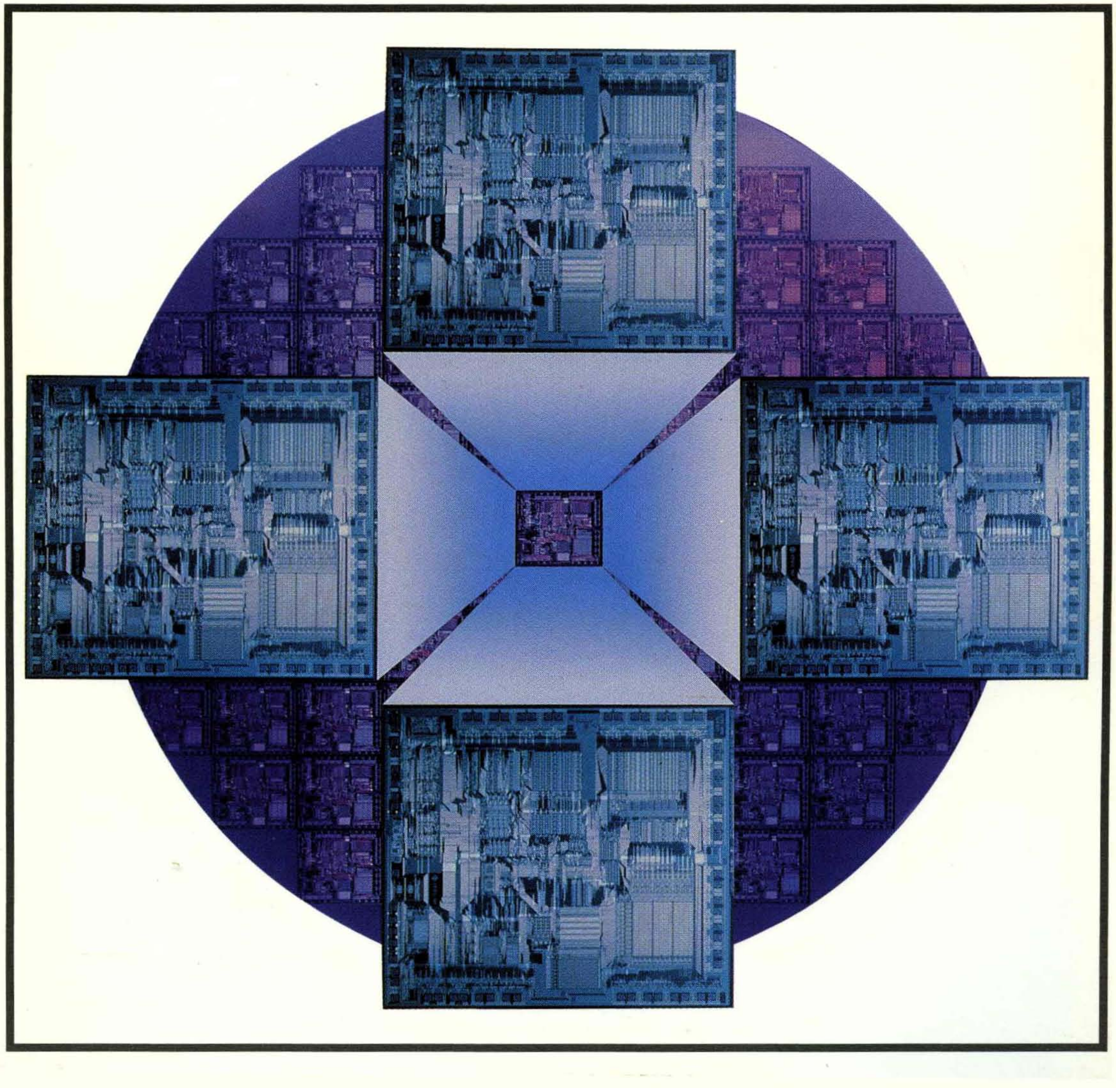
SIEMENS

8-Bit Single-Chip Microcontroller Handbook

1990/91



SIEMENS

8-Bit Single-Chip
Microcontroller Handbook

1990/91

SIEMENS

8-Bit Single-Chip Microcontroller Handbook

1990/91

Published by Siemens Components Inc., Integrated Circuit Division
2191 Laurelwood Rd., Santa Clara, CA 95054

For the circuits, descriptions, and tables indicated no responsibility is assumed as far as patents or other rights of third parties are concerned.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery, and prices please contact the Offices of Siemens Components.

Table of Contents
Summary of Types

1

Table of Contents

Page

Summary of Types

Type	Package	Description/Speed
8-Bit Single-Chip Microcontrollers		
SAB 8031A-N	PLCC 44	without ROM, 12 MHz
SAB 8031A-P	P-DIP 40	without ROM, 12 MHz
SAB 8031A-16-N	PLCC 44	without ROM, 16 MHz
SAB 8031A-16-P	P-DIP 40	without ROM, 16 MHz
SAB 8032B-N	PLCC 44	without ROM, 12 MHz
SAB 8032B-P	P-DIP 40	without ROM, 12 MHz
SAB 8032B-16-N	PLCC 44	without ROM, 16 MHz
SAB 8032B-16-P	P-DIP 40	without ROM, 16 MHz
SAB 8032B-20-N	PLCC 44	without ROM, 20 MHz
SAB 8032B-20-P	P-DIP 40	without ROM, 20 MHz
SAB 80C32-N	PLCC 44	without ROM, CMOS, 12 MHz
SAB 80C32-P	P-DIP 40	without ROM, CMOS, 12 MHz
SAB 80C32-16-N	PLCC 44	without ROM, CMOS, 16 MHz
SAB 80C32-16-P	P-DIP 40	without ROM, CMOS, 16 MHz
SAB 8051A-N	PLCC 44	4K x 8-bit, ROM, 12 MHz
SAB 8051A-P	P-DIP 40	4K x 8-bit, ROM, 12 MHz
SAB 8051A-16-N	PLCC 44	4K x 8-bit, ROM, 16 MHz
SAB 8051A-16-P	P-DIP 40	4K x 8-bit, ROM, 16 MHz
SAB 8052B-N	PLCC 44	4K x 8-bit, ROM, 12 MHz
SAB 8052B-P	P-DIP 40	8K x 8-bit, ROM, 12 MHz
SAB 8052B-16-N	PLCC 44	8K x 8-bit, ROM, 16 MHz
SAB 8052B-16-P	P-DIP 40	8K x 8-bit, ROM, 16 MHz
SAB 80C52-N	PLCC 44	8K x 8-bit, ROM, CMOS, 12 MHz
SAB 80C52-P	P-DIP 40	8K x 8-bit, ROM, CMOS, 12 MHz
SAB 80C52-16-N	PLCC 44	8K x 8-bit, ROM, CMOS, 16 MHz
SAB 80C32-16-P	P-DIP 40	8K x 8-bit, ROM, CMOS, 16 MHz
SAB 80512-N	PLCC 68	4K x 8-bit, ROM, 12 MHz
SAB 80513-N	PLCC 44	16K x 8-bit, ROM, 12 MHz
SAB 80513-P	P-DIP 40	16K x 8-bit, ROM, 12 MHz
SAB 80513-16-N	PLCC 44	16K x 8-bit, ROM, 16 MHz
SAB 80513-16-P	P-DIP 40	16K x 8-bit, ROM, 16 MHz
SAB 8352-5P	P-DIP 40	32K x 8-bit, ROM, 12 MHz
SAB 8352-5N	PLCC 44	32K x 8-bit, ROM, 12 MHz
SAB 8352-5P-16	P-DIP 40	32K x 8-bit, ROM, 16 MHz
SAB 8352-5N-16	PLCC 44	32K x 8-bit, ROM, 16 MHz
SAB 80515-N	PLCC 68	8K x 8-bit, ROM, 12 MHz
SAB 83515-4N	PLCC 68	16K x 8-bit, ROM, 12 MHz
SAB 80C515-N	PLCC 68	8K x 8-bit, ROM, CMOS, 12 MHz
SAB 80C515-16-N	PLCC 68	8K x 8-bit, ROM, 16 MHz
SAB 80C517-N	PLCC 84	8K x 8-bit, ROM, CMOS, 12 MHz
SAB 80532-N	PLCC 68	without ROM, 12 MHz
SAB 80533-N	PLCC 44	without ROM, 12 MHz
SAB 80533-P	P-DIP 40	without ROM, 12 MHz
SAB 80535-N	PLCC 68	without ROM, 12 MHz
SAB 80C535-N	PLCC 68	without ROM, CMOS, 12 MHz
SAB 80C535-16-N	PLCC 68	without ROM, 16 MHz
SAB 80C537-N	PLCC 84	without ROM, CMOS, 12 MHz

Summary of Types (Continued)

Type	Package	Description/Speed
8-Bit Single-Chip Microcontrollers Extended Temperature Range		
SAB 8031A-10-P-T40/110	P-DIP 40	−40°C to +110°C
SAB 8031A-12-P-T40/85	P-DIP 40	−40°C to +85°C
SAB 80C32-N-T40/85	PLCC 44	−40°C to +85°C, CMOS
SAB 80C32-P-T40/85	P-DIP 40	−40°C to +85°C, CMOS
SAB 80C32-N-T40/110	PLCC 44	−40°C to +110°C, CMOS
SAB 80C32-P-T40/110	P-DIP 40	−40°C to +110°C, CMOS
SAB 80C32-16-N-T40/85	PLCC 44	−40°C to +85°C, CMOS, 16 MHz
SAB 80C32-16-P-T40/85	P-DIP 40	−40°C to +85°C, CMOS, 16 MHz
SAB 8032B-N-T40/85	PLCC 44	−40°C to +85°C
SAB 8032B-P-T40/85	P-DIP 40	−40°C to +85°C
SAB 8032B-P-T40/100	P-DIP 40	−40°C to +100°C
SAB 8051A-10-P-T40/110	P-DIP 40	−40°C to +110°C
SAB 8051A-12-P-T40/85	P-DIP 40	−40°C to +85°C
SAB 80C52-N-T40/85	PLCC 44	−40°C to +85°C, CMOS
SAB 80C52-P-T40/85	P-DIP 40	−40°C to +85°C, CMOS
SAB 80C52-N-T40/110	PLCC 44	−40°C to +110°C, CMOS
SAB 80C52-P-T40/110	P-DIP 40	−40°C to +110°C, CMOS
SAB 80C52-16-N-T40/85	PLCC 44	−40°C to +85°C, CMOS, 16 MHz
SAB 80C52-16-P-T40/85	P-DIP 40	−40°C to +85°C, CMOS, 16 MHz
SAB 8052B-N-T40/85	PLCC 44	−40°C to +85°C
SAB 8052B-P-T40/85	P-DIP 40	−40°C to +85°C
SAB 8052B-P-T40/100	P-DIP 40	−40°C to +100°C
SAB 80513-P-T40/85	P-DIP 40	−40°C to +85°C
SAB 80513-N-T40/85	PLCC 44	−40°C to +85°C
SAB 80513-16-P-T40/85	P-DIP 40	−40°C to +85°C, 16 MHz
SAB 80513-16-N-T40/85	PLCC 44	−40°C to +85°C, 16 MHz
SAB 8352-5P-T40/85	P-DIP 40	−40°C to +85°C
SAB 8352-5N-T40/85	PLCC 44	−40°C to +85°C
SAB 8352-5P-16-T40/85	P-DIP 40	−40°C to +85°C, 16 MHz
SAB 8352-5N-16-T40/85	PLCC 44	−40°C to +85°C, 16 MHz
SAB 80512-N-T40/85	PLCC 68	−40°C to +85°C
SAB 80515-N-T40/85	PLCC 68	−40°C to +85°C
SAB 80515-N-T40/110	PLCC 68	−40°C to +110°C
SAB 80C515-N-T40/85	PLCC 68	−40°C to +85°C, CMOS
SAB 80C515-N-16-T40/85	PLCC 68	−40°C to +85°C, CMOS, 16 MHz
SAB 80C517-N-T40/85	PLCC 84	−40°C to +85°C, CMOS
SAB 80532-N-T40/85	PLCC 68	−40°C to +85°C
SAB 80535-N-T40/85	PLCC 68	−40°C to +85°C
SAB 80535-N-T40/110	PLCC 68	−40°C to +110°C
SAB 80C535-N-T40/85	PLCC 68	−40°C to +85°C, CMOS
SAB 80C535-N-16-T40/85	PLCC 68	−40°C to +85°C, CMOS, 16 MHz
SAB 80C537-N-T40/85	PLCC 84	−40°C to +85°C, CMOS

General Information

General Information

Type Designation Code of ICs

IC type designations are based on the Eurpoean Pro Electron System. The code system is explained in the Pro Electron brochure D 15, edition 1985, available at:

Pro Electron, Avenue Louise, 430 (B.12)
B-1060 Brussels, Belgium

Mounting Instructions

Plastic Package (Dual In-Line)

The 90° pins fit into holes with a diameter of 0.7mm to 0.9mm, spaced 2.54mm apart. See spacing x in Figure 1.

The bottom of the package will not touch the PC board after insertion because the pins have shoulders just below the package (see Figure 1).

After insertion of the package into the PC board it is advisable to bend the ends of two pins at an angle of approx. 30° to the board so that the package does not have to be pressed down during soldering. Plastic packages are soldered on that side of the PCB facing away from the package.

The maximum permissible soldering temperature is 350°C (max. 3s) for hand soldering and 260°C (max. 10s) for dip soldering and wave soldering.

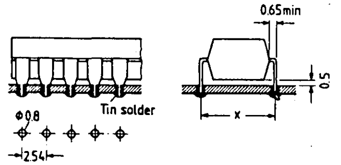


Figure 1. Plastic Package (Dual-In-Line)

Plastic Packages (SO and PLCC) for Surface Mounting (SMD)

Iron Soldering:	Soldering temperature 350°C for max. 3s; minimum distance between package and soldering point 1.5mm package temperature max. 150°C; no mechanical stress on the pins
Vapor phase soldering:	Soldering temperature 215°C, max. soldering time 40s
Wave soldering: (pins and package are dipped into the tin bath)	Soldering temperature 260°C, max. soldering time 8s

Storage, Pretreatment before Processing

The components are to be stored in a dry environment. When solder methods causing solder heat shock stresses are used (reflow soldering where the component is dipped into the solder bath, vapor-phase soldering) it is recommendable to subject IC's in plastic packages to a 24-hour drying phase at 125°C.

Other Points to Note

Ensure that no current is able to flow between the solder bath or soldering iron and the PCB. It is advisable to ground the pins that are to be soldered as well as the solder bath or soldering iron.

When the pins are being prepared and inserted in a PCB, circuits should be protected against static charge. Under no circumstances should the components be removed or inserted while the operating voltage is switched on.

The increase in chip temperature during the soldering process results in a temporary increase in electrostatic sensitivity of integrated circuits. Special precautions should therefore be taken against line transients, e.g. through the switching of inductances on magnetic chutes, etc.

Processing Guidelines for ICs

Integrated circuits (ICs) are **e**lectro**s**tatic-**s**ensitive (**ESS**) devices. The requirement for greater packing density has led to increasingly small structures on semiconductor chips with the result that today every IC, whether bipolar, MOS, or CMOS, has to be protected against electrostatics.

MOS and CMOS devices generally have integrated protective circuits and it is hardly possible any more for them to be destroyed by purely static electricity. On the other hand, there is acute danger from **e**lectro**s**tatic **d**ischarge (**ESD**).

Of the multiple of possible sources of discharge, charged devices should be mentioned in additon to charged persons. With low-resistive discharges it is possible for peak power amounting to kilowatts to be produced.

For the protection of devices the following principles should be observed:

a. Reduction of charging voltage, below 200V if possible. Means which are effective here are an increase in relative humidity to ≥ 60% and the replacement of highly charging plastics by antistatic materials.

b. With every kind of contact with the device pins a charge equalization is to be expected. This should always be highly resistive (ideally $R = 10^6\Omega$ to $10^8\Omega$).

All in all this means that ICs call for special handling, because uncontrolled charges, voltages from ungrounded equipment or persons, surge voltage spikes and similar influences can destroy a device. Even if devices have protective circuits (e.g. protective diodes) on their inputs, the following guidelines for their handling should nevertheless be observed.

Identification

The packing of ESS devices is provided with the following label by the manufacturer:

0165-3

Scope

The guidelines apply to the storage, transport, testing, and processing of all kinds of ICs, as well as the soldered circuit boards equipped with such components.

Handling of Devices

1. ICs must be left in their containers until they are processed.
2. ICs may only be handled at specially equipped work stations. These stations must have work surfaces covered with a conductive material of the order of 10^6 Ω/cm. to 10^9 Ω/cm.
3. With humidity of >50% a coat of pure cotton is sufficient. In the case of chargeable synthetic fibers the clothing should be worn close-fitting. The wrist strap must be worn snugly on the skin and be grounded through a resistor of 50 kΩ to 100 kΩ.
4. If conductive floors, $R = 5 \times 10^4\Omega$ to $10^7\Omega$ are provided, further protection can be achieved by using so-called MOS chairs and shoes with a conductive sole ($R \approx 10^5\Omega$ to $10^7\Omega$).
5. All transport containers for ESS devices and assembled circuit boards must first be brought to the same potential by being placed on the work surface or touched by the operator before the individual devices may be handled. The potential equalization should be through a resistor of $10^6\Omega$ to $10^8\Omega$.
6. When loading machines and production devices it should be noted that the devices come out of the transport magazine charged and can be damaged if they touch metal, e.g. machine parts.

   Example 1. Conductive (black) tubes.

   The devices may be destroyed in the tube by charged persons or come out of the tube charged if this is emptied by a charged person. Conductive tubes may only be handled at ESS work stations (high-resistance workstation and person grounding).

   Example 2. Anti-static (transparent) tubes.

   The devices cannot be destroyed by charged persons in the tube (there may be a rare exception in the case of custom ICs with unprotected gate pins). The devices can be endangered as in 1) when the tube is emptied if the latter, especially at low humidity, is no longer sufficiently antistatic after a long period of storage (> 1 year).

In both cases damage can be avoided by discharging the devices through a grounded adapter of high-resistance material ($\approx 10^6\ \Omega$/cm to $10^8\ \Omega$/cm) between the tube and the machine.

The use of metal tubes—especially of anodized aluminum—is not advisable because of the danger of low-resistance device discharge.

Storage

ESS devices should only be stored in identified locations provided for the purpose. During storage the devices should remain in the package in which they are supplied. The storage temperature should not exceed 60°C.

Transport

ESS devices in approved packing tubes should only be transported in suitable containers of conductive or longterm anti-static-treated plastic or possible unvarnished wood. Containers of high-charging plastic or very low-resistance materials are likewise unsuitable.

Transfer cars and their rollers should exhibit adequate electrical conductivity ($R < 10^6 \Omega$). Sliding contacts and grounding chains will not reliably eliminate charges.

Incoming Inspection

In incoming inspection the above guidelines should be observed. Otherwise any right for refund or replacement if devices fail inspection may be lost.

Material and Mounting

1. The drive belts of machines used for the processing of the devices, in as much as they come into contact with them (e.g. bending and cutting machines, conveyor belts), should be treated with anti-static spray (e.g. anti-static spray 100 from Kontaktchemie). It is better, however, to avoid the contact completely.
2. If ESS devices have to be soldered or desoldered manually, soldering irons with thyristor control cannot be used. Siemens EMI-suppression capacitors of the type B 81711-B31–B36 have proven very effective against line transients.
3. Circuit boards fitted and soldered with ESS devices are always to be considered as endangered.

Electrical Tests

1. The devices should be processed with observation of these guidelines. Before assembled and soldered circuit boards are tested, remove any shorting ring.
2. Test sockets must not be conducting any voltage when individual devices or assembled circuit boards are inserted or withdrawn, unless works' specifications state otherwise. Ensure that the test devices do not produce any voltage spikes, either when being turned on and off in normal operation or if the power fuse blows or other fuses respond.
3. Signal voltages may only be applied to the inputs of ICs when or after the supply voltage is turned on. They must be disconnected before or when the supply voltage is turned off.
4. Observe any notes and instructions in the respective data books/sheets.

Packing of Assembled PC Boards or Flatpack Units

The packing material should exhibit low volume conductivity:

$10^5\ \Omega/\text{cm} < \rho < 10^{10}\ \Omega/\text{cm}$.

In most cases—especially with humidity of > 40%—this requirement is fulfilled using simple corrugated board. Better protection is obtained with bags of conductive polyethylene foam (e.g. RCAS 1200 from Richmond of Redlands, California).

It must always be ensured that boards do not touch.

In special cases it may be necessary to provide protection against strong electric fields, such as can be generated by conveyor belts for example. For this purpose a sheath of aluminum foil is recommended, although direct contact between the film and the PCB must be avoided. Cardboard boxes with an aluminum-foil lining, such as those used for shipping of our devices, are available from Laber of Munich.

Ultrasonic Cleaning of ICs

The following recommendation applies to plastic packages. For cavity packages (metal and also ceramic) separate regulations have to be observed.

Freon and isopropyl alcohol (trade name: propanol) can be used as solvents. These solvents can also be used for plastic packages because they do not eat into the plastic material.

An ultrasonic bath in double halfwave operation is advisable because of the low component stress.

The ultrasonic limits are as follows:

sound frequency	$f > 40$ kHz
exposure	$t < 2$ min
alternating sound pressure	$p < 0.29$ bar
sound power	$N < 0.5$ W/cm^2/liter

Data Classification

Maximum Ratings

Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.

Characteristics

The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread. If not otherwise specified, typical characteristics will apply at $T_A = 25°C$ and for the given supply voltage.

Operating Range

In the operating range the functions given in the circuit description wil be fulfilled.

Quality Assurance System

The high quality and reliability of integrated circuits from Siemens is the result of a carefully arranged production which is systematically checked and controlled at each production stage.

The procedures are subject to a quality assurance system; full details are given in the brochure "Siemens Quality Assurance—Integrated Circuits' (SQS-IC).

Figure 2 shows the most important stages of the "SQS-IC". A quality assurance (QA) department which is independent of production and development, is responsible for the selected control measures, acceptance procedures, and information feedback loops. This department has state-of-the-art test and measuring equipment at its disposal, works according to approved methods of statistical quality control, and is provided with facilities for accelerated life and environmental tests used for both qualification and routine monitoring test.

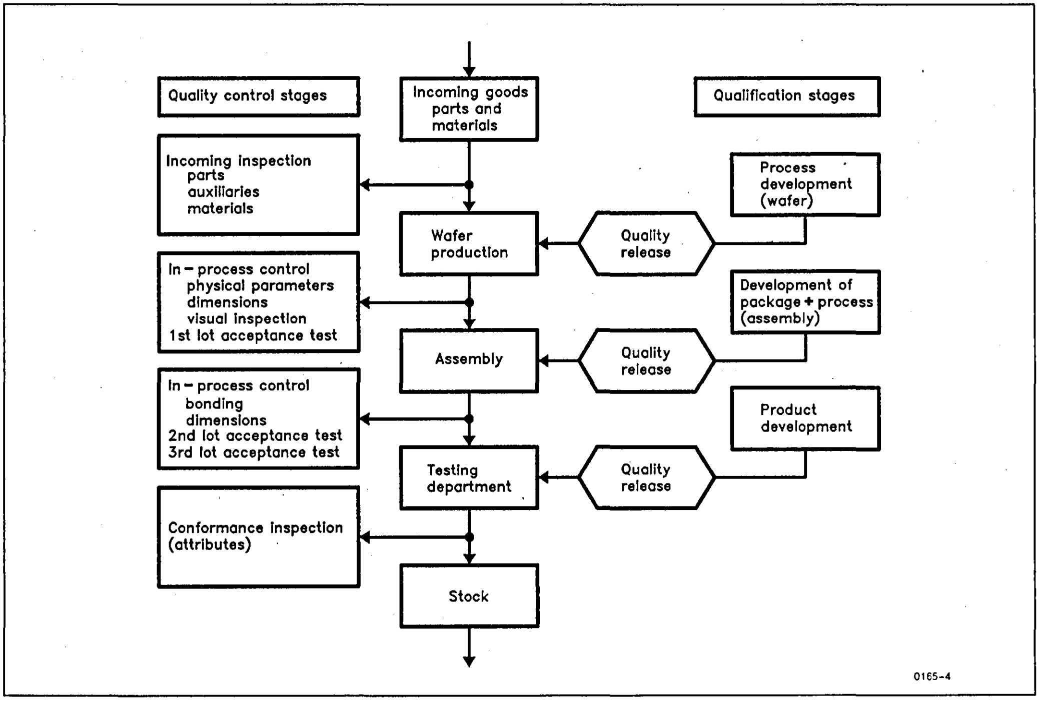

Quality control stages
Incoming goods parts and materials
Qualification stages
Incoming inspection parts auxiliaries materials
Process development (wafer)
Wafer production
Quality release
In-process control physical parameters dimensions visual inspection 1st lot acceptance test
Development of package + process (assembly)
Assembly
Quality release
In-process control bonding dimensions 2nd lot acceptance test 3rd lot acceptance test
Product development
Testing department
Quality release
Conformance inspection (attributes)
Stock
0165-4

Figure 2. Quality Assurance System

The latest methods and equipment for preparation and analysis are employed to achieve continuity of quality and reliability.

Conformance

Each integrated circuit is subjected to a final test at the end of the production process. These tests are carried out by computer-controlled, automatic test systems because hundreds of thousands of operating conditions as well as a large number of static and dynamic parameters have to be considered. Moreover, the test systems are extremely reliable and reproducible. The quality assurance department carries out a final check in the form of a lot-by-lot sampling inspection to additionally ensure this minimum percent defectives as well as the acceptable quality level (AQL). Sampling inspection is performed in accordance with the inspection plans of DIN 40080, as well as of the identical MIL-STD-105 or IEC 410.

Reliability

Measures Taken during Development

The reliability of ICs is already considerably influenced at the development stage. Siemens has, therefore, fixed certain design standards for the development of circuit and layout, specifying e.g. minimum width and spacing of conductive layers on a chip, dimensions and electrical parameters of protective circuits for electrostatic charge, etc. An examination with the aid of carefully arranged programs operated on large-scale computers, guarantees the immediate identification and elimination of unintentional violations of these design standards.

In-Process Control during Production

The manufacturing of integrated circuits comprises several hundred production steps. As each step is to be executed with utmost accuracy, the in-process control is of outstanding importance. Some processes require more than a hundred different test measures. The tests have been arranged such that the individual process steps can be reproduced continuously.

The decreasing failure rates reflect the never ending effort in this direction; they have been reduced considerably despite an immense rise in the IC's complexity.

Reliability Monitoring

The general course of the IC's failure rate versus time is shown by a so-called "bathtub" curve (Figure 3). The failure rate has its peak during the first few operating hours (early failure period). After the early failure period has decayed, the "constant" failure rate period starts during which the failures may occur at an approximately uniform rate. This period ends with a repeated rise of the curve during the wear-out failure period. For ICs, however, the latter period usually lies far beyond the service life specified for the individual equipment.

Reliability tests for ICs are usually destructive examinations. They are, therefore, carried out with samples. Most failure mechanisms can be accelerated by means of higher temperatures. Due to the temperature dependence of the failure mechanisms, it is possible to simulate future operational behavior within a short time by applying high temperatures; this is called accelerated life testing.

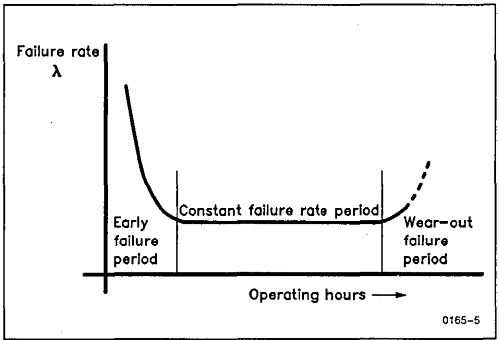


Figure 3. Reliability "Bath-Tub" Curve

The accerleration factor B for the life test can be obtained from the Arrhenius equation

$$B = \exp \quad \frac{E_A}{k}\left[\frac{1}{T_1} - \frac{1}{T_2}\right]$$

where T_2 is the temperature at which the life test is performed, T_1 is the assumed operating temperature, and k is the Boltzmann constant.

Important for factor B is the activation energy E_A. It lies between 0.3V and 1.3 eV and differs considerably for individual failure mechanisms.

For all Siemens ICs, the reliablity data from life tests is converted to an operating temperature of T_A = 55°C, assuming an average activation energy of 0.5 eV. The acceleration factor for life tests at 125°C is thus 22.3, compared with operational behavior. This method considers also failure mechanisms with low activation energy, i.e. which are only slightly accelerated by the temperature effect.

Various reliability tests are periodically performed with IC types that are representative of a certain production line—this is described in the brochure "SQS-IC". Such tests are e.g. humidity test at 85°C and 85% relative humidity, pressure cooker test, as well as life tests up to 1000 hours and more. Test results are available in the form of summary reports.

SAB-51 Architectural Overview

SAB-51 Architectural Overview

Siemens SAB-51 family of 8-bit microcontrollers consists of the devices listed in Table 1, all of which are based on the SAB-8051 architecture shown in Figure 1. The original 8051 was built in N-channel, silicon gate, Siemens MYMOS II technology and packaged in a 40-pin DIP. The 8051A which is in the advanced N-channel, silicon gate Siemens MYMOS III process, is the device currently in production.

All other microcontrollers listed in this book are backward compatible with the SAB-8051A.

SAB 8051 Architectural Block Diagram

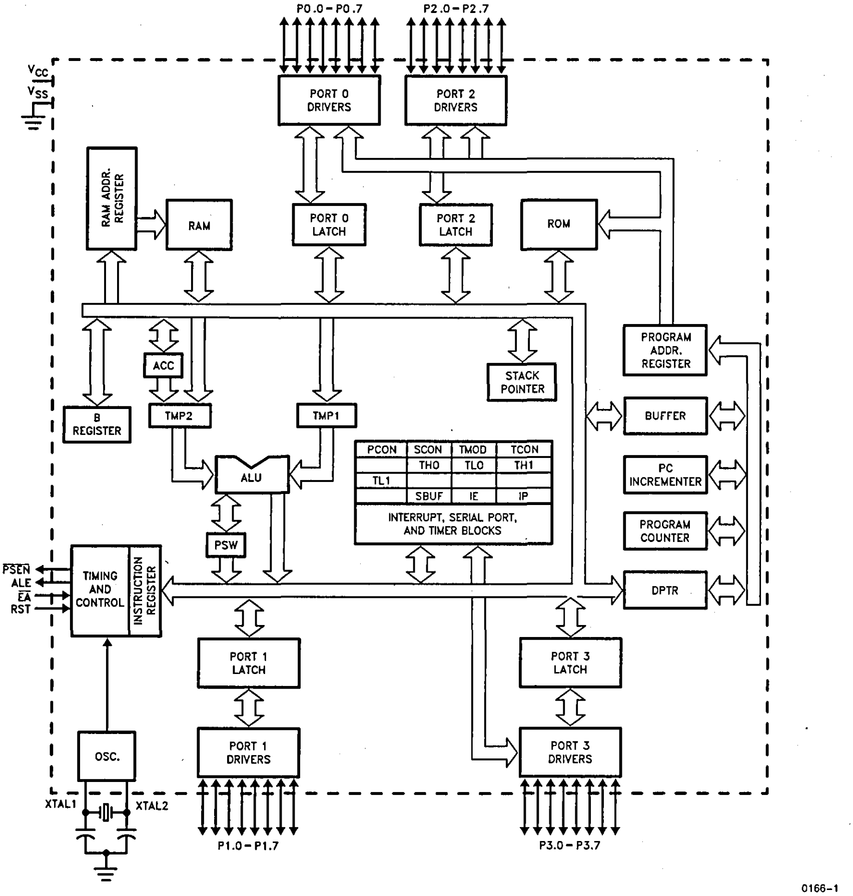


Figure 1. SAB 8051 Architectural Block Diagram

Table 1. The SAB-51 Family of Microcontrollers

Device * = ROMless Version	Clock Rate (MHz)	ROM (KB)	RAM (Byte)	I/O-Ports (8-Bit)	ADC Inputs (8-Bit Resol.)	Timer/ Counter (16-Bit)	Watch Dog Timer	Inter-rupt Sources/ Levels	Serial I/O	PWM	Div/ Mult Unit	Data Pointers (16-Bit)	Package
SAB 8051A SAB 8031A*	12, 16	4 —	128	4	—	2	—	5/2	USART	—	—	1	PDIP 40 PLCC 44
SAB 8052B SAB 8032B*	12, 16 20	8 —	256	4	—	3	—	6/2	USART	—	—	1	PDIP 40 PLCC 44
SAB 80512 SAB 80532*	12	4 —	128	6(I/O) + 1(I)	8	2	—	6/2	USART	—	—	1	PLCC 68
SAB 80513 SAB 8352-5	12, 16	16 32	256	4	—	3	—	6/2	USART	—	—	1	PDIP 40 PLCC 44
SAB 80515 SAB 80535* SAB 83515-4	12	8 — 16	256	6	8	3	1	12/4	USART	4-ch	—	1	PLCC 68
SAB 80C515 SAB 80C535*	12, 16	8 —	256	6(I/O) + 1(I)	8	3	1	12/4	USART	4-ch	—	1	PLCC 68
SAB 80C517 SAB 80C537*	12	8 —	256	7(I/O) + 1½ (I)	12	4	2	14/4	USART + UART	21-ch	Yes	8	PLCC 84

SAB 8051A/8031A, SAB 8051A-16/8031A-16

The SAB8051A is the original member of the SAB-51 family. Among the features of the SAB 8051A are:

- 8-Bit CPU Optimized for Control Applications
- Extensive Boolean Processing (Single-Bit Logic) Capabilities
- 32 Bidirectional and Individually Addressable I/O Lines
- 128 Bytes of On-Chip Data RAM
- RAM Power-Down Supply
- Two 16-Bit Timer/Counters
- Full Duplex UART
- 5-Source Interrupt Structure with 2 Priority Levels
- On-Chip Clock Oscillator
- 4 Kbytes of On-Chip Program Memory
- 64K Program Memory Address Space
- 64K Data Memory Address Space

The SAB 8051A/8031A is a standalone, high-performance single-chip microcontroller fabricated in +5V advanced Siemens MYMOS (III) technology and supplied in a 40-pin plastic P-DIP or 44-pin plastic leaded chip carrier (PL-CC-44) package.

The SAB8031A differs from the SAB8051A, in not having the on-chip program ROM. Instead, the SAB 8031A fetches all instructions from external memory.

The SAB8051A-16 differs from the SAB8051A only in the speed of operation. The SAB8051A can run with a clock oscillator frequency from 1.2 MHz to 12 MHz whereas the SAB 8051A-16 can run upto a clock oscillator frequency of 16 MHz.

The extended temperature versions of these parts are also available.

SAB 8052B/8032B, SAB 8052B-16/SAB 8032B-16, SAB 8032B-20

The SAB 8052B/8032B is identical to the SAB 8051A/8031A and is also fabricated in advanced N-channel, Silicon gate Siemens MYMOS III technology. It is pin for pin compatible with the SAB8051A. Its enhancements over the SAB8051A are as follows:

- 256 Bytes of On-Chip Data RAM
- Three Timer/Counters
- 6-Source Interrupt Structure
- 8 Kbytes of On-Chip Program ROM

The ROMless version of the 8052B is the 8032B. The 16 MHz version is the SAB 8052B-16/8032B-16. The ROMless version is also available in 20 MHz speed called the SAB 8032B-20. The extended temperature versions of these parts are also available. The earlier version of the SAB 8052B/8032B was fabricated in N-channel, silicon gate Siemens MYMOS II technology and was called the SAB 8052A/8032A. Whatever applies to the SAB 8052A/8032A also applies to the SAB 8052B/8032B except that the SAB 8052B/8032B is in the advanced N-channel, silicon gate Siemens MYMOS III technology.

SAB 80C52/80C32 SAB 80C52-16/SAB 80C32-16

The SAB 80C52/80C32 is a standalone, high-performance CMOS single-chip microcontroller, designed in Siemens ACMOS technology. It is functionally compatible with the SAB 8052A/8032A or the SAB 8052B/8032B devices in MYMOS technology. The ROMless version of the 80C52 is the 80C32. The 16 MHz version is the SAB 80C52-16/80C32-16.

Furthermore, it is backwardly compatible with the SAB 80C51/80C31. The low-power consumption properties of ACMOS technology allow applications where power consumption and dissipation are critical. In addition, the SAB 80C52/80C32 has two software-selectable modes of reduced activity for further power reduction—idle and power-down.

The SAB 80C52/80C32 is supplied in a 40-pin P-DIP package, or a 44-pin plastic leaded chip carrier (PLCC 44) package. The extended temperature versions of these parts are also available.

SAB 80513, SAB 8352-5
SAB 80513-16, SAB 8352-5-16

The SAB 80513 and SAB 8352-5 are new members of the Siemens SAB 8051 family of 8-bit microcontrollers. They are fabricated in N-channel silicon-gate Siemens MYMOS technology.

The SAB 80513 and SAB 8352-5 are stand-alone, high-performance, single-chip microcontrollers based on the SAB 8051 architecture. They maintain all features of the SAB 8051A and SAB 8052B (including Timer 2 of the SAB 8052B) and are thus fully compatible to both the SAB 8051A and SAB 8052B.

In addition, the SAB 80513 contains 16 Kbytes and SAB 8352-5 32 Kbytes of on-chip ROM, which make them powerful and cost-effective controllers for applications requiring more ROM space.

The SAB 80513 and SAB 8352-5 operate up to 12 MHz crystal oscillator frequency. The SAB 80513-16 and SAB 8352-5-16 operate up to 16 MHz crystal oscillator frequency.

Both microcontrollers are supplied in a 40-pin dual-in-line package or a 44-pin plastic leaded chip carrier (PLCC 44) package.

SAB 80512/80532

The SAB 80512/80532 is a new member of the Siemens SAB 8051 family of 8-bit microcontrollers. Maintaining all features of the SAB 8051A/8031A, it is backwardly compatible with the SAB 8051A/8031A. Furthermore the SAB 80512/80532 incorporates several enhancements, that significantly increase design flexibility and cost effectiveness. In addition to the SAB 8051A/8031A the SAB 80512/80532 contains an 8-bit A/D converter with 8 multiplexed inputs (these inputs can also be used as digital inputs), an own baud rate generator for the serial interface and two more I/O ports. The SAB 80532 is identical with the SAB 80512, except that it lacks the on-chip ROM.

The SAB 80512/80532 is fabricated in +5V advanced N-channel, silicon gate MYMOS technology of Siemens and supplied in a PLCC 68 package. For the industrial temperature range −40°C to +85°C, the SAB 80512/80532-T40/85 is available.

SAB 80515/80535
SAB 83515-4

The SAB 80515/80535 is a powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is fabricated in +5V N-channel, silicon-gate Siemens MYMOS technology.

The SAB 80515/80535 is a stand-alone, high-performance single-chip microcontroller based on the SAB 8051 architecture. While maintaining all the SAB 8051 operating characteristics, the SAB 80515/80535 incorporates several enhancements which significantly increase design flexibility and overall system performance. These features are:

- 8K × 8 ROM (SAB 80515 Only)
- 16K x 8 ROM (SAB 83515-4 Only)
- 256 × 8 RAM
- Six 8-Bit I/O Ports, One 8-Bit Input Port for Analog Signals
- Three 16-Bit Timer/Counters
- Programmable Timer/Counter Register-Array with Compare/Capture
  Auto Reload
  Pulse Width Modulator Capabilities
- Full-Duplex Serial Channel
- Twelve Interrupt Vectors, Four Priority Levels
- 8-Bit A/D Converter with Eight Multiplexed Inputs and Programmable Internal Reference Voltages
- 16-Bit Watchdog Timer
- V_{PD} Provides Standby Current for 40 Bytes of RAM
- Boolean Processor
- 256 Bit-Addressable Locations
- Most Instructions Executed in:
  1 µs (SAB 80515/80535)
- 4 µs Multiply and Divide
- Backwardly Compatible with SAB 8051
- 68-Pin Plastic Leaded Chip Carrier Package (PLCC 68)

The SAB 80535 is identical with the SAB 80515 except that it lacks the on-chip program memory. On the other hand, the SAB 83515-4 is also identical with the SAB 80515 and it contains 16K of on-chip ROM as opposed to 8K in 80515. The SAB 80515/80535 is supplied in a 68-pin plastic leaded chip carrier package (PLCC 68). The SAB 80515/80535 operates up to 12 MHz crystal oscillator frequency. The SAB 80515/80535 is also available in industrial temperature range (−40°C to +85°C) and in the automotive temperature range (−40°C to +110°C).

SAB 80C515/80C535, SAB 80C515-16/80C535-16

The SAB 80C515/80C535 is a new, powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is designed in Siemens ACMOS technology and is functionally compatible with the SAB 80515/80535 devices designed in MYMOS technology.

In addition, the SAB 80C515 has two software-selectable power saving modes: idle mode and the power-down mode. These modes replace the power-down supply mode via pin V_{PD} of the SAB 80515 (NMOS).

In case of the SAB 80C515 the analog port can also be used as a digital input port.

The SAB 80C535 is identical with the SAB 80C515 except that it lacks the on-chip program memory. The SAB 80C515-16/80C535-16 operates up to 16 MHz crystal oscillator frequency. The SAB 80C515/80C535 is supplied in a 68-pin plastic leaded chip carrier package (PLCC 68). For the industrial temperature range −40°C to +85°C, the SAB 80C515/80C535-T40/85 and SAB 80C515/80C535-T40/85-16 are also available.

SAB 80C517/80C537

The SAB80C517/80C537 is a new and the most powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is designed in Siemens ACMOS technology and is functionally compatible with the SAB 80C515/80C535 devices. While maintaining all the SAB 80C51 operating characteristics, the SAB 80C517/80C537 incorporates several enhancements which significantly increase design flexibility and overall system performance. These features are:

- 8 Kbyte On-Chip Program Memory
- 256 Byte On-Chip RAM
- 256 Directly Addressable Bits
- 1 μs Instruction Cycle at 12 MHz
- 64 of 111 Instructions Executed in One Cycle
- External Program and Data Memory Expandable to 64 Kbyte Each
- 8-Bit A/D Converter
  - — 12 Multiplexed Inputs
  - — Programmable Reference Voltages
  - — External/Internal Start of Conversion
- Two General Purpose 16-Bit Timers/Counters (Timer 0, Timer 1)
- Compare Capture Unit (CCU)
  - — One 16-Bit Timer/Counter, 1 MHz Clock
  - — One 16-Bit Compare-Timer, 6 MHz Clock with Dedicated Reload Register
  - — One 16-Bit Reload/Capture/Compare Register
  - — Four 16-Bit Capture/Compare Registers
  - — Eight 16-Bit Compare Registers
  - — Concurrent Compare
  - — Pulse Width Modulation or High Speed Output Possible on up to 21 Channels
  - — Fine Capture Input Channels
- Two Full Duplex Serial Interfaces with Own Baud Rate Generator
- Four Priority Level Interrupt System, 14 Interrupt Sources
- Extended Arithmetic Capabilities for Division and Multiplication (Mul./Div. Unit Operations to Fast 16/32-Bit)
- Eight Datapointers for Indirect Addressing
- Extended Fail Safe Mechanisms
  - — 16-Bit Programmable Watchdog Timer
  - — Oscillator Watchdog
  - — Hardware Disable for Power Saving Modes
- Extended Power Saving Modes (Slow Down, Idle, Power-Down)
- Nine Ports
  - — Seven Bidirectional 8-Bit Ports
  - — One 8-Bit, One 4-Bit Input Port
- 84 Pin PLCC Package

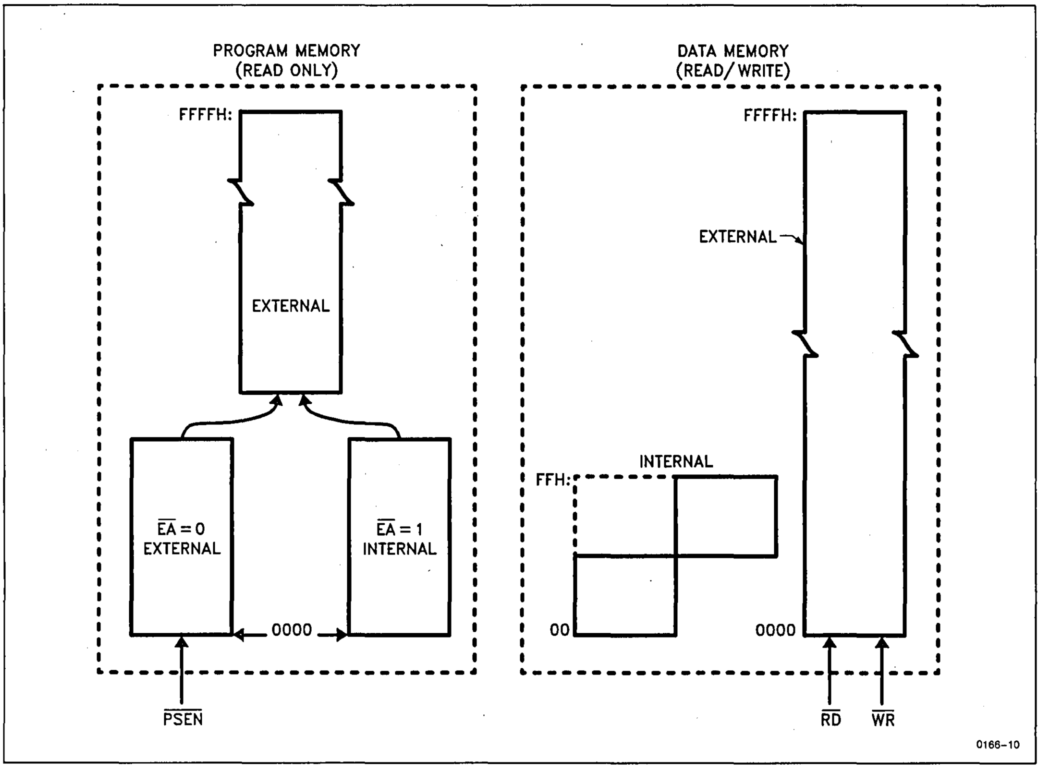


Figure 2. SAB-51 Memory Structure

Memory Organization in SAB-51 Devices

Logical Separation of Program and Data Memory

All SAB-51 devices have separate address spaces for Program and Data Memory, as shown in Figure 2. The logical separation of Program and Data Memory allows the Data Memory to be accessed by 8-bit addresses, which can be more quickly stored and manipulated by an 8-bit CPU. Nevertheless, 16-bit Data Memory addresses can also be generated through the DPTR register.

Program Memory can only be read, not written to. There can be up to 64 Kbytes of Program Memory. In the 8051A, 80C51 and 80512 the lowest 4 Kbytes of Program Memory are on-chip. The 8052B, 80515, 80C515 and 80C517 provide 8 Kbytes of on-chip Program Memory storage. The SAB 80513 has 16K of on-chip program memory.

In the ROMless versions all Program Memory is external. The read strobe for external Program Memory is the signal $\overline{\text{PSEN}}$ (Program Store Enable).

Data Memory occupies a separate address space from Program Memory. Up to 64 Kbytes of external RAM can be addressed in the external Data Memory space. The CPU generates read and write signals, $\overline{\text{RD}}$ and $\overline{\text{WR}}$, as needed during external Data Memory accesses.

External Program Memory and external Data Memory may be combined if desired by applying the $\overline{\text{RD}}$ and $\overline{\text{PSEN}}$ signals to the inputs of an AND gate and using the output of the gate as the read strobe to the external Program/Data memory.

Program Memory

Figure 3 shows a map of the lower part of the Program Memory. After reset, the CPU begins execution from location 0000H.

As shown in Figure 3, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

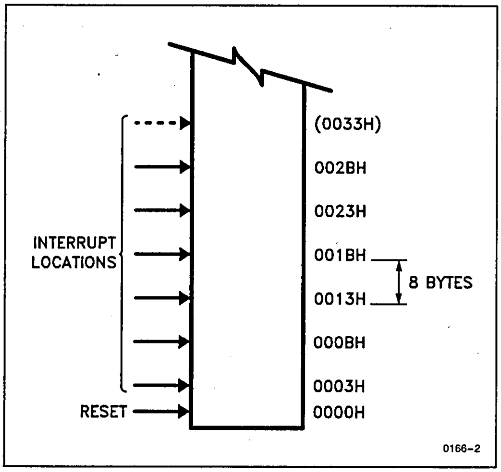


Figure 3. SAB-51 Program Memory

The interrupt service locations are spaced at 8-byte intervals: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

The lowest 4K (or 8K, in the 8052B, 80515 and 80C517) bytes of Program Memory can be either in the on-chip ROM or in an external ROM. This selection is made by strapping the $\overline{EA}$ (External Access) pin to either V_{CC} or V_{SS}.

In 8051A, 80C51 and 80512, if the $\overline{EA}$ pin is strapped to V_{CC}, then program fetches to addresses 0000H through 0FFFH are directed to the internal ROM. Program fetches to addresses 1000H through FFFFH are directed to external ROM.

In the 8052B and the other 8K ROM parts, $\overline{EA}$ = V_{CC} selects addresses 0000H through 1FFFH to be internal, and addresses 2000H through FFFFH to be external.

In the 80513, $\overline{EA}$ = V_{CC} selects addresses 0000H through 3FFFH to be internal and addresses 4000H through FFFH to be external.

If the $\overline{EA}$ pin is strapped to V_{SS}, then all program fetches are directed to external ROM. The ROMless parts 8031A, 8032B, 80532 etc. must have this pin externally strapped to V_{SS} to enable them to execute from external Program Memory.

The read strobe to external ROM, $\overline{PSEN}$, is used for all external program fetches. $\overline{PSEN}$ is not activated for internal program fetches.

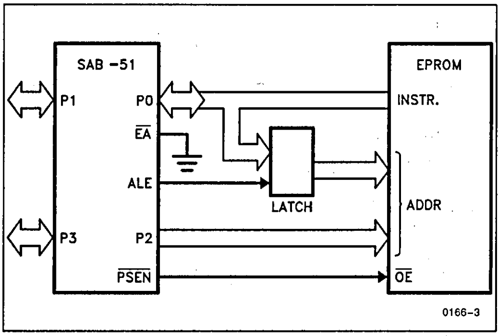


Figure 4. Executing from External Program Memory

The hardware configuration for external program execution is shown in Figure 4. Note that 16 I/O lines (Ports 0 and 2) are dedicated to bus functions during external Program Memory fetches. Port 0 (P0 in Figure 4) serves as a multiplexed address/data bus. It emits the low byte of the Program Counter (PCL) as an address, and then goes into a float state awaiting the arrival of the code byte from the Program Memory. During the time that the low byte of the Program Counter is valid on P0, the signal ALE (Address Latch Enable) clocks this byte into an address latch. Meanwhile, Port 2 (P2 in Figure 4) emits the high byte of the Program Counter (PCH). The $\overline{PSEN}$ strobes the EPROM and the code byte is read into the microcontroller.

Program Memory addresses are always 16 bits wide, even though the actual amount of Program Memory used may be less than 64 Kbytes. External program execution sacrifices two of the 8-bit ports, P0 and P2, to the function of addressing the Program Memory.

Data Memory

The right half of Figure 2 shows the internal and external Data Memory spaces available to the SAB-51 user.

Figure 5 shows a hardware configuration for accessing up to 2 Kbytes of external RAM. The CPU in this case is executing from internal ROM. Port 0 serves as a multiplexed address/data bus to the RAM and 3 lines of Port 2 are being used to page the RAM. The CPU generates $\overline{RD}$ and $\overline{WR}$ signals as needed during external RAM accesses.

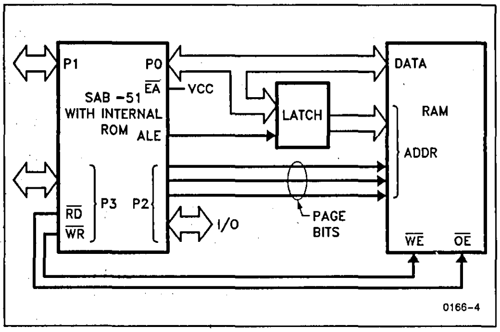


Figure 5. Accessing External Data Memory If the Program Memory is Internal, the Other Bits of P2 are Available as I/O.

There can be up to 64 Kbytes of external Data Memory. External Data Memory addresses can be either 1 or 2 bytes wide. One-byte addresses are often used in conjunction with one or more other I/O lines to page the RAM, as shown in Figure 5. Two-byte addresses can also be used, in which case the high address byte is emitted at Port 2.

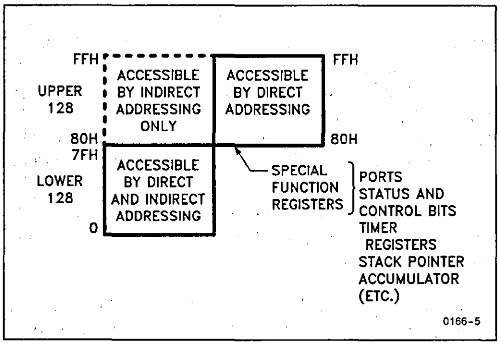

FFH
UPPER 128
80H
7FH
LOWER 128
0
ACCESSIBLE BY INDIRECT ADDRESSING ONLY
ACCESSIBLE BY DIRECT ADDRESSING
FFH
80H
ACCESSIBLE BY DIRECT AND INDIRECT ADDRESSING
SPECIAL FUNCTION REGISTERS
PORTS
STATUS AND CONTROL BITS
TIMER
REGISTERS
STACK POINTER
ACCUMULATOR
(ETC.)
0166-5

Figure 6. Internal Data Memory

Internal Data Memory is mapped in Figure 6. The memory space is shown divided into three blocks, which are generally referred to as the Lower 128, the Upper 128 and SFR space.

Internal Data Memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH access a different memory space. Thus Figure 6 shows the Upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

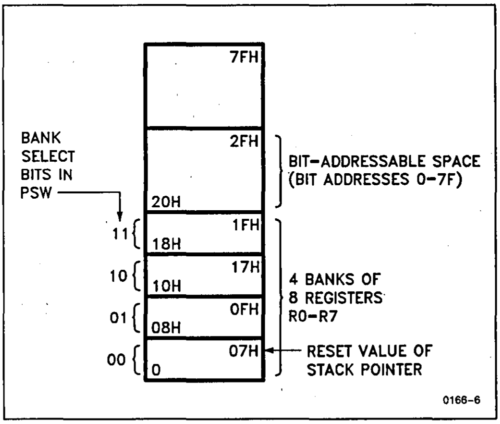

7FH
BANK SELECT BITS IN PSW
2FH
20H
BIT-ADDRESSABLE SPACE (BIT ADDRESSES 0-7F)
11
1FH
18H
10
17H
10H
01
0FH
08H
00
07H
0
4 BANKS OF 8 REGISTERS R0-R7
RESET VALUE OF STACK POINTER
0166-6

Figure 7. The Lower 128 Bytes of Internal RAM

The Lower 128 bytes of RAM are present in all SAB-51 devices as mapped in Figure 7. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

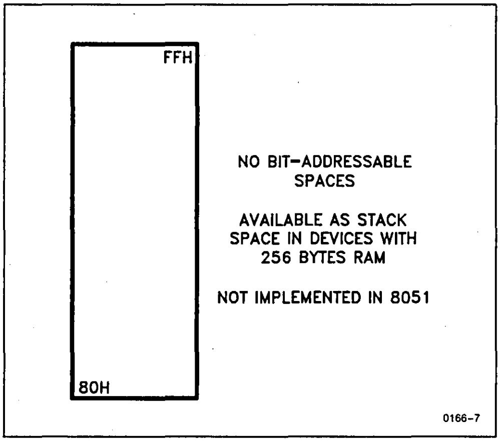

FFH
NO BIT-ADDRESSABLE SPACES
AVAILABLE AS STACK SPACE IN DEVICES WITH 256 BYTES RAM
NOT IMPLEMENTED IN 8051
80H
0166-7

Figure 8. The Upper 128 Bytes of Internal RAM

The next 16 bytes above the register banks form a block of bit-addressable memory space. The SAB-51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing. The Upper 128 (Figure 8) can only be accessed by indirect addressing. The Upper 128 bytes of RAM are not implemented in the 8051A, but are in the 8052B, 80515 and 80C517.

Figure 9 gives a brief look at the Special Function Register (SFR) space. SFR include the Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. In general, all SAB-51 microcontrollers have the same SFRs as the 8051. However, enhancements to the 8051 have additional SFRs that are not present in the 8051, nor perhaps in other proliferations of the family.

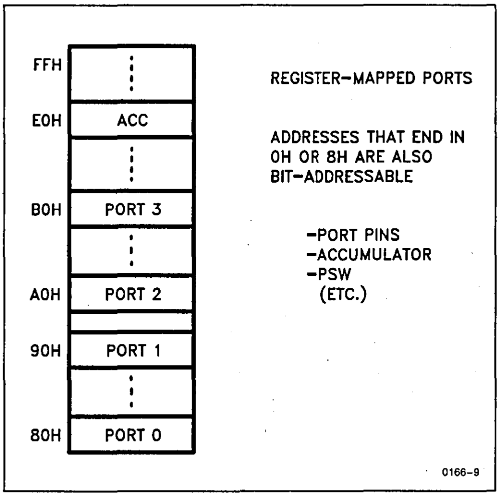


Figure 9. SFR Space

The bit addresses in this area are 80H through FFH.

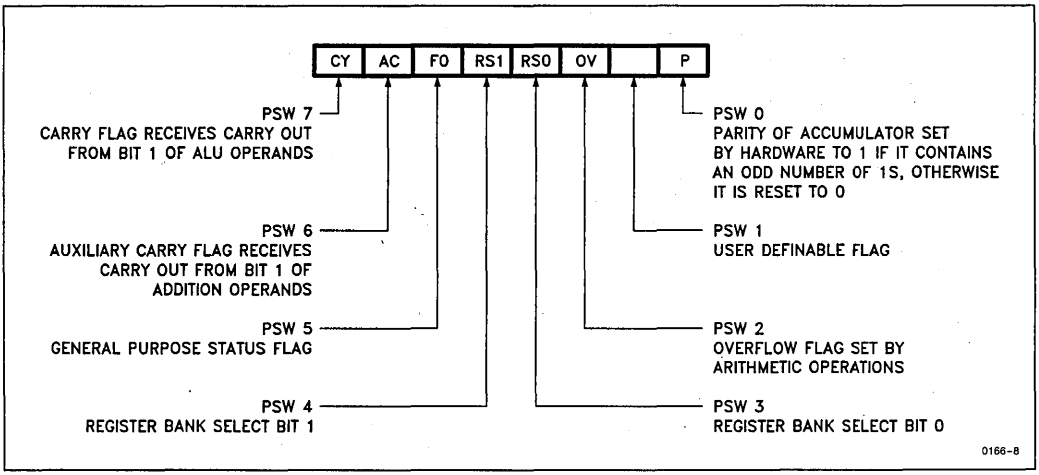

CY
AC
F0
RS1
RS0
OV
P
PSW 7
CARRY FLAG RECEIVES CARRY OUT FROM BIT 1 OF ALU OPERANDS
PSW 6
AUXILIARY CARRY FLAG RECEIVES CARRY OUT FROM BIT 1 OF ADDITION OPERANDS
PSW 5
GENERAL PURPOSE STATUS FLAG
PSW 4
REGISTER BANK SELECT BIT 1
PSW 0
PARITY OF ACCUMULATOR SET BY HARDWARE TO 1 IF IT CONTAINS AN ODD NUMBER OF 1S, OTHERWISE IT IS RESET TO 0
PSW 1
USER DEFINABLE FLAG
PSW 2
OVERFLOW FLAG SET BY ARITHMETIC OPERATIONS
PSW 3
REGISTER BANK SELECT BIT 0
0166-8

Figure 10. PSW (Program Status Word) Register SAB-51 Devices

The SAB-51 Instruction Set

All members of the SAB-51 family execute the same instruction set. SAB-51 instruction set is optimized for 8-bit control applications. It provides a variety of fast addressing modes for accessing the internal RAM to facilitate byte operations on small data structures. The instruction set provides extensive support for one-bit variables as a separate data type, allowing direct bit manipulation in control and logic systems that require Boolean processing.

An overview of the SAB-51 instruction set is presented below, with a brief description of how certain instructions might be used. Refer to the chapter on instruction set in this book for detailed information on the instructions.

Program Status Word

The Program Status Word (PSW) contains several status bits that reflect the current state of the CPU. The PSW, shown in Figure 10, resides in SFR space. It contains the Carry bit, the Auxiliary Carry (for BCD operations), the two register bank select bits, the Overflow flag, a Parity bit, and two user-definable status flags.

The Carry bit, other than serving the functions of a Carry bit in arithmetic operations, also serves as the "Accumulator" for a number of Boolean operations.

The bits RS0 and RS1 are used to select one of the four register banks shown in Figure 7. A number of instructions refer to these RAM locations as R0 through R7. The selection of which of the four banks is being referred to is made on the basis of the bits RS0 and RS1 at execution time.

The Parity bit reflects the number of 1s in the Accumulator: P = 1 if the Accumulator contains an odd number of 1s, and P = 0 if the Accumulator contains an even number of 1s. Thus the number of 1s in the Accumulator plus P is always even.

Two bits in the PSW are uncommitted and may be used as general purpose status flags.

Addressing Modes

The addressing modes in the SAB-51 instruction set are as follows:

Direct Addressing

In direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal Data RAM and SFRs can be directly addressed.

Indirect Addressing

In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed.

The address register for 8-bit addresses can be R0 or R1 of the selected register bank or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit "data pointer" register, DPTR.

Register Instructions

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specifcation within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of the eight registers in the selected bank is accessed. One of four banks is selected at execution time by the two bank select bits in the PSW.

Register-Specific Instructions

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point to it. The opcode itself does that. Instructions that refer to the Accumulator as A assemble as accumulator-specific opcodes.

Immediate Constants

The value of a constant can follow the opcode in Program Memory. For example,

```
MOV A, #100
```

loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.

Indexed Addressing

Only Program Memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program Memory. A 16-bit base register (either DPTR or the Program Counter) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program Memory is formed by adding the Accumulator data to the base pointer.

Another type of indexed addressing is used in the "case jump" instruction. In this case the destination address of a jump instruction is computed as the sum of the base pointer and the Accumulator data.

Arithmetic Instructions

With the on-chip arithmetic unit the SAB 80C517 has a special provision for fast multiplication and division. Refer to the SAB 80C517 User's Manual for detailed information on this logic. However, the following information applies to all the members of the SAB-51 family.

The menu of arithmetic instructions is listed in Table 2. The table indicates the addressing modes that can be used with each instruction to access the <byte> operand. For example, the ADD A, <byte> instruction can be written as:

```
ADD   A,7FH    (direct addressing)
ADD   A,@R0    (indirect addressing)
ADD   A,R7     (register addressing)
ADD   A,#127   (immediate constant)
```

The execution times listed in Table 2 assume a 12 MHz clock frequency. All of the arithmetic instructions execute in 1 μs except the INC DPTR instruction, which takes 2 μs, and the Multiply and Divide instructions, which take 4 μs.

Note that any byte in the internal Data Memory space can be incremented or decremented without going through the Accumulator.

One of the INC instructions operates on the 16-bit Data Pointer. The Data Pointer is used to generate 16-bit addresses for external memory, so being able to increment it in one 16-bit operation is a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

3

Table 2. A List of the SAB-51 Arithmetic Instructions

Mnemonic		Operation	Addressing Modes				Execution Time (μs)
			Dir	Ind	Reg	Imm	
ADD	A, <byte>	A = A + <byte>	X	X	X	X	1
ADDC	A, <byte>	A = A + <byte> + C	X	X	X	X	1
SUBB	A, <byte>	A = A − <byte> − C	X	X	X	X	1
INC	A	A = A + 1	Accumulator Only				1
INC	<byte>	<byte> = <byte> + 1	X	X	X		1
INC	DPTR	DPTR = DPTR + 1	Data Pointer Only				2
DEC	A	A = A − 1	Accumulator Only				1
DEC	<byte>	<byte> = <byte> − 1	X	X	X		1
MUL	AB	B:A = B × A	ACC and B Only				4
DIV	AB	A = Int [A/B] B = Mod [A/B]	ACC and B Only				4
DA	A	Decimal Adjust	Accumulator Only				1

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator, and the 8-bit remainder in the B register.

Oddly enough, DIV AB finds less use in arithmetic "divide" routines than in radix conversions and programmable shift operations. An example of the use of DIV AB in a radix conversion will be given later. In shift operations, dividing a number by 2^n shifts its n bits to the right. Using DIV AB to perform the division completes the shift in 4 μs and leaves the B register holding the bits that were shifted out.

The DA A instruction is for BCD arithmetic operations. In BCD arithmetic, ADD and ADDC instructions should always be followed by a DA A operation, to ensure that the result is also in BCD. Note that DA A will not convert a binary number to BCD. The DA A operation produces a meaningful result only as the second step in the addition of two BCD bytes.

Logical Instructions

Table 3 shows the list of SAB-51 logical instructions. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes perform the operation on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and <byte> contains 01010011B, then

ANL A,<byte>

will leave the Accumulator holding 00010001B.

The addressing modes that can be used to access the <byte> operand are listed in Table 3. Thus, the ANL A,<byte> instruction may take any of the forms:

ANL	A,7FH	(direct addressing)
ANL	A,@R1	(indirect addressing)
ANL	A,R6	(register addressing)
ANL	A,#53H	(immediate constant)

Table 3. A List of the SAB-51 Logical Instructions

Mnemonic		Operation	Addressing Modes				Execution Time (μs)
			Dir	Ind	Reg	Imm	
ANL	A, <byte>	A = A.AND. <byte>	X	X	X	X	1
ANL	<byte>, A	<byte> = <byte> .AND.A	X				1
ANL	<byte>, #data	<byte> = <byte> .AND.#data	X				2
ORL	A,<byte>	A = A.OR. <byte>	X	X	X	X	1
ORL	<byte>,A	<byte> = <byte> .OR.A	X				1
ORL	<byte>,#data	<byte> = <byte>.OR.#data	X				2
XRL	A,<byte>	A = A.XOR.<byte>	X	X	X	X	1
XRL	<byte>, A	<byte> = <byte> .XOR.A	X				1
XRL	<byte>, #data	<byte> = <byte> .XOR.#data	X				2
CRL	A	A = 00H	Accumulator Only				1
CPL	A	A = .NOT.A	Accumulator Only				1
RL	A	Rotate ACC Left 1 Bit	Accumulator Only				1
RLC	A	Rotate Left through Carry	Accumulator Only				1
RR	A	Rotate ACC Right 1 Bit	Accumulator Only				1
RRC	A	Rotate Right through Carry	Accumulator Only				1
SWAP	A	Swap Nibbles in A	Accumulator Only				1

All of the logical instructions that are Accumulator-specific execute in 1 μs (using a 12 MHz clock). The others take 2 μs.

Note that Boolean operations can be performed on any byte in the internal Data Memory space without going through the Accumulator. The XRL <byte>, #data instruction, for example, offers a quick and easy way to invert port bits, as in:

```
XRL   R1,#0FFH
```

If the operation is in response to an interrupt, not using the Accumulator saves the time and effort to stack it in the service routine.

The Rotate instructions (RL A, RLC A, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the LSB rolls into the MSB position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. This is a useful operation in BCD manipulations. For example, if the Accumulator contains a binary number which is known to be less than 100, it can be quickly converted to BCD by the following code.

```
MOV    B, #10
DIV    AB
SWAP   A
ADD    A, B
```

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The SWAP and ADD instructions move the tens digit to the high nibble of the Accumulator and the ones digit to the low nibble.

Data Transfers

Internal RAM

Table 4 shows the menu of instructions that are available for moving data around within the internal memory spaces, and the addressing modes that can be used with each one. With a 12 MHz clock, all of these instructions execute in either 1 μs or 2 μs.

The MOV <dest>, <src> instruction allows data to be transferred between any two internal RAM or SFR locations without going through the Accumulator. Remember the Upper 128 bytes of data RAM can be accessed only by indirect addressing and SFR space only by direct addressing.

Note that in all SAB-51 devices, the stack resides in on-chip RAM, and grows upwards. The PUSH instruction first increments the Stack Pointer (SP), then copies the byte into the stack. PUSH and POP use only direct addressing to identify the byte being saved or restored, but the stack itself is accessed by indirect addressing using the SP register. This means the stack can go into the Upper 128, if they are implemented, but not into SFR space.

Table 4. A List of the SAB-51 Data Transfer Instructions that Access Internal Data Memory Space

Mnemonic		Operation	Addressing Modes				Execution Time (μs)
			Dir	Ind	Reg	Imm	
MOV	A, <src>	A = <src>	X	X	X	X	1
MOV	<dest>, A	<dest> = A	X	X	X		1
MOV	<dest>, <src>	<dest> = <src>	X	X	X	X	2
MOV	DPTR, #data16	DPTR = 16-Bit Immediate Constant				X	2
PUSH	<src>	INC SP: MOV "@SP", <src>	X				2
POP	<dest>	MOV <dest>, "@SP": DEC SP	X				2
XCH	A, <byte>	ACC and <byte> Exchange Data	X	X	X		1
XCHD	A, @Ri	ACC and @Ri Exchange Low Nibbles		X			1

The Upper 128 Bytes are not implemented in the 8051A, 80C51 and 80512, nor in their ROMless counterparts. With these devices, if the SP points to the Upper 128, PUSHed bytes are lost, and POPped bytes are indeterminate.

The Data Transfer instructions include a 16-bit MOV that can be used to initialize the Data Pointer (DPTR) for look-up tables in Program Memory, or for 16-bit external Data Memory accesses. Refer to the SAB 80C517 Data Sheet for extended Data Pointer Capabilities.

The XCH A, <byte> instruction causes the Accumulator and addressed byte to exchange data. The XCHD A,@Ri instruction is similar, but only the low nibbles are involved in the exchange.

To see how XCH and XCHD can be used to facilitate data manipulations, consider first the problem of shifting an 8-digit BCD number two digits to the right. Figure 11 shows how this can be done using direct MOVs, and for comparison how it can be done using XCH instructions. To aid in understanding how the code works, the contents of the registers that are holding the BCD number and the content of the Accumulator are shown alongside each instruction to indicate their status after the instruction has been executed.

		2A	2B	2C	2D	2E	ACC
MOV	A, 2EH	00	12	34	56	78	78
MOV	2EH, 2DH	00	12	34	56	56	78
MOV	2DH, 2CH	00	12	34	34	56	78
MOV	2CH, 2BH	00	12	12	34	56	78
MOV	2BH, #0	00	00	12	34	56	78

(a) Using Direct MOVs: 14 Bytes, 9 µs

		2A	2B	2C	2D	2E	ACC
CLR	A	00	12	34	56	78	00
XCH	A, 2BH	00	00	34	56	78	12
XCH	A, 2CH	00	00	12	56	78	34
XCH	A, 2DH	00	00	12	34	78	56
XCH	A, 2EH	00	00	12	34	56	78

(b) Using XCH$_{S1}$ 9 Bytes, 5 µs

Figure 11. Shifting a BCD Number Two Digits to the Right

After the routine has been executed, the Accumulator contains the two digits that were shifted out on the right. Doing the routine with direct MOVs uses 14 code bytes and 9 µs of execution time (assuming a 12 MHz clock). The same operation with XCHs uses less code and executes almost twice as fast.

To right-shift by an odd number of digits, a one-digit shift must be executed. Figure 12 shows a sample of code that will right-shift a BCD number one digit, using the XCHD instruction. Again, the contents of the registers holding the number and of the Accumulator are shown alongside each instruction.

			2A	2B	2C	2D	2E	ACC
	MOV	R1, #2EH	00	12	34	56	78	XX
	MOV	R0, #2DH	00	12	34	56	78	XX
	Loop for R1 = 2EH:							
LOOP:	MOV	A, @R1	00	12	34	56	78	76
	XCHD	A, @R0	00	12	34	58	78	76
	SWAP	A	00	12	34	58	78	67
	MOV	@R1, A	00	12	34	58	67	67
	DEC	R1	00	12	34	58	67	67
	DEC	R0	00	12	34	58	67	67
	CJNE	R1, #2AH, LOOP						
	Loop for R1 = 2DH:		00	12	38	45	67	45
	Loop for R1 = 2CH:		00	18	23	45	67	23
	Loop for R1 = 2BH:		08	01	23	45	67	01
	CLR	A	08	01	23	45	67	00
	XCH	A, 2AH	00	01	23	45	67	08

Figure 12. Shifting a BCD Number One Digit to the Right

First, pointers R1 and R0 are set up to point to the two bytes containing the last four BCD digits. Then a loop is executed which leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH. The CJNE instruction (Compare and Jump if Not Equal) is a loop control that will be described later.

The loop is executed from LOOP to CJNE for R1 2EH, 2DH, 2CH and 2BH. At that point the digit that was originally shifted out on the right has propagated to location 2AH. Since that location should be left with 0s, the lost digit is moved to the Accumulator.

External RAM

Table 5 shows a list of the Data Transfer instructions that access external Data Memory. Only indirect addressing can be used. The choice is whether to use a one-byte address, @Ri, where Ri can be either R0 or R1 of the selected register bank, or a two-byte address, @DPTR. The disadvantage to using 16-bit addresses if only a few Kbytes of external RAM are involved is that 16-bit addresses use all 8 bits of Port 2 as address bus. On the other hand, 8-bit addresses allow one to address a few Kbytes of RAM, as shown in Figure 5, without having to sacrifice all of Port 2.

All of thes instructions execute in 2 μs, with a 12 MHz clock.

Table 5. A List of the SAB-51 Data Transfer Instructions that Access External Data Memory Space

Address Width	Mnemonic	Operation	Execution Time (μs)
8 Bits	MOVX A, @Ri	Read External RAM @Ri	2
8 Bits	MOVX @Ri, A	Write External RAM @Ri	2
16 Bits	MOVX A, @DPTR	Read External RAM @DPTR	2
16 Bits	MOVX @DPTR, A	Write External RAM @DPTR	2

Note that in all external Data RAM accesses, the Accumulator is always either the destination or source of the data.

The read and write strobes to external RAM are activated only during the execution of a MOVX instruction. Normally these signals are inactive, and in fact if they're not going to be used at all, their pins are available as extra I/O lines. More about that later.

Lookup Tables

Table 6 shows the two instructions that are available for reading loopup tables in Program Memory. Since these instructions access only Program Memory, the lookup tables can only be read, not updated. The mnemonic is MOVC for "move constant".

If the table access is to external Program Memory, then the read strobe is $\overline{\text{PSEN}}$.

Table 6. The SAB-51 Lookup Table Read Instructions

Mnemonic	Operation	Execution Time (μs)
MOVC A, @A + DPTR	Read PGM Memory at (A + DPTR)	2
MOVC A, @A + PC	Read PGM Memory at (A + PC)	2

The first MOVC instruction in Table 6 can accommodate a table of up to 256 entries, numbered 0 through 255. The number of the desired entry is loaded into the Accumulator, and the Data Pointer is set up to point to beginning of the table. Then

```
MOVC   A, @A + DPTR
```

copies the desired table entry into the Accumulator.

The other MOVC instruction works the same way, except the Program Counter (PC) is used as the table base, and the table is accessed through a subroutine. First the number of the desired entry is loaded into the Accumulator, and the subroutine is called:

```
MOV    A, ENTRY_ NUMBER

CALL   TABLE
```

The subroutine "TABLE" would look like this:

```
TABLE: MOVC   A, @A + PC
       RET
```

The table itself immediately follows the RET (return) instruction in Program Memory. This type of table can have up to 255 entries, numbered 1 through 255. Number 0 can not be used, because at the time the MOVC instruction is executed, the PC contains the address of the RET instruction. An entry numbered 0 would be the RET opcode itself.

Boolean Instructions

SAB-51 devices contain a complete Boolean (single-bit) processor. The internal RAM contains 128 addressable bits, and the SFR space can support up to 128 other addressable bits. All of the port lines are bit-addressable, and each one can be treated as a separate single-bit port. The instructions that access these bits are not just conditional branches, but a complete menu of move, set, clear, complement, OR and AND instructions. These kinds of bit operations are not easily obtained in other architectures with any amount of byte-oriented software.

Table 7. A List of the SAB-51 Boolean Instructions

Mnemonic	Operation	Execution Time (μs)
ANL C,Bit	C = C.AND.Bit	2
ANL C,/Bit	C = C.AND..NOT.Bit	2
ORL C,Bit	C = C.OR.Bit	2
ORL C,/Bit	C = C.OR..NOT.Bit	2
MOV C,Bit	C = Bit	1
MOV Bit, C	Bit = C	2
CLR C	C = 0	1
CLR Bit	Bit = 0	1
SETB C	C = 1	1
SETB Bit	Bit = 1	1
CPL C	C = .NOT.C	1
CPL Bit	Bit = .NOT.Bit	1
JC Rel	Jump if C = 1	2
JNC Rel	Jump if C = 0	2
JB Bit,Rel	Jump if Bit = 1	2
JNB Bit,Rel	Jump if Bit = 0	2
JBC Bit,Rel	Jump if Bit = 1; CLR Bit	2

The instruction set for the Boolean processor is shown in Table 7. All bit accesses are by direct addressing. Bit addresses 00H through 7FH are in the Lower 128 and bit addresses 80H through FFH are in SFR space.

Note how easily an internal flag can be moved to a port pin:

```
MOV   C,FLAG
MOV   P1.0,C
```

In this example, FLAG is the name of any addressable bit in the Lower 128 or SFR space. An I/O line (the LSB of Port 1, in this case) is set or cleared depending on whether the flag bit is 1 or 0.

The Carry bit in the PSW is used as the single-bit Accumulator of the Boolean processor. Bit instructions that refer to the Carry bit as C assemble as Carry-specific instructions (CLR C, etc). The Carry bit also has a direct address, since it resides in the PSW register, which is bit-addressable.

Note that the Boolean instruction set includes ANL and ORL operations, but not the XRL (Exclusive OR) operation. An XRL operation is simple to implement in software. Suppose, for example, it is required to form the Exclusive OR of two bits:

C = Bit1 .XRL. bit 2

The software to do that could be as follows:

```
        MOV   C,bit1
        JNB   bit2,OVER
        CPL   C
OVER: (Continue)
```

First, bit1 is moved to the Carry. If bit2 = 0, then C now contains the correct result. That is, bit1 .XRL. bit 2 = bit1 if bit2 = 0. On the other hand, if bit2 = 1C now contains the complement of the correct result. It need only be inverted (CPL C) to complete the operation.

This code uses the JNB instruction, one of a series of bit-test instructions which execute a jump if the addressed bit is set (JC, JB, JBC) or if the addressed bit is not set (JNC, JNB). In the above case, bit2 is being tested, and if bit2 = 0 the CPL C instruction is jumped over.

JBC executes the jump if the addressed bit is set, and also clears the bit. Thus a flag can be tested and cleared in one operation.

All the PSW bits are directly addressable, so the Parity bit, or the general purpose flags, for example, are also available to the bit-test instructions.

Relative Offset

The destination address for these jumps is specified to the assembler by a label or by an actual address in Program Memory. However, the destination address assembles to a relative offset byte. This is a signed (two's complement) offset byte which is added to the PC in two's complement arithmetic if the jump is excuted.

The range of the jump is therefore −128 to +127 Program Memory bytes relative to the first byte following the instruction.

Jump Instructions

Table 8 shows the list of unconditional jumps.

Table 8. Unconditional Jumps in SAB-51 Devices

Mnemonic	Operation	Execution Time (μs)
JMP addr	Jump to addr	2
JMP @A + DPTR	Jump to A + DPTR	2
CALL Addr	Call Subroutine at addr	2
RET	Return from Subroutine	2
RETI	Return from Interrupt	2
NOP	No Operation	1

The Table lists a single "JMP addr" instruction, but in fact there are three— SJMP, LJMP and AJMP— which differ in the format of the destination address. JMP is a generic mnemonic which can be used if the programmer does not care which way the jump is encoded.

The SJMP instruction encodes the destination address as a relative offset, as described above. The instruction is 2 bytes long, consisting of the opcode and the relative offset byte. The jump distance is limited to a range of −128 to +127 bytes relative to the instruction following the SJMP.

The LJMP instruction encodes the destination address as a 16-bit constant. The instruction is 3 bytes long, consisting of the opcode and two address bytes. The destination address can be anywhere in the 64K Program Memory space.

The AJMP instruction encodes the destination address as an 11-bit constant. The instruction is 2 bytes long, consisting of the opcode, which itself contains 3 of the 11 address bits, followed by another byte containing the low 8 bits of the destination address. When the instruction is executed, these 11 bits are simply substitued for the low 11 bits in the PC. The high 5 bits stay the same. Hence the destination has to be within the same 2K block as the instruction following the AJMP.

In all cases the programmer specifies the destination address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the destination address into the correct format for the given instruction. If the format required by the instruction will not support the distance to the specified destination address, a "Destination out of range" message is written into the List file.

The JMP @A + DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically, DPTR is set up with the address of a jump table, and the Accumulator is given an index to the table. In a 5-way branch, for example, an integer 0 through 4 is loaded into the Accumulator. The code to be executed might be as follows:

```
MOV     DPTR,#JUMP_TABLE
MOV     A,INDEX_NUMBER
RL      A
JMP     @A + DPTR
```

The RL A instruction converts the index number (0 through 4) to an even number on the range 0 through 8, because each entry in the jump table is 2 bytes long:

```
JUMP_TABLE:
          AJMP   CASE_0
          AJMP   CASE_1
          AJMP   CASE_2
          AJMP   CASE_3
          AJMP   CASE_4
```

Table 8 shows a single "CALL addr" instruction, but there are two of them—LCALL and ACALL— which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic which can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64K Program Memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2K block as the instruction following the ACALL.

In any case the programmer specifies the subroutine address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the address into the correct format for the given instructions.

Subroutines should end with a RET instruction, which returns execution to the instruction following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is done. If there is no interrupt in progress at the time RETI is executed, then the RETI is functionally identical to RET.

Table 9. Conditional Jumps in SAB-51 Devices

Mnemonic		Operation	Addressing Modes				Execution Time (μs)
			Dir	Ind	Reg	Imm	
JZ	Rel	Jump if A = 0	Accumulator Only				2
JNZ	Rel	Jump if A ≠ 0	Accumulator Only				2
DJNZ	<byte>,rel	Decrement and Jump if Not Zero	X		X		2
CJNE	A, <byte>,rel	Jump if A ≠ <byte>	X			X	2
CJNE	<byte>, #data,rel	Jump if <byte> ≠ #data		X	X		2

Table 9 shows the list of conditional jumps available to the SAB-51 user. All of these jumps specify the destination address by the relative offset method; and so are limited to a jump distance of −128 to +127 bytes from the instruction following the conditional jump instruction. Important to note, however, the user specifies to the assembler the actual destination address the same way as the other jumps: as a label or a 16-bit constant.

There is no Zero bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10:

```
        MOV  COUNTER, #10
LOOP:   (begin loop)
        •
        •
        •
        (end loop)
        DJNZ COUNTER, LOOP
(Continue)
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in Figure 12. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Figure 12, the two bytes were the data in R1 and the constant 2AH. The initial data in R1 was 2EH. Every time the loop as executed, R1 was decremented, and the looping was to continue until the R1 data reached 2AH.

Another application of this instruction is in "greater than, less than" comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry bit is set (1). If the first is greater than or equal to the second, then the Carry bit is cleared.

CPU Timing

All SAB-51 microcontrollers have an on-chip oscillator which can be used if desired as the clock source for the CPU. To use the on-chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller and capacitors to ground as shown in Figure 13.

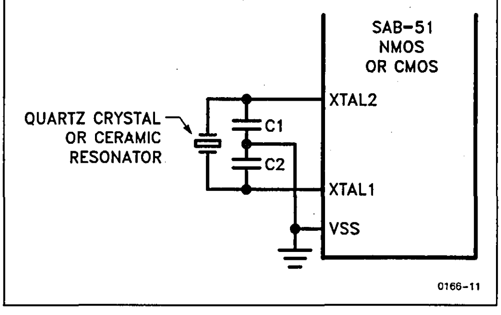


Figure 13. Using the On-Chip Oscillator

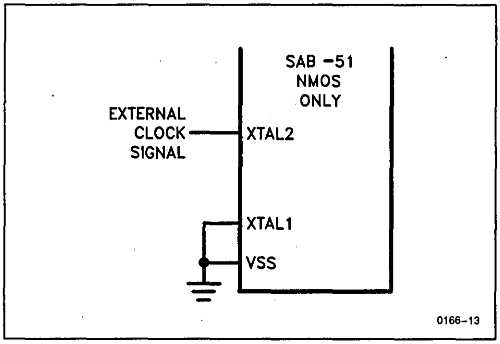


Figure 14. Using an External Clock

Example of how to drive the clock with an external oscillator is shown in Figure 14. Note that in the NMOS devices (8051, etc.) the signal at the XTAL2 pin actually drives the internal clock generator. In the CMOS devices 80C51, 80C52 the signal at the XTAL1 pin drives the internal clock generator, whereas in 80C515 and 80C517 the signal at the XTAL2 pin drives the internal clock generator. If only one pin is going to be driven with the external oscillator signal, make sure it is the right pin. For 80C515 and 80C517, refer to their data-sheets for the description of the clock oscillator pins.

The internal clock generator defines the sequence of states that make up the SAB-51 machine cycle.

Machine Cycles

A machine cycle consists of a sequence of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1 μs if the oscillator frequency is 12 MHz.

Each state is divided into a Phase 1 half and a Phase 2 half. Figure 15 shows the fetch/execute sequences in states and phases for various kinds of instructions. Normally two program fetches are generated during each machine cycle, even if the instruction being executed doesn't require it. If the instruction being executed doesn't need more code bytes, the CPU simply ignores the extra fetch, and the Program Counter is not incremented.

Execution of a one-cycle instruction (Figure 15A and B) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second fetch occurs during S4 of the same machine cycle. Execution is complete at the end of State 6 of this machine cycle.

The MOVX instructions take two machine cycles to execute. No program fetch is generated during the

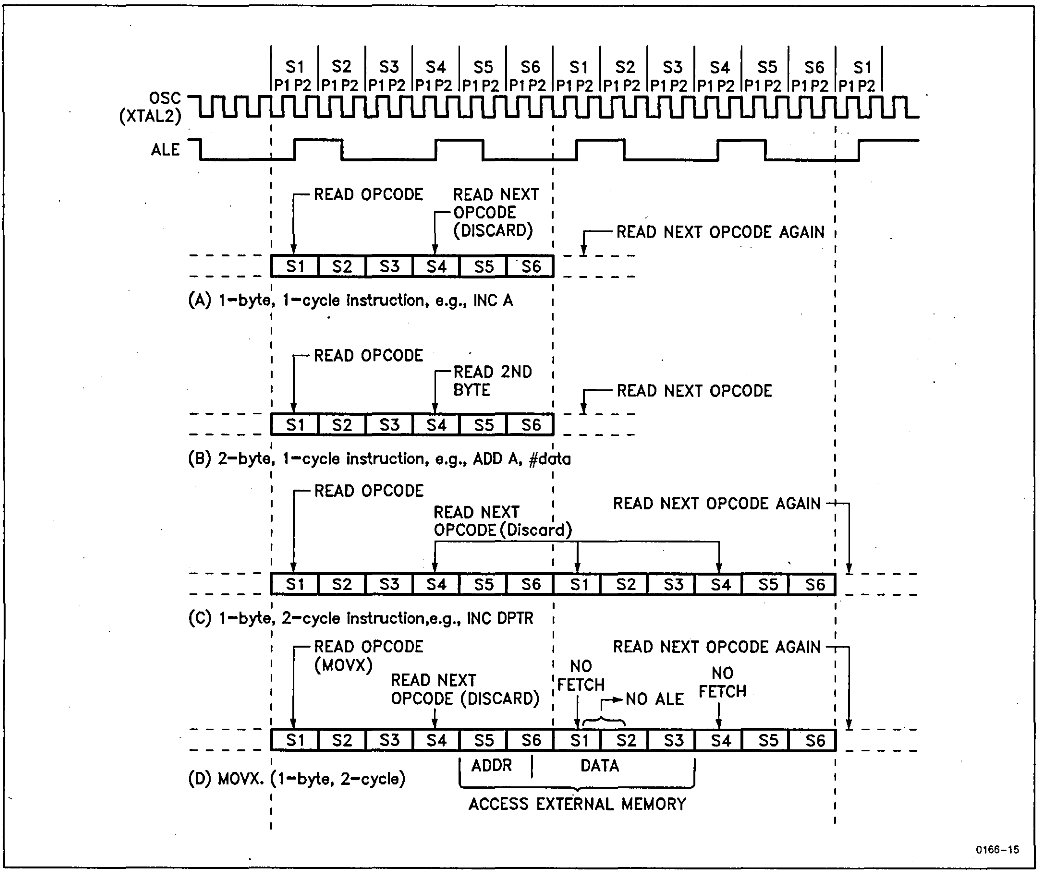

S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1
P1 P2
OSC (XTAL2)
ALE
READ OPCODE
READ NEXT OPCODE (DISCARD)
READ NEXT OPCODE AGAIN
(A) 1-byte, 1-cycle instruction, e.g., INC A
READ OPCODE
READ 2ND BYTE
READ NEXT OPCODE
(B) 2-byte, 1-cycle instruction, e.g., ADD A, #data
READ OPCODE
READ NEXT OPCODE (Discard)
READ NEXT OPCODE AGAIN
(C) 1-byte, 2-cycle instruction,e.g., INC DPTR
READ OPCODE (MOVX)
READ NEXT OPCODE (DISCARD)
NO FETCH
NO ALE
NO FETCH
READ NEXT OPCODE AGAIN
ADDR
DATA
ACCESS EXTERNAL MEMORY
(D) MOVX. (1-byte, 2-cycle)
0166-15

Figure 15. State Sequences in SAB-51 Devices

second cycle of a MOVX instruction. This is the only time program fetches are skipped. The fetch/execute sequence for MOVX instructions is shown in Figure 15(D).

The fetch/execute sequences are the same whether the Program Memory is internal or external to the chip. Execution times do not depend on whether the Program Memory is internal or external.

Figure 16 shows the signals and timing involved in program fetches when the Program Memory is external. If Program Memory is external, then the Program Memory read strobe $\overline{\text{PSEN}}$ is normally activated twice per machine cycle, as shown in Figure 16(A).

If an access to external Data Memory occurs, as shown in Figure 16(B), two $\overline{\text{PSEN}}$s are skipped, because the address and data bus are being used for the Data Memory access.

Note that a Data Memory bus cycle takes twice as much time as a Program Memory bus cycle. Figure 16 shows the relative timing of the addresses being emitted at Ports 0 and 2, and of ALE and $\overline{\text{PSEN}}$. ALE is used to latch the low address byte from P0 into the address latch.

When the CPU is executing from internal Program Memory, $\overline{\text{PSEN}}$ is not activated, and program addresses are not emitted. However, ALE continues to be activated twice per machine cycle and so is available as a clock output signal. Note, however, that one ALE is skipped during the execution of the MOVX instruction.

Interrupt Structure

The 8051A and 80C51 and their ROMless versions, provide 5 interrupt sources: 2

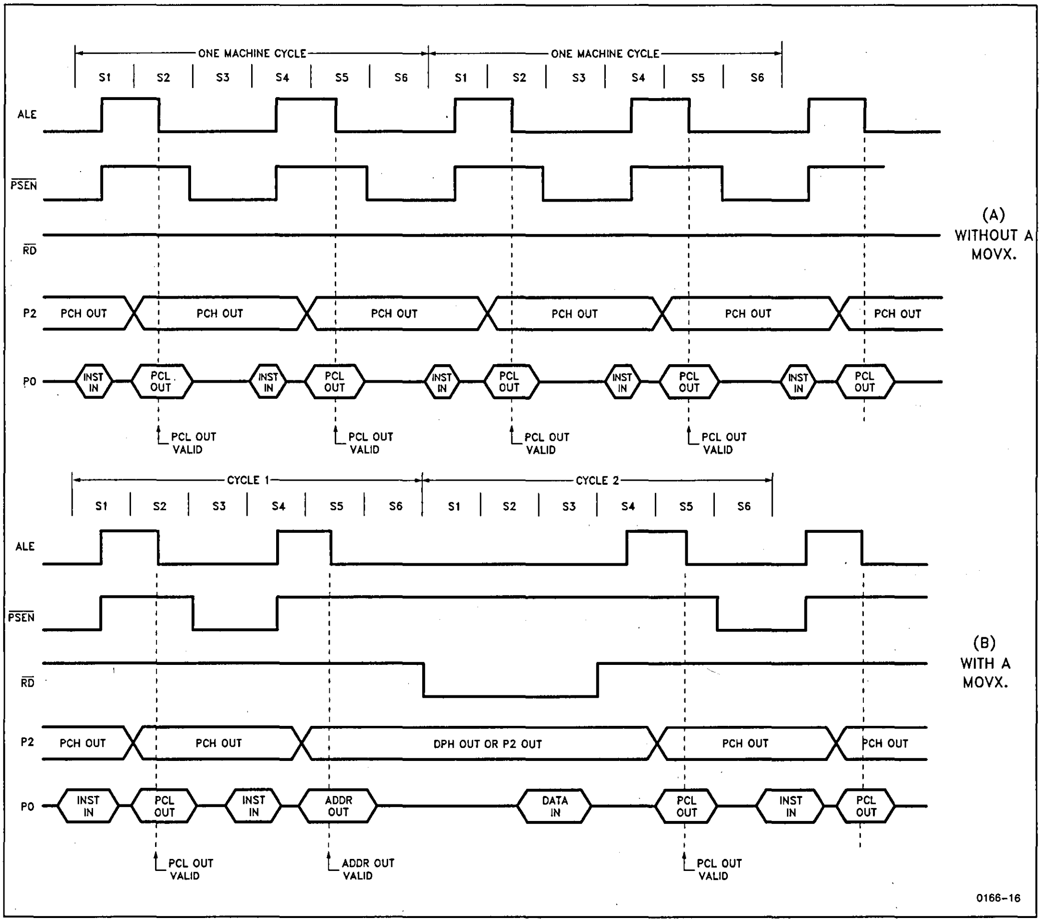


Figure 16. Bus Cycles in SAB-51 Devices Executing from External Program Memory

external interrupts, 2 timer interrupts, and the serial port interrupt. The 8052B and 80C52 provide these 5 plus a sixth interrupt that is associated with the third timer/counter which is present in the device. Additional interrupts are available on the 80512, 80515, 80C515 and 80C517. Refer to the appropriate chapters on these devices for further information on their interrupts.

What follows is an overview of the interrupt structure for these devices. More detailed information for specific members of the SAB-51 family is provided in the chapters of this handbook that describe the specific devices.

Interrupt Enables

Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the SFR named IE (Interrupt Enable). This register also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 17 shows the IE register for the 80528 and 80C52.

(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	reserved.
ET2	IE.5	enables or disables the Timer 2 overflow or capture interrupt. If ET2 = 0, the Timer 2 interrupt is disabled.
ES	IE.4	enables or disables the Serial Port interrupt. If ES = 0, the Serial Port interrupt is disabled.
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.
EX1	IE.2	enables or disables External Interrupt 1. If EX1 = 0, External Interrupt 1 is disabled.
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External Interrupt 0. If EX0 = 0, External Interrupt 0 is disabled.

Figure 17. IE (Interrupt Enable) Register in the 8052B and 80C52

Interrupt Priorities

Each interrupt source can also be individually programmed to one of two priority levels (SAB 80515, 80C515 and 80C517 have four priority levels for the interrupts, please refer to their respective chapters for detailed information). By setting clearing a bit in the SFR named IP (Interrupt Priority) Figure 18 shows the IP register in the 8052B and 80C52.

A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence.

Figure 19 shows, for the 8052B, how the IE and IP registers and the polling sequence work to determine which if any interrupt will be serviced.

(MSB)							(LSB)
—	—	PT2	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
—	IP.7	reserved
—	IP.6	reserved
PT2	IP.5	defines the Timer 2 interrupt priority level. PT2 = 1 programs it to the higher priority level.
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 program it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PX1	IP.2	defines the External Interrupt 1 priority level. PX1 = 1 programs it to the higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level, PT0 = 1 programs it to the higher priority level.
PX0	IP.0	defines the External Interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

Figure 18. IP (Interrupt Priority) Register in the 8052B and 80C52

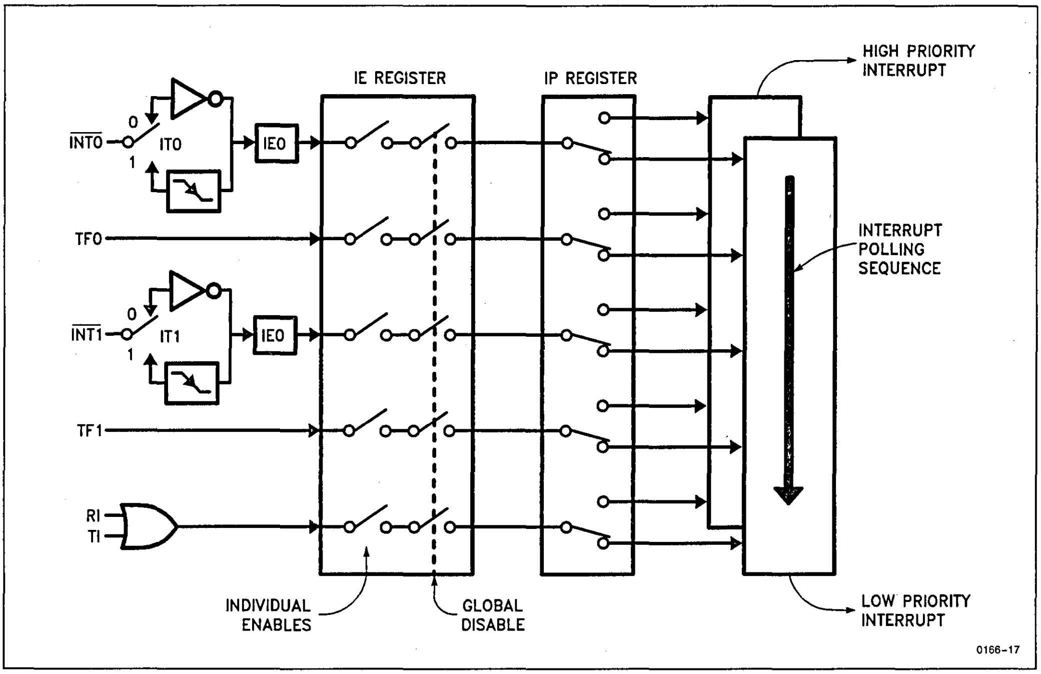


Figure 19. 8052 Interrupt Control System

In operation, all the interrupt flags are latched into the interrupt control system during State 5 of every machine cycle. The samples are polled during the following machine cycle. If the flag for an enabled interrupt is found to be set (1), the interrupt system generates an LCALL to the appropriate location in Program Memory, unless some other condition blocks the interrupt. Several conditions can block an interrupt, among them that an interrupt of equal or higher priority level is already in progress.

The hardware-generated LCALL causes the contents of the Program Counter to be pushed onto the stack, and reloads the PC with the beginning address of the service routine. As previously noted (Figure 3), the service routine for each interrupt begins at a fixed location.

Only the Program Counter is automatically pushed onto the stack, not the PSW or any other register. Having only the PC be automatically saved allows the programmer to decide how much time to spend saving which other registers. This enhances the interrupt response time, albeit at the expense of increasing the programmer's burden of responsibility. As a result, many interrupt functions that are typical in control applications—toggling a port pin, for example, or reloading a timer, or unloading a serial buffer—can often be completed in less time than it takes other architectures to commence them.

Simulating a Third Priority Level in Software

Some applications require more than the two priority levels that are provided by on-chip hardware in SAB-51 devices. In these cases, relatively simple software can be written to produce the same effect as a third priority level.

First, interrupts that are to have higher priority than 1 are assigned to priority 1 in the IP (Interrupt Priority) register. The service routines for priority 1 interrupts that are supposed to be interruptible by "priority 2" interrupts are written to include the following code:

```
          PUSH  IE
          MOV   IE,#MASK
          CALL  LABEL

                 *******
     (execute service routine)
                 *******

          POP   IE
          RET
LABEL:    RETI
```

As soon as any priority 1 interrupt is acknowledged, the IE (Interrupt Enable) register is re-defined so as to disable all but "priority 2" interrupts. Then, a CALL to LABEL executes the RETI instruction, which clears the priority 1 interrupt-in-progress flip-flop. At this point any priority 1 interrupt that is enabled can be serviced, but only "priority 2" interrupts are enabled. POPping IE restores the original enable byte. Then a normal RET (rather than another RETI) is used to terminate the service routine. The additional software adds 10 μs (at 12 MHz) to priority 1 interrupts.

8-Bit Single-Chip Microcontroller Components Data Sheets

SIEMENS

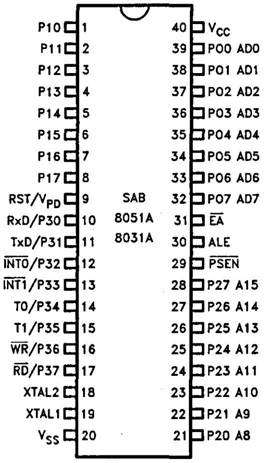
SAB 8051A/8031A
SAB 8051A-16/8031A-16
8-Bit Single-Chip Microcontroller

SAB 8051A/8051A-16 Microcontroller with factory-mask programmable ROM
SAB 8031A/8031A-16 Microcontroller for external ROM

- SAB 8051A/8031A, 12 MHz Operation SAB 8051A-16/8031A-16, 16 MHz Operation
- 4K × 8 ROM
- 128 × 8 RAM
- Four 8-Bit Ports, 32 I/O Lines
- Two 16-Bit Timer/Event Counters
- High-Performance Full-Duplex Serial Channel
- Boolean Processor
- Compatible with SAB 8080/8085 Peripherals
- External Memory Expandable up to 128 Kbytes
- 218 User Bit-Addressable Locations
- Most Instructions Execute in:
  1 μs (SAB 8051A/8031A)
  750 ns (SAB 8051A-16/8031A-16)
- 4 μs (3 μs) Multiply and Divide
- P-DIP 40 and PLCC 44 Packages

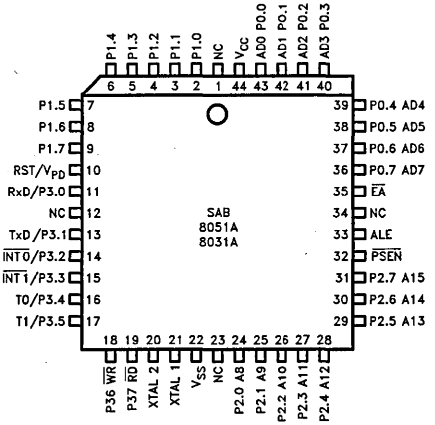
Pin Configurations

P-DIP 40



0192-1

PLCC 44



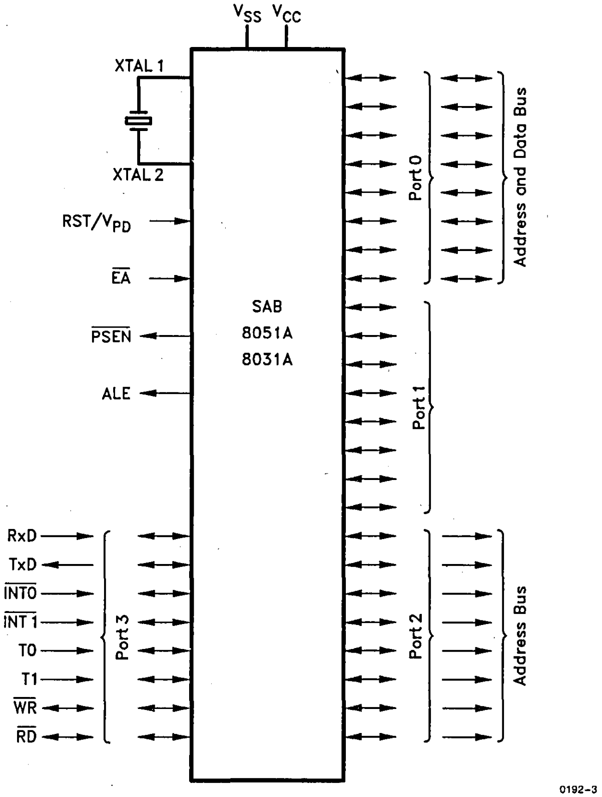
0192-2

The SAB 8051A/8031A is a stand-alone, high-performance single-chip microcontroller fabricated in +5V advanced Siemens MYMOS (III) technology and supplied in a 40-pin plastic P-DIP or 44-pin plastic leaded chip carrier (PLCC 44) package. It provides the hardware features, architectural enhancements and instructions that are necessary to make it a powerful and cost-effective controller for applications requiring up to 64 Kbytes of program memory and/or up to 64 Kbytes of data storage.

The SAB 8051A contains a non-volatile 4K × 8 read-only program memory; a volatile 128 × 8 read/write data memory; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multiprocessor communications, I/O expansion, or full-duplex UART; and on-chip oscillator and clock circuits. The SAB 8031A is identical with the SAB 8051A, except that it lacks the program memory.

For systems that require extra capability, the SAB 8051A can be expanded using standard TTL-compatible memories and the byte-oriented SAB 8080 and SAB 8085 peripherals.

Logic Symbol



Pin Definitions and Functions

Pin		Symbol	Input(I)
			Output(O)
				Function

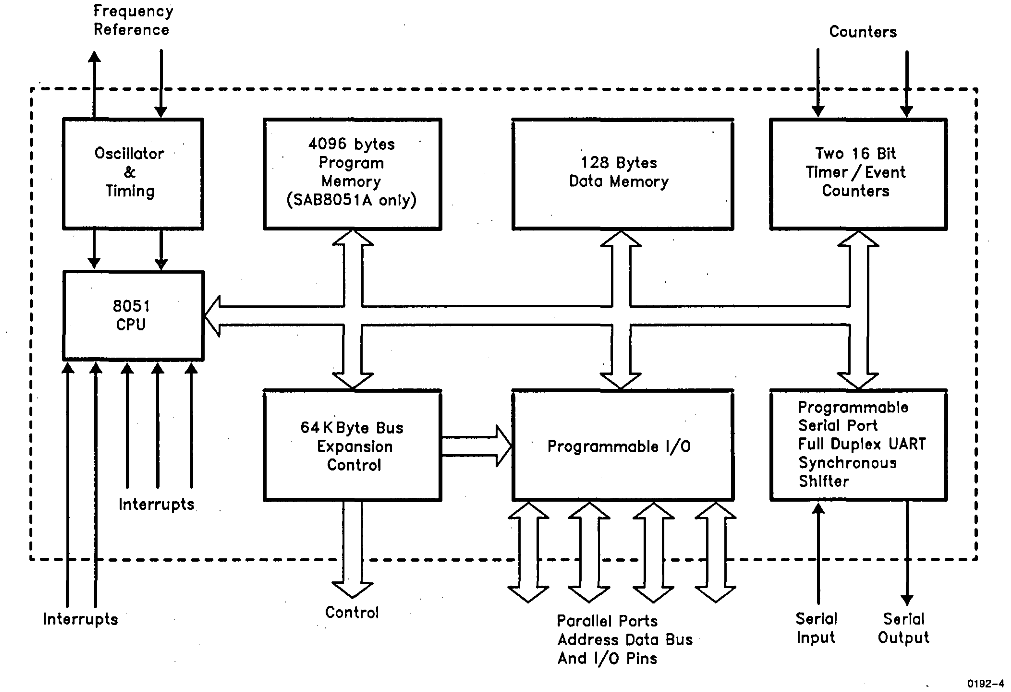
P-DIP-40	PLCC 44			
1–8	2–9	P1.0–P1.7	I/O	Port 1 is an 8-bit quasi-birirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads.
9	10	RST/V_{PD}	I	A high level on this pin resets the SAB 8051A. A small internal pulldown resistor permits power-on reset using only a capacitor connected to V_{CC}. If V_{PD} is held within its spec while V_{CC} drops below spec, V_{PD} will provide standby power to the RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC}.
10–17	11, 13–19	P3.0–P3.7	I/O	Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS TTL loads. The secondary functions are assigned to the pins of port 3, as follows: —RxD/data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous). —TxD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous). —$\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0. —$\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1. —T0 (P3.4). Input to counter 0. —T1 (P3.5). Input to counter 1. —$\overline{WR}$ (P3.6). The write control signal latches the data byte from port 0 into the external data memory. —$\overline{RD}$ (P3.7). The read control signal enables external data memory to port 0.
19 18	21 20	XTAL1 XTAL2		XTAL 1 input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to V_{SS} when external source is used on XTAL 2. XTAL 2 output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.
21–28	24–31	P2.0–P2.7	I/O	Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.
29	32	$\overline{PSEN}$	O	The program store enable output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.

Pin Definitions and Functions (Continued)

Pin		Symbol	Input(I)
			Output(O)
				Function

P-DIP-40	PLCC 44			
30	33	ALE	O	Provides address latch enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
31	35	$\overline{EA}$	I	When held at a TTL high level, the SAB 8051A executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the SAB 8051A fetches all instructions from external program memory. For the SAB 8031A this pin must be tied low.
39–32	43–36	P0.0–P0.7	I/O	Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads.
40	44	V_{CC}		+5V power supply during operation and program verification.
20	22	V_{SS}		Ground (0V)
	1, 12 23, 34	NC		No Connection

Block Diagram


Frequency Reference
Counters
Oscillator & Timing
4096 bytes Program Memory (SAB8051A only)
128 Bytes Data Memory
Two 16 Bit Timer / Event Counters
8051 CPU
64 K Byte Bus Expansion Control
Programmable I/O
Programmable Serial Port Full Duplex UART Synchronous Shifter
Interrupts
Interrupts
Control
Parallel Ports Address Data Bus And I/O Pins
Serial Input
Serial Output
0192-4

Absolute Maximum Ratings*

Ambient Temperature under Bias ....0°C to +70°C

Storage Temperature ....−65°C to +150°C

Voltage on Any Pin with
Respect to Ground (V_{SS}) ....−0.5V to 7V

Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Parameter	Symbol	Test Condition	Limit Values		Unit
			Min	Max	
Input Low Voltage	V_{IL}		−0.5	0.8	V
Input High Voltage (Except RST/VPD and XTAL2)	V_{IH}		2.0	V_{CC} + 0.5	V
Input High Voltage to RST/VPD for Reset, XTAL2	V_{IH1}	XTAL1 to V_{SS}	2.5	V_{CC} + 0.5	V
Power Down Voltage to RST/VPD	V_{PD}	V_{CC} = 0V	4.5	5.5	V
Output Low Voltage Ports 1, 2, 3	V_{OL}	I_{OL} = 1.6 mA		0.45	V
Output Low Voltage Port 0, ALE, $\overline{PSEN}$	V_{OL1}	I_{OL} = 3.2 mA		0.45	V
Output High Voltage Ports 1, 2, 3	V_{OH}	I_{OH} = −80 μA	2.4		V
Output High Voltage Port 0, ALE, $\overline{PSEN}$	V_{OH1}	I_{OH} = −400 μA	2.4		V
Logical 0 Input Current Ports 1, 2, 3	I_{IL}	V_{IL} = 0.45V		−500	μA
Logical 0 Input Current XTAL 2	I_{IL2}	XTAL1 = V_{SS} V_{IL} = 0.45V		−3.2	mA
Input High Current to RST/V_{PD} for Reset	I_{IH1}	V_{IN} = V_{CC} −1.5V		500	μA
Input Leakage Current to Port 0, $\overline{EA}$	I_{LI}	0V < V_{IN} < V_{CC}		±10	μA
Power Supply Current SAB 8031A/8051A SAB 8031A-16/8051A-16	I_{CC}	All Outputs Disconnected		 125 140	 mA mA
Power Down Current	I_{PD}	V_{CC} = 0V		10	mA
Capacitanco of I/O Buffer	C_{IO}	f_c = 1 MHz		10	pF

A.C. Characteristics for SAB 8051A/8031A

T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V
(C_L for Port 0, ALE and $\overline{\text{PSEN}}$ Outputs = 100 pF; C_L for All Other Outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1.2 MHz to 12 MHz		
		Min	Max	Min	Max	
ALE Pulse Width	t_{LHLL}	127		2 t_{CLCL} − 40		ns
Address Setup to ALE	t_{AVLL}	53		t_{CLCL} − 30		ns
Address Hold after ALE	t_{LLAX1}	48		t_{CLCL} − 35		ns
ALE to Valid Instruction In	t_{LLIV}		233		4 t_{CLCL} − 100	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	58		t_{CLCL} − 25		ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	215		3 t_{CLCL} − 35		ns
$\overline{\text{PSEN}}$ to Valid Instruction In	t_{PLIV}		150		3 t_{CLCL} − 100	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	t_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	t_{PXIZ}*		63		t_{CLCL} − 20	ns
Address Valid after $\overline{\text{PSEN}}$	t_{PXAV}*	75		t_{CLCL} − 8		ns
Address to Valid Instruction In	t_{AVIV}		302		5 t_{CLCL} − 115	ns
Address Float to $\overline{\text{PSEN}}$	t_{AZPL}	0		0		ns

NOTE:
*Interfacing the SAB 8051A to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1.2 MHz to 12 MHz		
		Min	Max	Min	Max	
$\overline{\text{RD}}$ Pulse Width	t_{RLRH}	400		6 t_{CLCL} − 100		ns
$\overline{\text{WR}}$ Pulse Width	t_{WLWH}	400		6 t_{CLCL} − 100		ns
Address Hold after ALE	t_{LLAX2}	132		2 t_{CLCL} − 35		ns
$\overline{\text{RD}}$ to Valid Data In	t_{RLDV}		252		5 t_{CLCL} − 165	ns
Data Hold after $\overline{\text{RD}}$	t_{RHDX}	0		0		ns
Data Float after $\overline{\text{RD}}$	t_{RHDZ}		97		2 t_{CLCL} − 70	ns
ALE to Valid Data In	t_{LLDV}		517		8 t_{CLCL} − 150	ns
Address to Valid Data In	t_{AVDV}		585		9 t_{CLCL} − 165	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{LLWL}	200	300	3 t_{CLCL} − 50	3 t_{CLCL} + 50	ns
Address to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{AVWL}	203		4 T_{CLCL} − 130		ns

External Data Memory Characteristics (Continued)

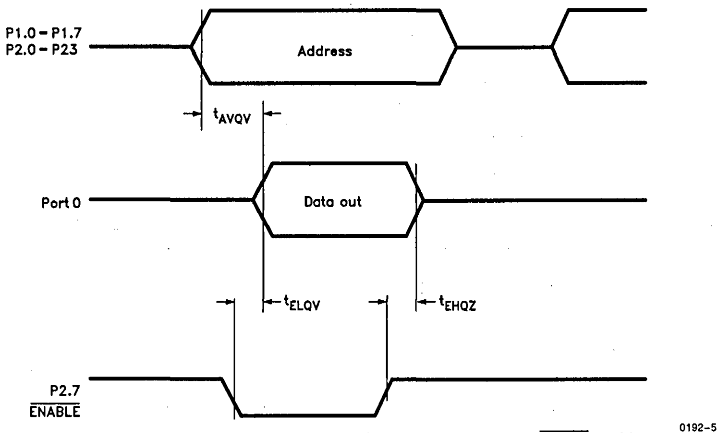
Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1.2 MHz to 12 MHz		
		Min	Max	Min	Max	
$\overline{WR}$ or $\overline{RD}$ High to ALE High	t_{WHLH}	43	123	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
Data Valid to $\overline{WR}$ Transition	t_{QVWX}	33		$t_{CLCL} - 50$		ns
Data Setup before $\overline{WR}$	t_{QVWH}	433		$7\ t_{CLCL} - 150$		ns
Data Hold after $\overline{WR}$	t_{WHQX}	33		$t_{CLCL} - 50$		ns
Address Float after $\overline{RD}$	t_{RLAZ}		0		0	ns

External Clock Drive XTAL2

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 1.2 MHz to 12 MHz		
		Min	Max	
Oscillator Period	t_{CLCL}	83.3	833.3	ns
High Time	t_{CHCX}	20	$t_{CLCL} - t_{CLCX}$	ns
Low Time	t_{CLCX}	20	$t_{CLCL} - t_{CHCX}$	ns
Rise Time	t_{CLCH}		20	ns
Fall Time	t_{CHCL}		20	ns

Waveforms: Refer to SAB 8051A/8031A

ROM Verification



Address:	P1.0–P1.7 = A0–A7	
	P2.0–P2.3 = A8–A11	
Data:	Port 0 = D0–D7	

Inputs:	P2.4–P2.6, $\overline{\text{PSEN}}$	= V_{SS}
	ALE, $\overline{\text{EA}}$	= V_{IH}
	R_{ST}/V_{PD}	= V_{IH1}

A.C. Characteristics for SAB 8051A-16/8031A-16

T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V
(C_L for Port 0, ALE and $\overline{\text{PSEN}}$ Outputs = 100 pF; C_L for All Other Outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		16 MHz Clock		Variable Clock 1/t_{CLCL} = 1.2 MHz to 16 MHz		
		Min	Max	Min	Max	
ALE Pulse Width	t_{LHLL}	85		2 t_{CLCL} – 40		ns
Address Setup to ALE	t_{AVLL}	33		t_{CLCL} – 30		ns
Address Hold after ALE	t_{LLAX1}	28		t_{CLCL} – 35		ns
ALE to Valid Instruction In	t_{LLIV}		150		4 t_{CLCL} – 100	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	38		t_{CLCL} – 25		ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	153		3 t_{CLCL} – 35		ns
$\overline{\text{PSEN}}$ to Valid Instruction In	t_{PLIV}		88		3 t_{CLCL} – 100	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	t_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	t_{PXIZ}*		48		t_{CLCL} – 15	ns
Address Valid after $\overline{\text{PSEN}}$	t_{PXAV}*	60		t_{CLCL} – 3		ns
Address to Valid Instruction In	t_{AVIV}		223		5 t_{CLCL} – 90	ns
Address Float to $\overline{\text{PSEN}}$	t_{AZPL}	0		0		ns

NOTE:
*Interfacing the SAB 8051A-16 to devices with float times up to 55 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		16 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1.2 MHz to 16 MHz		
		Min	Max	Min	Max	
$\overline{\text{RD}}$ Pulse Width	t_{RLRH}	275		$6\,t_{CLCL} - 100$		ns
$\overline{\text{WR}}$ Pulse Width	t_{WLWH}	275		$6\,t_{CLCL} - 100$		ns
Address Hold after ALE	$t_{LLAX\,2}$	90		$2\,t_{CLCL} - 35$		ns
$\overline{\text{RD}}$ to Valid Data In	t_{RLDV}		148		$5\,t_{CLCL} - 165$	ns
Data Hold after $\overline{\text{RD}}$	t_{RHDX}	0		0		ns
Data Float after $\overline{\text{RD}}$	t_{RHDZ}		55		$2\,t_{CLCL} - 70$	ns
ALE to Valid Data In	t_{LLDV}		350		$8\,t_{CLCL} - 150$	ns
Address to Valid Data In	t_{AVDV}		398		$9\,t_{CLCL} - 165$	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{LLWL}	138	238	$3\,t_{CLCL} - 50$	$3\,t_{CLCL} + 50$	ns
Address to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{AVWL}	120		$4\,T_{CLCL} - 130$		ns
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ High to ALE High	t_{WHLH}	23	103	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
Data Valid to $\overline{\text{WR}}$ Transition	t_{QVWX}	13		$t_{CLCL} - 50$		ns
Data Setup before $\overline{\text{WR}}$	t_{QVWH}	288		$7\,t_{CLCL} - 150$		ns
Data Hold after $\overline{\text{WR}}$	t_{WHQX}	13		$t_{CLCL} - 50$		ns
Address Float after $\overline{\text{RD}}$	t_{RLAZ}		0		0	ns

External Clock Drive XTAL2

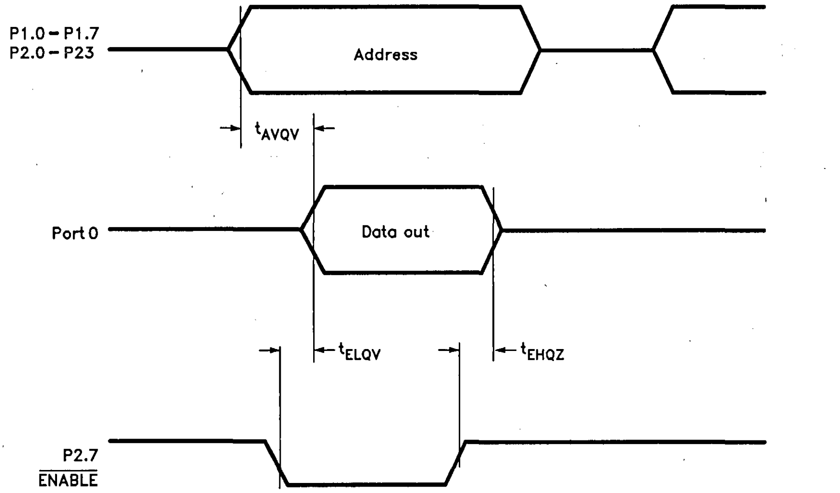
Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 1.2 MHz to 16 MHz		
		Min	Max	
Oscillator Period	t_{CLCL}	62.5	833.3	ns
High Time	t_{CHCX}	15	$t_{CLCL} - t_{CLCX}$	ns
Low Time	t_{CLCX}	15	$t_{CLCL} - t_{CHCX}$	ns
Rise Time	t_{CLCH}		15	ns
Fall Time	t_{CHCL}		15	ns

ROM Verification Characteristics for SAB 8051A-16

T_A = 25°C ±5°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Parameter	Symbol	Limit Values		Unit
		Min	Max	
Address to Valid Data	t_{AVQV}		48 t_{CLCL}	ns
$\overline{\text{ENABLE}}$ to Valid Data	t_{ELQV}		48 t_{CLCL}	ns
Data Float after $\overline{\text{ENABLE}}$	t_{EHQZ}	0	48 t_{CLCL}	ns
Oscillator Frequency	$1/t_{CLCL}$	4	6	MHz

ROM Verification

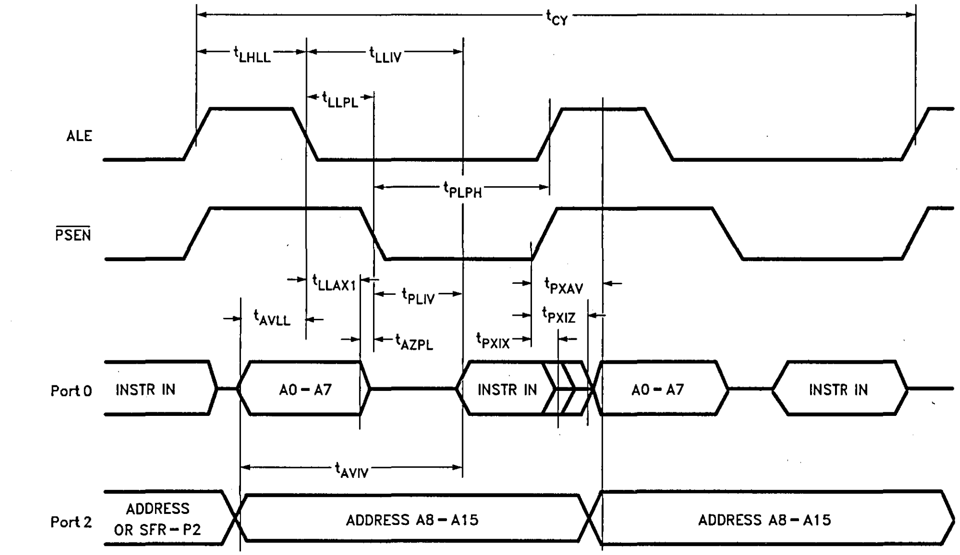


Address: P1.0-P1.7 = A0-A7
P2.0-P2.3 = A8-A11
Data: Port 0 = D0-D7

Inputs: P2.4-P2.6, $\overline{\text{PSEN}}$ = V_{SS}
ALE, $\overline{\text{EA}}$ = V_{IH}
R_{ST}/V_{PD} = V_{IH1}

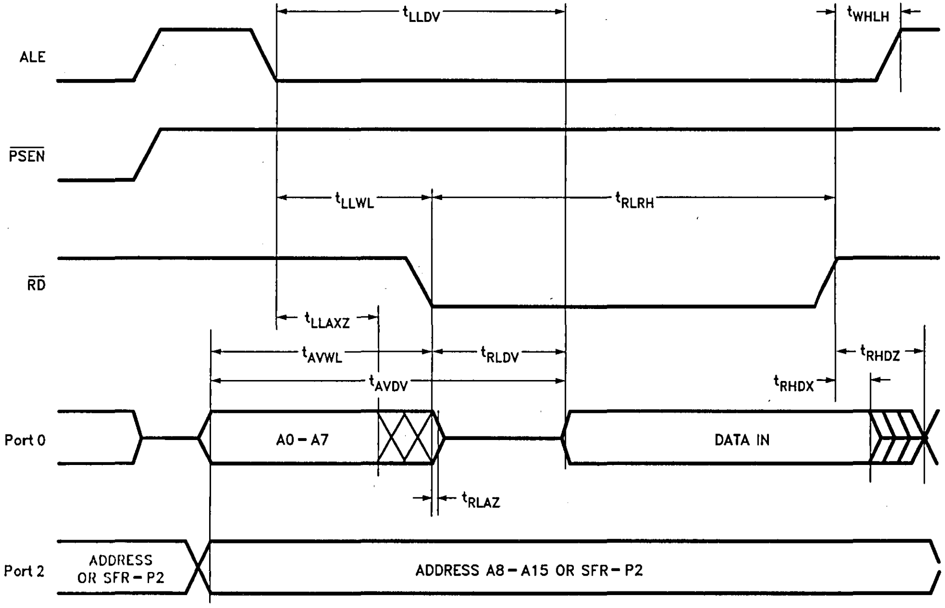
Waveforms

Program Memory Read Cycle



0192–6

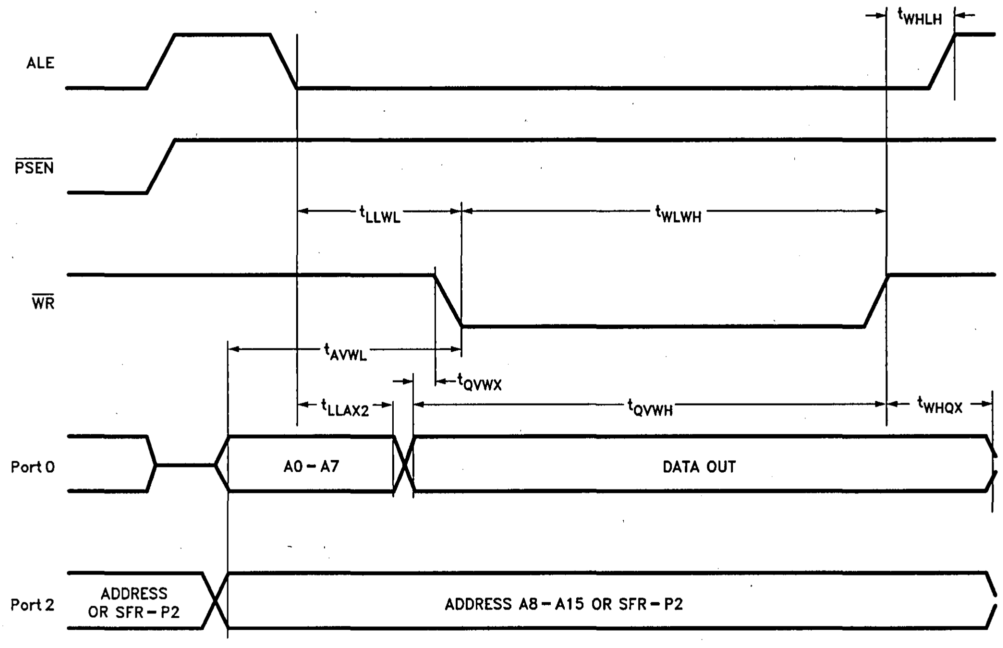
Data Memory Read Cycle



0192–7

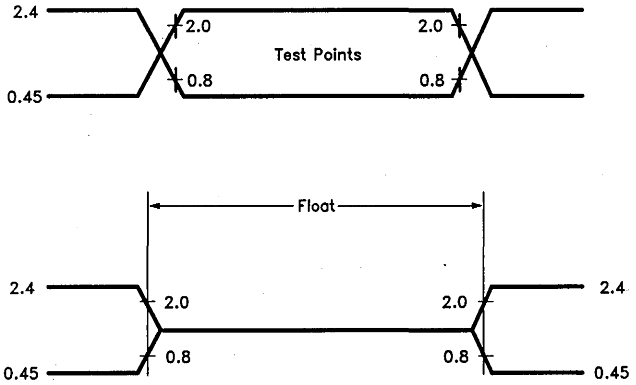
Waveforms (Continued)

Data Memory Write Cycle



0192-8

A.C. Testing Input, Output, Float Waveforms

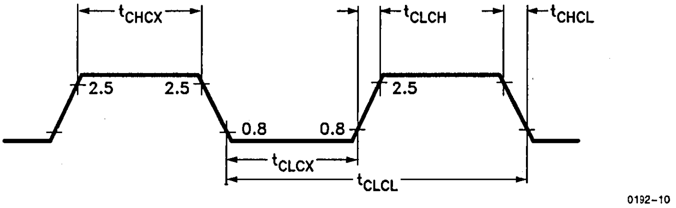


0192-9

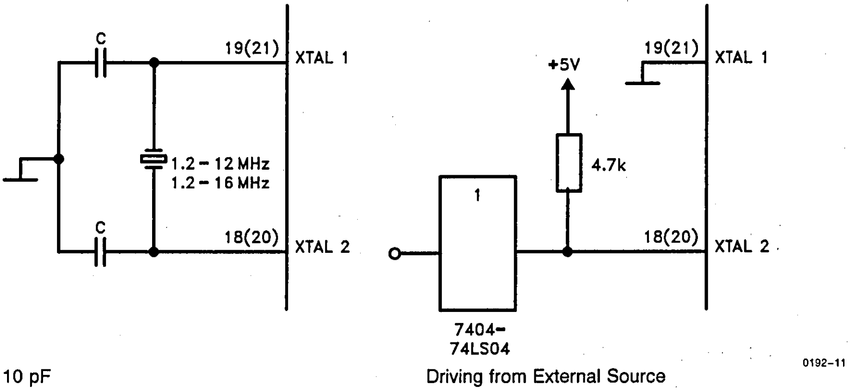
A.C. testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0".
Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".
For timing purposes, the float state is defined as the point at which a P0 pin sinks 3.2 mA or sources 400 μA at the voltage test levels.

Waveforms (Continued)

External Clock Cycle



Recommended Oscillator Circuits



C = 30 pF ±10 pF
Crystal Oscillator Mode
(Pin number in (. . .) are for PLCC 44 Package)

Driving from External Source

Ordering Information

Type	Description
SAB 8051A-P	8-Bit Single-Chip Microcontroller with Mask-Programmable ROM (P-DIP-40)
SAB 8031A-P	for External Memory (P-DIP-40)
SAB 8051A-16-P	with Mask-Programmable ROM (P-DIP-40)
SAB 8031A-16-P	for External Memory (P-DIP-40)
SAB 8051A-N	with Mask-Programmable ROM (PL-CC-44)
SAB 8031A-N	for External Memory (PL-CC-44)
SAB 8051A-16-N	with Mask-Programmable ROM (PL-CC-44)
SAB 8031A-16-N	for External Memory (PL-CC-44)

SIEMENS

Preliminary

SAB 8051A/8031A Ext. Temp 8-Bit Single-Chip Microcontroller

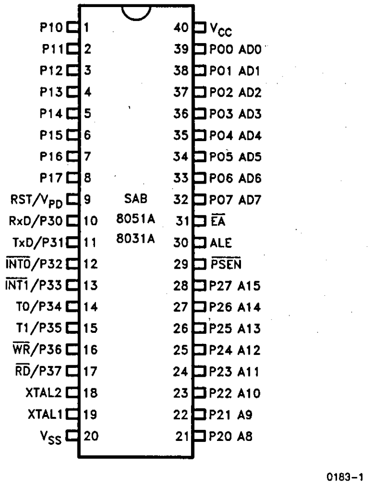
Extended Temperature Range: −40°C to +85°C
−40°C to +110°C

Mask-Programmable ROM
SAB 8051A-12-P-T40/85
SAB 8051A-10-P-T40/110

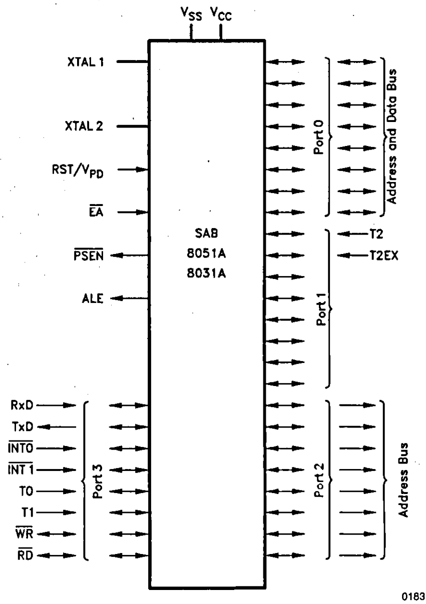
External ROM
SAB 8031A-12-P-T40/85
SAB 8031A-10-P-T40/110

- Advanced Version of the SAB 8031/8051 for Extended Temperature Range
- SAB 8051A/8031A-12-T40/85: 12 MHz Operation
- SAB 8051A/8031A-10-T40/110: 10 MHz Operation
- 4K × 8 ROM
- 128 × 8 RAM
- Four 8-Bit Ports, 32 I/O Lines
- Two 16-Bit Timer/Event Counters
- High-Performance Full-Duplex Serial Channel
- External Memory Expandable up to 128K
- Compatible with SAB 8080/8085 Peripherals
- Boolean Processor
- 218 User Bit-Addressable Locations
- Most Instructions Execute in 1 μs
- 4 μs Multiply and Divide

Pin Configuration



Logic Symbol



The SAB 8051A/8031A for the two extended temperature ranges (industrial temperature range: −40°C to +85°C, automotive temperature range: −40°C to +110°C) is fully compatible with the standard SAB 8051A/8031A with respect to architecture, instruction set, and software portability.

The SAB 8051A/8031A is a stand-alone, high-performance single-chip microcontroller fabricated in +5V advanced N-channel, silicon gate Siemens MYMOS technology and packaged in a 40-pin DIP.

The SAB 8051A contains a non-volatile 4K × 8 read-only program memory; a volatile 128 × 8 read/write data memory; 32 I/O lines; two 16-bit timer/counters; a five-source two-priority-level, nested interrupt structure; a serial I/O port for either multiprocessor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. The SAB 8031A is identical with the SAB 8051A, except that it lacks the program memory.

For systems that require extra capability, the SAB 8051A can be expanded using standard TTL compatible memories and the byte-oriented SAB 8080 and SAB 8085 peripherals.

Absolute Maximum Ratings*

Ambient Temperature under Bias
T40/85 .... −40°C to +85°C
T40/110 .... −40°C to +110°C
Storage Temperature .... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground (V_{SS}) .... −0.5V to +7V
Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = −40°C to +85°C for T40/85;
T_A = −40°C to + 110°C for T40/110

Parameter	Symbol	Test Condition	Limit Values		Unit
			Min	Max	
Input Low Voltage	V_{IL}		−0.5	0.8	V
Input High Voltage Except RST/VDP and XTAL2	V_{IH}		2.0	V_{CC}+0.5	V
Input High Voltage to RST/VPD for Reset, XTAL2	V_{IH1}	XTAL 1 to V_{SS}	2.5	V_{CC}+0.5	V
Power Down Voltage to RST/VPD	V_{PD}	V_{CC} = 0V	4.5	5.5	V
Output Low Voltage Ports 1, 2, 3	V_{OL}	I_{OL} = 1.6 mA		0.45	V
Output Low Voltage Port 0, ALE, $\overline{PSEN}$	V_{OL1}	I_{OL} = 3.2 mA		0.45	V
Output High Voltage Ports 1, 2, 3	V_{OH}	I_{OH} = −80 µA	2.4		V
Output High Voltage Port 0, ALE, $\overline{PSEN}$	V_{OH1}	I_{OH} = −400 µA	2.4		V
Logical 0 Input Current Ports 1, 2, 3	I_{IL}	V_{IL} = 0.45V		−500	µA
Logical 0 Input Current XTAL2	I_{IL2}	XTAL1 = V_{SS} V_{IL} = 0.45V		−3.2	mA
Input High Current to RST/VPD for Roset	I_{IH1}	V_{IN} = V_{CC} −1.5V		500	µA
Input Leakago Current to Port 0, $\overline{EA}$	I_{LI}	0 < V_{IN} < V_{CC}		±10	µA
Power Supply Current	I_{CC}			150	mA
Power Down Current	I_{PD}			15	mA
Capacitance of I/O Buffer	C_{IO}	f_c = 1 MHz		10	pF

AC Characteristics for T40/85: Refer to SAB 8051A/8031A Data Sheet.

AC Characteristics for T40/110

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = −40 to +110°C
(C_L for Port 0, ALE and $\overline{\text{PSEN}}$ Outputs = 100 pF; C_L for all other Outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		10 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1.2 MHz to 10 MHz		
		Min	Max	Min	Max	
ALE Pulse Width	t_{LHLL}	160		$2\,t_{CLCL} - 40$		ns
Address Setup to ALE	t_{AVLL}	70		$t_{CLCL} - 30$		ns
Address Hold after ALE	t_{LLAX1}	65		$t_{CLCL} - 35$		ns
ALE to Valid Instruction In	t_{LLIV}		300		$4\,t_{CLCL} - 100$	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	75		$t_{CLCL} - 25$		ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	265		$3\,t_{CLCL} - 35$		ns
$\overline{\text{PSEN}}$ to Valid Instruction In	t_{PLIV}		200		$3\,t_{CLCL} - 100$	ns
Input Instruction Hold After $\overline{\text{PSEN}}$	t_{PXIX}	0		0		ns
Input Instructions Float After $\overline{\text{PSEN}}$	t_{PXIZ}*		80		$t_{CLCL} - 20$	ns
Address Valid After $\overline{\text{PSEN}}$	t_{PXAV}*	92		$t_{CLCL} - 8$		ns
Address to Valid Instruction In	t_{AVIV}		385		$5\,t_{CLCL} - 115$	ns
Address Float to $\overline{\text{PSEN}}$	t_{AZPL}	0		0		ns

NOTE:
*Interfacing the SAB 8051A to devices with float times up to 92 ns is permissible. This limited bus contention will not cause any damage to Port 0 drivers.

AC Characteristics for T40/110 (Continued)

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = −40 to +110°C
(C_L for Port 0, ALE and $\overline{PSEN}$ Outputs = 100 pF; C_L for all other Outputs = 80 pF)

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		10 MHz Clock		Variable Clock 1/t_{CLCL} = 1.2 MHz to 10 MHz		
		Min	Max	Min	Max	
$\overline{RD}$ Pulse Width	t_{RLRH}	500		6 t_{CLCL} − 100		ns
$\overline{WR}$ Pulse Width	t_{WLWH}	500		6 t_{CLCL} − 100		ns
Address Hold After ALE	t_{LLAX2}	165		2 t_{CLCL} − 35		ns
$\overline{RD}$ to Valid Data In	t_{RLDV}		335		5 t_{CLCL} − 165	ns
Data Hold After $\overline{RD}$	t_{RHDX}	0		0		ns
Data Float After $\overline{RD}$	t_{RHDZ}		130		2 t_{CLCL} − 70	ns
ALE to Valid Data In	t_{LLDV}		650		8 t_{CLCL} − 150	ns
Address to Valid Data In	t_{AVDV}		735		9 t_{CLCL} − 165	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	250	350	3 t_{CLCL} − 50	3 t_{CLCL} + 50	ns
Address to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	270		4 t_{CLCL} − 130		ns
$\overline{WR}$ or $\overline{RD}$ High to ALE High	t_{WHLH}	60	140	t_{CLCL} − 40	t_{CLCL} + 40	ns
Data Valid to $\overline{WR}$ Transition	t_{QVWX}	50		t_{CLCL} − 50		ns
Data Setup Before $\overline{WR}$	t_{QVWH}	550		7 t_{CLCL} − 50		ns
Data Hold After $\overline{WR}$	t_{WHQX}	50		t_{CLCL} − 50		ns
Address Float After $\overline{RD}$	t_{RLAZ}		0		0	ns

NOTE:
*Interfacing the SAB 8051A to devices with float times up to 92 ns is permissible. This limited bus contention will not cause any damage to Port 0 drivers.

External Clock Drive XTAL2

Parameter	Symbol	Limit Values		Unit

		Variable Clock
		Freq. = 1.2 MHz to 12 MHz (T40/85)
		Freq. = 1.2 MHz to 10 MHz (T40/110)		

		Min	Max	
Oscillator Period T40/85 T40/110	t_{CLCL}	83.3 100	833.3	ns
High Time	t_{CHCX}	20	t_{CLCL}−t_{CLCX}	ns
Low Timo	t_{CLCX}	20	t_{CLCL}−t_{CHCX}	ns
Rise Time	t_{CLCH}		20	ns
Fall Timo	t_{CHCL}		20	ns

Waveforms: Refer to SAB 8051A/8031A Data Sheet

Ordering Information

Type	Description
SAB 8051A-12-P-T40/85	8-Bit Single-Chip-Microcomputer with Mask-Programmable ROM (Plastic)
SAB 8051A-10-P-T40/110	With Mask-Programmable ROM (Plastic)
SAB 8031A-12-P-T40/85	For External Memory (Plastic)
SAB 8031A-10-P-T40/110	For External Memory (Plastic)

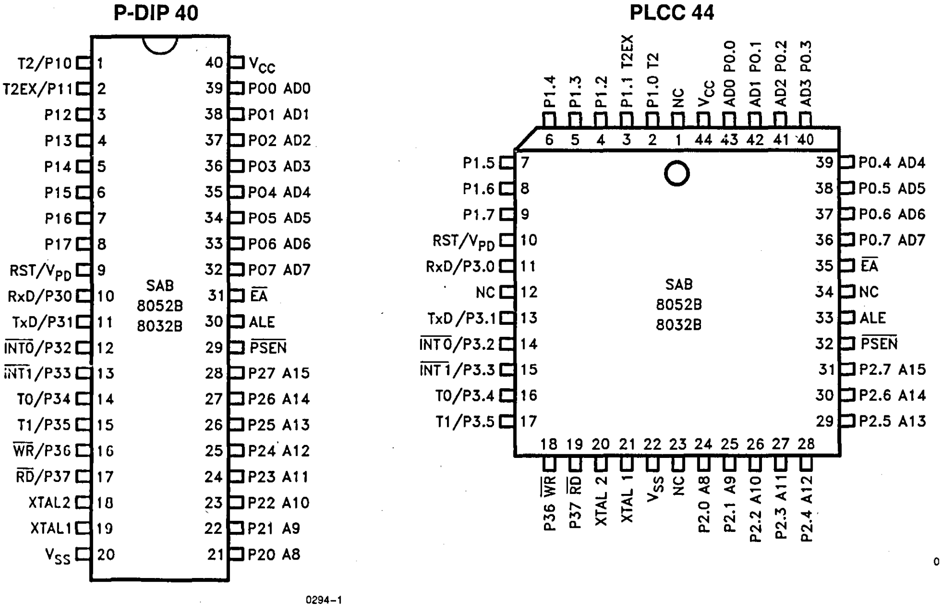
SIEMENS

SAB 8052B/8032B
SAB 8052B-16/8032B-16
8-Bit Single-Chip Microcontroller

SAB 8052B/8052B-16 Microcontroller with factory-mask programmable ROM
SAB 8032B/8032B-16 Microcontroller for external ROM

- SAB 8052B/8032B, 12 MHz Operation SAB 8052B-16/8032B-16, 16 MHz Operation
- 8K × 8 ROM (SAB 8052B only)
- 256 × 8 RAM
- Four 8-Bit Ports, 32 I/O Lines
- Three 16-Bit Timer/Event Counters
- High-Performance Full-Duplex Serial Channel
- External Memory Expandable up to 128 Kbytes
- Compatible with SAB 8080/8085 Peripherals
- Boolean Processor
- Most Instructions Execute in:
  1 μs (SAB 8052B/8032B)
  750 ns (SAB 8052B-16/8032B-16)
- 4 μs (3 μs) Multiply and Divide
- P-DIP 40 and PLCC 44 Packages
- Full Backward Compatibility with SAB 8051A/8031A

Pin Configurations


P-DIP 40

PLCC 44

The SAB 8052B/8032B is a stand-alone, high-performance single-chip microcontroller fabricated in +5V advanced Siemens MYMOS (III) technology and supplied in a 40-pin plastic P-DIP or 44-pin plastic leaded chip carrier (PLCC 44) package. It is backwardly compatible with the SAB 8051A/8031A. It provides the hardware features, architectural enhancements and instructions that are necessary to make it a powerful and cost-effective controller for applications requiring up to 64 Kbytes of program memory and/or up to 64 Kbytes of data storage.

The SAB 8052B contains a non-volatile 8K × 8 read-only program memory; a volatile 256 × 8 read/write data memory; 32 I/O lines; three 16-bit timer/counters; a five source, two-priority-level, nested interrupt structure; a serial I/O port for either multiprocessor communications, I/O expansion, or full-duplex UART; and on-chip oscillator and clock circuits. The SAB 8032B is identical with the SAB 8052B, except that it lacks the program memory.

For systems that require extra capability, the SAB 8052B can be expanded using standard TTL-compatible memories and the byte-oriented SAB 8080 and SAB 8085 peripherals.

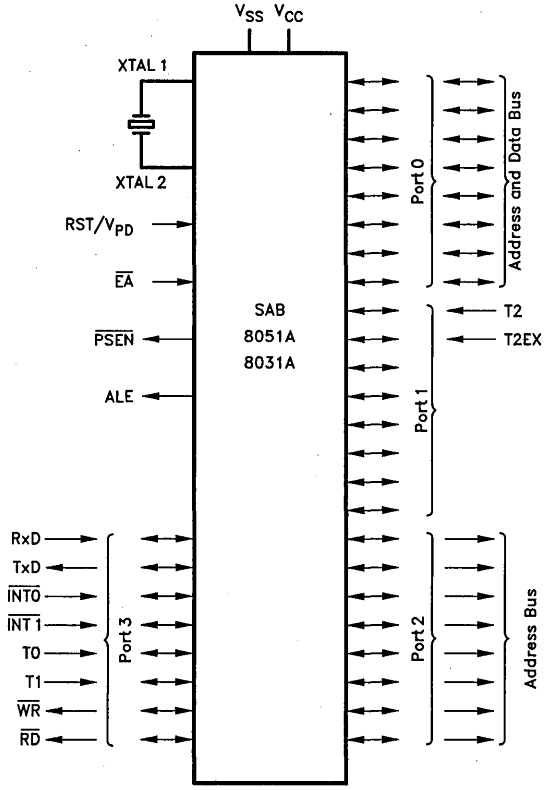
Pin Definitions and Functions

Symbol	Pin		Input (I) Output (O)	Function
	P-DIP-40	PLCC-44		
P1.0–P1.7	1–8	2–9	I/O	Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads.
RST/V_{PD}	9	10	I	A high level on this pin resets the SAB 8051A/8052B. A small internal pulldown resistor permits power-on reset using only a capacitor connected to V_{CC}. If V_{PD} is held within its specification while V_{CC} drops below specification, V_{PD} will provide standby power to the RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC}.
P3.0–P3.7	10–17	11, 13–19	I/O	Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS TTL loads. The secondary functions are assigned to the pins of port 3, as follows: —RxD/data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous). —TxD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous). —$\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0. —$\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1. —T0 (P3.4). Input to counter 0. —T1 (P3.5). Input to counter 1. —$\overline{WR}$ (P3.6). The write control signal latches the data byte from port 0 into the external data memory. —$\overline{RD}$ (P3.7). The read control signal enablos external data memory to port 0.

Pin Definitions and Functions (Continued)

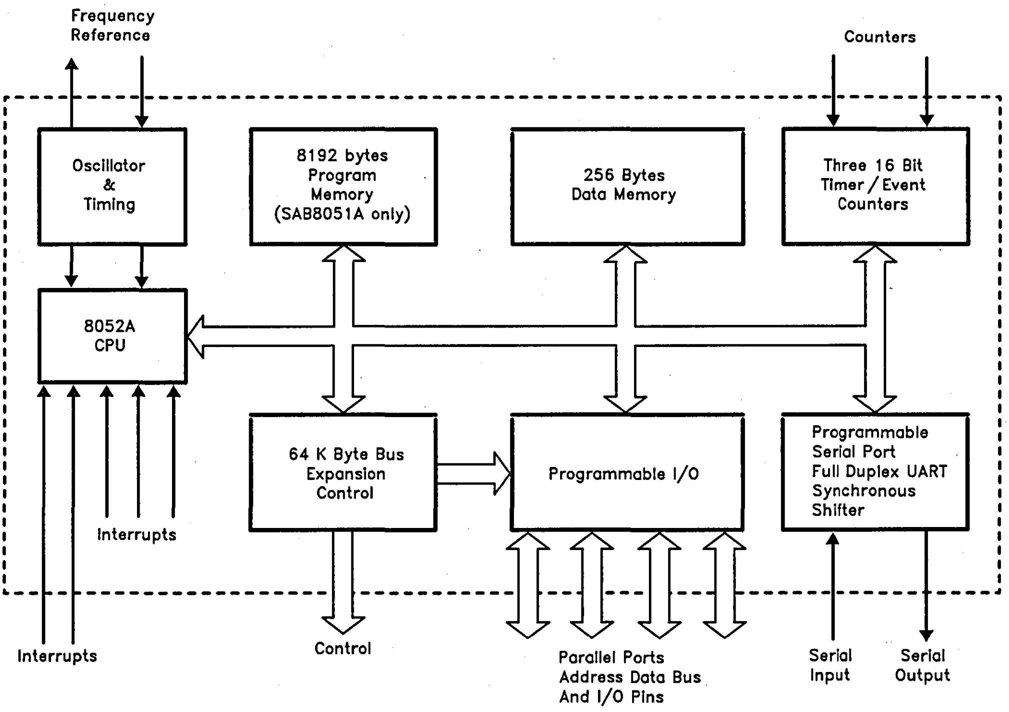
Symbol	Pin		Input (I) Output (O)	Function
	P-DIP-40	PLCC-44		
XTAL1 XTAL2	19 18	21 20		XTAL 1 input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to V_{SS} when external source is used on XTAL2. XTAL 2 output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.
P2.0–P2.7	21–28	24–31	I/O	Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.
$\overline{\text{PSEN}}$	29	32	O	The program store enable output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.
ALE	30	33	O	Provides address latch enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{\text{EA}}$	31	35	I	When held at a TTL high level, the SAB 8051A executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the SAB 8051A fetches all instructions from external program memory. For the SAB 8031A this pin must be tied low.
P0.0–P0.7	39–32	43–36	I/O	Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads.
V_{CC}	40	44		+ 5V power supply during operation and program verification.
V_{SS}	20	22		Ground (0V)
NC	—	1, 12 23, 24	—	No Connection

Logic Symbol



0294-3

Block Diagram


Frequency Reference
Counters
Oscillator & Timing
8192 bytes Program Memory (SAB8051A only)
256 Bytes Data Memory
Three 16 Bit Timer / Event Counters
8052A CPU
64 K Byte Bus Expansion Control
Programmable I/O
Programmable Serial Port Full Duplex UART Synchronous Shifter
Interrupts
Interrupts
Control
Parallel Ports Address Data Bus And I/O Pins
Serial Input
Serial Output

0294-4

Absolute Maximum Ratings*

Ambient Temperature under Bias ...... 0°C to 70°C
Storage Temperature .... $-65°C$ to $+150°C$
Voltage on Any Pin
with Respect to Ground (V_{SS}) .... $-0.5V$ to $+7V$
Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics $T_A = 0°C$ to $70°C$; $V_{CC} = 5V \pm 10\%$; $V_{SS} = 0V$

Parameter	Symbol	Test Conditions	Limit Values		Units
			Min	Max	
Input Low Voltage	V_{IL}		-0.5	0.8	V
Input High Voltage (except RST/V_{PD} and XTAL2)	V_{IH}		2.0	$V_{CC} + 0.5$	V
Input High Voltage to RST/V_{PD} for Reset, XTAL2	V_{IH1}	XTAL1 to V_{SS}	2.5	$V_{CC} + 0.5$	V
Power Down Voltage to RST/V_{PD}	V_{PD}	$V_{CC} = 0V$	4.5	5.5	V
Output Low Voltage Ports 1, 2, 3	V_{OL}	$I_{OL} = 1.6$ mA		0.45	V
Output Low Voltage Port 0, ALE, $\overline{PSEN}$	V_{OL1}	$I_{OL} = 3.2$ mA		0.45	V
Output High Voltage Ports 1, 2, 3	V_{OH}	$I_{OH} = -80\ \mu A$	2.4		V
Output High Voltage Port 0, ALE, $\overline{PSEN}$	V_{OH1}	$I_{OH} = -400\ \mu A$	2.4		V
Logical 0 Input Current Ports 1, 2, 3	I_{IL}	$V_{IL} = 0.45V$		-500	μA
Logical 0 Input Current XTAL2	I_{IL2}	XTAL1 = V_{SS} $V_{IL} = 0.45V$		-3.2	mA
Input High Current to RST/V_{PD} for Reset	I_{IH1}	$V_{IN} = V_{CC} - 1.5V$		500	μA
Input Leakage Current to Port 0, $\overline{EA}$	I_{LI}	$0 < V_{IN} < V_{CC}$		±10	μA
Power Supply Current SAB 8032B/8052B SAB 8032B-16/8052B-16	I_{CC}	All Outputs Disconnected		175	mA
Power Down Current	I_{PD}	$V_{CC} = 0V$	—	15	mA
Capacitance of I/O Buffer	C_{IO}	$f_C = 1$ MHz		10	pF

A.C. Characteristics for SAB 8052B/8032B

The A.C. Characteristics and Waveform Timings of the SAB 8052B/8032B and the SAB 8052B-16/8032B-16 are same as that of the SAB 8051A/8031A and SAB 8051A-16/8031A-16 respectively.

Ordering Information

Type	Description
	8-Bit Single-Chip Microcomputer
SAB 8052B-P	with Mask-Programmable ROM (P-DIP-40)
SAB 8032B-P	for External Memory (P-DIP-40)
SAB 8052B-16-P	with Mask-Programmable ROM (P-DIP-40)
SAB 8032B-16-P	for External Memory (P-DIP-40)
SAB 8052B-N	with Mask-Programmable ROM (PLCC-44)
SAB 8032B-N	for External Memory (PLCC-44)
SAB 8052B-16-N	with Mask Programmable ROM (PLCC-44)
SAB 8032B-16-N	for External Memory (PLCC-44)

SIEMENS

SAB 8052B/8032B Ext. Temp. 8-Bit Single-Chip Microcontroller

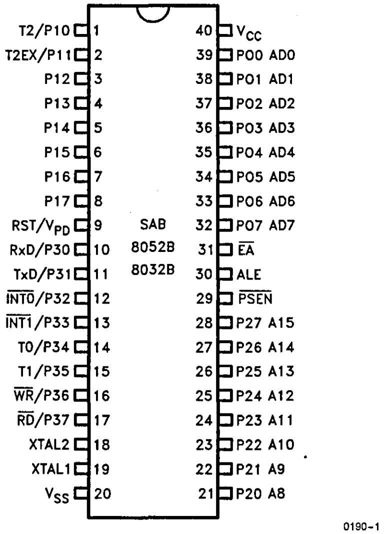
Extended Temperature Range: −40°C to +85°C
−40°C to +100°C

SAB 8052B-T40/85
SAB 8052B-T40/100 With mask-programmable ROM
SAB 8032B-T40/85
SAB 8032B-T40/100 For external ROM

- 8K × 8 ROM (SAB 8052B only)
- 256 × 8 RAM
- Four 8-Bit Ports, 32 I/O Lines
- Three 16-Bit Timer/Event Counters
- High-Performance Full-Duplex Serial Channel with Flexible Transmit/Receive Baud Rate Capability
- External Memory Expandable up to 128 Kbytes
- Boolean Processor
- Most Instructions Execute in 1 μs
- Multiply and Divide in 4 μs
- Six Interrupt Vectors, Two Priority Levels
- RAM Power-Down Supply
- P-DIP 40 and PLCC 44 Packages
- Full Backward Compatibility with SAB 8051/8031

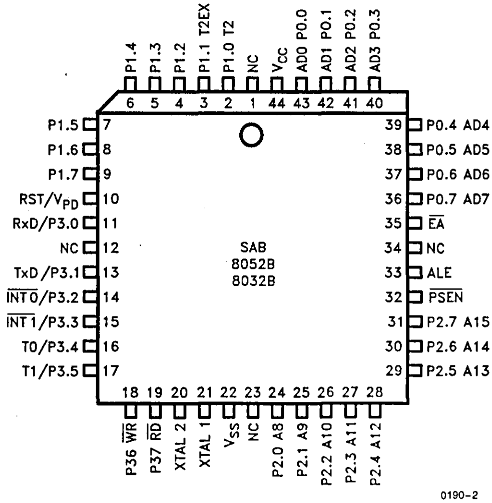
Pin Configurations

P-DIP 40



0190-1

PLCC 44



0190-2

The SAB 8052B/8032B for the two extended temperature ranges −40°C to +85°C and −40°C to +100°C is fully compatible with the standard SAB 8052B/8032B with respect to architecture, instruction set, and software portability.

The SAB 8052B/8032B is a stand-alone, high-performance single-chip microcontroller fabricated in +5V advanced N-channel, silicon gate Siemens MYMOS technology. Both extended temperature versions are available in a 40-pin plastic DIP (P-DIP 40) package: The SAB 8052B-T40/85 is also supplied in a 44-pin plastic leaded chip carrier (PLCC 44) package.

The SAB 8052B contains a non-volatile 8K × 8 read-only program memory; a volatile 256 × 8 read/write data memory; 32 I/O lines; three 16-bit timer/counters; a six-source, two-priority-level, nested interrupt structure; a serial I/O port for either multiprocessor communications, I/O expansion, or full-duplex UART; as well as on-chip oscillator and clock circuits. The SAB 8032B is identical with the SAB 8052B, except that it lacks the program memory.

For systems that require extra capability, the SAB 8052B can be expanded using standard TTL-compatible memories and the byte-oriented SAB 8080 and SAB 8085 peripherals.

Absolute Maximum Ratings*

Ambient Temperature under Bias
for T40/85 .... −40°C to +85°C
for T40/100 .... −40°C to +100°C

Storage Temperature (T_{stg}) ..... −65°C to +150°C

Voltage on any Pin
with Respect to Ground (V_{SS})...... −0.5 to +7V

Power Dissipation (P_D) 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = −40°C to +85°C for T40/85;
T_A = −40°C to +100°C for T40/100

Parameter	Symbol	Test Condition	Limit Values		Unit
			Min	Max	
Input Low Voltage	V_{IL}		−0.5	0.8	V
Input High Voltage (Except RST/V_{PD} and XTAL2)	V_{IH}		2.0	V_{CC} +0.5	V
Input High Voltage to RST/V_{PD} for Reset, XTAL2	V_{IH1}	XTAL1 to V_{SS}	2.5	V_{CC} +0.5	V
Power-Down Voltage to RST/V_{PD}	V_{PD}	V_{CC} = 0V	4.5	5.5	V
Output Low Voltage Ports 1, 2, 3	V_{OL}	I_{OL} = 1.6 mA		0.45	V
Output Low Voltage Port 0, ALE, $\overline{PSEN}$	V_{OL1}	I_{OL} = 3.2 mA		0.45	V
Output High Voltage Ports 1, 2, 3	V_{OH}	I_{OH} = −80 μA	2.4		V
Output High Voltage Port 0, ALE, $\overline{PSEN}$	V_{OH1}	I_{OH} = −400 μA	2.4		V
Logical 0 Input Current Ports 1, 2, 3	I_{IL}	V_{IL} = 0.45V		−500	μA
Logical 0 Input Current XTAL2	I_{IL2}	XTAL1 = V_{SS} V_{IL} = 0.45V	−2.5	−3.2	mA
Input High Current to RST/V_{PD} for Reset	I_{IH1}	V_{IN} = V_{CC} − 1.5V		500	μA
Input Leakage Current to Port 0, $\overline{EA}$	I_{LI}	0V < V_{IN} < V_{CC}		±10	μA
Power Supply Current	I_{CC}	All Outputs Disconnected		175	mA
Power-Down Current	I_{PD}	V_{CC} = 0V		15	mA
Capacitance of I/O Buffer	C_{IO}	f_c = 1 MHz		10	pF

AC Characteristics for T40/85: Refer to SAB 8051A/8031A Datasheet

AC Characteristics for T40/100

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = −40°C to +100°C;
(C_L for port 0, ALE and $\overline{\text{PSEN}}$ outputs = 100 pF; C_L for all other outputs = 80 pF)

Parameter	Symbol	Limit Values				Unit
		10 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1.2 MHz to 10 MHz		
		Min	Max	Min	Max	
Program Memory Characteristics						
ALE Pulse Width	t_{LHLL}	160		$2\,t_{CLCL} - 40$		ns
Address Setup to ALE	t_{AVLL}	70		$t_{CLCL} - 30$		ns
Address Hold after ALE	t_{LLAX1}	65		$t_{CLCL} - 35$		ns
ALE to Valid Instruction In	t_{LLIV}		300		$4\,t_{CLCL} - 100$	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	75		$t_{CLCL} - 25$		ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	265		$3\,t_{CLCL} - 35$		ns
$\overline{\text{PSEN}}$ to Valid Instruction In	t_{PLIV}		200		$3\,t_{CLCL} - 100$	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	t_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	t_{PXIZ}[1]		80		$t_{CLCL} - 20$	ns
Address Valid after $\overline{\text{PSEN}}$	t_{PXAV}[1]	92		$t_{CLCL} - 8$		ns
Address to Valid Instruction In	t_{AVIV}		385		$5\,t_{CLCL} - 115$	ns
Address Float to $\overline{\text{PSEN}}$	t_{AZPL}	0		0		ns
External Data Memory Characteristics						
$\overline{\text{RD}}$ Pulse Width	t_{RLRH}	500		$6\,t_{CLCL} - 100$		ns
$\overline{\text{WR}}$ Pulse Width	t_{WLWH}	500		$6\,t_{CLCL} - 100$		ns
Address Hold after ALE	t_{LLAX2}	165		$2\,t_{CLCL} - 35$		ns
$\overline{\text{RD}}$ to Valid Data In	t_{RLDV}		335		$5\,t_{CLCL} - 165$	ns
Data Hold after $\overline{\text{RD}}$	t_{RHDX}	0		0		ns
Data Float after $\overline{\text{RD}}$	t_{RHDZ}		130		$2\,t_{CLCL} - 70$	ns
ALE to Valid Data In	t_{LLDV}		650		$8\,t_{CLCL} - 150$	ns
Address to Valid Data In	t_{AVDV}		735		$9\,t_{CLCL} - 165$	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{LLWL}	250	350	$3\,t_{CLCL} - 50$	$3\,t_{CLCL} + 50$	ns
Address to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{AVWL}	270		$4\,t_{CLCL} - 130$		ns
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ High to ALE High	t_{WHLH}	60	140	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
Data Valid to $\overline{\text{WR}}$ Transition	t_{DVWX}	50		$t_{CLCL} - 50$		ns
Data Setup before $\overline{\text{WR}}$	t_{QVWH}	550		$4\,t_{CLCL} - 150$		ns
Data Hold after $\overline{\text{WR}}$	t_{WHQX}	50		$t_{CLCL} - 50$		ns
Address Float after $\overline{\text{RD}}$	t_{RLAZ}		0		0	ns

NOTE:
1. Interfacing the SAB 8052B to devices with float times up to 92 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

External Clock Drive XTAL2

Parameter	Symbol	Limit Values		Unit

		Variable Clock
		Freq. = 1.2 MHz to 12 MHz (T40/85)
		Freq. = 1.2 MHz to 10 MHz (T40/100)		

		Min	Max	
Oscillator Period T40/85 T40/100	t_{CLCL}	83.3 100	833.3	ns
High Time	t_{CHCX}	20	$t_{CLCL}-t_{CLCX}$	ns
Low Time	t_{CLCX}	20	$t_{CLCL}-t_{CHCX}$	ns
Rise Time	t_{CLCH}		20	ns
Fall Time	t_{TCHCL}		20	ns

Waveforms: Refer to SAB 8051A/8031A Datasheet

Ordering Information

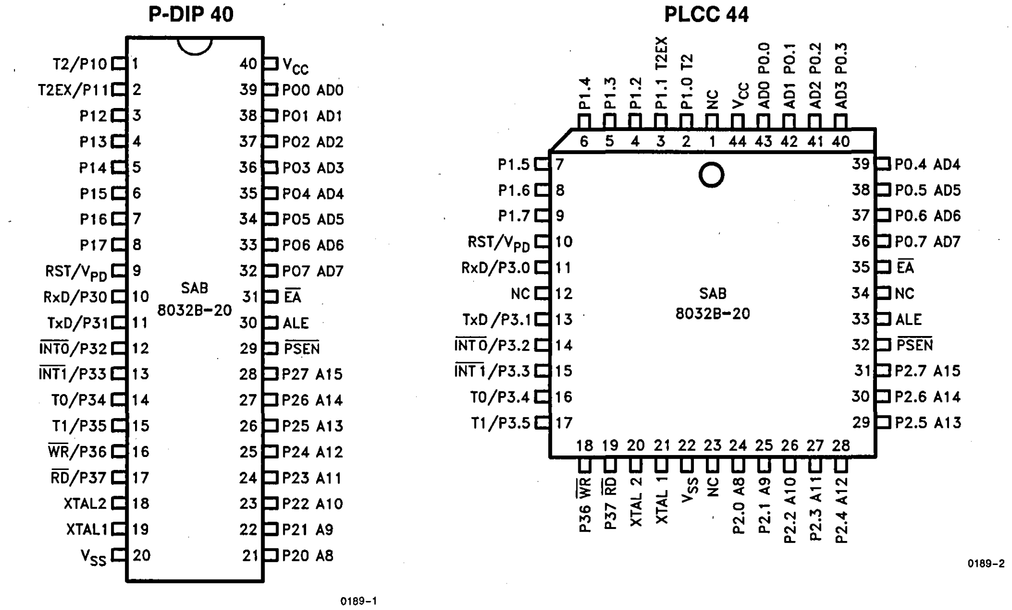
Type	Description
	8-Bit Single-Chip Microcontroller
SAB 8052B-P-T40/85	with Mask-Programmable ROM (P-DIP40)
SAB 8052B-P-T40/100	with Mask-Programmable ROM (P-DIP40)
SAB 8032B-P-T40/85	for External Memory (P-DIP40)
SAB 8032B-P-T40/100	for External Memory (P-DIP40)
SAB 8052B-N-T40/85	with Mask-Programmable ROM (PLCC44)
SAB 8032B-N-T40/85	for External Memory (PLCC44)

SAB 8032B-20
8-Bit Single-Chip Microcontroller

SAB 8032B-20 Microcontroller for external ROM

- SAB 8032B-20, 20 MHz Operation
- ROMless
- 256 × 8 Bytes RAM
- Four 8-Bit Ports, 32 I/O Lines
- Three 16-Bit Timer/Event Counters
- High-Performance Full-Duplex Serial Channel
- External Memory Expandable up to 128 Kbytes
- Compatible with SAB 8080/8085 Peripherals
- Boolean Processor
- Most Instructions Execute in: 600 ns (SAB 8032B-20)
- 2.4 μs Multiply and Divide
- P-DIP 40 and PLCC 44 Packages
- Full Backward Compatibility with SAB 8051A/8031A

Pin Configurations



The SAB 8032B-20 is a stand-alone, high-performance single-chip microcontroller fabricated in +5V advanced Siemens MYMOS (III) technology and supplied in a 40-pin plastic P-DIP or 44-pin plastic leaded chip carrier (PLCC 44) package. It is backwardly compatible with the SAB 8051A/8031A. It provides the hardware features, architectural enhancements and instructions that are necessary to make it a powerful and cost-effective controller for applications requiring up to 64 Kbytes of program memory and/or up to 64 Kbytes of data storage.

The SAB 8032-B contains no on-chip ROM; a volatile 256 × 8 bytes read/write data memory; 32 I/O lines; three 16-bit timer/counters; a five source, two-priority-level, nested interrupt structure; a serial I/O port for either multiprocessor communications, I/O expansion, or full-duplex UART; and on-chip oscillator and clock circuits.

Absolute Maximum Ratings*

Ambient Temperature under Bias ......0°C to 70°C

Storage Temperature .... −65°C to + 150°C

Voltage on Any Pin
with Respect to Ground (V_{SS}) .... −0.5V to +7V

Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics T_A = 0 to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Parameter	Symbol	Test Condition	Limit Values		Unit
			Min	Max	
Input Low Voltage	V_{IL}		−0.5	0.8	V
Input High Voltage (except RST/V_{PD} and XTAL2)	V_{IH}		2.0	V_{CC} + 0.5	V
Input High Voltage to RST/V_{PD} for Reset, XTAL2	V_{IH1}	XTAL1 to V_{SS}	2.5	V_{CC} + 0.5	V
Power Down Voltage to RST/V_{PD}	V_{PD}	V_{CC} = 0V	4.5	5.5	V
Output Low Voltage Ports 1, 2, 3	V_{OL}	I_{OL} = 1.6 mA		0.45	V
Output Low Voltage Port 0, ALE, $\overline{PSEN}$	V_{OL1}	I_{OL} = 3.2 mA		0.45	V
Output High Voltage Ports 1, 2, 3	V_{OH}	I_{OH} = −80 μA	2.4		V
Output High Voltage Port 0, ALE, $\overline{PSEN}$	V_{OH1}	I_{OH} = −400 μA	2.4		V
Logical 0 Input Current Ports 1, 2, 3	I_{IL}	V_{IL} = 0.45V		−500	μA
Logical 0 Input Current XTAL2	I_{IL2}	XTAL1 = V_{SS} V_{IL} = 0.45V		−3.2	mA
Input High Current to RST/V_{PD} for Reset	I_{IH1}	V_{IN} = V_{CC} − 1.5V		500	μA
Input Leakage Current to Port 0, $\overline{EA}$	I_{LI}	0 < V_{IN} < V_{CC}		±10	μA
Power Supply Current SAB 8032B/8052B SAB 8032B-16/8052B-16	I_{CC}	All Outputs Disconnected		175	mA
Power Down Current	I_{PD}	V_{CC} = 0V	—	15	mA
Capacitance of I/O Buffer	C_{IO}	f_C = 1 MHz		10	pF

A.C. Characteristics for SAB 8032B-20

T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V
(C_L for Port 0, ALE and $\overline{\text{PSEN}}$ Outputs = 100 pF; C_L for All Other Outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		20 MHz Clock		Variable Clock 1/t_{CLCL} = 1.2 MHz to 20 MHz		
		Min	Max	Min	Max	
ALE Pulse Width	t_{LHLL}	60		2 t_{CLCL} − 40		ns
Address Setup to ALE	t_{AVLL}	20		t_{CLCL} − 30		ns
Address Hold after ALE	t_{LLAX1}	20		t_{CLCL} − 30		ns
ALE to Valid Instruction In	t_{LLIV}		100		4 t_{CLCL} − 100	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	25		t_{CLCL} − 25		ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	115		3 t_{CLCL} − 35		ns
$\overline{\text{PSEN}}$ to Valid Instruction In	t_{PLIV}		75		3 t_{CLCL} − 75	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	t_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	t_{PXIZ}		40		t_{CLCL} − 10	ns
Address to Valid Instruction In	t_{AVIV}		190		5 t_{CLCL} − 60	ns
Address Float to $\overline{\text{PSEN}}$	t_{AZPL}	0		0		ns

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		20 MHz Clock		Variable Clock 1/t_{CLCL} = 1.2 MHz to 20 MHz		
		Min	Max	Min	Max	
$\overline{\text{RD}}$ Pulse Width	t_{RLRH}	200		6 t_{CLCL} − 100		ns
$\overline{\text{WR}}$ Pulse Width	t_{WLWH}	200		6 t_{CLCL} − 100		ns
Address Hold after ALE	$t_{LLAX\,2}$	70		2 t_{CLCL} − 35		ns
$\overline{\text{RD}}$ to Valid Data In	t_{RLDV}		100		5 t_{CLCL} − 150	ns
Data Hold after $\overline{\text{RD}}$	t_{RHDX}	0		0		ns
Data Float after $\overline{\text{RD}}$	t_{RHDZ}		40		2 t_{CLCL} − 60	ns
ALE to Valid Data In	t_{LLDV}		250		8 t_{CLCL} − 150	ns
Address to Valid Data In	t_{AVDV}		285		9 t_{CLCL} − 165	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{LLWL}	100	200	3 t_{CLCL} − 50	3 t_{CLCL} + 50	ns
Address to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{AVWL}	70		4 T_{CLCL} − 130		ns

External Data Memory Characteristics (Continued)

Parameter	Symbol	Limit Values				Unit
		20 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1.2 MHz to 20 MHz		
		Min	Max	Min	Max	
$\overline{WR}$ or $\overline{RD}$ High to ALE High	t_{WHLH}	20	80	$t_{CLCL} - 30$	$t_{CLCL} + 30$	ns
Data Valid to $\overline{WR}$ Transition	t_{QVWX}	5		$t_{CLCL} - 45$		ns
Data Hold after $\overline{WR}$	t_{WHQX}	10		$t_{CLCL} - 40$		ns
Address Float after $\overline{RD}$	t_{RLAZ}		0		0	ns

External Clock Drive XTAL2

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 1.2 MHz to 20 MHz		
		Min	Max	
Oscillator Period	t_{CLCL}	50	833.3	ns
High Time	t_{CHCX}	15	$t_{CLCL} - t_{CLCX}$	ns
Low Time	t_{CLCX}	15	$t_{CLCL} - t_{CHCX}$	ns
Rise Time	t_{CLCH}		15	ns
Fall Time	t_{CHCL}		15	ns

Waveforms: Refer to SAB 8051A/8031A Data Sheet

Ordering Information

Type	Description

SAB 8032B-20-N
SAB 8032B-20-P
	8-Bit Single-Chip Microcontroller
	for External Memory (PLCC-44)
	for External Memory (P-DIP-40)

SIEMENS

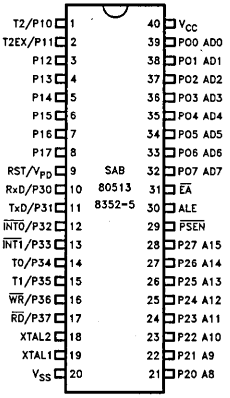
SAB 80513/80513-16
SAB 8352-5/8352-5-16
8-Bit Single-Chip Microcontroller

SAB 80513/80513-16	Microcontroller with 16K factory-mask programmable ROM, 12/16 MHz operation
SAB 8352-5/8352-5-16	Microcontroller with 32K factory-mask programmable ROM, 12/16 MHz operation

- 16K × 8 ROM (SAB 80513)
- 32K × 8 ROM (SAB 8352-5)
- 256 × 8 RAM
- Four 8-Bit I/O Ports
- Three 16-Bit Timer/Event Counters
- High-Performance Full-Duplex Serial Channel with Flexible Transmit/Receive Baud Rate Capability
- Six Interrupt Vectors, Two Priority Levels are Programmable
- Boolean Processor
- Most Instructions Execute in:
  1 μs (SAB 80513/80533)
  750 ns (SAB 80513-16/80533-16)
- 4 μs (3 μs) Multiply and Divide
- External Memory Expandable up to 128 Kbytes
- Fully Backward Compatible to SAB 8051A and SAB 8052B
- P-DIP 40 and PLCC 44 Packages
- Two temperature ranges available:
  0°C to +70°C
  −40°C to +85°C

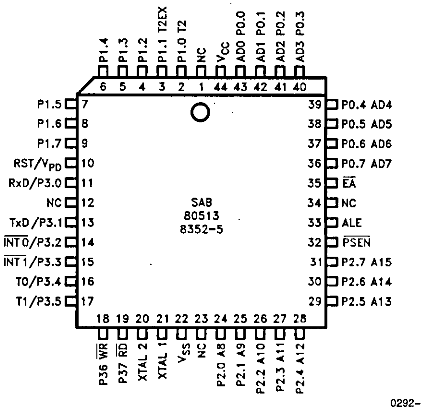
Pin Configurations

P-DIP 40



0292-1

PLCC 44

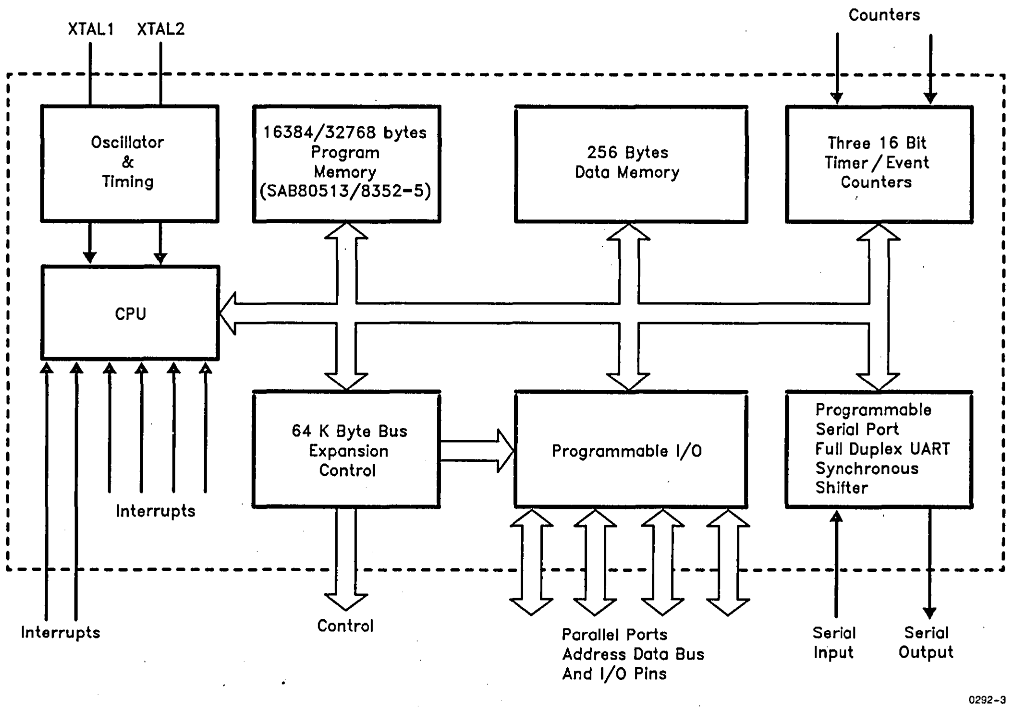


0292-2

The SAB 80513 and SAB 8352-5 are new members of the Siemens SAB 8051 family of 8-bit microcontrollers. It is fabricated in N-channel, silicon-gate Siemens MYMOS technology.

Pin Definitions and Functions: Refer to SAB 8052B/8032B

Block Diagram


XTAL1
XTAL2
Counters
Oscillator & Timing
16384/32768 bytes Program Memory (SAB80513/8352-5)
256 Bytes Data Memory
Three 16 Bit Timer / Event Counters
CPU
64 K Byte Bus Expansion Control
Programmable I/O
Programmable Serial Port Full Duplex UART Synchronous Shifter
Interrupts
Interrupts
Control
Parallel Ports Address Data Bus And I/O Pins
Serial Input
Serial Output
0292-3

The SAB 80513 and the SAB 8352-5 are stand-alone, high-performance single-chip microcontrollers based on the SAB 8051 architecture. They maintain all features of the SAB 8051A and SAB 8052B (including timer 2 of the SAB 8052B) and are thus fully compatible with both the SAB 8051A and SAB 8052B.

In addition, the SAB 80513 contains 16 Kbytes of on-chip ROM and the SAB 8352-5 contains 32 Kbytes of on-chip ROM, which makes them powerful and cost-effective controllers for applications requiring more ROM space.

Furthermore, they contain 256 byte RAM on-chip, four 8-bit ports, a powerful interrupt structure with six vectors and two programmable priority levels, a serial channel as well as on-chip oscillator and clock circuitry. Both parts are available in a 12 MHz and a 16 MHz crystal oscillator frequency version.

The SAB 80513 and SAB 8352-5 are supplied in a 40-pin dual-in-line package or a 44-pin plastic leaded chip carrier (PLCC 44) package.

The parts in addition to the standard temperatu range (0°C to +70°C) are also available in the tended temperature range (−40°C to +85°C) up 16 MHz crystal oscillator frequency.

Because of the different internal ROM size, the $\overline{EA}$ pin level for these two devices determines the appropriate program counter (PC) effective address: when $\overline{EA}$ pin is held at a high level, the SAB 80513 executes instructions from internal ROM when PC is less than 4000H, whereas the SAB 8352-5 executes instructions from internal ROM when PC is less than 8000H. When $\overline{EA}$ pin is held at a low level, the SAB 80513/8352-5 fetch all instructions from external program memory.

Absolute Maximum Ratings*

Ambient Temperature under Bias ...... 0°C to 70°C

Storage Temperature .... −65°C to +150°C

Voltage on Any Pin with Respect
to Ground (V_{SS}) .... −0.5V to +7V

Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

AC/DC Characteristics for SAB 80513/80513-16 and SAB 8352-5/8352-5-16 are same as that of the SAB 8052B/8052B-16.

Ordering Information

Type	Description
	8-Bit Single-Chip-Microcontroller
SAB 80513-P	with 16K Mask-Programmable ROM (P-DIP-40)
SAB 80513-N	with 16K Mask-Programmable ROM (PLCC-44)
SAB 80513-16-P	with 16K Mask-Programmable ROM (P-DIP-40), 16 MHz
SAB 80513-16-N	with 16K Mask-Programmable ROM (PLCC-44), 16 MHz
SAB 80513-P-T40/85	with 16K Mask-Programmable ROM (P-DIP-40), Ext. Temp.
SAB 80513-N-T40/85	with 16K Mask-Programmable ROM (PLCC-44), Ext. Temp.
SAB 80513-16-P-T40/85	with 16K Mask-Programmable ROM, 16 MHz (P-DIP-40), Ext. Temp.
SAB 80513-16-N-T40/85	with 16K Mask-Programmable ROM, 16 MHz (PLCC-44), Ext. Temp.
SAB 8352-5P	with 32K Mask-Programmable ROM (P-DIP-40)
SAB 8352-5N	with 32K Mask-Programmable ROM (PLCC-44)
SAB 8352-5P-16	with 32K Mask-Programmable ROM (P-DIP-40), 16 MHz
SAB 8352-5N-16	with 32K Mask-Programmable ROM (PLCC-44), 16 MHz
SAB 8352-5P-T40/85	with 32K Mask-Programmable ROM (P-DIP-40), Ext. Temp.
SAB 8352-5N-T40/85	with 32K Mask-Programmable ROM (PLCC-44), Ext. Temp.
SAB 8352-5P-16-T40/85	with 32K Mask-Programmable ROM, 16 MHz (P-DIP-40), Ext. Temp.
SAB 8352-5N-16-T40/85	with 32K Mask-Programmable ROM, 16 MHz (PLCC-44), Ext. Temp.

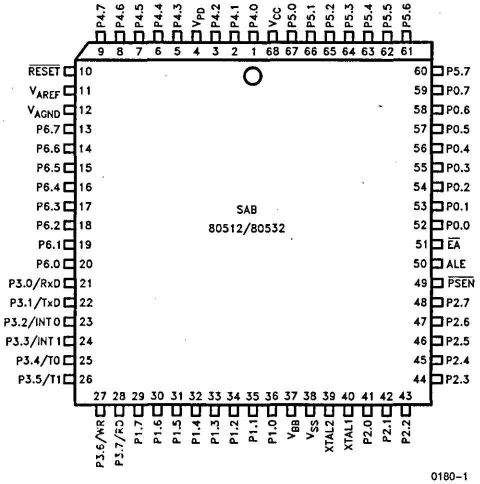
SIEMENS

SAB 80512/80532 8-Bit Single-Chip Microcontroller

SAB 80512	Microcontroller with factory-mask programmable ROM
SAB 80532	Microcontroller for external ROM
SAB 80512-T40/85	Extended temperature range: −40°C to +85°C
SAB 80532-T40/85	Extended temperature range: −40°C to +85°C

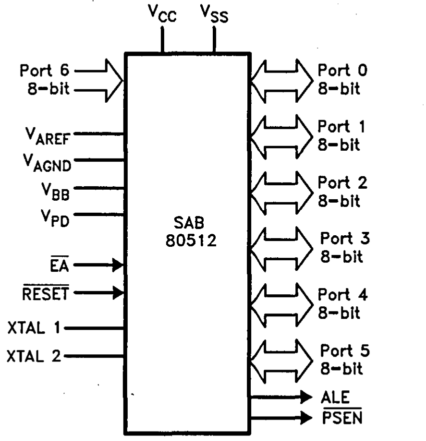
- 4K × 8 ROM (SAB 80512 only)
- 128 × 8 RAM
- Backwardly Compatible with SAB 8051A/ 8031A
- Seven 8-Bit Ports
- Two 16-Bit Timers/Event Counters
- High-Performance Full Duplex Serial Channel with Own Baud Rate Generator
- 8-Bit A/D Converter with Eight Multiplexed Inputs, Reference Voltages Externally Adjustable
- Six Interrupt Sources (2 External, 4 Internal), Two Priority Levels Programmable
- Boolean Processor
- 1 μs Instruction Cycle Time (at 12 MHz Osc. Frequency)
- 4 μs Multiply and Divide (at 12 MHz Osc. Frequency)
- External Program and Data Memory Expandable up to 64 Kbyte Each
- PLCC 68 Package

Pin Configuration



0180-1

Logic Symbol



0180-2

The SAB 80512/80532 is a new member of the Siemens SAB 8051 family of 8-bit microcontrollers. Maintaining all features of the SAB 8051A/8031A, it is fully backwardly compatible with the SAB 8051A/8031A. Furthermore the SAB 80512/80532 incorporates several enhancements that significantly increase design flexibility and cost effectiveness. In addition to the SAB 8051A/8031A the SAB 80512/80532 contains an 8-bit A/D converter with 8 multiplexed inputs (these inputs can also be used as digital inputs), an own baud rate generator for the serial interface and two more I/O ports. The SAB 80532 is identical with the SAB 80512, except that it lacks the on-chip ROM.

The SAB 80512/80532 is fabricated in +5V advanced N-channel, silicon gate MYMOS technology of Siemens and supplied in a PLCC 68 package. For the industrial temperature range −40°C to +85°C, the SAB 80512/80532-T40/85 is available.

Pin Definitions and Functions

Pin	Symbol	Input (I) Output (O)	Function
1-3, 5-9	P4.0–P4.7	I/O	Port 4 is an 8-bit quasi-bidirectional I/O port with internal pullup resistors. Port 4 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 4 pins being externally pulled low will source current (I_{IL}, on the DC characteristics) because of the internal pullup resistors.
4	V_{PD}		Power down supply voltage. If V_{PD} is held within its specifications while V_{CC} drops below the specification, V_{PD} will provide standby power to 40 byte of internal RAM (addr. 58H to 7FH). During normal operation of the SAB 80512, the RAM's current is supplied by V_{CC}, when V_{PD} is low.
10	$\overline{\text{RESET}}$	I	A low level on this pin for the duration of two machine cycles while the oscillator is running resets the SAB 80512. A small internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
11	V_{AREF}		Reference voltage for the A/D converter.
12	V_{AGND}		Reference ground for the A/D converter.
13–20	P6.7–P6.0	I	Port 6, 8-bit unidirectional input port. Port pins can be used for digital input if voltage levels meet the specified input high/low voltages and for the eight multiplexed analog inputs of the A/D converter, simultaneously.
21–28	P3.0–P3.7	I/O	Port 3 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 3 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs port 3 pins being externally pulled low will source current (I_{IL}, on the DC characteristics) because of the internal pullup resistors. It also contains the interrupt, timer, serial port and external memory strobe pins that are used by various options. The output latch corresponding

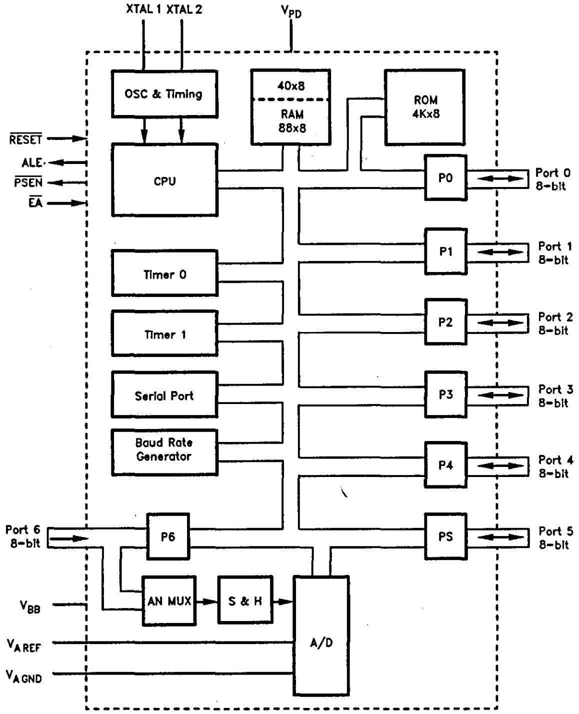
Pin Definitions and Functions (Continued)

Pin	Symbol	Input (I) Output (O)	Function
21-28	P3.0-P3.7	I/O	to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: —RxD (P3.0): serial port's receiver data input (asynchronous) or data input/output (synchronous) —TxD (P3.1): serial port's transmitter data output (asynchronous) or clock output (synchronous) —$\overline{\text{INT0}}$ (P3.2): interrupt 0 input/timer 0 gate control input —$\overline{\text{INT1}}$ (P3.3): interrupt 1 input/timer 1 gate control —T0 (P3.4): counter 0 input —T1 (P3.5): counter 1 input —$\overline{\text{WR}}$ (P3.6): the write control signal latches the data byte from port 0 into the external data memory —$\overline{\text{RD}}$ (P3.7): the read control signal enables the external data memory to port 0
29-36	P1.7-P1.0	I/O	Port 1 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 1 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs port 1 pins being externally pulled low will source current (I_{IL}, on the DC characteristics) because of the internal pullup resistors. The port is also used for the low order address byte during program verification.
37	V_{BB}		Substrate pin. Must be connected to V_{SS} with a capacitor (47 nF to 100 nF) for proper operation of the A/D converter.
39 40	XTAL2 XTAL1		XTAL2 Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL2 should be driven, while XTAL1 is pulled low. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times specified in the AC characteristics must be observed: XTAL1 Input to the inverting oscillator amplifier. Required when a crystal or ceramic resonator is used.
41-48	P2.0-P2.7	I/O	Port 2 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 2 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs port 2 pins being externally pulled low will source current (I_{IL}, on the DC characteristics) because of the internal pullup resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX@DPTR). In this application it uses strong internal pullup resistors when issuing 1s. During accesses to external data memory that use 8-bit addresses (MOVX@Ri), port 2 issues the contents of the P2 special function register.
49	$\overline{\text{PSEN}}$	O	The program store enable output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. Remains high during internal program execution.
50	ALE	O	Provides address latch enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.

Pin Definitions and Functions (Continued)

Pin	Symbol	Input (I) Output (O)	Function
51	$\overline{EA}$	I	When held at a TTL high level, the SAB 80512 executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the SAB 80512 fetches all instructions from external program memory. For the SAB 80532 this pin must be tied low.
52–59	P0.0–P0.7	I/O	Port 0 is an 8-bit open drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low order address and data bus during accesses to external program and data memory. In this application it uses strong internal pullup resistors when issuing 1s. Port 0 also outputs the code bytes during program verification. External pullup resistors are required during program verification.
60–67	P5.7–P5.0	I/O	Port 5 is an 8-bit bidirectional I/O Port with internal pullup resistors. Port 5 pins that have 1s written to them are pulled high by the internal pullup registers, and in that state can be used as inputs. As inputs Port 5 pins being externally pulled low will source current (I_{IL}, on the DC Characteristics) because of the internal pullup resistors.
68	V_{CC}		Supply voltage during normal operation and program verification.
38	V_{SS}		Ground (0V)

Block Diagram



0180-3

Functional Description

The SAB 80512/80532 is based on the architecture of the SAB 8051 microcontroller family. The SAB 80512 includes all features of the SAB 8051 and additionally offers peripheral extensions in three items:

— bit A/D converter with adjustable reference voltages

— two more ports

— dedicated baud rate generator

Different to the SAB 8051 is the inverted reset-input and the RAM power-down supply by a special pin (V_{PD}), which supplies 40 byte with a typical current of 2 mA. Beside the upward compatibilty to the SAB 8051 (all SAB 8051 software runs on the SAB 80512 without any changes) the SAB 80512 is also downward compatible to the SAB 80515. The SAB 80512 is packed into the PLCC 68 package and has the same pinning as the SAB 80515.

A/D Converter

The 8-bit A/D converter of the SAB 80512 has 8 multiplexed analog inputs and is using the successive approximation method. The sampling of an analog signal takes 5 machine cycles, the total conversion time is 15 machine cycles (15 μs at 12 MHz oscillator frequency). Conversion can be programmed to be single or continuous, at the end of a conversion an interrupt can be generated. The SAB 80512 provides variable external reference voltages V_{AGND} and V_{AREF} adjustable in a wide range. A compressed reference voltage range allows to increase the resolution of the converted analog input.

The lower reference voltage (V_{AGND}) can be varied within V_{SS} − 0.2V and 4V, the higher (V_{AREF}) within 1V and V_{CC} +5%. For proper operation of the A/D converter a minimum of 1V difference is required between the external voltages:

$$(V_{SS} - 0.2V) \leq V_{AGND} \leq (V_{AREF} - 1V)$$
$$(V_{AGND} + 1V) \leq V_{AREF} \leq (V_{CC} + 5\%)$$

Special Function Register

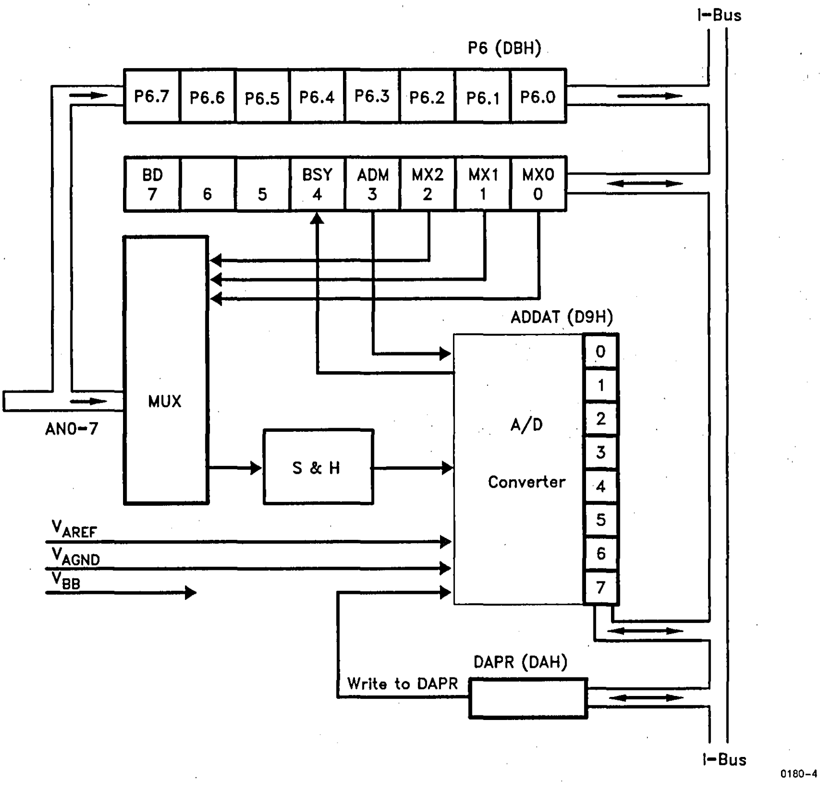
All registers, except the program counter and the four 8-register banks, reside in the special function register area. The 28 special function registers (SFRs) include arithmetic registers, pointers, and registers that provide an interface between the CPU and the on-chip peripheral functions. There are also 128 directly addressable bits within the SFR area.

I/O Ports

The SAB 80512 has six 8-bit I/O ports and one 8-bit input port. Port 0 is an open-drain bidirectional I/O port, while ports 1 to 5 are quasi-bidirectional I/O ports with internal pullups. That means, when configured as inputs, ports 1 to 5 will pull high and will source current when externally pulled low. Port 0 will float when configured as input. Port 6 is an input port only and can be used as digital input port, if the values meet the specified high/low voltages and as analog input for the A/D-converter.

Port 0 and port 2 can be used to expand the program and data memory externally. During an access to external memory, port 0 emits the low-order address byte. In this function, port 0 is not an open-drain port, but uses a strong internal pullup FET.

Block Diagram of A/D Converter



Special Function Registers

Address	Symbol	Name	Bit-Addressable
80H	P0	Port 0 Register	Yes
81H	SP	Stack Pointer	
82H	DPL	Data Pointer, Low-Byte	
83H	DPH	Data Pointer, High-Byte	
87H	PCON	Power Control Register	
88H	TCON	Timer Control Register	Yes
89H	TMOD	Timer Mode Register	
8AH	TL0	Timer 0, Low-Byte	
8BH	TL1	Timer 1, Low-Byte	
8CH	TH0	Timer 0, High-Byte	
8DH	TH1	Timer 1, High-Byte	
90H	P1	Port 1 Register	Yes
98H	SCON	Serial Port Control Register	Yes
99H	SBUF	Serial Port Buffer Register	
0A0H	P2	Port 2 Register	Yes
0A8H	IE	Interrupt Enable Register	Yes
0B0H	P3	Port 3 Register	Yes
0B8H	IP	Interrupt Priority Register	Yes
0C0H	IRCON	Interrupt Request Control	Yes
0D0H	PSW	Program Status Word Register	Yes
0D8H	ADCON	A/D Converter Control Register	Yes
0D9H	ADDAT	A/D Converter Data Register	
0DAH	DAPR	D/A Converter Start Register	
0DBH	P6	Port 6 Register	
0E0H	ACC	Accumulator Register	Yes
0E8H	P4	Port 4 Register	Yes
0F0H	B	B-Register	Yes
0F8H	P5	Port 5 Register	Yes

Absolute Maximum Ratings*

Temperature under Bias
for the SAB 80512/80532 ....0°C to +70°C
for the SAB 80512/
80532-T40/85 ....−40°C to +85°C

Storage Temperature ....−65°C to +150°C

Voltage on any Pin with
Respect to Ground (v_{SS}) ....−0.5V to +7V

Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = 0 to 70°C for SAB 80512/80532; T_A = −40°C to +85°C for SAB 80512/80532-T40/85

Parameter	Symbol	Test Conditions	Limit Values		Unit
			Min	Max	
Input Low Voltage	V_{IL}		−0.5	0.8	V
Input High Voltage (Except $\overline{\text{RESET}}$ and XTAL2)	V_{IH}		2.0	V_{CC}+ 0,5	V
Input High Voltage to XTAL2	V_{IH1}	XTAL1 to V_{SS}	2.5	V_{CC} + 0.5	V
Input High Voltage to $\overline{\text{RESET}}$	V_{IH2}		3.0		V
Power-Down Voltage	V_{PD}	V_{CC} = 0V	3	5.5	V
Output Low Voltage, Ports 1, 2, 3, 4, 5	V_{OL}	I_{OL} = 1.6 mA		0.45	V
Output Low Voltage, Port 0, ALE, $\overline{\text{PSEN}}$	V_{OL1}	I_{OL} = 3.2 mA		0.45	V
Output High Voltage, Ports 1, 2, 3, 4, 5	V_{OH}	I_{OH} = −80 μA	2.4		V
Output High Voltage, Port 0, ALE, $\overline{\text{PSEN}}$	V_{OH1}	I_{OH} = −400 μA	2.4		V
Logic 0 Input Current, Ports 1, 2, 3, 4, 5	I_{IL}	V_{IL} = 0.45V		−500	μA
Logic 0 Input Current, XTAL2	I_{IL2}	XTAL = V_{SS}, V_{IL} = 0.45V		−2.5	mA
Input Low Current to $\overline{\text{RESET}}$ for Reset	I_{IL3}	V_{IL} = 0.45V		−500	μA
Input Leakage Current to Port 0, $\overline{\text{EA}}$	I_{LI}	0V < V_{IN} < V_{CC}		±10	μA
Power Supply Current SAB 80512/80532 SAB 80512/80532-T40/85	I_{CC}	All Outputs Disconnected		175	mA
Power-Down Current	I_{PD}	V_{CC} = 0V		3	mA
Capacitance of I/O Buffer	C_{IO}	f_c = 1 MHz		10	pF

A/D Converter Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; (V_{SS} − 0.2V) ≤ V_{AGND} ≤ (V_{AREF} − 1V); (V_{AGND} + 1V) ≤ V_{AREF} ≤ (V_{CC} + 5%); T_A = 0°C to 70°C for SAB 80512/80532; T_A = −40°C to +85°C for SAB 80512/80532-T40/85

Parameter	Symbol	Test Conditions	Limit Values			Unit
			Min	Typ	Max	
Analog Input Voltage	V_{AINPUT}		V_{AGND} −0.2		V_{AREF} +0.2	V
Analog Input Capacitance(1)	C_I			25	70	pF
Load Time	t_L				2 t_{CY}	µs
Sample Time (Incl. Load Time)	t_S				5 t_{CY}	µs
Conversion Time (Incl. Sample Time)	t_C				15 t_{CY}	µs
Differential Non-Linearity	DNLE	V_{AREF} = V_{CC} V_{AGND} = V_{SS}		±¼	±½	LSB
Integral Non-Linearity	INLE			±¼	±½	
Offset Error				±¼	±½	
Gain Error				±¼	±½	
Total Unadjusted Error	TUE			±¼	±½	
V_{AREF} Supply Current(2)	I_{REF}				5	mA

NOTES:

1. The internal resistance of the analog source must be low enough to assure full loading of the sample capacitance (C_1) during load time (t_L). After charging of the internal capacitance (C_I) in the load time (t_L) the analog input must be held constant for the rest of the sample time (t_S).
2. The differential impedance r_D of the analog reference voltage source must be less than 1 kΩ at reference supply voltage.

AC Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = 0°C to 70°C for SAB 80512/80532; T_A = −40°C to +85°C for SAB 80512/80532-T40/85; (C_L for Port 0, ALE and $\overline{\text{PSEN}}$ Outputs = 100 pF; C_L for All Outputs = 80 pF)

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock 1/t_{CLCL} = 1.2 MHz to 12 MHz		
		Min	Max	Min	Max	
Program Memory Charactistics						
Cycle Time	t_{CY}	1000		12 t_{CLCL}		ns
ALE Pulse Width	t_{LHLL}	127		2 t_{CLCL} − 40		ns
Address Setup to ALE	t_{AVLL}	53		t_{CLCL} − 30		ns
Address Hold after ALE	t_{LLAX1}	48		t_{CLCL} − 35		ns
Address to Valid Instr In	t_{LLIV}		233		4 t_{CLCL} − 100	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	58		t_{CLCL} − 25		ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	215		3 t_{CLCL} − 35		ns
$\overline{\text{PSEN}}$ to Valid Instr In	t_{PLIV}		150		3 t_{CLCL} − 100	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	t_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	t_{PXIZ}*		63		t_{CLCL} − 20	ns
Address Valid after $\overline{\text{PSEN}}$	t_{PXAV}*	75		t_{CLCL} − 8		ns
Address to Valid Instr In	t_{AVIV}		302		5 t_{CLCL} − 115	ns
Address Float to $\overline{\text{PSEN}}$	t_{AZPL}	0		0		ns
External Data Memory Characteristics						
$\overline{\text{RD}}$ Pulse Width	t_{RLRH}	400		6 t_{CLCL} − 100		ns
$\overline{\text{WR}}$ Pulse Width	t_{WLWH}	400		6 t_{CLCL} − 100		ns
Address Hold after ALE	t_{LLAX2}	132		2 t_{CLCL} − 35		ns
$\overline{\text{RD}}$ to Valid Data In	t_{RLDV}		250		5 t_{CLCL} − 165	ns
Data Hold after $\overline{\text{RD}}$	t_{RHDX}	0		0		ns
Data Float after $\overline{\text{RD}}$	t_{RHDZ}		97		2 t_{CLCL} − 70	ns
ALE to Valid Data In	t_{LLDV}		517		8 t_{CLCL} − 150	ns
Address to Valid Data In	t_{AVDV}		585		9 t_{CLCL} − 165	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{LLWL}	200	300	3 t_{CLCL} − 50	3 t_{CLCL} + 50	ns
Address to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{AVWL}	203		4 t_{CLCL} − 130		ns
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ High to ALE High	t_{WHLH}	43	123	t_{CLCL} − 40	t_{CLCL} + 40	ns
Data Valid to $\overline{\text{WR}}$ Transition	t_{QVWX}	33		t_{CLCL} − 50		ns
Data Setup before $\overline{\text{WR}}$	t_{QVWH}	433		7 t_{CLCL} − 150		ns
Data Hold after $\overline{\text{WR}}$	t_{WHQX}	33		t_{CLCL} − 50		ns
Address Float after $\overline{\text{RD}}$	t_{RLAZ}		0		0	ns

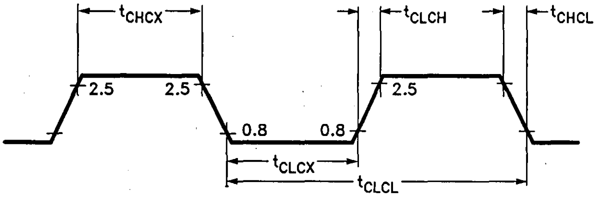
* Interfacing the SAB 80512 to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

AC Characteristics (Continued)

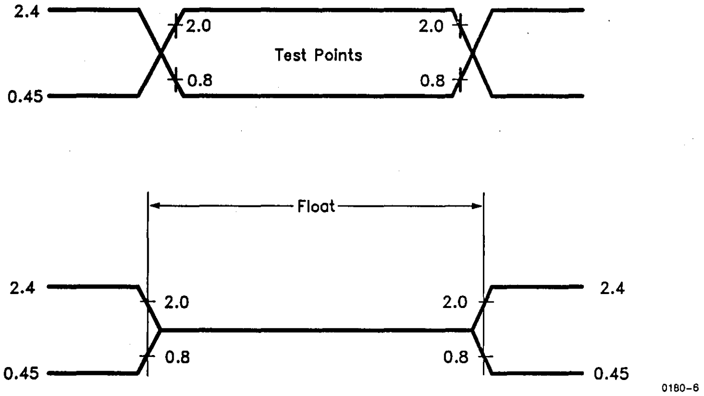
External Clock Drive XTAL2

Parameter	Symbol	Limit Values Variable Clock Freq. = 1.2 MHz to 12 MHz Min	 Max	Unit
Oscillator Period	t_{CLCL}	83.3	833.3	ns
High Time	t_{CHCX}	20	t_{CLCL}-t_{CLCX}	ns
Low Time	t_{CLCX}	20	t_{CLCL}-t_{CHCX}	ns
Rise Time	t_{CLCH}		20	ns
Fall Time	t_{CHCL}		20	ns

External Clock Cycle



AC Testing Input, Output, Float Waveforms

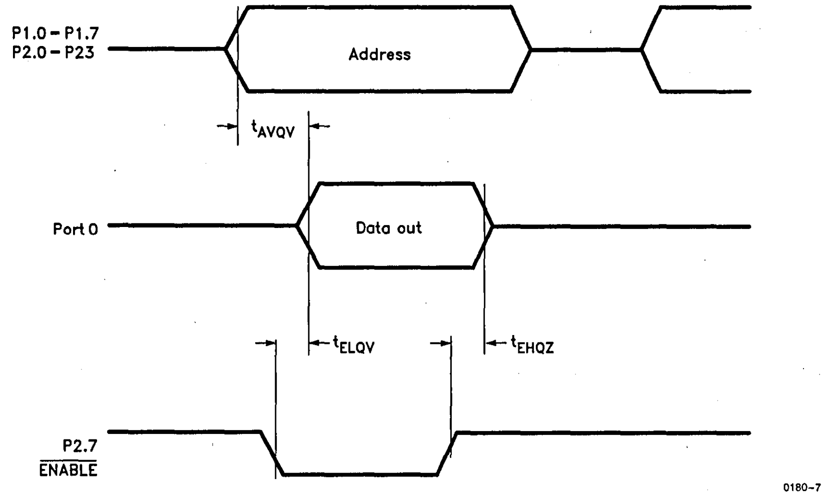


AC Testing inputs are driven at 2.4V for a logic "1" and at 0.45V for a logic "0".
Timing measurements are made at 2.0V for a logic "1" and at 0.8V for a logic "0".
For timing purposes, the float state is defined as the point where a P0 pin sinks 3.2 mA or sources 400 μA at voltage test levels.

ROM Verification Characteristics $T_A = 25°C; V_{CC} = 5V \pm 10\%; V_{SS} = 0V$

Parameter	Symbol	Limit Values		Unit
		Min	Max	
Address to Valid Data	t_{AVQV}		48 t_{CLCL}	ns
ENABLE to Valid Data	t_{ELQV}		48 t_{CLCL}	ns
Data Float after ENABLE	t_{EHQZ}	0	48 t_{CLCL}	ns
Oscillator Frequency	$1/t_{CLCL}$	4	6	MHz

ROM Verification

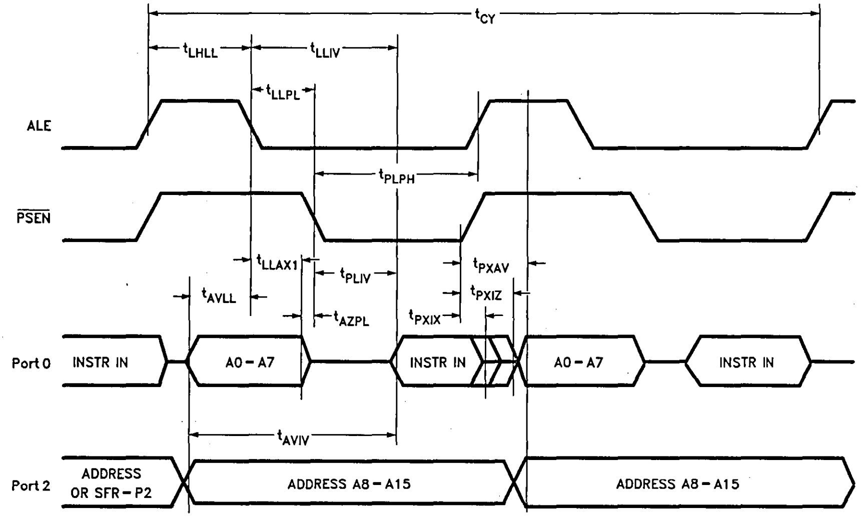


Address: P1.0–P1.7 = A0–A7
P2.0–P2.3 = A8–A11
Data: Port 0 = D0–D7

Inputs: P2.4–P2.6, $\overline{\text{PSEN}}$ = V_{SS}
ALE, $\overline{\text{EA}}$ = V_{IH}
$\overline{\text{RESET}}$ = V_{IL}

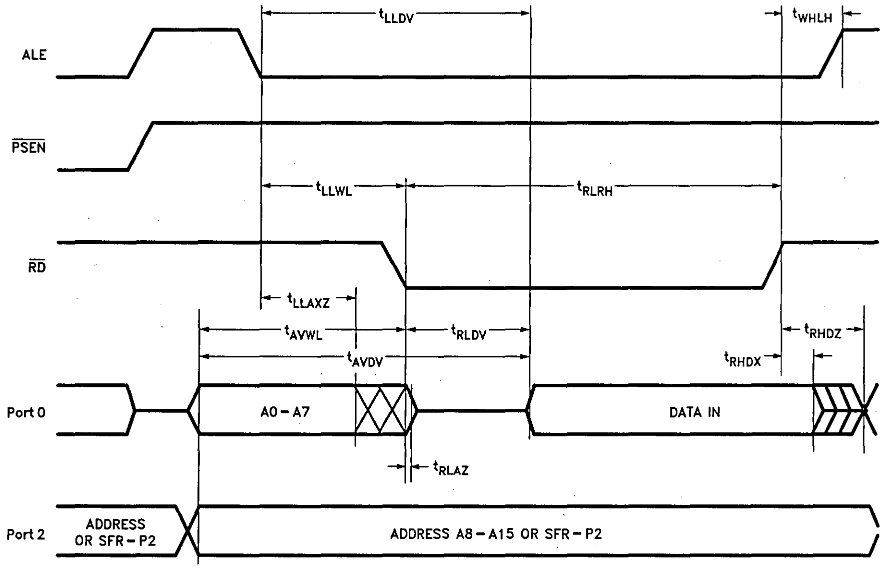
Waveforms

Program Memory Read Cycle



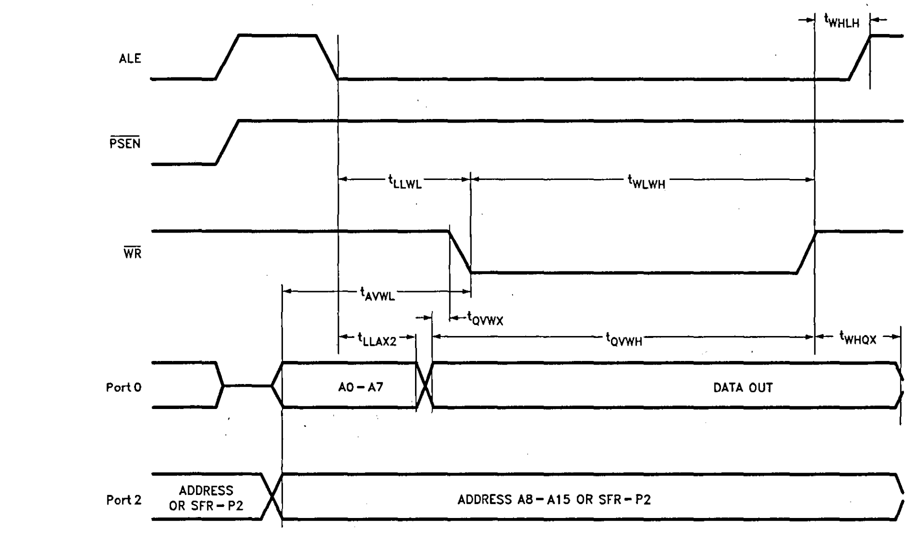
0180–8

Data Memory Read Cycle



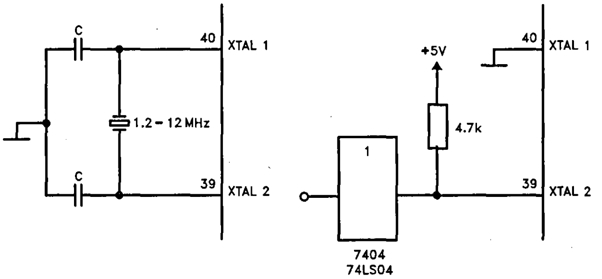
0180–9

Data Memory Write Cycle



0180-10

Recommended Oscillator Circuits



0180-11

C = 30 pF ± 10 pF

Crystal Oscillator Mode

Driving from External Source

Ordering Information

Type	Function
SAB 80512	8-Bit Single-Chip Microcontroller with ROM
SAB 80532	8-Bit Single-Chip Microcontroller for External ROM
SAB 80512-T40/85	Like SAB 80512 but for −40°C to +85°C
SAB 80532-T40/85	Like SAB 80532 but for −40°C to +85°C

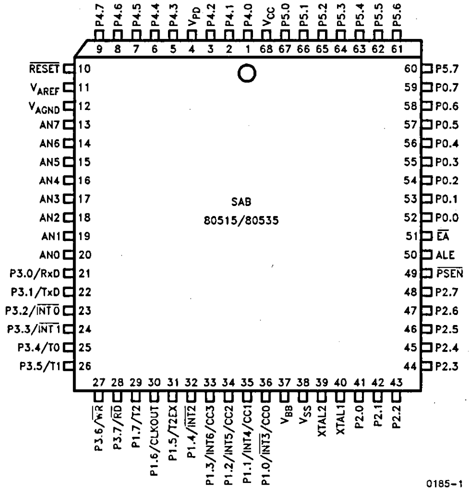
SIEMENS

SAB 80515/80535
8-Bit Single-Chip Microcontroller

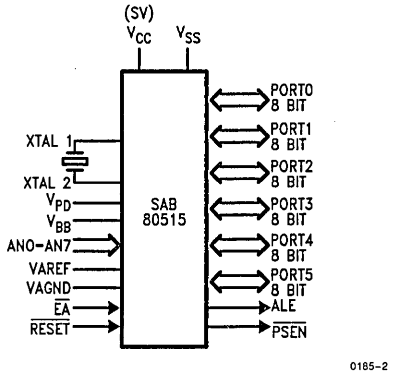
SAB 80515 Microcontroller with Factory-Mask Programmable ROM, 12 MHz Operation
SAB 80535 Microcontroller for External ROM, 12 MHz Operation

- 8K x 8 ROM (SAB 80515 Only)
- 256 x 8 RAM
- Six 8-bit ports, 48 I/O Lines
- Three 16-bit Timer/Event Counters
- PTRA with Highly Flexible Reload, Capture, Compare Capabilities
- High-Performance Full-Duplex Serial Channel
- Twelve Interrupt Vectors, Four Priority Levels
- 8-bit A/D Converter with 8 Multiplexed Analog Inputs and Programmable Internal Reference Voltages
- 16-bit Watchdog Timer
- V_{PD} Provides Standby Current for 40 Bytes of RAM
- Boolean Processor
- 256 Bit-Addressable Locations
- Most Instructions Execute in: 1 μs
- 4 μs Multiply and Divide
- External Memory Expandable to 128 Kbyte
- Backward-Compatible with SAB 8051A
- 68-Pin Plastic Leaded Chip Carrier Package (PLCC 68)

Pin Configuration



Logic Symbol



The SAB 80515/80535 is a member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is fabricated in +5V advanced N-channel, silicon-gate Siemens MYMOS technology and supplied in a 68-pin PLCC package. The SAB 80515/80535 is a standalone, high-performance single-chip microcontroller based on the SAB 8051 architecture. While maintaining all the SAB 8051 operating characteristics, the SAB 80515/80535 incorporates several enhancements which significantly increase design flexibility and overall system performance. The SAB 80535 is identical with the SAB 80515 except that it lacks the program memory.

Pin Definitions and Functions

Pin	Symbol	Input (I) Output (O)	Function
1–3, 5–9	P4.0–P4.7	I/O	Port 4 is an 8-bit quasi-bidirectional I/O port. Port 4 can sink/source 4 LS-TTL loads.
4	V_{PD}		Power down supply. If V_{PD} is held within its specs while V_{CC} drops below specs, V_{PD} will provide standby power to 40 byte of the internal RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC}.
10	$\overline{\text{RESET}}$	I	A low level on this pin for the duration of two machine cycles while the oscillator is running resets the SAB 80515. A small internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
11	V_{AREF}		Reference voltage for the A/D converter
12	V_{AGND}		Reference ground for the A/D converter
13–20	AN7–AN0	I	Multiplexed analog inputs
21–28	P3.0–P3.7	I/O	Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source 4 LS-TTL loads. The secondary functions are assigned to the pins of port 3, as follows: — RxD (P3.0): serial port's receiver data input (asynchronous) or data input/output (synchronous) — TxD (P3.1): serial port's transmitter data output (asynchronous) or clock output (synchronous) — $\overline{\text{INT0}}$ (P3.2): interrupt 0 input/timer 0 gate control input — $\overline{\text{INT1}}$ (P3.3): interrupt 1 input/timer 1 gate control input — T0 (P3.4): counter 0 input — T1 (P3.5): counter 1 input — $\overline{\text{WR}}$ (P3.6): the write control signal latches the data byte from port 0 into the external data memory — $\overline{\text{RD}}$ (P3.7): the read control signal enables the external data memory to port 0

Pin Definitions and Functions (Continued)

Pin	Symbol	Input (I) Output (O)	Function
29–36	P1.7–P1.0	I/O	Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. It also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output latch must be programmed to a one (1) for that function to operate (except when used for the compare functions). Port 1 can sink/source 4 LS-TTL loads. The secondary functions are assigned to the port 1 pins, as follows: — $\overline{\text{INT3}}$/CC0 (P1.0): interrupt 3 input/compare 0 output/ capture 0 input — INT4/CC1 (P1.1): interrupt 4 input/compare 1 output/ capture 1 input — INT5/CC2 (P1.2): interrupt 5 input/compare 2 output/ capture 2 input — INT6/CC3 (P1.3): interrupt 6 input/compare 3 output/ capture 3 input — $\overline{\text{INT2}}$ (P1.4): interrupt 2 input — T2EX (P1.5): timer 2 external reload trigger input — CLKOUT (P1.6): system clock output — T2 (P1.7): counter 2 input
37	V_{BB}		Substrate pin. Must be connected to V_{SS} through a capacitor (47 nF to 100 nF) for proper operation of the A/D converter.
39	XTAL2		XTAL2 is the output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal, ceramic resonator, or external source can be used.
40	XTAL1		XTAL1 is the input to the oscillator's high gain amplifier. Required when a crystal or ceramic resonator is used. Connect to V_{SS} when external source is used on XTAL2.
41–48	P2.0–P2.7	I/O	Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source 4 LS-TTL loads.
49	$\overline{\text{PSEN}}$	O	The program store enable output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. Remains high during internal program execution.
50	ALE	O	Provides address latch enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
51	$\overline{\text{EA}}$	I	When held at a TTL high level, the SAB 80515 executes instructions from the internal ROM when the PC is less than 8192. When held at a TTL low level, the SAB 80515 fetches all instructions from external program memory. For the SAB 80535 this pin must be tied low.

Pin Definitions and Functions (Continued)

Pin	Symbol	Input (I) Output (O)	Function
52–59	P0.0–P0.7	I/O	Port 0 is an 8-bit open-drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source 8 LS-TTL loads.
60–67	P5.7–P5.0	I/O	Port 5 is an 8-bit quasi-bidirectional I/O port. Port 5 can sink/source 4 LS-TTL loads.
68	V_{CC}		POWER SUPPLY (+5V power supply during normal operation and program verification)
38	V_{SS}		GROUND (0V)

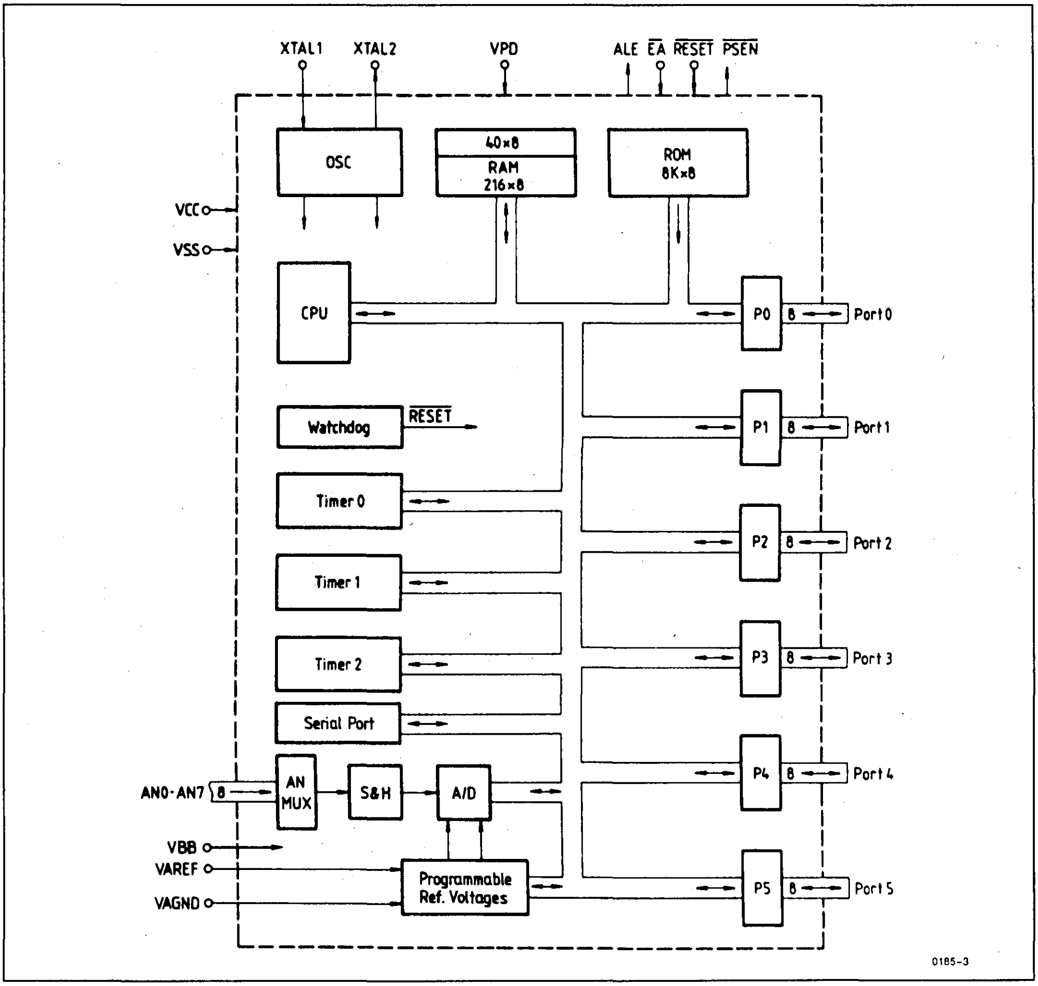


Figure 1. Block Diagram

Functional Description

The members of the SAB 80515 family of microcontrollers are:

— SAB 80515: Microcontroller, designed in Siemens MYMOS technology, with 8 Kbyte factory mask-programmable ROM

— SAB 80535: ROM-less version of the SAB 80515

— SAB 80C515: Microcontroller, designed in Siemens ACMOS technology, with 8 Kbyte factory mask-programmable ROM

— SAB 80C535: ROM-less version of the SAB 80C515

— SAB 80515K: Special ROM-less version of the SAB 80515 with an additional interface for program memory accesses. An external ROM that is accessed via this interface substitutes the SAB 80515's internal ROM.

The SAB 80535 is identical to the SAB 80515, except that it lacks the on-chip ROM. In this data sheet the term "SAB 80515" is used to refer to both the SAB 80515 and SAB 80535, unless otherwise noted.

Architecture

The architecture of the SAB 80515 is based on the SAB 8051 microcontroller family. The following features of the SAB 80515 are fully compatible with the SAB 8051 features:

— instruction set

— external memory expansion interface (port 0 and port 2)

— full-duplex serial port

— timer/counters 0 and 1

— alternate functions on port 3

— the lower 128 bytes of internal RAM and the lower 4 Kbytes of internal ROM

Different to the SAB 8051 are the RAM power-down supply, which supplies 40 byte with a typical current of 2 mA, and the powerful interrupt structure with 12 sources and 4 priority levels.

The SAB 80515 additionally contains 128 byte of internal RAM and 4 Kbyte of internal ROM, that means a total of 256 byte RAM and 8 Kbyte ROM on-chip. The SAB 80515 has a new 16-bit timer/counter with a 2:1 prescaler, reload mode, compare and capture capability. It also contains a 16-bit watchdog timer, an 8-bit A/D converter with 8 analog inputs and programmable reference voltages, two additional quasi-bidirectional 8-bit ports, and a programmable clock output (fosc/12).

CPU

The SAB 80515 is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12 MHz crystal, 58% of the instructions execute in 1.0 μs.

Memory Organization

The SAB 80515 manipulates operands in the four memory address spaces described below: (Refer to Figure 2.)

Program Memory

The SAB 80515 has 8 Kbytes of on-chip ROM, while the SAB 80535 has no internal ROM. The program memory can be externally expanded up to 64 Kbyte. If the $\overline{EA}$ pin is held high, the SAB 80515 executes out of internal ROM unless the address exceeds 1FFFH. Locations 2000H through 0FFFFH are then fetched from the external program memory. If the $\overline{EA}$ pin is held low, the SAB 80515 fetches all instructions from the external program memory. Since the SAB 80535 has no internal ROM, pin $\overline{EA}$ must be tied low when using this component.

Data Memory

The data memory address space consists of an internal and an external memory space. The internal data memory is divided into three physically separate and distinct blocks: the lower 128 byte of RAM; the upper 128 byte of RAM; and the 128-byte special function register (SFR) area. While the upper 128 byte of data memory and the SFR area share the same address locations, they are accessed through different addressing modes. The lower 128 byte of data memory can be accessed through direct or register-indirect addressing; the upper 128 byte of RAM can be accessed through register-indirect addressing; the special function registers are accessible through direct addressing.

Four 8-register banks, each bank consisting of eight 8-bit multi-purpose registers, occupy locations 0 through 1FH in the lower RAM area. The next 16 bytes, locations 20H through 2FH, contain 128 directly addressable bit locations. The stack can be located anywhere in the internal data memory address space, and the stack depths can be expanded up to 256 byte.

The external data memory can be expanded up to 64 Kbyte and can be accessed by instructions that use a 16-bit or an 8-bit address.

All registers, except the program counter and the four 8-register banks, reside in the special function register area. The 41 special function registers (SFR's) include arithmetic registers, pointers, and registers that provide an interface between the CPU and the on-chip peripheral functions. There are also 128 directly addressable bits within the SFR area. The special function registers are listed in the following table:

Symbol	Name	Address
*P0	Port 0	80H
SP	Stack Pointer	81H
DPL	Data Pointer, Low Byte	82H
DPH	Data Pointer, High Byte	83H
PCON	Power Control Register	87H
*TCON	Timer Control Register	88H
TMOD	Timer Mode Register	89H
TL0	Timer 0, Low Byte	8AH
TL1	Timer 1, Low Byte	8BH
TH0	Timer 0, High Byte	8CH
TH1	Timer 1, High Byte	8DH
*P1	Port 1	90H
*SCON	Serial Port Control Register	98H
SBUF	Serial Port Buffer Register	99H
*P2	Port 2	0A0H
*IEN0	Interrupt Enable Register 0	0A8H
IP0	Interrupt Priority Register 0	0A9H
*P3	Port 3	0B0H
*IEN1	Interrupt Enable Register 1	0B8H
IP1	Interrupt Priority Register 1	0B9H
*IRCON	Interrupt Request Control Register	0C0H
CCEN	Compare/Capture Enable Register	0C1H
CCL1	Compare/Capture Register 1, Low Byte	0C2H
CCH1	Compare/Capture Register 1, High Byte	0C3H
CCL2	Compare/Capture Register 2, Low Byte	0C4H
CCH2	Compare/Capture Register 2, High Byte	0C5H
CCL3	Compare/Capture Register 3, Low Byte	0C6H
CCH3	Compare/Capture Register 3, High Byte	0C7H
*T2CON	Timer 2 Control Register	0C8H
CRCL	Compare/Reload/Capture Register, Low Byte	0CAH
CRCH	Compare/Reload/Capture Register, High Byte	0CBH
TL2	Timer 2, Low Byte	0CCH
TH2	Timer 2, High Byte	0CDH
*PSW	Program Status Word Register	0D0H
*ADCON	A/D-Converter Control Register	0D8H
ADDAT	A/D-Converter Data Register	0D9H
DAPR	D/A-Converter Program Register	0DAH
*ACC	Accumulator	0E0H
*P4	Port 4	0E8H
*B	B Register	0F0H
*P5	Port 5	0F8H

The SFR's marked with an asterisk (*) are both bit and byte-addressable.
Figure 2 illustrates the memory address spaces of the SAB 80515.

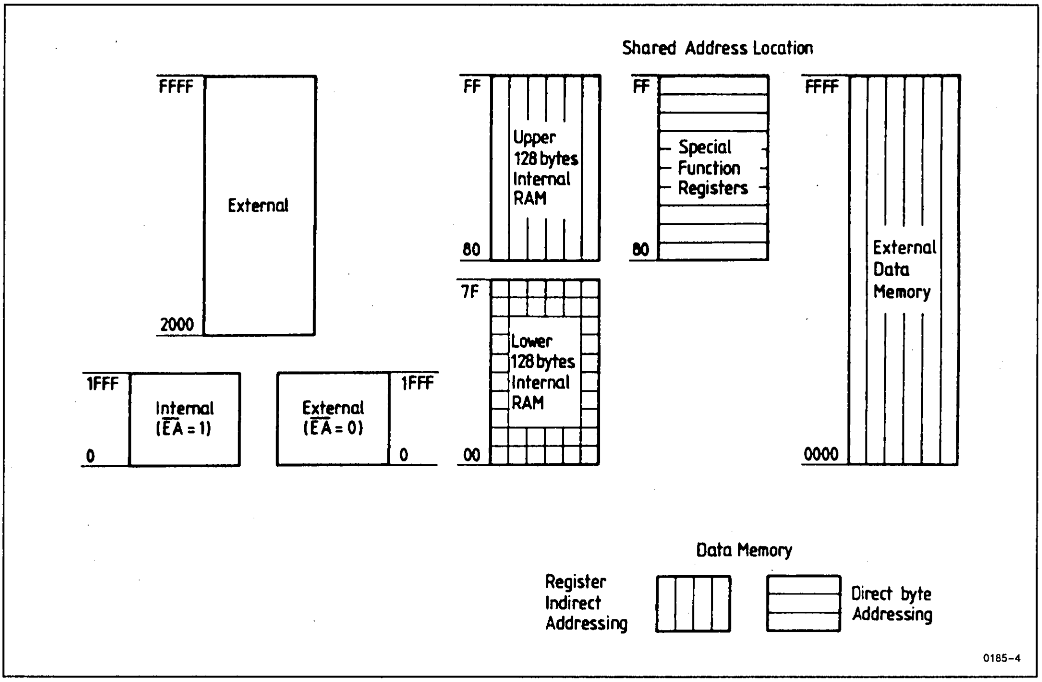


Figure 2. Memory Address Spaces

I/O Ports

The SAB 80515 has six 8-bit ports. Port 0 is an open-drain bidirectional I/O port, while ports 1 to 5 are quasi-bidirectional I/O ports with internal pullups. That means, when configured as inputs, ports 1 to 5 will pull high and will source current when externally pulled low. Port 0 will float when configured as input.

Port 0 and port 2 can be used to expand the program and data memory externally. During an access to external memory, port 0 emits the low-order address byte and reads/writes the data byte, while port 2 emits the high-order address byte. In this function, port 0 is not an open-drain port, but uses a strong internal pullup FET.

Ports 1 and 3 are provided for several alternate functions, as listed below:

Port	Symbol	Function
P1.0	$\overline{\text{INT3}}$/CC0	External Interrupt 3 Input, Compare 0 Output, Capture 0 Input
P1.1	INT4/CC1	External Interrupt 4 Input, Compare 1 Output, Capture 1 Input
P1.2	INT5/CC2	External Interrupt 5 Input, Compare 2 Output, Capture 2 Input
P1.3	INT6/CC3	External Interrupt 6 Input, Compare 3 Output, Capture 3 Input
P1.4	$\overline{\text{INT2}}$	External Interrupt 2 Input
P1.5	T2EX	Timer 2 External Reload Trigger Input
P1.6	CLKOUT	System Clock Output
P1.7	T2	Timer 2 External Counter Input
P3.0	RXD	Serial Input Port
P3.1	TXD	Serial Output Port
P3.2	$\overline{\text{INT0}}$	External Interrupt 0 Input, Timer 0 Gate Control
P3.3	$\overline{\text{INT1}}$	External Interrupt 1 Input, Timer 1 Gate Control
P3.4	T0	Timer 0 External Counter Input
P3.5	T1	Timer 1 External Counter Input
P3.6	$\overline{\text{WR}}$	External Data Memory Write Strobe
P3.7	$\overline{\text{RD}}$	External Data Memory Read Strobe

Timer/Counters

The SAB 80515 contains three 16-bit timer/counters which are useful in many applications for timing and counting. The input clock for each timer/counter is 1/12 of the oscillator frequency in the timer operation or can be taken from an external clock source for the counter operation (maximum count rate is 1/24 of the oscillator frequency).

Timer/Counters 0 and 1

These timer/counters can operate in four modes:

Mode 0: 8-bit timer/counter with 32:1 prescaler

Mode 1: 16-bit timer/counter

Mode 2: 8-bit timer/counter with 8-bit auto-reload

Mode 3: Timer/counter 0 is configured as one 8-bit timer/counter and one 8-bit timer; timer/counter 1 in this mode holds its count.

External inputs $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ can be programmed to function as a gate for timer/counters 0 and 1 to facilitate pulse width measurements.

PTRA

Programmable Timer/Counter Register Array (PTRA) of the SAB 80515 has timer/counter 2 as its time-base. Timer/counter 2 of the SAB 80515 is a 16-bit timer/counter with several additional features. It offers a 2:1 prescaler, a selectable gate function, and compare, capture and reload functions. Corresponding to the 16-bit timer register there are four 16-bit capture/compare registers, one of them can be used to perform a 16-bit reload on a timer overflow or external event. Each of these registers corresponds to a pin on port 1 for capture input/compare output.

Figure 3 shows a block diagram of the PTRA. The main features of the PTRA are:

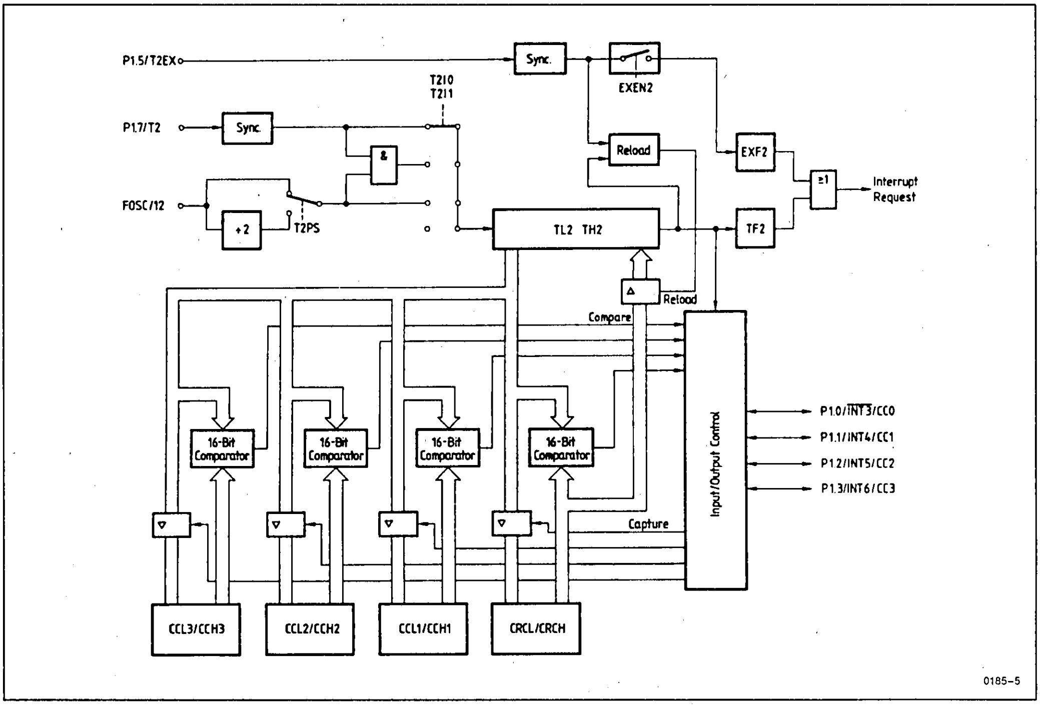


Figure 3. Block Diagram of PTRA

Reload

With the 16-bit CRC register, which is a concatanation of the 8-bit registers CRCL and CRCH, a 16-bit reload can be performed. There are two modes from which to select:

Mode 0: Reload is caused by a timer 2 overflow (auto-reload)

Mode 1: Reload is caused in response to a negative transition at pin T2EX (P1.5), which can also request an interrupt.

Compare

In the compare mode, the 16-bit values stored in the dedicated compare registers are compared to the contents of the timer 2 registers. If the count value in the timer 2 registers matches one of the stored values, an appropriate output signal is generated and an interrupt is requested. Two compare modes are provided:

Mode 0: Upon a match the output signal changes from low to high. It goes back to a low level when timer 2 overflows.

Mode 1: The transition of the output signal can be determined by software. A timer 2 overflow causes no output change.

Capture

This feature permits saving the actual timer/counter contents into a selected register upon an external event or a software write operation. Two modes are provided to latch the current 16-bit value in timer 2 registers into a dedicated capture register:

Mode 0: Capture is performed in response to a transition at the corresponding port 1 pins CC0 to CC3.

Mode 1: Write operation into the low-order byte of the dedicated capture register causes the timer 2 contents to be latched into this register.

Serial Port

The serial port of the SAB 80515 permits the full duplex communication between microcontrollers or between microcontrollers and peripheral devices. The serial port can operate in 4 modes:

Mode 0: Shift register mode. Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: 10 bits are transmitted (through RxD) or received (through TxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). The baud rate is variable.

Mode 2: 11 bits are transmitted (through RxD) or received (through TxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 3: 11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). Mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable.

The variable baud rates can be generated by timer 1 or an internal baud rate generator.

A/D Converter

The 8-bit A/D converter of the SAB 80515 has 8 multiplexed analog inputs and is using the successive approximation method. The sampling of an analog signal takes 5 machine cycles, the total conversion time is 15 machine cycles, (15 μs at 12 MHz oscillator frequency). Conversion can be programmed to be single or continuous, at the end of a conversion an interrupt can be generated.

The internal reference voltages IVAREF and IVAGND for the A/D converter are programmable in 16 steps with respect to the external reference voltages. This feature permits a second conversion with changed internal reference voltages to gain a higher resolution. In addition, the internal reference voltages can easily be adapted by software to the desired analog voltage range.

Figure 4 shows a block diagram of the A/D converter of the SAB 80515.

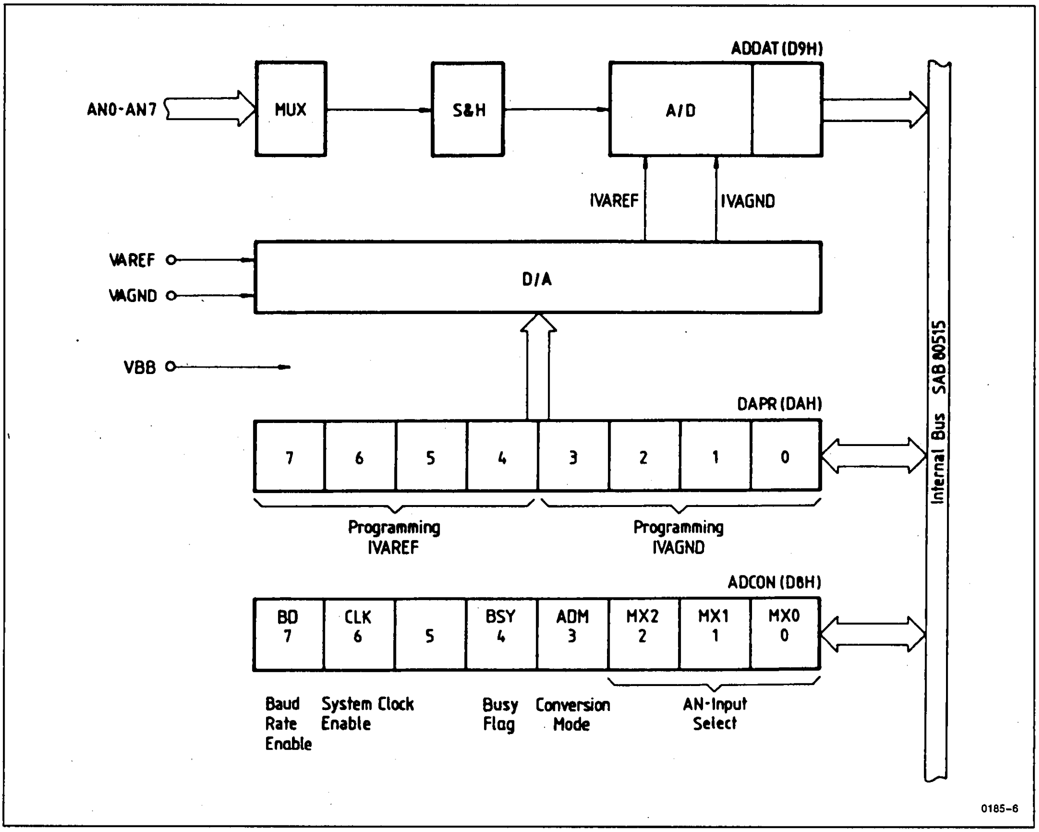


Figure 4. A/D Converter Block Diagram

Interrupt Structure

The 12 interrupt sources of the SAB 80515 are organized in 6 pairs:

External interrupt 0	A/D converter interrupt
Timer 0 interrupt	External interrupt 2
External interrupt 1	External interrupt 3
Timer 1 interrupt	External interrupt 4
Serial port interrupt	External interrupt 5
Timer 2 interrupt	External interrupt 6

Each interrupt source has its own vector address. It can be programmed to one of four priority levels and can individually be enabled/disabled. The minimum interrupt response time is 3 to 8 machine cycles.

External interrupts 0 and 1 can be activated by a low-level or a negative transition (selectable) at their corresponding input pin, external interrupts 2 and 3 can be programmed to be activated by a negative or a positive transition. The external interrupts 4 to 6 are activated by a positive transition. The interrupts 3 to 6 can be combined with the corresponding alternate functions compare (output) and capture (input) on port 1. For programming the priority levels, the interrupt vectors are combined in pairs. Each pair can be programmed individually to one of four priority levels by setting or clearing one bit in the special function register IP0 and one in IP1.

Figure 5 shows the interrupt request sources, and Figure 6 illustrates the priority level structure of the SAB 80515.

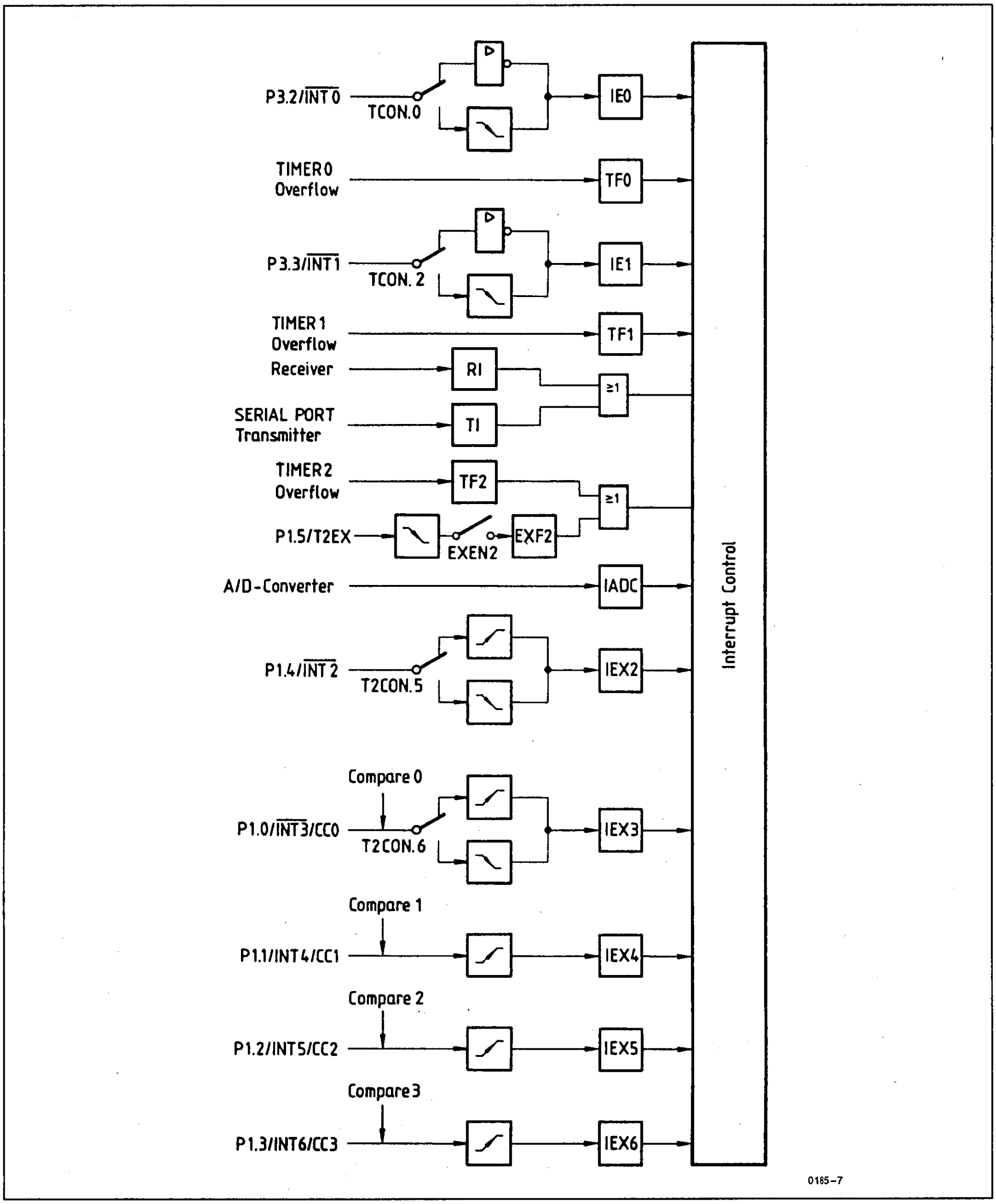


Figure 5. Interrupt Request Sources

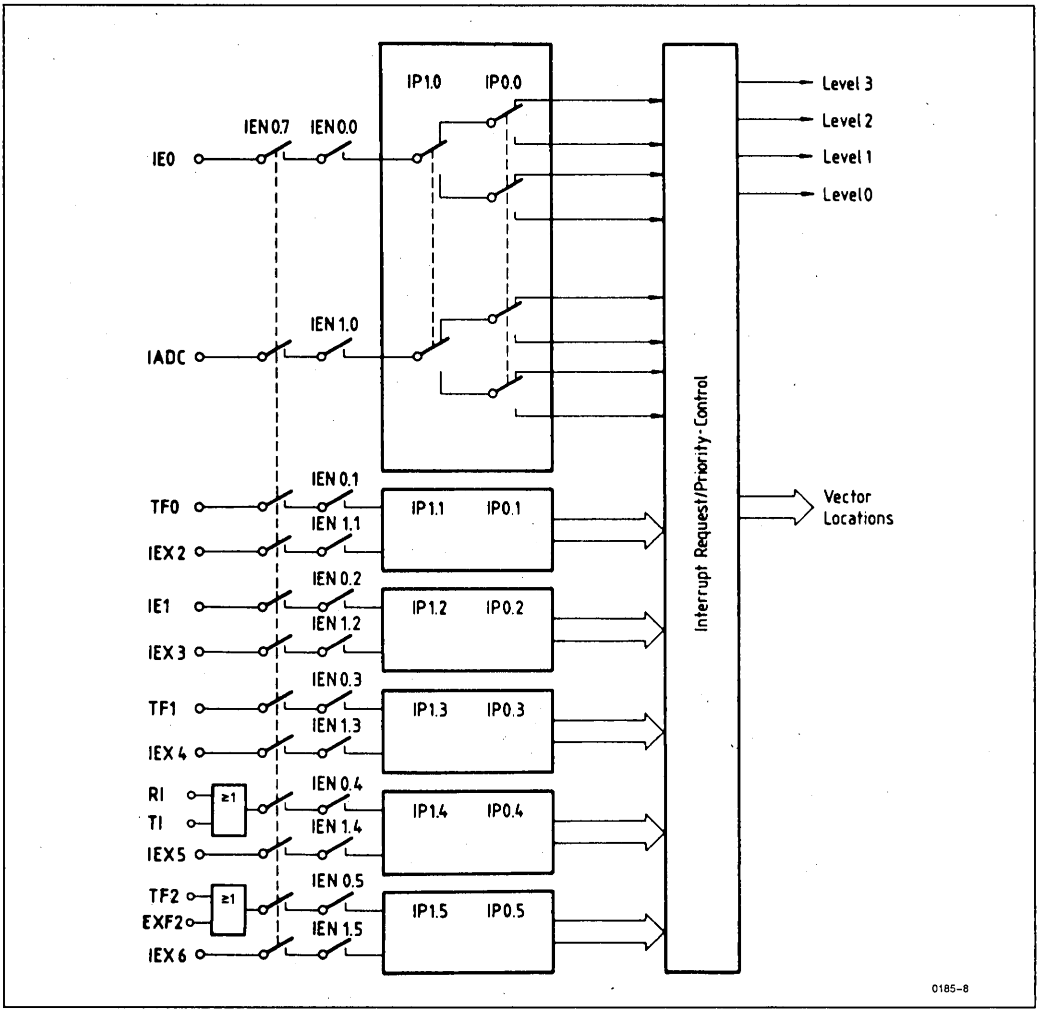


Figure 6. Priority Level Structure

Watchdog Timer

This feature is provided as a means of graceful recovery from software upset. After a reset, the watchdog timer is cleared and stopped. It can be started and cleared by software, but it cannot be stopped. If the software fails to clear the watchdog timer at least every 65532 machine cycles (about 65 ms if a 12 MHz oscillator frequency is used), a hardware reset will be initiated. The reset cause (external reset or reset caused by the watchdog) can be examined by software. To clear the watchdog, two bits in two different special function registers must be set by two consecutive instructions. This is done to prevent the watchdog from being cleared by unexpected op codes.

Absolute Maximum Ratings*

Ambient Temperature under Bias .... 0°C to + 70°C

Storage Temperature .... − 65°C to + 150°C

Voltage on Any Pin with
Respect to Ground (V_{SS}) .... − 0.5V to + 7V

Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5V ± 10%; V_{SS} = 0V;
T_A = 0°C to 70°C; for SAB 80515/80535

Parameter	Symbol	Test Conditions	Limit Values		Unit
			Min	Max	
Input Low Voltage	V_{IL}		− 0.5	0.8	V
Input High Voltage (except $\overline{\text{RESET}}$ and XTAL2)	V_{IH}		2.0	V_{CC} + 0.5	
Input High Voltage to XTAL2	V_{IH1}	XTAL1 to V_{SS}	2.5		
Input High Voltage to $\overline{\text{RESET}}$	V_{IH2}		3.0		
Power-Down Voltage	V_{PD}	V_{CC} = 0V	3	5.5	
Output Low Voltage, Ports 1, 2, 3, 4, 5	V_{OL}	I_{OL} = 1.6 mA		0.45	
Output Low Voltage, Port 0, ALE, $\overline{\text{PSEN}}$	V_{OL1}	I_{OL} = 3.2 mA			
Output High Voltage, Ports 1, 2, 3, 4, 5	V_{OH}	I_{OH} = − 80 μA	2.4		
Output High Voltage, Port 0, ALE, $\overline{\text{PSEN}}$	V_{OH1}	I_{OH} = − 400 μA			
Logic 0 Input Current, Ports 1, 2, 3, 4, 5	I_{IL}	V_{IL} = 0.45V		− 800	μA
Logic 0 Input Current, XTAL2	I_{IL2}	XTAL1 = V_{SS} V_{IL} = 0.45V		− 2.5	mA
Input Low Current to $\overline{\text{RESET}}$ for Reset	I_{IL3}	V_{IL} = 0.45		− 500	μA
Input Leakage Current to Port 0, $\overline{\text{EA}}$	I_{LI}	0V < V_{IN} < V_{CC}		± 10	
Power Supply Current SAB 80515/80535 SAB 80515-16/80535-16	I_{CC}	All Outputs Disconnected		210 TBD	mA
Power-Down Current	I_{PD}	V_{CC} = 0V		3	
Capacitance of I/O Buffer	C_{IO}	f_C = 1 MHz		10	pF

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

A/D Converter Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; $V_{AREF} = V_{CC}$ ±5%; $V_{AGND} = V_{SS}$ ±0.2V; $V_{IntAREF} - V_{IntAGND} \geq 1V$;
T_A = 0°C to +70°C for SAB 80515/80535

Parameter	Symbol	Test Condition	Limit Values			Unit
			Min	Typ	Max	
Analog Input Voltage	V_{AINPUT}		$V_{AGND} - 0.2$		$V_{AREF} + 0.2$	V
Analog Input Capacitance(1)	C_I			25		pF
Load Time	t_L				2 T_{CY}	µs
Sample Time (incl. Load Time)	t_S				5 T_{CY}	µs
Conversion Time	t_C				15 T_{CY}	
Differential Non-Linearity	DNLE	$V_{IntAREF} = V_{AREF} = V_{CC}$ $V_{IntAGND} = V_{AGND} = V_{SS}$		±½	±1	LSB
Integral Non-Linearity	INLE			±½	±1	LSB
Offset Error				±½	±1	LSB
Gain Error				±½	±1	LSB
Total Unadjusted Error(1)	TUE			±1	±2	LSB
V_{AREF} Supply Current(2)	IREF				5	mA
Internal Reference Error(2)	$V_{IntREFERR}$			±5	±15	mV

NOTES:

1. The internal resistance of the analog source must be low enough to assure full loading of the sample capacitance (C_I) during load time (t_L). After charging of the internal capacitance (C_I) in the load time (t_L) the analog input must be held constant for the rest of the sample time (t_S).
2. The differential impedance r_D of the analog reference voltage source must be less than 1 KΩ at reference supply voltage.

AC Characteristics for SAB 80515/80535

V_{CC} = 5V ±10%; V_{SS} = 0V
T_A = 0°C to 70°C
(CL for Port 0, ALE and $\overline{\text{PSEN}}$ Outputs = 100 pF; CL for All Other Outputs = 80 pF)

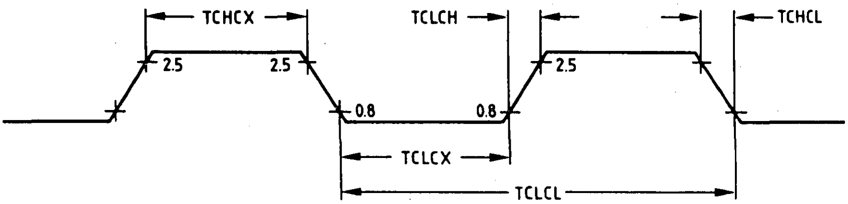
Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock 1/TCLCL = 1.2 MHz to 12 MHz		
		Min	Max	Min	Max	
Program Memory Characteristics						
Cycle Time	TCY	1000		12 TCLCL		ns
ALE Pulse Width	TLHLL	127		2 TCLCL − 40		
Address Setup to ALE	TAVLL	53		TCLCL − 30		
Address Hold after ALE	TLLAX1	48		TCLCL − 35		
ALE to Valid Instruction In	TLLIV		233		4 TCLCL − 100	
ALE to $\overline{\text{PSEN}}$	TLLPL	58		TCLCL − 25		
$\overline{\text{PSEN}}$ Pulse Width	TPLPH	215		3 TCLCL − 35		
$\overline{\text{PSEN}}$ to Valid Instruction In	TPLIV		150		3 TCLCL − 100	
Input Instruction Hold after $\overline{\text{PSEN}}$	TPXIX	0		0		
Input Instruction Float after $\overline{\text{PSEN}}$	TPXIZ*		63		TCLCL − 20	
Address Valid after $\overline{\text{PSEN}}$	TPXAV*	75		TCLCL − 8		
Address to Valid Instruction In	TAVIV		302		5 TCLCL − 115	
Address Float to $\overline{\text{PSEN}}$	TAZPL	0		0		
External Data Memory Characteristics						
$\overline{\text{RD}}$ Pulse Width	TRLRH	400		6 TCLCL − 100		ns
$\overline{\text{WR}}$ Pulse Width	TWLWH					
Address Hold after ALE	TLLAX2	132		2 TCLCL − 35		
$\overline{\text{RD}}$ to Valid Data In	TRLDV		252		5 TCLCL − 165	
Data Hold after $\overline{\text{RD}}$	TRHDX	0		0		
Data Float after $\overline{\text{RD}}$	TRHDZ		97		2 TCLCL − 70	
ALE to Valid Data In	TLLDV		517		8 TCLCL − 150	
Address to Valid Data In	TAVDV		585		9 TCLCL − 165	
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	TLLWL	200	300	3 TCLCL − 50	3 TCLCL + 50	
Address to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	TAVWL	203		4 TCLCL − 130		
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ High to ALE High	TWHLH	43	123	TCLCL − 40	TCLCL + 40	
Data Valid to $\overline{\text{WR}}$ Transition	TQVWX	33		TCLCL − 50		
Data Setup before $\overline{\text{WR}}$	TQVWH	433		7 TCLCL − 150		
Data Hold after $\overline{\text{WR}}$	TWHQX	33		TCLCL − 50		
Address Float after $\overline{\text{RD}}$	TRLAZ		0		0	

*Interfacing tho SAB 80515 to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damago to port 0 drivers.

External Clock Drive XTAL2

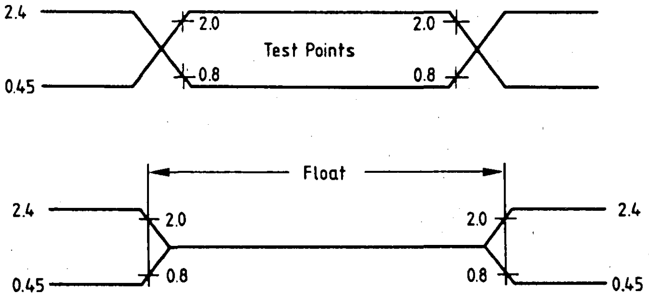
Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 1.2 MHz to 12 MHz		
		Min	Max	
External Clock Drive XTAL2				
Oscillator Period	TCLCL	83.3	833.3	ns
High Time	TCHCX	20	TCLCL − TCLCX	
Low Time	TCLCX		TCLCL − TCHCX	
Rise Time	TCLCH		20	
Fall Time	TCHCL			

External Clock Cycle



0185-09

AC Testing Input, Output, Float Waveforms



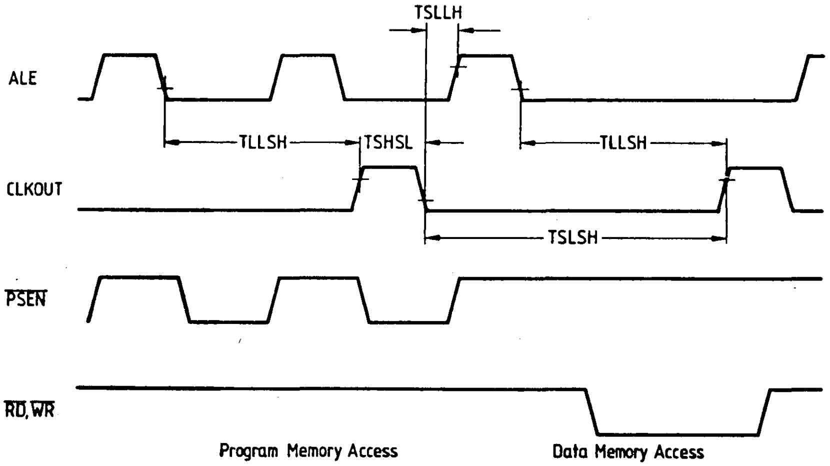
0185-10

NOTES:
AC testing inputs are driven at 2.4V for a logic "1" and at 0.45V for a logic "0".
Timing measurements are made at 2.0V for a logic "1" and at 0.8V for a logic "0".
For timing purposes, the float state is defined as the point where a P0 pin sinks 3.2 mA or sources 400 μA at voltage test levels.

System Clock Timing

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock 1/TCLCL = 1.2 MHz to 12 MHz		
		Min	Max	Min	Max	
ALE to CLKOUT	TLLSH	543		7 TCLCL − 40		ns
CLKOUT High Time	TSHSL	127		2 TCLCL − 40		
CLKOUT Low Time	TSLSH	793		10 TCLCL − 40		
CLKOUT Low to ALE High	TSLLH	43	123	TCLCL − 40	TCLCL + 40	

System Clock Timing



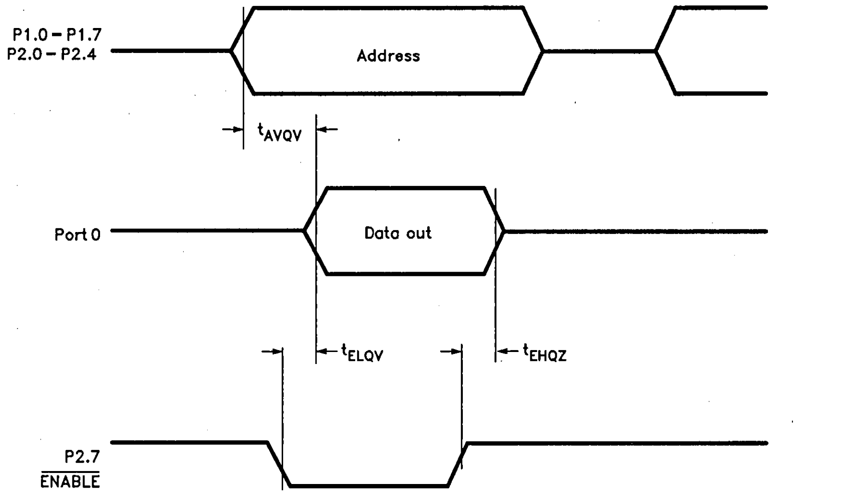
0185-11

ROM Verification Characteristics

TA = 25°C ±°C; VCC = 5V ±10%; VSS = 0V

Parameter	Symbol	Limit Values		Unit
		Min	Max	
Address to Valid Data	TAVQV		48 TCLCL	ns
ENABLE to Valid Data	TELQV			
Data Float after ENABLE	TEHQZ	0		
Oscillator Frequency	1/TCLCL	4	6	MHz

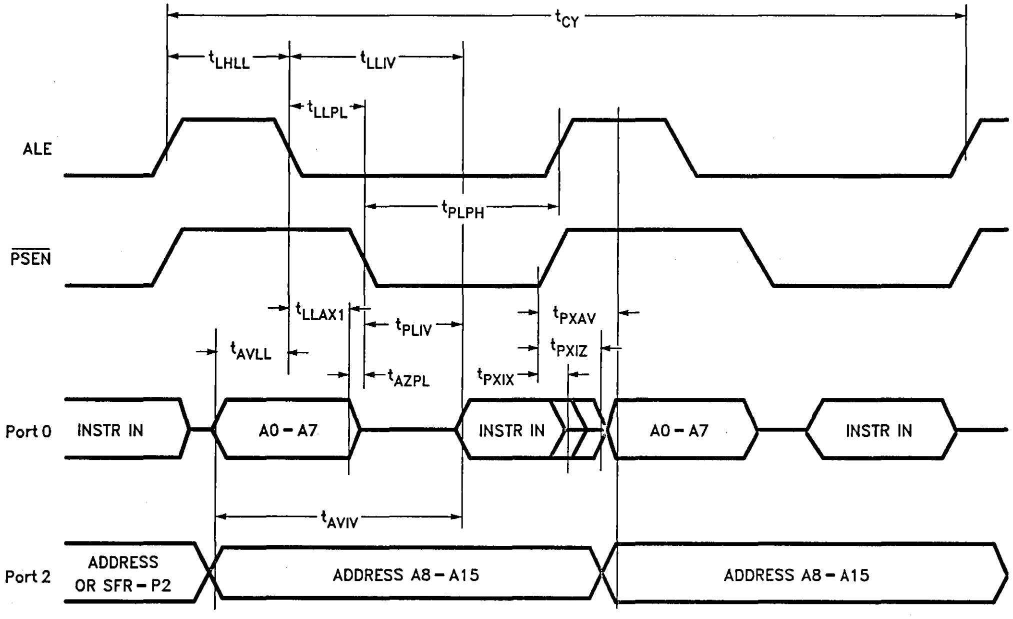
ROM Verification



Address: P1.0–P1.7 = A0–A7
P2.0–P2.4 = A8–A12
Data: Port 0 = D0–D7

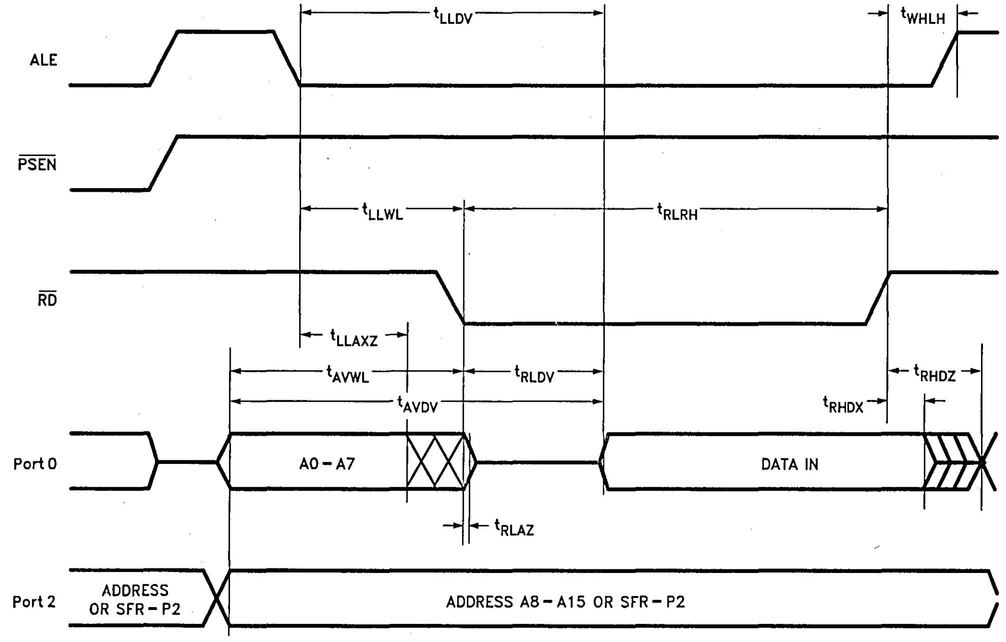
Inputs: P2.5–P2.6, $\overline{\text{PSEN}}$ = VSS
ALE, $\overline{\text{EA}}$ = V_{IH}
$\overline{\text{RESET}}$ = VIL

Program Memory Read Cycle



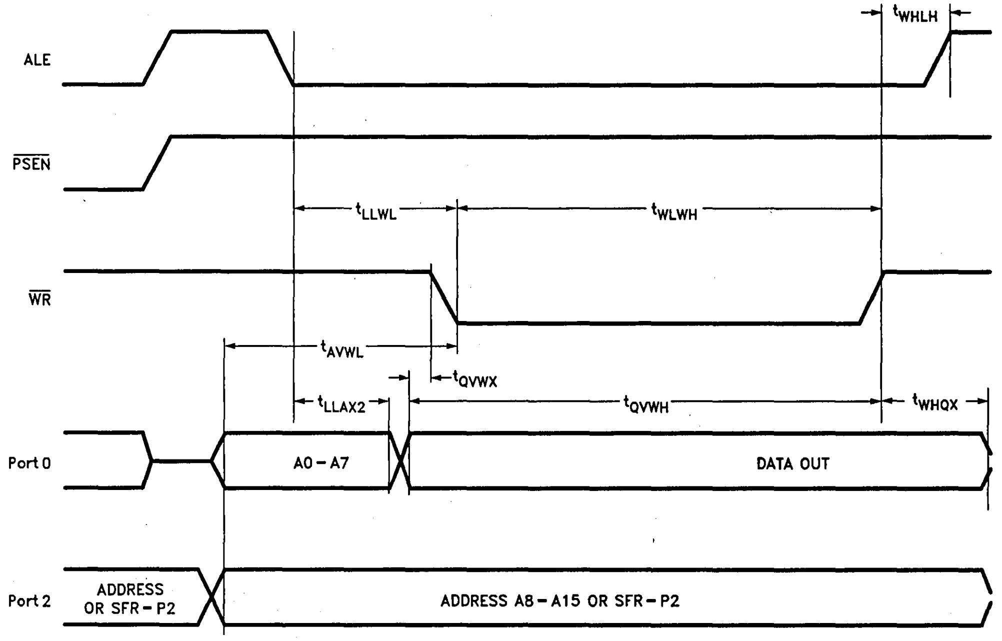
0185-13

Data Memory Read Cycle



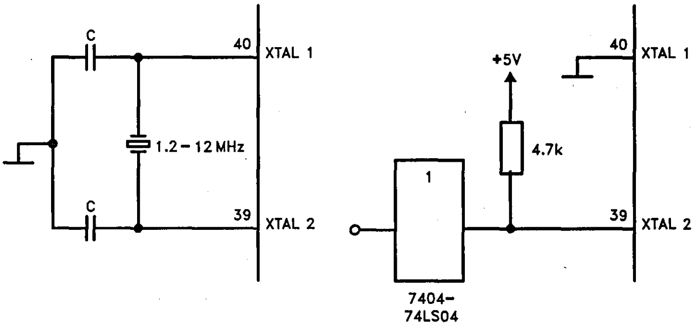
0185-14

Data Memory Write Cycle



0185-15

Recommended Oscillator Circuits



0185-16

C = 30 pF ± 10 pF
Crystal Oscillator Mode

Driving from External Source

Ordering Information

Type	Description
SAB 80515-N	8-Bit Single-Chip Microcontroller with Mask-Programmable ROM (PLCC 68)
SAB 80535-N	for External Memory (PLCC 68)

SIEMENS

SAB 80515/80535 Ext. Temp. 8-Bit Single-Chip Microcontroller

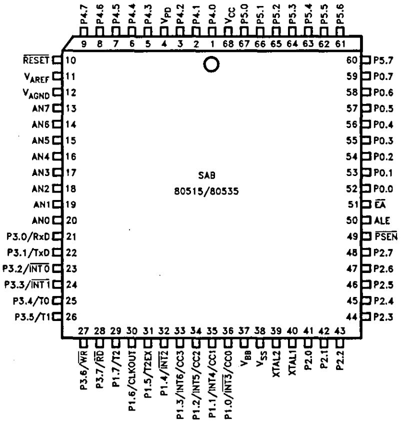
Extended Temperature Ranges:

T40/85	−40°C to +85°C	12 MHz operation
T40/110	−40°C to +110°C	10 MHz operation

SAB 80515-N-T40/85 SAB 80515-N-T40/110	Microcontroller with factory-mask programmable ROM
SAB 80535-N-T40/85 SAB 80535-N-T40/110	Microcontroller for external ROM

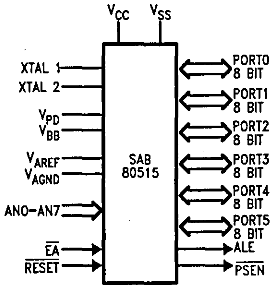
- Version of the SAB 80515/80535 for Two Extended Temperature Ranges
- 8k × 8 ROM (SAB 80515 only)
- 256 × 8 RAM
- Six 8-Bit I/O Ports, One 8-Bit Input Port for Analog Signals
- Three 16-Bit Timer/Event Counters
- Highly Flexible Reload, Capture, Compare Capabilities
- Full-Duplex Serial Channel
- Twelve Interrupt Vectors, Four Priority Levels
- V_{PD} Provides Standby Current for 40 bytes of RAM
- 8-Bit A/D Converter with Eight Multiplexed Inputs and Programmable Internal Reference Voltages
- 16-Bit Watchdog Timer
- Boolean Processor
- 256 Bit-Addressable Locations
- Most Instructions Execute in 1 μs
- 4 μs Multiply and Divide
- External Memory Expandable up to 128 Kbytes
- Backwardly Compatible with SAB 8051A
- 68-Pin Plastic Leaded Chip Carrier Package (PLCC 68)

Pin Configuration



0186-1

Logic Symbol



0186-2

The SAB 80515/80535 Ext. Temp. is a powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. The SAB 80515/80535 Ext. Temp. is available for the industrial temperature range (−40°C to +85°C) and the automotive temperature range (−40°C to +110°C). It is fully compatible with the standard SAB 80515/80535 with respect to architecture, instruction set and software portability. The SAB 80515/80535 Ext. Temp. is a stand-alone, high-performance single-chip microcontroller designed in +5V N-channel, silicon-gate Siemens MYMOS technology. While maintaining all the SAB 8051 operating characteristics, the SAB 80515/80535 Ext. Temp. incorporates several enhancements which significantly increase design flexibility and overall system performance.

The SAB 80535 is identical with the SAB 80515 except that it lacks the on-chip program memory. The SAB 80515/80535 Ext. Temp. is supplied in a 68-pin plastic leaded chip carrier package (PLCC 68).

Absolute Maximum Ratings*

Ambient Temperature under Bias
for SAB 80515/80535-
T40/85 .... −40°C to +85°C
for SAB 80515/80535-
T40/110 .... −40°C to +110°C

Storage Temperature .... −65°C to +150°C

Voltage on any Pin
with Respect to Ground (V_{SS}) .... −0.5V to +7V

Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5V ± 10%; V_{SS} = 0V; T_A = −40 to + 85°C for SAB 80515/80535-T40/85;
T_A = −40°C to +110°C for SAB 80515/80535-T40/110

Parameter	Symbol	Test Conditions	Limit Values		Unit
			Min	Max	
Input Low Voltage	V_{IL}		−0.5	0.8	V
Input High Voltage (except $\overline{\text{RESET}}$ and XTAL2)	V_{IH}		2.0	V_{CC} + 0.5	V
Input High Voltage to XTAL2	V_{IH1}	XTAL1 to V_{SS}	2.5	V_{CC} + 0.5	
Input High Voltage to $\overline{\text{RESET}}$	V_{IH2}		3.0		V
Power-Down Voltage	V_{PD}	V_{CC} = 0V	3	5.5	V
Output Low Voltage, Ports 1, 2, 3, 4, 5	V_{OL}	I_{OL} = 1.6 mA		0.45	V
Output Low Voltage, Ports 0, ALE, $\overline{\text{PSEN}}$	V_{OL1}	I_{OL} = 3.2 mA		0.45	V
Output High Voltage, Ports 1, 2, 3, 4, 5	V_{OH}	I_{OH} = −80 μA	2.4		V
Output High Voltage, Port 0, ALE, $\overline{\text{PSEN}}$	V_{OH1}	I_{OH} = −400 μA	2.4		V
Logic 0 Input Current, Ports 1, 2, 3, 4, 5	I_{IL}	V_{IL} = 0.45V		−800	μA
Logic 0 Input Current, XTAL2	I_{IL2}	XTAL1 = V_{SS} V_{IL} = 0.45V		−2.5	mA
Input Low Current to $\overline{\text{RESET}}$ for Reset	I_{IL3}	V_{IL} = 0.45V		−500	μA
Input Leakage Current to Port 0, $\overline{\text{EA}}$	I_{LI}	0V < V_{IN} < V_{CC}		±10	μA
Power Supply Current SAB 80515/80535-T40/85 SAB 80515/80535-T40/110	I_{CC}	All Outputs Disconnected		230 230	mA
Powor-Down Current	I_{PD}	V_{CC} = 0V		3	mA
Capacitance of I/O Buffer	C_{IO}	f_c = 1 MHz		10	pF

A/D Converter Characteristics: Refer to the SAB 80515/80535 Datasheet

AC Characteristics for T40/85: Refer to the SAB 80515/80535 Datasheet

AC Characteristics for T40/100

$V_{CC} = 5V \pm 10\%$; $V_{SS} = 0V$; $T_A = -40°C$ to $+110°C$ for SAB 80515/80535-T40/110; (C_L for port 0, ALE and $\overline{\text{PSEN}}$ outputs = 100 pF; C_L for all other outputs = 80 pF)

Parameter	Symbol	Limit Values				Unit
		10 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1.2 MHz to 10 MHz		
		Min	Max	Min	Max	
Program Memory Characteristics						
ALE Pulse Width	t_{LHLL}	160		$2\,t_{CLCL} - 40$		ns
Address Setup to ALE	t_{AVLL}	70		$t_{CLCL} - 30$		ns
Address Hold after ALE	t_{LLAX1}	65		$t_{CLCL} - 35$		ns
ALE to Valid Instruction In	t_{LLIV}		300		$4\,t_{CLCL} - 100$	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	75		$t_{CLCL} - 25$		ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	265		$3\,t_{CLCL} - 35$		ns
$\overline{\text{PSEN}}$ to Valid Instruction In	t_{PLIV}		200		$3\,t_{CLCL} - 100$	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	t_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	t_{PXIZ}*		80		$t_{CLCL} - 20$	ns
Address Valid after $\overline{\text{PSEN}}$	t_{PXAV}*	92		$t_{CLCL} - 8$		ns
Address to Valid Instruction In	t_{AVIV}		385		$5\,t_{CLCL} - 115$	ns
Address Float to $\overline{\text{PSEN}}$	t_{AZPL}	0		0		ns
External Data Memory Characteristics						
$\overline{\text{RD}}$ Pulse Width	t_{RLRH}	500		$6\,t_{CLCL} - 100$		ns
$\overline{\text{RD}}$ Pulse Width	t_{WLWH}	500		$6\,t_{CLCL} - 100$		ns
Address Hold after ALE	t_{LLAX2}	165		$2\,t_{CLCL} - 35$		ns
$\overline{\text{RD}}$ to Valid Data In	t_{RLDV}		335		$5\,t_{CLCL} - 165$	ns
Data Hold after $\overline{\text{RD}}$	t_{RHDX}	0		0		ns
Data Float after $\overline{\text{RD}}$	t_{RHDZ}		130		$2\,t_{CLCL} - 70$	ns
ALE to Valid Data In	t_{LLDV}		650		$8\,t_{CLCL} - 150$	ns
Address to Valid Data In	t_{AVDV}		735		$9\,t_{CLCL} - 165$	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{LLWL}	250	350	$3\,t_{CLCL} - 50$	$3\,t_{CLCL} + 50$	ns
Address to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{AVWL}	270		$4\,t_{CLCL} - 130$		ns
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ High to ALE High	t_{WHLH}	60	140	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
Data Valid to $\overline{\text{WR}}$ Transition	t_{DVWX}	50		$t_{CLCL} - 50$		ns
Data Setup before $\overline{\text{WR}}$	t_{QVWH}	550		$7\,t_{CLCL} - 150$		ns
Data Hold after $\overline{\text{WR}}$	t_{WHQX}	50		$t_{CLCL} - 50$		ns
Address Float after $\overline{\text{RD}}$	t_{RLAZ}		0		0	ns

* Interfacing the SAB 80515 to devices with float times up to 92 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

External Clock Drive XTAL2

Parameter	Symbol	Limit Values
		Variable Clock
		Freq. = 1.2 MHz to 12 MHz (T40/85)
		Freq. = 1.2 MHz to 10 MHz (T40/110)
				Unit

		Min	Max	
Oscillator Period T40/85 T40/110	t_{CLCL}	83.3 100	833.3 833.3	ns
High Time	t_{CHCX}	20	t_{CLCL}-t_{CLCX}	ns
Low Time	t_{CLCX}	20	t_{CLCLC}−t_{CHCX}	ns
Rise Time	t_{CLCH}		20	ns
Fall Time	t_{CHCL}		20	ns

System Clock Timing T40/110

Parameter	Symbol	Limit Values				Unit
		10 MHz Clock		Variable Clock 1/t_{CLCL} = 1.2 MHz to 10 MHz		
		Min	Max	Min	Max	
ALE to CLKOUT	t_{LLSH}	660		7 t_{CLCL} − 40		ns
CLKOUT High Time	t_{SHSL}	160		2 t_{CLCL} − 40		ns
CLKOUT Low Time	t_{SLSH}	960		10 t_{CLCL} − 40		ns
CLKOUT Low to ALE High	t_{SLLH}	60	140	t_{CLCL} − 40	t_{CLCL} + 40	ns

Waveforms: Refer to SAB 80515/80535 Data Sheet for the Waveforms

Ordering Information

Type	Description
	8-Bit Single Chip Microcomputer
SAB 80515-N-T40/85	with Mask-Programmable ROM (Plastic)
SAB 80535-N-T40/85	for External Memory (Plastic)
SAB 80515-N-T40/110	with Mask-Programmable ROM (Plastic)
SAB 80535-N-T40/110	for External Memory (Plastic)

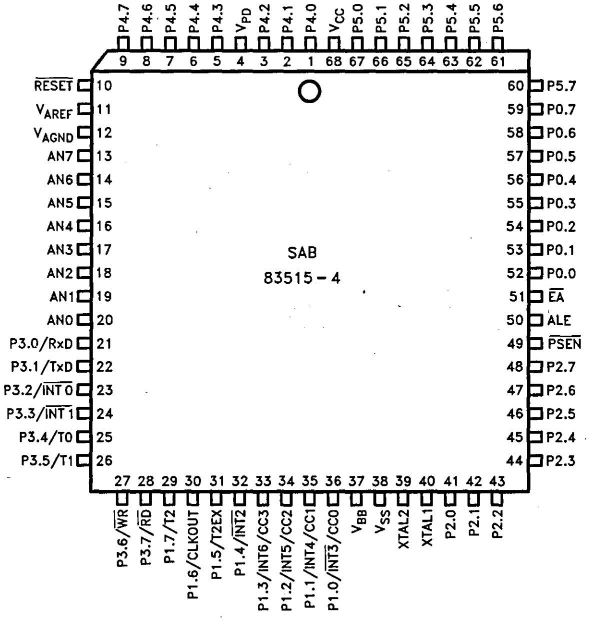
SIEMENS

SAB 83515-4
8-Bit Single-Chip Microcontroller

SAB 83515-4 Microcontroller with Factory-Mask Programmable ROM, 12 MHz Operation

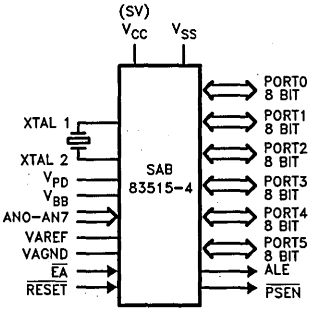
- 16K x 8 ROM
- 256 x 8 RAM
- Six 8-bit ports, 48 I/O Lines
- Three 16-bit Timer/Event Counters
- PTRA with Highly Flexible Reload, Capture, Compare Capabilities
- High-Performance Full-Duplex Serial Channel
- Twelve Interrupt Vectors, Four Priority Levels
- 8-bit A/D Converter with 8 Multiplexed Analog Inputs and Programmable Internal Reference Voltages
- 16-bit Watchdog Timer
- V_{PD} Provides Standby Current for 40 Bytes of RAM
- Boolean Processor
- 256 Bit-Addressable Locations
- Most Instructions Execute in 1 μs
- 4 μs Multiply and Divide
- External Memory Expandable to 128 Kbyte
- Backward-Compatible with SAB 8051A
- 68-Pin Plastic Leaded Chip Carrier Package (PLCC 68)

Pin Configuration



0291-1

Logic Symbol



0291-2

The SAB 83515-4 is a new member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is fabricated in +5V advanced N-channel, silicon-gate Siemens MYMOS technology and supplied in a 68-pin PLCC package. The SAB 83515-4 is a stand-alone, high-performance single-chip microcontroller based on the SAB 8051 architecture. While maintaining all the SAB 80515 operating characteristics, the SAB 83515-4 has 16 KBytes of ROM on-chip. This feature makes 83515-4 a cost effective microcontroller for applications of the 80515 which require 16K ROM space.

Absolute Maximum Ratings*

Ambient Temperature under Bias . . . . 0°C to +70°C

Storage Temperature . . . . −65°C to +150°C

Voltage on Any Pin with
Respect to Ground (V_{SS}) . . . . −0.5V to +7V

Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = 0°C to 70°C; for SAB 83515-4

Parameter	Symbol	Test Conditions	Limit Values		Unit
			Min	Max	
Input Low Voltage	V_{IL}		−0.5	0.8	V
Input High Voltage (except $\overline{\text{RESET}}$ and XTAL2)	V_{IH}		2.0	V_{CC} + 0.5	
Input High Voltage to XTAL2	V_{IH1}	XTAL1 to V_{SS}	2.5		
Input High Voltage to $\overline{\text{RESET}}$	V_{IH2}		3.0		
Power-Down Voltage	V_{PD}	V_{CC} = 0V	3	5.5	
Output Low Voltage, Ports 1, 2, 3, 4, 5	V_{OL}	I_{OL} = 1.6 mA		0.45	
Output Low Voltage, Port 0, ALE, $\overline{\text{PSEN}}$	V_{OL1}	I_{OL} = 3.2 mA			
Output High Voltage, Ports 1, 2, 3, 4, 5	V_{OH}	I_{OH} = −80 µA	2.4		
Output High Voltage, Port 0, ALE, $\overline{\text{PSEN}}$	V_{OH1}	I_{OH} = −400 µA			
Logic 0 Input Current, Ports 1, 2, 3, 4, 5	I_{IL}	V_{IL} = 0.45V		−800	µA
Logic 0 Input Current, XTAL2	I_{IL2}	XTAL1 = V_{SS} V_{IL} = 0.45V		−2.5	mA
Input Low Current to $\overline{\text{RESET}}$ for Reset	I_{IL3}	V_{IL} = 0.45		−500	µA
Input Leakage Current to Port 0, $\overline{\text{EA}}$	I_{LI}	0V < V_{IN} < V_{CC}		±10	
Power Supply Current	I_{CC}	All Outputs Disconnected		TBD	mA
Power-Down Current	I_{PD}	V_{CC} = 0V		3	
Capacitance of I/O Buffer	C_{IO}	f_C = 1 MHz		10	pF

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

AC Characteristics

The AC characteristics and waveform timings of the SAB 83515-4 are the same as that of the SAB 80515.

Ordering Information

Type	Description
SAB 83515-4N	8-Bit Single-Chip Microcontroller with Mask-Programmable ROM (PLCC 68)

SIEMENS

Advance Information

SAB 80C52/80C32, 80C52-16/80C32-16 SAB 80C52/80C32, 80C52-16/80C32-16 Ext. Temp. 8-Bit CMOS Microcontroller

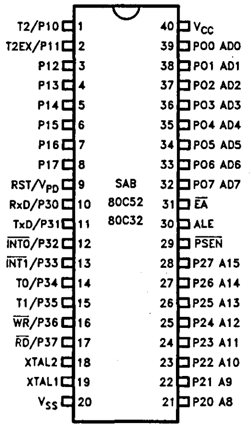
SAB 80C52(16)-P(N) CMOS microcontroller with factory-mask programmable ROM
SAB 80C32(16)-P(N) CMOS microcontroller for external ROM

SAB 80C52/80C32-T40/85(-16) Extended temperature range −40°C to +85°C
SAB 80C52/80C32-T40/110 Extended temperature range −40°C to +110°C

- SAB 80C52/80C32 (-T40/85, -T40/110) for 12 MHz operation
- SAB 80C52-16/80C32-16 (-T40/85) for 16 MHz operation
- 8K × 8 ROM (SAB 80C52 only)
- 256 × 8 RAM
- Four 8-Bit Ports, 32 I/O Lines
- Three 16-Bit Timer/Event Counters
- High-Performance Full-Duplex Serial Channel with Flexible Transmit/Receive Baud Rate Capability
- External Memory Expandable up to 128 Kbytes
- Boolean Processor
- Most Instructions Execute in 1 µs (750 ns)
- Multiply and Divide in 4 µs (3 µs)
- Six Interrupt Sources,Two Priority Levels
- Idle and Power-Down Operation
- P-DIP 40 and PLCC 44 Packages
- Full Backward Compatibility with SAB 80C51/80C31

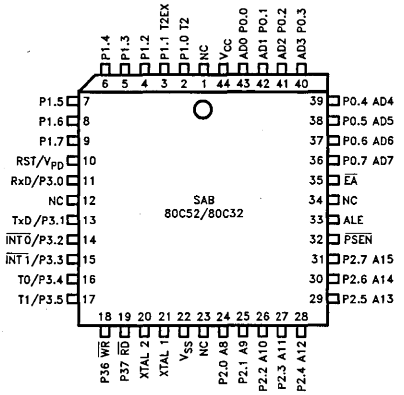
Pin Configurations

P-DIP 40



0188-1

PLCC 44



0188-2

The SAB 80C52/80C32 is a stand-alone, high-performance CMOS single-chip microcontroller, designed in Siemens ACMOS technology. It is functionally compatible with the SAB 8052A/8032A devices in MYMOS technology. The SAB 80C52/80C32 with the suffix "-16" operates up to 16 MHz crystal oscillator frequency. The part without the suffix operates up to a maximum frequency of 12 MHz. For the industrial temperature range −40°C to +85°C, the SAB 80C52/80C32-T40/85 (-16) and for the automotive temperature range −40°C to +110°C, the SAB 80C52/80C32-T40/110 are also available.

Furthermore, it is backwardly compatible with the SAB 80C51/80C31. The low-power properties of ACMOS technology allow applications where power consumption and dissipation are critical. In addition, the SAB 80C52/80C32 has two software-selectable modes of reduced activity for further power reduction—idle and power-down.

The SAB 80C52 contains a non-volatile 8K × 8 read-only program memory, a volatile 256 × 8 read/write data memory, 32 I/O lines, three 16-bit timer/counters, a six-source, two-priority-level interrupt structure, a serial I/O port, an on-chip oscillator, and clock circuits. The SAB 80C32 is identical, except that it lacks the program memory on the chip.

The SAB 80C52/80C32 is supplied in a 40-pin P-DIP package, or a 44-pin plastic lead chip carrier (PLCC 44) package.

Pin Definitions and Functions

Pin		Symbol	Input(I) Output (O)	Functions
P-DIP40	PLCC44			
1–8	2–9	P1.0–P1.7	I/O	Port 1 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 1 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, on the DC characteristics) because of the internal pullup resistors. Port 1 also receives the low-order address bytes during program verification. Port 1 also contains the timer 2 pins as a secondary function. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 1, as follows: —T2 (P1.0). Input to counter 2. —T2 EX (P1.10. Capture/Reload trigger of timer 2.
9	10	RST	I	A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits power-on reset using only an external capacitor to V_{CC}.

Pin Definitions and Functions (Continued)

Pin		Symbol	Input(I) Output (O)	Functions
P-DIP40	PLCC44			
10–17	11, 13–19	P3.0–P3.7	I/O	Port 3 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 3 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, on the DC characteristics) because of the internal pullup resistors. Port 3 also contains the interrupt, timer, serial port and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: $\overline{RxD}$/data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous). $\overline{TxD}$/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous). $\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0. $\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1. T0 (P3.4). Input to counter 0. T1 (P3.5). Input to counter 1. $\overline{WR}$ (P3.6). The write control signal latches the data byte from port 0 into the external data memory. $\overline{RD}$ (P3.7). The read control signal enables external data memory to port 0.
19 18	21 20	XTAL1 XTAL2		XTAL 1 Input to the inverting oscillator amplifier and input to the internal clock generator circuits. XTAL 2 Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL 1 should be driven, while XTAL 2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop. Minimum and maximum high and low times specified in the AC characteristics must be observed.

Pin Definitions and Functions (Continued)

Pin		Symbol	Input (I) Output (O)	Functions
P-DIP40	PLCC44			
21–28	24–31	P.20–P2.7	I/O	Port 2 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 2 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, on the DC characteristics) because of the internal pullup resist. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullup resistors when issuing 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ Ri), port 2 issues the contents of the P2 special function register.
29	32	$\overline{\text{PSEN}}$	O	$\overline{\text{PROGRAM STORE ENABLE}}$ This output issues a control signal that enables the external program memory to access the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.
30	33	ALE	O	ADDRESS LATCH ENABLE Provides signal used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
31	35	$\overline{\text{EA}}$	I	$\overline{\text{EXTERNAL ACCESS}}$ When held at a high level, the SAB 80C52 executes instructions from the internal ROM when the PC is less than 8192. When held at a low level, the SAB 80C52 fetches all instructions from the external program memory. For the SAB 80C32 this pin must be tied low.
39–32	43-36	P0.0–P0.7	I/O	Port 0 is an 8-bit open drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application it uses strong internal pullup resistors when issuing 1s. Port 0 also outputs the code bytes during program verification in the SAB 80C52. External pullup resistors are required during program verification.
40	44	V_{CC}		Supply voltage during normal, idle, and power-down operations.
20	22	V_{SS}		Circuit ground potential.
	1, 12 23, 24	NC		No connection

Functional Description

The SAB 80C52/80C32 is functionally compatible with the SAB 8052B/8032B products that are designed in Siemens MYMOS technology. Furthermore, the SAB 80C52/80C32 is backwardly compatible with the SAB 80C51/80C31 devices.

In addition, instead of the RAM backup power supply of the SAB 8052B/8032B, the SAB 80C52/80C32 offers two additional power control modes, the idle mode and the power-down mode. The control bits for the reduced power modes are in the special function register PCON.

— Idle Mode
In the idle mode, the CPU puts itself to sleep while all the on-chip peripherals stay active. The instruction that invokes the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The contents of the CPU, the on-chip RAM, and all the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt, at which time the process is picked up at the interrupt service routine and continued, or by a hardware reset which starts the processor in the same way as a power-on reset.

— Power-Down Mode
In the power-down mode the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. Only the contents of the on-chip RAM is preserved. A hardware reset is the only way to terminate power-down.

During power-down and idle mode the external pins will have the following status (See Table 1):

Table 1. Status of the External Pins during Idle and Power-Down Modes

Mode	Program Memory	ALE	PSEN	Port 0	Port 1	Port 2	Port 3
Idle	Internal	1	1	Data	Data/ Alternate Outputs	Data	Data/ Alternate Outputs
Idle	External	1	1	Float	Data/ Alternate Outputs	Address	Data/ Alternate Outputs
Power-Down	Internal	0	0	Data	Data/Last Output of Alternate Function	Data	Data/Last Output of Alternate Function
Power-Down	External	0	0	Float	Data/Last Output of Alternate Function	Data	Data/Last Output of Alternate Function

Absolute Maximum Ratings*

Ambient Temperature under Bias
SAB 80C52 .... 0°C to +70°C
SAB 80C52-T40/85 .... −40°C to +85°C
SAB 80C52-T40/110 .... −40°C to +110°C

Storage Temperature .... −65°C to +150°C

Voltage on Any Pin with
Respect to Ground (V_{SS})... −0.5V to V_{CC} +0.5V

Voltage on V_{CC} to V_{SS} .... −0.5V to 6.5V

Power Dissipation 1W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V;
T_A = 0°C to +70°C (SAB 80C52/80C32);
T_A = −40°C to +85°C (SAB 80C52/80C32-T40/85 (-16));
T_A = −40°C to +110°C (SAB 80C52/80C32-T40/110)

Parameter	Symbol	Test Condition	Limit Values		Unit
			Min	Max	
Input Low Voltage (Except EA)	V_{IL}		−0.5	0.2 V_{CC} − 0.1	V
Input Low Voltage (EA)	V_{IL1}		−0.5	0.2 V_{CC} − 0.3	V
Input High Voltage (Except XTAL1, RST)	V_{IH}		0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
Input High Voltage (XTAL1, RST)	V_{IH1}		0.7 V_{CC}	V_{CC} + 0.5	V
Output Low Voltage (Ports 1, 2, 3)	V_{OL}	I_{OL} = 1.6 mA (1)		0.45	V
Output Low Voltage (Port 0, ALE, PSEN)	V_{OL1}	I_{OL} = 3.2 mA (1)		0.45	V
Output High Voltage (Ports 1, 2, 3)	V_{OH}	I_{OH} = −80 μA, V_{CC} = 5V ±10%	2.4		V
		I_{OH} = −25 μA	0.75 V_{CC}		V
		I_{OH} = −10 μA	0.9 V_{CC}		V
Output High Voltage (Port 0 in External Bus Mode, ALE, PSEN)	V_{OH1}	I_{OH} = −800 μA, V_{CC} = 5V ±10%	2.4		V
		I_{OH} = −150 μA	0.75 V_{CC}		V
		I_{OH} = −80 μA (2)	0.9 V_{CC}		V
Logical 0 Input Current (Ports 1, 2, 3)	I_{IL}	V_{IN} = 0.45V		−50	μA
Logical 1-to-0 Transition Current (Ports 1, 2, 3)	I_{TL}	V_{IN} = 2V		−650	μA
Input Leakage Current (Port 0, EA)	I_{LI}	0.45V < V_{IN} < V_{CC}		±10	μA

D.C. Characteristics (Continued)

Parameter	Symbol	Test Condition	Limit Values		Unit
			Min	Max	
Reset Pulldown Resistor	R_{RST}		40	150	kΩ
Pin Capacitance	C_{IO}	$f_C = 1$ MHz, $T_A = 25°C$		10	pF
Power Supply Current:					
Active Mode, 12 MHz (6)	I_{CC}	$V_{CC} = 5V$ (4)	20		mA
Idle Mode, 12 MHz (6)	I_{CC}	$V_{CC} = 5V$ (5)	6.8		mA
Power Down Mode	I_{PD}	$V_{CC} = 2.0V–5V$ (3)	50		μA

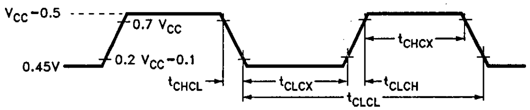
NOTES:

1. Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation. In the worst case (capacitive load > 100 pF), the noise pulse on ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt-trigger, or use an address latch with a Schmitt-trigger strobe input.
2. Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and PSEN to momentarily fall below the 0.9 V_{CC} specification when the address bits are stabilizing.
3. Power-down I_{CC} is measured with: $\overline{EA}$ = Port 0 = V_{CC}; XTAL1 = V_{SS}; XTAL2 = N.C.; $\overline{RESET}$ = V_{SS}; all other pins are disconnected.
4. I_{CC} (active mode) is measured with: XTAL1 driven with clock signal according to the figure below; XTAL2 = N.C.; $\overline{EA}$ = Port 0 = V_{CC}; $\overline{RESET}$ = V_{CC}; all other pins are disconnected. I_{CC} might be slightly higher if a crystal oscillator is used.
5. I_{CC} (idle mode) is measured with: XTAL1 driven with clock signal according to the figure below; XTAL2 = N.C.; $\overline{EA}$ = V_{SS}; Port 0 = V_{CC}; $\overline{RESET}$ = V_{SS}; all other pins are disconnected.
6. $I_{CC\,Max}$ at other frequencies is given by: Active Mode: $I_{CC\,Max}$ = 1.5 * fosc + 2.0
   Idle Mode: $I_{CC\,Max}$ = 0.4 * fosc + 2.0

where fosc is the oscillator frequency in MHz. I_{CC} values are given in mA and measured at V_{CC} = 5V (see also notes 4 and 5).

Clock Signal Waveform for I_{CC} Tests in Active and Idle Mode

$t_{CLCH} = t_{CHCL} = 5$ ns



0188-3

A.C. Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V,
T_A = 0°C to 70°C (SAB 80C52);
T_A = −40 to +85°C (SAB 80C52-T40/85); V_{CC} = 5V ± 10%
T_A = −40°C to +110°C (SAB 80C52-T40/110)
(C_L for Port 0, ALE and $\overline{PSEN}$ Outputs = 100 pF; C_L for All Other Outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 0.5 MHz to 12 MHz		
		Min	Max	Min	Max	
ALE Pulse Width	t_{LHLL}	127		2 t_{CLCL} − 40		ns
Address Setup to ALE	t_{AVLL}	53		t_{CLCL} − 30		ns
Address Hold after ALE	t_{LLAX}	48		t_{CLCL} − 35		ns
ALE to Valid Instruction in	t_{LLIV}		233		4 t_{CLCL} − 100	ns
ALE to $\overline{PSEN}$	t_{LLPL}	58		t_{CLCL} − 25		ns
$\overline{PSEN}$ Pulse Width	t_{PLPH}	215		3 t_{CLCL} − 35		ns
$\overline{PSEN}$ to Valid Instruction in	t_{PLIV}		150		3 t_{CLCL} − 100	ns
Input Instruction Hold after $\overline{PSEN}$	t_{PXIX}	0		0		ns
Input Instruction Float after $\overline{PSEN}$	t_{PXIZ} (1)		63		t_{CLCL} − 20	ns
Address Valid after $\overline{PSEN}$	t_{PXAV} (1)	75		t_{CLCL} − 8		ns
Address to Valid Instruction in	t_{AVIV} (1)		302		5 t_{CLCL} − 115	ns
Address Float to $\overline{PSEN}$	t_{PLAZ}		0		10	ns

NOTE:
1. Interfacing the SAB 80C52 to devices with float times up to 75 ns is permissable. This limited bus contention will not cause any damage to Port 0 drivers.

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 0.5 MHz to 12 MHz		
		Min	Max	Min	Max	
$\overline{RD}$ Pulse Width	t_{RLRH}	400		6 t_{CLCL} − 100		ns
$\overline{WR}$ Pulse Width	t_{WLWH}	400		6 t_{CLCL} − 100		ns
Address Hold after ALE	t_{LLAX2}	132		2t_{CLCL} − 35		ns
$\overline{RD}$ to Valid Data In	t_{RLDV}		252		5 t_{CLCL} − 165	ns
Data Hold after $\overline{RD}$	t_{RHDX}	0		0		ns
Data Float after $\overline{RD}$	t_{RHDZ}		97		2 t_{CLCL} − 70	ns
ALE to Valid Data In	t_{LLDV}		517		8 t_{CLCL} − 150	ns
Address to Valid Data In	t_{AVDV}		585		9 t_{CLCL} − 165	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	200	300	3 t_{CLCL} − 50	3 t_{CLCL} + 50	ns
Address to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	203		4 t_{CLCL} − 130		ns
$\overline{WR}$ or $\overline{RD}$ High to ALE High	t_{WHLH}	43	123	t_{CLCL} − 40	t_{CLCL} + 40	ns
Data Valid to $\overline{WR}$ Transition	t_{QVWX}	33		t_{CLCL} − 60		ns
Data Setup before $\overline{WR}$	t_{QVWH}	433		7 t_{CLCL} − 150		ns
Data Hold after $\overline{WR}$	t_{WHQX}	33		t_{CLCL} − 50		ns
Address Float after $\overline{RD}$	t_{RLAZ}		0		0	ns

External Clock Drive

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 0.5 MHz to 12 MHz		
		Min	Max	
Oscillator Period	t_{CLCL}	83.3	2000	ns
High Time	t_{CHCX}	20		ns
Low Time	t_{CLCX}	20		ns
Rise Time	t_{CLCH}		20	ns
Fall Time	t_{CHCL}		20	ns
Oscillator Frequency	$1/t_{CLCL}$	0.5	12	MHz

A.C. Characteristics for SAB 80C52-16/80C32-16

V_{CC} = 5V ±10%; V_{SS} = 0V,
T_A = 0°C to 70°C (SAB 80C52);
T_A = −40 to +85°C (SAB 80C52-T40/85)
(C_L for Port 0, ALE and $\overline{\text{PSEN}}$ Outputs = 100 pF; C_L for All Other Outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		16 MHz Clock		Variable Clock $1/t_{CLCL}$ = 0.5 MHz to 16 MHz		
		Min	Max	Min	Max	
ALE Pulse Width	t_{LHLL}	85		$2\ t_{CLCL} - 40$		ns
Address Setup to ALE	t_{AVLL}	33		$t_{CLCL} - 30$		ns
Address Hold after ALE	t_{LLAX1}	28		$t_{CLCL} - 35$		ns
ALE to Valid Instruction in	t_{LLIV}		150		$4\ t_{CLCL} - 100$	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	38		$t_{CLCL} - 25$		ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	153		$3\ t_{CLCL} - 35$		ns
$\overline{\text{PSEN}}$ to Valid Instruction in	t_{PLIV}		88		$3\ t_{CLCL} - 100$	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	t_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	t_{PXIZ} (1)		48		$t_{CLCL} - 15$	ns
Address Valid after $\overline{\text{PSEN}}$	t_{PXAV} (1)	60		$t_{CLCL} - 3$		ns
Address to Valid Instruction in	t_{AVIV}		223		$5\ t_{CLCL} - 90$	ns
Address Float to $\overline{\text{PSEN}}$	t_{PLAZ}	0		0		ns

NOTE:
1. Interfacing the SAB 80C52-16 to devices with float times up to 55 ns is permissable. This limited bus contention will not cause any damage to Port 0 drivers.

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		16 MHz Clock		Variable Clock $1/t_{CLCL}$ = 0.5 MHz to 16 MHz		
		Min	Max	Min	Max	
$\overline{RD}$ Pulse Width	t_{RLRH}	275		$6\,t_{CLCL} - 100$		ns
$\overline{WR}$ Pulse Width	t_{WLWH}	275		$6\,t_{CLCL} - 100$		ns
Address Hold after ALE	t_{LLAX2}	90		$2\,t_{CLCL} - 35$		ns
$\overline{RD}$ to Valid Data in	t_{RLDV}		148		$5\,t_{CLCL} - 165$	ns
Data Hold after $\overline{RD}$	t_{RHDX}	0		0		ns
Data Float after $\overline{RD}$	t_{RHDZ}		55		$2\,t_{CLCL} - 70$	ns
ALE to Valid Data in	t_{LLDV}		350		$8\,t_{CLCL} - 150$	ns
Address to Valid Data in	t_{AVDV}		398		$9\,t_{CLCL} - 165$	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	138	238	$3\,t_{CLCL} - 50$	$3\,t_{CLCL} + 50$	ns
Address to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	120		$4\,t_{CLCL} - 130$		ns
$\overline{WR}$ or $\overline{RD}$ High to ALE High	t_{WHLH}	23	103	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
Data Valid to $\overline{WR}$ Transition	t_{QVWX}	13		$t_{CLCL} - 50$		ns
Data Setup before $\overline{WR}$	t_{QVWH}	288		$7\,t_{CLCL} - 150$		ns
Data Hold after $\overline{WR}$	t_{WHQX}	13		$t_{CLCL} - 50$		ns
Address Float after $\overline{RD}$	t_{RLAZ}		0		0	ns

External Clock Drive XTAL1

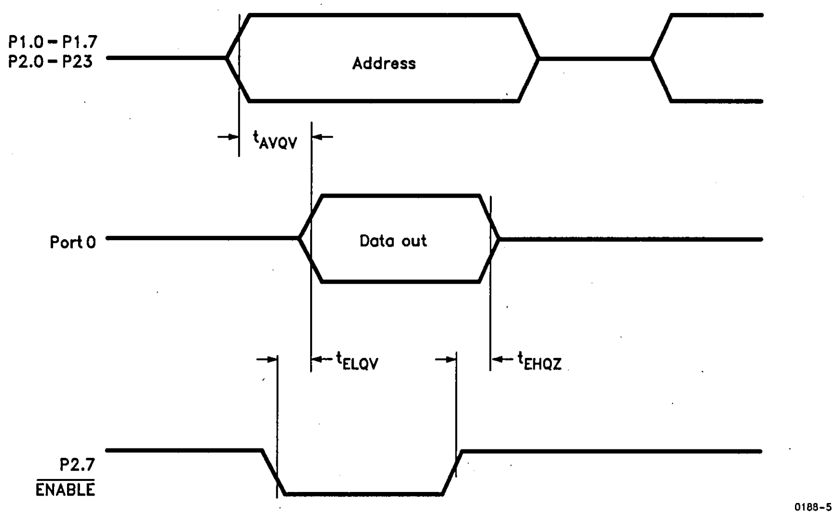
Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 0.5 MHz to 16 MHz		
		Min	Max	
Oscillator Period	t_{CLCL}	62.5	833.3	ns
High Time	t_{CHCX}	15	$t_{CLCL} - t_{CLCX}$	ns
Low Time	t_{CLCX}	15	$t_{CLCL} - t_{CHCX}$	ns
Rise Time	t_{CLCH}		15	ns
Fall Time	t_{CHCL}		15	ns

ROM Verification Characteristics

T_A = 25°C ±5°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Parameter	Symbol	Limit Values		Unit
		Min	Max	
Address to Valid Data	t_{AVQV}		$48\,t_{CLCL}$	ns
$\overline{ENABLE}$ to Valid Data	t_{ELQV}		$48\,t_{CLCL}$	ns
Data Float after $\overline{ENABLE}$	t_{EHQZ}	0	$48\,t_{CLCL}$	ns
Oscillator Frequency	$1/t_{CLCL}$	4	6	MHz

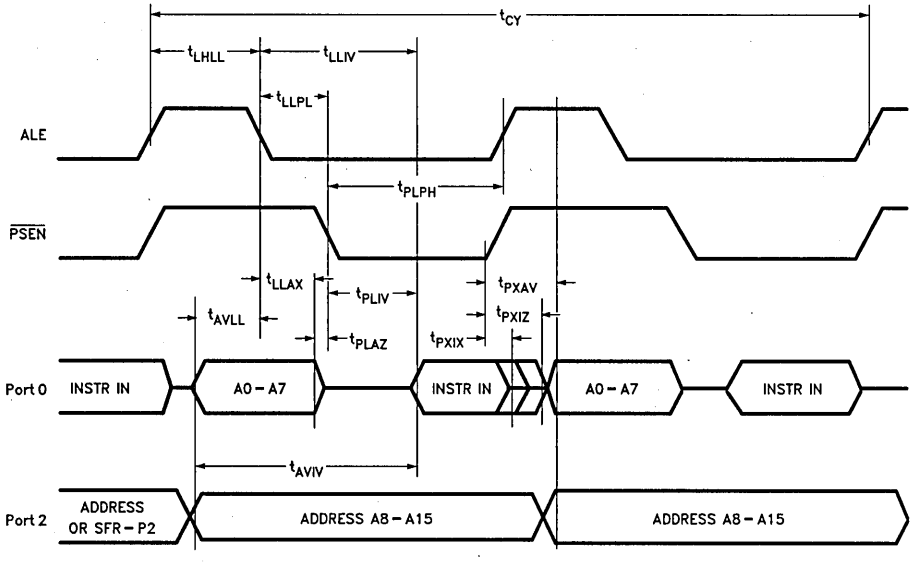
ROM Verification



Address:	P1.0–P1.7 = A0–A7	Inputs:	P2.4–P2.6, $\overline{\text{PSEN}}$ = V_{SS}
	P2.0–P2.3 = A8–A11		ALE, $\overline{\text{EA}}$ = V_{IH}
Data:	Port 0 = D0–D7		RST = V_{IH1}

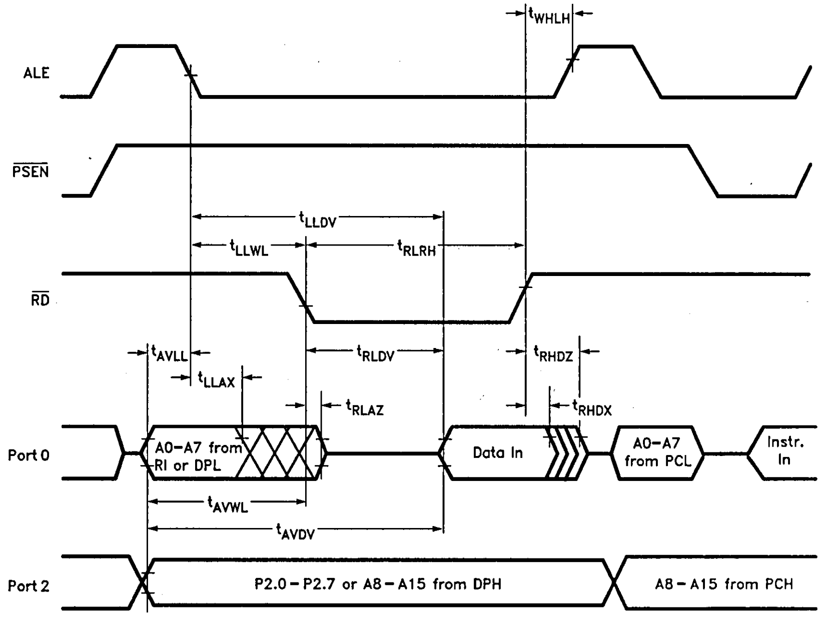
Waveforms

Program Memory Read Cycle



0188-6

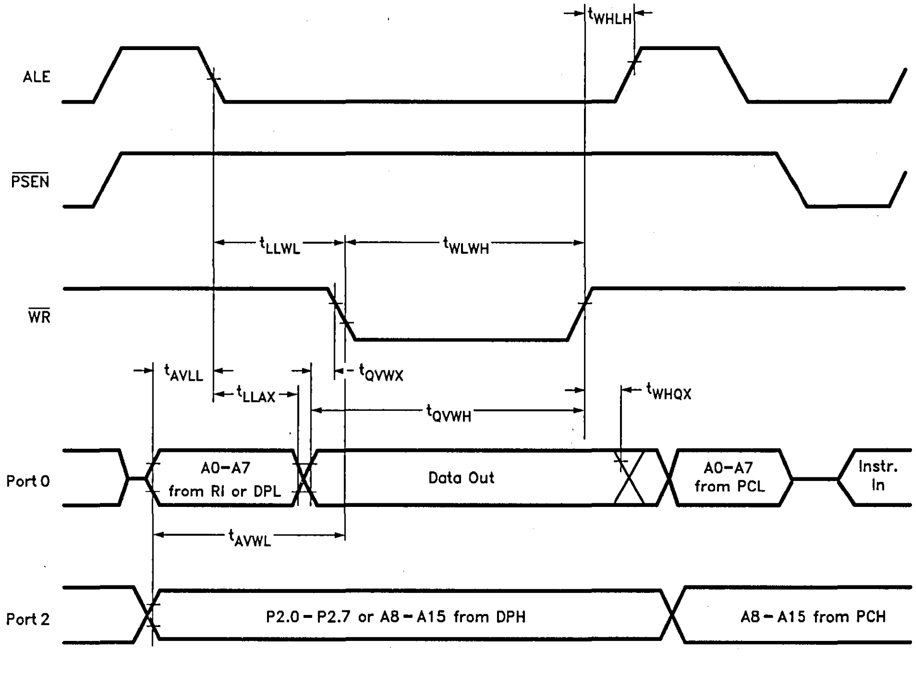
Data Memory Read Cycle



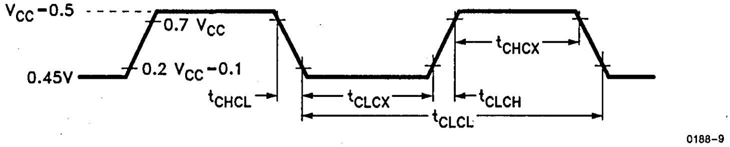
0188-7

Waveforms (Continued)

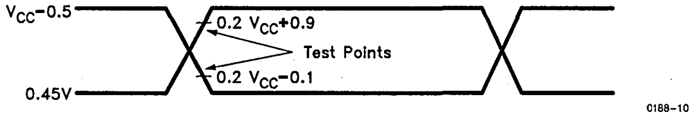
Data Memory Write Cycle



External Clock Cycle



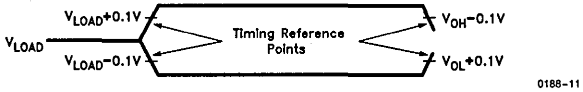
A.C. Testing: Input, Output Waveforms



NOTE:
A.C. Inputs during testing are driven at V_{CC} − 0.5V for a logic "1" and 0.45V for a logic "0".
Timing measurements are made at $V_{IH\ Min}$ for a logic "1" and $V_{IL\ Max}$ for a logic "0".

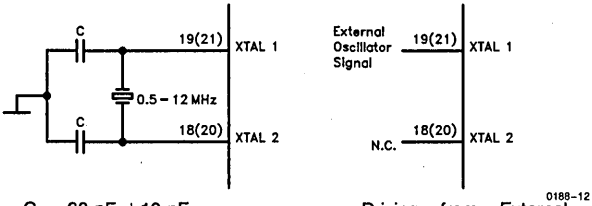
Waveforms (Continued)

A.C. Testing: Float Waveforms



NOTE:
For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

Recommended Oscillator Circuits



C = 30 pF ±10 pF
Crystal Oscillator Mode

Driving from External Source

Pin numbers in (. . .) are for PLCC44 Package

Ordering Information

Type	Function
	8-Bit CMOS Microcontroller
SAB 80C52-P	with Mask-Programmable ROM (P-DIP 40)
SAB 80C32-P	for External Memory (P-DIP 40)
SAB 80C52-N	with Mask-Programmable ROM (PLCC 44)
SAB 80C52-N	for External Memory (PLCC 44)
SAB 80C52-16-P	with Mask-Programmable ROM (P-DIP 40), 16 MHz
SAB 80C32-16-P	for External Memory (P-DIP 40), 16 MHz
SAB 80C52-16-N	with Mask-Programmable ROM (PLCC 44), 16 MHz
SAB 80C32-16-N	for External Memory (PLCC 44), 16 MHz
SAB 80C52-P-T40/85	Ext. Temp −40°C to +85°C (P-DIP 40), 12 MHz
SAB 80C32-P-T40/85	Ext. Temp. −40°C to +85°C (P-DIP 40), 12 MHz
SAB 80C52-N-T40/85	Ext. Temp. −40°C to +85°C (PLCC-44), 12 MHz
SAB 80C32-N-T40/85	Ext. Temp. −40°C to +85°C (PLCC-44), 12 MHz
SAB 80C52-P-T40/110	Ext. Temp. −40°C to +110°C (P-DIP 40), 12 MHz
SAB 80C32-P-T40/110	Ext. Temp. −40°C to +110°C (P-DIP 40), 12 MHz
SAB 80C52-N-T40/110	Ext. Temp. −40°C to +110°C (PLCC-44), 12 MHz
SAB 80C32-N-T40/110	Ext. Temp. −40°C to +110°C (PLCC-44), 12 MHz
SAB 80C52-16-P-T40/85	Ext. Temp. −40°C to +85°C (P-DIP 40), 16 MHz
SAB 80C32-16-P-T40/85	Ext. Temp. −40°C to +85°C (P-DIP 40), 16 MHz
SAB 80C52-16-N-T40/85	Ext. Temp. −40°C to +85°C (PLCC-44), 16 MHz
SAB 80C32-16-N-T40/85	Ext. Temp. −40°C to +85°C (PLCC-44), 16 MHz

SIEMENS

SAB 80C515/80C535, SAB 80C515-16/80C535-16 SAB 80C515-16/80C535-16 Ext. Temp 8-Bit CMOS Microcontroller

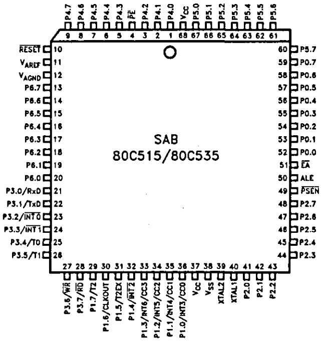
SAB 80C515 (-16)	CMOS microcontroller with factory-mask programmable ROM
SAB 80C535 (-16)	CMOS microcontroller for external ROM
SAB 80C515-T40/85 (-16) **SAB 80C535-T40/85 (-16)**	Extended temperature range: −40°C to +85°C

- SAB 80C515/80C535 (-T40/85) for 12 MHz Operation
- SAB 80C515/80C535 (-T40/85) -16 for 16 MHz Operation
- 8K × 8 ROM (SAB 80C515 only)
- 256 × 8 RAM
- Six 8-Bit I/O Ports, One Input Port for Digital or Analog Input
- Three 16-Bit Timer/Counters
- Highly Flexible Reload, Capture, Compare Capabilities
- Full-Duplex Serial Channel
- Boolean Processor
- 8-Bit A/D Converter with 8 Multiplexed Inputs and Programmable Internal Reference Voltages
- 16-Bit Watchdog Timer
- Twelve Interrupt Vectors, Four Priority Levels
- 256 Bit-Addressable Locations
- Most Instructions Execute in 1 μs (750 ns)
- 4 μs (3 μs) Multiply and Divide
- External Memory Expandable up to 128 Kbytes
- Backwardly Compatible with SAB 8051
- Functionally Compatible with SAB 80515 (NMOS)
- Idle and Power-Down Mode
- 68-Pin Plastic Leaded Chip Carrier Package (PLCC 68)

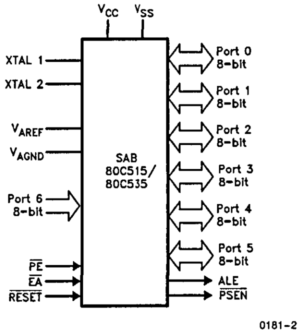
4

Pin Configuration

PLCC 68



Logic Symbol



The SAB 80C515/80C535 is a new, powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is designed in Siemens ACMOS technology and is functionally compatible with the SAB 80515/80535 devices designed in MYMOS technology.

The SAB 80C515/80C535 with the suffix "-16" operates up to 16 MHz crystal oscillator frequency. The part without the suffix operates up to a maximum frequency of 12 MHz.

The SAB 80C515/80C535 is a stand-alone, high-performance single-chip microcontroller based on the SAB 8051/80C51 architecture. While maintaining all the SAB 80C51 operating characteristics, the SAB 80C515/80C535 incorporates several enhancements which significantly increase design flexibility and overall system performance.

In addition, the low-power properties of Siemens ACMOS technology allow applications where power consumption and dissipation are critical. Furthermore, the SAB 80C515/80C535 has two software-selectable modes of reduced activity for further power reduction: idle and power-down mode.

The SAB 80C535 is identical with the SAB 80C515 except that it lacks the on-chip program memory. The SAB 80C515/80C535 is supplied in a 68-pin plastic leaded chip carrier package (PLCC 68). For the industrial temperature range −40°C to +85°C, the SAB 80C515/80C535-T40/85 and SAB 80C515/80C535-16-T40/85 are also available.

Pin Definitions and Functions

Symbol	Pin	Input (I) Output (O)	Function
P4.0–P4.7	1–3,5–9	I/O	Port 4 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 4 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 4 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors.
$\overline{PE}$	4	I	A low level on this pin enables the use of the power saving modes (idle mode and power-down mode). When $\overline{PE}$ is held on high level it is impossible to enter the power saving modes.
$\overline{RESET}$	10	I	A low level of this pin for the duration of two machine cycles while the oscillator is running resets the SAB 80C515. A small internal pullup resistor permits power-in reset using only a capacitor connected to V_{SS}.
V_{AREF}	11		Reference voltage for the A/D converter
V_{AGND}	12		Reference ground for the A/D converter
P6.7–P6.0	13–20	I	Port 6 is an 8-bit undirectional input port. Port pins can be used for digital input if voltage levels simultaneously meet the specifications for high/low input voltages and for the eight multiplexed analog inputs of the A/D converter.

Pin Definitions and Functions (Continued)

Symbol	Pin	Input (I) Output (O)	Function
P3.0–P3.7	21–38	I/O	Port 3 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 3 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 3 also contains the interrupt, timer, serial port and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: —RxD (P3.0): serial port's receiver data input (asynchronous) or data input/output (synchronous) —TxD (P3.1): serial port's transmitter data output (asynchronous) or clock output (synchronous) —$\overline{\text{INT0}}$ (P3.2): interrupt 0 input/timer 0 gate control input —$\overline{\text{INT1}}$ (P3.3): interrupt 1 input/timer 1 gate control input —T0 (P3.4): counter 0 input —T1 (P3.5): counter 1 input —$\overline{\text{WR}}$ (P3.6): the write control signal latches the data byte from port 0 into the external data memory —$\overline{\text{RD}}$ (P3.7): the read control signal enables the external data memory to port 0
P1.7–P1.0	29–36	I/O	Port 1 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 1 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. The port is used for the low-order address byte during program verification. Port 1 also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate (except when used for the compare functions). The secondary functions are assigned to the port 1 pins as follows: —$\overline{\text{INT3}}$/CC0 (P1.0): interrupt 3 input/compare 0 output/ capture 0 input —INT4/CC1 (P1.1): interrupt 4 input/compare 1 output/ capture 1 input —INT5/CC2 (P1.2): interrupt 5 input/compare 2 output/ capture 2 input —INT6/CC3 (P1.3): interrupt 6 input/compare 3 output/ capture 3 input —$\overline{\text{INT2}}$ (P1.4): interrupt 2 input —T2EX (P1.5): timer 2 external reload trigger input —CLKOUT (P1.6): system clock output —T2 (P1.7): counter 2 input

Pin Definitions and Functions (Continued)

Symbol	Pin	Input (I) Output (O)	Function
V_{CC}	37		Supply voltage during normal, idle, and power down operation. Internally connected to pin 68.
V_{SS}	38		GROUND (0V)
XTAL2 XTAL1	39 40		XTAL2 Input to the inverting oscillator amplifier and input to the internal clock generator circuits. XTAL1 Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL2 should be driven, while XTAL1 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times and rise/fall times specified in the AC characteristics must be observed.
P2.0–P2.7	41–48	I/O	Port 2 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 2 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullup resistors when issuing 1s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function register.
$\overline{\text{PSEN}}$	49	O	The program store enable output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. The signal remains high during internal program execution.
ALE	50	O	Provides address latch enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods, except during an external data memory access.
$\overline{\text{EA}}$	51	I	When held high, the SAB 80C515 executes instructions from the internal ROM as long as the PC is less than 8192. When held low, the SAB 80C515 fetches all instructions from external program memory. For the SAB 80C535 this pin must be tied low.
P0.0–P0.7	52–59	I/O	Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application it uses strong internal pullup resistors when issuing 1s. Port 0 also outputs the code bytes during program verification in the SAB 80C515. External pullup resistors are required during program verification.

Pin Definitions and Functions (Continued)

Symbol	Pin	Input (I) Output (O)	Function
P5.7–P5.0	60–67	I/O	Port 5 is an 8-bit bidirectional I/O port with internal pullup resistors. Port 5 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 5 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors.
V_{CC}	68		Supply voltage during normal, idle and power-down operations. Internally connected to pin 37.

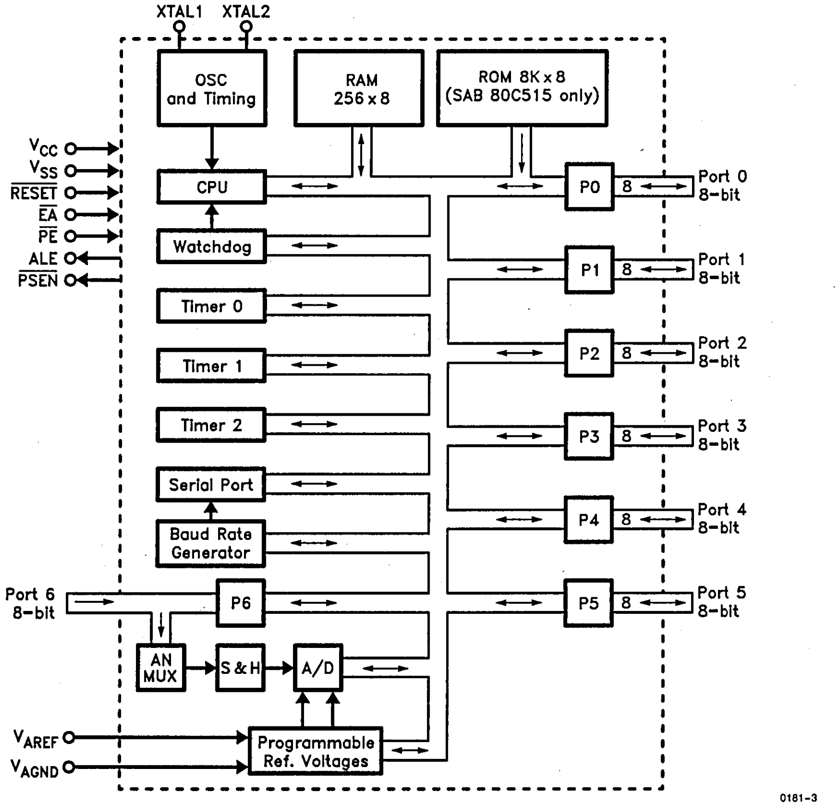


Figure 1. Block Diagram

Functional Description

The members of the SAB 80515 family of microcontrollers are:

— SAB 80C515: Microcontroller, designed in Siemens ACMOS technology, with 8 Kbyte factory mask-programmable ROM

— SAB 80C535: ROM-less version of the SAB 80C515

— SAB 80515: Microcontroller, designed in Siemens MYMOS technology, with 8 Kbyte factory mask-programmable ROM

— SAB 80535: ROM-less version of the SAB 80515

— SAB 80515K: Special ROM-less version of the SAB 80515 with an additional interface for program memory accesses. An external ROM that is accessed via this interface substitutes the SAB 80515's internal ROM.

The SAB 80C535 is identical to the SAB 80C515, except that it lacks the on-chip ROM. In this data sheet the term "SAB 80C515" is used to refer to both the SAB 80C515 and SAB 80C535, unless otherwise noted.

Architecture

The architecture of the SAB 80C515 is based on the SAB 8051/SAB 80C51 microcontroller family. The following features of the SAB 80C515 are fully compatible with the SAB 80C51 features:

— Instruction set

— External memory expansion interface (port 0 and port 2)

— Full-duplex serial port

— Timer/counter 0 and 1

— Alternate functions on port 3

— The lower 128 bytes of internal RAM and the lower 4 Kbytes of internal ROM

The SAB 80C515 additionally contains 128 bytes of internal RAM and 4 Kbytes of internal ROM, which results in a total of 256 bytes of RAM and 8 Kbytes of ROM on chip. The SAB 80C515 has a new 16-bit timer/counter with a 2:1 prescaler, reload mode, compare and capture capability. It also contains a 16-bit watchdog timer, an 8-bit A/D converter with programmable reference voltages, two additional quasi-bidirectional 8-bit ports, one 8-bit input port for analog or digital signals, and a programmable clock output ($f_{OSC}/12$).

Furthermore, the SAB 80C515 has a powerful interrupt structure with 12 vectors and 4 programmable priority levels.

Figure 1 shows a block diagram of the SAB 80C515.

CPU

The SAB80C515 is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in its bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12 MHz crystal, 58% of the instructions execute in 1.0 μs.

All registers, except the program counter and the four 8-register banks, reside in the special function register area. The 42 special functions registers include arithmetic registers, pointers, and registers that provide an interface between the CPU and the on-chip peripheral functions. There are also 128 directly addressable bits within the SFR area. The special function registers are listed in Table 1.

Table 1. Special Function Registers

Symbol	Name	Address
*P0	Port 0	80H
SP	Stack Pointer	81H
DPL	Data Pointer, Low Byte	82H
DPH	Data Pointer, High Byte	83H
PCON	Power Control Register	87H
*TCON	Timer Control Register	88H
TMOD	Timer Mode Register	89H
TL0	Timer 0, Low Byte	8AH
TL1	Timer 1, Low Byte	8BH
TH0	Timer 0, High Byte	8CH
TH1	Timer 1, High Byte	8DH
*P1	Port 1	90H
*SCON	Serial Channel Control Register	98H
SBUF	Serial Channel Buffer Register	99H
*P2	Port 2	0A0H
*IEN0	Interrupt Enable Register 0	0A8H
IP0	Interrupt Priority Register 0	0A9H
*P3	Port 3	0B0H
*IEN1	Interrupt Enable Register 1	0B8H
IP1	Interrupt Priority Register 1	0B9H
*IRCON	Interrupt Request Control Register	0C0H
CCEN	Compare/Capture Enable Register	0C1H
CCL1	Compare/Capture Register 1, Low Byte	0C2H
CCH1	Compare/Capture Register 1, High Byte	0C3H
CCL2	Compare/Capture Register 2, Low Byte	0C4H
CCH2	Compare/Capture Register 2, High Byte	0C5H
CCL3	Compare/Capture Register 3, Low Byte	0C6H
CCH3	Compare/Capture Register 3, High Byte	0C7H
*T2CON	Timer 2 Control Register	0C8H
CRCL	Compare/Reload/Capture Register, Low Byte	0CAH
CRCH	Compare/Reload/Capture Register, High Byte	0CBH
TL2	Timer 2, Low Byte	0CCH
TH2	Timer 2, High Byte	0CDH
*PSW	Program Status Word Register	0D0H
*ADCON	A/D Converter Control Register	0D8H
ADDAT	A/D Converter Data Register	0D9H
DAPR	D/A Converter Program Register	0DAH
P6	Port 6	0DBH
*ACC	Accumulator	0E0H
*P4	Port 4	0E8H
*B	B-Register	0F0H
*P5	Port 5	0F8H

The SFR's marked with an asterisk (*) are bit and byte-addressable.

I/O Ports

The SAB 80C515 has six 8-bit I/O ports and one 8-bit input port. Port 0 is an open-drain bidirectional I/O port, while ports 1 to 5 are quasi-bidirectional I/O ports with internal pull-up resistors. That means, when configured as inputs, ports 1 to 5 will be pulled high and will source current when externally pulled low. Port 0 will float when configured as input.

Port 0 and port 2 can be used to expand the program and data memory externally. During an access to external memory, port 0 emits the low-order address byte and reads/writes the data byte, while port 2 emits the high-order address byte. In this function, port 0 is not an open-drain port, but uses a strong internal pullup FET.

Ports 1 and 3 are provided for several alternate functions, as listed below:

Port	Symbol	Function
P1.0	$\overline{\text{INT3}}$/CC0	External interrupt 3 input, compare 0 output, capture 0 input
P1.1	INT4/CC1	External interrupt 4 input, compare 1 output, capture 1 input
P1.2	INT5/CC2	External interrupt 5 input, compare 2 output, capture 2 input
P1.3	INT6/CC3	External interrupt 6 input, compare 3 output, capture 3 input
P1.4	$\overline{\text{INT2}}$	External interrupt 2 input
P1.5	T2EX	Timer 2 external reload trigger input
P1.6	CLKOUT	System clock output
P1.7	T2	Timer 2 external counter input
P3.0	RXD	Serial port's receiver data input (asynchronous) or data input/output (synchronous)
P3.1	TXD	Serial port's transmitter data output (asynchronous) or clock output (synchronous)
P3.2	$\overline{\text{INT0}}$	External interrupt 0 input, timer 0 gate control
P3.3	$\overline{\text{INT1}}$	External interrupt 1 input, timer 1 gate control
P3.4	T0	Timer 0 external counter input
P3.5	T1	Timer 1 external counter input
P3.6	$\overline{\text{WR}}$	External data memory write strobe
P3.7	$\overline{\text{RD}}$	External data memory read strobe

The SAB 80C515 has a dual-purpose input port. As the ANx lines in the SAB 80515 (NMOS version), the eight port lines at port 6 can be used as analog inputs. But if the input voltages at port 6 meet the specified digital input levels (V_{IL} and V_{IH}), the port can also be used as digital input port. Reading the special function register P6 allows the user to input the digital values currently applied to the port pins. It is not necessary to select these modes by software; the voltages applied at port 6 pins can be converted to digital values using the A/D converter and at the same time the pins can be read via SFR P6. It must be noted, however, that the results in port P6 bits will be indeterminate if the levels at the corresponding pins are not within their respective VIL/VIH specifications. Futhermore, it is not possible to use port P6 as output lines. Special function register P6 is located at address 0DBH.

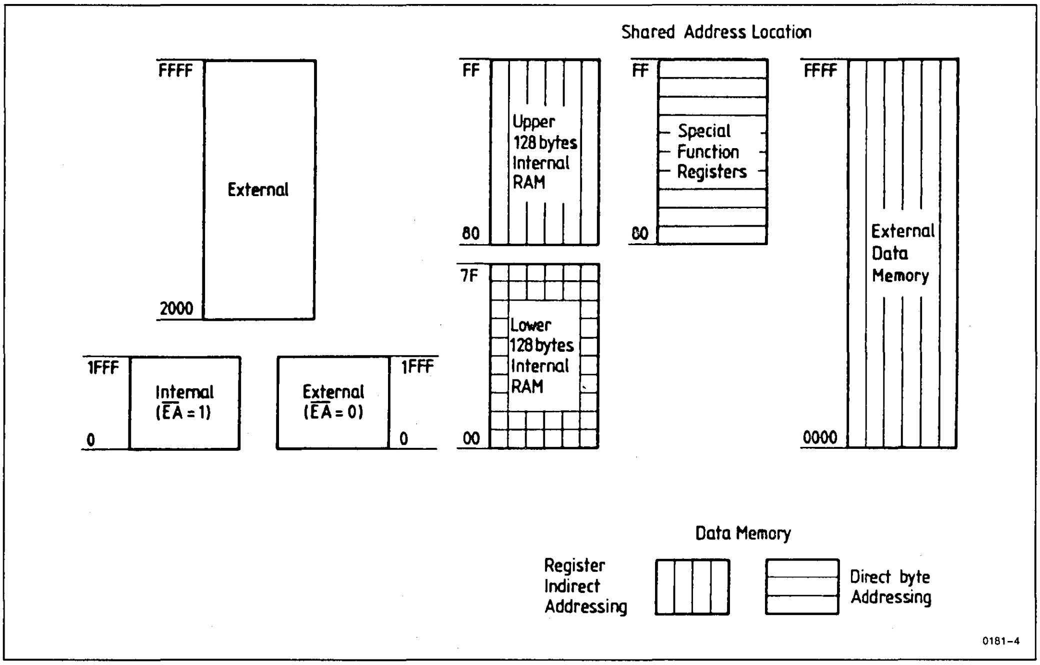


Figure 2. Memory Address Spaces

A/D Converter

The 8-bit A/D converter of the SAB 80C515 has eight multiplexed analog inputs (Port 6) and uses the successive approximation method.

It takes 5 machine cycles to sample an analog signal (during this sample time the input signal should be held constant); the total conversion time (including sample time) is 13 machine cycles (13 μs at 12 MHz oscillator frequency). Conversion can be programmed to be single or continuous; at the end of a conversion an interrupt can be generated.

A unique feature is the capability of internal reference voltage programming. The internal reference voltages $V_{IntAREF}$ and $V_{IntAGND}$ for the A/D converter both are programmable to one of 16 steps with respect to the external reference voltages. This feature permits a conversion with a smaller internal reference voltage range to gain a higher resolution. In addition, the internal reference voltages can easily be adapted by software to the desired analog input voltage range.

Figure 3 shows a block diagram of the A/D converter.

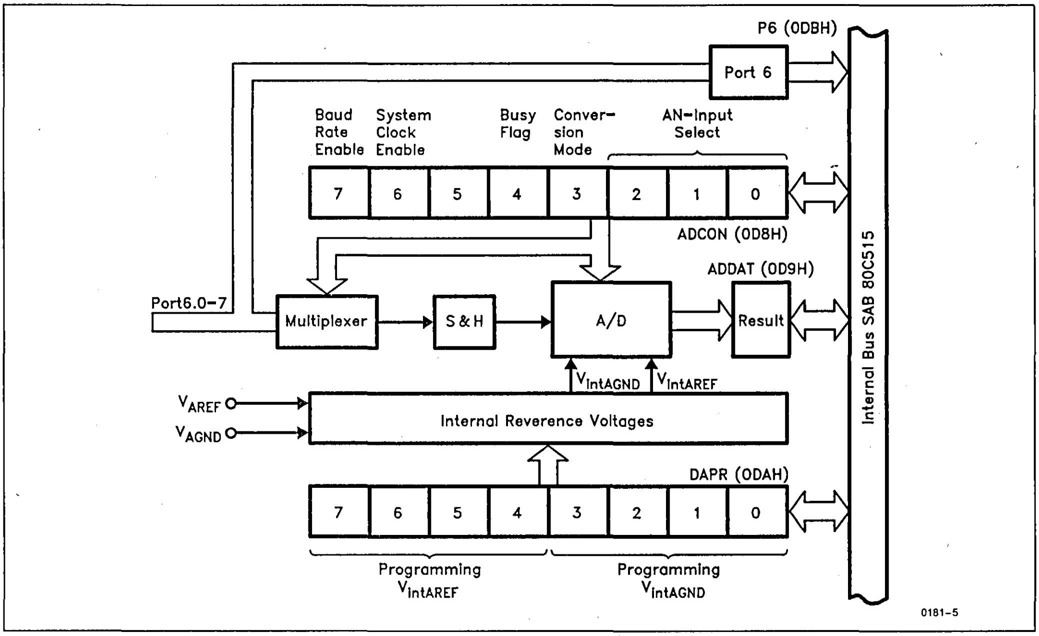


Figure 3. Block Diagram of the A/D Converter

Watchdog Timer

This feature is provided as a means of graceful recovery from a software upset. After a reset, the watchdog timer is cleared and stopped. It can be started and cleared by software, but it cannot be stopped during active mode of the device. If the software fails to clear the watchdog timer at least every 65532 machine cycles (about 65 ms if a 12 MHz oscillator frequency is used), a hardware reset be initiated. The reset cause (external reset or reset caused by the watchdog) can be examined by software. To clear the watchdog, two bits in two different special function registers must be set by two consecutive instructions (bits IEN0.6 and IEN1.6). This is done to prevent the watchdog from being cleared by unexpected opcodes.

It must be noted, however, that the watchdog timer is halted during the idle mode and power down mode of the processor (see section "power saving modes" below). Therefore it is possible to use the idle mode in combination with the watchdog timer function. But even the watchdog timer cannot reset the device when one of the power saving modes is entered accidentally.

For these reasons several precautions are taken against unintentional entering of the power-down or idle mode (see below).

Power Saving Modes

The ACMOS technology of the SAB 80C515 allows two new power saving modes of the device: The idle mode of the power-down mode. These modes replace the power-down supply mode via pin V_{PD} of the SAB 80515 (NMOS). The SAB 80C515 is supplied via pins V_{CC} also during idle and power down operation.

However, there are applications where unintentional entering of these power saving modes must be absolutely avoided. Such critical applications often use the watchdog timer to prevent the system from program upsets. Then accidental entering of the power saving modes would even stop the watchdog timer and would circumvent the watchdog timer's task of system protection.

Thus, the SAB 80C515 has an extra pin that allows it to disable both of the power saving modes. When pin $\overline{PE}$ is held high, idle mode and power-down mode are completely disabled and the instruction sequences that are used for entering these modes (see below) will NOT affect the normal operation of the device. When $\overline{PE}$ is held low, the use of the idle mode and of power-down mode is possible as described in the following sections.

Pin $\overline{PE}$ has a weak internal pullup resistor. Thus, when left open, the power saving modes are disabled.

The Special Function Register PCON

In the NMOS version SAB 80515 the SFR PCON (address 87H) contains only bit SMOD; in the CMOS version SAB 80C515 there are more bits used (see Table 2).

The bits PDE, PDS and IDLE, IDLS select the power-down mode or the idle mode, respectively, when the use of the power saving modes is enabled by pin $\overline{PE}$ (see below). If the power-down mode and the idle mode are set at the same time, power-down takes precedence.

Futhermore, register PCON contains two general purpose flags. For example, the flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during an Idle. Then an instruction that activates Idle can also set one or both flag bits. When Idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

The reset value of PCON is 000X0000B.

Idle Mode

In the idle mode the oscillator of the SAB 80C515 continues to run, but the CPU is gated off from the clock signal. However, the interrupt system, the serial port, the A/D converter, and all timers with the exception of the watchdog timer are further provided with the clock. The CPU status is preserved in its entirety: the stack pointer, program counter, program status word, accumulator, and all other registers maintain their data during idle mode.

Table 2. SFR PCON (87H)

SMOD	PDS	IDLS	—	GF1	GF0	PDE	IDLE	87H
7	6	5	4	3	2	1	0	

Symbol	Position	Function
SMOD	PCON.7	When set, the baud rate of the serial channel in mode 1, 2, 3 is doubled.
PDS	PCON.6	Power-down start bit. The instruction that sets the PDS flag bit is the last instruction before entering the power-down mode.
IDLS	PCON.5	Idle start bit. The instruction that sets the IDLS flag bit is the last instruction before entering the idle mode.
—	PCON.4	Reserved
GF1	PCON.3	General Purpose Flag
GF0	PCON.2	General Purpose Flag
PDE	PCON.1	Power-down enable bit. When set, starting of the power-down mode is enabled.
IDLE	PCON.0	Idle mode enable bit. When set, starting of the idle mode is enabled.

The reduction of power consumption, which can be achieved by this feature depends on the number of peripherals running.

If all timers are stopped and the A/D converter and the serial interface are not running, the maximum power reduction can be achieved. This state is also the test condition of the idle mode I_{CC} (see DC characteristics, Note 5).

So the user has to take care which peripheral should continue to run and which has to be stopped during idle mode. Also the state of all port pins—either the pins controlled by their latches or controlled by their secondary functions—depends on the status of the controller when entering idle mode.

Normally the port pins hold the logical state they had at the time idle mode was activated. If some pins are programmed to serve their alternate functions they still continue to output during idle mode if the assigned function is on. This applies to the compare outputs as well as to the clock output signal or to the serial interface in case it cannot finish reception or transmission during normal operation. The control signals ALE and $\overline{\text{PSEN}}$ hold at logic high levels (see Table 3).

As in normal operation mode, the ports can be used as inputs during idle mode. Thus a capture or reload operation can be triggered, the timers can be used to count external events, and external interrupts will be detected.

The idle mode is a useful feature which makes it possible to "freeze" the processor's status—either for a pre-defined time, or until an external event reverts the controller to normal operation, as discussed below. The watchdog timer is the only peripheral which is automatically stopped during idle mode. If it were not disabled on entering idle mode, the watchdog timer would reset the controller, thus abandoning the idle mode.

When idle mode is used, pin $\overline{\text{PE}}$ must be held on low level. The idle mode is then entered by two consecutive instructions. The first instruction sets the flag bit IDLE (PCON.0) and must not set bit IDLS (PCON.5), the following instruction sets the start bit IDLS (PCON.5) and must not set bit IDLE (PCON.0). The hardware ensures that a concurrent setting of both bits, IDLE and IDLS, does not initiate the idle mode. Bits IDLE and IDLS will automatically be cleared after being set. If one of these register bits is read the value that appears is 0 (see Table 2). This double instruction is implemented to minimize the chance of an unintentional entering of the idle mode which would leave the watchdog timer's task of system protection without effect.

Note that PCON is not a bit-addressable register, so the above mentioned sequence for entering the idle mode is obtained by byte-handling instructions, as shown in the following example:

```
ORL PCON,00000001B    ;Set bit IDLE, bit IDLS
                       must not be set
ORL PCON,00100000B    ;Set bit IDLS, bit IDLE
                       must not be set
```

The instruction that sets bit IDLS is the last instruction executed before going into idle mode.

Table 3. Status of External Pin during Idle and Power-Down Modes

Outputs	Last Instruction Executed from Internal Code Memory		Last Instruction Executed from External Code Memory	
	Idle	Power-Down	Idle	Power-Down
ALE	HIGH	LOW	HIGH	LOW
$\overline{\text{PSEN}}$	HIGH	LOW	HIGH	LOW
PORT 0	Data	Data	Float	Float
PORT 1	Data/Alternate Outputs	Data/Last Output	Data/Alternate Outputs	Data/Last Output
PORT 2	Data	Data	Address	Data
PORT 3	Data/Alternate Outputs	Data/Last Output	Data/Alternate Outputs	Data/Last Output
PORT 4	Data	Data	Data	Data
PORT 5	Data	Data	Data	Data

There are two ways to terminate the idle mode:

— The idle mode can be terminated by activating any enabled interrupt. This interrupt will be serviced and normally the instruction to be executed following the RETI instruction will be the one following the instruction that sets the bit IDLS.

— The other way to terminate the idle mode, is a hardware reset. Since the oscillator is still running, the hardware reset must be held active only for two machine cycles for a complete reset.

Power-Down Mode

In the power-down mode, the on-chip oscillator is stopped. Therefore all functions are stopped; only the contents of the on-chip RAM and the SFR's are maintained. The port pins controlled by their port latches output the values that are held by their SRF's. The port pins which serve the alternate output functions show the values they had at the end of the last cycle of the instruction which initiated the power-down mode; when the clockout signal (CLKOUT, P1.6) is enabled, it will stop at low level. ALE and $\overline{\text{PSEN}}$ hold at logic low level (see Table 3).

To enter the power-down mode the pin $\overline{\text{PE}}$ must be on low level. The power-down mode then is entered by two consecutive instructions. The first instruction has to set the flag bit PDE (PCON.1) and must not set bit PDS (PCON.6), the following instruction has to set the start bit PDS (PCON.6) and must not set bit PDE (PCON.1). The hardware ensures that a concurrent setting of both bits, PDE and PDS, does not initiate the power-down mode. Bits PDE and PDS will automatically be cleared after having been set and the value shown by reading one of these bits is always 0 (see Table 2). This double instruction is implemented to minimize the chance of unintentionally entering the power-down mode which could possibly "freeze" the chip's activity in an undesired status.

Note that PCON is not a bit-addressable register, so the above mentioned sequence for entering the power-down mode is obtained by byte-handling instructions, as shown in the following example:

```
ORL PCON,00000010B ;Set bit PDE, bit PDS must
                    not be set
ORL PCON,01000000B ;Set bit PDS, bit PDE must
                    not be set
```

The instruction that sets bit PDS is the last instruction executed before going into power-down mode.

The only exit from power-down mode is a hardware reset. Reset will redefine all SFR's, but will not change the contents of the internal RAM.

In the power-down mode of operation, V_{CC} can be reduced to minimize power consumption. It must be ensured, however, that V_{CC} is not reduced before the power-down mode is invoked, and that V_{CC} is restored to its normal operating level, before the power-down mode is terminated. The reset signal that terminates the power-down mode also restarts the oscillator. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (similar to power-on reset).

Differences in Pin Assignments of the SAB 80C515 and SAB 80515

Since the SAB 80C515 is designed in CMOS technology, this device requires no V_{BB} pin, because the die's substrate is internally connected to V_{CC}.

Furthermore, the RAM backup power supply via pin V_{PD} is replaced by the software-controlled power-down mode and power supply via V_{CC}.

Therefore, pins V_{BB} and V_{PD} of the NMOS version SAB 80515 are used for other functions in the SAB 80C515.

Pin 4 (the former pin V_{PD}) is the new $\overline{\text{PE}}$ pin which enables the use of the power saving modes.

Pin 37 (the former pin V_{BB}) becomes an additional V_{CC} pin. Thus, it is possible to insert a decoupling capacitor between pin 37 (V_{CC}) and pin 38 (V_{SS}) very close to the device, thereby avoiding long wiring and reducing the voltage distortion resulting from high dynamic current peaks.

There is a difference between the NMOS and CMOS version concerning the clock circuitry. When the device is driven from an external source, pin XTAL2 must be driven by the clock signal; pin XTAL1, however, must be left be open in the SAB 80C515 (must be tied low in the NMOS version). When using the oscillator with a crystal there is no difference in the circuitry.

Thus, due to its pin compatibility the SAB 80C515 normally substitutes any SAB 80515 without redesign of the user's printed circuit board; but the user has to take care that the two V_{CC} pins are hardwired on-chip. In any case, it is recommended that power is supplied on both V_{CC} pins of the SAB 80C515 to improve the power supply to the chip. If the power saving modes are to be used, pin $\overline{\text{PE}}$ must be tied low, otherwise these modes are disabled.

Absolute Maximum Ratings

Ambient Temperature under Bias
SAB 80C515 .... 0°C to +70°C
SAB 80C515-T40/85 .... −40°C to +85°C
Storage Temperature .... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground (V_{SS}) .. −0.5V to V_{CC} + 0.5V
Voltage on V_{CC} to V_{SS} .... −0.5 to +6.5V
Power Dissipation 2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V;
T_A = 0°C to +70°C for SAB 80C515/80C535 (-16)
T_A = −40°C to +85°C for SAB 80C515/80C535-T40/85 (-16)

Parameter	Symbol	Conditions	Limit Values		Unit
			Min	Max	
Input Low Voltage (except $\overline{\text{EA}}$)	V_{IL}		−0.5	0.2 V_{CC} − 0.1	V
Input Low Voltage ($\overline{\text{EA}}$)	V_{IL1}		−0.5	0.2 V_{CC} − 0.3	V
Input High Voltage (except $\overline{\text{RESET}}$ and XTAL2)	V_{IH}		0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
Input High Voltage to XTAL2	V_{IH1}		0.7 V_{CC}	V_{CC} + 0.5	V
Input High Voltage to $\overline{\text{RESET}}$	V_{IH2}		0.6 V_{CC}	V_{CC} + 0.5	V
Output Low Voltage, Ports 1, 2, 3, 4, 5	V_{OL}	I_{OL} = 1.6 mA (1)		0.45	V
Output Low Voltage, Port 0, ALE, $\overline{\text{PSEN}}$	V_{OL1}	I_{OL} = 3.2 mA (1)		0.45	V
Output High Voltage, Ports 1, 2, 3, 4, 5	V_{OH}	I_{OH} = −80 µA V_{CC} = 5V ±10% I_{OH} = −10 µA	2.4 0.9 V_{CC}		
Output High Voltage (Port 0 in External Bus Mode, ALE, PSEN)	V_{OH1}	I_{OH} = −400 µA, V_{CC} = 5V ±10% I_{OH} = −40 µA (2)	2.4 0.9 V_{CC}		V V
Logic 0 Input Current, Ports 1, 2, 3, 4, 5	I_{IL}	V_{IN} = 0.45V		−50	µA
Input Low Current to $\overline{\text{RESET}}$ for Reset	I_{IL2}	V_{IN} = 0.45V		−100	µA
Logical 1-to-0 Transition Current, Ports 1, 2, 3, 4, 5	I_{TL}	V_{IN} = 2V		−650	µA
Input Leakage Current (Port 0, $\overline{\text{EA}}$)	I_{LI}	0.45 < V_{IN} < V_{CC}		±10	µA
Pin Capacitance	C_{IO}	f_c = 1 MHz, T_A = 25°C		10	pF

D.C. Characteristics (Continued)

V_{CC} = 5V ±10%; V_{SS} = 0V;
T_A = 0°C to +70°C for SAB 80C515/80C535 (-16)
T_A = −40°C to +85°C for SAB 80C515/80C535-T40/85 (-16)

Parameter	Symbol	Conditions	Limit Values		Unit
			Min	Max	
Power Supply Current:					
Active Mode, 12 MHz(6)		V_{CC} = 5V(4)		35	mA
Idle Mode, 12 MHz(6)		V_{CC} = 5V(5)		13	mA
Active Mode, 16 MHz(6)		V_{CC} = 5V(4)		46	mA
Idle Mode, 16 MHz(6)		V_{CC} = 5V(5)		17	mA
Power-Down Mode		V_{CC} = 2V to 5.5V(3)		50	μA

NOTES:

1. Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and ports 1, 3, 4, and 5. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation. In the worst case (capacitive loading > 100 pF), tho noise pulse on ALE line may exceed 0.8V. Then, it may be desirable to qualify ALE with a Schmitt-trigger, or use an address latch with a Schmitt-trigger strobe input.
2. Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and PSEN to momentarily fall below the 0.9 V_{CC} specification when the address bits are stabilizing.
3. Power-down I_{CC} is measured with: $\overline{EA}$ = Port 0 = Port 6 = V_{CC}; XTAL2 = N.C.; XTAL1 = V_{SS}; $\overline{RESET}$ = V_{SS}; all other pins are disconnected.
4. I_{CC} (active mode) is measured with: XTAL2 driven with the clock signal according to the figure below; XTAL1 = N.C.; $\overline{EA}$ = Port 0 = Port 6 = V_{CC}, $\overline{RESET}$ = V_{SS}; all other pins are disconnected. I_{CC} might be slightly higher if a crystal oscillator is used.
5. I_{CC} (idle mode) is measured with: XTAL2 driven with the clock signal according to the figure below; XTAL1 = N.C.; $\overline{EA}$ = V_{SS}; Port 0 = V_{CC}; $\overline{RESET}$; all other pins are disconnected; all on-chip peripherals are disabled.
6. I_{CC} at other frequencies is given by:
   Active Mode: I_{CC} max (mA) = 2.67 *f_{OSC} (MHz) + 3.00
   Idle Mode: I_{CC} max (mA) = 0.88 *f_{OSC} (MHz) + 2.50
   where f_{OSC} is the oscillator frequency in MHz.
   I_{CC} max is given in mA and measured at V_{CC} = 5V (see also notes 4 and 5)
7. The output impedance of the analog source must be low enough to assure full loading of the sample capacitance (C_I) during load time (t_L). After charging of the internal capacitance (C_I) in the load time (t_L), the analog input must be held constant for the rest of the sample time (t_S).
8. The differential impedance r_D of the analog reference voltage source must be less than 1 kΩ at reference supply voltage.
9. Exceeding these limit values at one or more input channels will cause additional current which is sinked/sourced at these channels. This may also affect the accuracy of other channels which are operated within these specifications.

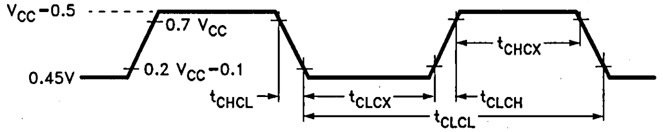
A/D Converter Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; V_{AREF} = V_{CC} ±5%; V_{AGND} = V_{SS} ±0.2V; I_{VAREF}-I_{VAGND} ≥ 1V
T_A = 0°C to +70°C for SAB 80C515/80C535 (-16)
T_A = −40°C to +85°C for SAB 80C515/80C535-T40/85 (-16)

Parameter	Symbol	Conditions	Limit Values			Unit
			Min	Typ	Max	
Analog Input Voltage	V_{AINPUT}	(Note 9)	V_{AGND} − 0.2		V_{VAREF} + 0.2	V
Analog Input Capacitance(6)	C_I	(Note 7)		25		pF
Load Time	T_L				2 T_{CY}	µs
Sample Time (Inc. Load Time)	T_s				7 T_{CY}	µs
Conversion Time (Inc. Sample Time)	T_C				13 T_{CY}	µs
Differential Non-Linearity	DNLE	I_{VAREF} = V_{AREF} = V_{CC} I_{VAGND} = V_{AGND} = V_{SS} (Note 7)		±½	±1	LSB
Integral Non-Linearity	INLE			±½	±1	LSB
Offset Error				±½	±1	LSB
Gain Error				±½	±1	LSB
Total Unadjusted Error	TUE			±1	±2	LSB
V_{AREF} Supply Current(7)	I_{REF}	(Note 8)			5	mA
Internal Reference Error(7)	$V_{intREFERR}$	(Note 8)			TBD	mV

Clock Signal Waveform for I_{CC} Tests in Active and Idle Mode

t_{CLCH} = t_{CHCL} = 5 ns



0181-6

A.C. Characteristics for SAB 80C515/80C535

V_{CC} = 5V ±10%; V_{SS} = 0V
(C_L for Port 0, ALE and $\overline{PSEN}$ Outputs = 100 pF; C_L for All Outputs = 80 pF)
T_A = 0°C to +70°C for SAB 80C515/80C535
T_A = −40°C to +85°C for SAB 80C515/80C535-T40/85

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock 1/T_{CLCL} 0.5 MHz to 12 MHz		
		Min	Max	Min	Max	
Program Memory Characteristics						
ALE Pulse Width	T_{LHLL}	127		2 T_{CLCL} − 40		ns
Address Setup to ALE	T_{AVLL}	53		T_{CLCL} − 30		ns
Address Hold after ALE	T_{LLAX}	48		T_{CLCL} − 35		ns
ALE to Valid Instruction In	T_{LLIV}		233		4 T_{CLCL} − 100	ns
ALE to $\overline{PSEN}$	T_{LLPL}	58		T_{CLCL} − 25		ns

A.C. Characteristics for SAB 80C515/80C535 (Continued)

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/T_{CLCL}$ 0.5 MHz to 12 MHz		
		Min	Max	Min	Max	
Program Memory Characteristics (Continued)						
$\overline{\text{PSEN}}$ Pulse Width	T_{PLPH}	215		$3\,T_{CLCL} - 35$		ns
$\overline{\text{PSEN}}$ to Valid Instruction In	T_{PLIV}		150		$3\,T_{CLCL} - 100$	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	T_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	T_{PXIZ}(1)		63		$T_{CLCL} - 20$	ns
Address Valid after $\overline{\text{PSEN}}$	T_{PXAV}(1)	75		$T_{CLCL} - 8$		ns
Address to Valid Instruction In	T_{AVIV}		302		$5\,T_{CLCL} - 115$	ns
Address Float to $\overline{\text{PSEN}}$	T_{AZPL}	0		0		ns
External Data Memory Characteristics						
$\overline{\text{RD}}$ Pulse Width	T_{RLRH}	400		$6\,T_{CLCL} - 100$		ns
$\overline{\text{WR}}$ Pulse Width	T_{WLWH}	400		$6\,T_{CLCL} - 100$		ns
Address Hold after ALE	T_{LLAX2}	132		$2\,T_{CLCL} - 35$		ns
$\overline{\text{RD}}$ to Valid Data In	T_{RLDV}		252		$5\,T_{CLCL} - 165$	ns
Data Hold after $\overline{\text{RD}}$	T_{RHDX}	0		0		ns
Data Float after $\overline{\text{RD}}$	T_{RHDZ}		97		$2\,T_{CLCL} - 70$	ns
ALE to Valid Data In	T_{LLDV}		517		$8\,T_{CLCL} - 150$	ns
Address to Valid Data In	T_{AVDV}		585		$9\,T_{CLCL} - 165$	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	T_{LLWL}	200	300	$3\,T_{CLCL} - 50$	$3\,T_{CLCL} + 50$	ns
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ High to ALE High	T_{WHLH}	43	123	$T_{CLCL} - 40$	$T_{CLCL} + 40$	ns
Address Valid to $\overline{\text{WR}}$	T_{AVWL}	203		$4\,T_{CLCL} - 130$		ns
Data Valid to $\overline{\text{WR}}$ Transition	T_{QVWX}	33		$T_{CLCL} - 50$		ns
Data Setup before $\overline{\text{WR}}$	T_{QVWH}	433		$7\,T_{CLCL} - 150$		ns
Data Hold after $\overline{\text{WR}}$	T_{WHQX}	33		$T_{CLCL} - 50$		ns
Address Float after $\overline{\text{RD}}$	T_{RLAZ}		0		0	ns
External Clock Drive						
Oscillator Period	T_{CLCL}	83.3	2000			ns
Oscillator Frequency	$1\,T_{CLCL}$	0.5	12			MHz
High Time	T_{CHCX}	20				ns
Low Time	T_{CLCX}	20				ns
Rise Time	T_{CLCH}		20			ns
Fall Time	T_{CHCL}		20			ns

NOTE:
1. Interfacing the SAB 80C515 to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

System Clock Timing

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/T_{CLCL}$ = 0.5 MHz to 12 MHz		
		Min	Max	Min	Max	
ALE to CLKOUT	T_{LLSH}	543		$7\,T_{CLCL} - 40$		ns
CLKOUT High Time	T_{SHSL}	127		$2\,T_{CLCL} - 40$		ns
CLKOUT Low Time	T_{SLSH}	793		$10\,T_{CLCL} - 40$		ns
CLKOUT Low to ALE High	T_{SLLH}	43	123	$T_{CLCL} - 40$	$T_{CLCL} + 40$	ns

A.C. Characteristics for SAB 80C515-16/80C535-16

V_{CC} = 5V ±10%; V_{SS} = 0V
(C_L for Port 0, ALE and $\overline{\text{PSEN}}$ Outputs = 100 pF; C_L for All Outputs = 80 pF)
T_A = 0°C to +70°C for SAB 80C515-16/80C535-16
T_A = −40°C to +85°C for SAB 80C515-16/80C535-16-T40/85

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		16 MHz Clock		Variable Clock $1/T_{CLCL}$ = 0.5 MHz to 16 MHz		
		Min	Max	Min	Max	
ALE Pulse Width	T_{LHLL}	85		$2\,T_{CLCL} - 40$		ns
Address Setup to ALE	T_{AVLL}	33		$T_{CLCL} - 30$		ns
Address Hold after ALE	T_{LLAX}	28		$T_{CLCL} - 35$		ns
ALE to Valid Instruction In	T_{LLIV}		150		$4\,T_{CLCL} - 100$	ns
ALE to $\overline{\text{PSEN}}$	T_{LLPL}	38		$T_{CLCL} - 25$		ns
$\overline{\text{PSEN}}$ Pulse Width	T_{PLPH}	153		$3\,T_{CLCL} - 35$		ns
$\overline{\text{PSEN}}$ to Valid Instruction In	T_{PLIV}		88		$3\,T_{CLCL} - 100$	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	T_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	T_{PXIZ}*		48		$T_{CLCL} - 20$	ns
Address Valid after $\overline{\text{PSEN}}$	T_{PXAV}*	60		$T_{CLCL} - 8$		ns
Address to Valid Instruction In	T_{AVIV}		223		$5\,T_{CLCL} - 115$	ns
Address Float to $\overline{\text{PSEN}}$	T_{AZPL}	0		0		ns

*Interface the SAB 80C515-16 to devices with float times up to 55 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

External Data Memory Characteristics for SAB 80C515-16/80C535-16

Parameter	Symbol	Limit Values				Unit
		16 MHz Clock		Variable Clock $1/T_{CLCL}$ = 0.5 MHz to 16 MHz		
		Min	Max	Min	Max	
$\overline{\text{RD}}$ Pulse Width	T_{RLRH}	275		$6\ T_{CLCL} - 100$		ns
$\overline{\text{WR}}$ Pulse Width	T_{WLWH}	275		$6\ T_{CLCL} - 100$		ns
Address Hold after ALE	T_{LLAX2}	90		$2\ T_{CLCL} - 35$		ns
$\overline{\text{RD}}$ to Valid Data In	T_{RLDV}		148		$5\ T_{CLCL} - 165$	ns
Data Hold after $\overline{\text{RD}}$	T_{RHDX}	0		0		ns
Data Float after $\overline{\text{RD}}$	T_{RHDZ}		55		$2\ T_{CLCL} - 70$	ns
ALE to Valid Data In	T_{LLDV}		350		$8\ T_{CLCL} - 150$	ns
Address to Valid Data In	T_{AVDV}		398		$9\ T_{CLCL} - 165$	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	T_{LLWL}	138	238	$3\ T_{CLCL} - 50$	$3\ T_{CLCL} + 50$	ns
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ High to ALE High	T_{WHLH}	23	103	$T_{CLCL} - 40$	$T_{CLCL} + 40$	ns
Address Valid to $\overline{\text{WR}}$	T_{AVWL}	120		$4\ T_{CLCL} - 130$		ns
Data Valid to $\overline{\text{WR}}$ Transition	T_{QVWX}	13		$T_{CLCL} - 50$		ns
Data Setup before $\overline{\text{WR}}$	T_{QVWH}	288		$7\ T_{CLCL} - 150$		ns
Data Hold after $\overline{\text{WR}}$	T_{WHQX}	13		$T_{CLCL} - 50$		ns
Address Float after $\overline{\text{RD}}$	T_{RLAZ}		0		0	ns

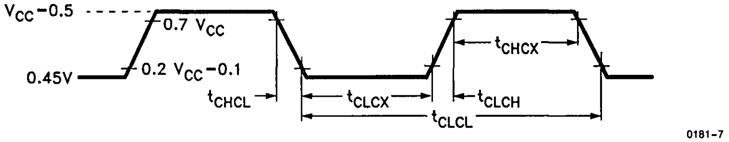
External Clock Drive for SAB 80C515-16/80C535-16

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq = 0.5 MHz to 16 MHz		
		Min	Max	
Oscillator Period	T_{CLCL}	62.5	2000	ns
Oscillator Frequency	$1/T_{CLCL}$	0.5	16	MHz
High Time	T_{CHCX}	15		ns
Low Time	T_{CLCX}	15		ns
Rise Time	T_{CLCH}		15	ns
Fall Time	T_{CHCL}		15	ns

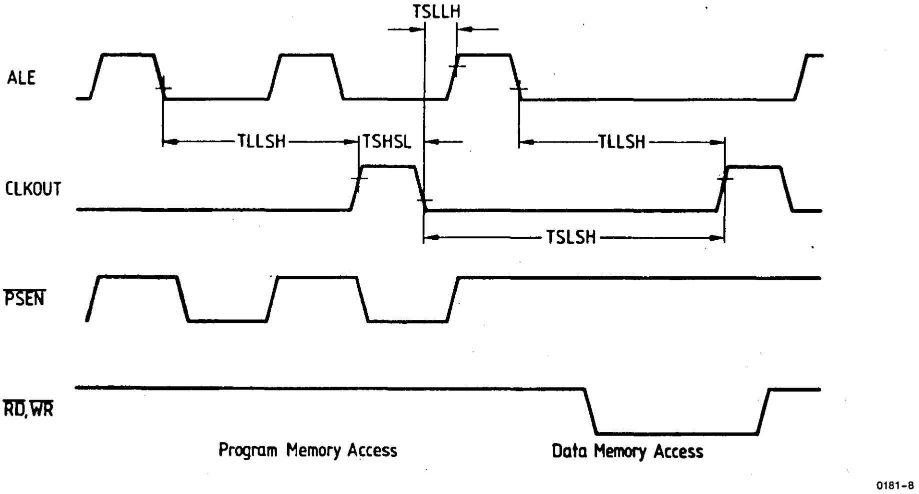
System Clock Timing for SAB 80C515-16/80C535-16

Parameter	Symbol	Limit Values				Unit
		16 MHz Clock		Variable Clock $1/T_{CLCL}$ = 0.5 MHz to 16 MHz		
		Min	Max	Min	Max	
ALE to CLKOUT	T_{LLSH}	398		$7\ T_{CLCL} - 40$		ns
CLKOUT High Time	T_{SHSL}	85		$2\ T_{CLCL} - 40$		ns
CLKOUT Low Time	T_{SLSH}	585		$10\ T_{CLCL} - 40$		ns
CLKOUT Low to ALE High	T_{SLLH}	23	103	$T_{CLCL} - 40$	$T_{CLCL} + 40$	ns

External Clock Cycle



System Clock Timing

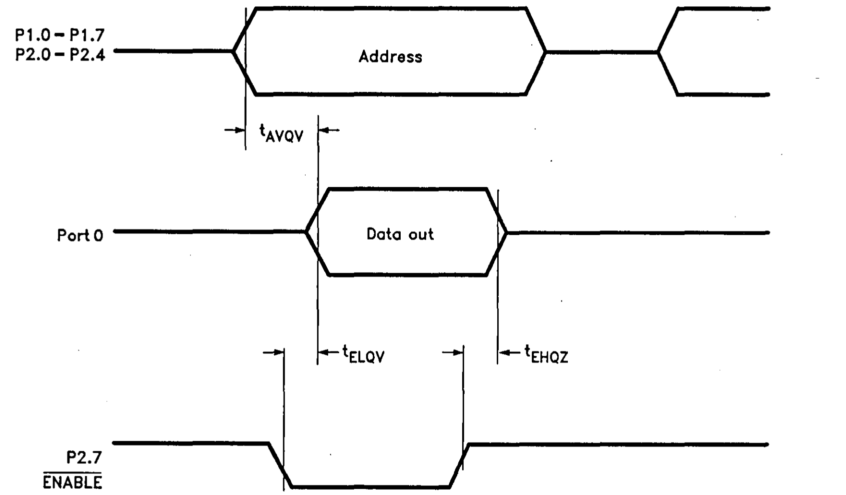


ROM Verification Characteristics

T_A = 25°C ±5°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Parameter	Symbol	Limit Values		Unit
		Min	Max	
Address to Valid Data	T_{AVQV}		$48\ T_{CLCL}$	ns
ENABLE to Valid Data	T_{ELQV}		$48\ T_{CLCL}$	ns
Data Float after ENABLE	T_{EHQZ}	0	$48\ T_{CLCL}$	ns
Oscillator Frequency	$1/T_{CLCL}$	4	6	MHz

ROM Verification

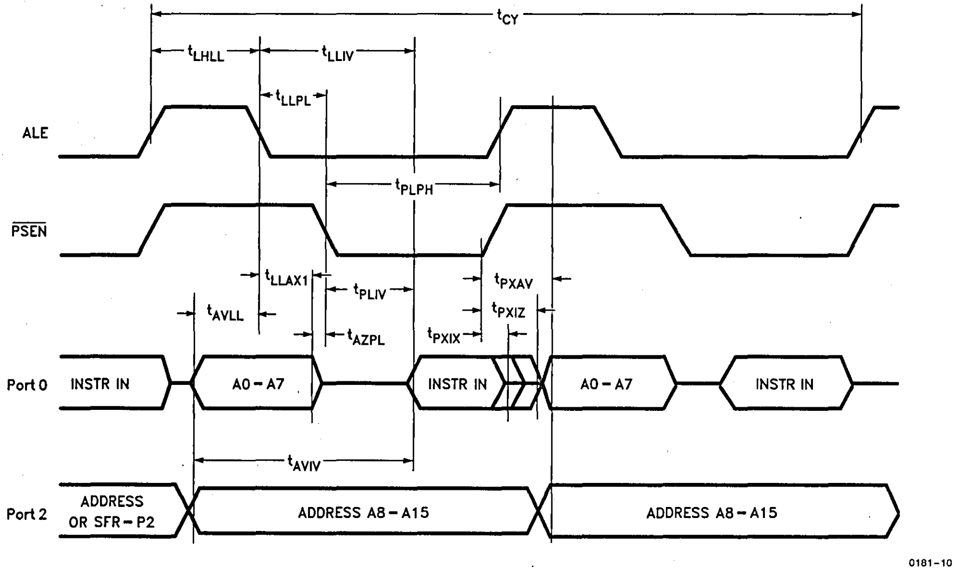


Address: P1.0–P1.7 = A0–A7
P2.0–P2.4 = A8–A12
Data: Port 0= D0–D7

Inputs: P2.5–P2.6, $\overline{PSEN}$ = V_{SS}
$\overline{ALE, EA}$ = V_{IH}
$\overline{RESET}$ = V_{IL}

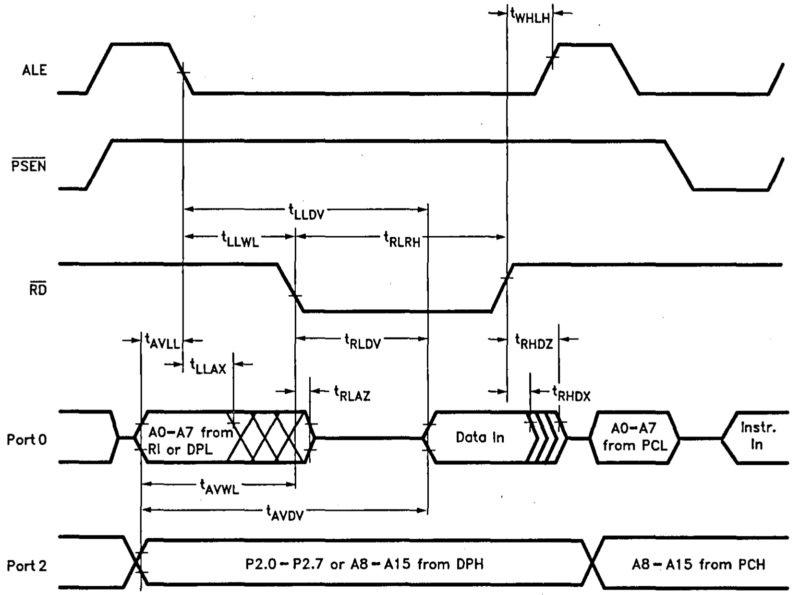
Waveforms

Program Memory Read Cycle



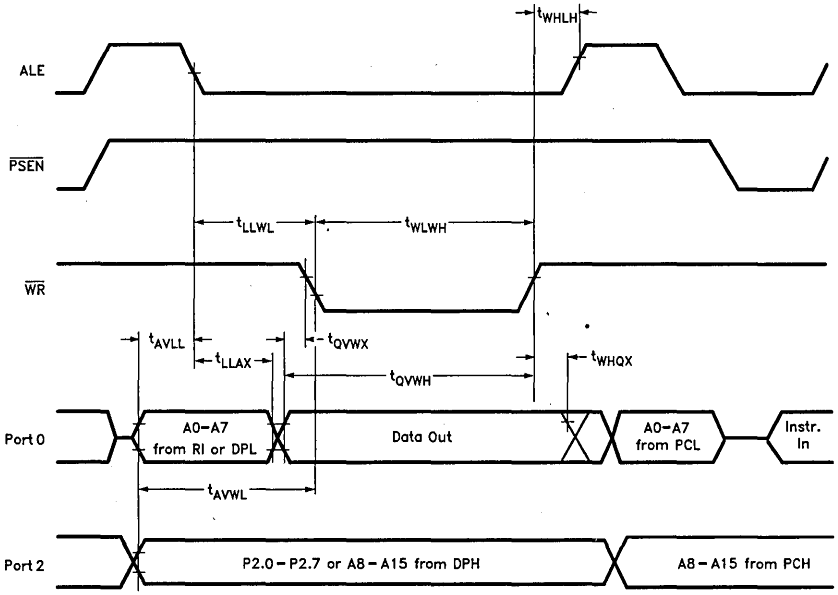
Waveforms (Continued)

Data Memory Read Cycle



0181-11

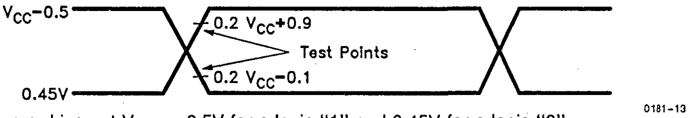
Data Memory Write Cycle



0181-12

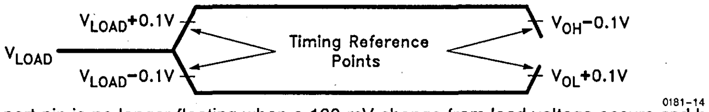
Waveforms (Continued)

AC Testing: Input, Output Waveforms



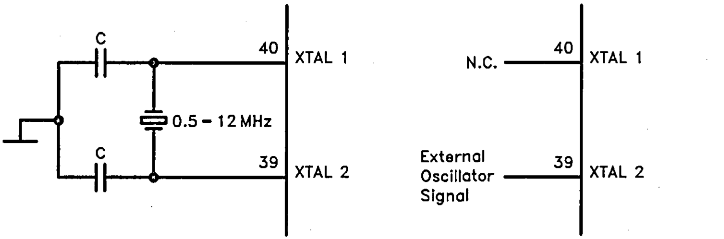
AC inputs during testing are driven at V_{CC} − 0.5V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at V_{IHmin} for a logic "1" and V_{ILmax} for a logic "0".

AC Testing: Float Waveforms



For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV deviation from the load voltage V_{OH}/V_{OL} occurs. I_{OL}/I_{OH} ≥ ±20 mA.

Recommended Oscillator Circuits



C = 30 pF ±10 pF
(incl. stray capacitance)
Crystal Oscillator Mode

Driving from External Source

Ordering Information

Type	Description
	8-bit CMOS Microcontroller
SAB 80C515-N	with Mask-Programmable ROM (Plastic), 12 MHz
SAB 80C535-N	for External Memory (Plastic), 12 MHz
SAB 80C515-16-N	with Mask-Programmable ROM (Plastic), 16 MHz
SAB 80C535-16-N	for External Memory (Plastic), 16 MHz
SAB 80C515-N-T40/85	with Mask-Programmable ROM (Plastic), Ext. Temperature
SAB 80C535-N-T40/85	for External Memory (Plastic), Ext. Temperature
SAB 80C515-16-N-T40/85	with Mask-Programmable ROM, Ext. Temp., 16 MHz
SAB 80C535-16-N-T40/85	for External Memory (Plastic), Ext. Temp., 16 MHz

SIEMENS

Advance Information

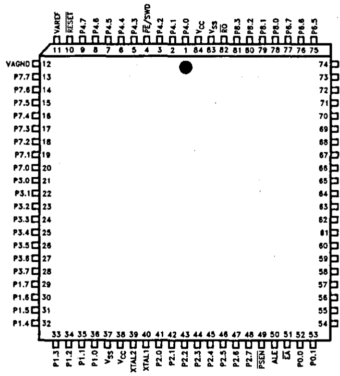
SAB 80C517/80C537 High Performance 8-Bit Single-Chip CMOS Microcontroller

SAB 80C517	CMOS microcontroller with factory-mask programmable ROM
SAB 80C537	CMOS microcontroller for external ROM
SAB 80C517-T40/85	Extended temperature range: −40°C to +85°C
SAB 80C537-T40/85	Extended temperature range: −40°C to +85°C

- 8K × 8 ROM (SAB 80C517 only)
- 256 × 8 On-Chip RAM
- Superset of SAB 80C51 architecture:
  — 1 μs Instruction Cycle Time at 12 MHz
  — 256 Direct Addressable Bits
  — Boolean Processor
  — External Data and Program Memory Expandable up to 64 Kbytes each
- Fully Backward Compatible with SAB 80C515
- Four 16-Bit Timer/Counters
- Powerful 16-Bit Compare/Capture-Unit (CCU) with up to 21 High-Speed or PWM Output Channels and 5 Capture Inputs
- Two Full Duplex Serial Interfaces
- Fast 32-Bit Division, 16-Bit Multiplication, 32-Bit Normalize and Shift operation using the on-chip MUL/DIV-Unit (MDU)
- Eight Datapointers for External Memory Addressing
- Fourteen Interrupt Vectors, Four Priority Levels Selectable
- 8-Bit A-to-D Converter with 12 Multiplexed Inputs and Programmable Reference Voltages
- Versatile "Fail Safe"-Provisions (Watchdog . . .)
- Extended Power Saving Modes
- Nine Ports: 56 I/O-Lines, 12 Input-Lines

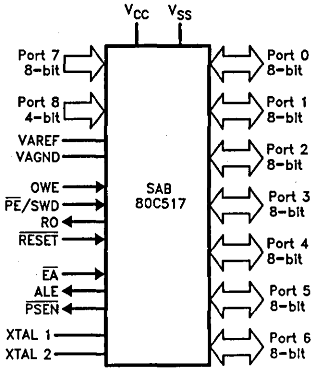
Pin Configuration

PLCC 84



0193-1

Logic Symbol



0193-2

The SAB 80C517/80C537 is a high-end member of the Siemens SAB 8051 family of microcontrollers. It is designed in Siemens ACMOS IV technology and based on SAB 8051 architecture. ACMOS IV is a technology which combines high speed, density characteristics with low power consumption.

While maintaining all the SAB 80C515 features and operating characteristics the SAB 80C517 is expanded in its arithmetic capabilities, "fail safe" characteristics, analog signal processing and timer capabilities. The SAB 80C537 is identical with the SAB 80C517 except that it lacks the on-chip program memory. The SAB 80C517/80C537 is supplied in an 84-pin plastic leaded chip carrier package (PLCC 84).

Pin Definitions and Functions

Pin	Symbol	Input (I) Output (O)	Function
1-3, 5-9	P4.0-P4.7	I/O	**Port 4** is a bidirectional I/O port with internal pullup resistors. Port 4 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 4 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors.
4	$\overline{PE}$/SWD	I	A low level on this pin allows the software to enter the power down, idle and slow down mode. In case the low level is also seen during reset, the watchdog timer function is off on default. Use of the power saving modes is blocked, when this pin is held on high level. A high level during reset performs an automatic start of the watchdog timer immediately after reset. When left unconnected this pin is pulled high by a small internal pullup.
10	$\overline{RESET}$	I	A low level on this pin for the duration of two machine cycles while the oscillator is running resets the SAB 80C517. A small internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
11	V_{AREF}		**Reference voltage** for the A/D converter
12	V_{AGND}		**Reference ground** for the A/D converter
13-20	P7.7-P7.0	I	**Port 7** is an 8-bit undirectional input port. Port pins can be used for digital input, if voltage levels meet the specified input high/low voltages, and for the lower 8-bit of the multiplexed analog inputs of the A/D converter, simultaneously.

Pin Definitions and Functions (Continued)

Pin	Symbol	Input (I) Output (O)	Function
21–28	P3.0–P3.7	I/O	**Port 3** is a bidirectional I/O port with internal pullup resistors. Port 3 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 3 also contains the interrupt, timer, serial port 0 and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to pins of port 3 as follows: —RxD(P3.0): receiver data input (asynchronous) or data input/output (synchronous) of serial interface 0 —TxD(P3.1): transmitter data output (asynchronous) or clock output (synchronous) of serial interface 0 —$\overline{\text{INT0}}$(P3.2): interrupt 0 input/Timer 0 gate control —$\overline{\text{INT1}}$(P3.3): interrupt 1 input/Timer 1 gate control —T0(P3.4): counter 0 input —T1(P3.5): counter 1 input —$\overline{\text{WR}}$(P3.6): the write control signal latches the data byte from port 0 into the external data memory. —$\overline{\text{RD}}$(P3.7): the read control signal enables the external data memory to port 0
29–36	P1.7–P1.0	I/O	**Port 1** is a bidirectional I/O port with internal pullup resistors. Port 1 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. It is used for the low order address byte during program verification. It also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output latch must be programmed to a one (1) for that function to operate (except when used for the compare functions). The secondary functions are assigned to the port 1 pins as follows: —$\overline{\text{INT3}}$/CC0(P1.0): interrupt 3 input/compare 0 output/compare 0 input —INT4/CC1(P1.1): interrupt 4 input/compare 1 output/capture 1 input —INT5/CC2(P1.2): interrupt 5 input/compare 2 output/capture 2 input —INT6/CC3(P1.3): interrupt 6 input/compare 3 output/capture 3 input —$\overline{\text{INT2}}$/CC4(P1.4): interrupt 2 input/compare 4 output/capture 4 input —T2EX(P1.5): Timer 2 external reload trigger input —CLKOUT(P1.6): system clock output —T2(P1.7): counter 2 input

Pin Definitions and Functions (Continued)

Pin	Symbol	Input (I) Output (O)	Function
37	V_{SS}		Circuit ground potential
38	V_{CC}		Supply terminal for all operating modes
39 40	XTAL2 XTAL1		XTAL2 Input to the inverting oscillator amplifier and input to the internal clock generator circuits. XTAL1 Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL2 should be driven, while XTAL1 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times as well as rise/fall times specified in the AC-characteristics must be observed.
41–48	P2.0–P2.7	I/O	**Port 2** is a bidirectional I/O port with internal pullup resistors. Port 2 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullup resistors when issuing 1s. During accesses to external data memory that use 8-bit addresses (MOVX @RI), port 2 issues the contents of the P2 special function register.
49	$\overline{\text{PSEN}}$	O	The **Program Store Enable** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. Remains high during internal program execution.
50	ALE	O	The **Address Latch Enable** output is used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
51	$\overline{\text{EA}}$	I	**External Access Enable.** When held at high level, the SAB 80C517 executes instructions from the internal ROM when the PC has address less than 8192. When held at low level, the SAB 80C517 fetches all instructions from external program memory. For the SAB 80C537 this pin must be tied low.

Pin Definitions and Functions (Continued)

Pin	Symbol	Input (I) Output (O)	Function
52–59	P0.0–P0.7	I/O	Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pullup resistors when issuing 1s. Port 0 also outputs the code bytes during program verification in the SAB 80C517. External pullup resistors are required during program verification.
60	V_{SS}		Circuit ground potential
61–68	P5.7–P5.0	I/O	**Port 5** is a bidirectional I/O port with internal pullup resistors. Port 5 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 5 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. This port also serves the alternate function "Concurrent Compare". The secondary functions are assigned to the port 5 pins as follows: —CCM0(P5.0): concurrent compare 0 —CCM1(P5.1): concurrent compare 1 —CCM2(P5.2): concurrent compare 2 —CCM3(P5.3): concurrent compare 3 —CCM4(P5.4): concurrent compare 4 —CCM5(P5.5): concurrent compare 5 —CCM6(P5.6): concurrent compare 6 —CCM7(P5.7): concurrent compare 7
69	OWE	I	A high level on this pin enables the oscillator watchdog. When left unconnected this pin is pulled high by a small internal pullup. When held at low level the oscillator watchdog function is off.
70–77	P6.0–P6.7	I/O	**Port 6** is a bidirectional I/O port with internal pullup resistors. Port 6 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 6 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 6 also contains the external A/D converter control pin and the transmit and receive pins for serial channel 1. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 6, as follows: —$\overline{\text{ADST}}$(P6.0): external A/D-converter start pin —RxD1(P6.1): receiver data input of serial interface1 —TxD(P6.2): transmitter data output of serial interface1

Pin Definitions and Functions (Continued)

Pin	Symbol	Input (I) Output (O)	Function
78–81	P8.0–P8.3	I	**Port 8** is a 4-bit unidirectional input port. Port pins can be used for digital input, if voltage levels meet the specified input high/low voltages, and for the higher 4-bit of the multiplexed analog inputs of the A/D converter, simultaneously.
82	$\overline{\text{RO}}$	O	**Reset Output** pin. This pin outputs the internally synchronized reset request signal. This signal may be generated by an external hardware reset, a watchdog timer reset or an oscillator watchdog reset. The reset output is active low.
83	V_{SS}		Circuit ground potential
84	V_{CC}		Supply terminal for all operating modes

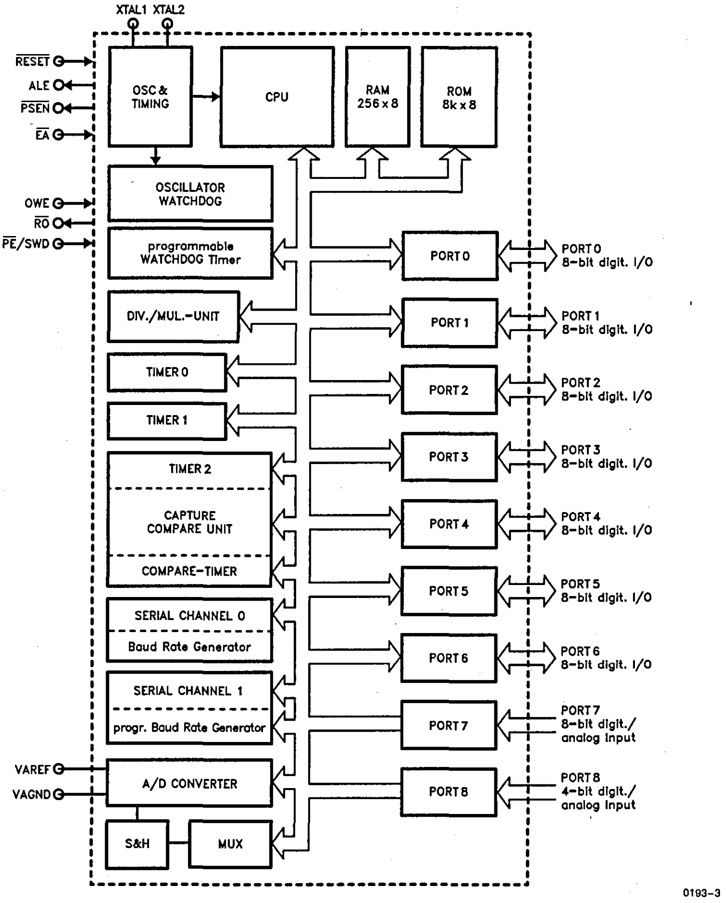


Figure 1. Block Diagram SAB 80C517

Functional Description

The following description gives a brief summary of the SAB 80C517's architecture and its peripheral functions. It is mainly based on a list of the SAB 80C517's special function registers given in Table 1. In the following text, any reference to the SAB 80C517 applies to all versions of this microcontroller unless otherwise noted.

The functional description contains the following sections:

1.0 Architecture

The SAB 80C517 is based on 8051 architecture. It is a fully compatible member of the SIEMENS SAB 8051/80C51 microcontroller family. It is significantly enhanced as compared to the SAB 80C515. The SAB 80C517 is therefore 100% compatible with code written for the SAB 80C51 and SAB 80C515.

2.0 CPU

Having an 8-bit CPU with extensive facilities for bit-handling and binary BCD arithmetics, the SAB 80C517 is optimized for control applications. With a 12 MHz crystal, 58% of the instructions execute in 1 μs.

Being designed to close the performance gap to the 16-bit microcontroller world, the SAB 80C517's CPU is supported by a powerful 32-bit/ 16-bit arithmetic unit and a more flexible addressing of external memory by eight 16-bit Datapointers.

3.0 Memory Organization

Identical to the SAB 8051 architecture, the SAB 80C517 has separate address spaces for program and data memory. Figure 2 illustrates the mapping of address spaces.

Program Memory

The SAB 80C517 has 8 Kbyte of on-chip ROM, while the SAB 80C537 has no internal ROM. The external program memory can be expanded up to 64 Kbytes. Pin $\overline{EA}$ controls whether program fetches below address 2000H are done from internal or external memory.

Data Memory

The data memory space consists of an **internal** and **external** memory space.

External Data Memory

Up to 64 Kbytes external data memory can be addressed by instructions that use 8-bit or 16-bit indirect addressing. A 16-bit external data memory address requires one of the eight 16-bit datapointers for addressing. For an 8-bit address, registers R0 and R1 can also be used for addressing the external data memory.

Multiple Datapointers

As a functional enhancement to standard 8051 controllers, the SAB 80C517 contains eight 16-bit datapointers. The instruction set uses only one of these datapointer registers at a time. The selection of the datapointer register is done using another special function register DPSEL (Data Pointer Select, Addr. 92H). Figure 3 illustrates the addressing mechanism.

Internal Data Memory

The internal data memory is divided into three physically distinct blocks:

- — the lower 128 bytes of RAM including four banks of eight registers each
- — the upper 128 bytes of RAM
- — the 128-byte special function register area

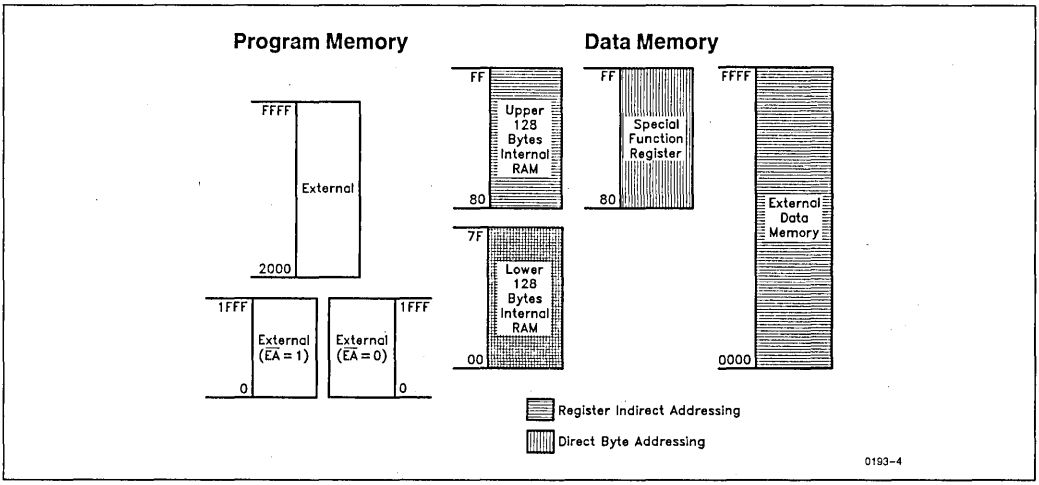
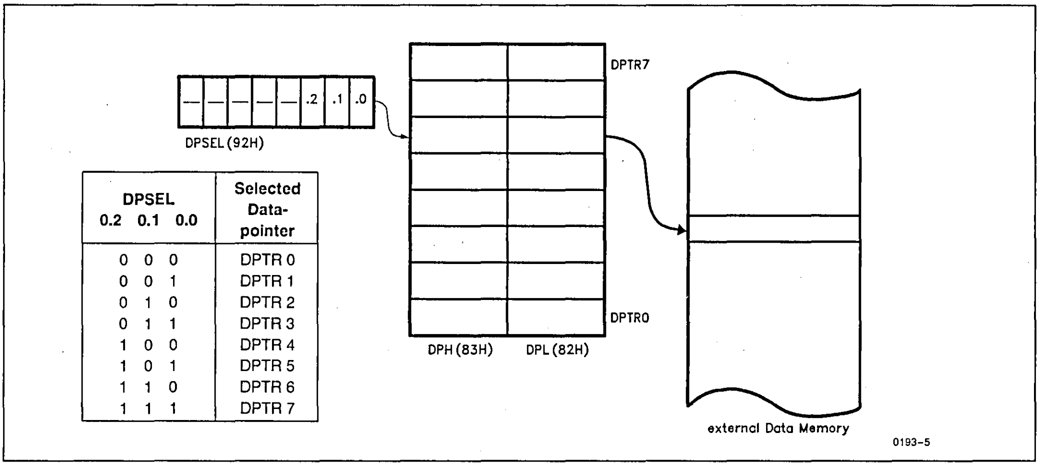


Figure 2. Memory Mapping of the SAB 80C517



DPSEL 0.2 0.1 0.0	Selected Data-pointer
0 0 0	DPTR 0
0 0 1	DPTR 1
0 1 0	DPTR 2
0 1 1	DPTR 3
1 0 0	DPTR 4
1 0 1	DPTR 5
1 1 0	DPTR 6
1 1 1	DPTR 7

Figure 3. Addressing of External Data Memory

A mapping of the internal data memory is also shown in Figure 2. The overlapping address spaces are accessed by different addressing modes. The stack can be located anywhere in the internal data memory.

4.0 Special Function Registers

All registers, except the program counter and the four general purpose register banks, reside in the special function register area. The 81 special function registors include arithmetic registers, pointers, and registers that provide an interface between the CPU and the on-chip peripherals. There are also 128 directly addressable bits within the SFR area. The special function registers are listed in Table 1. In this table they are organized in groups which refer to the functional blocks of the SAB 80C517. Block names and symbols are listed in alphabetical order. Bit-addressable Special Function Registers are marked with a dot in the fifth column. Special Function Registers with bits belonging to more than one functional block of the SAB 80C517 are marked with an asterisk (*) in column two.

Table 1. Special Function Registers of the SAB 80C517

Block	Symbol	Name	Address	
CPU	ACC	Accumulator	0E0H	●
	B	B-Register	0F0H	●
	DPH	Data Pointer, High Byte	83H	
	DPL	Data Pointer, Low Byte	82H	
	DPSEL	Data Pointer Select Register	92H	
	PSW	Program Status Word Register	0D0H	●
	SP	Stack Pointer	81H	
A/D-Converter	ADCON0	A/D Converter Control Reg. 0	0D8H	●
	ADCON1	A/D Converter Control Reg. 1	0DCH	
	ADDAT	A/D Converter Data Register	0D9H	
	DAPR	D/A Converter Program Reg.	0DAH	
Compare/Capture Unit/ (CCU)	CCEN	Comp./Capture Enable Reg.	0C1H	
	CC4EN	Comp./Capture 4 Enable Reg.	0C9H	
	CCH1	Comp./Capt. Reg. 1, High Byte	0C3H	
	CCH2	Comp./Capt. Reg. 2, High Byte	0C5H	
	CCH3	Comp./Capt. Reg. 3, High Byte	0C7H	
	CCH4	Comp./Capt. Reg. 4, High Byte	0CFH	
	CCL1	Comp./Capt. Reg. 1, Low Byte	0C2H	
	CCL2	Comp./Capt. Reg. 2, Low Byte	0C4H	
	CCL3	Comp./Capt. Reg. 3, Low Byte	0C6H	
	CCL4	Comp./Capt. Reg. 4, Low Byte	0CEH	
	CMEN	Compare Enable Register	0F6H	
	CMH0	Compare Reg. 0, High Byte	0D3H	
	CMH1	Compare Reg. 1, High Byte	0D5H	
	CMH2	Compare Reg. 2, High Byte	0D7H	
	CMH3	Compare Reg. 3, High Byte	0E3H	
	CMH4	Compare Reg. 4, High Byte	0E5H	
	CMH5	Compare Reg. 5, High Byte	0E7H	
	CMH6	Compare Reg. 6, High Byte	0F3H	
	CMH7	Compare Reg. 7, High Byte	0F5H	
	CML0	Compare Register 0, Low Byte	0D2H	
	CML1	Compare Register 1, Low Byte	0D4H	
	CML2	Compare Register 2, Low Byte	0D6H	
	CML3	Compare Register 3, Low Byte	0E2H	
	CML4	Compare Register 4, Low Byte	0E4H	
	CML5	Compare Register 5, Low Byte	0E6H	
	CML6	Compare Register 6, Low Byte	0F2H	
	CML7	Compare Register 7, Low Byte	0F4H	
	CMSEL	Compare Input Select	0F7H	
	CRCH	Com./Rel./Capt. Reg. High Byte	0CBH	
	CRCL	Com./Rel./Capt. Reg. Low Byte	0CAH	
	CTCON	Com. Timer Control Reg.	0E1H	
	CTRELH	Com. Timer Rel. Reg., High Byte	0DFH	
	CTRELL	Com. Timer Rel. Reg., Low Byte	0DEH	
	TH2	Timer 2, High Byte	0CDH	
	TL2	Timer 2, Low Byte	0CCH	
	T2CON	Timer 2 Control Register	0C8H	●

Table 1. Special Function Registers of the SAB 80C517 (Continued)

Block	Symbol	Name	Address	
Interrupt System	IEN0	Interrupt Enable Register 0	0A8H	●
	CTCON*	Comp. Timer Control Reg.	0E1H	
	IEN1	Interrupt Enable Register 1	0B8H	●
	IEN2	Interrupt Enable Register 2	9AH	
	IP0	Interrupt Priority Register 0	0A9H	
	IP1	Interrupt Priority Register 1	0B9H	
	IRCON	Interr. Request Control Reg.	0C0H	●
	TCON*	Timer Control Register	88H	●
	T2CON*	Timer 2 Control Register	0C8H	●
MUL/DIV Unit	ARCON	Arithmetic Control Register	0EFH	
	MD0	Multiplication/Division Reg. 0	0E9H	
	MD1	Multiplication/Division Reg. 1	0EAH	
	MD2	Multiplication/Division Reg. 2	0EBH	
	MD3	Multiplication/Division Reg. 3	0ECH	
	MD4	Multiplication/Division Reg. 4	0EDH	
	MD5	Multiplication/Division Reg. 5	0EEH	
Ports	P0	Port 0	80H	●
	P1	Port 1	90H	●
	P2	Port 2	0A0H	●
	P3	Port 3	0B0H	●
	P4	Port 4	0E8H	●
	P5	Port 5	0F8H	●
	P6	Port 6	0FAH	
	P7	Port 7, Analog/Digital Input	0DBH	
	P8	Port 8, Analog/Dig. Input, 4-Bit	0DDH	
Pow.Sav.M.	PCON	Power Control Register	87H	
Serial Channels	ADCON0*	A/D Converter Control Reg.	0D8H	●
	PCON*	Power Control Register	87H	
	S0BUF	Serial Channel 0 Buffer Reg.	99H	
	S0CON	Serial Channel 0 Control Reg.	98H	●
	S1BUF	Serial Channel 1 Buffer Reg.	9CH	
	S1CON	Serial Channel 1 Control Reg.	9BH	
	S1REL	Serial Channel 1 Reload Reg.	9DH	
Timer 0 Timer 1	TCON	Timer Control Register	88H	●
	TH0	Timer 0, High Byte	8CH	
	TH1	Timer 1, High Byte	8DH	
	TL0	Timer 0, Low Byte	8AH	
	TL1	Timer 1, Low Byte	8BH	
	TMOD	Timer Mode Register	89H	
Watchdog	IEN0*	Interrupt Enable Register 0	0A8H	●
	IEN1*	Interrupt Enable Register 1	0B8H	●
	IP0*	Interrupt Priority Register 0	0A9H	
	IP1*	Interrupt Priority Register 1	0B9H	
	WDTREL	Watchdog Timer Reload Reg.	86H	

Table 2. Register Contents after Reset

Register	Contents
PC	**00H**
ACC	00H
ADCON0	00H
ADCON1	XXXX 0000B
ADDAT	00H
ARCON	0XXXX XXXXB
B	00H
CCL1-4	00H
CCH1-4	00H
CCEN	00H
CC4EN	00H
CMEN	00H
CMH0-7	00H
CML0-7	00H
CMSEL	00H
CRCL, CRCH	00H
CTCON	XXXX 0000B
CTRELL, CTRELH	00H
DAPR	00H
DPSEL	XXXXX000B
DPTR0-7	000H
IEN0, INE1	00H
IEN2	XXXX XX00B
IP0, IP1	00H
IRCON	00H
MD 0-5	XXH
P0-P6	0FFH
PCON	00H
PSW	00H
S0BUF, S1BUF	0XXH
S0CON	00H
S1CON	0X00 0000B
S1REL	00H
SP	07H
TCON	00H
TL0, TH0	00H
TL2, TH2	00H
TMOD	00H
T2CON	00H
WDTREL	00H

X means, that the value is undeterminate.

5.0 On-Chip Peripherals

Given below is a functional description of all Special Function Registers and Register Bits which are used to program the SAB 80C517's peripherals.

5.1 A/D Converter

The SAB 80C517 contains an 8-bit A/D-Converter with 12 multiplexed input channels. Reference voltages are internally programmable. A block diagram of the A/D converter is shown in Figure 4.

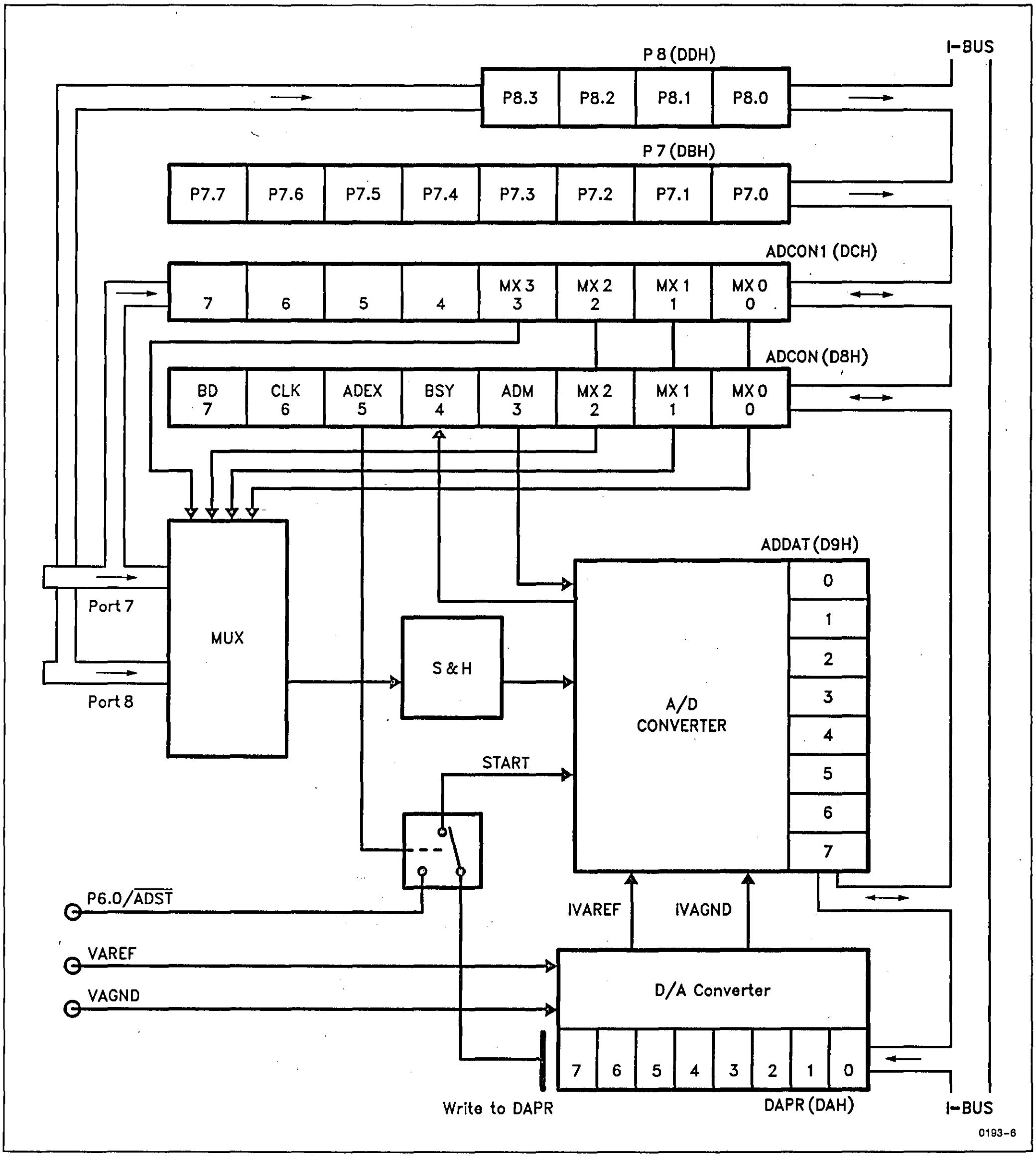


Figure 4. Block Diagram A/D Converter

Special Function Registers of the A/D Converter

DAPR

0DAH	Programming of $V_{intAREF}$	Programming of $V_{intAGND}$	DAPR

D/A Converter Program Register. Each 4 bit nibble is used to program the internal reference voltages. Write-access to DAPR starts conversion.

$$V_{intAGND} = V_{intAGND} + \frac{DAPR\,(0.3 - 0.0)}{16}(V_{AREF} - V_{AGND})$$

with DAPR (0.3 − 0.0) < 13;

$$V_{intAREF} = V_{intAGND} + \frac{DAPR\,(0.7 - 0.4)}{16}(V_{AREF} - V_{AGND})$$

with DAPR (0.7 − 0.4) > 3

If DAPR (0.3 − 0.0) or DAPR (0.7 − 0.4) = 0 then the internal references voltages correspond to the external reference voltages V_{AGND} and V_{AREF}.

ADDAT

0D9H	Conversion Result	ADDAT

This register contains the 8-bit conversion result.

ADCON 0

	0DFH	0DEH	0DDH	0DCH	0DBH	0DAH	0D9H	0D8H	
0D8H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0	ADCON 0

▨ These bits are not used in controlling A/D Converter Functions

Bit	Function
MX0 MX1 MX2	Select lower 8 input channels of the A/D converter. See Table 3
ADM	A/D conversion mode. When set, a continuous conversion is selected. If ADM = 0, the converter stops after one conversion.
BSY	Busy flag. This flag indicates whether a conversion is in progress (BSY = 1). The flag is cleared by hardware when the conversion is completed.
ADEX	Internal/external start of conversion. When set, the external start of conversion by P6.0/$\overline{\text{ADST}}$ is enabled.

ADCON1

0DCH	—	—	—	—	MX3	MX2	MX1	MX0	ADCON 1

A/D-converter control register 1. Contains channel selection bits MX0 to MX3. For channel selection see Table 3. Bits MX0 to MX2 can be written or read either in ADCON0 or in ADCON1.

Table 3. Selection of the Analog Input Channels

MX3	MX2	MX1	MX0	Selected Channel	Pin
0	0	0	0	Analog Input 0	P7.0
0	0	0	1	Analog Input 1	P7.1
0	0	1	0	Analog Input 2	P7.2
0	0	1	1	Analog Input 3	P7.3
0	1	0	0	Analog Input 4	P7.4
0	1	0	1	Analog Input 5	P7.5
0	1	1	0	Analog Input 6	P7.6
0	1	1	1	Analog Input 7	P7.7
1	X*	0	0	Analog Input 8	P8.0
1	X	0	1	Analog Input 9	P8.1
1	X	1	0	Analog Input 10	P8.2
1	X	1	1	Analog Input 11	P8.3

*X means that the value may be 0 or 1.

5.2 Compare/Capture-Unit (CCU)

The Compare Capture Unit is a complex timer/register array for applications that require High Speed I/O, Pulse Width Modulation and more timer/counter capabilities. The CCU consists of

- — one 16-bit timer/counter (Timer 2) with 2-bit prescaler, reload capability and a max. clock frequency of $f_{osc/12}$.
- — one 16-bit timer (Compare Timer) with 8-bit prescaler, reload capability and a max. clock frequency of $f_{osc/2}$.
- — thirteen 16-bit Compare Registers
- — five of which can be used as 16-bit Capture Registers
- — up to 21 output lines controlled by the CCU
- — seven interrupts which can be generated by CCU-events..

Figure 5 shows a block diagram of the CCU. Eight compare registers (CM0 to CM7) can individually be assigned either to Timer 2 or to the Compare Timer. Diagrams of the two timers are shown in Figures 6 and 7. The four Compare/Capture Registers and the Compare/Reload/Capture Register are always connected to Timer 2. Depending upon the selection of the register type and the timer, two compare modes can be selected. Table 4 illustrates possible combinations and the corresponding output lines.

Compare

Mode 0: Upon a match the output signal changes from low to high. It goes back to low level whon Timer 2 overflows.

Mode 1: The transition of the output signal can be determined by software. A Timer 2 overflow does not necessarily cause an output change.

Compare Registers CM0 to CM7 use additional shadow latches when operated in mode 0. Figure 8 shows the function of these shadow latches. The shadow latches are implemented to prevent from loss of compare matches which may occur when loading of the compare values is not correlated with the timer count. The shadow latches are automatically loaded from the Compare Registers, every time the timer overflows.

Capture

Mode 0: Capture is performed in response to a transition at the corresponding port 1 pins CC0 to CC3.

Mode 1: Write operation into the low-order byte of the dedicated capture register causes the Timer 2 contents to be latched into this register.

Reload of Timer 2

Mode 0: Reload is caused by a timer overflow (auto-reload).

Mode 1: Reload is caused in response to a negative transition at pin T2EX (P1.5), which also can request an interrupt.

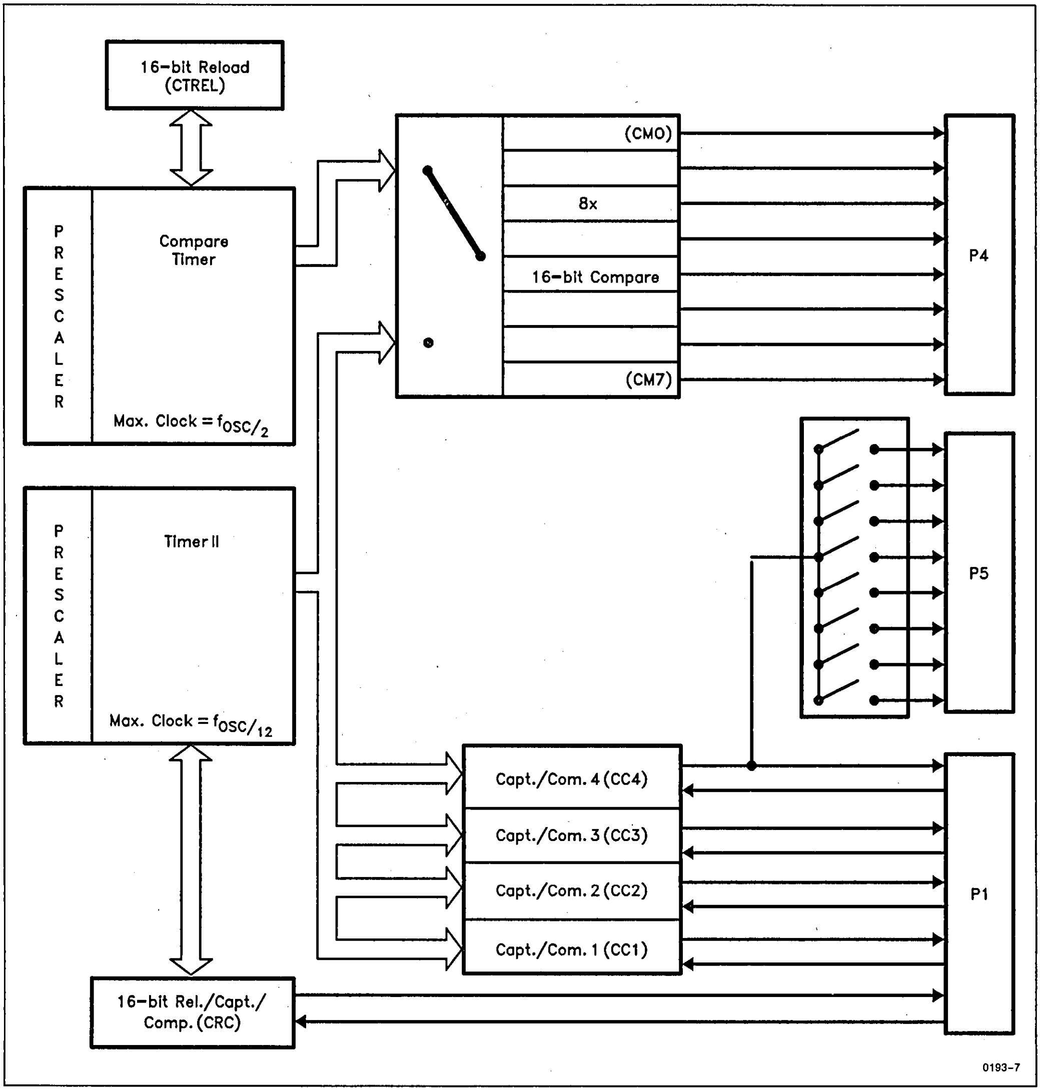


Figure 5. Block Diagram of the Compare/Capture-Unit

Table 4. CCU Compare Configuration

Assigned Timer	Compare Reg.	Compare Output at	Possible Modes
Timer 2	CRCH/CRCL	P1.0/$\overline{\text{INT3}}$/CC0	Comp. Mode 0, 1 + Reload
	CC1H/CC1L	P1.1/INT4/CC1	Comp. Mode 0, 1
	CC2H/CC2L	P1.2/INT5/CC2	Comp. Mode 0, 1
	CC3H/CC3L	P1.3/INT6/CC3	Comp. Mode 0, 1
	CC4H/CC4L	P1.4/$\overline{\text{INT2}}$/CC4	Comp. Mode 0, 1
	CC4H/CC4L	P5.0/CCM0	Comp. Mode 1
	:	:	:
	CC4H/CC4L	P5.7/CCM7	Comp. Mode 1
	CM0H/CM0L	P4.0/CM0	Comp. Mode 1
	:	:	:
	CM7H/CM7L	P4.7/CM7	Comp. Mode 1
Compare Timer	CM0H/CM0L	P4.0/CM0	Comp. Mode 0 (with Shadow Latches)
	:	:	:
	:	:	:
	:	:	:
	CM7H/CM7L	P4.7/CM7	Comp. Mode 0 (with Shadow Latches)

Special function registers of the CCU:

T2H/T2L

0CCH	Timer 2, Low	TL2
0CDH	Timer 2, High	TH2

16-bit Timer-Register. Contains actual count of Timer 2.

CTRELH/CTREL

0DEH	Compare-Timer Reload, Low	CTRELL
0DFH	Compare/Timer Reload, High	CTRELH

16-bit Compare Timer Reload Register. After Overflow the Compare Timer is automatically reloaded with the contents of this register.

A write-to-CTRELL starts or restarts a running Compare Timer with the contents of CTREL. If loading a CTREL with a 16-bit value is intended, the high byte CTRELH must be written first. The prescaler of the Compare-Timer (register CTCON) should be set up before the timer is started.

CCH1/CCL1 to CCH3/CCL3

0C2H	Compare/Capture 1, Low	CCL1
0C3H	Compare/Capture 1, High	CCH1
0C4H	Compare/Capture 2, Low	CCL2
0C5H	Compare/Capture 2, High	CCH2
0C6H	Compare/Capture 3, Low	CCL3
0C7H	Compare/Capture 3, High	CCH3

16-bit dual function Compare/Capture-Registers. The compare function of each register controls one port output line (see Table 4) on port 1. These registers exclusively operate with Timer 2 as a time-base. On a compare match an interrupt is requested.

CCH4/CCL4

0CEH	Compare/Capture 4, Low	CCL4
0CFH	Compare/Capture 4, High	CCH4

16-bit dual function Compare/Capture register. In addition to the standard Compare/Capture functions this register provides the "Concurrent Compare" feature which allows a simultaneous control of up to 9 output lines (at Port 1 and Port 5) with one compare register. It operates exclusively with Timer 2. On a compare match an interrupt can be generated.

CRCL/CRCH

0CAH	Compare/Reload/Capture, Low	CRCL
0CBH	Compare/Reload/Capture, High	CRCH

16-bit three function Compare/Reload/Capture Register. Provides Compare and Capture function using Timer 2. Can also be used as a 16-bit reload-register for Timer 2.

CMH0/CML0 to CMH7/CML7

0D2H	Compare 0, Low	CML0
0D3H	Compare 0, High	CMH0
0D4H	Compare 1, Low	CML1
0D5H	Compare 1, High	CMH1
0D6H	Compare 2, Low	CML2
0D7H	Compare 2, High	CMH2
0E2H	Compare 3, Low	CML3
0E3H	Compare 3, High	CMH3
0E4H	Compare 4, Low	CML4
0E5H	Compare 4, High	CMH4
0E6H	Compare 5, Low	CML5
0E7H	Compare 5, High	CMH5
0F2H	Compare 6, Low	CML6
0F3H	Compare 6, High	CMH6
0F4H	Compare 7, Low	CML7
0F5H	Compare 7, High	CMH7

16-bit Compare registers. The compare function of each register controls one output line at port 4 (see Table 4). These registers can individually be assigned either to Timer 2 or to the Compare Timer. When operated with the Compare timer, shadow latches are used to latch the contents at timer overflow.

CCEN

0C1H	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0	CCEN

Compare/Capture Enable Register selects Compare or Capture function for registers CRC, CC1 to CC3.

Bit		Function
COCAH0	**COCAL0**	**Compare/Capture Mode for CRC Register**
0	0	Compare/Capture Disabled
0	1	Capture on Falling/Rising Edge at Pin P1.0/$\overline{\text{INT3}}$/CC0
1	0	Compare Enabled
1	1	Capture on Write Operation into Register CRCL
COCA1H	**COCAL1**	**Compare/Capture Mode for CC Register 1**
0	0	Compare/Capture Disabled
0	1	Capture on Falling/Rising Edge at Pin P1.1/INT4/CC1
1	0	Compare Enabled
1	1	Capture on Write Operation into Register CCL1
COCAH2	**COCAL2**	**Compare/Capture Mode for CC Register 2**
0	0	Compare/Capture Disabled
0	1	Capture on Falling/Rising Edge at Pin P1.2/INT5/CC2
1	0	Compare Enabled
1	1	Capture on Write Operation into Register CCL2
COCAH3	**COCAL3**	**Compare/Capture Mode for CC Register 3**
0	0	Compare/Capture Disabled
0	1	Capture on Falling/Rising Edge at Pin P1.3/INT6/CC3
1	0	Compare Enabled
1	1	Capture on Write Operation into Register CCL3

CC4EN

0C9H	—	COCON2	COCON1	COCON0	COCOEN	COCAH4	COCAL4	COMO	CC4EN

Selects Compare or Capture function, number of concurrent compares and compare mode of register CC4.

Bit		Function
COACH4	**COCAL4**	**Compare/Capture Mode for CC4 Register**
0	0	Compare/Capture Disabled
0	1	Capture on Falling/Rising Edge at Pin P1.4/$\overline{\text{INT2}}$/CC4
1	0	Compare Enabled
1	1	Capture on Write Operation into Register CCL4
COMO		Compare Mode Bit. When set Compare Mode 1 is selected for CC4 COMO = 0 selects Compare Mode 0.
COCOEN		Enables the Compare Mode 1 and the concurrent Compare Output for CC4. Setting of this bit automatically sets bit COMO.
COCON2 COCON1 COCON0		Selects additional concurrent Compare Outputs at Port 5. See table below.

COCON 2	COCON 1	COCON 0	Function
0	0	0	One Additional Output of CC4 at P5.0
0	0	1	Additional Outputs of CC4 at P5.0 to P5.1
0	1	0	Additional Outputs of CC4 at P5.0 to P5.2
0	1	1	Additional Outputs of CC4 at P5.0 to P5.3
1	0	0	Additional Outputs of CC4 at P5.0 to P5.4
1	0	1	Additional Outputs of CC4 at P5.0 to P5.5
1	1	0	Additional Outputs of CC4 at P5.0 to P5.6
1	1	1	Additional Outputs of CC4 at P5.0 to P5.7

CMEN

0F6H	CMEN.7	CMEN.6	CMEN.5	CMEN.4	CMEN.3	CMEN.2	CMEN.1	CMEN.0	CMEN

Contains enable bits for compare-registers CM0 to CM7. When set, compare function is enabled and the CCU is tied to the output lines.

Bit	Function
CMEN.7	Compare Enable Bit for CM7
CMEN.6	Compare Enable Bit for CM6
CMEN.5	Compare Enable Bit for CM5
CMEN.4	Compare Enable Bit for CM4
CMEN.3	Compare Enable Bit for CM3
CMEN.2	Compare Enable Bit for CM2
CMEN.1	Compare Enable Bit for CM1
CMEN.0	Compare Enable Bit for CM0

CMSEL

0F7H	CMSEL.7	CMSEL.6	CMSEL.5	CMSEL.4	CMSEL.3	CMSEL.2	CMSEL.1	CMSEL.0	CMSEL

Contains select bits for registers CM0 to CM7. When set, CMLx/CMHx are assigned to the Compare Timer and compare mode 0 is enabled. The compare registers are assigned to Timer 2 if CMSEL.X = 0. In this case compare mode 1 is selected.

Bit	Function
CMSEL.7	Select Bit for CM7
CMSEL.6	Select Bit for CM6
CMSEL.5	Select Bit for CM5
CMSEL.4	Select Bit for CM4
CMSEL.3	Select Bit for CM3
CMSEL.2	Select Bit for CM2
CMSEL.1	Select Bit for CM1
CMSEL.0	Select Bit for CM0

Table 4 illustrates possible assignments of the compare-registers for the two timers, the corresponding compare outputs and the compare modes.

T2CON

	0CFH	0CEH	0CDH	0CCH	0CBH	0CAH	0C9H	0C8H	
0C8H	T2PS	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0	T2CON

▨ These bits are not used in controlling the CCU.

Timer 2 control register. Bit-addressable register which controls Timer 2 function and compare mode of registers CRC, CC1 to CC3.

Bit		Function
T2I1	**T2I0**	**Timer 2 Input Selection**
0	0	No Input Selected, Timer 2 Stops
0	1	Timer Function, Input Frequency = $f_{osc/12}$ (T2PS = 0) or $f_{osc/24}$ (T2PS = 1)
1	0	Counter Function, External Input Signal at Pin T2/P1.7
1	1	Gated Timer Function, Input Controlled by Pin T2/P1.7
T2R1	**T2R0**	**Timer 2 Reload Mode Selection**
0	X	Reload Disabled
1	0	Mode 0: Auto-Reload upon Timer 2 Overflow (TF2)
1	1	Mode 1: Reload upon Falling Edge at Pin T2EX/P1.5
T2CM		Compare Mode Bit for Registers CRC, CC1 through CC3. When set, Compare Mode 1 is selected. T2CM = 0 selects Compare Mode 0.
T2PS		Prescaler Select Bit. When set, Timer 2 is clocked in the "Timer" or "Gated Timer" function with 1/24 of the oscillator frequency. T2PS = 0 Gates $f_{osc/12}$ to Timer 2. T2PS must be 0 for the counter operation of Timer 2.

CTCON

0E1H	—	—	—	—	CTF	CLK2	CLK1	CLK0	CTCON

Compare Timer Control Register. Contains clock selection bits for the Compare-Timer and the Compare Timer Overflow flag.

Bit	Function
CLK2 CLK1 CLK0	Compare Timer Input Clock Selection. See table below.
CTF	Compare Timer Overflow Flag. Must be cleared by software. If the Compare Timer Interrupt is Enabled, CTF = 1 will cause an Interrupt.

CLK2	CLK1	CLK0	Function
0	0	0	Compare Timer Input Clock is $f_{osc/2}$
0	0	1	Compare Timer Input Clock is $f_{osc/4}$
0	1	0	Compare Timer Input Clock is $f_{osc/8}$
0	1	1	Compare Timer Input Clock is $f_{osc/16}$
1	0	0	Compare Timer Input Clock is $f_{osc/32}$
1	0	1	Compare Timer Input Clock is $f_{osc/64}$
1	1	0	Compare Timer Input Clock is $f_{osc/128}$
1	1	1	Compare Timer Input Clock is $f_{osc/256}$

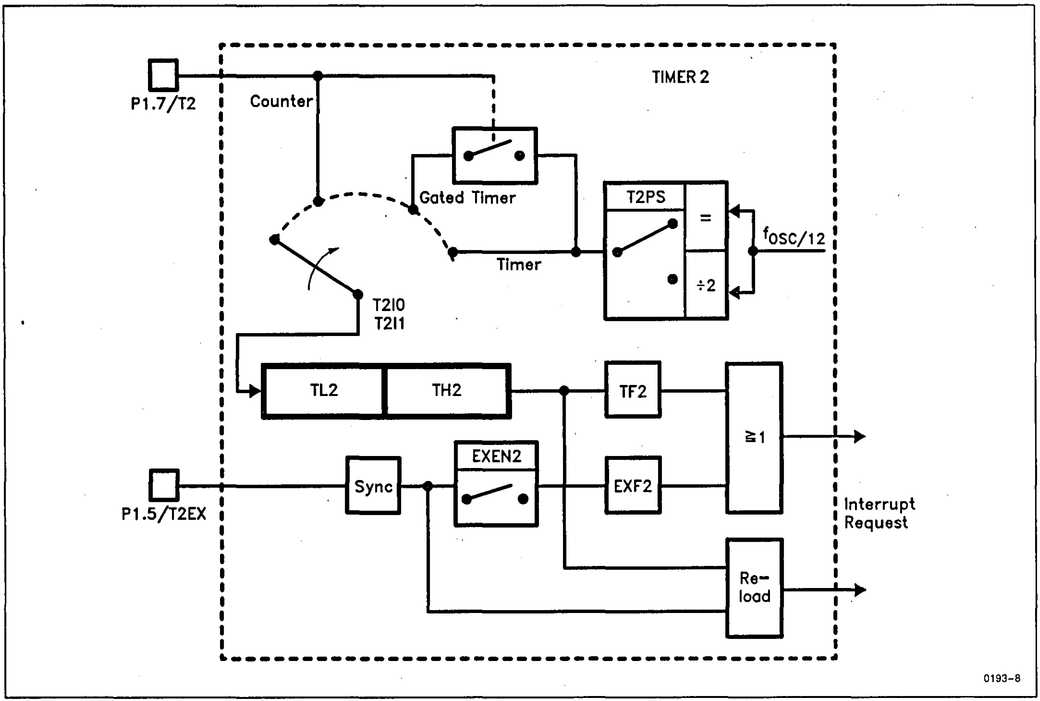


Figure 6. Block Diagram of Timer 2

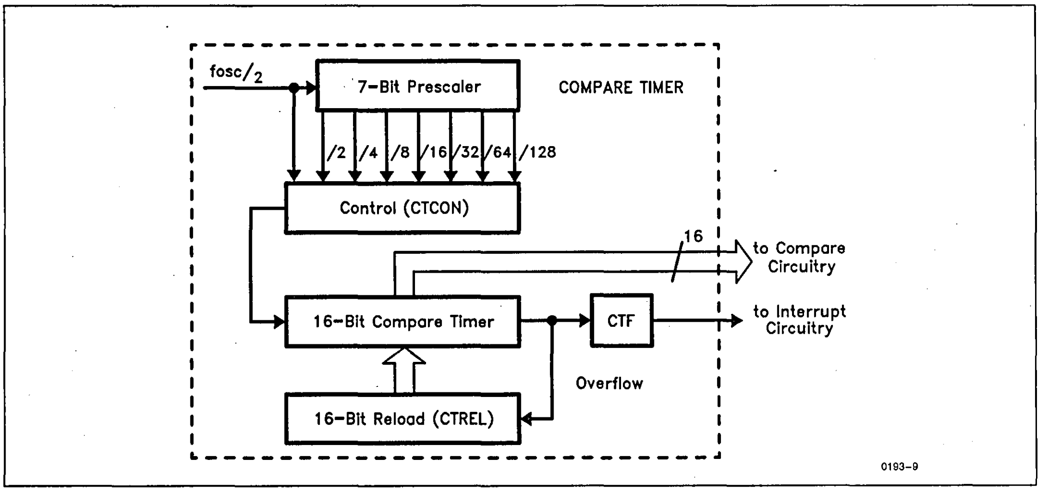


Figure 7. Block Diagram of the Compare Timer

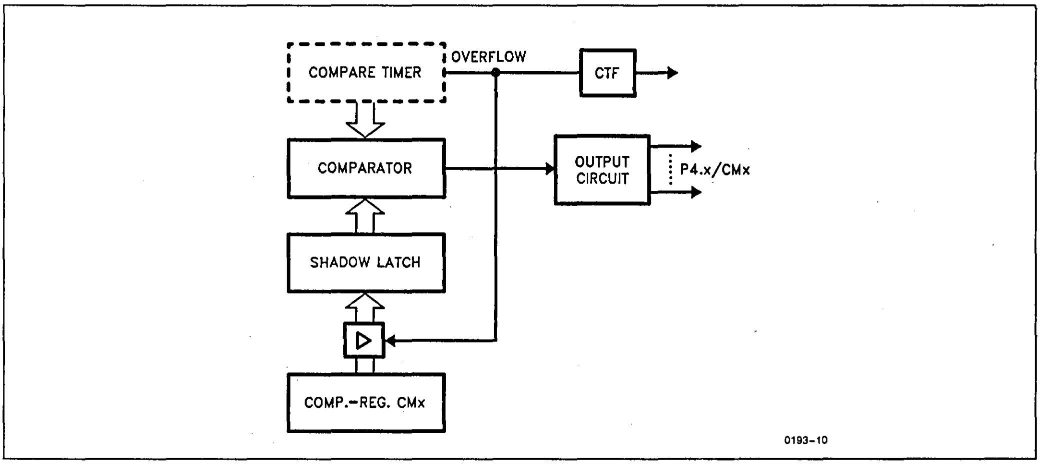


Figure 8. Compare-Mode 0 with Registers CM0 to CM7

5.3 Interrupt Structure

The SAB 80C517 has 14 interrupt vectors with the following vector addresses and request flags:

Table 5. Interrupt Sources and Vectors

Interrupt Request Flags	Interrupt Vector Address	Interrupt Source
IE0	003H	External Interrupt 0
TF0	00BH	Timer 0 Overflow
IE1	0013H	External Interrupt 1
TF1	001BH	Timer 1 Overflow
RI0/TI0	0023H	Serial Channel 0
TF2/EXF2	002BH	Timer 2 Overflow/Ext. Reload
IADC	0043H	A/D Converter
IEX2	004BH	External Interrupt 2
IEX3	0053H	External Interrupt 3
IEX4	005BH	External Interrupt 4
IEX5	0063H	External Interrupt 5
IEX6	006BH	External Interrupt 6
RI1/TI1	0083H	Serial Channel 1
CTF	008BH	Compare Timer Overflow

Each interrupt vector can be individually enabled/disabled. The response time to an interrupt request is more than 3 machine cycles and less than 9 machine cycles.

External interrupts 0 and 1 can be activated by a low-level or a negative transition (selectable) at their corresponding input pin, external interrupts 2 and 3 can be programmed for triggering on a negative or a positive transition. The external interrupts 2 to 6 are combined with the corresponding alternate functions compare (output) and capture (input) on port 1.

For programming of the priority levels the interrupt vectors are combined in pairs or triples. Each pair or triple can be programmed individually to one of the four priority levels by setting or clearing one bit in special function register IP0 and one in IP1. Figure 9 shows the interrupt request sources, their enabling bits and the prioirty level structure.

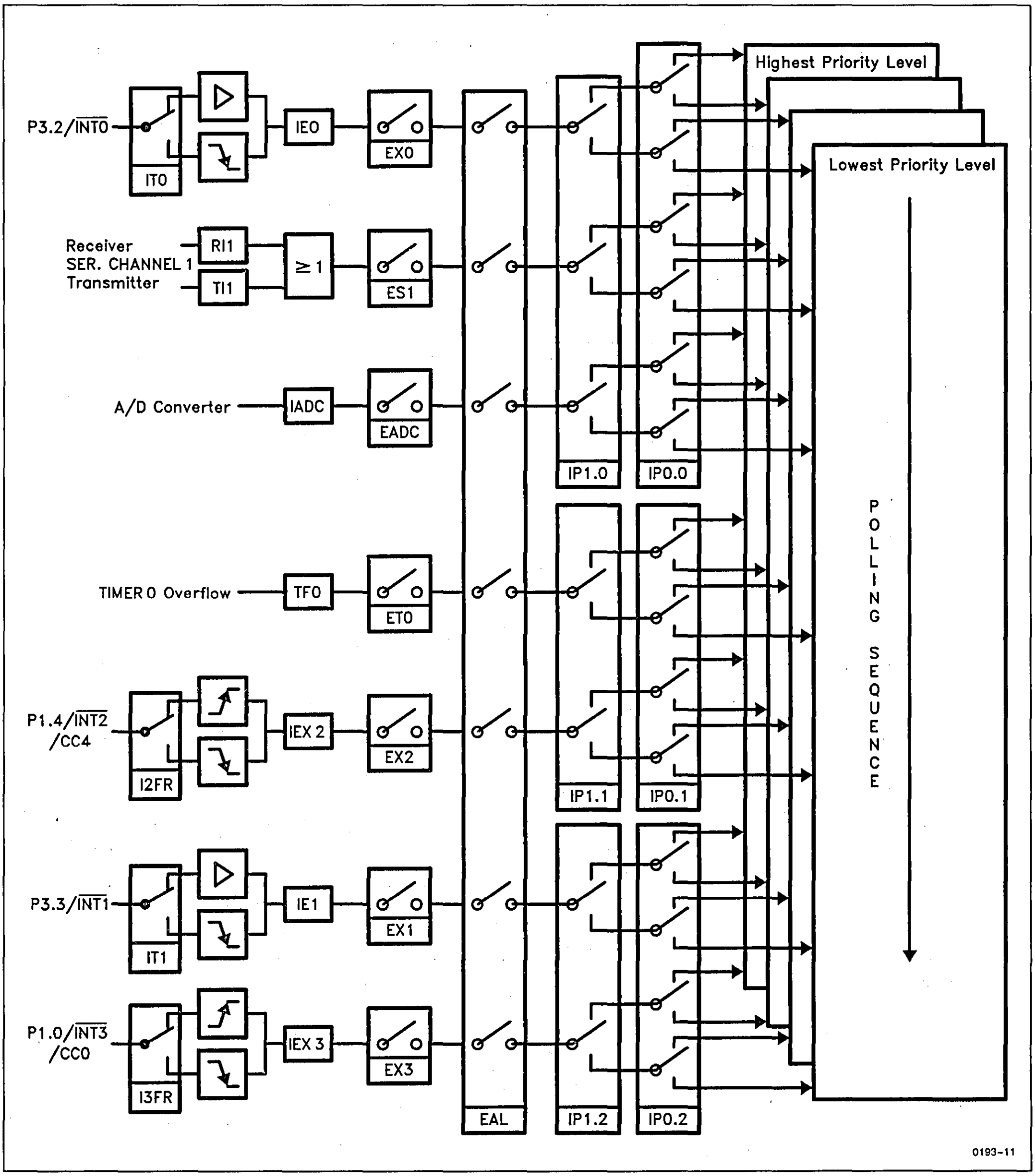


Figure 9. Interrupt Structure of the SAB 80C517

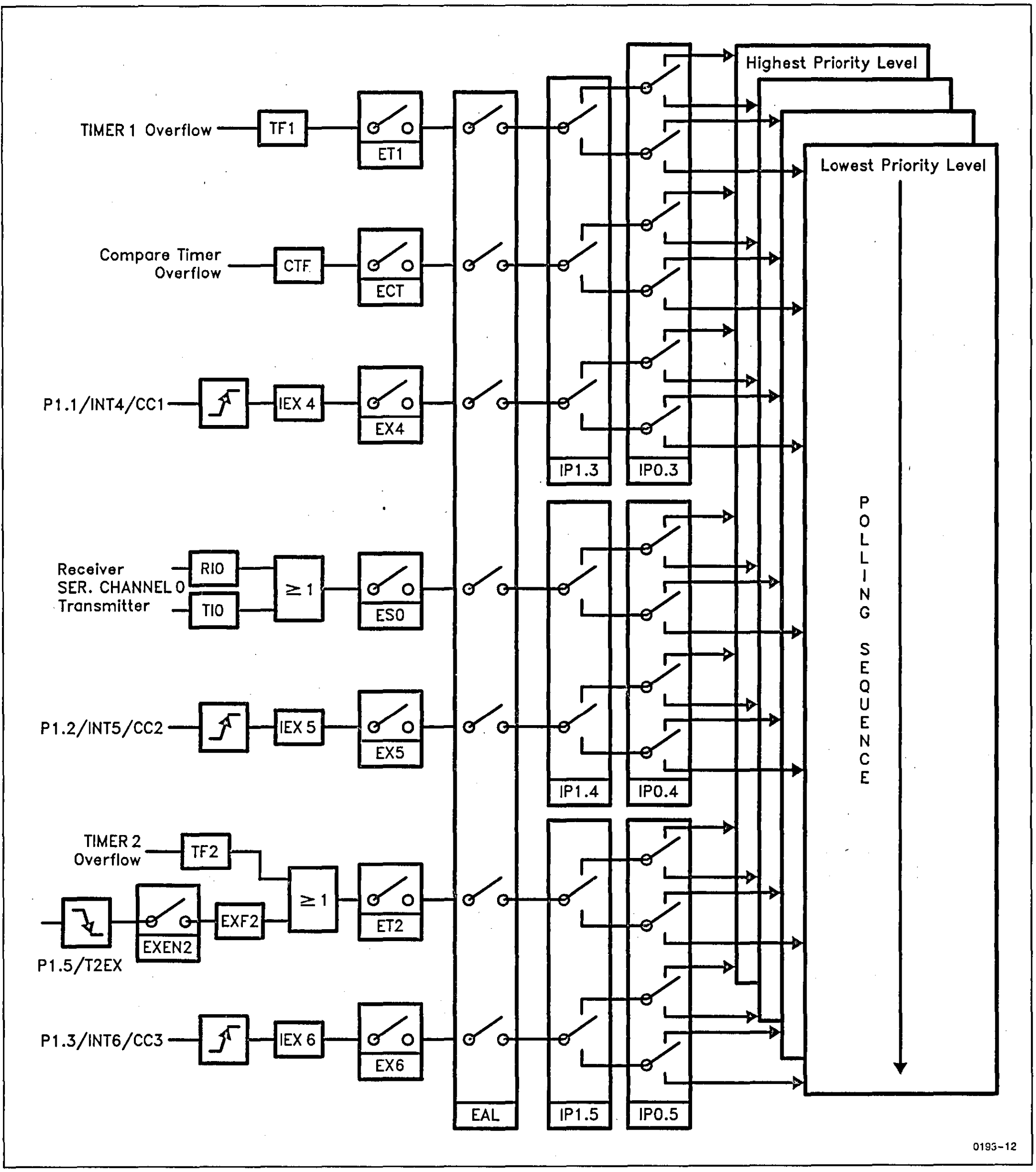


Figure 9. Interrupt Structure of the SAB 80C517 (Continued)

IRCON

	0C7H	0C6H	0C5H	0C4H	0C3H	0C2H	0C1H	0C0H	
0C0H	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	IRCON

Bit	Function
IADC	A/D converter interrupt request flag. Set by hardware at the end of a conversion. Must be cleared by software.
IEX2	External interrupt 2 edge flag. Set by hardware when external interrupt edge is detected or when a compare event occurs at pin 1.4/$\overline{\text{INT2}}$/CC4. Cleared by hardware when interrupt is processed.
IEX3	External interrupt 3 edge flag. Set by hardware when external interrupt edge is detected or when a compare event occurs at pin 1.0/$\overline{\text{INT3}}$/CC0. Cleared by hardware when interrupt is processed.
IEX4	External interrupt 4 edge flag. Set by hardware when external interrupt edge is detected or when a compare event occurs at pin 1.1/$\overline{\text{INT4}}$/CC1. Cleared by hardware when interrupt is processed.
IEX5	External interrupt 5 edge flag. Set by hardware when external interrupt edge is detected or when a compare event occurs at pin 1.2/$\overline{\text{INT5}}$/CC2. Cleared by hardware when interrupt is processed.
IEX6	External interrupt 6 edge flag. Set by hardware when external interrupt edge is detected or when a compare event occurs at pin 1.3/$\overline{\text{INT6}}$/CC3. Cleared by hardware when interrupt is processed.
TF2	Timer 2 overflow flag. Set by a Timer 2 overflow. Must be cleared by software. If the Timer 2 interrupt is enabled, TF2 = 1 will cause an interrupt.
EXF2	Timer 2 external reload flag. Set when a reload is caused by a negative transition on pin T2EX while EXEN2 = 1. When the Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. Can be used as an additional external interrupt when the reload function is not used. EXF2 must be cleared by software.

TCON

	8FH	8EH	C5H	8CH	8BH	8AH	89H	88H	
88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

▨ These bits are not used for interrupt control.

Bit	Function
IT0	Interrupt 0 Type Control Bit. Set/cleared by software to specify falling edge/low-level triggered external interrupts
IE0	Interrupt 0 Edge Flag. Set by hardware when external interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	Interrupt 1 Type Control Bit. Set/cleared by software to specify falling edge/low-level triggered external interrupts.
IE1	Interrupt 1 Edge Flag. Set by hardware when external interrupt edge is detected. Cleared by hardware when interrupt is processed.
TF0	Timer 0 Overflow Flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TF1	Timer 1 Overflow Flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

T2CON

	0CFH	0CEH	0CDH	0CCH	0CBH	0CAH	0C9H	0C8H	
0C8H	T2PS	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0	T2CON

▨ These bits are not used for interrupt control.

Bit	Function
I2FR	External interrupt 2 falling/rising edge flag. When set, the interrupt 2 request flag IEX2 will be set on a positive transition at pin P1.4/$\overline{\text{INT2}}$. I2FR = 0 specifies external interrupt 2 to be negative-transition activated.
I3FR	External interrupt 3 falling/rising edge flag. When set, the interrupt 3 request flag IEX3 will be set on a positive transition at pin P1.0/$\overline{\text{INT}}$. I3FR = specifies external interrupt 3 to be negative-transition active.

CTCON

0C8H	—	—	—	—	CTF	CLK 2	CLK 1	CLK 0	CTCON

▨ These bits are not used for interrupt control.

Bit	Function
CTF	Compare Timer Overflow Flag. Set by hardware at a rollover of the compare timer. Must be cleared by software. If the compare timer interrupt is enabled, CTF = 1 will cause an interrupt.

IEN0

0A8H	EAL	WOT ▨	ET2	ES0	ET1	EX1	ET0	EX0	IEN0

▨ These bits are not used for interrupt control.

Bit	Function
EX0	Enables or Disables External Interrupt 0. If EXO = 0, external interrupt 0 is disabled.
ET0	Enables or Disables the Timer 0 Overflow Interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.
EX1	Enables or Disables External Interrupt 1. If EX1 = 0, external interrupt 1 is disabled.
ET1	Enables or Disables the Timer 1 Overflow Interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.
ES0	Enables or Disables the Serial Channel 0 Interrupt. If ES0 = 0, the serial channel 0 interrupt is disabled.
ET2	Enables or Disables the Timer 2 Overflow or External Reload Interrupt. If ET2 = 0, the Timer 2 interrupt is disabled.
EAL	Enables or Disables All Interrupts. If EAL = 0, no interrupt will be acknowledged. If EAL = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.

IEN1

	0BFH	0BEH	0BDH	0BCH	0BBH	0BAH	0B9H	0B8H	
0B8H	EXEN2	SWDT ▨	EX6	EX5	EX4	EX3	EX2	EADC	IEN1

▨ These bits are not used for interrupt control.

Bit	Function
EADC	Enables or disables the A/D converter interrupt. If EADC = 0, the A/D converter interrupt is disabled.
EX2	Enables or disables external interrupt 2/capture/compare interrupt 4. If EX2 = 0, external interrupt 2 is disabled.
EX3	Enables or disables external interrupt 3/capture/compare interrupt 0. If EX3 = 0, external interrupt 3 is disabled.
EX4	Enables or disables external interrupt 4/capture/compare interrupt 1. If EX4 = 0, external interrupt 4 is disabled.
EX5	Enables or disables external interrupt 5/capture/compare interrupt 2. If EX5 = 0, external interrupt 5 is disabled.
EX6	Enables or disables external interrupt 6/capture/compare interrupt 3. If EX6 = 0, external interrupt 6 is disabled.
EXEN2	Enables or disables the Timer 2 external reload interrupt. EXEN2 = 0 disables the Timer 2 external reload interrupt. The external reload function is not affected by EXEN2.

IEN2

09AH	—	—	—	—	—	—	ECT	ES1	IEN2

Bit	Function
ES1	Enable Serial Interrupt of interface 1. Enables or disables the interrupt of serial interface 1. If ES1 = 0, the interrupt is disabled.
ECT	Enable Compare Timer Interrupt. Enables or disables the interrupt at compare timer overflow. If ECT = 0, the interrupt is disabled

IP0/IP1

0A9H	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0
0B9H	—	—	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	IP1

▨ These bits are not used for interrupt control.

Corresponding bit-locations in both registers are used to set the interrupt priority level of an interrupt pair or triple.

Bit		Function
IP1.x	**IP0.x**	
0	0	Set Priority Level 0 (Lowest)
0	1	Set Priority Level 1
1	0	Set Priority Level 2
1	1	Set Priority Level 3 (Highest)

Bit	Corresponding Interrupt Pair or Triple
IP1.0/IP0.0	IE0/RI1 + TI1/IADC
IP1.1/IP0.1	TF0/CTF/IEX2
IP1.2/IP0.2	IE1/IEX3
IP1.3/IP0.3	TF1/IEX4
IP1.4/IP0.4	RI0 + TI0/IEX5
IP1.5/IP0.5	TF2 + EXF2/IEX6

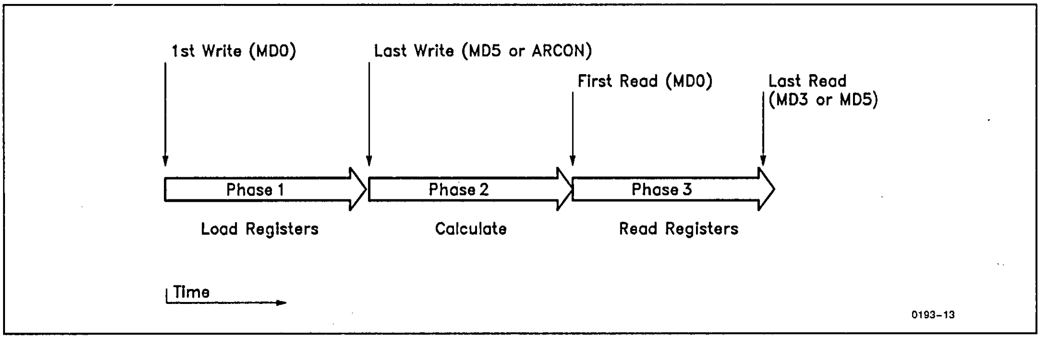
5.4 Multiplication/Division Unit

This on-chip arithmetic unit provides fast 32-bit division, 16-bit multiplication as well as shift and normalize features. All operations are integer operations.

Operation	Result	Remainder	Execution Time
32-Bit/16-Bit	32-Bit	16-Bit	6 t_{cy} (1)
16-Bit/16-Bit	16-Bit	16-Bit	4 t_{cy}
16-Bit × 16-Bit	32-Bit		4 t_{cy}
32-Bit Normalize			6 t_{cy} (2)
32-Bit Shift Left/Right			6 t_{cy} (2)

NOTE:
1. 1 t_{cy} = 1 µs @12 MHz oscillator frequency
2. The maximal shift speed is 6 shifts/cycle.

The MDU consists of six registers used for operands and results and one control register. Operation of the MDU can be divided in three phases:



To start an operation, register MD0 to MD5 (or ARCON) must be written to in a certain sequence according to Table 6 or 7. The order the registers are accessed determines the type of the operation. A shift operation is started by a final write operation to register ARCON (see also the register description).

Table 6. Performing a MDU-Calculation

Operation	32-Bit/16-Bit		16-Bit/16-Bit		16-Bit * 16-Bit	
FIRST WRITE	MD0 MD1	D'endL D'end	MD0	D'endL	MD0	M'andL
	MD2	D'end	MD1	D'endH	MD4	M'orL
	MD3 MD4	D'endH D'orL	MD4	D'orL	MD1	M'andH
LAST WRITE	MD5	D'orH	MD5	D'orH	MD5	M'orH
FIRST READ	MD0 MD1	QuoL Quo	MD0	QuoL	MD0	PrL
	MD2	Quo	MD1	QuoH	MD1	
	MD3 MD4	QuoH RemL	MD4	RemL	MD2	
LAST READ	MD5	RemH	MD5	RemH	MD3	PrH

Table 7. Shift Operation with the CCU

Operation	Normalize, Shift Left, Shift Right	
FIRST WRITE	MD0	Least Significant Byte
	MD1	
	MD2	
	MD3	Most Significant Byte
LAST WRITE	ARCON	Start of Conversion
FIRST READ	MD0	Least Significant Byte
	MD1	
	MD2	
LAST READ	MD3	Most Significant Byte

Abbreviations

D'end	Dividend, 1st operation of division
D'or	Divisor, 2nd operation of division
Quo	Quotient, result of division
M'and	Multiplicand, 1st operand of multiplication
M'or	Multiplicator, 2nd operand of multiplication
Pr	Product, result of multiplication
Rem	Remainder
... L	means, that this byte is the least significant of the 16-bit or 32-bit operand
... H	means, that this byte is the most significant of the 16-bit or 32-bit operand

MD0 to MD5

0E9H	Multiplication/Division Register 0	MD0
0EAH	Multiplication/Division Register 1	MD1
0EBH	Multiplication/Division Register 2	MD2
0ECH	Multiplication/Division Register 3	MD3
0EDH	Multiplication/Division Register 4	MD4
0EEH	Multiplication/Division Register 5	MD5

Operand registers of the MDU must be loaded in a certain sequence to start a MDU-operation (see Table 6 or 7). Registers also contain result and remainder after operation. MD0 is the first byte to be written in any operation. Writing to MD5 completes the trigger-procedure of multiplication and division.

ARCON

0EFH	MDEF	MDOV	SLR	SC.4	SC.3	SC.2	SC.1	SC.0	ARCON

Arithmetic control register. Contains control flags and the shift counter of the MDU. Triggers a shift or a normalize operation in register MD0 to MD3 when being written to.

Bit	Function
MDEF	Error flag. Indicates an improperly performed operation. MDEF is set by hardware when an operation is retriggered by a write access to MDx before the first operation has been completed. MDEF is automatically cleared after being read.
MDOV	Overflow flag. Exclusively controlled by hardware. MDOV is set by the following events: —division by zero —multiplication with a result greater than 0FFFFH —Start of normalizing if MD3.7 = 1
SLR	Shift direction bit. When set, shift right is performed. SLR = 0 selects shift left operation.
SC.4 SC.3 SC.2 SC.1 SC.0	Shift counter. When preset with 00000B, normalizing is selected. After operation, SC.0 to SC.4 contain number of normalizing shifts performed. When set with a value ≠ 0 shift operation is started. The number of shifts performed is determined by the count written to SC.0 to SC.4.

5.5 I/O Ports

The SAB 80C517 has seven 8-bit I/O-ports and two input ports (8-bit and 4-bit wide).

Port 0 is an open-drain bidirectional I/O port, while ports 1 to 6 are quasi-bidirectional I/O ports with internal pull-up resistors. That means, when configured as inputs, ports 1 to 6 will be pulled high and will source current when externally pulled low. Port 0 will float when configured as input.

Port 0 and port 2 can be used to expand the program and data memory externally. During an access to external memory, port 0 emits the low-order address byte and reads/writes the data byte, while port 2 emits the high-order address byte. In this function, port 0 is not an open-drain port, but uses a strong internal pullup FET. Port 1, 3, 4, 5 and port 6 provide several alternate functions. Please see the "Pin Description" for details.

The SAB 80C517 has two dual-purpose input ports. The twelve port lines at port 7 and port 8 can be used as analog inputs for the A/D converter. If input voltages at P7 and P8 meet the specified digital input levels (V_{IL} and V_{IH}) the port can also be used as digital input port.

P0

	87H	86H	85H	84H	83H	82H	81H	80H	
80H	P.07	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	P0

During any access to external memory the CPU writes 0FFH to the port 0 latch therefore obliterating any previous contents the port latch had.

P2

	0A7H	0A6H	0A5H	0A4H	0A3H	0A2H	0A1H	0A0H	
0A0H	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	P2

The contents written to the port 2 latch are not affected during external memory access. If an eight bit address is used to address external memory, the contents of the port 2 latch remain at the port 2 pins.

P1, P3, P4, P5, P6

	097H	096H	095H	094H	093H	092H	091H	090H	
90H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1
	0B7H	0B6H	0B5H	0B4H	0B3H	0B2H	0B1H	0B0H	
0B0H	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	P3
	0EFH	0EEH	0EDH	0ECH	0EBH	0EAH	0E9H	0E8H	
0E8H	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	P4
	0FFH	0FEH	0FDH	0FCH	0FBH	0FAH	0F9H	0F8H	
0F8H	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0	P5
0FAH	Port 6								P6

Port pins show the information written to these port latches, when used as general purpose port. When an alternate function is used, the port pin is controlled by the respective peripheral unit. Therefore the port latch must contain a "one" for that function to operate. The same applies when the port pins are used as inputs. Ports 1, 3, 4 and 5 are bit-addressable.

P7, P8

0DCH	Port 7	P7
0DDH	Port 8	P8

Reading these port latches allows the user to input the digital values currently applied to the port pins. This digital input function is independent from the analog input function for the A/D converter. Contents of P7 and P8 are indeterminate if the levels at the corresponding pins are not within their respective V_{IL}/V_{IH} specifications.

5.6 Power Saving Modes

The SAB 80C517 provides three modes in which power consumption can be significantly reduced.

- The *Slow Down Mode*

  The controller keeps up the full operating functionality, but is driven with the eighth part of its normal operating frequency. Slowing down the frequency greatly reduces power consumption.
- The *Idle Mode*

  The CPU is gated off from the oscillator, but all peripherals are still supplied by the clock and able to work.
- The *Power Down Mode*

  Operation of the SAB 80C517 is stopped, the oscillator is turned off. This mode is used to save the contents of the internal RAM with a very low standby current.

All of these modes are entered by software. Special function register PCON (Power Control register, address is 87H) is used to select one of these modes.

Hardware Enable for Power Saving Modes

A dedicated Pin ($\overline{\text{PE}}$/SWD) of the SAB 80C517 allows blocking of the power saving modes. Since this pin is mostly used in noise-critical application it is combined with an automatic start of the Watchdog Timer (see Watchdog Timer for further description).

$\overline{\text{PE}}$/SWD = V_{IH} (logic high level):

Using the power saving modes is not possible. The instruction sequences used for entering of these modes will not affect the normal operation of the device.

$\overline{\text{PE}}$/SWD = V_{IL} (logic low level):

All power saving modes can be activated by software.

When left unconnected, Pin $\overline{\text{PE}}$/SWD is pulled to high level by a weak internal pullup. This is done to provide system protection on default.

The logic-level applied to pin $\overline{\text{PE}}$/SWD can be changed during program execution to allow or to block the use of the power saving modes without any effect on the on-chip watchdog circuitry.

Power Down Mode

The Power Down Mode is entered by two consecutive instructions immediately following each other. The first instruction has to set the flag PDE (Power Down Enable) and must not set PDS (Power Down Set). The following instruction has to set the start bit PDS. Bits PDE and PDS will automatically be cleared after having been set.

The instruction that sets bit PDS is the last instruction executed before going into Power Down Mode. The only exit from power down mode is a hardware reset.

The status of all output lines of the controller can be looked up in Table 8.

Idle Mode

During Idle Mode all peripherals of the SAB 80C517 are still supplied by the oscillator clock. Thus the user has to take care which peripheral should continue to run and which has to be stopped during Idle.

The procedure to enter the Idle mode is similar to entering the power down mode. The two bits IDLE and IDLS must be set by two consecutive instructions to minimize the chance of unintentional activation of the Idle Mode.

There are two ways to terminate the idle mode:

- The idle mode can be terminated by activating any enabled interrupt. This interrupt will be serviced and in normal cases the instruction to be executed following the RETI instruction will be the one following the instruction that sets the bit IDLS.
- The other way to terminate the idle mode, is a hardware reset. Since the oscillator is still running, the hardware reset must be held active only for two machine cycles for a complete reset.

Normally the port pins hold the logical state they had at the time idle mode was activated. If some pins are programmed to serve their alternate functions they still continue to output during idle mode if the assigned function is on. The control signals ALE and $\overline{\text{PSEN}}$ hold at logic high levels (see Table 8).

Slow Down Mode

During slow down operation all signal frequencies that are derived from the oscillator clock, are divided by eight, also the clockout signal and the watchdog timer count.

The Slow Down Mode is enabled by setting bit SD. The controller actually enters the Slow Down Mode after a short synchronization period (max. 2 machine cycles).

The slow down mode is disabled by clearing bit SD.

Table 8. Status of External Pin during Idle and Power Down

Outputs	Last Instruction Executed from Internal Code Memory		Last Instruction Executed from External Code Memory	
	Idle	Power Down	Idle	Power Down
ALE	HIGH	LOW	HIGH	LOW
$\overline{\text{PSEN}}$	HIGH	LOW	HIGH	LOW
PORT 0	DATA	DATA	FLOAT	FLOAT
PORT 1	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT
PORT 2	DATA	DATA	ADDRESS	DATA
PORT 3	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT
PORT 4	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT
PORT 5	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT
PORT 6	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT	DATA/ALTERNATE OUTPUTS	DATA/LAST OUTPUT

Special Function Registers for Power Down Mode:

PCON

87H	▨ SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE	PCON

▨ This bit is not used in controlling the power saving modes.

Bit	Function
PDS	Power down start bit. The instruction that sets the PDS flag bit is the last instruction before entering the power down mode.
IDLS	Idle start bit. The instruction that sets the IDLS flag bit is the last instruction before entering the idle mode.
SD	When set, the slow done mode is enabled.
GF1	General purpose flag
GF0	General purpose flag
PDE	Power down enable bit. When set, starting the power down mode is enabled.
IDLE	Idle mode enable bit. When set, starting the idle mode is enabled.

5.7 Serial Interfaces

The SAB 80C517 has two serial interfaces. Both interfaces are full duplex and receive buffered. They are functionally identical with the serial interface of the SAB 8051 when working as asynchronous channels. Serial interface 0 additionally has a synchronous mode.

5.7.1 Serial Interface 0

Serial Interface 0 can operate in 4 modes:

Mode 0: Shift register mode:

Serial data enters and exits through RxD0. TxD0 outputs the shift clock. 8 data bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: 8-bit UART, variable baud rate:

10 bits are transmitted (through TxD0) or received (through RxD0): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB80 in special function register S0CON. The baud rate is variable.

Mode 2: 9-bit UART, fixed baud rate.

11 bits are transmitted (through TxD0) or received (through RxD0): a start bit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). On transmission, the 9th data bit (TB80 in S0CON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB80 or a second stop bit by setting TB80 to 1. On reception the 9th data bit goes into RB80 in special function register S0CON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 3: 9-bit UART, variable baud rate:

11 bits are transmitted (through TxD0) or received (through RxD0): a start bit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). In fact, mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable.

Variable Baud Rates for Serial Interface 0:

Variable baud rates for modes 1 and 3 of Serial Interface 0 can be derived from either Timer 1 or from the oscillator via a special prescaler ("BD").

Timer 1 may be operated in mode 1 (to generate slow baud rates) or mode 2. The dedicated baud rate generator "BD" provides the two standard baud rates 4800 baud or 9600 baud. Tables 9 and 10 show possible configurations and the according baud rates.

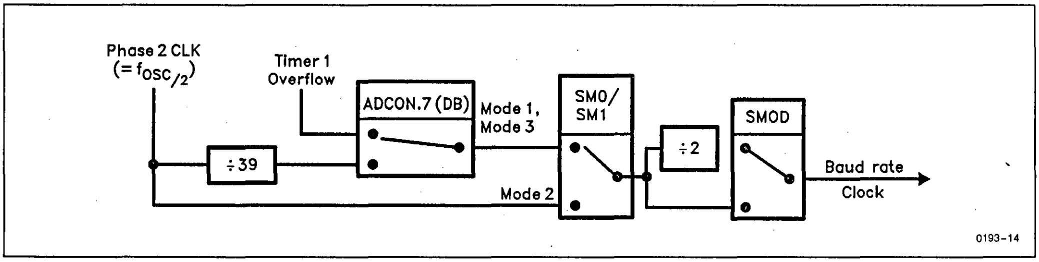


Figure 10. Generation of the Baud Rates for Serial Interface 0

Special Function Register for Serial Interface 0:

S0CON

	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	
98H	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0	S0CON

Bit		Function	
SM0	**SM1**		
0	0	Serial Mode 0:	Shift Register Mode
0	1	Serial Mode 1:	8-Bit UART, Variable Baud Rate
1	0	Serial Mode 2:	9-Bit UART, Fixed Baud Rate
1	1	serial Mode 3:	9-Bit UART, Variable Baud Rate
SM20		Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3, if SM20 is set to 1 then RI0 will not be activated if the received 9th data bit (RB80) is 0. In mode 1, if SM20 = 1 then RI0 will not be activated if a valid stop bit was not received. In mode 0, SM20 should be 0.	
REN0		Receiver Enable. Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.	
TB80		Transmitter Bit 8. Is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.	
RB80		Receiver Bit 8. In modes 2 and 3, is the 9th data bit that was received. In mode 1, if SM20 = 0, RB80 is the stop bit that was received. In mode 0, RB80 is not used.	
TI0		Transmitter Interrupt. Is the transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.	
RI0		Receiver Interrupt. Is the receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or during the stop bit time in the other modes, in any serial reception. Must be cleared by software.	

S0BUF

99H	Serial Interface 0 Buffer Register	S0BUF

Receive and transmit buffer of Serial Interface 0. Writing to S0BUF loads the transmit register and initiates transmission. Reading out S0BUF accesses a physically separate receive register.

ADCON0

	0DFH	0DEH	0DDH	0DCH	0DBH	0DAH	0D9H	0D8H	
0D8H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0	ADCON0

▨ These bits are not used in controlling serial interface 0.

Bit	Function
BD	Baud Rate Enable. When set, the baud rate in mode 1 and 3 of serial interface 0 is taken from a dedicated prescaler (see Figure 10). Standard baud rates 4800 baud and 9600 baud at 12 MHz oscillator frequency can be achieved.

PCON

87H	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE	PCON

▨ These bits are not used in controlling serial interface 0.

Bit	Function
SMOD	When set, the baud rate of Serial Interface 0 in modes 1, 2, 3 is doubled.

5.7.2. Serial Interface 1

Serial Interface 1 can operate in two asynchronous modes:

Mode A: 9-bit UART, variable baud rate.

11 bits are transmitted (through TxD1) or received (through RxD1): a start bit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). On transmission, the 9th data bit (TB81 in S1CON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB81 or a second stop bit by setting TB81 to 1. On reception the 9th data bit goes into RB81 in special function register S1CON, while the stop bit is ignored.

Mode B: 8-bit UART, variable baud rate.

10 bits are transmitted (through TxD1) or received (through RxD1): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB81 in special function register S1CON.

Variable Baud Rates for Serial Interface 1

Variable Baud Rates for modes A and B of Serial Interface 1 can be derived from a dedicated baud rate generator.

The baud rate clock $\left(\text{baud rate} = \dfrac{\text{baud rate clock}}{16}\right)$ is generated by an 8-bit free running timer with programmable reload register (see Figure 11).

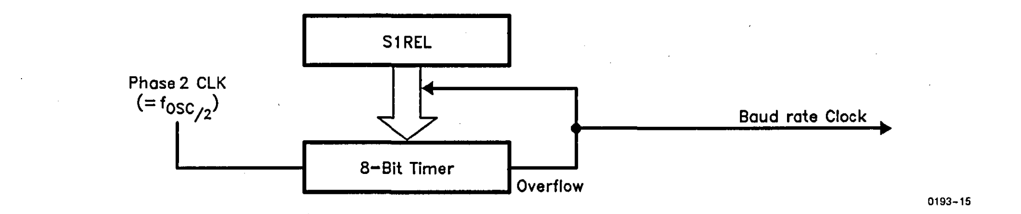


Figure 11. Baud Rate Generator for Serial Interface 1

Table 9. Calculating the Baud Rates

	Baud Rate (Derived from)	Interface Mode	Baud Rate
Serial Interface 0	Timer 1 in Mode 1	1, 3	$\frac{2^{SMOD}}{2} \times \frac{1}{16} \times$ (Timer 1 Overflow Rate)
	Timer 1 in Mode 2	1, 3	$\frac{2^{SMOD}}{2} \times \frac{1}{16} \times \frac{f_{OSC}}{12\,(256 - (TH1))}$
	Oscillator	2	$\frac{2^{SMOD}}{2} \times \frac{1}{16} \times \frac{f_{OSC}}{2}$
	BD	1, 3	$\frac{2^{SMOD}}{2} \times \frac{f_{OSC}}{1250}$
Serial Interface 1	8-Bit Baud Rate Generator	A	$\frac{1}{16} \times \frac{f_{OSC}}{2\,(256 - (S1REL))}$
		B	$\frac{1}{16} \times \frac{f_{OSC}}{2\,(256 - (S1REL))}$

Table 10. Baud Rate Generation

Function		Serial Interface 0		Serial Interface 1
8-Bit Synchronous Channel	Mode	Mode 0		
	Baud Rate	1 MHz @ f_{OSC} = 12 MHz		
	Baud Rate (Derived from)	f_{OSC}		
8-Bit UART	Mode	Mode 1		Mode B
	Baud Rate*	1K–62.5K	4800, 9600	1.5K–375K
	Baud Rate (Derived from)	Timer 1	BD	8-Bit Baud Rate Generator
9-Bit UART	Mode	Mode 2	Mode 3	Mode A
	Baud Rate*	187.5K/375K	1K–62.5K	1.5K–375K
	Baud Rate (Derived from)	$f_{OSC/2}$	Timer 1	8-Bit Baud Rate Generator

*Baud Rate values are given for 12 MHz oscillator frequency.

Special Function Register for Serial Interface 1:

S1CON

9BH	SM	—	SM21	REN1	TB81	RB81	TI1	RI1	S1CON

Bit	Function

SM	SM = 0: Serial Mode A; 9-bit UART
	SM = 1: Serial Mode B; 8-bit UART

SM21	Enables the multiprocessor communication feature in mode A. If SM21 is set to 1 then RI1 will not be activated if the received 9th data bit (RB81) is 0. In mode B, if SM21 = 1 then RI1 will not be activated if a valid stop bit was not received.
REN1	Receiver Enable of interface 1. Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB81	Transmitter Bit 8 of interface 1. Is the 9th data bit that will be transmitted in mode A. Set or cleared by software as desired.
RB81	Receiver Bit 8 of interface 1. Is the 9th data bit that was received in mode A. In mode B, if SM21 = 0, RB81 is the stop bit that was received.
TI1	Transmitter Interrupt of interface 1. Is the transmit interrupt flag. Set by hardware at the beginning of the stop bit in any serial transmission. Must be cleared by software.
RI1	Receiver Interrupt of interface 1. Is the receive interrupt flag. Set by hardware at the halfway through the stop bit time in any serial reception. Must be cleared by software.

S1REL

9DH	Serial Interface 1 Reload Register	S1REL

8-bit reload register for baud rate generator of Serial Interface 1.

S1BUF

9CH	Serial Interface 1 Buffer Register	S1BUF

Receive and transmit buffer of Serial Interface 1. Writing to S1BUF loads the transmit register and initiates transmission. Reading out S1BUF accesses a physically separate receive register.

5.8 Timer/Counters 0 and 1

These timer/counters are fully compatible with Timer/Counter 0 or 1 of the SAB 8051 and can operate in four modes:

Mode 0: 8-bit timer/counter with 32:1 prescaler

Mode 1: 16-bit timer/counter

Mode 2: 8-bit timer/counter with 8-bit auto reload

Mode 3: Timer/Counter 0 is configured as one 8-bit timer; Timer/Counter 1 in this mode holds its count.

External inputs $\overline{INT0}$ and $\overline{INT1}$ can be programmed to function as a gate for Timer/Counters 0 and 1 to facilitate pulse width measurements.

Special Function Registers:

TCON

	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	
88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

▨ These bits are not used in controlling Timer/Counter 0 and 1.

Bit	Function
TR0	Timer 0 run control bit. Set/cleared by software to turn Timer/counter 0 on/off.
TF0	Timer 0 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	Timer 1 run control bit. Set/cleared by software to turn Timer/counter 1 on/off.
TF1	Timer 1 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

TMOD

89H	GATE	$\overline{C/T}$	M1	M0	GATE	$\overline{C/T}$	M1	M0	TMOD
	Timer 1				Timer 0				

Timer/counter 0/1 mode control register

Bit		Function
GATE		Gating control. When set, timer/counter "x" is enabled only while "$\overline{INTx}$" pin is high and "TRx" control bit is set. When cleared timer "x" is enabled whenever "TRx" control bit is set.
$\overline{C/T}$		Counter or timer select bit. Set for counter operation (input from "Tx" input pin). Cleared for timer operation (input from internal system clock).
M1 0	**M0** 0	8-bit timer/counter. "THx" operates as 8-bit timer/ counter "TLx" serves as 5-bit prescaler.
0	1	16-bit timer/counter. "THx" and "TLx" are cascaded; there is no prescaler.
1	0	8-bit auto-reload timer/counter. "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	Timer 0: TL0 is an 8-bit timer/counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	Timer/tcounter 1 stops

5.9 Watchdog Units

The SAB 80C517 offers two enhanced fail safe mechanisms, which allow an automatic recovery from hardware failure or software upset:

— programmable Watchdog Timer (WDT), variable from 512 μs up to about 1.1s time out period @12 MHz. Upward compatible with the SAB 80515 watchdog timer.

— oscillator watchdog (OWD), monitors the on-chip oscillator and forces the microcontroller to go into reset state, in case the on-chip oscillator fails.

Programmable Watchdog Timer

The WDT can be activated by hardware or software.

Hardware initialization is done when Pin $\overline{\text{PE}}$/SWD (Pin 4) is held high during RESET. The SAB 80C517 then starts program execution with the WDT running. Pin $\overline{\text{PE}}$/SWD doesn't allow dynamic switching of the WDT.

Software initialization is done by setting bit SWDT. A refresh of the Watchdog Timer is done by setting bits WDT and SWDT consecutively.

A block diagram of the Watchdog Timer is shown in Figure 12.

When a Watchdog Timer reset occurs, the Watchdog Timer keeps on running, but a status flag WDTS is set. This flag can also be manipulated by software (see Figure 14).

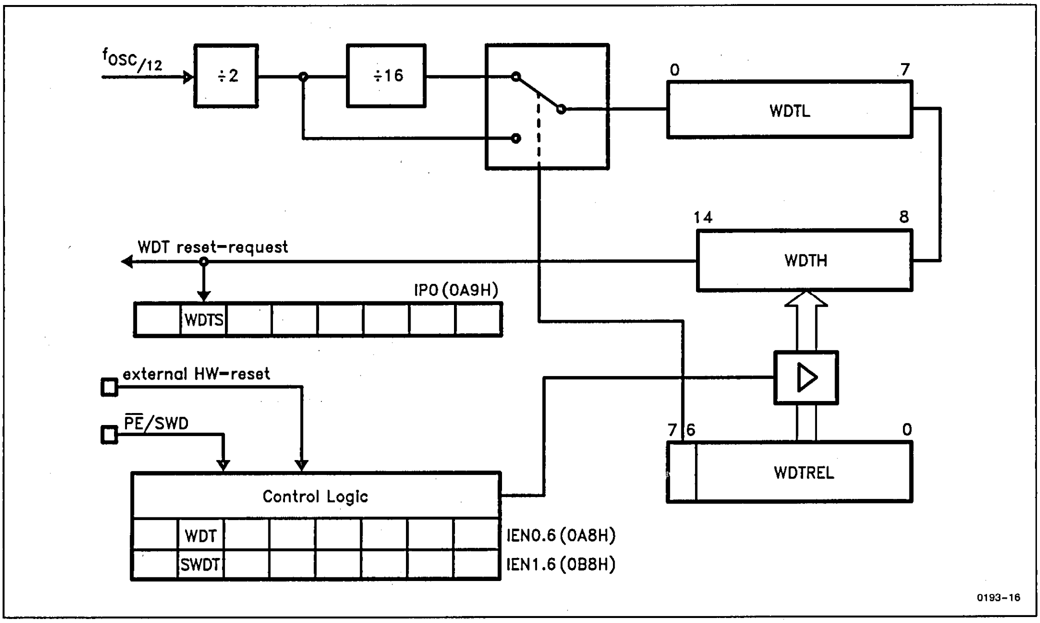


Figure 12. Block Diagram of the Programmable Watchdog Timer

Oscillator Watchdog

The Oscillator Watchdog monitors the on-chip quartz oscillator. A detected oscillator failure ($f_{OSC} \leq 300$ KHz) causes a hardware reset. The reset state is held until the on-chip oscillator is working again. The Oscillator Watchdog feature is enabled by a high level at pin OWE (Pin 69). An Oscillator Watchdog reset sets status flag OWDS which can be examined and modified by software (see Figure 14). Figure 13 shows a block diagram of the Oscillator Watchdog.

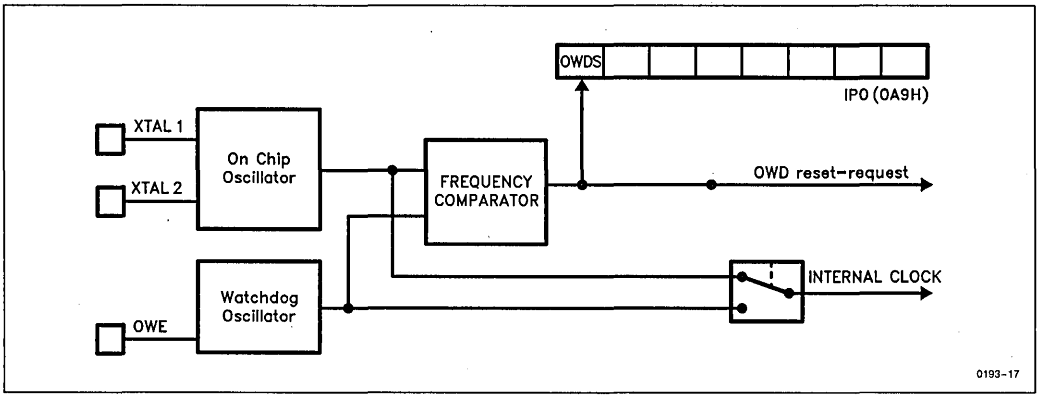


Figure 13. Functional Block Diagram of the Oscillator Watchdog

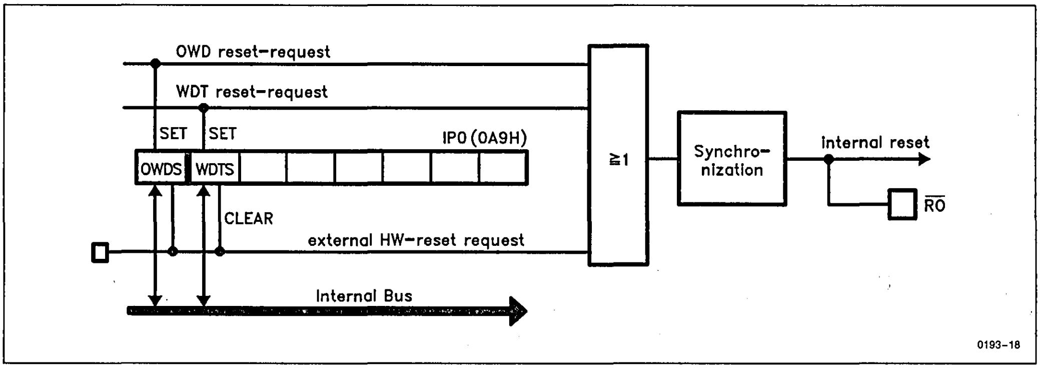


Figure 14. Watchdog Status Flags and Reset-Requests

Special Function Registers:

IEN0

	0AFH	0AEH	0ADH	0ACH	0ABH	0AAH	0A9H	0A8H	
0A8H	EAL	WDT	ET2	ES0	ET1	EX1	ET0	EX0	IEN0

▨ These bits are not used in controlling the fail safe mechanisms.

Bit	Function
WDT	Watchdog Timer refresh flag. Set to initiate a refresh of the Watchdog Timer. Must be set directly before SWDT is set to prevent unintentional refreshing of the Watchdog Timer.

IEN1

	0BFH	0BEH	0BDH	0BCH	0BBH	0BAH	0B9H	0B8H	
0B8H	EXEN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IEN1

▨ These bits are not used in controlling the fail safe mechanisms.

Bit	Function
SWDT	Watchdog Timer start flag. Initially set to activate the Watchdog Timer. When directly set after setting of bit WDT, a Watchdog Timer refresh is performed.

IP0

0A9H	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0

▨ These bits are not used in controlling the fail safe mechanisms.

Bit	Function
OWDS	Oscillator Watchdog Timer status flag. Set by hardware when an Oscillator Watchdog reset occurred. Can be cleared or set by software.
WDTS	Watchdog Timer status flag. Set by hardware when a Watchdog Timer reset occurred. Can be cleared or set by software.

WDTREL

086H	Watchdog Timer Reload Register	WDTREL

Bit	Function
WDTREL.7	Prescaler Select Bit. When set, the Watchdog Timer is clocked through an additional divide-by-16 prescaler (see Figure 12).
WDTREL.6 to WDTREL.0	Seven bit reload value for the high-byte of the Watchdog Timer. This value is loaded to the WDT when a refresh is triggered by a consecutive setting of bits WDT and SWDT.

Absolute Maximum Ratings

Ambient Temperature under Bias	
SAB 80C517	0°C to +70°C
SAB 80C517-T40/85	−40°C to +85°C
Storage Temperature	−65°C to +150°C
Voltage on V_{CC} Pins with Respect to Ground (V_{SS})	−0.5V to +6.5V
Voltage on any Pin with Respect to Ground (V_{SS})	−0.5V to V_{CC} + 0.5V
Power Dissipation	2W

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = 0°C to +70°C for SAB 80C517/80C537;
T_A = −40°C to +85°C for SAB 80C517/80C537-T40/85

Parameter	Symbol	Test Conditions	Limit Values		Unit
			Min	Max	
Input Low Voltage (except $\overline{EA}$)	V_{IL}		−0.5	0.2 V_{CC} − 0.1	V
Input Low Voltage ($\overline{EA}$)	V_{IL1}		−0.5	0.2 V_{CC} − 0.3	V
Input High Voltage (except $\overline{RESET}$ and XTAL2)	V_{IH}		−0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
Input High Voltage to XTAL2	V_{IH1}		0.7 V_{CC}	V_{CC} + 0.5	V
Input High Voltage to $\overline{RESET}$	V_{IH2}		0.6 V_{CC}	V_{CC} + 0.5	V
Output Low Voltage, Ports 1, 2, 3, 4, 5, 6	V_{OL}	I_{OL} = 1.6 mA(1)		0.45	V
Output Low Voltage, Port 0, ALE, $\overline{PSEN}$, $\overline{RO}$	V_{OL1}	I_{OL} = 3.2 mA(1)		0.45	V
Output High Voltage, Ports 1, 2, 3, 4, 5, 6	V_{OH}	I_{OH} = −80 μA, I_{OH} = −10 μA	2.4 0.9 V_{CC}		V
Output High Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$, $\overline{RO}$)	V_{OH1}	I_{OH} = −800 μA(2), I_{OH} = −80 μA(2)	2.4 0.9 V_{CC}		V
Logic 0 Input Current, Ports 1, 2, 3, 4, 5, 6	I_{IL}	V_{IN} = 0.45V		−50	μA
Logical 1-to-0 Transition Current, Ports 1, 2, 3, 4, 5, 6	I_{TL}	V_{IN} = 2V		−650	μA
Input Leakage Current (Port 0, $\overline{EA}$, OWE, $\overline{PE}$/SWD	I_{LI}	0.45 < V_{IN} < V_{CC}		±10	μA
Input Low Current to $\overline{RESET}$ for Reset	I_{IL2}	V_{IN} = 0.45V		−100	μA
Pin Capacitance	C_{IO}	f_C = 1 MHz, T_A = 25°C		10	pF
Power Supply Current Active Mode Idle Mode	I_{CC}	V_{CC} = 5V, f_{OSC} = 12 MHz (4), (5)		 50 14	mA
Power Down Current	I_{PD}	V_{CC} = 2V to 5.5V(3)		50	μA

A/D Converter Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V, T_A = 0°C to 70°C for
SAB 80C517/80C537; T_A −40°C to +85°C for
SAB 80C517/80C537-T40/85; V_{AREF} = V_{CC} ±5%; V_{AGND} = V_{SS} ±0.2V; I_{VAREF} − I_{VAGND} ≥ 1V

Parameter	Symbol	Test Conditions	Limit Values			Unit
			Min	Typ	Max	
Analog Input Voltage	V_{AINPUT}	(Note 8)	V_{AGND} − 0.2		V_{AREF} + 0.2	V
Analog Input Capacitance	C_I	(Note 6)		25	60	pF
Load Time(6)	T_L	(Note 6)			2 TCY	µs
Sample Time (Inc. Load Time)	T_S	(Note 6)			7 TCY	µs
Conversion Time (Inc. Sample Time)	T_C	(Note 6)			13 TCY	µs
Differential Non-Linearity	DNLE	I_{VAREF} = V_{AREF} = V_{CC}, I_{VAGND} = V_{AGND} = V_{SS}		±½	±1	LSB
Integral Non-Linearity	INLE			±½	±1	
Offset Error				±½	±1	
Gain Error				±½	±1	
Total Unadjusted Error(6)	TUE	(Note 6)		±1	±2	
Internal Reference Error	$V_{intREFERR}$	(Note 7)			TBD	
V_{AREF} Supply Current (7)	I_{REF}	(Note 7)			5	mA

NOTES:

1. Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and ports 1, 3, 4, 5 and 6. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation. In the worst case (capacitive loading > 100 pF), the noise pulse on an ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt-trigger, or use an address latch with a Schmitt-trigger strobe input.
2. Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V_{CC} specification when the address lines are stabilizing.
3. Power down I_{CC} is measured with all output pins disconnected; $\overline{EA}$ = $\overline{RESET}$ = V_{CC}; Port 0 = Port 7 = Port 8 = V_{CC}; XTAL1 = N.C; XTAL2 = V_{SS}; $\overline{PE}$/SWD = OWE = V_{SS}.
4. I_{CC} (active mode) is measured with all output pins disconnected; XTAL2 driven with TCLCH, TCLCL = 5 ns, V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} − 0.5V; XTAL1 = N.C.; $\overline{EA}$ = OWE = $\overline{PE}$/SWD = V_{CC}; Port 0 = Port 7 = Port 8 = V_{CC}; $\overline{RESET}$ = V_{SS}. I_{CC} would be slightly higher if a crystal oscillator is used.
5. Idle I_{CC} is measured with all output pins disconnected and with all peripherals disabled; XTAL2 driven with TCLCH, TCLCL = 5 ns, V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} − 0.5V; XTAL1 = N.C.; $\overline{RESET}$ = OWE = V_{CC}; Port 0 = Port 7 = Port 8 = V_{CC}; $\overline{EA}$ = $\overline{PE}$/SWD = V_{SS}.
6. The output impedance of the analog source must be low enough to assure full loading of the sample capacitance (C_I) during load time (T_L). After charging of the internal capacitance (C_I) in the load time (T_L) the analog input must be held constant for the rest of the sample time (T_S).
7. The differential impedance r_D of the analog reference voltage source must be less than 1 KΩ at reference supply voltage.
8. Exceeding the limit values at one or more input channels will cause additional current which is sinked/sourced at these channels. This may also affect the accuracy of other channels which are operated within the specification.

AC Characteristics

V_{CC} = 5V ±10%; V_{SS} = 0V; T_A = 0°C to 70°C for SAB 80C517/80C537; T_A = −40°C to +85°C for SAB 80C51780C537-T40/85; (C_L for port 0, ALE and $\overline{\text{PSEN}}$ outputs = 100 pF; C_L for all outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1 MHz to 12 MHz		
		Min	Max	Min	Max	
ALE Pulse Width	t_{LHLL}	127		$2\,t_{CLCL} - 40$		ns
Address Setup to ALE	t_{AVLL}	53		$t_{CLCL} - 30$		ns
Address Hold after ALE	t_{LLAX}	48		$t_{CLCL} - 35$		ns
Address to Valid Instr In	t_{LLIV}		233		$4\,t_{CLCL} - 100$	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	58		$t_{CLCL} - 25$		ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	215		$3\,t_{CLCL} - 35$		ns
$\overline{\text{PSEN}}$ to Valid Instr In	t_{PLIV}		150		$3\,t_{CLCL} - 100$	ns
Input Instruction Hold after PSEN	t_{PXIX}	0		0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	t_{PXIZ}*		63		$t_{CLCL} - 20$	ns
Address Valid after $\overline{\text{PSEN}}$	t_{PXAV}*	75		$t_{CLCL} - 8$		ns
Address to Valid Instr In	t_{AVIV}		302		$5\,t_{CLCL} - 115$	ns
Address Float to $\overline{\text{PSEN}}$	t_{AZPL}		0		0	ns

*Interfacing the SAB 80C517 to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1 MHz to 12 MHz		
		Min	Max	Min	Max	
$\overline{\text{RD}}$ Pulse Width	t_{RLRH}	400		$6\,t_{CLCL} - 100$		ns
$\overline{\text{WR}}$ Pulse Width	t_{WLWH}	400		$6\,t_{CLCL} - 100$		ns
Address Hold after ALE	t_{LLAX2}	132		$2\,t_{CLCL} - 35$		ns
$\overline{\text{RD}}$ to Valid Data In	t_{RLDV}		252		$5\,t_{CLCL} - 165$	ns
Data Hold after $\overline{\text{RD}}$	t_{RHDX}	0		0		ns
Data Float after $\overline{\text{RD}}$	t_{RHDZ}		97		$2\,t_{CLCL} - 70$	ns
ALE to Valid Data In	t_{LLDV}		517		$8\,t_{CLCL} - 150$	ns

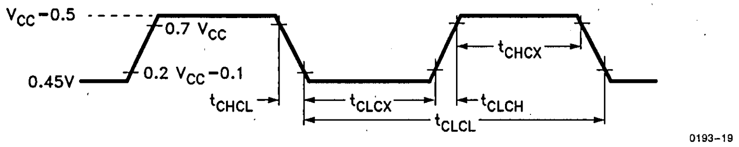
External Data Memory Characteristics (Continued)

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1 MHz to 12 MHz		
		Min	Max	Min	Max	
Address to Valid Data In	t_{AVDV}		585		$9\ t_{CLCL} - 165$	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	200	300	$3\ t_{CLCL} - 50$	$3\ t_{CLCL} + 50$	ns
$\overline{WR}$ or $\overline{RD}$ High to ALE High	t_{WHLH}	43	123	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
Address Valid to $\overline{WR}$	t_{AVWL}	203		$4\ t_{CLCL} - 130$		ns
Data Valid to $\overline{WR}$ Transition	t_{QVWX}	33		$t_{CLCL} - 50$		ns
Data Setup before $\overline{WR}$	t_{QVWH}	433		$7\ t_{CLCL} - 150$		ns
Data Hold after $\overline{WR}$	t_{WHQX}	33		$t_{CLCL} - 50$		ns
Address Float after $\overline{RD}$	t_{RLAZ}		0		0	ns

External Clock Drive

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 1 MHz to 12 MHz		
		Min	Max	
Oscillator Period	t_{CLCL}	83, 3	1000	ns
High Time	t_{CHCX}	20		ns
Low Time	t_{CLCX}	20		ns
Rise Time	t_{CLCH}		20	ns
Fall Time	t_{CHCL}		20	ns
Oscillator Frequency	$1/t_{CLCL}$	1	12	MHz

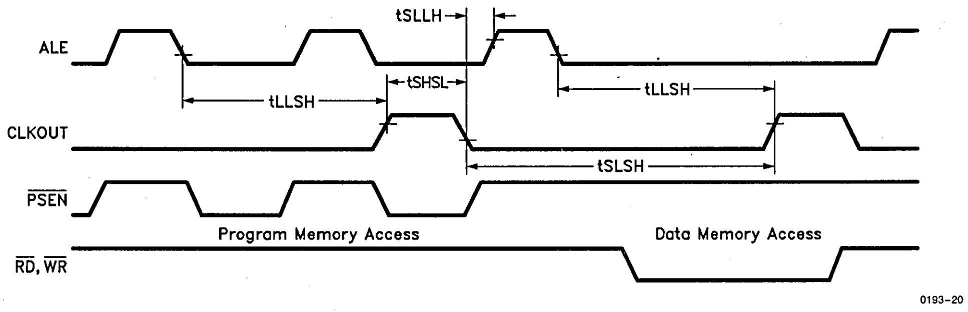
External Clock Cycle



System Clock Timing

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1 MHz to 12 MHz		
		Min	Max	Min	Max	
ALE to CLKOUT	t_{LLSH}	543		$7\ t_{CLCL} - 40$		ns
CLKOUT High Time	t_{SHSL}	127		$2\ t_{CLCL} - 40$		ns
CLKOUT Low Time	t_{SLSH}	793		$10\ t_{CLCL} - 40$		ns
CLKOUT Low to ALE High	t_{SLLH}	43	123	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns

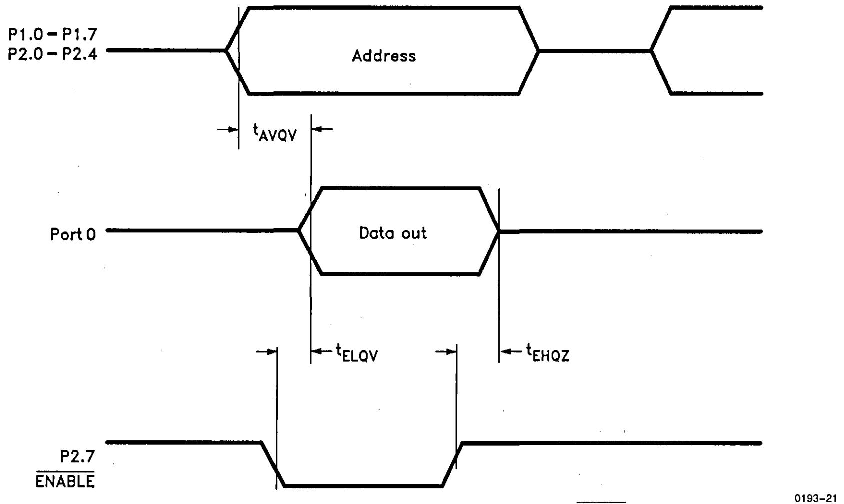
System Clock Timing



ROM Verification Characteristics

Parameter	Symbol	Limit Values		Unit
		Min	Max	
Address to Valid Data	t_{AVQV}		$48\ t_{CLCL}$	ns
ENABLE to Valid Data	t_{ELQV}			
Data Float after ENABLE	t_{EHQZ}	0		
Oscillator Frequency	$1t_{CLCL}$	4	6	MHz

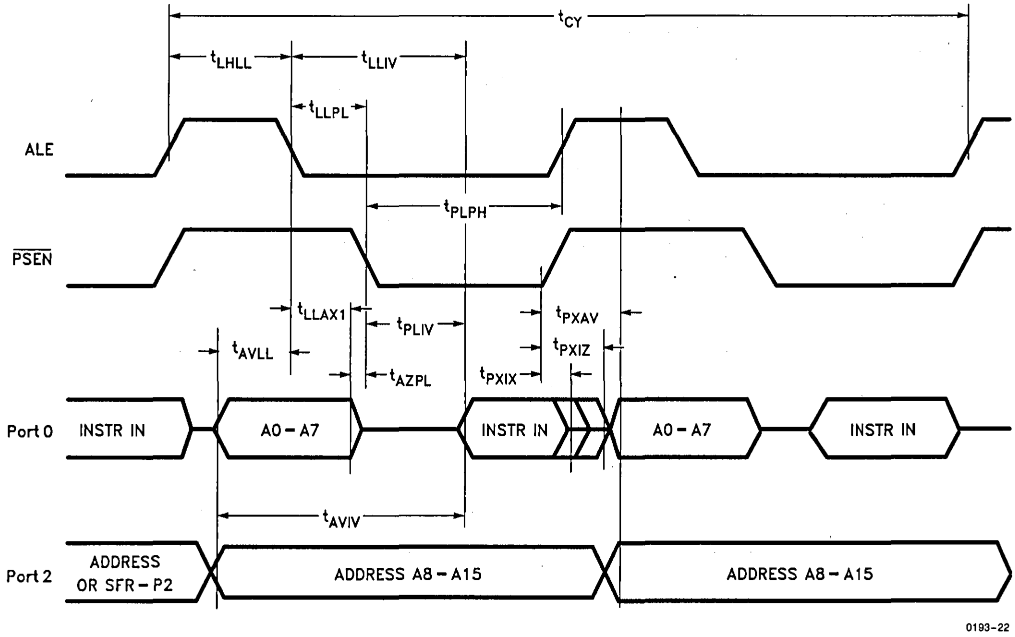
ROM Verification



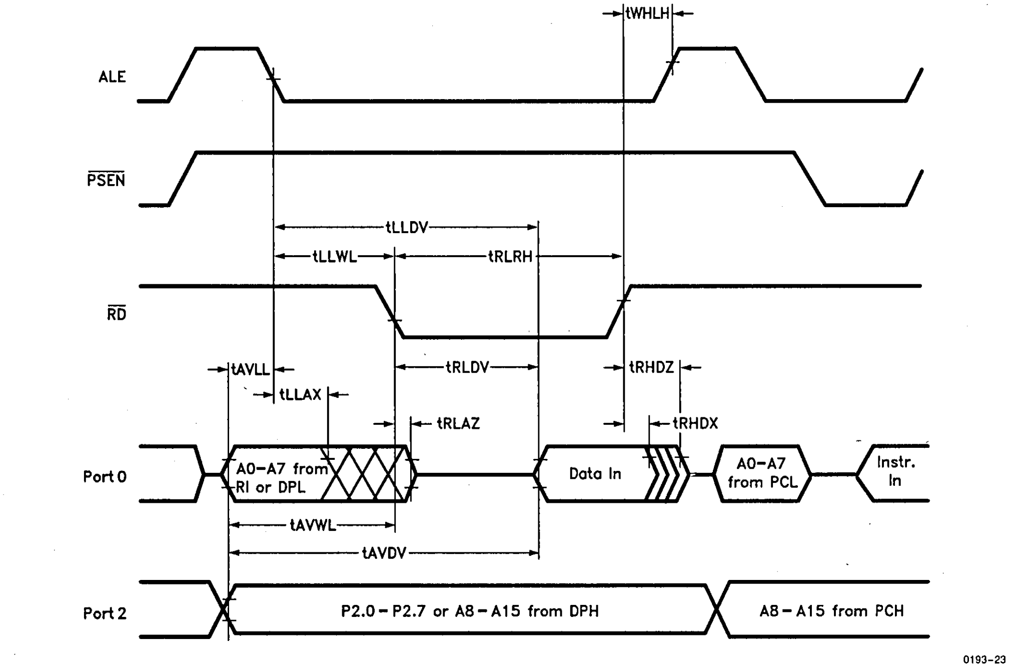
Address: P1.0 - P1.7 = A0 – A7
P2.0 – P2.4 = A8 - A12
Data: Port 0 = D0 – D7

Inputs: P2.5 - 2.6, $\overline{PSEN}$ = V_{SS}
ALE, $\overline{EA}$ = V_{IH}
$\overline{RESET}$ = V_{IL}

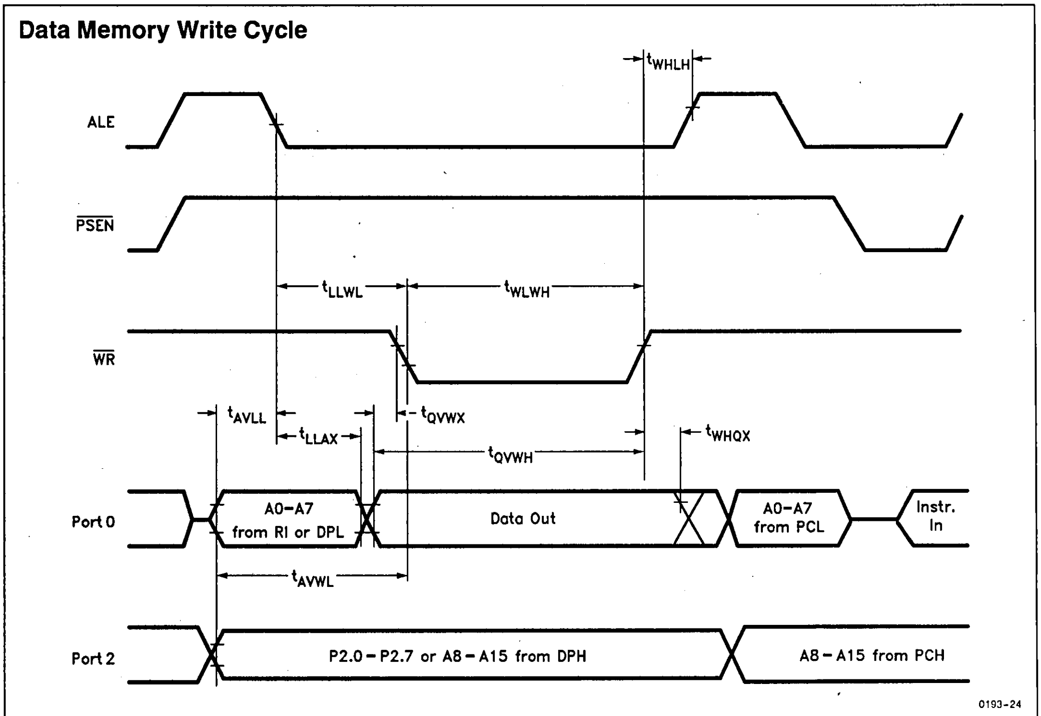
Program Memory Read Cycle



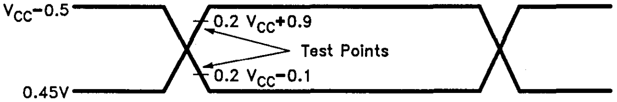
Data Memory Read Cycle



Data Memory Write Cycle



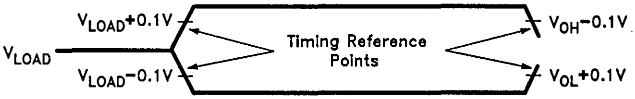
AC Testing: Input, Output Waveforms



0193-25

AC Inputs during testing are driven at V_{CC} − 0.5V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at V_{IHmin} for a logic "1" and V_{ILmax} for a logic "0".

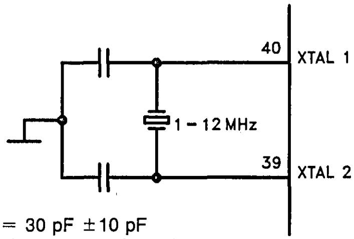
AC Testing: Float Waveforms



0193-26

For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. I_{OL}/I_{OH} ≥ ±20 mA.

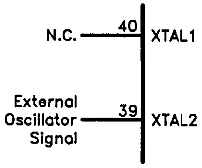
Recommended Oscillator Circuits



C = 30 pF ±10 pF
(incl. stray capacitance)

0193-27

Crystal Oscillator Mode



0193-28

Driving from External Source

Instruction Set

5.0 Instruction Set

The instruction set of the SAB 8051 family of microcontrollers includes 111 instructions, 49 of which are single-byte, 45 two-byte, and 17 three-byte instructions. The instruction opcode format consists of a function mnemonic followed by a "destination, source" operand field. This field specifies the data type and addressing method(s) to be used.

All members of the 8051 family can be programmed with the same instruction set common to the basic member, the SAB 8051.

All microcontrollers are 100% software compatible with the SAB 8051 and may be programmed with 8051 assembler or high-level language, ASM51 or PLM51 respectively.

5.1 Addressing Modes

The SAB 8051 family of microcontrollers uses five addressing modes:

- register
- direct
- register indirect
- immediate
- base register plus index register indirect

Table 5-1 summarizes which memory spaces may be accessed by each of the addressing modes.

Table 5-1

Addressing Modes	Associated Memory Spaces
Register Addressing	R0 through R7 of Selected Register Bank ACC, B, CY (Bit), DPTR
Direct Addressing	Lower 128 Bytes of Internal RAM Special-Function Register
Register Indirect Addressing	Internal RAM (@R1, @R0, SP) External Data Memory (@R1, @R0, @DPTR)
Immediate Addressing	Program Memory
Base Register plus Register Addressing	Program Memory (@DPTR + A, @PC + A)

Register Addressing

Register addressing accesses the eight working registers (R0–R7) of the selected register bank. The least significant bits of the instruction opcode indicate which register is to be used. ACC, B, DPTR and CY, the Boolean processor accumulator can also be addressed as registers.

Direct Addressing

Direct addressing is the only method of accessing the special-function registers. The 128 bytes of internal RAM are also directly addressable.

Register Indirect Addressing

Register indirect addressing uses the contents of either R0 or R1 (in the selected register bank) as a pointer to locations in a 256-byte block: the 128 bytes of internal RAM or the lower 256 bytes of external data memory. Note that the special-function registers are not accessible by this method. Access to the full 64 Kbyte of external data memory address space is accomplished by using the 16-bit data pointer. Execution of PUSH and POP instructions also uses register indirect addressing. The stack may reside anywhere in the internal RAM.

Immediate Addressing

Immediate addressing allows constants to be part of the instruction in program memory.

Base Register plus Index Register Addressing

Base register plus index register addressing allows a byte to be accessed from program memory via an indirect move from the location whose address is the sum of a base register (DPTR or PC) and index register ACC. This mode facilitates look-up-table accesses.

Boolean Processor

The Boolean processor is a bit processor integrated within the SAB 8051 family of microcontrollers. It has its own instruction set, accumulator (the carry flag), and bit-addressable RAM and I/O.

- set bit
- clear bit
- complement bit
- jump if bit is set
- jump if bit is not set
- jump if bit is set and clear bit
- move bit from/to carry

Addressable bits, or their complements, may be logically ANDed or ORed with the contents of the carry flag. The result is returned to the carry register.

5.2 Introduction to the Instruction Set

The instruction set is divided into four functional groups:

- data transfer
- arithmetic
- logic
- control transfer

5.2.1 Data Transfer

Data operations are divided into three classes:

- general-purpose
- accumulator-specific
- address-object

None of these operations affects the PSW flag settings except a POP or MOV directly to the PSW.

General-Purpose Transfers

- MOV performs a bit or byte transfer from the source operand to the destination operand.
- PUSH increments the SP register and then transfers a byte from the source operand to the stack location currently addressed by SP.
- POP transfers a byte operand from the stack location addressed by the SP to the destination operand and then decrements SP.

Accumulator-Specific Transfers

- XCH exchanges the byte source operand with register A (accumulator)
- XCHD exchanges the low-order nibble of the source operand byte with the low-order nibble of A.
- MOVX performs a byte move between the external data memory and the accumulator. The external address can be specified by the DPTR register (16 bits) or the R1 or R0 register (8 bits).
- MOVC moves a byte from program memory to the accumulator. The operand in A is used as an index into a 256-byte table pointed to by the base register (DPTR or PC). The byte operand accessed is transferred to the accumulator.

Address-Object Transfer

- MOV DPTR, # data loads 16 bits of immediate data into a pair of destination registers, DPH and DPL.

5.2.2 Arithmetic

The SAB 8051 family of microcontrollers has four basic mathematical operations. Only 8-bit operations using unsigned arithmetic are supported directly. Refer to the technical description of the SAB 80C517 for 16-bit and 32-bit arithmetic operations. The overflow flag, however, permits the addition and subtraction operation to serve for both unsigned and signed binary integers. Arithmetic can also be performed directly on packed BCD representations.

Addition

- INC (increment) adds one to the source operand and puts the result in the operand.
- ADD adds A to the source operand and returns the result to A.
- ADDC (add with carry) adds A and the source operand, then adds one (1) if CY is set, and puts the result in A.
- DA (decimal-add-adjust for BCD addition) corrects the sum which results from the binary addition of two-digit decimal operands. The packed decimal sum formed by DA is returned to A. CY is set if the BCD result is greater than 99; otherwise it is cleared.

Subtraction

- SUBB (subtract with borrow) subtracts the second source operand from the first operand (the accumulator), subtracts one (1) if CY is set and returns the result to A.
- DEC (decrement) subtracts one (1) from the source operand and returns the result to the operand.

Multiplication

- MUL performs an unsigned multiplication of the A register, returning a double-byte result. A receives the low-order byte, B receives the high-order byte. OV is cleared if the top half of the result is zero and is set if it is not zero. CY is cleared. AC is unaffected.

Division

- DIV performs an unsigned division of the A register by the B register, and returns the integer quotient to A and returns the fractional remainder to the B register. Division by zero leaves indeterminate data in registers A and B and sets OV; otherwise OV is cleared. CY is cleared. AC is unaffected.

Flags

Unless otherwise stated in the above descriptions, the flags of PSW are affected as follows:

- CY is set if the operation causes a carry to or a borrow from the resulting high-order bit. Otherwise CY is cleared.
- AC is set if the operation results in a carry from the low-order four bits of the result (during addition), or a borrow from the high-order bits to the low-order bits (during subtraction); otherwise AC is cleared.
- OV is set if the operation results in a carry to the high-order bit of the result but not a carry from the bit, or vice versa; otherwise OV is cleared. OV is used in two's-complement arithmetic, because it is set when the signal result cannot be represented in 8 bits.
- P is set if the modulo 2 sum of the eight bits in the accumulator is 1 (odd parity); otherwise P is cleared (even parity). When a value is written to the PSW register, the P bit remains unchanged, as it always reflects the parity of A.

5.2.3 Logic

The SAB 8051 family of microcontrollers perform basic logic operations on both bit and byte operands.

Single-Operand Operations

- CLR sets A or any directly addressable bit to zero (0).
- SETB sets any directly bit-addressable bit to one (1).
- CPL is used to complement the contents of the A register without affecting any flag, or any directly addressable bit location.
- RL, RLC, RR, RRC, SWAP are the five operations that can be performed on A. RL rotate left RR, rotate right, RLC rotate left through carry, RRC rotate right through carry, and SWAP rotate left four. For RLC and RRC the CY flag becomes equal to the last bit rotated out. SWAP rotates A left four places to exchange bits 3 through 0 with bits 7 through 4.

Two-Operand Operations

- ANL performs bitwise logical ANDing of two operands (for both bit and byte operands) and returns the result to the location of the first operand.
- ORL performs bitwise logical ORing of two source operands (for both bit and byte operands) and returns the result to the location of the first operand.
- XRL performs bitwise logical exclusive ORing of two source operands (byte operands) and returns the result to the location of the first operand.

5.2.4 Control Transfer

There are three classes of control transfer operations: unconditional calls, returns and jumps, conditional jumps, and interrupts. All control transfer operations cause, some upon a specific condition, the program execution to continue a non-sequential location in program memory.

Unconditional Calls, Returns and Jumps

Unconditional calls, returns and jumps transfer control from the current value of the program counter to the target address. Both direct and indirect transfers are supported.

- ACALL and LCALL push the address of the next instruction onto the stack and then transfer control to the target address. ACALL is a 2-byte instruction used when the target address is in the current 2K page. LCALL is a 3-byte instruction that addresses the full 64K program space. In ACALL, immediate data (i.e. an 11-bit address field) is concatenated to the five most significant bits of the PC (which is pointing to the next instruction). If ACALL is in the last 2 bytes of a 2K page then the call will be made to the next page since the PC will have been incremented to the next instruction prior to execution.

- RET transfers control to the return address saved on the stack by a previous call operation and decrements the SP register by two (2) to adjust the SP for the popped address.
- AJMP, LJMP and SJMP transfer control to the target operand. The operation of AJMP and LJMP are analogous to ACALL and LCALL. The SJMP (short jump) instruction provides for transfers within a 256-byte range centered about the starting address of the next instruction (−128 to +127).
- JMP@A + DPTR performs a jump relative to the DPTR register. The operand in A is used as the offset (0–255) to the address in the DPTR register. Thus the effective destination for a jump can be anywhere in the program memory space.

Conditional Jumps

Conditional jumps perform a jump contingent upon a specific condition. The destination will be within a 256-byte range centered about the starting address of the next instruction (−128 to +127).

- JZ performs a jump if the accumulator is zero.
- JNZ performs a jump if the accumulator is not zero.
- JC performs a jump if the carry flag is set.
- JNC performs a jump if the carry flag is not set.
- JB performs a jump if the directly addressed bit is set.
- JNB performs a jump if the directly addressed bit is not set.
- JBC performs a jump if the directly addressed bit is set and then clears the directly addressed bit.
- CJNE compares the first operand to the second operand and performs a jump if they are not equal. CY is set if the first operand is less than the second operand; otherwise it is cleared. Comparisons can be made between A and directly addressable bytes in internal data memory or an immediate value and either A, a register in the selected register bank, or a register indirect addressed byte of the internal RAM.
- DJNZ decrements the source operand and returns the result to the operand. A jump is performed if the result is not zero. The source operand of the DJNZ instruction may be any directly addressable byte in the internal data memory. Either direct or register addressing may be used to address the source operand.

Interrupt Returns

- RETI transfers control as RET does, but additionally enables interrupts of the current priority level.

5.3 Instruction Definitions

All 111 instructions of the SAB 8051 family of microcontrollers can essentially be condensed to 54 basic operations, in the following ordered alphabetically according to the operation mnemonic section.

A brief example of how the instruction might be used is given as well as its effect on the PSW flags. The number of bytes and machine cycles required, the binary machine language encoding, and a symbolic description or restatement of the function is also provided.

Note: Only the carry, auxiliary carry, and overflow flags are discussed. The parity bit is computed after every instruction cycle that alters the accumulator. Similarly, instructions which alter directly addressed registers could affect the other status flags if the instruction is applied to the PSW. Status flags can also be modified by bit manipulation.

Instruction	Flag			Instruction	Flag		
	CY	OV	AC		CY	OV	AC
ADD	X	X	X	SETB C	1		
ADDC	X	X	X	CLR C	0		
SUBB	X	X	X	CPL C	X		
MUL	0	X		ANL C, bit	X		
DIV	0	X		ANL C,/bit	X		
DA	X			ORL C, bit	X		
RRC	X			ORL C,/bit	X		
RLC	X			MOV C, bit	X		
CJNE	X						

Notes on Data Addressing Modes

Rn — Working register R0–R7

direct — 128 internal RAM locations, any I/O port, control or status register

@Ri — Indirect internal or external RAM location addressed by register R0 or R1

#data — 8-bit constant included in instruction

#data 16 — 16-bit constant included as bytes 2 and 3 of instruction

bit — 128 software flags, any I/O pin, control or status bit

A — Accumulator

Notes on Program Addressing Modes

addr 16 — Destination address for LCALL and LJMP may be anywhere within the 64-Kbyte program memory address space.

addr 11 — Destination address for ACALL and AJMP will be within the same 2-Kbyte page of program memory as the first byte of the following instruction.

rel — SJMP and all conditional jumps include an 8-bit offset byte. Range is +127/−128 bytes relative to the first byte of the following instruction.

ACALL addr11

Function: Absolute Call

Description: ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the stack pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, op code bits 7-5, and the second byte of the instruction. The subroutine called must therefore start within the same 2k block of the program memory as the first byte of the instruction following ACALL. No flags are affected.

Example: Initially SP equals 07H. The label "SUBRTN" is at program memory location 0345H. After executing the instruction,

```
ACALL   SUBRTN
```

at location 0123H, SP will contain 09H, internal RAM locations 08H and 09H will contain 25H and 01H, respectively, and the PC will contain 0345H.

Operation:

```
ACALL
(PC) ← (PC) + 2
(SP) ← (SP) + 1
((SP)) ← (PC7-0)
(SP) ← (SP) + 1
((SP)) ← (PC15-8)
(PC10-0) ← page address
```

Encoding:

a10 a9 a8 1	0 0 0 1		a7 a6 a5 a4	a3 a2 a1 a0

Bytes: 2

Cycles: 2

ADD A, <src-byte>

Function: Add

Description: ADD adds the byte variable indicated to the accumulator, leaving the result in the accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occured.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B). The instruction,

```
ADD   A,R0
```

will leave 6DH (01101101B) in the Accumulator with the AC flag cleared and both the carry flag and OV set to 1.

ADD A,Rn

Operation: ADD
(A) ← (A) + (Rn)

Encoding:

0 0 1 0	1 r r r

Bytes: 1

Cycles: 1

ADD A,direct

Operation: ADD
(A) ← (A) + (direct)

Encoding:

0 0 1 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ADD A,@Ri

Operation: ADDC
(A) ← (A) + (C) + ((Ri))

Encoding:

0 0 1 0	0 1 1 i

Bytes: 1

Cycles: 1

ADD A,#data

Operation: ADD
(A) ← (A) + #data

Encoding:

0 0 1 0	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

ADDC A, <src-byte>

Function: Add with Carry

Description: ADDC simultaneously adds the byte variable indicated, the carry flag and the accumulator contents, leaving the result in the accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occured.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) with the carry flag set. The instruction,

ADDC A,R0

will leave 6EH (01101110B) in the accumulator with AC cleared and both the Carry flag and OV set to 1.

ADDC A,Rn

Operation: ADDC
(A) ← (A) + (C) +(Rn)

Encoding:

0 0 1 1	1 r r r

Bytes: 1

Cycles: 1

ADDC A,direct

Operation: ADDC
(A) ← (A) + (C) + (direct)

Encoding:

0 0 1 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ADDC A,@Ri

Operation: ADDC
(A) ← (A) + (C) + ((Ri))

Encoding:

0 0 1 1	0 1 1 i

Bytes: 1

Cycles: 1

ADDC A,#data

Operation: ADDC
(A) ← (A) + (C) + #data

Encoding:

0 0 1 1	0 1 0 0		immediate data

Bytes: 2

Cycles: 1

AJMP addr11

Function: Absolute Jump

Description: AJMP transfers program execution to the indicated address, which is formed at run-time by concatenating the high-order five bits of the PC (*after* incrementing the PC twice), op code bits 7–5, and the second byte of the instruction. The destination must therefore be within the same 2k block of program memory as the first byte of the instruction following AJMP.

Example: The label "JMPADR" is at program memory location 0123H. The instruction,

AJMP JMPADR

is at location 0345H and will load the PC with 0123H.

Operation: AJMP
(PC) ← (PC) + 2
(PC10–0) ← page address

Encoding:

a10 a9 a8 0	0 0 0 1		a7 a6 a5 a4	a3 a2 a1 a0

Bytes: 2

Cycles: 2

ANL <dest-byte>,<src-byte>

Function: Logical-AND for byte variables

Description: ANL performs the bitwise logical-AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the accumulator or immediate data.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: If the accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) then the instruction,

ANL A,R0

will leave 81H (10000001B) in the accumulator.

When the destination is a directly addressed byte, this instruction will clear combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the accumulator at run-time. The instruction,

ANL P1,#01110011B

will clear bits 7, 3, and 2 of output port 1.

ANL A,Rn

Operation: ANL
(A) ← (A) ∧ (Rn)

Encoding:

0 1 0 1	1 r r r

Bytes: 1

Cycles: 1

ANL A,direct

Operation: ANL
(A) ← (A) ∧ (direct)

Encoding:

0 1 0 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ANL A,@Ri

Operation: ANL
(A) ← (A) ∧ ((Ri))

Encoding:

0 1 0 1	0 1 1 i

Bytes: 1

Cycles: 1

ANL A,#data

Operation: ANL
(A) ← (A) ∧ #data

Encoding:

0 1 0 1	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

ANL direct,A

Operation: ANL
(direct) ← (direct) ∧ (A)

Encoding:

0 1 0 1	0 0 1 0	direct address

Bytes: 2

Cycles: 1

ANL direct,#data

Operation: ANL
(direct) ← (direct) ∧ #data

Encoding:

0 1 0 1	0 0 1 1	direct address	immediate data

Bytes: 3

Cycles: 2

ANL C, <src-bit>

Function: Logical-AND for bit variables

Description: If the Boolean value of the source bit is a logical 0 then clear the carry flag; otherwise leave the carry flag in its current state. A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, *but the source bit itself is not affected.* No other flags are affected.

Only direct addressing is allowed for the source operand.

Example: Set the carry flag if, and only if, P1.0 = 1, ACC. 7 = 1, and OV = 0:

```
MOV   C,P1.0     ;Load carry with input pin state
ANL   C,ACC.7    ;AND carry with accumulator bit 7
ANL   C,/OV      ;AND with inverse of overflow flag
```

ANL C,bit

Operation: ANL
(C) ← (C) ∧ (bit)

Encoding:

1 0 0 0	0 0 1 0	bit address

Bytes: 2

Cycles: 2

ANL C,/bit

Operation: ANL
(C) ← (C) ∧ ¬ (bit)

Encoding:

1 0 1 1	0 0 0 0	bit address

Bytes: 2

Cycles: 2

CJNE <dest-byte>, <src-byte>, rel

Function: Compare and Jump if Not Equal

Description: CJNE compares the magnitudes of the first two operands, and branches if their values are not equal. The branch destination is computed by adding the signed relative displacement in the last instruction byte to the PC, after incrementing the PC to the start of the next instruction. The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte>; otherwise, the carry is cleared. Neither operand is affected.

The first two operands allow four addressing mode combinations: the accumulator may be compared with any directly addressed byte or immediate data, and any indirect RAM location or working register can be compared with an immediate constant.

Example: The accumulator contains 34H. Register 7 contains 56H. The first instruction in the sequence,

```
                CJNE   R7,#60H, NOT_EQ
;               ...    .....                ;  R7 = 60H
NOT_EQ:         JC     REQ_LOW              ;  IF R7 < 60H
;               ...    .....                ;  R7 > 60H
```

sets the carry flag and branches to the instruction at label NOT_EQ. By testing the carry flag, this instruction determines whether R7 is greater or less than 60H.

If the data being presented to Port 1 is also 34H, then the instruction,

```
WAIT:   CJNE   A,P1,WAIT
```

clears the carry flag and continues with the next instruction in sequence, since the accumulator does equal the data read from P1. (If some other value was being input on P1, the program will loop at this point until the P1 data changes to 34H.)

5

CJNE A,direct,rel

Operation:

```
(PC) ← (PC) + 3
if (A) <> (direct)
then (PC) ← (PC) + relative offset

if (A) < (direct)
then (C) ← 1
else (C) ← 0
```

Encoding:

1 0 1 1	0 1 0 1	direct address	rel. address

Bytes: 3

Cycles: 2

CJNE A,#data,rel

Operation:

```
(PC) ← (PC) + 3
if (A) <> data
then (PC) ← (PC) + relative offset

if (A) < data
then (C) ← 1
else (C) ← 0
```

Encoding:

1 0 1 1	0 1 0 0	immediate data	rel. address

Bytes: 3

Cycles: 2

CJNE Rn,#data,rel

Operation:

```
(PC) ← (PC) + 3
if (Rn) <> data
then (PC) ← (PC) + relative offset

if (Rn) < data
then (C) ← 1
else (C) ← 0
```

Encoding:

1 0 1 1	1 r r r	immediate data	rel. address

Bytes: 3

Cycles: 2

CJNE @Ri,#data,rel

Operation:

```
(PC) ← (PC) + 3
if ((Ri)) <> data
then (PC) ← (PC) + relative offset

if ((Ri)) < data
then (C) ← 1
else (C) ← 0
```

Encoding:

1 0 1 1	0 1 1 i	immediate data	rel. address

Bytes: 3

Cycles: 2

CLR A

Function: Clear accumulator

Description: The accumulator is cleared (all bits set to zero). No flags are affected.

Example: The accumulator contains 5CH (01011100B). The instruction,

CLR A

will leave the accumulator set to 00H (00000000B).

Operation: CLR
(A) ← 0

Encoding:

1 1 1 0	0 1 0 0

Bytes: 1

Cycles: 1

CLR bit

Function: Clear bit

Description: The indicated bit is cleared (reset to zero). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.

Example: Port 1 has previously been written with 5DH (01011101B). The instruction,

CLR P1.2

will leave the port set to 59H (01011001B).

CLR C

Operation: CLR
(C) ← 0

Encoding:

1 1 0 0	0 0 1 1

Bytes: 1

Cycles: 1

CLR bit

Operation: CLR
(bit) ← 0

Encoding:

1 1 0 0	0 0 1 0	bit address

Bytes: 2

Cycles: 1

CPL A

Function: Complement accumulator

Description: Each bit of the accumulator is logically complemented (one's complement). Bits which previously contained a one are changed to zero and vice versa. No flags are affected.

Example: The accumulator contains 5CH (01011100B). The instruction,

```
CPL   A
```

will leave the accumulator set to 0A3H (10100011B).

Operation: CPL
$(A) \leftarrow \neg (A)$

Encoding:

1 1 1 1	0 1 0 0

Bytes: 1

Cycles: 1

CPL bit

Function: Complement bit

Description: The bit variable specified is complemented. A bit which had been a one is changed to zero and vice versa. No other flags are affected. CLR can operate on the carry or any directly addressable bit.

Note:

When this instruction is used to modify an output pin, the value used as the original data will be read from the output data latch, *not* the input pin.

Example: Port 1 has previously been written with 5DH (01011101B). The instruction sequence,

```
CPL   P1.1
CPL   P1.2
```

will leave the port set to 5BH (01011011B).

CPL C

Operation: CPL
$(C) \leftarrow \neg (C)$

Encoding:

1 0 1 1	0 0 1 1

Bytes: 1

Cycles: 1

CPL bit

Operation: CPL
$(bit) \leftarrow \neg (bit)$

Encoding:

1 0 1 1	0 0 1 0	bit address

Bytes: 2

Cycles: 1

DA A

Function: Decimal adjust accumulator for addition

Description: DA A adjusts the eight-bit value in the accumulator resulting from the earlier addition of two variables (each in packed-BCD format), producing two four-bit digits. Any ADD or ADDC instruction may have been used to perform the addition.

If accumulator bits 3–0 are greater than nine (xxxx1010–xxxx1111), or if the AC flag is one, six is added to the accumulator producing the proper BCD digit in the low-order nibble. This internal addition would set the carry flag if a carry-out of the low-order four-bit field propagated through all high-order bits, but it would not clear the carry flag otherwise.

If the carry flag is now set, or if the four high-order bits now exceed nine (1010xxxx–1111xxxx), these high-order bits are incremented by six, producing the proper BCD digit in the high-order nibble. Again, this would set the carry flag if there was a carry-out of the high-order bits, but wouldn't clear the carry. The carry flag thus indicates if the sum of the original two BCD variables is greater than 100, allowing multiple precision decimal addition. OV is not affected.

All of this occurs during the one instruction cycle. Essentially, this instruction performs the decimal conversion by adding 00H, 06H, 60H, or 66H to the accumulator, depending on initial accumulator and PSW conditions.

Note:

DA A *cannot* simply convert a hexadecimal number in the accumulator to BCD notation, nor does DA A apply to decimal subtraction.

Example: The accumulator holds the value 56H (01010110B) representing the packed BCD digits of the decimal number 56. Register 3 contains the value 67H (01100111B) representing the packed BCD digits of the decimal number 67. The carry flag is set. The instruction sequence,

```
ADDC   A,R3
DA     A
```

will first perform a standard two's-complement binary addition, resulting in the value 0BEH (10111110) in the accumulator. The carry and auxiliary carry flags will be cleared.

The decimal adjust instruction will then alter the accumulator to the value 24H (00100100B), indicating the packed BCD digits of the decimal number 24, the low-order two digits of the decimal sum of 56, 67, and the carry-in. The carry flag will be set by the decimal adjust instruction, indicating that a decimal overflow occurred. The true sum 56, 67, and 1 is 124.

BCD variables can be incremented or decremented by adding 01H or 99H. If the Accumulator initially holds 30H (representing the digits of 30 decimal), then the instruction sequence,

```
ADD    A,#99H
DA     A
```

will leave the carry set and 29H in the accumulator, since 30 + 99 = 129. The low-order byte of the sum can be interpreted to mean 30 − 1 = 29.

Operation:

```
DA
contents of accumulator are BCD
if [[(A3-0) > 9] V [(AC) = 1]]
then (A3-0) <- (A3-0) + 6
and
if [[(A7-4) > 9] V [(C) = 1]]
then (A7-4) <- (A7-4) + 6
```

Encoding:

1 1 0 1	0 1 0 0

Bytes: 1

Cycles: 1

DEC byte

Function: Decrement

Description: The variable indicated is decremented by 1. An original value of 00H will underflow to 0FFH. No flags are affected. Four operand addressing modes are allowed: accumulator, register, direct, or register-indirect.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Register 0 contains 7FH (01111111B). Internal RAM locations 7EH and 7FH contain 00H and 40H, respectively. The instruction sequence,

```
DEC   @R0
DEC   R0
DEC   @R0
```

will leave register 0 set to 7EH and internal RAM locations 7EH and 7FH set to 0FFH and 3FH.

DEC A

Operation:

```
DEC
(A) <- (A) - 1
```

Encoding:

0 0 0 1	0 1 0 0

Bytes: 1

Cycles: 1

DEC Rn

Operation: DEC
(Rn) ← (Rn) − 1

Encoding:

0 0 0 1	1 r r r

Bytes: 1

Cycles: 1

DEC direct

Operation: DEC
(direct) ← (direct) − 1

Encoding:

0 0 0 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

DEC @Ri

Operation: DEC
((Ri)) ← ((Ri)) − 1

Encoding:

0 0 0 1	0 1 1 i

Bytes: 1

Cycles: 1

DIV AB

Function: Divide

Description: DIV AB divides the unsigned eight-bit integer in the accumulator by the unsigned eight-bit integer in register B. The accumulator receives the integer part of the quotient; register B receives the integer remainder. The carry and OV flags will be cleared.

Exception: If B had originally contained 00H, the values returned in the accumulator and B register will be undefined and the overflow flag will be set. The carry flag is cleared in any case.

Example: The accumulator contains 251 (0FBH or 11111011B) and B contains 18 (12H or 00010010B). The instruction,

DIV AB

will leave 13 in the accumulator (0DH or 00001101B) and the value 17 (11H or 00010001B) in B, since 251 = (13 x 18) + 17. Carry and OV will both be cleared.

Operation: DIV
(A15–8)
(B7-0) ← (A)/(B)

Encoding:

1 0 0 0	0 1 0 0

Bytes: 1

Cycles: 4

DJNZ <byte>,<rel-addr>

Function: Decrement and Jump if not zero

Description: DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00H will underflow to 0FFH. No flags are affected. The branch destination would be computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Internal RAM locations 40H, 50H, and 60H contain the values, 01H, 70H, and 15H, respectively. The instruction sequence,

```
DJNZ  40H,LABEL_1
DJNZ  50H,LABEL_2
DJNZ  60H,LABEL_3
```

will cause a jump to the instruction at label LABEL_2 with the values 00H, 6FH, and 15H in the three RAM locations. The first jump was *not* taken because the result was zero.

This instruction provides a simple way of executing a program loop a given number of times, or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The instruction sequence,

```
         MOV     R2, #8
TOGGLE:  CPL     P1.7
         DJNZ    R2,TOGGLE
```

will toggle P1.7 eight times, causing four output pulses to appear at bit 7 of output port 1. Each pulse will last three machine cycles; two for DJNZ and one to alter the pin.

DJNZ Rn,rel

Operation: DJNZ
(PC) ← (PC) + 2
(Rn) ← (Rn) − 1
if (Rn) > 0 or (Rn) < 0
then (PC) ← (PC) + rel

Encoding:

1 1 0 1	1 r r r	rel. address

Bytes: 2

Cycles: 2

DJNZ direct,rel

Operation: DJNZ
(PC) ← (PC) + 2
(direct) ← (direct) − 1
if (direct) > 0 or (direct) < 0
then (PC) ← (PC) + rel

Encoding:

1 1 0 1	0 1 0 1	direct address	rel. address

Bytes: 3

Cycles: 2

INC <byte>

Function: Increment

Description: INC increments the indicated variable by 1. An original value of 0FFH will overflow to 00H. No flags are affected. Three addressing modes are allowed: register, direct, or register-indirect.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Register 0 contains 7EH (011111110B). Internal RAM locations 7EH and 7FH contain 0FFH and 40H, respectively. The instruction sequence,

```
INC   @R0
INC   R0
INC   @R0
```

will leave register 0 set to 7FH and internal RAM locations 7EH and 7FH holding (respectively) 00H and 41H.

INC A

Operation: INC
(A) ← (A) + 1

Encoding:

0 0 0 0	0 1 0 0

Bytes: 1

Cycles: 1

INC Rn

Operation: INC
(Rn) ← (Rn) + 1

Encoding:

0 0 0 0	1 r r r

Bytes: 1

Cycles: 1

INC direct

Operation: INC
(direct) ← (direct) + 1

Encoding:

0 0 0 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

INC @Ri

Operation: INC
((Ri)) ← ((Ri)) + 1

Encoding:

0 0 0 0	0 1 1 i

Bytes: 1

Cycles: 1

INC DPTR

Function: Increment data pointer

Description: Increment the 16-bit data pointer by 1. A 16-bit increment (modulo 2^{16}) is performed; an overflow of the low-order byte of the data pointer (DPL) from 0FFH to 00H will increment the high-order byte (DPH). No flags are affected.

This is the only 16-bit register which can be incremented.

Example: Registers DPH and DPL contain 12H and 0FEH, respectively. The instruction sequence,

```
INC   DPTR
INC   DPTR
INC   DPTR
```

will change DPH and DPL to 13H and 01H.

Operation: INC
(DPTR) ← (DPTR) + 1

Encoding:

1 0 1 0	0 0 1 1

Bytes: 1

Cycles: 2

JB bit,rel

Function: Jump if bit set

Description: If the indicated bit is a one, jump to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The accumulator holds 56 (01010110B). The instruction sequence,

```
JB   P1.2,LABEL1
JB   ACC.2,LABEL2
```

will cause program execution to branch to the instruction at label LABEL2.

Operation:

```
JB
(PC) ← (PC) + 3
if (bit) = 1
then (PC) ← (PC) + rel
```

Encoding:

0 0 1 0	0 0 0 0	bit address	rel. address

Bytes: 3

Cycles: 2

JBC bit,rel

Function: Jump if bit is set and clear bit

Description: If the indicated bit is one, branch to the address indicated; otherwise proceed with the next instruction. *In either case, clear the designated bit.* The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

Note:

When this instruction is used to test an output pin, the value used as the original data will be read from the output data latch, *not* the input pin.

Example: The accumulator holds 56H (01010110B). The instruction sequence,

```
JBC   ACC.3,LABEL1
JBC   ACC.2,LABEL2
```

will cause program execution to continue at the instruction identified by the label LABEL2, with the accumulator modified to 52H (01010010B).

Operation:

```
JBC
(PC) ← (PC) + 3
if (bit) = 1
then (bit) ← 0
     (PC) ← (PC) + rel
```

Encoding:

0 0 0 1	0 0 0 0	bit address	rel. address

Bytes: 3

Cycles: 2

JC rel

Function: Jump if carry is set

Description: If the carry flag is set, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.

Example: The carry flag is cleared. The instruction sequence,

```
JC     LABEL1
CPL    C
JC     LABEL 2
```

will set the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Operation:

```
JC
(PC) ← (PC) + 2
if (C) = 1
then (PC) ← (PC) + rel
```

Encoding:

0 1 0 0	0 0 0 0		rel. address

Bytes: 2

Cycles: 2

JMP @A+DPTR

Function: Jump indirect

Description: Add the eight-bit unsigned contents of the accumulator with the sixteen-bit data pointer, and load the resulting sum to the program counter. This will be the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo 2^{16}): a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the accumulator nor the data pointer is altered. No flags are affected.

Example: An even number from 0 to 6 is in the Accumulator. The following sequence of instructions will branch to one of four AJMP instructions in a jump table starting at JMP_TBL:

```
            MOV     DPTR, #JMP_TBL
            JMP     @A+DPTR
JMP_TBL:    AJMP    LABEL0
            AJMP    LABEL1
            AJMP    LABEL2
            AJMP    LABEL3
```

If the accumulator equals 04H when starting this sequence, execution will jump to label LABEL2. Remember that AJMP is a two-byte instruction, so the jump instructions start at every other address.

Operation:

```
JMP
(PC) ← (A) + (DPTR)
```

Encoding:

0 1 1 1	0 0 1 1

Bytes: 1

Cycles: 2

JNB bit,rel

Function: Jump if bit not set

Description: If the indicated bit is a zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The accumulator holds 56H (01010110B). The instruction sequence,

```
JNB   P1.3,LABEL1
JNB   ACC.3,LABEL2
```

will cause program execution to continue at the instruction at label LABEL2.

Operation:

```
JNB
(PC) ← (PC) + 3
if (bit) = 0
then (PC) ← (PC) + rel
```

Encoding:

0 0 1 1	0 0 0 0	bit address	rel. address

Bytes: 3

Cycles: 2

JNC rel

Function: Jump if carry is not set

Description: If the carry flag is a zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.

Example: The carry flag is set. The instruction sequence,

```
JNC   LABEL1
CPL   C
JNC   LABEL2
```

will clear the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Operation:

```
JNC
(PC) ← (PC) + 2
if (C) = 0
then (PC) ← (PC) + rel
```

Encoding:

0 1 0 1	0 0 0 0	rel. address

Bytes: 2

Cycles: 2

JNZ rel

Function: Jump if accumulator not zero

Description: If any bit of the accumulator is a one, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The accumulator is not modified. No flags are affected.

Example: The accumulator originally holds 00H. The instruction sequence,

```
JNZ   LABEL1
INC   A
JNZ   LABEL2
```

will set the accumulator to 01H and continue at label LABEL2.

Operation:

```
JNZ
(PC) ← (PC) + 2
if (A) ≠ 0
then (PC) ← (PC) + rel
```

Encoding:

0 1 1 1	0 0 0 0		rel. address

Bytes: 2

Cycles: 2

JZ rel

Function: Jump if accumulator zero

Description: If all bits of the accumulator are zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The accumulator is not modified. No flags are affected.

Example: The accumulator originally contains 01H. The instruction sequence,

```
JZ    LABEL1
DEC   A
JZ    LABEL2
```

will change the Accumulator to 00H and cause program execution to continue at the instruction identified by the label LABEL2.

Operation:

```
JZ
(PC) ← (PC) + 2
if (A) = 0
then (PC) ← (PC) + rel
```

Encoding:

0 1 1 0	0 0 0 0		rel. address

Bytes: 2

Cycles: 2

LCALL addr16

Function: Long call

Description: LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the stack pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64 kbyte program memory address space. No flags are affected.

Example: Initially the stack pointer equals 07H. The label "SUBRTN" is assigned to program memory location 1234H. After executing the instruction,

LCALL SUBRTN

at location 0123H, the stack pointer will contain 09H, internal RAM locations 08H and 09H will contain 26H and 01H, and the PC will contain 1234H.

Operation:

```
LCALL
(PC) ← (PC) + 3
(SP) ← (SP) + 1
((SP)) ← (PC7-0)
(SP) ← (SP) + 1
((SP)) ← (PC15-8)
(PC) ← addr15-0
```

Encoding:

0 0 0 1	0 0 1 0	addr15...addr8	addr7...addr0

Bytes: 3

Cycles: 2

LJMP addr16

Function: Long Jump

Description: LJMP causes an unconditional branch to the indicated address, by loading the high-order and low-order bytes of the PC (respectively) with the second and third instruction bytes. The destination may therefore be anywhere in the full 64K program memory address space. No flags are affected.

Example: The label "JMPADR" is assigned to the instruction at program memory location 1234H. The instruction,

LJMP JMPADR

at location 0123H will load the program counter with 1234H.

Operation:

```
LJMP
(PC) ← addr15-0
```

Encoding:

0 0 0 0	0 0 1 0	addr15...addr8	addr7...addr0

Bytes: 3

Cycles: 2

MOV <dest-byte>,<src-byte>

Function: Move byte variable

Description: The byte variable indicated by the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

This is by far the most flexible operation. Fifteen combinations of source and destination addressing modes are allowed.

Example: Internal RAM location 30H holds 40H. The value of RAM location 40H is 10H. The data present at input port 1 is 11001010B (0CAH).

```
MOV  R0,#30H     ;R0 <= 30H
MOV  A,@R0       ;A <= 40H
MOV  R1,A        ;R1 <= 40H
MOV  R,@R1       ;B <= 10H
MOV  @R1,P1      ;RAM (40H) <= 0CAH
MOV  P2,P1       ;P2 <0CAH
```

leaves the value 30H in register 0, 40H in both the accumulator and register 1, 10H in register B, and 0CAH (11001010B) both in RAM location 40H and output on port 2.

MOV A,Rn

Operation: MOV
(A) ← (Rn)

Encoding:

1 1 1 0	1 r r r

Bytes: 1

Cycles: 1

MOV A, direct*

Operation: MOV
(A) ← (direct)

Encoding:

1 1 1 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

*MOV A,ACC is not a valid instruction.

MOV A,@Ri

Operation: MOV
(A) ← ((Ri))

Encoding:

1 1 1 0	0 1 1 i

Bytes: 1

Cycles: 1

MOV A,#data

Operation: MOV
(A) ← #data

Encoding:

0 1 1 1	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

MOV Rn,A

Operation: MOV
(Rn) ← (A)

Encoding:

1 1 1 1	1 r r r

Bytes: 1

Cycles: 1

MOV Rn,direct

Operation: MOV
(Rn) ← (direct)

Encoding:

1 0 1 0	1 r r r	direct address

Bytes: 2

Cycles: 2

MOV Rn,#data

Operation: MOV
(Rn) ← #data

Encoding:

0 1 1 1	1 r r r	immediate data

Bytes: 2

Cycles: 1

MOV direct,A

Operation: MOV
(direct) ← (A)

Encoding:

1 1 1 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

MOV direct,Rn

Operation: MOV
(direct) ← (Rn)

Encoding:

1 0 0 0	1 r r r	direct address

Bytes: 2

Cycles: 2

MOV direct,direct

Operation: MOV
(direct) ← (direct)

Encoding:

1 0 0 0	0 1 0 1	dir. addr. (src)	dir. addr. (dest)

Bytes: 3

Cycles: 2

MOV direct,@Ri

Operation: MOV
(direct) ← ((Ri))

Encoding:

1 0 0 0	0 1 1 i	direct address

Bytes: 2

Cycles: 2

MOV direct,#data

Operation: MOV
(direct) ← #data

Encoding:

0 1 1 1	0 1 0 1	direct address	immediate data

Bytes: 3

Cycles: 2

MOV @Ri,A

Operation: MOV
((Ri)) ← (A)

Encoding:

1 1 1 1	0 1 1 i

Bytes: 1

Cycles: 1

MOV @Ri,direct

Operation: MOV
((Ri)) ← (direct)

Encoding:

1 0 1 0	0 1 1 i	direct addr.

Bytes: 2

Cycles: 2

MOV @Ri,#data

Operation: MOV
((Ri)) ← #data

Encoding:

0 1 1 1	0 1 1 i	immediate data

Bytes: 2

Cycles: 1

MOV <dest-bit>,<src-bit>

Function: Move bit data

Description: The Boolean variable indicated by the second operand is copied into the location specified by the first operand. One of the operands must be the carry flag; the other may be any directly addressable bit. No other register or flag is affected.

Example: The carry flag is originally set. The data present at input port 3 is 11000101B. The data previously written to output port 1 is 35H (00110101B).

```
MOV   P1.3,C
MOV   C,P3.3
MOV   P1.2,C
```

will leave the carry cleared and change port 1 to 39H (00111001B).

MOV C,bit

Operation: MOV
(C) ← (bit)

Encoding:

1 0 1 0	0 0 1 0	bit address

Bytes: 2

Cycles: 1

MOV bit,C

Operation: MOV
(bit) ← (C)

Encoding:

1 0 0 1	0 0 1 0	bit address

Bytes: 2

Cycles: 2

MOV DPTR,#data16

Function: Load data pointer with a 16-bit constant

Description: The data pointer is loaded with the 16-bit constant indicated. The 16-bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the low-order byte. No flags are affected.

This is the only instruction which moves 16 bits of data at once.

Example: The instruction,

```
MOV    DPTR,#1234H
```

will load the value 1234H into the data pointer: DPH will hold 12H and DPL will hold 34H.

Operation: MOV
(DPTR) ← #data15-0
DPH □ DPL ← #data15-8 □ #data7-0

Encoding:

1 0 0 1	0 0 0 0	immed. data15-8	immed. data7-0

Bytes: 3

Cycles: 2

MOVC A,@A+<base-reg>

Function: Move code byte

Description: The MOVC instructions load the accumulator with a code byte, or constant from program memory. The address of the byte fetched is the sum of the original unsigned eight-bit accumulator contents and the contents of a 16-bit base register, which may be either the data pointer or the PC. In the latter case, the PC is incremented to the address of the following instruction before being added to the accumulator; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example: A value between 0 and 3 is in the accumulator. The following instructions will translate the value in the accumulator to one of four values defined by the DB (define byte) directive.

```
REL_PC:  INC     A
         MOVC    A,@A+PC
         RET
         DB      66H
         DB      77H
         DB      88H
         DB      99H
```

If the subroutine is called with the accumulator equal to 01H, it will return with 77H in the accumulator. The INC A before the MOVC instruction is needed to "get around" the RET instruction above the table. If several bytes of code separated the MOVC from the table, the corresponding number would be added to the accumulator instead.

MOVC A,@A+DPTR

Operation: MOVC
(A) ← ((A) + (DPTR))

Encoding:

1 0 0 1	0 0 1 1

Bytes: 1

Cycles: 2

MOVC A,@A + PC

Operation: MOVC
(PC) ← (PC) + 1
(A) ← ((A) + (PC))

Encoding:

1 0 0 0	0 0 1 1

Bytes: 1

Cycles: 2

MOVX <dest-byte>,<src-byte>

Function: Move external

Description: The MOVX instructions transfer data between the accumulator and a byte of external data memory, hence the "X" appended to MOV. There are two types of instructions, differing in whether they provide an eight-bit or 16-bit indirect address to the external data RAM.

In the first type, the contents of R0 or R1 in the current register bank provide an eight-bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or for a relatively small RAM array. For somewhat larger arrays, any output port pins can be used to output higher-order address bits. These pins would be controlled by an output instruction preceding the MOVX.

In the second type of MOVX instruction, the data pointer generates a 16-bit address. P2 outputs the high-order eight address bits (the contents of DPH) while P0 multiplexes the low-order eight bits (DPL) with data. The P2 special function register retains its previous contents while the P2 output buffers are emitting the contents of DPH. This form is faster and more efficient when accessing very large data arrays (up to 64 kbytes), since no additional instructions are needed to set up the output ports.

It is possible in some situations to mix the two MOVX types. A large RAM array with its high-order address lines driven by P2 can be addressed via the data pointer, or with code to output high-order address bits to P2 followed by a MOVX instruction using R0 or R1.

Example: An external 256 byte RAM using multiplexed address/data lines (e.g., an SAB 8155 RAM/I/O/Timer) is connected to the SAB 80515 port 0. Port 3 provides control lines for the external RAM. Ports 1 and 2 are used for normal I/O. Registers 0 and 1 contain 12H and 34H. Location 34H of the external RAM holds the value 56H. The instruction sequence,

```
MOVX   A,@R1
MOVX   @R0,A
```

copies the value 56H into both the accumulator and external RAM location 12H.

MOVX A,@Ri

Operation: MOVX
(A) ← ((Ri))

Encoding:

1 1 1 0	0 0 1 i

Bytes: 1

Cycles: 2

MOVX A,@DPTR

Operation: MOVX
(A) ← ((DPTR))

Encoding:

1 1 1 0	0 0 0 0

Bytes: 1

Cycles: 2

MOVX @Ri,A

Operation: MOVX
((Ri)) ← (A)

Encoding:

1 1 1 1	0 0 1 i

Bytes: 1

Cycles: 2

MOVX @DPTR,A

Operation: MOVX
((DPTR)) ← (A)

Encoding:

1 1 1 1	0 0 0 0

Bytes: 1

Cycles: 2

MUL AB

Function: Multiply

Description: MUL AB multiplies the unsigned eight-bit integers in the accumulator and register B. The low-order byte of the 16-bit product is left in the accumulator, and the high-order byte in B. If the product is greater than 255 (0FFH) the overflow flag is set; otherwise it is cleared. The carry flag is always cleared.

Example: Originally the accumulator holds the value 80 (50H). Register B holds the value 160 (0A0H). The instruction,

MUL AB

will give the product 12,800 (3200H), so B is changed to 32H (00110010B) and the accumulator is cleared. The overflow flag is set, carry is cleared.

Operation: MUL
(A7–0) ← (A) X (B)
(B15–8)

Encoding:

1 0 1 0	0 1 0 0

Bytes: 1

Cycles: 4

NOP

Function: No operation

Description: Execution continues at the following instruction. Other than the PC, no registers or flags are affected.

Example: It is desired to produce a low-going output pulse on bit 7 of port 2 lasting exactly 5 cycles. A simple SETB/CLR sequence would generate a one-cycle pulse, so four additional cycles must be inserted. This may be done (assuming no interrupts are enabled) with the instruction sequence,

```
CLR     P2.7
NOP
NOP
NOP
NOP
SETB    P2.7
```

Operation: NOP

Encoding:

0 0 0 0	0 0 0 0

Bytes: 1

Cycles: 1

ORL <dest-byte> <src-byte>

Function: Logical-OR for byte variables

Description: ORL performs the bitwise logical-OR operation between the indicated variables, storing the results in the destination byte. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the accumulator or immediate data.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: If the accumulator holds 0C3H (11000011B) and R0 holds 55H (01010101B) then the instruction,

```
ORL   A,R0
```

will leave the accumulator holding the value 0D7H (11010111B).

When the destination is a directly addressed byte, the instruction can set combinations of bits in any RAM location or hardware register. The pattern of bits to be set is determined by a mask byte, which may be either a constant data value in the instruction or a variable computed in the accumulator at run-time. The instruction,

```
ORL   P1,#00110010B
```

will set bits 5, 4, and 1 of output port 1.

ORL A,Rn

Operation: ORL
(A) ← (A) V (Rn)

Encoding:

0 1 0 0	1 r r r

Bytes: 1

Cycles: 1

ORL A,direct

Operation: ORL
(A) ← (A) V (direct)

Encoding:

0 1 0 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ORL A,@Ri

Operation: ORL
(A) ← (A) V ((Ri))

Encoding:

0 1 0 0	0 1 1 i

Bytes: 1

Cycles: 1

ORL A,#data

Operation: ORL
(A) ← (A) V #data

Encoding:

0 1 0 0	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

ORL direct,A

Operation: ORL
(direct) ← (direct) V (A)

Encoding:

0 1 0 0	0 0 1 0	direct address

Bytes: 2

Cycles: 1

ORL direct,#data

Operation: ORL
(direct) ← (direct) V #data

Encoding:

0 1 0 0	0 0 1 1	direct address	immediate data

Bytes: 3

Cycles: 2

ORL C,<src-bit>

Function: Logical-OR for bit variables

Description: Set the carry flag if the Boolean value is a logical 1; leave the carry in its current state otherwise . A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Example: Set the carry flag if, and only if, P1.0 = 1, ACC. 7 = 1, or OV = 0:

```
MOV   C,P1.0     ;Load carry with input pin p10
ORL   C,ACC.7    ;OR carry with the accumulator bit 7
ORL   C,/OV      ;OR carry with the inverse of OV.
```

ORL C,bit

Operation: ORL
$(C) \leftarrow (C) \vee (bit)$

Encoding:

0 1 1 1	0 0 1 0		bit address

Bytes: 2

Cycles: 2

ORL C,/bit

Operation: ORL
$(C) \leftarrow (C) \vee \neg (bit)$

Encoding:

1 0 1 0	0 0 0 0		bit address

Bytes: 2

Cycles: 2

POP direct

Function: Pop from stack

Description: The contents of the internal RAM location addressed by the stack pointer is read, and the stack pointer is decremented by one. The value read is then transferred to the directly addressed byte indicated. No flags are affected.

Example: The stack pointer originally contains the value 32H, and internal RAM locations 30H through 32H contain the values 20H, 23H, and 01H, respectively. The instruction sequence,

```
POP   DPH
POP   DPL
```

will leave the stack pointer equal to the value 30H and the data pointer set to 0123H. At this point the instruction,

```
POP   SP
```

will leave the stack pointer set to 20H. Note that in this special case the stack pointer was decremented to 2FH before being loaded with the value popped (20H).

Operation: POP
(direct) ← ((SP))
(SP) ← (SP) − 1

Encoding:

1 1 0 1	0 0 0 0		direct address

Bytes: 2

Cycles: 2

PUSH direct

Function: Push onto stack

Description: The stack pointer is incremented by one. The contents of the indicated variable is then copied into the internal RAM location addressed by the stack pointer. Otherwise no flags are affected.

Example: On entering an interrupt routine the stack pointer contains 09H. The data pointer holds the value 0123H. The instruction sequence,

```
PUSH   DPL
PUSH   DPH
```

will leave the stack pointer set to 0BH and store 23H and 01H in internal RAM locations 0AH and 0BH, respectively.

Operation: PUSH
(SP) ← (SP) + 1
((SP)) ← (direct)

Encoding:

1 1 0 0	0 0 0 0		direct address

Bytes: 2

Cycles: 2

RET

Function: Return from subroutine

Description: RET pops the high- and low-order bytes of the PC successively from the stack, decrementing the stack pointer by two. Program execution continues at the resulting address, generally the instruction immediately following an ACALL or LCALL. No flags are affected.

Example: The stack pointer originally contains the value 0BH. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

RET

will leave the stack pointer equal to the value 09H. Program execution will continue at location 0123H.

Operation: RET
(PC15-8) ← ((SP))
(SP) ← (SP) − 1
(PC7-0) ← ((SP))
(SP) ← (SP) − 1

Encoding:

0 0 1 0	0 0 1 0

Bytes: 1

Cycles: 2

RETI

Function: Return from interrupt

Description: RETI pops the high- and low-order bytes of the PC successively from the stack, and restores the interrupt logic to accept additional interrupts at the same priority level as the one just processed. The stack pointer is left decremented by two. No other registers are affected; the PSW is *not* automatically restored to its pre-interrupt status. Program execution continues at the resulting address, which is generally the instruction immediately after the point at which the interrupt request was detected. If a lower- or same-level interrupt had been pending when the RETI instruction is executed, that one instruction will be executed before the pending interrupt is processed.

Example: The stack pointer originally contains the value 0BH. An interrupt was detected during the instruction ending at location 0122H. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

RETI

will leave the stack pointer equal to 09H and return program execution to location 0123H.

Operation: RETI
(PC15-8) ← ((SP))
(SP) ← (SP) − 1
(PC7-0) ← ((SP))
(SP) ← (SP) − 1

Encoding:

0 0 1 1	0 0 1 0

Bytes: 1

Cycles: 2

RL A

Function: Rotate accumulator left

Description: The eight bits in the accumulator are rotated one bit to the left. Bit 7 is rotated into the bit 0 position. No flags are affected.

Example: The accumulator holds the value 0C5H (11000101B). The instruction,

RL A

leaves the accumulator holding the value 8BH (10001011B) with the carry unaffected.

Operation: RL
(An + 1) ← (An) n = 0 – 6
(A0) ← (A7)

Encoding:

0 0 1 0	0 0 1 1

Bytes: 1

Cycles: 1

RLC A

Function: Rotate accumulator Left through carry flag

Description: The eight bits in the accumulator and the carry flag are together rotated one bit to the left. Bit 7 moves into the carry flag; the original state of the carry flag moves into the bit 0 position. No other flags are affected.

Example: The accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction,

RLC A

leaves the accumulator holding the value 8AH (10001010B) with the carry set.

Operation: RLC
(An + 1) ← (An) n = 0 – 6
(A0) ← (C)
(C) ← (A7)

Encoding:

0 0 1 1	0 0 1 1

Bytes: 1

Cycles: 1

RR A

Function: Rotate accumulator right

Description: The eight bits in the accumulator are rotated one bit to the right. Bit 0 is rotated into the bit 7 position. No flags are affected.

Example: The accumulator holds the value 0C5H (11000101B). The instruction,

```
RR   A
```

leaves the accumulator holding the value 0E2H (11100010B) with the carry unaffected.

Operation:

```
RR
(An) ← (An + 1)   n = 0 − 6
(A7) ← (A0)
```

Encoding:

0 0 0 0	0 0 1 1

Bytes: 1

Cycles: 1

RRC A

Function: Rotate accumulator right through carry flag

Description: The eight bits in the accumulator and the carry flag are together rotated one bit to the right. Bit 0 moves into the carry flag; the original value of the carry flag moves into the bit 7 position. No other flags are affected.

Example: The accumulator holds the value 0C5H (11000101B), the carry is zero. The instruction,

```
RRC   A
```

leaves the accumulator holding the value 62 (01100010B) with the carry set.

Operation:

```
RRC
(An) ← (An + 1)   n = 0 − 6
(A7) ← (C)
(C) ← (A0)
```

Encoding:

0 0 0 1	0 0 1 1

Bytes: 1

Cycles: 1

SETB <bit>

Function: Set bit

Description: SETB sets the indicated bit to one. SETB can operate on the carry flag or any directly addressable bit. No other flags are affected.

Example: The carry flag is cleared. Output port 1 has been written with the value 34H (00110100B). The instructions,

```
SETB   C
SETB   P1.0
```

will leave the carry flag set to 1 and change the data output on port 1 to 35H (00110101B).

SETB C

Operation: SETB
(C) ← 1

Encoding:

1 1 0 1	0 0 1 1

Bytes: 1

Cycles: 1

SETB bit

Operation: SETB
(bit) ← 1

Encoding:

1 1 0 1	0 0 1 0	bit address

Bytes: 2

Cycles: 1

SJMP rel

Function: Short jump

Description: Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction to 127 bytes following it.

Example: The label "RELADR" is assigned to an instruction at program memory location 0123H. The instruction,

SJMP RELADR

will assemble into location 0100H. After the instruction is executed, the PC will contain the value 0123H.

Note:

Under the above conditions the instruction following SJMP will be at 102H. Therefore, the displacement byte of the instruction will be the relative offset (0123H−0102H) = 21H. Put another way, an SJMP with a displacement of 0FEH would be a one-instruction infinite loop.

Operation: SJMP
(PC) ← (PC) + 2
(PC) ← (PC) + rel

Encoding:

1 0 0 0	0 0 0 0		rel. address

Bytes: 2

Cycles: 2

SUBB A,<src-byte>

Function: Subtract with borrow

Description: SUBB subtracts the indicated variable and the carry flag together from the accumulator, leaving the result in the accumulator. SUBB sets the carry (borrow) flag if a borrow is needed for bit 7, and clears C otherwise. (If C was set *before* executing a SUBB instruction, this indicates that a borrow was needed for the previous step in a multiple precision subtraction, so the carry is subtracted from the accumulator along with the source operand.) AC is set if a borrow is needed for bit 3, and cleared otherwise. OV is set if a borrow is needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.

When subtracting signed integers OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.

The source operand allows four addressing modes: register, direct, register-indirect, or immediate.

Example: The accumulator holds 0C9H (11001001B), register 2 holds 54H (01010100B), and the carry flag is set. The instruction,

SUBB A,R2

will leave the value 74H (01110100B) in the accumulator, with the carry flag and AC cleared but OV set.

Notice that 0C9H minus 54H is 75H. The difference between this and the above result is due to the (borrow) flag being set before the operation. If the state of the carry is not known before starting a single or multiple-precision subtraction, it should be explicitly cleared by a CLR C instruction.

SUBB A,Rn

Operation: SUBB
(A) ← (A) − (C) − (Rn)

Encoding:

1 0 0 1	1 r r r

Bytes: 1

Cycles: 1

SUBB A,direct

Operation: SUBB
(A) ← (A) − (C) − (direct)

Encoding:

1 0 0 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

SUBB A,@Ri

Operation: SUBB
(A) ← (A) − (C) − ((Ri))

Encoding:

1 0 0 1	0 1 1 i

Bytes: 1

Cycles: 1

SUBB A,#data

Operation: SUBB
(A) ← (A) − (C) − #data

Encoding:

1 0 0 1	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

SWAP A

Function: Swap nibbles within the accumulator

Description: SWAP A interchanges the low- and high-order nibbles (four-bit fields) of the accumulator (bits 3−0 and bits 7−4). The operation can also be thought of as a four-bit rotate instruction. No flags are affected.

Example: The accumulator holds the value 0C5H (11000101B). The instruction,

SWAP A

leaves the accumulator holding the value 5CH (01011100B).

Operation: SWAP
(A3−0) ⟷ (A7−4)

Encoding:

1 1 0 0	0 1 0 0

Bytes: 1

Cycles: 1

XCH A,<byte>

Function: Exchange accumulator with byte variable

Description: XCH loads the accumulator with the contents of the indicated variable, at the same time writing the original accumulator contents to the indicated variable. The source/destination operand can use register, direct, or register-indirect addressing.

Example: R0 contains the address 20H. The accumulator holds the value 3FH (00111111B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

```
XCH    A,@R0
```

will leave RAM location 20H holding the values 3FH (00111111B) and 75H (01110101B) in the accumulator.

XCH A,Rn

Operation: XCH
(A) ⇄ (Rn)

Encoding:

1 1 0 0	1 r r r

Bytes: 1

Cycles: 1

XCH A,direct

Operation: XCH
(A) ⇄ (direct)

Encoding:

1 1 0 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

XCH A,@Ri

Operation: XCH
(A) ⇄ ((Ri))

Encoding:

1 1 0 0	0 1 1 i

Bytes: 1

Cycles: 1

XCHD A,@Ri

Function: Exchange digit

Description: XCHD exchanges the low-order nibble of the accumulator (bits 3–0, generally representing a hexadecimal or BCD digit), with that of the internal RAM location indirectly addressed by the specified register. The high-order nibbles (bits 7–4) of each register are not affected. No flags are affected.

Example: R0 contains the address 20H. The accumulator holds the value 36H (00110110B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

```
XCHD   A,@R0
```

will leave RAM location 20H holding the value 76H (01110110B) and 35H (00110101B) in the accumulator.

Operation: XCHD
(A3–0) ⇄ ((Ri)3–0)

Encoding:

1 1 0 1	0 1 1 i

Bytes: 1

Cycles: 1

XRL <dest-byte>,<src-byte>

Function: Logical Exclusive-OR for byte variables

Description: XRL performs the bitwise logical Exclusive-OR operation between the indicated variables, storing the results in the destination. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the accumulator or immediate data.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.)

Example: If the accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) then the instruction,

XRL A,R0

will leave the accumulator holding the value 69H (01101001B).

When the destination is a directly addressed byte, this instruction can complement combinations of bits in any RAM location or hardware register. The pattern of bits to be complemented is then determined by a mask byte, either a constant contained in the instruction or a variable computed in the accumulator at run-time. The instruction,

XRL P1,#00110001B

will complement bits 5, 4, and 0 of output port 1.

XRL A,Rn

Operation: XRL
(A) ← (A) $\veebar$ (Rn)

Encoding:

0 1 1 0	1 r r r

Bytes: 1

Cycles: 1

XRL A,direct

Operation: XRL
(A) ← (A) $\veebar$ (direct)

Encoding:

0 1 1 0	0 1 0 1	direct address

XRL A,@Ri

Operation: XRL
(A) ← (A) ∀ ((Ri))

Encoding:

0 1 1 0	0 1 1 i

Bytes: 1

Cycles: 1

XRL A,#data

Operation: XRL
(A) ← (A) ∀ #data

Encoding:

0 1 1 0	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

XRL direct,A

Operation: XRL
(direct) ← (direct) ∀ (A)

Encoding:

0 1 1 0	0 0 1 0	direct address

Bytes: 2

Cycles: 1

XRL direct,#data

Operation: XRL
(direct) ← (direct) ∀ #data

Encoding:

0 1 1 0	0 0 1 1	direct address	immediate data

Bytes: 3

Cycles: 2

5.4 Instruction Set Summary

Mnemonic		Description	Byte	Cycle
Arithmetic Operations				
ADD	A, Rn	Add Register to Accumulator	1	1
ADD	A, Direct	Add Direct Byte to Accumulator	2	1
ADD	A, @Ri	Add Indirect RAM to Accumulator	1	1
ADD	A, #Data	Add Immediate Data to Accumulator	2	1
ADDC	A, Rn	Add Register to Accumulator with Carry Flag	1	1
ADDC	A, Direct	Add Direct Byte to A with Carry Flag	2	1
ADDC	A, @Ri	Add Indirect RAM to A with Carry Flag	1	1
ADDC	A, #Data	Add Immediate Data to A with Carry Flag	2	1
SUBB	A, Rn	Subtract Register from A with Borrow	1	1
SUBB	A, Direct	Subtract Direct Byte from A with Borrow	2	1
SUBB	A, @Ri	Subtract Indirect RAM from A with Borrow	1	1
SUBB	A, #Data	Subtract Immediate Data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment Register	1	1
INC	Direct	Increment Direct Byte	2	1
INC	@Ri	Increment Indirect RAM	1	1
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement Register	1	1
DEC	Direct	Decrement Direct Byte	2	1
DEC	@Ri	Decrement Indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUL	AB	Multiply A and B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
Logical Operations				
ANL	A, Rn	AND Register to Accumulator	1	1
ANL	A, Direct	AND Direct Byte to Accumulator	2	1
ANL	A, @Ri	AND Indirect RAM to Accumulator	1	1
ANL	A, #Data	AND Immediate Data to Accumulator	2	1
ANL	Direct, A	AND Accumulator to Direct Byte	2	1

5.4 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Cycle
Logical Operations (Continued)				
ANL	Direct, #Data	AND Immediate Data to Direct Byte	3	2
ORL	A, Rn	OR Register to Accumulator	1	1
ORL	A, Direct	OR Direct Byte to Accumulator	2	1
ORL	A, @Ri	OR Indirect RAM to Accumulator	1	1
ORL	A, #Data	OR Immediate Data to Accumulator	2	1
ORL	Direct, A	OR Accumulator to Direct Byte	2	1
ORL	Direct, #Data	OR Immediate Data to Direct Byte	3	2
XRL	A, Rn	Exclusive OR Register to Accumulator	1	1
XRL	A, Direct	Exclusive OR Direct Byte to Accumulator	2	1
XRL	A, @Ri	Exclusive OR Indirect RAM to Accumulator	1	1
XRL	A, #Data	Exclusive OR Immediate Data to Accumulator	2	1
XRL	Direct, A	Exclusive OR Accumulator to Direct Byte	2	1
XRL	Direct, #Data	Exclusive OR Immediate Data to Direct Byte	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through Carry Flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry Flag	1	1
SWAP	A	Swap Nibbles within the Accumulator	1	1
Data Transfer				
MOV	A, Rn	Move Register to Accumulator	1	1
MOV	A, Direct*	Move Direct Byte to Accumulator	2	1
MOV	A, @Ri	Move Indirect RAM to Accumulator	1	1
MOV	A, #Data	Move Immediate Data to Accumulator	2	1
MOV	Rn, A	Move Accumulator to Register	1	1
MOV	Rn, Direct	Move Direct Byte to Register	2	2
MOV	Rn, #Data	Move Immediate Data to Register	2	1
MOV	Direct, A	Move Accumulator to Direct Byte	2	1
MOV	Direct, Rn	Move Register to Direct Byte	2	2
MOV	Direct, Direct	Move Direct Byte to Direct Byte	3	2

*MOV A, ACC is not a valid instruction.

5.4 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Cycle
Data Transfer (Continued)				
MOV	Direct, @Ri	Move Indirect RAM to Direct Byte	2	2
MOV	Direct, #Data	Move Immediate Data to Direct Byte	3	2
MOV	@Ri, A	Move Accumulator to Indirect RAM	1	1
MOV	@Ri, Direct	Move Direct Byte to Indirect RAM	2	2
MOV	@Ri, #Data	Move Immediate Data to Indirect RAM	2	1
MOV	DPTR, #Data16	Load Data Pointer with a 16-Bit Constant	3	2
MOVC	A, @A+DPTR	Move Code Byte Relative to DPTR to Accumulator	1	2
MOVC	A, @A + PC	Move Code Byte Relative to PC to Accumulator	1	2
MOVX	A, @ Ri	Move External RAM (8-Bit Addr.) to Accumulator	1	2
MOVX	A, @DPTR	Move External RAM (16-bit Addr.) to Accumulator	1	2
MOVX	@Ri, A	Move A to External RAM (8-bit Addr.)	1	2
MOVX	@DPTR, A	Move A to External RAM (16-Bit Addr.)	1	2
PUSH	Direct	Push Direct Byte onto Stack	2	2
POP	Direct	Pop Direct Byte from Stack	2	2
XCH	A, Rn	Exchange Register with Accumulator	1	1
XCH	A, Direct	Exchange Direct Byte with Accumulator	2	1
XCH	A, @Ri	Exchange Indirect RAM with Accumulator	1	1
XCHD	A, @Ri	Exchange Low-Order Digit Indirect RAM with A	1	1
Boolean Variable Manipulation				
CLR	C	Clear Carry Flag	1	1
CLR	Bit	Clear Direct Bit	2	1
SETB	C	Set Carry Flag	1	1
SETB	Bit	Set Direct Bit	2	1
CPL	C	Complement Carry Flag	1	1
CPL	Bit	Complement Direct Bit	2	1
ANL	C, Bit	AND Direct Bit to Carry Flag	2	2
ANL	C, /Bit	AND Complement of Direct Bit to Carry	2	2
ORL	C, Bit	OR Direct Bit to Carry Flag	2	2
ORL	C, /Bit	OR Complement of Direct Bit to Carry	2	2
MOV	C, Bit	Move Direct Bit to Carry Flag	2	1
MOV	Bit, C	Move Carry Flag to Direct Bit	2	2

5.4 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Cycle
Program and Machine Control				
ACALL	Addr 11	Absolute Subroutine Call	2	2
LCALL	Addr 16	Long Subroutine Call	3	2
RET		Return from Subroutine	1	2
RETI		Return from Interrupt	1	2
AJMP	Addr 11	Absolute Jump	2	2
LJMP	Addr 16	Long Jump	3	2
SJMP	Rel	Short Jump (Relative Addr.)	2	2
JMP	@A + DPTR	Jump Indirect Relative to the DPTR	1	2
JZ	Rel	Jump if Accumulator is Zero	2	2
JNZ	Rel	Jump if Accumulator is No Zero	2	2
JC	Rel	Jump if Carry Flag is Set	2	2
JNC	Rel	Jump if Carry Flag is Not Set	2	2
JB	Bit, Rel	Jump if Direct Bit is Set	3	2
JNB	Bit, Rel	Jump if Direct Bit is Not Set	3	2
JBC	Bit, Rel	Jump if Direct Bit is Set and Clear Bit	3	2
CJNE	A, Direct, Rel	Compare Direct Byte to A and Jump if Not Equal	3	2
CJNE	A, #Data, Rel	Comp. Immed. to A and Jump if Not Equal	3	2
CJNE	Rn, #Data, Rel	Comp. Immed. to Reg. and Jump if Not Equal	3	2
CJNE	@Ri, #Data, Rel	Comp. Immed. to Ind. and Jump if Not Equal	3	2
DJNZ	Rn, Rel	Decrement Register and Jump if Not Zero	2	2
DJNZ	Direct, Rel	Decrement Direct and Jump if Not Zero	3	2
NOP		No Operation	1	1

Notes on Data Addressing Modes:

Rn — Working register R0-R7

Direct — 128 internal RAM locations, any I/O port, control or status register

@Ri — Indirect internal or external RAM location addressed by register R0 or R1

#Data — 8-bit constant included in instruction

#Data16 — 16-bit constant included as bytes 2 and 3 of instruction

Bit — 128 software flags, any I/O pin, control or status bit

A — Accumulator

Notes on Program Addressing Modes:

Addr 16 — Destination address for LCALL and LJMP may be anywhere within the 64 kbyte program memory address space.

Addr 11 — Destination address for ACALL and AJMP will be within the same 2 kbyte page of program memory as the first byte of the following instruction.

Rel — SJMP and all conditional jumps include an 8-bit offset byte. Range is +127/−128 bytes relative to first byte of the following instruction.

Instruction Op Codes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	Code Addr
02	3	LJMP	Code Addr
03	1	RR	A
04	1	INC	A
05	2	INC	Data Addr
06	1	INC	@ R0
07	1	INC	@ R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	Bit Addr, Code Addr
11	2	ACALL	Code Addr
12	3	LCALL	Code Addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	Data Addr
16	1	DEC	@ R0
17	1	DEC	@ R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	Bit Addr, Code Addr
21	2	AJMP	Code Addr
22	1	RET	
23	1	RL	A
24	2	ADD	A, #Data
25	2	ADD	A, Data Addr
26	1	ADD	A, @ R0
27	1	ADD	A, @ R1
28	1	ADD	A, R0
29	1	ADD	A, R1
2A	1	ADD	A, R2
2B	1	ADD	A, R3
2C	1	ADD	A, R4
2D	1	ADD	A, R5
2E	1	ADD	A, R6
2F	1	ADD	A, R7
30	3	JNB	Bit Addr, Code Addr
31	2	ACALL	Code Addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A, #Data
35	2	ADDC	A, Data Addr
36	1	ADDC	A, @ R0
37	1	ADDC	A, @ R1
38	1	ADDC	A, R0
39	1	ADDC	A, R1
3A	1	ADDC	A, R2
3B	1	ADDC	A, R3
3C	1	ADDC	A, R4
3D	1	ADDC	A, R5
3E	1	ADDC	A, R6
3F	1	ADDC	A, R7
40	2	JC	Code Addr
41	2	AJMP	Code Addr
42	2	ORL	Data Adrr, A
43	3	ORL	Data Addr, #Data
44	2	ORL	A, #Data
45	2	ORL	A, Data Addr
46	1	ORL	A, @ R0
47	1	ORL	A, @ R1
48	1	ORL	A, R0
49	1	ORL	A, R1
4A	1	ORL	A, R2
4B	1	ORL	A, R3
4C	1	ORL	A, R4
4D	1	ORL	A, R5
4E	1	ORL	A, R6
4F	1	ORL	A, R7
50	2	JNC	Code Addr
51	2	ACALL	Code Addr
52	2	ANL	Data Adrr, A
53	3	ANL	Data Addr, #Data
54	2	ANL	A, #Data
55	2	ANL	A, Data Addr
56	1	ANL	A, @ R0
57	1	ANL	A, @ R1
58	1	ANL	A, R0
59	1	ANL	A, R1
5A	1	ANL	A, R2
5B	1	ANL	A, R3
5C	1	ANL	A, R4
5D	1	ANL	A, R5
5E	1	ANL	A, R6
5F	1	ANL	A, R7
60	2	JZ	Code Addr
61	2	AJMP	Code Addr
62	2	XRL	Data Adrr, A
63	3	XRL	Data Addr, #Data
64	2	XRL	A, #Data
65	2	XRL	A, Data Addr
66	1	XRL	A, @ R0
67	1	XRL	A, @ R1

Instruction Op Codes in Hexadecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands
68	1	XRL	A, R0
69	1	XRL	A, R1
6A	1	XRL	A, R2
6B	1	XRL	A, R3
6C	1	XRL	A, R4
6D	1	XRL	A, R5
6E	1	XRL	A, R6
6F	1	XRL	A, R7
70	2	JNZ	Code Addr
71	2	ACALL	Code Addr
72	2	ORL	C, Bit Addr
73	1	JMP	@A + DPTR
74	2	MOV	A, #Data
75	3	MOV	Data Addr, #Data
76	2	MOV	@ R0, #Data
77	2	MOV	@ R1, #Data
78	2	MOV	R0, #Data
79	2	MOV	R1, #Data
7A	2	MOV	R2, #Data
7B	2	MOV	R3, #Data
7C	2	MOV	R4, #Data
7D	2	MOV	R5, #Data
7E	2	MOV	R6, #Data
7F	2	MOV	R7, #Data
80	2	SJMP	Code Addr
81	2	AJMP	Code Addr
82	2	ANL	C, Bit Adrr
83	1	MOVC	A, @A + PC
84	1	DIV	AB
85	3	MOV	Data Addr, Data Addr
86	2	MOV	data addr, @R0
87	2	MOV	data addr, @R1
88	2	MOV	data addr, R0
89	2	MOV	data addr, R1
8A	2	MOV	data addr, R2
8B	2	MOV	data addr, R3
8C	2	MOV	data addr, R4
8D	2	MOV	data addr, R5
8E	2	MOV	data addr, R6
8F	2	MOV	data addr, R7
90	3	MOV	DPTR, #Data
91	2	ACALL	Code Addr
92	2	MOV	Bit Addr, C
93	1	MOVC	A, @A + DPTR
94	2	SUBB	A, #Data
95	2	SUBB	A, Data Addr
96	1	SUBB	A, @R0
97	1	SUBB	A, @R1
98	1	SUBB	A, R0
99	1	SUBB	A, R1
9A	1	SUBB	A, R2
9B	1	SUBB	A, R3

Hex Code	Number of Bytes	Mnemonic	Operands
9C	1	SUBB	A, R4
9D	1	SUBB	A, R5
9E	1	SUBB	A, R6
9F	1	SUBB	A, R7
A0	2	ORL	C, /Bit Addr
A1	2	AJMP	Code Addr
A2	2	MOV	C, Bit Adrr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		Reserved	
A6	2	MOV	@R0, Data Addr
A7	2	MOV	@R1, Data Addr
A8	2	MOV	R0, Data Addr
A9	2	MOV	R1, Data Addr
AA	2	MOV	R2, Data Addr
AB	2	MOV	R3, Data Addr
AC	2	MOV	R4, Data Addr
AD	2	MOV	R5, Data Addr
AE	2	MOV	R6, Data Addr
AF	2	MOV	R7, Data Addr
B0	2	ANL	C, /Bit Addr
B1	2	ACALL	Code Addr
B2	2	CPL	Bit Adrr
B3	1	CPL	C
B4	3	CJNE	A, #Data, Code Addr
B5	3	CJNE	A, Data Addr, Code Addr
B6	3	CJNE	@R0, #Data, Code Addr
B7	3	CJNE	@R1, #Data, Code Addr
B8	3	CJNE	R0, #Data, Code Addr
B9	3	CJNE	R1, #Data, Code Addr
BA	3	CJNE	R2, #Data, Code Addr
BB	3	CJNE	R3, #Data, Code Addr
BC	3	CJNE	R4, #Data, Code Addr
BD	3	CJNE	R5, #Data, Code Addr
BE	3	CJNE	R6, #Data, Code Addr
BF	3	CJNE	R7, #Data, Code Addr
C0	2	PUSH	Data Addr
C1	2	AJMP	Code Addr
C2	2	CLR	Bit Adrr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A, Data Addr
C6	1	XCH	A, @R0
C7	1	XCH	A, @R1
C8	1	XCH	A, R0
C9	1	XCH	A, R1
CA	1	XCH	A, R2
CB	1	XCH	A, R3
CC	1	XCH	A, R4
CD	1	XCH	A, R5
CE	1	XCH	A, R6
CF	1	XCH	A, R7

Instruction Op Codes in Hexadecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands
D0	2	POP	Data Addr
D1	2	ACALL	Code Addr
D2	2	SETB	Bit Adrr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	Data Addr, Code Addr
D6	1	XCHD	A, @R0
D7	1	XCHD	A, @R1
D8	2	DJNZ	R0, Code Addr
D9	2	DJNZ	R1, Code Addr
DA	2	DJNZ	R2, Code Addr
DB	2	DJNZ	R3, Code Addr
DC	2	DJNZ	R4, Code Addr
DD	2	DJNZ	R5, Code Addr
DE	2	DJNZ	R6, Code Addr
DF	2	DJNZ	R7, Code Addr
E0	1	MOVX	A, @DPTR
E1	2	AJMP	Code Addr
E2	1	MOVX	A, @R0
E3	1	MOVX	A, @R1
E4	1	CLR	A
E5	2	MOV	A, Data Addr*
E6	1	MOV	A, @R0
E7	1	MOV	A, @R1

Hex Code	Number of Bytes	Mnemonic	Operands
E8	1	MOV	A, R0
E9	1	MOV	A, R1
EA	1	MOV	A, R2
EB	1	MOV	A, R3
EC	1	MOV	A, R4
ED	1	MOV	A, R5
EE	1	MOV	A, R6
EF	1	MOV	A, R7
F0	1	MOVX	@DPTR, A
F1	2	ACALL	Code Addr
F2	1	MOVX	@R0, A
F3	1	MOVX	@R1, A
F4	1	CPL	A
F5	2	MOV	Data Addr, A
F6	1	MOV	@R0, A
F7	1	MOV	@R1, A
F8	1	MOV	R0, A
F9	1	MOV	R1, A
FA	1	MOV	R2, A
FB	1	MOV	R3, A
FC	1	MOV	R4, A
FD	1	MOV	R5, A
FE	1	MOV	R6, A
FF	1	MOV	R7, A

*MOV A,ACC is not a valid instruction

SAB 80512/80532 Single-Chip Microcontroller User's Manual

SIEMENS

Microcomputer Components

SAB 80512/80532
Single-Chip Microcontroller

User's Manual

6

Contents

Table of Contents

Introduction

1.0 Introduction

The SAB 80512 is another member of the SIEMENS SAB 8051 family. Based on the well-known industry standard 8-bit 8051 architecture, the performance and functionality of the SAB 80512 lies between that of the SAB 8051 and the SAB 80515. In addition to having all the operating characteristics of the SAB 8051, it meets market requirements of today's microcontrollers: high cost-effectiveness and ability to implement applications which need more digital I/O or an easy-to-handle on-chip A/D converter.

Listed below is a summary of the features available in the SAB 80512:

- Fully backward compatible with SAB 8051
- Pin-compatible with SAB 80515
- 4 Kbyte on-chip ROM
- 128 byte on-chip RAM
- 256 directly addressable bits
- Power-down supply for 40 bytes of RAM
- Full-duplex serial port, 4 modes of operation
- Additional baud-rate generator with two selectable baud rates
- Two 16-bit timer/counters
- Six parallel I/O ports, i.e. 48 I/O lines
- One general-purpose 8-bit input port
- A/D converter, 8 multiplexed channels, user-adjustable reference voltages
- Boolean processor
- 6 interrupt sources (2 external, 4 internal), two priority levels
- 1 μs instruction cycle at 12 MHz
- 4 μs multiply and divide
- External program and data memory, expandable up to 64 Kbytes each
- Compatible with standard SAB 8085 peripherals and memories
- Space-saving PLCC-68 package

The SAB 80532 is especially designed for applications with external program memory but without the on-chip ROM.

In this manual, any reference to the SAB 80512 applies to all versions of the SAB 80512 microcontroller unless otherwise noted. The different versions include:

SAB 80512	—	ROM version
SAB 80532	—	ROM-less version

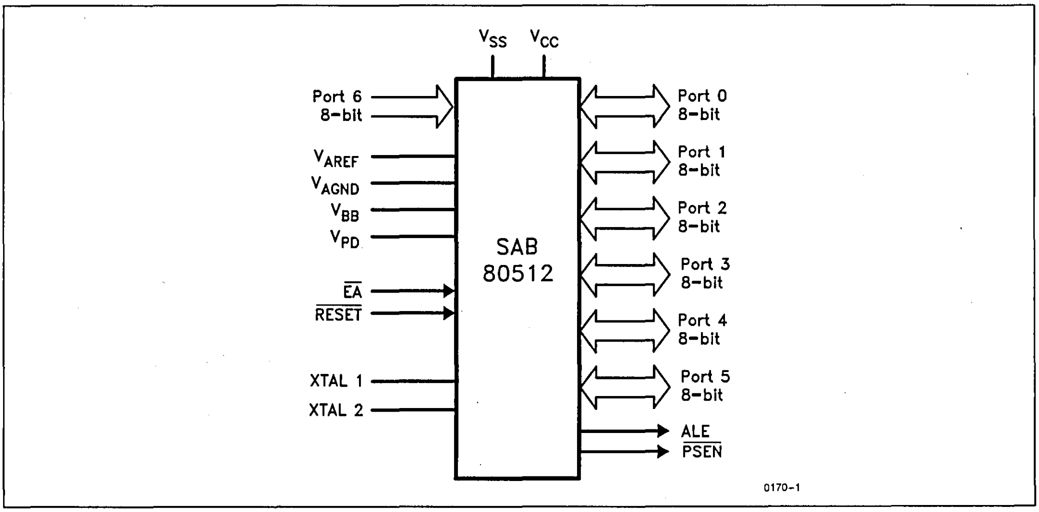


Figure 1-1. Logic Symbol

Architecture

2.0 Architecture

The core of the SAB 80512 includes the complete SAB 8051, thereby providing 100% upward compatibility between the SAB 8051 and the SAB 80512. This means that all existing 8051 programs or user's program libraries can be used without restriction and may be easily extended for the new SAB 80512. Furthermore, the SAB 80512 contains two additional I/O ports and one general input port. The serial channel can optionally be operated with two selectable baud rates (4800 baud or 9600 baud) provided by an independent baud-rate generator. An 8-bit resolution A/D converter with externally adjustable reference voltages has been integrated to allow analog signal processing. Figure 2-1 shows a block diagram of the SAB 80512. New blocks added to the 8051 are marked grey.

Readers who are familiar with the SAB 8051 may concentrate on sections 2.4 and 3.0 where the reset conditions and the new peripheral components are described.

For readers who are new to the 8051 microcontroller family the following section gives a general view of the basic characteristics of the SAB 80512. The operational details are given in later chapters.

In addition to the internal RAM there is a further 128-byte address space for the special-function registers, which are described in following sections.

Because of its Harvard architecture, the SAB 80512 distinguishes between an external program memory portion (as mentioned above) and up to 64 Kbytes of external data memory accessed by a set of special instructions.

Peripheral Control

All on-chip peripheral components—I/O ports, serial interface, timer, interrupt controller and A/D converter—are handled and controlled by the so-called special-function registers. These registers constitute the easy-to-handle interface with the peripherals. This peripheral control concept, as implemented in the SAB 8051, provides the high flexibility for further expansion as in the SAB 80512.

Moreover, some of the special-function registers, like accumulator, B-register, program status word (PSW), stack pointer (SP) and data pointer (DPTR) are used by the CPU and maintain the machine status.

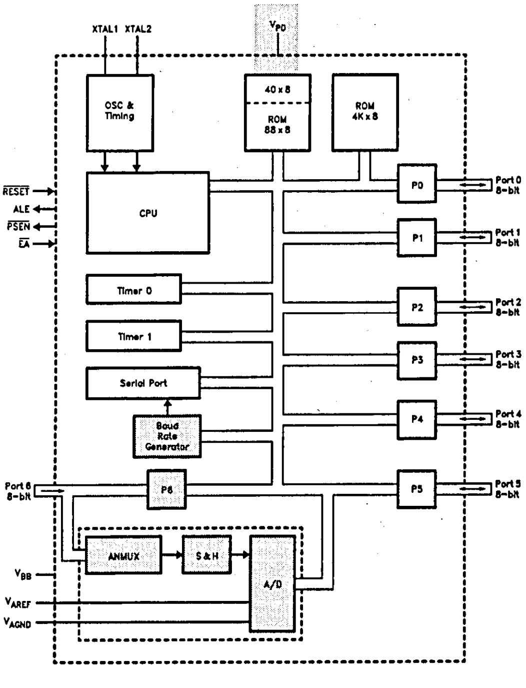


Figure 2-1. Functional Block Diagram

2.1 Central Processing Unit

The CPU is designed to operate on bits and bytes. The instructions, which may be up to 3 bytes, are performed in one, two or four machine cycles. One machine cycle requires twelve oscillator cycles. The instruction set has extensive facilities for data transfer, logic and arithmetic instructions. The Boolean processor has its own full featured and bit-based instructions within the instruction set. The SAB 80512 uses five addressing modes: direct access, immediate, register, register-indirect access, and for accessing the external data or program memory portions a base register plus index-register indirect addressing.

General Description

The CPU of the SAB 80512 consists of the instruction decoder, the arithmetic section and the program control section. Each program instruction is decoded by the instruction decoder. This unit generates the internal signals controlling the functions of the individual sections in the CPU. The signals have effect on the source and destination of data transfers, and control the ALU processing.

The arithmetic section of the processor performs extensive data manipulation and comprises the arithmetic/logic unit (ALU), A-register, B-register and PSW register. The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under the control of the instruction decoder. The ALU performs the arithmetic operations add, subtract, multiply, divide, increment, decrement, BCD-decimal-add-adjust, and compare, and the logic operations AND, OR, exclusive-OR, complement, and rotate (right, left or swap nibble (left four)). Also included is a Boolean processor performing the bit operations of set, clear, complement, jump-if-not-set, jump-if-set-and-clear and move-to/from-carry. Between any addressable bit (or its complement) and the carry flag, it can perform the bit operations of logical AND or logical OR with the result returned to the carry flag. The A, B and PSW registers are described in a later section.

The program control section controls the sequence in which the instructions stored in program memory are executed. The 16-bit program counter (PC) holds the address of the next instruction to be executed. The PC is manipulated by the control transfer instructions listed in the chapter "Instruction Set". The conditional branch logic enables internal and external events to cause a change in the program execution sequence.

2.1.1 CPU Timing

A machine cycle consists of 6 states (12 oscillator periods). Each state is divided into a phase 1 half during which the phase 1 clock is active, and a phase 2 half during which the phase 2 clock is active. Thus, a machine cycle consists of 12 oscillator periods, numbered S1P1 (state 1, phase 1) through S6P2 (state 6, phase 2). Each state lasts two oscillator periods. In general, arithmetic and logical operations take place during phase 1 and internal register-to-register transfers take place during phase 2.

The diagrams in Figure 2-2 show the fetch/execute timing related to the internal states and phases. Since these internal clock signals are not accessible by the user, the XTAL2 oscillator signals and the ALE (address latch enable) signal are shown for external reference. ALE is normally activated twice during each machine cycle: once during S1P2 and S2P1, and a second time during S4P2 and S5P1.

Execution of a one-cycle instruction begins at S1P2 when the op-code is latched into the instruction register. In the case of a two-byte instruction, the second byte is read during S4 of the same machine cycle. In the case of a one-byte instruction, there is still a fetch at S4, but the byte read (which would be the next op-code) is ignored, and the program counter is not incremented. In any case, execution is completed at the end of S6P2.

Figures 2-2A, B show the timing of a 1-byte, 1-cycle instruction and for a 2-byte, 1-cycle instruction.

Most SAB 80512 instructions execute in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than two cycles to complete; they take four cycles. Normally, two code bytes are fetched from the program memory during every machine cycle. A MOVX instruction is the only exception. MOVX is a one-byte, 2-cycle instruction that accesses external data memory. During a MOVX, the two fetches in the second cycle are skipped while the external data memory is being addressed and strobed. Figures 2-2C, D show the timing for a normal 1-byte, 2-cycle instruction and for a MOVX instruction.

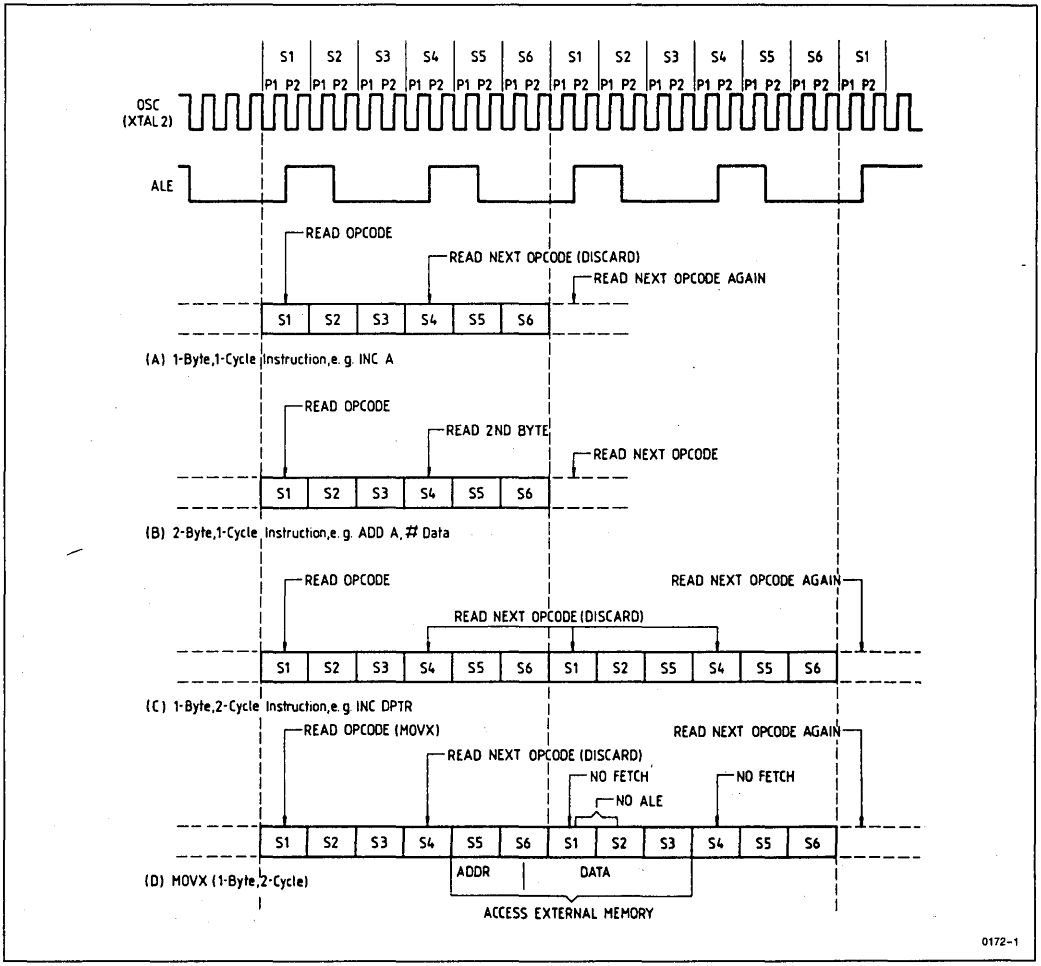

S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1
P1 P2
OSC (XTAL 2)
ALE
READ OPCODE
READ NEXT OPCODE (DISCARD)
READ NEXT OPCODE AGAIN
(A) 1-Byte,1-Cycle Instruction,e. g. INC A
READ OPCODE
READ 2ND BYTE
READ NEXT OPCODE
(B) 2-Byte,1-Cycle Instruction,e. g. ADD A,# Data
READ OPCODE
READ NEXT OPCODE AGAIN
READ NEXT OPCODE (DISCARD)
(C) 1-Byte,2-Cycle Instruction,e. g. INC DPTR
READ OPCODE (MOVX)
READ NEXT OPCODE AGAIN
READ NEXT OPCODE (DISCARD)
NO FETCH
NO ALE
NO FETCH
(D) MOVX (1-Byte,2-Cycle)
ADDR
DATA
ACCESS EXTERNAL MEMORY
0172-1

Figure 2-2. Fetch/Execute Sequence

2.2 Memory Organization

The SAB 80512 has an internal 4-Kbyte ROM. The program memory can be expanded externally up to 64 Kbytes (see Bus Expansion Control). The internal RAM has 128 bytes. Within this address space there are 128 bit-addressable locations and four register banks, each with 8 general-purpose registers.

The SAB 80512 CPU manipulates operands in the following four memory address spaces:

- Up to 64 Kbytes of program memory
- Up to 64 Kbytes of external data memory
- 128 bytes of internal data memory
- A 128-byte special-function register area

2.2.1 Program Memory

The program memory of the SAB 80512 consists of an internal and an external memory portion (see Figure 2-3). 4 Kbytes of program memory may reside on-chip (SAB 80512 only), while the SAB 80532 has no internal ROM. The program memory can be externally expanded up to 64 Kbytes. If the $\overline{EA}$ pin is held high, the SAB 80512 executes out of the internal program memory unless the address exceeds FFFH. Locations 1000H through 0FFFFH are then fetched from the external program memory. If the $\overline{EA}$ pin is held low, the SAB 80512 fetches all instructions from the external program memory. Since the SAB 80532 has no internal program memory, pin $\overline{EA}$ must be tied low when using this device. In either case, the 16-bit program counter is used for addressing.

Locations 03H through 2BH in the program memory are used by interrupt service routines.

2.2.2 Data Memory

The data memory address space consists of an internal and an external memory portion.

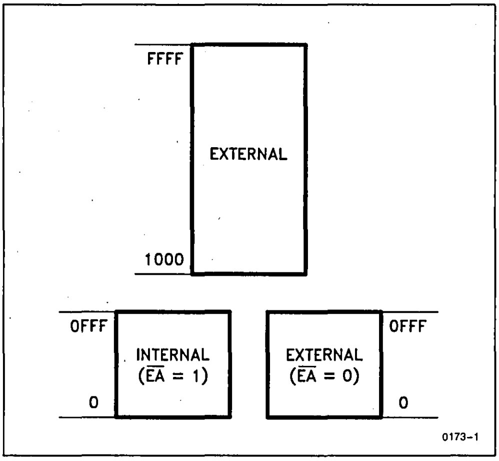


Figure 2-3. Program Memory Address Space

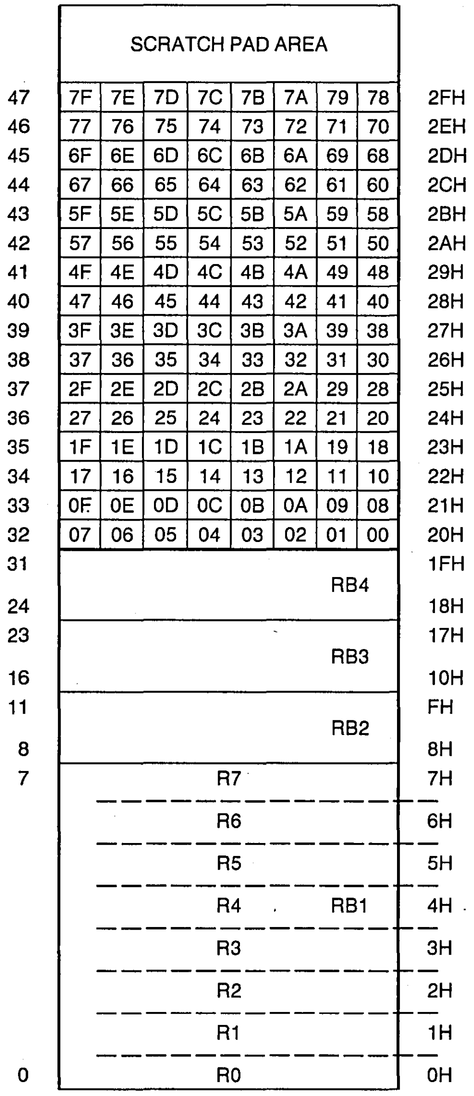
Internal Data Memory

The internal data memory address space is divided into two physically separate and distinct blocks: the lower 128 bytes of RAM and the 128-byte special-function register (SFR) area (see Figure 2-5). While the SFR area can only be accessed by direct addressing, the internal RAM is accessed by two addressing modes: direct and indirect. For details see Instruction Set.

The internal RAM is grouped in three address spaces. A general-purpose register area occupies locations 0 through 1FH (see Figure 2-4).

The next 16 bytes, locations 20H through 2FH, contain 128 directly addressable bits. These bits can be referred to in two ways, both of which are acceptable by the ASM51. One way is to refer to their address, i.e., 0 to 7FH. The other way is to refer to bytes 20H to 2FH. Thus, bits 0 to 7 can also be referred to as bits 20.0–20.7, and bits 8–0FH are the same as 21.0–21.7, and so on. Each of the 16 bytes in this segment may also be addressed as a complete byte.

Locations 30H to 7FH can be used as a scratch pad area.



	SCRATCH PAD AREA								
47	7F	7E	7D	7C	7B	7A	79	78	2FH
46	77	76	75	74	73	72	71	70	2EH
45	6F	6E	6D	6C	6B	6A	69	68	2DH
44	67	66	65	64	63	62	61	60	2CH
43	5F	5E	5D	5C	5B	5A	59	58	2BH
42	57	56	55	54	53	52	51	50	2AH
41	4F	4E	4D	4C	4B	4A	49	48	29H
40	47	46	45	44	43	42	41	40	28H
39	3F	3E	3D	3C	3B	3A	39	38	27H
38	37	36	35	34	33	32	31	30	26H
37	2F	2E	2D	2C	2B	2A	29	28	25H
36	27	26	25	24	23	22	21	20	24H
35	1F	1E	1D	1C	1B	1A	19	18	23H
34	17	16	15	14	13	12	11	10	22H
33	0F	0E	0D	0C	0B	0A	09	08	21H
32	07	06	05	04	03	02	01	00	20H
31									1FH
	RB4								
24									18H
23									17H
	RB3								
16									10H
11									FH
	RB2								
8									8H
7	R7								7H
	R6								6H
	R5								5H
	R4				RB1				4H
	R3								3H
	R2								2H
	R1								1H
0	R0								0H

Figure 2-4. Mapping of the Internal Data Memory

Using the stack pointer (SP)—a special function register described in Section 2.2.3—the stack can be located anywhere in the internal data memory address space. The stack depth is limited only by the internal RAM available (128 bytes maximum). However, pay attention to the fact that the stack should not be overwritten by other data, and vice versa.

External Data Memory

Figures 2-4 and 2-5 contain memory maps which diagram the internal/external data memory. To address data memory external to the chip, the "MOVX" instructions are used. Refer to Instruction Set or External Bus Interface for detailed descriptions of these operations. A maximum of 64 Kbytes of external data memory can be accessed by instructions using a 16-bit address.

2.2.3 General Purpose Registers

The lower 32 locations of the internal RAM are assigned to four banks with eight general-purpose registers (GPRs) each. Only one of these banks may be enabled at a time. Two bits in the program status word, PSW.3 and PSW.4, select the active register bank (see description of the PSW). This allows fast context switching, which is useful when entering subroutines or interrupt service routines. ASM51 and the device SAB 80512 default to register bank 0 after reset.

The 8 general-purpose registers of the selected register bank may be accessed by register addressing. With register addressing the instruction opcode indicates which register is to be used. For indirect addressing R0 and R1 are used as a pointer or index register to address internal or external memory (e.g. MOV @ R0).

Reset initializes the stack pointer to location 07H and is incremented once to start from location 08H which is also the first register (R0) of register bank 1. Thus, if more than one register bank is required, the SP should be initialized to a different location of the RAM, which is not used for data storage.

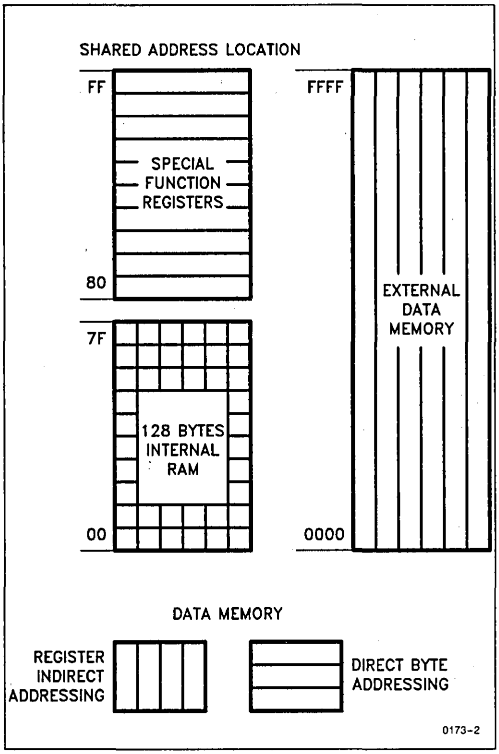


Figure 2-5. Data Memory Address Spaces

2.2.4 Special-Function Registers

The special function register (SFR) area has two important functions. Firstly, all CPU registers except the program counter and the four register banks reside here. The CPU registers are the arithmetic registers like A, B, PSW and pointers like SP, DPH and DPL.

Secondly, a number of registers constitute the interface between the CPU and all on-chip peripherals. This means that all control and data transfers to and from the peripherals use this register interface exclusively.

The special-function register area is located in the address space above the internal RAM between addresses 80H and FFH. All 28 special-function registers of the SAB 80512 reside here. The fifteen SFRs located on addresses dividable by eight are bit-addressable.

Because the SFR area is memory mapped, accesses to the special function registers are as easy as to the internal RAM, and they may be processed with most instructions. In addition, if the special functions are not used, some of them may be utilized as general scratch pad registers. Note, however, that all SFRs can be accessed by direct addressing only. The special function registers are listed in Table 2-1, and register maps are drawn in Figures 2-6a and 2-6b.

Accumulator, SFR Address 0E0H

ACC is the symbol for the accumulator register. The mnemonics for accumulator-specific instructions refer to the accumulator simply as A.

Program Status Word Register (PSW), SFR Address 0D0H

The PSW register contains program status information as shown in Figure 2-7.

B-Register, SFR Address 0F0H

The B-register is used during multiply and divide and serves as source as well as destination. For other instructions it can be treated as another scratch pad register.

Stack Pointer, SFR Address 081H

The stack pointer (SP) register is 8 bits wide. It is incremented before data are stored during PUSH and CALL executions and decremented after data are popped during POP and RET (RETI) executions, i.e. it points always to the last stack byte valid. While the stack may reside anywhere in on-chip RAM, the stack pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H above register bank zero. The SP can be read or written under software control.

Table 2-1. Special-Function Register

Symbol	Name	Address
P0	Port 0	80H
SP	Stack Pointer	81H
DPL	Data Pointer, Low Byte	82H
DPH	Data Pointer, High Byte	83H
PCON	Power Control Register	87H
TCON	Timer Control Register	88H
TMOD	Timer Mode Register	89H
TL0	Timer 0, Low Byte	8AH
TL1	Timer 1, Low Byte	8BH
TH0	Timer 0, High Byte	8CH
TH1	Timer 1, High Byte	8DH
P1	Port 1	90H
SCON	Serial Channel Control Register	98H
SBUF	Serial Channel Buffer Register	99H
P2	Port 2	0A0H
IE	Interrupt Enable Register	0A8H
P3	Port 3	0B0H
IP	Interrupt Priority Register	0B8H
IRCON	Interrupt Request Control Register	0C0H
PSW	Program Status Word Register	0D0H
ADCON	A/D Converter Control Register	0D8H
ADDAT	A/D Converter Data Register	0D9H
DAPR	A/D Converter Program Register	0DAH
P6	Port 6	0DBH
ACC	Accumulator	0E0H
P4	Port 4	0E8H
B	B-Register	0F0H
P5	Port 5	0F8H

	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H	
0B0H	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	P3
	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H	
0A8H	EAL	—	—	ES	ET1	EX1	ET0	EX0	IE
	A7H	A6H	A5H	A4H	A3H	A2H	A1H	A0H	
0A0H	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	P2
99H									SBUF
	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	
98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON
	97H	96H	95H	94H	93H	92H	91H	90H	
90H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1
8DH									TH1
8CH									TH0
8BH									TL1
8AH									TL0
89H	1 GATE	1 C/F	1M1	1M0	0 GATE	0 C/F	0M1	0M0	TMOD
	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	
88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON
87H	SMOD								PCON
83H									DPH
82H									DPL
81H									SP
	87H	86H	85H	84H	83H	82H	81H	80H	
80H	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	P0

Figure 2-6A. Special-Function Register Map, Address 80H to 0B0H

	FFH	FEH	FDH	FCH	FBH	FAH	F9H	F8H	
F8H	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0	P5
	H7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	
F0H	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	B
	EFH	EEH	EDH	ECH	EBH	EAH	E9H	E8H	
E8H	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	P4
	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H	
E0H	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	ACC
0DBH									P6
0DAH									DAPR
0D9H									ADDAT
	DFH	DEH	DDH	DCH	DBH	DAH	D9H	D8H	
0D8H	BD	—	—	BSY	ADM	MX2	MX1	MX0	ADCON
	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H	
0D0H	CY	AC	F0	RS1	RS0	OV	F1	P	PSW
	C7H	C6H	C5H	C4H	C3H	C2H	C1H	C0H	
0C0H	—	—	—	—	—	—	—	IADC	IRCON
	BFH	BEH	BDH	BCH	BBH	BAH	B9H	B8H	
0B8H	—	—	—	PS	PT1	PX1	PT0	PX0	IP

Figure 2-6B. Special-Function Register Map, Address 0B8H to F8H

Data Pointer, SFR Address 082H and 083H

The 16-bit data pointer (DPTR) register is a concatenation of registers DPH (data pointer's high-order byte) and DPL (data pointer's low order byte). These pointers are used in register-indirect addressing to move program memory constants and external data memory variables, as well as to branch within the 64-Kbyte program memory address space.

Ports 0 to 6

P0 to P5 are the SFR latches to the corresponding port 0 to 5. The port SFRs 0 to 5 are bit-addressable. Port 6 is a general-purpose input port and has no internal latch. That means, port 6 lines are used for the 8 multiplexed input lines of the A/D converter but can also be used as digital inputs. P6 is the associated SFR when the digital value is to be ready by the CPU. P6 can be read only. Because of this, Port 6 is used as the 8 multiplexed input lines of the A/D converter as well as for digital inputs.

CY	AC	F0	RS1	RS0	OV	F1	P	Bit
0D7H	0D6H	0D5H	0D4H	0D3H	0D2H	0D1H	0D0H	Address

<table>
<tr><th>Symbol</th><th>Position</th><th>Name and Significance</th></tr>
<tr><td>CY</td><td>PSW.7</td><td>Carry Flag</td></tr>
<tr><td>AC</td><td>PSW.6</td><td>Auxiliary Carry Flag (for BCD Operations)</td></tr>
<tr><td>F0</td><td>PSW.5</td><td>General-Purpose User Flag 0</td></tr>
<tr><td>RS1
RS0</td><td>PSW.4
PSW.3</td><td>Register Bank Select Control Bits 1 and 0. Set/cleared by software to determine working register bank:
<table>
<tr><th>RS1</th><th>RS0</th><th colspan="2">Enabled Working Register Bank</th></tr>
<tr><td>0</td><td>0</td><td>Bank 0</td><td>00H–07H</td></tr>
<tr><td>0</td><td>1</td><td>Bank 1</td><td>08H–0FH</td></tr>
<tr><td>1</td><td>0</td><td>Bank 2</td><td>10H–17H</td></tr>
<tr><td>1</td><td>1</td><td>Bank 3</td><td>18H–1FH</td></tr>
</table>
</td></tr>
<tr><td>OV</td><td>PSW.2</td><td>Overflow Flag</td></tr>
<tr><td>F1</td><td>PSW.1</td><td>General-Purpose User Flag 1</td></tr>
<tr><td>P</td><td>PSW.0</td><td>Parity Flag. Set/cleared by hardware in each instruction cycle to indicate an odd/even number of "one" bits in the accumulator, i.e. even parity.</td></tr>
</table>

Figure 2-7. Program Status Word Register (0D0H)

Peripheral Control, Data and Status Register

The following table lists the control, status, and data registers which handle the on-chip peripherals.

In the table the register names are organized in groups and each of these groups refers to one peripheral unit. More details on register programming are given in the description of the corresponding peripheral units.

Timer 0/1	Serial Channel	Interrupt System	A/D Converter
TCON	SCON	IE	ADCON
TMOD	SBUF	IP	ADDAT
TL0	PCON	IRCON	DAPR
TH0	(ADCON)		
TL1			
TH1			

2.3 External Bus Interface

The external bus interface of the SAB 80512 consists of an 8-bit data bus (port 0), a 16-bit address bus (port 0 and port 2) and five control lines. The address latch enable signal (ALE) is used to demultiplex address and data of port 0. The program memory is accessed by the program store enable signal ($\overline{\text{PSEN}}$) twice a machine cycle. A separate external access line ($\overline{\text{EA}}$) is used to inform the controller, while executing out of the lower 4 kbytes of the program memory, whether to operate out of the internal or external program memory. The read or write strobe ($\overline{\text{RD}}$, $\overline{\text{WR}}$) is used for accessing the external data memory.

The SAB 80512 allows external memory expansion. To accomplish this, the external bus interface common to most 8051-based controllers is utilized.

2.3.1 Accessing External Memory

It is possible to distinguish between accesses to external program memory and external data memory or other peripheral components, respectively. This distinction is made by hardware: Accesses to external program memory use the signal $\overline{\text{PSEN}}$ (program store enable) as a read strobe. Accesses to external data memory use $\overline{\text{RD}}$ and $\overline{\text{WR}}$ (alternate functions of P3.7 and P3.6, see Section 3.1.2) to strobe the memory. Port 0 and port 2 (with exceptions) provide data and address signals. In this section only the port 0 and port 2 functions relevant to external memory accesses are described (for more detailed information see Chapter 3.1).

Fetches from external program memory always use a 16-bit address. Accesses to external data memory can use either a 16-bit address (MOVX @ DPTR) or an 8-bit address (MOVX @ Ri).

Role of P0 and P2 as Data/Address Bus

When used for accessing external memory, port 0 provides the data byte time-multiplexed with the low byte of the address. In this state, port 0 is disconnected from its own port latch, and the address/data signals drive both FETs in the port 0 output buffers. In this application, the port 0 pins are not open-drain outputs and do not require external pullup resistors.

During any access to external memory, the CPU writes 0FFH to the port 0 latch (the special-function register), thus obliterating whatever information the port 0 SFR may have been holding.

Whenever a 16-bit address is used, the high byte of the address comes out on port 2, where it is held during the read or write cycle. During this time, the port 2 lines are disconnected from the port 2 latch (the special-function register).

Thus, the port 2 latch does not have to contain 1s, and the contents of the port 2 SFR are not modified.

If an 8-bit address is being used (MOVX @Ri), the contents of the port 2 SFR remain at the port 2 pins throughout the external memory cycle. This will facilitate paging. It should be noted that if a port 2 pin outputs an address bit that is a 1, strong pullups will be used for the entire read/write cycle and not only for two oscillator periods.

Timing

The timing of the external bus interface, in particular the relation of the control signals ALE, $\overline{\text{PSEN}}$ and $\overline{\text{RD}}$/$\overline{\text{WR}}$ to port 0 and port 2 information, is illustrated in Figures 2-8A to 2-8C.

Data memory: in a write cycle, the data byte to be written appears on port 0 just before $\overline{\text{WR}}$ is activated, and remains there until after $\overline{\text{WR}}$ is deactivated. In a read cycle, the incoming byte is accepted at port 0 before the read strobe is deactivated.

Program memory: signal $\overline{\text{PSEN}}$ works as a read strobe. For more detailed information see Section 2.3.2.

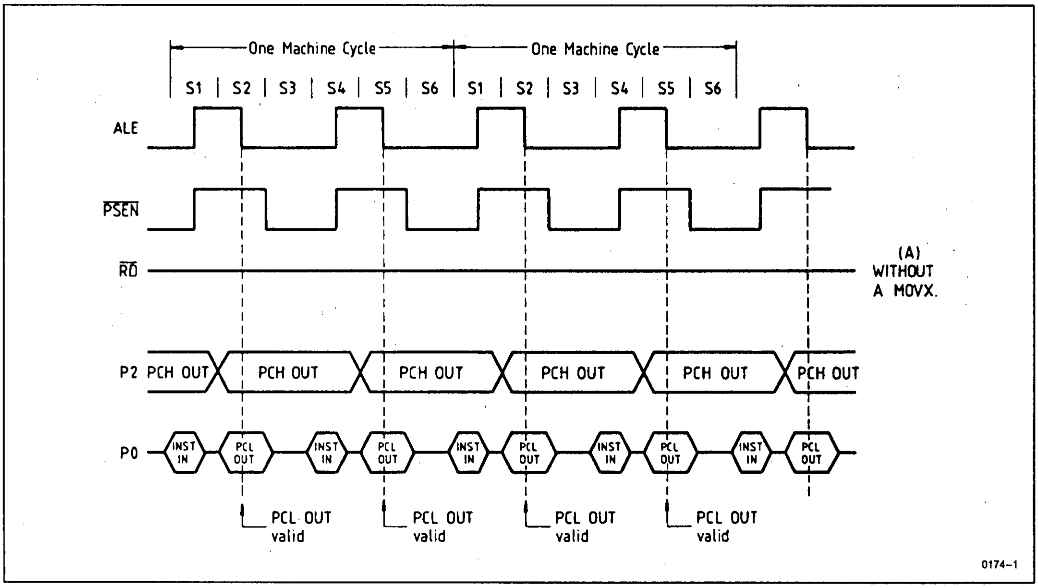


Figure 2-8A. External Program Memory Fetch

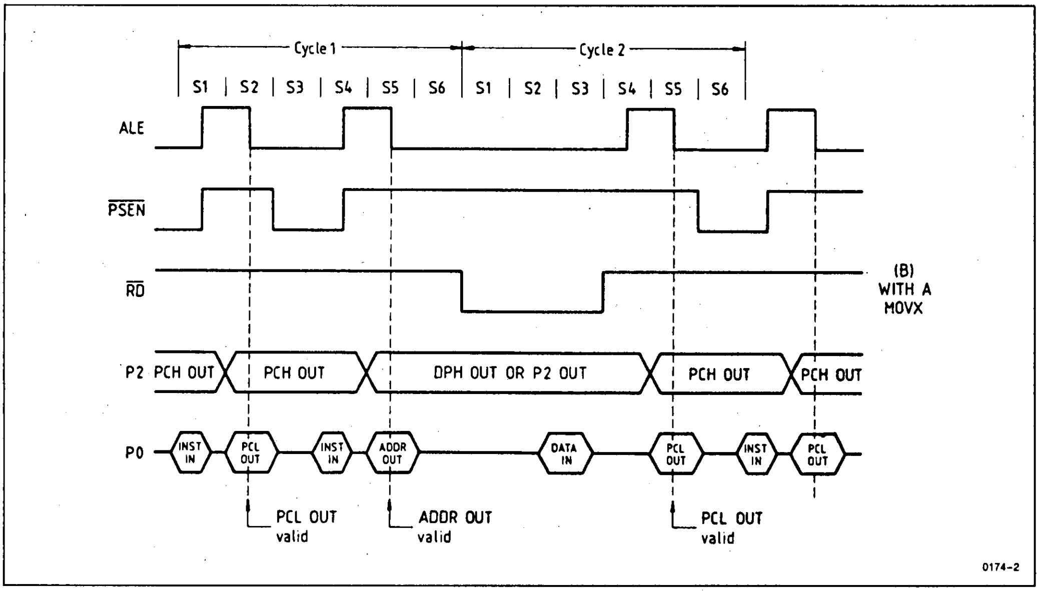


Figure 2-8B. External Data Memory Read

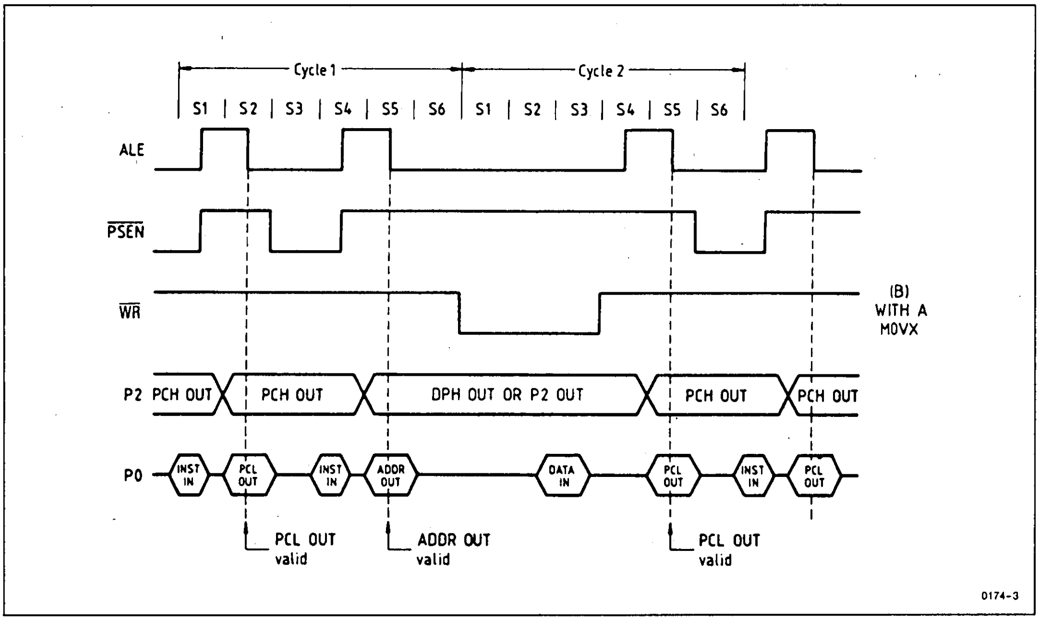


Figure 2-8C. External Data Memory Write

External Program Memory is Accessed under Two Conditions:

1. Whenever signal $\overline{EA}$ is active; or
2. Whenever the program counter (PC) contains a number that is larger than 0FFFH

This requires the ROM-less version SAB 80532 to have $\overline{EA}$ wired low to allow the lower 4k program bytes to be fetched from external memory.

When the CPU is executing out of external program memory, all 8 bits of port 2 are dedicated to an output function and may not be used for general-purpose I/O. The contents of the port 2 SFR are not affected. During external program memory fetches port 2 lines output the high byte of the PC, and during accesses to external data memory they output either DPH or the port 2 SFR (depending on whether the external data memory access is a MOVX @ DPTR or a MOVX @ Ri).

Since the SAB 80532 has no internal program memory, accesses to program memory are always external, and port 2 is at all times dedicated to output the high-order address byte. This means that port 0 and port 2 of the SAB 80532 can never be used as general-purpose I/O. This also applies to the SAB 80512 when it is operated only with an external program memory.

2.3.2 $\overline{PSEN}$, Program Store Enable

The read strobe for external fetches is $\overline{PSEN}$. $\overline{PSEN}$ is not activated for internal fetches. When the CPU is accessing external program memory, $\overline{PSEN}$ is activated twice every cycle (except during a MOVX instruction) no matter whether or not the byte fetched is actually needed for the current instruction. When $\overline{PSEN}$ is activated its timing is not the same as for $\overline{RD}$. A complete $\overline{RD}$ cycle, including activation and deactivation of ALE and $\overline{RD}$, takes 12 oscillator periods. A complete $\overline{PSEN}$ cycle, including activation and deactivation of ALE and $\overline{PSEN}$ takes 6 oscillator periods. The execution sequence for these two types of read cycles is shown in Figures 2-8A/B.

2.3.3 ALE, Address Latch Enable

The main function of ALE is to provide a properly timed signal to latch the low byte of an address from P0 into an external latch during fetches from external memory. The address byte is valid at the negative transition of ALE. For this reason, ALE is activated twice per machine cycle. This activation takes place even when the cycle involves no external fetch. The only time an ALE pulse does not come out is during an access to external data memory when $\overline{RD}/\overline{WR}$ signals are active. The first ALE of the second cycle of a MOVX instruction is missing (see Figures 2-8B/C). Consequently, in any system that does not use data memory, ALE is activated at a constant rate of ⅙ of the oscillator frequency, and can be used for external clocking or timing purposes.

2.3.4 Overlapping of External Data and Program Memory Spaces

In some applications it is required to execute a program from the same physical memory that is used to store data. In the SAB 80512, the external program and data memory spaces can be combined by ANDing $\overline{\text{PSEN}}$ and $\overline{\text{RD}}$. A positive logic AND of these two signals produces an active low read strobe that can be used for the combined physical memory. Since the $\overline{\text{PSEN}}$ cycle is faster than the $\overline{\text{RD}}$ cycle, the external memory must be fast enough to accommodate the $\overline{\text{PSEN}}$ cycle.

2.4 System Reset

The reset function incorporated in the SAB 80512 allows an easy and automatic start-up with low hardware expense and forces the controller to a predefined default state.

The reset input is an active low input on pin 10. An internal Schmitt trigger is used at the input for noise rejection. Since the reset is synchronized internally, the $\overline{\text{RESET}}$ pin must be held low for at least two machine cycles (24 oscillator periods) while the oscillator is running. The internal reset is executed during the second cycle in which $\overline{\text{RESET}}$ is low and is repeated every cycle until RESET goes high again. During reset, ALE and $\overline{\text{PSEN}}$ are configured as inputs and may not be stimulated externally. An external stimulation at these lines during reset activates several test modes, which are reserved for the fab test.

A pullup resistor is internally connected to V_{CC} to allow a power-up reset with an external capacitor only. An automatic reset can be obtained when V_{CC} is applied by connecting the reset pin to GND through a capacitor as shown in Figure 2-9. After V_{CC} is turned on, the capacitor must hold the voltage at the reset pin for a specified time at a level which remains under the higher threshold of the Schmitt trigger to effect a complete reset. Under normal conditions this must last at least 10 ms for a crystal oscillator and 50 µs for a ceramic oscillator. The time required is the oscillator start-up time, plus 2 machine cycles.

To ensure proper entry into the initialization software, a hardware branch to location zero is made immediately following reset.

The system state of the SAB 80512 is determined by the contents of its special-function registers. The default values to which they are forced during reset are listed in Table 2-2. After reset is internally accomplished the port latches of port 0 to 5 default in 0FFH. This leaves port 0 floating, because it is an open-drain port when not used as data/address bus. All other I/O ports lines (port 1 through 5) output a one (1).

Port 6, which is an input port only, has no internal latch and therefore the contents of the special function register P6 depend on the levels applied to the port 6 lines.

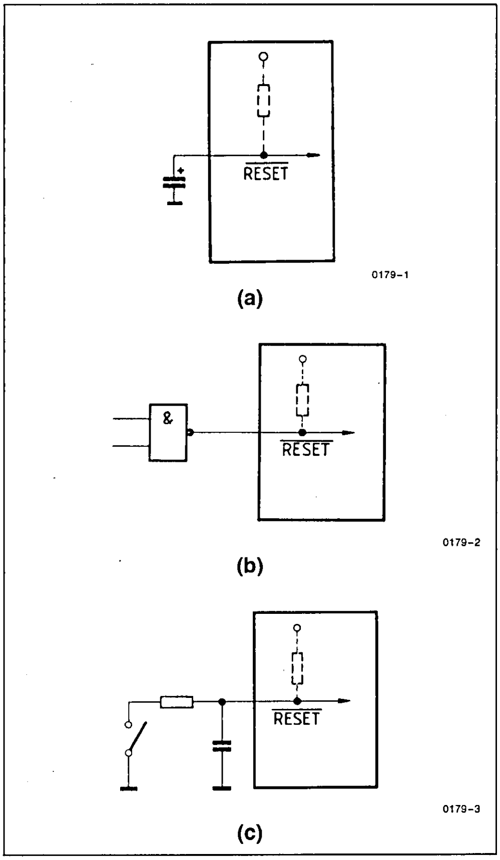


Figure 2-9. Reset Circuits

Table 2-2. Register Contents after Reset

Register	Contents	Register	Contents
P0–P5	0FFH	IE	00H
SP	07H	IP	00H
DPTR	0000H	IRCON	00H
PCON	(0XXXXXXX)	PSW	00H
TCON	00H	ADCON	00H
TMOD	00H	ADDAT	00H
TL0, TH0	00H	DAPR	Indeterminate
TL1, TH1	00H	A	00H
SCON	00H	B	00H
SBUF	Indeterminate		

X means that the value is indeterminate

On-Chip Peripheral Components

3.0 On-Chip Peripheral Components

This chapter gives detailed information about all on-chip peripherals of the SAB 80512 except for the integrated interrupt controller, which is described separately in Chapter 4.0. Sections 3.1 and 3.2 are associated with the various I/O facilities, while the remaining sections describe the miscellaneous functions such as the timers, A/D converter, oscillator and the standby power supply.

3.1 Parallel I/O

3.1.1 Port Structures

Digital I/O

The SAB 80512 allows digital I/O on 48 lines grouped into 6 bidirectional 8-bit ports. Each port bit consists of a latch (special-function register P0 to P5), an output driver and an input buffer.

The output drivers of port 0 and 2 and the input buffers of port 0 have the alternate function of accessing external memory. In this application, port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise, the port 2 pins continue to emit the P2 SFR contents (see Chapter 3, External Bus Interface).

Digital/Analog Input Port

Port 6 is available as an input port only and provides for two functions. When used for digital input, the SFR P6 contains the digital value, applied to port 6 lines. When used for analog inputs the required analog channel is selected by a 3-bit field in SFR ADCON, as described in section 3.4.

If a digital value is to be read, the voltage levels have to be held within specs (V_{IL}/V_{IH}). Since P6 is not a bit-addressable register, all 8 input lines are read at the same time. Nevertheless, there is the possibility of using port 6 simultaneously for analog and digital input. However, the user has to ensure that all bits of P6 are masked by software which have an undetermined value caused by their analog function.

To guarantee high-quality A/D conversion, digital inputs at some lines of port 6 should not toggle while a neighboring port 6 pin is used as analog input. This could produce crosstalk to the analog signal.

Digital Port Circuits

Figure 3-1 shows a functional diagram of a typical bit latch and I/O buffer which is the core of each of the 6 I/O ports. The bit latch (one bit in the port's SFR) is represented as a typ-D flipflop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flipflop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, while others activate the "read pin" signal.

Port 1 through 5 output drivers have internal pullups (see Figure 3-2). Each I/O line can be used independently as an input or output. To be used as an input, the port bit must contain a one (1) (that means for Figure 3-2: $\overline{Q} = 0$), which turns off the output driver FET. Then, for ports 1 through 5, the pin is pulled high by the internal pullup, but can be pulled low by an external source. When externally pulled low the port pins source current. For this reason they are sometimes called "quasi bidirectional". In fact, the pullups mentioned above and drawn in Figure 3-2 are pullup arrangements as shown in Figure 3-3. Two pullup FETs are used because the deep depletion type is able to restrict the low level input current (I_{IL}) when externally pulled low, but it is not strong enough to drive a fast 0 to 1 transition at the port pin. For the latter purpose an additional pullup is turned on for the two phases (S1P1 and S1P2) in which the transition occurs. The extra pullup can drive about 100 times the current that the normal pullup can. It should be noted that all internal pullups are FETs, not linear resistors.

Port 0, contrary to ports 1 through 5, is considered "true" bidirectional, because the pin floats when configured as input. Thus this port differs in not having internal pullups. The pullup FET in the P0 output driver (see Figure 3-4a) is used only when the port is emitting 1s during the external memory accesses. Otherwise the pullup is off. Consequently, P0 lines that are being used as general-purpose output port lines are open drain lines.

Writing a 1 to the bit latch leaves both output FETs off, so that the pin floats. In that condition it can be used as high-impedance input. If port 0 is configured as general I/O port and has to emit ones (1s), external pullups are required.

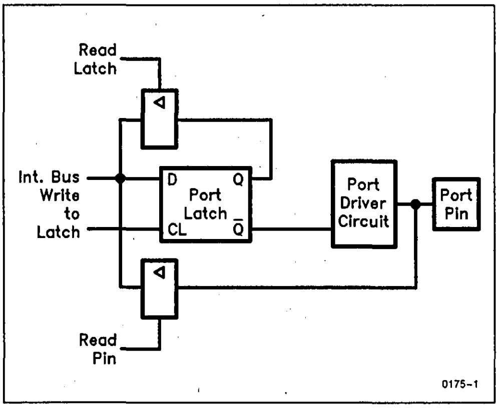


Figure 3-1. Basic Structure of Port Circuitry

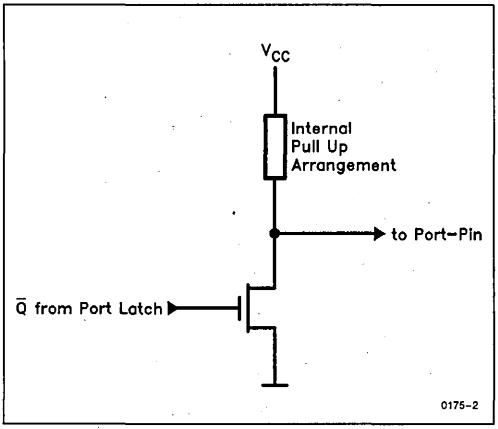


Figure 3-2. Basic Output Driver Circuit of Ports 1 through 5

Port 0 and Port 2 Used as Address/Data Bus

As shown in Figures 3-4a and 3-4b, the output drivers of ports 0 and 2 are switchable to an internal address and address/data bus for use in external memory accesses. In this application they may not be used as general purpose I/O. The switch is done by an internal control signal dependent on the input level as the $\overline{EA}$ pin and/or the contents of the program counter. If the ports are configured as an address/data bus, the port latches are disconnected from the driver circuit. During this time, the P2 SFR remains unchanged while the P0 SFR gets 1s written to it. Being an address/data bus, port 0 uses a pullup FET as shown in Figure 3-4a. When a 16-bit address is used, port 2 uses the additional strong pullups to emit 1s for the entire external memory cycle instead of the weak ones used during normal port activity.

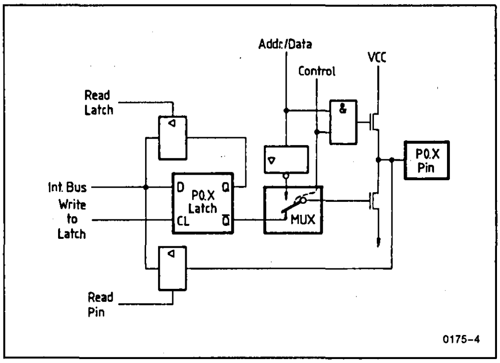


Figure 3-4a. Port 0 Circuitry

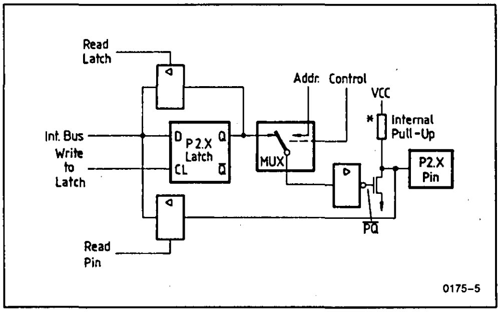


Figure 3-4b. Port 2 Circuitry

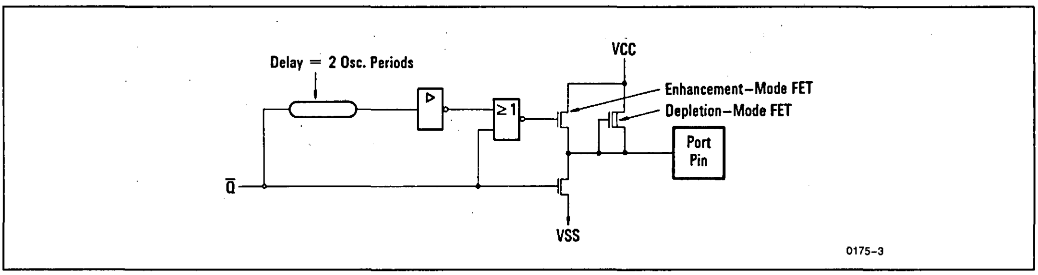


Figure 3-3. Output Driver Circuit of Ports 1 through 5, Detailed Circuit Diagram

3.1.2 Alternate Functions

All pins of port 3 are multifunctional. They are not only port pins, but also serve for various special features as listed in Table 3-1.

Figure 3-5 shows a functional diagram of port 3 latch. The input buffer and output driver circuit are like those in port 1, port 4 or 5. To pass the alternate function to the output pin and vice versa, however, the gate between the latch and driver circuit has to be open. Thus, to use the alternate input or output functions, the corresponding bit latch in the port SFR has to contain a one (1); otherwise to pulldown FET is on and the port pin is stuck at 0. After Reset all port latches contain ones (1).

Table 3-1. Alternate Functions of Port 3

Port	Pin	Alternate Function
P3.0	RXD	Serial Input Channel
P3.1	TXD	Serial Output Channel
P3.2	$\overline{\text{INT0}}$	External Interrupt 0
P3.3	$\overline{\text{INT1}}$	External Interrupt 1
P3.4	T0	Timer 0 External Counting Input
P3.5	T1	Timer 1 External Counting Input
P3.6	$\overline{\text{WR}}$	External Data Memory Write Strobe
P3.7	$\overline{\text{RD}}$	External Data Memory Read Strobe

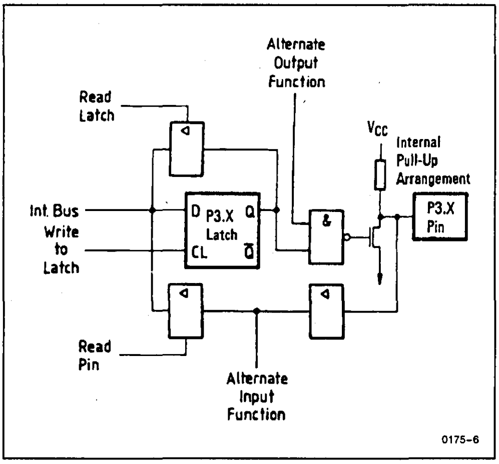


Figure 3-5. Port 3 Circuit

3.1.3 Port Handling

In the execution of an instruction that changes the value of a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are in fact sampled by their output buffers only during phase 1 of any clock period (during phase 2 the output buffer holds the value it saw during the previous phase 1). Consequently, the new value in the port latch will not actually appear at the output pin until the next phase 1, which will be at S1P1 of the next machine cycle.

Port Loading and Interfacing

The output buffers of ports 1 through 5 can each drive 4 LS-TTL inputs.

Port 0 output buffers can each drive 8 LS-TTL inputs. They do, however, require external pullups to drive floating inputs, except when being used as the address/data bus.

Read Modify Write Feature

Some instructions that read a port read the latch and others read the pin. The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions. The instructions listed in Table 3-2 are the read-modify-write instructions. When the destination operand is a port, or a port bit, these instructions read the latch rather than the pin. Note that all other instructions that can be used to read a port, exclusively read the port pin.

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a "1" is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and possibly interpret it as 0. Reading the latch rather than the pin will return the correct value of "1".

Table 3-2. Read-Modify-Write Instructions

ANL	Logic AND; e.g. ANL P1, A
ORL	Logic OR; e.g. ORL P2, A
XRL	Logic Exclusive OR; e.g. XRL P3, A
JBC	Jump if Bit is Set and Clear Bit: e.g. JBC P1.1., LABEL
CPL	Complement Bit; e.g., CPL P3.0
INC	Increment; e.g. INC P2
DEC	Decrement; e.g. DEC P2
DJNZ	Decrement and Jump if Not Zero; e.g. DJNZ P3, LABEL
MOV PX.Y, C	Move Carry Bit to Bit Y of Port X
CLR PX.Y	Clear Bit Y of Port X
SETB PX.Y	Set Bit Y of Port X

3.2 Serial Interface

The SAB 80512 includes a serial interface (USART) which provides one synchronous and three asynchronous operating modes. This serial channel is full duplex, meaning it can transmit and receive simultaneously. In addition it is receive buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register (however, if the first byte still has not been read by the time reception of the second byte is complete, the last received byte will be lost).

The serial interface uses a number of special-function registers for control and communication to the CPU which are listed below:

SBUF The receive **and** transmit register are both accessed by this SFR. Writing to this serial channel buffer loads the transmit register, and reading SBUF accesses a physically separate receive register.

SCON This register controls the serial channel and provides information about its status. It also contains the interrupt request flags for the serial channel.

PCON Only one bit in register PCON is of interest for serial communication. Bit 7 (PCON.7 or SMOD) doubles the selected baud rate.

ADCON Only bit 7 of ADCON is used for the serial interface. It enables an additional timer of the 80512 used for generating two selectable baud rates (more details in section 3.2.3).

Two port pins of port 3 are used for input/output of the serialized data:

P3.0/RxD RxD is the alternate function of Port 3.0. Through this pin data is received in mode 1, 2 and 3. In mode 0, RxD is used for input and output.

P3.1/TxD TxD is the alternate function of Port 3.1. Through this pin data is transmitted in mode 1, 2 and 3. In mode 0, TxD outputs the shift clock.

Note that in this application both port bit latches must contain a one (1)! Otherwise the output or input line is stuck at 0 and no transfer is possible.

The baud rate for the serial interface can be derived from several possible sources. It is either directly generated from a fractional part of the oscillator frequency or from timer 1, or from a dedicated baudrate timer. The serial interface is able to operate in one of the following four modes:

Mode 0: Shift Register Mode:

Serial data enter and exit through RxD. TxD outputs the shift clock. 8 data bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: 8-bit UART, Variable Baud Rate:

10 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB8 in special-function register SCON. The baud rate is variable.

Mode 2: 9-Bit UART, Fixed Baud Rate:

11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). On transmission, the 9th data bit (TB8 in SCON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB8 or a second stop bit by setting TB8 to 1. On reception the 9th data bit goes into RB8 in special-function register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 3: 9-Bit UART, Variable Baud Rate:

11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). In fact, mode 3 is the same as mode 2 in all respects except for the baud rate. The baud rate in mode 3 is variable.

3.2.1 Setup and Control

For correct setup of the serial channel SFR SCON (address 98H) has to be initialized, just as the source of the baud rate clock (and the speed) has to be determined.

Figure 3-6 shows the function of the control bits in SCON.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI	Bit
9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	Address

Symbol	Position	Function
SM0 SM1	SCON.7 SCON.6	Serial Mode 0,1. Serial port mode selection, see Table 3-3.
SM2	SCON.5	Serial Mode 2. Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3, if SM2 is set to 1, the RI wil not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1, RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
REN	SCON.4	Receiver Enable. Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB8	SCON.3	Transmitter Bit 8. It is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
RB8	SCON.2	Receiver Bit 8. In modes 2 and 3, it is the 9th data bit that was received. In mode 1, if SM1 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmitter Interrupt. This is the transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
RI	SCON.0	Receiver Interrupt. Is the receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or during the stop bit time in the other modes, in any serial reception. Must be cleared by software.

Figure 3-6. Serial Port Control Register SCON (98H)

Table 3-3. Serial Port Mode Selection

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	fosc$_{/12}$
0	0	1	8-Bit UART	Variable
1	0	2	9-Bit UART	fosc$_{/64}$ or fosc$_{/32}$
1	1	3	9-Bit UART	Variable

One of four serial modes is selected with bits SM0 and SM1 (see Table 3-3). It should be noted that reception is possible only if bit REN is set. The function of bit SM2, which enables the multiprocessor communication feature, is explained in the following section.

In modes 2 and 3, which are the 9-bit UART modes, bits TB8 and RB8 hold the contents of the 9th bit transmitted or received, respectively. The 9-bit can be programmed and checked by the user's software, e.g. it can be used as a parity bit.

RI and TI are the interrupt request flags and indicate that a transfer has been completed. Furthermore, they indicate whether it was reception or transmission which generated the serial port interrupt. Of course, both flags can also be polled by the "Jump on Bit" instruction (JB bit, rel. address).

The RI and TI flags are set during the stop bit clock period (or at the end of the 8th bit clock period in mode 0) and have to be cleared before completion of a new reception or transmission. This must be done in order to properly indicate completed transfers.

After initialization, and provided the proper baud-rate clock is generated, transmission is activated by any instruction that uses SBUF as a destination register. Reception is initiated by the incoming start bit (1-to-0 transition at RxD) for mode 1 through 3 assumed REN = 1. For mode 0, reception is initiated by the condition REN = 1 and RI = 0.

3.2.2 Multiprocessor Communication

Modes 2 and 3 of the serial interface, respectively, have a special provision for multiprocessor communication. In these modes 9 data bits are received. The 9th bit goes into RB8 and is followed by a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor communication is as follows:

When the master processor wants to transmit a block of data to one of the several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be incoming. After having received a complete message, the slave is setting SM2. The slaves that were not addressed leave their SM2 set and go on about their business, ignoring the incoming data bytes.

SM2 has no effect in mode 0. In mode 1 SM2 can be used to check the validity of the stop bit. If SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

3.2.3 Generating Baud Rates

As already mentioned there are several possibilities to generate the baud rate clock for the serial interface depending on the mode in which it is operated. To clarify the terminology, something should be said about the difference between "baud rate clock" and "baud rate". For internal synchronization, the serial interface requires a clock rate which is 16 times the baud rate as is mentioned in the description of the various operating modes later on. Therefore, the baud-rate generators have to provide a "baud-rate clock" to the serial interface which—there divided by 16—results in the actual "baud rate". However, all formulas given in the following section already include the factor and calculate the final baud rate.

Mode 0

The baud rate in mode 0 is fixed:

$$\text{Mode 0 baud rate} = \frac{\text{oscillator frequency}}{12}$$

Thus, this rate is equivalent to the machine cycle rate and is named fosc$_{/12}$.

Mode 2

This mode provides a 9-bit UART with two fixed baud rates. Since the baud-rate clock is directly derived from the oscillator frequency in this mode the fastest baud rates for the UART of the SAB 80512 can be achieved.

The baud rate in mode 2 depends on the value of bit SMOD in special-function register PCON (see Figure 3-7). If SMOD = (which is the value after reset), the baud rate is $\frac{1}{64}$ of the oscillator frequency. If SMOD = 1, the baud rate is $\frac{1}{32}$ of the oscillator frequency.

$$\text{Mode 2 baud rate} = \frac{2^{SMOD}}{64} \times \text{oscillator frequency}$$

Mode 1 and Mode 3

In these modes the baud rate is variable and can be generated alternatively by a dedicated baud rate generator or by timer 1.

Using the Baud Rate Generator:

If either 4800 or 9600 baud are required, then a special feature of the SAB 80512 can be used. A prescaler supplied by the phase 2 clock (fosc$_{/12}$) provides a baude rate clock to the serial interface. The commonly used rate of 4800 baud at 12 MHz oscillator frequency may be doubled to 9600 baud when bit SMOD in SFR PCON (87H) is set.

By setting bit BD, special-function register bit ADCON.7, the "internal baud rate generator" can easily be activated. It thereby frees timer 1 for general-purpose use.

$$\text{Baud Rate (BD = 1)} = \frac{2^{SMOD}}{2500} \times \text{oscillator frequency}$$

SMOD	—	—	—	—	—	—	—	87H
PCON.7	PCON.6	PCON.5	PCON.4	PCON.3	PCON.2	PCON.1	PCON.0	

Note! This register is not bit-addressable.

Symbol	Position	Function
SMOD	PCON.7	When set, the baud rate of the serial channel in mode 1, 2, 3, is doubled.
—	PCON.6	Reserved
—	PCON.5	Reserved
—	PCON.4	Reserved
—	PCON.3	Reserved
—	PCON.2	Reserved
—	PCON.1	Reserved
—	PCON.0	Reserved

Using Timer 1 to Generate Baud Rates:

In this case the baud rates in mode 1 and 3 are determined by the timer 1 overflow rate and the value of SMOD as follows:

$$\text{Mode 1, 3, baud rate} = \frac{2^{SMOD}}{32} \times \begin{matrix}\text{(timer 1}\\ \text{Overflow Rate)}\end{matrix}$$

The timer 1 interrupt should be disabled in this application. The timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications it is configured for "timer" operation in the auto-reload mode (high nibble of TMOD = 0010B). In that case the baud rate is given by the formula:

$$\text{Mode 1, 3, baud rate} = \frac{2^{SMOD}}{32} \times \frac{\text{oscillator frequency}}{12 \times (256 - (TH1))}$$

One can achieve very low baud rates with timer 1 by leaving the timer 1 interrupt enabled, configuring the timer to run as 16-bit timer (high nibble of TMOD = 0001B), and using the timer 1 interrupt to do a 16-bit software reload.

Table 3-4 lists various commonly used baud rates and how they can be obtained from timer 1.

3.2.4 Modes of Operation

This section gives a more detailed description of the various operating modes.

Mode 0 Synchronous Mode

Serial data enter and exit through RxD. TxD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency. In this mode the serial interface can be viewed as a shift register.

Any instruction that uses SBUF as a destination register activates the "Write-to-SBUF" signal at S6P2. One full machine cycle later the output of the "shift register" is enabled to the alternate output function line P3.0. In state 3 phase 1 of this cycle, the first transition of the shift clock also occurs at the alternate output function line P3.1 (see Figure 3-8). The shift clock is low during S3, S4 and S5 of every machine cycle and high during S6, S1, S2 while the interface is transmitting. The shift clock remains high before and after transmission. At S6 P2 of every machine cycle in which a transmission takes place, the contents of the shift register are shifted one position to the right. In the 10th machine cycle after "write-to-SBUF", RxD outputs a one (1) and sets TI.

Reception is initiated by the condition REN = 1 and RI = 0. At S3P1 of the cycle following the one in which RI was cleared, the first 1-to-0 transition of the shift clock occurs at the alternate output function line of P3.1. At S6P1 in every machine cycle in which a reception is activated, the contents of the receive shift register are shifted one position to the left. The value that comes in from RxD is the value that was sampled at the P3.0 pin at S5P2 in the same machine cycle. At S1P1 in the 10th machine cycle after the write-to-SCON that cleared RI, reception is disabled and RI is set.

Table 3-4. Timer 1 Generated Commonly Used Baud Rates

Baud Rate		fosc MHz	SMOD	Timer 1		
				C/T	Mode	Reload Value
Mode 0 Max:	1 MHz	12.0	X	X	X	X
Mode 2 Max:	375.0 Kbaud	12.0	1	X	X	X
Mode 1, 3:	62.5 Kbaud	12.0	1	0	2	FFH
	19.5 Kbaud	11.059	1	0	2	FDH
	9.6 Kbaud	11.059	0	0	2	FDH
	4.8 Kbaud	11.059	0	0	2	FAH
	2.4 Kbaud	11.059	0	0	2	F4H
	1.2 Kbaud	11.059	0	0	2	E8H
	110 Baud	6.0	0	0	2	72H
	110 Baud	12.0	0	0	1	FEEBH

Mode 1, 8-Bit UART

Ten bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception through RxD, the stop bit goes into RB8 (SCON). A transmission is activated by any instruction that uses SBUF as a destination register. The associated timing is shown in Figure 3-9. The baud rate for the serial interface is determined by the timer 1 overflow rate or by the internal baud-rate generator.

However, transmission is internally synchronized to a divide-by-16 counter and not to the "write-to-SBUF" signal. This divide by 16 counter is clocked by the "baud rate clock".

All eight data bits are shifted out through TxD after the start bit is transmitted. This occurs at TxD in S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. The bit time is determined by the baud rate selected.

When the MSB of the data byte is at the output position of the internal shift register, the next shift disables the register output, the stop bit is placed on TxD, and TI is set. This occurs at the 10th divide-by-16 rollover after "write-to-SBUF".

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose, RxD is sampled at a rate of 16 times whatever baud rate has been estabilshed. When a transition is detected, the divide-by-16 counter is immediate reset. The rollovers of this counter are thereby aligned with the boundaries of the incoming bits.

The 16 states of the counter divide each bit time by 16. At the 7th, 8th and 9th counter state of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to look for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the internal input shift register, and reception of the rest of the frame will proceed.

When the start bit arrives at the last position in the 9-bit input shift register, one last shift will be made. After this last shift SBUF and RB8 will take over the contents of the shift register and RI will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1. RI = 0
2. either SM2 = 0 or the receive stop bit = 1

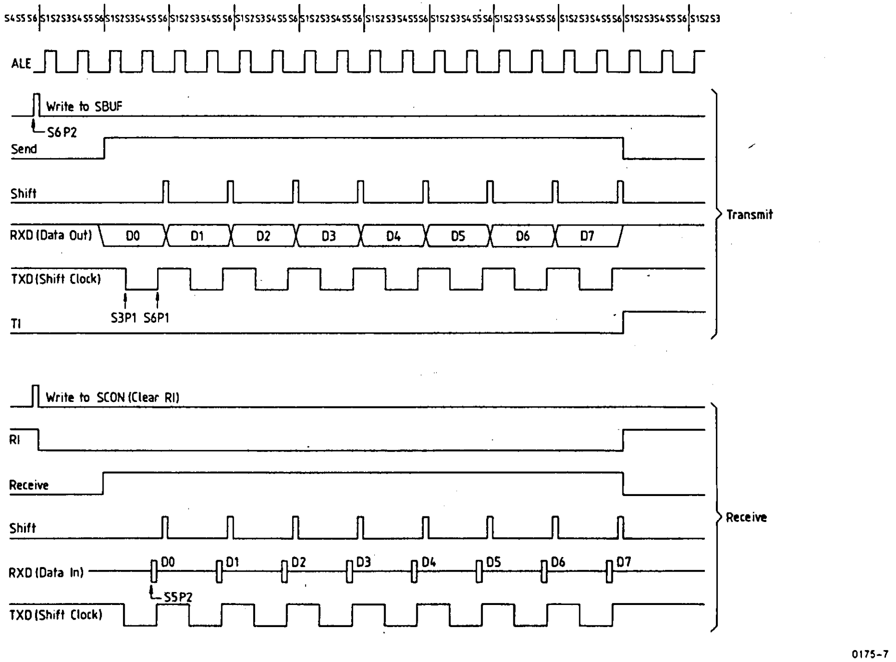

S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3 S4 S5 S6 | S1 S2 S3
ALE
Write to SBUF
S6P2
Send
Shift
RXD (Data Out)
D0
D1
D2
D3
D4
D5
D6
D7
TXD (Shift Clock)
S3P1
S6P1
TI
Transmit
Write to SCON (Clear RI)
RI
Receive
Shift
RXD (Data In)
D0
D1
D2
D3
D4
D5
D6
D7
S5P2
TXD (Shift Clock)
Receive
0175-7

Figure 3-8. Serial Port Mode 0, Timing

If none of these two conditions is met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to look for a 1-to-0 at RxD.

Mode 2, 9-Bit UART

Mode is functionally identical to mode 3 and therefore described in the next section. The only exception is that in mode 2 the baud rate can be programmed to two fixed values: either 1/32 or 1/64 of the oscillator frequency. Note that the serial interface cannot achieve these high baud rates in mode 3. The baud-rate clock in mode 3 is generated either by the additional baud rate generator or by timer 1, which is incremented by a rate of $f_{OSC/12}$ (1 MHz at 12 MHz oscillator frequency).

Mode 3, 9-Bit UART

Eleven bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmission, the 9th data bit (TB8) can be assigned the value of 0 or 1. On reception, the 9th data bit goes into RB8 in SCON. Figure 3-10 shows the associated timings for mode 3. The receive portion is exactly the same as in mode 1. The transmit portion differs from mode 1 only in the 9th bit of the transmit shift register. Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads TB8 into the 9th bit position of the internal transmit shift register. Transmission then commences at S1P1 of the machine cycle following the next rollover of the divide-by-16 counter (thus the bit times are synchronized to the divide-by-16 counter, and not to the "write-to-SBUF" signal).

After the first rollover of this counter, the start bit is passed to TxD. After the start bit is transmitted all 9 bits (including TB8) of the transmit register are shifted out. The last shift occurs at the 11th divide-by-16 rollover after "write-to-SBUF" and puts the stop bit to the pin, as well as it sets TI.

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset. The rollovers of this counter are thereby aligned with the boundaries of the incoming bits.

The 16 states of the counter divide each bit time by 16. At the 7th, 8th and 9th counter states of each bit time, the bit detector samples the value of RxD. The

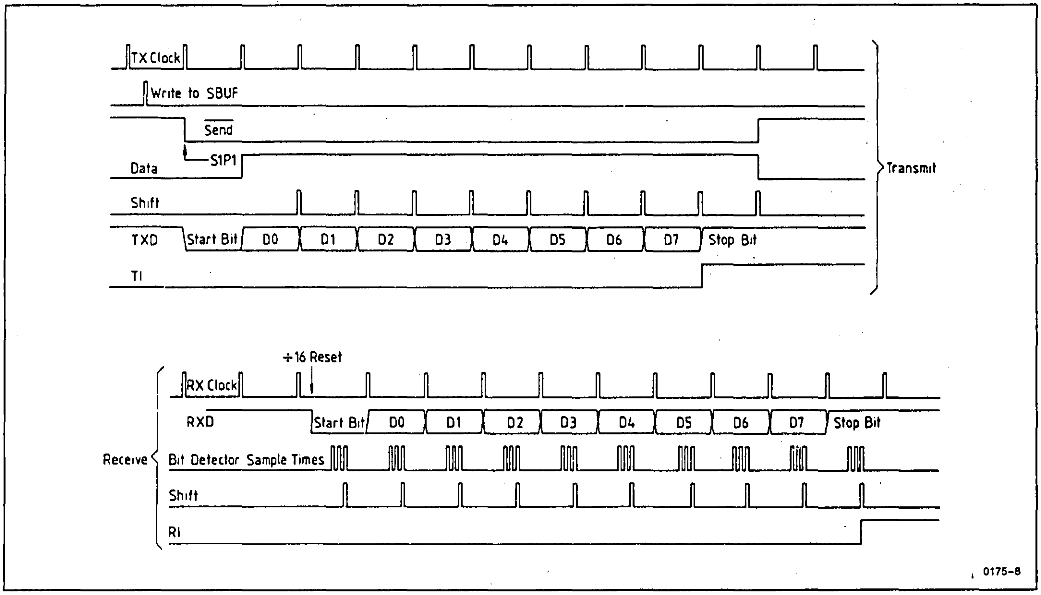


Figure 3-9. Serial Port Mode 1, Timing

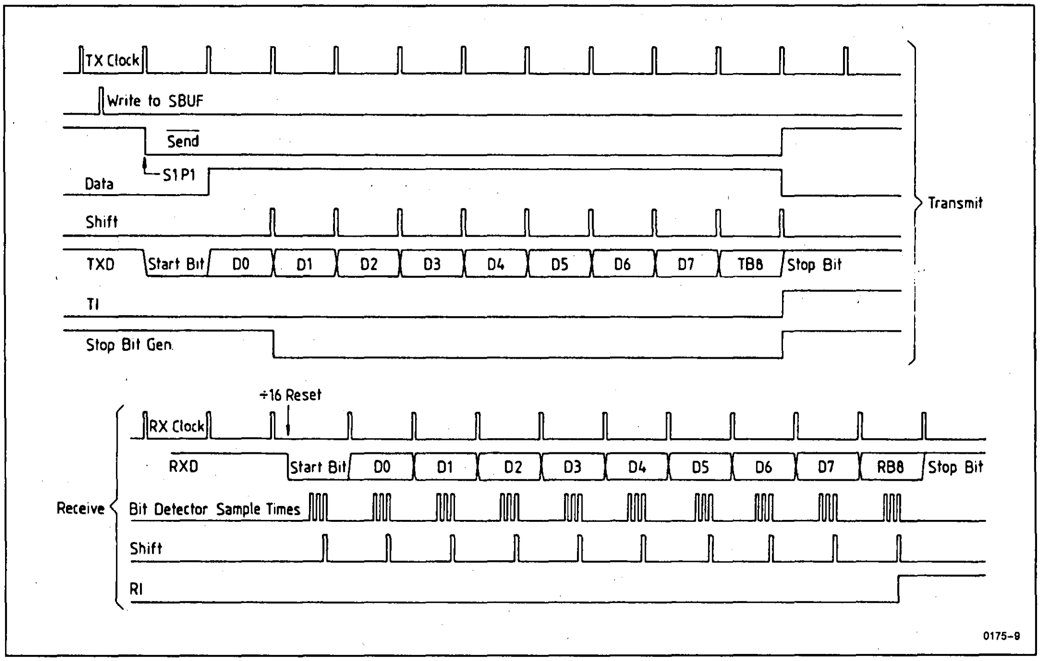


Figure 3-10. Serial Port Mode 2 and Mode 3, Timing

value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to look for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the internal input shift register, and reception of the reset of the frame will proceed.

When the start bit arrives at the last position in the 9-bit input shift register, one last shift will be made. After this last shift SBUF and RB8 will take over the contents of the shift register and RI will be generated. This will be done if, and only if, the following conditions are met at the time the final shift pulse is generated.

1. RI = 0

2. either SM2 = 0 or the received 9th data bit = 1

If none of these two conditions is met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, the first 8 data bits go into SBUF. One bit time later, regardless of whether the above conditions are met or not, the unit goes back to look for a 1-to-0 transition at the RxD input.

Note that in mode 3 the value of the received stop bit is irrelevant to SBUF, RB8, or RI.

3.3 Timer 0 and Timer 1

The SAB 80512 has two general purpose timers, Timer 0 and Timer 1. These may also be configured to operate as event counters.

In "timer" function, the register is incremented every machine cycle. Thus one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In "counter" function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 or T1. In this function the external input is sampled during S5P2 of every machine cycle. When the samples show a high level in one cycle and a low level in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected.

Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is $\frac{1}{24}$ of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

In addition to the "timer" and "counter" selection, timer 0 and timer 1 have four operating modes from which to select.

Each timer consists of two 8-bit registers (TH0 and TL0 for timer 0, TH1 and TL1 for timer 1) which may be combined to one timer configuration depending on the mode that is established. The functions of the timers are controlled by two special function registers TCON and TMOD shown in Figures 3-11 and 3-12.

In the following descriptions the symbols TH0 and TL0 are used to specify the high byte and low byte of timer 0 (TH1 and TL1 for timer 1, respectively). The operating modes are described and shown for timer 0. If not explicitly noted, this applies also to timer 1.

GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
Timer 1				Timer 0			

GATE **Gating Control:** When set, timer/counter "x" is enabled only while "$\overline{\text{INTx}}$" pin is high and "TRx" control bit is set. When cleared timer "x" is enabled whenever "TRx" control bit is set.

$C/\overline{T}$ **Counter or Timer Select Bit:** Cleared for timer operation (input from internal system clock). Set for counter operation (input from "Tx" input pin).

M1	M0	Operating Mode:
0	0	**8-Bit Timer/Counter:** "THx" operates as 8-bit timer/counter "TLx" serves as 5-bit prescaler.
0	1	**16-Bit Timer/Counter:** "THx" and "TLx" are cascaded; there is no prescaler.
1	0	**8-Bit Auto-Reload Timer/ Counter:** "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	2	**Timer 0: TL0 is an 8-bit timer/ counter controlled by the standard timer 0 control bits.** TH0 is an 8-bit timer only controlled by timer 1 control bits.
1	1	Timer 1: Timer/count 1 stops.

Figure 3-12. Timer/Counter Mode Control Register TMOD (89H)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	Bit
8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	Address

Symbol	Position	Function
IT0	TCON.0	**Interrupt 0 Type Control Bit:** Set/cleared by software to specify falling edge/low-level triggered external interrupts.
IE0	TCON.1	**Interrupt 0 Edge Flag:** Set by hardware when external interrupt edge is detected. Cleared when interrupt processed.
IT1	TCON.2	**Interrupt 1 Type Control Bit:** Set/cleared by software to specify falling edge/low-level triggered external interrupts.
IE1	TCON.3	**Interrupt 1 Edge Flag:** Set by hardware when external interrupt edge is detected. Cleared when interrupt processed.
TR0	TCON.4	**Timer 0 Run Control Bit:** Set/cleared by software to turn timer/counter 0 on/off.
TF0	TCON.5	**Timer 0 Overflow Flag:** Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	TCON.6	**Timer 1 Run Control Bit:** Set/cleared by software to turn timer/counter 1 on/off.
TF1	TCON.7	**Timer 1 Overflow Flag:** Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

Figure 3-11. Timer Control Register TCON (88H)

Mode 0

Putting either timer into mode 0 configures it as an 8-bit counter with a divide-by-32 prescaler. Figure 3-13 shows the mode 0 operation.

In this mode the timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the timer overflow flag TF0. The counted input is enabled to the timer when TR0 = 1 and either GATE = 0 or $\overline{\text{INT0}}$ = 1 (setting GATE = 1 allows the timer to be controlled by external input ($\overline{\text{INT0}}$, to facilitate pulse width measurements). TR0 is a control bit in the special function register TCON; GATE is located in TMOD.

The 13-bit register consists of all 8 bits of TH0 and the lower 5 bits of TL0. The upper 3 bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers.

Mode 0 operation is the same for timer 0 as for timer 1. Substitute the designations TR1, TF1, TH1, TL1 and $\overline{\text{INT1}}$ for the corresponding timer 1 signals in Figure 3-13. There are two different gate bits, one for timer 1 (TMOD.7) and one for timer 0 (TMOD.3).

Mode 1

Mode 1 is the same as mode 0, except that the timer register is run with all 16 bits. Mode 1 is shown in Figure 3-14.

Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in Figure 3-15. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is preset by software. The reload leaves TH0 unchanged.

Mode 3

Timer 1 in mode 3 simply holds its count. The effect is the same as with setting TR1 = 0. Timer 0 in mode 3 establishes TL0 and TH0 as two separate counters. The logic for mode 3 on timer 0 is shown in Figure 3-16. TL0 uses the timer 0 control bits: $\overline{\text{C/T}}$, GATE, TR0, $\overline{\text{INT0}}$, and TF0. TH0 is locked into

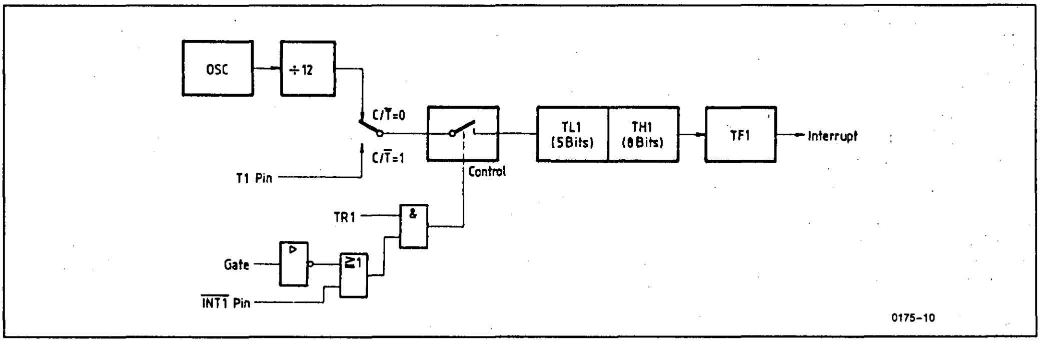

OSC
÷12
C/$\overline{\text{T}}$=0
C/$\overline{\text{T}}$=1
T1 Pin
Control
TL1 (5Bits)
TH1 (8Bits)
TF1
Interrupt
TR1
&
Gate
≥1
$\overline{\text{INT1}}$ Pin
0175-10

Figure 3-13. Timer/Counter 0/1, Mode 0: 13-Bit Counter

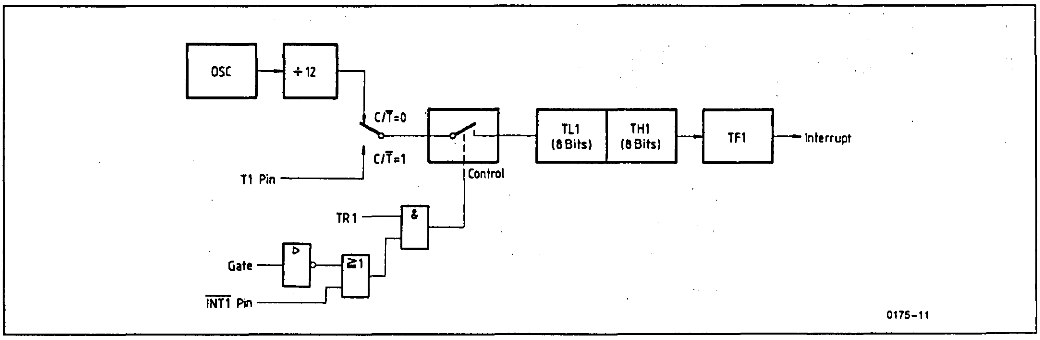

OSC
÷12
C/$\overline{\text{T}}$=0
C/$\overline{\text{T}}$=1
T1 Pin
Control
TL1 (8Bits)
TH1 (8Bits)
TF1
Interrupt
TR1
&
Gate
≥1
$\overline{\text{INT1}}$ Pin
0175-11

Figure 3-14. Timer/Counter 0/1, Mode 1: 16-Bit Counter

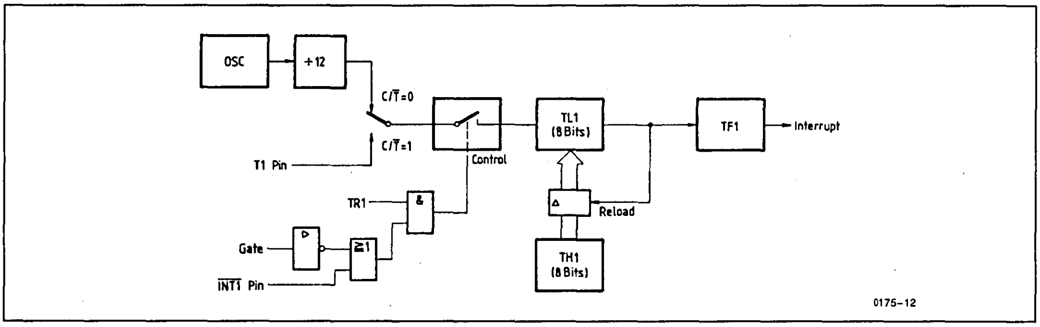


Figure 3-15. Timer/Counter 0/1 Mode 2: 8-Bit Auto-Reload

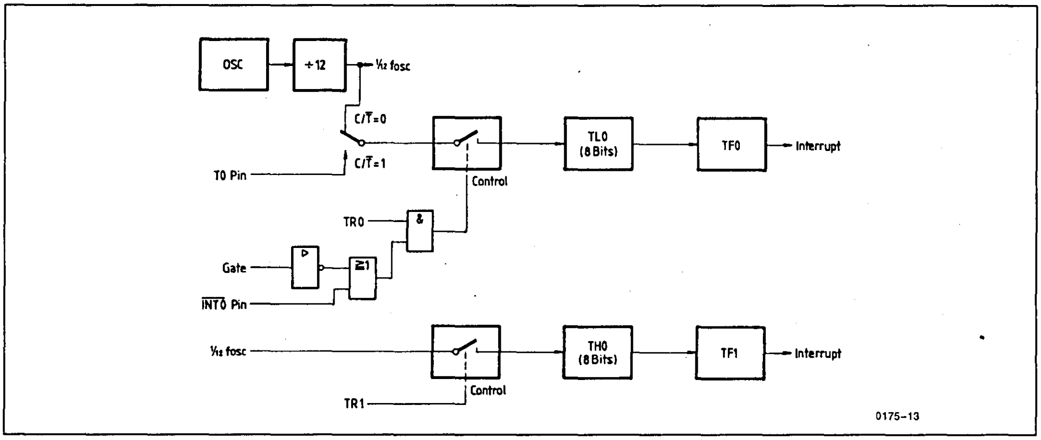


Figure 3-16. Timer/Counter 0 Mode 3: Two 8-Bit Counters

a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from timer 1. TH0 now controls the "timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer or counter. With timer 0 in mode 3, an SAB 80512 may seem to have three timer/counters. When timer 0 is in mode 3, timer 1 can be turned on and off by switching it out of and into its own mode 3, or can still be used by the serial channel as a baud-rate generator or, in fact, in any application not requiring an interrupt.

3.4 A/D Converter

The SAB 80512 provides an A/D converter with the following features:

- 8 multiplexed input channels, which can also be used as digital inputs (Port 6)
- External reference voltages adjustable in a wide range
- 8-bit resolution **within** the selected reference voltage range
- 15 μs conversion time (including sample time)
- Interrupt request generation after each conversion

For the conversion, the method of successive approximation via a capacitor network is used. The externally applied reference voltages can be varied to reduce the reference voltage range of the A/D converter and therefore to achieve higher resolution.

Figure 3-17 shows a block diagram of the A/D converter. There are three user-accessible special-function registers: ADCON (A/D converter control register), ADDAT (A/D converter data register) and DAPR (A/D converter start register).

3.4.1 Function and Control

Initialization

Special-function register ADCON, which is illustrated in Figure 3-18, is used to set the operating modes, to

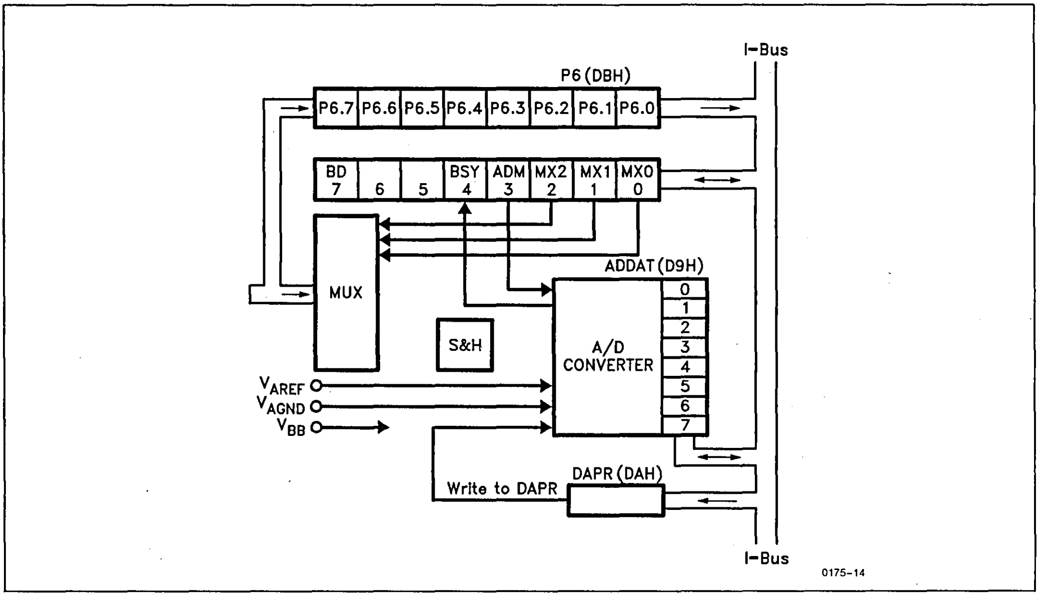


Figure 3-17. A/D Converter Control Register

check the status, and to select one of the eight analog input channels at port 6.

Two operating modes are supported by means of bit ADM (ADCON.3) which is used to select single or continuous conversion mode. As long as ADM is set the converter starts a new conversion directly after the previous one has been completed. The three-bit field consisting of bits MX0 to MX2 (ADCON.0 to ADCON.2) contains the binary coded information to select one of the 8 analog input channels (see Table 3-5). Bit BSY (ADCON.3) indicates the A/D converter status. If it is a one (1), a conversion is in progress. For details see Section 3.4.2.

Start of Conversion

Start of conversion is triggered by a write-to-DAPR instruction. The start procedure is triggered by the write operation itself and does not depend on the value written. Reading the DAPR register has no effect on the conversion and determines no result.

3.4.2 External Reference Voltages

The A/D converter in the SAB 80512 allows for external adjustment of the reference voltages. Thus the voltage at the pins VAREF/VAGND, which are the input pins for the upper/lower reference voltage, can be varied in a wide range. This feature may be used to adjust the reference voltage range to the range of the expected analog input. A compressed reference voltage range affords increased resolution of the converted analog input.

The lower reference voltage (VAGND) can be varied with V_{SS} − 0.2V and VAREF − 1V, and the upper one (VAREF) within VAGND + 1V and V_{CC} + 5%. Thus for proper operation of the A/D converter, a minimum of 1V difference is required between the external voltages:

$(V_{SS} - 0.2V) \leq VAGND \leq (VAREF - 1V)$
$(VAGND + 1V) \leq VAREF \leq (V_{CC} + 5\%)$

Table 3-5. Selection of the Analog Input Channels

MX2	MX1	MX0	Selected Channel	Pin
0	0	0	Analog Input 0	P6.0
0	0	1	Analog Input 1	P6.1
0	1	0	Analog Input 2	P6.2
0	1	1	Analog Input 3	P6.3
1	0	0	Analog Input 4	P6.4
1	0	1	Analog Input 5	P6.5
1	1	0	Analog Input 6	P6.6
1	1	1	Analog Input 7	P6.7

For example, if the external voltage range extends from VAGND = 0V to VAREF = 5V, the resulting resolution is approximately 20 mV per digital step. A minimum voltage range of 1V (e.g. VAREF = 3V, VAGND = 2V) results in approximately 4 mV resolution.

Note that the errors of the A/D converter (specified in the Appendix) refer to a reference voltage range of 5V (VAGND = V_{SS}, VAREF = V_{CC}).

Reducing the reference voltage range does increase the resolution but not the accuracy of the 8-bit ADC, hence, the specified errors apply to any reference voltage range chosen.

3.4.3 A/D Converter Timing and Conversion Time

A conversion is started by writing into special-function register DAPR. A write-to-DAPR will start a new conversion even if a conversion is currently in progress. The conversion begins with the next machine cycle in which the "MOVE-TO-DAPR" has been completed. The busy flag will be set in the same cycle as the write-to-DAPR operation occurs.

The conversion needs 15 machine cycles to be completed. This equates to a conversion time of 15 μs at 12 MHz oscillator frequency.

If a continuous conversion is established, the next conversion is automatically started in the following machine cycle. After a conversion has been started by writing into SFR DAPR, the analog voltage at the selected input channel is sampled for the duration of 5 machine cycles (5 μs at 12 MHz oscillator frequency), which will then be internally held at the sampled level for the rest of the conversion time. The external analog source has to be strong enough to charge the internal sample hold capacitance, which is 70 pF at a maximum, throughout the first three cycles of the sample time. The requirements on the internal resistance of the analog source can be derived from this time, which is specified as Load Time (TL) in the Appendix.

Conversion of the sampled analog voltage takes place during the 6th to 15th machine cycle after being started. In the 15th machine cycle the converted result is moved to ADDAT and the busy flag (BSY) is cleared. The A/D converter interrupt is generated by bit IADC in register IRCON and is set at the end of the 11th machine cycle after a conversion has been started. This means that if an interrupt is initiated, the converted result is ready for access at the same time the first instruction of the interrupt service routine is executed, provided the interrupt can be serviced after the minimally possible response time.

3.5 RAM Backup Power Supply

The power down mode in the SAB 80512 enables the reduction of V_{CC} to zero while saving 40 bytes of the on-chip RAM through a backup supply connected to the VPD pin. In the following description, the terms V_{CC} and V_{PD} are used to specify the voltages at pin V_{CC} and pin V_{PD}.

BD	—	—	BSY	ADM	MX2	MX1	MX0	Bit
0DFH	0DEH	0DDH	0DCH	0DBH	0DAH	0D9H	0D8H	Address

Symbol	Position	Function
MX0	ADCON.0	Analog Input Channel Select, See Table 3-5
MX1	ADCON.1	Analog Input Channel Select, See Table 3-5
MX2	ADCON.2	Analog Input Channel Select, See Table 3-5
ADM	ADCON.3	A/D Conversion Mode. When set, a continuous conversion is selected. If ADM = 0, the converter stops after one conversion.
BSY	ADCON.4	Busy Flag. This flag indicates whether a conversion is in progress (BSY = 1) or not (BSY = 0).
—	ADCON.5	Reserved
—	ADCON.6	Reserved
BD	ADCON.7	Baud Rate Enable. When set, the baud rate in mode 1 and 3 of the serial channel is taken from the internal baud rate generator (see Section 3.2).

Figure 3-18. A/D Converter Control Register ADCON (0D8H)

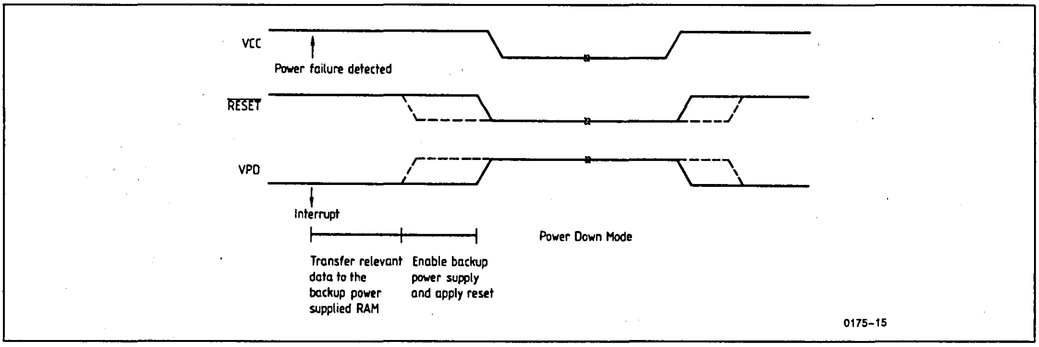

VCC
Power failure detected
RESET
VPD
Interrupt
Power Down Mode
Transfer relevant data to the backup power supplied RAM
Enable backup power supply and apply reset
0175-15

Figure 3-19. Reset and RAM Backup Power Timing

If $V_{CC} > V_{PD}$, the 40 bytes are supplied from V_{CC}. In this case V_{PD} may be low or at any voltage less than V_{CC}.

If $V_{CC} < V_{PD}$, the current for the 40 bytes is drawn from V_{PD}. It is also permissible to hold V_{PD} equal to or higher than V_{CC} during normal operation.

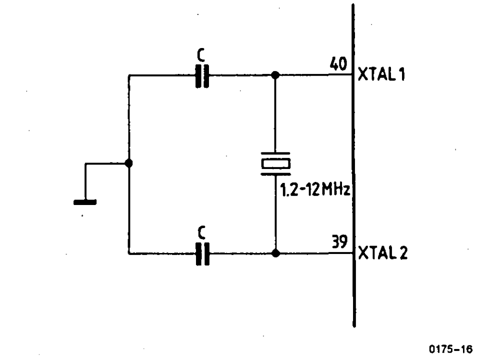
The addresses of these backup powered RAM locations range from 88 to 127 (58H to 7FH). The current drawn from the backup power supply is specified in the Appendix.

Note that the user must take provisions to preserve necessary machine state conditions. He also has to ensure that an external reset is applied to the chip in time, as shown in Figure 3-19. This is to be done to ensure that the controller does not continue operating when V_{CC} drops below specs.

Thus, to use this feature, the user's system—upon detection that a power failure is imminent—would interrupt the processor in some manner to transfer relevant data to the 40 bytes in on-chip RAM and enable the backup power supply to the V_{PD} pin. Then a reset should be executed before V_{CC} falls below its operation limit. When power returns, a power-on reset should be made, and the backup supply needs to stay on long enough so that normal operation can be resumed. Figure 3-19 illustrates the timing of a power failure.

3.6 Oscillator and Clock Circuit

XTAL1 and XTAL2 are the input and output of a single-stage on-chip inverter which can be configured with off-chip components as a Pierce oscillator. The oscillator, in any case, drives the internal clock generator. The clock generator provides the internal clocking signals to the chip. These signals are at half the oscillator frequency and define the internal phases, states and machine cycles, as described in Section 2.1.1. Figure 3-20 shows the recommended oscillator circuit.



C = 30 pF ±10 pF for Quartz Crystal

Figure 3-20. Recommended Oscillator Circuit

In this application the on-chip oscillator is used as a crystal-controlled, positive reactance oscillator (a more detailed schematic is given in Figure 3-21). It is operated in its fundamental response mode as an inductive reactance in parallel resonance with a capacitance external to the chip. The crystal specifications and capacitance values are not critical. 30 pF can be used in these positions at any frequency with a good quality crystal. A ceramic resonator can be used in place of the crystal in cost-critical applications. When a ceramic resonator is used, C is normally selected to be of somewhat higher values, typically 47 pF. The manufacturer of the ceramic resonator should be consulted for recommendations on the values of these capacitors.

To drive the SAB 80512 with an external clock source, apply the external clock signal to XTAL2 and ground to XTAL1, as shown in Figure 3-22. A pullup resistor is suggested (to increase noise margin), but is optional if V_{OH} of the driving gate fits the V_{IH2} specification of XTAL2.

Sometimes an external clock with the frequency of the oscillator is needed. For this application the circuit shown in Figure 3-23 is recommended. The CMOS driver (or inverter) should be placed as closely as possible to the oscillator circuit. Be sure to take into account the impedances of the circuit and the CMOS driver input.

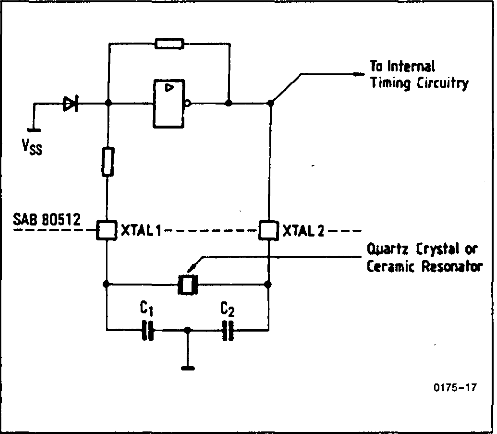

To Internal Timing Circuitry
V_{SS}
SAB 80512
XTAL 1
XTAL 2
Quartz Crystal or Ceramic Resonator
C_1
C_2
0175-17

Figure 3-21. On-Chip Oscillator Circuitry

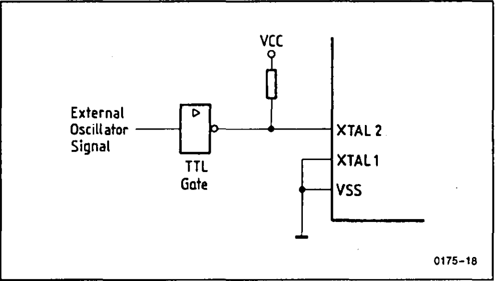


Figure 3-22. Driving with an External Clock Source

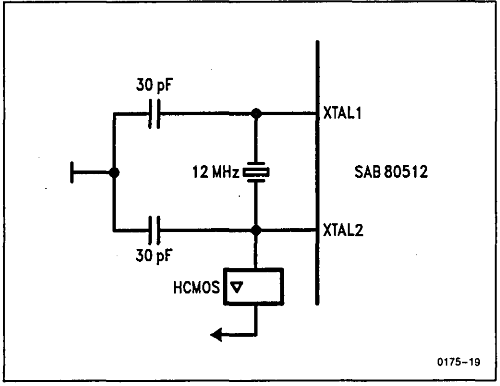


Figure 3-23. Generating a System Clock from the Oscillator Circuit

Interrupt System

4.0 Interrupt System

The SAB 80512 provides six interrupt sources. Four interrupts can be generated by the on-chip peripherals (i.e., Timer 0, Timer 1, Serial Channel and A/D Converter), and two interrupts may be triggered externally.

The interrupt structure of the SAB 80512 has been mainly adapted from the 8051. Thus, each interrupt source can be enabled individually and can be set on one of two priority levels.

Figure 4-1 gives a general view of all interrupt sources and illustrates the request and control flags described in the next sections.

4.1 Interrupt Structure

A common mechanism is used to generate the various interrupts whereby each source has its own request flag located in a special-function register (e.g., TCON, IRCON, SCON). Provided the peripheral or external source meets the condition for an interrupt, the dedicated request flag is set, whether an interrupt is enabled or not. For example, each timer 0 overflow sets the corresponding request flag TF0. If it is already set, it retains a one (1). But the interrupt is not necessarily serviced.

Now each interrupt requested by the corresponding flag can be enabled or disabled individually by the enable bits in SFR IE (see Figure 4-2). This determines whether the interrupt will actually be performed. In the following section the interrupt sources are discussed separately.

The external interrupts 0 and 1 ($\overline{\text{INT0}}$ and $\overline{\text{INT1}}$) can each be either level-activated or negative transition-activated, depending on bits IT0 and IT1 in register TCON (see Figure 3-11). The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated this interrupt is cleared by the hardware when the service routine is vectored to, only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source directly controls the request flag, rather than the on-chip hardware.

The timer 0 and timer 1 interrupts are generated by TF0 and TF1, which are set by a rollover in their respective timer/counter registers (exception see Section 3.3 for timer 0 in mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

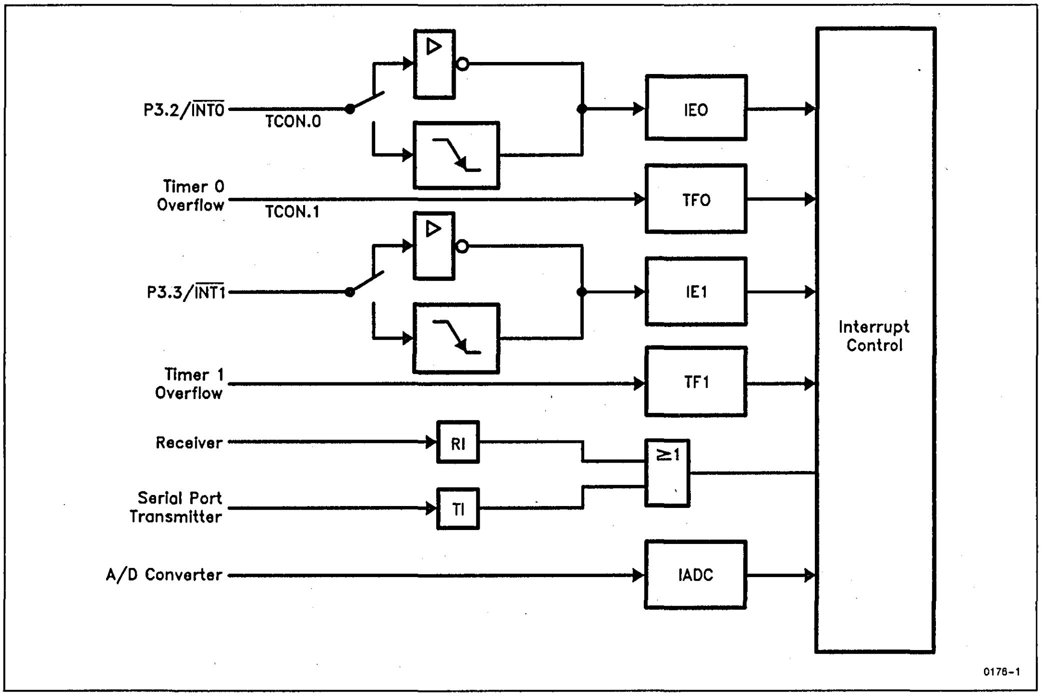


Figure 4-1. Interrupt Requesting Sources

The serial channel interrupt is generated by the request flags RI and TI. That is, the two request flags are logically ORed together. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine of the interface will normally have to determine whether it was the receive interrupt flag (RI) or the transmission interrupt flag (TI) that generated the interrupt, and the flag will have to be cleared by software.

The A/D Converter interrupt is generated by IADC in register IRCON (Bit IRCON.0 see Figure 4-3) and is set on the 11th machine cycle after a conversion has been started. That is, if an interrupt is generated, in any case the converted result in ADDAT is valid on the first instruction of the interrupt service routine (11 cycles plus 4 cycles minimal interrupt response time results in 15 cycles; the time the A/D Converter requires for one conversion). If the continuous conversion mode is established, IADC is set in the 11th machine cycle of the last conversion to be completed. If an A/D Converter interrupt is generated, flag IADC will have to be cleared by software.

EA	EADC	—	ES	ET1	EX1	ET0	EX0	Bit
0AFH	0AEH	0ADH	0ACH	0ABH	0AAH	0A9H	0A8H	Address

Symbol	Position	Function
EX0	IE.0	Enables or disables external interrupt 0. If EX0 = 0, external interrupt 0 is enabled.
ET0	IE.1	Enables or disables the timer 0 overflow interrupt. If ET0 = 0, the timer 0 interrupt is disabled.
EX1	IE.2	Enables or disables external interrupt 1. If EX1 = 0, external interrupt 1 is disabled.
ET1	IE.3	Enables or disables the timer 1 overflow interrupt. If ET1 = 0, the timer 1 interrupt is disabled.
ES	IE.4	Enables or disables the serial port interrupt. If ES = 0, the serial port interrupt is disabled.
—	IE.5	Reserved
EADC	IE.6	Enables or disables A/D Converter Interrupt. If EADC = 0, the A/D Converter Interrupt is disabled.
EA	IE.7	Enables or disables all interrupts. If EA = 0, no interrupt is will be acknowledged. If EA = 1, eachj interrupt source is individually enabled or disabled by setting or clearing its enable bit.

Figure 4-2. Interrupt Enable Register IE (0A8H)

—	—	—	—	—	—	—	IADC	Bit
0C7H	0C6H	0C5H	0C4H	0C3H	0C2H	0C1H	0C0H	Address

Symbol	Position	Function
IADC	IRCON.0	A/D Converter interrupt request flag. Set by hardware in the 11th cycle of a conversion. Must be cleared by software.
	IRCON.1	Reserved
	IRCON.2	Reserved
	IRCON.3	Reserved
	IRCON.4	Reserved
	IRCON.5	Reserved
	IRCON.6	Reserved
	IRCON.7	Reserved

Figure 4-3. Interrupt Request Control Register IRCON (0C0H)

All of the bits that generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software. The only exceptions are request flags IE0 and IE1. If the external interrupts 0 and 1 are programmed to be level-activated, IE0 and IE1 are controlled by the external source via pin $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, respectively. In this case, writing a one (1) to the request flag IE0 and/or IE1 will have no effect. In this mode, interrupts 0 and 1 can only be generated in software by writing a 0 to the corresponding pins $\overline{\text{INT0}}$ (P3.2) and $\overline{\text{INT1}}$ (P3.3), provided this will not affect any peripheral circuit connected to the pins.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in the special function registers IE (Figure 4-2). Note that IE contains also a global disable bit, EA, which disables all interrupts at once.

4.2 Priority Level Structure

Each interrupt source can be programmed individually to one of two priority levels by setting or clearing a bit in the special-function register IP (Figure 4-4).

An interrupt can itself be interrupted by an interrupt with higher priority, but not by another interrupt of the same or a lower priority. Thus a high-priority interrupt cannot be interrupted by any other interrupt source.

If two or more requests of different priority levels are received simultaneously, the request of the highest priority is serviced first. If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced first. Thus within each priority level there is a second priority structure determined by the polling sequence, as follows:

Source	Priority within Level
1. IE0	(Highest)
2. TF0	
3. IE1	
4. TF1	
5. RI + TI	
6. IADC	(Lowest)

Note that this "priority within level" structure is only used to resolve **simultaneous requests of the same priority level.**

—	—	PADC	PS	PT1	PX1	PT0	PX0	Bit
0BFH	0BEH	0BDH	0BCH	0BBH	0BAH	0B9H	0B8H	Address

Symbol	Position	Function
—	IP.7	Reserved
PADC	IP.6	Reserved
PS	IP.5	Defines the ADC interrupt priority level. PADC = 1 programs it to the higher level.
PT1	IP.4	Defines the serial interrupt priority level. PS = 1 programs it to the higher level.
PX1	IP.3	Defines the timer 1 interrupt priority level. PT1 = 1 programs it to the higher level.
PT0	IP.2	Defines the external interrupt 1 priority level. PX1 = 1 programs it to the higher level.
PX0	IP.1	Defines the timer 0 interrupt priority level. PT0 = 1 programs it to the higher level.
	IP.0	Defines the external interrupt 0 priority level. PX0 = 1 programs it to the higher level.

Figure 4-4. Interrupt Priority Register IP (0B8H)

4.3 How Interrupts Are Handled

The interrupt flags are sampled at S5P2 in every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions.

1) An interrupt of equal or higher priority is already in progress.
2) The current (polling) cycle is not in the final cycle in the execution of the instruction in progress.
3) The instruction in progress is RETI or any write access to registers IE or IP.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any write access to registers IE or IP, then at least one more instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note then that if any interrupt flag is active but not being responded to for one of the above conditions, and if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Each polling cycle interrogates only the pending interrupt requests.

The polling cycle/LCALL sequence is illustrated in Figure 4-5.

Note that if an interrupt of higher priority level goes active prior to S5P2 in the machine cycle labeled C3 in Figure 4-5, then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine being executed.

Thus the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it does not. It never clears the serial port (RI, TI) or A/D Converter flag (IADC). This has to be done in the user's software. It clears an external interrupt flag (IE0 or IE1) only if it was transition-activated. The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to, as shown below.

Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI + TI	0023H
IADC	002BH

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from where it left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress.

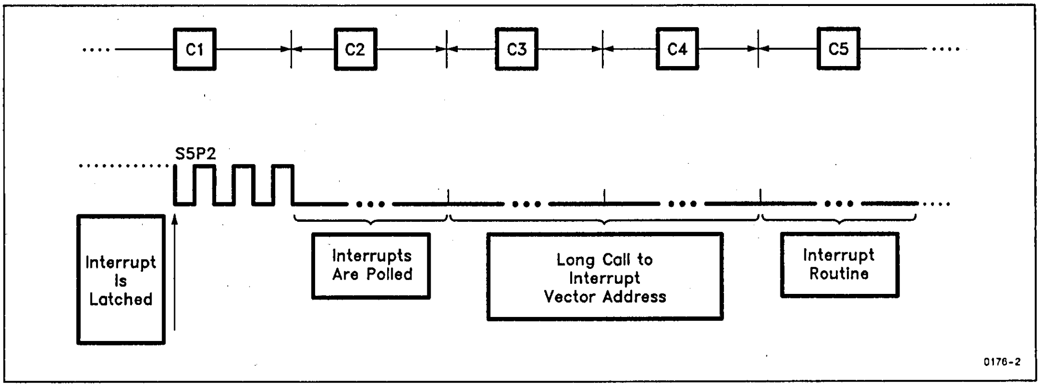
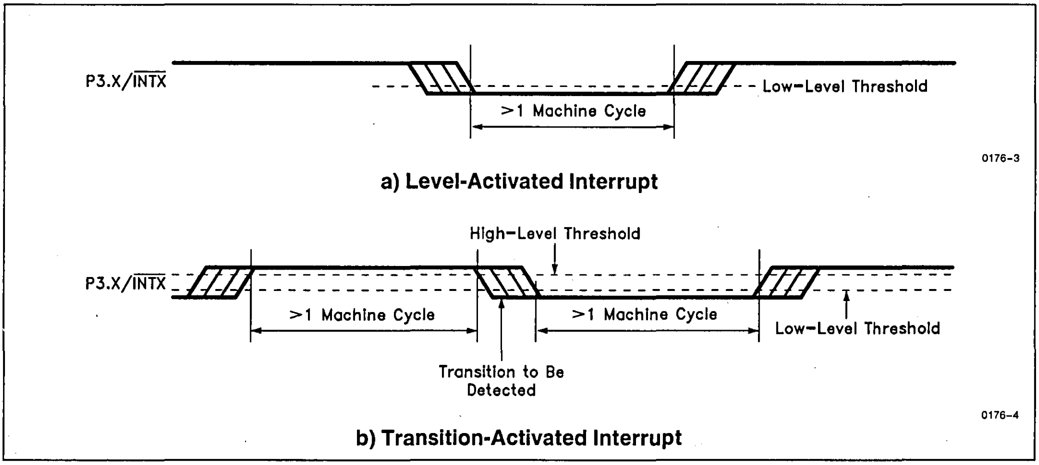


Figure 4-5. Interrupt Response Timing Diagram



a) Level-Activated Interrupt

b) Transition-Activated Interrupt

Figure 4-6. External Interrupt Detection

4.4 External Interrupts

The external interrupts 0 and 1 can be programmed to be level-activated or negative transition-activated by setting or clearing bit IT0 or IT1, respectively, in register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the $\overline{\text{INTx}}$ pin. If ITx = 1, external interrupt x is negative edge-triggered. In this mode, if successive samples of the $\overline{\text{INTx}}$ pin show a high level in one cycle and a low level in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

Since the external interrupt pins are sampled once in each machine cycle, an input high or low should be held for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle to ensure that the transition is recognized so that the corresponding interrupt request flag will be set. The external interrupt request flag will automatically be cleared by the CPU when the service routine is called.

If the external interrupt 0 or 1 is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, otherwise another interrupt will be generated.

Note that once a level-activated interrupt 0 or 1 is removed, it is not remembered by the controller if it was not directly serviced.

However, if the interrupt is transition-activated, a detected transition sets the request flag and this flag remains set until it is cleared by software or by hardware after servicing the interrupt. Thus an interrupt request triggered by a negative edge is remembered for any period of time.

4.5 Response Time

If an external interrupt is recognized, its corresponding request flag is set at S5P2 in every machine cycle. The value is not actually polled by the circuitry until the next machine cycle. If the request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles. Thus a minimum of three complete machine cycles will elapse between activation and external interrupt request and the beginning of execution of the first instruction of the service routine. Figure 4-5 shows the interrupt response timings.

A longer response time would result if the request is blocked by one of the three previously listed conditions. If an interrupt of equal or higher priority is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles since the longest instructions (MUL and DIV) are only 4 cycles long; and, if the instruction in progress is RETI or an access to registers IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction, if the instruction is MUL or DIV).

Thus, in a single interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

SAB 80515/80535 Single-Chip Microcontroller User's Manual

SIEMENS

Microcomputer Components

SAB 80515/80535
Single-Chip Microcontroller

User's Manual

Contents

Table of Contents

Introduction

1.0 Introduction

The SAB 80515 is the newest member of the Siemens SAB 8051 8-bit microcontroller family, based on the SAB 8051 architecture. While maintaining all the SAB 8051's operating characteristics, the SAB 80515 incorporates several enhancements which significantly increase design flexibility and overall system performance.

SAB 80515 features are:

- — 8 Kybte on-chip program memory
- — 256 byte on-chip RAM
- — Six 8-bit parallel ports
- — Full-duplex serial port, 4 modes of operation, fixed or variable baud rates
- — Three 16-bit timer/counters
- — 16-bit reload, compare, capture capability
- — A/D converter, 8 multiplexed analog inputs, programmable reference voltages
- — 16-bit watchdog timer
- — Power-down supply for 40 byte of RAM
- — Boolean processor
- — 256 directly addressable bits
- — 12 interrupt sources (7 external, 5 internal), 4 priority levels
- — Stack depth up to 256 byte
- — 1 µs instruction cycle at 12 MHz operation
- — 4 µs multiply and divide
- — External program and data memory expandable up to 64 Kbyte each
- — Compatible with standard SAB 8080/8085 peripherals and memories
- — Space-saving PLCC-68 package

The SAB 80535 is the ROMless version of the SAB 80515.

In this manual, the term 'SAB 80515' is used to refer to both the SAB 80515 and the SAB 80535, unless otherwise noted.

Figure 1 shows the logic symbol, Figure 2 a block diagram of the SAB 80515.

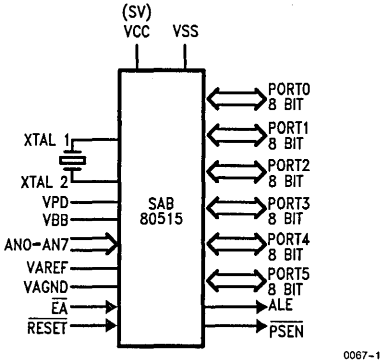


Figure 1. Logic Symbol

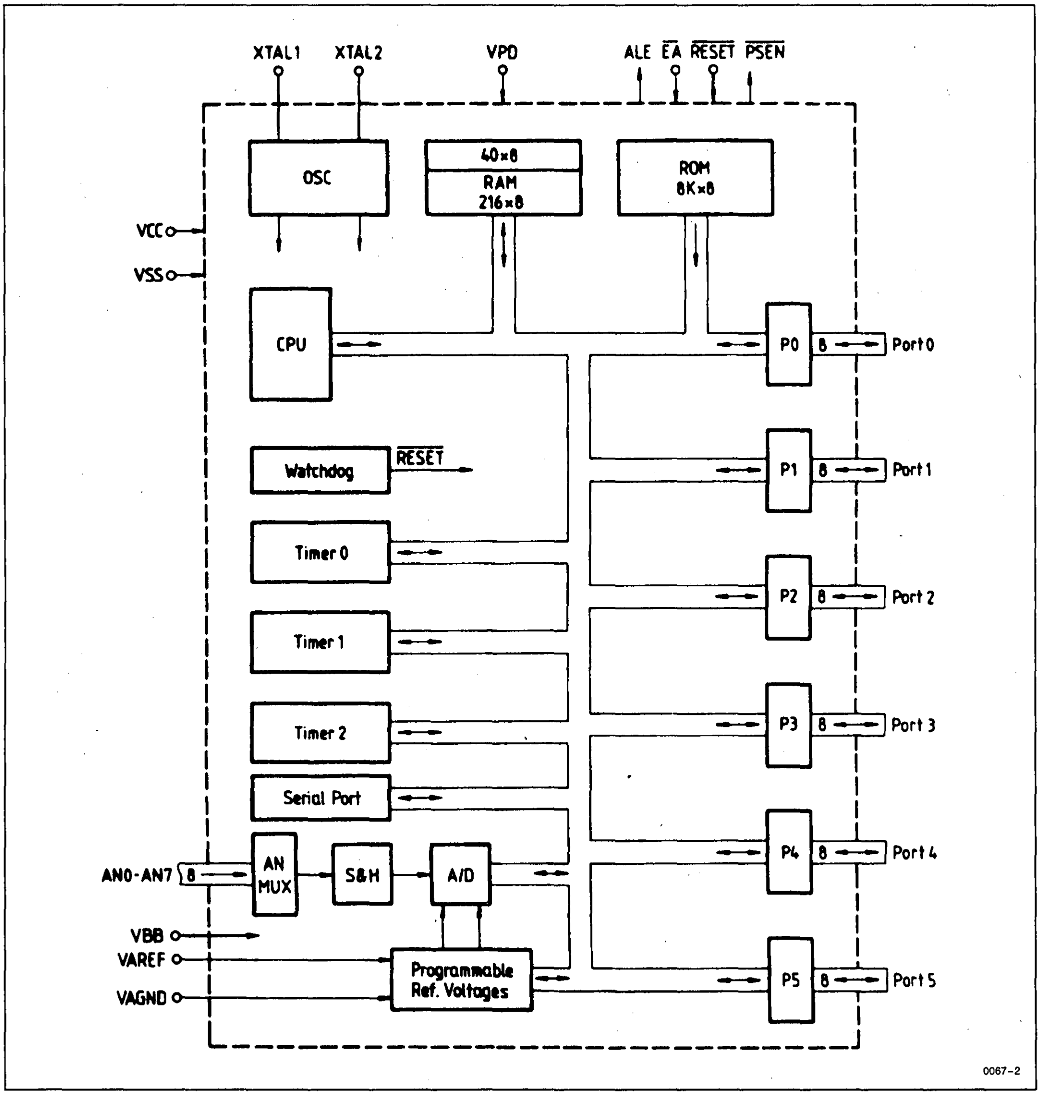


Figure 2. Block Diagram

Architecture

2.0 Architecture

The SAB 80515 is classified as being an 8-bit machine since the internal ROM, RAM, special function registers, arithmetic/logic unit and external data bus are each 8 bits wide. The SAB 80515 supports bit, nibble, byte and double-byte operations. The SAB 80515 has extensive facilities for byte transfer, logic and integer arithmetic operations. It excels at bit-handling capability since data transfer, logic and conditional branch operations can be performed directly on Boolean variables.

2.1 CPU

The CPU (Central Processing Unit) of the SAB 80515 consists of the instruction decoder, the arithmetic section, and the program control section. Each program instruction is decoded by the instruction decoder. This unit generates the internal signals controlling the functions of the individual units within the CPU. They have an effect on the source and destination of data transfers, and control the ALU processing.

The arithmetic section of the processor performs extensive data manipulation and is comprised of the arithmetic/logic unit (ALU), an A register, B register and PSW register. The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under the control of the instruction decoder. The ALU performs the arithmetic operations add, subtract, multiply, divide, increment, decrement, BCD-decimal-add-adjust and compare, and the logic operations AND, OR, Exclusive OR, complement and rotate [right, left, or swap nibble (left four)]. Also included is a Boolean processor performing the bit operations of set, clear, complement, jump-if-set, jump-if-not-set, jump-if-set-and-clear and move to/from carry. Between any addressable bit (or its complement) and the carry flag it can perform the bit operations of logical AND or logical OR with the result returned to the carry flag. The A, B and PSW registers are described in a later section.

The program control section controls the sequence in which the instructions stored in program memory are executed. The 16-bit program counter (PC) holds the address of the next instruction to be executed. The PC is manipulated by the control transfer instructions listed in section 5.1.4. The conditional branch logic enables events internal and external to the processor to cause a change in the program execution sequence.

2.2 Memory Organization

The SAB 80515 CPU manipulates operands in the following four memory address spaces:

— up to 64 kbytes of program memory

— up to 64 kbytes of external data memory

— 256 bytes of internal data memory

— a 128-byte special function register area

2.2.1 Program Memory

The program memory address space of the SAB 80515 consists of an internal and an external memory portion. The SAB 80515 has 8 kbytes of program memory on-chip, while the SAB 80535 has no internal ROM. The program memory can be externally expanded up to 64 kbytes. If the $\overline{EA}$ pin is held high, the SAB 80515 executes out of the internal program memory unless the address exceeds 1FFFH. Locations 2000H through 0FFFFH are then fetched from the external program memory. If the $\overline{EA}$ pin is held low, the SAB 80515 fetches all instructions from the external program memory. Since the SAB 80535 has no internal program memory, pin $\overline{EA}$ must be tied low when using this device. In either case, the 16-bit program counter is the addressing mechanism.

Locations 00 through 6BH in the program memory are used by interrupt service routines.

Figure 4 illustrates the program memory address space.

2.2.2 Data Memory

The data memory address space consists of an internal and an external memory space. The internal data memory is divided into three physically separate and distinct blocks: the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128-byte special function register (SFR) area. While the upper RAM area and the SFR area share the same address locations, they are accessed through different addressing modes. These modes are discussed in section 4.3.

Figure 4 shows a mapping of the internal data memory. Four 8-register banks occupy locations 0 through 31 in the lower RAM area. Only one of these banks may be enabled at a time (through a two-bit field in the PSW, see description of PSW). The next 16 bytes, locations 32 through 47, contain 128 directly addressable bit locations. The stack can be located anywhere in the internal data memory address space. The stack depth is only limited by the internal RAM available (256 byte maximum). The 64 kbyte external data memory area can be accessed by instructions using a 16-bit or an 8-bit address.

The special function register area is described in the next section.

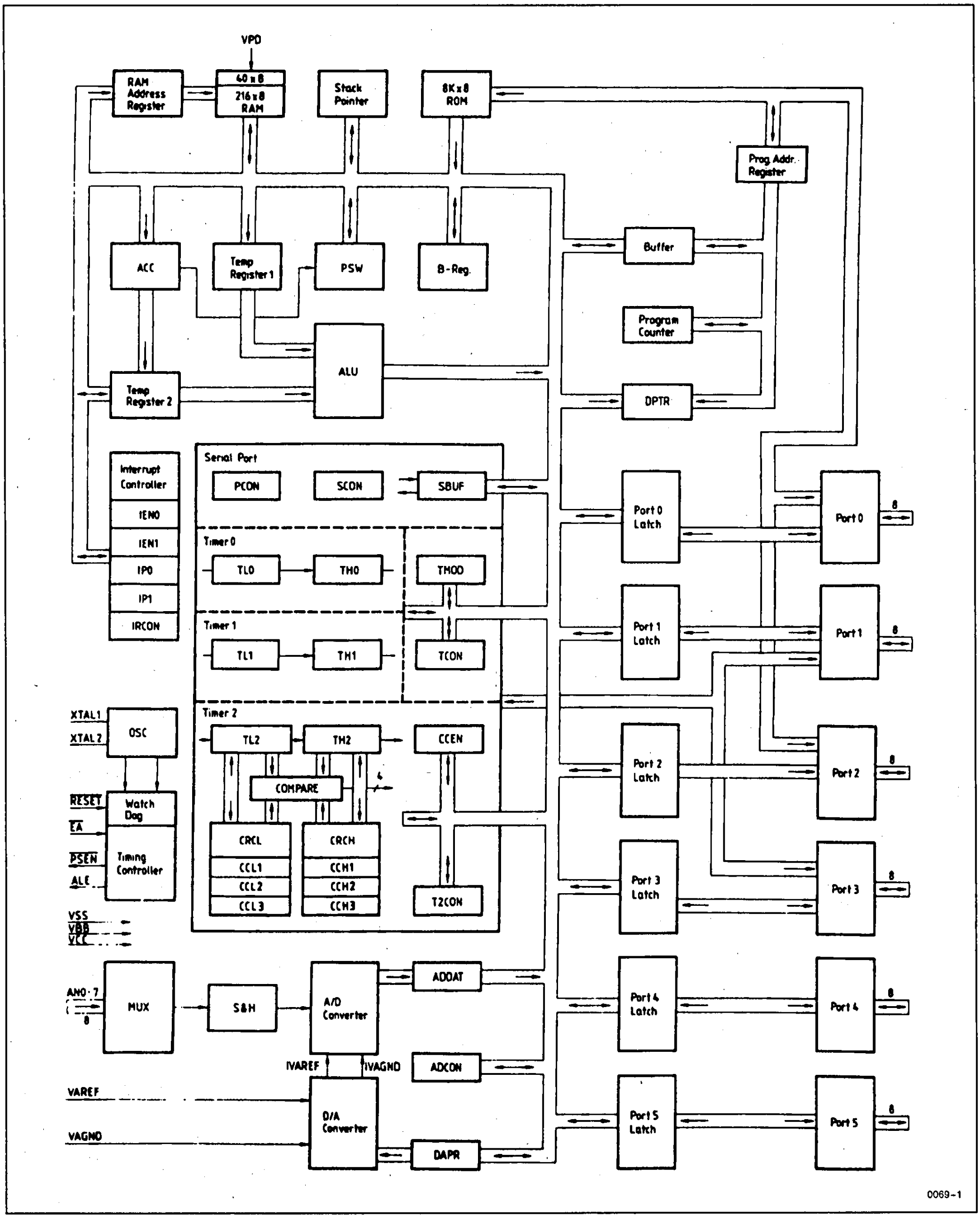


Figure 3. Detailed Block Diagram

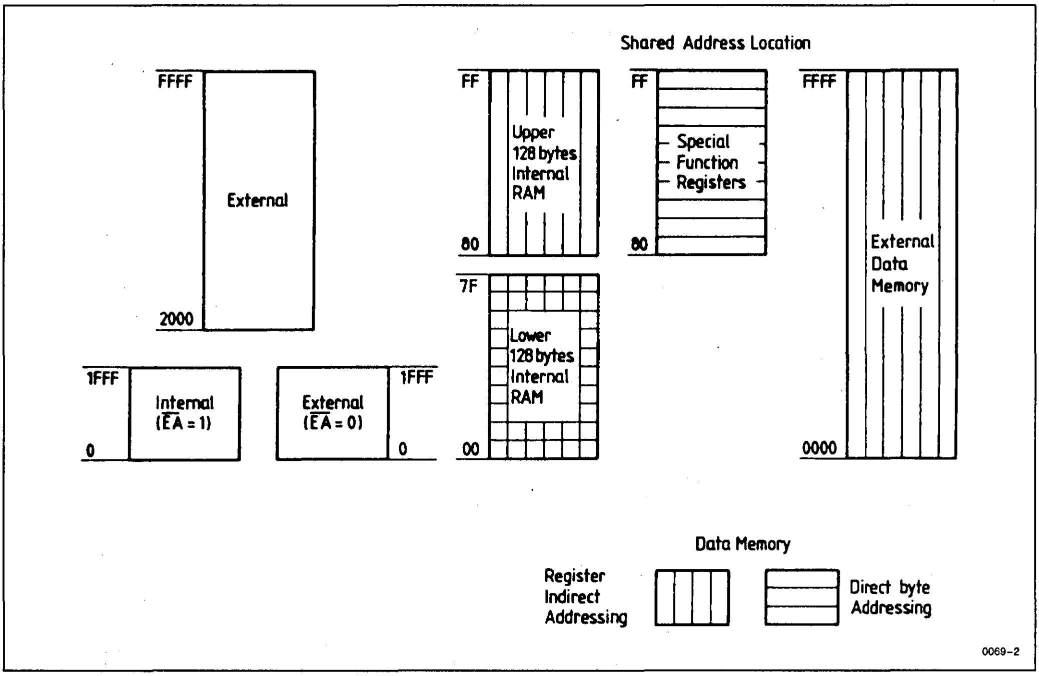


Figure 4. Program and Internal Data Memory Address Spaces

2.2.3 Special Function Registers

The address space of the special function registers is comprised of locations 128 through 255. All registers except the program counter and the four 8-register banks reside here. The 41 special function registers (SFRs) include arithmetic registers (A, B, PSW), pointers (SP, DPH, DPL) and registers providing an interface between the CPU and the on-chip peripheral functions. There are also 128 directly addressable bits within the special function registers. Memory-mapping the SFRs allows accessing them as easily as the internal RAM. For this reason, they can be processed with most instructions. The special function registers are listed in Table 1.

Table 1. Special Function Registers

Symbol	Name	Address
* P0	Port 0	80H
SP	Stack Pointer	81H
DPL	Data Pointer, Low Byte	82H
DPH	Data Pointer, High Byte	83H
PCON	Power Control Register	87H
* TCON	Timer Control Register	88H
TMOD	Timer Mode Register	89H
TL0	Timer 0, Low Byte	8AH
TL1	Timer 1, Low Byte	8BH
TH0	Timer 0, High Byte	8CH
TH1	Timer 1, High Byte	8DH
* P1	Port 1	90H
* SCON	Serial Port Control Register	98H
SBUF	Serial Port Buffer Register	99H
* P2	Port 2	0A0H
* IEN0	Interrupt Enable Register 0	0A8H
IP0	Interrupt Priority Register 0	0A9H
* P3	Port 3	0B0H
* IEN 1	Interrupt Enable Register 1	0B8H
IP1	Interrupt Priority Register 1	0B9H
* IRCON	Interrupt Request Control Register	0C0H
CCEN	Compare/Capture Enable Register	0C1H
CCL1	Compare/Capture Register 1, Low Byte	0C2H
CCH1	Compare/Capture Register 1, High Byte	0C3H
CCL2	Compare/Capture Register 2, Low Byte	0C4H
CCH2	Compare/Capture Register 2, High Byte	0C5H
CCL3	Compare/Capture Register 3, Low Byte	0C6H
CCH3	Compare/Capture Register 3, High Byte	0C7H
* T2CON	Timer 2 Control Register	0C8H
CRCL	Compare/Reload/Capture Register, Low Byte	0CAH
CRCH	Compare/Reload/Capture Register, High Byte	0CBH
TL2	Timer 2, Low Byte	0CCH
TH2	Timer 2, High Byte	0CDH
* PSW	Program Status Word Register	0D0H
* ADCON	A/D Converter Control Register	0D8H
ADDAT	A/D Converter Data Register	0D9H
DAPR	D/A Converter Program Register	0DAH
* ACC	Accumulator	0E0H
* P4	Port 4	0E8H
* B	B Register	0F0H
* P5	Port 5	0F8H

The SFRs marked with an asterisk (*) are both bit and byte-addressable.

Accumulator

ACC is the accumulator register. The mnemonics for accumulator-specific instructions, however, refer to the accumulator simply as A.

Program Status Word (PSW)

The PSW register contains program status information as shown in Figure 5. A more detailed description of the bits contained in the PSW is given in section 5.1.2.

B Register

The B register is used during multiply and divide and serves as both source and destination. For other instructions it can be treated as another scratch pad register.

Stack Pointer

The stack pointer (SP) register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions and decremented after data is popped during a POP and RET (RETI) execution. While the stack may reside anywhere in on-chip RAM, the stack pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H above register bank 0. The SP can be read or written to under software control.

CY	AC	F0	RS1	RS0	OV	F1	P	Bit
0D7H	0D6H	0D5H	0D4H	0D3H	0D2H	0D1H	0D0H	Address

Figure 5. Program Status Word Register PSW (0D0H)

Symbol	Position	Name and Significance			
CY	PSW.7	Carry Flag			
AC	PSW.6	Auxiliary Carry Flag (for BCD Operations)			
F0	PSW.5	General Purpose User Flag 0			
RS1	PSW.4	Register Bank Select Control Bits 1 and 0. Set/cleared			
RS0	PSW.3	by software to determine working register bank:			
		RS1	RS0	Enabled Working Register Bank	
		0	0	Bank 0	00H–07H
		0	1	Bank 1	08H–0FH
		1	0	Bank 2	10H–17H
		1	1	Bank 3	18H–1FH
OV	PSW.2	Overflow Flag			
F1	PSW.1	General Purpose User Flag 1			
P	PSW.0	Parity Flag. Set/Cleared by Hardware each instruction cycle to indicate an odd/even number of "one" bits in the accumulator, i.e. even parity.			

Data Pointer

The 16-bit data pointer (DPTR) register is a concatenation of registers DPH (data pointer's high-order byte) and DPL (data pointer's low-order byte). The DPTR is used in register-indirect addressing to move program memory constants and external data memory variables, as well as to branch within the 64 kbyte program memory address space.

Ports 0 to 5

P0, P1, P2, P3, P4 and P5 are the SFR latches of ports 0, 1, 2, 3, 4 and 5. A more detailed description of the ports is given in later sections.

7

Peripheral Control, Data and Status Registers

The following registers contain the control, data and status information of the peripheral devices:

PCON TCON TMOD TL0 TL1 TH0 TH1
SCON SBUF IEN0 IP0 IEN1 IP1 IRCON
CCEN CCL1 CCH1 CCL2 CCH2 CCL3 CCH3
T2CON CRCL CRCH TL2 TH2 ADCON ADDAT
DAPR

Most of these registers are bit-addressable to facilitate control of peripheral functions. A more detailed description of each register is given in the respective sections in chapter 3, "On-Chip Peripheral Components".

2.3 Oscillator and Clock Circuit

XTAL1 and XTAL2 are the input and output of a single-stage on-chip inverter which can be configured with off-chip components as a Pierce-oscillator. The on-chip circuitry and selection of off-chip components to configure the oscillator are discussed in section 3.10.

The oscillator, in any case, drives the internal clock generator. The clock generator provides the internal clocking signals to the chip. The internal clocking signals are at half the oscillator frequency, and define the internal phases, states, and machine cycles, which are described in the next section. Figure 6 shows the recommended oscillator circuit.

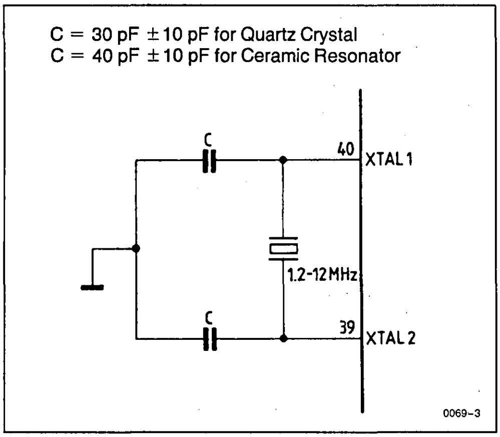

C = 30 pF ±10 pF for Quartz Crystal
C = 40 pF ±10 pF for Ceramic Resonator

Figure 6. Recommended Oscillator Circuit

2.4 CPU Timing

A machine cycle consists of 6 states (12 oscillator periods). Each state is divided into a phase 1 half, during which the phase 1 clock is active, and a phase 2 half, during which the phase 2 clock is active. Thus, a machine cycle consists of 12 oscillator periods, numbered S1P1 (state 1, phase 1) through S6P2 (state 6, phase 2). Each phase lasts for one oscillator period. Each state lasts for two oscillator periods. Typically, arithmetic and logical operations take place during phase 1 and internal register-to-register transfers take place during phase 2.

The diagrams in Figure 7 show the fetch/execute timing referenced to the internal states and phases. Since these internal clock signals are not user-accessible, the XTAL2 oscillator signal and the ALE (address latch enable) signal are shown for external reference. ALE is normally activated twice during each machine cycle: once during S1P2 and S2P1, and again during S4P2 and S5P1.

Execution of a one-cycle instruction begins at S1P2, when the op code is latched into the instruction register. If it is a two-byte instruction, the second byte is read during S4 of the same machine cycle. If it is a one-byte instruction, there is still a fetch at S4, but the byte read (which would be the next op code) is ignored, and the program counter is not incremented. In any case, execution is complete at the end of S6P3. Figures 8A and 8B show the timing for a 1-byte, 1-cycle instruction and for a 2-byte, 1-cycle instruction.

Most SAB 80515 instructions execute in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than two cycles to complete; they take four cycles. Normally, two code bytes are fetched from the program memory during every machine cycle. The only exception to this is when a MOVX instruction is executed. MOVX is a 1-byte, 2-cycle instruction that accesses external data memory. During a MOVX, two fetches are skipped while the external data memory is being addressed and strobed.

Figures 7C and 7D show the timing for a normal 1-byte, 2-cycle instruction and for a MOVX instruction.

2.5 Accessing External Memory

Accesses to external memory are of two types: accesses to external program memory and accesses to external data memory. Accesses to external program memory use the signal, $\overline{\text{PSEN}}$ (program store enable) as the read strobe. Accesses to external data memory use $\overline{\text{RD}}$ and $\overline{\text{WR}}$ (alternate functions of P3.7 and P3.6, see section 3.1) to strobe the memory. Port 0 and port 2 (with exceptions) are used to provide data and address signals. In this section only the port 0 and port 2 functions relevant to external memory accesses are described (for more detailed information see section 3.1).

Fetches from external program memory always use a 16-bit address. Accesses to external data memory can use either a 16-bit address (MOVX @DPTR) or an 8-bit address (MOVX @Ri).

Whenever a 16-bit address is used, the high byte of the address comes out on port 2, where it is held for the duration of the read or write cycle. During this time the port 2 latch (the special function register) does not have to contain 1s, and the contents of port 2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the port 2 SFR will reappear in the next cycle.

If an 8-bit address is being used (MOVX @Ri), the contents of the port 2 SFR remain at the port 2 pins throughout the external memory cycle. This will facilitate paging. In any case, the low byte of the address is time-multiplexed with the data byte on port 0. The address/data signal drives both FETs in the port 0 output buffers. Thus, in this application, the port 0 pins are not open-drain outputs, and do not require external pullup resistors. Signal ALE (address latch enable) should be used to capture the address byte

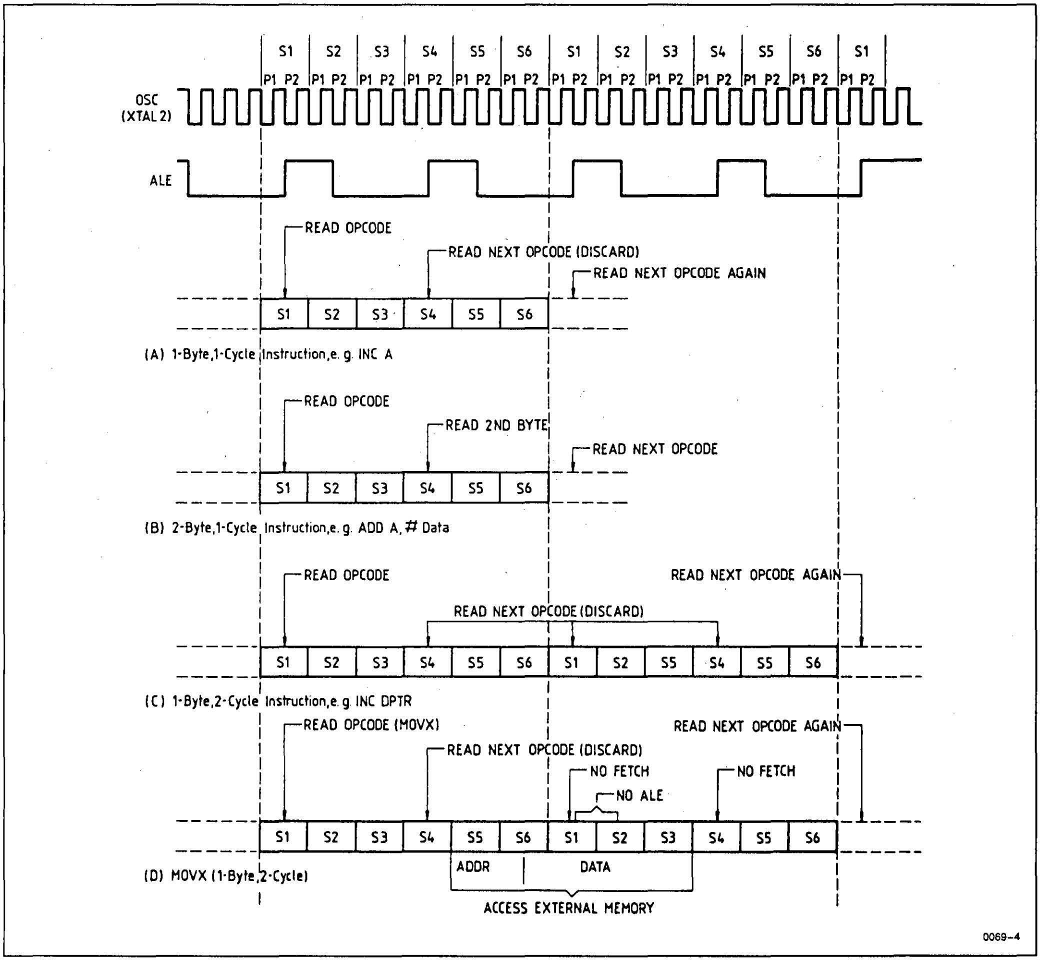

S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1
P1 P2
OSC (XTAL 2)
ALE
READ OPCODE
READ NEXT OPCODE (DISCARD)
READ NEXT OPCODE AGAIN
(A) 1-Byte,1-Cycle Instruction,e. g. INC A
READ OPCODE
READ 2ND BYTE
READ NEXT OPCODE
(B) 2-Byte,1-Cycle Instruction,e. g. ADD A, # Data
READ OPCODE
READ NEXT OPCODE AGAIN
READ NEXT OPCODE (DISCARD)
(C) 1-Byte,2-Cycle Instruction,e. g INC DPTR
READ OPCODE (MOVX)
READ NEXT OPCODE AGAIN
READ NEXT OPCODE (DISCARD)
NO FETCH
NO ALE
NO FETCH
(D) MOVX (1-Byte,2-Cycle)
ADDR
DATA
ACCESS EXTERNAL MEMORY
0069-4

Figure 7. Fetch/Execute Sequence

into an external latch. The address byte is valid at the negative transition of ALE. Then, in a write cycle, the data byte to be written appears on port 0 just before $\overline{\text{WR}}$ is activated, and remains there until after $\overline{\text{WR}}$ is deactivated. In a read cycle, the incoming byte is accepted at port 0 before the read strobe is deactivated.

During any access to external memory, the CPU writes 0FFH to the port 0 latch (the special function register), thus obliterating whatever information the port 0 SFR may have been holding.

External program memory is accessed under two conditions:

1. whenever signal $\overline{\text{EA}}$ is active; or
2. whenever the program counter (PC) contains a number that is larger than 1FFFH.

This requires that the ROMless version SAB 80535 has $\overline{\text{EA}}$ wired low to enable the lower 8k program bytes to be fetched from external memory.

When the CPU is executing out of external program memory, all 8 bits of port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program memory fetches they output the high byte of the PC, and during accesses to external data memory they output either DPH or the port 2 SFR (depending on whether the external data memory access is a MOVX @DPTR or a MOVX @Ri).

Since the SAB 80535 has no internal program memory, accesses to program memory are always external, and port 2 is always dedicated to output the high-order address byte. So for the SAB 80535, port 0 and port 2 can never be used as general purpose I/O. This applies also to the SAB 80515 when it is operated with only an external program memory.

It should be noted that, if a port 2 bit outputs an address bit that is a 1, then the enhancement-mode transistor at the output buffer will be turned on for the entire read/write cycle and not only for two oscillator periods (see section 3.1.2).

2.6 $\overline{\text{PSEN}}$

The read strobe for external fetches is $\overline{\text{PSEN}}$. $\overline{\text{PSEN}}$ is not activated for internal fetches. When the CPU is accessing external program memory, $\overline{\text{PSEN}}$ is activated twice every cycle (except during a MOVX instruction) no matter whether or not the byte fetched is actually needed for the current instruction. When $\overline{\text{PSEN}}$ is activated its timing is not the same as for $\overline{\text{RD}}$. A complete $\overline{\text{RD}}$ cycle, including activation and deactivation of ALE and $\overline{\text{RD}}$, takes 12 oscillator periods. A comlpete $\overline{\text{PSEN}}$ cycle, including activation and deactivation of ALE and $\overline{\text{PSEN}}$, takes 6 oscillator periods. The execution sequences for these two types of read cycles are shown in Figure 8.

2.7 ALE

The main function of ALE is to provide a properly timed signal to latch the low byte of an address from P0 to an external latch during fetches from external memory. For that purpose ALE is activated twice every machine cycle. This activation takes place even when the cycle involves no external fetch. The only time an ALE pulse doesn't come out is during an access to external data memory. The first ALE of the second cycle of a MOVX instruction is missing (see Figure 9). Consequently, in any system that does not use external data memory, ALE is activated at a constant rate of ⅙ the oscillator frequency, and can be used for external clocking or timing purposes.

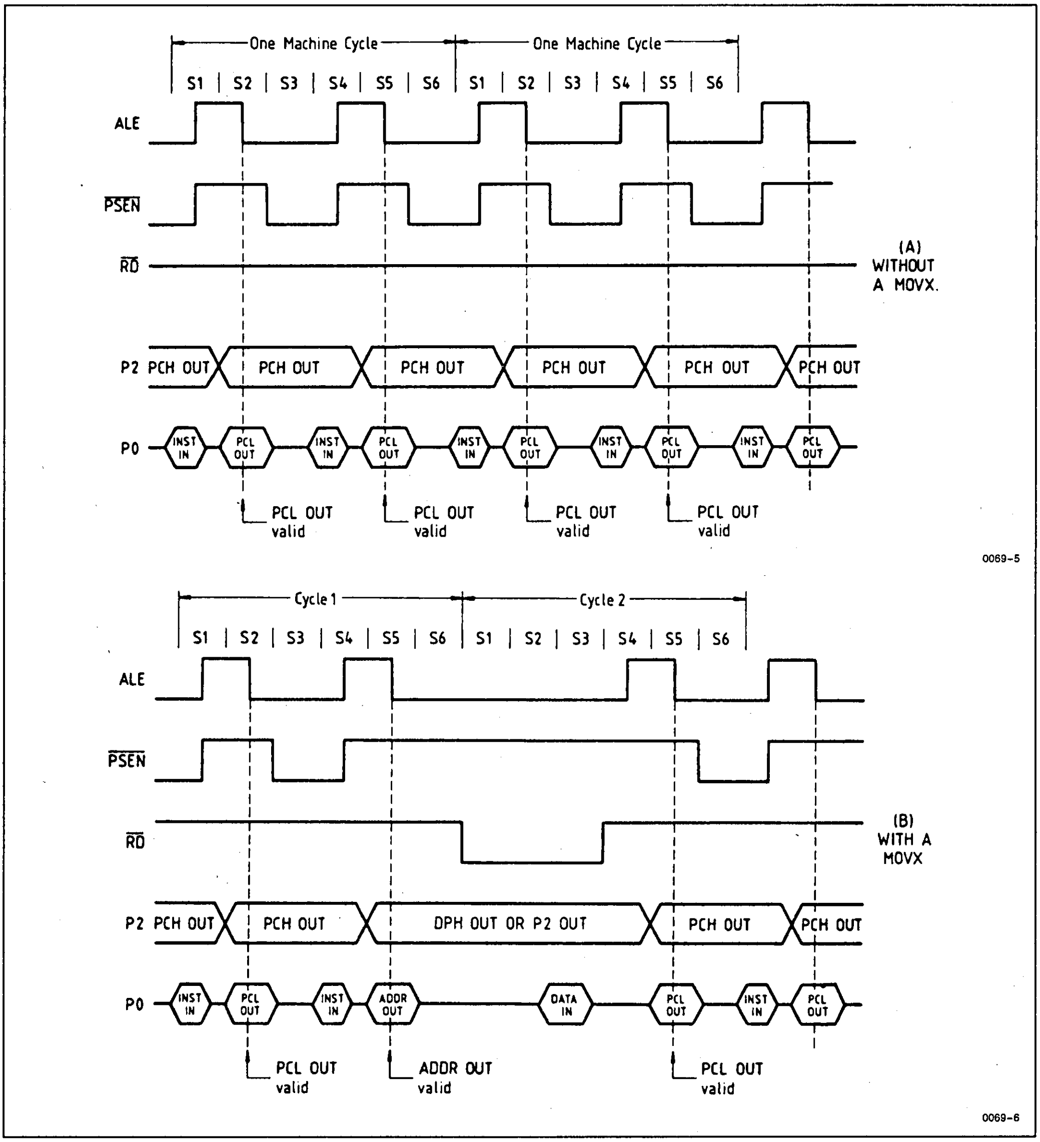


Figure 8. External Program Memory Execution

2.8 Overlapping External Program and Data Memory Spaces

In some applications it is desirable to execute a program from the same physical memory that is being used to store data. In the SAB 80515, the external program and data memory spaces can be combined by ANDing $\overline{\text{PSEN}}$ and $\overline{\text{RD}}$. A positive-logic AND of these two signals produces an active-low read strobe that can be used for the combined physical memory. Since the $\overline{\text{PSEN}}$ cycle is faster than the $\overline{\text{RD}}$ cycle, the external memory needs to be fast enough to accommodate the $\overline{\text{PSEN}}$ cycle.

2.9 $\overline{\text{RESET}}$

The reset input is an active low input with a pullup resistor connected to VCC. A Schmitt trigger is used at the input for noise rejection. A reset is accomplished by holding the reset pin, $\overline{\text{RESET}}$, low for at least 2 machine cycles (24 oscillator periods) while the oscillator is running. It leaves the internal registers as follows:

Table 2. Register Contents after Reset

Register	Contents	Register	Contents
P0–P5	0FFH	SP	07H
DPTR	0000H	PCON	(0XXXXXXX)
TCON	00H	TMOD	00H
TL0, TH0	00H	TL1, TH1	00H
TL2, TH2	00H	SCON	00H
IEN0, IEN1	00H	SBUF	undefined
IRCON	00H	IP0, IP1	00H
CCL1, CCH1	00H	CCEN	00H
CCL3, CCH3	00H	CCL2, CCH2	00H
T2CON	00H	CRCL, CRCH	00H
ADCON	00H	PSW	00H
DAPR	00H	ADDAT	00H
B	00H	ACC	00H

The internal RAM is not affected by a reset. Unlike the SAB 8051, the reset pin of the SAB 80515 is a high-impedance input. The reason for this is that the SAB 80515 uses an extra pin to backup the internal RAM (see section 3.8).

An automatic reset can be obtained when VCC is turned on by connecting the reset pin to GND through a capacitor as shown in Figure 9a. After VCC is turned on, the capacitor must hold the voltage at the reset pin at a level remaining under the higher threshold of the Schmitt trigger to effect a complete reset. This must last at least 10 ms for a crystal oscillator and 50 µs for a ceramic oscillator. The time required is the oscillator start-up time, plus 2 machine cycles.

Figures 9b and 9c show two additional examples of a reset circuit for the SAB 80515.

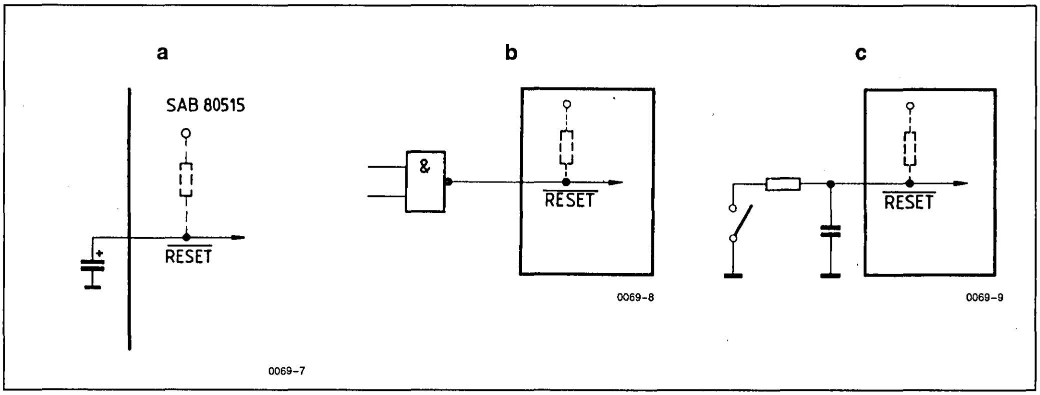


Figure 9(a–c). Reset Circuits

On-Chip Peripheral Components

3.0 On-Chip Peripheral Components

This chapter describes the on-chip peripheral components of the SAB 80515.

3.1 Port Structures and Operation

All six 8-bit ports of the SAB 80515 are bidirectional. Each consists of a latch (special function registers P0 through P5), an output driver, and an input buffer.

The output drivers of ports 0 and 2, and the input buffers of port 0, are used when accessing external memory (see Section 2.5). In this application, port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise the port 2 pins continue to emit the P2 SFR contents.

All the port 1 and port 3 pins are multifunctional. They are not only port pins, but also serve the functions of various special features as listed in Table 3. The alternate functions of port 3 are the same as for the SAB 8051/8052.

To use the alternate functions on port pins P3.0 to P3.7 and P1.4 to P1.7 the corresponding bit latch in the port SFR has to contain a one (1). Otherwise the port pin is stuck at 0. If the alternate input functions (interrupt input, capture input) on pins P1.0 to P1.3 are used, the corresponding bit latches must contain a one (1), too. If these pins are used as compare outputs, the value stored in the bit latches depends on the established compare modes as described in Section 3.3.2.

3.1.1 I/O Configuration

Figure 11 shows a functional diagram of a typical bit latch and I/O buffer in each of the six ports. The bit latch (one bit in the port's SFR) is represented as a D-type flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port, activate the "read latch" signal, and others activate the "read pin" signal. More about that in the section "Read-Modify-Write Feature" (3.1.4).

As shown in Figures 11a and 11c, the output drivers of ports 0 and 2 are switchable to an internal address and address/data bus by an internal control signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but the P0 SFR gets 1s written to it. Also shown in Figure 11, is that if a P1 or P3 bit latch contains a one (1), then the output level is controlled by the signal labeled "alternate output function". The actual P1.X or P3.X pin level is always available

Table 3. Alternate Functions on Port 1 and Port 3

Port Pin		Alternate Function
P1.0	$\overline{\text{INT3}}$/CC0	External Interrupt 3/Capture 0/ Compare 0
P1.1	INT4/CC1	External Interrupt 4/Capture 1/ Compare 1
P1.2	INT5/CC2	External Interrupt 5/Capture 2/ Compare 2
P1.3	INT6/CC3	External Interrupt 6/Capture 3/ Compare 3
P1.4	$\overline{\text{INT2}}$	External Interrupt 2
P1.5	T2EX	Timer 2 External Reload Trigger Input
P1.6	CLKOUT	System Clock Output
P1.7	T2	Timer 2 Input
P3.0	RxD	Serial Input Port
P3.1	TxD	Serial Output Port
P3.2	$\overline{\text{INT0}}$	External Interrupt 0
P3.3	$\overline{\text{INT1}}$	External Interrupt 1
P3.4	T0	Timer 0 Input
P3.5	T1	Timer 1 Input
P3.6	WR	External Data Memory Write Strobe
P3.7	RD	External Data Memory Read Strobe

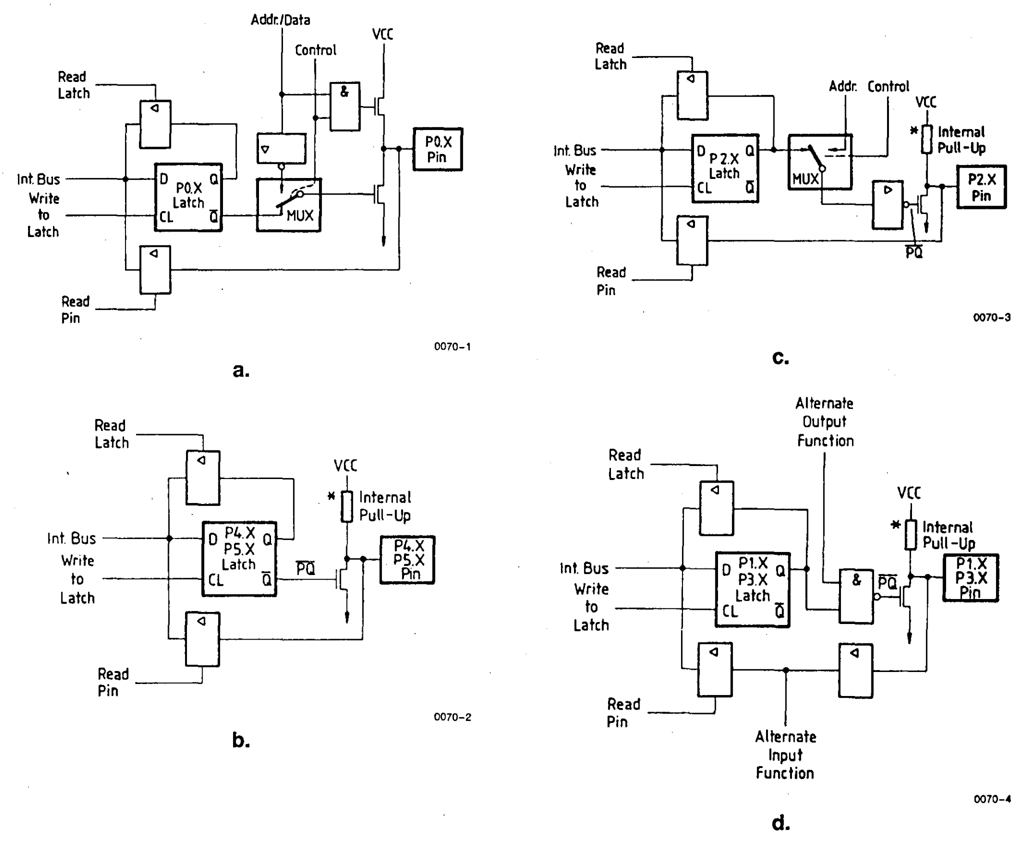
to the pin's alternate function, if any. For exceptions applying to pins P1.0 to P1.3 see Section 3.3.2.

Ports 1 through 5 have internal pullup FETs. Port 0 has open-drain outputs. Each I/O line can be used independently as an input or output. Port 0 and port 2 may not be used as general-purpose I/O when being used as the address/data bus To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. Then, for ports 1 through 5, the pin is pulled high by the internal pullup, but can be pulled low by an external source.

Port 0 differs in not having internal pullups. The pullup FET in the P0 output driver (see Figure 11a) is used only when the port is emitting 1s during external memory accesses: Otherwise the pullup is off. Consequently P0 lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, so the pin floats. In that condition it can be used as a high-impedance input.

Because ports 1 through 5 have fixed internal pullups they are sometimes called "quasi-bidirectional" ports. When configured as inputs they pull high and will source current when externally pulled low. Port 0, on the other hand, is considered "true" bidirectional, because when configured as an input it floats.

All the port latches in the SAB 80515 have 1s written to them by the reset function. If a 0 is subsequently written to a port latch, it can be reconfigured as an input by writing a 1 to it.



*) See Figure 12 for details on the internal pullup

Figure 11. Port Bit Latches and I/O Buffers

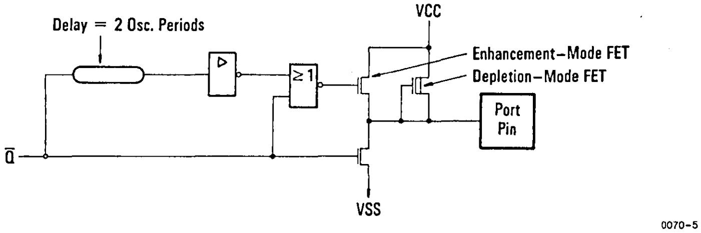


Figure 12. Ports 1 to 5 Internal Pullup Configurations

3.1.2 Writing to a Port

When executing an instruction that changes the value in a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are in fact sampled by their output buffers only during phase 1 of any clock period (during phase 2 the output buffer holds the value it saw during the previous phase 1). Consequently, the new value in the port latch won't actually appear at the output pin until the next phase 1, which will be at S1P1 of the next machine cycle.

If the change requires a 0-to-1 transition in port 1 through 5, an additional pullup is turned on during S1P1 and S1P2 of the cycle in which the transition occurs. This is done to increase the transition speed. The extra pullup can source about 100 times the current that the normal pullup can. It should be noted that the internal pullups are field-effect transistors, not linear resistors. The pullup arrangements are shown in Figure 12.

In the SAB 80515, the fixed part of the pullup is a depletion-mode transistor with the gate wired to the source. This transistor will allow the pin to source typically 0.25 mA (a typical value) when shorted to ground. In parallel with the fixed pullup is an enhancement-mode transistor, which is activated during S1 whenever the port bit does a 0-to-1 transition. During this interval, i.e. a 0-to-1 transition if the port pin is shorted to ground, this extra transistor will allow the pin to source an additional typical 30 mA (a transitional value which should not be treated as spec).

As already discussed in section 2.5 for port 2, this extra transistor is turned on for the entire external memory cycle every time port 2 outputs an address bit that is a one (1). That means, port 2 cannot be used as general-purpose I/O when the SAB 80515 executes out of external program memory.

3.1.3 Port Loading and Interfacing

The output buffers of port 1 through 5 can each drive 4 LS-TTL inputs. Port 0 output buffers can each drive 8 LS-TTL inputs. They do, however, require external pullups to drive NMOS inputs, except when being used as the address/data bus.

3.1.4 Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called, "read-modify-write" instructions. The instructions listed in Table 4 are the read-modify-write instructions. When the destination operand is a port, or a port bit, these instructions read the latch rather than the pin.

It is not obvious that the last three instructions in this list are read-modify-write instructions, by they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a 1 is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as 0. Reading the latch rather than the pin will return the correct value of 1.

Table 4. Read-Modify-Write Instruction

ANL	Logical AND; e.g. ANL P1, A
ORL	Logical OR; e.g. ORL P2, A
XRL	Logical Exclusive OR; e.g. XRL P3,A
JBC	Jump if Bit is Set and Clear Bit; e.g. JBC P1.1, LABEL
CPL	Complement Bit; e.g. CPL P3.0
INC	Increment; e.g. INC P2
DEC	Decrement; e.g. DEC P2
DJNZ	Decrement and Jump if Not Zero; e.g. DJNZ P3, LABEL
MOV PX.Y,C	Move Carry Bit to Bit Y of Port X
CLR PX.Y	Clear Bit Y or Port X
SET PX.Y	Set BIt Y of Port X

3.2 Timer 0 and Timer 1

The SAB 80515 has three 16-bit timer/counters: timer 0, timer 1, and timer 2 (timer 2 is discussed separately in Section 3.3). Timer 0 and timer 1 can be configured to operate either as timers or event counters.

In "timer" function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In "counter" function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 or T1. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected.

Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

In addition to the "timer" and "counter" selection, timer 0 and timer 1 have four operating modes from which to select.

Each timer consists of two 8-bit registers (TH0 and TL0 for timer 0, TH1 and TL1 for timer 1) which may be combined to one timer configuration depending on the mode that is established. The functions of the timers are controlled by two special function registers TCON and TMOD, shown in Figures 13 and 14.

The "timer" or "counter" function is selected by control bits C/$\overline{T}$ in the special function register TMOD. The two timer counters have four operating modes which are selected by bit pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both timer/counters, mode 3 is different. The four operating modes are described in the following.

In the following descriptions the symbols TH0 and TL0 are used to specify the high-byte and low-byte of timer 0 (TH1 and TL1 for timer 1, respectively). The operating modes are described and shown for timer 1. If not explicitly noted, this applies also to timer 0.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	Bit
8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	Address

Symbol	Position	Function
IT0	TCON.0	Interrupt 0 Type Control Bit. Set/cleared by software to specify falling edge/low-level triggered external interrupts.
IE0	TCON.1	Interrupt 0 Edge Flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 Type Control Bit. Set/cleared by software to specify falling edge/low-level triggered external interrupts.
IE1	TCON.3	Interrupt 1 Edge Flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TR0	TCON.4	Timer 0 Run Control Bit. Set/cleared by software to turn timer/counter 0 on/off.
TF0	TCON.5	Timer 0 Overflow Flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	TCON.6	Timer 1 Run Control Bit. Set/cleared by software to turn timer/counter 1 on/off.
TF1	TCON.7	Timer 1 Overflow Flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

Figure 13. Timer Control Register TCON (88H)

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Timer 1				Timer 0			

GATE Gating Control. When set, timer/counter "x" is enabled only while "$\overline{INTx}$" pin is high and "TRx" control bit is set. When cleared timer "x" is enabled whenever "TRx" control bit is set.

C/$\overline{T}$ Timer or Counter Select Bit. Cleared for timer operation (input from internal system clock). Set for counter operation (input from "Tx" input pin).

M1	M0	Operating Mode
0	0	8-Bit Timer/Counter. "TLx" serves as 5-bit prescaler.
0	1	16-Bit Timer/Counter. "THx" and "TLx" are cascaded; there is no prescaler.
1	0	8-Bit Auto-Reload Timer/Counter. "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	Timer 0: TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only controlled by timer 1 control bits.
1	1	Timer 1: Timer/counter 1 stops.

Figure 14. Timer/Counter Mode Control Register TMOD (89H)

3.2.1 Mode 0

Putting either timer into mode 0 configures it as an 8-bit counter with a divide-by-32 prescaler. Figure 15 shows the mode 0 operation as it applies to timer 1.

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the timer interrupt flag TF1. The counted input is enabled to the timer when TR1 = 1 and either GATE = 0 or $\overline{INT1}$ = 1 (setting GATE = 1 allows the timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements). TR1 is a control bit in the special function register TCON; GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for timer 0 as for timer 1. Substitute TR0, TF0, TH0, TL0, and $\overline{INT0}$ for the corresponding timer 1 symbols in Figure 15. There are two different GATE bits, one of timer 1 (TMOD.7) and one for timer 0 (TMOD.3).

3.2.2 Mode 1

Mode 1 is the same as mode 0, except that the timer register is being run with all 16 bits. Mode 1 is shown in Figure 16.

3.2.3 Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL1) with automatic reload, as shown in Figure 17. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

3.2.4 Mode 3

Timer 1 in mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in mode 3 establishes TL0 and TH0 as two separate counters. The logic for mode 3 on timer 0 is shown in Figure 18. TL0 uses the timer 0 control bits: C/$\overline{T}$, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from timer 1. Thus, TH0 now controls the "timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer or counter. With timer 0 in mode 3, an SAB 80515 can look like it has four timer/counters. When timer 0 is in mode 3, timer 1 can be turned on and off by switching it out of and into its own mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

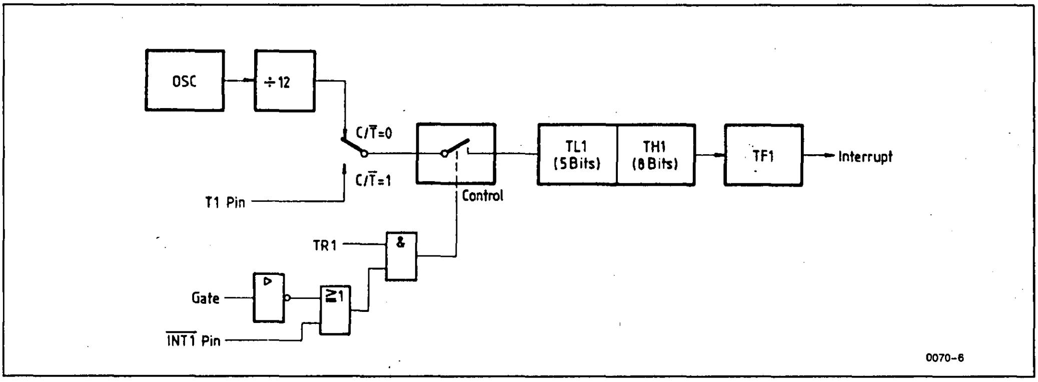


Figure 15. Timer/Counter 1 Mode 0: 13-Bit Counter

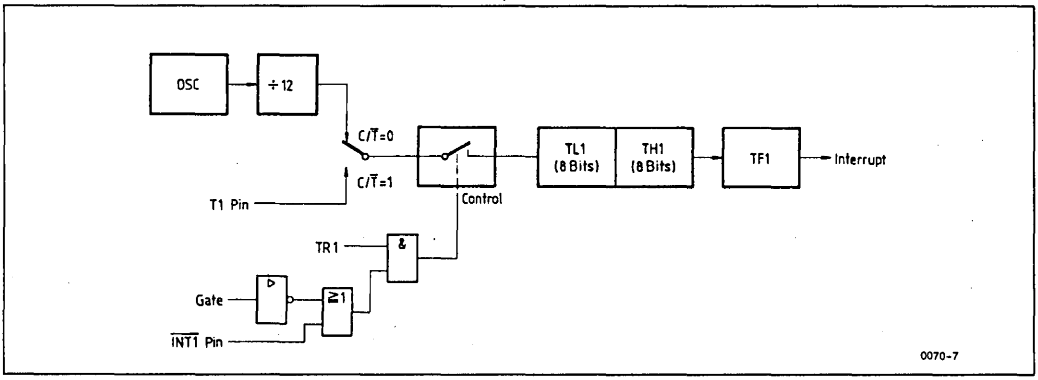


Figure 16. Timer/Counter 1 Mode 1: 16-Bit Counter

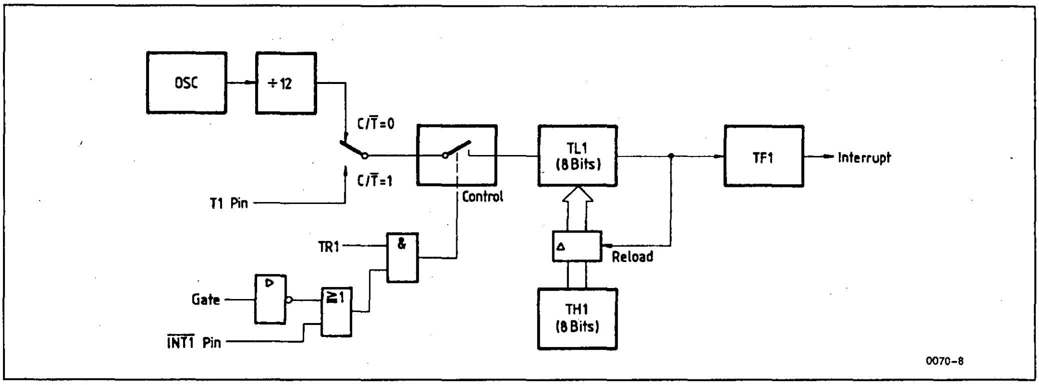


Figure 17. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload

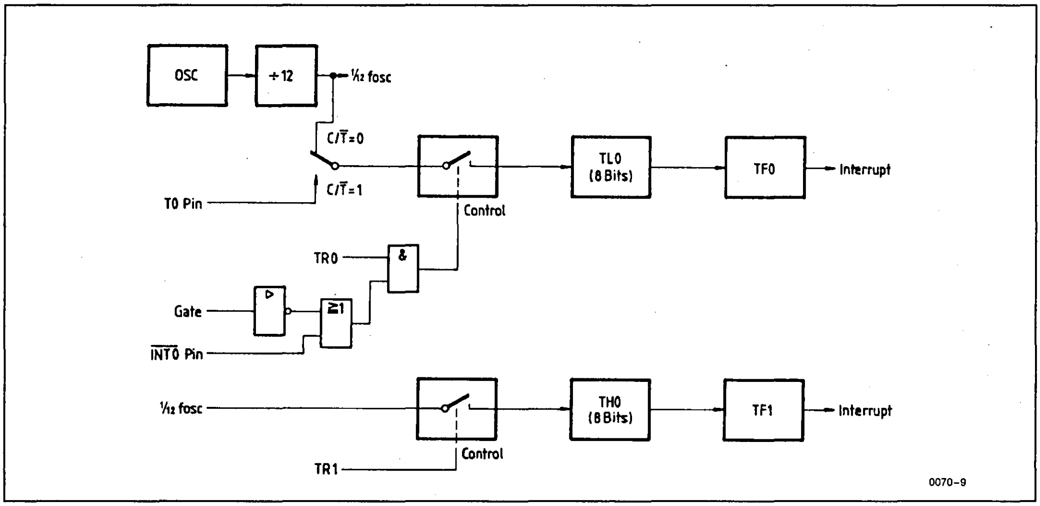


Figure 18. Timer/Counter 0 Mode 3: Two 8-Bit Counters

3.3 PTRA Unit

The term PTRA Unit (Programmable Timer/Counter Register Array) refers to a complex circuit consisting of the following registers:

T2CON	Timer 2 control register
TL2	Timer 2 register, low-byte
TH2	Timer 2 register, high-byte
CRCL	Compare/reload/capture register, low-byte
CRCH	Compare/reload/capture register, high-byte
CCL1	Compare/capture register 1, low-byte
CCH1	Compare/capture register 1, high-byte
CCL2	Compare/capture register 2, low-byte
CCH2	Compare/capture register 2, high-byte
CCL3	Compare/capture register 3, low-byte
CCH3	Compare/capture register 3, high-byte
CCEN	Compare/capture enable register

For brevity, the double-byte compare/reload/capture register is called CRC register, the three double-byte compare/capture registers are called CC registers 1 to 3.

Six bits of port 1 are used by the timer 2 circuit for special functions:

P1.0/$\overline{\text{INT3}}$/CC0	Compare output/capture input for the CRC register
P1.1/INT4/CC1	Compare output/capture input for CC register 1
P1.2/INT5/CC2	Compare output/capture input for CC register 2
P1.3/INT6/CC3	Compare output/capture input for CC register 3
P1.5/T2EX	External reload trigger input
P1.7/T2	External count or gate input to timer 2

To use the special functions on pins P1.5/T2EX and P1.7/T2 a one (1) must first be written into the appropriate bit latches. For pins P1.0 to P1.3 it depends on the special function, whether the bit latches must contain a one (1) or not. Should those pins be used as interrupt or capture inputs, the corresponding bit latches must contain a one (1). If those pins are used as compare outputs, the value written to the bit latches depends on the compare modes established.

In addition to the operational modes "timer" or "counter", timer 2 provides the features of:

— 16-bit reload

— 16-bit compare

— 16-bit capture

Figure 19 shows a block diagram of the timer 2 circuit.

The timer 2 can operate either as timer, event counter, or gated timer. In timer function, the count rate is derived from the oscillator frequency. A 2:1 prescaler offers the possibility to select a count rate of 1/12 or 1/24 of the oscillator frequency. Thus, the 16-bit timer 2 register (consisting of TL2 and TH2) is incremented every machine cycle or every second machine cycle. The prescaler is selected by bit T2PS in special function register T2CON (see Figure 20). If T2PS is cleared, the input frequency is 1/12 of the oscillator frequency; if T2PS is set, the 2:1 prescaler gates 1/24 of the oscillator frequency to the timer.

In gated timer function, the external input pin T2 (P1.7) functions as a gate to the input of timer 2. If T2 is high, the counted input is gated to the timer. T2 = 0 stops the counting procedure. This will facilitate pulse width measurements.

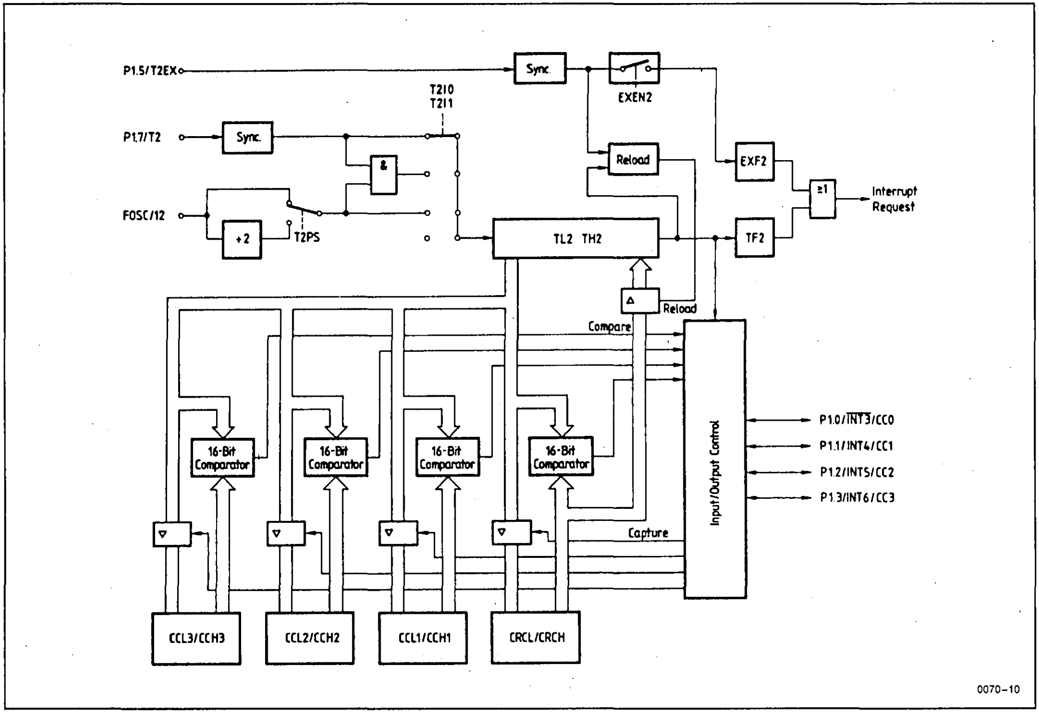


Figure 19. Timer 2 Block Diagram

T2PS	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0	Bit
0CFH	0CEH	0CDH	0CCH	0CBH	0CAH	0C9H	0C8H	Address

Symbol	Position	Function
T2I0 T2I1	T2CON.0 T2CON.1	Timer 2 Input Selection. See Table 5.
T2CM	T2CON.2	Compare Mode Bit. When set, compare mode 1 is selected. T2CM = 0 selects compare mode 0.
T2R0 T2R1	T2CON.3 T2CON.4	Timer 2 Reload Mode Selectin. See Table 6.
I2FR	T2CON.5	External Interrupt 2 Falling/Rising Edge Flag. When set, the interrupt 2 request flag IEX2 will be set on a positive transition at pin P1.4/$\overline{\text{INT2}}$. I2FR = 0 specifies external interrupt 2 to be negative-transition active.
I3FR	T2CON.6	External Interrupt 3 Falling/Rising Edge Flag. When set, the interrupt 3 request flag IEX3 will be set on a positive transition at pin P1.0/$\overline{\text{INT3}}$/CC0. I3FR = 0 specifies external interrupt 3 to be negative-transition active.
T2PS	T2CON.7	Prescaler Select Bit. When set, timer 2 is clocked in the "timer" or "gated timer" function with 1/24 of the oscillator frequency. T2PS = 0 gates fosc/12 to timer 2. T2PS must be 0 for the counter operation of timer 2.

Figure 20. Timer 2 Control Register T2CON (0C8H)

In counter function, the timer 2 register is incremented in response to a 1-to-0 transition at its corresponding external input pin T2 (P1.7). In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S1P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

Note:
The prescaler must be off for proper counter operation of timer 2, that means, T2PS must be 0.

In either case, no matter whether timer 2 is configured as timer, event counter, or gated timer, a rolling over of the count from all 1s to all 0s sets the timer 2 overflow flag TF2 (bit 6 in SFR IRCON, Interrupt Request Control) which can generate an interrupt.

The input clock to timer 2 is selected by bits T2I0, T2I1 and T2PS as listed in Table 5.

Table 5. Timer 2 Input Selection

T2I1	T2I0	Function
0	0	No Input Selected, Timer 2 Stops
0	1	Timer Function, Input Frequency = fosc/12 (T2PS = 0) or fosc/24 (T2PS = 1)
1	0	Counter Function, External Input Signal at Pin T2/P1.7
1	1	Gated Timer Function. Input Controlled by Pin T2/P1.7

3.3.1 Reload

The reload mode for timer 2 is selected by bits T2R0 and T2R1 in SFR T2CON as illustrated in Table 6. In mode 0, when timer 2 rolls over from all 1s to all 0s, it not only sets TF2 but also causes the timer 2 registers to be loaded with the 16-bit value in the CRC register which is preset by software. The reload will happen in the same machine cycle in which TF2 is set, thus overwriting the count value 0000H. In mode 1, a 16-bit reload from the CRC register is caused by a negative transition at the corresponding input pin T2EX/P1.5. In addition, this transition will set flag EXF2 if bit EXEN2 in SFR IEN1 is set. If the timer 2 interrupt is enabled, setting EXF2 will generate an interrupt. The external input pin T2EX is sampled during S5P2 of every machine cycle. When the sampling shows a high in one cycle and a low in the next cycle, a transition will be recognized. The reload of the timer 2 registers will then take place during S2P1 of the cycle following the one in which the transition was detected.

Figure 21 shows a functional diagram of the timer 2 reload modes.

T2R1 = 0 disables the reload modes 0 and 1. If the reload modes are disabled, and if EXEN2 is set, a negative transition at pin T2EX/P1.5 can be used as additional external interrupt input. More detailed information on the interrupts of the SAB 80515 can be found in Section 3.6.

Table 6. Timer 2 Reload Mode Selection

T2R1	T2R0	Mode
0	X	Reload Disabled
1	0	Mode 0: Auto-Reload upon Timer 2 Overflow (TF2)
1	1	Mode 1: Reload upon Falling Edge at Pin T2EX/P1.5

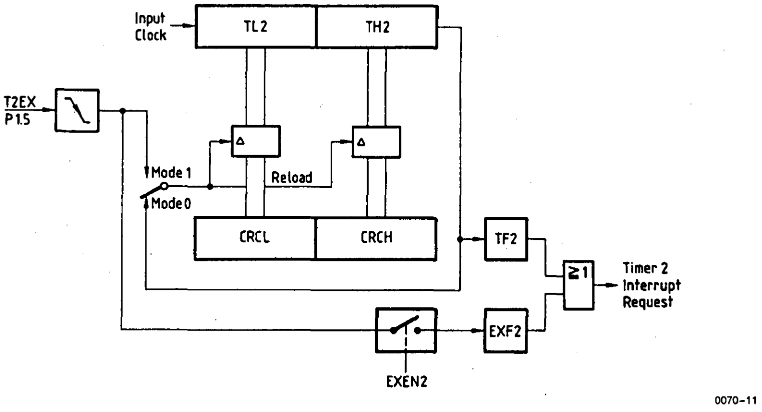


Figure 21. Timer 2 in Reload Mode

3.3.2 Compare

In compare mode, the 16-bit values stored in the dedicated compare registers are compared with the contents of the timer 2 registers (TL2 and TH2). If the count value in the timer 2 registers matches the stored one an appropriate output signal is generated at the corresponding sort 1 pin, and interrupt is requested.

The compare modes are enabled by setting the appropriate bits in SFR CCEN (compare/capture enable register, see Figure 27). There are two different compare modes which are selected by bit T2CM in T2CON.

In mode 0, upon a match, the output signal changes from low to high. It goes back to a low level on timer 2 overflow. As long as compare mode 0 is enabed, the appropriate output pin is controlled by the timer 2 circuit, and not by the user. Writing to the port will operate as "dummy" instruction. Figure 22 shows a functional diagram of the port 1 latches P1.0 to P1.3 in compare mode 0. The port latch is directly controlled by the two signals Timer2 overflow and compare. The input line from the internal bus and the "write-to-latch" line are disconnected when compare mode 0 is enabled.

In mode 1, the software determines the transition of the output signal. If mode 1 is enabled, and the software writes to the appropriate output pin at port 1, the new value won't appear at the output pin until the next compare event occurs. Thus, one can select whether the output signal makes a 1-to-0 or a 0-to-1 transition, at the time the timer 2 count matches the stored compare value. Figure 23 shows a functional diagram of the port 1 latches P1.0 to P1.3 in compare mode 1. In this function, the "port latch" consists of two separate latches. The "left" latch can be written to under software control, but this value will only be transferred to the "right" latch (and to the port pin) in response to a compare event. Note that the "right" latch is transparent as long as the internal compare signal is active. While the compare signal is active a write operation to the port will change both latches. A "read-modify-write" instruction (see section 3.1.4) will read the user-controlled "left" latch, and write the modified value back to this "left" latch.

In both compare modes, the new value arrives at the port 1 pin within the same machine cycle in which the internal compare signal is activated.

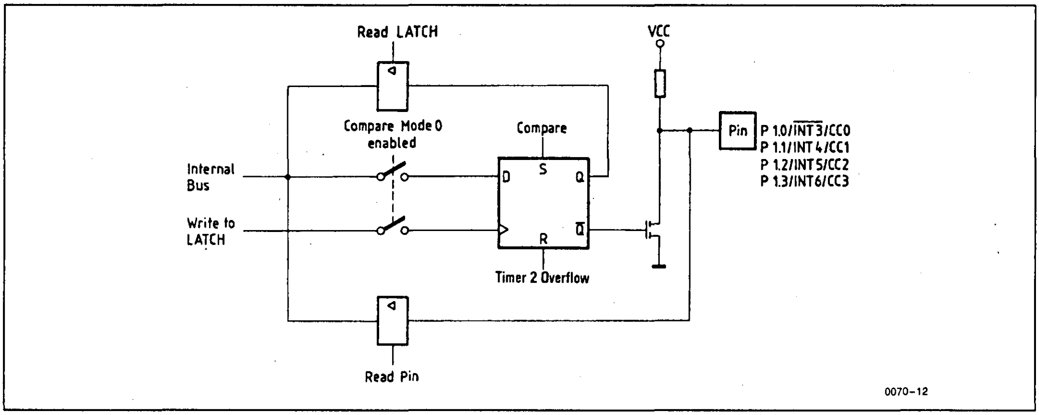


Figure 22. Functional Diagram of Port Latches P1.0 to P1.3 in Compare Mode 0

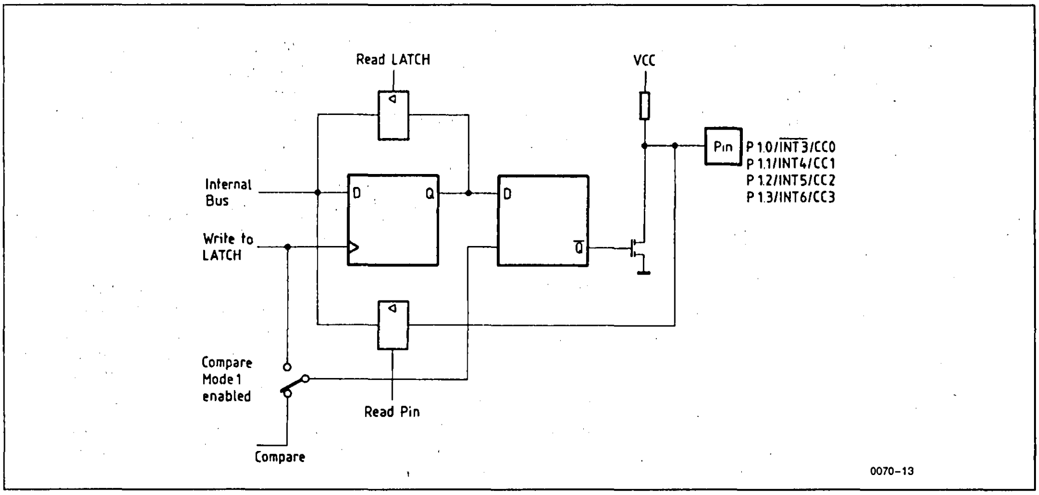
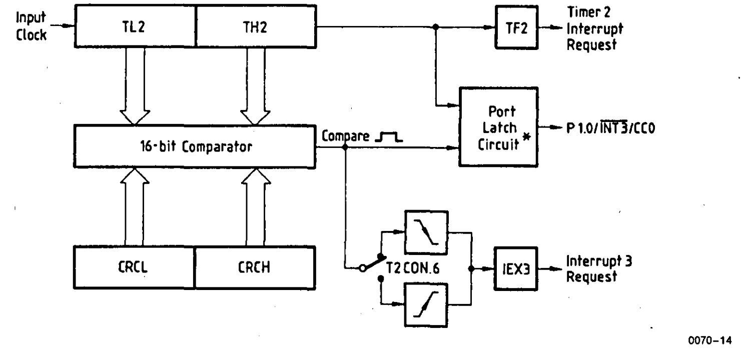


Figure 23. Functional Diagram of Port Latches P1.0 to P1.3 in Compare Mode 1



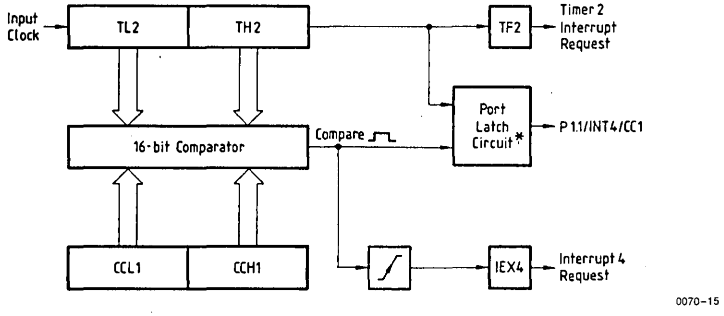
*) See Figures 22 and 23

Figure 24. Functional Diagram of Timer 2 in Compare Mode Using the CRC Register

Figure 24 shows a functional diagram of timer 2 in the compare mode using the CRC register. Figure 25 shows the compare modes with reference ot the CC register 1. Except for the symbolic names, this diagram applies also to the CC registers 2 and 3.

Note that the compare signal is active as long as the timer 2 contents is equal to the one of the appropriate compare register, and that it has a rising and a falling edge. Thus, when using the CRC register, it can be selected whether an interrupt should be caused when the compare signal goes active or inactive, depending on the status of bit I3FR in T2CON. For the CC registers 1 to 3 an interrupt is always requested when the compare signal goes active.

If compare function is enabled, the corresponding port 1 pin is dedicated to act as output. The level at the port pin can be read under software control, but the input line from the port pin to the interrupt system is disconnected. Thus, a change of the pin's level will not cause a setting of the corresponding interrupt flag. In the compare modes, the external interrupt request flags can only be set by the internally generated compare signal.



*) See Figures 22 and 23

Figure 25. Functional Diagram of Timer 2 in Compare Mode Using CC Register 1

3.3.3 Capture

Each of the three compare/capture registers and the CRC register can be used to latch the current 16-bit value in the timer 2 registers TL2 and TH2. Two different modes are provided for this function. In mode 0, an external event causes a latching of the timer 2 contents to a dedicated capture register. In mode 1, a capture will occur upon writing to the low-order byte of the dedicated 16-bit capture register. This mode is provided to allow the software reading the timer 2 contents "on the fly".

In mode 0, the external event causing a capture is

- for CC registers 1 to 3: a positive transition at pins CC1 to CC3 of CC registers 1 to 3,
- for the CRC register: a positive or negative transition, depending on the status of bit I3FR in SFR T2CON, at pin CC0. If bit I3FR is cleared, a capture occurs in response to a negative transition, if bit I3FR is set a capture occurs in response to a positive transition at pin P1.0/$\overline{\text{INT3}}$/CC0.

In this mode, the appropriate port 1 pin is used as input and the port latch must be programmed to contain a one (1). The external input is sampled during S5P2 in every machine cycle. When the sampling shows a low (high for input CC0, if it is programmed to be negative-transition-active) in one cycle and a high (low) in the next cycle, a transition is recognized. The timer 2 contents is latched to the appropriate capture register during S3P1 in the cycle following the one in which the transition was identified.

In mode 0, a transition on the external capture inputs CC0 to CC3 will also cause setting of the corresponding external interrupt request flags IEX3 to IEX6. If the interrupts are enabled, an external capture signal will cause the CPU to vector to the appropriate interrupt service routine.

In mode 1, a capture occurs in response to a MOV instruction to the low-order byte of a capture register. The "write-to-register" signal (e.g. "write to CRCL") is used to initiate a capture. The value written to the dedicated capture register is irrelevant for this function. The timer 2 contents will be latched into the appropriate capture register in the cycle following the MOV instruction. In this mode no interrupt request will be generated.

In both capture modes, the value latched in the machine cycle in which the capture occurs will be the actual contents of timer 2 in that machine cycle.

Figures 26a and 26b show functional diagrams of the capture function of timer 2. Figure 26a illustrates the operation for the CRC register, while Figure 26b shows the operation applying to the compare/capture register 1. This operation is the same for CC register 1 as well as for the CC registers 2 and 3. Substitute the symbols for the corresponding signals and names of CC registers 2 and 3 in Figure 26b.

The two capture modes can be established individually for each capture register by bits in SFR CCEN (compare/capture enable register), with two bits for each capture register. That means, other than for the compare modes, it is possible to select mode 0 for one capture register and mode 1 for another register simultaneously. The bit positions and functions of CCEN are listed in Figure 27.

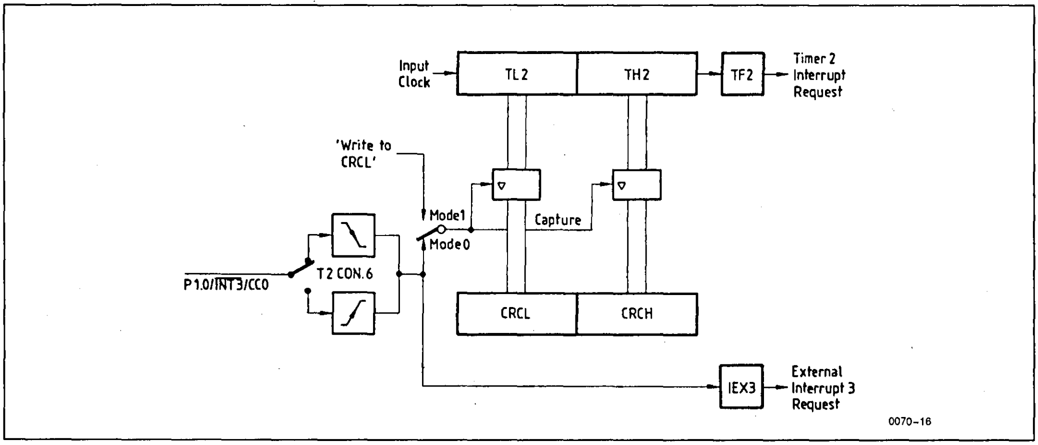


Figure 26a. Functional Diagram of Timer 2 in Capture Mode Using the CRC Register

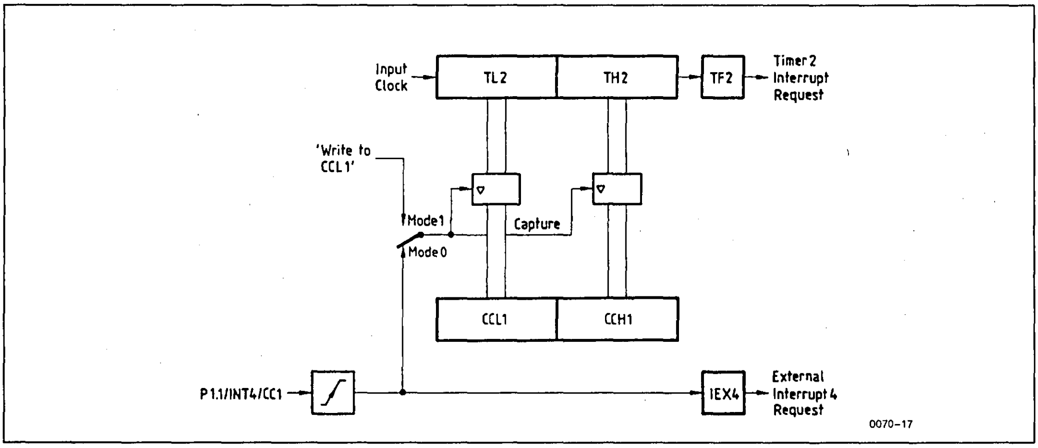


Figure 26b. Functional Diagram of Timer 2 in Capture Mode Using CC Register 1

7	6	5	4	3	2	1	0	Bit

Bit		Function
1	**0**	**CRC Register**
0	0	Compare/Capture Disabled
0	1	Capture on Falling/Rising Edge at Pin P1.0/$\overline{\text{INT3}}$/CC0
1	0	Compare Enabled
1	1	Capture on Write Operation into Register CRCL
3	**2**	**CC Register 1**
0	0	Compare/Capture Disabled
0	1	Capture on Rising Edge at Pin P1.1/INT4/CC1
1	0	Compare Enabled
1	1	Capture on Write Operation into Register CCL1
5	**4**	**CC Register 2**
0	0	Compare/Capture Disabled
0	1	Capture on Rising Edge at Pin P1.2/INT5/CC2
1	0	Compare Enabled
1	1	Capture on Write Operation into Register CCL2
7	**6**	**CC Register 3**
0	0	Compare/Capture Disabled
0	1	Capture on Rising Edge at Pin P1.3/INT6/CC3
1	0	Compare Enabled
1	1	Capture on Write Operation into Register CCL3

Figure 27. Compare/Capture Enable Register CCEN (0C1H)

3.4 Serial Interface

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register (however, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost). The serial port receive and transmit registers are both accessed at special function register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in 4 modes:

Mode 0: Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: 10 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB8 in special function register SCON. The baud rate is variable.

Mode 2: 11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmission, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On reception, the 9th data bit goes into RB8 in special function register SCON, while the stop bit is ignored. The baud rate is programmable to either $^1/_{32}$ or $^1/_{64}$ the oscillator frequency.

Mode 3: 11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). In fact, mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1. The control, mode, and status bits of the serial port in special function register SCON are illustrated in Figure 28.

3.4.1 Multiprocessor Communication

Modes 2 and 3 of the serial interface of the SAB 80515 have a special provision for multiprocessor communication. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor communications is as follows.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in mode 0, and in mode 1 can be used to check the validity of the stop bit. In a mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

3.4.2 Baud Rates

The baud rate in mode 0 is fixed:

$$\text{Mode 0 baud rate} = \frac{\text{oscillator frequency}}{12}$$

SM0	SM1	SM2	REN	TB8	RB8	TI	RI	Bit
9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	Address

Symbol	Position	Function
SM0 SM1	SCON.7 SCON.6	Serial Port Mode Selection, see Table 7.
SM2	SCON.5	Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
REN	SCON.4	Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB8	SCON.3	Is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
RB8	SCON.2	In modes 2 and 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Is the transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
RI	SCON.0	Is the receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes, in any serial reception. Must be cleared by software.

Figure 28. Serial Port Control Register SCON (98H)

Table 7. Serial Port Mode Selection

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	fosc/12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	fosc/64 or fosc/32
1	1	3	9-Bit UART	Variable

The baud rate in mode 2 depends on the value of bit SMOD in special function register PCON (see Section 3.11). If SMOD = 0 (which is the value on reset), the baud rate is $\frac{1}{64}$ of the oscillator frequency. If SMOD = 1, the baud rate is $\frac{1}{32}$ of the oscillator frequency.

$$\text{Mode 2 baud rate} = \frac{2^{\text{SMOD}}}{64} \text{ x (oscillator frequency)}$$

The baud rates in modes 1 and 3 are determined by the timer 1 overflow rate or can be generated by the internal baud rate generator.

3.4.2.1 Using Timer 1 to Generate Baud Rates

When timer 1 is used as the baud rate generator, the baud rates in modes 1 and 3 are determined by the timer 1 overflow rate and the value of SMOD as follows:

$$\text{Modes 1, 3 baud rate} = \frac{2^{\text{SMOD}}}{32} \text{ x (timer 1 overflow rate)}$$

The timer 1 interrupt should be disabled in this application. The timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications, it is configured for "timer" operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case, the baud rate is given by the formula:

$$\text{Modes 1, 3 baud rate} = \frac{2^{\text{SMOD}}}{32} \text{ x } \frac{\text{oscillator frequency}}{12 \text{ x } [256 - (\text{TH1})]}$$

One can achieve very low baud rates with timer 1 by leaving the timer 1 interrupt enabled, configuring the timer to run as 16-bit timer (high nibble of TMOD = 0001B), and using the timer 1 interrupt to do a 16-bit software reload.

Table 8 lists various commonly used baud rates and how they can be obtained from timer 1.

3.4.2.2 Internal Baud Rate Generator

In modes 1 and 3, the SAB 80515 provides the possibility to use the internal baud rate generator. To enable this feature, bit BD (bit 7 of special function register ADCON) must be set. This baud rate generator divides the oscillator frequency by 2500. Bit SMOD (PCON.7) can be used to divide the resulting frequency by 2. At 12 MHz oscillator frequency, the commonly used baud rates 4800 baud (SMOD = 0) and 9600 baud (SMOD = 1) are available. The baud rate is determined by SMOD and the oscillator frequency as follows:

$$\text{Modes 1, 3 baud rate} = \frac{2^{\text{SMOD}}}{2500} \text{ x (oscillator frequency)}$$

The following sections give a more detailed description of the operational modes of the serial port.

3.4.3 More About Mode 0

Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Figures 29a and 29b show a simplified functional diagram of the serial port in mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write-to-SBUF" signal at S6P2 also loads a 1 into the 9th bit position of the transmit shift register and tells the TX control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between "write-to-SBUF" and activation of SEND.

Table 8. Timer 1 Generated Commonly Used Baud Rates

Baud Rate		fosc MHz	SMOD	Timer 1		
				C/$\overline{T}$	Mode	Reload Value
Mode 0 max.:	1 MHz	12.0	X	X	X	X
Mode 2 max.:	375 kbaud	12.0	1	X	X	X
Mode 1, 3:	62.5 kbaud	12.0	1	0	2	FFH
	19.2 kbaud	11.059	1	0	2	FDH
	9.6 kbaud	11.059	0	0	2	FDH
	4.8 kbaud	11.059	0	0	2	FAH
	2.4 kbaud	11.059	0	0	2	F4H
	1.2 kbaud	11.059	0	0	2	E8H
	137.5 kbaud	11.986	0	0	2	1DH
	110 Baud	6.0	0	0	2	72H
	110 Baud	12.0	0	0	1	FEEBH

SEND enables the output of the shift register to the alternate output function line of P3.0, and also enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1, and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift register is shifted one position to the right.

As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX control block to do one last shift and then deactivate SEND and set TI. Both of these actions occur at S1P1 in the 10th machine cycle after "write-to-SBUF".

Reception is initiated by the condition REN = 1 and RI = 0. At S6P2 in the next machine cycle, the RX control unit writes the bits 1111 1110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1 SHIFT CLOCK makes transitions at S3P1 and S6P1 in every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted one position to the left. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 in the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX control block to do one last shift and load SBUF. At S1P1 in the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.

3.4.4 More About Mode 1

Ten bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB8 in SCON.

The baud rate is determined by the timer 1 overflow rate or by the internal baud rate generator.

Figures 30a and 30b show a simplified functional diagram of the serial port in mode 1, and associated timings for transmit and receive.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write-to-SBUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX control block that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the "write-to-SBUF" signal).

The transmission begins with activation of $\overline{\text{SEND}}$, which puts the start bit to TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX control unit to do one last shift and then deactivate $\overline{\text{SEND}}$ and set TI. This occurs at the 10th divide-by-16 rollover after "write-to-SBUF".

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in mode 1 is a 9-bit register), it flags the Rx control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1) RI = 0, and

2) either SM2 = 0 or the received stop bit = 1

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, no matter whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RxD.

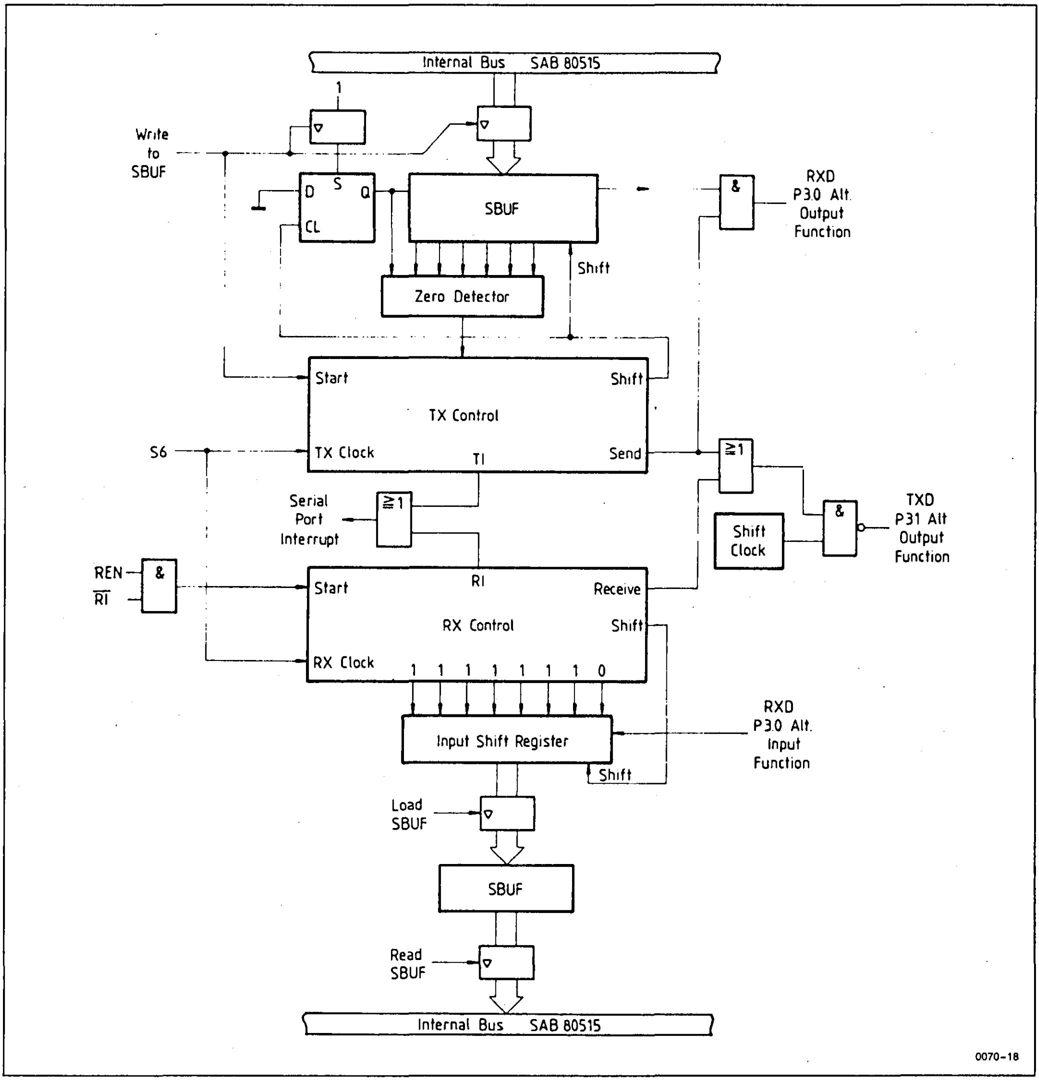


Figure 29a. Serial Port Mode 0, Function Diagram

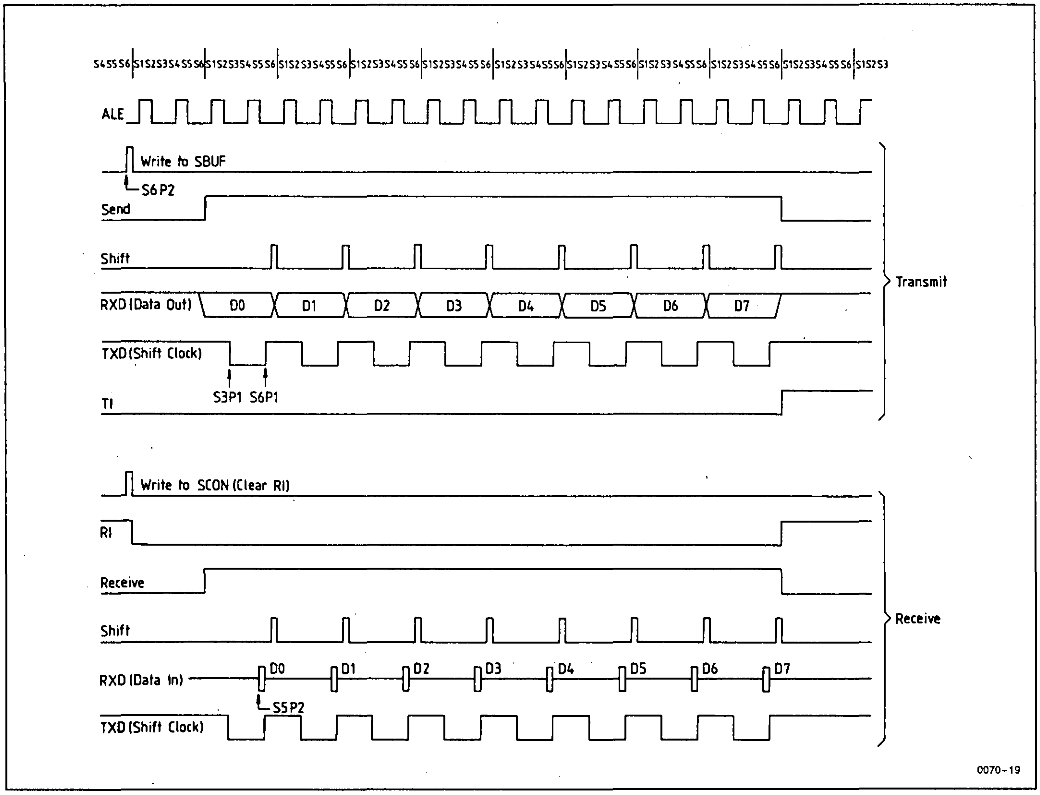

S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1 S2 S3
ALE
Write to SBUF
S6 P2
Send
Shift
RXD (Data Out)
D0
D1
D2
D3
D4
D5
D6
D7
TXD (Shift Clock)
S3P1
S6P1
TI
Transmit
Write to SCON (Clear RI)
RI
Receive
Shift
RXD (Data In)
D0
D1
D2
D3
D4
D5
D6
D7
S5 P2
TXD (Shift Clock)
Receive
0070-19

Figure 29b. Serial Port Mode 0, Timing

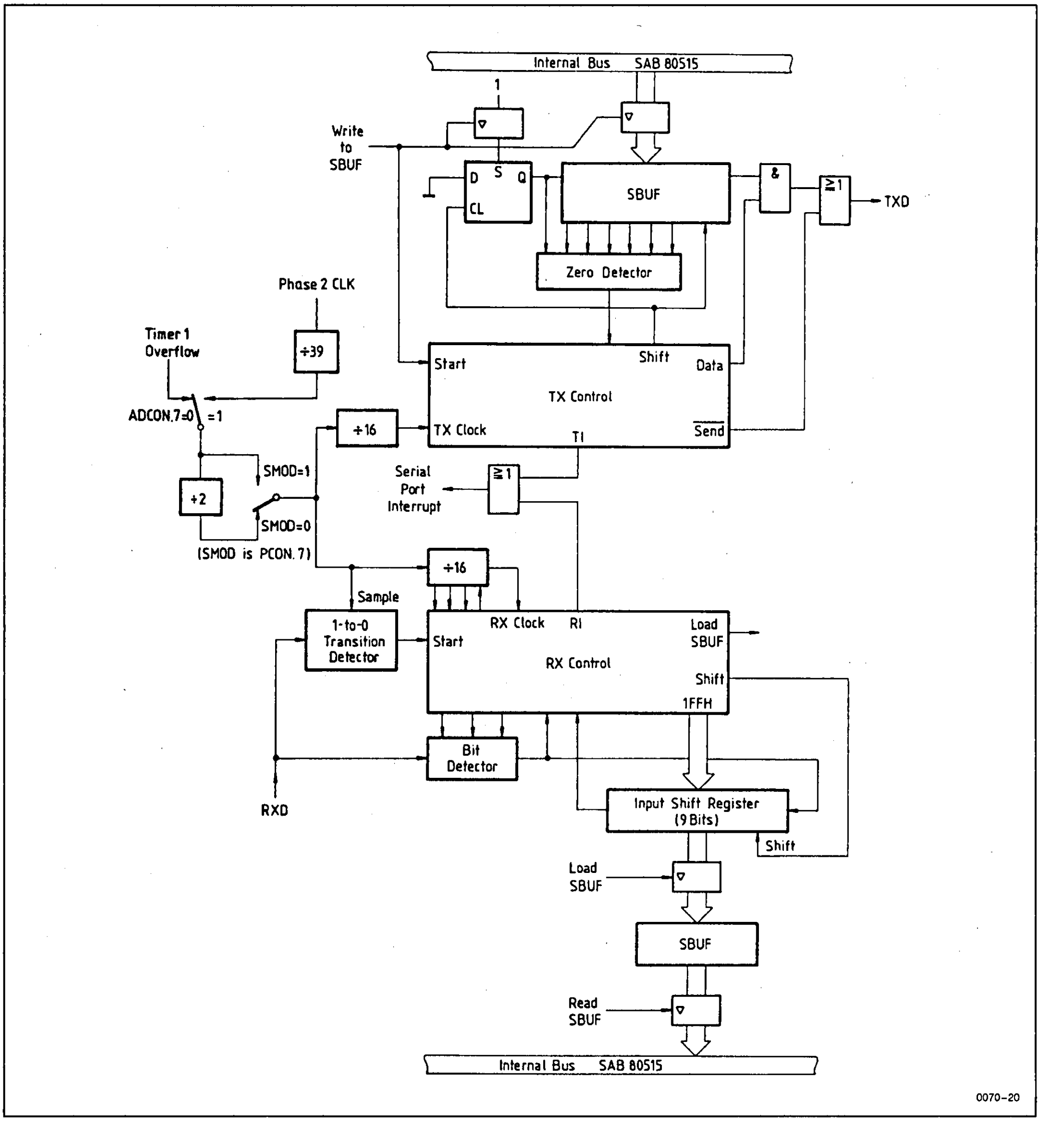


Figure 30a. Serial Port Mode 1, Functional Diagram

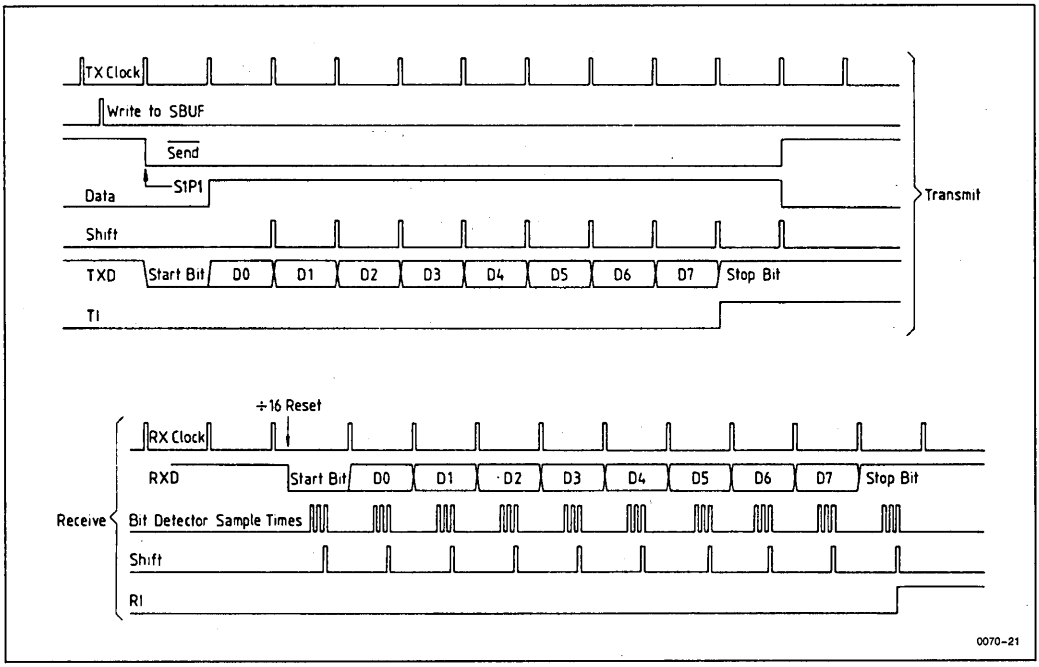


Figure 30b. Serial Port Mode 1, Timing

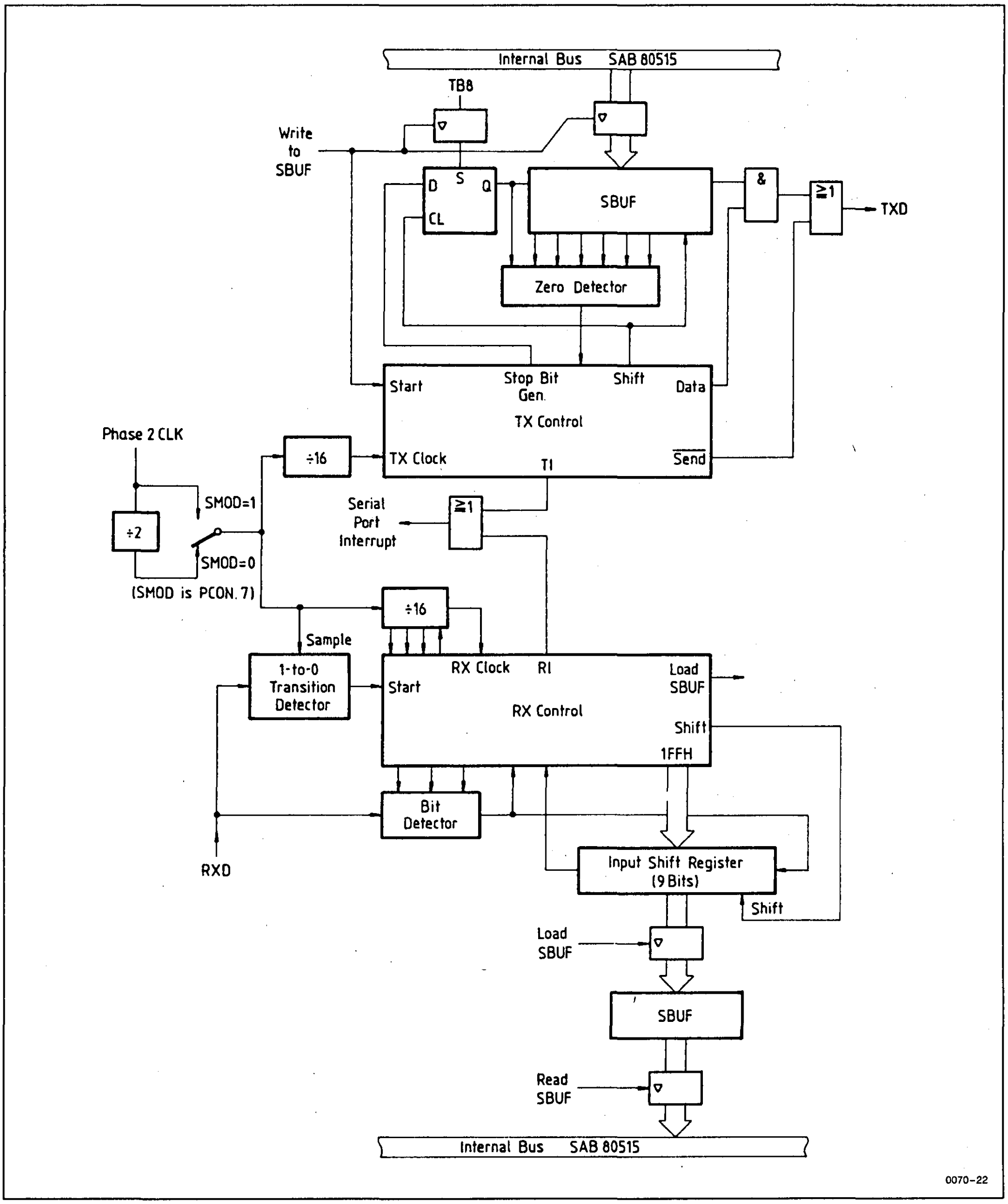


Figure 31a. Serial Port Mode 2, Functional Diagram

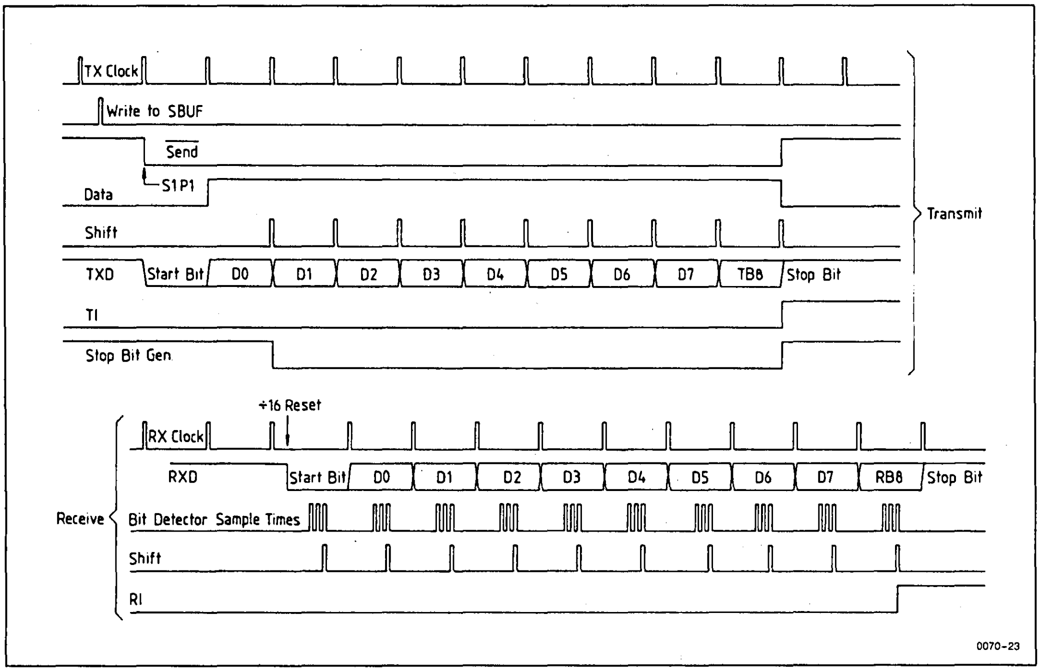


Figure 31b. Serial Port Mode 2, Timing

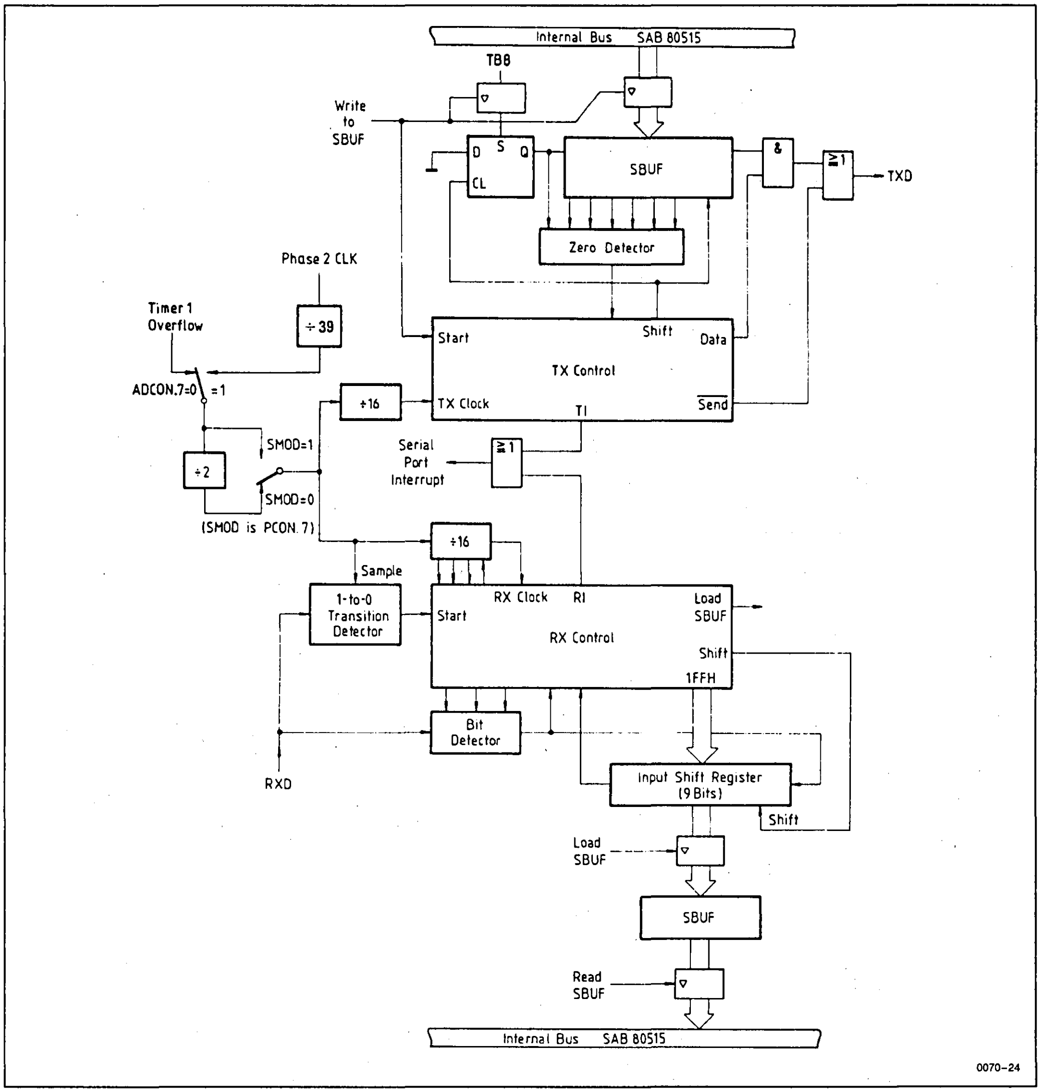


Figure 31c. Serial Port Mode 3, Functional Diagram

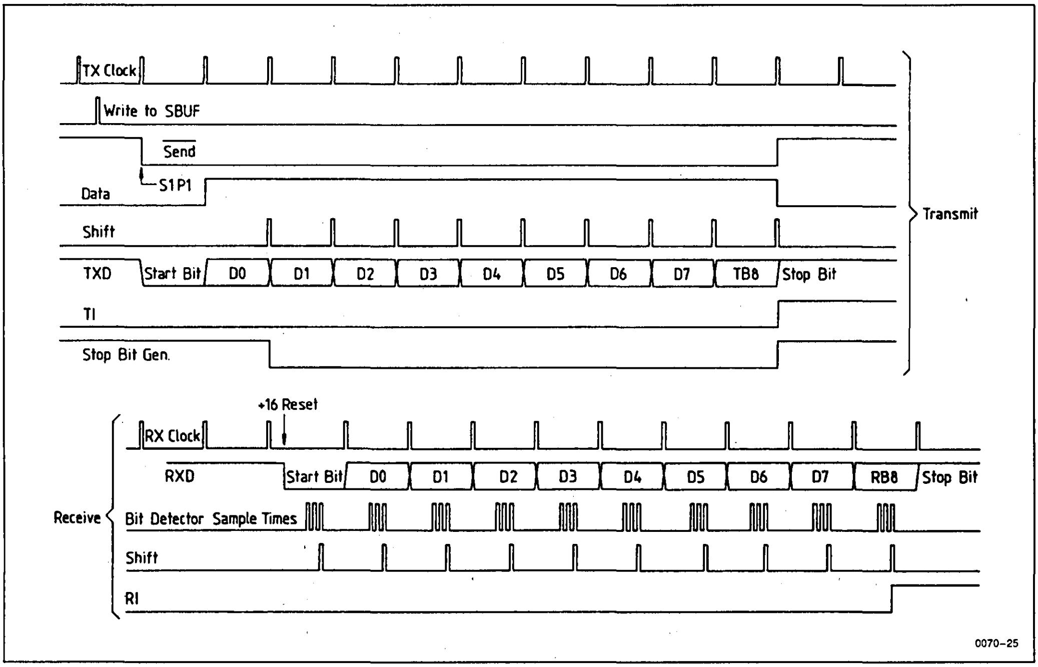


Figure 31d. Serial Port Mode 3, Timing

3.4.5 More About Modes 2 and 3

Eleven bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and stop bit (1). On transmission, the 9th data bit (TB8) can be assigned the value of 0 or 1. On reception, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency in mode 2. Mode 3 may have a variable baud rate generated from timer 1 or by the internal baud rate generator.

Figures 31 a, b, c, and d show a functional diagram of the serial port in modes 2 and 3 and associated timings. The receive portion is exactly the same as in mode 1. The transmit portion differs from mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write-to-SBUF" signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the "write-to-SBUF" signal).

The transmission begins with activation of $\overline{\text{SEND}}$, which puts the start bit to TxD. One bit time later, DATA is activated which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeros are clocked in. Thus, as data bits shift out to the right, zeros are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just left of the TB8, and all positions to the left of that contain zeros.

This condition flags the TX control unit to do one last shift and then deactivate $\overline{\text{SEND}}$ and set TI. This occurs at the 11th divide-by-16 rollover after "write-to-SBUF".

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least

2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in modes 2 and 3 is a 9-bit register), it flags the RX control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set R1, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1) RI = 0, and
2) either SM2 = 0 or the received 9th data bit = 1

If either of these two conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th bit goes into RB8, the first 8 data bits go into SBUF. One bit time later, no matter whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition at the RxD input.

Note that the value of the received stop bit is irrelevant to SBUF, RB8, or RI.

3.5 A/D Converter

The SAB 80515 provides an 8-bit A/D converter with eight multiplexed analog input channels on-chip. In addition, the A/D converter has a sample and hold circuit and offers the feature of software-programmable reference voltages. For the conversion, the method of successive approximation with a capacitor network is used.

Figure 32 shows a block diagram of the A/D converter. There are three user-accessible special function registers: ADCON (A/D converter control register), ADDAT (A/D converter data register), and DAPR (D/A converter program register) for the programmable reference voltages.

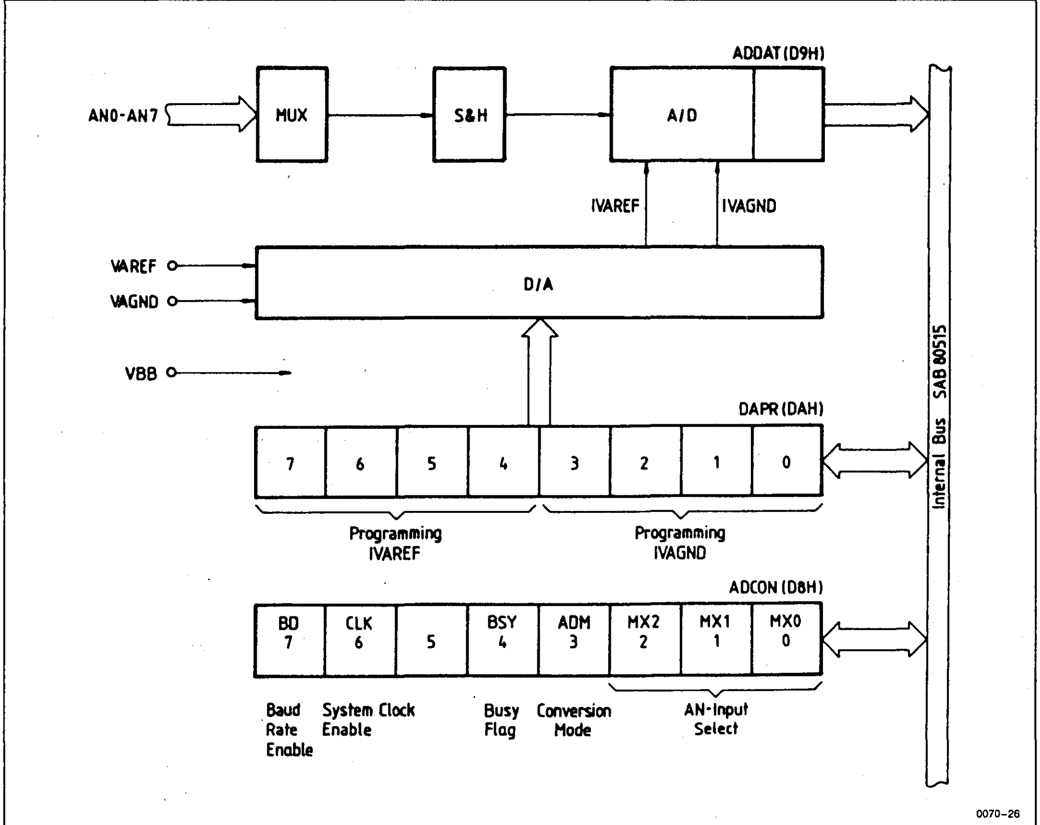


Figure 32. A/D Converter Block Diagram

Special function register ADCON, which is illustrated in Figure 33, is used to select one of the eight analog input channels to be converted, to specify a single or continuous conversion, and to check the status bit BSY which signals whether a conversion is in progress or not.

BD	CLK	—	BSY	ADM	MX2	MX1	MX0	Bit
0DFH	0DEH	0DDH	0DCH	0DBH	0DAH	0D9H	0D8H	Address

Symbol	Position	Function
MX0	ADCON.0	Analog Input Channel Selection, see Table 9.
MX1	ADCON.1	
MX2	ADCON.2	
ADM	ADCON.3	A/D Conversion Mode. When set, a continuous conversion is selected. If ADM = 0, the converter stops after one conversion.
BSY	ADCON.4	Busy Flag. This flag indicates whether a conversion is in progress (BSY = 1) or not (BSY = 0).
—	ADCON.5	Reserved (must be 0).
CLK	ADCON.6	System Clock Enable. When set, a clock signal with 1/12 the oscillator frequency is gated to pin P1.6/CLKOUT (see Section 3.9). CLK = 0 disables the clock output.
BD	ADCON.7	Baud Rate Enable. When set, the baud rate in mode 1 and 3 of the serial port is taken from the internal baud rate generator (see Section 3.4.2.2).

Figure 33. A/D Converter Control Register ADCON (0D8H)

Table 9. Selection of the Analog Input Channels

MX2	MX1	MX0	Selected Channel	Pin
0	0	0	Analog Input 0	AN0
0	0	1	Analog Input 1	AN1
0	1	0	Analog Input 2	AN2
0	1	1	Analog Input 3	AN3
1	0	0	Analog Input 4	AN4
1	0	1	Analog Input 5	AN5
1	1	0	Analog Input 6	AN6
1	1	1	Analog Input 7	AN7

The special function register ADDAT holds the converted digital 8-bit data result. The data remains in ADDAT until it is overwritten by the next converted data. The new converted value will appear in ADDAT in the 15th machine cycle after a conversion has been started. ADDAT can be read and written to under software control. If the A/D converter of the SAB 80515 is not used, register ADDAT can be used as an addition general-purpose register.

3.5.1 Programming the Internal Reference Voltages

The SFR DAPR is provided for programming the internal reference voltages IVAREF and IVAGND. For this purpose the internal reference voltages can be programmed in steps of 1/16 with respect to the external reference voltages (VAREF − VAGND) by four bits each in register DAPR. Bits 0 to 3 specify IVAGND, while bits 4 to 7 specify IVAREF. A minimum of 1V difference is required between the internal reference voltages for proper operation of the A/D converter. That means, the internal reference voltage IVAREF must always be programmed four steps higher than IVAGND (in respect of the external reference voltage VAREF which is specified as V_{CC} ±5%). The values of IVAGND and IVAREF are given by the formula:

$$\text{IVAGND} = \text{VAGND} + \frac{\text{DAPR}(0-3)}{16}(\text{VAREF} - \text{VAGND})$$

with $\text{DAPR}(0-3) \neq 0 \text{ and } \text{DAPR}(0-3) < 13$;

$$\text{IVAREF} = \text{VAGND} + \frac{\text{DAPR}(4-7)}{16}(\text{VAREF} - \text{VAGND})$$

with $\text{DAPR}(4-7) > 3$;

where DAPR(0−3) is the contents of the low-order nibble, and DAPR(4-7) the contents of the high-order nibble of DAPR, taken as an unsigned decimal integer.

If DAPR(0–3) or DAPR(4–7) = 0, the internal reference voltages correspond to the external reference voltages VAGND and VAREF, respectively.

If VAINPUT > IVAREF, the conversion result is 0FFH, if VAINPUT < IVAGND, the conversion result is 00H (VAINPUT is the analog input voltage).

Figure 34 shows special function register DAPR.

7	6	5	4	3	2	1	0	Bit
Digital Value for IVAREF				Digital Value for IVAGND				

Figure 34. D/A Converter Program Register DAPR (0DAH)

If the external reference voltages VAGND = 0V and VAREF = +5V (in respect of GND and VCC) are applied, then the following internal reference voltages IVAGND and IVAREF shown in Table 10 can be adjusted via the special function register DAPR.

Table 10. Adjustable Internal Reference Voltages

Step	DAPR(0–3)
	DAPR(4–7)
		IVAGND
		(V)
			IVAREF
			(V)

0	0000	0.0	5.0
1	0001	0.3125	—
2	0010	0.625	—
3	0011	0.9375	—
4	0100	1.25	1.25
5	0101	1.5625	1.5625
6	0110	1.875	1.875
7	0111	2.1875	2.1875
8	1000	2.5	2.5
9	1001	2.8125	2.8125
10	1010	3.125	3.125
11	1011	3.4375	3.4375
12	1100	3.75	3.75
13	1101	—	4.0625
14	1110	—	4.375
15	1111	—	4.6875

Items marked with "—" are not allowed according to the rules listed before (IVAREF at least four steps higher than IVAGND).

3.5.2 A/D Converter Timing and Conversion Time

A conversion is started by writing into special function register DAPR. A "write-to-DAPR" will start a new conversion even if a conversion is currently in progress. The conversion begins with the next machine cycle. The busy flag will be set in the same machine cycle as the "write-to-DAPR" operation occurs. If a value is written to DAPR the A/D conversion starts and the conversion time is 15 μs at 12 MHz oscillator frequency.

After a conversion has been started by writing into SFR DAPR, the analog voltage at the selected input channel is sampled for 5 machine cycles (5 μs at 12 MHz oscillator frequency), which will then be held at the sampled level for the rest of the conversion time.

The most critical time of the sample period is the load time. The load time t_L is part of the sample time, t_S, and it last for 2 machine cycles. It is the time during which the entire internal capacitance of the A/D converter is charged by the analog source. The remaining 3 machine cycles of the sample time are used for adjusting the comparator of the A/D converter. The output impedance of the analog source must be low enough to assure full loading of the sample and hold capacitance during load time, t_L. After charging the internal capacitance of the A/D converter during load time t_L, the analog input must be held constant for the rest of the sample time t_S.

Conversion of the sampled analog voltage takes place between the 6th and 15th machine cycle after sampling has been completed. In the 15th machine cycle the converted result is moved to ADDAT, the busy flag (BSY) is cleared, and the A/D converter interrupt request flag IADC (bit 0 in SFR interrupt control register IRCON, see Section 3.6) is set. If a continuous conversion is established, the next conversion is automatically started in the following machine cycle.

The special feature of programmable internal reference voltages allows adjusting the internal voltage range to the range of the external analog input voltage. Or it may be used to increase the resolution of the converted analog input voltage by starting a second conversion with a compressed internal reference voltage range closely to the previously measured analog value. Figures 35a and 35b illustrate these applications.

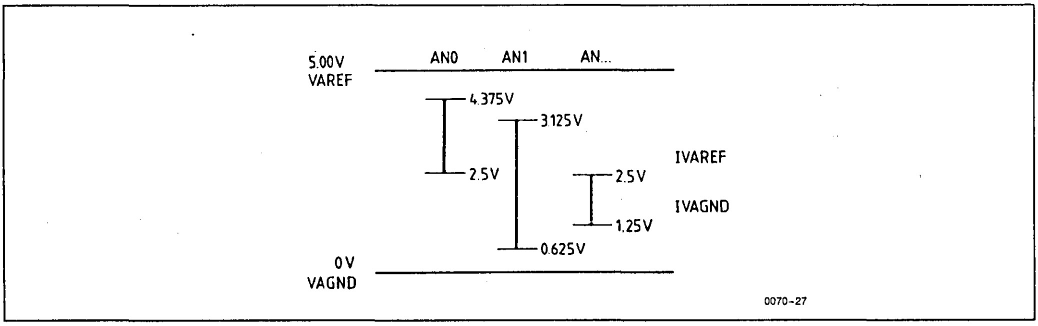


Figure 35a. Adjusting the Internal Reference Voltages to the Range of the External Analog Voltages

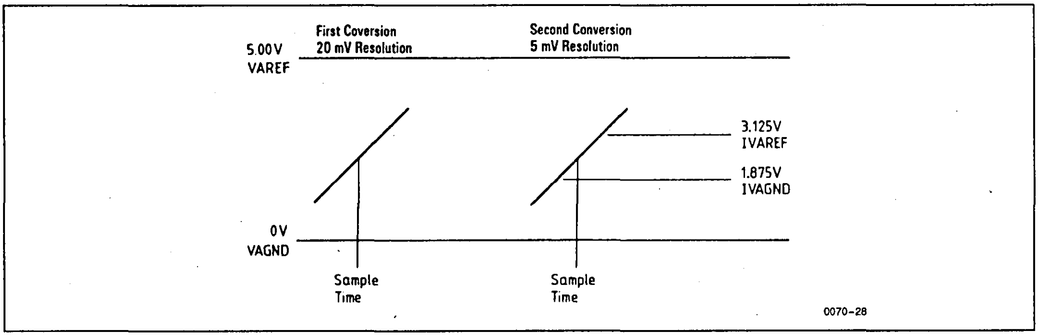


Figure 35b. Increasing the Resolution of the A/D Result by Doing a Second Conversion

3.6 Interrupt Structure

The interrupt structure of the SAB 80515 provides 12 interrupt sources and 4 priority levels. The 12 interrupt sources are organized as 6 pairs. Table 11 lists the interrupt sources and pairs of the SAB 80515.

Table 11. Interrupt Sources

External Interrupt 0	A/D Converter Interrupt
Timer 0 Interrupt	External Interrupt 2
External Interrupt 1	External Interrupt 3
Timer 1 Interrupt	External Interrupt 4
Serial Port Interrupt	External Interrupt 5
Timer 2 Interrupt	External Interrupt 6

Some of these interrupt sources are activated by one, others are activated by two internal or external events. Each interrupt source has its own vector location in the program memory address space 00H to 6BH. In the following section the interrupt sources are discussed separately.

The external interrupts 0 and 1 ($\overline{\text{INT0}}$ and $\overline{\text{INT1}}$) can each be either level-activated or negative transition-activated, depending on bits IT0 and IT1 in register TCON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated this interrupt is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source directly controls the request flag, rather than the on-chip hardware.

The timer 0 and timer 1 interrupts are generated by TF0 and TF1, which are set by a rollover in their respective timer/counter registers (except see Section 3.2.4 for timer 0 in mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The serial port interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared in software.

The timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register IRCON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared in software.

The A/D converter interrupt is generated by bit IADC in register IRCON. It is set in the 15th machine cycle, after a conversion has been started by a "write-to-DAPR", or, if continuous conversions are established, after the last conversion has been completed, depending on whether the internal reference voltages IVAGND and IVAREF have to be adjusted or not. When an A/D converter interrupt is generated, flag IADC will have to be cleared in software.

The external interrupt 2 ($\overline{\text{INT2}}$) can be either positive or negative transition-activated, depending on bit I2FR in register T2CON. The flag that actually generates this interrupt is bit IEX2 in register IRCON. If an external interrupt 2 is generated, flag IEX2 is cleared by hardware when the service routine is vectored to.

Like the external interrupt 2, the external interrupt 3 can be either positive or negative transition-activated, depending on bit I3FR in register T2CON. The flag that actually generates this interrupt is bit IEX3 in register IRCON. In addition, this flag will be set if a compare event occurs at pin P1.0/$\overline{\text{INT3}}$/CC0 (timer 2 registers contents matches the contents of the CRC register), regardless of the compare mode established, the transition occurring at the pin, and of the external interrupt 3 being positive or negative transition-activated. Flag IEX3 is cleared by the on-chip hardware when the service routine is vectored to.

The external interrupts 4 (INT4), 5 (INT5), and 6 (INT6) are positive transition-activated. The flags that actually generate these interrupts are bits IEX4, IEX5, and IEX6 in register IRCON. In addition, these flags will be set if a compare event occurs at the corresponding output pin P1.1/INT4/CC1, P1.2/INT5/CC2, and P1.3/INT6/CC3, regardless of the compare mode established and the transition at the respective pin. When an interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

All of these bits that generate interrupts can be set or cleared by software, with the same result as though they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be canceled in software. The only exception are request flags IE0 and IE1. If the external interrupts 0 and 1 are programmed to be level-activated, IE0 and IE1 are controlled by the external source via pin $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, respectively. Thus, writing a one to these bits will not set the request flags IE0 and/or IE1. In this mode, external interrupts 0 and 1 can only be generated in software by writing a 0 to the corresponding pins $\overline{\text{INT0}}$ (P3.2) and $\overline{\text{INT1}}$ (P3.3), provided this will not affect any peripheral circuit connected to the pins.

Figure 36 shows the special function register IR-CON.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in the special function registers IEN0 and IEN1 (Figure 37a and 37b). Note that IEN0 contains also a global disable bit, EAL, which disables all interrupts at once. Also note that in the SAB 8051 the interrupt priority register IP is located at address 0B8H; in the SAB 80515 this location is occupied by register IEN1.

EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	Bit
0C7H	0C6H	0C5H	0C4H	0C3H	0C2H	0C1H	0C0H	Address

Symbol	Position	Function
IADC	IRCON.0	A/D Converter Interrupt Request Flag. Set by hardware at the end of a conversion. Must be cleared by software.
IEX2	IRCON.1	External Interrupt 2 Edge Flag. Set by hardware when external interrupt edge was detected. Cleared when interrupt processed.
IEX3	IRCON.2	External Interrupt 3 Edge Flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin P1.0/$\overline{\text{INT3}}$/CC0. Cleared when interrupt processed.
IEX4	IRCON.3	External Interrupt 4 Edge Flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin P1.1/INT4/CC1. Cleared when interrupt processed.
IEX5	IRCON.4	External Interrupt 5 Edge Flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin P1.2/INT5/CC2. Cleared when interrupt processed.
IEX6	IRCON.5	External Interrupt 6 Edge Flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin P1.3/INT6/CC3. Cleared when interrupt processed.
TF2	IRCON.6	Timer 2 Overflow Flag. Set by a timer 2 overflow and must be cleared by software. If the timer 2 interrupt is enabled, TF2 = 1 will cause an interrupt.
EXF2	IRCON.7	Timer 2 External Reload Flag. Set when a reload is caused by a negative transition on pin T2EX and EXEN2 = 1. When the timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the timer 2 interrupt routine. Can be used as an additional external interrupt when the reload function is not used. EXF2 must be cleared by software.

Figure 36. Interrupt Request Control Register IRCON (0C0H)

EAL	WDT	ET2	ES	ET1	EX1	ET0	EX0	Bit
0AFH	0AEH	0ADH	0ACH	0ABH	0AAH	0A9H	0A8H	Address

Symbol	Position	Function
EX0	IEN0.0	Enables or Disables External Interrupt 0. If EX0 = 0, external interrupt 0 is disabled.
ET0	IEN0.1	Enables or Disables the Timer 0 Overflow Interrupt. If ET0 = 0, the timer 0 interrupt is disabled.
EX1	IEN0.2	Enables or Disables External Interrupt 1. If EX1 = 0, external interrupt 1 is disabled.
ET1	IEN0.3	Enables or Disables the Timer 1 Overflow Interrupt. If ET1 = 0, the timer 1 interrupt is disabled.
ES	IEN0.4	Enables or Disables the Serial Port Interrupt. If ES = 0, the serial port interrupt is disabled.
ET2	IEN0.5	Enables or Disables the Timer 2 Overflow or External Reload Interrupt. If ET2 = 0, the timer 2 interrupt is disabled.
WDT	IEN0.6	Watchdog Timer Reset Flag. Set to initiate a reset of the watchdog timer (details in Section 3.7).
EAL	IEN0.7	Enables or Disables All Interrupts. If EAL = 0, no interrupt will be acknowledged. If EAL = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.

Figure 37a. Interrupt Enable Register IEN0 (0A8H)

EXEN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	Bit
0BFH	0BEH	0BDH	0BCH	0BBH	0BAH	0B9H	0B8H	Address

Symbol	Position	Function
EADC	IEN1.0	Enables or Disables the A/D Converter Interrupt. If EADC = 0, the A/D converter interrupt is disabled.
EX2	IEN1.1	Enables or Disables External Interrupt 2. If EX2 = 0, external interrupt 2 is disabled.
EX3	IEN1.2	Enables or Disables External Interrupt 3/Capture/Compare Interrupt 0. If EX3 = 0, external interrupt 3 is disabled.
EX4	IEN1.3	Enables or Disables External Interrupt 4/Capture/Compare Interrupt 1. If EX3 = 0, external interrupt 4 is disabled.
EX5	IEN1.4	Enables or Disables External Interrupt 5/Capture/Compare Interrupt 2. If EX5 = 0, external interrupt 5 is disabled.
EX6	IEN1.5	Enables or Disables External Interrupt 6/Capture/Compare Interrupt 3. If EX6 = 0, external interrupt 6 is disabled.
SWDT	IEN1.6	Watchdog Timer Start/Reset Bit. Set to start/reset the watchdog timer (details in Section 3.7).
EXEN2	IEN1.7	Enables or Disables the Timer 2 External Reload Interrupt. EXEN2 = 0 disables the timer 2 external reload interrupt. The external reload function is not affected by EXEN2.

Figure 37b. Interrupt Enable Register IEN1 (0B8H)

3.6.1 Priority Level Structure

Each pair of interrupt sources can be programmed individually to one of four priority levels by setting or clearing one bit in the special function register IP0 and one in IP1 (Figure 38). A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another interrupt of the same or a lower priority. An interrupt of the highest priority level can't be interrupted by another interrupt source.

If two or more requests of different priority levels are received simultaneously, the requests of the highest priority is serviced first. If request of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced first. If requests from two interrupt sources of one interrupt pair are received simultaneously, the "left" interrupt source of each pair is serviced first. Thus within each priority level there is a second priority structure determined by the polling sequence, as follows:

High →	Low	Priority
Interrupt Source Pair		
IE0	IADC	High
TF0	IEX2	
IE1	IEX3	↓
TF1	IEX4	
RI + TI	IEX5	Low
TF2 + EXF2	IEX6	

Note that the "priority within level" structure is only used to resolve simultaneous requests of the same priority level.

Figure 39 shows a block diagram of the priority level structure and Figure 40 illustrates the requesting sources of the SAB 80515's interrupt structure.

—	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0

—	—	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0

The priority level of each pair of interrupt sources is determined by corresponding bits in IP0 and IP1 as follows:

Bits		Corresponding Interrupt Pair
IP1.0	IP0.0	IE0/IADC
0	0	Priority Level 0 (Lowest)
0	1	Priority Level 1
1	0	Priority Level 2
1	1	Priority Level 3 (Highest)
IP1.1	IP0.1	TF0/IEX2
IP1.2	IP0.2	IE1/IEX3
IP1.3	IP0.3	TF1/IEX4
IP1.4	IP0.4	RI+TI/IEX5
IP1.5	IP0.5	TF2+EXF2/IEX6

IP0.6 is the watchdog timer status bit WDTS. IP0.7, IP1.6, and IP1.7 are reserved.

Figure 38. Interrupt Priority Registers IP0 (0A9H) and IP1 (0B9H)

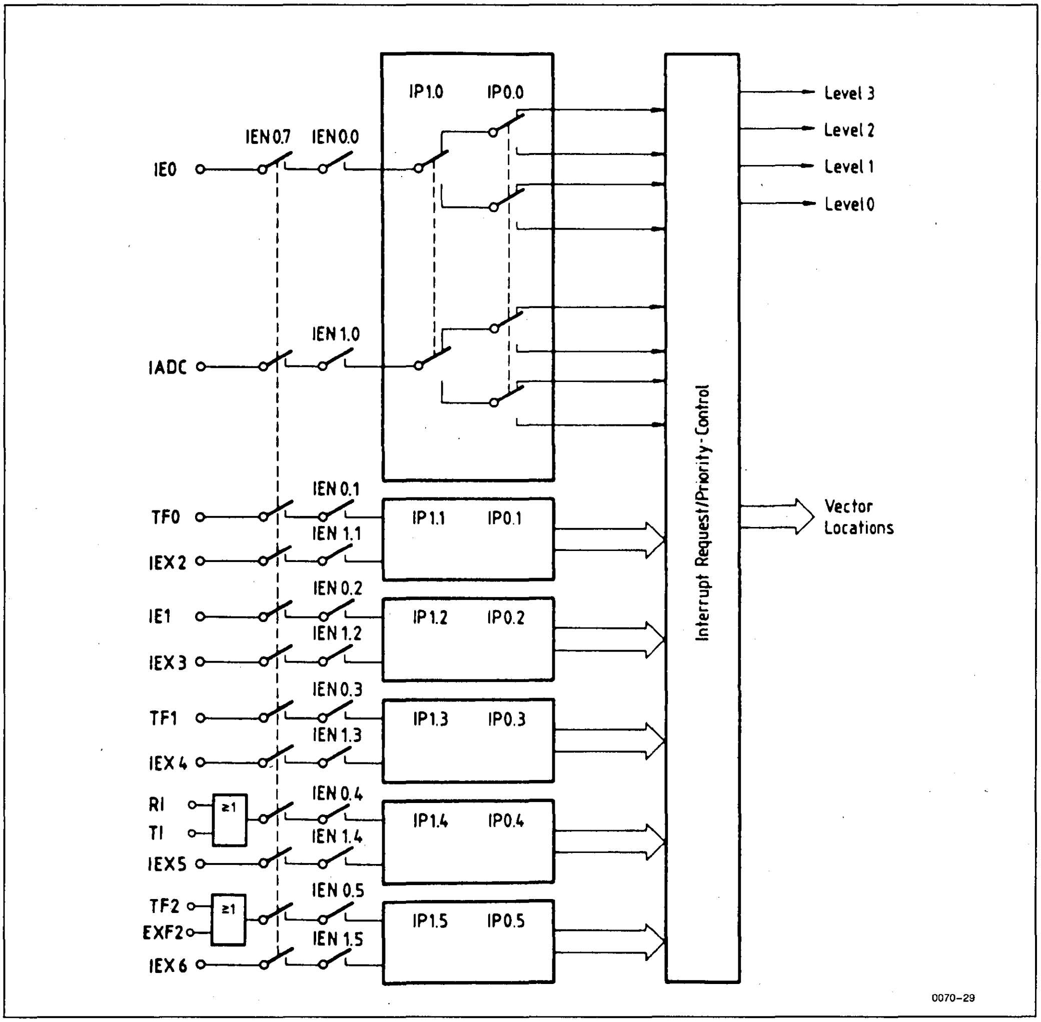


Figure 39. Interrupt Priority Level Structure

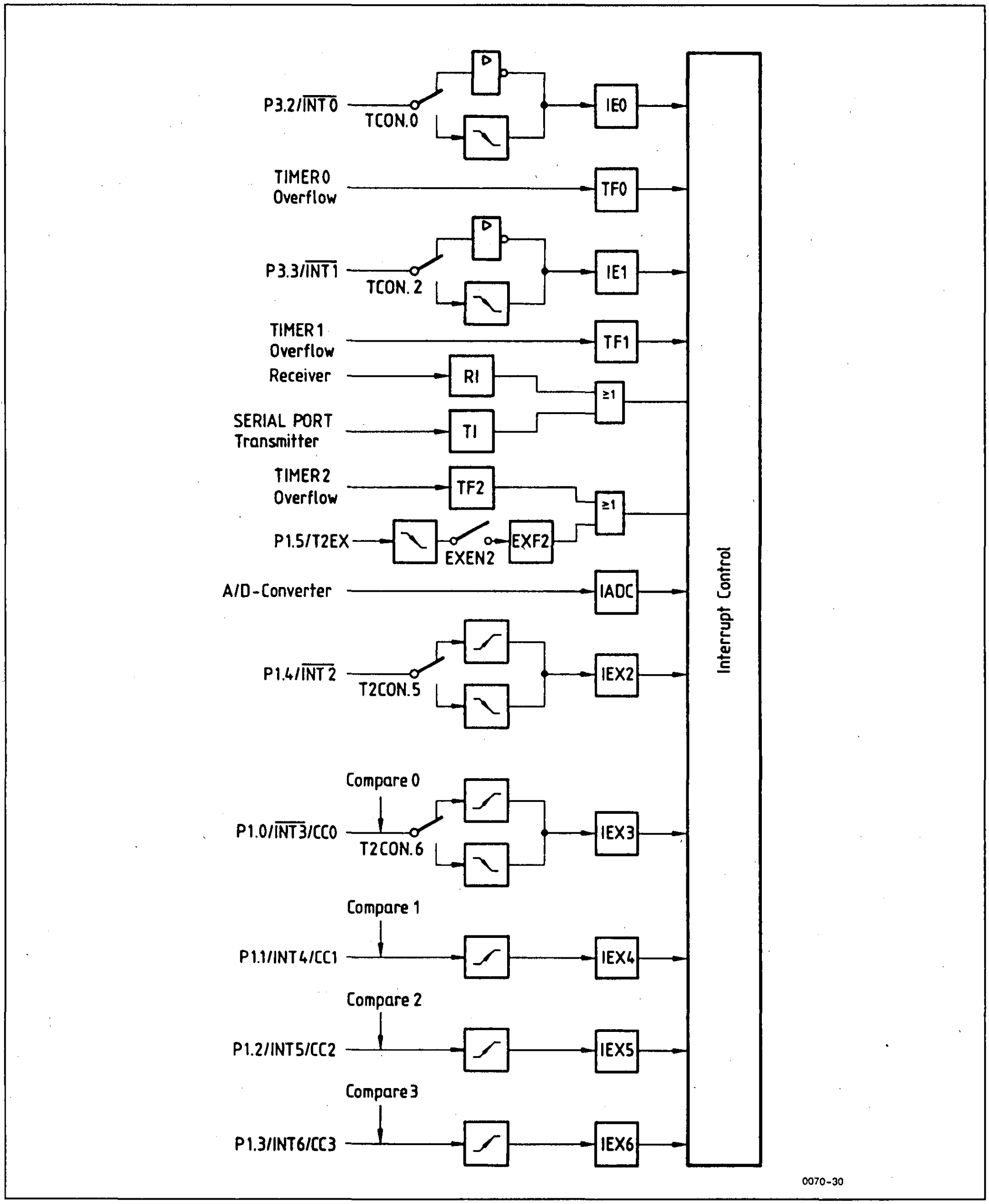


Figure 40. Interrupt Requesting Sources

3.6.2 How Interrupts Are Handled

The interrupt flags are sampled at S5P2 in every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1) An interrupt of equal or higher priority is already in progress.
2) The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3) The instruction in progress is RETI or a write access to any of the registers IEN0, IEN1, IP0, or IP1.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to registers IEN0, IEN1, IP0, or IP1, then at least one more instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with every machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note then that if any interrupt flag is active but not being responded to for one of the above conditions, if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in Figure 41.

Note that if an interrupt of higher priority level goes active prior to S5P2 in the machine cycle labeled C3 in Figure 41, then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine being executed.

Thus the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it doesn't. It never clears the serial port (RI, TI), timer 2 (TF0, EXF2), or A/D converter flags. This has to be done in the user's software. It clears an external interrupt flag (IE0 or IE1) only if it was transition-activated. External interrupt flags IEX2 to IEX6 are always cleared. The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to, as shown below.

Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI+TI	0023H
TF2+EXF2	002BH
IADC	0043H
IEX2	004BH
IEX3	0053H
IEX4	005BH
IEX5	0063H
IEX6	006BH

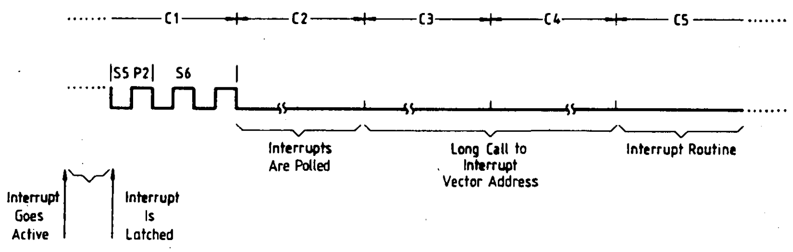


Figure 41. Interrupt Response Timing Diagram

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from where it was left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress.

3.6.3 External Interrupts

The external interrupts 0 and 1 can be programmed to be level-activated or negative transition-activated by setting or clearing bit IT0 or IT1, respectively, in register TCON. If ITx = 0 (x = 0 or 1), external interrupt x is triggered by a detected low at the $\overline{\text{INTx}}$ pin. If ITx = 1, external interrupt x is negative edge-triggered In this mode, if successive samples of the $\overline{\text{INTx}}$ pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

If the external interrupt 0 or 1 is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

The external interrupts 2 and 3 can be programmed to be negative or positive transition-activated by setting or clearing bit I2FR or I3FR in register T2CON. If IxFR = 0 (x = 2 or 3), external interrupt x is negative transition-activated. If IxFR = 1, external interrupt x is triggered by a positive transition.

The external interrupts 4, 5, and 6 are activated by a positive transition. The external timer 2 reload trigger interrupt request flag EXF2 will be activated by a negative transition at pin P1.5/T2EX but only if bit EXEN2 is set.

Since the external interrupt pins ($\overline{\text{INT2}}$ to INT6) are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin low (high for $\overline{\text{INT2}}$ and $\overline{\text{INT3}}$, if they are programmed to be negative transition-active) for at least one cycle, and then hold it high (low) for at least one cycle to ensure that the transition is recognized so that the corresponding interrupt request flag will be set. The external interrupt request flags will automatically be cleared by the CPU when the service routine is called.

3.6.4 Response Time

If an external interrupt is recognized, its corresponding request flag is set at S5P2 in every machine cycle. The value is not actually polled by the circuitry until the next machine cycle. If the request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the request service routine will be the next instruction to be executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles will elapse between activation of an external interrupt request and the beginning of executing the first instruction of the service routine. Figure 41 shows interrupt response timings.

A longer response time would result if the request is blocked by one of the three previously listed conditions. If an interrupt of equal or higher priority level is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles, since the longest instructions (MUL and DIV) are only 4 cycles long; and, if the instruction in progress is RETI or an access to registers IEN0, IEN1, IP0, or IP1, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction if the instruction is MUL or DIV).

Thus, in a single interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

3.7 Watchdog Timer

As a means of graceful recovery from software or hardware upset a watchdog timer is provided in the SAB 80515. If the software fails to clear the watchdog timer at least every 65532 µs, an internal hardware reset will be initiated. The software can be designed such that the watchdog times out if the program does not progress properly. The watchdog will also time out if the software error was due to hardware-related problems. This prevents the controller from malfunctioning for longer than 65 ms if a 12 MHz oscillator is used.

The watchdog timer is a 16-bit counter which is incremented once every machine cycle. After an external reset the watchdog timer is disabled and cleared to 0000H. The counter is started by setting bit SWDT (bit 6 in SFR IEN1). After having been started, the watchdog timer cannot be stopped by software and bit WDTS (watchdog timer status, bit 6 in SFR IP0) is set. It can only be cleared to 0000H by first setting bit WDT (IEN0.6) and with the next instruction setting SWDT. Bit WDT will automatically be cleared during the third machine cycle after having been set. This double instruction clearing of the watchdog timer was implemented to minimize the chance of unintentionally clearing the watchdog. To prevent the watchdog from overflowing, it must be cleared periodically.

If the software fails to clear the watchdog in time, an internally generated watchdog reset is entered at the counter state FFFCH, which lasts four machine cycles. This internal reset differs from an external reset only to the extent that the watchdog timer is not disabled and bit WDTS (watchdog timer status, bit 6 in SFR IP0) is set. Bit WDTS allows the software to examine from which source the reset was initiated. If it is set, the reset was caused by a watchdog timer overflow.

3.8 RAM Backup Power Supply

The power-down mode in the SAB 80515 allows to reduce VCC to zero while saving 40 bytes of the on-chip RAM through a backup supply connected to the VPD pin. In the following, the terms VCC VPD are used to specify the voltages on pin VCC and pin VPD, respectively.

If VCC > VPD, the 40 bytes are supplied from VCC. VPD may then be low. If VCC < VPD, the current for the 40 bytes is drawn from VPD. The addresses of these backup-powered RAM locations range from 88 to 127 (58H to 7FH). The current drawn from the backup power supply is typically 1 mA, max. 3 mA.

To utilize this feature, the user's system—upon detecting that a power failure is imminent—would interrupt the processor in some manner to transfer relevant data to the 40 byte in on-chip RAM and enable the backup power supply to the VPD pin. Then a reset should be accomplished before VCC falls below its operating limit. When power returns, a power-on reset should be accomplished, and the backup supply needs to stay on long enough to resume normal operation. Figure 42 illustrates the timing on a power failure.

3.9 System Clock Output

For peripheral devices requiring a system clock, the SAB 80515 provides a clock output signal derived from the oscillator frequency as an alternate output function on pin P1.6/CLKOUT. If bit CLK is set (bit 6 of special function register ADCON), a clock signal with 1/12 the oscillator frequency is gated to pin P1.6/CLKOUT. To use this function the port 1 pin must first be programmed to a one (1).

Figure 43 shows the timing of this system clock signal with respect to signal ALE and the internal states. The system clock is high during S3P1 and S3P2 of every machine cycle and low during all other states. Thus, the duty cycle of the clock signal is 1:6. Also shown is the timing with respect to an external data memory access. The system clock coincides with the last state (S3) in which a $\overline{RD}$ or $\overline{WR}$ signal is active.

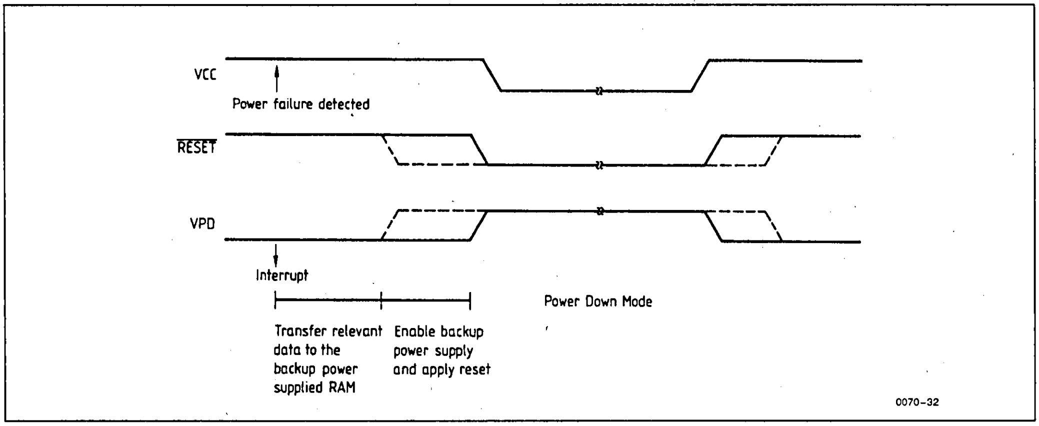

VCC
Power failure detected
RESET
VPD
Interrupt
Power Down Mode
Transfer relevant data to the backup power supplied RAM
Enable backup power supply and apply reset
0070-32

Figure 42. Reset and RAM Backup Power Timing

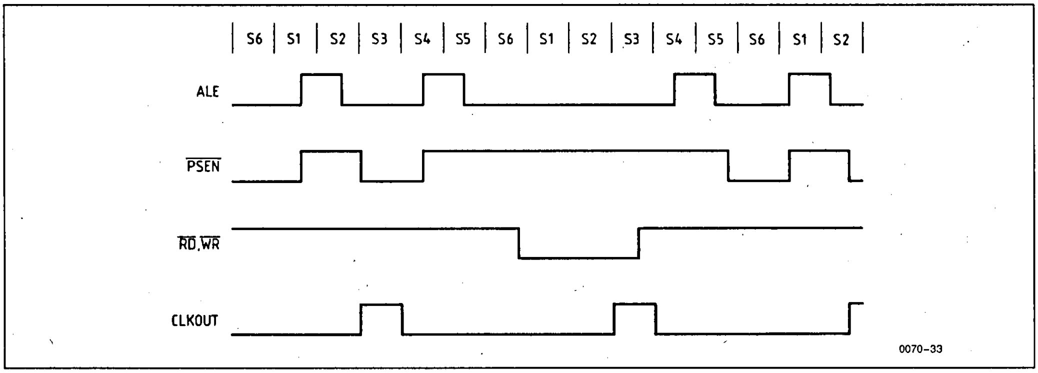


Figure 43. System Clock Timing

3.10 More about the On-Chip Oscillator

The on-chip oscillator of the SAB 80515, like in the SAB 8051, is a single-stage inverter (Figure 44), intended for use as crystal-controlled, positive reactance oscillator (Figure 45). In this application the crystal is operated in its fundamental response mode as an inductive reactance in parallel resonance with a capacitance external to the crystal. The crystal specifications and capacitance values (C1 and C2 in Figure 45) are not critical. 30 pF can be used in these positions at any frequency with good quality crystals. A ceramic resonator can be used in place of the crystal in cost-critical applications. When a ceramic resonator is used, C1 and C2 are normally selected to be of somewhat higher values, typically 47 pF. The manufacturer of the ceramic resonator should be consulted for recommendations on the values of these capacitors.

To drive the SAB 80515 with an external clock source, apply the external clock signal to XTAL2, and ground XTAL1, as shown in Figure 46. A pullup resistor is suggested because the logic levels at XTAL2 are not TTL.

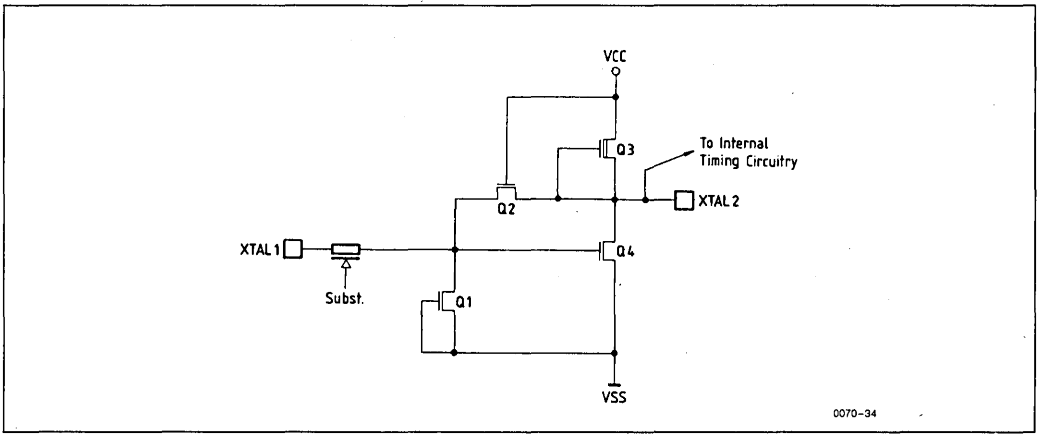


Figure 44. On-Chip Oscillator Circuitry

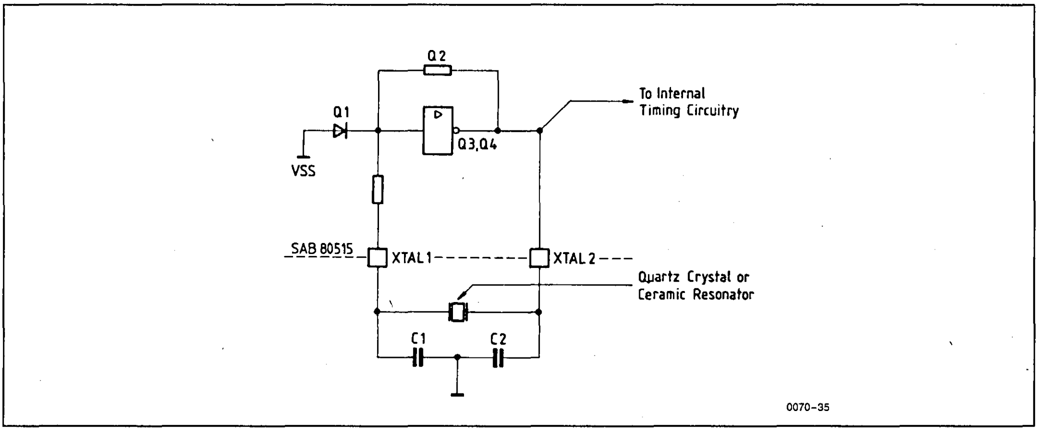


Figure 45. Using the On-Chip Oscillator

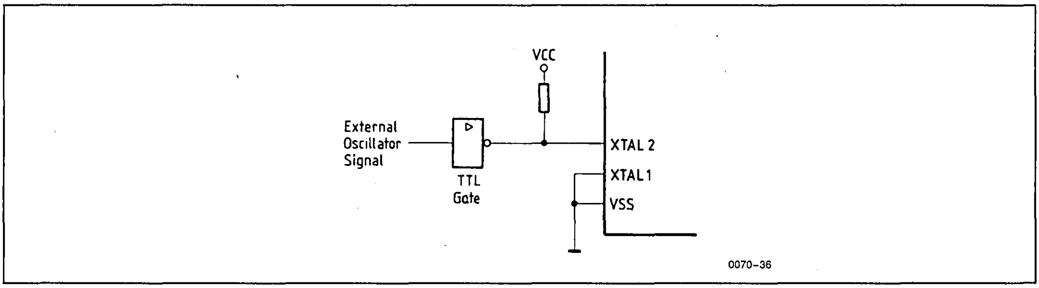


Figure 46. Driving with an External Clock Source

3.11 Register PCON

The special function register PCON is located at address 87H. In this register only bit 7, which is SMOD, is implemented. The other bit positions (PCON.0 to PCON.6) are reserved and should not be used. SMOD is used to double the baud rate for the serial port. If SMOD is set to one, the baud rate is doubled when the serial port is operating in either mode 1, 2, or 3 (see Section 3.4.2). The reset value of SMOD is 0. Note that PCON is not bit-addressable, therefore byte instructions must be used to alter SMOD.

Memory Organization, Addressing Modes and Boolean Processor

4.0 Memory Organization, Addressing Modes and Boolean Processor

4.1 Introduction

The SAB 80515 architecture provides on-chip memory as well as off-chip memory expansion capabilities. Several addressing mechanisms are incorporated to allow for an optimal instruction set.

4.2 Memory Organization

The SAB 80515 has four basic memory address spaces:

— 64 kbyte program memory

— 64 kbyte external data memory

— 256 byte internal data memory

— 41 special function registers

4.2.1 Program Memory Address Space

The 64 kbyte program memory space consists of an internal and an external memory portion, illustrated in Figure 48. If the $\overline{EA}$ pin is held high, the SAB 80515 executes out of internal program memory unless the address exceeds 1FFFH. Locations 2000H through 0FFFFH are then fetched from external program memory. If the $\overline{EA}$ pin is held low, the SAB 80515 fetches all instructions from external program memory. In either case, the 16-bit program counter is the addressing mechanism.

Locations 03 through 6BH in program memory are used by interrupt service routines as discussed in section 3.6.

4.2.2 Data Memory Address Space

The data memory address space consists of an internal and an external memory space. External data memory is accessed when a MOVX instruction is executed.

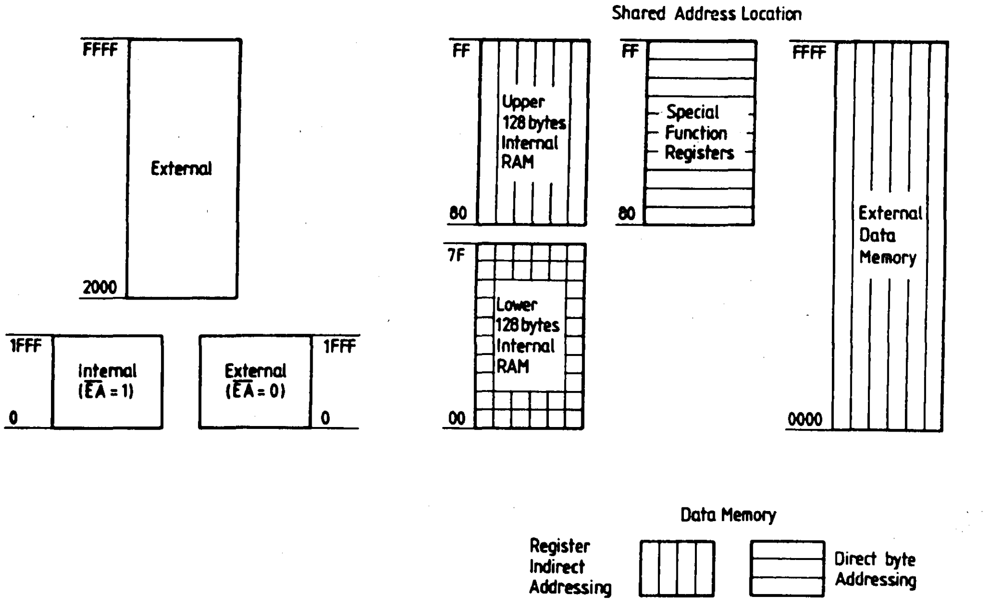


Figure 48. Program Memory and Data Memory Address Spaces

The internal data memory is divided into three physically separate and distinct blocks: the lower 128 bytes of RAM; the upper 128 bytes of RAM; and the 128 byte special function register (SFR) area. While the upper RAM area and the SFR area share the same address locations, they are accessed through different addressing modes. These modes are discussed in a later section.

Figure 49 shows a mapping of internal data memory. Four 8-register banks occupy locations 0 through 31 in the lower RAM area. Only one of these banks may be enabled at a time (through a two-bit field in the PSW). The next sixteen bytes, locations 32 through 47 contain 128 bit-addressable locations. 16 of the 41 special function registers are also bit-addressable.

Figure 48 shows the data memory address spaces, Figure 50 shows the RAM bit addresses, and Figure 51 the special function register bit locations.

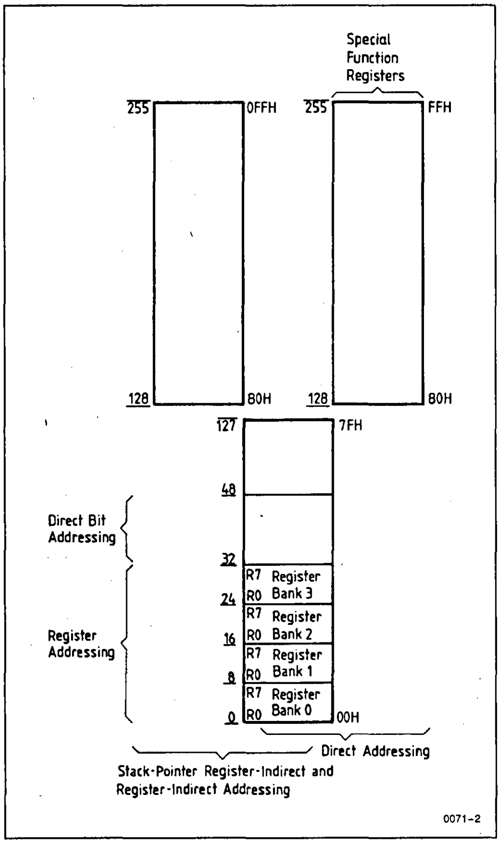
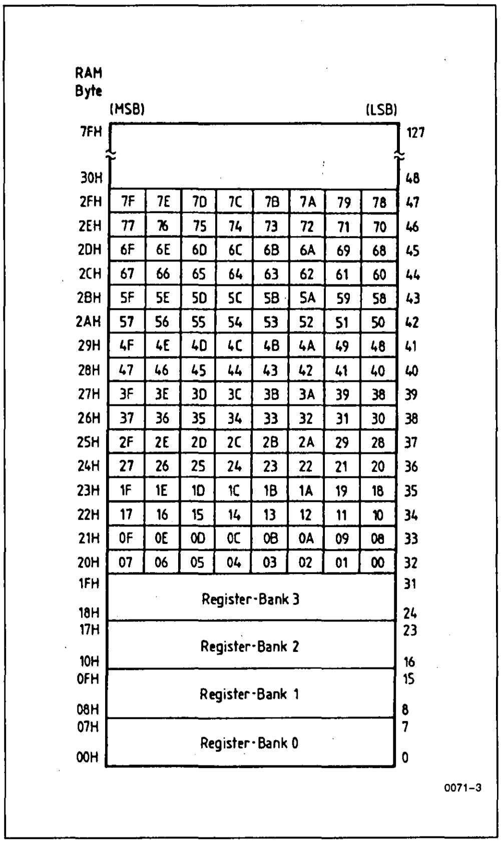


Figure 49. Internal Data Memory Address Space



RAM Byte	(MSB)							(LSB)	
7FH									127
30H									48
2FH	7F	7E	7D	7C	7B	7A	79	78	47
2EH	77	76	75	74	73	72	71	70	46
2DH	6F	6E	6D	6C	6B	6A	69	68	45
2CH	67	66	65	64	63	62	61	60	44
2BH	5F	5E	5D	5C	5B	5A	59	58	43
2AH	57	56	55	54	53	52	51	50	42
29H	4F	4E	4D	4C	4B	4A	49	48	41
28H	47	46	45	44	43	42	41	40	40
27H	3F	3E	3D	3C	3B	3A	39	38	39
26H	37	36	35	34	33	32	31	30	38
25H	2F	2E	2D	2C	2B	2A	29	28	37
24H	27	26	25	24	23	22	21	20	36
23H	1F	1E	1D	1C	1B	1A	19	18	35
22H	17	16	15	14	13	12	11	10	34
21H	0F	0E	0D	0C	0B	0A	09	08	33
20H	07	06	05	04	03	02	01	00	32
1FH 18H	Register-Bank 3								31 24
17H 10H	Register-Bank 2								23 16
0FH 08H	Register-Bank 1								15 8
07H 00H	Register-Bank 0								7 0

0071-3

Figure 50. Internal Data Memory Bit-Addressable Locations

F8H	FFH	FEH	FDH	FCH	FBH	FAH	F9H	F8H	P5
F0H	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	B
E8H	EFH	EEH	EDH	ECH	EBH	EAH	E9H	E8H	P4
E0H	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H	ACC
	BD	CLK	–	BSY	ADM	MX2	MX1	MX0	
D8H	DFH	DEH	DDH	DCH	DBH	DAH	D9H	D8H	ADCON
	CY	AC	F0	RS1	RS0	OV	F1	P	
D0H	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H	PSW
	T2PS	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0	
C8H	CFH	CEH	CDH	CCH	CBH	CAH	C9H	C8H	T2CON
	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	
C0H	C7H	C6H	C5H	C4H	C3H	C2H	C1H	C0H	IRCON
	EXEN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	
B8H	BFH	BEH	BDH	BCH	BBH	BAH	B9H	B8H	IEN1
B0H	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H	P3
	EAL	WDT	ET2	ES	ET1	EX1	ET0	EX0	
A8H	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H	IEN0
A0H	A7H	A6H	A5H	A4H	A3H	A2H	A1H	A0H	P2
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
98H	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	SCON
90H	97H	96H	95H	94H	93H	92H	91H	90H	P1
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	TCON
80H	87H	86H	85H	84H	83H	82H	81H	80H	P0

0071-4

Figure 51. Special Function Register Bit-Addressable Locations

4.3 Addressing Modes

The SAB 80515 uses five addressing modes:

—register
—direct
—register indirect
—immediate
—base-register plus index-register indirect

Table 12 summarizes which memory spaces may be accessed by each of the addressing modes.

4.3.1 Register Addressing

Register addressing accesses the eight working registers (R0–R7) of the selected register bank. The least significant bits of the instruction op code indicates which register is to be used. ACC, B, DPTR and CY, the Boolean processor accumulator, can also be addressed as registers.

Table 12

Addressing Modes	Associated Memory Spaces
Register Addressing	—R0 through R7 of Selected Register Bank —ACC, B, CY (Bit), DPTR
Direct Addressing	—Lower 128 Byte of Internal RAM —Special Function Registers
Register Indirect Addressing	—Internal RAM (@R1, @R0, SP) —External Data Memory (@R1, @R0, @DPTR)
Immediate Addressing	—Program Memory
Base-Register plus Index-Register Addressing	—Program Memory (@DPTR + A, @PC +A)

4.3.2 Direct Addressing

Direct addressing is the only method of accessing the special function registers. The lower 128 byte of internal RAM are also directly addressable.

4.3.3 Register-Indirect Addressing

Register-indirect addressing uses the contents of either R0 or R1 (in the selected register bank) as a pointer to locations in a 256-byte block: the 256 bytes of internal RAM or the lower 256 bytes of external data memory. Note that the special function registers are not accessible by this method. Access to the full 64 kbytes of external data memory address space is accomplished by using the 16-bit data pointer.

Execution of PUSH and POP instructions also uses register-indirect addressing. The stack may reside anywhere in internal RAM.

4.3.4 Immediate Addressing

Immediate addressing allows constants to be part of the instruction in program memory.

4.3.5 Base-Register plus Index-Register Addressing

Base-register plus index-register addressing allows a byte to be accessed from program memory via an indirect move from the location whose address is the sum of a base register (DPTR or PC) and index register, ACC. This mode facilitates look-up table accesses.

4.4 Boolean Processor

The Boolean processor is a bit processor integrated within the SAB 80515. It has its own instruction set, accumulator (the carry flag), and bit-addressable RAM and I/O.

The bit manipulation instructions allow:

—set bit
—clear bit
—complement bit
—jump if bit is set
—jump if bit is not set
—jump if bit is set and clear bit
—move bit from/to carry

Addressable bits, or their complements, may be logically ANDed or ORed with the contents of the carry flag. The result is returned to the carry register.

Microcontroller Application Notes
Article Reprints/Application Briefs

Prepared by
Ashutosh Ahluwalia

November 1988

Applications Using Operation of Timer 2 in the SAB 80515/80535—Generating Pulse Width Modulated Signals

SAB 80515/80535

Application Note

Introduction

This application introduces the user to the features of the Timer 2 on the SAB 80515/80535. It gives basic application hints for programming and using the functions of the Timer 2 structure. An example on generating pulse-width-modulated signals with minimum software is also given.

Timer 2 Structure

The Timer 2 of the SAB 80515 with its 16-bit compare/reload/capture register and three 16-bit compare/capture registers is capable of generating pulse width modulated output signals with very little software effort. This unit is also referred as Programmable Timer/Counter Register Array (PTRA).

The SAB 80515 offers two different compare modes, explained in more detail below, which are useful for the generation of these output signals.

In either compare mode, the values stored in the selected compare registers are compared continuously to the count value of Timer 2. For this purpose, each compare register has its own comparator circuit. Within one machine cycle, all comparisons are done simultaneously.

Each of the four registers has a fixed relation to a port 1 pin, illustrated in the following table:

Symbols

Register	Port Latch	Alternate Function Symbols
CRC Register	P1.0	$\overline{\text{INT3}}$/CC0
CC1 Register	P1.1	INT4/CC1
CC2 Register	P1.2	INT5/CC2
CC3 Register	P1.3	INT6/CC3

The alternate functions for which port pins P1.0 to P1.3 can be used are shown in the following table:

Port Pin		Alternate Function
P1.0	$\overline{\text{INT3}}$	External Interrupt 3 input; can be selected to be active on a negative or positive transition.
	CC0	Compare output from or capture input to CRC register.
P1.1	INT4	External Interrupt 4 input; active on a positive transition.
	CC1	Compare output from or capture input to CC1 register.
P1.2	INT5	External Interrupt 5 input; active on a positive transition.
	CC2	Compare output from or capture input to CC2 register.
P1.3	INT6	External Interrupt 6 input; active on a positive transition.
	CC3	Compare output from or capture input to CC3 register.

The selected modes of the compare/capture registers determine which of the possible alternate functions is actually used. For brevity, the description of the use of alternate functions is given for the CRC register and port pin P1.0 only.

If no compare or capture modes are enabled for the CRC register, pin P1.0 can be used for general purpose I/O or as an external interrupt input. In the latter case, the port latch must contain a one (1) to allow the external source to control the pin. It is, however, possible to cause an interrupt by toggling the port latch by software. Bit I3FR in SFR T2CON is used to select the active edge for $\overline{\text{INT3}}$ (refer to the SAB 80515 User's Manual for details). If the external capture mode is enabled for the CRC register (SFR CCEN = XXXXXX01B), a transition at P1.0 will not only cause the request flag IEX3 (SFR IRCON) to be set, but will also latch the current Timer 2 contents into the CRC register. If the pin is not controlled by an external source, software may toggle the port latch to do a capture of the Timer 2 content. This facilitates reading the timer "on-the-fly", since reading the timer directly requires a certain read procedure to be observed in order to obtain the correct value (to read the 16-bit register, only 8-bit MOV instructions are available). This feature is in addition to the special capture mode provided in the SAB 80515 for a software capture.

If either of the two possible compare modes is enabled for the CRC register, the port pin P1.0 is used as an output. The functions of the two compare modes are described separately in the following sections.

In compare mode (T2CM = 0), the port latch is controlled only by the Timer 2 overflow and the match signal of the comparator related to the CRC register. The user has no access to the port latch as long as compare mode 0 is enabled (SFR CCEN = XXXXXX10B). The input line from pin P1.0 to the interrupt request flag IEX3 is disconnected and IEX3 is controlled by the output of the comparator. If Timer 2 overflows, the port latch is set to zero. If a match is detected between the value stored in the CRC register and the content of Timer 2, then the port latch is set to one. The setting of the request flag IEX3 depends on the selected active edge. If I3FR = 0, IEX3 is set with the positive transition of the comparator output signal. If I3FR = 0, IEX3 will be set when the comparator output goes inactive, i.e. shows a negative transition.

In compare mode 1 (T2CM = 1), the port latch P1.0 is separated into two latches: one is connected to the internal bus and can be read or written to under software control. This is referred to as latch A. The other latch, called latch B, is a transparent latch with the input connected to latch A. Its output connected to the port pin and its clock input controlled by the comparator output. Figure 1 illustrates these two port latches. The function of compare mode 1 is illustrated by the following example:

Assume that port P1.0 contains a one, register CRC contains a value <xxxx>, IEX3 is programmed to be positive transition active, and Timer 2 is running. When compare mode 1 is enabled (SFR CCEN = XXXXXX10B), the port latch is separated, and both latch A and latch B contain a one. The user may now write a zero into latch A. Latch B is unchanged, since the clock input is low. When a match is detected between the value <xxxx> stored in the CRC register and the contents of Timer 2, the comparator output signal goes active. This causes the value of latch A to be transferred to latch B and pin P1.0 is set to zero. Simultaneously, the interrupt request flag IEX3 is set, informing the user of the successful match.

User response depends on the application. Writing a one to latch A causes a positive transition at the next compare event within the following Timer 2 period. Changing both the port latch and the compare value in register CRC causes a new match either within the current Timer 2 period if the new value is higher than the actual Timer 2 content, or within the next Timer 2 period otherwise. If the user changes neither the port latch nor the CRC register, the port pin will remain in its state regardless of further compare events. However, each further compare event (one during each Timer 2 period, if compare mode 1 is enabled) will set the interrupt request flag IEX3. This enables the user to count the compare events until a certain number is reached. Servicing the port latch at that point will cause a new transition of the pin at the next match.

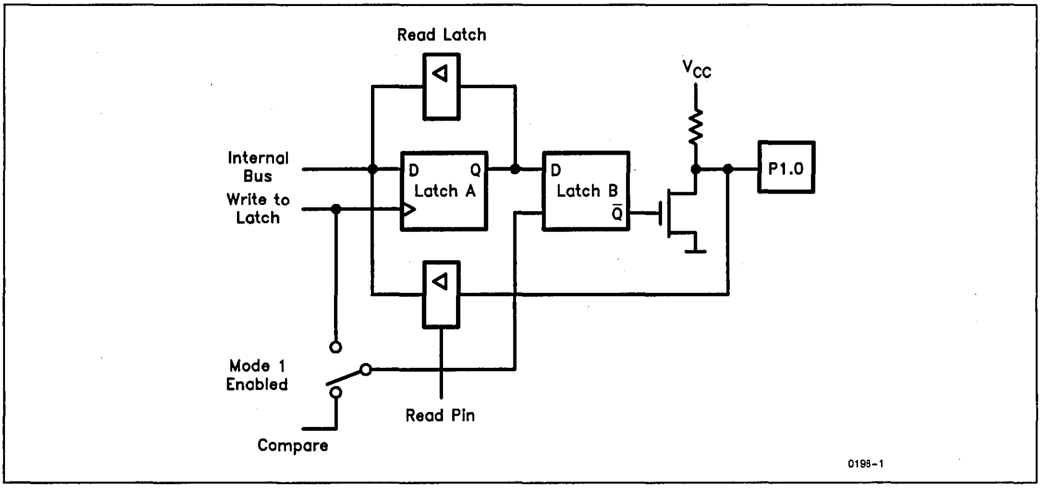


Figure 1. Functional Diagram of Port Latches P1.0 to P1.3 in Compare Mode 1

NOTE:

Since the transition detection circuit is installed directly in front of the interrupt request flag, IEX3 may be set when a compare mode is enabled or disabled. For an explanation of this, consider the following conditions:

- IEX3 is programmed to be set on a negative transition (I3FR = 0)
- Port latch P1.0 contains a one, so a logic high level is applied to the input of the detection circuit at IEX3
- The CRC register contains 0000H and Timer 2 contains a value > 0000H.
  If a compare mode is now enabled for the CRC register, the input line to the transition detector is switched from the port pin to the output of the comparator, which is zero (no match). The transition detector recognizes a negative transition at its input and causes IEX3 to be set. The same procedure applies to the other condition: I3FR = 1/P1.0 = 0/ comparator output = 1.

For registers CC1 to CC3 and their appropriate port pins, the compare and capture features function in a similar way, except that the setting of the interrupt flags and the capture are caused only by a positive transition at the respective port pins.

Using Compare Mode 0

The following section shows a concrete example for generating pulse width modulated output signals using the compare mode 0. The CRC register is used to do a 16-bit reload of Timer 2 in order to vary the period of the signals, while registers CC1 to CC3 are selected to operate in compare mode 0. The appendix contains a listing of the example program.

First an initialization routine is executed. It selects the input frequency for Timer 2, the compare and reload mode, loads the reload value into the CRC register, and starts the timer. Then the first loading of the compare registers CC1 to CC3 is done in the Timer 2 overflow interrupt service routine. The main program, which is not detailed in this example, computes the compare values and stores them in registers R2 to R7, from which the values are loaded into registers CC1 to CC3 during the interrupt procedure. If the compare values were left unchanged over several Timer 2 periods, the following output signals would result.

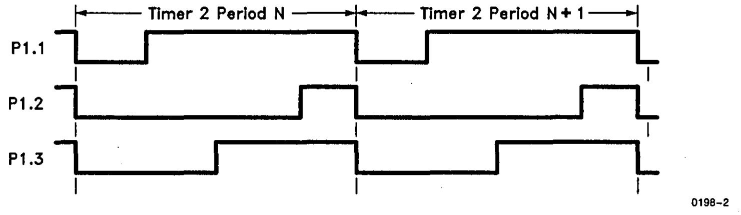


Figure 2. Pulse Width Modulation in Compare Mode 0

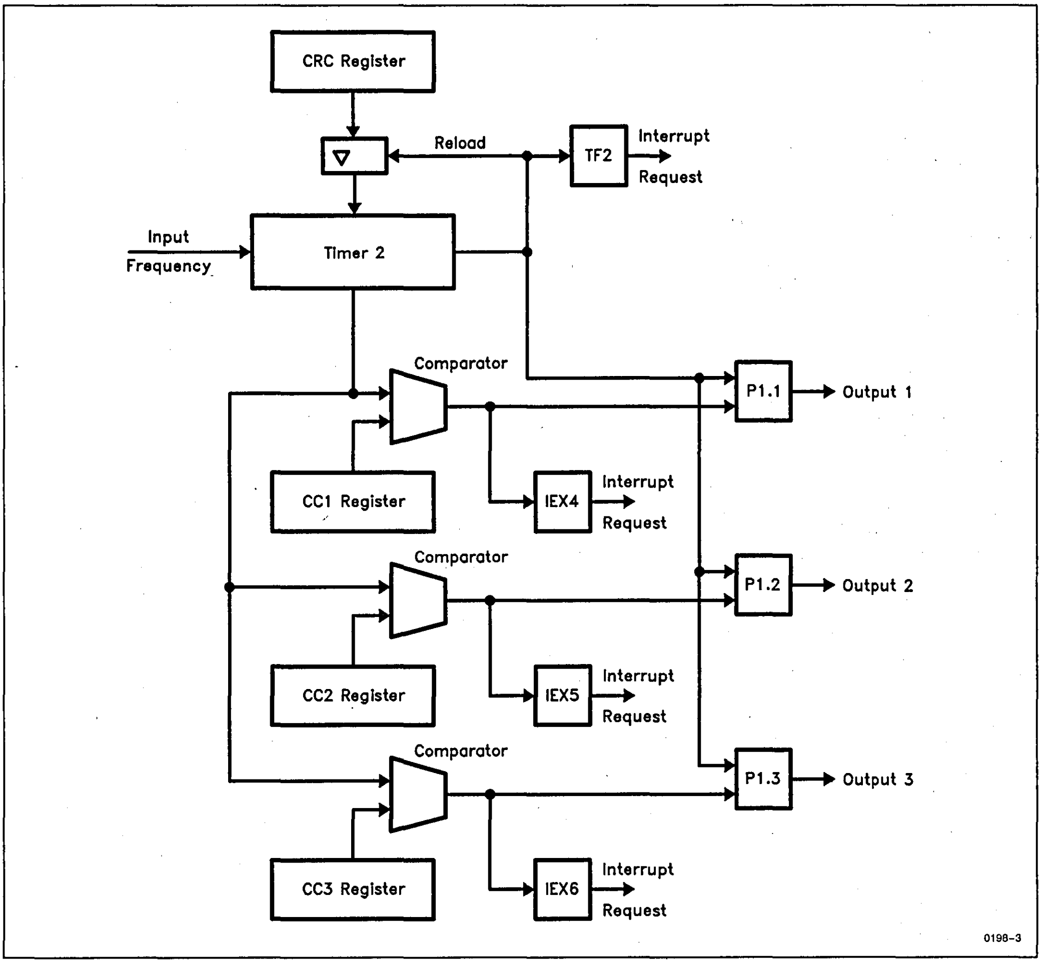


Figure 3. PTRA Configuration for the Generation of PWM Signals

Figure 3 illustrates the configuration of the PTRA structure after the initialization. The CRC register serves as a 16-bit reload register, while registers CC1 to CC3 are selected as 16-bit compare registers. The comparator output signal of each compare register is connected to the appropriate interrupt request flag (IEX4 to IEX6). Since the port pin P1.0 is not used in this application, it can be used as general purpose I/O or as an external interrupt input. It cannot, however, be used as a capture input to the CRC register, since selecting the reload mode for the CRC register disables the capture and compare modes for this register.

In this example, all three output signals have the same period as determined by the reload value stored in the CRC register. On an overlow of Timer 2, the port latches P1.1 to P1.3 are reset and the value of the CRC register is loaded into Timer 2. The timer continues incrementing from this start value up to the next overflow. The time period results from the formula

TP = (12/fosc) * (65536 − <reload value>)

where <reload value> is the content, of the CRC register. The frequency of the output signals is 1/TP. The assembler ASM51 provides an easy way to compute the reload value for a given time period. If the time period is given in μs (at 12 MHz in this example), the reload value can be programmed in the following way:

```
TIME_PERIOD EQU 1234 ;1234 µs time
                      period at 12 MHz

MOV         CRCL#LOW(-TIME_PERIOD)
MOV         CRCH#HIGH(-TIME_PERIOD)
```

Note the minus sign, which states that the value stored is 65536 − TIME__PERIOD.

The time during which an output signal is high is determined by the value stored in the associated compare register. The same mathematical relation applies here as for the reload value:

HT = (12/fosc) * (65536 − <compare value>)

where <compare value> is the contents of the respective compare register CC1 to CC3. Note that if a compare value is smaller than the reload value, the associated output pin will remain low, since there will never be a match.

If the compare values are not changed by the main program from one Timer 2 period to the next, the compare registers need not be reloaded in the Timer 2 interrupt service routine. A software flag called READY is used to indicate whether or not the compare registers have to be reloaded. If the main program has computed new compare values, it sets flag READY after storing these values in registers R2 to R7. The first instruction in the Timer 2 interrupt service routine checks the READY flag and returns to the main program if READY = 0. Otherwise the compare registers are loaded with the contents of R2 to R7. This procedure shortens the time needed for servicing the Timer 2 interrupt if the compare values are not changed.

The READY flag also has a second function. The instruction sequence which loads registers R2 to R7 can be interrupted by the Timer 2 interrupt. To avoid loading indeterminate values into the compare registers, the reload procedure in the Timer 2 interrupt service routine must not be executed until all new compare values are stored in R2 to R7. Since the READY flag will be reset in this case, the routine which loads the compare registers will be skipped in the interrupt service routine.

Before loading the compare registers with new values in the Timer 2 interrupt service routine, all compare modes are disabled (CCEN = 00000000B). This is done for a reason similar to that described above. When the 16-bit compare registers are reloaded with 8-bit MOV instructions, indeterminate compare values may occur in the registers. These values may cause unexpected match conditions. To avoid this, first compare is disabled. The registers are then loaded with new values, and finally the compare is again enabled. At the end the READY flag is reset.

Because vectoring to the Timer 2 interrupt service routine, disabling compare mode, and reloading the compare registers all take time, there is a limit to the smallest low pulse at the output pins. Vectoring to the interrupt routine in a single interrupt routine takes no more than 8 and no fewer than 3 cycles. Checking the READY flag, disabling compare, reloading the 3 16-bit registers, and enabling compare again will take 18 cycles. Therefore, the smallest low pulse should not be less than 0026D (26 cycles), otherwise a match is not recognized.

Figure 4 shows a detailed circuit diagram of an SAB 80535 with an external program memory represented by a 8 Kbyte SAB 2764A EPROM. This diagram also shows the basic power supply connections when using the SAB 80535.

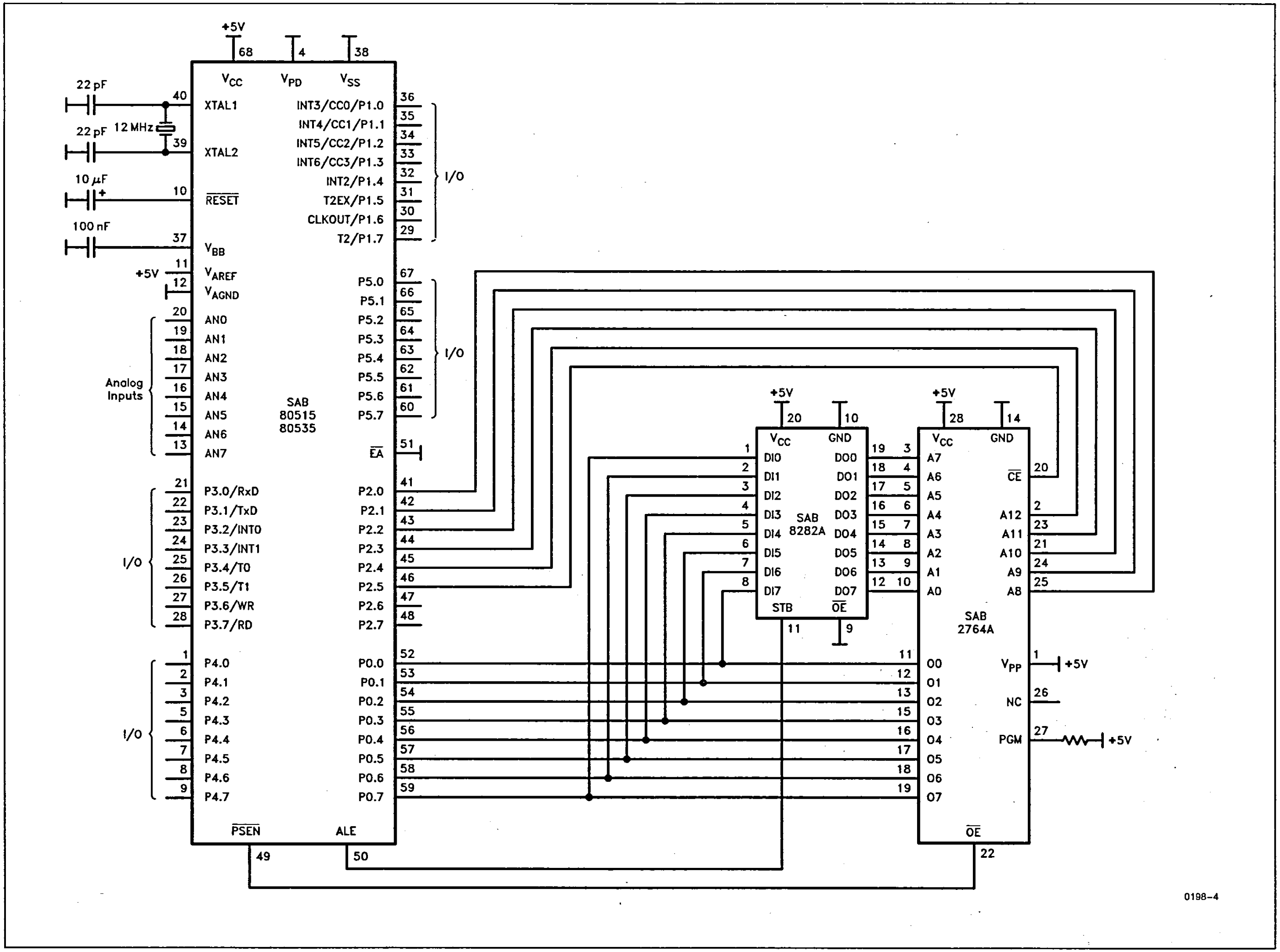


Figure 4. Circuit Diagram

```
MCS-51 MACRO ASSEMBLER    T2COMP

ISIS-II MCS-51 MACRO ASSMEBLER V2.1
OBJECT MODULE PLACED IN :F1:T2COMP.OBJ
ASSEMBLER INVOKED BY:  ASM51 :F1:T2COMP.SRC NOSYMBOLS DATE(04.03.86)

LOC  OBJ     LINE  SOURCE
               1   $NOMOD51
               2   $INCLUDE (REG515.PDF)
  = 1          3 +1$NOLIST
             160
  1234       161   TIME PERIOD      EQU         1234H       ; TIME PERIOD COMPARE SIGNALS
  0000       162   READY            BIT         0           ; READY FLAG DEFINITION
  0021       163   COMPARE_1_LOW    DATA        21H         ; TEMPORARY STORAGE LOCATIONS
  0022       164   COMPARE_1_HIGH   DATA        22H         ;  FOR COMPARE VALUES
  0026       165   COMPARE_3_HIGH   DATA        26H
             166
             167   ;*********************************************************
             168   ;*                                                       *
             169   ;*      INITIALIZATION ROUTINE                           *
             170   ;*                                                       *
             171   ;*********************************************************
             172
             173   INIT:
0000 75CACC  174       MOV          CRCL,#LOW(-TIME_PERIOD) ; LOAD RELOAD VALUE
0003 75CBED  175       MOV          CRCH,#HIGH(-TIME_PERIOD);
0006 75CB10  176       MOV          T2CON,#00010000B        ; RELOAD ON TIMER OVERFLOW
             177                                            ; NO PRESCALER; COMPARE MODE 0
0009 C200    178       CLR          READY                   ; CLEAR FLAG READY
000B D2C8    179       SETB         T2I0                    ; START TIMER 2, FOSC/12
             180
000D 020045  181       JMP          MAIN_PROG               ; JUMP TO MAIN PROGRAM
             182
             183
```

```
MCS-51 MACRO ASSEMBLER    T2COMP (Continued)

LOC  OBJ             LINE     SOURCE

                     184  ;*******************************************************
                     185  ;*                                                     *
                     186  ;* TIMER 2 OVERFLOW INTERRUPT SERVICE ROUTINE.         *
                     187  ;* LOADS NEW COMPARE VALUES INTO THE COMPARE REGISTERS.*
                     188  ;*                                                     *
                     189  ;*******************************************************
                     190
----                 191      CSE6  AT TIMER2
                     192
                     193  TF2_INT:
002B 300014          194      JNB   READY,INT_END      ; CHECK FLAG READY
002E 75C100          195      MOV   CCEN,$0            ; DISABLE COMPARE
0031 8AC2            196      MOV   CCL1,R2            ; LOAD NEW COMPARE VALUES
0033 8BC3            197      MOV   CCH1,R3            ;
0035 8CC3            198      MOV   CCL2,R4            ;
0037 8DC3            199      MOV   CCH2,R5            ;
0039 8EC6            200      MOV   CCL3,R6            ;
003B 8FC7            201      MOV   CCH3,R7            ;
003D 75C1A8          202      MOV   CCEN,#10101000B    ; ENABLE COMPARE 1 TO 3
0040 C200            203      CLR   READY              ; CLEAR FLAG READY
                     204 INT_END:
0042 C2C6            205      CLR   TF2                ; CLEAR TIMER 2 OVERFLOW FLAG
0044 32              206      RETI
                     207
                     208  ;********************************************************
                     209  ;*                                                      *
                     210  ;* MAIN PROGRAM.                                        *
                     211  ;* COMPUTES THE NEW COMPARE VALUES AND STORES THEM INTO *
                     212  ;* REGISTERS R2 TO T7 IN CURRENT REGISTER BANK.         *
                     213  ;*                                                      *
                     214  ;* SINCE IT DEPENDS ON THE APPLICATION, THE MAIN PROGRAM*
                     215  ;* IS NOT DETAILED HERE.                                *
                     216  ;*                                                      *
                     217  ;********************************************************
```

```
                                                                                                 1

MCS-51 MACRO ASSEMBLER   T2COMP (Continued)

LOC  OBJ          LINE     SOURCE

                  218
                  219   MAIN_PROG:
                  220   ;              ●
                  221   ;              ●
                  222   ;              ●
0045 C200         223                  CLR  READY               ; CLEAR FLAG READY
0047 AA21         224                  MOV  R2,COMPARE_1_LOW    ; STORE NEW COMPARE VALUES
0049 AB22         225                  MOV  R3,COMPARE_1_HIGH   ;
                  226   ;              ●
                  227   ;              ●
                  228   ;              ●
004B AF26         229                  MOV  R7,COMPARE_3_HIGH   ;
004D D200         230                  SETB READY               ; SET FLAG READY
                  231   ;              ●
                  232   ;              ●
                  233   ;              ●
                  234   ;              ●
                  235                  END
REGISTER BANK(S) USED: 0
ASSEMBLY COMPLETE, NO ERRORS FOUND
```

SIEMENS

November 1988

Operation of the A/D Converter in the SAB 80515/80535

SAB 80515/80535

Application Note

This application note discusses the Analog to Digital converter in the SAB 80515/80535 single chip microcontroller. It provides the SAB 80515/80535 user with detailed information on the features and characteristics of the converter. Included in this application note are design tips for using the converter, as well as an application example with program listing which illustrates how an 8-bit digital result obtained as a result of conversion of an analog input can be displayed on a terminal over the serial channel of the SAB 80515/80535.

The A/D Converter of the SAB 80515/80535

The SAB 80515/80535 is an 8-bit single chip microcontroller, which contains an on-chip A/D converter. It provides a simple interface between analog and digital circuitry. It permits the replacement of discrete A/D components with this on-board circuitry.

This application note illustrates the use of the SAB 80515/80535's A/D converter. The electrical characteristics and operating conditions are discussed in more detail than in the User's Manual.

The following topics are covered:

- Fundamentals and principles of A/D conversion with the SAB 80515/80535
- Transfer characteristics and error definition
- Electrical characteristics of the SAB 80515/80535's A/D converter
- Design considerations of the device
- Sample application of the device

Principles of A/D Conversion with the SAB 80515/80535

An Analog to Digital converter converts analog input signals (voltages) applied at the "analog inputs" into the corresponding digital value. The range of the analog inputs which result in digital values between 00H and FFH in the device's 8-bit converter is defined by the reference voltages (reference ground VAGND and reference voltage VAREF). These voltages are applied externally to the device.

Different principles may be used in A/D conversion. They differ mainly in accuracy, resolution, conversion time, costs, etc. The A/D converter of the SAB 80515/80535 uses the principle of successive-approximation. This technique is much faster than the more common dual slope conversion and allows tracking of signals at higher frequencies. Successive-approximation method uses binary fractions (½, ¼, ⅛, etc.). The unknown input value is compared first to the ½ of the reference value, determining the most significant bit of the result. Depending on this MSB, the unknown value is compared with ¼ or (¼ + ½) ref. value thus determining the next bit of the result. Following this procedure eight times produces an eight-bit result. The conversion time is independent of the input value.

To ensure meaningful data, the input value to the comparator cannot change during conversion. To ensure this, the SAB 80515/80535 samples the input and latches this value at the comparator input during the conversion. This avoids fluctuations in the analog input to the comparator. Suppression of noise in the analog input value as is normal with dual slope converters is not required, since there is no input signal integration.

For the above reasons, the reference voltage must be held at a constant level during the whole conversion time. Unlike the analog inputs, there is no sample-and-hold circuit at the reference input. To avoid unpredictable results, the user must provide a constant noise-free reference voltage to the device.

The successive approximation converter does not use an R-2R network because the variations in resistance values affect the accuracy of the A/D converter. Instead of the R-2R approach, a network of 256 capacitors is used. These capacitors are organized in binary weighted groups. Switching these groups generates the binary weighted reference values. Using this principle improves the converter accuracy and also the immunity against temperature and frequency changes.

Note that this technique affects the electrical characteristics at the analog input pins (see next section for further information).

A special feature of the SAB 80515/80535's A/D converter is the ability to program internal reference voltages. This feature allows reference ground (IVAGND) and reference voltage (IVAREF) to be defined as fractions of externally applied references. Device software makes it possible to select for each among 16 values in equal steps, one for IVAGND and one for IVAREF. It is also possible to program IVAGND = VAGND and IVAREF = VAREF.

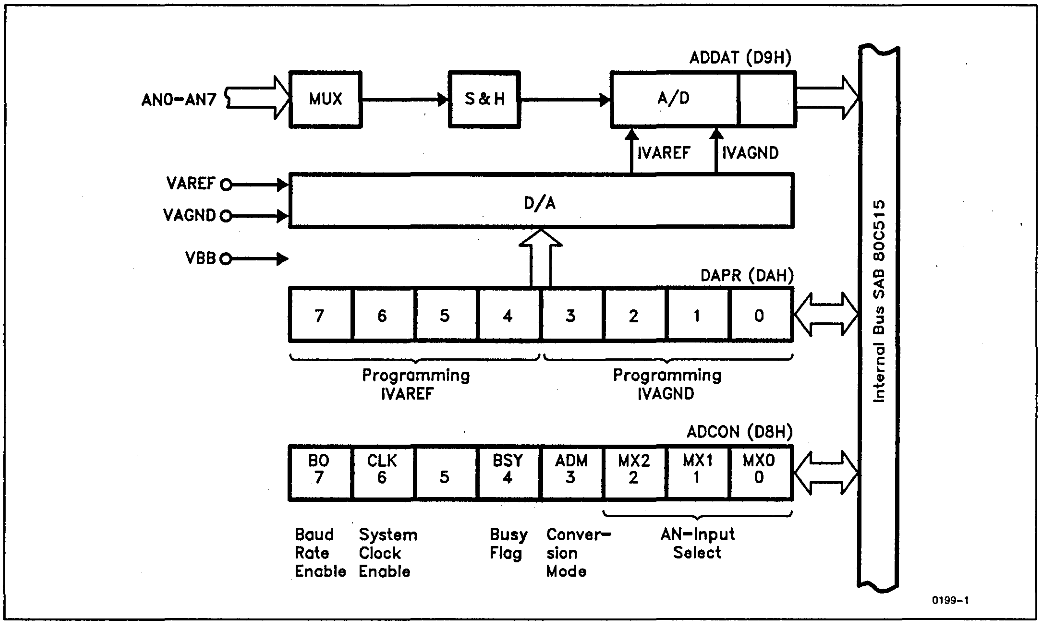


Figure 1. A/D Converter Block Diagram

Figure 1 shows the block diagram of the SAB 80515/80535's A/D converter. It shows the following main features:

— Eight analog inputs, selectable via a multiplexer

— Sample & hold function for analog inputs

— External references available to the converter after internal division

— Special Function Registers ADCON, ADDAT, DAPR, used for converter control, status and results

The Special Function Registers (SFR) are described in detail in the SAB 80515 User's Manual. Some points to consider when programming the SFRs include:

- The A/D conversion is initiated by writing the reference selection value into the SFR DAPR (addr.: DAH). Writing a "00H" starts the conversion using the externally applied references (VAGND and VAREF) directly.

  Writing into DAPR during a conversion causes an interrupt to the conversion followed by a restart.

- Altering the value of SFR ADCON (addr.: D8H) during a conversion causes unpredictable results.
- Once a conversion is complete, the SFR ADDAT (addr.: D9H) contains the result of the conversion. If the A/D converter isn't used, the SFR ADDAT is available for general purpose read/write storage.
- The conversion time is 15 machine cycles. Using a 12 MHz oscillator, gives a conversion time of 15 μs. This includes the sample time.

Transfer Characteristics and Error Definition

Figure 2 shows the ideal transfer characteristic for a 3-bit A/D converter.

Since possible analog input values are a continuum, they must be quantizised by partitioning the continuum into 2^n discrete digital values. The number n equals the number of bits in the converter. All analog values within a given range are represented by the same digital value, which corresponds to the nominal mid-range value. Each converter has inherent quantization uncertainty of ±½ LSB (Least Significant Bit). The ideal transfer characteristic shown in Figure 2 gives the first digital transition (from "00H" to "01H") at the analog value of ½ LSB.

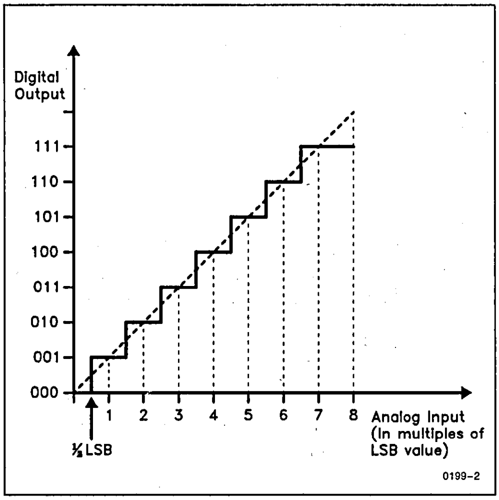


Figure 2. Ideal Transfer Characteristic

Based on the ideal transfer characteristic, four different error types are defined:

- Offset error
- Integral non-linearity
- Differential non-linearity
- Gain error

Each of these is described in detail below.

The offset error (Figure 3) is the mean adjustment in input voltage required to bring the digital output to the first digital transition (from "00H" to "01H") of the converter. The deviation from the ideal value of this transition (at ½ LSB) is the offset of the converter.

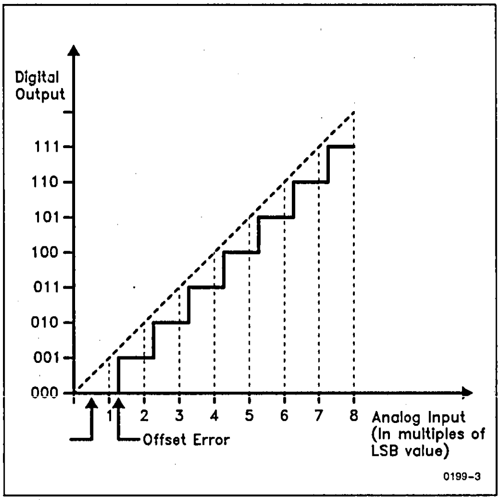


Figure 3. Offset Error

This error may vary over the full temperature range. An adjustment is therefore exact only for a fixed temperature.

The integral nonlinearity is also known as the linearity error. This is defined as the maximum deviation of the actual transfer function from the ideal straight line at any point along the function. It can be expressed as a percentage of full scale or, as shown in Figure 4, in multiples of the LSB value. The value of integral nonlinearity assumes that other errors, such as offset and gain, have been adjusted to zero. Linearity error cannot be adjusted and is an inherent characteristic of the converter.

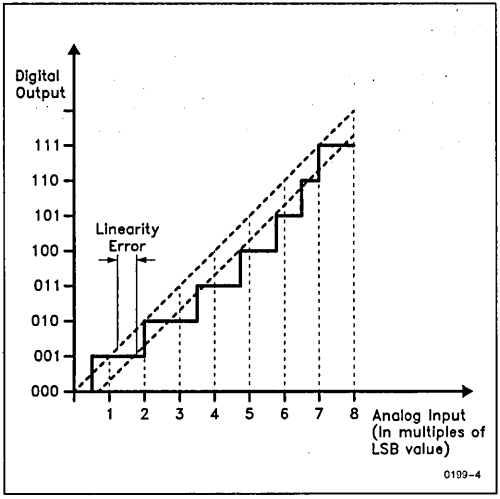


Figure 4. Integral Nonlinearity

The differential nonlinearity (Figure 5) is the maximum deviation of any quantum from its ideal analog input value between any adjacent pairs of digital numbers, over the full range of the digital output. If each transition is exactly 1 LSB, the differential nonlinearity is zero. If the transitions are 1 LSB ±1 LSB, then there is the possiblity of missing code, i.e. digital value misses, e.g. the output might jump from 011 to 101, missing out 100.

Conversely, differential nonlinearity of less than 1 LSB, automatically excludes missing codes.

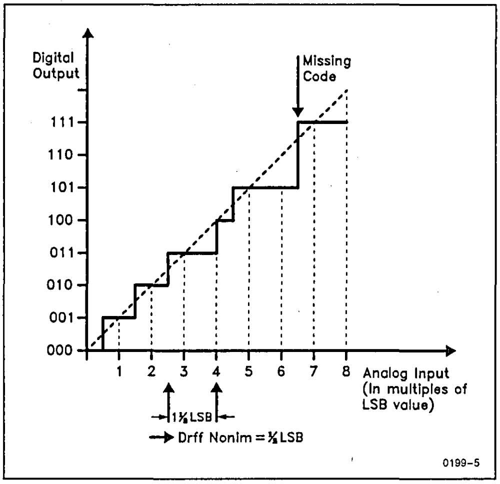


Figure 5. Differential Nonlinearity

The gain error (Figure 6) is also known as the scale factor error. It is the difference in slope between the real and the ideal transfer characteristic. It may be expressed in LSB's or as a percentage of analog magnitude.

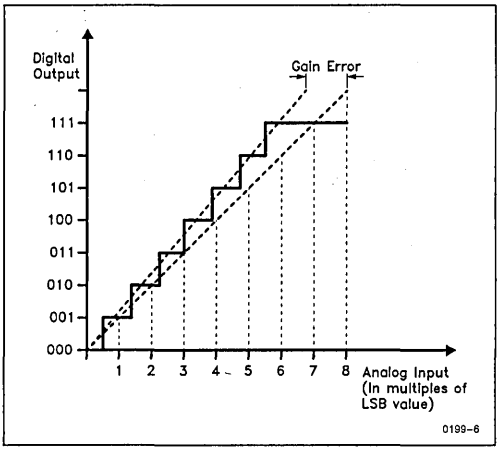


Figure 6. Gain Error

This error can be adjusted by changing the reference voltage or adjusting the input voltage division. The temperature drift of the gain error is generally smaller than that of the offset error.

All these errors are specified for the SAB 80515/80535 in the A/D characteristics section of the data sheet. The LSB unit refers to an 8-bit resolution. Therefore, 1 LSB is (VAREF − VAGND)/256, giving 20 mV for a reference voltage of 5.12V.

The specified errors are valid over the total operating temperature range specified for the device (0°C to +70°C for standard parts, −40°C to +85°C for extended temperature range parts).

Using the internal programming ability for the reference voltages IVAGND and IVAREF does not affect the errors and the accuracy of the A/D converter itself. The absolute values of these errors will be the same over the full reference range, as well as in a smaller internally programmed reference range. The resolution is increased by using narrower references. As an example, if the internally programmed references are IVAGND = 2.5V and IVAREF = 3.75V, the resolution is ≈5 mV. This allows a more exact measurement of differences between several successive analog voltages; the differences can be determined in steps of 5 mV, rather than steps of 20 mV (full reference range). This feature is useful for the measurement of analog differences in closed control loops. Once an initial value is established using the full range, a second conversion with changed internal reference voltages gains higher resolution.

Electrical Characteristics of the SAB 80515/80535's A/D Converter

The electrical operating conditions and maximum ratings on the pins of the SAB 80515/80535's A/D converter are given in the datasheet.

The Analog Input Pins AN0–AN7

MAXIMUM RATINGS

For the analog inputs the same maximum ratings are valid as for all other pins. These maximum ratings are specified in the SAB 80515/80535's datasheet. Thus, the maximum voltage at the pins is restricted to −0.5V up to +7V with respect to ground (V_{SS}). The device may never be exposed to voltages above these values, otherwise the chip might be damaged.

Normal Operating Characteristics

As mentioned above, the A/D converter of the SAB 80515/80535 uses a capacitor network instead of a R-2R network. This affects the input impedance of the analog inputs.

The input impedance of the analog inputs is mainly capacitive with a negligible resistive component. This capacitance is relevant at pins AN0–AN7 only during the load time, which is part of the conversion time. At other times, the inputs have very high impedance typical leakage currents of nA's, as shown in Figure 7.

The load time is the period which is available to charge the inputs before they are sampled during the remainder of the samle time when the comparator is adjusted. The rest of the conversion time is taken up by the actual conversion process.

The input impedance requires that the analog source from which the analog value is generated must be capable of fully charging all the analog input pins AN0–7 to the value to be measured within the load time. In the worst case the capacitance must be charged to the full input voltage from the completely discharged state.

The input capacitance CI of the converter and the internal resistance of the analog source form a RC lowpass filter, which has a charging function shown in Figure 8.

Where:

t: Time

V(t): Voltage after time t

Ri: Internal resistance of the analog source

CI: Input capacitance of the A/D converter. This formula determines the change over a given time period.

t	V/Vmax [%]
1 RC	63%
2 RC	87%
5 RC	99.3%
8 RC	99.97%
10 RC	99.995%

The maximum value of Ri can be calculated for a charging time of at least 10 Ri × CI (error less than 0.005%) as:

$$T_L \geq 10 \times (Ri \times CI)$$

$$\boxed{RI \leq \frac{T_L}{10 \times CI}}$$

Assuming the worst-case conditions of:

Load time (FOSC = 12 MHz): 2 μs
Max. Input Capacity: 50 pF (typ. 25 pF)

We Get:

$$Ri \leq \frac{2\ \mu s}{10 \times 50\ pF} = 4\ k\Omega$$

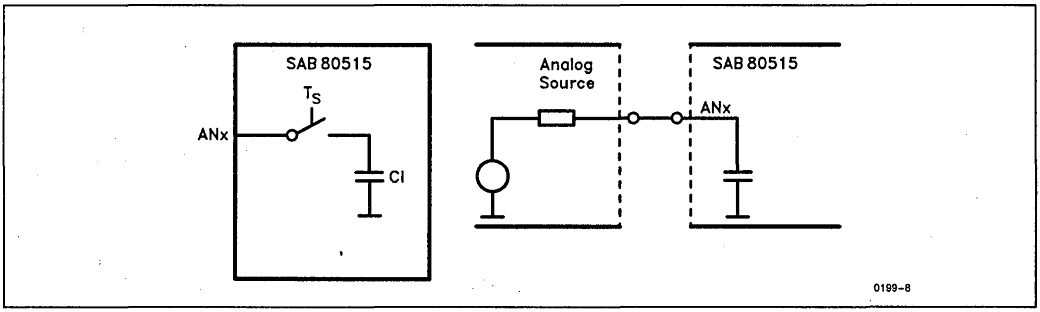
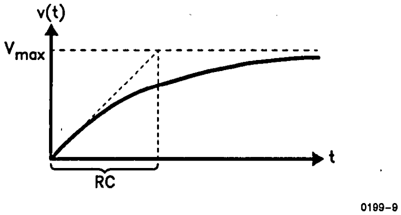


Figure 7. Analog Interface, an RC Network



$V(t) = Vmax(1 - e^{-t/Ri} - CT)$

Figure 8. Ri, CI Change Time

Ri should therefore be less than 4 kΩ (see also datasheet SAB 80515/80535).

These conditions are reached only under worst-case conditions.

The Reference Voltage Pins (VAGND and VAREF)

Absolute Maximum Ratings

As with other pins, the maximum voltage range at these pins is −0.5V to +7V, as given in the datasheet.

Normal Operating Characteristics

The normal operating conditions for VAGND and VAREF are also specified in the SAB 80515/80535 datasheet, under A/D characteristics. The operating conditions under which conversion is valid are:

$VAGND = V_{SS} \pm 0.2V$

$VAREF = V_{CC} \pm 5\%$

In other words, externally applied reference voltages must not deviate from the digital power supply voltage for more than these values. Although adjustment of the external reference is not possible internal programming of the references can be done under software control.

Exceeding the values specified will not damage the part if the maximum ratings are not exceeded. However, the specified accuracy is no longer guaranteed.

The allowed operating voltages of the analog inputs AN0–AN7 are different from the maximum ratings. They are defined by the voltages at VAGND and VAREF:

$VAGND - 0.2V \leq VAINPUT \leq VAREF + 0.2V$

An exact conversion requires that the reference voltages be held constant during the whole conversion time. The normal digital power supply for V_{CC} is therefore appropriate for VAREF because of noise and glitches on this voltage. It is recommended that the digital power supply and the analog reference supply be separate. The VAREF current is 5 mA max. and the differential internal resistance of the reference supply must be less than 1 kΩ. This is a result of similar charging processes to those at the analog inputs and the reference power supply must take this into account.

The V_{BB} Pin

This pin is connected to the substrate of the chip. A back-bias generator generates a negative voltage (with respect to ground) in the substrate. This generator has noise too. Noise from this could affect the A/D converter operation. It must therefore be eliminated by a capacitor between V_{BB} and V_{SS} (ground). The capacitor should be between 47 nF and 100 nF (not 500 pF–1000 pF, as described in an older release of the datasheet).

Design Considerations

Reference Voltage Supply

A design with the SAB 80515/80535's A/D converter must follow rules similar to other A/D converter designs. Generation of the reference voltage supply is critical. It is recommended that analog and digital grounds are connected together. Care must be taken to avoid the digital current at V_{SS} injecting noise into the analog ground potential. The ground lines to V_{SS} and VAGND should be isolated from each other as much as possible. Figure 9 illustrates the recommended supply connections.

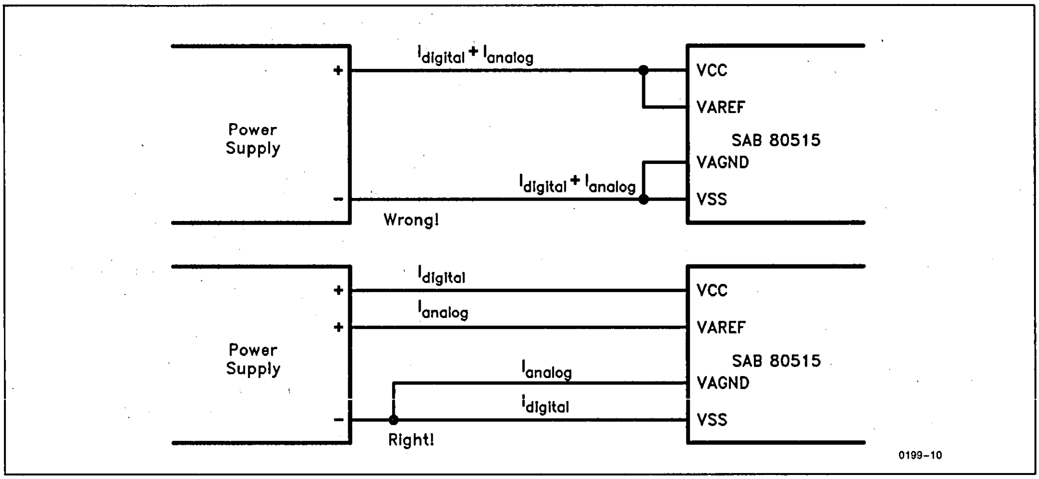


Figure 9. Power Supply Connections

For best reference voltage stability on VAREF, a separate power supply should be provided. If, for cost reasons, only one supply voltage is used for both digital and analogy supply, the voltage at VAREF should be stabilized with a lowpass filter. Note that the differential internal resistance of the reference supply must not exceed 1 kΩ, as described above.

The supply should be grounded across a storage capacitor (tantalum) and a smaller HF-capacitor (ceramic) as near the device as possible.

Analog Input Supply

The use of analog power sources with higher internal resistances than 4 kΩ is possible under certain conditions. The internal resistances requirement is a result of the charging current necessary to overcome the analog input capacitance. This current is only needed during the sample time. If the analog value changes only slowly with time, a large external capacitor at the analog input is able to supply the charging current during the sample time. This is shown in Figure 10.

The external capacitor should be at least 1000 times the value of the internal capacitance (50 nF = 1000 × 50 pF). The error induced by the external capacitor is therefore kept less than 0.1%.

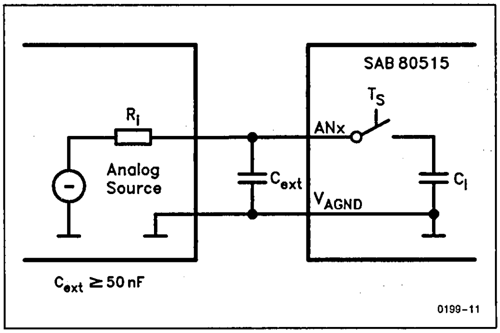


Figure 10. Analog Input Supply

Using an external capacitor, the analog source must replace only that charge lost by the external capacitor. The time available for this is the time between two consecutive conversions on this channel. This depends on user software, but is at least three times the sample time if continuous conversion on the same channel is selected.

The equation for Ri can be applied here:

$$Ri \leq \frac{T}{10 \times CI}$$

The difference is the time "T" between two consecutive conversions on the channel. This time is longer than the sample time, which permits higher internal resistance in the analog power source.

An analog input overload protection can be provided by two diodes, as shown in Figure 11. A Schottky diode satisfies the rating for the connection to VAGND because of its lower forward voltage characteristic.

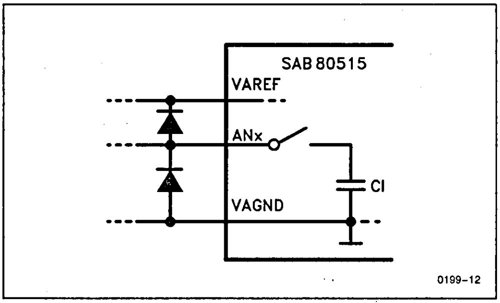


Figure 11. Protection Diodes for Analog Inputs

Application Example

The following application shows how to use the A/D converter for 8 different analog signals applied at the pins AN0–AN7. The analog values are generated by eight potentiometers. A terminal can be used as an output device through the SAB 80515/80535's serial channel.

The "single conversion" operating mode is used. Operation begins by writing a "00H" to DAPR, which deselects the internal reference voltage programming. The program polls the BSY flag for the termination of the conversion. The result is converted into BCD format and sent to the terminal. This procedure is followed for each of the eight analog channels.

Programming details are given in the program listings.

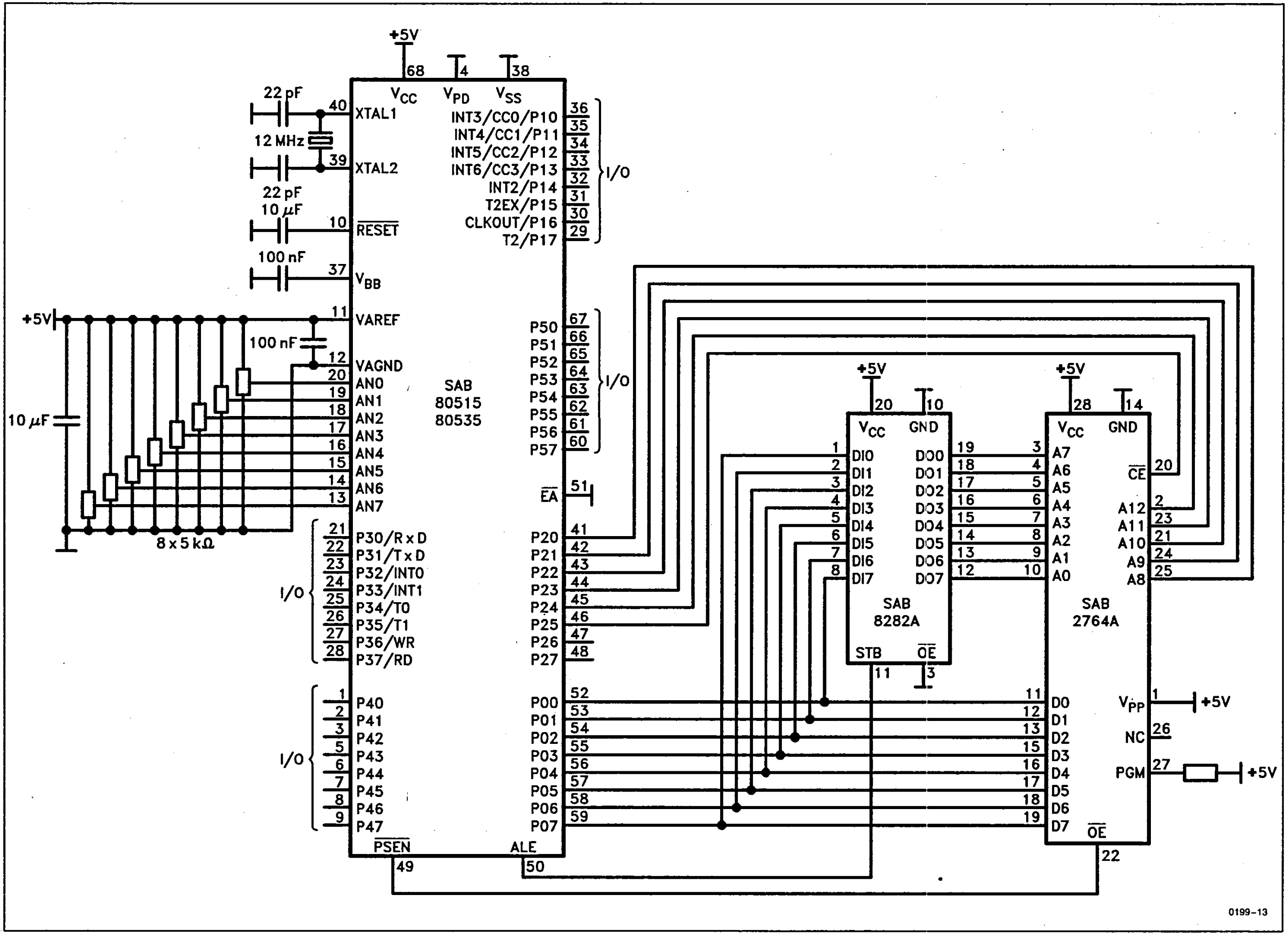
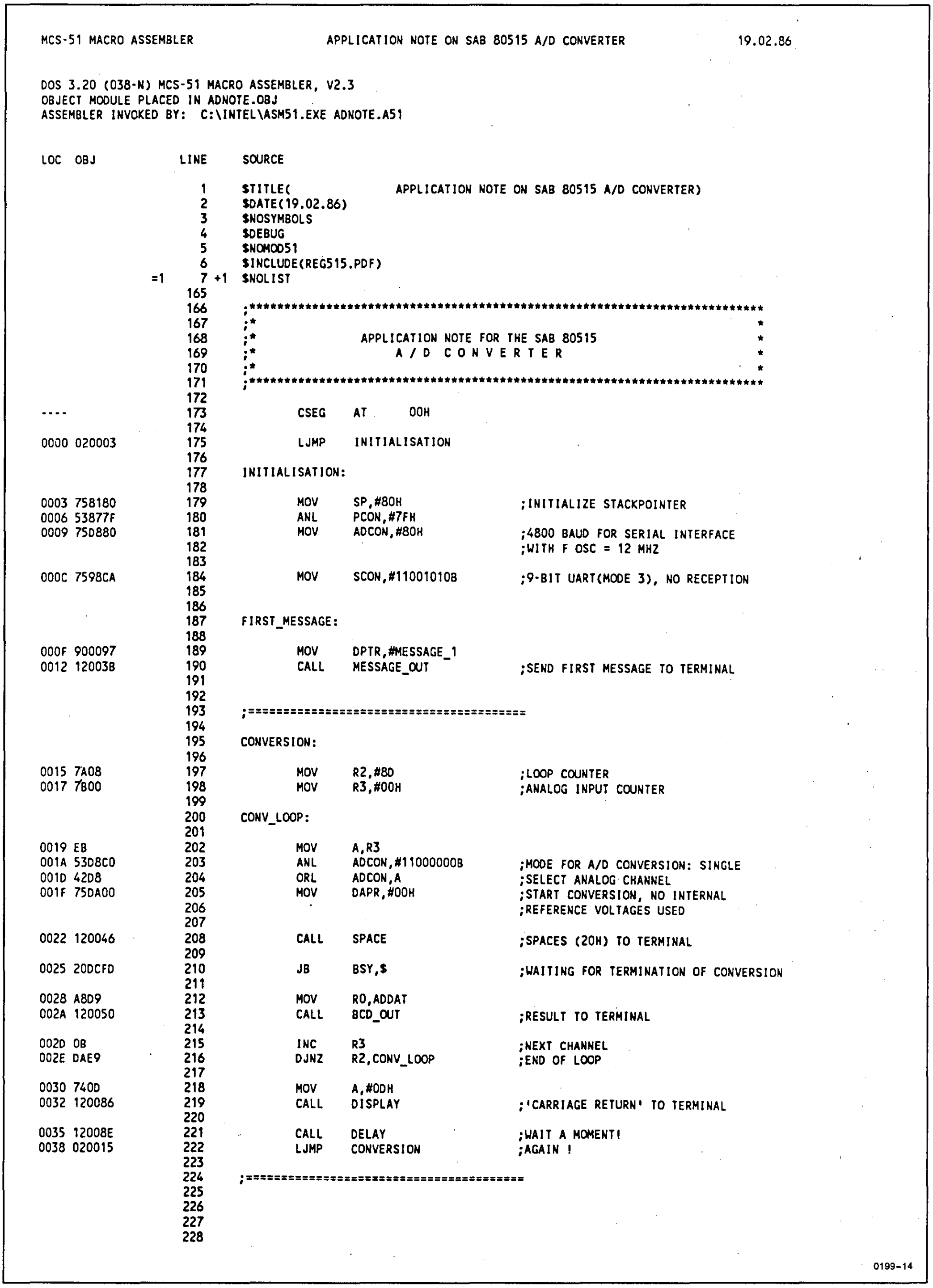
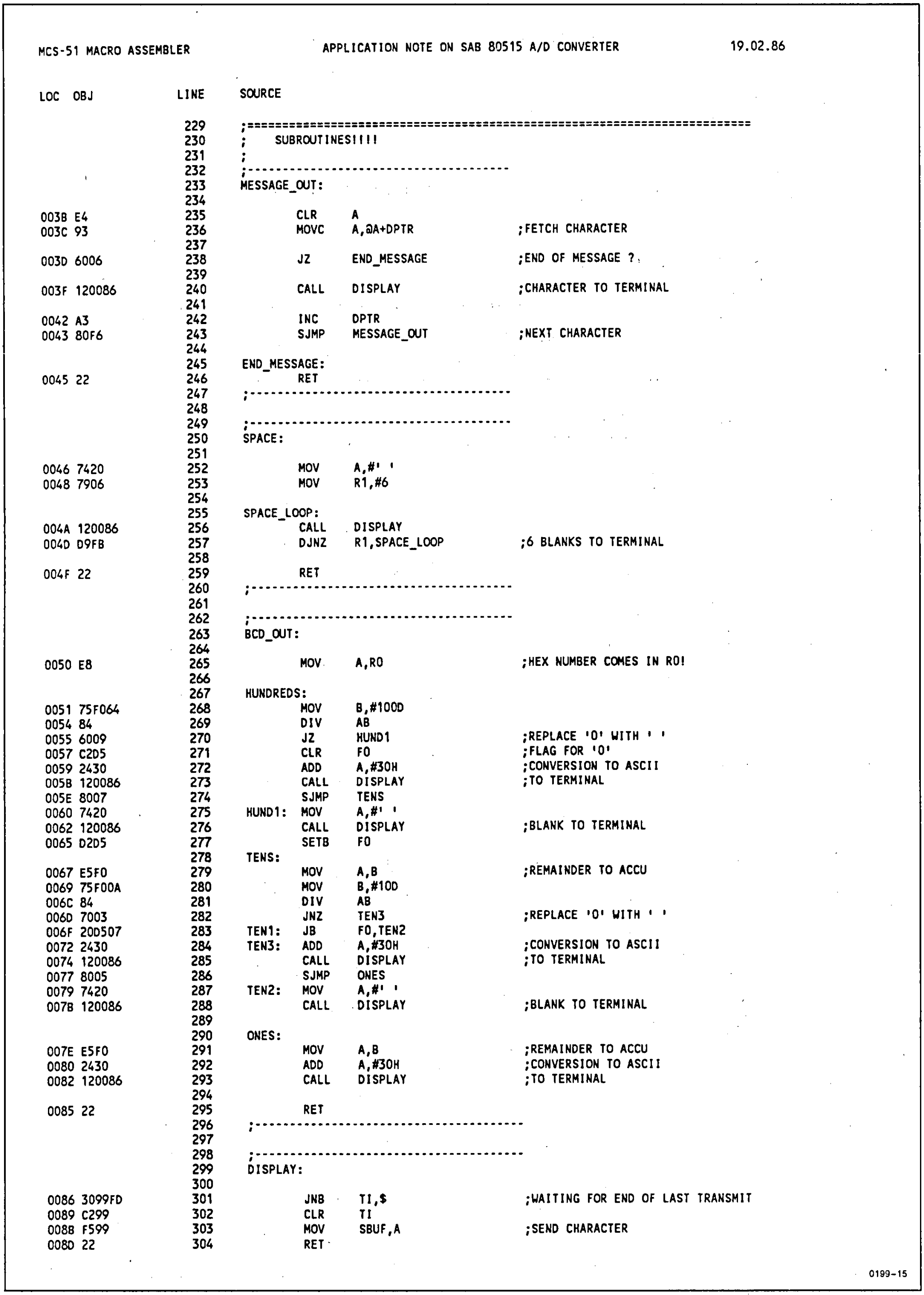


Figure 12. Circuit Diagram



```
MCS-51 MACRO ASSEMBLER              APPLICATION NOTE ON SAB 80515 A/D CONVERTER          19.02.86

DOS 3.20 (038-N) MCS-51 MACRO ASSEMBLER, V2.3
OBJECT MODULE PLACED IN ADNOTE.OBJ
ASSEMBLER INVOKED BY:  C:\INTEL\ASM51.EXE ADNOTE.A51


LOC  OBJ            LINE     SOURCE

                       1     $TITLE(           APPLICATION NOTE ON SAB 80515 A/D CONVERTER)
                       2     $DATE(19.02.86)
                       3     $NOSYMBOLS
                       4     $DEBUG
                       5     $NOMOD51
                       6     $INCLUDE(REG515.PDF)
                =1     7 +1  $NOLIST
                     165
                     166     ;*************************************************************************
                     167     ;*                                                                       *
                     168     ;*                  APPLICATION NOTE FOR THE SAB 80515                   *
                     169     ;*                      A / D  C O N V E R T E R                         *
                     170     ;*                                                                       *
                     171     ;*************************************************************************
                     172
----                 173             CSEG    AT      00H
                     174
0000 020003          175             LJMP    INITIALISATION
                     176
                     177     INITIALISATION:
                     178
0003 758180          179             MOV     SP,#80H                 ;INITIALIZE STACKPOINTER
0006 53877F          180             ANL     PCON,#7FH
0009 75D880          181             MOV     ADCON,#80H              ;4800 BAUD FOR SERIAL INTERFACE
                     182                                             ;WITH F OSC = 12 MHZ
                     183
000C 7598CA          184             MOV     SCON,#11001010B         ;9-BIT UART(MODE 3), NO RECEPTION
                     185
                     186
                     187     FIRST_MESSAGE:
                     188
000F 900097          189             MOV     DPTR,#MESSAGE_1
0012 12003B          190             CALL    MESSAGE_OUT             ;SEND FIRST MESSAGE TO TERMINAL
                     191
                     192
                     193     ;====================================
                     194
                     195     CONVERSION:
                     196
0015 7A08            197             MOV     R2,#8D                  ;LOOP COUNTER
0017 7B00            198             MOV     R3,#00H                 ;ANALOG INPUT COUNTER
                     199
                     200     CONV_LOOP:
                     201
0019 EB              202             MOV     A,R3
001A 53D8C0          203             ANL     ADCON,#11000000B        ;MODE FOR A/D CONVERSION: SINGLE
001D 42D8            204             ORL     ADCON,A                 ;SELECT ANALOG CHANNEL
001F 75DA00          205             MOV     DAPR,#00H               ;START CONVERSION, NO INTERNAL
                     206                                             ;REFERENCE VOLTAGES USED
                     207
0022 120046          208             CALL    SPACE                   ;SPACES (20H) TO TERMINAL
                     209
0025 20DCFD          210             JB      BSY,$                   ;WAITING FOR TERMINATION OF CONVERSION
                     211
0028 A8D9            212             MOV     R0,ADDAT
002A 120050          213             CALL    BCD_OUT                 ;RESULT TO TERMINAL
                     214
002D 0B              215             INC     R3                      ;NEXT CHANNEL
002E DAE9            216             DJNZ    R2,CONV_LOOP            ;END OF LOOP
                     217
0030 740D            218             MOV     A,#0DH
0032 120086          219             CALL    DISPLAY                 ;'CARRIAGE RETURN' TO TERMINAL
                     220
0035 12008E          221             CALL    DELAY                   ;WAIT A MOMENT!
0038 020015          222             LJMP    CONVERSION              ;AGAIN !
                     223
                     224     ;====================================
                     225
                     226
                     227
                     228
```

0199-14

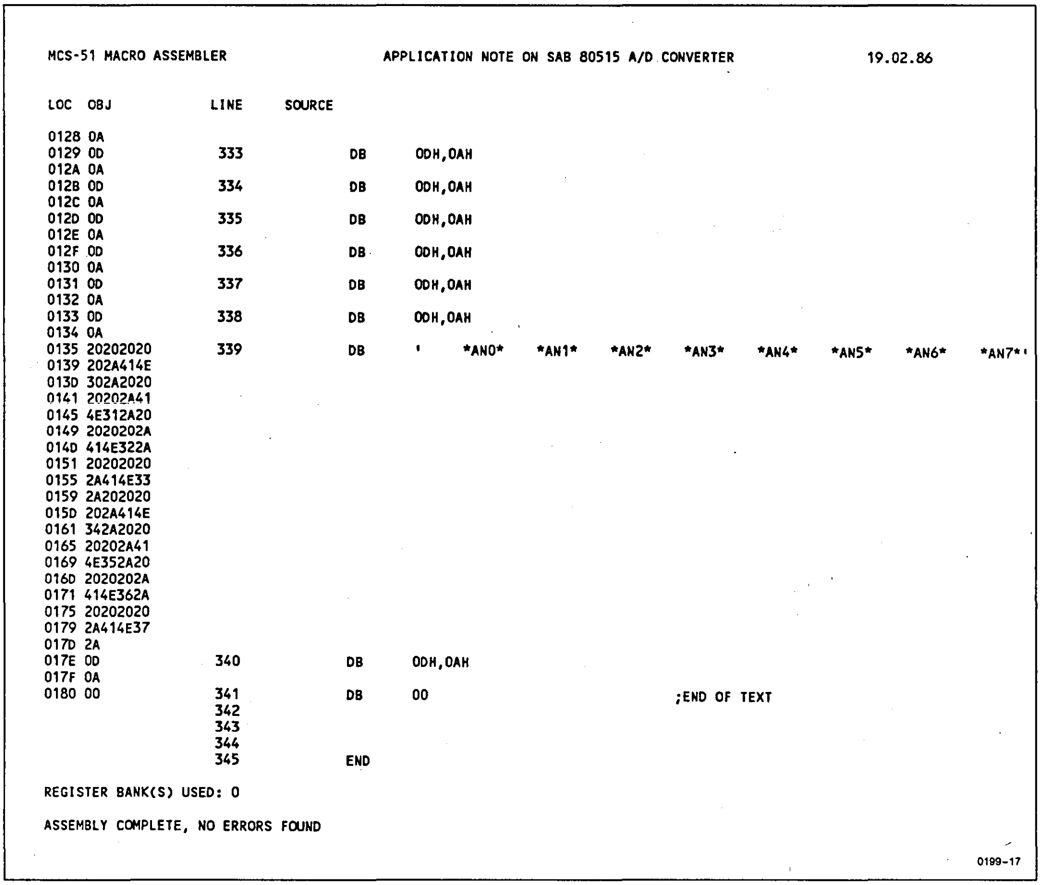


```
MCS-51 MACRO ASSEMBLER              APPLICATION NOTE ON SAB 80515 A/D CONVERTER            19.02.86

LOC  OBJ             LINE     SOURCE

                      229     ;=====================================================================
                      230     ;    SUBROUTINES!!!!
                      231     ;
                      232     ;-------------------------------------
                      233     MESSAGE_OUT:
                      234
003B E4               235             CLR     A
003C 93               236             MOVC    A,@A+DPTR               ;FETCH CHARACTER
                      237
003D 6006             238             JZ      END_MESSAGE             ;END OF MESSAGE ?
                      239
003F 120086           240             CALL    DISPLAY                 ;CHARACTER TO TERMINAL
                      241
0042 A3               242             INC     DPTR
0043 80F6             243             SJMP    MESSAGE_OUT             ;NEXT CHARACTER
                      244
                      245     END_MESSAGE:
0045 22               246             RET
                      247     ;-------------------------------------
                      248
                      249     ;-------------------------------------
                      250     SPACE:
                      251
0046 7420             252             MOV     A,#' '
0048 7906             253             MOV     R1,#6
                      254
                      255     SPACE_LOOP:
004A 120086           256             CALL    DISPLAY
004D D9FB             257             DJNZ    R1,SPACE_LOOP           ;6 BLANKS TO TERMINAL
                      258
004F 22               259             RET
                      260     ;-------------------------------------
                      261
                      262     ;-------------------------------------
                      263     BCD_OUT:
                      264
0050 E8               265             MOV     A,R0                    ;HEX NUMBER COMES IN R0!
                      266
                      267     HUNDREDS:
0051 75F064           268             MOV     B,#100D
0054 84               269             DIV     AB
0055 6009             270             JZ      HUND1                   ;REPLACE '0' WITH ' '
0057 C2D5             271             CLR     F0                      ;FLAG FOR '0'
0059 2430             272             ADD     A,#30H                  ;CONVERSION TO ASCII
005B 120086           273             CALL    DISPLAY                 ;TO TERMINAL
005E 8007             274             SJMP    TENS
0060 7420             275     HUND1:  MOV     A,#' '
0062 120086           276             CALL    DISPLAY                 ;BLANK TO TERMINAL
0065 D2D5             277             SETB    F0
                      278     TENS:
0067 E5F0             279             MOV     A,B                     ;REMAINDER TO ACCU
0069 75F00A           280             MOV     B,#10D
006C 84               281             DIV     AB
006D 7003             282             JNZ     TEN3                    ;REPLACE '0' WITH ' '
006F 20D507           283     TEN1:   JB      F0,TEN2
0072 2430             284     TEN3:   ADD     A,#30H                  ;CONVERSION TO ASCII
0074 120086           285             CALL    DISPLAY                 ;TO TERMINAL
0077 8005             286             SJMP    ONES
0079 7420             287     TEN2:   MOV     A,#' '
007B 120086           288             CALL    DISPLAY                 ;BLANK TO TERMINAL
                      289
                      290     ONES:
007E E5F0             291             MOV     A,B                     ;REMAINDER TO ACCU
0080 2430             292             ADD     A,#30H                  ;CONVERSION TO ASCII
0082 120086           293             CALL    DISPLAY                 ;TO TERMINAL
                      294
0085 22               295             RET
                      296     ;-------------------------------------
                      297
                      298     ;-------------------------------------
                      299     DISPLAY:
                      300
0086 3099FD           301             JNB     TI,$                    ;WAITING FOR END OF LAST TRANSMIT
0089 C299             302             CLR     TI
008B F599             303             MOV     SBUF,A                  ;SEND CHARACTER
008D 22               304             RET
```

0199-15

```
MCS-51 MACRO ASSEMBLER        APPLICATION NOTE ON SAB 80515 A/D CONVERTER        19.02.86

LOC  OBJ            LINE     SOURCE

                     305     ;-------------------------------------
                     306
                     307     ;-------------------------------------
                     308     DELAY:
                     309
008E 7F06            310             MOV     R7,#6
                     311     DELAY_LOOP:
0090 DDFE            312             DJNZ    R5,$
0092 DEFC            313             DJNZ    R6,DELAY_LOOP
0094 DFFA            314             DJNZ    R7,DELAY_LOOP
0096 22              315             RET
                     316     ;-------------------------------------
                     317
                     318
                     319     MESSAGE_1:
0097 1B              320             DB      1BH,'E'                ;CLEAR SCREEN
0098 45
0099 0D              321             DB      0DH,0AH                ;CR, LF
009A 0A
009B 0D              322             DB      0DH,0AH
009C 0A
009D 0D              323             DB      0DH,0AH
009E 0A
009F 0D              324             DB      0DH,0AH
00A0 0A
00A1 20202020        325             DB      '              ***    A / D  C O N V E R T E R   D E M O    ***'
00A5 20202020
00A9 20202020
00AD 20202A2A
00B1 2A202020
00B5 2041202F
00B9 20442020
00BD 43204F20
00C1 4E205620
00C5 45205220
00C9 54204520
00CD 52202020
00D1 44204520
00D5 4D204F20
00D9 2020202A
00DD 2A2A
00DF 0D              326             DB      0DH,0AH
00E0 0A
00E1 0D              327             DB      0DH,0AH
00E2 0A
00E3 20202020        328             DB      '              ***          S A B   8 0 5 1 5               ***'
00E7 20202020
00EB 20202020
00EF 20202A2A
00F3 2A202020
00F7 20202020
00FB 20202053
00FF 20412042
0103 20202038
0107 20302035
010B 20312035
010F 20202020
0113 20202020
0117 20202020
011B 2020202A
011F 2A2A
0121 0D              329             DB      0DH,0AH
0122 0A
0123 0D              330             DB      0DH,0AH
0124 0A
0125 0D              331             DB      0DH,0AH
0126 0A
0127 0D              332             DB      0DH,0AH
```

0199–16



```
MCS-51 MACRO ASSEMBLER              APPLICATION NOTE ON SAB 80515 A/D CONVERTER           19.02.86

LOC  OBJ            LINE     SOURCE

0128 0A
0129 0D              333            DB      0DH,0AH
012A 0A
012B 0D              334            DB      0DH,0AH
012C 0A
012D 0D              335            DB      0DH,0AH
012E 0A
012F 0D              336            DB      0DH,0AH
0130 0A
0131 0D              337            DB      0DH,0AH
0132 0A
0133 0D              338            DB      0DH,0AH
0134 0A
0135 20202020        339            DB      '     *AN0*     *AN1*     *AN2*     *AN3*     *AN4*     *AN5*     *AN6*     *AN7*'
0139 202A414E
013D 302A2020
0141 20202A41
0145 4E312A20
0149 2020202A
014D 414E322A
0151 20202020
0155 2A414E33
0159 2A202020
015D 202A414E
0161 342A2020
0165 20202A41
0169 4E352A20
016D 2020202A
0171 414E362A
0175 20202020
0179 2A414E37
017D 2A
017E 0D              340            DB      0DH,0AH
017F 0A
0180 00              341            DB      00                          ;END OF TEXT
                     342
                     343
                     344
                     345            END

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

0199-17

SIEMENS

March 1990

On-Chip A/D Converters in Siemen's SAB 8051-Based Microcontrollers

SAB-51 Family of Microcontrollers

Application Note

This application note discusses technical details of the on-chip A/D converter integrated into several members of the SIEMENS SAB 8051 Microcontroller Family. The information is a continuation of an earlier application note on the A/D converter in the SAB 80515/80535.

Operation of the A/D Converter in the SAB 80515

The technical details given are background information on this on-chip peripheral, intended to help in designing these microcontrollers into diffcult and electrically noisy environments.

This application note includes discussion of the A/D Converter specification, as well as recent updates to this specification.

SAB 80515's A/D Converter Architecture

The following description of the on-chip A/D converter unit (ADCU) concentrates on hardware and specification details of this particular microcontroller peripheral.

Conversion Principle

The ADCU conversion uses the successive approximation principle. Instead of an R-2R resistor ladder with which the analog input is compared, the ADCU uses an array of 256 small capacitors which are charged through the analog input. These capacitors not only perform the sample-and-hold function, but are also grouped into the binary weights used in the conversion process itself.

The "top ends" of all capacitors are connected to the comparator and, through the switch S1, to the positive reference voltage VAREF. The comparator itself is described in the next section. The connection to the "bottom" of each group of binary weighted capacitors can be connected to either the selected analog input or to one of the two internal reference voltages.

For the sake of explanation, it is assumed that the internal reference voltages are connected directly to the reference voltages applied to pins VAGND and VAREF.

Figure 1 shows a block diagram of the ADCU. The conversion process itself is described in the following three steps. The timings used in the explanation are based on the use of a crystal oscillator of 12 MHz.

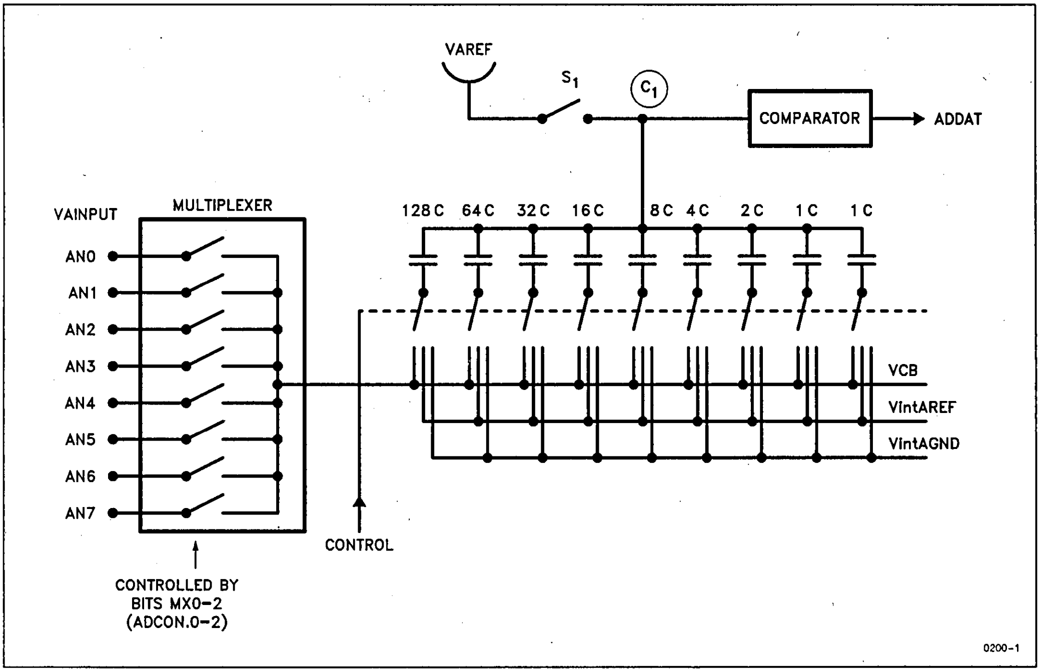


Figure 1. Detailed Block Diagram

Step 1 (1.0 μs–2.0 μs)

At the start of the conversion process, switch S1 is closed. This switch connects the top of all capacitors to the reference voltage VAREF. The bottom of each group of capacitors is connected to one analog input by selecting one of eight multiplexer channels with bits ADCON0–ADCON2.

Interpretation of Step 1

This step is the first part of the sample phase and is known as the load time (TL) in the specification. During these 2 μs, the complete capacitor-array has to be precharged in the analog input through the VAREF-pin. At the end of this period, the voltage VCB at the bottom of the capacitor network must be the same as the analog input being measured (= VAINPUT).

Step 2 (3.0 μs–5.0 μs)

Switch S1 is opened. The top of the capacitor array remains connected to the high impedance comparator input and the bottom to the analog input. During the next 3 μs, the comparator is adjusted to the voltage of node C1 (for details of the comparator see next section). This potential still equals to the voltage at VAREF since node C1 was precharged to that voltage during step 1.

Interpretation of Step 2

This second part of the sample phase follows the load phase. The analog input is still connected to the bottom of the capacitor array. Any change in the voltage at the analog input pin such as a spike or transition will be capacitively coupled to node C1. This will disturb the adjustment of the comparator. During this time the analog input must therefore be held strictly constant.

Step 3 (Remaining Conversion Time)

During this time the actual conversion process (successive approximation) takes place:

The bottom of each group of capacitors is disconnected from the analog input and pulled to VINTAGND. Node C1 is currently at a potential VC1 = (VAREF − VAINPUT).

The first group of capacitor-cells (128C, corresponding to the Most Significant Bit (MSB)), is then switched to VINTAREF. This causes a charge transfer in the capacitor array and thereby a superimposition of VC1 with VAREF/2, i.e. the potential of node C1 alters to become VC1 = (VAREF − VAINPUT + VAREF/2). The new VC1 is now compared to VAREF (adjustment-voltage of the comparator). Depending on the result of the comparison, this 128C-group remains at VINTAREF (for VC1 < VINTAREF) or is switched back to VINTAGND (for VC1 > VINTAREF).

The same process is repeated with the next group of capacitor-cells (in this case the 64C capacitor). The conversion is complete when the last capacitor-group (1C) of one capacitor cell has been compared.

Upon completion of the comparison, the voltage VC1 applied to the comparator can be calaculated by the formula:

$$\mathrm{VAREF} - \mathrm{VAINPUT} + \sum_{i=0}^{7} \mathrm{bi} \times \frac{\mathrm{VINTAREF}}{2^{8-i}}$$

All bits bi (i = 0 to 7) are latched in the position of the switches at the bottom of each capacitor group, thereby providing the result of the conversion process.

Interpretation of Step 3

The actual conversion is done in step 3 when the analog input voltage is measured against the weighted capacitor network. Since the analog input is disconnected from the capacitor network, changes at the analog input no longer affect the conversion result. Note that during this time, charge transfer in the capacitor-array still causes dynamic current spikes drawn from the analog reference. However, from the user's point of view, step 3 is the least critical phase of the whole conversion process.

The Comparator

The accuracy of an A/D-converter is very dependent on the comparator used. The previous section, especially step 2, stated that the ADCU comparator is susceptible to noise or spikes through the analog input pin during certain time windows. A standard comparator (such as is used in a common OP-AMP application) compares the input value with a reference, which is always present. The ADCU comparator can be viewed as an inverting three-stage amplifier with some coupling capacitors. It is precharged initially with a reference voltage which is then used as a threshold point.

The principle of this three-stage comparator can be explained through the function of one stage. It consists of a coupling capacitor in series with an inverting amplifier as shown in Figure 2a. A feedback switch allows the inverter's output to be short-circuited with its input.

The feedback-switch is closed during the load phase of the comparator-stage, as discussed in step 1. This causes the coupling capacitor to buffer the differential voltage VD between the input stage and the operating point of the inverting amplifier. Opening the feedback switch activates the comparator at the previously adjusted potential (Figure 2b). The amplifier responds to small variations from this potential with an amplified output. A concatenation of three of these stages results in an amplification which digitizes even the smallest variation at the comparator input.

Additionally, the three stages of the adjustment process can be serialized by opening the three feedback switches one after another. This compensates for unintentional noise in the coupling capacitors which might be generated by opening the switches during the 3 μs duration of step 2 mentioned above. This explains why the comparator is susceptible to noise during this period. Any spikes or noise capacitively coupled to its input during the last phase of the sample time might shift the comparator threshold.

A smooth analog input signal should be supplied during the whole sample phase (TS). A transition on the analog input signal during the critical time of the sample phase may give unexpected results. On the other hand, this comparator concept, combined with the conversion principle described above, provides high immunity against temperature changes and results in a narrow variation of device parameters during production.

Expansion and Update of the ADCU Specification

The ADCU specifications in the datasheet give generally condensed information on A/D converter characteristics. This section provides the reader with a more detailed interpretation of the main points in the specification. It also updates the specification with more recent information.

VAREF/VAGND Voltages

The limits of the device reference voltage inputs are a function of the microcontroller type and operational supply voltage. Some devices have a standard ratiometric ADCU on board, which allows an adjustment of the reference over a wide range of external voltages. Other devices have internally adjustable reference voltages, the voltage range being controlled by software.

The latter type cannot function correctly with any variation in the reference voltages. The voltages applied to VAREF and VAGND are specified within a very narrow range of the supply voltage for such devices.

Whichever type of microcontroller is used, the reference voltage, once adjusted to the desired value within specifications, must be held steady during the entire conversion process. This, in turn, requires a low impedance to the reference source. Reference voltage impedance and analog source impedance requirements are discussed in the following sections.

For reference voltage specifications refer to either the A/D Converter Characteristics given in the respective datasheets or to the list given below in Table 1.

For all parts VAREF − VAGND must be greater than 1V or four steps of the internally programmable reference voltages, whichever is less.

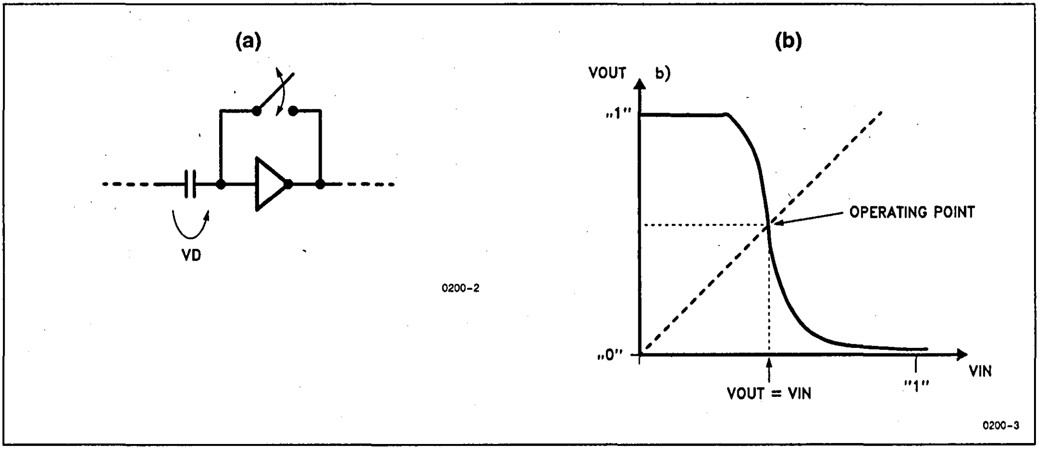


Figure 2. One Stage of the Comparator

Table 1. Voltage Limits on V_{AREF} and V_{AGND}

Device	VSS	VCC	VAGNDmin	VAGNDmax	VAREFmin	VAREFmax
SAB 80512	0V	5V ±10%	VSS − 0.2V	VAREF − 1V	VAGND + 1V	VCC + 5%
SAB 80515	0V	5V ±10%	VSS − 0.2V	VSS + 0.2V	VCC − 5%	VCC + 5%
SAB 80C515	0V	5V ±10%	VSS − 0.2V	VSS + 0.2V	VCC − 5%	VCC + 5%

Timing

Timing requirements have already been discussed in the description of the conversion principle.

The load time TL is an important ADCU characteristic and has therefore been added to the specification.

The load time TL is specified as the period during which the ADCU internal capacitance must be charged by the analog source. TL is the first part of the sample time TS. The ADCU uses the remaining part to adjust its comparator. The entire conversion time TC consists of TS (which includes TL) and a part during which the conversion is performed. TC is specified in the datasheet and differs slightly among device types.

Analog and Reference Voltage Source Impedance Requirements

A/D Converter Characteristics in the datasheets contains absolute values required for impedances of the reference and analog source only. This section gives pointers on determining the ADCU's requirements on the analog part of an application as well.

Impedance of the Analog Source

Two parameters from the ADCU's specification are necessary to determine the maximum impedance of a signal source such as a sensor:

- — the load time TL
- — the maximum capacitance of the on-chip capacitor array (CTot).

Figure 3 shows a simple equivalent circuit for the ADCU during load time TL. At the moment a conversion is started, a multiplexer switch to the selected channel is closed and the analog input is connected to the internal capacitors of the ADCU. This is shown as SMUX in Figure 3.

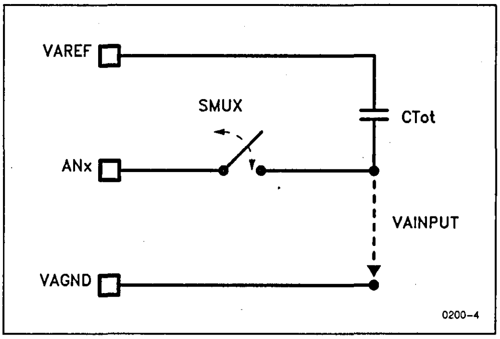


Figure 3. Equivalent Circuit for the ADCU during Load Time

The ADCU's entire internal capacitance consists of the capacitor array, which is typically 25 pF, and stray capacitance from the substrate equivalent to another 20 pF–40 pF to substrate. The actual amount depends on the microcontroller itself. In any case, the internal capacitance total must be charged by the analog source. This implies that a constant input impedance cannot be specified for the ADCU.

The switch SMUX and the internal traces to the capacitors array have very low resistance. During TL, the ADCU can be regarded as a capacitor with a capacitance of CTot. The internal resistance of the whole analog circuitry connected to the analog inputs of the ADCU can be defined as RI. RI and CTot form an RC-element, with time constant t determined by "t = RI * CTot".

Given that an accuracy better than 0.05% is required, the formula

$VC = (1 - e^{-TL/\tau}) * VAINPUT$

gives $\tau = 0.13 * TL$

As an example, assume that the load time of the ADCU is TL = 2 μs and its maximum internal capacitance CTot = 65 pF. The above formulae result in a maximum impedance for the analog source of

$RI = 0.13\ TL/CTot = 4\ k\Omega$

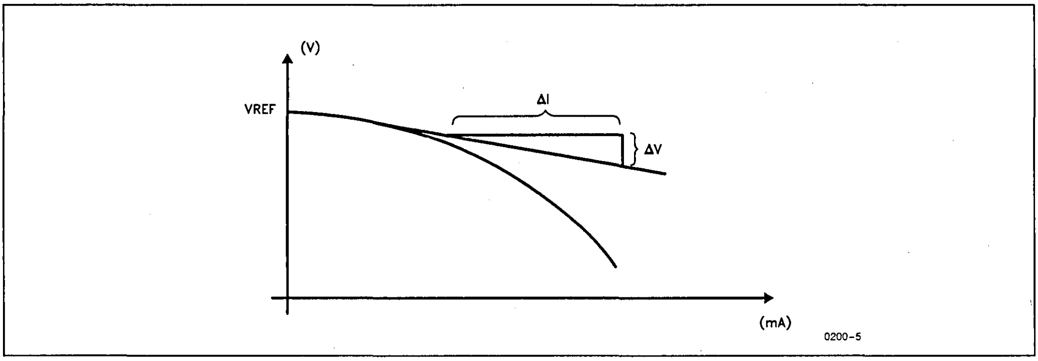


Figure 4. Definition of the Reference Supply Impedance

Observing the above rules for the analog circuit impedance ensures that at least 99.95% of VAINPUT is sampled during load time TL. Sampling actually continues during the whole time TS to improve the match between VAINPUT and the voltage at CTot.

The value for RI given above is meant as an example. For a given application, the calculation should be done using the values for CTot and the accuracy desired.

Impedance of the Reference Source

During the conversion process, each group of capacitor cells is individually switched to either VAREF or VAGND. Due to this switching and associated charge transfers in the capacitor array, the reference source must supply additional current over and above the current some parts use to generate internal reference voltages.

The reference output impedance must be low enough to supply this additional current and is therefore specified as a differential impedance. The diagram in Figure 4 illustrates the differential impedance, rD, which is given by the formula:

$$rD = \frac{dV}{dI}$$

where V = VREF

The actual value for rD can be referenced under A/D converter characteristics in the the datasheet.

3.4 Specification of Errors

Several error sources which modify the ideal transfer characteristic of an A/D converter were defined and discussed in the application note titled "Operation of the A/D converter in the SAB 80515/80535". The error sources included:

- Offset Error
- Integral Non-Linearity
- Differential Non-Linearity
- Gain Error

Figure 5a shows a diagram of the ideal transfer characteristic of any A/D converter. A 3-bit converter is shown here for clarity. All analog values within a given quantum are represented by the same digital value, which corresponds to the mid value of this quantum. Connecting all of these mid values leads to a straight line, which, for an ideal A/D converter, is the diagonal in the 1st quadrant. This also implies that the first step to the digital value 01 occurs when an analog input voltage is applied which corresponds to a ½ LSB. All error definitions stated in the datasheet refers to this ideal transfer characteristic.

All the above errors affect the A/D converter ideal transfer characteristic. Their effect is defined as the Total Unadjusted Error (TUE), which is now an A/D converter characteristic.

The TUE of the ADCU is not simply the sum of several individually measured errors. Since some ADC errors can cancel each other, e.g. the offset and the gain error, the Total Unadjusted Error can be less than the sum of individual errors.

Some ADCU errors can be compensated for. A negative offset error can be adjusted in software by subtracting known offset from the actually measured value. The TUE, however is an "unadjusted" error, which means that no correction method has been taken into account. It therefore gives a true image of the converter accuracy.

Figure 5b shows the TUE as a maximum deviation of the actual transfer characteristic (bold line) from the ideal transfer characteristic corresponding to the full scale range. The TUE defines the range of the transfer characteristic where the real value must lie.

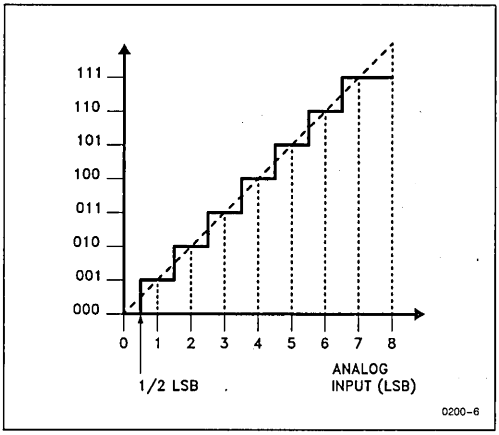


Figure 5a. Ideal Transfer Characteristic

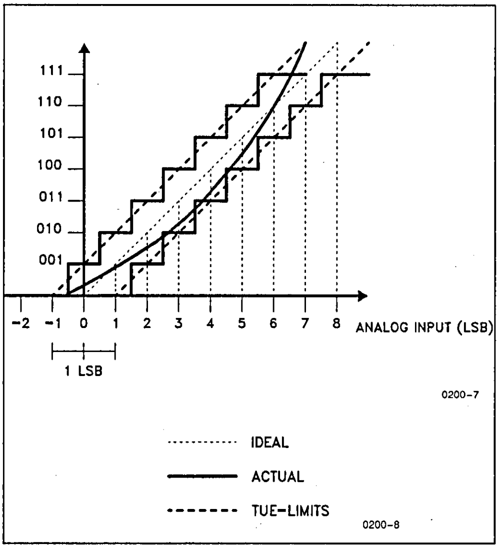


Figure 5b. The Total Unadjusted Error

Accuracy/Resolution in a Reduced Reference Voltage Range

General Considerations

All A/D converter units of the SAB 8051 family are ratiometric A/D converters. Some (e.g the SAB 80512) allow a reduction of the reference voltage range external to the chip, others (e.g., the SAB 80515) provide software-adjustable internal reference voltages. This section deals with the effect of reducing the reference voltage range on ADCU accuracy.

Generally, the absolute **accuracy** of an 8-bit A/D converter cannot be improved by using a smaller reference voltage range. The errors in an ADCU originate in physical phenomena like temperature substrate noise, etc., and these cannot be reduced by reducing the reference voltage.

However, this is not the point in ratiometric A/D converters. The main advantage of these converters is that they are able to provide higher resolution. Since the distinction between accuracy and resolution can become a source of confusion, it is worth discussing further.

Accuracy versus Resolution

The advantage of ratiometric A/D converters is that they increase the resolution of the analog input. Increasing the resolution just means that the quantum of analog values which corresponds to one digital value is reduced.

In case of the Siemens Microcontrollers, this is performed by selecting a smaller reference voltage range, i.e. by varying the lower or higher reference voltage. All ADCUs in the Siemens SAB 8051 family provide the equivalent of a 10-bit resolution by a reduction of the reference voltage range to as little as 1V. An ADCU such as that in the SAB 80515 allows for a dynamic (software controlled) variation of the reference voltage. Therefore this converter, which is actually an 8-bit converter, can look like a 10-bit converter over the full conversion-range from 0V to 5V using appropriate software. The absolute accuracy of this "10-bit converter", is the absolute deviation between the analog input and the digital result. This is not necessarily any better than that of the "8-bit converter" (VAGND = 0V and VAREF = 5V), although it can be improved by implementing correction methods in the control program. The resolution, on the other hand, is improved by using a smaller reference voltage range. An analog interval which gave one digital step with a 5V reference voltage gives four digital steps in the same interval, and therefore an additional two bits of resolution.

The adjustable reference voltage feature, in combination with software error correction algorithms make precise A-to-D conversions with low-cost on-chip converters possible.

Errors in a Reduced Reference Voltage Range

All error definitions in the datasheet A/D converter characteristics refer to the standard conversion range of 0V to 5V, (the "8-bit range"). For the purpose of the following discussion, the term "LSB8" is used to describe one LSB (least significant bit) over this range. Similarly, "LSB10" describes one LSB of a conversion made within a range of a quarter of the full reference voltage range.

As discussed above, the four different errors which affect the accuracy of an A/D converter are:

Offset Error
Gain Error
DNLE (Differential Nonlinearity Error)
INLE (Integral Nonlinearity Error)

This section discusses the effect of a variation in reference voltage range on these errors.

Offset- and Gain Error, Integral Nonlinearity

These errors are systematic errors. This means that they occur independent of application, disregarding temperature drift. Reducing of the reference voltage range will not reduce these errors.

Take as an example an ADCU with a reference voltage range of VAGND = 0V and VAREF = 5.0V. Assume a negative offset error of ½ LSB8 and no other errors. This means that the first transition of the transfer characteristic (the change from 00H to 01H) occurs at an input voltage of VAINPUT = 0.00V. According to the ideal transfer characteristic, the absolute offset error is approximately −10 mV. If we now reduce the upper internal reference voltage to 1.25V, then 1 LSB (now an "LSB10") of the digital result now refers to a quantum of approximately 5 mV. For VAINPUT = 0.00V, with the same Offset error the conversion result is 2 LSB10 (ADDAT = 02H). Thus, an absolute offset error is independent of the reference voltage selected. In other words, the error is the same regardless which "window" within the full scale range is selected.

A similar analysis shows that the same applies to both the Integral Nonlinearity Error and to the Gain Error if these are considered separately.

Both the Offset Error, the Integral Nonlinearity Error can be disregarded for our ADCUs, since they can be compensated for in software. The Gain Error is more significant. Since it is given as the difference in slope measured at its biggest extent, it is by definition a scaled error comparison of an actual A-to-D converter transfer characteristic and comparing with an ideal characteristic shows a combination of errors. Some of these compensate each other and can be determined only at certain points on the transfer characteristic.

To determine the actual "absolute error" at a given input voltage, both Gain and Offset Errors must be summed. Differential Nonlinearity is not considered yet. It will be discussed in detail later. The following example demonstrates Gain and Offset errors for different reference voltage ranges.

To simplify matters, again consider the 3-bit converter. This has an Offset error of −2 LSB and a Gain error of +1 LSB over full range. Figure 6 shows the effect of both errors on the transfer characteristic of the converter. Reducing the reference voltage range has no influence on the absolute accuracy of the conversion. The dotted window in Figure 6 shows the reduced reference voltage range. In this case, it is half the size of the full scale range. Using a terminology similar to that above, LSBs are therefore LSB3 in the full range and LSB4 in the reduced.

The dotted window in Figure 6 illustrates that changing the reference voltages just extracts a window out of the full scale range and increases resolution. The actual Offset and Gain Error line crosses the y-axis of the small diagram at the binary value 011 three LSB4 units. This three LSB4 error is a combined error of four LSB4 Offset Error and −1 LSB4 Gain Error. The Gain Error in the reduced range is scaled down to one LSB4 (½ LSB3) because this error is measured in the middle of the transfer characteristic. The lower reference voltage of the reduced range is at 2 LSB3 and the offset error also 2 LSB3, which together gives 4 LSB3 on the x-axis. This is the middle of the full range.

Summarising, it can be said that Offset-, Gain-Error and Integral Nonlinearity of the ratiometric A-to-D converters in the SIEMENS 8051-family can be regarded as absolute errors. Absolute errors are the deviation of the actual from the ideal value at a given input voltage. Changing the reference voltages has no effect on these errors. The number of LSBs in which the errors are expressed depends on the reference range and therefore on the "LSB-Unit" (LSB8 or LSB10).

Differential Nonlinearity Error (DNLE)

The Offset and Gain Error, described above are, in most cases, bigger than Nonlinear Errors. On the other hand, they are stable and can be compensated for by software.

The Differential Nonlinearity Error is a statistical error with more than one source. A major part is due to the noise inherent in any mixed analog/digital system. This noise devices from the controller's environment on the PC-board and from the device itself. Even good A/D Converter test-boards carry some noise on the analog lines. Distinguishing which part of the DNLE is induced by the test equipment and which by the chip itself is difficult. There is also a small DNLE in the A/D-converter which can't be avoided. This "self-induced" DNLE comes from on-chip noise and noise on the chip's substrate. More complex origins are inherent in conversion method principles. Empirical measurements show that DNLE component is not a constant error and therefore varies with reference voltage range. The DNLE due to the converter itself in fact shows a dependence to the reference voltage selected. *This* DNL component scales with the size of the window selected.

10-Bit Resolution?

The datasheet error definition disagrees with the 10-bit resolution claimed above. 10-bit resolution is possible, even with a differential nonlinearity error of 1 eight-bit-LSB (LSB8) maximum. This corresponds to approximately 20 mV.

A DNLE of 1 LSB8 would result in missing codes in the transfer characteristics of the 10-bit range, which means that 10-bit resolution appears to be "inappropriate".

However, the DNLE discussed in the above paragraph is an exception to the principle of "absolute errors". The "self-induced" DNLE of the chip presents no problem for 10-bit resolution. The problem is the chip's susceptibility to external noise. Keeping this noise as small as possible is a major task for the design engineer. Further application notes will give some design hints for device connections necessary for high resolution with minimal errors.

In summary, the typical DNLE is generally under 1 LSB8 and depends heavily on the PC-board environment in which the microcontroller is operated. Observing the design rules for dealing with weak analog signals leads to a typical DNLE of less than 1 LSB10 (10-bit LSB) with no missing code.

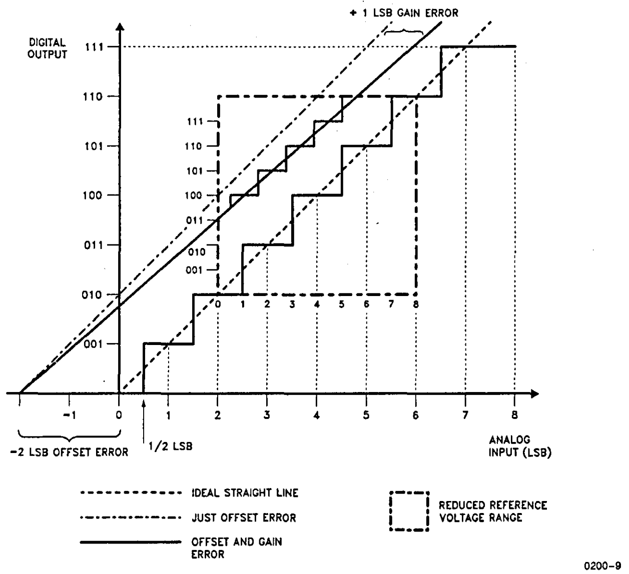


Figure 6. Offset and Gain Error in a Reduced Reference Voltage Range

All errors other than DNLE are systematic errors. They affect the absolute accuracy of A/D conversion, but cannot lead to missing codes and therefore don't affect resolution.

Errors in the Internal Reference Voltages

For microcontrollers which provide on-board ADCUs with programmable reference voltages, another relevant parameter is the accuracy of the internal reference voltages. The absolute accuracy of a conversion result measured in a reduced reference voltage range also depends on the accuracy with which the internal reference voltages can be adjusted.

The internal reference voltages are generated by a simple digital-to-analog converter. It is essentially a tapped resistor ladder consisting of 16 equal resistors.

A write-to-DAPR instruction connects the internal reference voltage nodes to the appropriate taps. The lower and upper nibble of the SFR DAPR determines the tap position for the lower and higher reference voltage respectively. The accuracy of the internal reference voltages therefore depends on the accuracy of the internal resistor network. In the ideal case where all resistors of the network have exactly the same value, the differential voltage between two taps of the resistor ladder is exactly $1/16$ of the reference voltage applied externally.

Slight variations in the ratio of the resistance values are inherent in the device and are unavoidable. These variations result in a so-called internal reference error which is now a part of the A/D Converter Characteristics specification.

This VINTREFERR defines the maximum deviation of the actual internal reference voltage from the ideal value. The ideal value depends on the reference voltage applied to the chip. The formulae given in the user's manual of the SAB 80515/80535 can be used to calculate the internal reference voltages.

Example 1:
VAGND = 0.00V
VAREF = 5.12V DAPR = C4H

$$\left\{\begin{array}{l} \text{VINTAGND} = \text{VAGND} + \dfrac{\text{DAPR (0-3)}}{16} \bullet (\text{VAREF} - \text{VAGND}) = 1.280\text{V} \\[2ex] \text{VINTAREF} = \text{VAGND} + \dfrac{\text{DAPR(4-7)}}{16} \bullet \text{VAREF} - \text{VAGND}) = 3.840\text{V} \end{array}\right.$$

The above values for VINTAGND and VINTAFER are the ideal values. With VINTREFERR = ±15 mV, we get the following:

$1.265\text{V} < \text{VINTAGND} < 1.295\text{V}$

$3.825\text{V} < \text{VINTAREF} < 3.855\text{V}$

The resolution of the conversion in this example is 10 mV (9-bit). That means that in a 9-bit conversion with a reference voltage range of DAPR = C4H, the above deviation of the internal reference voltage must be taken into account.

Consequences of VINTREFERR

The importance of this error depends on the application.

The following two examples show the effect of the VINTREFERR in typical applications:

A. Measuring Absolute Voltages

This is a standard application for any A/D-Converter.

For the calculation of the Total Unadjusted Error in a **reduced reference voltage range**, VINTREFERR must be taken into account. If both internal reference voltages are set to taps of the resistor ladder, then the VINTREFERR or VINTAGND can be considered as an additional offset error. VINTREFERR of VINTAREF is additional gain error in the conversion. The following example shows the effect of the VINTREFERR on the accuracy of the conversion:

VAGND = 0.00V DAPR = 84H

VAREF = 5.12V VAINPUT = 1.500V

$$\left\{\begin{array}{l} \text{VINTAGND} = 1.280\text{V (ideal)} \\ \text{VINTAREF} = 2.56\text{V (ideal)} \\ \text{Step Width} = \dfrac{\text{VINTAREF} - \text{VINTAGND}}{256} = 5\text{ mV} \end{array}\right.$$

The correct result of the conversion would be

$$\text{ADDAT} = \frac{\text{VAINPUT} - \text{VINTAGND}}{\text{Step Width}} = 2\text{CH}$$

Lets assume that a hypothetic VINTREFERR leads to the following internal reference voltages.

VINTAGND = 1.28V − 5 mV

VINTAREF = 2.56V − 10 mV

This gives a new step width of 4.98 mV and an absolute offset error caused by the new VINTAGND of one LSB10. The actual result is now

$$\text{ADDAT} = \frac{1.500\text{V} - 1.275\text{V}}{4.98 \text{ mV}} = 45\text{D} = 2\text{DH}$$

Therefore, the error caused by the internal references in this example is one LSB10.

B. Measuring Differential Voltages

The feature of software-adjustable reference voltages is ideal for a tracking converter application.

The requirements for a tracking converter are high resolution and monotonic behaviour. Both the requirements are accomplished by the ADCU of the SAB 80(C)515. In this case, the internal reference error is not significant. The maximum VINTREFERR merely changes the analog quantum for one digital value by

$$\pm\frac{\text{VINTREFERR}}{256}$$

As an example, taking the same parameters as in example 1:

VAREF = 5.12V VAGND = 0.00V

The smallest possible reference voltage range is therefore:

$$\frac{\text{VINTAREF} - \text{VINTAGND}}{4} = 1.28\text{V}$$

The resolution of the conversion is in this case 5.00 mV. That is, the analog quantum corresponding to each LSB is 5.00 mV. If we assume the worst case for both internal reference (−15 mV for VINTAGND and +15 mV for VINTAREF or vice versa), the analog equivalent to one LSB would expand or compress by 0.117 mV.

In other words this results in a step width error in the worst case of ~2.3%. The absolute amount of the error therefore increases with the differential voltage of the signal between two sample points.

Summarising, it may be said that the VINTREFERR, the error of the internal reference, cannot be disregarded in applications which use the internal reference voltages since this error affects the absolute accuracy of a conversion.

For clarity, the above examples deal with rather large errors. Typical values for the VINTREFERR are less than 5 mV.

Further, it should be mentioned that specifications are constantly being updated. Values are given in this application note to illustrate errors specified in the datasheet. The current datasheet should always be consulted for the latest error definitions and values.

References:

1. Application Note—Operation of the A/D Converter in the SAB 80515
2. H. Kessler, R. Rossgotterer

   Ein 8-Bit-Analog-Digital-Wandler in MOS-Silizium-Gate-Technologie nach dem Ladungsverteilungsverfahren

   Siemens Forshc.-u. Entwickl.-Ber. Bd. 8 (1979) Nr. 5
3. SAB 80515/80535, 8-Bit Single-Chip Microcontroller, Datasheet

   SAB 80512/80532, 8-Bit Single-Chip Microcontroller, Datasheet

   SAB 80C515/80C535 8-Bit Single-Chip Microcontroller, Datasheet

   SAB 80512K, 8-Bit Single-Chip Microcontroller, ROM-less Version, Datasheet

   SAB 80515K, 8-Bit Single-Chip Microcontroller, ROM-less Version, Datasheet
4. Microcomputer Components

   SAB 80515/80535 8-Bit Single-Chip Microcontroller User's Manual 7.85
5. Microcomputer Components

   SAB 80512/80532 8-Bit-Single-Chip Microcontroller User's Manual 2.88

March 1990

Ten Bits of Resolution with the 8-Bit On-Chip A/D Converter of the SAB 80515/80535

SAB 80515/80535

Application Note

This application note describes a software routine to achieve 10 bits of resolution with the A/D converter of the SAB 80515/535.

Analog-to-digital (A/D) and digital-to-analog (D/A) converters are devices which interface physical parameters, which are analog, to digital computation and control. Some applications in which A/D converters are used include processing systems, sampled-data control systems, data telemetry systems and automatic test systems. Besides A/D converters, these systems usually employ transducers to interface with physical/analog quantities such as temperature, pressure, flow, acceleration and position, as well as microcontrollers or microprocessors to process the acquired data.

Siemens SAB 80515/535 microcontroller facilitates the design of a data conversion system by providing an on-chip A/D converter. The inherent 8031 based architecture of the SAB 80515/535 allows direct handling of 8-bit quantities only. However, by combining the on-chip special features of the A/D converter and the processing capability of the SAB 80515/535, a 10-bit result can be achieved and stored in two 8-bit registers.

Quantization and Resolution of an A/D Converter

Quantizing is the process of transforming a continuous analog signal into a set of discrete output states. Resolution of an A/D converter is the number of bits required to describe the output states. The number of output states for a binary coded A/D converter is 2^n where n is the number of bits. Thus an 8-bit A/D converter has an eight bit resolution and 256 output states. A 10-bit A/D converter has a 10-bit resolution and 1024 output states.

In any part of the input range of the A/D converter, there is a small range of analog values within which the same digital output is produced. This small range is known as the analog quantization size or quantum Q. The quantum is found by dividing the full scale analog range by the number of output states.

Therefore, $Q = FSR/2^n$ where FSR is the Full Scale Range and n is the number of bits.

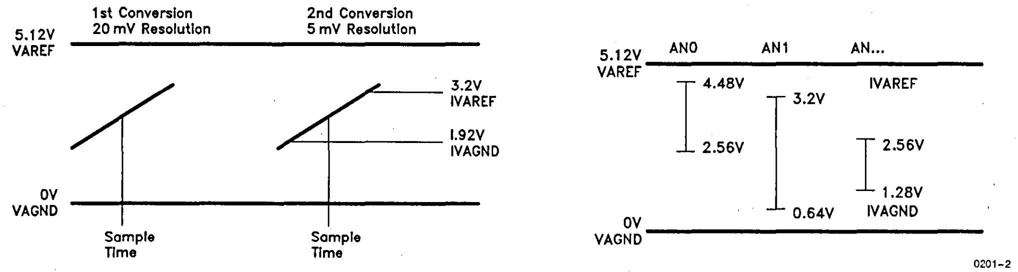
The quantum Q, introduces an error and for a given analog input value to an A/D converter, the output error will vary anywhere from 0 to plus or minus Q/2. This error is called quantization noise. It can be reduced only by increasing the resolution of the converter, thereby making the quantization finer.

With 1024 possible quantized output states from a 10-bit A/D converter, a quantum for a full scale range of 5.12V will be equal to $5.12V/2^{10} = 5$ mV. The same level of quantum can be achieved with the 8-bit A/D converter on the SAB 80515/535. By programming the internal reference voltage sources, the entire analog input range of 5.12V may be divided into four ranges of 1.28V each. For a full scale range of 1.28V, the quantum is then equal to $1.28/2^8 = 5$ mV. Therefore, the four ranges, each with 256 quantized output states will give a total of 1024 digital output states with the same quantum level as achieved by using a 10-bit A/D converter.

A/D Converter with Programmable Reference Voltages

The 8-bit A/D converter of the SAB 80515/535 has 8 multiplexed analog inputs and its operation is based on the method of successive approximation by using a capacitive load distribution. The use of capacitors in place of a resistive network ensures a better immunity against temperature and frequency changes, thereby providing a better accuracy of the A/D converter. The analog signal at the selected input channel is sampled for 5 machine cycles (5 μs at an oscillator frequency of 12 MHz), which will then be held constant at the sampled level for the rest of the conversion time of 10 μs at an oscillator frequency of 12 MHz. One-time or continuous conversions may be performed. The end of a conversion may cause an interrupt.

Moreover, the two internal reference voltages IVAREF and IVAGND can be programmed for a 4-bit resolution (16 steps), referred to the externally applied reference voltage VAREF. Each 4-bit value, one for IVAGND and another for IVAREF, is put in a D/A converter program register called DAPR, DAPR is an 8-bit register in which the low-order nibble holds the digital value of IVAGND, while the high-order nibble holds the digital value of IVAREF. As soon as a digital value is written to DAPR, the corresponding analog levels for the IVAREF and IVAGND are computed and the A/D conversion is then performed in this range. By reducing the range of conversion, the resolution of the A/D converter can be increased up to 10 bits (Figure 1). Different internal analog voltage ranges may be set for each input by means of software.



1. Improved resolution by second measurement

2. Matching various external voltage ranges to the A/D converter

Figure 1. Increase in A/D Converter Resolution

Software

A simple way to achieve 10 bits of resolution with A/D converter of the SAB 80515/535 is to divide the entire 5.12V range (VAREF = 5.12V and VAGND = 0V) into four equal parts of 1.28V each. This would give four ranges starting from 0V to 1.28V, 1.28V to 2.56V, 2.56V to 3.84 and 3.84V to 5.12V. First an 8-bit conversion is performed by programming the IVAREF and IVAGND to 5.12V and 0V respectively. The 8-bit result obtained from this conversion has unique value for the two Most Significant Bits (MSBs) for each of the four ranges, i.e. 00XX XXXXB for range from 0V to 1.28V, 01XX XXXXB for range from 1.28V to 2.56, etc. These two MSBs constitute the two MSBs of the final 10-bit result. Based on this first conversion a decision is made as to which of the four ranges the analog signal belongs. The IVAREF and IVAGND are then programmed to have the respective values in the narrow range and another conversion is performed. The 8-bit result obtained, gives the next 8 bits of the final 10-bit result.

This straight-forward method of achieving 10-bit resolution introduces an error when the analog signal lies at the boundary of two adjacent ranges. The irreducible error which results from the quantizing process may produce the digital result in one range, while the actual analog signal may belong to the adjacent range. This indeterminateness of the correct range introduces an additional error of ½ LSB.

To circumvent this problem, another approach is adopted, in which the entire range of 5.12V is divided into several sub-ranges of 1.28V such that each sub-range overlaps the other at its mid-point, as shown in Figure 2. The sub-ranges are numbered from 1 through 7. The even-numbered sub-range is offset by 0.64V from its preceding odd-numbered sub-range, thereby overlapping the upper half of the preceding sub-range. The algorithm to select a sub-range for a given analog signal is discussed later but the basic principle to achieve 10-bit resolution is the same as before.

First, an 8-bit result of the A/D conversion is performed in the 5.12V range and then a decision is made as to which of the seven sub-ranges the signal belongs. The second A/D conversion in the selected sub-range then gives the lower 8 bits of the final 10-bit result in one of the two following ways. If an odd-numbered sub-range is selected, then the 8-bit result directly gives the lower 8 bits of the 10-bit result. However, if an even-numbered sub-range is selected, which is offset by 0.64V from the odd-numbered sub-range, an 80H (digital equivalent of FSR/2 = 1.28V/2 = 0.64V) is added to the digital result to compensate for this offset. Any carry from this addition will also modify the value of the two MSBs of the final 10-bit result.

Algorithm

Figure 2 graphically represents the assignment of the DAPR register in the narrow range. The left nibble of the 8-bit result obtained from the A/D converter in the full range is used as a pointer in the DAPR look-up table to get a value for DAPR register for the second conversion in the narrow range. The second bit of the selected DAPR value for the narrow range decides whether an even or odd-numbered sub-range is selected. When this bit is '0' (for an odd-numbered sub-range), the third and fourth bits of the DAPR value become the two MSBs of the final 10-bit result. When an even-numbered sub-range is selected, an 80H is added to the A/D conversion result obtained in the narrow range. Again, the third and fourth bits of the selected DAPR value corresponds to the two MSBs of the final 10-bit result. The carry generated by the addition of 80H to the digital result is added to these bits to determine the two MSBs of the final 10-bit result. These two bits concatenated with the 8-bit result obtained by programming the DAPR register in the narrow range gives the final 10-bit result.

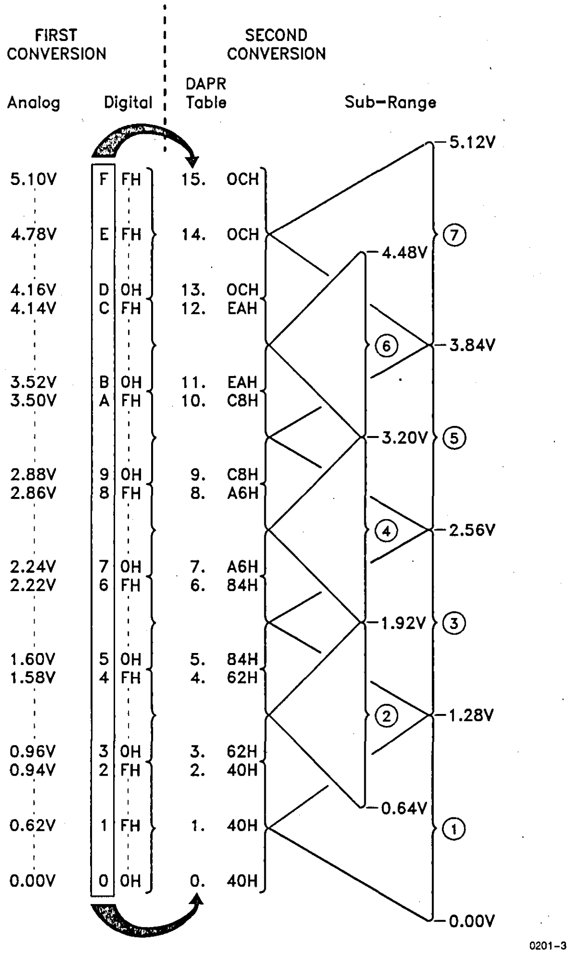


Figure 2. A/D Converter Sub-Ranges

System Design Hints

The recommended design for use of the SAB 80515/535 is shown in Figure 3. Standard design rules such as low impedance wire and minimized connector lengths will ensure low system resistance and inductance. In addition, there are several key points to consider for the most accurate and stable performance of the A/D converter.

First, a dedicated reference voltage for the A/D converter is highly desirable. Since V_{CC} is typically noisy and unstable, the V_{CC} line is unsuitable as an analog reference. In addition, many standard voltage regulators are not stable enough to meet the requirements of the SAB 80515, if maximum accuracy is desired. Strict adherence to the VAREF specification is recommended.

Additionally, separate digital and analog ground lines (tied near the source) are recommended to provide a smooth analog GND. However, if there is a large voltage differential between the ground lines, it is desirable to tie the ground lines close to the chip. By doing so, the noise in the digital ground will be reflected in the analog ground, resulting in a loss of accuracy.

Lastly, Figure 3 is a schematic for the NMOS implementation of the SAB 80515/535. In the CMOS implementation of this device, the V_{BB} pin becomes another V_{CC} pin. Therefore, when designing for both NMOS and CMOS devices, it will be necessary to bring a V_{CC} line close to the V_{BB} pin, to be connected via a jumper pin, as required by the CMOS device. In addition, when using CMOS devices, the long capacitor connected between V_{SS} and V_{CC} (pin 68) may be removed so that only one short 100 nF capacitor remains between V_{SS} and V_{CC} (pins 37 and 38).

Software Hints

A typical software routine to achieve 10 bit resolution is given in Appendix A. To reduce noise, the address lines should remain as quiescent as possible to minimize the power draw from the V_{CC}. Therefore, it is recommended that other tasks which change the address lines are not initiated while waiting for the end of conversion.

In addition, the Jump if Busy (JB BSY,addr) instruction should be placed at the beginning of a page in the program memory where the upper 14 address lines don't change. An example is shown:

Good Example:	*Bad Example:*
Addr.	Addr.
XX X0H JB	0F FFH JB
XX X1H BSY	10 00H BSY
XX X2H rel. address	10 01H rel. address

To ensure highest accuracy, self-calibration routines should be performed each time before the A/D converter is used, to monitor the offset error at baseline, and gain error of the device. A simple test for offset may be performed by inputting an analog zero, and checking the digital result. Likewise, gain error may be determined by feeding known voltages into the A/D converter and comparing the digital outputs.

References

1. Siemens SAB 80515/535 User's Manual.

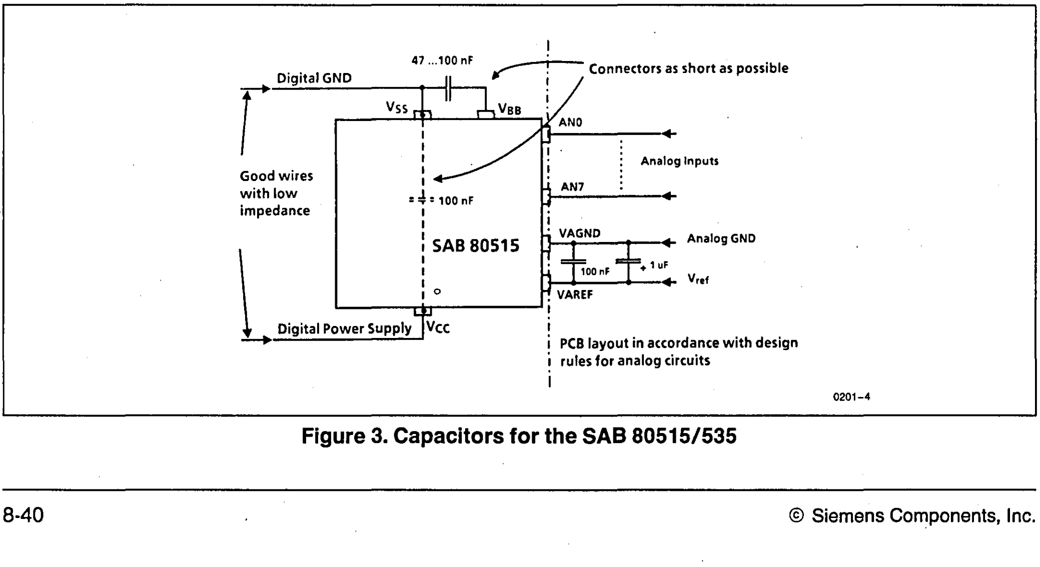
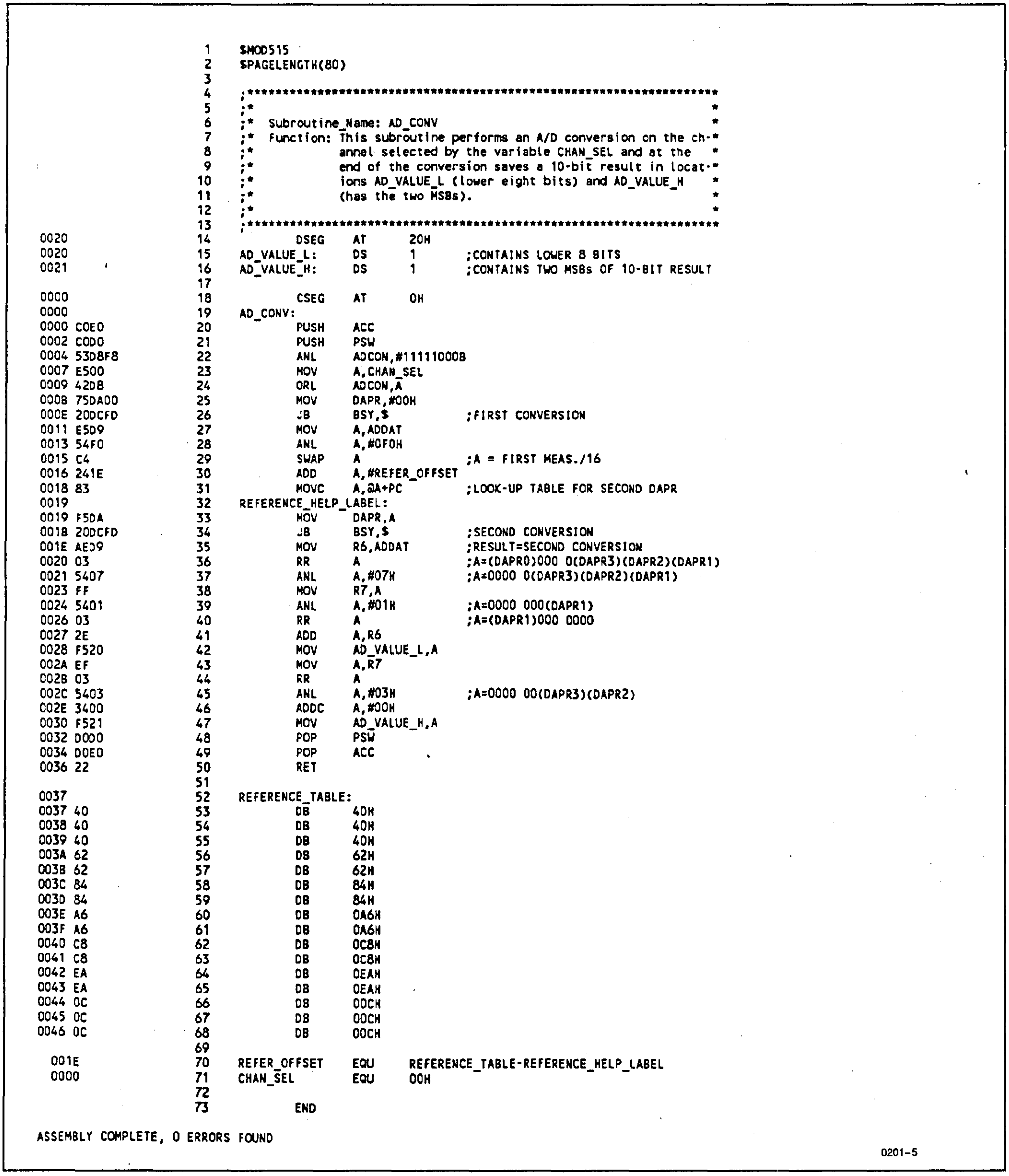


Figure 3. Capacitors for the SAB 80515/535



```
                 $MOD515
                 $PAGELENGTH(80)

                 ;**********************************************************************
                 ;*                                                                    *
                 ;*  Subroutine_Name: AD_CONV                                          *
                 ;*  Function: This subroutine performs an A/D conversion on the ch-*
                 ;*            annel selected by the variable CHAN_SEL and at the      *
                 ;*            end of the conversion saves a 10-bit result in locat-*
                 ;*            ions AD_VALUE_L (lower eight bits) and AD_VALUE_H       *
                 ;*            (has the two MSBs).                                     *
                 ;*                                                                    *
                 ;**********************************************************************
0020                     DSEG    AT      20H
0020             AD_VALUE_L:     DS      1       ;CONTAINS LOWER 8 BITS
0021             AD_VALUE_H:     DS      1       ;CONTAINS TWO MSBs OF 10-BIT RESULT

0000                     CSEG    AT      0H
0000             AD_CONV:
0000 C0E0                PUSH    ACC
0002 C0D0                PUSH    PSW
0004 53D8F8              ANL     ADCON,#11111000B
0007 E500                MOV     A,CHAN_SEL
0009 42D8                ORL     ADCON,A
000B 75DA00              MOV     DAPR,#00H
000E 20DCFD              JB      BSY,$           ;FIRST CONVERSION
0011 E5D9                MOV     A,ADDAT
0013 54F0                ANL     A,#0F0H
0015 C4                  SWAP    A               ;A = FIRST MEAS./16
0016 241E                ADD     A,#REFER_OFFSET
0018 83                  MOVC    A,@A+PC         ;LOOK-UP TABLE FOR SECOND DAPR
0019             REFERENCE_HELP_LABEL:
0019 F5DA                MOV     DAPR,A
001B 20DCFD              JB      BSY,$           ;SECOND CONVERSION
001E AED9                MOV     R6,ADDAT        ;RESULT=SECOND CONVERSION
0020 03                  RR      A               ;A=(DAPR0)000 0(DAPR3)(DAPR2)(DAPR1)
0021 5407                ANL     A,#07H          ;A=0000 0(DAPR3)(DAPR2)(DAPR1)
0023 FF                  MOV     R7,A
0024 5401                ANL     A,#01H          ;A=0000 000(DAPR1)
0026 03                  RR      A               ;A=(DAPR1)000 0000
0027 2E                  ADD     A,R6
0028 F520                MOV     AD_VALUE_L,A
002A EF                  MOV     A,R7
002B 03                  RR      A
002C 5403                ANL     A,#03H          ;A=0000 00(DAPR3)(DAPR2)
002E 3400                ADDC    A,#00H
0030 F521                MOV     AD_VALUE_H,A
0032 D0D0                POP     PSW
0034 D0E0                POP     ACC
0036 22                  RET

0037             REFERENCE_TABLE:
0037 40                  DB      40H
0038 40                  DB      40H
0039 40                  DB      40H
003A 62                  DB      62H
003B 62                  DB      62H
003C 84                  DB      84H
003D 84                  DB      84H
003E A6                  DB      0A6H
003F A6                  DB      0A6H
0040 C8                  DB      0C8H
0041 C8                  DB      0C8H
0042 EA                  DB      0EAH
0043 EA                  DB      0EAH
0044 0C                  DB      00CH
0045 0C                  DB      00CH
0046 0C                  DB      00CH

  001E           REFER_OFFSET    EQU     REFERENCE_TABLE-REFERENCE_HELP_LABEL
  0000           CHAN_SEL        EQU     00H


                         END

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

0201-5

SIEMENS

November 1988

Bidirectional, Speed Regulated Moving Message Display by Using the Timer 2 & 8-Bit A/D Converter of the SAB 80515/80535

SAB 80515/80535

Application Note

This application note introduces the user to one of the features of Timer 2 and A/D converter of the SAB 80515/535. Included in this application note is a description of both the software and hardware implementations of the SAB 80515/535 to use its Timer 2 and 8-bit A/D converter for the bidirectional, speed regulated moving message display. The program listing demonstrates how the Timer 2 and the 8-bit A/D converter of the SAB 80515/535 can be combined to generate time delays controlled by analog levels. The hardware circuitry shows an interface of the SAB 80515/535 with a simulated analog input, a 2 kbyte EPROM, and intelligent display chips of Siemens used in memory mapped I/O scheme.

The SAB 80515/535 microcontroller with on-chip A/D converter and a 16-bit Timer (Timer 2) with reload capability offers a solution which can be applied to a wide range of industrial applications. These applications vary from analog controlled digital delays to controlled frequency converters for pulse width modulation.

In the present application example, the above features of the SAB 80515/535 are used in conjunction to generate the software delays. The software delay results in by varying the voltage level of the analog signal applied to the A/D converter of the SAB 80515/535.

A/D Converter

The SAB 80515/535 provides an 8-bit A/D converter with eight multiplexed analog input channels on-chip. In addition, the A/D converter has a sample and hold circuit and offers the feature of software-programmable reference voltages. For the conversion, the method of successive approximation with a capacitor network is used.

Figure 1 shows a block diagram of the A/D converter. There are three user-accessible special function registers:

— ADCON (A/D converter control register)

— ADDAT (A/D converter data register)

— DAPR (D/A converter program register) for the programmable reference voltages.

Special function register ADCON is used to select one of the eight analog input channels to be converted, to specify a single or continuous conversion, and to check the status bit BSY which signals whether a conversion is in progress or not.

The special function register ADDAT holds the converted digital 8-bit data result. The data remains in ADDAT until it is overwritten by the next converted data. The new converted value will appear in ADDAT in the 15th machine cycle after a conversion has been started. ADDAT can be read and written to under software control. If the A/D converter of the SAB 80515/535 is not used, register ADDAT can be used as an additional general-purpose register.

The special function register DAPR is provided for programming the internal reference voltages IVAREF and IVAGND. In the present application DAPR holds a value of 00H. For this value of DAPR, IVAREF and IVAGND are same as VAREF and VAGND respectively.

A/D Conversion

A conversion is started by writing to the special function register DAPR. A "Write-to-DAPR" will start a new conversion even if a conversion is currently in progress. The conversion begins with the next machine cycle. The busy flag BSY will be set in the same machine cycle as the "write-to-DAPR" operation occurs. If the value written to DAPR is 00H, meaning that no adjustment of the internal reference voltages is desired, the conversion needs 15 machine cycles to be completed. Thus, the conversion time is 15 μs for 12 MHz oscillator frequency.

After a conversion has been started by writing into the special function register DAPR, the analog voltage at the selected input channel is sampled for 5 machine cycles (5 μs at 12 MHz oscillator frequency), which will then be held at the sampled level for the rest of the conversion time. The external analog source must be strong enough to source the current in order to load the sample & hold capacitance, being 25 pF, within those 5 machine cycles.

Conversion of the sampled analog voltage takes place between the 6th and 15th machine cycle after sampling has been completed. In the 15th machine cycle the converted result is moved to ADDAT.

Timer 2

The SAB 80515 has three 16-bit Timer/Counters: Timer 0, Timer 1 and Timer 2. These Timers can be configured to operate either as timers or event counters. Timer 2 is the time base of the programmable Timer/Counter Register Array (PTRA) unit. In addition to the operational modes "Timer" or "counter", Timer 2, being the time base for the PTRA unit, provides the features of:

— 16-bit reload

— 16-bit compare

— 16-bit capture

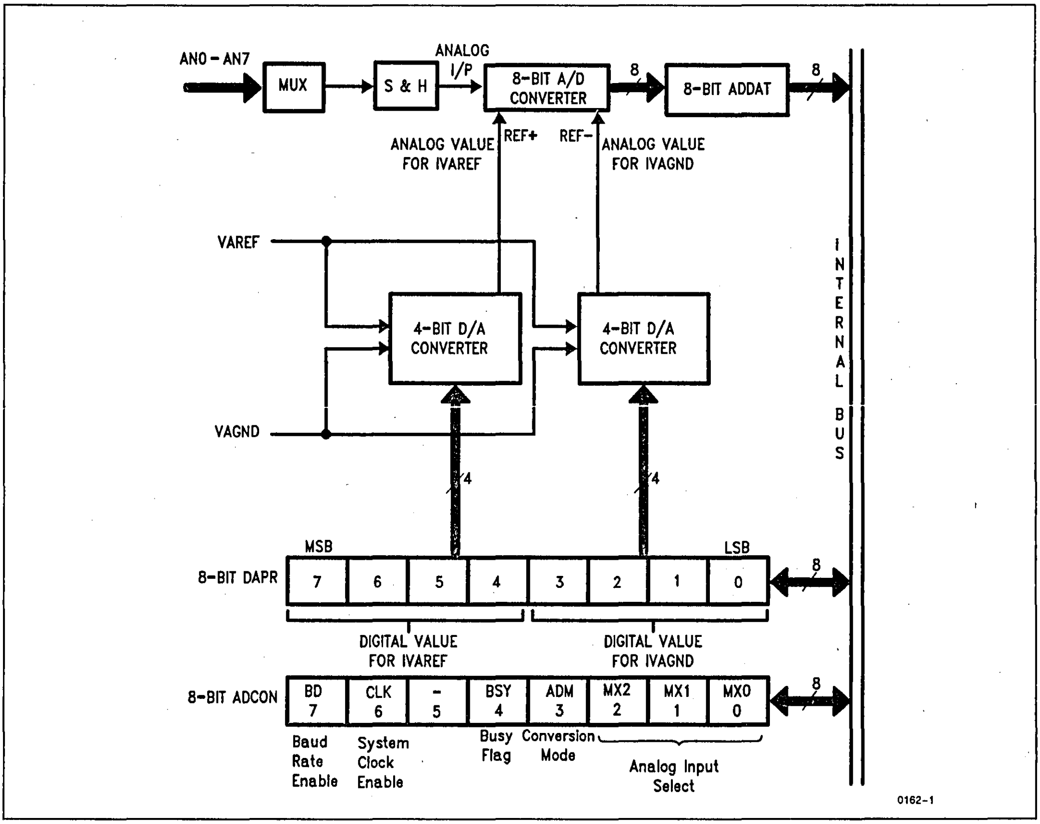


Figure 1. Block Diagram of A/D Converter

The reload mode of Timer 2 is used in this application to generate software delays. For explanation of the other modes please refer to the users' manual.

Reload

The reload mode for Timer 2 is selected by bits T2R0 and T2R1 in special function register T2CON as illustrated in Table 1. In mode 0, when Timer 2 rolls over from all 1s to all 0s, it not only sets TF2 but also causes the Timer 2 registers to be loaded with the 16-bit value in the CRC (compare/reload/capture) register which is preset by software. The reload will happen in the same machine cycle in which TF2 is set, thus overwriting the count value 0000H.

Table 1. Timer 2 Reload Mode Selection

T2R1	T2R0	Mode
0	X	Reload Disabled
1	0	Mode 0: Auto-Reload upon Timer 2 Overflow (TF2)
1	1	Mode 1: Reload upon Falling Edge at Pin T2EX/P1.5

Figure 2 below shows a functional diagram of the Timer 2 reload modes.

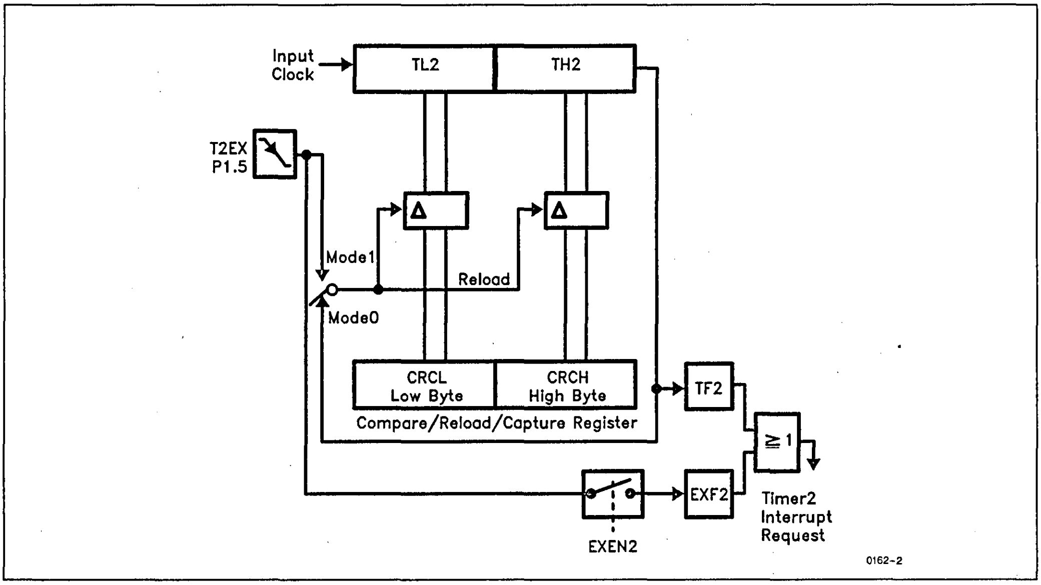


Figure 2. Timer 2 in Reload Mode

PD2435

PD2435 is a CMOS
4-character
5 x 7 dot matrix
alphanumeric
programmable display with ROM to decode 96 ASCII alphanumeric characters and enough RAM to store the display's complete four digit ASCII message with software programmable attributes. The CMOS IC incorporates special interface control circuitry to allow the user to control the module as a fully supported microprocessor peripheral.

Microprocessor Interface

The interface to the microprocessor is through the address lines (A0–A2), the data bus (D0–D7), two chip select lines ($\overline{\text{CE0}}$, CE1), and ($\overline{\text{RD}}$) and ($\overline{\text{WR}}$) lines. The $\overline{\text{CE0}}$ should be held low when executing a read or write operation. The read and write lines are both active low. A valid write will enable the data as input lines.

Programming the PD2435

There are five registers within the PD2435. Four of the registers are used to hold the ASCII code of the four display characters. The fifth register is the Control Word, which is used to blink, blank, clear or dim the entire display to change the presentation (attributes) of individual characters.

The PD2435 block diagram shows the major blocks and internal registers.

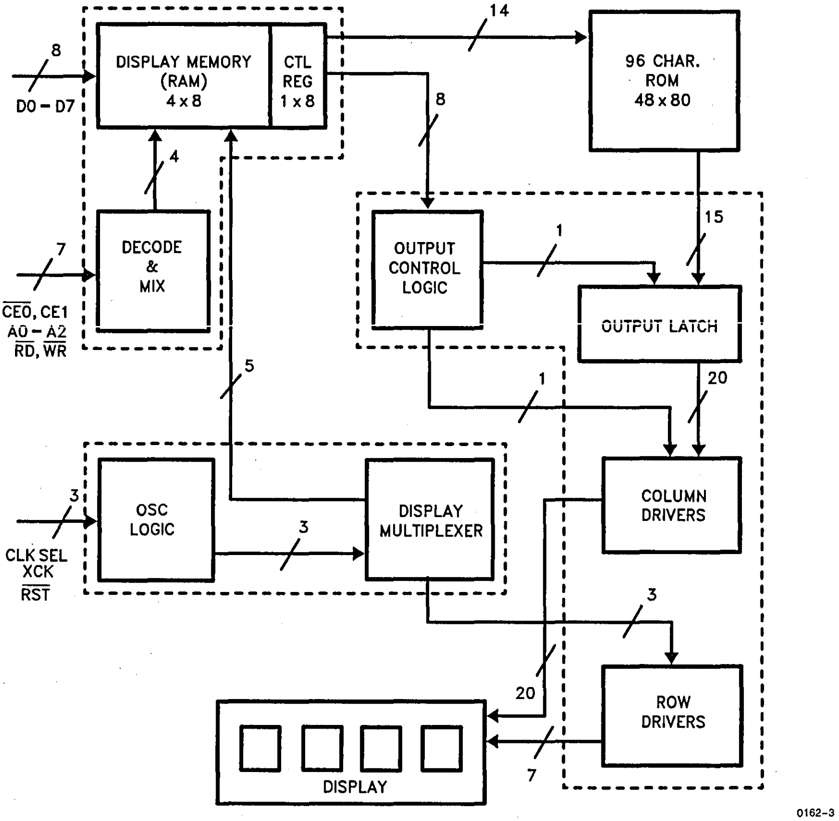


Figure 3. PD2435 Block Diagram

Application

The speed regulated moving message display is an example where a digitized value of the controlling analog signal is used to compute a reload value for the Timer 2. The Timer 2 is operated in mode 0 where this reload value becomes a starting point for the Timer to count up. On overflow the Timer automatically takes the restart value for counting from reload register CRC. While the Timer is counting up, a new reload value is computed using the present A/D value.

Hardware

The circuit used in this application offers the advantage in requiring a minimum of components. The single chip microcomputer SAB 80535 operates in conjunction with four alphanumeric programmable display chips PD 2435 to form a 16-digit long display.

The ASCII-coded data is transferred from the SAB 80535 to the display ICs via the data port P0 and using the control signal WR (P3.6) of the SAB 80535. The address pins from the ports P0 and P2 of the SAB 80535 are used to address the EPROM as well as the display chips in a memory-mapped I/O scheme. The display chips are addressed as memory locations with the following addresses.

Display Chip	Control Register Address	Digits Address
1	1000H	1004H–1007H
2	2000H	2004H–2007H
3	4000H	4004H–4007H
4	8000H	8004H–8007H

A push button is interfaced to port P3.2 of the SAB 80535 to provide an external interrupt to the microcontroller.

Firmware Description

Besides controlling speed of the moving message, there is a provision to interrupt the moving message to roll it backwards until the beginning of the message. The microcontroller reads the code and the message to display from an EPROM 2716A interfaced to the ports P0 and P2 of the SAB 80535. A virtual image of the message is created in the internal RAM of the SAB 80535. Four display chips PD2435 are interfaced to the SAB 80535 in a memory-mapped scheme and can be addressed as external memory to the SAB 80535. The virtual image of the message in internal RAM of the SAB 80535 is used to manipulate data to be displayed on the display chips. The internal RAM used for the display can be viewed as an area divided into two portions:

1. For active display
2. As a data buffer

The active display area is the replica of the data being displayed on the display chips. In this case the 16-digit display would need 16 RAM locations which correspond to 16 digits currently being displayed. The data buffer contains the rest of the message which is not being displayed. The message is shifted character by character in the RAM area When the message cn the display moves from right to left, the RAM buffer acts in "First In First Out" mode and when the message on the display moves from left to right, the data to the display from the microcontroller RAM buffer is supplied in the "Last In First Out" scheme.

Between display of every character there is a software delay which depends upon the level of the analog signal supplied to the AN0 pin of the SAB 80535. The external interrupt 0 (at port P3.2) is used to interrupt the microcontroller to inform that the message needs to be scrolled backwards. On getting this interrupt the software sets the flag bit 0 which remains set until the message is scrolled back to the beginning of the message.

List of Components

Name	Number
SAB 80535	1
2716A	1
PD2435	4
12 MHz Crystal	1
74LS373	1
22 pF Capcitors	2
100 nF Capacitor	1
4.7 μf Capacitor	1
1k Resistor	1
10k Pot	1

Reference Material for ICs

1. SAB 80515/80535 User's Manual.
2. PD2435 Data-Sheet or Optoelectronic Data Book (1987/88).

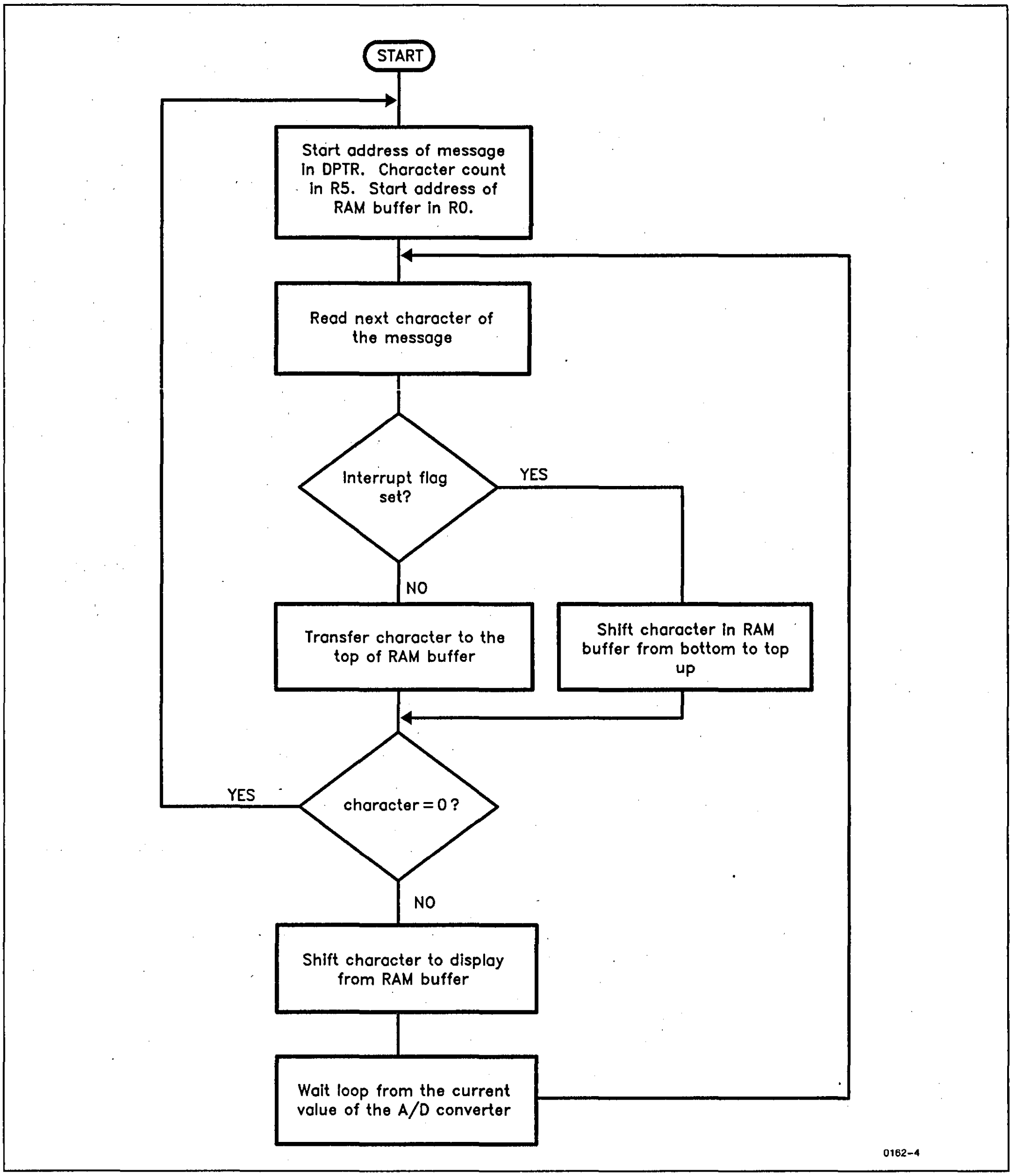
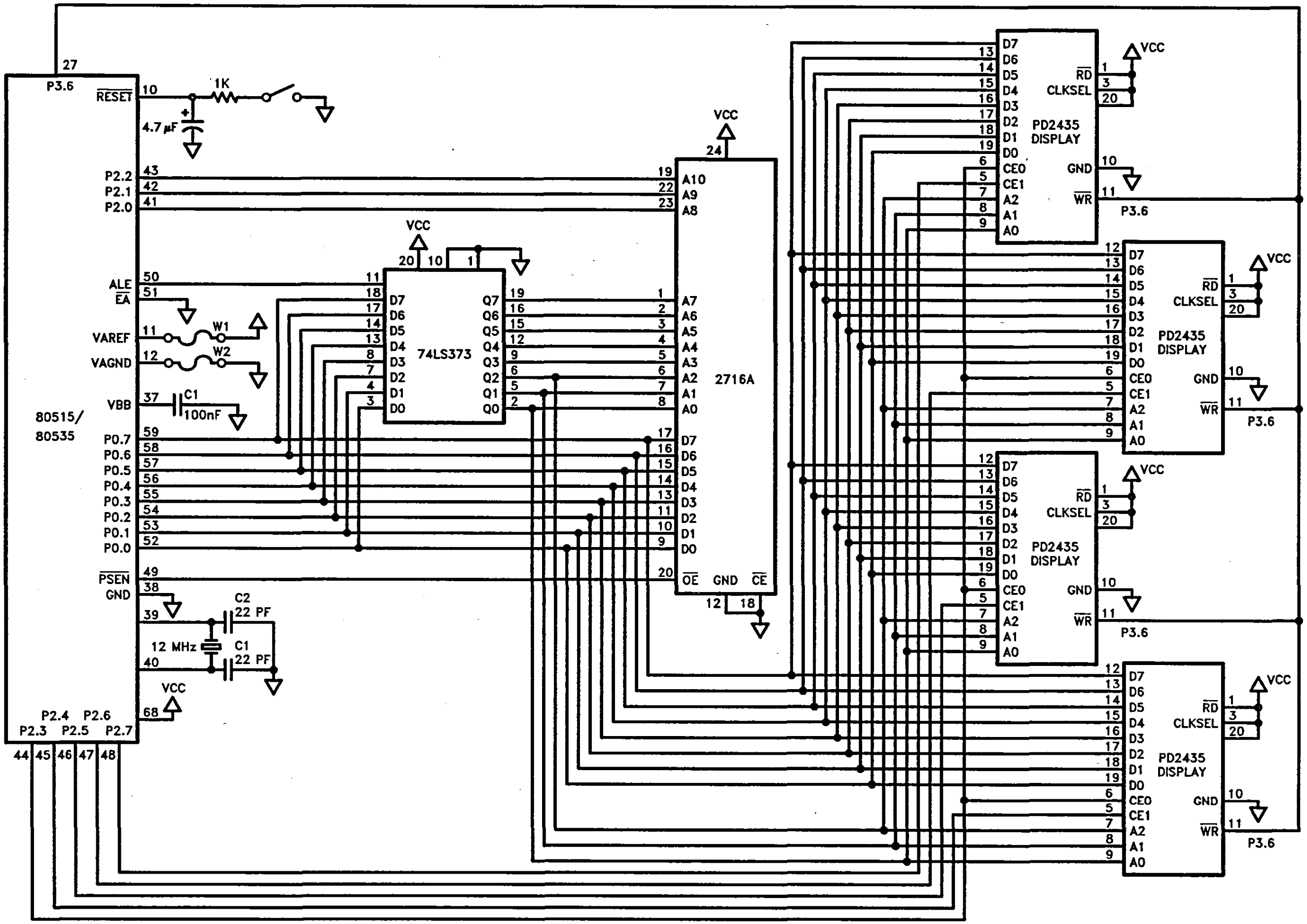

START
Start address of message in DPTR. Character count in R5. Start address of RAM buffer in R0.
Read next character of the message
Interrupt flag set?
YES
NO
Transfer character to the top of RAM buffer
Shift character in RAM buffer from bottom to top up
character = 0 ?
YES
NO
Shift character to display from RAM buffer
Wait loop from the current value of the A/D converter
0162-4

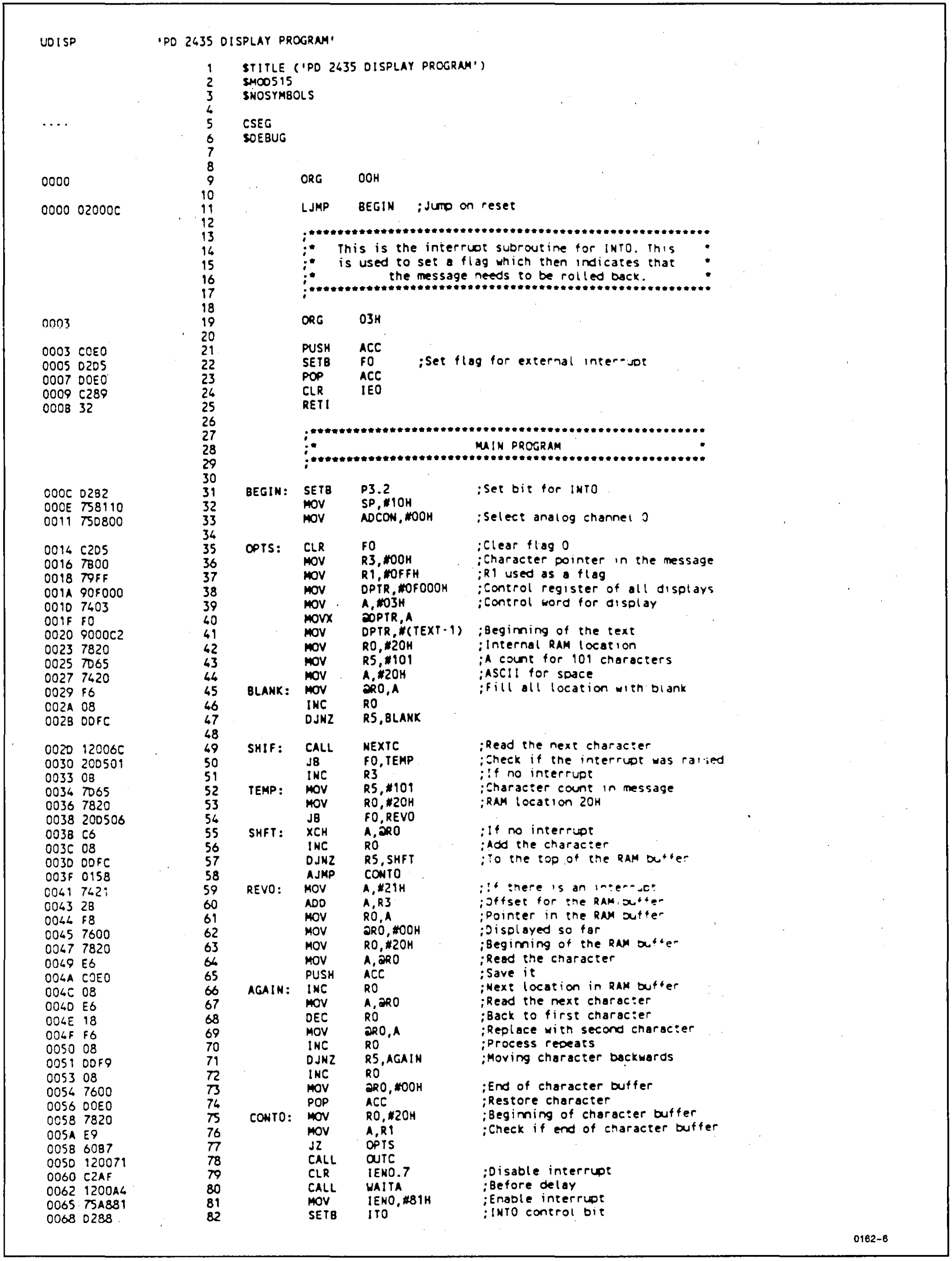
Figure 4. Program Flow-Chart

Interface Circuit



0162-5

Program Listing



```
UDISP            'PD 2435 DISPLAY PROGRAM'

                            $TITLE ('PD 2435 DISPLAY PROGRAM')
                            $MOD515
                            $NOSYMBOLS

....                        CSEG
                            $DEBUG


0000                                  ORG     00H

0000 02000C                           LJMP    BEGIN   ;Jump on reset

                                      ;****************************************************
                                      ;
                                      ;*   This is the interrupt subroutine for INTO. This    *
                                      ;*   is used to set a flag which then indicates that    *
                                      ;*           the message needs to be rolled back.        *
                                      ;****************************************************
                                      ;

0003                                  ORG     03H

0003 C0E0                             PUSH    ACC
0005 D2D5                             SETB    F0      ;Set flag for external interrupt
0007 D0E0                             POP     ACC
0009 C289                             CLR     IE0
000B 32                               RETI

                                      ;****************************************************
                                      ;
                                      ;*                        MAIN PROGRAM                   *
                                      ;****************************************************
                                      ;

000C D282                   BEGIN:    SETB    P3.2              ;Set bit for INTO
000E 758110                           MOV     SP,#10H
0011 75D800                           MOV     ADCON,#00H        ;Select analog channel 0

0014 C2D5                   OPTS:     CLR     F0                ;Clear flag 0
0016 7B00                             MOV     R3,#00H           ;Character pointer in the message
0018 79FF                             MOV     R1,#0FFH          ;R1 used as a flag
001A 90F000                           MOV     DPTR,#0F000H      ;Control register of all displays
001D 7403                             MOV     A,#03H            ;Control word for display
001F F0                               MOVX    @DPTR,A
0020 9000C2                           MOV     DPTR,#(TEXT-1)    ;Beginning of the text
0023 7820                             MOV     R0,#20H           ;Internal RAM location
0025 7D65                             MOV     R5,#101           ;A count for 101 characters
0027 7420                             MOV     A,#20H            ;ASCII for space
0029 F6                     BLANK:    MOV     @R0,A             ;Fill all location with blank
002A 08                               INC     R0
002B DDFC                             DJNZ    R5,BLANK

002D 12006C                 SHIF:     CALL    NEXTC             ;Read the next character
0030 20D501                           JB      F0,TEMP           ;Check if the interrupt was raised
0033 0B                               INC     R3                ;If no interrupt
0034 7D65                   TEMP:     MOV     R5,#101           ;Character count in message
0036 7820                             MOV     R0,#20H           ;RAM location 20H
0038 20D506                           JB      F0,REVO
003B C6                     SHFT:     XCH     A,@R0             ;If no interrupt
003C 08                               INC     R0                ;Add the character
003D DDFC                             DJNZ    R5,SHFT           ;To the top of the RAM buffer
003F 0158                             AJMP    CONTO
0041 7421                   REVO:     MOV     A,#21H            ;If there is an interrupt
0043 2B                               ADD     A,R3              ;Offset for the RAM buffer
0044 F8                               MOV     R0,A              ;Pointer in the RAM buffer
0045 7600                             MOV     @R0,#00H          ;Displayed so far
0047 7820                             MOV     R0,#20H           ;Beginning of the RAM buffer
0049 E6                               MOV     A,@R0             ;Read the character
004A C0E0                             PUSH    ACC               ;Save it
004C 08                     AGAIN:    INC     R0                ;Next location in RAM buffer
004D E6                               MOV     A,@R0             ;Read the next character
004E 18                               DEC     R0                ;Back to first character
004F F6                               MOV     @R0,A             ;Replace with second character
0050 08                               INC     R0                ;Process repeats
0051 DDF9                             DJNZ    R5,AGAIN          ;Moving character backwards
0053 08                               INC     R0
0054 7600                             MOV     @R0,#00H          ;End of character buffer
0056 D0E0                             POP     ACC               ;Restore character
0058 7820                   CONTO:    MOV     R0,#20H           ;Beginning of character buffer
005A E9                               MOV     A,R1              ;Check if end of character buffer
005B 60B7                             JZ      OPTS
005D 120071                           CALL    OUTC
0060 C2AF                             CLR     IEN0.7            ;Disable interrupt
0062 1200A4                           CALL    WAITA             ;Before delay
0065 75A881                           MOV     IEN0,#81H         ;Enable interrupt
0068 D288                             SETB    IT0               ;INTO control bit
```

0162-6

Program Listing (Continued)

```
006A 012D                    AJMP    SHIF

                             ;*******************************************************
                             ;* The routine moves a character of the message to ACC. *
                             ;*******************************************************

006C A3              NEXTC:  INC     DPTR
006D 7400                    MOV     A,#0
006F 93                      MOVC    A,@A+DPTR              ;Move the character to Acc.
0070 22                      RET

                             ;*******************************************************
                             ;*   This routine displays and moves a character over   *
                             ;*    the four digits of the PD2435 and then repeats    *
                             ;*          for the next display chip and so on.        *
                             ;*******************************************************

0071 C0E0            OUTC:   PUSH    ACC
0073 C082                    PUSH    DPL
0075 C083                    PUSH    DPH
0077 7A04                    MOV     R2,#4          ;For four digits(0 to 3) in a chip
0079 901004                  MOV     DPTR,#1004H    ;Digit 0 in first display chip
007C 120098                  CALL    OUTCO
007F 902004                  MOV     DPTR,#2004H    ;Digit 0 in second display chip
0082 120098                  CALL    OUTCO
0085 904004                  MOV     DPTR,#4004H    ;Digit 0 in third display chip
0088 120098                  CALL    OUTCO
008B 908004                  MOV     DPTR,#8004H    ;Digit 0 in fourth display chip
008E 120098                  CALL    OUTCO
0091 D083                    POP     DPH
0093 D082                    POP     DPL
0095 D0E0                    POP     ACC
0097 22                      RET

                             ;*******************************************************
                             ;* This is a nested subroutine. It moves a nonzero hex  *
                             ;*  value (ASCII) from left in right of the four digit  *
                             ;*                        display.                      *
                             ;*******************************************************

0098 E6              OUTCO:  MOV     A,@R0
0099 6007                    JZ      FIN
009B F0                      MOVX    @DPTR,A
009C 08                      INC     R0
009D A3                      INC     DPTR
009E DAF8                    DJNZ    R2,OUTCO
00A0 7A04                    MOV     R2,#4
00A2 F9              FIN:    MOV     R1,A
00A3 22                      RET

                             ;*******************************************************
                             ;*  This subroutine generates the software delay. The  *
                             ;*  delay is generated by the timer 2. The start count  *
                             ;*  of the timer 2 is computed from the present value   *
                             ;*                 of the A/D converter.                *
                             ;*******************************************************

00A4 7E03            WAITA:  MOV     R6,#03H
00A6 7D10            WAITB:  MOV     R5,#10H
00A8 75DA00          WAITC:  MOV     DAPR,#00H
00AB E5D9                    MOV     A,ADDAT
00AD 75F0FF                  MOV     B,#255         ;For computing reload value
00B0 A4                      MUL     AB             ;Reload value is computed
00B1 F5CA                    MOV     CRCL,A         ;Load the reload value low
00B3 85F0CB                  MOV     CRCH,B         ;Load the reload value high
00B6 75C811                  MOV     T2CON,#11H
00B9 10C602          WAITD:  JBC     TF2,WAITE
00BC 01B9                    AJMP    WAITD
00BE DDE8            WAITE:  DJNZ    R5,WAITC
00C0 DEE4                    DJNZ    R6,WAITB
00C2 22                      RET

                             ;*******************************************************
                             ;*                      MESSAGE                        *
                             ;*******************************************************

00C3 20202020        TEXT:   DB      '                '
00C7 20202020
00CB 20202020
00CF 20202020
00D3 5349454D                DB      'SIEMENS MICROCONTROLLER SAB 80515/535'
00D7 454E5320
00DB 4D494352
00DF 4F434F4E
```

0162-7

Program Listing (Continued)

```
00E3 54524F4C
00E7 4C455220
00EB 53414220
00EF 38303531
00F3 352F3533
00F7 35
00F8 20202020      161        DB    '          SAB 80515/535          ',0
00FC 20202020
0100 20202020
0104 53414220
0108 38303531
010C 352F3533
0110 35202020
0114 20202020
0118 20202020
011C 20202020
0120 00
                   162        END

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

0162-8

SIEMENS

November 1988

Heating and Air Conditioning Control in Cars with the Microcontroller SAB 80515/80535

SAB 80515/80535

Application Note

The heating and air conditioning unit in the car should provide the driver with conditions of comfortable temperature, fresh air flow, defogged and defrosted windows, low energy consumption and easy operation.

The currently mass-produced systems meet these requirements to a limited degree. In response, Siemens began to develop μC-controlled installations and equipped several test vehicles with various systems which proved to be highly satisfactory.

Its performance-oriented processor and flexible on-chip periphery (e.g. analog-to-digital converter, timer function, large number of inputs/outputs) make the SAB 80515 especially suitable for this type of application. The majority of the peripheral components are fully utilized in this application.

The temperature in the car reaches its nominal value and is kept constant by means of a two-stage mixing value control. A rise in outside temperature automatically activates the compressor of the air conditioning unit. The air entering the car is distributed upwards and downwards by an electrically controlled value, depending on the temperature of the air. The optimal speed is also determined by the microcontroller as a function of the various input values. The electronics also control actuators such as air circulation and water valve.

The device is operated by means of several keys. An LED display indicates the nominal or outside temperature. Individual LEDs indicate special conditions which can be selected by the user independent of the automatic functions.

Control Elements and Sensors (Figure 1)

The temperature inside the car is controlled in accordance with a selectable nominal value. The most important actuators for this purpose are the mixing valve and the compressor of the air conditioning unit. The mixing valve determines which part of the air entering the passenger area must pass through the heat exchanger of the heating unit. The valve can be fine-tuned by the microcomputer.

If there is no heating requirement, the water flow to the heat exchanger is stopped by a digitally controlled valve. As a result, the temperature is further reduced in the summer time.

Depending on the output of the system, the air conditioning unit ensures that the nominal value of the temperature inside the car is obtained despite higher outside temperatures. The compressor of the air conditioning unit is enabled/disabled by the microcontroller.

In addition, the electronics influences the distribution of the temperature layers inside the car by a nearly stepless adjustable distribution valve. The valve determines whether the air is to be moved towards the roof or the floor of the car. Through this type of control, the air close to the roof of the car should be at a temperature lower than that close to the floor.

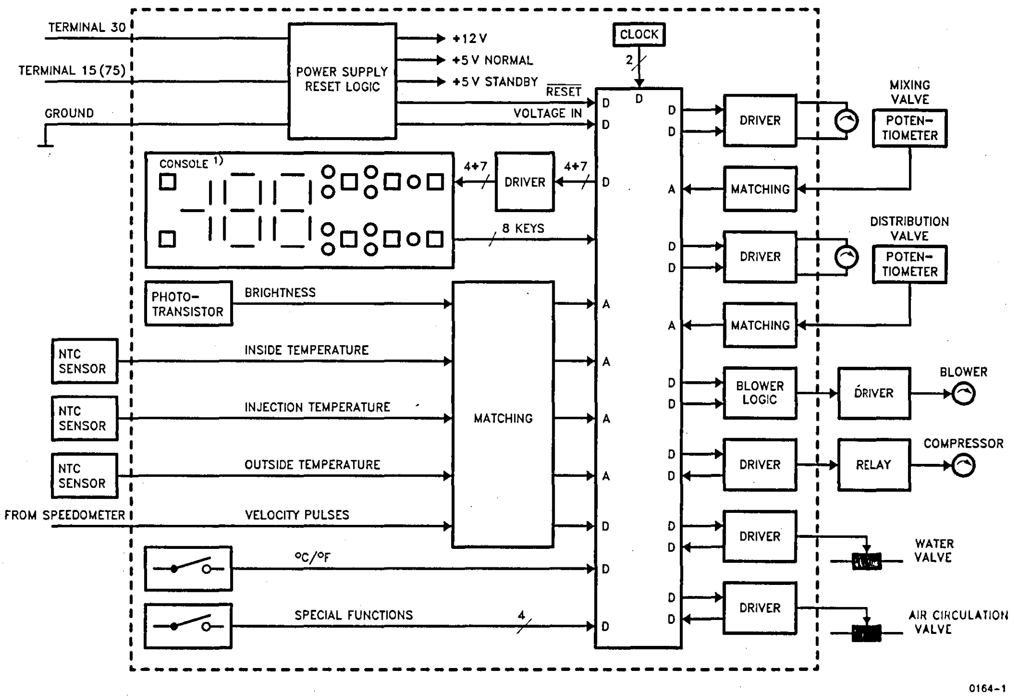
The fresh air flow is also electronically controlled. Depending on the different temperatures and the road speed of the car, the microcontroller computes the optimal speed for the blower, which can be changed almost continuously.

On the basis of the temperature conditions, the processor determines the requirement for fresh air flow or circulation of the air inside the car. The corresponding valve is digitally controlled.

In order to achieve the described functions, the system uses three sensors to measure the temperature inside the car, the temperature of the air entering the car as well as the outside temperature. A speed sensor informs the processor about the car's current road speed.

Operating and Displaying Unit (Figure 2)

A display optionally indicates the nominal or outside temperature. Functions which deviate from standard operations are indicated by LEDs located next to the keys. The brightness level of the display and the LEDs is controlled by the processor in accordance with the ambient light measured by a phototransistor.



D— Digital Signal
A— Analog Signal
1See Figure 2 for Detail

Figure 1. Block Diagram

With the aid of eight keys the following functions can be performed (Figure 2):

S1, S2: Changes in nominal temperature ("+" and "−" key)
Through instantaneous pressure or sustained pressure on the key, the nominal value can be changed in 1°C/1°F steps, that is from 16°C–30°C/60°F–86°F. In addition, the extreme values "LO" and "HI" can be set, and the mixing valve will continue to remain in the minimal (cold) or maximal (hot) position.

By depressing the keys, the following functions allow the user to switch over from normal (automatic) setting to one, two or three fixed values. After a fixed value has been selected, the corresponding LED or a combination of two LEDs lights up.

S3: Distribution key for switching the air distribution to automatic, only upwards, in the center (upwards and downwards, both LEDs light up) or only downwards.

S4: Blower key for switching the blower to automatic, full speed, half speed (both LEDs light up) and "OFF".

S5: Air supply key for switching the air supply to automatic, air circulation, or fresh air.

S6: Compressor key for switching the compressor to automatic, "ON" and "OFF".

S7: Outside temperature key for switching the display to nominal temperature (standard function) or outside temperature. The outside temperature is displayed in 1°C/1°F steps at a range between −40°C/−40°F and +60°C/+140°F.

S8: Defrost key for switching the device from its previous function (standard) to the defrost function.

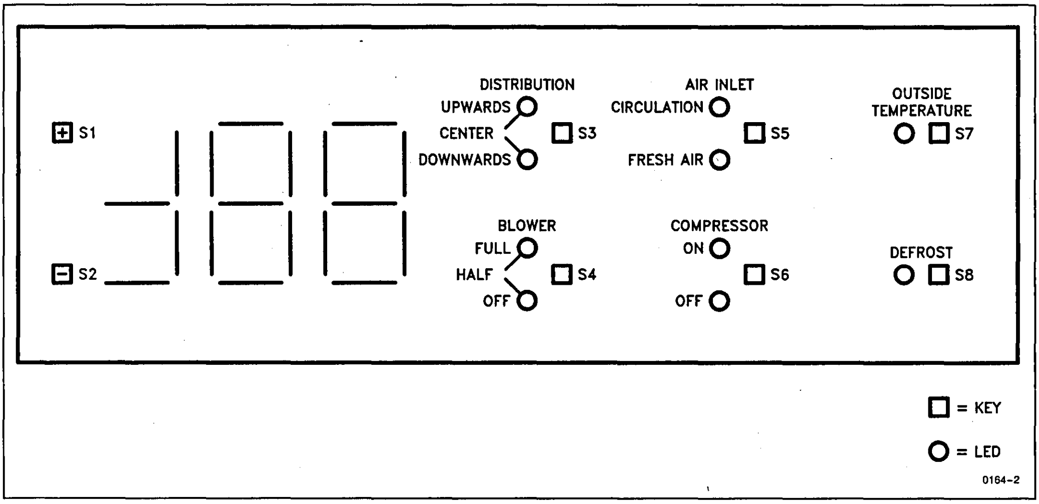


Figure 2. Console

The windows of the car can thus rapidly be defrosted and defogged.

The setting elements take the following positions during the defrost function which has priority over all other settings:

Mixing valve:	Max. Heating
Distributor valve:	Only Upwards
Blower:	Max. Number of Speed
Air supply:	Fresh Air
Compressor:	ON
Water Valve:	ON

As long as the defrost function is in operation, the remaining functions (with the exception of display switch-over for the temperature) cannot be operated. The corresponding LEDs are not driven. After the defrost status is finished, the previous functions apply again.

The nominal temperature as well as set fixed values are saved after the car ignition has been turned off. During initial start-up or after reconnecting the battery, a mean nominal temperature (22°C/71°F) is set and the automatic functions apply.

Major Hardware and Software Functions

Voltage Supply, Reset Logic (Figure 3)

Since various conditions—e.g. nominal temperature—are to be stored after the ignition has been turned off, a continuous 5V supply is required which is supplied directly by the battery (terminal 30). A diode/capacitor combination protects against reversed polarity and extreme voltage peaks. The voltage regulator, which is used, is characterized by a lower power dissipation and continues to operate during low input voltages. The SAB 80515 stores the data; 40 bytes of its internal RAM are saved during standby operation with a typ. supply current of 1 mA.

When the car ignition is turned on, the normal 5V operating voltage as well as a filtered 12V voltage are available for supplying the drivers. The criterion for the connection of these voltages is the status of terminal 15. Preferable terminal 75 should be used if included in the device, since it will remain disabled while the car is started.

When the ignition is turned off, the processor receives a signal via P07 prior to the drop in voltage of the standard 5V supply. Subsequently, the processor will wait for the reset signal which immediately precedes the voltage switch off. After the ignition has been turned on again, $\overline{\text{RES}}$ continues to be in "L" to reset the SAB 80515. This time period required for reset is ensured by an RC network in combination with diodes and Schmitt triggers.

Clock Supply

The SAB 80515 oscillator resonates at a frequency of 6 MHz by means of a ceramic resonator. The result is an instruction cycle time of 2 μs.

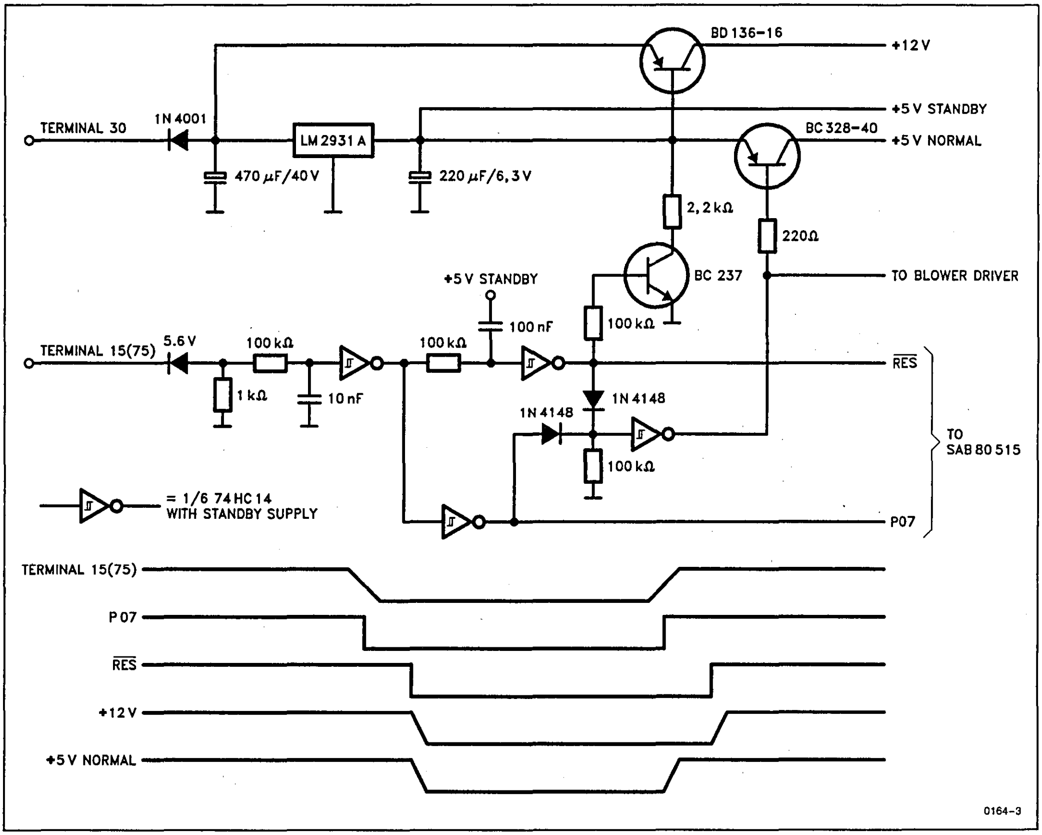


Figure 3. Voltage Supply, Reset Logic

Acquisition and Preprocessing of Input Value Parameters

— Temperature and brightness (Figure 4)

The inside air, fresh air, and outside air temperatures are measured with an NTC sensor S 861 or S 867 (encapsulated). The most suitable locations for installations of the sensors in particular car types must be determined experimentally. The values of the pull-up resistors between the signal line and the analog reference voltage have been selected to ensure optimal accuracy within the required temperature range. Short-term interference pulses are filtered out by an RC network. The phototransistor BP 103B on the face of the device generates a voltage across a resistor in accordance with the ambient light.

The SAB 80515 reads these analog units at regular intervals via the multiplexer and the analog-to-digital converter located on the chip. Since the sensors are connected to the analog references, the result is not affected by the absolute value of these voltages. After the conversion low-frequency fluctuations are suppressed via software averaging. On the basis of tables and linear interpolation, the SAB 80515 computes the values required, i.e. the temperature and the value for control of the display's brightness.

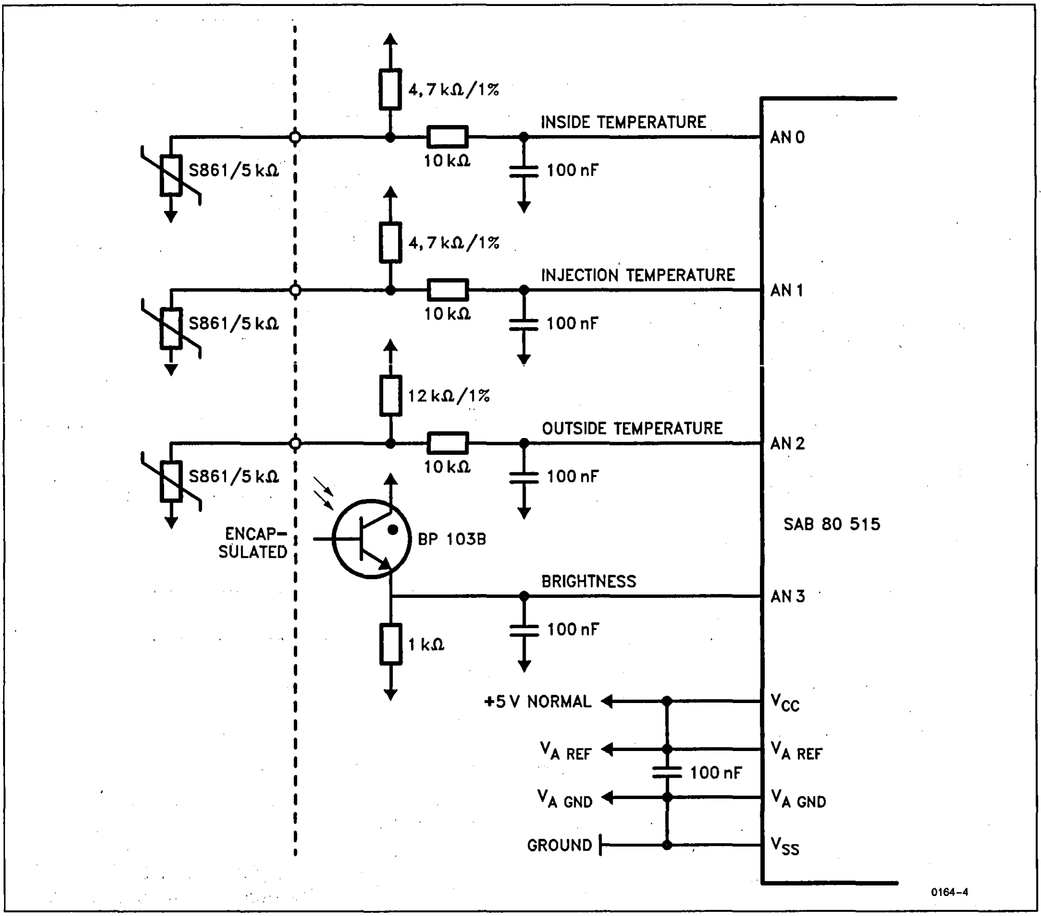


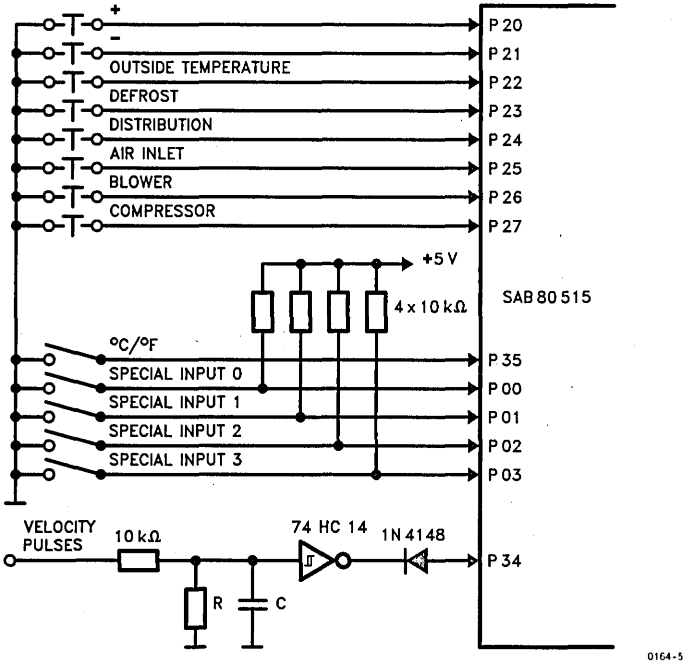
Figure 4. Temperature and Brightness

—Speed (Figure 5)

The speed is derived from the generator for electronic speedometers included in most cars An RC wiring (perhaps with voltage division) and a Schmitt trigger filter out interference in the sensor pulses and adjust the voltage amplitude. With timer 1, the processor counts the pulses received within a defined timer period.

—Keys/Switch (Figure 5)

Since these components are located inside the device, they can be protected against bouncing using software. Because of its many I/O ports, the SAB 80515 can read in information directly. A matrix with decoupling diodes is not required. Also, pull-up resistors are not required at the inputs of the SAB 80515—with the exception of PO. With the hidden °F switch, the unit for displaying the nominal and outside temperatures can be selected. The special inputs are used for activating special test functions (see section on "testing and optimization support").



R: Value according to velocity pulses
C: Value according to max. frequency of the velocity pulses

Figure 5. Inputs for Keys, Switches and Speed

—Display (Figure 6)

The display comprises a three-digit 7-segment LED display (configured from HG 1107 elements). The foremost digit utilizes only four segments. There are also 10 single LEDs LG 3160 for indicating special conditions. The processor drives the display in a four-step multiplex method. For selecting the digit, the outputs P54–P57 go to HIGH in successive order. During this time the information for the segments is present at outputs P40–P46. Four Darlington transistors BC 517 are used as actuator drivers, and seven transistors BC 237 as segment drivers. The voltage source is the 12V supply.

The multiplexed display and the brightness are controlled by timer 2 of the SAB 80515. The component is used as timer in the auto-reload with the oscillator frequency divided by 12. During each overflow the timer is automatically loaded with the content of the CRC register—in this case FF00. This leads to a time interval of 256 × 2 = 512 μs between two overflows.

The interval determines the length of a multiplex clock. The interrupt triggered by each overflow results in the output of the new segment information at port 4. The allocated multiplex location is released through port 5.

The display brightness level is determined by the processor on the basis of the ambient light measured by the phototransistor, and a table stored in the ROM. The compare function of timer 2 sets the brightness level: as soon as the timer reaches the value of the compare register, an additional interrupt is triggered. In the associated routine, the processor sets the actuator outputs P54–P57 to "L". This creates an off-period until the timer overflows, the duration of which depends on the content of the compare register. This register can be loaded at any time with the value determined from the ambient light.

—Regulating the inside temperature with the mixing valve (Figure 7)

The temperature inside the car depends largely on the position of the mixing valve, which the SAB 80515 computes by means of the so-called cascade control.

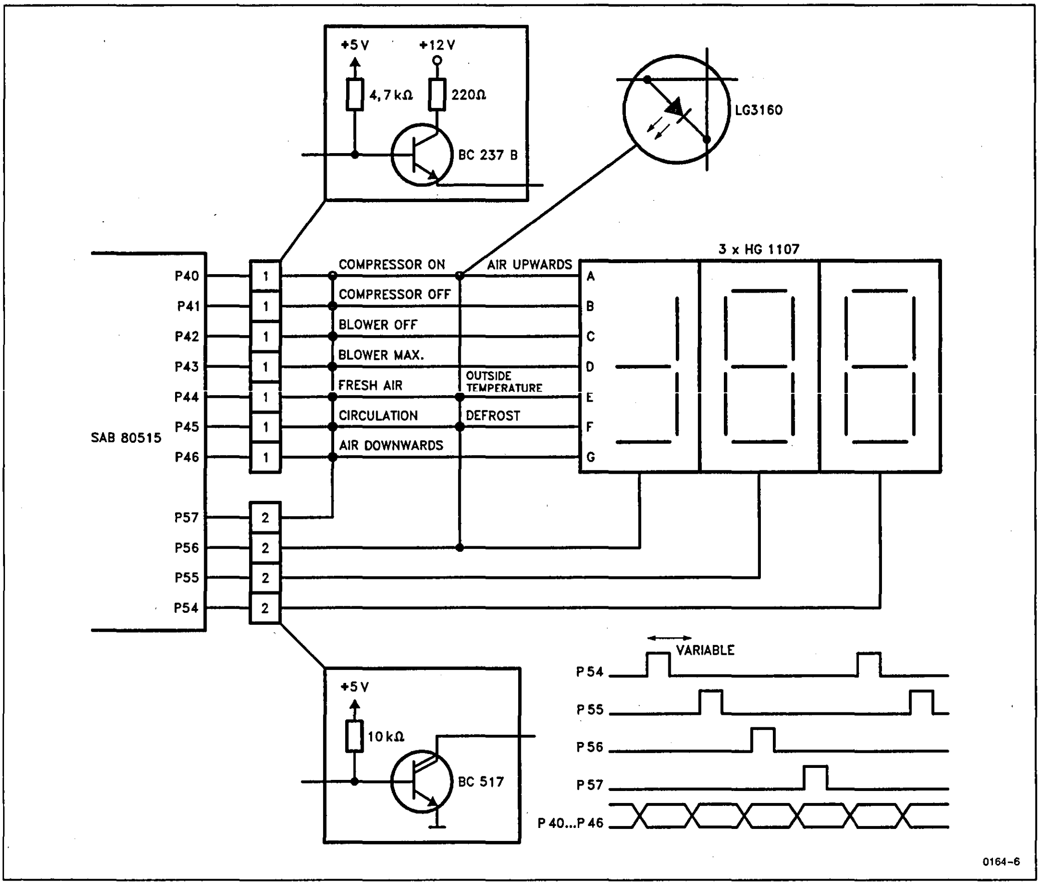


Figure 6. Display

The deviation of the temperature inside the car $T_{\mathrm{I\ act}}$ from the set nominal value $T_{\mathrm{I\ nom}}$ determines in an outer control circuit the nominal value for the injection temperature $T_{\mathrm{In\ nom.}}$ Through the inner, faster control circuit the mixing valve is adjusted so that the injection temperature actually reaches the value $T_{\mathrm{In\ nom.}}$ When compared with a less complex control of the mixing valve by means of the difference between the nominal and actual value of the temperature inside the car, this two-stage system results in improved stability. In addition, interference which influences the injection temperature can be quickly rectified (e.g. changes in motor or outside temperature, activation/deactivation of compressor, switchover from/to fresh air/air circulation). Also, the min. and max. ratings for the injection temperature can easily be established providing the necessary comfort for the passengers. With properly set parameters the time characteristics as compared to a simple control are equally satisfactory.

The nominal values for the injection temperature and the mixing value position are computed according to a digital PID (proportional, integral, and differential) algorithm. Although the variety of parameters which can be set for both controls permit a wide range of adjustments, the expenditure is considerable. Therefore, to facilitate the test and optimization phase, all parameters can be displayed and changed during travel by depressing the respective key (see section 3.10 "testing and optimization support"). For example, by setting the differential portion to zero, a PI characteristic can be obtained.

When "HI" or "LO" is displayed in place of the nominal temperature, the control algorithm is switched off and the mixing valve is positioned at maximum or minimum heating output.

The control algorithm is also inactive in the defrost status which calls for max. heat output. When switching over to normal operation, the valve returns to its former position.

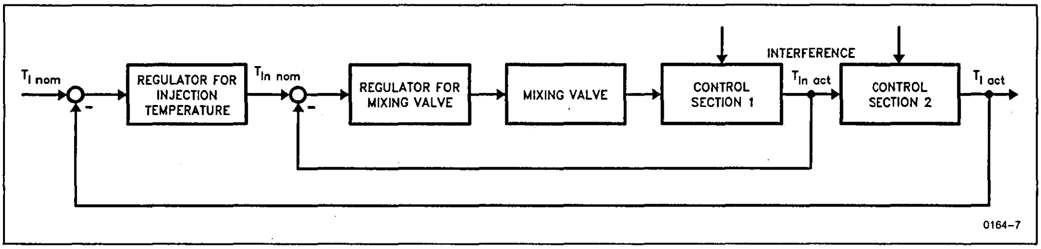


Figure 7. Block Diagram of Temperature Regulation

Establishing the Nominal Value for the Distribution Valve

As can be seen in Figure 8, the position of the distribution valve normally depends on the difference between the injection temperature and the temperature inside the car. Cooler air is usually injected upwards while heating air is injected downwards towards to floor of the car. The effective nominal value is set by the SAB 80515 with reference to the actual end positions of the valve (see section 3.5).

During the special functions "upward air distribution" and "defrost", the air is blown only upwards or during "downward air distribution" only downwards at floor level. When "center air distribution" has been selected, half of the air volume is blown upwards and half downwards.

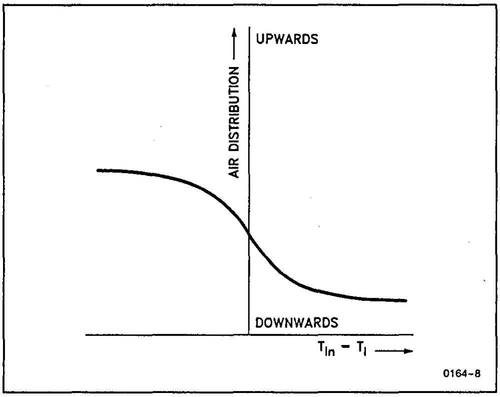


Figure 8. Position of Distribution Valve versus Injection and Inside Temperature

Setting the Mixing and Distribution Valve (Figure 9)

Both valves are set in the same manner, that is by motors and gears which run or stop in both directions. The components TLE 4201 drive the motor, while the μC controls them via ports P50 and P51 (mixing valve) as well as P52 and P53 (distribution valve). The analog-to-digital converter of the processor is informed of the value position by means of the voltage on a potentiometer which is connected to the valve and supplied by the analog reference voltages. An RC network filters out interference. When the difference between the nominal and actual value of a valve exceeds a certain tolerance margin, the motor is driven in the respective direction.

The valves should reach their end positions (mechanical stops) but the motor, for mechanical reasons, should not be driven continuously in these positions. Since it is difficult to solve this problem by an accurate adjustment of the potentiometer, the system recognizes a mechanical stop when the difference between the actual and the nominal value remains the same although the motor is running. In response, the motor is switched off and the actual value is stored. After that the system will stop when this value has been reached. Only after a certain period of time (approx. 10 minutes) or each time the ignition has been turned on, the user can change the stop by depressing the respective key for selection of a max. position. As a prerequisite for this type of stop recognition, the electrical region of the potentiometer should not be fully utilized by the valve angle.

The slight, relatively rapid fluctuations in the valve nominal value in the regulation or control mode are suppressed to prevent mechanical wear and tear. This suppression is of no consequence to the passengers.

Switching Over the Air Supply

During the automatic function, fresh air will be supplied when the following conditions for air circulation are not met:

— outside temperature > nominal temperature +10K (°C)

— outside temperature > inside temperature

To prevent the valve from switching continuously, a hysteresis of 2K in each direction is used.

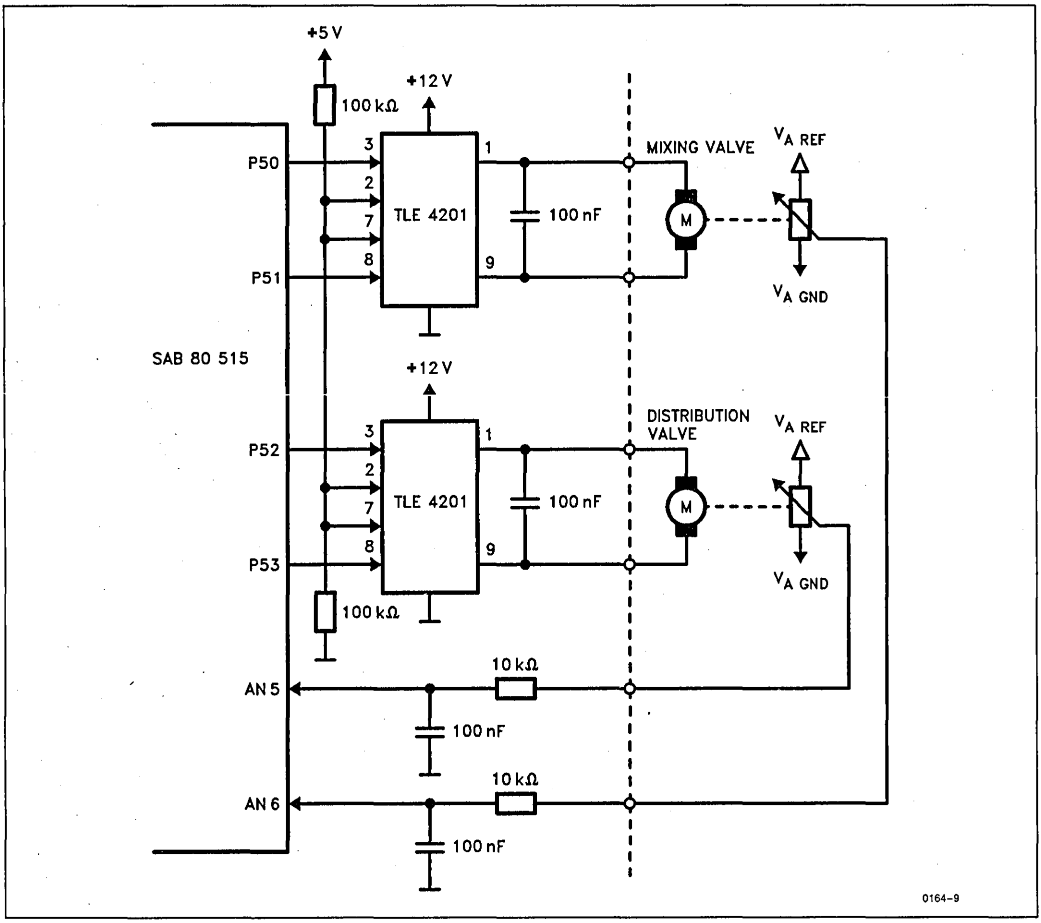


Figure 9. Control of Mixing and Distribution Valve

During "defrost" or "fresh air supply" the system takes in fresh air; during "ambient air supply", the air in the car is circulated.

The magnetic valve for the air circulation valve is switched on via P15 by the SAB 80515 with a Darlington transistor BC 879. A resistor on the output of the transistor protects against short-term interference (see Figure 10). A short-circuit in the electrical supply can be detected via P05. In this case, the processor immediately stops the control of the valve, but periodically attempts to reactivate it every few seconds. The magnetic valve controls a vacuum motor for activating the air circulation valve.

Water Valve Activation

When there is no heating requirement, the electronics inhibit the water flow to the heat exchanger by means of a valve. As a result, the temperature inside the car can be reduced by several degrees in the summer when compared to operation with closed mixing valve. The criterion for inhibiting the water flow is the mixing valve's position at the lower stop. Only after the mixing valve has changed its position by a defined distance from the stop, will the processor enable the valve again.

The magnetic valve for inhibiting the water flow is controlled in a manner similar to that used for the air circulation valve (see Figure 10). P16 and P06 are used as outputs or feedback pins. As can be seen in Figure 10, the magnetic valve pneumatically activates the inhibit valve.

Enabling/Disabling the Compressor

During the automatic function, the compressor is disabled only if the outside temperature drops by more than 10K below the nominal temperature. Again a hysteresis of 2K (K = °C) is applied for the switching procedure.

During "defrost" or "compressor ON", the air conditioning unit operates continuously, but stops completely during "compressor OFF".

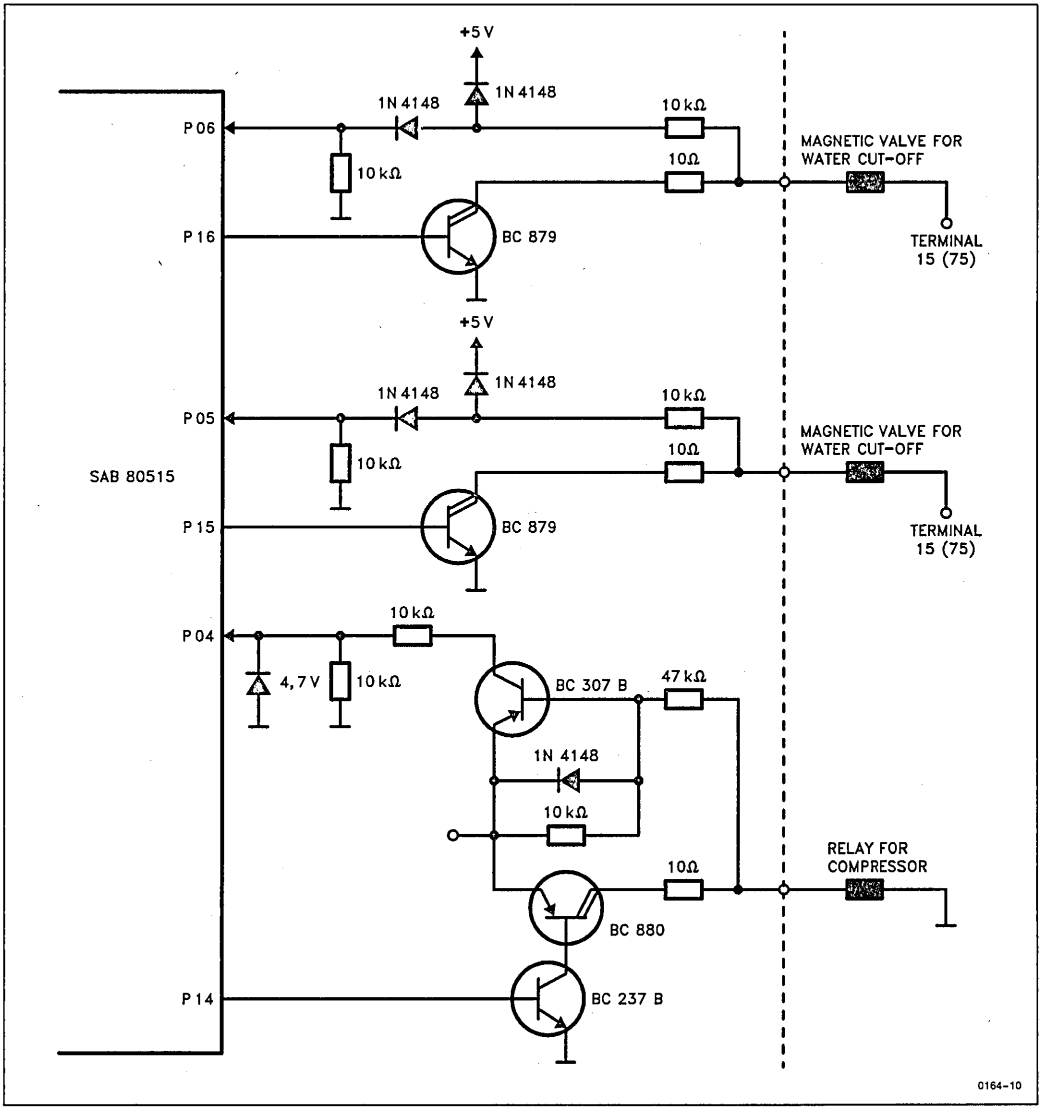


Figure 10. Control of Water Valve, Circulation Valve and Compressor

The compressor relay is driven (see Figure 10) by P14 of the SAB 80515 as well as two transistors BC 237 and BC 880 for increasing the current and converting the levels. A drive signal short-circuited to ground—after level conversion by transistor BC 307—can be detected via P04 and the system is then switched off for several seconds. The external compressor relay activates the magnetic coupling for driving the compressor, however, only if the (electronically independent) defroster for the carburetor does not respond.

To Drive the Blower

Initially, the speed of the blower is a function of two variables as can be seen in Figure 11. An increase in speed as a function of the nominal-actual temperature difference leads to rapid temperature adjustment. During extreme outside temperatures, the heating or air conditioning effect has to be supported continuously by the blower. The curve minimum is therefore displaced since the average thermal effect of sun rays has been taken into account. The two functions are additively combined.

When both the inside temperature and the injection temperature lie below or above the nominal value for the inside temperature, the blower speed is reduced. Otherwise the already uncomfortably cold car would get colder or, if already too hot, hotter.

Two points were included when considering the dependence on the road speed; during higher speeds the dynamic air pressure increasingly replaces the blower output. In response the blower speed is reduced in proportion to the speed or set to zero, if required. However, during lower speeds or when the car is parked, the noise generated by the blower is irritating, and polluted air is brought in e.g. during heavy traffic. The speed of the blower is therefore reduced.

These automatic functions can be overriden with the blower or defrost key. The blower reaches its max. speed during "full speed blower capacity" or "defrost", operates at a medium speed at "half blower capacity" or not at all in "blower off".

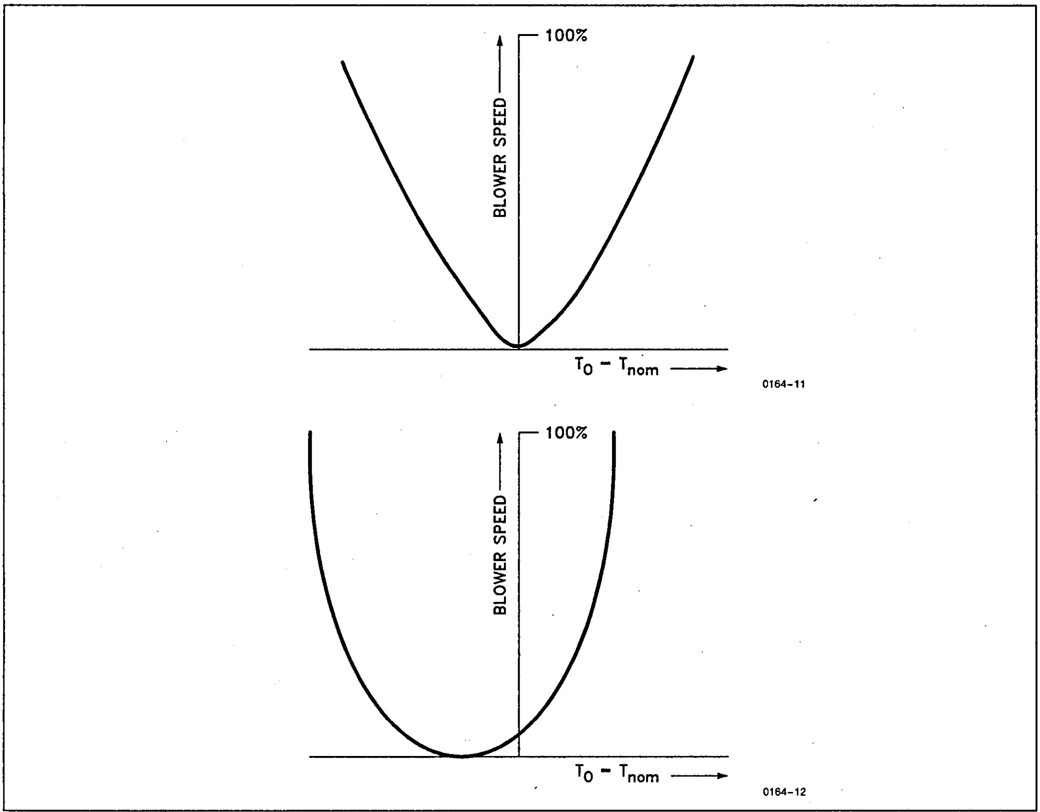


Figure 11. Examples for Blower Speed Characteristics

The blower is driven (Figure 12) by the pulse-width modulated signal generated by the microcontroller at P12 with the aid of timer 2. The timer—as described in section 2—has been programmed for an overflow every 512 μs. The compare/capture register 2 operates in the compare mode (mode 0), port 12 is in "L" during timer overflow, or in "H" when the content of the timer and compare register is the same. Therefore, by changing the content of the compare register, the pulse duty factor at P12 can be varied. An HCMOS inverter and an RC combination convert the microcontroller output signal into an analog voltage ranging between 0V and 5V. This voltage drives the blower driver located outside the electronics.

During the standby status, a signal from the voltage supply prevents a voltage from being applied to the blower.

When the blower is operating at full capacity, there should be no voltage drop across the power transistor of the blower driver. Therefore the transistor is by-passed with a relay. For this part of the driver, the SAB 80515 connects a 12V signal to the blower output using two transistors (BC 237 and BC 307). The relay is switched off—to prevent wear and tear—when the speed of the blower is reduced.

Testing and Optimization Support

By encoding at ports P00–P02, the following quantities can be displayed in place of the nominal or outside temperature:

1. Inside temperature T_I in °C
2. Injection temperature T_In in °C
3. Mixing valve setting in %
4. Distribution valve setting in %
5. Blower drive in %
6. Status of compressor, air circulation as well as water valve

7a. Memory address of internal RAM, which can be set

7b. Memory content associated with this address, which can be varied

The controlling and regulating procedures of the system can be monitored with displays 1 through 6.

During operation, the system can be accessed and all memory contents can be indicated with display 7. The settings are performed in the same manner as the changes in nominal temperature, namely with keys "+", and "−". The outside temperature key is used to switch between the memory address and content. There are functions suitable for manual control, and there are those which should be left in the automatic mode.

The possibility of user access has been providod for adjusting the parameters of the two-stago PID control to the respective vehicle. The parameters aro not established by the program. Instead they are stored in the RAM in the memory area saved during standby operation. Only after the voltage has been switched off, the parameters can be loaded with

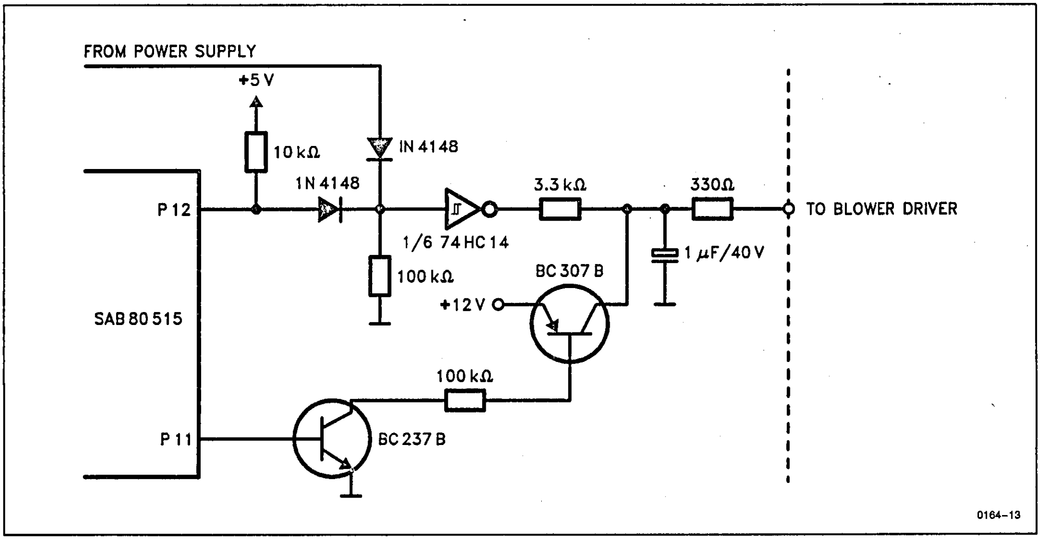


Figure 12. Blower Control

fixed values during initialization. The test engineer can therefore change the parameters according to the test results, although the device has already been installed.

In order to provide defined start conditions for a test, all controlling and regulating functions can be switched off during the setting procedures with P03.

After the test has been completed, the established optimal parameters values can be permanently programmed in the EPROM or in the masked ROM.

Use of the On-Chip Periphery of the SAB 80515

Table 1 includes the functions of the integrated periphery of the SAB 80515. As can be seen, almost all elements are utilized.

Table 1. Use of the SAB 80515 On-Chip Periphery

Periphery	Application
Analog-to-Digital Converter	for measuring —temperatures —brightness level and for setting mixing and distribution value
Timer 0	for generating a standard time clock
Timer 1	for measuring road speed
Timer 2	for controlling the time for LED multiplex display (with brightness control) for controlling the pulse/pause ratio for the blower
Watchdog Timer	for system monitoring
Serial Inteface	for diagnostic purposes
Ports	for interrogating and driving digital and time analog inputs/ outputs

Alternatives and Upgrades

Changes in the functions of the described sample—provided the sensors and actuators remain the same—can easily be realized by merely modifying the program or the stored tables. However, when different or additional sensors or actuators are used, the hardware must be changed as well.

For example the mixing valve can be replaced by a clocked valve which alternately releases and interrupts the water flow between the cooler and heat exchanger. According to the pulse/pause ratio of the drive signal the heat exchanger temperature changes and thus the air injected into the car. The previously described hot water valve is in this case omitted.

Also, in addition to the distribution valve, other elements for changing the air distribution can be controlled, e.g. the vent flaps at the dashboard.

If the serial interface of the SAB 80515 is not used, the system could be diagnosed during practical application and inspections.

SIEMENS

November 1988

E^{2}PROM Interface with a Siemens 8031 based Microcontroller

SAB-51 Family

Application Note

8

This application note provides users with a solution to interface an E^{2}PROM to an 8031 based microcontroller over the I^{2}C bus. In this application example, a Siemens microcontroller the SAB 80535 and a Siemens E^{2}PROM the SDA 2526 or the SDA 2516 are used.

An on-chip Electrically Erasable Programmable Read-Only Memory (E^{2}PROM) in a microcontroller becomes a very useful peripheral because it allows the system parameters to be stored and reprogrammed without having to remove the microcontroller from the board. An E^{2}PROM can also store internally the system specific information and can quite often replace a battery backed-up CMOS RAM. The write-cycle time of an E^{2}PROM is considerably longer than that of most RAM chips and therefore the device is referred to as a "read-only" memory. Most E^{2}PROMs have a limited number of write/erase endurance cycles and for this reason alone an on-chip E^{2}PROM in a microcontroller is sometimes not desirable. However, Siemens microcontrollers do not have an on-chip E^{2}PROM. Therefore, the user either has to rely on an external E^{2}PROM chip or perform the same function by battery back-up of the microcontroller's internal RAM.

The following application example offers the users a solution of interfacing an E^{2}PROM to a Siemens microcontroller over a two line I^{2}C interface.

Inter-Integrated Circuit Bus

The Inter-Integrated Circuit (I^{2}C) bus is a mechanism for serially communicating with peripheral devices. The bi-directional bus consists of two wires, and it can support multiple masters and operate at various data rates.

The physical part of an I^{2}C interface is a set of two wires, the serial clock (SCL) line and the serial data (SDA) line. The clock line is used to clock data out of a transmitting device into a receiving device. The data line is used to carry the data bits from a transmitting device to a receiving device. It is also used to transfer the acknowledge signal from a receiving device to a transmitting device.

When transferring address, data or acknowledge bits the SDA line can change its state only while the clock is low. If the data line changes from high to low while the SCL line is high, a start condition is initiated. A change from low to high while the SCL line is high initiates a stop condition. The basic relationship between the SCL and SDA lines is shown in Figure 1.

Hardware

Figure 2 illustrates the hardware interface between a Siemens microcontroller such as the SAB 80535 and a Siemens E^{2}PROM such as the SDA 2526 or the SDA 2516. The bit manipulation capability of the SAB 80515/535 allows any two port pins of the microcontroller to be used for the E^{2}PROM with an I^{2}C bus interface. In this example, port pin P1.0 is used for the SDA line and port pin P1.1 is used for the SCL line of the I^{2}C bus. One could select any two port pins for the interface by making an appropriate change in the software.

Via the I^{2}C bus the memory is controlled by the microcontroller (master) during two operating modes:

a. Read-out cycle.

b. Reprogramming cycle or the write cycle.

In both operating modes the microcontroller has to provide 3 bytes and an additional acknowledge clock on the bus after the start condition. These three bytes contain information like chip select for data input, memory word address and chip select for data output. For more information on the operation of the SDA 2526 or the SDA 2516, please refer to their data-sheets.

Software

The software listing of the subroutines used to read and program (write) the SDA 2526 or the SDA 2516 is attached.

The subroutine Re_EEPROM allows the user to read the E^{2}PROM. The following parameters need to be transferred to this subroutine to complete the read operation successfully.

a. Register R1 holds the starting address of the E^{2}PROM.

b. Register R2 holds the number of bytes to be read.

c. Register R0 holds the destination address in the internal RAM of the SAB 80535.

The subroutine Pr_EEPROM allows the user to program the E^{2}PROM. After programming a byte, the software executes a time delay of about 30 ms (worst case programming duration for the SDA 2526 or the SDA 2516) before writing the next byte. Registers R0, R1 and R2 need to be programmed with the following values before calling this subroutine.

a. Register R1 holds the starting address of the E^2PROM

b. Register R2 holds the number of bytes to be programmed.

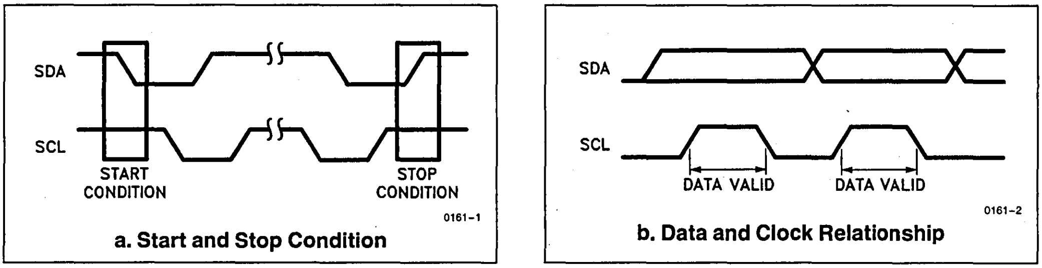
c. Register R0 holds the source address of the data in the internal RAM of the SAB 80535.

References

i. Siemens SAB 80515/535 User's Manual.

ii. Siemens SDA 2526/2516 Data Sheets.

iii. Siemens I^2C Bus Driver Subroutines by R. Mirthes and P. Walter.



a. Start and Stop Condition

b. Data and Clock Relationship

Figure 1. I^2C Bus Basic Timing

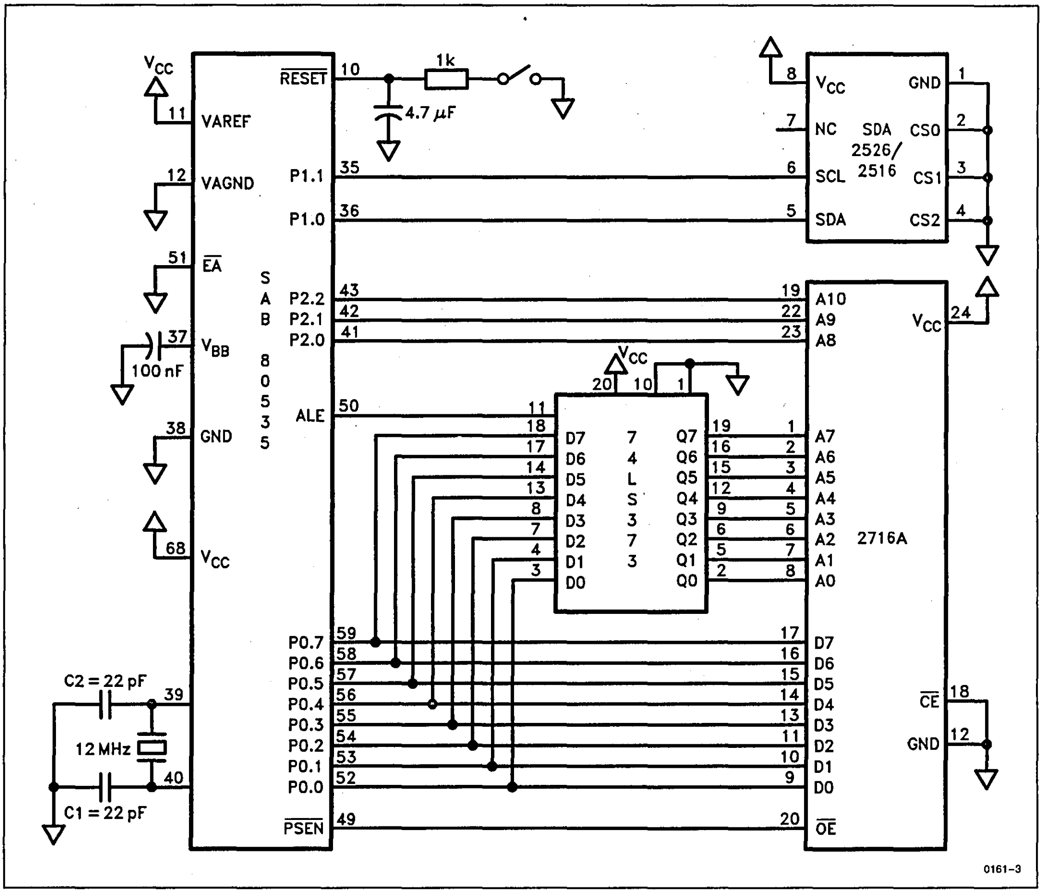
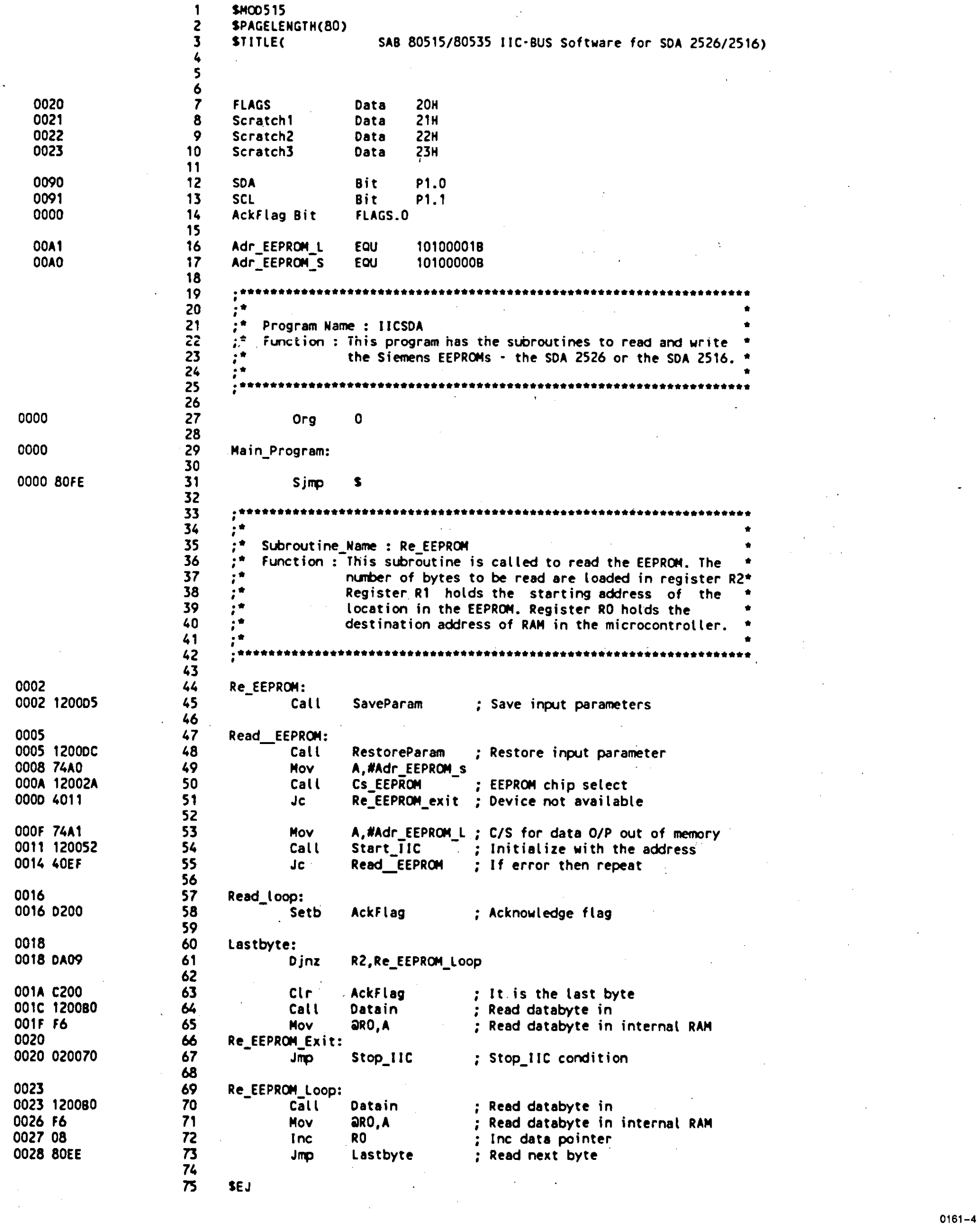


Figure 2. Schematic Diagram

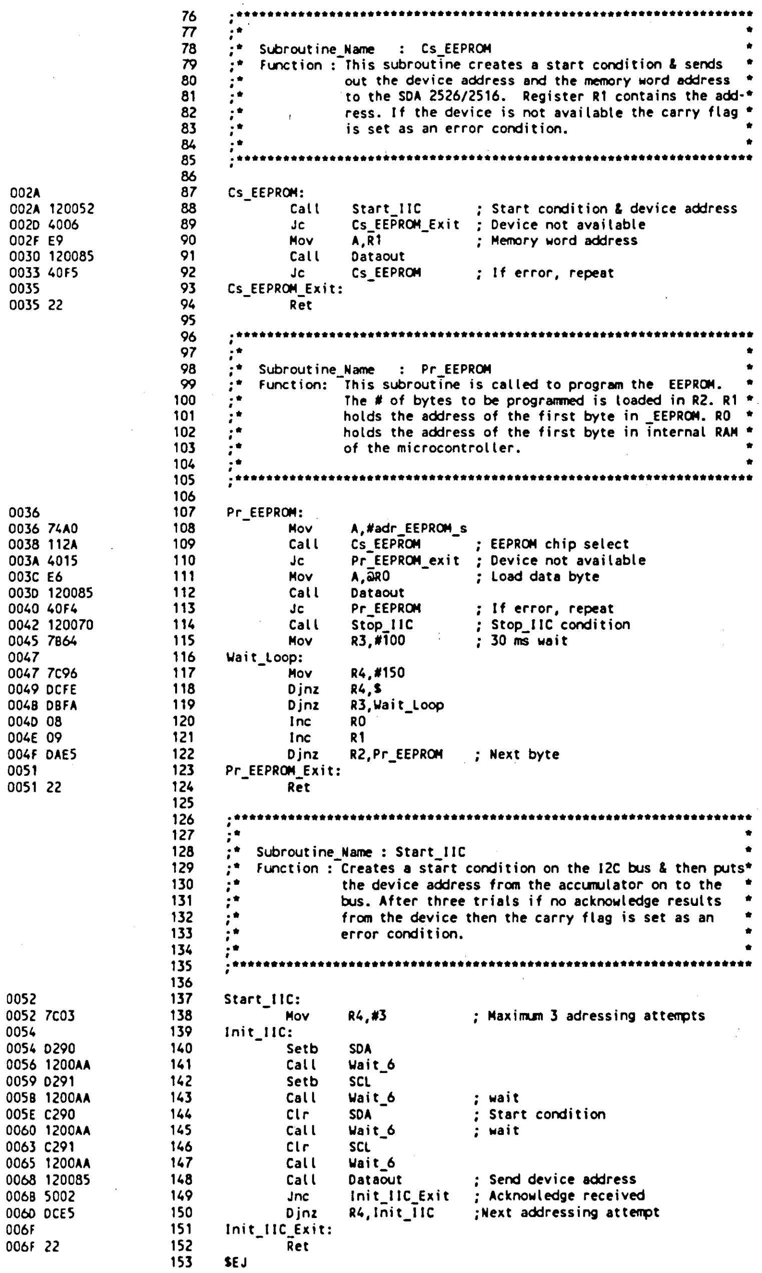
Program Listings



```
                $MOD515
                $PAGELENGTH(80)
                $TITLE(          SAB 80515/80535 IIC-BUS Software for SDA 2526/2516)



 0020           FLAGS           Data    20H
 0021           Scratch1        Data    21H
 0022           Scratch2        Data    22H
 0023           Scratch3        Data    23H

 0090           SDA             Bit     P1.0
 0091           SCL             Bit     P1.1
 0000           AckFlag Bit     FLAGS.0

 00A1           Adr_EEPROM_L    EQU     10100001B
 00A0           Adr_EEPROM_S    EQU     10100000B

                ;****************************************************************
                ;*                                                              *
                ;*  Program Name : IICSDA                                       *
                ;*  Function : This program has the subroutines to read and write  *
                ;*             the Siemens EEPROMs - the SDA 2526 or the SDA 2516. *
                ;*                                                              *
                ;****************************************************************

0000                   Org     0

0000            Main_Program:

0000 80FE              Sjmp    $

                ;****************************************************************
                ;*                                                              *
                ;*  Subroutine_Name : Re_EEPROM                                 *
                ;*  Function : This subroutine is called to read the EEPROM. The  *
                ;*             number of bytes to be read are loaded in register R2*
                ;*             Register R1  holds the  starting address  of  the  *
                ;*             location in the EEPROM. Register R0 holds the      *
                ;*             destination address of RAM in the microcontroller.  *
                ;*                                                              *
                ;****************************************************************

0002            Re_EEPROM:
0002 1200D5            Call    SaveParam       ; Save input parameters

0005            Read__EEPROM:
0005 1200DC            Call    RestoreParam    ; Restore input parameter
0008 74A0              Mov     A,#Adr_EEPROM_s
000A 12002A            Call    Cs_EEPROM       ; EEPROM chip select
000D 4011              Jc      Re_EEPROM_exit  ; Device not available

000F 74A1              Mov     A,#Adr_EEPROM_L ; C/S for data O/P out of memory
0011 120052            Call    Start_IIC       ; Initialize with the address
0014 40EF              Jc      Read__EEPROM    ; If error then repeat

0016            Read_loop:
0016 D200              Setb    AckFlag         ; Acknowledge flag

0018            Lastbyte:
0018 DA09              Djnz    R2,Re_EEPROM_Loop

001A C200              Clr     AckFlag         ; It is the last byte
001C 120080            Call    Datain          ; Read databyte in
001F F6                Mov     @R0,A           ; Read databyte in internal RAM
0020            Re_EEPROM_Exit:
0020 020070            Jmp     Stop_IIC        ; Stop_IIC condition

0023            Re_EEPROM_Loop:
0023 120080            Call    Datain          ; Read databyte in
0026 F6                Mov     @R0,A           ; Read databyte in internal RAM
0027 08                Inc     R0              ; Inc data pointer
0028 80EE              Jmp     Lastbyte        ; Read next byte

                $EJ
```

0161-4

Program Listings (Continued)



```
                ;****************************************************************
                ;*                                                              *
                ;*  Subroutine_Name   :  Cs_EEPROM                              *
                ;*  Function : This subroutine creates a start condition & sends *
                ;*             out the device address and the memory word address *
                ;*             to the SDA 2526/2516.  Register R1 contains the add-*
                ;*             ress. If the device is not available the carry flag *
                ;*             is set as an error condition.                      *
                ;*                                                              *
                ;****************************************************************

002A            Cs_EEPROM:
002A 120052             Call    Start_IIC        ; Start condition & device address
002D 4006               Jc      Cs_EEPROM_Exit   ; Device not available
002F E9                 Mov     A,R1             ; Memory word address
0030 120085             Call    Dataout
0033 40F5               Jc      Cs_EEPROM        ; If error, repeat
0035            Cs_EEPROM_Exit:
0035 22                 Ret

                ;****************************************************************
                ;*                                                              *
                ;*  Subroutine_Name   :  Pr_EEPROM                              *
                ;*  Function:  This subroutine is called to program the  EEPROM.  *
                ;*             The # of bytes to be programmed is loaded in R2. R1 *
                ;*             holds the address of the first byte in _EEPROM. R0  *
                ;*             holds the address of the first byte in internal RAM *
                ;*             of the microcontroller.                           *
                ;*                                                              *
                ;****************************************************************

0036            Pr_EEPROM:
0036 74A0               Mov     A,#adr_EEPROM_s
0038 112A               Call    Cs_EEPROM        ; EEPROM chip select
003A 4015               Jc      Pr_EEPROM_exit   ; Device not available
003C E6                 Mov     A,@R0            ; Load data byte
003D 120085             Call    Dataout
0040 40F4               Jc      Pr_EEPROM        ; If error, repeat
0042 120070             Call    Stop_IIC         ; Stop_IIC condition
0045 7B64               Mov     R3,#100          ; 30 ms wait
0047            Wait_Loop:
0047 7C96               Mov     R4,#150
0049 DCFE               Djnz    R4,$
004B DBFA               Djnz    R3,Wait_Loop
004D 08                 Inc     R0
004E 09                 Inc     R1
004F DAE5               Djnz    R2,Pr_EEPROM     ; Next byte
0051            Pr_EEPROM_Exit:
0051 22                 Ret

                ;****************************************************************
                ;*                                                              *
                ;*  Subroutine_Name : Start_IIC                                 *
                ;*  Function : Creates a start condition on the I2C bus & then puts*
                ;*             the device address from the accumulator on to the  *
                ;*             bus. After three trials if no acknowledge results  *
                ;*             from the device then the carry flag is set as an   *
                ;*             error condition.                                   *
                ;*                                                              *
                ;****************************************************************

0052            Start_IIC:
0052 7C03               Mov     R4,#3            ; Maximum 3 adressing attempts
0054            Init_IIC:
0054 D290               Setb    SDA
0056 1200AA             Call    Wait_6
0059 D291               Setb    SCL
005B 1200AA             Call    Wait_6           ; wait
005E C290               Clr     SDA              ; Start condition
0060 1200AA             Call    Wait_6           ; wait
0063 C291               Clr     SCL
0065 1200AA             Call    Wait_6
0068 120085             Call    Dataout          ; Send device address
006B 5002               Jnc     Init_IIC_Exit    ; Acknowledge received
006D DCE5               Djnz    R4,Init_IIC      ;Next addressing attempt
006F            Init_IIC_Exit:
006F 22                 Ret
                $EJ
```

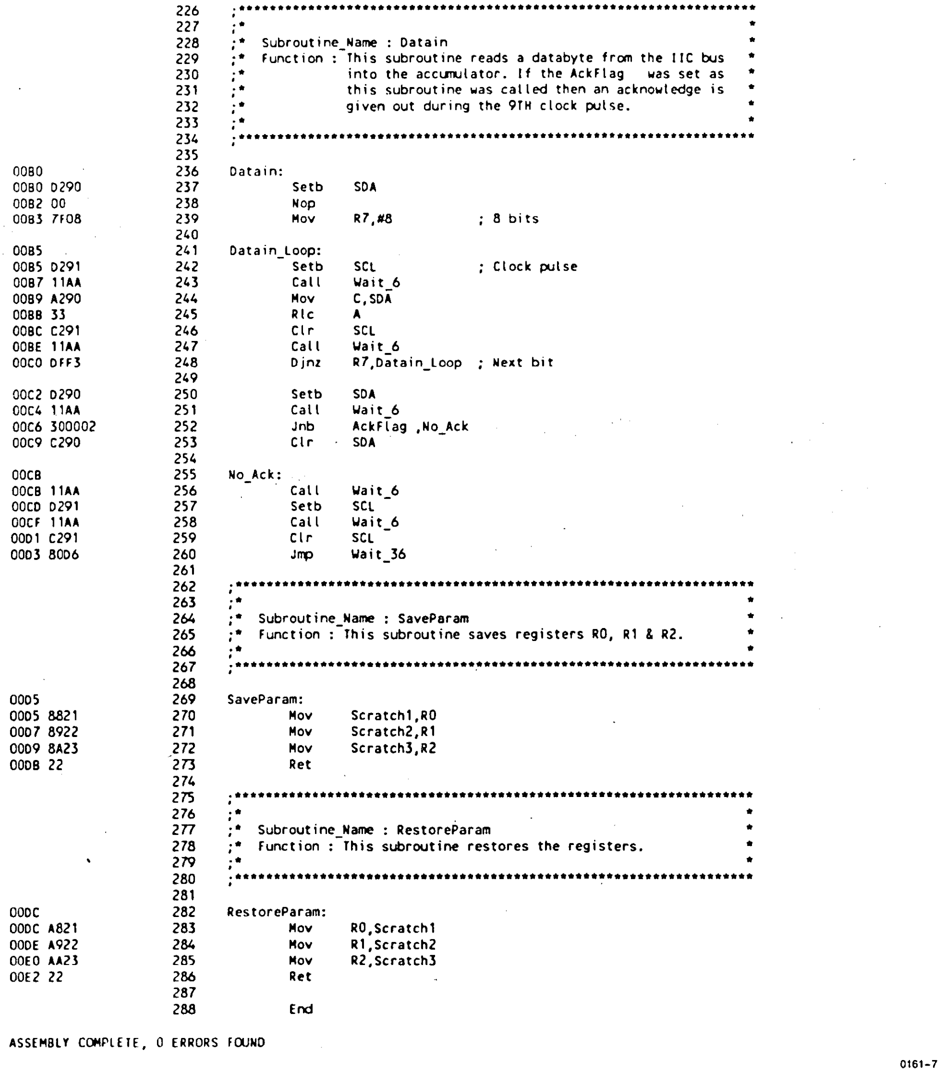
0161-5

Program Listings (Continued)

```
            ;*****************************************************************
            ;*                                                               *
            ;*  Subroutine_Name : Stop_IIC                                   *
            ;*  Function : This sobroutine creates a stop condition on the IIC *
            ;*             bus. Then the status of the SDA line is checked - if*
            ;*             low then clock pulses are sent out until SDA line  *
            ;*             goes high.                                        *
            ;*                                                               *
            ;*****************************************************************

0070        Stop_IIC:
0070 C291           Clr     SCL
0072 1200AA         Call    Wait_6
0075 C290           Clr     SDA
0077 1200AA         Call    Wait_6
007A D291           Setb    SCL
007C 1200AA         Call    Wait_6
007F D290           Setb    SDA

0081        IsSDALow:
0081 3090EC         Jnb     SDA,Stop_IIC    ; SDA is not high
0084 22             Ret


            ;*****************************************************************
            ;*                                                               *
            ;*  Subroutine_Name : Dataout                                    *
            ;*  Function : This subroutine transmits a databyte on the IIC bus *
            ;*             from the accumulator. The carry flag is set if an  *
            ;*             error results.                                    *
            ;*                                                               *
            ;*****************************************************************

0085        Dataout:
0085 7F08           Mov     R7,#8           ; 8 bits in a byte

0087        Dataout_Loop:
0087 33             Rlc     A
0088 9290           Mov     SDA,C           ; Send data bit out
008A 00             Nop
008B D291           Setb    SCL             ; Set the clock bit
008D 1200AA         Call    Wait_6
0090 C291           Clr     SCL
0092 1200AA         Call    Wait_6
0095 DFF0           Djnz    R7,Dataout_Loop
0097 33             Rlc     A
0098 D290           Setb    SDA
009A 00             Nop
009B D291           Setb    SCL             ; Clock pulse for acknowledge
009D 1200AA         Call    Wait_6
00A0 A290           Mov     C,SDA           ; Get acknowledge bit
00A2 C291           Clr     SCL
00A4 1200AB         Call    Wait_36
00A7 40C7           Jc      Stop_IIC
00A9 22             Ret

            ;*****************************************************************
            ;*                                                               *
            ;*  Subroutine_Name : Wait_6, Wait_36                            *
            ;*  Function : These subroutines cause delays.                   *
            ;*                                                               *
            ;*****************************************************************

00AA        Wait_6:
00AA 22             Ret

00AB        Wait_36:
00AB 7B06           Mov     R3,#6
00AD DBFE           Djnz    R3,$
00AF 22             Ret

            $EJ
```

0161-6

Program Listings (Continued)



```
             ;****************************************************************
             ;*                                                              *
             ;*  Subroutine_Name : Datain                                    *
             ;*  Function : This subroutine reads a databyte from the IIC bus *
             ;*             into the accumulator. If the AckFlag  was set as  *
             ;*             this subroutine was called then an acknowledge is *
             ;*             given out during the 9TH clock pulse.             *
             ;*                                                              *
             ;****************************************************************

0080         Datain:
0080 D290            Setb    SDA
0082 00              Nop
0083 7F08            Mov     R7,#8           ; 8 bits

0085         Datain_Loop:
0085 D291            Setb    SCL             ; Clock pulse
0087 11AA            Call    Wait_6
0089 A290            Mov     C,SDA
008B 33              Rlc     A
008C C291            Clr     SCL
008E 11AA            Call    Wait_6
00C0 DFF3            Djnz    R7,Datain_Loop  ; Next bit

00C2 D290            Setb    SDA
00C4 11AA            Call    Wait_6
00C6 300002          Jnb     AckFlag ,No_Ack
00C9 C290            Clr     SDA

00CB         No_Ack:
00CB 11AA            Call    Wait_6
00CD D291            Setb    SCL
00CF 11AA            Call    Wait_6
00D1 C291            Clr     SCL
00D3 80D6            Jmp     Wait_36

             ;****************************************************************
             ;*                                                              *
             ;*  Subroutine_Name : SaveParam                                 *
             ;*  Function : This subroutine saves registers R0, R1 & R2.     *
             ;*                                                              *
             ;****************************************************************

0005         SaveParam:
0005 8821            Mov     Scratch1,R0
0007 8922            Mov     Scratch2,R1
0009 8A23            Mov     Scratch3,R2
000B 22              Ret

             ;****************************************************************
             ;*                                                              *
             ;*  Subroutine_Name : RestoreParam                              *
             ;*  Function : This subroutine restores the registers.          *
             ;*                                                              *
             ;****************************************************************

000C         RestoreParam:
000C A821            Mov     R0,Scratch1
000E A922            Mov     R1,Scratch2
00E0 AA23            Mov     R2,Scratch3
00E2 22              Ret

                     End

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

0161-7

SIEMENS

November 1988

Programmable Timer/Counter Register Array in Microcontrollers

SAB 80515/80535

Article Reprint

Abstract

In the past the addition of on-chip memory to microprocessors was the distinctive feature which separated microcontrollers from microprocessors. Since then many bells and whistles have been added to microcontrollers to clearly distinguish them from microprocessors. To appreciate the depth of the microcontroller function cornucopia, consider the following (seemingly endless, but nevertheless partial) list of available features: ROM, RAM, EPROM, E^2PROM, A/D converter, PWM, full-duplex serial ports, on-chip counter/timer, latched I/O, Boolean manipulation etc. Far from dead, the 8-bit microcontroller market continues to grow, as more functions with ever increasing performance are integrated on a single-chip by taking advantage of the advanced NMOS and CMOS processes, tighter design rules, and architectural improvements. One such enhanced functional unit of the microcontroller is a timer/counter unit, which in the past had only been used to generate delays and count pulses. The microcontroller architectures have continued to evolve and expand the scope of the timer/counter functions. This article makes an attempt to re-classify a timer unit of the Siemens 80535/515 (a member of Siemens 8051-based microcontrollers) to give a new wrinkle to control applications. If you have ever felt a need to measure frequencies and pulse widths, or had an application to combine digital and analog functions, or wanted to distinguish noise from real signals, or needed to generate PWM signals, then your solution is the PTRA (Programmable Timer/Counter Register Array) of the Siemens 80535/515 microcontroller. Intel's 8051FA microcontroller has a similar timer unit called the PCA (Programmable Counter Array). But before we draw any conclusions it would be worthwhile to review the features of the PTRA unit available in the Siemens 80535/515 and compare it to the PCA unit of Intel's 8051FA. We will then discuss an application example to evaluate the capabilities and limitations of such timer units.

What is in a name?

A Timer circuit is not new in microcomputers, but of all the on-chip peripheral functions of the 8-bit SAB 80535/515 microcomputer—RAM, ROM, analog-to-digital converter, serial port—none is more useful or important. Two of the SAB 80535/515's forerunners, the SAB 8031/51 and SAB 8032/52, incorporated timers, and it is from these circuits that a new enhanced timer function is derived. The time-base for the PTRA of the SAB 80535/515 is a programmable 16-bit timer/counter also known as the Timer 2. Timer 2 is the only one that can serve the PTRA.

The PTRA consists of the following registers.

T2CON	Timer 2 Control Register
TL2	Timer 2 Register, Low-Byte
TH2	Timer 2 Register, High-Byte
CRCL	Compare/Reload/Capture Register, Low-Byte
CRCH	Compare/Reload/Capture Register, High-Byte
CCL1	Compare/Capture Register 1, Low-Byte
CCH1	Compare/Capture Register 1, High-Byte
CCL2	Compare/Capture Register 2, Low-Byte
CCH2	Compare/Capture Register 2, High-Byte
CCL3	Compare/Capture Register 3, Low-Byte
CCH3	Compare/Capture Register 3, High-Byte
CCEN	Compare/Capture Enable Register

For brevity, the double-byte Compare/Reload/Capture register is called CRC register, and the three double-byte Compare/Capture registers are called CC registers 1 to 3. The PTRA shares Port 1 pins for hardware interfacing as shown in Table 1.

In addition to supporting the operational modes of a timer or a counter, the PTRA provides the following features for the Timer 2.

—16 bit reload
—16 bit compare
—16 bit capture

Figure 1 shows a block diagram of the PTRA unit.

The Timer 2 of PTRA can operate either as timer, event counter, or gated timer. In timer mode, the count rate is derived from the oscillator frequency. A 2:1 prescaler makes it possible to select a count rate of 1/12 or 1/24 of the oscillator frequency. In either case, no matter whether Timer 2 is configured as timer, event counter, or gated timer; when the count rolls over from all 1's to all 0's it sets the Timer 2 overflow flag which can generate an interrupt. The timer overflow function can serve as a periodic interrupt for performing a keyboard scan or for refreshing the screen display. It can also be used to indicate end of delay for the execution of time delayed events. In counter mode, the Timer 2 register is incremented in response to a 1-to-0 transition at its corresponding external input pin T2 (P1.7).

The reload mode of the PTRA unit allows contents of Timer 2 to be reloaded from the CRC register at the time when Timer 2 rolls over from all 1's to all 0's—a feature which is quite useful in varying periods of pulse width modulated signals. The 16-bit reload from the CRC register to Timer 2 registers can also be caused by a negative transition at port pin P1.5.

Table 1. Hardware Interfacing for the PTRA Unit

Port Pin	Name	Function
P1.0	INT3/CC0	Compare Output/Capture Input for the CRC Register
P1.1	INT4/CC1	Compare Output/Capture Input for CC Register 1
P1.2	INT5/CC2	Compare Output/Capture Input for CC Register 2
P1.3	INT6/CC3	Compare Output/Capture Input for CC Register 3
P1.5	T2EX	External Reload Trigger Input
P1.7	T2	External Count or Gate Input to Timer 2

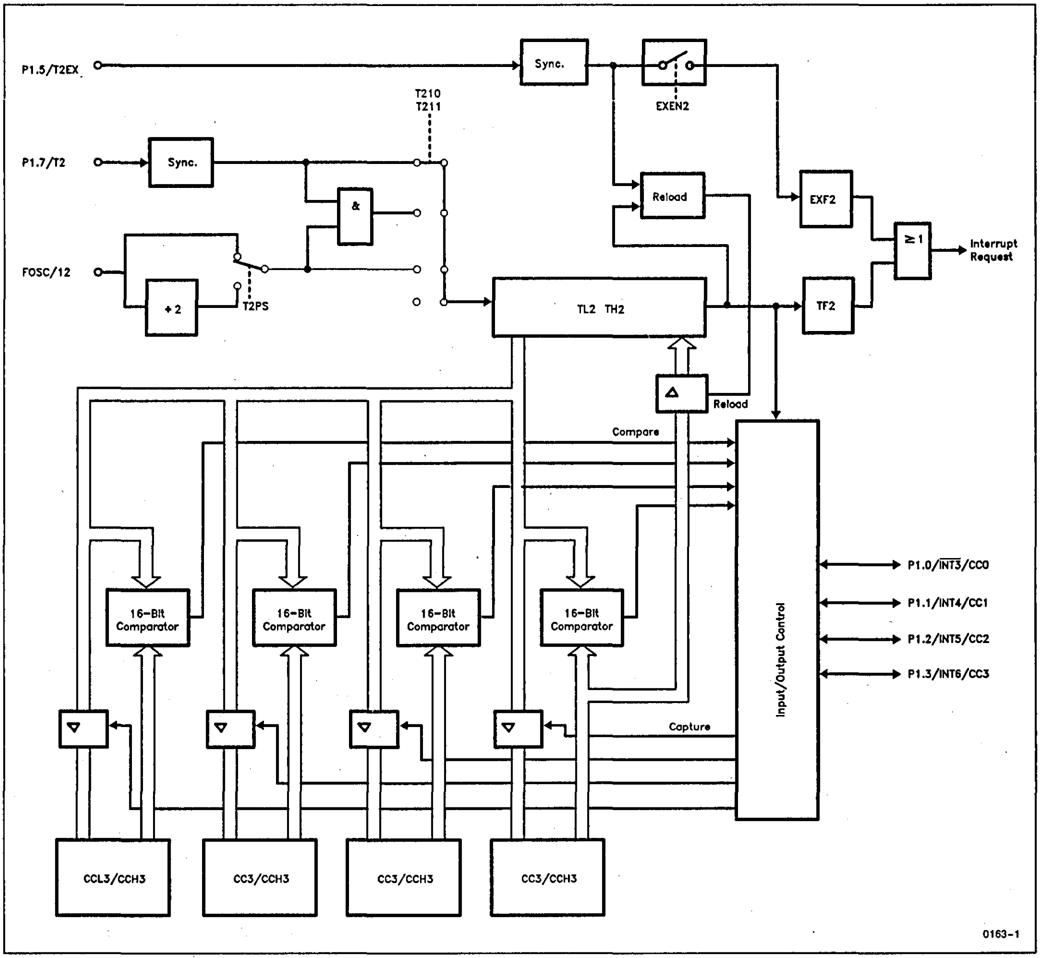


Figure 1. Block Diagram of the PRTA Unit

In the compare mode, the 16-bit values stored in the dedicated compare registers of the PTRA are compared with the contents of the Timer 2 registers TL2 and TH2. If the count value in the Timer 2 register matches the stored value an appropriate output signal is generated at the corresponding port 1 pin, and an interrupt can also be generated. In one of the two compare modes, the Timer 2 of PTRA can be used to generate PWM signals as shown by the functional diagram in Figure 2.

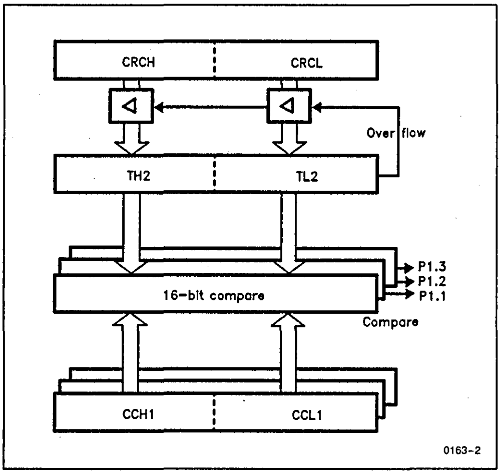


Figure 2. PWM Using the PTRA Unit

In the PTRA set-up (compare mode-0) for PWM signal generation, the output signals at port pins P1.1 through P1.3 change from low to high upon a match of the corresponding CC1 through CC3 register contents with contents of the Timer 2 registers. The outputs then go back to a low level on Timer 2 overflow, and Timer 2 restarts by reloading the CRC register contents. The CRC value controls the period and the registers CC1 through CC3 control the pulse widths of the PWM signals generated at port pins P1.1 through P1.3 respectively. In the second mode of operation the output port pins P1.0 through P1.3 can be caused to toggle at compare events, thereby resulting in the generation of square waves at these outputs. In this mode the Timer 2 overflow has no affect on the output port pins. In this mode of operation one can also generate pulses of varying widths from a few to several thousand microseconds without any major software interruption.

Each of the three compare/capture and the CRC registers of the PTRA unit can be used to latch the current 16-bit value of the Timer 2 in one of them. This latching could either be caused by an external event or upon writing to the low-order byte of the dedicated 16-bit capture register. Either a rising or a falling edge can be selected to cause the capture from Timer 2 into the CRC register, whereas for the CC registers 1 to 3 a positive transition causes the capture. In the former case, this capability is useful in applications to determine the state of an input pin, or in discriminating noise (a narrow signal) from real signal, or in measuring pulse widths.The other application of input capture is to determine the frequency of a signal applied to the input pin. By capturing the counter value on successive rising or falling edges, the timer determines the number of elapsed counts. Then, knowing the count rate, the frequency of the input signal can be computed.

The PCA unit of Intel's 8051FA can perform similar functions including 16-bit capture/compare with capture activated by negative or positive or both edge transition. In addition, one of the compare registers could be preset and programmed to cause an internal reset on a match, thereby acting as a watchdog timer—a function which is supported by another timer in the SAB 80535/515. The PCA unit of 8051FA supports 8-bit PWM as opposed to 16-bit by the PTRA, and has no reload capability for the timer. On the other hand the timer of the PCA can be programmed to count at a rate of FOSC/4 and supports High Speed Output (HSO) operation. In HSO mode of operation the PCA is configured such that a dedicated software bit can be pre-set and when a match occurs, the module reverses the logic level of its I/O pin.

The 16-bit Timer 2 of the PTRA can also be compared with the timers available on 16-bit microcomputers such as NEC 78312 and Intel's 8096, which have similar performance but only have two and one PWM outputs respectively. Even Motorola's 68HC11 offers capabilities similar to that of the PTRA of the SAB 80515/535 but it lacks a PWM output.

PWM Outputs

A microcontroller with PWM outputs becomes a useful component in a converter design (Figure 3) to control three phase motors. If the three phase voltage supplied by the converter is variable in frequency and amplitude (a function which can easily be performed by the PWM outputs of the PTRA unit) an efficient control of a three phase motor can be realized. This three phase switching control through a microcontroller will allow motors to have

—Long Operating Times
—High Speed
—Variable RPM
—Frequency Reversing Cycles
—Low Noise, Maintenance-Free Operation

The key role of a microcontroller in such an application is to provide PWM control signals which in effect could be used directly to control the converter.

The PWM outputs characterized by their variable duty cycle and fixed frequency waveforms can also be integrated to provide an approximation to analog outputs. Converting the PWM signal to an analog signal varies in difficulty, depending upon the requirements of the system. Simply, analog signals can be generated from PWM waveforms by using a simple RC or an active filter. Applications such as motor control and switching power supplies actually require a PWM signal, not a true analog one.

PWM and 3 Phase Motor Control

In the control of a 3-phase motor, the motor is driven by a power amplifier stage which is controlled by PWM signals as shown in the block diagram of Figure 3. The microcontroller in this converter diagram acts as a driver and generates the necessary signals to drive the power amplifier stage. The amplifier performs an important function in the converter by providing the correct amount of current and voltage needed to drive the motor. However, a continuous sinusoidal voltage applied to this amplifier results in operational inefficiency caused by the dissipation of large amounts of power in the output transistors. Therefore a pulsed voltage is applied to the motor, which at short intervals switches each motor winding on at full voltage, then switches it off. On and off times of the pulse sequence using PWM method are controlled by the microcontroller in such a way that the average value of all on-cycles represents a sine-wave shape. The running of a 3-phase motor requires a sinusoidal current flow in its windings and does not depend on the shape of the applied voltage. The following section discusses the synthesis of a sinusoidal voltage waveform from a PWM output.

Synthesis of a Sine-Waveform Using PWM

As shown in Figure 4, a sine-wave can be formed by a number of synthesis points—each represented by a PWM signal corresponding to the amplitude of the sampled point. If T_p is the time period of the PWM signal then the maximum amplitude (positive peak) of the sine-wave will be represented by a PWM signal which has a high-time = T_p. The high-time of the PWM signal can be varied by selecting different compare values (in compare mode of the PTRA) which corresponds to different amplitude levels in a sine-wave. If T_S is the time period of the synthesised sine-wave and S is the number of systhesis points for one full period of the sine-wave, then

$$T_S = S.T_p$$

It is essential that a timer unit must have the following features to generate PWM signals,

1. Compare registers.
2. PWM outputs (at least three needed for a 3-phase motor control).
3. Capability to auto-reload the timer (a feature which allows to vary the T_p).

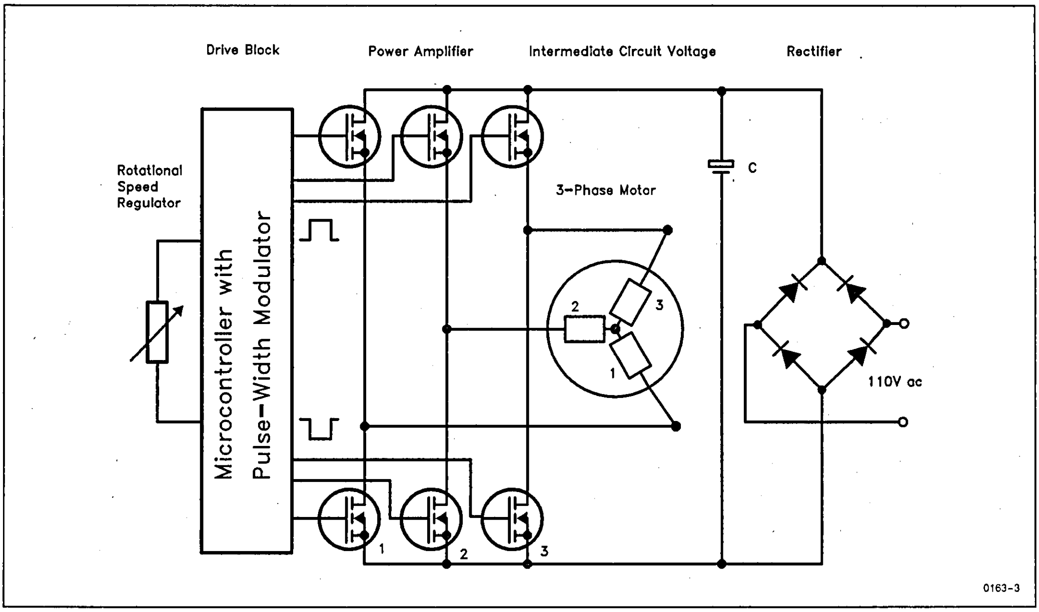

Drive Block
Power Amplifier
Intermediate Circuit Voltage
Rectifier
Rotational Speed Regulator
Microcontroller with Pulse-Width Modulator
3-Phase Motor
C
2
3
1
110V ac
1
2
3
0163-3

Figure 3. Circuit Diagram of a Converter for 3-Phase Motors

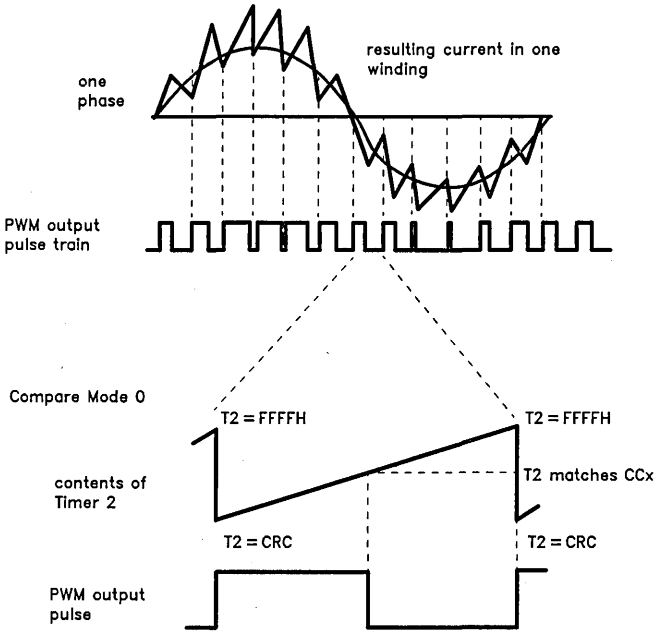


Figure 4. Pulse Width Modulation Using Timer 2 Cycle

The pulse width modulation of samples representing a sine-wave is shown in Figure 5. A high-time of 50% duty cycle represents an amplitude of zero magnitude, whereas an almost 100% high-time represents the positive peak and an almost 0% high-time represents the negative peak. In the SAB 80535/515 the maximum count rate of Timer 2 is 750 ns at 16 MHz oscillator frequency. The full count value (0FFFFH) minus the reload value of Timer 2 represents the T_p. For example, a reload value of 0FC00H will result in a T_p of 1024 counts (0FFFFH−0FC00H) or 1024 × 0.75 = 770.25 μs. The count of 1024 can also be interpreted as 1024 different compare values to represent 1024 different samples point in a given period of a sine-wave whereas T_p when expressed as a unit of time represents the PWM frequency. This would imply a positive peak could be represented by a pulse which will remain high for 770.25 μs. The full count value minus the reload value of Timer 2 is also referred to as the resolution of the PWM signal. As the reload value approaches the full count, the T_p count decreases, implying an increase in the PWM frequency but decrease in the resolution of the PWM signal. For example, using a reload value of 0FC00H in the PTRA unit of the SAB 80535/515 one can achieve a PWM frequency of 1.302 kHz with a resolution of 10 bits. Similarly, using the PTRA unit at 16 MHz oscillator frequency, one can achieve 8-bit resolution with a PWM frequency of 5.208 kHz, a 6-bit resolution with a PWM frequency of 20.83 kHz and a 5-bit resolution with a PWM frequency of 41.67 kHz. Clearly, there is a compromise between the PWM frequency and resolution of the PWM signal. One can also have more than one PWM pulse per sample point. If N is the number of equal pulses per synthesis point then

$$T_s = S.N.T_p$$

or $$f_s = f_p/S.N$$

where,

f_s = frequency of the synthesised sine-wave

f_p = frequency of the PWM signal

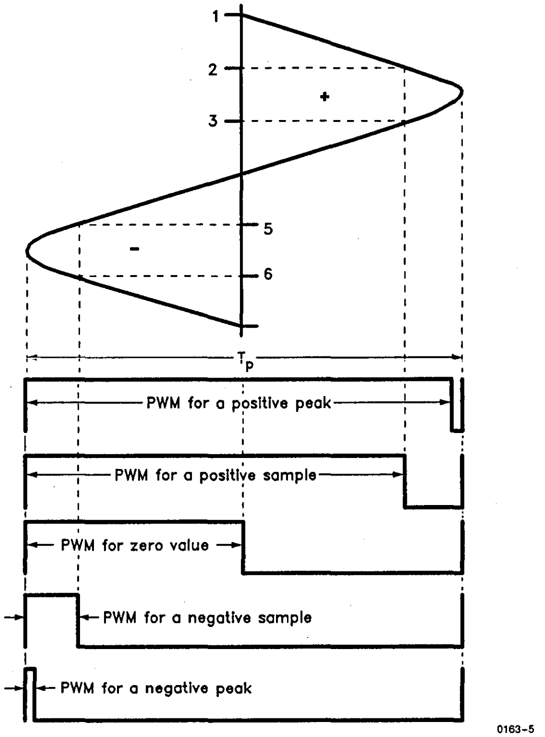


Figure 5. Synthesis of a Sine-Wave Using PWM

The PWM frequency depends upon the crystal oscillator frequency of the microcontroller. The number of synthesis points for each full sine-wave cycle must be a multiple of six as otherwise a 120 degree phase displacement cannot be achieved for a three phase operation. Based on these requirements, the Table 2 illustrates some values for f_S.

From Table 2 it is clear that a series of sine-waves with a wide range of frequencies can be synthesized using the PWM technique. However, an improper selection of the f_p could cause oscillations and audible "chirps" when a mechanical resonance is met at a particular frequency while controlling a 3-phase motor. As f_p increases the resolution of the PWM decreases and only fewer samples are available to synthesize a sine-wave. The maximum synthesized sine-wave frequency f_s is obtained at a high PWM frequency with fewer synthesis points per period and as the f_p decreases or the number of synthesis points increase the frequency of the synthesized sine-wave decreases. Clearly, continuous sinusoidal frequency variations require alterations of the PWM frequency—a job which the SAB 80535/515 can perform quite easily.

Conclusion

Besides performing the PWM support to control 3-phase motors, the SAB 80535/515 has additional on-chip features which makes it the ideal solution for many control applications. On-chip peripherals such as three timer/counters, 6 x 8-bit ports, A/D converter with programmable reference voltages, watchdog timer, baud-rate generator and a serial interface allow versatile applications in the field of automotive, industrial and consumer electronics.

References

1. Siemens SAB 80535/515 user's manual.
2. Frequency converter for driving asynchronous three-phase motors designed with new micro and power electronic components Siemens application note.

Table 2. Different Values for f_s

Timer 2 Reload Value	f_p in Hz	Clock in MHz	Timer 2 Count Rate	S	N	f_s in Hz
0FFC0H	20833	16	750 ns	18	1	1157.4
0FFC0H	20833	16	750 ns	18	2	578.69
0FFC0H	20833	16	750 ns	18	4	289.35
0FFC0H	20833	16	750 ns	54	1	385.80
●	●	●	●	●	●	●
●	●	●	●	●	●	●
0F000H	326	16	750 ns	12	1	27.12
0F000H	326	16	750 ns	12	2	13.56
●	●	●	●	●	●	●
●	●	●	●	●	●	●
0FFC0H	15625	12	1000 ns	24	1	651.04
0FFC0H	15625	12	1000 ns	24	2	325.52
0FFC0H	15625	12	1000 ns	54	1	289.35
●	●	●	●	●	●	●
0F000H	244	12	1000 ns	24	1	10.16

SIEMENS

February 1990

Implementation of the ISDN Oriented Modular (IOM) Interface Using the SAB 80515/80535 Microcontroller

SAB 80515/80535

Application Note

Introduction

This application note describes how any two port pins of the SAB 80515/535 microcontroller can be programmed to perform the data transmit and receive line functions of the ISDN Oriented Modular (IOM) interface.

Digitization of telephone networks not only increases transmission rates but allows more than one signal to be sent over a telephone line simultaneously. This capability to offer voice and a multiplicity of data services over one network with standard interfaces and call set-up procedures is the real driving force behind today's telecom/datacom revolution.

This one network is called ISDN—the Integrated Services Digital Network.

Integrated Telecommunication and data communication services together on one system.

Services Voice, data, text, picture, video, telex, facsimile, telemetry, alarms, and more.

Digital Digital transmission from end to end, with voice digitization handled at the terminal.

Network One worldwide network, based on existing public telephone lines, providing standard interfaces and call procedures familiar to everyone.

ISDN—Standards for Interconnectability

The international Telegraph and Telephone Consultative Committee (CCITT) has published a series of recommendations that have become the industry standard. These recommendations include ISDN definitions for two different groups of transmission speeds:

Basic Access Rate: 144 Kbits/s user information.

Basic access is configured as two 64 Kbits/s B-channels for voice or data transmission and one 16 Kbits/s D-channel for signaling or packet switched data. This is abbreviated as 2xB + D. It is used to connect voice/data terminals to either a private branch exchange (PBX) or directly to the public central office exchange (CO).

Primary Access Rate: 1.544 Mbits/s user information in USA and Japan.

Primary rates are used to transmit large amounts of data. The increased transmission rate helps to reduce the time of transmission and hence the cost of transmission. 64 Kbits/s channels are configured as 23xB + D in USA/Japan.

Siemens provides a number of dedicated solutions in the form of devices; system integration and software for ISDN. To make the interconnectivity of such ISDN devices possible, Siemens developed the ISDN Oriented Modular (IOM) architecture. The Siemens IOM (rev 2) architecture is fast becoming the de-facto standard for telecom designs. Many IC companies (Siemens, AMD, National and AT&T) now support this standard. Although a rich array of devices currently exists to support the bus, the Telecom market is expanding at such a rate that not all applications are covered. It is now Siemens' intent to have IOM interface available on all ISDN devices. Complete ISDN systems from terminals, terminal adapters, network terminators and transmission repeaters to line cards for digital exchange systems can be optimally designed using the IOM family of ICs.

ISDN Oriented Modular (IOM) Architecture

The IOM interface is a standard 4-wire local interface for the interconnection of ISDN-devices within the ISDN basic access. It consists of a receive and a transmit data line, an 8 KHz frame signal (FSC) and a 512 KHz data clock signal (DCL). The interface is tailor-made to fit the needs of the ISDN basic access. Typical applications shown in Figure 1 are:

In a repeater; two identical transmission devices are set back-to-back to achieve an increase in range.

In an NT (Network Termination); two different transmission devices are combined to convert between the S and U interface.

In the terminal (TE) and the exchange; the IOM interface interlinks the layer 1 and layer 2 device. In addition to the point to point configuration, the IOM-interface can support a point-multipoint configuration.

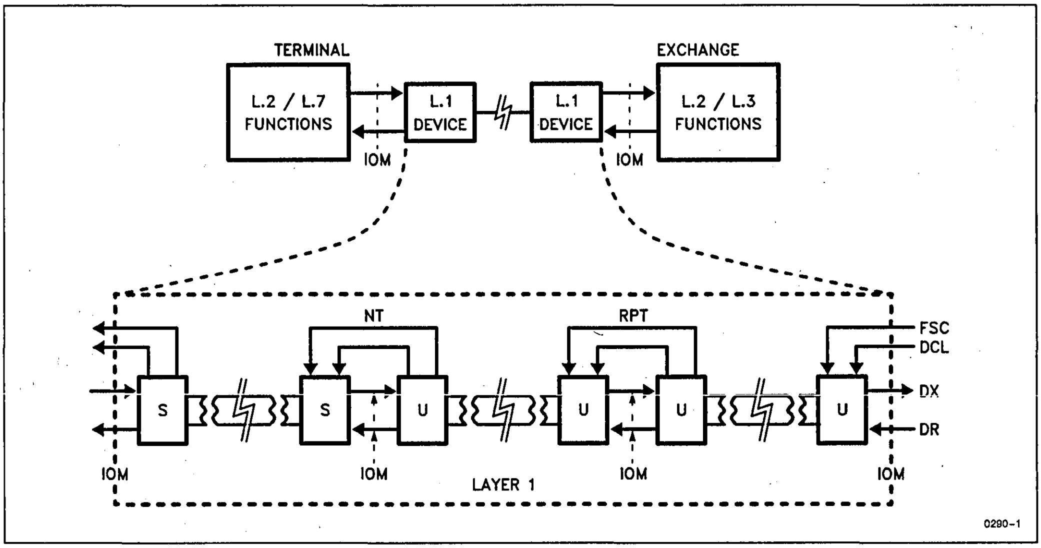
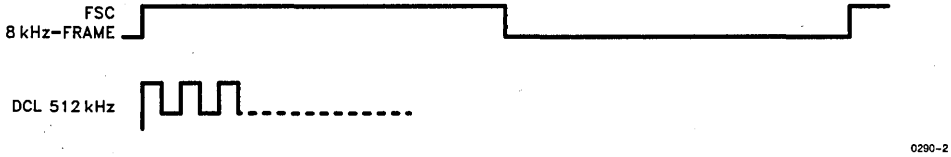


Figure 1. Application of the IOM-Interface in the Basic Access



IOM D_{IN}	B1	B2	Monitor	D	C/I
IOM D_{OUT}	B1	B2	Monitor	D	C/I

B1, B2: Circuit switched voice/data
D : D-channel
C/I : Command/Indicate

Figure 2

In all applications the ISDN data rate of 144 Kbits (2xB + 1xD) has to be transferred across the IOM interface transparently. In addition, there is a need for the exchange of control information for activating/deactivating layer 1 and for switching test loops. A few applications require further capacity to transfer maintenance information via this interface. All this information is transferred in a time-division multiplexed mode based on an 8 KHz frame structure (Figure 2) and assigned to the following four octets per frame and direction:

The 64 Kbits/sec channels B1 and B2 occupy the first two octets.

The third octet (monitor channel) is used for transferring the maintenance information.

The two D-channel-bits, the four Command/Indicate-bits (which control the activation/deactivation procedure), the T-bit and the E-bit are transferred in the fourth octet. The E-bit supports the handling of the monitor channel and the T-bit is reserved for a transparent 1 Kbit channel.

These four octets per frame constitute a bit rate of 256 Kbps. The transmission rate depends on one of the following modes:

Normal Mode:	Data Rate: 256 Kbps Bit Rate: 256 Kbps Clock Frequency: 512 KHz
MUX Mode:	Data Rate: 256 Kbps Bit Rate: 2.048 Mbps Clock Frequency: 4.096 MH

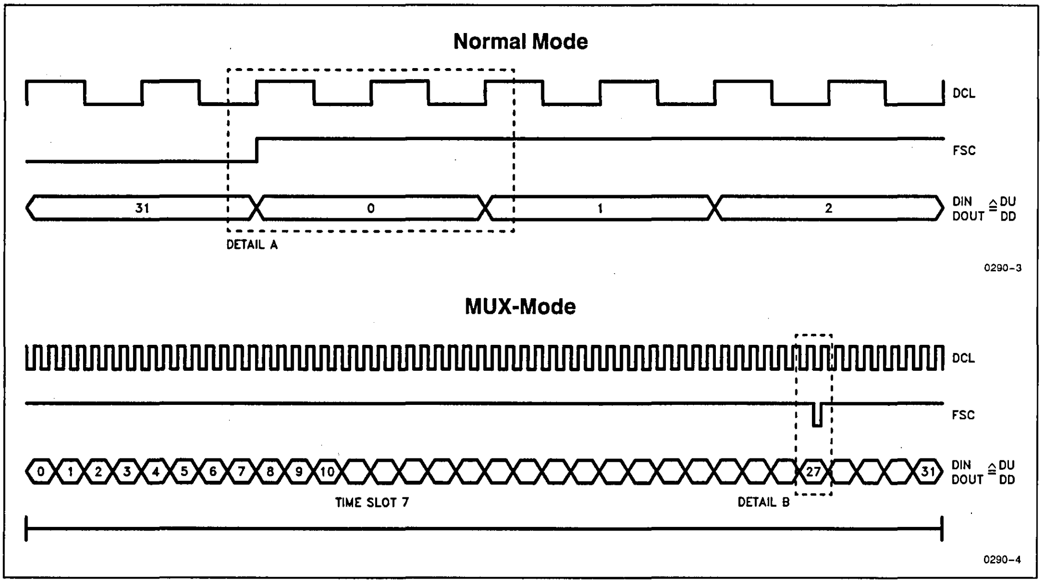


Figure 3. IOM Interface Timing

In the multiplex mode the outputs of up to eight layer 1 devices may be connected together to form an eight-time-slot IOM interface bus with a 2 MHz data rate. The physical timing of the IOM-interface in different modes is shown in Figure 3.

SAB 80515/535 Role

The primary role of the SAB 80515/535 microcontroller in generating the IOM interface in normal mode is to accept the frame synchronization (FSC) signal as an input and based on this signal identify the four octets on the receive and transmit data lines. Any two port pins of the SAB 80515/535 can be used as the transmit and receive data lines. The PTRA unit of the SAB 80515/535 plays a major role not only in defining the octets but also in generating and receiving the data bit stream over the transmit and receive lines in synchronization with the frame and data clock signals. In this respect the SAB 80515/535 falls into a unique family of microcontrollers which can perform this function.

PTRA Unit of the SAB 80515/535

The Programmable Timer/Counter Register Array (PTRA) of the SAB 80535/515 has a time-base which is a programmable 16-bit timer/counter also known as the Timer 2. Timer 2 is the only one that can serve the PTRA.

The PTRA consists of the following registers:

T2CON	Timer 2 control register
TL2	Timer 2 register, low-byte
TH2	Timer 2 register, high-byte
CRCL	Compare/reload/capture register, low-byte
CRCH	Compare/reload/capture register, high-byte
CCL1	Compare/capture register 1, low-byte
CCH1	Compare/capture register 1, high-byte
CCL2	Compare/capture register 2, low-byte
CCH2	Compare/capture register 2, high-byte
CCL3	Compare/capture register 3, low-byte
CCH3	Compare/capture register 3, high-byte
CCEN	Compare/capture register enable register

For brevity, the double-byte Compare/Reload/Capture register is called CRC register, and the three double-byte Compare/Capture registers are called CC registers 1 to 3. The PTRA shares Port 1 pins for hardware interfacing as shown in Table 1.

Table 1. Hardware Interfacing for the PTRA Unit

Port Pin	Name	Function
P1.0	INT3/CC0	Compare Output/Capture Input for the CRC Register
P1.1	INT4/CC1	Compare Output/Capture Input for CC Register 1
P1.2	INT5/CC2	Compare Output/Capture Input for CC Register 2
P1.3	INT6/CC3	Compare Output/Capture Input for CC Register 3
P1.5	T2EX	External Reload Trigger Input
P1.7	T2	External Count or Gate Input to Timer 2

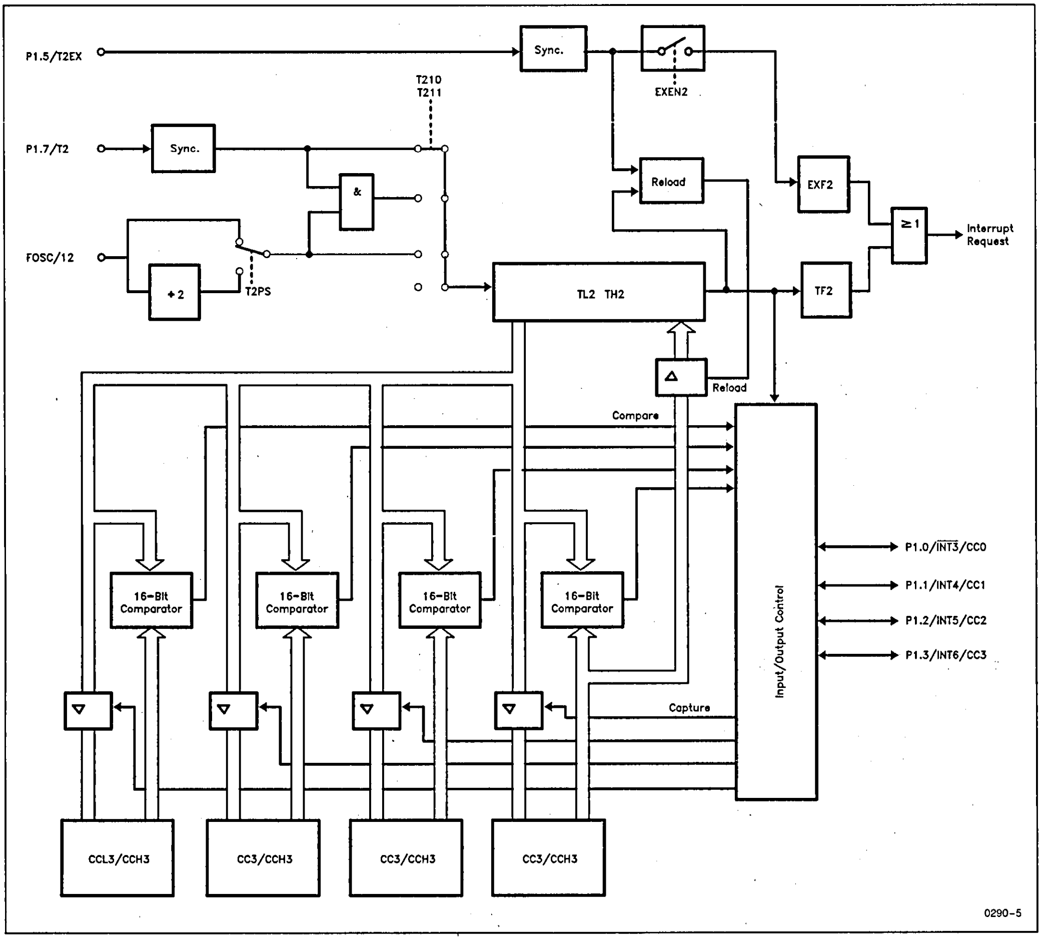


Figure 4. Block Diagram of the PTRA Unit

In addition to supporting the operational modes of a timer or a counter, the PTRA provides the following features for the Timer 2.

—16-bit reload
—16-bit compare
—16-bit capture

Figure 4 shows a block diagram of the PTRA unit.

The Timer 2 of PTRA can operate either as timer, event counter, or gated timer. In timer mode, the count rate is derived from the oscillator frequency. A 2:1 pre-scaler makes it possible to select a count rate of 1/12 or 1/24 of the oscillator frequency. In either case, no matter whether Timer 2 is configured as timer, event counter, or gated timer; when the count rolls over from all 1's to all 0's it sets the Timer 2 overflow flag which can generate an interrupt. In counter mode, the Timer 2 register is incremented in response to a 1-to-0 transition at its corresponding external input pin T2 (P1.7).

The reload mode of the PTRA unit allows contents of Timer 2 to be reloaded from the CRC register at the time when Timer 2 rolls over from all 1's to all 0's. The 16-bit reload from the CRC register to Timer 2 registers can also be caused by a negative transition at port pin P1.5.

In the compare mode, the 16-bit values stored in the dedicated compare registers of the PTRA are compared with the contents of the Timer 2 registers TL2 and TH2. If the count value in the Timer 2 register matches the stored value an appropriate output signal is generated at the corresponding port 1 pin, and an interrupt can also be enabled.

In the PTRA compare mode-0, the output signals at port pins P1.1 through P1.3 change from low to high upon a match of the corresponding CC1 through CC3 register contents with contents of the Timer 2 registers. The outputs then go back to a low level on Timer 2 overflow, and Timer 2 restarts by reloading the CRC register contents. In the second mode of operation the output port pins P1.0 through P1.3 can be caused to toggle at compare events, thereby resulting in the generation of square waves at these outputs. In this mode the Timer 2 overflow has no affect on the output port pins. Each of the three compare/capture and the CRC registers of the PTRA unit can be used to latch the current 16-bit value of the timer 2 in one of them. This latching could either be caused by an external event or upon writing to the low-order byte of the dedicated 16-bit capture register. Either a rising or a falling edge can be selected to cause the capture from Timer 2 into the CRC register, whereas for the CC registers 1 to 3 a positive transition causes the capture.

IOM Interface Using the SAB 80515/535

The high speed of the SAB 80515/535 microcontroller allows its machine cycle (1/12 of oscillator frequency) to be twice as fast as the DCL which means the microcontroller can easily handle two DCLs per data bit. To achieve this, the crystal oscillator frequency is selected at 24 times the data clock i.e., 24x512 KH = 12.288 MHz. As a result, the Timer 2, which is programmed to operate in mode 1, also counts as twice the rate of the data clock. Timer 2 is programmed to increment at the rate of 1024 KHz as opposed to counting at the data clock rate of 512 KHz. Though the latter could be accomplished by connecting the data clock to port pin P1.7 of the microcontroller and letting the Timer 2 count in the external count mode. But this is not desirable as the count rate of 1024 KHz gives a higher resolution for the compare feature. However, the synchronization is achieved by programming the Timer 2 to count at a value of 0FF80H every time the falling edge of the FSC (Frame Synchronization) signal is detected at port pin P1.5. The falling edge of the FSC signal coincides with the occurrence of the first data bit in octet 3 (monitor channel).

With 0FF80H being the start value of Timer 2 which also marks the beginning of third octet, the other octets occur 32 counts apart. This derives from the fact that there are two DCLs per data bit (4 machine cycles or 4 counts per data bit) and there are 8 data bits per octet. Based on this analogy, the octets 3, 4, 1 and 2 occur at the Timer 2 count values of 0FF80H, 0FFA0H, 0FFC0H and 0FFE0H respectively as shown in Figure 5. In other words, if the compare interrupt was invoked at these count values, the user will be in the desired octet to transmit or receive a data bit stream. However, the delay involved in processing the interrupt will also delay the occurrence of a dat bit at the desired DCL. To understand the compare interrupt handling in the SAB 80515/535, one will first have to understand the basic machine cycle timings of the SAB 80515/535.

A machine cycle consists of 6 states (12 oscillator periods). Each state is divided into two phases—phase 1 and phase 2. Thus a machine cycle consists of 12 phases or 12 oscillator periods, numbered S1P1 (state 1, phase 1) through S6P2 (state 6, phase 2). Each phase lasts or one oscillator period and each state lasts for two oscillator periods. Timer 2 is incremented in S1P1 of every machine cycle. An internal compare signal is generated in S2P2, if Timer 2 increments to the compare count in one of the compare registers CC1 through CC3. The compare signal is active as long as the Timer 2 contents are equal to one of the compare registers. For the compare registers CC1 to CC3 an interrupt is always requested and a flag is set when the compare signal goes active. The interrupt flags are sampled in S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition in S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine. As shown in Figure 6 a standard interrupt handling procedure can take about 4 machine cycles. For exceptions please refer to the SAB 80515/535 user's manual.

Based on above facts, in the present application, two DCLs (4 machine cycles) have already elapsed before the program enters into the compare subroutine. To avoid this delay, the compare interrupt subroutine is invoked 30 machine cycles or 15 DCLs before the desired octet. Therefore the compare event for different octets occur at the following count values:

Octet 1, count value = 0FFA2H
Octet 2, count value = 0FFC2H
Octet 3, count value = 0FFE2H
Octet 4, count value = 0FF82H

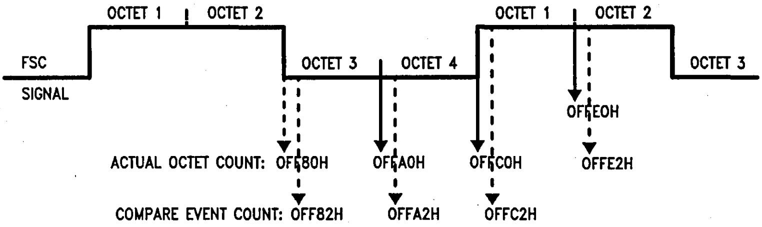


Figure 5. Octet Placement in FSC Signal

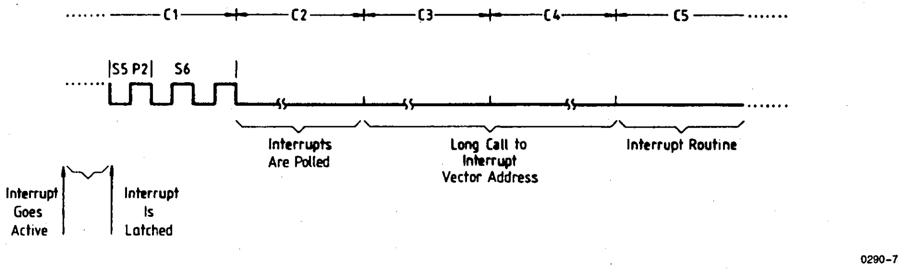


Figure 6. Interrupt Response Timing Diagram

In either transmit or receive mode, when compare interrupt occurs at one of the above count values, the microcontroller performs no-operation until 30 machine cycles elapse. At that instant, the microcontroller sends out or receives the data bit stream as that is the beginning of the desired octet.

System Implementation

The program listing (refer to Appendix A) has two subroutines—one for transmit and the other for receive mode. In either case the data variable "count" holds the lower byte of the desired octet to perform the compare match. In transmit mode the data is taken from external memory and in the receive mode the data is stored in the external memory starting at address 000H. The total time spent in the compare interrupt subroutine to transmit or receive a byte varies from 75 to 82 machine cycles. This variation is caused due to the time spent in deciding whether or not the byte transmitted or received is the last byte. The next byte is not transmitted or received until the next octet in subsequent frame i.e., 128 machine cycles later. Therefore the CPU is free for 128 machine cycles minus the time spent in machine cycles in the compare interrupt subroutine. However more time could be made available for the CPU by cutting down the number of NOPs in the compare interrupt subroutine and by increasing the compare count to compensate for the number of NOPs removed. In that case the compare event could occur as close as 4 machine cycles before the desired octet count.

Any two port pins of the SAB 80515/535 can be used as the data transmit and data receive lines. In this application port pin P3.0 is used as the receive data line and the port pin P3.1 is used as the transmit data line. Figure 7 shows the basic SAB 80535 circuit required to run the subroutines. The pins used for the IOM interface are also shown. The other on-chip peripherals namely the serial port, Timer 0 and Timer 1, the watchdog timer, the A/D converter and 30 I/O lines are still available to perform any other system functions.

Figure 8 shows an IDSN application where the SAB 80515/535 is being used as a terminal interfaced to an ISDN network simulator over the S-interface.

References

1. Siemens, 8-Bit Single-Chip Microcontroller Handbook, 1989/90.
2. Siemens, Telecommunications Data Book, 1987.
3. Total ISDN Commitment, a Siemens Brochure.

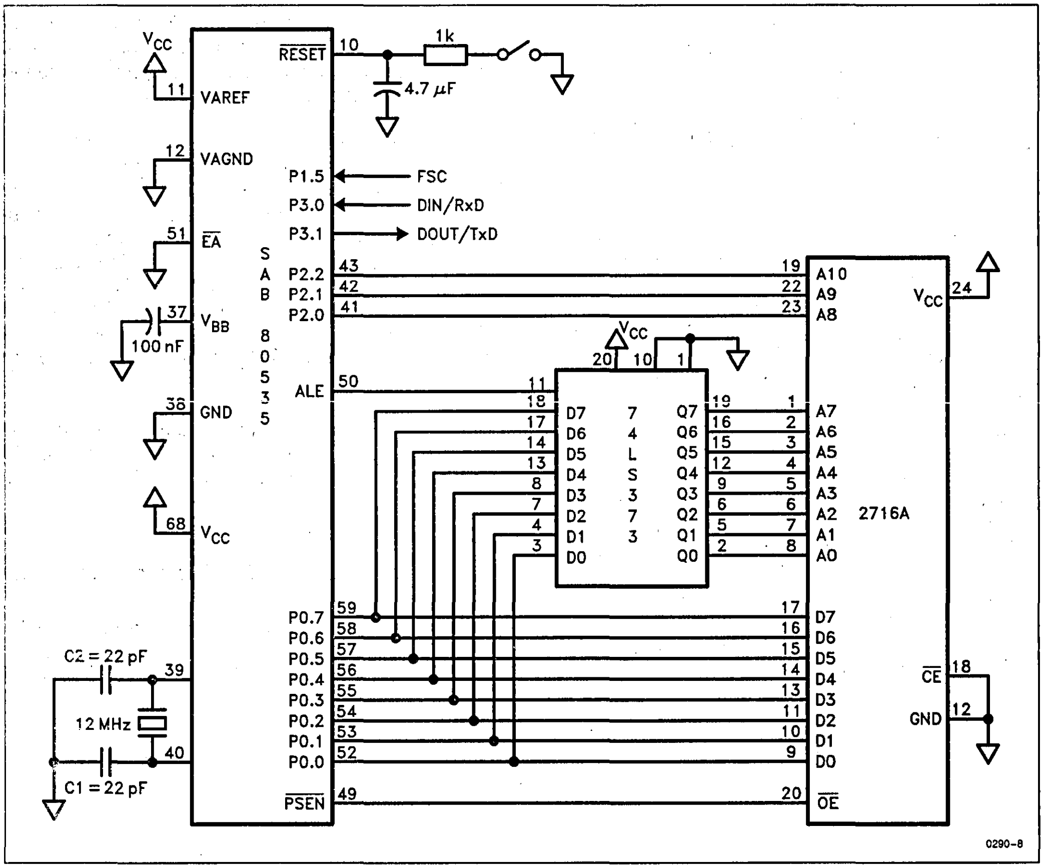


Figure 7. SAB 80535 Schematic Diagram

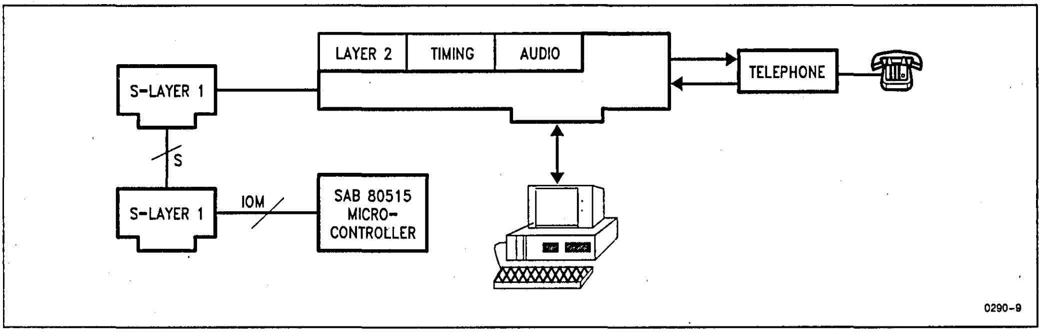


Figure 8. NT—Simulator Application Using the SAB 80515/5353 as a Terminal

```
               $Mod515
               $Pagelength(70)
               $Debug
               $Title (IOM Interface     Appendix A)



               ;*************************************************************
               ;*   IOM Interface Program:                                  *
               ;*   In the following program the variable "Count"           *
               ;*   count in which the data byte is to be transmitted.      *
               ;*   The valid octet counts are: for octet 1, count =        *
               ;*   0A2H; octet 2, count = 0C2H; octet 3, count = 0E2H;     *
               ;*   octet 4, count = 82H. These bytes are stored in the     *
               ;*   external data memory. The bytes appear in the           *
               ;*   desired octet in every subsequent frame until           *
               ;*   the last byte is transmitted. The external              *
               ;*   oscillator frequency is 12.288 MHz (24 x 512 KHz).      *
               ;*************************************************************


 00A2              Count  EQU     0A2H         ;Octet 1=A2,2=C2,3=E2,4=82.

0000               Cseg   at      0H

0000 020100               LJMP    Main_Begin

               ;*************************************************************
               ;*   Write_IOM_Subroutine:                                   *
               ;*   The following subroutine happens 30 machine cycles      *
               ;*   (15 DCLs) before the occurrence of the desired          *
               ;*   octet. The data byte is output on port bit P3.1         *
               ;*   coinciding with the occurrence of the desired           *
               ;*   octet. The data byte is taken from external             *
               ;*   memory and the data pointer to external                 *
               ;*   memory is incremented after a byte is transmitted       *
               ;*   over the IOM interface. The subroutine is               *
               ;*   exited before the arrival of the falling edge           *
               ;*   of the FSC at port pin P1.5.                            *
               ;*************************************************************
005B                      Org     5BH          ;Compare 1 interrupt routine.

005B           Write_IOM_Subroutine:           ;4 m/c for servicing the
                                                routine.
005B C2BB                 CLR     IEN1.3       ;7th m/c into the frame.

005D 00                   NOP                  ;8th m/c into the frame.
005E 00                   NOP                  ;9th m/c into the frame.
005F 00                   NOP                  ;10th m/c into the frame.
0060 00                   NOP                  ;11th m/c into the frame.
0061 00                   NOP                  ;12th m/c into the frame.
0062 00                   NOP                  ;13th m/c into the frame.
0063 00                   NOP                  ;14th m/c into the frame.
0064 00                   NOP                  ;15th m/c into the frame.
0065 00                   NOP                  ;16th m/c into the frame.
0066 00                   NOP                  ;17th m/c into the frame.
0067 00                   NOP                  ;18th m/c into the frame.
```

```
0068 00         NOP                       ;19th m/c into the frame.
0069 00         NOP                       ;20th m/c into the frame.
006A 00         NOP                       ;21th m/c into the frame.
006B 00         NOP                       ;22th m/c into the frame.
006C 00         NOP                       ;23th m/c into the frame.
006D 00         NOP                       ;24th m/c into the frame.
006E 00         NOP                       ;25th m/c into the frame.
006F 00         NOP                       ;26th m/c into the frame.
0070 00         NOP                       ;27th m/c into the frame.
0071 00         NOP                       ;28th m/c into the frame.

0072 13         RRC   A                   ;29th m/c into the frame.
0073 00         NOP                       ;30th m/c into the frame.
0074 92B1       MOV   P3.1,C              ;Two more cycles.

0076 13         RRC   A                   ;New octet, first bit arrives.
0077 00         NOP                       ;34th cycle.
0078 92B1       MOV   P3.1,C              ;Second bit sent.

007A 13         RRC   A                   ;Third bit to carry flag.
007B 00         NOP                       ;38th cycle.
007C 92B1       MOV   P3.1,C              ;Third bit sent

007E 13         RRC   A                   ;Fourth bit to carry flag.
007F 00         NOP                       ;42nd cycle.
0080 92B1       MOV   P3.1,C              ;Fourth bit sent.

0082 13         RRC   A                   ;Fifth bit to carry flag.
0083 00         NOP                       ;46th cycle.
0084 92B1       MOV   P3.1,C              ;Fifth bit sent.

0086 13         RRC   A                   ;Sixth bit to carry flag.
0087 00         NOP                       ;50th cycle.
0088 92B1       MOV   P3.1,C              ;Sixth bit sent.

008A 13         RRC   A                   ;Seventh bit to carry flag.
008B 00         NOP                       ;54th cycle.
008C 92B1       MOV   P3.1,C              ;Seventh bit sent.

008E 13         RRC   A                   ;Eighth bit to carry flag.
008F 00         NOP                       ;58th cycle.
0090 92B1       MOV   P3.1,C              ;Eighth bit sent.

0092 00         NOP                       ;122th DCL (61st cycle).
0093 00         NOP                       ;62nd cycle.
0094 00         NOP                       ;63rd cycle.
0095 D2B1       SETB  P3.1                ;32nd DCL (64th cycle).

0097 A3         INC   DPTR                ;Increment the data pointer.
0098 E0         MOVX  A,@DPTR             ;Load the acc. with next byte.
0099 A883       MOV   R0,DPH              ;To compare if end of array
009B B8FF08     CJNE  R0,#0FFH,CONTI      ;Compare the high byte first.
009E A882       MOV   R0,DPL              ;Load the lower byte for comparison.
00A0 B8FF03     CJNE  R0,#0FFH,CONTI      ;Compare the lower byte.

00A3 900000     MOV   DPTR,#00H           ;Initialize the data pointer.

```

```
00A6          CONTI:
00A6 C2C3                CLR  IRCON.3
00A8 D2BB                SETB IEN1.3
00AA 32                  RETI



              ;*********************************************************
              ;*   Write_IOM_Set-Up:                                   *
              ;*   In this program the registers are set up to         *
              ;*   allow compare to happen at the "count" value. The   *
              ;*   PTRA unit of the SAB 80535 is programmed in the     *
              ;*   compare mode 1. The "count" value is assigned       *
              ;*   to the compare register 1 and the corresponding     *
              ;*   interrupt is enabled. The Timer 2 of the PTRA       *
              ;*   unit is programmed to restart at a value of FF80H   *
              ;*   on the arrival of the falling edge of the           *
              ;*   FSC signal.                                         *
              ;*********************************************************

0100                     ORG  100H
              Main_Begin:
0100 900000              MOV  DPTR,#00H      ;DPTR = (Start of data array).
0103 75C100              MOV  CCEN,#00H      ;Disable compare mode.
0106 75B0FF              MOV  P3,#0FFH       ;SDO, SDI lines are high.

0109 75C3FF              MOV  CCH1,#0FFH     ;Compare register has a value
010C 75C2A2              MOV  CCL1,#Count    ;corresponding to the octet desired.
010F 75CBFF              MOV  CRCH,#0FFH
0112 75CA80              MOV  CRCL,#080H     ;Reload = 0FFFH - 128 (32x4)
0115 75CDFF              MOV  TH2,#0FFH
0118 75CC81              MOV  TL2,#81H       ;To avoid compare at FF80H
011B D2CA                SETB T2CM           ;T2 in compare mode 1
011D 75C108              MOV  CCEN,#08H      ;Compare mode enabled
0120 D2F                 SETB EAL            ;Master interrupt Bit enabled.

0122 758910              MOV  TMOD,#10H      ;Timer1 to cause more than
0125 758DFF              MOV  TH1,#0FFH      ;quarter of a frame delay
0128 758BD0              MOV  TL1,#256-48    ;to avoid match before reload.

012B          Start:
012B 2096FD              JB   P1.6,$         ;Prepare for start pulse arrival.
012E 3096FD              JNB  P1.6,$         ;Wait for the start pulse, 1 m/c.
0131 D28E                SETB TR1            ;So far 2 m/c into the frame.
0133 308FFD              JNB  TF1,$          ;Wait for another 49 m/c

0136 75C81D              MOV  T2CON,#1DH     ;T2 auto reload from P1.5.
0139 D2BB                SETB IEN1.3         ;Compare interrupt 1 enable.

013B 213B                AJMP $              ;Wait for a match.

                         end

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

```
                $Mod515
                $Pagelength(73)
                $Debug
                $Title (IOM Interface     Appendix A)



                ;***************************************************************
                ;*   IOM Interface Program:                                    *
                ;*   In the following program the variable "Count"             *
                ;*   holds the octet count from which the data byte is to be   *
                ;*   received. The valid octet counts are: for octet 1,        *
                ;*   count = 0A2H; octet 2, count = 0C2H; octet 3,             *
                ;*   count = 0E2H; octet 4, count = 82H. These bytes are       *
                ;*   stored in the external data memory. The bytes are read    *
                ;*   from the desired octet in subsequent frames until the     *
                ;*   last byte is transmitted. The external crystal            *
                ;*   oscillator frequency is 12.288 MHz (24 x 512 KHz).        *
                ;***************************************************************


  00A2              Count  EQU     0A2H       ;Octet 1=A2,2=C2,3=E2,4=82.

0000                Cseg   at      0H

0000 020100                LJMP    Main_Begin

                ;***************************************************************
                ;*   Read_IOM_Subroutine:                                      *
                ;*   The following subroutine happens 30 machine cycles        *
                ;*   (15 DCLs) before the occurrence of the desired            *
                ;*   octet. The data byte is read from port bit                *
                ;*   P3.0, 3-4 DCLs after the occurrence of the desired        *
                ;*   octet. The data byte is stored in external memory         *
                ;*   and the data pointer to external memory                   *
                ;*   is incremented after the byte is received                 *
                ;*   over the IOM interface. The subroutine is                 *
                ;*   exited before the next compare event.                     *
                ;***************************************************************
005B                       Org     5BH        ;Compare 1 interrupt routine.

005B            Write_IOM_Subroutine:         ;4 m/c for servicing the routine.

005B C2BB                  CLR     IEN1.3     ;7th m/c into the frame.

005D 00                    NOP                ;8th m/c into the frame.
005E 00                    NOP                ;9th m/c into the frame.
005F 00                    NOP                ;10th m/c into the frame.
0060 00                    NOP                ;11th m/c into the frame.
0061 00                    NOP                ;12th m/c into the frame.
0062 00                    NOP                ;13th m/c into the frame.
0063 00                    NOP                ;14th m/c into the frame.
0064 00                    NOP                ;15th m/c into the frame.
0065 00                    NOP                ;16th m/c into the frame.
0066 00                    NOP                ;17th m/c into the frame.
0067 00                    NOP                ;18th m/c into the frame.
0068 00                    NOP                ;19th m/c into the frame.
```

```
0069 00        NOP                       ;20th m/c into the frame.
006A 00        NOP                       ;21th m/c into the frame.
006B 00        NOP                       ;22th m/c into the frame.
006C 00        NOP                       ;23th m/c into the frame.
006D 00        NOP                       ;24th m/c into the frame.
006E 00        NOP                       ;25th m/c into the frame.
006F 00        NOP                       ;26th m/c into the frame.
0070 00        NOP                       ;27th m/c into the frame.
0071 00        NOP                       ;28th m/c into the frame.
0072 00        NOP                       ;29th m/c into the frame.
0073 00        NOP                       ;30th m/c into the frame.
0074 00        NOP                       ;31th m/c into the frame.
0075 00        NOP                       ;32th m/c into the frame.

0076 00        NOP                       ;New octet.
0077 A2B0      MOV    C,P3.0             ;First bit read.
0079 13        RRC    A                  ;First bit into the acc.
007A 00        NOP                       ;36th m/c inth the frame

007B 00        NOP                       ;37th m/c into the frame.
007C A2B0      MOV    C,P3.0             ;Second bit read.
007E 13        RRC    A                  ;Second bit into the acc.
007F 00        NOP                       ;40th m/c into the frame.

0080 00        NOP                       ;41st m/c into the frame.
0081 A2B0      MOV    C,P3.0             ;Third bit read.
0083 13        RRC    A                  ;Third bit into the acc.
0084 00        NOP                       ;44th cycle.

0085 00        NOP                       ;45th m/c into the frame.
0086 A2B0      MOV    C,P3.0             ;Fourth bit read.
0088 13        RRC    A                  ;Fourth bit into the acc.
0089 00        NOP                       ;48th machine cycle.

008A 00        NOP                       ;49th m/c into the frame.
008B A2B0      MOV    C,P3.0             ;Fifth bit read.
008D 13        RRC    A                  ;Fifth bit into the acc.
008E 00        NOP                       ;52nd cycle.

008F 00        NOP                       ;53rd machine cycle.
0090 A2B0      MOV    C,P3.0             ;Sixth bit read.
0092 13        RRC    A                  ;Sixth bit into the acc.
0093 00        NOP                       ;56th machine cycle.

0094 00        NOP                       ;57th machine cycle.
0095 A2B0      MOV    C,P3.0             ;Seventh bit read.
0097 13        RRC    A                  ;Seventh bit into the acc.
0098 00        NOP                       ;60th machine cycle.

0099 00        NOP                       ;61st machine cycle.
009A A2B0      MOV    C,P3.0             ;Eighth bit read.
009C 13        RRC    A                  ;Eighth bit into the acc.

009D A3        INC    DPTR               ;Increment the data pointer.
009E F0        MOVX   @DPTR,A            ;Save the acc. at the next byte.
009F A883      MOV    R0,DPH             ;To compare if end of array
00A1 B8FF08    CJNE   R0,#0FFH,CONTI     ;Compare the high byte first.
00A4 A882      MOV    R0,DPL             ;Load the lower byte for comparison.
```

```
00A6 B8FF03         CJNE   R0,#0FFH,CONTI  ;Compare the lower byte.

00A9 900000         MOV    DPTR,#00H       ;Initialize the data pointer.

00AC        CONTI:
00AC C2C3           CLR    IRCON.3
00AE D2BB           SETB   IEN1.3
00B0 32             RETI



            ;********************************************************
            ;*   Read_IOM_Set-Up:                                    *
            ;*   In this program the registers are set up to         *
            ;*   allow compare to happen at the "count" value. The   *
            ;*   PTRA unit of the SAB 80535 is programmed in the     *
            ;*   compare mode 1. The "count" value is assigned       *
            ;*   to the compare register 1 and the corresponding     *
            ;*   interrupt is enabled. The Timer 2 of the PTRA       *
            ;*   unit is programmed to restart at a value of FF80H   *
            ;*   on the arrival of the falling edge of the           *
            ;*   FSC signal.                                         *
            ;********************************************************

0100                ORG    100H
0100        Main_Begin:
0100 900000         MOV    DPTR,#00H       ;DPTR = (Start of data array).
0103 75C100         MOV    CCEN,#00H       ;Disable compare mode.
0106 75B0FF         MOV    P3,#0FFH        ;SDO, SDI lines are high.

0109 75C3FF         MOV    CCH1,#0FFH      ;Compare register has a value
010C 75C2A2         MOV    CCL1,#Count     ;corresponding to the octet
desired.
010F 75CBFF         MOV    CRCH,#0FFH
0112 75CA80         MOV    CRCL,#080H      ;Reload = 0FFFFH - 128 (32x4)
0115 75CDFF         MOV    TH2,#0FFH
0118 75CC81         MOV    TL2,#81H        ;To avoid compare at FF80H
011B D2CA           SETB   T2CM            ;T2 in compare mode 1
011D 75C108         MOV    CCEN,#08H       ;Compoare mode enabled
0120 D2AF           SETB   EAL             ;Master interrupt Bit enabled.

0122 758910         MOV    TMOD,#10H       ;Timer1 to cause more than
0125 758DFF         MOV    TH1,#0FFH       ;quarter of a frame delay
0128 758BD0         MOV    TL1,#256-48     ;to avoid match before reload.

012B        Start:
012B 2096FD         JB     P1.6,$          ;Prepare for start pulse arrival.
012E 3096FD         JNB    P1.6,$          ;Wait for the start pulse, 1 m/c.
0131 D28E           SETB   TR1             ;So far 2 m/c into the frame.
0133 308FFD         JNB    TF1,$           ;Wait for another 49 m/c

0136 75C81D         MOV    T2CON,#1DH      ;T2 auto reload from P1.5.
0139 D2BB           SETB   IEN1.3          ;Compare interrupt 1 enable.

013B 213B           AJMP   $               ;Wait for a match.

            end
ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

February 1990

Memory Access Using the Eight Data Pointers of the SAB 80C517/80C537

SAB 80C517/80C537

Application Brief

Memory Access Using the Eight Data Pointers of the SAB 80C517/80C537

The SAB 80C517/80C537 has eight 16-bit data pointers (DPTR) as opposed to one in other derivatives of the SAB 8051 family of microcontrollers. This allows a fast and an efficient transfer of data bytes to, from and between program and external data memory spaces. An application of this feature and the resulting increase in processing power is illustrated in the following example. A comparison is made between a block transfer of data using the 8051 and the 80C517.

The microcontrollers based on the 8051 architecture have the capability to expand their memory addressing by going to external program and data memory spaces. This feature is quite useful in applications that require more memory space than furnished by the on-board ROM and RAM. However, data in the program memory and data in the external memory can only be accessed through indirect addressing using a 16-bit register called the Data Pointer (DPTR). The following instructions are used to access data or memory mapped peripheral addresses using the Data Pointer:

```
MOVC  A,@A+DPTR    ;Read from Program
                    memory
MOVX  A,@DPTR      ;Read external Data
                    memory
MOVX  @DPTR,A      ;Write external Data
                    memory
```

Almost all microcontrollers based on the 8051 architecture have only one such data pointer. This puts a heavy toll on data transfers requiring access to blocks of data stored at different addresses within the memory space. For example, in order to transfer a block of data from one address to another, one will first have to save the source address before using the Data Pointer for the destination address and vice versa. This operation not only slows down the data access as a certain amount of time is spent in saving and retrieving the data pointer but also uses the valuable internal RAM memory space to save the Data Pointer. The limitation of the 8051 architecture with one on-board Data Pointer is soon realized when the application requires multiple data transfers in the shortest possible time or when applications require access to a number of external peripherals and/or data memory locations. To some extent this limitation is eliminated in the SAB 80C517/80C537 by providing eight on-chip Data Pointers.

In order to keep the compatibility of the SAB 80C517/80C537 with the 8051, whose instruction set allows the handling of one single 16-bit Data Pointer only, it is not possible to add eight Data Pointers with eight different addresses. Instead all eight Data Pointers of the SAB 80C517/80C537 have the address 83H for the High-Byte (DPH) and 82H for the Low-Byte (DPL). Although the user's program may store up to eight different 16-bit addresses in these eight Data Pointer registers, only one register at any given time is active as a Data Pointer. The Data Pointer in use is selected by another special function register called DPSEL (**D**ata **P**ointer **SEL**ect register). The bit 0 to 2 of this register DPSEL selects one of the eight Data Pointers. The selected Data Pointer can then be loaded, read and manipulated in the usual manner. It can also be used with MOVC/MOVX types of instructions. Only the selected Data Pointer gets affected while the other seven Data Pointers remain unchanged. However, to activate any other Data Pointer, it is only sufficient to reprogram the lower three bits of the DPSEL register. This implementation guarantees the complete compatibility with the 8051 architecture for the Data Pointer in use.

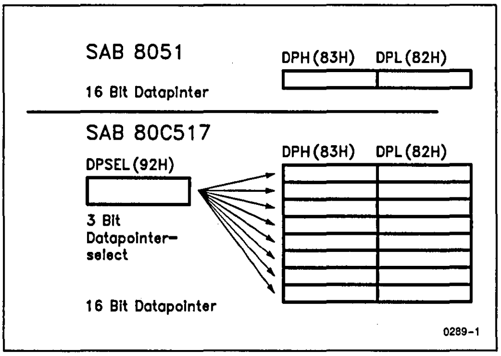


Figure 1. Data Pointer Register in the 8051-Type Microcontroller vs. the Data Pointer Array in the SAB 80C517/80C537

The following programming example demonstrates the increase in processing power that results from using the additional Data Pointers of the SAB 80C517/80C537. This program copies a block of data (e.g., a set of parameters) from the program memory (ROM) to the external data memory (RAM).

Transfer Program for the 8051

```
;DECLARATION OF THE ARRAYS
;------------------------------------------------------------
     CSEG AT 800H     ;ROM!
PAR_ARRAY: DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
           DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
                        ;.....etc.
           DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

END_OF_ARRAY   EQU $

     XSEG AT 0000H    ;EXTERNAL RAM!
BUFFER:        DS 512
;------------------------------------------------------------
DECLARATION OF THE SHADOW DATAPOINTERS
     DSEG AT 40H
SHADOW_DPL_0;   DS 1
SHADOW_DPH_0;   DS 1

SHADOW_DPL_1;   DS 1
SHADOW_DPH_1;   DS 1
;------------------------------------------------------------
     CSEG AT 00H
;LOAD THE SHADOW DATAPOINTERS
     MOV  SHADOW_DPL_0,#LOW(PAR_ARRAY)
     MOV  SHADOW_DPH_0,#HIGH(PAR_ARRAY)

     MOV  SHADOW_DPL_1,#LOW(BUFFER)
     MOV  SHADOW_DPH_1,#HIGH(BUFFER)
;------------------------------------------------------------
LOOP:MOV  DPL,SHADOW_DPL_0 ;LOAD THE ACTUAL DPTR WITH
     MOV  DPH,SHADOW_DPH_0 ;POINTER TO PAR_ARRAY

     MOV  A,DPL      ;CHECK FOR END_ADDRESS
     CJNE A,#LOW(END_OF_ARRAY),GO_ON
     MOV  A,DPH
     CJNE A,#HIGH(END_OF_ARRAY),GO_ON
     LJMP END_LOOP

GO_ON:
     CLR  A
     MOVC A,@A+DPTR  ;READ ELEMENT FROM PAR_ARRAY
     INC  DPTR       ;AND INCREMENT POINTER
     MOV  SHADOW_DPL_0,DPL ;SAVE THE ACTUAL DPTR
     MOV  SHADOW_DPH_0,DPH ;TO SHADOW_POINTER

     MOV  DPL,SHADOW_DPL_1 ;LOAD THE ACTUAL DPTR WITH
     MOV  DPH,SHADOW_DPH_1 ;POINTER TO BUFFER

     MOVX @DPTR,A         ;WRITE ELEMENT TO EXT. RAM
     INC  DPTR       ;AND INCREMENT POINTER

     MOV  SHADOW_DPL_1,DPL ;SAVE THE ACTUAL DPTR
     MOV  SHADOW_DPH_1,DPH ;TO SHADOW_POINTER

     SJMP LOOP

END
```

Figure 2. Program Listing using the 8051-Type Microcontroller vs the Program Listing using the SAB 80C517/80C537

Transfer Program for the 80C517

```
;DECLARATION OF THE ARRAYS
;--------------------
     CSEG AT 800H      ;ROM!
PAR_ARRAY: DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
           DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
                         .....
           DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

END_OF_ARRAY   EQU $

     XSEG AT 0000H    ;EXTERNAL RAM!
BUFFER:         DS 512
;--------------------
     CSEG AT 00H
;LOAD THE ACTUAL DATAPOINTERS
     MOV  DPSEL,#00H ;SELECT AND LOAD POINTER
     MOV  DPTR,#PAR_ARRAY;TO PAR_ARRAY

     INC  DPSEL      ;SELECT AND LOAD POINTER
     MOV  DPTR,#BUFFER    ;TO BUFFER
;--------------------
LOOP;MOV  DPSEL,#00H ;SELECT FIRST DPTR (POINTER
                     ;TO PAR_ARRAY)

     MOV  A,DPL      ;CHECK FOR END ADDRESS
     CJNE A,#LOW(END_OF_ARRAY),GO_ON
     MOV  A,DPH
     CJNE A,#HIGH(END_OF_ARRAY),GO_ON
     LJMP END_LOOP

GO_ON:
     CLR  A
     MOVC A,@A+DPTR  ;READ ELEMENT FROM PAR_ARRAY
     INC  DPTR       ;AND INCREMENT POINTER
     INC  DPSEL      ;SELECT THE SECOND DPTR (POINTER
                     ;TO BUFFER)
     MOVX @DPTR,A         ;WRITE ELEMENT TO EXT. RAM
     INC  DPTR       ;AND INCREMENT POINTER

     SJMP LOOP

END
```

Figure 2. Program Listing Using the 8051-Type Microcontroller vs the Program Listing Using the SAB 80C517/80C537 (Continued)

Program Description

First the memory spaces utilized are declared in the program—an operation which is the same for both the 8051 and the 80C517 program. This is done using the assembler directives CSEG for source data field in the ROM and XSEG for destination data field in the external RAM. However, in the 8051 program, internal RAM space needs to be reserved to allow saving of either the source or the destination pointer when one of them is not in use. This is implemented using the assembler directive DSEG. As the data pointer is a 16-bit register, two 8-bit locations are needed for both the source and destination pointers. These memory locations have the symbols SHADOW_DPL_0 and SHADOW_DPH_0 and SHADOW_DPL_1 and SHADOW_DPH_1 for the source and the destination pointers respectively. This procedure of memory declaration is however not needed in the 80C517 program.

Following the declaration is the program itself which in case of the 8051 requires the loading of the shadow data pointers into the internal RAM whereas in the 80C517 program the source and the destination addresses are directly written into the Data Pointer array. In data transfer subroutine of both the programs the initial part is quite similar in which it is checked whether or not the byte transferred is the last byte in the source array. Following this check is the actual transfer in which the byte read from source array in ROM is written into the appropriate location in the destination array in the external RAM. However, the way this transfer is handled by the 8051 routine is different from that of the 80C517 routine.

In the 8051 program, the Data Pointer must be loaded with the appropriate address from the shadow data pointers in the internal RAM, prior to each read and write operation. And after every read and write operation this address must again be saved in these shadow data pointers. This operation of loading and saving is completely eliminated in the 80C517 because of the abundance of the Data Pointer registers. However, in the 80C517 prior to every read and write operation the desired Data Pointer is selected by programming the DPSEL register.

The differences are obvious: In the 8051 program two bytes of memory space per address which need to be saved are needed in the internal RAM (4 bytes in the present example). However, using the 80C517 up to eight addresses can be directly stored in the new data pointer array, thus not occupying any space in the internal RAM. In addition the transfer of data is less time consuming with the 80C517. Most of the time spent in this routine is the time in performing the actual transfer of bytes. The transfer time increases as the number of bytes to be transferred increases. The 8051 program requires lots of time for loading and storing the Data Pointer. In 80C517, however, only the selection of the desired data pointer is necessary. One loop execution time in the 8051 program is 30 machine cycles while 80C517 for the same loop requires only 17 machine cycles; almost twice as fast.

SIEMENS

November 1988

Oscillator Design Considerations for SAB-51 Family of Microcontrollers Operating at Frequencies Higher than 16 MHz

SAB-51 Family

Application Brief

For the SAB-51 (Mymos Technology only) family of microcontrollers specified to run at frequencies 18 and 20 MHz, the following care must be taken when they are used at these frequencies.

Problem

To use the on-chip oscillator, a crystal or ceramic resonator is connected between the XTAL1 and XTAL2 pins of the SAB-51 family of microcontrollers. In all the Mymos-based microcontrollers, XTAL1 is the input of the on-chip oscillator amplifier. XTAL2 is the output of this amplifier, driving the internal clock generator of all the microcontrollers. The gain of the on-chip oscillator amplifier decreases with increase in the crystal frequency, thus reducing the amplitude of the oscillation. This effect is more predominant at frequencies higher than 16 MHz. At crystal frequencies of 18 MHz and 20 MHz, the following work-arounds are recommended to have sufficiently high amplitude of oscillations.

Work Around #1

Using an external clock at XTAL2 pin (refer to Figure 1) for driving the microcontrollers bypasses the on-chip oscillator amplifier, and the amplitude of the oscillations is then controlled by the external circuit. This is the best way to supply clock to microcontrollers at 18 MHz or 20 MHz clock oscillator frequency.

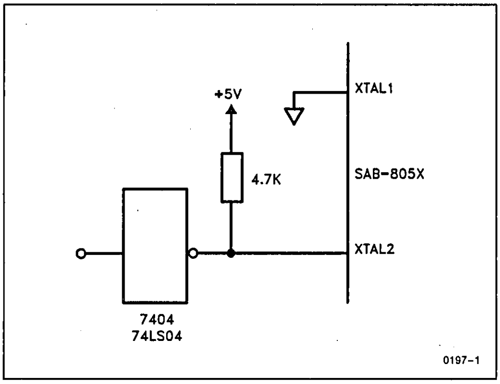


Figure 1. Driving from External Source

Work Around #2

If for some reason, work around #1 is not possible to implement, a crystal oscillator with a careful design can be used.

At such high frequencies, capacitive and inductive couplings between the oscillator circuitry and other signals are a major source of miscounts in the internal clocking circuitry. Surrounding the oscillator components with "quiet" traces (V_{CC} and ground, for example) will alleviate capacitive coupling to signals that have fast transition times. To minimize inductive coupling, the PCB layout should minimize the areas of the loops formed by the oscillator components. These are the loops that should be checked:

XTAL1 through the resonator to XTAL2;
XTAL1 through the capacitor to the V_{SS} pin;
XTAL2 through the capacitor to the V_{SS} pin;

The traces between the grounded ends of the capacitors and the V_{SS} pin should be kept as short as possible.

In addition, a higher amplitude at XTAL2 can be achieved by changing the ratio of the capacitors at XTAL1 and XTAL2. For frequencies higher than 16 MHz (18 MHz and 20 MHz), the configuration in Figure 2 increases the signal amplitude at XTAL2; the influence on the start-up time, however, is negligible. This is only a recommendation, the final design should however be verified against worst case conditions of temperature, V_{CC} levels, device tolerances, etc.

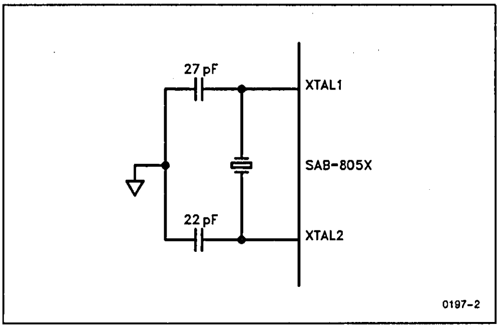


Figure 2. Crystal Oscillator Mode

Software Support/Development Tools for the SAB-51 Family of Microcontrollers

February 1990

Software Support/Development Tools for the SAB-51 Family of Microcontrollers

SAB-51 Family

The hardware and software development support tools for the SAB-51 family of microcontrollers are available on IBM PC, XT, AT or a compatible. There are various solutions available from third party vendors, designed to be operated in an IBM PC compatible environment.

On account of the software compatibility of all the SAB-51 family of microcontrollers with the SAB 8051, the language and utility programs ASM51, PLM51, "C" and RL51 can be used without any restriction for any of the SAB-51 family of microcontrollers. However, the definitions of the new symbolic names as well as the register and bit addresses for the microcontrollers other than the SAB 8051 must be linked to the source program.

For example, in the case of the SAB 80515, a file called REG515.DCL is added to the PLM51 source program by an "Include" directive. Similarly, an "Include" directive is used to link a file called REG515.PDF to the ASM51 source program. In addition, the control instruction NOMOD51 (NOMO) must either be specified at the beginning of the source program, or in the assembler invocation, to avoid a multiple definition of the symbolic names. This control instruction causes default SAB 8051 declarations of register and bit symbols to be ignored during assembly. The file REG515.PDF includes all SAB 80515 register and bit definitions, including those that are declared with MOD51 for SAB 8051. The contents of the files REG515.PDF and REG515.DCL are listed in Appendix A.

Some third party vendors have already provided control directives such as MOD515, MOD512, etc. for their assemblers/compilers which eliminates the need to include files as mentioned above. In that case a mere mention of the control instructions MOD515, MOD512, etc. performs the job and the procedure of including a symbol declaration file is therefore transparent to the user.

Given below is a list of third party support products available for Siemens proprietary microcontrollers. These companies can also be contacted for development tools for the SAB 8031/8051/8032/8052 etc.

Third Party Hardware and Software Support Products

NOTE:
For third party support contact the appropriate vendor for pricing, availability, and other information.

Metalink Corporation:
325 East Elliot Rd., Suite 23
Chandler, AZ 85225
Phone: (602) 926-0797

Hardware Support Product

MetalCE-515, MetalCE-537: Full In-Circuit-Emulator support for 80515 and 80C537 with IBM PC compatible host system. Link to host system via RS232C serial interface.

MetalCE-512: Full In-Circuit-Emulator support for 80512 with IBM PC compatible host system. Link to host system via RS232C serial interface.

Software Support Product

MetaWARE-51: Cross assembly package for IBM PC compatible systems. Fully supports the SAB 80515/80535, SAB 80512/80532 and SAB 80C517/80C537.

Nohau Corporation:
51 East Campbell Ave., Suite 144
Campbell, CA 95008
Phone: (408) 866-1820

Hardware Support Product

EMUL515-PC, EMUL517-PC: Full In-Circuit-Emulator support for 80515/80C515 and 80C537 with IBM PC/XT/AT compatible host system. Link to host system via RS232C serial interface or directly to the mother-board.

EMUL532-PC: Full In-Circuit-Emulator support for 80532 with IBM PC/XT/AT compatible host system. Link to host system via RS232C serial interface or directly to the mother-board.

Software Support Product

EMUL51-PC/C51: "C" Cross compiler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

EMUL51-PC/AVO: Cross assembler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

Signum Systems:
1820 14th St., Suite 203
Santa Monica, CA 90404
Phone: (213) 450-6096

Hardware Support Product

EM-515: Full In-Circuit-Emulator support for 80515/80535 with IBM PC/XT/AT compatible host system. Link to host system via RS232C serial interface.

EM-512: Full In-Circuit-Emulator support for 80512/80532 with IBM PC/XT/AT compatible host system. Link to host system via RS232C serial interface.

Software Support Product

ASM51: Cross assembler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

PL/M51: Cross compiler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

Allen Systems:
2151 Fairfax Road
Columbus, OH 43221
Phone: (614) 488-7122

Hardware Support Product

DP-535: Prototype development board for the SAB 80535 with a monitor program EPROM for program debugging.

DP-532: Prototype development board for the SAB 80532 with a monitor program EPROM for program debugging.

Software Support Product

CA-51: Cross assembler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

Archemides Software:
2159 Union St.,
San Francisco, CA 94123
Phone: (415) 567-4010

Software Support Product

C-51: "C" language cross-compiler package which includes a macro-assembler, linker and librarian for SAB-51 family of microcontrollers. Available for IBM PC compatible (MSDOS based), Microvax/VAX (running VMS/Ultrix) and SUN systems.

SimCASE: Software simulator package for SAB-51 family of microcontrollers for IBM PC compatible systems.

Franklin Software, Inc.:
888 Saratoga Ave., #2
San Jose, CA 95129
Phone: (408) 296-8051

Software Support Product

C-51: "C" language C2055-compiler package which includes a macro-assembler, linker and librarian for SAB-51 family of microcontrollers. Available for IBM PC compatible (MSDOS based) systems.

SIM-51: Software simulator package for SAB-51 family of microcontrollers for IBM PC compatible systems.

Micro Computer Control:
P.O. Box 275,
Hopewell, NJ 08525
Phone: (609) 466-1751

Software Support Product

ASM51: Cross-assembler package for SAB-51 family of microcontrollers. Available for IBM PC compatible systems (MSDOS based).

SIM-51: SAB 8051 simulator.

C-51: "C" language cross-compiler package for SAB-51 family of microcontrollers. Available for IBM PC compatible systems (MSDOS based).

2500AD Software Inc.:
109 Brookdale Ave.,
Box 480
Buena Vista, CO 81211
Phone: (719) 395-8683

Software Support Product

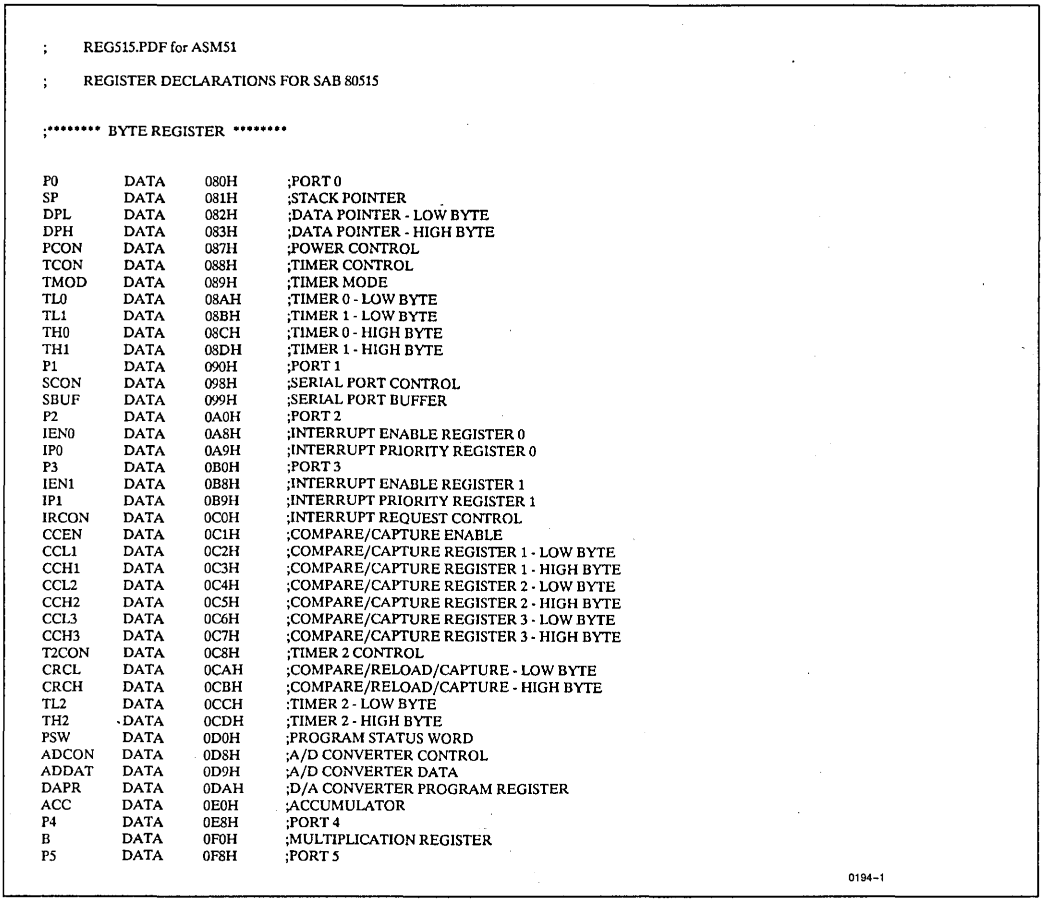
A80515: Cross-assembler package for SAB-51 family of microcontrollers. Available for IBM PC compatible (MSDOS based), Microvax/VAX (running VMS/Ultrix) and SUN systems.

S80515: SAB 80515 simulator. Available for IBM PC compatible (MSDOS based), Microvax/VAX (running VMS/Ultrix) and SUN systems.

C80515: "C" cross-compiler package for SAB-51 family of microcontrollers. Available for IBM PC compatible (MSDOS based), Microvax/VAX (running VMS/Ultrix) and SUN systems.

Appendix A

Register Definitions for SAB 80515/80535



```
;      REG515.PDF for ASM51

;      REGISTER DECLARATIONS FOR SAB 80515

;******** BYTE REGISTER ********

P0      DATA    080H    ;PORT 0
SP      DATA    081H    ;STACK POINTER
DPL     DATA    082H    ;DATA POINTER - LOW BYTE
DPH     DATA    083H    ;DATA POINTER - HIGH BYTE
PCON    DATA    087H    ;POWER CONTROL
TCON    DATA    088H    ;TIMER CONTROL
TMOD    DATA    089H    ;TIMER MODE
TL0     DATA    08AH    ;TIMER 0 - LOW BYTE
TL1     DATA    08BH    ;TIMER 1 - LOW BYTE
TH0     DATA    08CH    ;TIMER 0 - HIGH BYTE
TH1     DATA    08DH    ;TIMER 1 - HIGH BYTE
P1      DATA    090H    ;PORT 1
SCON    DATA    098H    ;SERIAL PORT CONTROL
SBUF    DATA    099H    ;SERIAL PORT BUFFER
P2      DATA    0A0H    ;PORT 2
IEN0    DATA    0A8H    ;INTERRUPT ENABLE REGISTER 0
IP0     DATA    0A9H    ;INTERRUPT PRIORITY REGISTER 0
P3      DATA    0B0H    ;PORT 3
IEN1    DATA    0B8H    ;INTERRUPT ENABLE REGISTER 1
IP1     DATA    0B9H    ;INTERRUPT PRIORITY REGISTER 1
IRCON   DATA    0C0H    ;INTERRUPT REQUEST CONTROL
CCEN    DATA    0C1H    ;COMPARE/CAPTURE ENABLE
CCL1    DATA    0C2H    ;COMPARE/CAPTURE REGISTER 1 - LOW BYTE
CCH1    DATA    0C3H    ;COMPARE/CAPTURE REGISTER 1 - HIGH BYTE
CCL2    DATA    0C4H    ;COMPARE/CAPTURE REGISTER 2 - LOW BYTE
CCH2    DATA    0C5H    ;COMPARE/CAPTURE REGISTER 2 - HIGH BYTE
CCL3    DATA    0C6H    ;COMPARE/CAPTURE REGISTER 3 - LOW BYTE
CCH3    DATA    0C7H    ;COMPARE/CAPTURE REGISTER 3 - HIGH BYTE
T2CON   DATA    0C8H    ;TIMER 2 CONTROL
CRCL    DATA    0CAH    ;COMPARE/RELOAD/CAPTURE - LOW BYTE
CRCH    DATA    0CBH    ;COMPARE/RELOAD/CAPTURE - HIGH BYTE
TL2     DATA    0CCH    ;TIMER 2 - LOW BYTE
TH2     DATA    0CDH    ;TIMER 2 - HIGH BYTE
PSW     DATA    0D0H    ;PROGRAM STATUS WORD
ADCON   DATA    0D8H    ;A/D CONVERTER CONTROL
ADDAT   DATA    0D9H    ;A/D CONVERTER DATA
DAPR    DATA    0DAH    ;D/A CONVERTER PROGRAM REGISTER
ACC     DATA    0E0H    ;ACCUMULATOR
P4      DATA    0E8H    ;PORT 4
B       DATA    0F0H    ;MULTIPLICATION REGISTER
P5      DATA    0F8H    ;PORT 5
```

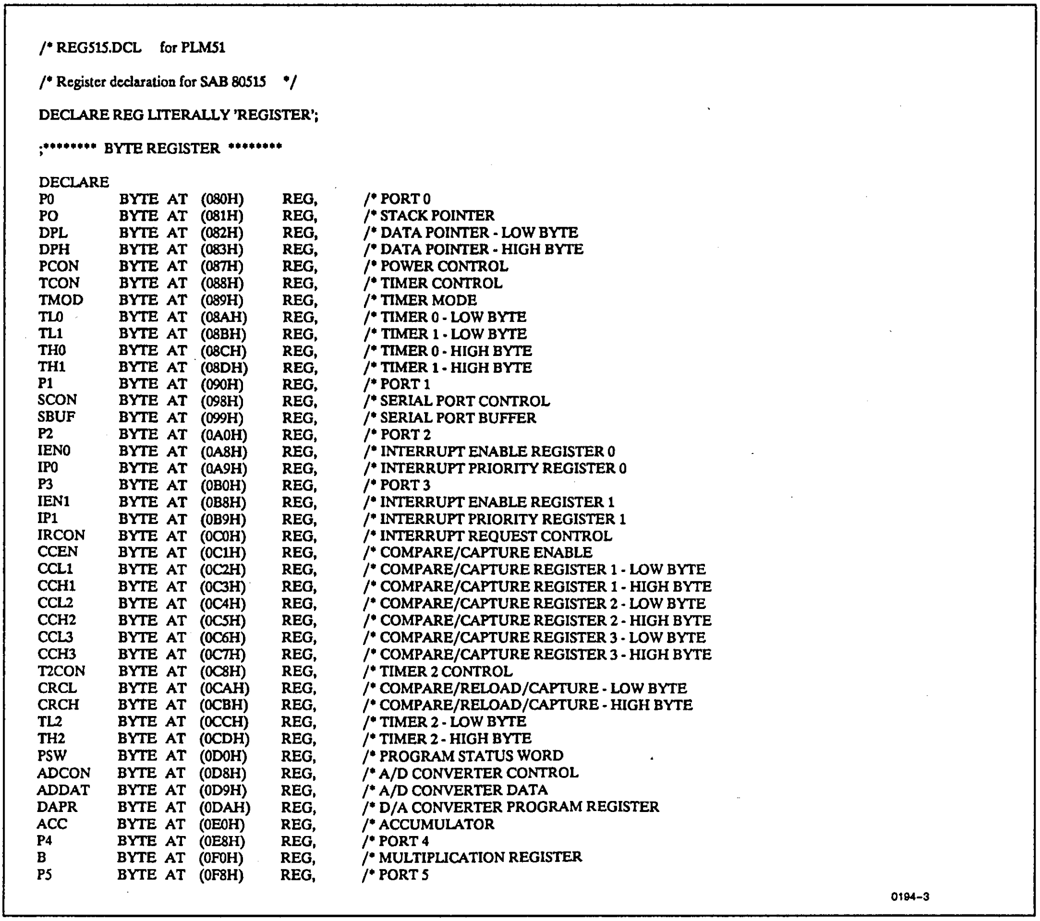
0194-1

Register Definitions for SAB 80515/80535 (Continued)

```
******** BIT REGISTERS ********


IT0     BIT    088H    ;TCON.0 - EXT. INTERRUPT 0 TYPE
IE0     BIT    089H    ;TCON.1 - EXT. INTERRUPT 0 EDGE FLAG
IT1     BIT    08AH    ;TCON.2 - EXT. INTERRUPT 1 TYPE
IE1     BIT    08BH    ;TCON.3 - EXT. INTERRUPT 1 EDGE FLAG
TR0     BIT    08CH    ;TCON.4 - TIMER 0 ON/OFF CONTROL
TF0     BIT    08DH    ;TCON.5 - TIMER 0 OVERFLOW FLAG
TR1     BIT    08EH    ;TCON.6 - TIMER 1 ON/OFF CONTROL
TF1     BIT    08FH    ;TCON.7 - TIMER 1 OVERFLOW FLAG
INT3    BIT    090H    ;P1.0 - EXTERNAL INTERRUPT 3/CAPTURE 0/COMPARE 0
INT4    BIT    091H    ;P1.1 - EXTERNAL INTERRUPT 4/CAPTURE 1/COMPARE 1
INT5    BIT    092H    ;P1.2 - EXTERNAL INTERRUPT 5/CAPTURE 2/COMPARE 2
INT6    BIT    093H    ;P1.3 - EXTERNAL INTERRUPT 6/CAPTURE 3/COMPARE 3
INT2    BIT    094H    ;P1.4 - EXTERNAL INTERRUPT 2
T2EX    BIT    095H    ;P1.5 - TIMER 2 EXTERNAL RELOAD TRIGGER INPUT
CLKOUT  BIT    096H    ;P1.6 - SYSTEM CLOCK OUTPUT
T2      BIT    097H    ;P1.7 - TIMER 2 INPUT
RI      BIT    098H    ;SCON.0 - RECEIVE INTERRUPT FLAG
TI      BIT    099H    ;SCON.1 - TRANSMIT INTERRUPT FLAG
RB8     BIT    09AH    ;SCON.2 - RECEIVE BIT 8
TB8     BIT    09BH    ;SCON.3 - TRANSMIT BIT 8
REN     BIT    09CH    ;SCON.4 - RECEIVE ENABLE
SM2     BIT    09DH    ;SCON.5 - SERIAL MODE CONTROL BIT 2
SM1     BIT    09EH    ;SCON.6 - SERIAL MODE CONTROL BIT 1
SM0     BIT    09FH    ;SCON.7 - SERIAL MODE CONTROL BIT 0
EX0     BIT    0A8H    ;IEN0.0 - EXTERNAL INTERRUPT 0 ENABLE
ET0     BIT    0A9H    ;IEN0.1 - TIMER 0 INTERRUPT ENABLE
EX1     BIT    0AAH    ;IEN0.2 - EXTERNAL INTERRUPT 1 ENABLE
ET1     BIT    0ABH    ;IEN0.3 - TIMER 1 INTERRUPT ENABLE
ES      BIT    0ACH    ;IEN0.4 - SERIAL PORT INTERRUPT ENABLE
ET2     BIT    0ADH    ;IEN0.5 - TIMER 2 INTERRUPT ENABLE
WDT     BIT    0AEH    ;IEN0.6 - WATCHDOG TIMER RESET
EAL     BIT    0AFH    ;IEN0.7 - GLOBAL INTERRUPT ENABLE
RXD     BIT    0B0H    ;P3.0 - SERIAL PORT RECEIVE INPUT
TXD     BIT    0B1H    ;P3.1 - SERIAL PORT TRANSMIT OUTPUT
INT0    BIT    0B2H    ;P3.2 - EXTERNAL INTERRUPT 0 INPUT
INT1    BIT    0B3H    ;P3.3 - EXTERNAL INTERRUPT 1 INPUT
T0      BIT    0B4H    ;P3.4 - TIMER 0 COUNT INPUT
T1      BIT    0B5H    ;P3.5 - TIMER 1 COUNT INPUT
WR      BIT    0B6H    ;P3.6 - WRITE CONTROL FOR EXT. MEMORY
RD      BIT    0B7H    ;P3.7 - READ CONTROL FOR EXT. MEMORY
EADC    BIT    0B8H    ;IEN1.0 - A/D CONVERTER INTERRUPT ENABLE
EX2     BIT    0B9H    ;IEN1.1 - EXTERNAL INTERRUPT 2 ENABLE
EX3     BIT    0BAH    ;IEN1.2 - EXTERNAL INTERRUPT 3 ENABLE
EX4     BIT    0BBH    ;IEN1.3 - EXTERNAL INTERRUPT 4 ENABLE
EX5     BIT    0BCH    ;IEN1.4 - EXTERNAL INTERRUPT 5 ENABLE
EX6     BIT    0BDH    ;IEN1.5 - EXTERNAL INTERRUPT 6 ENABLE
SWDT    BIT    0BEH    ;IEN1.6 - WATCHDOG TIMER START
EXEN2   BIT    0BFH    ;IEN1.7 - TIMER 2 EXTERNAL RELOAD INTERRUPT ENABLE
IADC    BIT    0C0H    ;IRCON.0 - A/D CONVERTER INTERRUPT REQUEST
IEX2    BIT    0C1H    ;IRCON.1 - EXTERNAL INTERRUPT 2 EDGE FLAG
IEX3    BIT    0C2H    ;IRCON.2 - EXTERNAL INTERRUPT 3 EDGE FLAG
IEX4    BIT    0C3H    ;IRCON.3 - EXTERNAL INTERRUPT 4 EDGE FLAG
IEX5    BIT    0C4H    ;IRCON.4 - EXTERNAL INTERRUPT 5 EDGE FLAG
IEX6    BIT    0C5H    ;IRCON.5 - EXTERNAL INTERRUPT 6 EDGE FLAG
TF2     BIT    0C6H    ;IRCON.6 - TIMER 2 OVERFLOW FLAG
EXF2    BIT    0C7H    ;IRCON.7 - TIMER 2 EXTERNAL RELOAD FLAG
T2I0    BIT    0C8H    ;T2CON.0 - TIMER 2 INPUT SELECT BIT 0
T2I1    BIT    0C9H    ;T2CON.1 - TIMER 2 INPUT SELECT BIT 1
T2CM    BIT    0CAH    ;T2CON.2 - COMPARE MODE
T2R0    BIT    0CBH    ;T2CON.3 - TIMER 2 RELOAD MODE SELECT BIT 0
T2R1    BIT    0CCH    ;T2CON.4 - TIMER 2 RELOAD MODE SELECT BIT 1
I2FR    BIT    0CDH    ;T2CON.5 - EXTERNAL INTERRUPT 2 FALLING/RISING EDGE FLAG
I3FR    BIT    0CEH    ;T2CON.6 - EXTERNAL INTERRUPT 3 FALLING/RISING EDGE FLAG
T2PS    BIT    0CFH    ;T2CON.7 - PRESCALER SELECT BIT
P       BIT    0D0H    ;PSW.0 - ACCUMULATOR PARITY FLAG
F1      BIT    0D1H    ;PSW.1 - FLAG 1
OV      BIT    0D2H    ;PSW.2 - OVERFLOW FLAG
RS0     BIT    0D3H    ;PWS.3 - REGISTER BANK SELECT 0
RS1     BIT    0D4H    ;PSW.4 - REGISTER BANK SELECT 1
F0      BIT    0D5H    ;PSW.5 - FLAG 0
AC      BIT    0D6H    ;PSW.6 - AUXILIARY CARRY FLAG
CY      BIT    0D7H    ;PSW.7 - CARRY FLAG
MX0     BIT    0D8H    ;ADCON.0 - ANALOG INPUT CHANNEL SELECT BIT 0
MX1     BIT    0D9H    ;ADCON.1 - ANALOG INPUT CHANNEL SELECT BIT 1
MX2     BIT    0DAH    ;ADCON.2 - ANALOG INPUT CHANNEL SELECT BIT 2
ADM     BIT    0DBH    ;ADCON.3 - A/D CONVERSION MODE
BSY     BIT    0DCH    ;ADCON.4 - BUSY FLAG
CLK     BIT    0DEH    ;ADCON.6 - SYSTEM CLOCK ENABLE
BD      BIT    0DFH    ;ADCON.7 - BAUD RATE ENABLE
```

0194-2

Register Definitions for SAB 80515/80535 (Continued)



```
/* REG515.DCL    for PLM51

/* Register declaration for SAB 80515   */

DECLARE REG LITERALLY 'REGISTER';

;******** BYTE REGISTER ********

DECLARE
P0        BYTE  AT  (080H)     REG,        /* PORT 0
PO        BYTE  AT  (081H)     REG,        /* STACK POINTER
DPL       BYTE  AT  (082H)     REG,        /* DATA POINTER - LOW BYTE
DPH       BYTE  AT  (083H)     REG,        /* DATA POINTER - HIGH BYTE
PCON      BYTE  AT  (087H)     REG,        /* POWER CONTROL
TCON      BYTE  AT  (088H)     REG,        /* TIMER CONTROL
TMOD      BYTE  AT  (089H)     REG,        /* TIMER MODE
TL0       BYTE  AT  (08AH)     REG,        /* TIMER 0 - LOW BYTE
TL1       BYTE  AT  (08BH)     REG,        /* TIMER 1 - LOW BYTE
TH0       BYTE  AT  (08CH)     REG,        /* TIMER 0 - HIGH BYTE
TH1       BYTE  AT  (08DH)     REG,        /* TIMER 1 - HIGH BYTE
P1        BYTE  AT  (090H)     REG,        /* PORT 1
SCON      BYTE  AT  (098H)     REG,        /* SERIAL PORT CONTROL
SBUF      BYTE  AT  (099H)     REG,        /* SERIAL PORT BUFFER
P2        BYTE  AT  (0A0H)     REG,        /* PORT 2
IEN0      BYTE  AT  (0A8H)     REG,        /* INTERRUPT ENABLE REGISTER 0
IP0       BYTE  AT  (0A9H)     REG,        /* INTERRUPT PRIORITY REGISTER 0
P3        BYTE  AT  (0B0H)     REG,        /* PORT 3
IEN1      BYTE  AT  (0B8H)     REG,        /* INTERRUPT ENABLE REGISTER 1
IP1       BYTE  AT  (0B9H)     REG,        /* INTERRUPT PRIORITY REGISTER 1
IRCON     BYTE  AT  (0C0H)     REG,        /* INTERRUPT REQUEST CONTROL
CCEN      BYTE  AT  (0C1H)     REG,        /* COMPARE/CAPTURE ENABLE
CCL1      BYTE  AT  (0C2H)     REG,        /* COMPARE/CAPTURE REGISTER 1 - LOW BYTE
CCH1      BYTE  AT  (0C3H)     REG,        /* COMPARE/CAPTURE REGISTER 1 - HIGH BYTE
CCL2      BYTE  AT  (0C4H)     REG,        /* COMPARE/CAPTURE REGISTER 2 - LOW BYTE
CCH2      BYTE  AT  (0C5H)     REG,        /* COMPARE/CAPTURE REGISTER 2 - HIGH BYTE
CCL3      BYTE  AT  (0C6H)     REG,        /* COMPARE/CAPTURE REGISTER 3 - LOW BYTE
CCH3      BYTE  AT  (0C7H)     REG,        /* COMPARE/CAPTURE REGISTER 3 - HIGH BYTE
T2CON     BYTE  AT  (0C8H)     REG,        /* TIMER 2 CONTROL
CRCL      BYTE  AT  (0CAH)     REG,        /* COMPARE/RELOAD/CAPTURE - LOW BYTE
CRCH      BYTE  AT  (0CBH)     REG,        /* COMPARE/RELOAD/CAPTURE - HIGH BYTE
TL2       BYTE  AT  (0CCH)     REG,        /* TIMER 2 - LOW BYTE
TH2       BYTE  AT  (0CDH)     REG,        /* TIMER 2 - HIGH BYTE
PSW       BYTE  AT  (0D0H)     REG,        /* PROGRAM STATUS WORD
ADCON     BYTE  AT  (0D8H)     REG,        /* A/D CONVERTER CONTROL
ADDAT     BYTE  AT  (0D9H)     REG,        /* A/D CONVERTER DATA
DAPR      BYTE  AT  (0DAH)     REG,        /* D/A CONVERTER PROGRAM REGISTER
ACC       BYTE  AT  (0E0H)     REG,        /* ACCUMULATOR
P4        BYTE  AT  (0E8H)     REG,        /* PORT 4
B         BYTE  AT  (0F0H)     REG,        /* MULTIPLICATION REGISTER
P5        BYTE  AT  (0F8H)     REG,        /* PORT 5
```

0194-3

Register Definitions for SAB 80515/80535 (Continued)

```
******** BIT REGISTERS ********

IT0     BIT  AT  (088H)  REG,   /* TCON.0 - EXT. INTERRUPT 0 TYPE
IE0     BIT  AT  (089H)  REG,   /* TCON.1 - EXT. INTERRUPT 0 EDGE FLAG
IT1     BIT  AT  (08AH)  REG,   /* TCON.2 - EXT. INTERRUPT 1 TYPE
IE1     BIT  AT  (08BH)  REG,   /* TCON.3 - EXT. INTERRUPT 1 EDGE FLAG
TR0     BIT  AT  (08CH)  REG,   /* TCON.4 - TIMER 0 ON/OFF CONTROL
TF0     BIT  AT  (08DH)  REG,   /* TCON.5 - TIMER 0 OVERFLOW FLAG
TR1     BIT  AT  (08EH)  REG,   /* TCON.6 - TIMER 1 ON/OFF CONTROL
TF1     BIT  AT  (08FH)  REG,   /* TCON.7 - TIMER 1 OVERFLOW FLAG
INT3    BIT  AT  (090H)  REG,   /* P1.0 - EXTERNAL INTERRUPT 3/CAPTURE 0/COMPARE 0
INT4    BIT  AT  (091H)  REG,   /* P1.1 - EXTERNAL INTERRUPT 4/CAPTURE 1/COMPARE 1
INT5    BIT  AT  (092H)  REG,   /* P1.2 - EXTERNAL INTERRUPT 5/CAPTURE 2/COMPARE 2
INT6    BIT  AT  (093H)  REG,   /* P1.3 - EXTERNAL INTERRUPT 6/CAPTURE 3/COMPARE 3
INT2    BIT  AT  (094H)  REG,   /* P1.4 - EXTERNAL INTERRUPT 2
T2EX    BIT  AT  (095H)  REG,   /* P1.5 - TIMER 2 EXTERNAL RELOAD TRIGGER INPUT
CLKOUT  BIT  AT  (096H)  REG,   /* P1.6 - SYSTEM CLOCK OUTPUT
T2      BIT  AT  (097H)  REG,   /* P1.7 - TIMER 2 INPUT
RI      BIT  AT  (098H)  REG,   /* SCON.0 - RECEIVE INTERRUPT FLAG
TI      BIT  AT  (099H)  REG,   /* SCON.1 - TRANSMIT INTERRUPT FLAG
RB8     BIT  AT  (09AH)  REG,   /* SCON.2 - RECEIVE BIT 8
TB8     BIT  AT  (09BH)  REG,   /* SCON.3 - TRANSMIT BIT 8
REN     BIT  AT  (09CH)  REG,   /* SCON.4 - RECEIVE ENABLE
SM2     BIT  AT  (09DH)  REG,   /* SCON.5 - SERIAL MODE CONTROL BIT 2
SM1     BIT  AT  (09EH)  REG,   /* SCON.6 - SERIAL MODE CONTROL BIT 1
SM0     BIT  AT  (09FH)  REG,   /* SCON.7 - SERIAL MODE CONTROL BIT 0
EX0     BIT  AT  (0A8H)  REG,   /* IEN0.0 - EXTERNAL INTERRUPT 0 ENABLE
ET0     BIT  AT  (0A9H)  REG,   /* IEN0.1 - TIMER 0 INTERRUPT ENABLE
EX1     BIT  AT  (0AAH)  REG,   /* IEN0.2 - EXTERNAL INTERRUPT 1 ENABLE
ET1     BIT  AT  (0ABH)  REG,   /* IEN0.3 - TIMER 1 INTERRUPT ENABLE
ES      BIT  AT  (0ACH)  REG,   /* IEN0.4 - SERIAL PORT INTERRUPT ENABLE
ET2     BIT  AT  (0ADH)  REG,   /* IEN0.5 - TIMER 2 INTERRUPT ENABLE
WDT     BIT  AT  (0AEH)  REG,   /* IEN0.6 - WATCHDOG TIMER RESET
EAL     BIT  AT  (0AFH)  REG,   /* IEN0.7 - GLOBAL INTERRUPT ENABLE
RXD     BIT  AT  (0B0H)  REG,   /* P3.0 - SERIAL PORT RECEIVE INPUT
TXD     BIT  AT  (0B1H)  REG,   /* P3.1 - SERIAL PORT TRANSMIT OUTPUT
INT0    BIT  AT  (0B2H)  REG,   /* P3.2 - EXTERNAL INTERRUPT 0 INPUT
INT1    BIT  AT  (0B3H)  REG,   /* P3.3 - EXTERNAL INTERRUPT 1 INPUT
T0      BIT  AT  (0B4H)  REG,   /* P3.4 - TIMER 0 COUNT INPUT
T1      BIT  AT  (0B5H)  REG,   /* P3.5 - TIMER 1 COUNT INPUT
WR      BIT  AT  (0B6H)  REG,   /* P3.6 - WRITE CONTROL FOR EXT. MEMORY
RD      BIT  AT  (0B7H)  REG,   /* P3.7 - READ CONTROL FOR EXT. MEMORY
EADC    BIT  AT  (0B8H)  REG,   /* IEN1.0 - A/D CONVERTER INTERRUPT ENABLE
EX2     BIT  AT  (0B9H)  REG,   /* IEN1.1 - EXTERNAL INTERRUPT 2 ENABLE
EX3     BIT  AT  (0BAH)  REG,   /* IEN1.2 - EXTERNAL INTERRUPT 3 ENABLE
EX4     BIT  AT  (0BBH)  REG,   /* IEN1.3 - EXTERNAL INTERRUPT 4 ENABLE
EX5     BIT  AT  (0BCH)  REG,   /* IEN1.4 - EXTERNAL INTERRUPT 5 ENABLE
EX6     BIT  AT  (0BDH)  REG,   /* IEN1.5 - EXTERNAL INTERRUPT 6 ENABLE
SWDT    BIT  AT  (0BEH)  REG,   /* IEN1.6 - WATCHDOG TIMER START
EXEN2   BIT  AT  (0BFH)  REG,   /* IEN1.7 - TIMER 2 EXTERNAL RELOAD INTERRUPT ENABLE
IADC    BIT  AT  (0C0H)  REG,   /* IRCON.0 - A/D CONVERTER INTERRUPT REQUEST
IEX2    BIT  AT  (0C1H)  REG,   /* IRCON.1 - EXTERNAL INTERRUPT 2 EDGE FLAG
IEX3    BIT  AT  (0C2H)  REG,   /* IRCON.2 - EXTERNAL INTERRUPT 3 EDGE FLAG
IEX4    BIT  AT  (0C3H)  REG,   /* IRCON.3 - EXTERNAL INTERRUPT 4 EDGE FLAG
IEX5    BIT  AT  (0C4H)  REG,   /* IRCON.4 - EXTERNAL INTERRUPT 5 EDGE FLAG
IEX6    BIT  AT  (0C5H)  REG,   /* IRCON.5 - EXTERNAL INTERRUPT 6 EDGE FLAG
TF2     BIT  AT  (0C6H)  REG,   /* IRCON.6 - TIMER 2 OVERFLOW FLAG
EXF2    BIT  AT  (0C7H)  REG,   /* IRCON.7 - TIMER 2 EXTERNAL RELOAD FLAG
T2I0    BIT  AT  (0C8H)  REG,   /* T2CON.0 - TIMER 2 INPUT SELECT BIT 0
T2I1    BIT  AT  (0C9H)  REG,   /* T2CON.1 - TIMER 2 INPUT SELECT BIT 1
T2CM    BIT  AT  (0CAH)  REG,   /* T2CON.2 - COMPARE MODE
T2R0    BIT  AT  (0CBH)  REG,   /* T2CON.3 - TIMER 2 RELOAD MODE SELECT BIT 0
T2R1    BIT  AT  (0CCH)  REG,   /* T2CON.4 - TIMER 2 RELOAD MODE SELECT BIT 1
I2FR    BIT  AT  (0CDH)  REG,   /* T2CON.5 - EXTERNAL INTERRUPT 2 FALLING/RISING EDGE FLA
I3FR    BIT  AT  (0CEH)  REG,   /* T2CON.6 - EXTERNAL INTERRUPT 3 FALLING/RISING EDGE FLA
T2PS    BIT  AT  (0CFH)  REG,   /* T2CON.7 - PRESCALER SELECT BIT
P       BIT  AT  (0D0H)  REG,   /* PSW.0 - ACCUMULATOR PARITY FLAG
F1      BIT  AT  (0D1H)  REG,   /* PSW.1 - FLAG 1
OV      BIT  AT  (0D2H)  REG,   /* PSW.2 - OVERFLOW FLAG
RS0     BIT  AT  (0D3H)  REG,   /* PWS.3 - REGISTER BANK SELECT 0
RS1     BIT  AT  (0D4H)  REG,   /* PSW.4 - REGISTER BANK SELECT 1
F0      BIT  AT  (0D5H)  REG,   /* PSW.5 - FLAG 0
AC      BIT  AT  (0D6H)  REG,   /* PSW.6 - AUXILIARY CARRY FLAG
CY      BIT  AT  (0D7H)  REG,   /* PSW.7 - CARRY FLAG
MX0     BIT  AT  (0D8H)  REG,   /* ADCON.0 - ANALOG INPUT CHANNEL SELECT BIT 0
MX1     BIT  AT  (0D9H)  REG,   /* ADCON.1 - ANALOG INPUT CHANNEL SELECT BIT 1
MX2     BIT  AT  (0DAH)  REG,   /* ADCON.2 - ANALOG INPUT CHANNEL SELECT BIT 2
ADM     BIT  AT  (0DBH)  REG,   /* ADCON.3 - A/D CONVERSION MODE
BSY     BIT  AT  (0DCH)  REG,   /* ADCON.4 - BUSY FLAG
CLK     BIT  AT  (0DEH)  REG,   /* ADCON.6 - SYSTEM CLOCK ENABLE
BD      BIT  AT  (0DFH)  REG,   /* ADCON.7 - BAUD RATE ENABLE
```

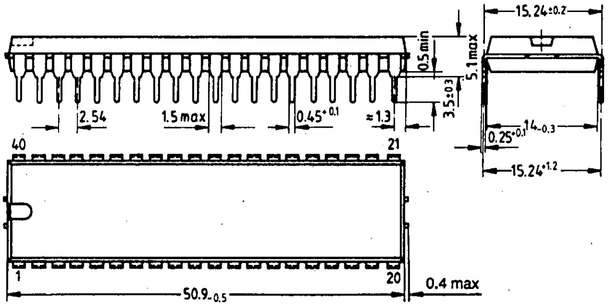
0194-4

Summary of Package Outlines

10

Package Outlines

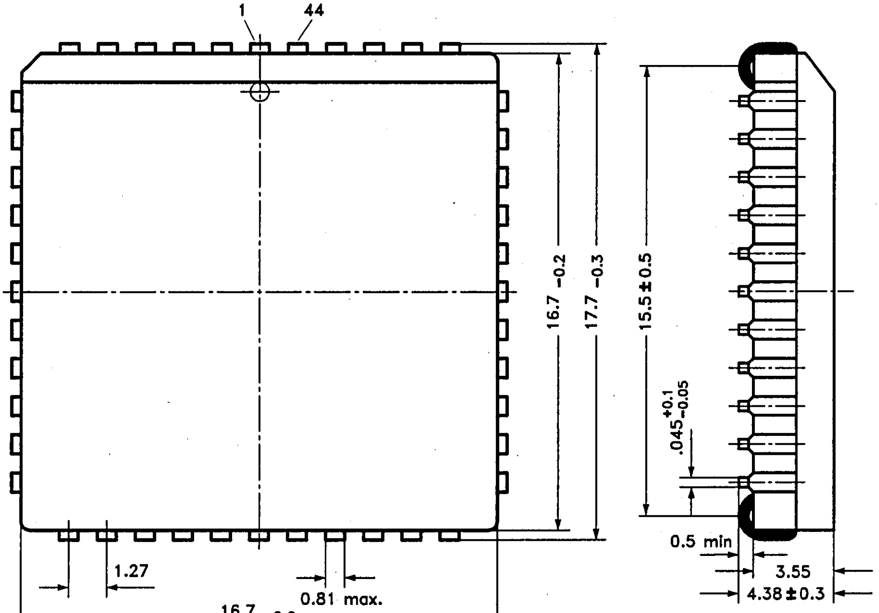
Plastic package, P-DIP-40
(Dual-In-Line Package)
20 B 40 DIN 41870 T10



Approx. weight 5.9 g

Dimensions in mm

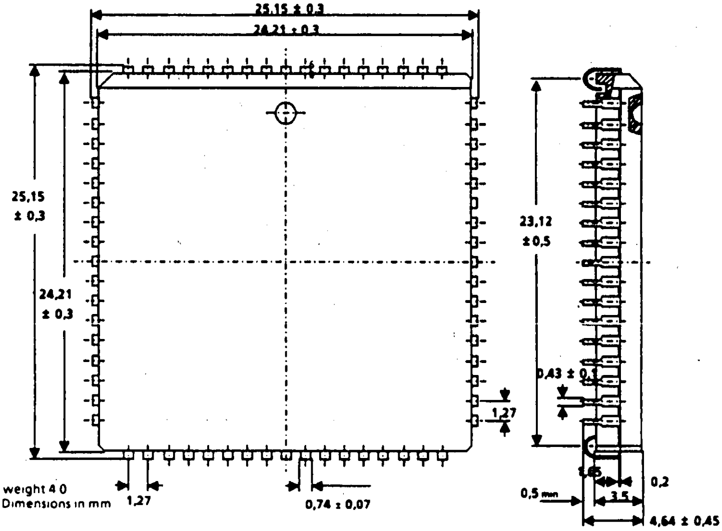
Plastic package, PL-CC-44
(Plastic Leaded—Chip Carrier)-**SMD**



Dimensions in mm

Plastic Package, PLCC-68 (SMD)

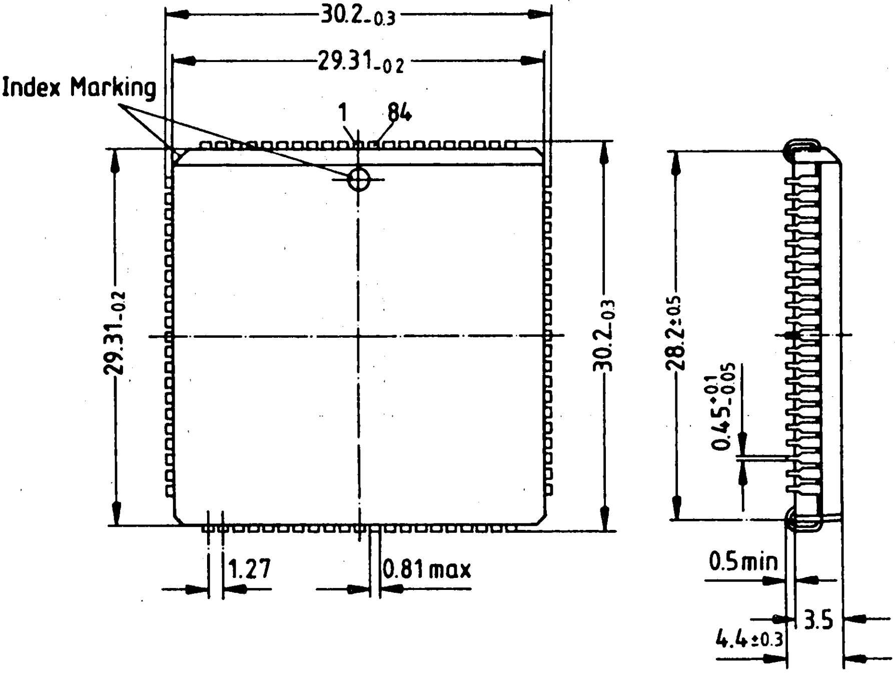
(plastic leaded chip carrier)



weight 4 0
Dimensions in mm

Plastic Package, PLCC-84 (SMD)

(plastic leaded chip carrier)



Dimensions in mm

Siemens Sales Offices

11

Siemens Components, Inc.

Semiconductor Group North American Sales Offices

UNITED STATES

- **Siemens Components, Inc.**
  2 Lowell Research Ctr. Dr.
  Suite 105
  Lowell, MA 01852
  508/454-0113

- **Siemens Components, Inc.**
  307 Fellowship Road
  Suite 202
  Mt. Laurel, NJ 08054
  609/273-6677

- **Siemens Components, Inc.**
  120 Wood Avenue South
  Suite 606
  Iselin, NJ 08830
  201/603-0600

- **Siemens Components, Inc.**
  6525 The Corners Parkway
  Suite 206
  Norcross, GA 30092
  404/449-3981

- **Siemens Components, Inc.**
  39209 W. Six Mile Rd.
  Suite 209
  Livonia, MI 48152
  313/462-1195

- **Siemens Components, Inc.**
  2340 River Road
  Suite 218
  Des Plaines, IL 60018
  708/296-5050

- **Siemens Components, Inc.**
  625 The City Dr. South
  Suite 320
  Orange, CA 92668
  714/385-1274

- **Siemens Components, Inc.**
  2191 Laurelwood Road
  Santa Clara, CA 95054
  408/980-7968

- **Siemens Components, Inc.**
  3003 LBJ Freeway, #115
  Dallas, TX 75234
  214/620-2294

CANADA

- **Siemens Electric Ltd.**
  1180 Courtney Park Drive
  Mississauga, Ont. L5T 1P2
  416/564-1995

- **Siemens Electric Ltd.**
  3600 Billings Court, Suite 100
  Burlington, Ont. L7N 3N6
  416/333-3773

- **Siemens Electric Ltd.**
  101A-5855 9th Street S.E.
  Calgary, Alberta T2H 1Z9
  403/252-2278

- **Siemens Electric Ltd.**
  1024 Winnipeg Street
  Regina, Saskatchewan S4R 8P8
  306/352-9910

- **Siemens Electric Ltee.**
  7300 Trans Canada Highway
  Pointe Claire, Que. H9R 4R6
  514/695-7300

- **Siemens Electric Ltee.**
  1500 rue Janelle
  Drummondville, Que. J2C 3E5
  819/472-1155

- **Siemens Electric Ltd.**
  9404 41st Avenue
  Edmonton, Alberta T6E 6G8
  403/450-6762

- **Siemens Electric Ltd.**
  109 Ilsley Avenue, Unit #1
  Dartmouth, N.S. B3B 1S8
  902/469-9791

- **Siemens Electric Ltd.**
  333 Consortium Court
  London, Ontario N6E 2S8
  519/685-7282

- **Siemens Electric Ltd.**
  108 Prospect Street, Suite 1
  Fredericton, New Brunswick
  E3B 2T9
  506/458-9788

- **Siemens Electric Ltee.**
  366 rue St-Patrick
  Lasalle, Que. H8N 2W7
  514/365-3315

- **Siemens Electric Ltd.**
  303 Moodie Drive, Suite 3400
  Nepean, Ontario K2H 9R4
  613/721-8180

- **Siemens Electric Ltd.**
  106B - 701 Cynthia Street
  Saskatoon, Saskatchewan
  S7L 6B7
  306/652-3101

- **Siemens Electric Ltee.**
  128 - 1990 boul. Charest ouest
  Ste-Foy, Que. G1N 4K8
  418/687-4524

- **Siemens Electric Ltd.**
  109 Clyde Ave.
  Donovans Industrial Park
  P.O. Box 668
  Mount Pearl, Nfld. A1N 2X1
  709/364-5131

- **Siemens Electric Ltd.**
  1034 rue King Est
  Sherbrooke, Que. J1G 1E4
  819/563-9656

- **Siemens Electric Ltd.**
  P.O. Box 1204
  Timmins, Ont. P4N 7J5
  705/267-7755

- **Siemens Electric Ltd.**
  8687 Yukon Street
  Vancouver, B.C. V5X 4T5
  604/321-8687

- **Siemens Electric Ltd.**
  1260 Border Street
  Winnipeg, Manitoba R3H 0M6
  204/633-8655

- **Siemens Electric Ltd.**
  654 Burnside Road West
  Victoria, B.C. V8Z 1M8
  604/727-6465

Issued by Integrated Circuit Division
2191 Laurelwood Road, Santa Clara, CA 95054 (408) 980-4500

Siemens Components, Inc.
Printed in U.S.A.

Siemens Components, Inc.

U.S. Regional Distributors

• **ALABAMA**
Hall-Mark
Huntsville
205/837-8700

Marshall
Huntsville
205/881-9235

• **ARIZONA**
Hall-Mark
Phoenix
602/437-1200

Insight Electronics
Tempe
602/829-1800

Marshall
Phoenix
602/496-0290

• **CALIFORNIA—Northern**
Hall-Mark
San Jose
408/432-4000

Citrus Heights
916/722-8600

Marshall
Milpitas
408/942-4600

Rancho Cordova
916/635-9700

Western Microtechnology
Saratoga
408/725-1660

• **CALIFORNIA—Southern**
Hall-Mark
Chatsworth
818/773-4500

San Diego
619/268-1201

Irvine
714/727-6000

Insight Electronics
San Diego
619/587-0471

Irvine
714/727-2111

Agoura
818/707-2100

Marshall
San Diego
619/578-9600

Irvine
714/458-5301

Chatsworth
818/407-4100

Western Micro
Agoura Hills
818/707-0377

Orange
714/637-0200

San Diego
619/453-8430

• **COLORADO**
Hall-Mark
Englewood
303/790-1662

Marshall, Thornton
303/451-8383

• **CONNECTICUT**
Hall-Mark
Chesire
203/271-2844

Marshall
Wallingford
203/265-3822

• **FLORIDA**
Hall-Mark
Orlando
305/855-4020

Pompano Beach
305/971-9280

Largo
813/530-4543

Marshall
Alta Monte Springs
407/767-8585

St. Petersburg
813/573-1399

Ft. Lauderdale
305/977-4880

• **GEORGIA**
Hall-Mark
Norcross
404/447-8000

Marshall
Norcross
404/923-5750

• **ILLINOIS**
Advent Electronics
Rosemont
312/297-6200

Hall-Mark
Wood Dale
312/860-3800

Marshall
Schaumburg
312/490-0155

• **INDIANA**
Advent Electronics
Indianapolis
317/872-4910

Hall-Mark
Indianapolis
317/872-8875

Marshall
Indianapolis
317/297-0483

• **IOWA**
Advent Electronics
Cedar Rapids
319/363-0221

• **KANSAS**
Hall-Mark
Lenexa
913/888-4747

Marshall
Lenexa
913/492-3121

• **MARYLAND**
Hall-Mark
Columbia
301/988-9800

Marshall
Silver Springs
301/622-1118

• **MASSACHUSETTS**
Hall-Mark
Billerica
617/935-9777,
508/667-0902

Marshall
Wilmington
508/658-0810

Western Microtechnology
Burlington
617/273-2800

Mass Comp, Inc.
West Peabody
617/535-7270

• **MICHIGAN**
Advent
Farmington Hills
313/477-1650

Hall-Mark
Livonia
313/462-1205

Marshall
Livonia
313/525-5850

• **MINNESOTA**
Hall-Mark
Eden Prairie
612/941-2600

Marshall
Plymouth
612/559-2211

• **MISSOURI**
Hall-Mark
Earth City
314/291-5350

Marshall
Bridgeton
314/291-4650

• **NEW JERSEY**
Hall-Mark
Fairfield
201/575-4415

Mt. Laurel
609/235-1900

Marshall
Fairfield
201/882-0320

Mt. Laurel
609/234-9100

• **NEW YORK**
Hall-Mark
Ronkonkoma
516/737-0600

Fairport
716/425-3300

Marshall
Hauppauge
516/273-2424

Johnson City
607/798-1611

Rochester
716/235-7620

Summit Distributors
Buffalo
716/887-2800

Rochester
716/334-8110

• **NORTH CAROLINA**
Hall-Mark
Raleigh
919/872-0712

Marshall
Raleigh
919/878-9882

• **OHIO**
Hall-Mark
Solon
216/349-4632

Worthington
614/888-3313

Marshall
Solon
216/248-1788

Dayton
513/898-4480

• **OKLAHOMA**
Hall-Mark
Tulsa
918/254-6110

• **OREGON**
Marshall
Beaverton
503/644-5050

Western Microtechnology
Beaverton
503/629-2082

• **PENNSYLVANIA**
Marshall
Pittsburgh
412/963-0441

• **TEXAS**
Allied/Hall-Mark
Ft. Worth
817/265-9341

Hall-Mark
Dallas
214/553-4300

Austin
512/258-8848

Houston
713/781-6100

Marshall
Carrollton
214/233-5200

Austin
512/837-1991

Houston
713/895-9200

El Paso
915/593-0706

Harlington
512/542-4589

• **UTAH**
Marshall
Salt Lake City
801/485-1551

• **WASHINGTON**
Marshall
Bothell
206/486-5747

Western Microtechnology
Redmond
206/881-6737

• **WISCONSIN**
Hall-Mark
New Berlin
414/797-7844

Marshall
Waukesha
414/797-8400

Issued by Integrated Circuit Division
2191 Laurelwood Road, Santa Clara, CA 95054 (408) 980-4500

Siemens Components, Inc.
Printed in U.S.A.

Issued by Integrated Circuit Division
2191 Laurelwood Road, Santa Clara, CA 95054 (408) 980-4500

Siemens Components, Inc.

M14T023
Printed in U.S.A.
10K/0490/RRD/DD

Issued by Integrated Circuit Division
2191 Laurelwood Road, Santa Clara, CA 95054 (408) 980-4500

Siemens Components, Inc.

M14T023
Printed in U.S.A.
10K/0490/RRD/DD