intel

OEM COMPUTERS

SBC 80/10
SINGLE BOARD COMPUTER
HARDWARE REFERENCE MANUAL

$500

SBC-80/10

HARDWARE REFERENCE MANUAL

Intel Corporation
3065 Bowers Avenue
Santa Clara, California 95050

98-230B

SBC-80/10

HARDWARE REFERENCE MANUAL

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The SBC-80/10 is a member of Intel's complete line of OEM computer systems which take full advantage of Intel's LSI technology to provide economical, self-contained computer based solutions for OEM applications. The SBC-80/10 is a complete computer system on a single 6.75-by-12 inch printed circuit card. The CPU, system clock, read/write memory, non-volatile read-only-memory, I/O ports and drivers, serial communications interface, bus control logic and drivers all reside on the board.

Intel's powerful 8-bit n-channel MOS 8080A CPU, fabricated on a single LSI chip, is the central processor for the SBC-80/10. The 8080A contains six 8-bit general purpose registers and an accumulator. The six general purpose registers may be addressed individually or in pairs providing both single and double precision operators. The 8080A has a 16-bit program counter which allows direct addressing of up to 64K bytes of memory. An external stack, located within any portion of memory, may be used as a last in/first out stack to store and retrieve the contents of the program counter, flags, accumulator and all of the six general purpose registers. A sixteen bit stack pointer controls the addressing of this external stack. This stack provides almost unlimited subroutine nesting. Sixteen-line address and eight-line bi-directional data busses are used to facilitate easy interface to memory and I/O.

The powerful 8080A instruction set allows the user to write

efficient programs in a minimum amount of time. The accumulator group instructions include arithmetic and logical operators with direct, register indirect, and immediate addressing modes. Move, load, and store instruction groups provide the ability to move either 8 or 16 bits of data between memory, the six working registers and the accumulator using all addressing modes. The ability to branch to different portions of a program is provided with jump, jump conditional, and computed jumps. The ability to conditionally and unconditionally call to and return from subroutines is provided. The RESTART (or single byte call instruction) is used for interrupt operation. Double precision operators such as stack manipulation and double add instructions extend both the arithmetic and interrupt handling capability of the 8080A. The ability to increment and decrement memory, the six general registers and the accumulator is provided as well as extended increment and decrement instructions to operate on the register pairs and stack pointer.

The SBC-80/10 contains 1K 8-bit words of read/write memory using Intel's 8111 low power static RAMs. Sockets for up to 4K 8-bit words of non-volatile read-only memory are provided on the board. Read-only memory may be added in 1K byte increments by using Intel's 8708 erasable and electrically reprogrammable ROMs (EPROMs) or Intel® 8308 metal mask ROMs. The 8080A performs read and write operations at maximum speed when accessing all on-board memory.

The SBC-80/10 contains 48 programmable parallel I/O lines implemented using two Intel® 8255 Programmable Peripheral Interface devices.

The software is used to configure the I/O lines in combinations of unidirectional input/output, and bidirectional ports. Therefore, the I/O interface may be customized to meet specified peripheral requirements. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. Hence, the flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate optional line drivers and terminators for each application.

A programmable serial communications interface using Intel's 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the board. The USART can be programmed by the systems software to provide virtually any serial data transmission technique presently in use (including IBM Bi-Sync). The mode of operation (i.e., synchronous or asynchronous), data format, control character format, parity, and asynchronous serial transmission rate (within limitations given later) are all under program control. The 8251 provides full duplex, double buffered transmission and receive capability. Parity, overrun, and framing error detection are all incorporated in the USART. The inclusion of jumper selectable teletype, or RS232C compatible interfaces on the board in conjunction with the USART provide a direct interface to a teletype, CRT, RS232C compatible devices, and asynchronous and synchronous modems.

A single-level interrupt may originate from any one of six sources including the USART, Programmable I/O interface, and two user designated interrupt request lines. When an interrupt request is recognized, a RESTART 7 instruction is generated. The processor responds by suspending program execution and executing a user defined interrupt service

routine originating at location 38_{16}.

Memory and I/O expansion may be achieved using standard Intel boards. Memory may be expanded to 65,536 bytes by adding user specified combinations of SBC-016 16K byte RAM board, SBC-406 6K byte and SBC-416 16K byte PROM boards. Input/output capacity may be expanded in increments of 4 input ports and 4 output ports using SBC-508 Input/Output boards. Expandable backplanes and cardcages are available to support multi-board systems.

The development cycle of SBC-80/10 based OEM products may be significantly reduced using the Intellec MDS™ Microcomputer Development System. The resident assembler, text editor, and system monitor greatly simplify the design, development, and debug of SBC-80/10 based system software. A unique In-Circuit Emulator (ICE-80) MDS option provides the capability of executing and debugging OEM system software directly on the SBC-80/10.

Intel's high-level language, PL/M, can be used to significantly decrease the time required to develop OEM system software.

CHAPTER 2

FUNCTIONAL DESCRIPTION

For descriptive purposes, the circuitry on the SBC-80/10 can be divided into six functional blocks:

1) CPU Set
2) System Bus Interface
3) Random Access Memory (RAM)
4) Read Only Memory (ROM/PROM) Logic
5) Serial I/O Interface
6) Parallel I/O Interface

as shown in Figure 2-1.

The CPU Set consists of the 8080A Control Processor, the 8224 Clock Generator and the 8238 System Controller. The CPU Set is the heart of the SBC-80/10. It performs all system processing functions and provides a stable timing reference for all other circuitry in the system. The CPU Set generates all of the address and control signals necessary to access memory and I/O ports both on the SBC-80/10 and external to the SBC-80/10. The CPU Set is capable of fetching and executing any of the 8080's seventy-eight instructions. The CPU Set responds to interrupt requests originating both on and off the SBC-80/10, to HOLD requests from modules wishing to acquire control of the system bus, and to WAIT requests from memory or I/O devices having an access time which is slower than the 8080's cycle time.

The System Bus Interface includes an assortment of circuitry which gates interrupt requests, HOLD requests, READY (no wait inputs and the system reset input to the appropriate pins of the CPU Set. Other circuits drive the various external system control signals. The

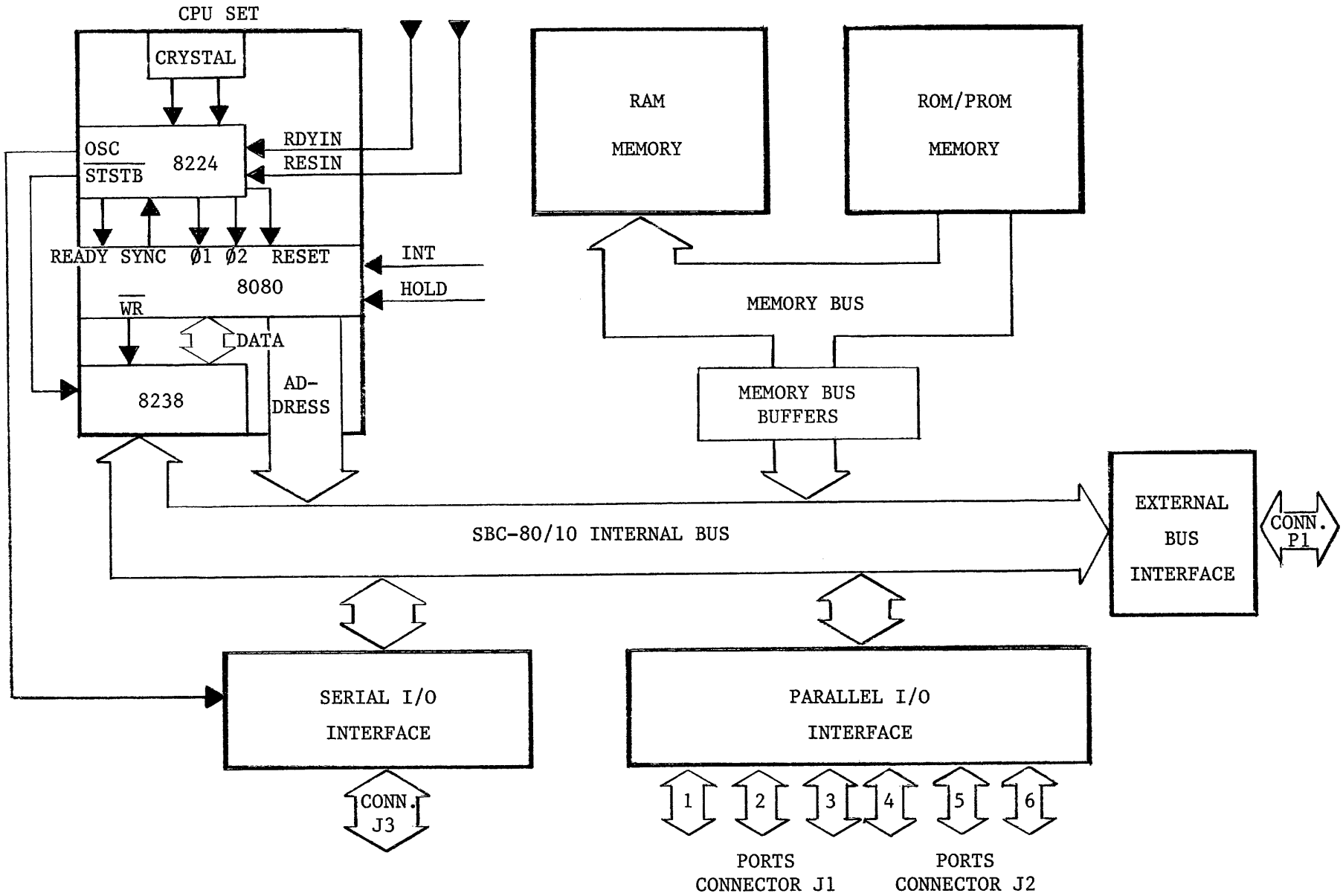


FIGURE 2-1. SBC-80/10 FUNCTIONAL BLOCK DIAGRAM

System Bus Interface also includes two 8216 bi-directional bus drivers which drive the memory data bus on the SBC-80/10. Six 8226 devices drive the external system data and address busses.

The Random Access Memory (RAM) section provides the SBC-80/10 user with 1024 × 8-bits of on-board read/write storage. Eight Intel® 8111 Static MOS RAM chips (256 × 4-bits each) are mounted on the SBC-80/10, along with all of the necessary acknowledgment and memory address decoding logic.

The Read Only Memory (ROM/PROM) section provides the user with the necessary provisions for installing 4096 × 8-bits of ROM or PROM. The SBC-80/10 has four 24-pin sockets that can accept either Intel's 8708 Erasable and Electrically Programmable Read Only Memory chips (1024 × 8-bits each) or Intel's 8308 Static MOS Read Only Memory chips (also 1024 × 8-bits each). The ROM/PROM logic also includes the necessary acknowledgment and memory address decoding circuitry.

The Serial I/O Interface, using Intel's 8251 USART device, provides a bi-directional serial data communications channel that can be programmed to operate with most of the current serial data transmission protocols. Synchronous or asynchronous mode, baud rate, character length, number of stop bits and the choice of even, odd or no parity are all program selectable. The user also has the option of configuring the Serial I/O Interface as an EIA RS232 interface or as a Teletype-compatible current loop interface.

The Parallel I/O Interface, using two Intel® 8255 Programmable Peripheral interface devices, provides 48 signal lines for the transfer and

control of data to or from peripheral devices. Eight lines already have a bidirectional driver and termination network permanently installed. This bidirectional network allows these eight lines to be inputs, outputs, or bidirectional (selected via jumpers). The remaining 40 lines, however, are uncommitted. Sockets are provided for the installation of drivers or termination networks as required to meet the specific needs of the user system.

CHAPTER 3

THEORY OF OPERATION

In the preceding chapter we introduced each of the SBC-80/10 functional blocks and defined what each block was capable of doing. In this chapter we shall go one step further and describe how each block performs its particular function(s). The text will constantly refer to the SBC-80/10 schematics, provided in Appendix A.

Note: Both active-high (positive true) and active-low (negative true) signals appear on the SBC-80/10 schematics. To eliminate any confusion when reading this chapter, the following convention will be adhered to: whenever a signal is active-low, its mnemonic is followed by a slash; for example, MRDC/ means that the level on that line will be low when the memory read command is true (active). If the signal is subsequently inverted, thus making it active-high, the slash is omitted; for example, MRDC means that the level on that line will be high when the memory read command is true.

3.1 THE CPU SET

The CPU Set consists of three Intel® integrated circuit devices:

* 8080A Central Processor Unit
* 8224 Clock Generator
* 8238 System Controller

and an 18.432 MHz crystal that establishes the frequency of oscillation for the 8224 device via a 10pF capacitor, as shown in Figure 3-0. Together, the elements in the CPU Set perform all central processing functions. The following paragraphs describe how the elements within the CPU Set interact with all other logic on the SBC-80/10. The interaction between the IC's within the CPU Set, however, is not described. Instead, the reader is referred to the Intel® "8080 Microcomputer Systems User's Manual" for a detailed description of the 8080, 8224 and 8238 devices.

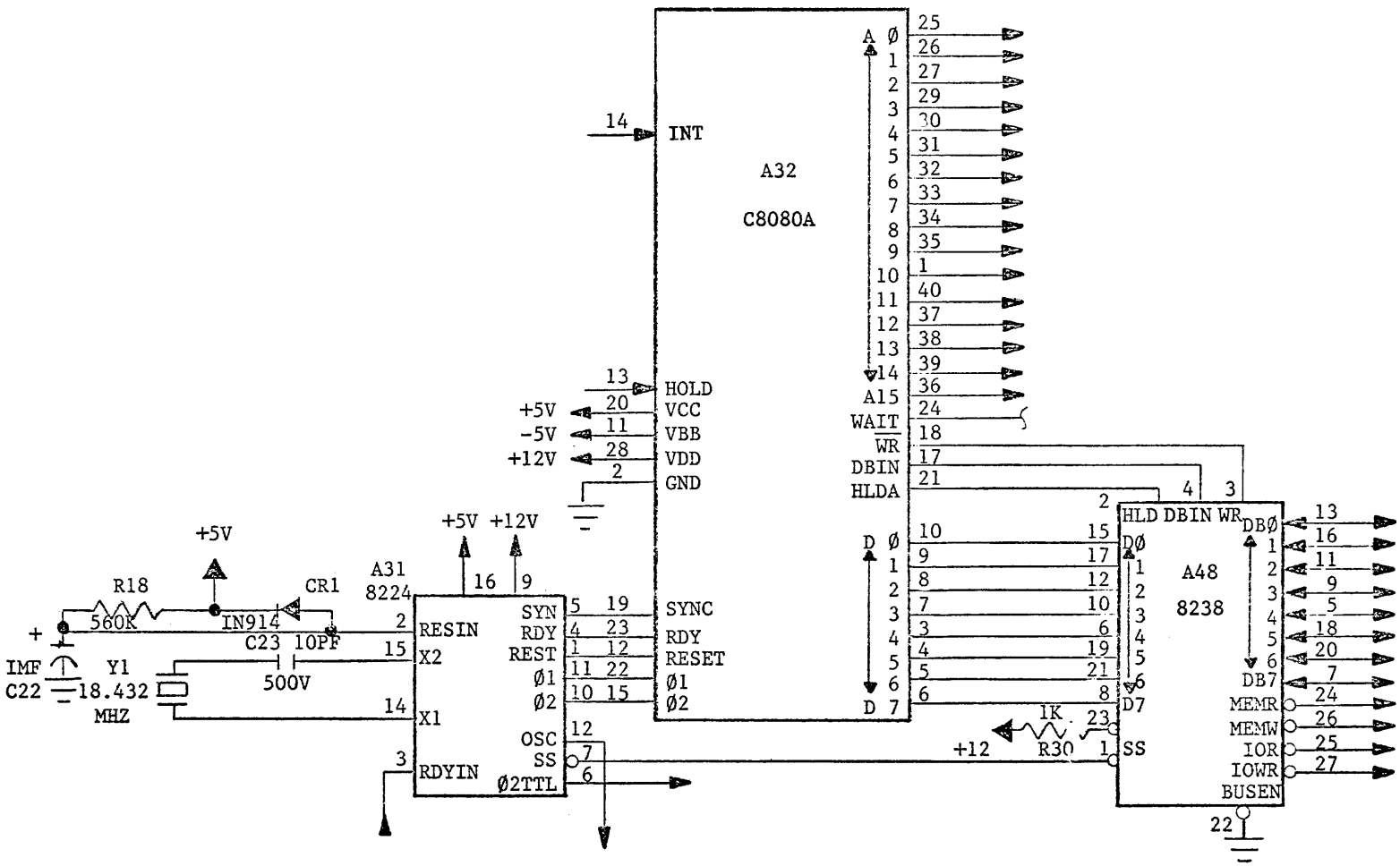


FIGURE 3-0. THE CPU SET

The CPU Set is shown on sheet 1 of the SBC-80/10 schematic (Appendix A).

3.1.1 INSTRUCTION TIMING

The activities of the CPU Set are cyclical. The CPU fetches an instruction, performs the operations required, fetches the next instruction, and so on. This orderly sequence of events requires precise timing. The 8224 Clock Generator, provides the primary timing reference for the CPU Set. The crystal in conjunction with a 10pF capacitor tunes an oscillator within the 8224 to precisely 18.432 MHz. The 8224 "divides" the oscillations by nine to produce two-phase timing inputs (ϕ1 and ϕ2) for the 8080. The ϕ1 and ϕ2 signals define a cycle of approximately 488 ns. duration. A TTL level phase 2 (Ø2TTL) signal is also derived and made available to external logic. In addition, the output of the oscillator is buffered and brought out on OSC so that other system timing signals can be derived from this stable, crystal controlled source (e.g., the serial I/O baud rate is derived from OSC). All processing activities of the CPU Set are referred to the period of the ϕ1 and ϕ2 clock signals.

Within the 8080 CPU Set, an instruction cycle is defined as the time required to fetch and execute an instruction. During the fetch, a selected instruction (one, two or three bytes) is extracted from memory and deposited in the CPU's operating registers. During the execution part, the instruction is decoded and translated into specific processing activities.

Every instruction cycle consists of one, two, three, four or five machine cycles. A machine cycle is required each time the CPU accesses

memory or an I/O port. The fetch portion of an instruction cycle requires one machine cycle for each byte to be fetched. The duration of the execution portion of the instruction cycle depends on the kind of instruction that has been fetched. Some instructions do not require any machine cycles other than those necessary to fetch the instruction; other instructions, however, require additional machine cycles to write or read data to/from memory or I/O devices.

Each machine cycle consists of three, four or five states. A state is the smallest unit or processing activity and is defined as the interval between two successive positive-going transitions of the Ø1 clock pulse.

There are three exceptions to the defined duration of a state. They are the WAIT state, the hold (HLDA) state and the halt (HLTA) state. Because the WAIT, the HLDA, and the HLTA states depend upon external events, they are by their nature of indeterminate length. Even these exceptional states, however, must be synchronized with the pulses of the driving clock. Thus the duration of all states, including these, are integral multiples of the clock pulse.

To summarize, then, each clock period marks a state; three to five states summarize a machine cycle; and one to five machine cycles comprise an instruction cycle. A full instruction cycle requires anywhere from four to seventeen states for its completion, depending on the kind of instruction involved.

There is just one consideration that determines how many machine cycles are required in any given instruction cycle: the number of times that the processor must reference a memory address or an I/O address, in order to fetch and execute the instruction. Like many processors, the 8080 is so constructed that

it transmits one address per machine cycle. Thus, if the fetching and execution of an instruction requires two memory references, then the instruction cycle associated with that instruction consists of two machine cycles. If five such references are called for, then the instruction cycle contains five machine cycles.

Every instruction cycle has at least one reference to memory, during which the instruction is fetched. An instruction cycle must always have a fetch, even if the execution of instruction requires no further references to memory. The first machine cycle in every instruction cycle is therefore a FETCH. Beyond that, there are no fast rules. It depends on the kind of instruction. The input (INP) and the output (OUT), instructions each require three machine cycles: a FETCH, to obtain the instruction; a MEMORY READ, to obtain the address of the object peripheral; and an INPUT or an OUTPUT machine cycle, to complete the transfer.

Every machine cycle within an instruction cycle consists of three to five active states (referred to as T1, T2, T3, T4, and T5). The actual number of states depends upon the instruction being executed, and on the particular machine cycle within the greater instruction cycle. Figure 3-1 shows the timing relationships in a typical FETCH machine cycle. Events that occur in each state are referred to transitions of the $\phi 1$ and $\phi 2$ clock pulses.

At the beginning of each machine cycle (in state T1), the 8080 activates its SYNC output and issues status information on its data bus. The 8224 accepts SYNC and generates an active-low status strobe (STSTB/) as soon as the status data is stable on the data bus. The status information indicates the type of machine cycle in progress.

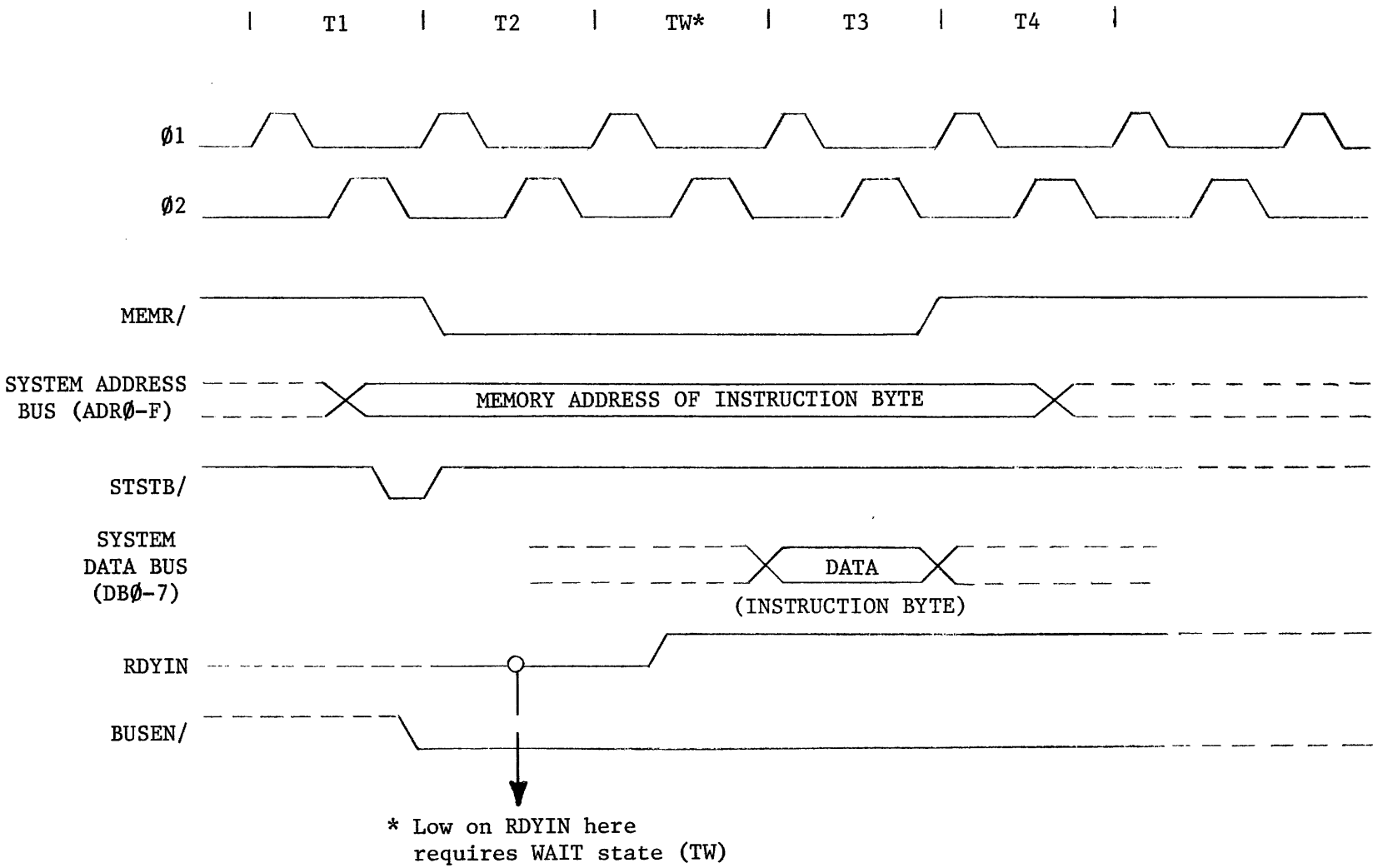


FIGURE 3-1. TYPICAL FETCH MACHINE CYCLE

The 8238 System Controller accepts the status bits from the 8080 and STSTB/ from the 8224, and uses them to generate the appropriate control signals (MEMR/, MEMW/, IOR/ and IOWR/) for the current machine cycle.

The rising edge of ∅2 during T1 loads the processor's address lines (A0 - A15). These lines become stable within a brief delay of the ∅2 clocking pulse, and they remain stable until the first ∅2 pulse after state T3. This gives the processor ample time to read the data returned from memory.

Once the processor has sent an address to memory, there is an opportunity for the memory to request a WAIT. This it does by pulling the 8224's RDYIN line low. As long as the RDYIN line remains low, the CPU Set will idle, giving the memory time to respond to the addressed data request. The 8224 synchronizes RDYIN with internal processor timing and applies the result to the 8080's READY input. The processor responds to a wait request by entering an alternative state (TW) at the end of T2, rather than proceeding directly to the T3 state. A wait period may be of indefinite duration. The 8080 remains in the waiting condition until its READY line again goes high. The cycle may then proceed, beginning with the rising edge of the next ∅1 clock. A WAIT interval will therefore consist of an integral number of TW states and will always be a multiple of the clock period.

The events that take place during the T3 state are determined by the kind of machine cycle in progress. In a FETCH machine cycle, the CPU Set interprets the data on its data bus as an instruction. During a MEMORY READ, signals on the same bus are interpreted as a data word.

The CPU Set itself outputs data on this bus during a MEMORY WRITE machine cycle. And during I/O operations, the CPU Set may either transmit or receive data, depending on whether an INPUT or an OUTPUT operation is involved. Consider the following two examples.

Figure 3-2 illustrates the timing that is characteristic of an input instruction cycle. During the first machine cycle (M1), the first byte of the two-byte IN instruction is fetched from memory. The 8080 places the 16-bit memory address on the system bus near the end of state T1. The 8238 activates the memory read control signal (MEMR/) during states T2 and T3 (and any intervening wait states, if required). During the next machine cycle (M2), the second byte of the instruction is fetched. During the third machine cycle (M3), the IN instruction is executed. The 8080 duplicates the 8-bit I/O address on address lines ADR0-7 and ADR8-F. The 8238 activates the I/O read control signal (IOR/) during states T2 and T3 of this cycle. In all cases the system bus enable input (BUSEN/) to the 8238 allows for normal operation of the data bus buffers and the read/write control signals. If BUSEN/ goes high the data bus output buffers and control signal buffers are forced into a high-impedance state.

Figure 3-3 illustrates an instruction cycle during which the CPU Set outputs data. During the first two machine cycles (M1 and M2), the CPU Set fetches the two-byte OUT instruction. During the third machine cycle (M3), the OUT instruction is executed. The 8080 duplicates the 8-bit I/O address on lines ADR0-7 and ADR8-F. The 8238 activates an advanced I/O write control signal (IOWR/) at the beginning of state T2 of this cycle. The nature and implications of the 8238 timing

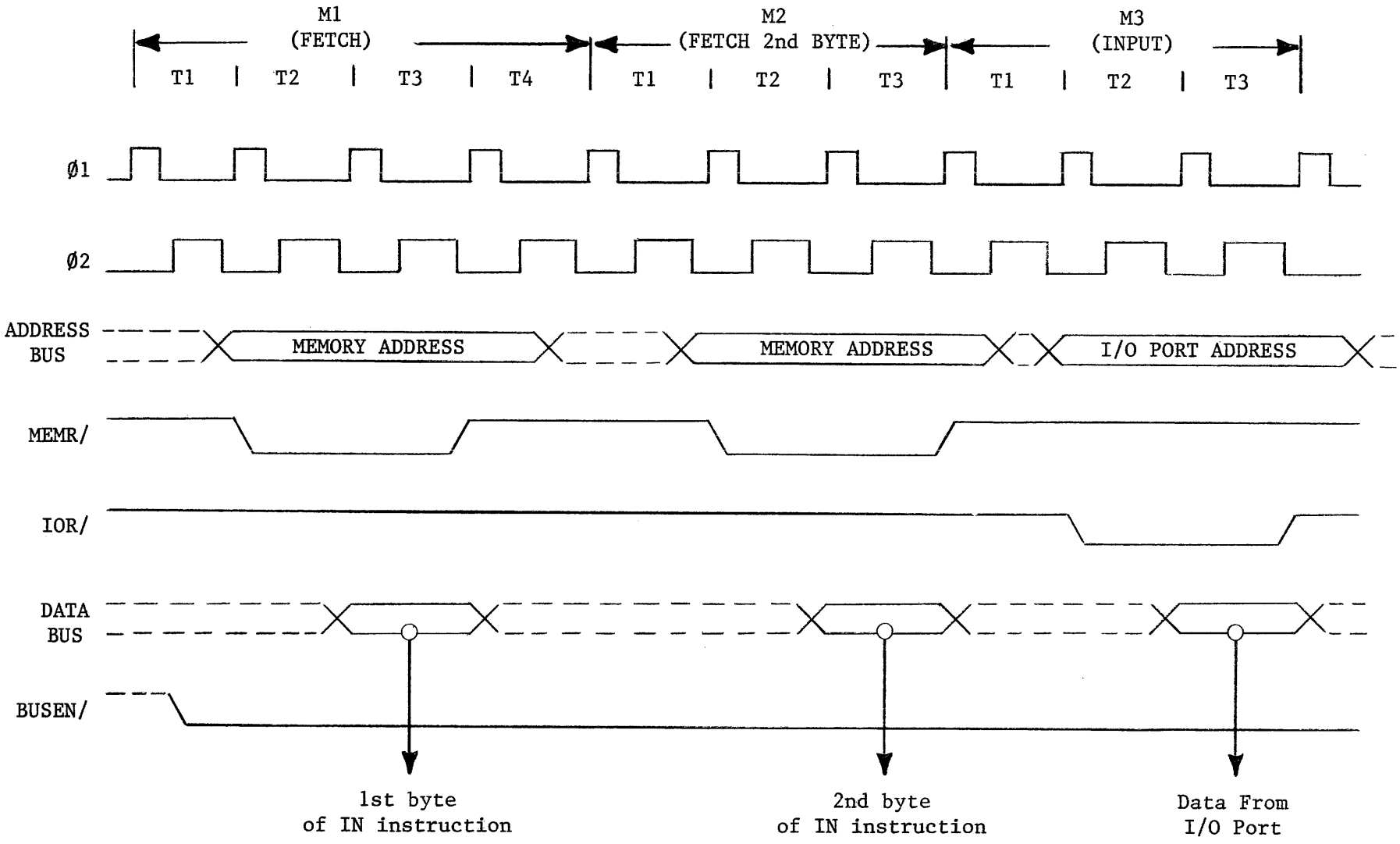


FIGURE 3-2. INPUT INSTRUCTION CYCLE

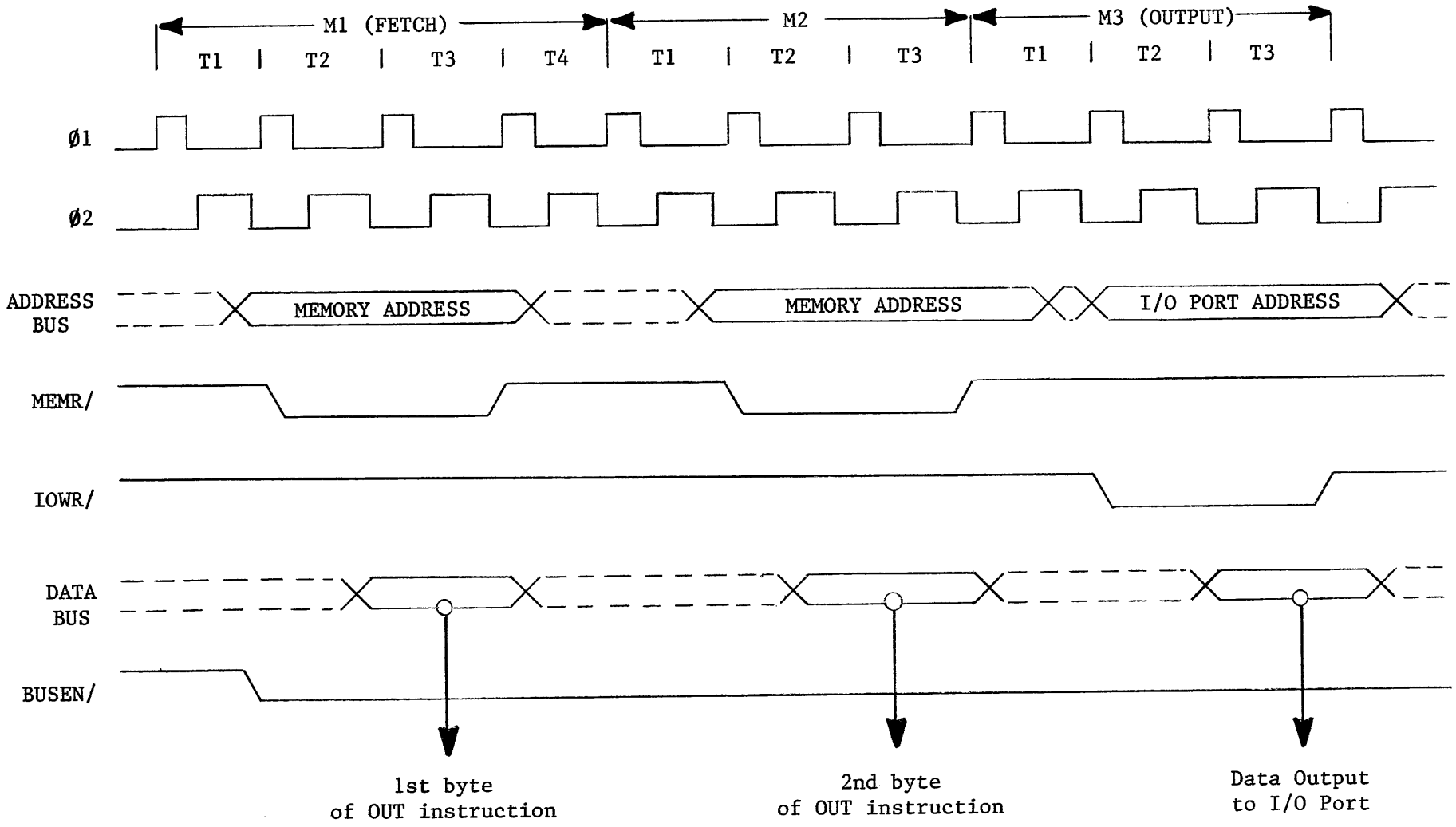


FIGURE 3-3. OUTPUT INSTRUCTION CYCLE

will be explained later (p. 3-17). The 8238 outputs the data onto the system bus at the end of state T2. Data on the bus remains stable throughout the remainder of the machine cycle. BUSEN/ must be low to prevent the output and control buffers from being forced into the high-impedance state.

Observe that a RDYIN signal is necessary for completion of an output machine cycle. Unless such an indication is present, the processor enters the TW state, following the T2 state. Data on the output lines remains stable in the interim, and the processing cycle will not proceed until the RDYIN line again goes high.

The 8080 generates a WR/ output for qualification of the advanced I/O write (IOWR/) and memory write (MEMW/) control signals from the 8238, during those machine cycles in which the CPU Set outputs data. The negative-going leading edge of WR/ is referred to the rising edge of the first ∅1 clock pulse following T2. WR/ remains low until re-triggered by the leading edge of ∅2, during the state following T3. Note that any TW states intervening between T2 and T3 of the output machine cycle will necessarily extend WR/.

All processor machine cycles consist of at least three states: T1, T2, and T3 as just described. If the CPU Set has to wait for a RDYIN response, then the machine cycle may also contain one or more TW states. During the three basic states, data is transferred to or from the CPU Set.

After the T3 state, however, it becomes difficult to generalize. T4 and T5 states are available, if the execution of a particular instruction requires them. But not all machine cycles make use of these states. It depends upon the kind of instruction being executed, and

on the particular machine cycle within the instruction cycle. The processor will terminate any machine cycle as soon as its processing activities are completed, rather than proceeding through the T4 and T5 states every time. Thus the 8080 may exit a machine cycle following the T3, the T4, or the T5 state and proceed directly to the T1 state of the next machine cycle.

3.1.2 INTERRUPT SEQUENCES

The 8080 has the built-in capacity to handle external interrupt requests. Peripheral logic can initiate an interrupt simply by driving the processor's interrupt (INT) line high. The interrupt (INT) input is asynchronous, and a request may therefore originate at any time during any instruction cycle. Internal logic re-clocks the external request, so that a proper correspondence with the driving clock is established. An interrupt request (INT) arriving during the time that the interrupt enable line (INTE) is high, acts in coincidence with the ∅2 clock to set the internal interrupt latch. This event takes place during the last state of the instruction cycle in which the request occurs, thus ensuring that any instruction in progress is completed before the interrupt can be processed.

The INTERRUPT machine cycle which follows the arrival of an enabled interrupt request resembles an ordinary FETCH machine cycle in most respects. The contents of the program counter are latched onto the address lines during T1, but the counter itself is not incremented during the INTERRUPT machine cycle, as it otherwise would be. In this way, the pre-interrupt status of the program counter is preserved, so that data in the counter may be saved in the stack.

This in turn permits an orderly return to the interrupted program after the interrupt request has been processed.

Because the 8238's INTA/ output (pin 23) is tied to +12 volts, the 8238 blocks incoming data and automatically inserts a Restart (RST 7) instruction onto the 8080 data bus during state T3, when the interrupt is acknowledged by the 8080. RST is a special one-byte call instruction that facilitates the processing of interrupts (the ordinary program call instruction is three bytes long). The RST 7 instruction causes the 8080 to branch program control to the instruction being stored in memory location 38_{16}.

3.1.3 HOLD SEQUENCES

By activating the 8080's HOLD input, an external device can cause the CPU Set to suspend its normal operations and relinquish control of the address and data busses. The CPU Set responds to a request of this kind by floating its address and data outputs, so that these exhibit a high impedance to other devices sharing the busses. At the same time, the processor acknowledges the HOLD by placing a high on on its HLDA output pin. During an acknowledged HOLD, the address and data busses are under control of the peripheral which originated the request, enabling it to conduct memory transfers without processor intervention.

3.1.4 HALT SEQUENCES

When a halt instruction (HLT) is executed, the 8080 enters the halt state after state T2 of the next machine cycle. There are only three ways in which the 8080 can exit the halt state:

- A high on the 8224 reset input (RESIN/) will always reset the 8080 to state T1; reset also clears the program counter.
- A HOLD input will cause the 8080 to enter the hold state, as previously described. When the HOLD line goes low, the 8080 re-enters the halt state on the rising edge of the next ∅1 clock pulse.
- An interrupt (i.e., INT goes high while INTE is enabled) will cause the 8080 to exit the halt state and enter state T1 on the rising edge of the next ∅1 clock pulse. NOTE: The interrupt enable (INTE) flag must be set when the halt state is entered; otherwise, the 8080 will only be able to exit via a reset signal.

3.1.5 START-UP SEQUENCE

When power is applied initially to the 8080, the processor begins operating immediately. The contents of its program counter, stack pointer, and the other working registers are naturally subject to random factors and cannot be specified. For this reason, the CPU Set power-up sequence begins with a reset. An external RC network is connected to the 8224's RESIN/ input. The slow transition of the power supply rise is sensed by an internal Schmitt Trigger which converts the slow transition into a clean, fast edge on the RESIN/ line when the input level reaches a predetermined value.

An active RESIN/ input to the 8224 produces a synchronized RESET signal which restores the processor's internal program counter to zero. Program execution thus begins with memory location zero, following

a reset. Systems which require the processor to wait for an explicit start-up signal will store a halt instruction (HLT) in this location. A manual or an automatic INTERRUPT will be used for starting. In other systems, the processor may begin executing its stored program immediately. Note, however, that the reset has no effect on status flags, or on any of the processor's working registers (accumulator, indices, or stack pointer). The contents of these registers remain indeterminate, until initialized explicitly by the program.

In addition to generating a RESET signal, the RESIN/ input causes the 8224's status strobe (STSTB/) output to remain true (low). This allows both the 8080 and 8238 to be reset by a power-up sequence or an externally generated RESIN/ condition.

3.2 SYSTEM BUS INTERFACE LOGIC

The System Bus Interface logic consists of three general groups of circuitry:

1) assorted gates that accept the various bus control signals, the interrupt request lines, the ready indications and then applies these signals to the CPU Set,

2) the system bus drivers, and

3) the Failsafe circuitry which generates an acknowledgment during interrupt sequences and during those cycles in which an acknowledgment is not returned because a non-existent device was inadvertently addressed.

Each group is described in the following paragraphs.

3.2.1 SYSTEM CONTROL SIGNAL LOGIC

Interrupt Requests:

Four interrupt request lines are ORed together at A17-6 (ref. Appendix A) and applied to the 8080's INT input. Two of the interrupt request lines are from external sources: EXT INTR 1/ which enters the SBC-80/10

at connector J1 pin 49 and EXT INTR 2/ which enters the SBC-80/10 at P1-42. The other two interrupt requests originate on the SBC-80/10: INT 55/ is an interrupt request from ports 1 or 2 in the Parallel I/O Interface (see Section 3.6.2); and INT 51/ is an interrupt request from the 8251 USART in the Serial I/O Interface (see Section 3.5.4).

Hold Requests:

If the SBC-80/10 is operating in a system with other modules sharing a common external bus, another module can acquire control of the external bus by activating the 8080's HOLD/ input (connector pin P1-15). HOLD/ is inverted and applied to the 8080's HOLD pin. As described in Section 3.1.3, the 8080 will subsequently activate its hold acknowledge (HLDA) output. HLDA is, in turn, latched by a 74LS74 flip flop (at A29). The Q output from the D-type latch (DHLDA) disables the 8097circuits (A47) that drive the external read/write control outputs: MRDC/, MWTC/, IORC/ and IOWC/. DHLDA also disables the external system address and data bus drivers by asserting a high at their active-low chip select (CS/) input pins. As a result of DHLDA, all of the above-mentioned drivers enter the high-impedance state. The $\overline{Q}$ output from the DHLDA output informs other modules of this condition via the BUSY/ output (connector pin P1-17). BUSY/ is driven by transistor Q5.

System Reset:

Connector pin P1-14 on the SBC-80/10 can be used to accept an externally generated SYSTEM RESET signal and to transfer a SBC-80/10

generated RESET signal to other modules in the system. If jumper pair 54-55 is connected, a RESET from the 8224 will be gated through the Q4 transistor to connector pin P1-14, thus resetting other modules in the system during power-up sequences. An externally generated SYSTEM RESET is accepted at P1-14, buffered, applied to the 8080's RESET input and made available to other logic on the SBC-80/10.

I/O Ready Generation

During each serial or parallel I/O cycle, a "ready" indication (IORDYIN/) is returned to the CPU Set. The three chip select lines for the 8251 and the two 8255 devices are ORed together (at A17-8 on sheet 3 of the schematic). The resultant output is then NANDed (at A44-11) with the I/O read (IOR) or the advanced I/O write (ADV IOW) signal to produce IORDYIN/. Recall from Section 3.1 that the 8238 System Controller (in the CPU Set) generates the I/O write control output at the beginning of all I/O write cycles. The IOW/ signal occurs earlier than the 8080's WR/ output. The 8238's IOW/ signal, alone, is labeled ADV IOW/. IOW/ is also synchronized with the 8080's WR/ output to produce the system write command IOWC/. ADV IOW/ allows the ready indication to be returned early enough to avoid an unnecessary wait state (see Figure 3-4). The IOWC/ signal causes an I/O device to actually write the data, later in the I/O cycle.

Ready Inputs:

Recall from Section 3.1.1 that the CPU Set must see a ready indication before proceeding to internal state T3 during all machine cycles. The 74S20 section at A57 on sheet 1 of the schematic OR's

the following ready indications:

1) INT ACK/ or TIME OUT ACK/ from the Failsafe logic (see Section 3.2.3),

2) IORDYIN/ from the Serial and Parallel I/O Interfaces,

3) PROM RDYIN/ from the ROM/PROM logic (see Section 3.4), and

4) RAM RDYIN/ from the RAM section (see Section 3.3).

The resultant output indicates an on-board memory or I/O access and is used to disable the external data bus drivers at A53 and A54. This output from A57-8 is also ORed (at A30-3) with the externally generated AACK/ (connector pin P1-25) and XACK/ (connector pin P1-23) inputs. The output from A30-3 is then applied to the CPU Set's RDYIN input (pin 3 on the 8224). When the SBC-80/10 CPU Set accesses an external module, the AACK/ or XACK/ input informs the CPU Set that the external device is ready. AACK/ is an advanced acknowledge that allows certain OEM modules to be accessed faster.

Figure 3-4 illustrates basic timing for the ready indications.

Bus Clock Generation:

The OSC output from the CPU Set (18.432 MHz frequency) is applied to the clock input of a 74LS74 D-type flip flop (at A29-11 on sheet 1 of the schematic). The $\overline{Q}$ output from this latch is tied to its own D input. Consequently, the Q output exhibits half the frequency of the OSC input. This 9.216 MHz output is buffered and made available to external modules on the common clock (CCLK/) line (via connector pin P1-31) and the bus clock (BCLK/) line (via connector pin P1-13).

3.2.2 SYSTEM BUS DRIVERS

The SBC-80/10 internal memory data bus (DM0-DM7) is driven by

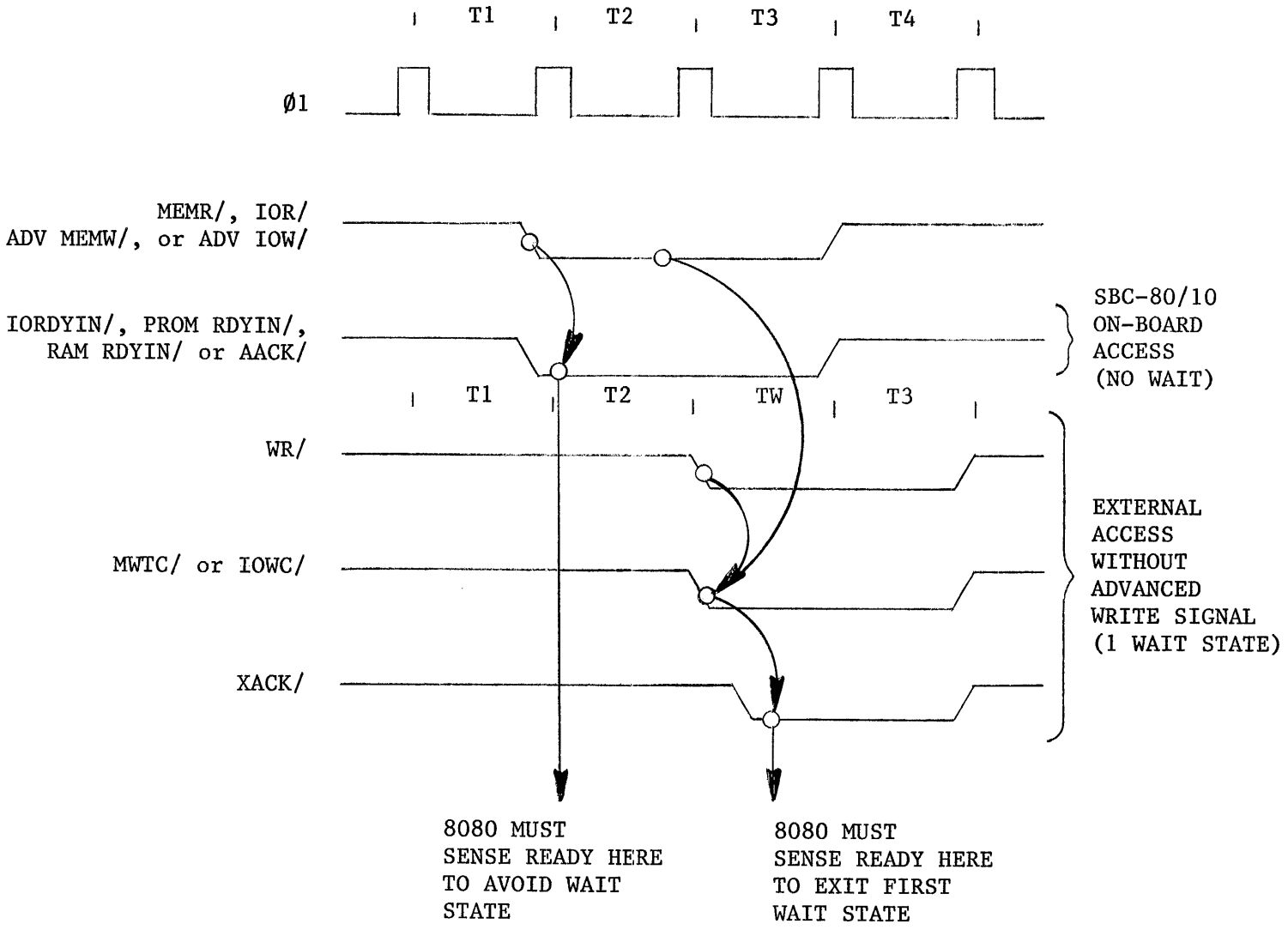


FIGURE 3-4. READY TIMING

two 8216 bidirectional bus drivers, shown at A55 and A56 on sheet 3 of the schematic. All data being transferred to/from the RAM memory (see Section 3.3) or ROM/PROM memory (see Section 3.4) is routed through these two devices. The chip select (CS/) input is provided by the MEM CMD/ signal which is the result of ORing RAM RDYIN/ and PROM RDYIN/. The direction enable (DIEN) input to the 8216's is provided by the memory read (MEMR) signal.

When the SBC-80/10 communicates with an external module, the data is driven by two 8226 bidirectional data bus drivers at A53 and A54 on sheet 1 of the schematic. The direction input to the 8226's is provided by the OR of memory read (MEMR) and I/O read (IOR). The 8226 devices will be disabled during 8080 HOLD sequences. The eight data bus lines to the 8226 bus drivers enter/leave the SBC-80/10 via the P1 edge connector.

The external 16-bit system address bus is driven by four 8226 bidirectional bus drivers. However, because the direction enable pin (EN/) on these 8226 devices is tied to ground, they can only be used to transmit addresses to external modules; they will not receive addresses from external modules. Consequently, the SBC-80/10 can access other modules, but other modules cannot access the memory or I/O controllers on the SBC-80/10. Like the data bus drivers, these 8226 devices are disabled during 8080 HOLD sequences.

3.2.3 FAILSAFE TIMER

When the 8080 acknowledges an interrupt request, the 8238 System Controller "forces" an RST 7 instruction onto the 8080's data bus

(see Section 3.1.2). In order to read this RST 7 instruction, however, the 8080 must sense a ready indication. The 8080 acknowledges an interrupt by setting status bit 0 (DO) during the status output portion of each machine cycle (i.e., when STATUS STROBE is true). When this occurs, the 9602 one-shot (shown at A28 on sheet 5 of the schematc) is reset causing a low signal on its output (INTR ACK/). This output is then gated through to the RDYIN pin on the 8224 as described in Section 3.2.1.

The Failsafe timer also performs another function. If the CPU Set tries to access a memory or I/O device but that device, for some reason, does not return a ready indication, then the 8080 remains in a wait state until ready is received. The Failsafe timer is designed to prevent hanging the system up in this way. The 9602 one-shot is triggered by STATUS STROBE at the beginning of each machine cycle. If the one-shot is not re-triggered (i.e., if another cycle does not begin) within 9 ms., then the 9602 times out and its output (also labeled TIME OUT ACK/) is gated through to the RDYIN pin on the 8224, thus allowing the 8080 to exit the wait state. This can be very helpful during system debugging.

3.3 RANDOM ACCESS MEMORY (RAM)

The Random Access Memory (RAM) provides the SBC-80/10 user with 1024 (1k) $\times$ 8-bits of read/write storage that requires no clocks or refreshing to operate. The RAM logic consists of eight Intel 8111 256 $\times$ 4 -bit Static MOS RAM chips, an Intel 3205 three-to-eight decoder for chip selection and assorted gates as shown on sheet 2 of the SBC-80/10 schematic (Appendix A).

The 8111 RAM devices used on the SBC 80/10 have a maximum access time of 500 nsec. Each chip has eight address inputs (A0-A7) that select one of the 256 four-bit segments, active-low write (W/) and chip enable (CE/) inputs and an output disable (OD) input. Each chip also has four common data input/output pins (I/01-I/04). A high on the OD input disables output and allows the I/O pins to be used for input. During memory read accesses, the data is read out nondestructively and has the same polarity as the input data.

The least significant system address lines (ADR0-ADR7) are applied to the eight address input pins on each 8111 RAM. The most significant eight system address lines (ADR8-ADRF) feed a 3205 decoder. Each of the four most significant decoder outputs are applied to the chip enable (CE/) inputs on two RAM chips. One RAM in each pair reads or writes data bits 0 to 3 (DM0-DM3) while the other RAM reads or writes data bits 4 to 7 (DM4-DM7) for each RAM access. One of the decoder outputs will be activated (low) whenever the value on the system address bus is within the range 3C00-3FFF (hexadecimal).

During memory write cycles, the advanced memory write signal (ADV MEMW/) is applied to the write input (W/) on each RAM. A high on the active-low memory read line (MEMR/) allows the selected RAM's I/O pins to be used to accept the data which is to be written into the addressed location. During memory read cycles, the level on ADV MEMW/ is high but is low on MEMR/ thus allowing the addressed data to be read out and onto the data bus.

During all RAM access cycles, the active decoder output is NANDed with ADV MEMW or MEMR (at A44-3) to produce a ready indication

for the CPU Set (RAM RDYIN/). The 8238 System Controller (see Section 3.1) generates ADV MEMW or MEMR early enough in the memory cycle to allow RAM RDYIN/ to appear at the CPU Set in time to prevent the occurrence of any wait states. Figure 3-5 illustrates RAM access timing.

Whenever SBC-80/10 RAM is accessed, the data is transferred to/from the RAM chips on the memory data bus (DM0-DM7). Lines DM0-DM7 are interfaced to the system data bus through two Intel 8216 bidirectional bus drivers (shown at A55 and A56 on sheet 3 of the schematic) as described in Section 3.2.

3.4 READ ONLY MEMORY (ROM/PROM)

The SBC-80/10 has provision for installing 4096 (4K) × 8-bit words of read only memory in sockets already on the PC board. Four Intel 8708 1K × 8-bit Erasable and Electrically Programmable Read Only Memory (PROM) chips or four 8308 1K × 8-bit static MOS mask Read Only Memory (ROM) chips can be installed in the four 24-pin sockets shown on sheet 3 of the SBC-80/10 schematic (Appendix A).

In addition to the four 24-pin sockets, the ROM/PROM logic includes an Intel 3205 decoder for address decoding and several assorted gates used in generating the ready indication.

Address lines ADR0-ADR9 are applied to the address pins A0-A9 at each of the four sockets. The remaining address lines, ADRA-ADRF are decoded by the 3205 device at A42. Each of the four least significant decoder outputs are applied to the chip select (CS/) pin at one of four sockets. One chip select line will be activated whenever the value on the system address bus is between 0000 and 0FFF (hexadecimal).

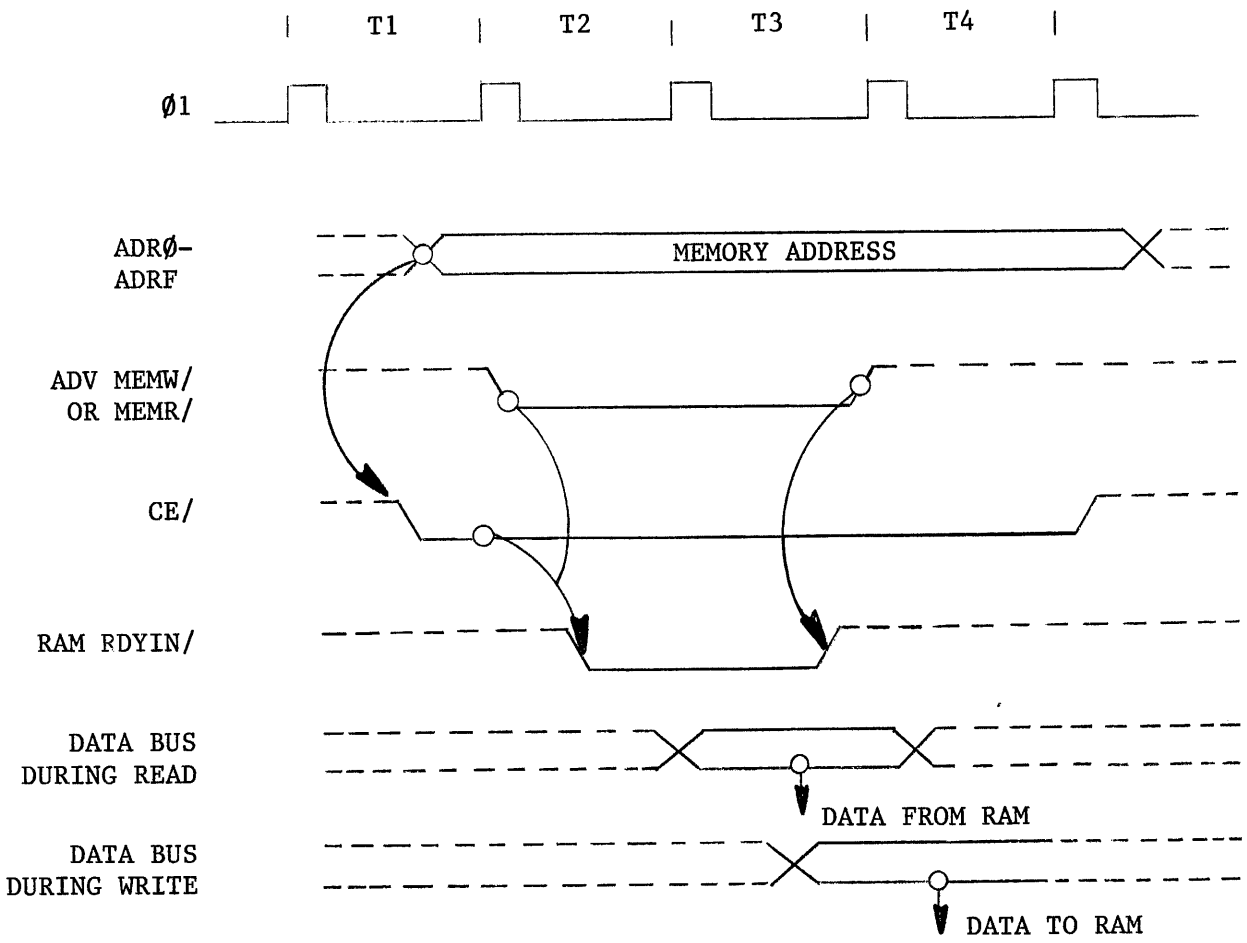


FIGURE 3-5. RAM ACCESS TIMING

In addition, when the four most significant address lines are low (i.e., the address is less than 0FFF) during a memory read cycle, the output from the 74LS00 section at A39-3 is NANDed with MEMR to produce a ready indication (PROM RDYIN/) for the CPU Set. PROM RDYIN/ is thus generated in time to allow all ROM/PROM reads to occur without any wait states. PROM RDYIN/ has the same timing as RAM RDYIN/, as shown in Figure 3-5.

Whenever one of the ROM/PROM devices are read, the data from the chips output pins (01-08) is placed on the memory data bus (DM0-DM7) which is interfaced to the system bus via two Intel 8216 bidirectional bus drivers (at A55 and A56), as described in Section 3.2.

3.5 SERIAL I/O INTERFACE

The Serial I/O Interface logic provides the SBC-80/10 with a serial data communications channel that can be programmed to operate with most of the current serial data transmission protocols, synchronous or asynchronous. Baud rate, character length, number of stop bits and even/odd parity are program selectable. In addition, the serial I/O Interface can be configured (through jumper connections) as an EIA RS232C interface or as a Teletype-compatible current loop interface.

The Serial I/O Interface logic consists primarily of an Intel 8251 USART device and a counting network for baud rate selection, as shown on sheet 4 of the SBC-80/10 schematic (Appendix A). Before describing the specific operation of the Serial I/O logic however, we will summarize the general operational characteristics of the 8251

USART, because it essentially defines the character of the Serial I/O Interface.

3.5.1 INTEL 8251 OPERATIONAL SUMMARY

The 8251 is a Universal Synchronous/Asynchronous Receiver/Transmitter designed specifically for the 8080 Micro-computer System. Like other I/O devices in the 8080 Micro-computer System its functional configuration is programmed by the systems software for maximum flexibility. The 8251 can support virtually any serial data technique currently in use (including IBM "Bi-Sync").

Modem Control

The 8251 has a set of control inputs and outputs that can be used to simplify the interface to almost any Modem. The modem control signals are general purpose in nature and can be used for functions other than Modem control, if necessary.

$\overline{\text{DSR}}$ (Data Set Ready)

The $\overline{\text{DSR}}$ input signal is general purpose in nature. Its condition can be tested by the CPU using a Status Read operation. The $\overline{\text{DSR}}$ input is normally used to test Modem conditions such as Data Set Ready.

$\overline{\text{DTR}}$ (Data Terminal Ready)

The $\overline{\text{DTR}}$ output signal is general purpose in nature. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{\text{DTR}}$ output signal is normally used for Modem control such as Data Terminal Ready or Rate Select.

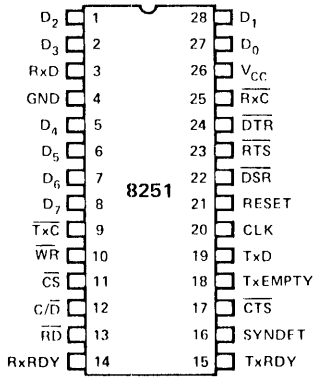
$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output signal is general purpose in nature. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{RTS}$ output signal is normally used for Modem control such as Request to Send.

$\overline{CTS}$ (Clear to Send)

A "low" on this input enables the 8251 to transmit data (serial) if the TxEN bit in the Command byte is set to a "one". This is very important to remember!

USART
PIN CONFIGURATION



Pin Name	Pin Function
D_7-D_0	Data Bus (8 bits)
C/D	Control or Data is to be Written or Read
RD	Read Data Command
WR	Write Data or Control Command
CS	Chip Enable
CLK	Clock Pulse (TTL)
RESET	Reset
TxC	Transmitter Clock
TxD	Transmitter Data
RxC	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready (has character for 8080)
TxRDY	Transmitter Ready (ready for char. from 8080)

Pin Name	Pin Function
DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDET	Sync Detect
RTS	Request to Send Data
CTS	Clear to Send Data
TxE	Transmitter Empty
V_{CC}	+5 Volt Supply
GND	Ground

FIGURE 3-6. 8251 PIN ASSIGNMENTS

$\overline{TXRDY}$ (Transmitter Ready)

This output signals the CPU that the transmitter is ready to accept a data character. It can be used as an interrupt to the system or for polled operation when the CPU can check TXRDY using a status read operation. $\overline{TXRDY}$ is active only when $\overline{CTS}$ is enabled. $\overline{TXRDY}$ is automatically reset when a character is loaded from the CPU.

TXE (Transmitter Empty)

When the 8251 has no characters to transmit, the TxE output will go "high". It resets automatically upon receiving a character from the CPU. TXE can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplexed operational mode.

In SYNChronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be transmitted automatically as "fillers".

$\overline{\text{TXC}}$ (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the frequency of $\overline{\text{TXC}}$ is equal to the actual Baud Rate (1X). In Asynchronous transmission mode, the frequency of $\overline{\text{TXC}}$ is a multiple of the actual Baud Rate. A portion of the mode instruction selects the value of the multiplier; it can be 1X, 16X or 64X the Baud Rate.

For Example:

If Baud Rate equals 110 Baud,

$\overline{\text{TXC}}$ equals 110 Hz (1X)

$\overline{\text{TXC}}$ equals 1.76 kHz (16X)

$\overline{\text{TXC}}$ equals 7.04 kHz (64X).

If Baud Rate equals 9600 Baud,

$\overline{\text{TXC}}$ equals 614.4 kHz (64X).

The falling edge of $\overline{\text{TXC}}$ shifts the serial data out of the 8251.

RXRDY (Receiver Ready)

This output indicates that the 8251 contains a character that is ready to be input to the CPU. RXRDY can be connected to the interrupt structure of the CPU or for polled operation the CPU can check the condition of RXRDY using a status read operation. RXRDY is automatically reset when the character is read by the CPU.

$\overline{\text{RXC}}$ (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the frequency of $\overline{\text{RXC}}$ is equal to the actual Baud Rate (1X). In Asynchronous Mode, the frequency of $\overline{\text{RXC}}$ is a multiple of the actual Baud Rate. A portion of the mode instruction selects the value of the multiplier; it can be 1X, 16X or 64X the Baud Rate.

For Example:

If Baud Rate equals 300 Baud,
$\overline{\text{RXC}}$ equals 300 Hx (1X)
$\overline{\text{RXC}}$ equals 4800 Hz (16X)
$\overline{\text{RXC}}$ equals 19.2 kHz (64X).
If Baud Rate equals 2400 Baud,
$\overline{\text{RXC}}$ equals 2400 Hz (1X)
$\overline{\text{RXC}}$ equals 38.4 kHz (16X)
$\overline{\text{RXC}}$ equals 153.6 kHz (64X).

Data is sampled into the 8251 on the rising edge of $\overline{\text{RXC}}$.

Note: In most communications systems, the 8251 will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the Same. Both $\overline{\text{TXC}}$ and $\overline{\text{RXC}}$ will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

SYNDET (SYNC Detect)

This pin is used in SYNCHronous Mode only. It is used as either input or output, programmable through the Control Word. It is reset to "low" upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "high" to indicate that the 8251 has located the SYNC character in the Receive mode. If the 8251 is programmed to use double Sync characters, then SYNDET will go "high" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

When used as an input, (external SYNC detect mode), a positive going signal will cause the 8251 to start assembling data characters on the falling edge of the next $\overline{RXC}$. Once in SYNC, the "high" input signal can be removed. The duration of the high signal should be at least equal to the period of $\overline{RXC}$.

Programming the 8251

Prior to starting data transmission or reception, the 8251 must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251 and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction,
2. Command Instruction.

Both the Mode and Command instructions must conform to a specified sequence for proper device operation. The Mode Instruction must be inserted immediately following a Reset operation, prior to using the 8251 for data communication.

All control words written into the 8251 after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251 at any time in the data block during the operation of the 8251. To return to the Mode Instruction format a bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251 back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters (see Figure 3-7).

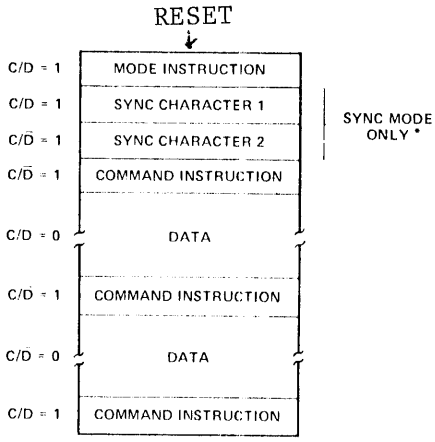


FIGURE 3-7. TYPICAL 8251 DATA BLOCK

Mode Instruction:

This format defines the general operational characteristics of the 8251. It must follow a Reset operation (internal or external). Once the Mode instruction has been written into the 8251 by the CPU, SYNC characters or Command instructions may be inserted.

The 8251 can be used for either synchronous or asynchronous

data communications. The two least significant bits of the Mode Instruction control word specify synchronous or asynchronous operation. The format for the remaining bits in the control word depends on the mode chosen by bits 0 and 1. Figure 3-8 shows the control word format for the asynchronous mode, while Figure 3-9 illustrates the control word format for the synchronous mode.

Command Instruction:

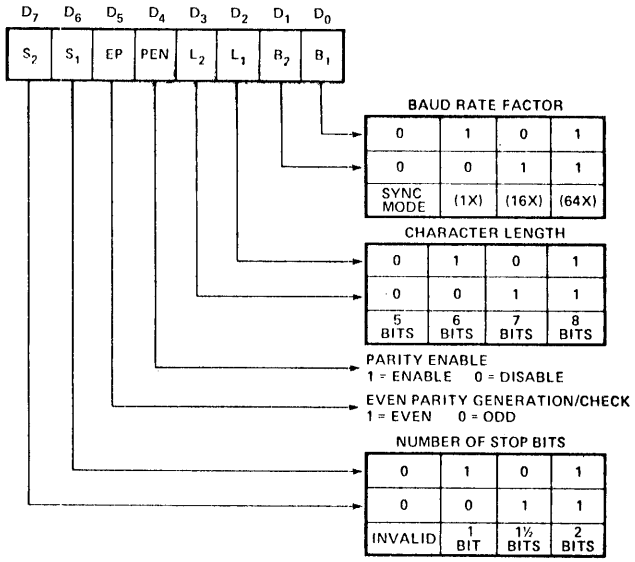
Once the functional definition of the 8251 has been programmed by the Mode Instruction and the Sync Characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command Instruction.

Once the Mode instruction has been written into the 8251 and Sync characters inserted, if necessary, then all further "control writes" ($C/\overline{D} = 1$) will load the Command Instruction. A Reset operation (internal or external) will return the 8251 to the Mode Instruction Format.

Figure 3-10 illustrate the format of a Command Instruction control word.

Status Read Definition

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251 has facilities that allow the programmer to "read" the status of the device at any time during the functional operation.



Mode Instruction Format, Asynchronous Mode

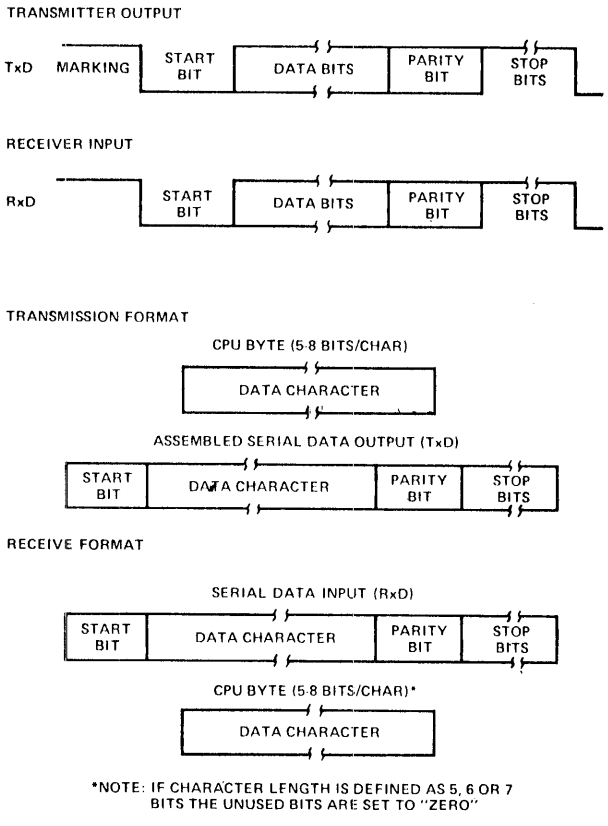
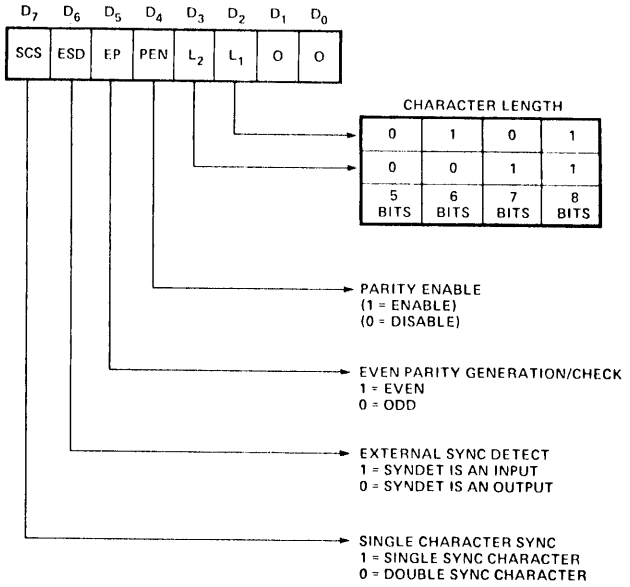


FIGURE 3-8. ASYNCHRONOUS MODE.



Mode Instruction Format, Synchronous Mode

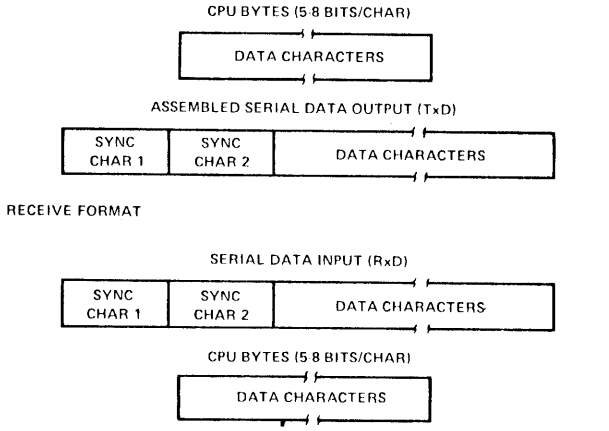


FIGURE 3-9. SYNCHRONOUS MODE.

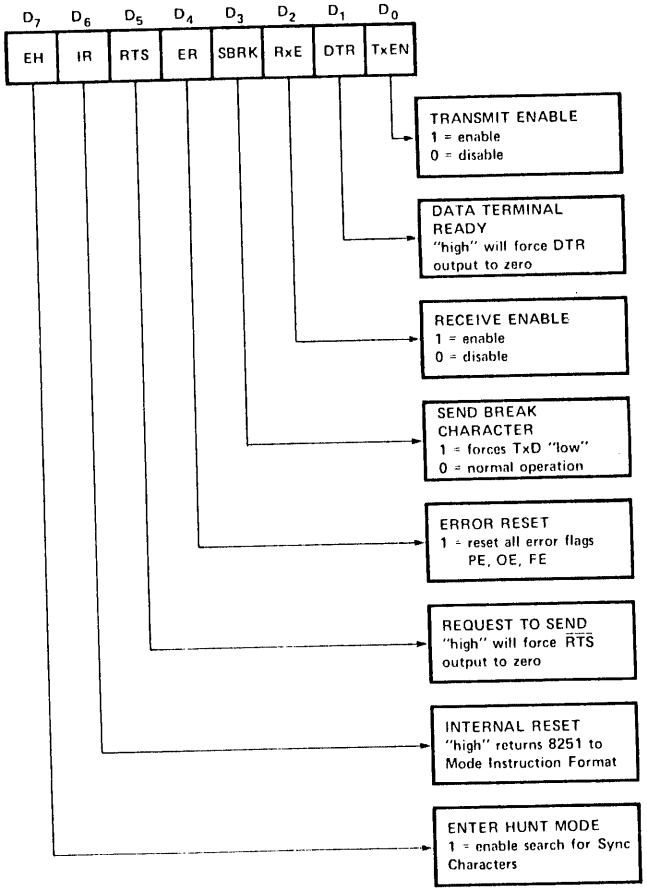


FIGURE 3-10. COMMAND INSTRUCTION FORMAT

A normal "read" command is issued by the CPU with the C/$\overline{D}$ input at one to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251 can be used in a completely Polled environment or in an interrupt driven environment (refer to Figure 3-11).

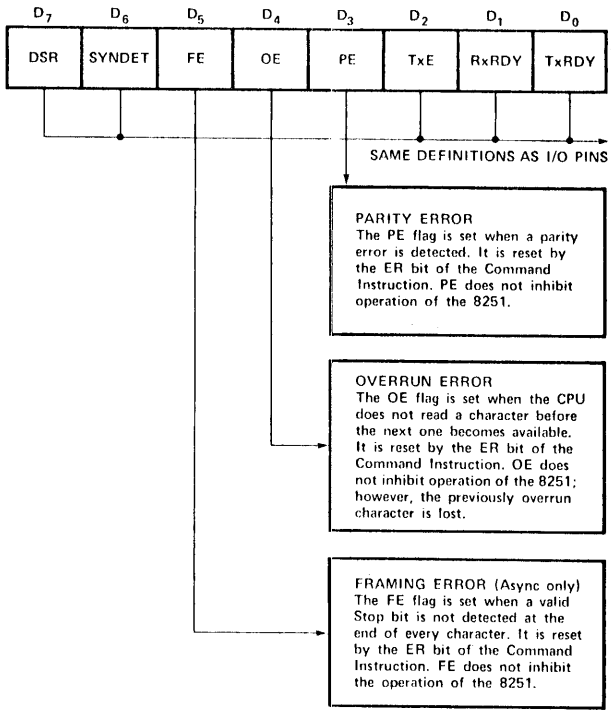


FIGURE 3-11. STATUS READ FORMAT

8251 DATA TRANSFERS

Once programmed, the 8251 is ready to perform its communication functions. The TXRDY output is raised "high" to signal the CPU that the 8251 is ready to receive a character. This output (TXRDY) is reset automatically when the CPU writes a character into the 8251. On the other hand, the 8251 receives serial data from the MODEM or

I/O device; upon receiving an entire character the RXRDY output is raised "high" to signal the CPU that the 8251 has a complete character ready for the CPU to fetch. RXRDY is reset automatically upon the CPU read operation.

The 8251 cannot begin transmission until the TXEN (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send (CTS) input. The TXD output will be held in the marking state upon Reset.

Asynchronous Mode (Transmission):

Whenever a data character is sent by the CPU the 8251 automatically adds a Start bit (low level) and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TXD output. The serial data is shifted out on the falling edge of $\overline{TXC}$ at a rate equal to 1/16 or 1/64 that of the $\overline{TXC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TXD if commanded to do so.

When no data characters have been loaded into the 8251 the TXD output remains "high" (marking) unless a Break (continuously low) has been programmed.

Asynchronous Mode (Receive):

The RXD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center. If a low is detected again, it is a valid START bit, and the bit

counter will start counting. The bit counter locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RXD pin with the rising edge of $\overline{RXC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. This character is then loaded into the parallel I/O buffer of the 8251. The RXRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN flag is raised (thus the previous character is lost). All of the error flags can be reset by a command instruction. The occurrence of any of these errors will not stop the operation of the 8251.

Synchronous Mode (Transmission):

The TXD output is continuously high until the CPU sends its first character to the 8251 which usually is a SYNC character. When the $\overline{CTS}$ line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TXC}$. Data is shifted out at the same rate as the $\overline{TXC}$.

Once transmission has started, the data stream at $\overline{TXD}$ output must continue at the $\overline{TXC}$ rate. If the CPU does not provide the 8251 with a character before the 8251 becomes empty, the SYNC characters (or character if in single SYNC word mode) will be automatically inserted in the TXC data stream. In this case, the TXEMPTY pin will momentarily go high to signal that the 8251 is empty and SYNC characters are being sent out. The TXEMPTY pin is internally reset by the next character being written into the 8251.

Synchronous Mode (Receive):

In this mode, character synchronization can be internally or externally achieved. If the internal SYNC mode has been programmed, the receiver starts in a HUNT mode. Data on the RXD pin is then sampled in on the rising edge of RXC. The content of the RX buffer is continuously compared with the first SYNC character until a match occurs. If the 8251 has been programmed for two SYNC characters, the subsequent received character is also compared. When both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ.

In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin. The high level can be removed after one $\overline{RXC}$ cycle.

Parity error and overrun error are both checked in the same way as in the Asynchronous receive mode.

The CPU can command the receiver to enter the HUNT mode if synchronization is lost.

3.5.2 Serial I/O Configurations

The 8251 USART presents a parallel, eight-bit interface to the CPU Set via the system data bus (DB0-DB7) and presents a EIA RS232C * or TTY current loop* interface to an external device (via edge connector J3). The 8251's interface with the CPU Set is enabled by a low

* Electrical interfaces provided on SBC 80/10

level on its chip select (CS/) pin. CS/ is low when the I/O address on the system address bus is between EC and EF (hexadecimal). Address bits 2 through 7 are decoded (at A14) to produce the CS/ input. The

TABLE 3-0. SERIAL COMMUNICATION (8251) ADDRESS ASSIGNMENTS

I/O ADDRESS (BASE 16)	COMMAND	FUNCTION
ED OR EF	OUTPUT	CONTROL WORD
EC OR EE	OUTPUT	DATA
ED OR EF	INPUT	STATUS
EC OR EE	INPUT	DATA

least significant address bit, ADR0, is applied to the 8251's C/$\overline{D}$ input (pin 12) thus indicating a control (if set) or data (if reset) byte on the data bus.

An output instruction (IOW/ is true) to port ED or EF (CS/ is low and ADR0 is high) causes the 8251 to accept a control byte through its data bus pins. The control byte can be either a mode instruction or a command instruction, depending on the sequence in which it is sent. The various bits in the mode control word specify the baud rate multiplexer, character length, parity and the number of stop bits as described in Section 3.5.1. Note that the actual baud rate selected is dependent on the configuration of the baud rate jumper network (refer to Section 3.5.3). The various bits in the command control word instruct the USART to enable/disable the receiver and transmitter, to reset errors, to reset internal control and return to the mode control cycle, and to set/clear the Data Terminal Ready output.

An output instruction to port EC or EE (CS/ and ADR0 are low) causes the 8251 USART to accept a data byte through its data bus pins. Bit 0 is the least significant bit and bit 7 is the most significant bit. The 8251 will subsequently transmit the data byte (if the transmitter is enabled), in serial fashion, to the external device as described in Section 3.5.1.

An input instruction (IOR/ is true) to port ED or EF (CS/ is low and ADR0 is high) causes the 8251 USART to place a status byte onto the system bus. The status bits are the result of status and error checking functions performed within the USART (see Section 3.5.1).

An input instruction (IOR/ is true) to port EC or EE (CS/ and ADR0 are low) causes the USART to output a data byte (previously

received from the external device) from its data bus pins. Bit 0 is the least significant bit and bit 7 is the most significant bit.

Timing for the USART's internal function is provided by the ϕ2TTL signal (see Section 3.1.1). The USART is reset by the occurrence of a high level on the RESET line.

The 8251 USART transmits and receives serial data, synchronously or asynchronously, as described in Section 3.5.1. By jumper-connecting the 8251 pins to different external lines, the Serial I/O logic can present either a Teletype-compatible current loop interface or an EIA RS232C interface to an external device. If the TTY-compatible current loop interface is used, the connections listed in Table 4-1 are required (see Section 4.1).

If the EIA RS232C interface is used, the connections listed in Table 4-2 are required (see Section 4.1).

3.5.3 BAUD RATE CLOCK GENERATION

The baud rate clock network consists of a 93S16 'divide-by-15' counter, two 74161 'divide-by-16' counters and wire-wrap jumpers for baud rate clock selection. The 93S16 counter is driven by the oscillator output (OSC) from the CPU Set. The QD output from this counter, in turn, drives the two 74161 counters. The outputs from these counters, each providing a different clock frequency, are tied to jumper pins that can be connected to the BAUD RATE CLK line. The available frequencies are listed in Table 4-3 (located in Section 4.2). Recall that the effective baud rate of the 8251 USART is also dependent on the state of the 8251's internal frequency divider and the mode of operation (refer to Section 3.5.1). The 8251 is capable of dividing the baud rate clock by 1, 16 or 64.

3.5.4 SERIAL I/O INTERRUPTS

The Serial I/O logic can be configured with different forms of an interrupt request mechanism. By connecting jumper pair 16-17 and disconnecting 15-16, the user can allow the 8251's Receiver Ready (RXRDY) output (pin 14) to generate an interrupt request (INT51/) to the CPU Set. RXRDY goes high whenever the receiver enable bit of the command word has been set and the 8251 contains a character that is ready to be input to the CPU Set. The user can also choose to have the 8251's Transmitter Ready (TXRDY) or the Transmitter Empty (TXE) output activate the INT51/ interrupt request. If jumper pair 19-21 is connected, a high on TXRDY (pin 15) will activate INT51/. If jumper pair 18-19 is connected instead, an active TXE (pin 18) output will generate INT51/. TXE goes high when the 8251 has no characters to transmit. TXRDY is high when the 8251 is ready to accept a character from the CPU Set. Both TXE and TXRDY are enabled by setting the transmit enable bit of the command word. Notice on the schematic that, if jumper pairs 19-20 and 15-16 are connected, Serial I/O interrupts are inhibited.

Upon receiving an interrupt, the program can determine the actual condition which is responsible for the interrupt (RXRDY, TXRDY or TXE) by reading the status of the 8251 device as described in Section 3.5.1. The interrupt request will be removed when the data is transferred to/ from the 8251, as required. Note that the TXE or TXRDY output will be high, and consequently maintain an interrupt request, during all idle periods, since the 8251's transmit buffer will remain empty. To disable the transmitter, and the resultant interrupt request, the program can issue a command instruction to the 8251 with the TXEN bit (bit 0) equal to zero (refer to Section 3.5.1). The transmitter should not be disabled until TXE is high.

3.6 PARALLEL I/O INTERFACE

The Parallel I/O Interface logic on the SBC-80/10 provides forty- eight (48) signal lines for the transfer and control of data to or from peripheral devices. Eight lines have a bidirectional driver and termination network permanently installed. The remaining forty lines are uncommitted. Sockets are provided for the installation of active driver networks or passive termination networks. The optional drivers and terminators are installed in groups of four by insertion into the 14-pin sockets.

All forty-eight signal lines eminate from the I/O ports on two Intel 8255 Programmable Peripheral Interface devices, as shown on sheet 5 of the SBC-80/10 schematic (Appendix A). The two 8255 devices allow for a wide variety of I/O configurations. Before describing the possible configurations, however, we will summarize the general operational characteristics of the 8255 device.

3.6.1 INTEL 8255 OPERATIONAL SUMMARY

The 8255 contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255.

Port A: One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C: One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into

two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with Ports A and B.

The 8080 CPU dictates the operating characteristics of the ports by outputting two different types of control words to the 8255:

(1) mode definition control word (bit 7 = 1)

2) port C bit set/reset control word (bit 7 = 0)

Bit 7 of each control word specifies its format, as shown in Figures 3-12 and 3-13, respectively.

Mode Selection

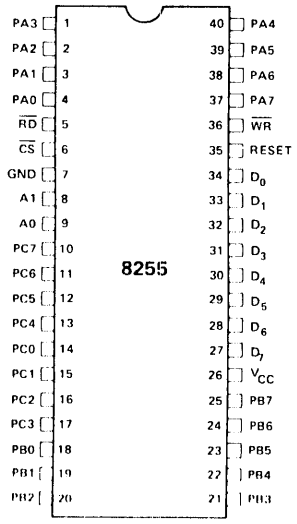
There are three basic modes of operation that can be selected by the system software:

Mode 0 - Basic Input/Output
Mode 1 - Strobed Input/Output
Mode 2 - Bi-Directional Bus

When the RESET input goes "high" all ports will be set to the Input mode 0 (i.e., all 24 lines will be in the high impedance state). After the RESET is removed the 8255 can remain in the Input mode with no additional initialization required. During the execution of the system program, the other modes may be selected using a single OUTput instruction. This allows a single 8255 to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed except for $\overline{OBF}$ in modes 1 and 2.

PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (BI DIRECTIONAL)
RESET	RESET INPUT
$\overline{CS}$	CHIP SELECT
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
V_{CC}	+5 VOLTS
GND	Ø VOLTS

FIGURE 3-12. 8255 PIN ASSIGNMENTS.

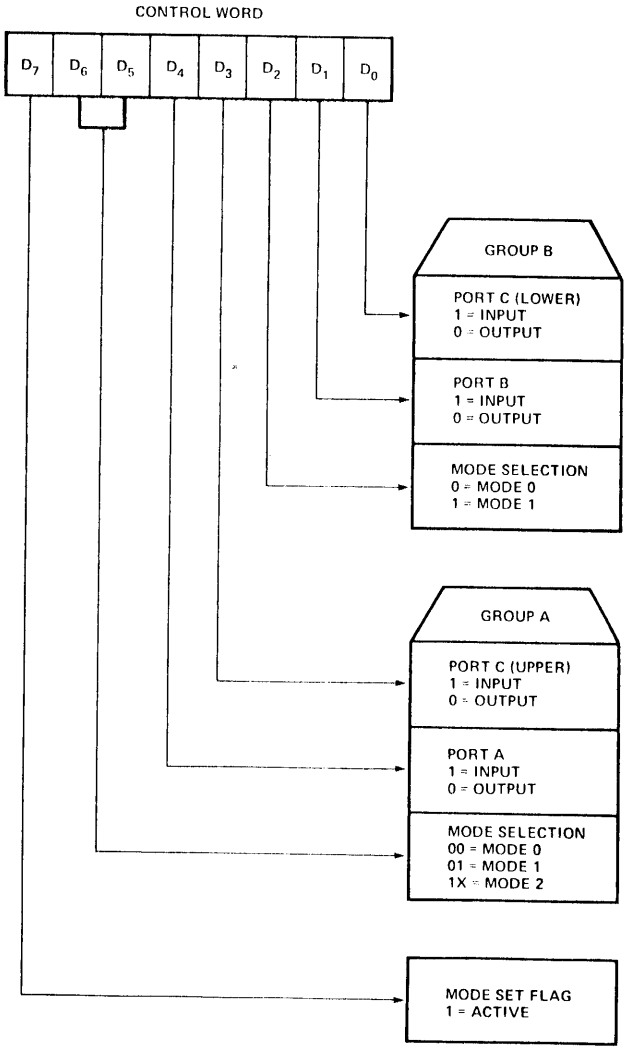


FIGURE 3-13. MODE DEFINITION CONTROL WORD FORMAT.

Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction (see Figure 3-14). This feature reduces software requirements in Control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255 is programmed to operate in Mode 1 or Mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from Port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the Bit set/reset function of Port C.

This function allows the Programmer to disallow or allow specific I/O devices to interrupt the CPU without effecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) - INTE is SET - Interrupt enable
(BIT-RESET) - INTE is RESET - Interrupt disable

Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

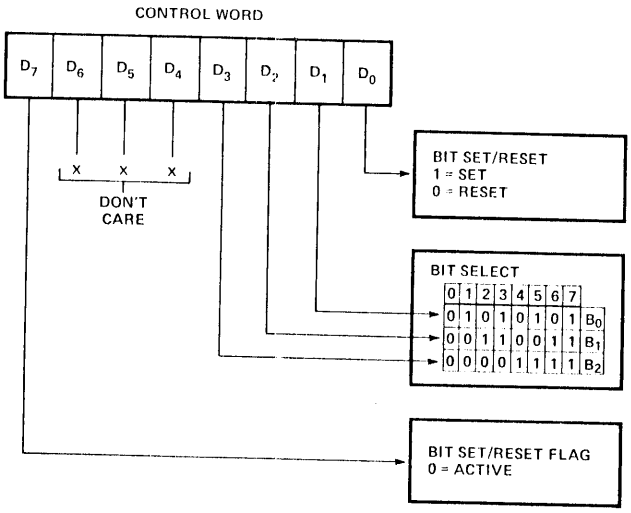


FIGURE 3-14. BIT SET/RESET CONTROL WORD FORMAT.

Operating Modes

Mode 0 (Basic Input/Output):

This functional configuration provides simple Input and Output operations for each of the three ports. No "hand-shaking" is required, data is simply written to or read from a specified port. Mode 0 timing is illustrated in Figure 3-15.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.

· 16 different Input/Output configurations are possible in this Mode. Figure 3-16 shows two possible configurations.

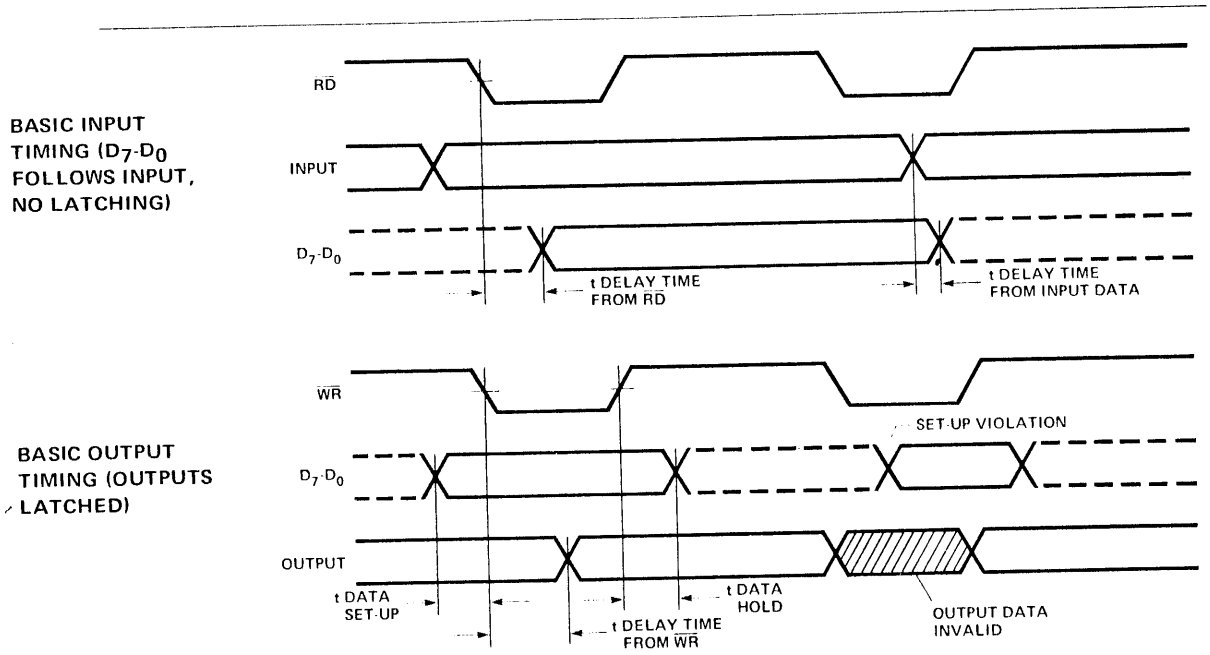


FIGURE 3-15. 8255 MODE 0 TIMING

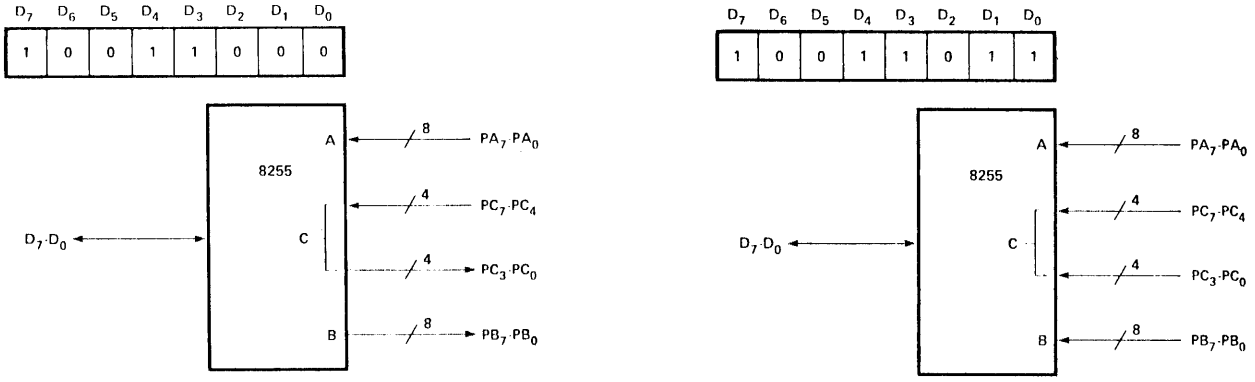


FIGURE 3-16. EXAMPLES OF MODE 0 CONFIGURATION.

Mode 1 (Strobed Input/Output):

This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In Mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Defintions:

- Two transfer ports (A and B).
- Each transfer port contains one 8-bit data port and 4 bits from one half of the control/data port (Port C).
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.

Input Control Signal Defintion for Mode 1

$\overline{STB}$ (Strobe Input)

A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by the falling edge of the STB input and is reset by the rising edge of the $\overline{RD}$ input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the rising edge of $\overline{STB}$ if IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to

request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC4.

INTE B

Controlled by bit/reset of PC2.

Figure 3-17 illustrates the Mode 1 input configuration, while Figure 3-18 shows the basic timing for Mode 1 input.

Output Control Signal Definition for Mode 1

$\overline{OBF}$ (Output Buffer Full F/F)

The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The OBF F/F will be set by the rising edge of the $\overline{WR}$ input and reset by the falling edge of the $\overline{ACK}$ input signal.

$\overline{ACK}$ (Acknowledge Input)

A "low" on this input informs the 8255 that the data from Port A or Port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set by the rising edge of $\overline{ACK}$ if $\overline{OBF}$ is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by bit/reset of PC6.

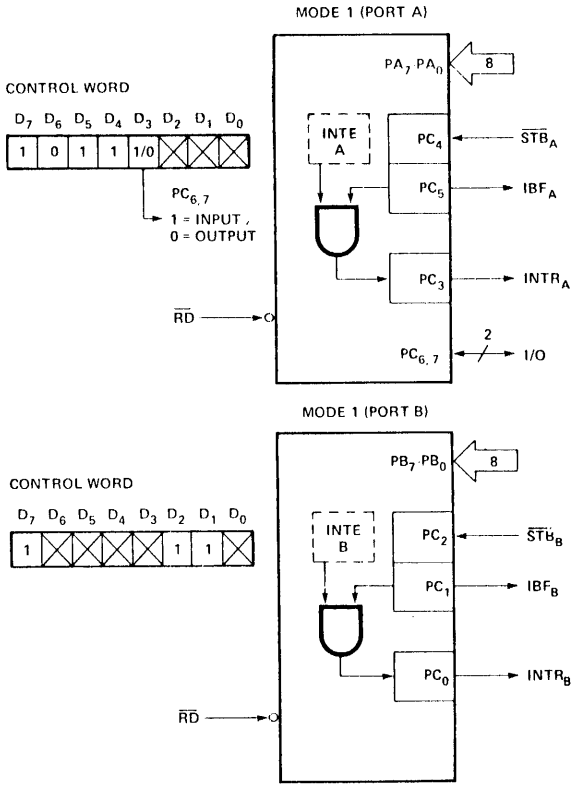


FIGURE 3-17. MODE 1 INPUT CONFIGURATION

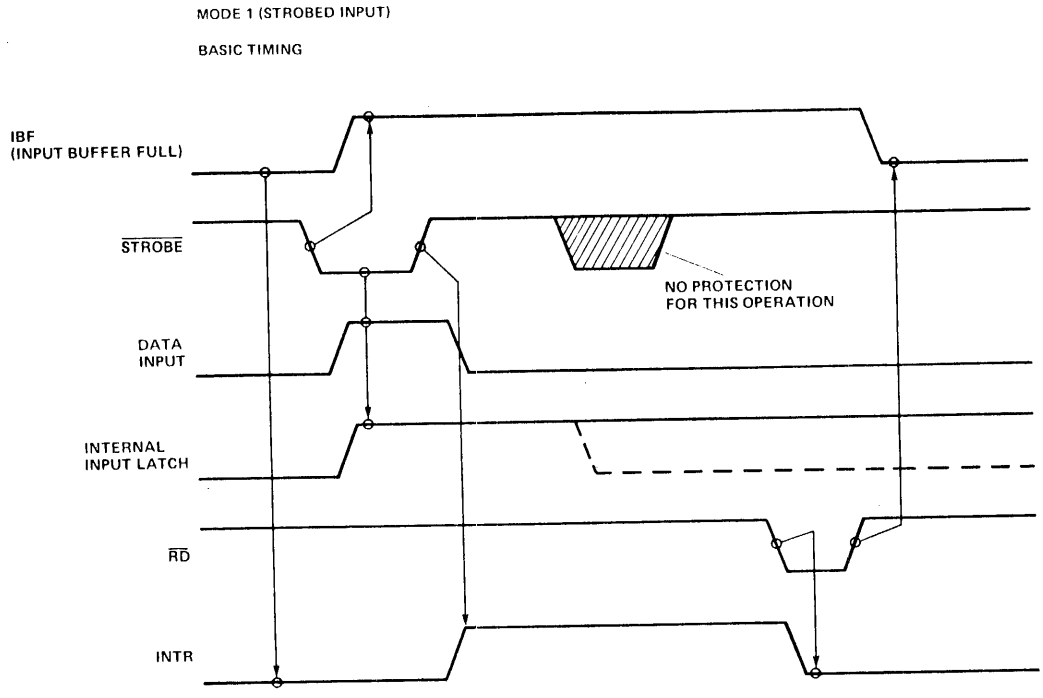


FIGURE 3-18. 8255 MODE 1 INPUT TIMING

INTE B

Controlled by bit set/reset of PC2.

Figure 3-19 illustrates the Mode 1 output configuration , while Figure 3-20 shows basic Mode 1 output timing.

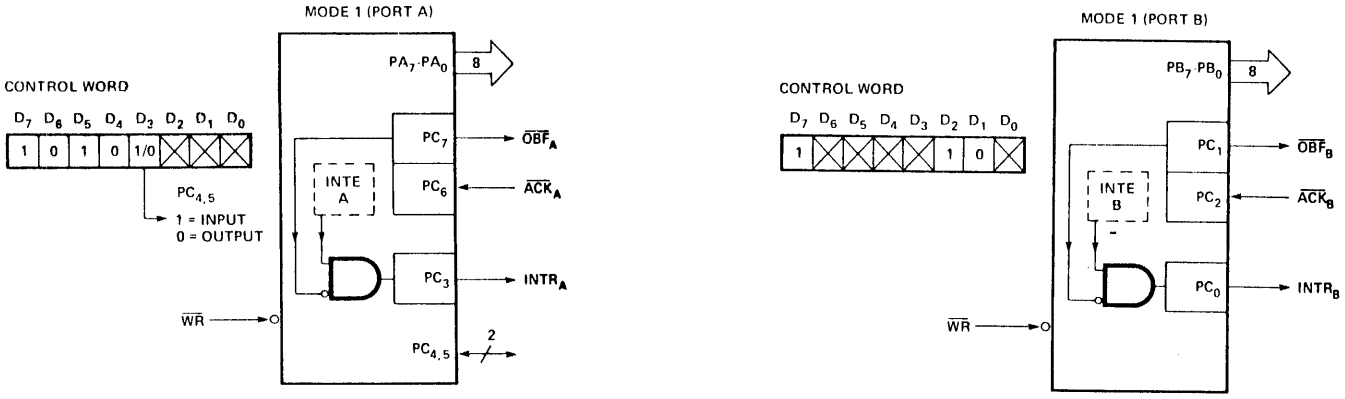


FIGURE 3-19. MODE 1 OUTPUT CONFIGURATION.

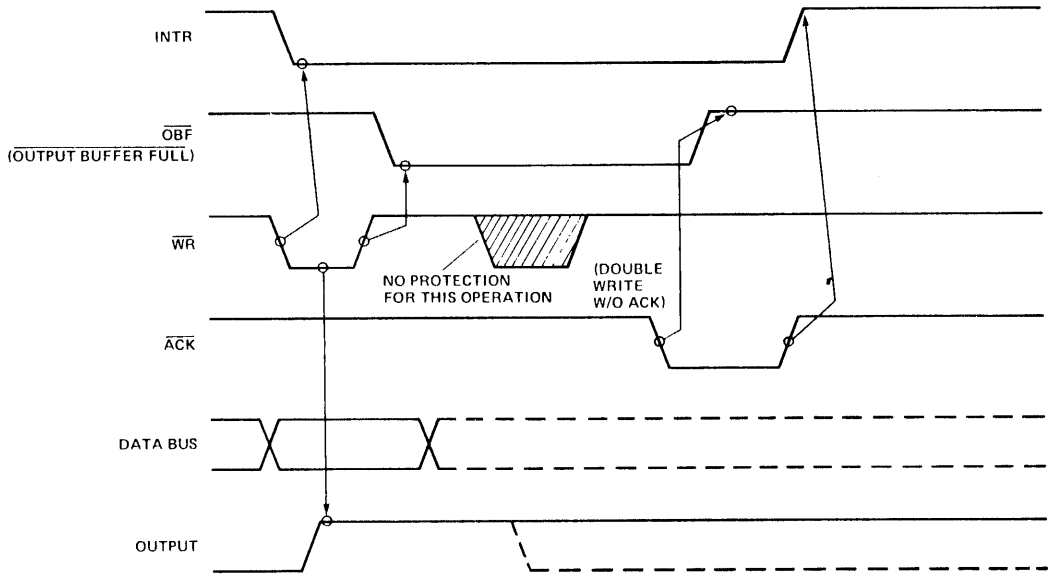


FIGURE 3-20. MODE 1 BASIC OUTPUT TIMING

Mode 2 (Strobed Bi-Directional Bus I/O):

This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bi-directional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to Mode 1. Interrupt generation and enable/disable functions are also available.

Mode 2 Basic Functional Definitions:

- Used in Port A only.
- One 8-bit, bi-directional data Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional data port (Port A).

Bi-Directional Bus I/O Control Signal Definition

INTR (Interrupt Request)

A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operation Control Signals

$\overline{OBF}$ (Output Buffer Full)

The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to Port A.

$\overline{ACK}$ (Acknowledge)

A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high-impedance state.

INTR A and B (The INTE flip-flop associated with $\overline{OBF}$)

Controlled by bit set/reset of PC6 (INTE1)

Input Operation Control Signals

$\overline{STB}$ (Strobed Input)

A "low" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE flip-flop associated with IBF)

Controlled by bit set/reset PC4 (INTE 2)

$INTR_A = PC6 \cdot OBF_A + PC4 \cdot INF_A$

Figure 3-21 illustrates the port configuration for Mode 2, Figure 3-22 shows Mode 2 timing, and Table 3-1 summarizes 8255 Mode definition.

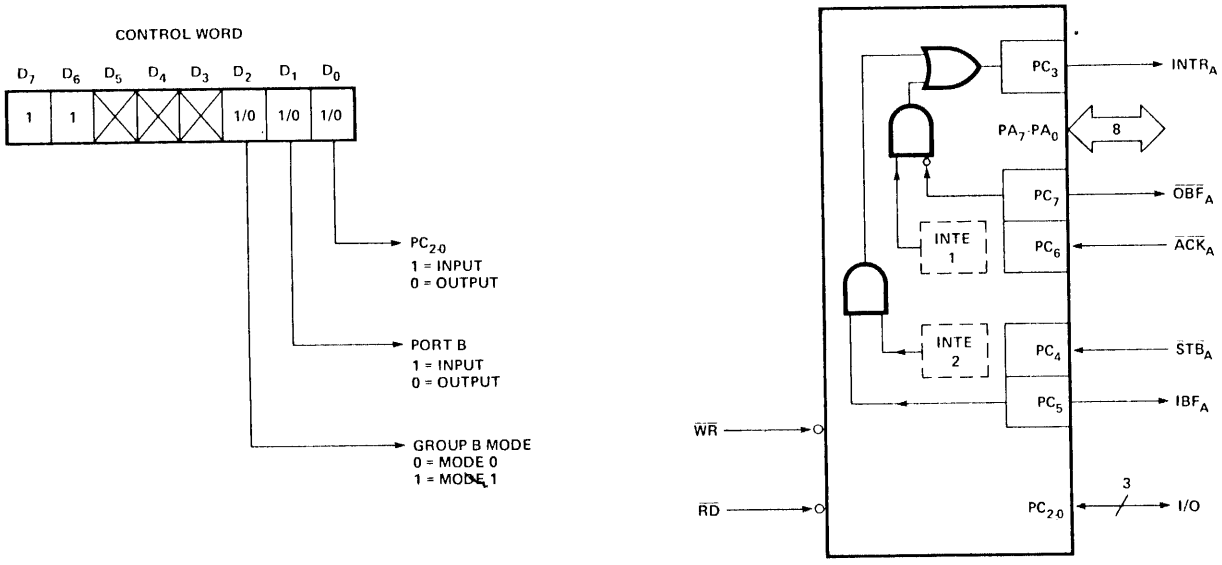


FIGURE 3-21. MODE 2 PORT CONFIGURATION

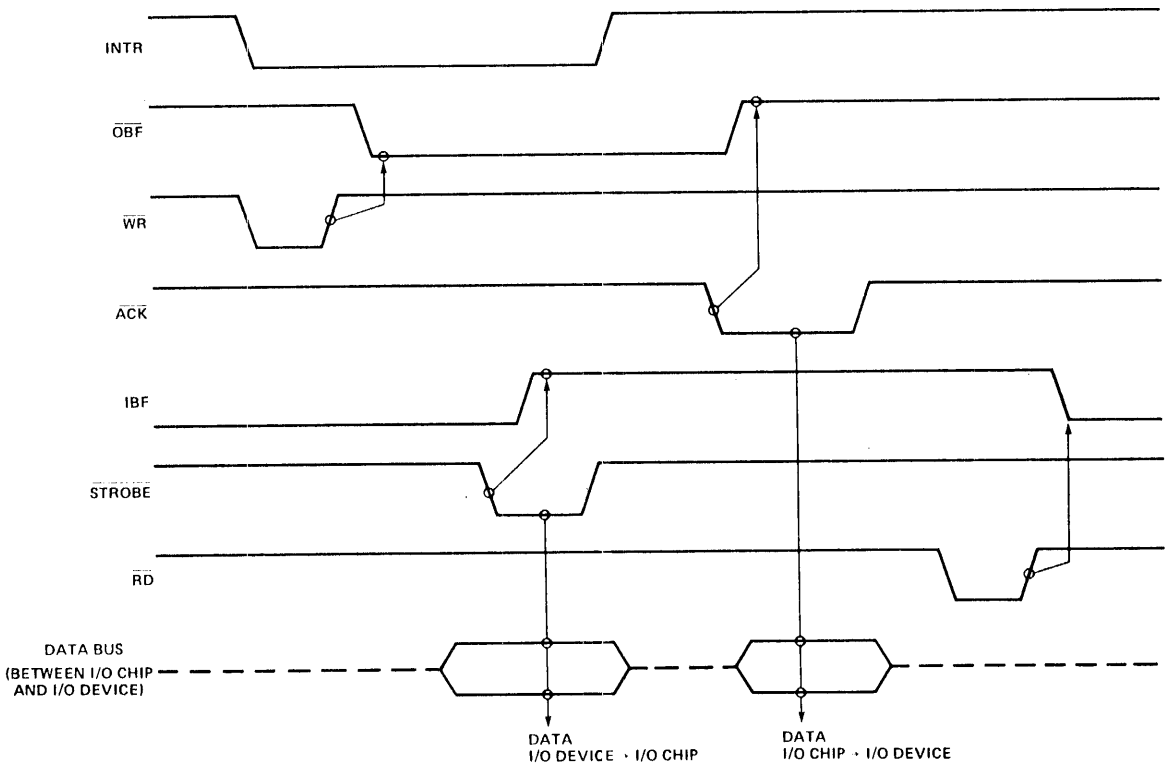


FIGURE 3-22. MODE 2 TIMING

MODE DEFINITION SUMMARY TABLE

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	⟷	
PA_1	IN	OUT	IN	OUT	⟷	
PA_2	IN	OUT	IN	OUT	⟷	
PA_3	IN	OUT	IN	OUT	⟷	
PA_4	IN	OUT	IN	OUT	⟷	
PA_5	IN	OUT	IN	OUT	⟷	
PA_6	IN	OUT	IN	OUT	⟷	
PA_7	IN	OUT	IN	OUT	⟷	
PB_0	IN	OUT	IN	OUT	——	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	——	
PB_2	IN	OUT	IN	OUT	——	
PB_3	IN	OUT	IN	OUT	——	
PB_4	IN	OUT	IN	OUT	——	
PB_5	IN	OUT	IN	OUT	——	
PB_6	IN	OUT	IN	OUT	——	
PB_7	IN	OUT	IN	OUT	——	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

TABLE 3-1. 8255 MODE DEFINITION SUMMARY

3.6.2 PARALLEL I/O CONFIGRATIONS

Referring to sheet 5 of the schematic, we see that there are two 8255 devices, one located at A19, the other at A20. For convenience the following device designations will be used: The device at A19 is called the "group 1" device, while the device at A20 is referred to as the "group 2" device. Each device has three eight-bit ports. The "group 1" ports are designated Ports 1, 2 and 3 while the "group 2" ports are designated Ports 4, 5 and 6.

The group 1 and group 2 devices both communicate with the CPU Set using the same signal lines: the 8-bit data bus, DB0-DB7, and seven control/address lines; ADR0, ADR1, RESET, IOR/, IOW/, CS1/, and CS2/. The data lines bring control bytes or data bytes to an 8255 or deliver data from an 8255 to the CPU Set. The chip select control signals (CS1/ and CS2/) select the group 1 and group 2 devices, respectively, when the proper I/O address appears on the system address bus. CS1/ and CS2/ are the result of decoding address bits 2 through 7 (ADR2-ADR7), as shown on sheet 4 of the schematic (at A14). The two least significant address bits select the control register (when programming an 8255) or one of the three I/O ports (when reading or writing data). IOR/ (8255 $\rightarrow$ CPU Set) and IOW/ (CPU Set $\rightarrow$ 8255) indicate the direction of data flow, as summarized in Table 3-2. Specific I/O addresses for the six ports and two 8255 control registers on the SBC-80/10 are listed in Table 3-3.

A high on the RESET line clears all internal 8255 registers including the control register; all ports (A, B and C) are set for input.

Though both 8255's maintain the same interface (at different

TABLE 3-2. 8255 BASIC OPERATION

A1	A0	IOR/	IOW/	CS/	Input Operation (Read)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
					Output Operation (Write)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
x	x	x	x	1	Data Bus → High-Impedance
1	1	0	1	0	Illegal

TABLE 3-3. PARALLEL I/O PORT ADDRESSES

Port	8255 Device Location	*Eight-Bit Address (Hexadecimal)
1	8255 #1 Port (A)	E4
2	8255 #1 Port (B)	E5
3	8255 #1 Port (C)	E6
-	8255 #1 Control	E7 For I/O write only.
4	8255 #2 Port (A)	E8
5	8255 #2 Port (B)	E9
6	8255 #2 Port (C)	EA
-	8255 #2 Control	EB For I/O write only.

*Note: If address = 111001xx, CS1/ is activated.
If address = 111010xx, CS2/ is activated.

I/O addresses) with the CPU Set, the interface between the group 1 device and edge connector J1 is significantly different than the interface between the group 2 device and its associated edge connector (J2). This gives the user a great deal of flexibility when configuring the system's external parallel I/O devices. Because of those flexible "external" interfaces, however, not all ports are capable of operating in each 8255 mode, though all ports can be programmed as either input or output. The group 1 ports can fully utilize the 8255's multi-mode and external interrupt capabilities as described in Section 3.6.1. The group 2 ports, however, are limited to a single mode of operation. The allowable port configurations for both groups are summarized below:

Port 1 (Group 1 Port A)

Mode 0 Input
Mode 0 Output (Latched)
Mode 1 Input (Strobed)
Mode 1 Output (Latched)
Mode 2 Bidirectional

Port 2 (Group 1 Port B)

Mode 0 Input
Mode 0 Output (Latched)
Mode 1 Input (Strobed)
Mode 1 Output (Latched)

Port 3 (Group 1 Port C)

Mode 0 8 Bit Input
Mode 0 8 Bit Output (Latched)

Note: Control mode dependent upon Port A and B mode.

Ports 4 and 5 (Group 2 Port A, B)

Mode 0 Input
Mode 0 Output (Latched)

Port 6 (Group 2 Port C)

Mode 0 8 Bit Input
Mode 0 8 Bit Output
Mode 0 4 Bit Input/4 Bit Output (Unlatched/latched)
Mode 0 4 Bit Output/4 Bit Input (Unlatched/latched)

Group 1

Port 1 is the most versatile of the six ports. It can be programmed to function in any one of the three 8255 operating modes. This first port is the only port that already includes a permanent bidirectional driver/termination network (two 8226 bus driver devices at A1 and A2).

Before Port 1 is programmed for input or output in any one of three operating modes (as described in Section 3.6.1), certain jumper connections must be made to allow the port to function properly in the chosen mode. The 40-41-42-43 jumper pad specifies the direction of data flow for the two 8226 bidirectional bus drivers. If input in mode 0 or mode 1 is to be programmed for Port 1, jumper pair 41-42 should be connected. If output in mode 0 or mode 1 is to be used, jumper pair 40-41 should be connected. If Port 1 is to be programmed for bidirectional mode 2, then jumper pair 41-43 should be connected. This connection allows the output acknowledge, ACK/, that is input on bit 6 of Port 3 to dynamically dictate direction for the two 8226 devices.

Another jumper pad (48-49-50-51) enables interrupts for Port 1 when it is in mode 1 or mode 2. Jumper pair 49-50 should be connected to allow the INTR output (see Section 3.6.1) from bit 3 of Port 3 to activate an interrupt request (INT55/) from the 74LS02 gate at A45. In mode 0, during which there is no provision for interrupts, jumper pairs 48-49 and 50-51 must be connected to allow use of bit 3 of port 3 and to inhibit Port 1 interrupts.

Because the 8226 bus drivers are inverting devices, all data input to or output from Port 1 is considered to be negative true with respect to the levels at the J1 edge connector.

Port 2 can be programmed for input or output in either mode 0 or mode 1 (see Section 3.6.1). If Port 2 is to be used for input (in either mode), terminator networks must be installed in the sockets at A5 and A6. Because these networks must be passive, data that is input to Port 2 will be positive true. If Port 2 is to be used for output (in either mode), driver networks must be installed in the sockets at A5 and A6. Assuming that the drivers are inverting devices, then the data being output will be negative true at the J1 edge connector.

When Port 2 is programmed for mode 1, interrupts can be enabled by connecting jumper pair 45-46. This connection allows the INTR output from bit 0 of Port 3 to activate the interrupt request (INT55/) to the CPU set. When Port 2 is in mode 0, jumper pairs 44-45 and 46-47 must be connected to allow use of bit 0 of Port 3 and to inhibit Port 2 interrupts.

As was described in Section 3.6.1, the use of Port 3 is dependent on the modes programmed for Ports 1 and 2. If Port 1 is in mode 1 or mode 2, bits 3, 4, 5, 6 and 7 of Port 3 can have dedicated control functions.

```
Port 3 bit 3 → INTR (interrupt request) - input or output
Port 3 bit 4 ← STB/ (input strobe)                    } mode 1 input
Port 3 bit 5 → IBF  (input buffer full flag)          } or mode 2
Port 3 bit 6 ← ACK/ (output acknowledge)              } mode 1 output
Port 3 bit 7 → OBF/ (output buffer full flag)         } or mode 2
```

If Port 2 is in mode 1, bits 0, 1 and 2 of Port 3 have dedicated control functions:

Port 3 bit 0 →	INTR (interrupt request)	- input or output
Port 3 bit 1 →	IBF (input buffer full)	} input only
Port 3 bit 2 ←	STB/ (input strobe	} input only
Port 3 bit 1 →	OBF/ (output buffer full)	} output only
Port 3 bit 2 ←	ACK/ (output acknowledge)	} output only

While certain Port 3 bits are available if Port 1 is in mode 1 or if Port 2 is in mode 0, the use of Port 3 as an eight-bit data path is restricted to those configurations that have both Port 1 and Port 2 programmed for mode 0. In this case all 8 bits of Port 3 can be programmed for mode 0 input (termination networks must be installed in the sockets at A3 and A4) or output (driver networks must be installed at A3 and A4). Note: If Port 1 and 2 are not both in mode 0, then a driver network must be installed in the sockets at A3 and a termination network must be installed at A4, so that the Port 3 control lines can function properly.

Group 2

The three ports on the group 2 device can be programmed for input or output, but only in mode 0. If Port 4 is programmed for input, termination networks must be installed in the sockets at A7 and A8. The data being input will be in positive true form. If Port 4 is programmed for output, driver networks must be installed at A7 and A8. Assuming that inverting drivers are used, then the data will be considered negative true at the J2 edge connector.

If Port 5 is programmed for input, termination networks must be installed in the sockets at A21 and A11. If Port 5 is programmed for output, driver networks must be installed at A21 and A11.

All eight bits of Port 6 can be programmed for input or output, or four bits can be programmed for input while the other four bits are programmed for output (see Section 3.6.1). Driver termination networks must be installed in the sockets at A9 and A10 as listed in Table 3-4.

TABLE 3-4. Port 6 I/O CONFIGURATIONS

	Sockets at A9	Sockets at A10
8-bit Input	Terminators*	Terminators*
8-bit Output	Drivers**	Drivers**
Upper 4-bits Input/ Lower 4-bits Output	Terminators*	Drivers**
Lower 4-bits Input/ Upper 4-bits Output	Drivers**	Terminators*

*Positive-true data.
**Negative-true data if inverting drivers.

In Section 4.2, all of the user options for configuring parallel I/O on the SBC-80/10 are summarized for convenient reference.

CHAPTER 4

USER SELECTABLE OPTIONS

The SBC-80/10 provides the user with a powerful, but flexible I/O capability for both parallel and serial transfers. The serial I/O Interface, using Intel's 8251 USART, provides a serial data communications channel that can be programmed to operate with most of the current serial data transmission protocols. Synchronous or asynchronous mode, baud rate, character length, number of stop bits and even/odd parity are all program selectable. In addition, the user has the option, through jumper connections, of configuring the Serial I/O Interface as an ETA RS232C interface or as a Teletype-compatible current loop interface.

The Parallel I/O Interface, using two Intel 8255 Programmable Peripheral Interface devices, provides 48 signal lines for the transfer and control of data to or from peripheral devices. Eight lines already have a bidirectional driver and termination network permanently installed. The remaining 40 lines, however, are uncommitted. Sockets are provided for the installation of active driver networks or passive termination networks as required to meet the specific needs of the user system.

In this chapter, we will reiterate each of the options available to the user, and summarize, for easy reference, the specific information required to implement the user's tailored I/O configuration. Section 4.1 deals with the Serial I/O Interface, while Section 4.2 covers Parallel I/O options. Section 4.3 will describe general options not covered in the other two sections.

4.1 SERIAL I/O INTERFACE OPTIONS

There are three general areas of Serial I/O options:

1) choice of interface type, RS232C or current loop,

2) baud rate and program-selectable mode options,

3) choice of an interrupt request mechanism.

The first two are covered in the following paragraphs; the third, choice of interrupt mechanism, is quite simple and is fully explained in Section 3.5.4.

4.1.1 INTERFACE TYPE

The user has the choice of configuring the Serial I/O logic to present either an EIA RS232C or a 20 mA current loop interface to an external device. If a Teletype-compatible current loop interface is used, the 8251 I/O pins should be connected to the external Teletype lines as listed in Table 4-1. The reader control logic is controlled by the output DSR/ from the 8251. If an EIA RS232C interface is used, the 8251 can assume the role of a "data set" (see Table 4-2a) or a "data processing terminal" (see Table 4-2b). Pin definitions for the 8251 USART are listed in Section 3.5.1.

4.1.2 BAUD RATE AND PROGRAM-SELECTABLE SERIAL I/O OPTIONS

Before beginning Serial I/O operations, the 8251 must be program-initialized to support the desired mode of operation. The CPU initializes the 8251 by outputting a set of control bytes to the USART device. These control words specify:

* synchronous or asynchronous operation,
* baud rate factor,
* character length,
* number of stop bits,
* even/odd parity.
* parity/no parity

TABLE 4-1. 20 mA CURRENT LOOP SERIAL I/O INTERFACE

8251 PIN MNEMONIC	PIN NO.		CONNECTOR PIN NO.	JUMPER CONNECTIONS
TXD	19	TTY Tx	J3-25	1-2
DTR/	24	TTY RD CONTROL	J3-6	23-24
(1) RTS/	23	(CTS/)	-	27-29, 30-31
(1) CTS/	17	(RTS/)	-	27-29
(2) TXC	9	(Baud Rate Clk)	-	33-34 (8-4, 56-57)
(2) RXC	25	(Baud Rate Clk)	-	35-36 (8-4, 56-57)
TXD	3	TTY Rx	J3-22	38-39
-	-	TTY Rx RET	J3-23	-
-	-	TTY Tx RET	J3-24	-
-	-	TTY RD CTL RET	J3-16	-

Notes: (1) The 8251's RTS/ output is connected to the CTS/ input through jumper pair 27-28. The command instruction word for the 8251 must enable RTS/.

(2) TXC and RXC are connected to the Baud Rate Clk line via jumpers 33-34 and 35-36. The Baud Rate Clk should be configured for 110 baud by connecting jumpers 8-4 and 56-57 (see Table 4-3), and the 8251 should be programmed for a baud rate factor of 64 (see Section 4.2).

TABLE 4-2a. RS232C INTERFACE, "DATA SET" ROLE

8251 PIN MNEMONIC	PIN NO.	LINE FUNCTION	CONNECTOR PIN No.	JUMPER CONNECTIONS
RXD	3	TRANSMITTED DATA	J3-3	37-38
TXD	19	RECEIVED DATA	J3-5	2-3
(1) CTS/	17	REQ TO SEND	J3-7	27-28
RTS/	23	CLEAR TO SEND	J3-7	29-30
DTR/	24	DATA SET READY	J3-11	22-23
(2) DSR/	22	DATA TERMINAL RDY	J3-14	25-26
-	-	PROTECTIVE GROUND	J3-1	-
-	-	SIGNAL GROUND	J3-13	-

TABLE 4-2b. RS232C INTERFACE, "DATA PROCESSING TERMINAL" ROLE

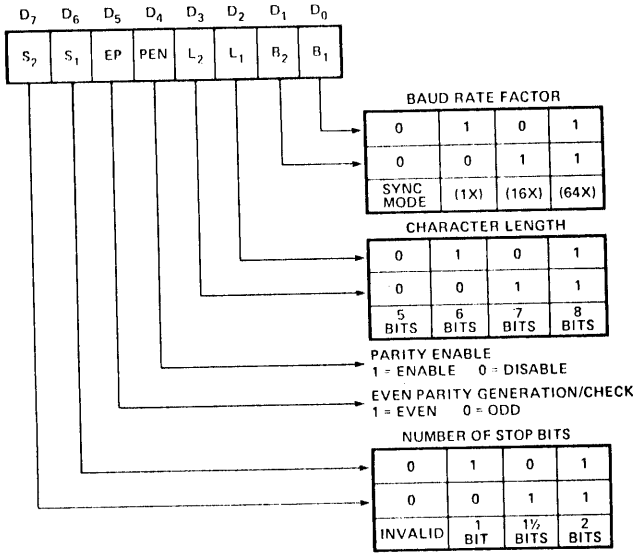
8251 PIN MNEMONIC	PIN NO.	LINE FUNCTION	CONNECTOR PIN NO.	JUMPER CONNECTIONS
TXD	19	TRANSMITTED DATA	J3-5	2-3
RXD	3	RECEIVED DATA	J3-3	37-38
RTS/	23	REQ TO SEND	J3-9	29-30
(1) CTS/	17	CLEAR TO SEND	J3-7	27-28
DTR/	24	DATA TERMINAL RDY	J3-11	22-23
(3) TXC	9	TRANSMIT CLOCK	J3-14	32-33
(2) DSP/	22	DATA SET RDY	J3-14	25-26
(3) RXC	25	RECEIVE CLOCK	J3-22	36-39
-	-	PROTECTIVE GROUND	J3-1	-
-	-	SIGNAL GROUND	J3-13	-

Notes:
(1) The CTS/ input pin on the 8251 must be "low" to enable the 8251 to transmit.
(2) When connector pin J3-14 is jumpered (25-26) to the DSR/ input, J3-14 cannot be used to supply on external transmit clock.
(3) In the asynchronous mode, TXC and RXC can be connected to externally supplied clocks via jumpers 32-33 and 36-39, or they can be connected to the internal Baud Rate Clk via jumpers 33-34 and 35-36, regardless of the mode.

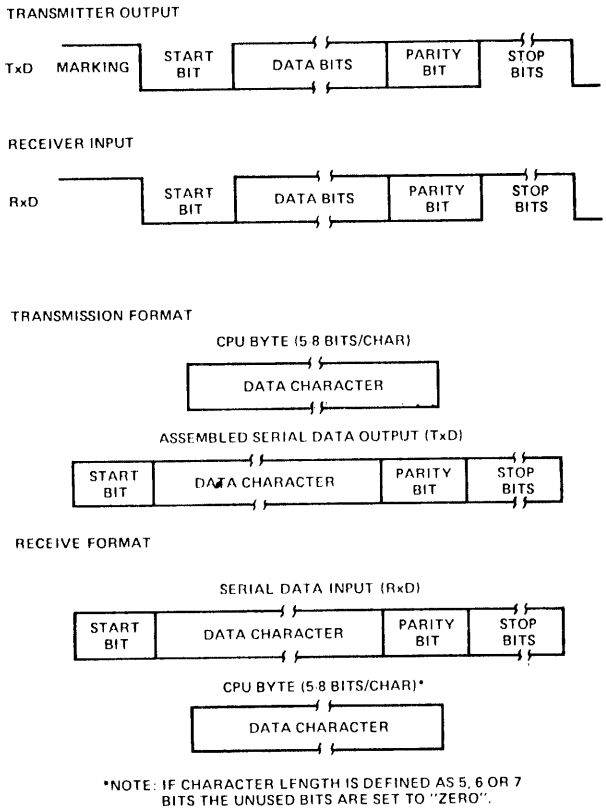
As explained in Section 3.5.1, there are two types of control words: (1) Mode instruction and (2) Command instruction. The Mode instruction initializes the 8251 USART. Because the USART supports either synchronous or asynchronous operation, the Mode instruction has one format for synchronous operation and another for asynchronous. The two least significant bits of the Mode instruction byte specify the format. If DO and D1 both equal 0, synchronous operation is indicated; otherwise, it is asynchronous. The Mode instruction format for asynchronous operation is illustrated in Figure 4-1. The Mode instruction for synchronous operation is shown in Figure 4-2.

Notice in Figure 4-1 that the baud rate factor is specified by the two least significant bits of the instruction byte (labeled B1 and B2). During asynchronous communications, the Baud Rate Clock frequency supplied to the 8251's TXC and RXC input pins is divided by the baud rate factor to produce the effective baud rate (i.e., the frequency at which data bits are serially transmitted by the 8251 USART). Consequently, the Baud Rate Clock, as well as the program-selected baud rate factor, must be considered in implementing the desired effective baud rate. The Baud Rate Clock frequency is selected through various jumper connections as shown on sheet 4 of the SBC-80/10 schematic (Appendix A). The selection of an effective baud rate is summarized in Table 4-3.

Notice from the schematic that TXC and RXC inputs can be supplied by externally supplied clocks (via connector pins J3-14 and J3-22, respectively), instead of using the Baud Rate Clock, if jumpers 32-33 and 36-39 are connected and jumpers 33-34 and 35-36 are disconnected.

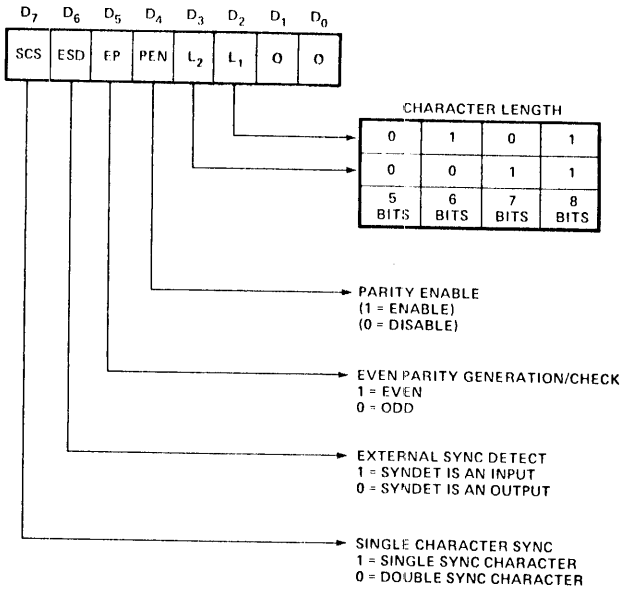


Mode Instruction Format, Asynchronous Mode

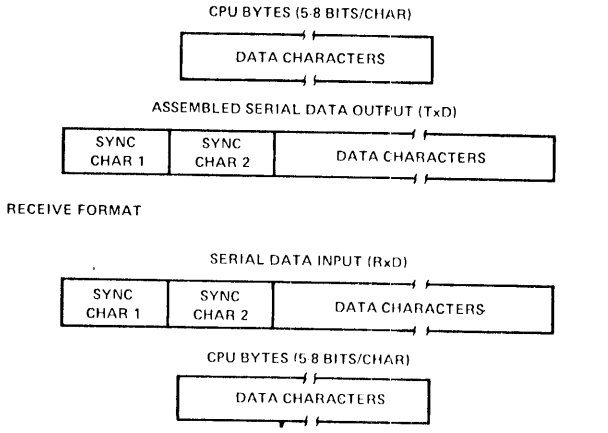


Asynchronous Mode

FIGURE 4-1. ASYNCHRONOUS OPERATION



Mode Instruction Format, Synchronous Mode



Synchronous Mode, Transmission Format

FIGURE 4-2. SYNCHRONOUS OPERATION

TABLE 4-3. BAUD RATE SELECTION

JUMPER CONNECTION	EFFECTIVE BAUD RATE (Hz)		
	SYNCHRONOUS MODE	ASYNCHRONOUS MODE	
		BAUD RATE FACTOR=16 (2)	BAUD RATE FACTOR=64 (2)
10-4	-	19,200 (3)	4800
11-4	-	9600	2400
12-4	-	4800	1200
5-4	38,400	2400	600
6-4	19,200	1200	300
7-4	9600	600	150
(1) 8-4	4800	300	75
(1) 8-4, 56-57	6980	-	110 (TTY)

Note: (1) If jumper pair 56-57 is not connected, the frequency at jumper pole 8 is 4.8 KHZ. If jumper 56-57 is connected, however, the frequency at jumper pole 8 is 6.98 KHZ which, with a programmed baud rate factor of 64, provides an effective baud rate of approximately 110 baud for Teletype use.

(2) Baud rate factor is software selectable.

(3) Caution: Baud Rate Factor = 16

4.2 PARALLEL I/O OPTIONS

The Parallel I/O Interface consists of six 8-bit I/O ports implemented with two Intel 8255 Programmable Peripheral Interface devices. The primary user considerations in determining how to use each of the six I/O ports are:

1) Choice of operating mode (as defined in Section 3.6.1),

2) direction of data flow (input, output or bidirectional),

3) choice of driver/termination networks for port's data path.

In the following paragraphs, we will define the capabilities of each port and summarize, in tables, that information which is necessary to use the port in each of its potential configurations. Each table will list the port I/O address, the control register address and the format for the control word which is output to the 8255 by the CPU Set and which specifies the particular configuration to be used. Each

table will also summarize all of the relevant information concerning the choice and use of driver/termination networks, the data polarity, the connecting of jumpers and what they enable, and any restrictions on the use of the other two ports in each group. Examples of suitable driver/termination networks are listed in Section 5.1.

4.2.1 PORT 1 (GROUP 1 PORT A)

Port 1 is the only port that already includes a permanent bi-directional driver/termination network (two 8226 Bidirectional Bus Drivers). Port 1 is also the only port which can be programmed to function in any one of the three 8255 operating modes, which were defined in Section 3.6.1. Before Port 1 is programmed for input or output in any one of the three modes, certain jumper connections must be made to allow the port to function properly in the chosen mode. Other jumper connections must be made to enable interrupts when Port 1 is in mode 1 or mode 2. In all, there are five potential configurations for Port 1. All of the necessary information for implementing each configuration has been summarized in the following tables:

PORT 1 CONFIGURATIONS		TABLE
Mode	Direction	
1. Mode 0	Input	Table 4-4
2. Mode 0	Output (Latched)	Table 4-5
3. Mode 1	Input (Strobed)	Table 4-6
4. Mode 1	Output (Latched)	Table 4-7
5. Mode 2	Bidirectional	Table 4-8

TABLE 4-4. PORT 1, MODE 0 INPUT CONFIGURATION

PORT 1 ADDRESS: E4, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	1	x	x	x	x

DRIVER/TERMINATION NETWORKS: Two Intel 8226 Bidirectional Bus Drivers permanently installed at A1 and A2.

DATA POLARITY: Negative-true.

JUMPER CONNECTIONS: 41-42 to enable input at 8226's.

PORT 2 RESTRICTIONS: None; port 2 can be programmed for mode 0 or mode 1, input or output (see Section 4.2.2)

PORT 3 RESTRICTIONS: None; port 3 can be programmed for mode 0, 8-bit input or output, unless port 2 is in mode 1. (see Section 4.2.3).

TABLE 4-5. PORT 1, MODE 0 LATCHED OUTPUT CONFIGURATION

PORT 1 ADDRESS: E4, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	0	x	x	x	x

DRIVER/TERMINATION NETWORKS: Two Intel 8226 Bidirectional Bus Drivers permanently installed at A1 and A2.

DATA POLARITY: Negative-true.

JUMPER CONNECTIONS: 40-41 to enable output at 8226's.

PORT 2 RESTRICTIONS: None; port 2 can be programmed for mode 0 or mode 1, input or output (see Section 4.2.2)

PORT 3 RESTRICTIONS: None; port 3 can be programmed for mode 0, input or output, unless port 2 is in mode 1 (see Section 4.2.3).

TABLE 4-6. PORT 1, MODE 1 STROBED INPUT CONFIGURATION

PORT 1 ADDRESS: E4, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	1	1	x	x	x	x

DRIVER/TERMINATION NETWORKS: Two Intel 8226 Bidirectional Bus Drivers permanently installed at A1 and A2. A driver network must be installed at A3 and a termination network must be installed at A4.

DATA POLARITY: Negative-true. The polarity of Port 3 control outputs is dependent on the type of driver installed at A3.

JUMPER CONNECTIONS: 41-42 to enable input at 8226's; connect 49-50 to enable interrupt request via INT55/.

PORT 2 RESTRICTIONS: None; port 2 can be programmed for mode 0 or mode 1, input or output (see Section 4.2.2).

PORT 3 RESTRICTIONS: Port 3 bits perform the following dedicated functions:

*Bits 0, 1 and 2 - dedicated to control of port 2 if port 2 is in mode 1 (see Tables 4-9 to 4-12).
*Bit 3 - INTR (interrupt request) output for port 1.
*Bit 4 - STB/ (strobe) input for port 1.
*Bit 5 - IBF (input buffer full) output for port 1.
*Bit 6 - can be used for input only. Bit 3 of control word = 1
*Bit 7 - cannot be used.

TABLE 4-7. PORT 1, MODE 1 LATCHED OUTPUT CONFIGURATION

PORT 1 ADDRESS: E4, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	1	0	x	x	x	x

DRIVER/TERMINATION NETWORKS: Two Intel 8226 Bidirectional Bus Drivers permanently installed at A1 and A2. A driver network must be installed at A3 and a termination network must be installed at A4.

DATA POLARITY: Negative-true. The polarity of Port C control outputs is dependent on the type of driver installed at A3.

JUMPER CONNECTIONS: 41-42 to enable output at 8226's; connect 49-50 to enable interrupt request via INT55/.

PORT 2 RESTRICTIONS: None; port 2 can be programmed for mode 0 or mode 1, input or output (see Section 4.2.2).

PORT 3 RESTRICTIONS: Port 3 bits perform the following dedicated functions:

*Bits 0, 1 and 2 - dedicated to the control of port 2 if port 2 is in mode 1 (see Tables 4-11 and 4-12).
*Bit 3 - INTR (interrupt request) output for port 1.
*Bit 4 - can be used for input if bit 3 of control word = 1
*Bit 5 - cannot be used if PC4 is used; can be used for output if control word bit 3 = 0 (PC4 cannot be used then).
*Bit 6 - ACK/ (acknowledge) input for port 1.
*Bit 7 - OBF/ (output buffer full) output for port 1.

TABLE 4-8. PORT 1, MODE 2 BIDIRECTIONAL CONFIGURATION

PORT 1 ADDRESS: E4, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	1	x	x	x	x	x	x

DRIVER/TERMINATION NETWORKS: Two Intel 8226 Bidirectional Bus Drivers permanently installed at A1 and A2. A driver network must be installed at A3 and a termination network must be installed at A4.

DATA POLARITY: Negative-true. The polarity of Port C control outputs is dependent on the type of driver installed at A3.

JUMPER CONNECTIONS: 41-43 to allow ACK/ input on PC6 to dynamically change data direction at 8226's (input when ACK/ = 1 and output when ACK/ = 0); connect 49-50 to enable interrupt request via INT55/.

PORT 2 RESTRICTIONS: None.

PORT 3 RESTRICTIONS: Port 3 bits perform the following dedicated functions:

*Bits 0 and 1 - can be used for output if bit 3 of control word = 0
*Bit 2 - cannot be used if PC0 and PC1 are used; can be used for input if control word bit 3 = 1 (PC0 and PC1 cannot be used then).
*Bit 3 - INTR (interrupt request) output for port 1.
*Bit 4 - STB/ (strobe input for port 1.
*Bit 5 - IBF (input buffer full) output for port 1.
*Bit 6 - ACK/ (acknowledge) input for port 1.
*Bit 7 - OBF/ (output buffer full) output for port 1.

4.2.2 PORT 2 (GROUP 1 Port B)

Port 2 can be programmed for input or output in either mode 0 or mode 1. If Port 1 is in mode 2, however, Port 2 must be programmed for mode 0. If Port 2 is to be used for input, in either mode, terminator networks must be installed in the sockets at A5 and A6. If Port 2 is to be used for output, in either mode, driver networks must be installed in the sockets at A5 and A6. When Port 2 is programmed for mode 1, interrupts can be enabled by connecting jumper pair 45-46. The four potential configurations for Port 2 are summarized in the following tables:

PORT 2 CONFIGURATIONS		TABLE
Mode	Direction	
1. Mode 0	Input	Table 4-9
2. Mode 0	Output (Latched)	Table 4-10
3. Mode 1	Input (Strobed)	Table 4-11
4. Mode 1	Output (Latched)	Table 4-12

TABLE 4-9. PORT 2, MODE 0 INPUT CONFIGURATION

PORT 2 ADDRESS: E5, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	x	x	x	x	0	1	x

DRIVER/TERMINATION NETWORKS: Termination networks must be installed at A5 and A6.

DATA POLARITY: Positive-true.

JUMPER CONNECTION: None.

PORT 1 RESTRICTIONS: None (see Section 4.2.1).

PORT 3 RESTRICTIONS: None, port 3 can be programmed for mode 0, input or output, unless port 1 is in mode 1 or mode 2 (see Section 4.2.3).

TABLE 4-10. PORT 2, MODE 0 LATCHED OUTPUT CONFIGURATION

PORT 2 ADDRESS: E5, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	x	x	x	x	0	0	x

DRIVER/TERMINATION NETWORKS: Driver networks must be installed at A5 and A6.

DATA POLARITY: Negative-true, assuming that inverting drivers are at A5 and A6.

JUMPER CONNECTIONS: None.

PORT 1 RESTRICTIONS: None (see Section 4.2.1).

PORT 3 RESTRICTIONS: None, port 3 can be programmed for mode 0 or mode 1, 8-bit input or output, unless port 1 is in mode 1 or mode 2 (see Section 4.2.3).

TABLE 4-11. PORT 2, MODE 1 STROBED INPUT CONFIGURATION

PORT 2 ADDRESS: E5, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	x	x	x	1	1	x

DRIVER/TERMINATION NETWORKS: Termination networks must be installed at A5 and A6. A driver network must be installed at A3 and a termination network must be installed at A4.

DATA POLARITY: Positive-true. The polarity of Port C control outputs is dependent on the type of driver installed at A3.

JUMPER CONNECTIONS: 45-46 to enable interrupt request via INT55/.

PORT 1 RESTRICTIONS: None.

PORT 3 RESTRICTIONS: Port 3 bits perform the following dedicated functions:

*Bit 0 - INTR (interrupt request) output for port 2.
*Bit 1 - IBF (input buffer full) output for port 2.
*Bit 2 - STB/ (strobe) input for port 2.
*Bit 3 to Bit 7 - dedicated to control of port 1 if port 1 is in mode 1 (see Tables 4-4 to 4-7).

TABLE 4-12. PORT 2, MODE 1 LATCHED OUTPUT CONFIGURATION

PORT 2 ADDRESS: E5, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	x	x	x	1	0	x

DRIVER/TERMINATION NETWORKS: Driver networks must be installed at A5 and A6. A driver network must be installed at A3 and a termination network must be installed at A4.

DATA POLARITY: Negative-true, assuming that inverting drivers are at A5 and A6. The polarity of Port C control outputs is dependent on the type of driver installed at A3.

JUMPER CONNECTIONS: 45-46 to enable interrupt request via INT55/

PORT 1 RESTRICTIONS: None.

PORT 3 RESTRICTIONS: Port 3 bits perform the following dedicated functions:

*Bit 0 - INTR (interrupt request) output for port 2.
*Bit 1 - OBF/ (output buffer full) output for port 2.
*Bit 2 - ACK/ (acknowledge) input for port 2.
*Bit 3 - P3-7 - dedicated to control of port 1 if port 1 is in mode 1 (see Tables 4-4 to 4-7).

4.2.3 PORT 3 (GROUP 1 PORT C)

The use of Port 3 is dependent on the modes programmed for Ports 1 and 2 (refer to Tables 4-4 to 4-12). While certain Port 3 bits are available if Port 1 is in mode 1 or if Port 2 is in mode 0, the use of Port 3 as an 8-bit data path is restricted to those configurations that have both Port 1 and Port 2 programmed for mode 0. In this case, all eight bits of Port 3 can be programmed for mode 0 input (see Table 4-13) or output (see Table 4-14). A 4-bit input/4-bit output configuration is never possible for group 1 Port 3.

Note: If Ports 1 and 2 are not both in mode 0, then a driver network must be installed in the sockets at A3 and a termination network must be installed at A4, so that the Port 3 control lines can function properly.

4.2.4 PORTS 4 AND 5 (GROUP 2 PORTS A AND B)

Ports 4 and 5 can be programmed for input or output but only in mode 0. The two potential configurations for each port are summarized in the following tables:

CONFIGURATIONS			TABLE
PORT	MODE	DIRECTION	
1. Port 4	Mode 0	Input	Table 4-15
2. Port 4	Mode 0	Output (Latched)	Table 4-16
1. Port 5	Mode 0	Input	Table 4-17
2. Port 5	Mode 0	Output (Latched)	Table 4-18

TABLE 4-13. PORT 3, MODE 0, 8-BIT INPUT CONFIGURATION

PORT 3 ADDRESS: E6, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	x	1	0	x	1

DRIVER/TERMINATION NETWORKS: Termination networks must be installed at A3 and A4.

DATA POLARITY: Positive-true.

JUMPER CONNECTIONS: 46-47 and 44-45 to disable port 2 interrupts and enable P3-0; connect 48-49 and 50-51 to disable port 1 interrupts and enable P3-3.

PORT 1 AND 2 RESTRICTIONS: Both ports 1 and 2 must be in mode 0.

TABLE 4-14. PORT 3, MODE 0, 8-BIT LATCHED OUTPUT CONFIGURATION

PORT 3 ADDRESS: E6, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	x	0	0	x	0

DRIVER/TERMINATION NETWORKS: Driver networks must be installed at A3 and A4.

DATA POLARITY: Negative-true, assuming that inverting drivers are installed at A3 and A4.

JUMPER CONNECTIONS: 46-47, and 44-45 to disable port 2 interrupts and enable P3-0; connect 48-49 and 50-51 to disable port 1 interrupts and enable P3-3.

PORT 1 AND 2 RESTRICTIONS: Both ports 1 and 2 must be in mode 0.

TABLE 4-15. PORT 4, MODE 0, INPUT CONFIGURATION

PORT 4 ADDRESS: E8, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	1	x	0	x	x

DRIVER/TERMINATION NETWORKS: Termination networks must be installed at A7 and A8.

DATA POLARITY: Positive-true.

JUMPER CONNECTIONS: None.

PORT 2 AND 3 RESTRICTIONS: None; ports 5 and 6 can be programmed for mode 0, input or output (also see Section 4.2.5).

TABLE 4-16. PORT 4, MODE 0 LATCHED OUTPUT CONFIGURATION

PORT 4 ADDRESS: E8, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	0	x	0	x	x

DRIVER/TERMINATION NETWORKS: Driver networks must be installed at A7 and A8.

DATA POLARITY: Negative-true, assuming that inverting drivers are installed at A7 and A8.

JUMPER CONNECTIONS: None.

PORT 5 AND 6 RESTRICTIONS: None; ports 5 and 6 can be programmed for mode 0, input or output (also see Section 4.2.5).

TABLE 4-17. PORT 5, MODE 0 INPUT CONFIGURATION

PORT 5 ADDRESS: E9, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	x	0	1	x	x

DRIVER/TERMINATION NETWORKS: Termination networks must be installed at A11 and A21.

DATA POLARITY: Positive-true.

JUMPER CONNECTIONS: None.

PORT 4 AND 6 RESTRICTIONS: None; ports 4 and 6 can be programmed for mode 0, input or output (also see Section 4.2.5).

TABLE 4-18. PORT 5, MODE 0 LATCHED OUTPUT CONFIGURATION

PORT 5 ADDRESS: E9, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	x	0	0	x	x

DRIVER/TERMINATION NETWORKS: Driver networks must be installed at A11 and A21.

DATA POLARITY: Negative-true, assuming that inverting drivers are installed at A11 and A21.

JUMPER CONNECTIONS: None.

PORT 4 AND 6 RESTRICTIONS: None; ports 4 and 6 can be programmed for mode 0, input or output (also Section 4.2.5).

4.2.5 PORT 6 (GROUP 2 PORT C)

All eight bits of Port 6 can be programmed for mode 0 input or output, or four bits can be programmed for mode 0 input while the other four bits are programmed for mode 0 output. The four potential configurations for Port 6 are summarized in the following tables:

PORT 6 CONFIGURATIONS	TABLE
1. MODE 0 8-BIT INPUT	Table 4-19
2. MODE 0 8-BIT OUTPUT (LATCHED)	Table 4-20
3. MODE 0 UPPER 4-BIT INPUT/LOWER 4-BIT OUTPUT	Table 4-21
4. MODE 0 UPPER 4-BIT OUTPUT/LOWER 4-BIT INPUT	Table 4-22

TABLE 4-19. PORT 6, MODE 0, 8-BIT INPUT CONFIGURATION

PORT 6 ADDRESS: EA, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	x	1	0	x	1

DRIVER/TERMINATION NETWORKS: Termination networks must be installed at A9 and A10.

DATA POLARITY: Positive-true.

JUMPER CONNECTIONS: None.

PORT 4 AND 5 RESTRICTIONS: None (see Section 4.2.4).

TABLE 4-20. PORT 6, MODE 0, 8-BIT LATCHED OUTPUT CONFIGURATION

PORT 6 ADDRESS: EA, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	x	0	0	x	0

DRIVER/TERMINATION NETWORKS: Driver networks must be installed at A9 and A10.

DATA POLARITY: Negative-true, assuming that inverting drivers are installed at A9 and A10.

JUMPER CONNECTIONS: None.

PORT 4 AND 5 RESTRICTIONS: None (see Section 4.2.4).

TABLE 4-21. PORT 6, MODE 0 UPPER 4-BIT INPUT/LOWER 4-BIT LATCHED OUTPUT CONFIGURATION

PORT 6 ADDRESS: EA, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	x	1	0	x	0

DRIVER/TERMINATION NETWORKS: A termination network must be installed at A9 and a driver network must be installed at A10.

DATA POLARITY: The upper 4-bits will be in positive-true form; however, the lower four bits will be in negative-true form if an inverting driver is installed at A10.

JUMPER CONNECTIONS: None.

PORT 4 AND 5 RESTRICTIONS: None (see Section 4.2.4).

TABLE 4-22. PORT 6, MODE 0 UPPER 4-BIT LATCHED OUTPUT/LOWER 4-BIT INPUT CONFIGURATION

PORT 6 ADDRESS: EA, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT:

7	6	5	4	3	2	1	0
1	0	0	x	0	0	x	1

DRIVER/TERMINATION NETWORKS: A driver network must be installed at A9 and a termination network must be installed at A10.

DATA POLARITY: The lower 4-bits will be in positive-true form; however, the upper 4-bits will be in negative-true form if an inverting driver is installed at A9.

JUMPER CONNECTIONS: None.

PORT 4 AND 5 RESTRICTIONS: None (see Section 4.2.4).

TABLE 4-23. PARALLEL I/O ADDRESS AND SOCKET ASSIGNMENTS

PORT	I/O ADDRESS	SOCKET NUMBERS
1	E4	BI-DIRECTIONAL DRIVER/ TERMINATOR AT A1, A2
2	E5	A5, A6
3	E6	A3, A4*
4	E8	A7, A8
5	E9	A11, A21
6	EA	A9, A10**

*Note requirements specified in Tables 4-4 through 4-14.

**Note requirements specified in Tables 4-15 through 4-22.

4.3 GENERAL OPTIONS

There are several other options that may be useful. Details are provided in the following paragraphs.

4.3.1 SYSTEM RESET OUTPUT

The user can enable a SYSTEM RESET output from the SBC 80/10 by connecting jumper pair 54-55. This allows the reset signal which is generated on the SBC 80/10 during power-up sequences (see Section 3.1.5) to be made available to other modules in the system via connector P1-14. Notice on the schematic that a SYSTEM RESET input is accepted by the SBC 80/10 and P1-14 and applied to the 8080 regardless of jumper connections.

4.3.2 DISABLE BUS CLOCK SIGNALS

The bus clock BCLK/ (connector pin P1-13) or the constant clock CCLK/ (P-31) outputs can be disabled (if more drive, or a different frequency is needed) by disconnecting jumper pair 61-63 or 62-64, respectively. When connected, both BCLK/ and CCLK/ provide a 9.216 MHz timing reference to other modules.

4.3.3 ACKNOWLEDGE INPUTS

The SBC bus has defined two types of acknowledges; transfer acknowledge (XACK/) and advance acknowledge (AACK/). XACK/ is the required responses of a memory or I/O port which indicates that the specified read/write operation has been completed. That is, data has been placed on (READ command) or accepted from (WRITE command) the system data bus lines. XACK/ is asynchronous with BCLK/. AACK/ is an advance acknow-

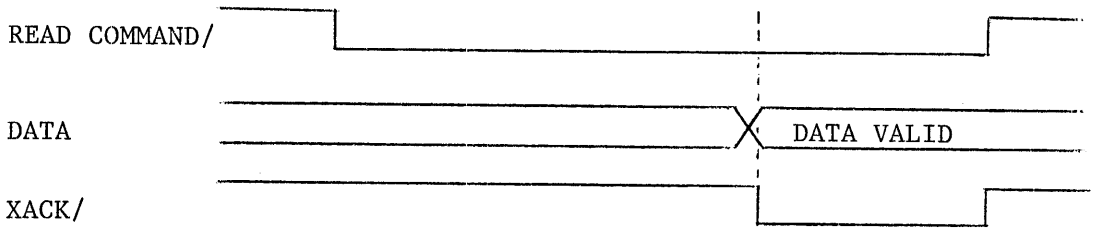


FIGURE 4-3. READ COMMAND WITH XACK/

ledge in response to a memory or I/O port command. This acknowledge is used only with 8080 CPU-based systems. Figure 4-4 shows timing of the SBC 80/10 "READING" memory using the AACK/ signal.

AACK/ is a response to a READ command that data will be valid on the bus by the time the 8080 needs it. Thus, if the access time of the slave device is less than t_{ACC} (command to data needed by the 8080), the slave module has t_{8KD} (command to 8080 sample point of bus acknowledge) to indicate that the data on the bus will be valid when the 8080 needs it. If the access time of a module is less than t_{8KD} then XACK/ can be used and the 8080 will run at maximum speed.

If the access time of a module is greater than t_{8KD}, but less than t_{ACC}, AACK/ must be used, if the 8080 is to run at maximum speed. If AACK/ is not used and instead XACK/ is used, the 8080 will execute one more wait state than is necessary. AACK/ is asynchronous with BCLK/.

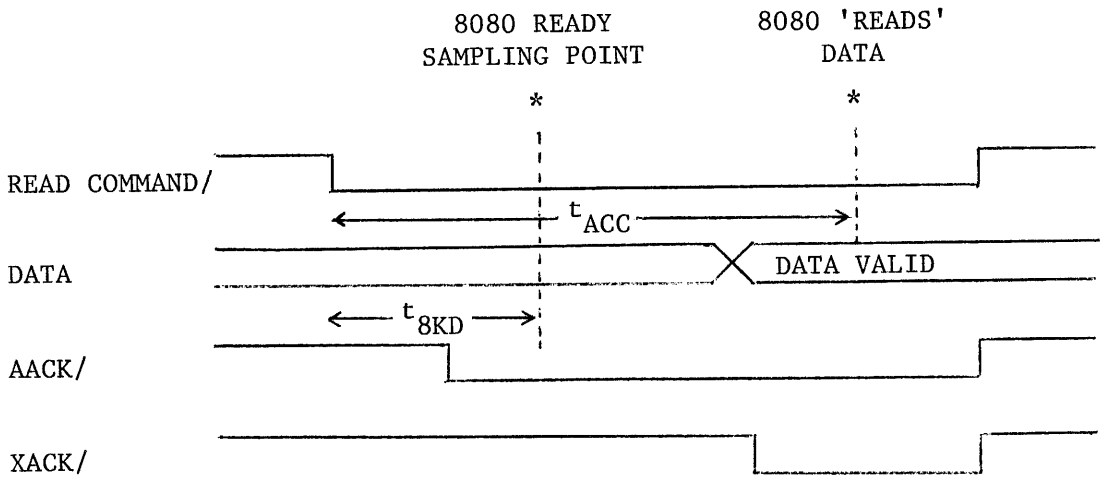


FIGURE 4-4. READ COMMAND WITH AACK/

AACK/ is also an advance response to a WRITE command indicating that the slave module will have accepted the data from the system bus by the time the 8080 has completed the WRITE. Figure 4-5 shows timing of the SBC 80/10 "WRITING" memory using the AACK/ signal.

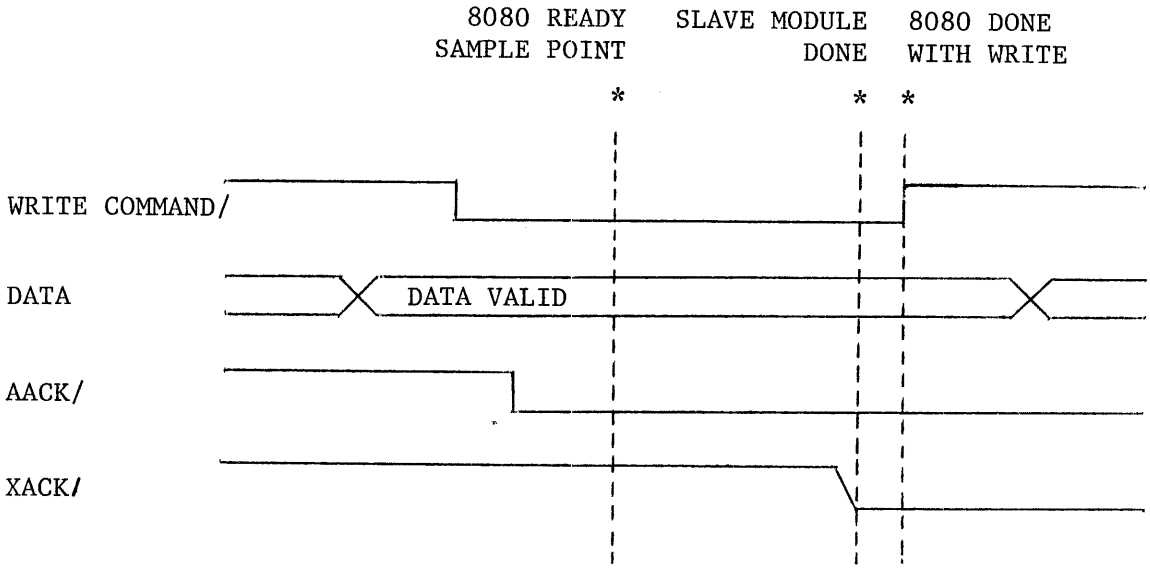


FIGURE 4-5. WRITE COMMAND WITH AACK/

When modules that generate proper AACK/ are used with the SBC 80/10, jumper pair 52-53 should be connected to allow AACK/ to be accepted (at P1-25) and gated to the RDYIN pin on the 8224 clock generator. If this option is used, caution should be taken to insure that all the modules on the bus meet the SBC 80/10 timing requirements

4.3.4 INTERRUPT SOURCES

There are six sources of interrupts on the SBC 80/10 board, two from the serial I/O section (see 3.5.4), two from the parallel I/O section (see 3.6), and two from external sources. One of the external sources is INTR1/. INTR1/ is connected to the SBC bus (P1-42) and is the only interrupt line that other SBC modules can use to interrupt the SBC 80/10. The other external interrupt is EXT INTRØ/. This interrupt is connected to the parallel I/O connector J1 (pin 49) and can be used by an external device to interrupt the SBC 80/10. Both external interrupts are negative true logic, a TTL low (V_{IN} < 0.4 volts) will cause the 8080 to interrupt and execute a RST 7 instruction. The processor can then read in the status registers of the possible interrupting devices to determine which device generated the interrupt. Then the processor can jump to the correct interrupt service routine, service that device, enable interrupts, and return.

4.4 DEFAULT OPTIONS

Table 4-24 lists the default options jumpered on the SBC 80/10. These options permit the SBC 80/10 to communicate to a TTY; they also provide power-up reset, bus clock, and the communication clock to the system bus. If the SBC 80/10 is driving the bus clock (BCLK/) and/or the communications clock (CCLK/), the system bus must be limited to 7 inches. This limitation is due to the SBC 80/10's limited drive capability on these clock lines. The system bus can be extended beyond 7 inches if the user provides BCLK/ and CCLK/.

TABLE 4-24. DEFAULT OPTION ON THE SBC 80/10

DEFAULT JUMPERS	REFERENCE	DESCRIPTION
1 - 2	4.1.1	Connect 8251 T_xD to 20 mA Current Loop Driver
23 - 24		Connect 8251 DTR/ to TTY Reader Control Circuit
39 - 38		Connect 8251 R_xD to 20 mA Current Loop Receiver
4 - 8	4.1.2	Generates 6.98K Baud Rate Clock
57 - 56		Generates 6.98K Baud Rate Clock
34 - 33		Connect 8251 T_x Clock to Baud Rate Clock
35 - 36		Connect 8251 R_x Clock to Baud Rate Clock
27 - 29		Connect 8251 RTS/ to 8251 CTS/
19 - 20	3.5.4	Disable T_xRDY Interrupt from 8251
16 - 15	3.5.4	Disable R_xRDY Interrupt from 8251
26 - 25	4.1.2	Connect DTR/ Receiver to 8251 DSR/ Input
30 - 31	4.1.2	Connect Set Clear to Send Driver to +12V
40 - 41	4.2.1	Enable Port 1 Bi-directional Drivers to Input
54 - 55	4.3.1	Connect Power-Up Reset to System Bus
62 - 64	4.3.2	Connect 9.216 MHz Clock to Communication Clock Line
61 - 63	4.3.2	Connect 9.216 MHz Clock to Bus Clock Line

CHAPTER 5

SYSTEM INTERFACING

The SBC-80/10, with its memory and I/O ports, is a complete computer on a single printed circuit board. However, the SBC-80/10 can also serve as a primary master module within an expanded system, communicating with numerous memory and I/O modules. In this chapter we identify each of the SBC-80/10's external connections and define all signals on the external system bus.

5.1 ELECTRICAL CONNECTIONS

The SBC-80/10 comes on a 12.00 × 6.75 inch printed circuit board, 0.50 inch thick and weighing 12 oz. The SBC-80/10 requires DC power at the following levels:

		(1)	(2)
V_{CC} = +5V ± 5%		I_{CC} = 4.0A max.	I_{CC} = 2.9A
V_{DD} = +12V ± 5%		I_{DD} = 0.40A max.	I_{DD} = 140mA
V_{BB} = -5V ± 5%		I_{BB} = 0.20A max	I_{BB} = 2mA
V_{AA} = -12V ± 5%		I_{AA} = 0.175 max.	I_{AA} = 175mA

Notes:
(1) The values assume that four 8708 PROM's are present and that ten optional 220/330 Ω termination networks being driven low have been installed in the Parallel I/O Interface.
(2) These values assume that the 8708 PROM's and optional termination networks are not present.

The SBC-80/10 has five edge connectors, as shown in Figure 5-1. Edge connectors at the top of the module are designed for compatibility with both flat cable and round cable hardware. All parallel I/O functions are paired with an independent signal ground pin. This allows flat cable implementation to utilize an alternate signal/ground scheme

for reduction of cross talk. Round cables may easily be implemented as twisted pair with an individual ground pin for every return wire. The serial connection hardware has similar flexibility but ground return lines are not as extensive. The connector is wired for RS232C compatibility, thus, only one signal ground is provided.

The Parallel I/O Interface communicates with external I/O devices via two 50-pin double-sided PC edge connectors (J1 and J2), 0.1 inch centers. External devices can be attached to J1 or J2 using any of the following mating connectors:

J1 and J2 Mating Connectors

Connector Type	Vendor	Part No.
Flat Cable	3M AMP	3415-0001 2-86792-3
Soldered	AMP VIKING TI	2-583715-3 3VH25/1JV-5 H312125
Wire-wrap	TI VIKING CDC ITT	H 311125 3VH25/1JND-5 VPB01B25D00A1 EC4A050A1A
Crimp	AMP	1-583717-1

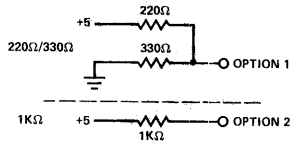
Tables 5-1 and 5-2 provide pin lists for the J1 and J2 connectors, respectively. The following TTL line drivers and Intel terminators are all compatible with the I/O driver sockets in the Parallel I/O Interface:

Driver	Characteristic	Sink Current (ma)
7438	I, OC	48
7437	I	48
7432	NI	16
7426	I, OC	16
7409	NI, OC	16
7408	NI	16
7403	I, OC	16
7400	I	16

Note: I = inverting; N.I. = non-inverting
OC = open collector

I/O Terminators:

Terminators: 220Ω/330Ω divider or 1 kΩ pull up



SBC-901

SBC-902

See Appendix C for schematics

TABLE 5-1. PIN ASSIGNMENTS FOR CONNECTOR J1
(Parallel I/O Interface - Group 1)

PIN	SIGNAL	PIN	SIGNAL
1	PORT 2 - BIT 3	2	GND
3	PORT 2 - BIT 2	4	
5	PORT 2 - BIT 1	6	
7	PORT 2 - BIT 0	8	
9	PORT 2 - BIT 4	10	
11	PORT 2 - BIT 5	12	
13	PORT 2 - BIT 6	14	
15	PORT 2 - BIT 7	16	
17	PORT 3 - BIT 3	18	
19	PORT 3 - BIT 2	20	
21	PORT 3 - BIT 4	22	
23	PORT 3 - BIT 6	24	
25	PORT 3 - BIT 0	26	
27	PORT 3 - BIT 5	28	
29	PORT 3 - BIT 1	30	
31	PORT 3 - BIT 7	32	
33	PORT 1 - BIT 7	34	
35	PORT 1 - BIT 6	36	
37	PORT 1 - BIT 5	38	
39	PORT 1 - BIT 4	40	
41	PORT 1 - BIT 1	42	
43	PORT 1 - BIT 0	44	
45	PORT 1 - BIT 2	46	
47	PORT 1 - BIT 3	48	
49	EXT INTR 1/	50	GND

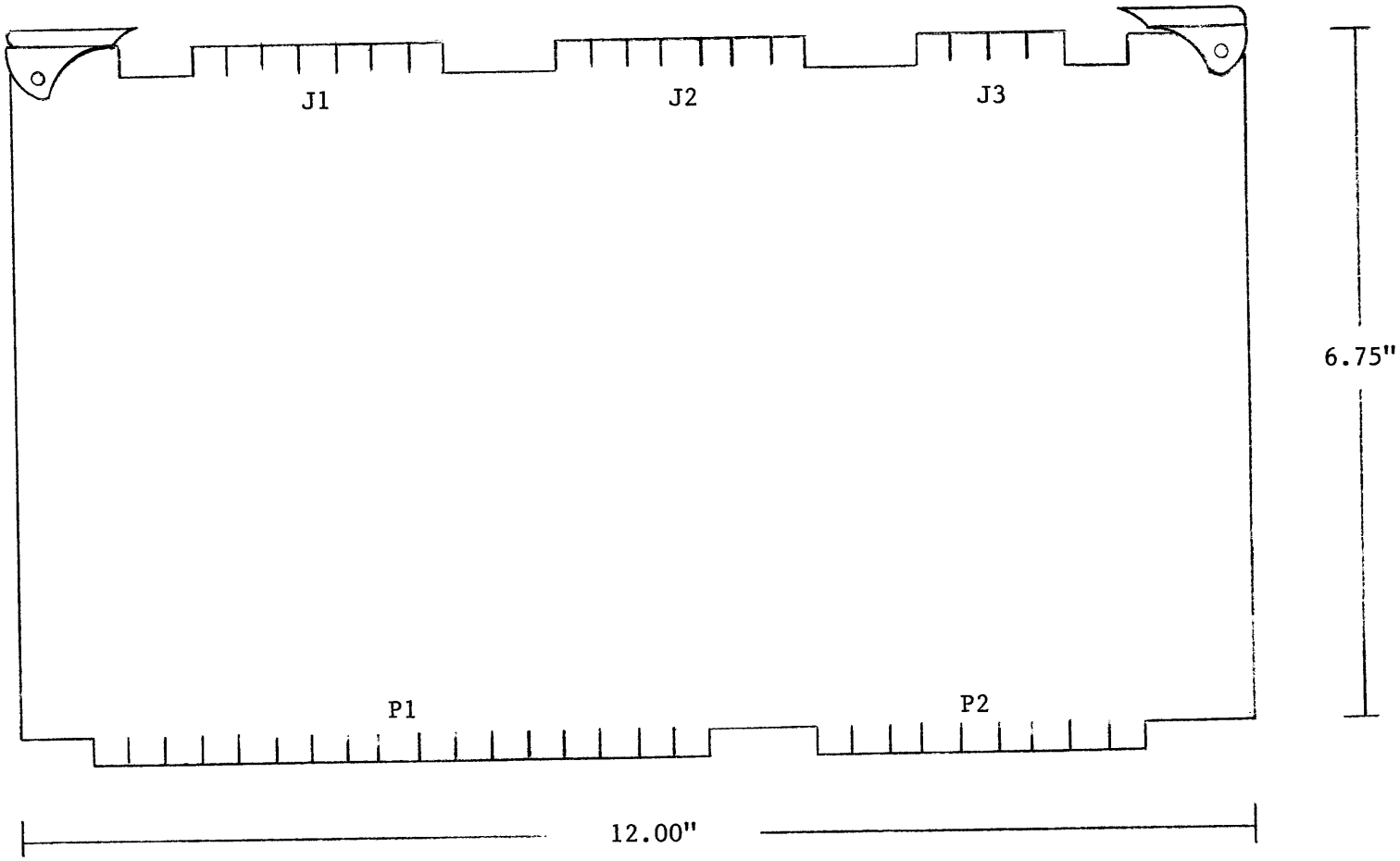


FIGURE 5-1. SBC-80/10 EDGE CONNECTORS

TABLE 5-2. PIN ASSIGNMENTS FOR CONNECTOR J2
(Parallel I/O Interface - Group 2)

PIN	SIGNAL	PIN	SIGNAL
1	GND	2	GND
3	PORT 5 - BIT 3	4	
5	PORT 5 - BIT 0	6	
7	PORT 5 - BIT 1	8	
9	PORT 5 - BIT 2	10	
11	PORT 5 - BIT 4	12	
13	PORT 5 - BIT 5	14	
15	PORT 5 - BIT 6	16	
17	PORT 5 - BIT 7	18	
19	PORT 6 - BIT 3	20	
21	PORT 6 - BIT 2	22	
23	PORT 6 - BIT 1	24	
25	PORT 6 - BIT 0	26	
27	PORT 6 - BIT 4	28	
29	PORT 6 - BIT 5	30	
31	PORT 6 - BIT 6	32	
33	PORT 6 - BIT 7	34	
35	PORT 4 - BIT 7	36	
37	PORT 4 - BIT 6	38	
39	PORT 4 - BIT 5	40	
41	PORT 4 - BIT 4	42	
43	PORT 4 - BIT 0	44	
45	PORT 4 - BIT 1	46	
47	PORT 4 - BIT 2	48	
49	PORT 4 - BIT 3	50	GND

The Serial I/O Interface communicates with an external I/O device via a 26-pin double-sided PC edge connector (J3), 0.1 inch centers. An external device can be connected to J3 using a 3M 3462-0001 flat cable connector or one of the following soldered connectors: TI H312113 or AMP 1-583715-1. Table 5-3 provides a pin list for connector J3.

The SBC-80/10 communicates with other system modules via an 86-pin double-sided edge connector (P1), 0.156 inch centers. This

edge connector will accept any of the following mating connectors: CDC VPB01E43A000A1, Micro Plastics MP-0156-43-BW-4 or ARCO AE 443WP1. Section 5.2 defines each of the external system bus signals and includes a pin list for P1 (Table 5-5).

TABLE 5-3. PIN ASSIGNMENTS FOR CONNECTOR J3
(Serial I/O Interface)

PIN	SIGNAL NAME	PIN	SIGNAL NAME
1	CHASSIS GND	2	
3	TRANSMITTED DATA	4	
5	RECEIVED DATA	6	TTY RD CONTROL
7	REQ TO SEND	8	
9	CLEAR TO SEND	10	
11	DATA SET READY	12	
13	GND	14	Tx CLK/DATA TERMINAL RDY
15	DATA CARRIER RETURN	16	TTY RD CONTROL RETURN
17		18	
19		20	
21		22	RECEIVE CLK/TTY Rx DATA RETURN
23	TTY Rx DATA		
25	TTY Tx DATA	24	TTY Tx DATA RETURN
		26	GND

The 60-pin double-sided edge connector labeled P2 in Figure 5-1 allows access to various test points on the SBC-80/10 (see Table 5-4). The following wire-wrap connectors will attach to P2:

CDC VPB01B30A00A2,
TI H311130 and
AMP PE5-14559

TABLE 5-4. PIN ASSIGNMENTS FOR CONNECTOR P2
(Auxilliary Connector)

SIGNAL NAME	PIN ASSIGNMENT	COMMENT
OSC	P2 - 28	TEST POINT
RAM 3C00 ENABLE/	P2 - 30	
RAM 3D00 ENABLE/	P2 - 32	
RAM 3E00 ENABLE/	P2 - 34	
RAM 3F00 ENABLE/	P2 - 36	
OSC INH/	P2 - 40	
DATA BUS INH/	P3 - 42	
BAUD RATE CLK TTY	P2 - 44	
COUNT 1 ENABLE 1	P2 - 46	
BAUD RATE CLK	P2 - 50	
COUNT 2 ENABLE/	P2 - 52	
TIME OUT ENABLE/	P2 - 54	
B & C CLK SET/	P2 - 55	
STATUS STROBE	P2 - 57	
RDY IN INH/	P2 - 57	
BAUD RATE CLEAR/	P2 - 58	
OSC/2	P2 - 60	TEST POINT

5.2 EXTERNAL SBC 80/10 SYSTEM BUS SUMMARY

A significant measure of the SBC-80/10's power and flexibility can be attributed to its external system bus. In expanded systems, the external bus structure allows for master-slave relationships between the various system modules. The bus includes its own clock (BCLK/) which is derived independently from the processor clock. Actual transfers via the bus, however, proceed asynchronously with respect to the bus clock. Thus, the transfer speed is dependent on the transmitting and receiving devices only. Once a module has gained

control of the bus by activating the BPRN input to the SBC-80/10, single or multiple read/write transfers can proceed at a maximum rate of 5 million data bytes per second. The 16 system address lines allow the SBC-80/10 to support up to 65,536 bytes of storage. The signal lines on the external system bus are defined as follows:

BCLK/ — Bus clock; used to synchronize bus control circuits on all master modules. BCLK/ has a period of 101.725 nanoseconds (9.8304 MHz frequency), 30% - 70% duty cycle. BCLK/ may be slowed, stopped or single stepped, if desired (see Section 4.4).

INIT/ — Initialization signal; resets the entire system to a known internal state.

BPRN — Bus priority input signal; indicates to the SBC-80/10 that a higher priority master module is requesting use of the system bus. BPRN suspends the processing activity and drivers of the SBC-80/10.

BUSY/ — Bus busy signal; indicates that the bus is currently in use. BUSY/ prevents all other master modules from gaining control of the bus. BUSY/ is driven by the HLDA/ output from the SBC-80/10 in response to a BPRN input. It indicates that the bus is available.

MRDC/ — Memory read command; indicates that the address of a memory location has been placed on the system address lines and specifies that the contents of the addressed location are to be read and placed on the system data bus.

MWTC/ — Memory write command; indicates that the address of a memory location has been placed on the system address lines and that a data word has been placed on the system data bus. MWTC/ specifies that the data word is to be written into the addressed memory location.

IORC/ — I/O read command; indicates that the address of an input port has been placed on the system address bus and that the data at that input port is to be read and placed on the system data bus.

IOWC/ — I/O write command; indicates that the address of an output port has been placed on the system address bus and that the contents of the system data bus are to be output to the addressed port.

XACK/ — Transfer acknowledge signal; the required response of an external memory location or I/O port which indicates that the specified read/write operation has been completed. That is, data has been placed on, or accepted from, the system data bus lines.

AACK/ — Advance acknowledge signal; used with 8080 CPU-based systems. 8080/ is an advance acknowledge, in response to a memory read command, that allows the memory to complete the access without requiring the CPU to wait.

TABLE 5-5. PIN ASSIGNMENTS FOR CONNECTOR P1
(External SBC 80/10 System Bus)

	(COMPONENT SIDE)			(CIRCUIT SIDE)		
	PIN	MNEMONIC	DESCRIPTION	PIN	MNEMONIC	DESCRIPTION
POWER SUPPLIES	1	GND	Signal GND	2	GND	Signal GND
	3	VCC	+ 5VDC	4	VCC	+ 5VDC
	5	VCC	+ 5VDC	6	VCC	+ 5VDC
	7	VDD	+12VDC	8	VDD	+12VDC
	9	VBB	- 5VDC	10	VBB	- 5VDC
	11	GND	Signal GND	12	GND	Signal GND
BUS CONTROLS	13	BCLK/	Bus Clock	14	INIT/	Initialize
	15	BPRN	Bus Pri. In	16	[1]	
	17	BUSY/	Bus Busy	18	[1]	
	19	MRDC/	Mem Read Cmd	20	MWTC/	Mem Write Cmd
	21	IORC/	I/O Read Cmd	22	IOWC/	I/O Write Cmd
	23	XACK/	XFER Acknow	24	[1]	
SPARES	25	AACK/	Special	26		
	27			28		
	29			30		
	31	CCLK/	Constant Clock	32		
	33	[1]		34		
INTERRUPTS	35	[1]		36	[1]	
	37	[1]		38	[1]	
	39	[1]		40	[1]	
	41	[1]		42	INTR1/	Interrupt request
ADDRESS	43	ADRD/	Address Bus	44	ADRF/	Address Bus
	45	ADRC/		46	ADRD/	
	47	ADRA/		48	ADRB/	
	49	ADR8/		50	ADR9/	
	51	ADR6/		52	ADR7/	
	53	ADR4/		54	ADR5/	
	55	ADR2/		56	ADR3/	
	57	ADR0/		58	ADR1/	
DATA	59	[1]	Data Bus	60	[1]	Data Bus
	61	[1]		62	[1]	
	63	[1]		64	[1]	
	65	[1]		66	[1]	
	67	DAT6/		68	DAT7/	
	69	DAT4/		70	DAT5/	
	71	DAT2/		72	DAT3/	
	73	DAT0/		74	DAT1/	
POWER SUPPLIES	75	GND	Signal GND	76	GND	Signal GND
	77	VBB[1]	-10VDC	78	VBB[1]	-10VDC
	79	VAA	-12VDC	80	VAA	-12VDC
	81	VCC	+ 5VDC	82	VCC	+ 5VDC
	83	VCC	+ 5VDC	84	VCC	+ 5VDC
	85	GND	Signal GND	86	GND	Signal GND

[1] Used by Intellec® MDS Bus.

CCLK/	Constant clock; provides a clock signal of constant frequency (9.8304 MHz) for use by option memory and I/O expansion boards. CCLK/ coincides with BCLK/ and has a period of 101.725 nanoseconds, 30% - 70% duty cycle (see Section 4.4).
INTR1/	Externally generated interrupt request.
ADR0/-ADRF/	16 Address lines: used to transmit the address of the memory location or I/O port to be accessed. ADRF/ is the most significant bit.
DAT0/-DAT7/	Bi-directional data lines; used to transmit/receive information to/from a memory location or I/O port. DAT7/ is the most significant bit.

5.3 RS232C CABLING

When the Serial I/O Interface is configured as an RS232C interface, the J3 edge connector can be cabled such that a RS232C pin-compatible connector is presented to the user's terminal or modem. A 26-pin mating connector, 3M 3462-0001, should be attached to the J3 edge connector on the SBC-80/10 and to a 25-wire flat cable, 3M 3349/25. The flat cable is, in turn, attached to the RS232C pin-compatible connector, 3M 3483-1000. Table 5-6 equates the J3 edge connector pins with the associated RS232-compatible pins on the 3M 3483-1000 connector.

Note: Using this 3M cable assemble, the RS232C connector pin-outs are MDS compatible. That is, if the SBC 80/10 is set up to drive a TTY, an MDS modified TTY can be used directly with the SBC 80/10.

SIGNAL NAME	J3 CONNECTOR PIN NO.	RS232C CONNECTOR PIN NO.
CHASSIS GND	1	1
	2	14
TRANSMITTED DATA	3	2
	4	15
RECEIVED DATA	5	3
TTY RD CONTROL	6	16
REQ TO SEND	7	4
	8	17
CLEAR TO SEND	9	5
	10	12
DATA SET READY	11	6
	12	19
GND	13	7
Tx CLK/DATA TERMINAL RDY	14	20
DATA CARRIER RETURN	15	8
TTY RD CONTROL RETURN	16	21
	17	9
	18	22
	19	10
	20	23
	21	11
RECEIVE CLK/TTY Rx DATA RETURN	22	24
TTY Rx DATA	23	12
TTY Tx DATA RETURN	24	25
TTY Tx DATA	25	13

5.4 TELETYPE MODIFICATIONS

The ASR-33 Teletype must receive the following internal modifications and external connections, for use with the SBC-80/10 Board.

Internal Modifications

(1) The current source resistor value must be changed to 1450Ω. This is accomplished by moving a single wire (see Figure 5-7).

(2) A full duplex hook-up must be created internally. This is accomplished by moving two wires on a terminal strip (see Figures 5-6 and 5-9).

(3) The receiver current level must be changed from 60 mA to 20 mA. This is accomplished by moving a single wire (see Figures 5-6 and 5-9).

(4) A relay circuit must be introduced into the paper tape reader drive circuit. The circuit consists of a relay, resistor, a diode, a thyractor and a suitable mounting fixture. This change requires the assembly of a small "vector" board with the relay circuit on it. It may be mounted in the Teletype by using two tapped holes in the base plate (see Figure 5-3). The relay circuit may then be added without alteration of the existing circuit (see Figures 5-4, 5-5 and 5-6). That is, wire "A" (Figure 5-9), to be connected to the brown wire in Figure 5-4 may be spliced into the brown wire near its connector plug. The "line" and "local" wires must then be connected to the mode switch (see Figures 5-8 and 5-9).

External Connections

(1) A two-wire receive loop must be created. This is accomplished by the connection of two wires between the Teletype and the BOARD in accordance with Figure 5-9.

(2) A two-wire send loop similar to the receive loop must be created. (See Figure 5-9.)

(3) A two-wire tape reader loop connecting the reader control relay to the BOARD must be created. (See Figure 5-9.)

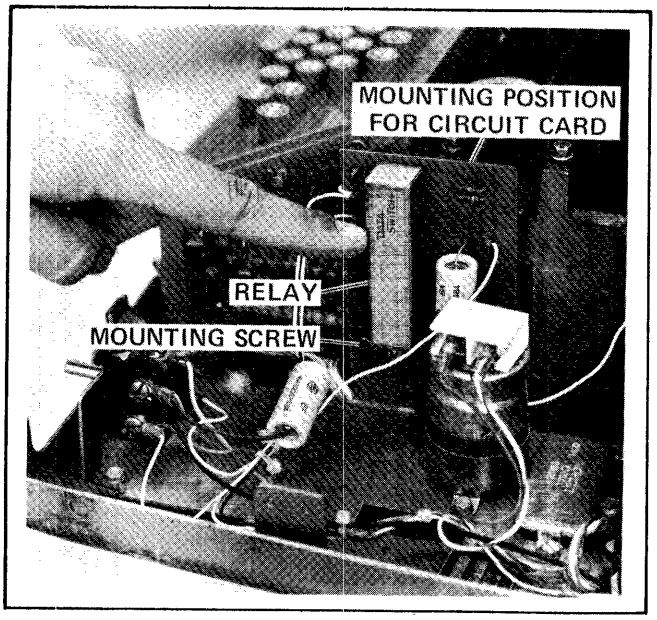


Figure 5-3. Relay Circuit (Alternate)

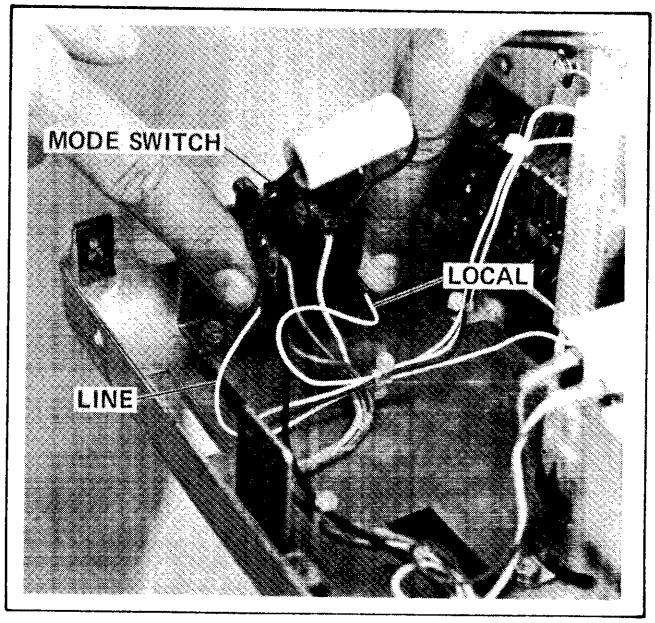


Figure 5-5. Mode Switch

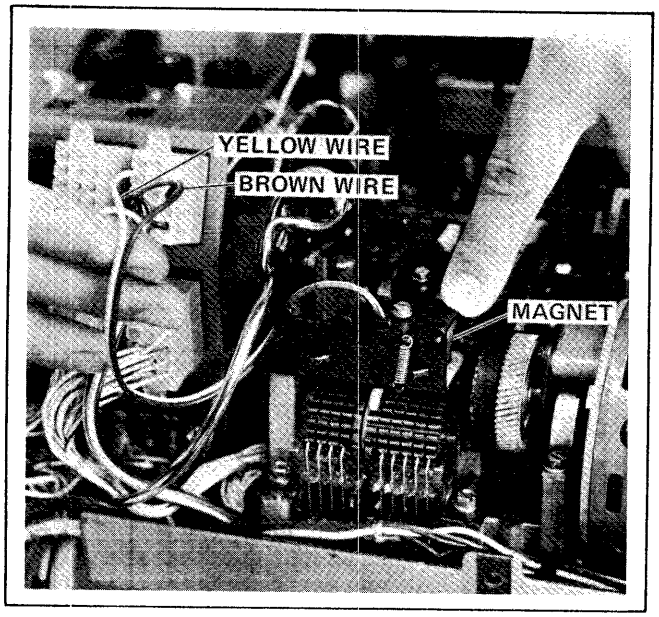


Figure 5-4. Distributor Trip Magnet

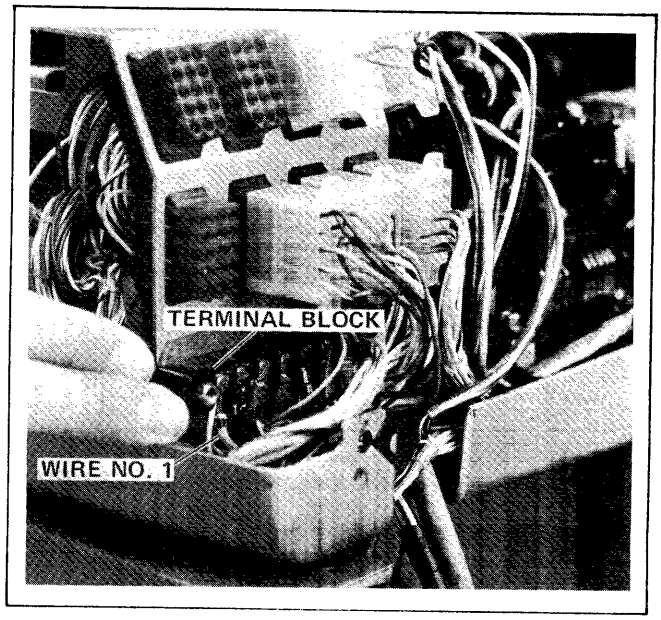


Figure 5-6. Terminal Block

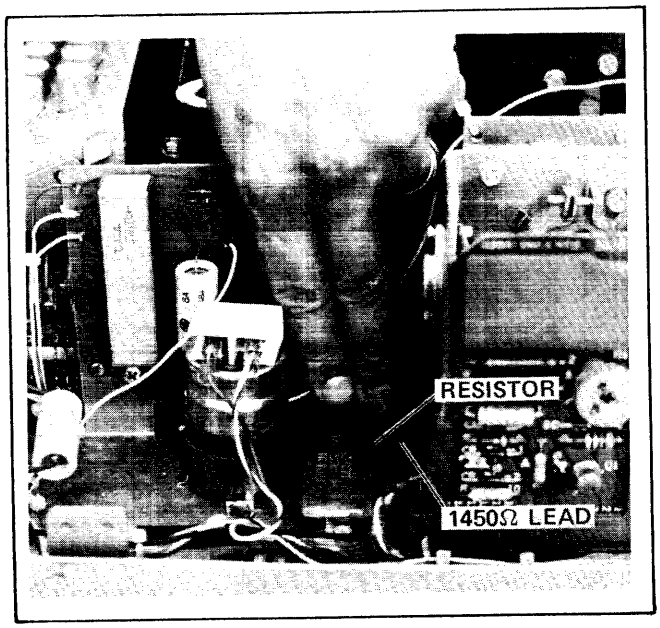


Figure 5-7. Current Source Resistor

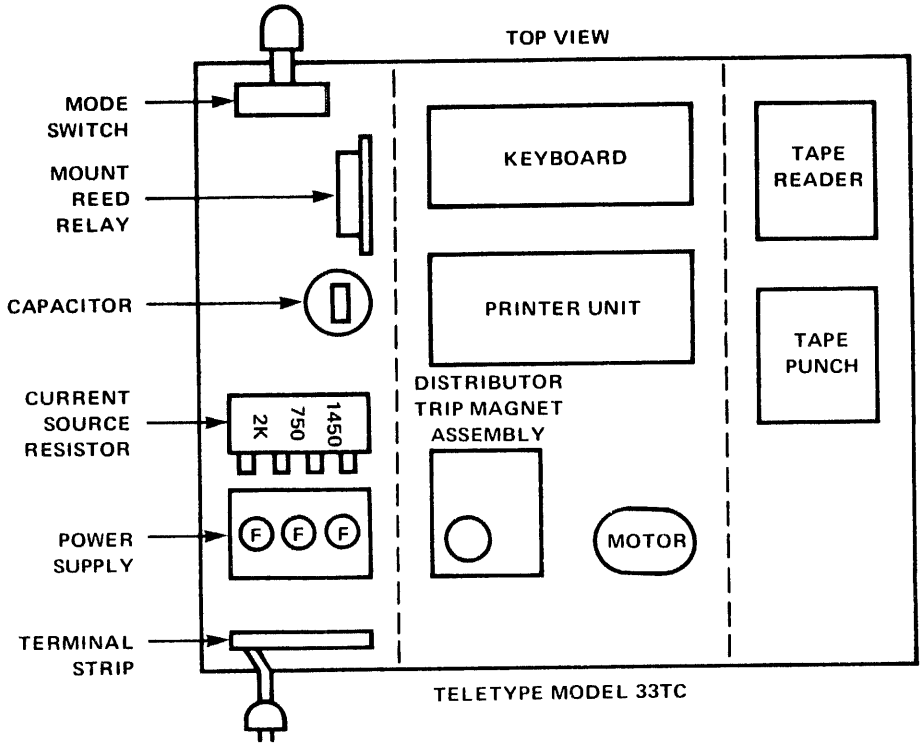


Figure 5-8. Teletype Layout

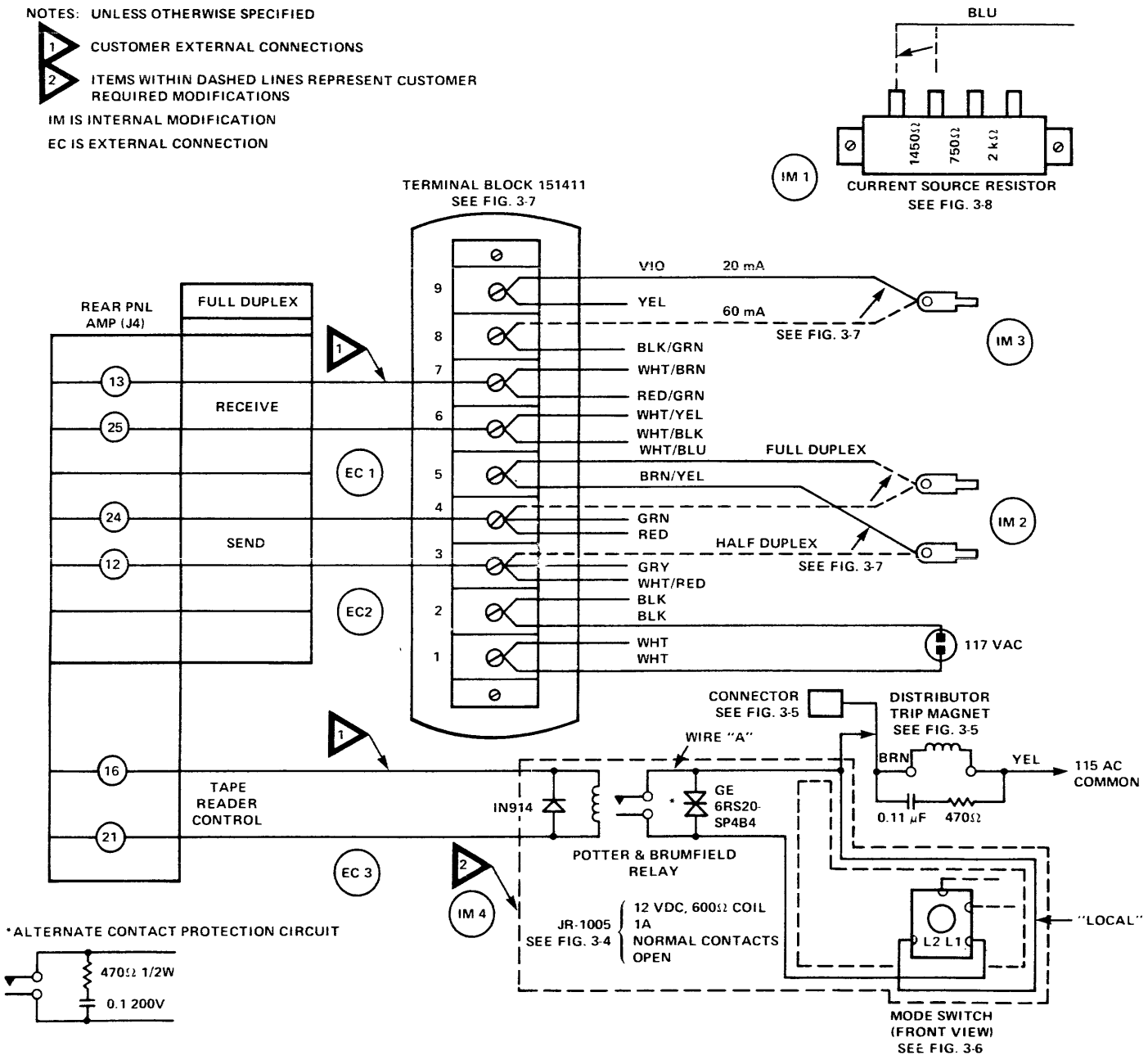


Figure 5-9. Teletype Modification

CHAPTER 6

COMPATIBLE EQUIPMENT

The SBC-80/10 is designed to operate with several other OEM modules. It is mechanically compatible with both Intellec® MDS chassis requirements and SBC-604 and SBC-614 4-module card holder, designed specifically for OEM applications. Details are presented in the following sections.

6.1 SBC-104 COMBINATION MEMORY AND I/O EXPANSION BOARD

The SBC-80/10 is completely compatible with the SBC-104 OEM Combination Memory and I/O Board module. The SBC-80/10 can be interfaced with up to 10 combination modules. Table 6-1 summarizes inter-module access characteristics.

The board includes 48 programmable I/O lines, an RS232C communications interface, 4K bytes of RAM memory and capacity for up to 4K bytes of EPROM/ROM memory. Eight interrupt request lines and a pending interrupt request register are on the board. Memory, I/O and interrupt register addresses are jumper selectable.

6.2 MASTER MODULES

The SBC-80/10 module may operate in conjunction with other master modules. But, its bus exchange mechanism must operate on an asynchronous basis (higher performance modules support the synchronous bus exchange mechanism of the Intellec® MDS family). Its use is limited by the following

constraints. The SBC-80/10 must be placed in the lowest priority position on the bus. Its operation will be totally suspended in a "hold" condition during the bus access of other masters. The SBC-80/10 memory and I/O ports are inaccessible to other OEM CPU Modules.

6.3 SBC-016 16K RAM MODULE

SBC-016 16K RAM Modules allow the user to expand the read/write storage capability of his system in increments of 16,384 bytes. This RAM Module includes thirty-two 2107 dynamic RAM elements (4096 bits each). Because the RAM elements are dynamic, they require periodic refreshing which is provided by logic on the module. See Table 6-1 for access characteristics.

Up to three 16K RAM Modules can be interfaced to the SBC-80/10. These RAM modules can be put in address locations 4000_{16}, 8000_{16}, or $C000_{16}$ only.

TABLE 6-1. SBC-80/10 INTER-MODULE ACCESS CHARACTERISTICS

MODULE	INSTRUCTION	CPU CYCLES		CPU WAIT STATES		CYCLE TIME (μsec)		REFRESH DEGRADATION	
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
COMBINATION	MR	8	8	1	1	3.9		0	+1
	MW	9	9	2	2	4.4			
	IOR	11	11	1	1	5.4			
	IOW	12	12	2	2	5.9			
16K RAM	MR	8	8	1	1	3.9		0	+2
	MW	9	9	2	2	4.4		0	+2
6K PROM*	MR	9	9	2	2	4.4			
	MW	BUS TIME OUT							
GP I/O	IOR	10	15	0	5	4.9	7.4		
	IOW	13	15	2	5	6.3	7.4		
16K PROM	MR	8	8	1	1	3.9			
	MW	BUS TIME OUT							

This chart assumes instruction fetch is from on board memory and data fetch is from off board memory.

*Assumes 6K PROM module uses 1702As (1 μs access time).

REFERENCE:

MR . . . . MEMORY READ: MOV A,M . . . . 7 CYCLES
MW . . . . MEMORY WRITE: MOV M,A . . . . 7 CYCLES
IOR . . . I/O READ: IN Addr . . . . 10 CYCLES
IOW . . . I/O WRITE: OUT Addr . . . 10 CYCLES

The number of cycles assumes all memory is on board.

However, they may only reside in memory addresses above 3FFF (hexadecimal).

6.4 SBC-406 6K PROM MODULE

SBC-406 6K PROM Modules allow the user to expand the read only storage capability of his system in increments of 2K, 4K or 6K bytes. This PROM Module can include up to twenty-four 8702A PROM devices (256 X 8-bits each). Intel's 1702A PROM's or 1302 ROM's (both pin compatible with the 8702A) can also be used on the PROM Module, in place of the 8702A's.

The 24 memory devices are organized into a 4K (4096) byte memory bank and a 2K (2048) byte memory bank. The user independently selects the address range for the 4K and 2K memory banks on 4K and 2K boundaries, respectively. See Table 6-1 for access characteristics.

Up to ten 6K PROM Modules can be interfaced to the SBC-80/10. However, they may only reside in memory addresses 2000 to 38FF or above 3FFF (hexadecimal).

Note: The 6K PROM Module requires a -10V supply in addition to standard OEM voltages.

6.5 SBC-416 16K PROM MODULE

The SBC-416 allows the user to expand his nonvolatile memory with 8708 EPROMs or 8308 ROMs in 1K bytes. Address locations for memory on the board are jumper selectable in 4K blocks. Beginning addresses for any block of memory must start at a multiple of 4K (0, 4K, 8K, 12K, etc.).

6.6 SBC-508 GENERAL PURPOSE I/O MODULE

The SBC-508 GP I/O Module includes four input and four output ports. Each output port latches 8-bit data words and issues a framed strobe pulse, of selectable duration, to the peripheral device. All outputs are driven by TTL level buffer drivers. Each input port also supports 8-bits of data, latched or unlatched. All inputs are terminated by dual-in-line socket-mounted resistor packs. The I/O Module also includes provisions for accepting eight external interrupt requests, buffering them and driving them to a CPU module. See Table 6-1 for access characteristics.

Up to nine BP I/O Modules can be interfaced to the SBC-80/10. I/O addresses selected for the GP I/O Module in this application must not coincide with the SBC-80/10's dedicated I/O addresses.

6.7 MODULAR BACKPLANE AND CARDCAGE

The SBC-604/614 Modular Backplane and Cardcage is designed specifically for OEM modules such as the SBC-80/10. Each card holder supports up to 4 modules. The modules may be electrically and mecahnically "ganged" together for expanded capability. Provisions for power supply distribution, air circulation and bus exchange functions are featured on the OEM card holders (see Section 4.4).

6.8 INTELLEC® MDS CARDCAGE

The mechanical characteristics of the Intellec MDS cardcage are compatible with the SBC 80/10.

Note: Intellec MDS power supplies do not support a -5V supply although a -5V bus conductor is built into the cardcage motherboard. An auxiliary power supply or converter connected to this line will suffice.

CHAPTER 7

SBC-80/10 SPECIFICATIONS

7.1 DC POWER REQUIREMENTS

DC Power Requirements are given in Table 7-1.

7.2 AC CHARACTERISTICS

AC Characteristics are given in Tables 7-2 and 7-3 and Figures 7-1, 7-2 and 7-3.

7.3 DC CHARACTERISTICS

DC Characteristics are given in Table 7-4.

7.4 ENVIRONMENT

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must, therefore, be maintained within the limits of $0^{o}C$ to $55^{o}C$. Exercise caution in locating the module, giving particular attention to radiant and conducive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance to permit the convective dissipation of heat from the elements on the card.

Relative humidity should not exceed 90%, non-condensing.

7.5 SBC-80/10 BOARD OUTLINE

See Figure 7-4.

7.6 SBC 80/10 Compatible Connectors

Table 7-5 lists compatible connectors which mate to the SBC 80/10 PC edge connectors.

TABLE 7-1. DC POWER REQUIREMENTS

	(1)	(2)
V_{CC} = +5V ± 5%	I_{CC} = 4.0A max.	I_{CC} = 2.9A
V_{DD} = +12V ± 5%	I_{DD} = 0.40A max.	I_{DD} = 140mA
V_{BB} = -5V ± 5%	I_{BB} = 0.20A max	I_{BB} = 2mA
V_{AA} = -12V ± 5%	I_{AA} = 0.175 max.	I_{AA} = 175mA

Notes:

(1) The values assume that four 8708 PROM's are present and that ten optional 220/330 Ω termination networks being driven low have been installed in the Parallel I/O Interface.

(2) These values assume that the 8708 PROM's and optional termination networks are not present.

TABLE 7-2. SBC-80/10 AC CHARACTERICS

PARAMETER	OVERALL		WITH BUS EXCHANGE				DESCRIPTION	REMARKS
			READ		MEMORY WRITE			
	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)		
t_{AS}	82		82		658		Address Setup Time to Command	
t_{AH}	61				61		Address Hold Time	
t_{DS}	140				140		Data Setup Time to Command	
t_{DH}					61		Data Hold Time	
t_{RDY}			65	215			First Ready Sampling Point of Current Cycle	
t_{RDY2}			548	683	-60	70	Second Ready Sampling Point of Current Cycle	
t_{RDY3}			1031	1176	416		Third Ready Sampling Point of Current Cycle	
t_{CY}	483	493					Cycle Time	
t_{WC}			596	796	1015	1115	Command Width	1
t_{ACC}			344				Read Access Time	1
t_{8KD}			65				8080 ACK Response Time for Mimimum Delay	1
t_{8KO}	100		100		100		8080 ACK Turn Off Delay	1
t_{XKD}			50				XACK Delay From Valid Data or Write	
t_{XKO}	100		100		100		XACK Turn Off Delay	
t_{DBS}		3500					Bus Sample to Exchange Initiation	1 Assume HOLD/ becomes active prior to DAD instruction
t_{BS}	483	493					Bus Sampling Point Delay	
t_{DBY}	358	1100					Bus Busy Turn On Delay	

1 Memory and I/O access occurs with no wait states.

TABLE 7-3. SBC-80/10 AC CHARACTERISTICS

PARAMETER	OVERALL		CONTINUOUS BUS CONTROL				DESCRIPTION	REMARKS
			READ		MEMORY WRITE			
	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)		
t_{AS}	82		82		658		Address Setup Time to Command	
t_{AH}					79		Address Hold Time	
t_{DS}	140				140		Data Setup Time to Command	
t_{DH}	79				79		Data Hold Time	
t_{RDY}			65	215			First Ready Sampling Point of Current Cycle	
t_{RDY2}			548	683	-60	70	Second Ready Sampling Point of Current Cycle	
t_{RDY3}			1031	1176	416		Third Ready Sampling Point of Current Cycle	
t_{CY}	483	493					Cycle Time	
t_{SEP}			613		259	▷2	Command Separation	
t_{WC}			596	796	1015	1115	Command Width	▷1
t_{ACC}			344				Read Access Time	▷1
t_{8KD}			65				8080 ACK Response Time for Minimum Delay	▷1
t_{8KO}	100		100		100		8080 ACK Turn Off Delay	
t_{XKD}			50				XACK Delay From Valid Data or Write	
t_{XKO}	100		100		100		XACK Turn Off Delay	
t_{BCY}	107	110					Bus Clock Cycle Time	
t_{BW}	25	85					Bus Clock Low or High Periods	
t_{INT}	3000						Initialization Width	After all voltages have stabilized

▷1 MAX assumes no acknowledge delays.

▷2 Write Command to next Read Command separation.

TABLE 7-4. SBC 80/10 DC CHARACTERISTICS

SIGNALS	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNITS
ADRØ/-ADRF/ ADDRESS	V_{OL}	Output Low Voltage	I_{OL} = 50 mA		0.6	V
	V_{OH}	Output High Voltage	I_{OH} = -10 mA	2.4		V
	V_{IL}	Input Low Voltage			0.95	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		-0.25	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		10	µA
	C_L	Capacitive Load			18	pF
MROC/,MWTC/ IORC/,IOWC/	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = -5.2 mA	2.4		V
	I_{LH}	Output Leakage High	V_O = 2.4		40	µA
	I_{LL}	Output Leakage Low	V_O = 0.4		-40	µA
	C_L	Capacitive Load			15	pF
DATØ/-DAT7/	V_{OL}	Output Low Voltage	I_{OL} = 50 mA		0.6	V
	V_{OH}	Output High Voltage	I_{OH} = -10 mA	2.4		V
	V_{IL}	Input Low Voltage			0.95	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		-0.25	mA
	I_{LH}	Output Leakage High	V_O = 5.25		100	µA
	I_{LL}	Output Leakage Low	V_O = 0.45		100	µA
	C_L	Capacitive Load			18	pF
INT1/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		-2.2	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.5V		1	mA
	C_L	Capacitive Load			18	pF
BPRN/,XACK AACK	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.5		-2.6	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.7V		0.30	mA
	C_L	Capacitive Load			18	pF
BUSY/ OPEN COLLECTOR	V_{OL}	Output Low Voltage	I_{OL} = 25 mA		0.4	V
	C_L	Capacitive Load			20	pF
INT (SYSTEM RESET)	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.6	V
	V_{OH}	Output High Voltage	OPEN COLLECTOR			
	V_{IL}	Input Low Voltage			0.7	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 5.5		0.1	mA
	I_{IH}	Input Current at High V	V_{IN} = 0.3		-0.7	mA
	C_L	Capacitive Load			38	pF
BCLK + CCLK	V_{OL}	Output Low Voltage	I_{OL} = 20 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = -1 mA	2.7		V
	C_L	Capacitive Load			18	pF

TABLE 7-4. SBC 80/10 DC CHARACTERISTICS (Continued)

SIGNALS	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNITS
EXT INTRØ/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V	6.8		mA
	I_{IH}	Input Current at High V	V_{IN} = 5.5V		2	mA
	C_I	Capacitive Load			18	pF
PORT E4 BIDIRECTIONAL DRIVERS	V_{OL}	Output Low Voltage	I_{OL} = 20 mA		.45	V
	V_{OH}	Output High Voltage	I_{OH} = -12 mA	2.4		V
	V_{IL}	Input Low Voltage			.95	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		5.25	mA
	I_{LH}	Output Leakage High	V_O = 5.25		.30	mA
	I_{LL}	Output Leakage Low	V_O = 0.45		5.25	mA
	C_L	Capacitive Load			18	pF
8255 DRIVER/ RECEIVER	V_{OL}	Output Low Voltage	I_{OL} = 1.7 mA		.45	V
	V_{OH}	Output High Voltage	I_{OH} = -50 µA	2.4		V
	V_{IL}	Input Low Voltage			.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		10	µA
	I_{IH}	Input Current at High V	V_{IN} = 5.0		10	µA
	C_L	Capacitive Load			18	pF

TABLE 7-5

SBC 80/10 COMPATIBLE CONNECTOR HARDWARE

FUNCTION	CONNECTOR TYPE	VENDOR	INTEL PART #	VENDOR PART #
PARALLEL I/O [1]	FLAT CABLE | FLAT CABLE	3M AMP		3415-0001 2-86792-3
SERIAL I/O [1]	FLAT CABLE | FLAT CABLE	3M AMP		3462-0001 88106-1
PARALLEL I/O [1]	SOLDERED | SOLDERED	AMP VIKING TI		2-583715-3 3VH25/1JV5 H312125
SERIAL I/O [1]	SOLDERED | SOLDERED	TI AMP		H312113 1-583715-1
AUXILIARY [1]	SOLDERED | SOLDERED	VIKING TI		3VH30/1JN5 H312130
BUS [1]	SOLDERED | SOLDERED	CDC MICRO PLASTICS ARCO VIKING		VPB01E43D00A1E MP-0156-43-BW-4 AE443WP1 LESS EARS 2VH43/1AV5
PARALLEL I/O [1]	WIREWRAP | WIREWRAP	TI VIKING CDC ITT CANNON		H311125 3VH25/1JND5 VPB01B25D00A1 EC4A050A1A
SERIAL I/O [1]	WIREWRAP | WIREWRAP	TI		H311113
PARALLEL I/O [1]	CRIMP | CRIMP	AMP		1-583717 1
SERIAL I/O [1]	CRIMP | CRIMP			
AUXILIARY [2] [1]	WIREWRAP | WIREWRAP	CDC TI AMP		VPB01B30D00A1 H311130 PE5-14559
BUS [2] [1]	WIREWRAP | WIREWRAP	CDC VIKING		VFB01E43D00A1 2VH43/1ANE5

[1] Connector heights are not guaranteed to conform to OEM packaging equipment. Intel OEM and MDS motherboards offer complete mechanical compatibility.

[2] Wirewrap pin lengths are not guaranteed to conform to OEM packaging equipment. Intel connectors and OEM and MDS motherboards offer complete mechanical compatibility.

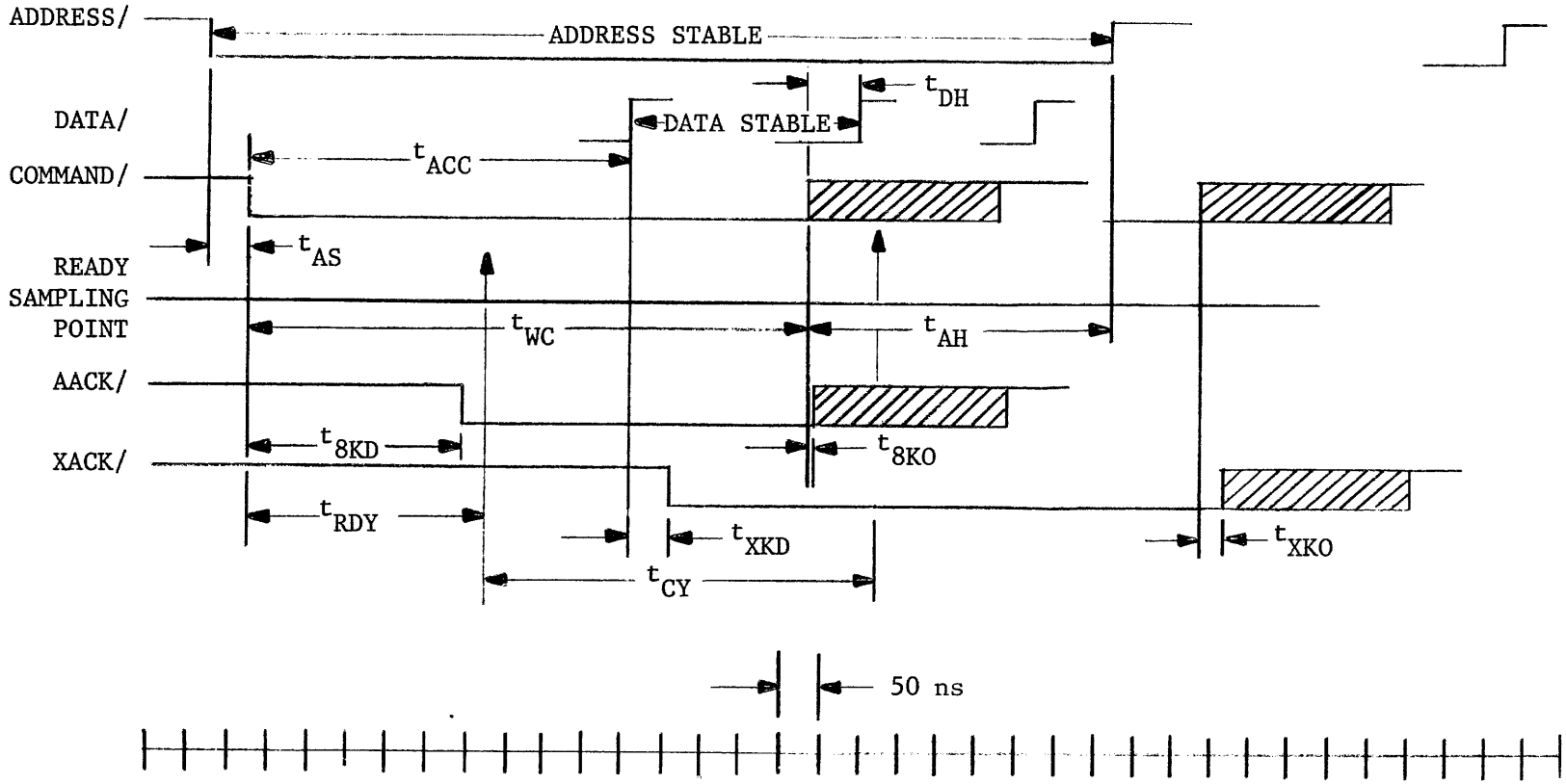


FIGURE 7-1. MEMORY AND I/O READ TIMING (CONTINUOUS BUS CONTROL)

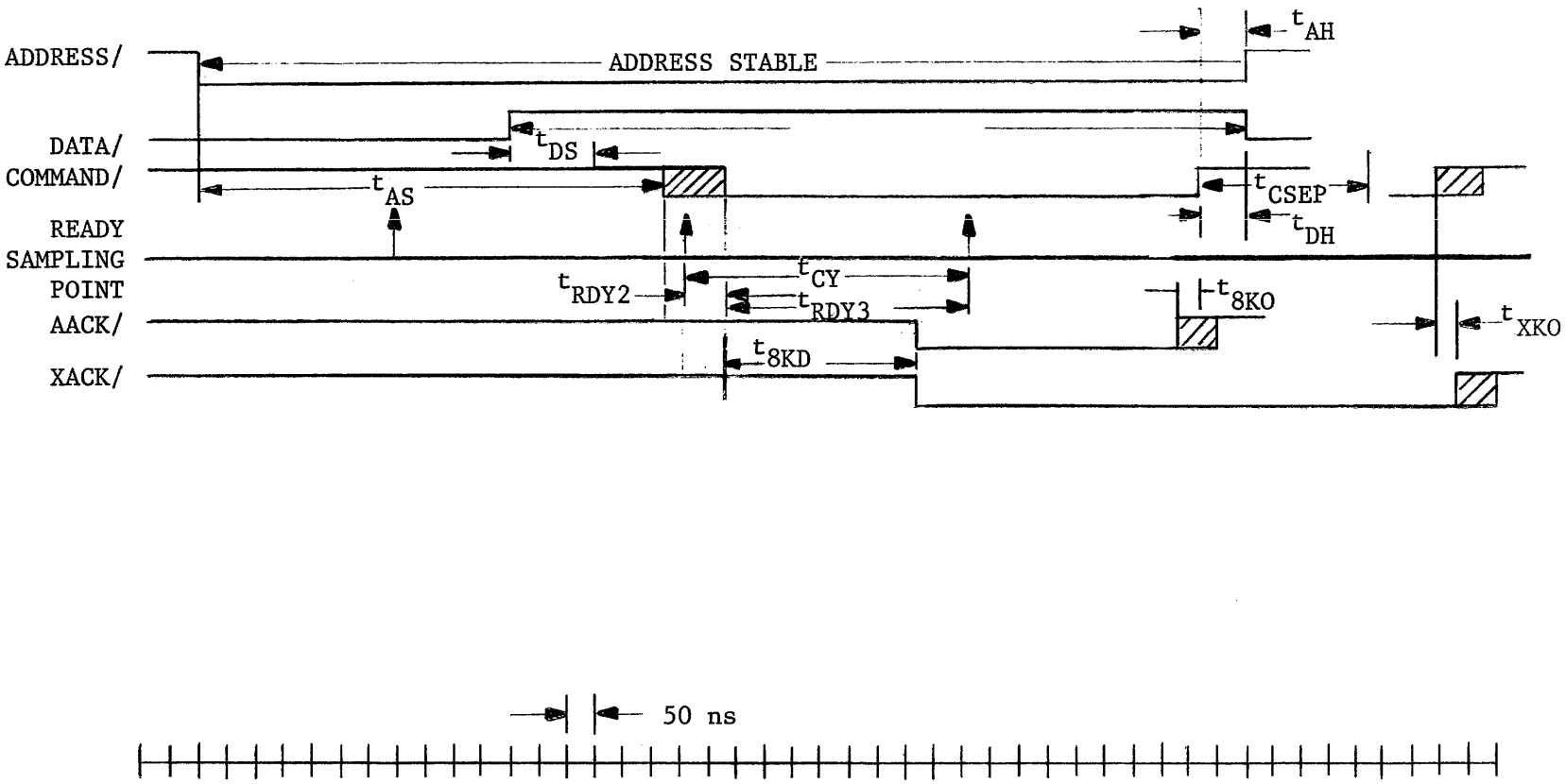


FIGURE 7-2. MEMORY AND I/O WRITE TIMING (CONTINUOUS BUS CONTROL)

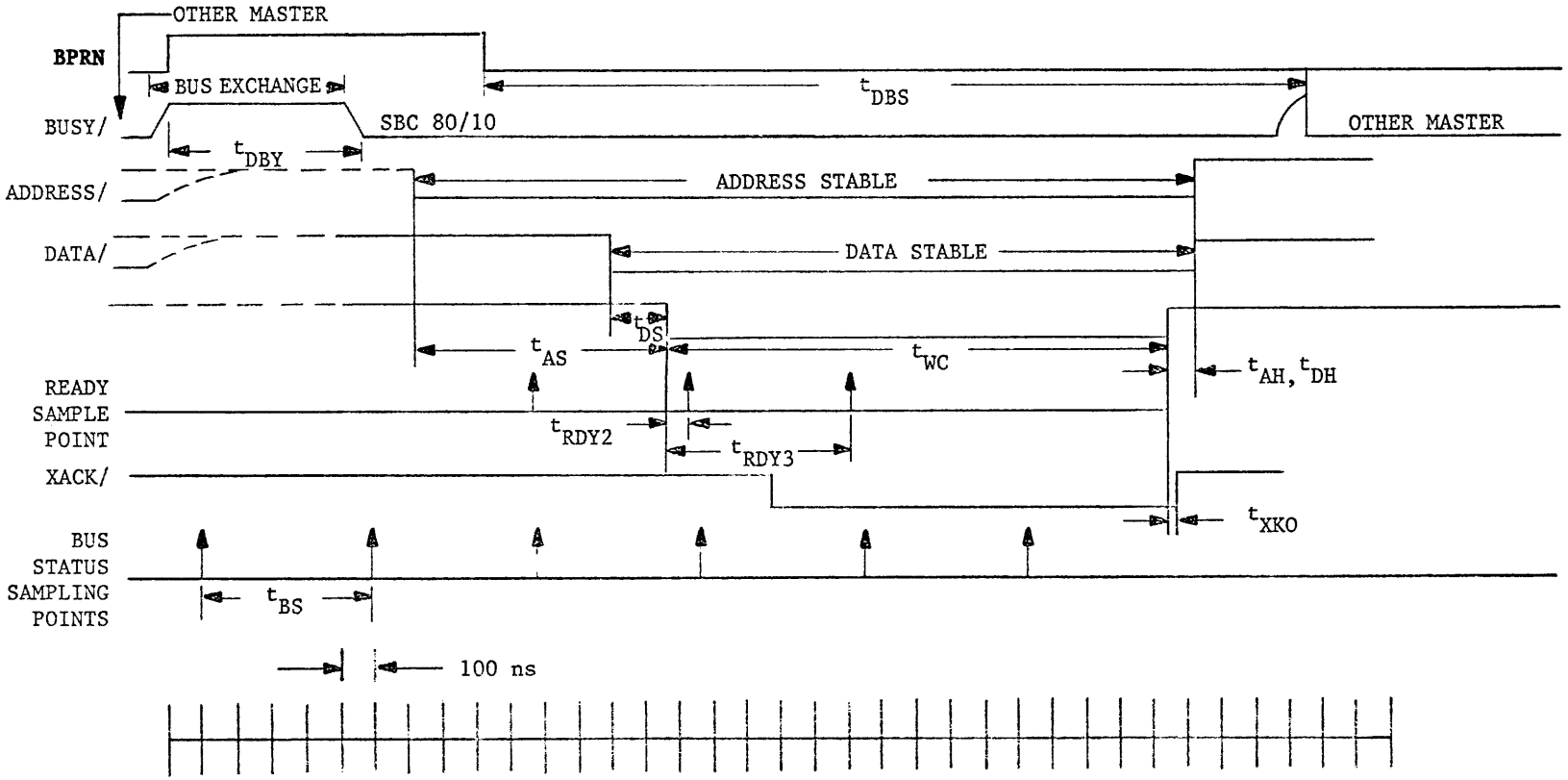


FIGURE 7-3. BUS EXCHANGE (WRITE)

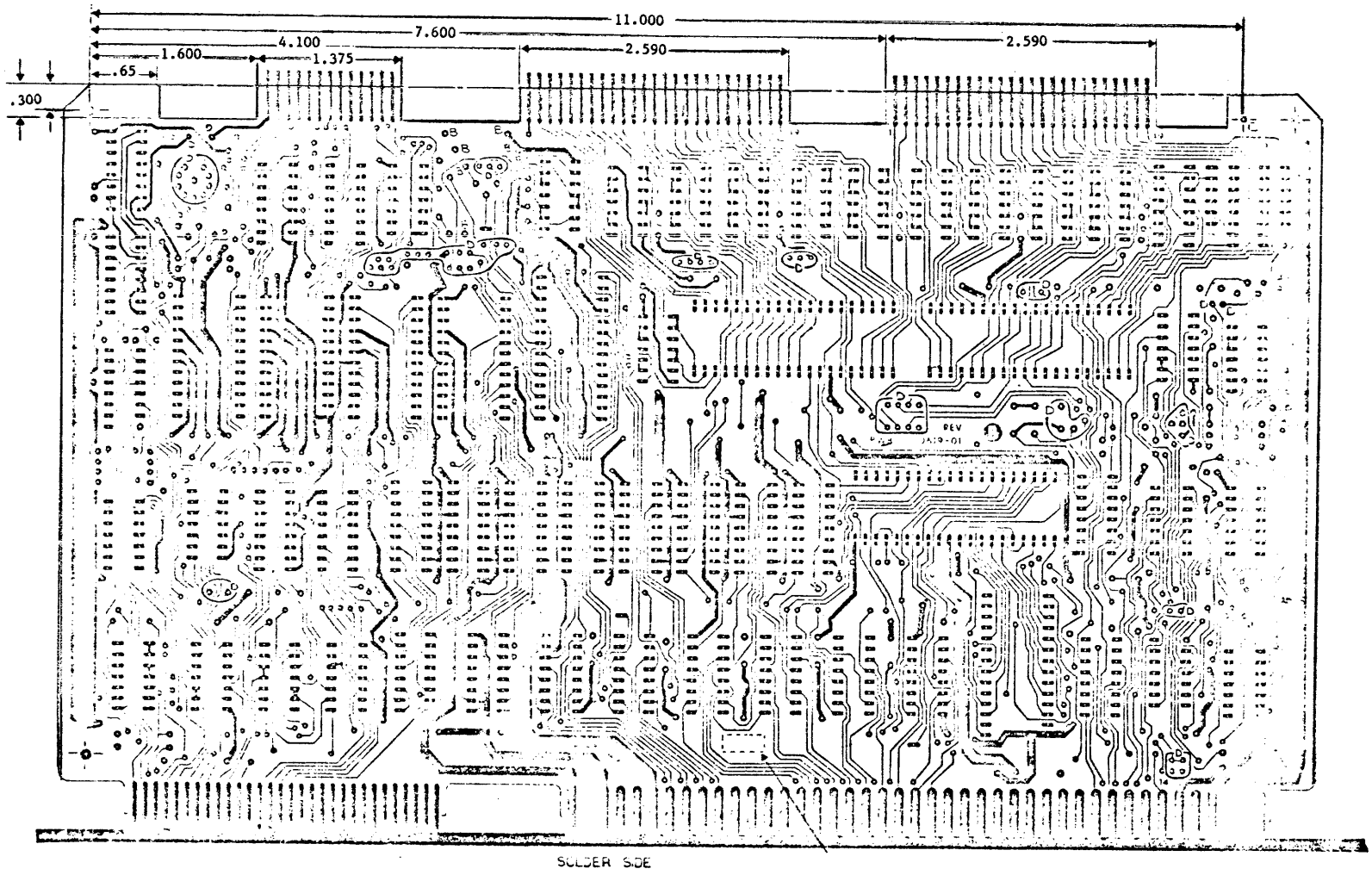
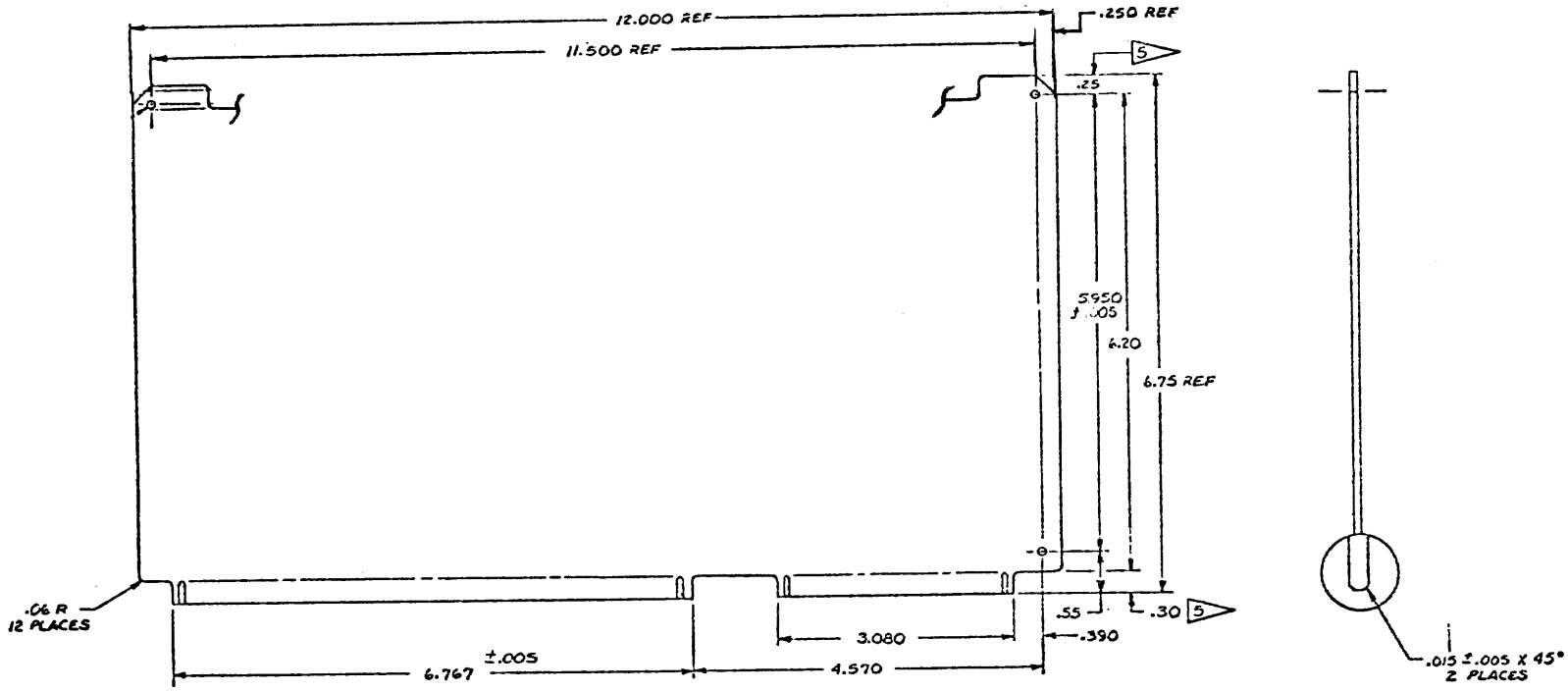


Figure 7-4 SBC 80/10 Dimension
Drawing - Page 1 of 2



NOTES:

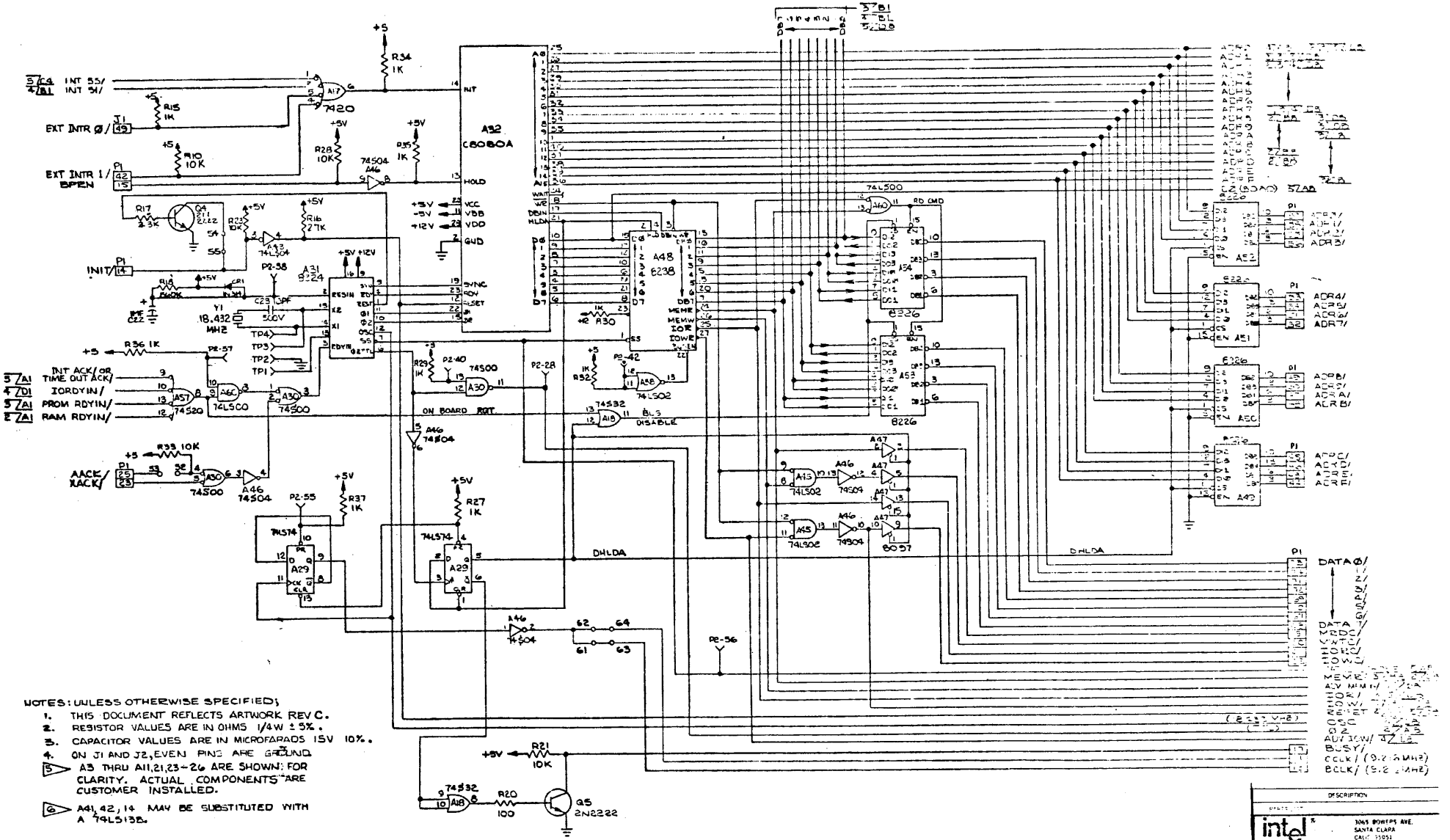
1. MATERIAL : .062 THK, 1 OZ COPPER CLAD, NATURAL EPOXY GLASS, TYPE G10 (2OZ AFTER PLATING THRU)
2. BOARD EDGES ARE LOCATED FROM INDEX HOLES. INDEX HOLES ARE ON .050 GRID INTERSECTION AND ARE USED FOR ARTWORK REGISTRATION AND MAY BE USED AS TOOLING HOLES, PLATING OPTIONAL.
3. HOLES ARE PLATED THRU WITH COPPER WALL THICKNESS OF .0007 MINIMUM.
4. HOLE SIZES SPECIFIED ARE AFTER PLATING; ±.003 TOLERANCE
5. CONTACT FINGERS ARE OVERPLATED WITH A MINIMUM OF 50 MILLIONTHS GOLD OVER NICKEL TO DIMENSION SHOWN.
6. APPLY SOLDER MASK OVER SOLDER PLATE USING MATERIAL; MECUMASK GREEN
7.
8. DRILL FROM CIRCUIT SIDE.
9. TRACE WIDTHS MUST BE WITHIN .004 OF ARTWORK NEGATIVES.
10. APPLY SILKSCREEN ON COMPONENT SIDE, AFTER SOLDER MASK IS APPLIED, USING WHITE EPOXY INK.

Figure 7-4 SBC 80/10 Dimension Drawing - Page 2 of 2

APPENDIX A

SBC-80/10 SCHEMATICS

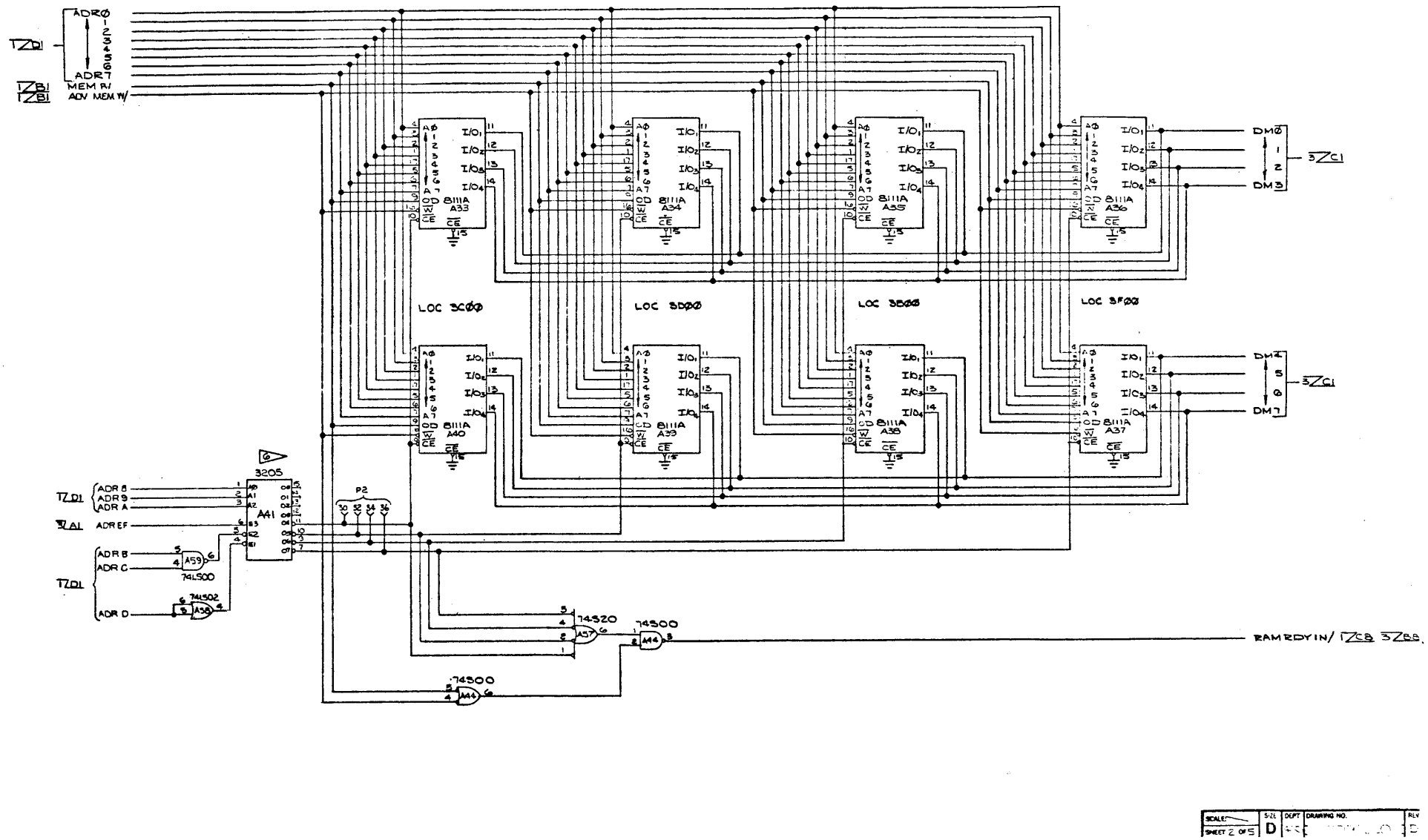
Schematic drawings for the SBC-80/10 are provided in this appendix. Information and diagrams in this section are subject to change without notice. References should be made to schematics shipped with this module.

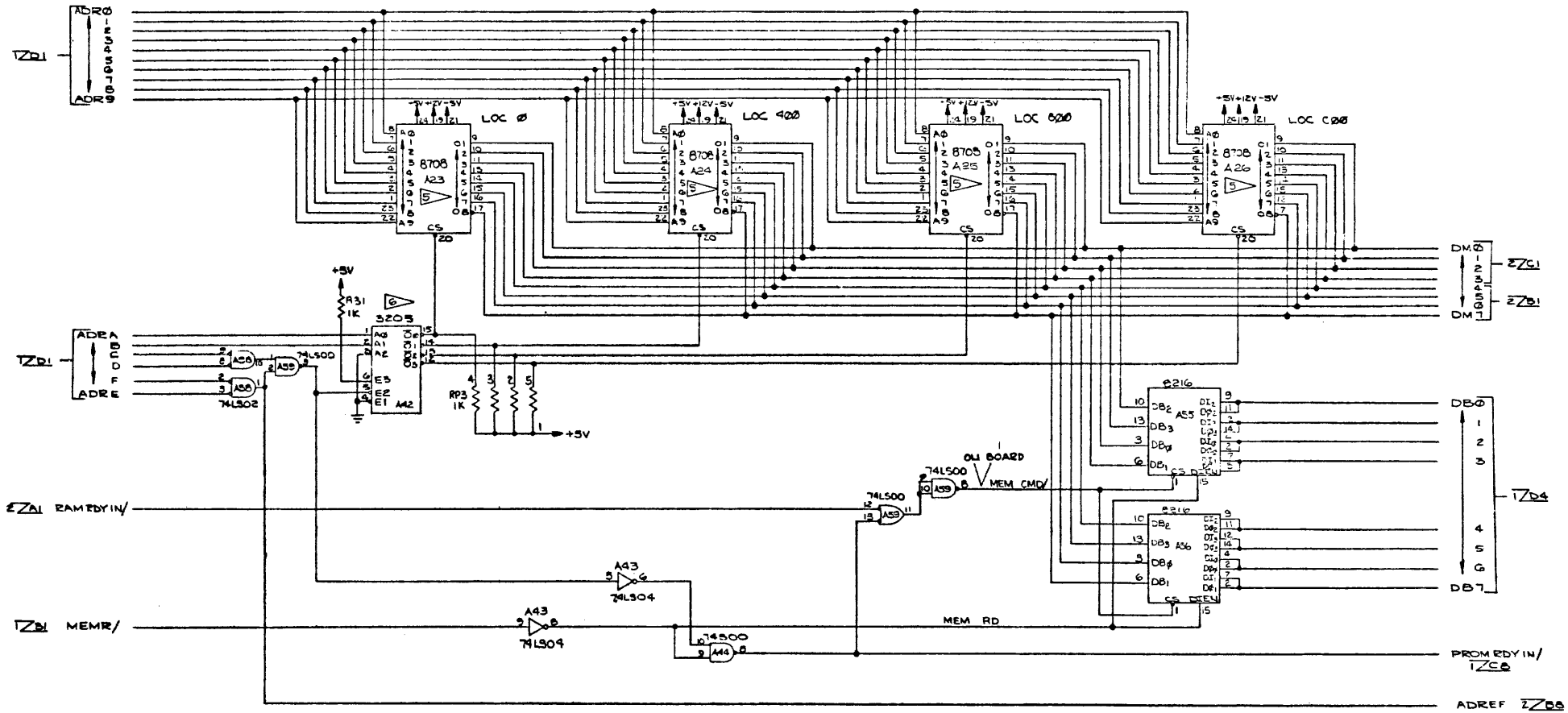


NOTES: UNLESS OTHERWISE SPECIFIED;

1. THIS DOCUMENT REFLECTS ARTWORK REV C.
2. RESISTOR VALUES ARE IN OHMS 1/4W ± 5%.
3. CAPACITOR VALUES ARE IN MICROFARADS 15V 10%.
4. ON J1 AND J2, EVEN PINS ARE GROUND.
5. A3 THRU A11,21,23-26 ARE SHOWN; FOR CLARITY. ACTUAL COMPONENTS ARE CUSTOMER INSTALLED.
6. A41,42,14 MAY BE SUBSTITUTED WITH A 74LS138.

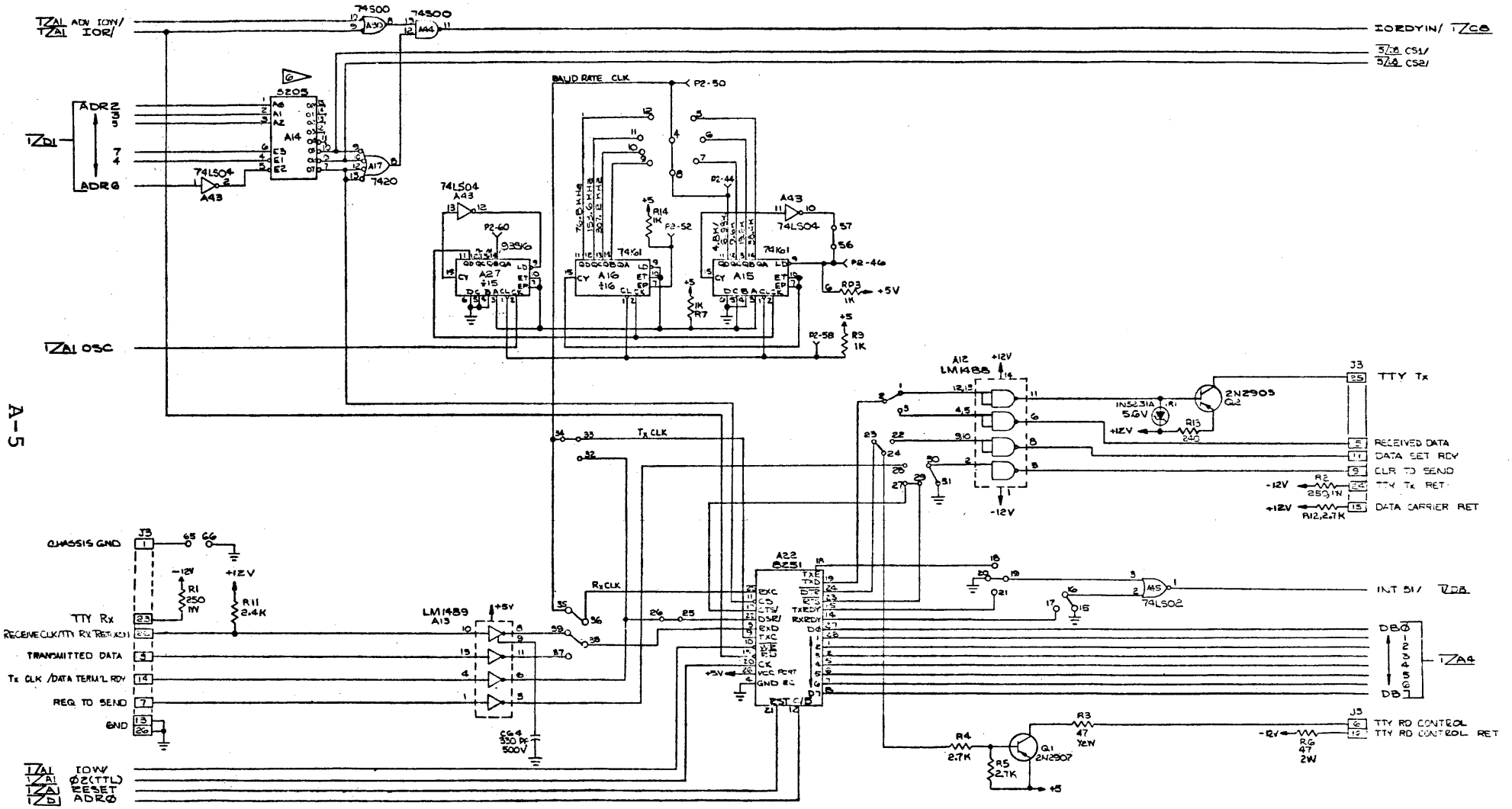
DESCRIPTION		
intel	3065 BOWERS AVE. SANTA CLARA CALIF. 95051	
SCHEMATIC, CIRCUIT BD AND COMPUTER SYS		
SIZE D	DWG NO.	REV


ADR Ø
ADR 7
MEM R/
ADV MEM W/
8111A
LOC 3CØØ
LOC 3DØØ
LOC 3EØØ
LOC 3FØØ
3205
A41
74S20
74S00
74S02
74S00
RAMRDYIN/



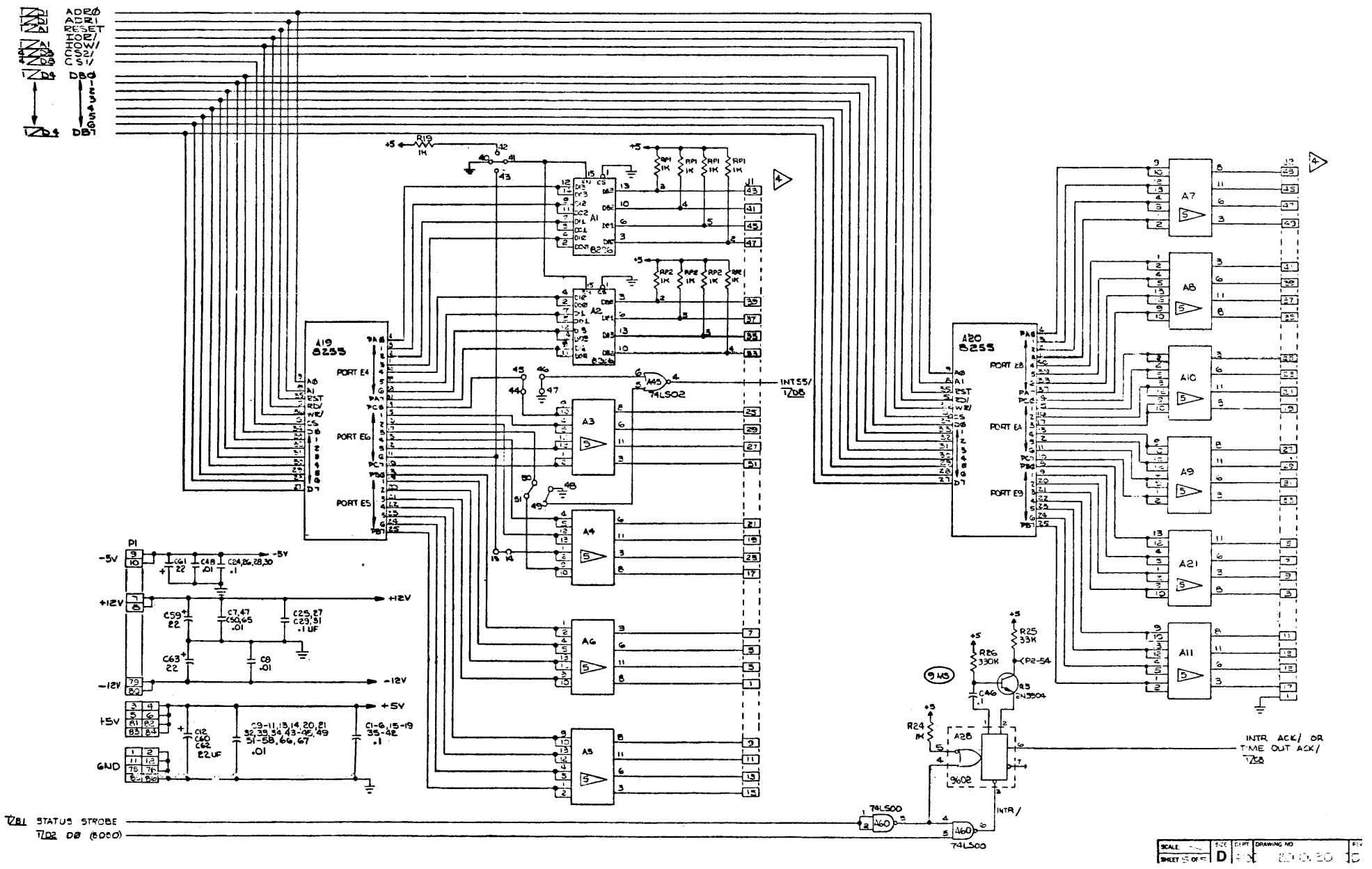
A-4

SCALE	SIZE	DEPT	DRAWING NO.	REV
SHEET 5 OF 15	D			D



A-5

SCALE:	SIZE	DEPT	DRAWING NO	REV
SHEET 1 OF 5	D			D


ADR0
ADR1
RESET
IOR/
IOW/
CS2/
CS1/
DB0
DB7
R19
1K
A19
8255
A20
8255
PORT EA
PORT EB
PORT EC
PORT EA
PORT EB
PORT EC
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A21
A11
A28
A45
74LS02
INT55/
TL08
P1
-5V
+12V
-12V
+5V
GND
C61 22
C48 .01
C24,26,28,30 .1
C59 22
C7,47 C50,65 .01
C25,27 C29,51 .1 UF
C63 22
C8 .01
C12 C60 C62 22 UF
C9-11,13,14,20,21 32,33,34,43-45,49 51-58,66,67 .01
C1-6,15-19 35-42 .1
R25 33K
R26 330K
P2-54
Q5 2N3904
C46 .1
R24 1K
9602
74LS00
A60
INTR/
INTR ACK/ OR TIME OUT ACK/
TL08
TL01 STATUS STROBE
TL02 D0 (8080)

APPENDIX B

8080 INSTRUCTION SET SUMMARY

A computer, no matter how sophisticated, can only do what it is "told" to do. One "tells" the computer what to do via a series of coded instructions referred to as a Program. The realm of the programmer is referred to as Software, in contrast to the Hardware that comprises the actual computer equipment. A computer's software refers to all of the programs that have been written for that computer.

When a computer is designed, the engineers provide the Central Processing Unit (CPU) with the ability to perform a particular set of operations. The CPU is designed such that a specific operation is performed when the CPU control logic decodes a particular instruction. Consequently, the operations that can be performed by a CPU define the computer's Instruction Set.

Each computer instruction allows the programmer to initiate the performance of a specific operation. All computers implement certain arithmetic operations in their instruction set, such as an instruction to add the contents of two registers. Often logical operations (e.g., OR the contents of two registers) and register operate instructions (e.g., increment a register) are included in the instruction set. A computer's instruction set will also have instructions that move data between registers, between a register and memory, and between register and an I/O device. Most instruction sets also provide Conditional Instructions. A conditional instruction specifies an operation to be performed only if certain conditions have been met; for example, jump to a particular instruction if the result of the last operation was zero. Conditional instructions provide a program with a decision-making capability.

By logically organizing a sequence of instructions into a coherent program, the programmer can "tell" the computer to perform a very specific and useful function.

The computer, however, can only execute programs whose instructions are in a binary coded form (i.e., a series of 1's and 0's), that is called Machine Code. Because it would be extremely cumbersome to program in machine code, programming languages have been developed. There are programs available which convert the programming language instructions into machine code that can be interpreted by the processor.

One type of programming language is Assembly Language. A unique assembly language mnemonic is assigned to each of the computer's instructions. The programmer can write a program (called the Source Program) using these mnemonics and certain operands; the source program is then converted into machine instructions (called the Object Code). Each assembly language instructions is converted into one machine code instruction (1 or more bytes) by an Assembler program. Assembly languages are usually machine dependent (i.e., they are usually able to run on only one type of computer).

THE 8080 INSTRUCTION SET

The 8080 instruction set includes five different types of instructions:

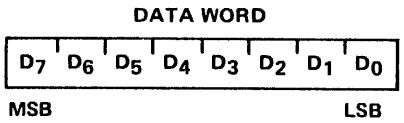
- *Data Transfer Group* – move data between registers or between memory and registers.
- *Arithmetic Group* – add, subtract, increment or decrement data in registers or in memory.
- *Logical Group* – AND, OR, EXCLUSIVE-OR, compare, rotate or complement data in registers or in memory.
- *Branch Group* – conditional and unconditional jump instructions, subroutine call instructions and return instructions.
- *Stack, I/O and Machine Control Group* – includes I/O instructions, as well as instructions for maintaining the stack and internal control flags.

Instruction and Data Formats

Memory for the 8080 is organized into 8-bit quantities, called Bytes. Each byte has a unique 16-bit binary address corresponding to its sequential position in memory.

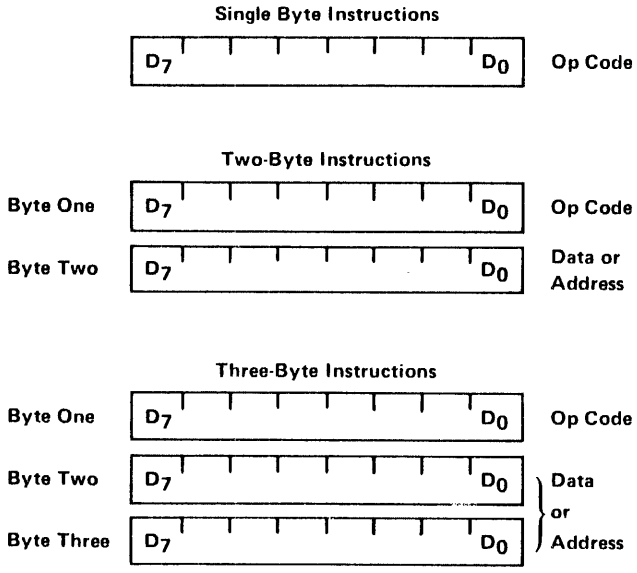
The 8080 can directly address up to 65,536 bytes of memory, which may consist of both read-only memory (ROM) elements and random-access memory (RAM) elements (read/write memory).

Data in the 8080 is stored in the form of 8-bit binary integers:



When a register or data word contains a binary number, it is necessary to establish the order in which the bits of the number are written. In the Intel 8080, BIT 0 is referred to as the Least Significant Bit (LSB), and BIT 7 (of an 8-bit number) is referred to as the Most Significant Bit (MSB).

The 8080 program instructions may be one, two or three bytes in length. Multiple byte instructions must be stored in successive memory locations; the address of the first byte is always used as the address of the instructions. The exact instruction format will depend on the particular operation to be executed.



Addressing Modes

Often the data that is to be operated on is stored in memory. When multi-byte numeric data is used, the data, like instructions, is stored in successive memory locations, with the least significant byte first, followed by increasingly significant bytes. The 8080 has four different modes for addressing data stored in memory or in registers:

- *Direct* – Bytes 2 and 3 of the instruction contain the exact memory address of the data item (the low-order bits of the address are in byte 2, the high-order bits in byte 3).

- *Register* – The instruction specifies the register-pair in which the data is located.

- *Register Indirect* – The instruction specifies a register-pair which contains the memory address where the data is located (the high-order bits of the address are in the first register of the pair, the low-order bits in the second).

- *Immediate* – The instruction contains the data itself. This is either an 8-bit quantity or a 16-bit quantity (least significant byte first, most significant byte second).

Unless directed by an interrupt or branch instruction, the execution of instructions proceeds through consecutively increasing memory locations. A branch instruction can specify the address of the next instruction to be executed in one of two ways:

- *Direct* – The branch instruction contains the address of the next instruction to be executed. (Except for the 'RST' instruction, byte 2 contains the low-order address and byte 3 the high-order address.)

- *Register Indirect* – The branch instruction indicates a register-pair which contains the address of the next instruction to be executed. (The high-order bits of the address are in the first register of the pair, the low-order bits in the second.)

The RST instruction is a special one-byte call instruction (usually used during interrupt sequences).

RST includes a three-bit field; program control is transferred to the instruction whose address is eight times the contents of this three-bit field.

Condition Flags

There are five condition flags associated with the execution of instructions on the 8080. They are Zero, Sign, Parity, Carry, and Auxiliary Carry, and are each represented by a 1-bit register in the CPU. A flag is "set" by forcing the bit to 1; "reset" by forcing the bit to 0.

Unless indicated otherwise, when an instruction affects a flag, it affects it in the following manner:

Zero: If the result of an instruction has the value 0, this flag is set; otherwise it is reset.

Sign: If the most significant bit of the result of the operation has the value 1, this flag is set; otherwise it is reset.

Parity: If the modulo 2 sum of the bits of the result of the operation is 0 (i.e., if the result has even parity), this flag is set; otherwise it is reset (i.e., if the result has odd parity).

Carry: If the instruction resulted in a carry (from addition), or a borrow (from subtraction of a comparison) out of the high-order bit, this flag is set; otherwise it is reset.

Auxiliary Carry: If the instruction caused a carry out of bit 3 and into bit 4 of the resulting value, the auxiliary carry is set; otherwise it is reset. This flag is affected by single precision additions, subtractions, increments, decrements, comparisons, and logical operations, but is principally used with additions and increments preceding a DAA (Decimal Adjust Accumulator) instruction.

Symbols and Abbreviations

The following symbols and abbreviations are used in the subsequent description of the 8080 instructions:

SYMBOLS	MEANING
accumulator	Register A
addr	16-bit address quantity
data	8-bit data quantity
data 16	16-bit data quantity
byte 2	The second byte of the instruction
byte 3	The third byte of the instruction
port	8-bit address of an I/O device
r,r1,r2	One of the registers A,B,C,D,E,H,L
DDD,SSS	The bit pattern designating one of the registers A,B,C,D,E,H,L. (DD= destination, SSS=source):

DDD or SSS	REGISTER NAME
111	A
000	B
001	C
010	D
011	E
100	H
101	L

rp	One of the register pairs: B represents the B,C pair with B as the high-order register and C as the low-order register; D represents the D,E pair with D as the high-order register and E as the low-order register; H represents the H,L pair with H as the high-order register and L as the low-order register; SP represents the 16-bit stack pointer register.
RP	The bit pattern designating one of the register pairs B,D,H,SP:

RP	REGISTER PAIR
00	B-C
01	D-E
10	H-L
11	SP

rh	The first (high-order) register of a designated pair.
rl	The second (low-order) register of a designated register pair.

PC	16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits, respectively).
SP	16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits, respectively).
r_m	Bit m of the register r (bits are number 7 through 0 from left to right).
Z,S,P,CY,AC	The condition flags: Zero, Sign, Parity, Carry, and Auxiliary Carry, respectively.
( )	The contents of the memory location or registers enclosed in the parentheses.
←	"Is transferred to"A
∧	Logical AND
∨	Exclusive OR
V	Inclusive OR
+	Addition
–	Two's complement subtraction
*	Multiplication
↔	"Is exchanged with"
—	The one's complement (e.g., $(\overline{A})$)
n	The restart number 0 through 7
NNN	The binary representation 000 through 111 for restart number 0 through 7, respectively.

Description Format

The following pages provide a detailed description of the instruction set of the 8080. Each instruction is described in the following manner:

1. The MAC 80 assembler format, consisting of the instruction mnemonic and operand fields, is printed in BOLDFACE on the left side of the first line.

2. The name of the instruction is enclosed in parenthesis on the right side of the first line.

3. The next line(s) contain a symbolic description of the operation of the instruction.

4. This is followed by a narative description of the operand of the instruction.

5. The following line(s) contain the binary fields and patterns that comprise the machine instruction.

6. The last four lines contain incidental information about the execution of the instruction. The number of machine cycles and states required to execute the instruction are listed first. If the instruction has two possible execution times, as in a Conditional Jump, both times will be listed, separated by a slash. Next, any significant data addressing modes (see Page A-2) are listed. The last line lists any of the five Flags that are affected by the execution of the instruction.

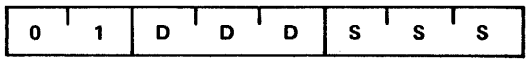
Data Transfer Group

This group of instructions transfer data to and from registers and memory. Condition flags are not affected by any instruction in this group.

MOV r1, r2 (Move Register)

(r1) ← (r2)

The content of register r2 is moved to register r1.



Cycles: 1
States: 5
Addressing: register
Flags: none

MOV r,M (Move from memory)

(r) ← ((H) (L))

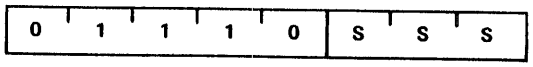
The content of the memory location, whose address is in registers H and L, is moved to register r.

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MOV M, r (Move to memory)

((H)(L)) ← (r)

The content of register r is moved to the memory location whose address is in registers H and L.

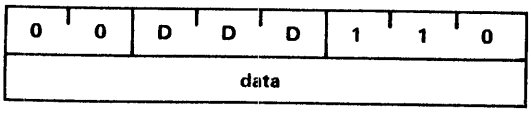


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MVI r, data (Move Immediate)

(r) ← (byte 2)

The content of byte 2 of the instruction is moved to register r.

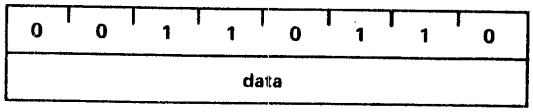


Cycles: 2
States: 7
Addressing: immediate
Flags: none

MVI M, data (Move to memory immediate)

((H)(L)) ← (byte 2)

The content of byte 2 of the instruction is moved to the memory location whose address is in registers H and L.



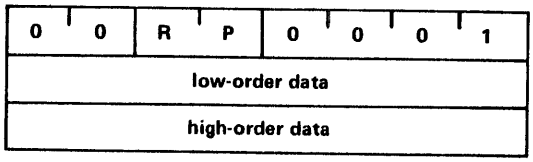
Cycles: 3
States: 10
Addressing: immed./reg. indirect
Flags: none

LXI rp, data 16 (Load register pair immediate)

(rh) ← (byte 3),

(rl) ← (byte 2)

Byte 3 of the instruction is moved into the high-order register (rh) of the register pair rp. Byte 2 of the instruction is moved into the low-order register (rl) of the register pair rp.

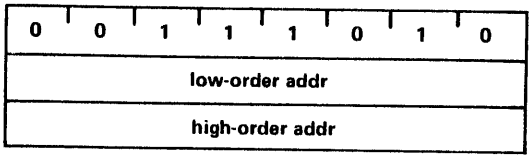


Cycles: 3
States: 10
Addressing: immediate
Flags: none

LDA addr (Load Accumulator direct)

(A) ← ((byte 3)(byte 2))

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register A.

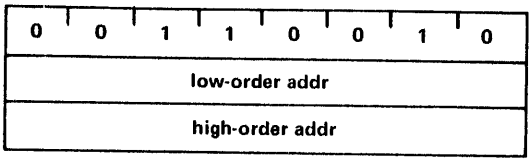


Cycles: 4
States: 13
Addressing: direct
Flags: none

STA addr (Store Accumulator direct)

((byte 3)(byte 2)) ← (A)

The content of the accumulator is moved to the memory location whose address is specified in byte 2 and byte 3 of the instruction.



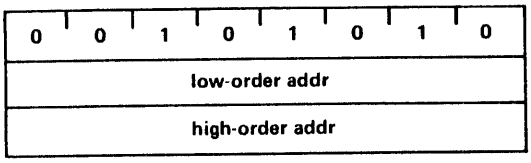
Cycles: 4
States: 13
Addressing: direct
Flags: none

LHLD addr (Load H and L direct)

(L) ← ((byte 3)(byte 2))

(H) ← ((byte 3)(byte 2) + 1)

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register L. The content of the memory location at the succeeding address is moved to register H.



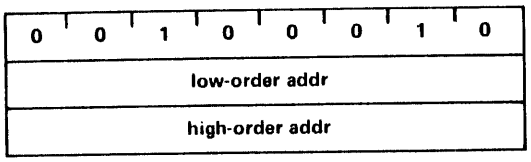
Cycles: 5
States: 16
Addressing: direct
Flags: none

SHLD addr (Store H and L direct)

((byte 3)(byte 2)) ← (L)

((byte 3)(byte 2) + 1) ← (H)

The content of register L is moved to the memory location whose address is specified in byte 2 and byte 3. The content of register H is moved to the succeeding memory location.

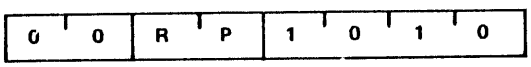


Cycles: 5
States: 16
Addressing: direct
Flags: none

LDAX rp (Load accumulator indirect)

(A) ← ((rp))

The content of the memory location, whose address is in the register pair rp, is moved to register A. Note: Only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.

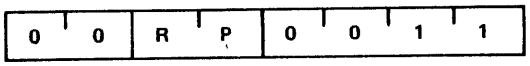


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

STAX rp (Store accumulator indirect)

((rp)) ← (A)

The content of register A is moved to the memory location whose address is in the register pair rp. Note: Only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.



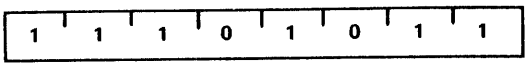
Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

XCHG (Exchange H and L with D and E)

(H) ↔ (D)

(L) ↔ (E)

The contents of registers H and L are exchanged with the contents of registers D and E.

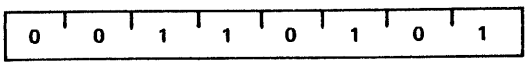


Cycles: 1
States: 4
Addressing: register
Flags: none

DCR M (Decrement memory)

((H)(L)) ← ((H)(L)) – 1

The content of the memory location whose address is contained in the H and L registers is decremented by one. Note: All condition flags except CY are affected.

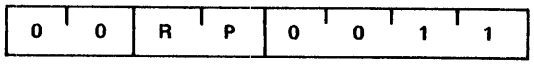


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

INX rp (Increment register pair)

(rh)(rl) ← (rh)(rl) + 1

The content of the register pair rp is incremented by one. Note: No condition flags are affected.



Cycles: 1
States: 5
Addressing: register
Flags: none

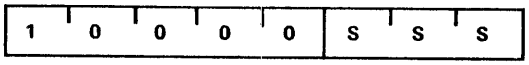
Arithmetic Group

This group of instructions performs arithmetic operations on data in registers and memory.

Unless otherwise indicated, all instructions in this group affect the Zero, Sign, Parity, Carry, and Auxiliary Carry flags according to the standard rules.

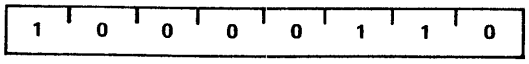
All subtraction operations are performed via two's complement arithmetic, and set the carry flag to one to indicate a borrow and clear it to indicate no borrow.

ADD r (Add Register)
(A) ← (A) + (r)
The content of register r is added to the content of the accumulator. The result is placed in the accumulator.



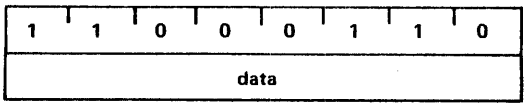
Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADD M (Add Memory)
(A) ← (A) + ((H)(L))
The content of the memory location whose address is contained in the H and L register is added to the content of the accumulator. The result is placed in the accumulator.



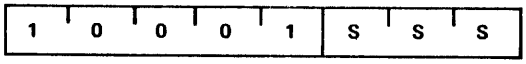
Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ADI data (Add Immediate)
(A) ← (A) + (byte 2)
The content of the second byte of the instruction is added to the constant of the accumulator. The result is placed in the accumulator.



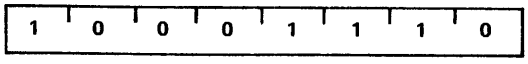
Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ADC r (Add Register with Carry)
(A) ← (A) + (r) + (CY)
The content of register r and the content of the carry bit are added to the content of the accumulator. The result is placed in the accumulator.



Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADC M (Add Memory with Carry)
(A) ← (A) + ((H)(L)) + (CY)
The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are added to the accumulator. The result is placed in the accumulator.

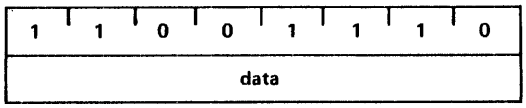


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ACI data (Add Immediate with Carry)

(A) ← (A) + (byte 2) + (CY)

The content of the second byte of the instruction and the content of the CY flag are added to the contents of the accumulator. The result is placed in the accumulator.

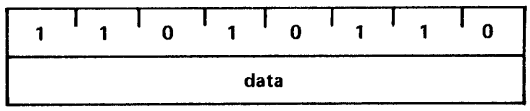


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SUI data (Subtract Immediate)

(A) ← (A) – (byte 2)

The content of the second byte of the instruction is subtracted from the content of the accumulator. The result is placed in the accumulator.

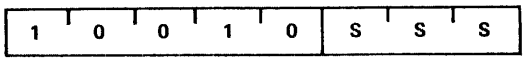


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SUB r (Subtract Register)

(A) ← (A) – (r)

The content of register r is subtracted from the content of the accumulator. The result is placed in the accumulator.

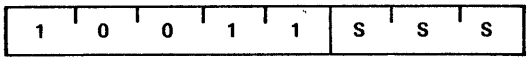


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SBB r (Subtract Register with Borrow)

(A) ← (A) – (r) – (CY)

The content of register r and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

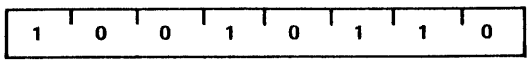


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SUB M (Subtract Memory)

(A) ← (A) – ((H)(L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the content of the accumulator. The result is placed in the accumulator.

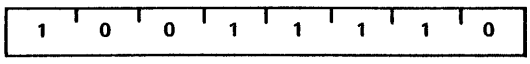


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SBB M (Subtract Memory with Borrow)

(A) ← (A) – ((H)(L)) – (CY)

The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

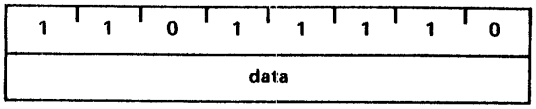


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SBI data (Subtract Immediate with Borrow)

(A) ← (A) – (byte 2) – (CY)

The contents of the second byte of the instruction and the contents of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

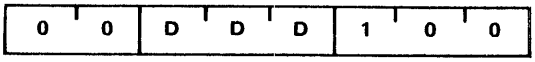


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

INR r (Increment Register)

(r) ← (r) + 1

The content of register r is incremented by one. Note: All condition flags except CY are affected.

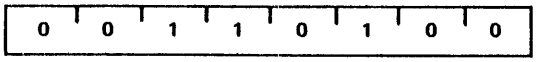


Cycles: 1
States: 5
Addressing: register
Flags: Z,S,P,AC

INR M (Increment Memory)

((H)(L)) ← ((H)(L)) + 1

The content of the memory location whose address is contained in the H and L registers is incremented by one. Note: All condition flags except CY are affected.

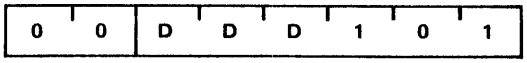


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

DCR r (Decrement Register)

(r) ← (r) – 1

The content of register r is decremented by one. Note: All condition flags except CY are affected.

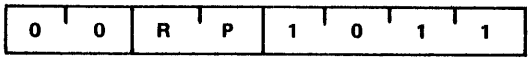


Cycles: 1
States: 5
Addressing: register
Flags: Z,S,P,AC

DCX rp (Decrement register pair)

(rh)(rl) ← (rh)(rl) – 1

The content of the register pair rp is decremented by one. Note: No condition flags are affected.

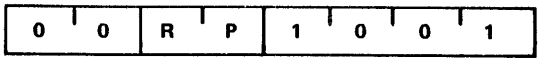


Cycles: 1
States: 5
Addressing: register
Flags: none

DAD rp (Add register pair to H and L)

(H)(L) ← (H)(L) + (rh)(rl)

The content of the register pair rp is added to the content of the register pair H and L. The result is placed in the register pair H and L. Note: Only the CY flag is affected. It is set if there is a carry out of the double precision add; otherwise it is reset.



Cycles: 3
States: 10
Addressing: register
Flags: CY

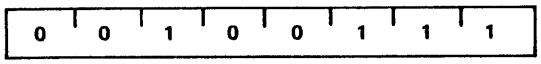
DAA (Decimal Adjust Accumulator)

The 8-bit number in the accumulator is adjusted to form two 4-bit Binary-Coded-Decimal digits by the following process:

1. If the value of the least significant 4 bits of the accumulator is greater than 9 or if the AC flag is set, 6 is added to the accumulator.

2. If the value of the most significant 4 bits of the accumulator is now greater than 9, or if the CY flag is set, 6 is added to the most significant 4 bits of the accumulator.

NOTE: All flags are affected.



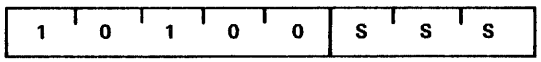
Cycles: 1
States: 4
Flags: Z,S,P,CY,AC

Logical Group

This group of instructions performs logical (Boolean) operations on data in registers and memory and on condition flags.

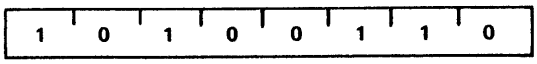
Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Auxiliary Carry, and Carry flags according to the standard rules.

ANA r (AND Register)
(A) ← (A) ∧ (r)
The content of register r is logically ANDed with the content of the accumulator. The result is placed in the accumulator. The CY flag is cleared.



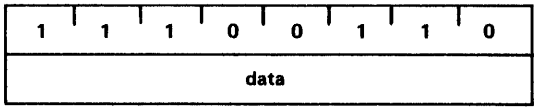
Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ANA M (AND memory)
(A) ← (A) ∧ ((H)(L))
The contents of the memory location whose address is contained in the H and L registers is logically ANDed with the content of the accumulator. The CY flag is cleared.



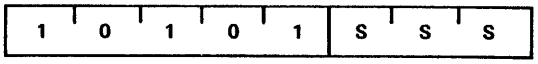
Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ANI data (AND immediate)
(A) ← (A) ∧ (byte 2)
The content of the second byte of the instruction is logically ANDed with the contents of the accumulator. The result is placed in the accumulator. The CY flag is cleared.



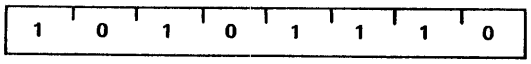
Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

XRA r (Exclusive OR Register)
(A) ← (A) ∀ (r)
The content of register r is exclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.



Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

XRA M (Exclusive OR Memory)
(A) ← (A) ∀ ((H)(L))
The content of the memory location whose address is contained in the H and L registers is exclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

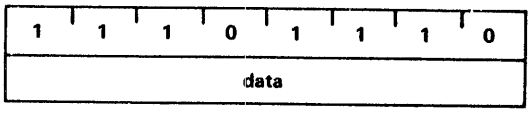


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

XRI data (Exclusive OR immediate)

(A) ← (A) V (byte 2)

The content of the second byte of the instruction is exclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

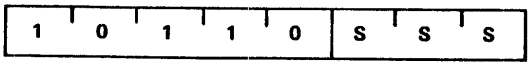


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ORA r (OR Register)

(A) ← (A) V (r)

The content of register r is inclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

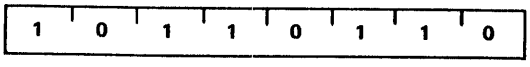


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ORA M (OR Memory)

(A) ← (A) V ((H)(L))

The content of the memory location whose address is contained in the H and L registers is inclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

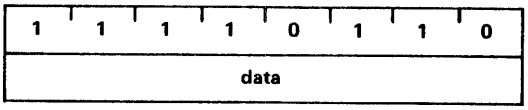


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ORI data (OR Immediate)

(A) ← (A) V (byte 2)

The content of the second byte of the instruction is inclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

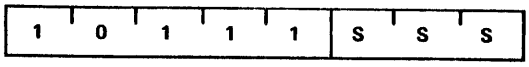


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

CMP r (Compare Register)

(A) – (r)

The content of register r is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if (A) = (r). The CY flag is set to 1 if (A) < (r).

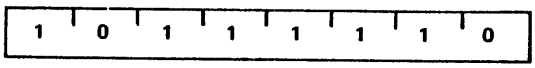


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

CMP M (Compare memory)

(A) – ((H)(L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 of (A) = ((H)(L)). The CY flag is set to 1 if (A) < ((H)(L)).

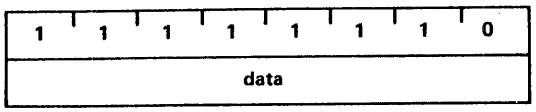


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S.P,CY,AC

CPI data (Compare immediate)

(A) – (byte 2)

The content of the second byte of the instruction is subtracted from the accumulator. The condition flags are set by the result of the subtraction. The Z flag is set to 1 if (A) = (byte 2). The CY flag is set to 1 if (A) < (byte 2).



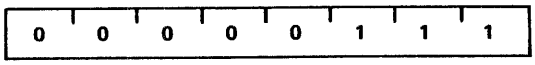
Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

RLC (Rotate left)

$(A_{n+1}) \leftarrow (A_n)$; $(A_0) \leftarrow (A_7)$

$(CY) \leftarrow (A_7)$

The content of the accumulator is rotated left one position. The low-order bits and the CY flag are both set to the value shifted out of the high-order bit position. Only the CY flag is affected.



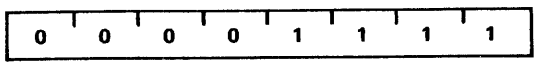
Cycles: 1
States: 4
Flags: CY

RRC (Rotate right)

$(A_n) \leftarrow (A_{n-1})$; $(A_7) \leftarrow (A_0)$

$(CY) \leftarrow (A_0)$

The content of the accumulator is rotated right one position. The high-order bit and the CY flag are both set to the value shifted out of the low-order bit position. Only the CY flag is affected.



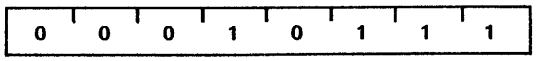
Cycles: 1
States: 4
Flags: CY

RAL (Rotate left through carry)

$(A_{n+1}) \leftarrow (A_n)$; $(CY) \leftarrow (A_7)$

$(A_0) \leftarrow (CY)$

The content of the accumulator is rotated left one position through the CY flag. The low-order bit is set equal to the CY flag and the CY flag is set to the value shifted out of the high-order bit. Only the CY flag is affected.



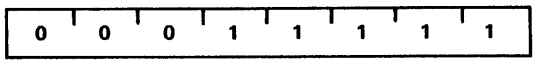
Cycles: 1
States: 4
Flags: CY

RAR (Rotate right through carry)

$(A_n) \leftarrow (A_{n+1})$; $(CY) \leftarrow (A_0)$

$(A_7) \leftarrow (CY)$

The content of the accumulator is rotated right one position through the CY flag. The high-order bit is set to the CY flag and the CY flag is set to the value shifted out of the low-order bit. Only the CY flag is affected.

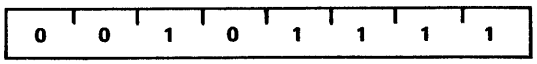


Cycles: 1
States: 4
Flags: CY

CMA (Complement accumulator)

(A) ← (A)

The contents of the accumulator are complemented (zero bits become 1, one bits become 0). No flags are affected.

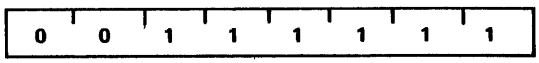


Cycles: 1
States: 4
Flags: none

CMC (Complement carry)

(CY) ← (CY)

The CY flag is complemented. No other flags are affected.

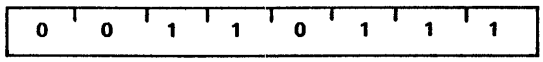


Cycles: 1
States: 4
Flags: CY

STC (Set carry)

(CY) ← 1

The CY flag is set to 1. No other flags are affected.



Cycles: 1
States: 4
Flags: CY

Branch Group

This group of instructions alter normal sequential program flow.

Condition flags are not affected by an instruction in this group.

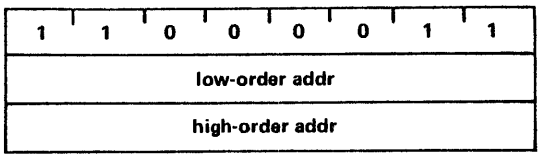
The two types of branch instructions are unconditional and conditional. Unconditional transfers simply perform the specified operation on register PC (the program counter). Conditional transfers examine the status of one of the four processor flags to determine if the specified branch is to be executed. The conditions that may be specified are as follows:

CONDITION		CCC
NZ	– not zero (Z=0)	000
Z	– zero (Z = 1)	001
NC	– no carry (C = 0)	010
C	– carry (CY = 1)	011
PO	– parity odd (P = 0)	100
PE	– parity even (P = 1)	101
P	– plus (S = 0)	110
M	– minus (S = 1)	111

JMP addr (Jump)

(PC) → (byte 3)(byte 2)

Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.



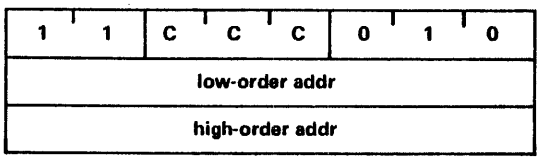
Cycles: 3
States: 10
Addressing: immediate
Flags: none

Jcondition addr (Conditional jump)

If (CCC),

(PC) ← (byte 3)(byte 2)

If the specified condition is true, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise, control continues sequentially.



Cycles: 3
States: 10
Addressing: immediate
Flags: none

CALL addr (Call)

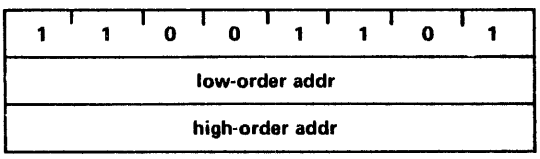
((SP) – 1) ← (PCH)

((SP) – 2) ← (PCL)

(SP) ← (SP) – 2

(PC) ← (byte 3)(byte 2)

The high-order 8 bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order 8 bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

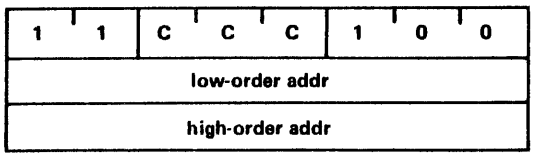


Cycles: 5
States: 17
Addressing: immed./reg. indirect
Flags: none

Ccondition addr (Condition call)

If (CCC),
((SP) – 1) ← (PCH)
((SP) – 2) ← (PCL)
(SP) ← (SP) – 2
(PC) ← (byte 3)(byte 2)

If the specified condition is true, the actions specified in the CALL instruction (see above) are performed; otherwise, control continues sequentially.

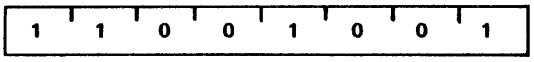


Cycles: 3/5
States: 11/17
Addressing: immed./reg. indirect
Flags: none

RET (Return)

(PCL) ← ((SP));
(PCH) ← ((SP) + 1);
(SP) ← (SP) + 2;

The content of the memory location whose address is specified in register SP is moved to the low-order 8 bits of register PC. The content of the memory location whose address is one more than the content of register SP is moved to the high-order 8 bits of register PC. The content of register SP is incremented by 2.

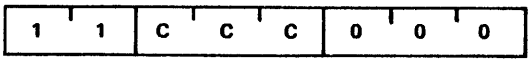


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: none

Rcondition (Conditional return)

If (CCC),
(PCL) ← ((SP))
(PCH) ← ((SP) + 1)
(SP) ← (SP) + 2

If the specified condition is true, the actions specified in the RET instruction (see above) are performed; otherwise, control continues sequentially.

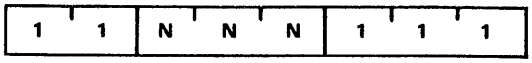


Cycles: 1/3
States: 5/11
Addressing: reg. indirect
Flags: none

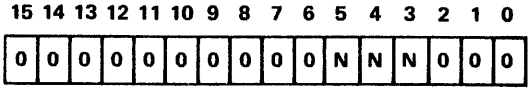
RST n (Restart)

((SP) – 1) ← (PCH)
((SP) – 2) ← (PCL)
(SP) ← (SP) – 2
(PC) ← 8 * (NNN)

The high-order 8 bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order 8 bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. Control is transferred to the instruction whose address is eight times the content of NNN.



Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

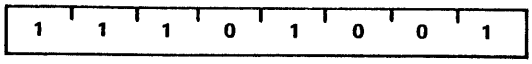


Program Counter After Restart

PCHL (Jump H and L indirect – move H and L to PC)

(PCH) ← (H)
(PCL) ← (L)

The content of register H is moved to the high-order 8 bits of register PC. The content of register L is moved to the low-order 8 bits of register PC.



Cycles: 1
States: 5
Addressing: register
Flags: none

Stack, I/O, and Machine Control Group

This group of instructions performs I/O, manipulates the Stack, and alters internal control flags.

Unless otherwise specified, condition flags are not affected by any instructions in this group.

FLAG WORD

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z	0	AC	0	P	1	CY

PUSH rp (Push)

((SP) – 1) ← (rh)
((SP) – 2) ← (rl)
(SP) ← (SP) – 2

The content of the high-order register of register pair rp is moved to the memory location whose address is one less than the content of register SP. The content of the low-order register of register pair rp is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Note: Register pair rp=SP may not be specified.

1	1	R	P	0	1	0	1

Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

POP rp (Pop)

(rl) ← ((SP))
(rh) ← ((SP) + 1)
(SP) ← (SP) + 2

The content of the memory location, whose address is specified by the content of register SP, is moved to the low-order register of register pair rp. The content of the memory location, whose address is one more than the content of register SP, is moved to the high-order register of register pair rp. The content of register SP is incremented by 2. Note: Register pair rp=SP may not be specified.

1	1	R	P	0	0	0	1

Cycles: 3
States: 10
Addressing: reg. indirect.
Flags: none

PUSH PSW (Push processor status word)

((SP) – 1) ← (A)
$((SP) - 2)_0$ ← (CY, $((SP) - 2)_1$ ← 1
$((SP) - 2)_2$ ← (P), $((SP) - 2)_3$ ← 0
$((SP) - 2)_4$ ← (AC), $((SP) - 2)_5$ ← 0
$((SP) - 2)_6$ ← (Z), $((SP) - 2)_7$ ← (S)
(SP) ← (SP) – 2

The content of register A is moved to the memory location whose address is one less than register SP. The contents of the condition flags are assembled into a processor status word and the word is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two.

1	1	1	1	0	1	0	1

Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

POP PSW (Pop processor status word)

(CY) ← $((SP))_0$
(P) ← $((SP))_2$
(AC) ← $((SP))_4$
(Z) ← $((SP))_6$
(S) ← $((SP))_7$
(A) ← ((SP) + 1)
(SP) ← (SP) + 2

The content of the memory location whose address is specified by the content of register SP is used to restore the condition flags. The content of the memory location whose address is one more than the content of register SP is moved to register A. The content of register SP is incremented by 2.

1	1	1	1	0	0	0	1

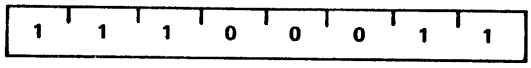
Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

XTHL (Exchange stack top with H and L)

(L) ↔ ((SP))

(H) ↔ ((SP) + 1)

The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP.

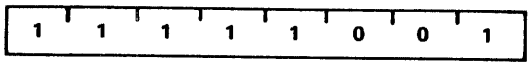


Cycles: 5
States: 18
Addressing: reg. indirect
Flags: none

SPHL (Move HL to SP)

(SP) ← (H)(L)

The contents of registers H and L (16 bits) are moved to register SP.

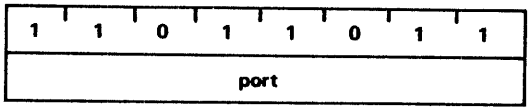


Cycles: 1
States: 5
Addressing: register
Flags: none

IN port (Input)

(A) ← (data)

The data placed on the 8-bit bidirectional data bus by the specified port is moved to register A.

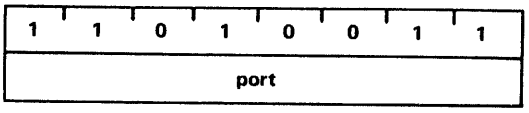


Cycles: 3
States: 10
Addressing: direct
Flags: none

OUT port (Output)

(data) ← (A)

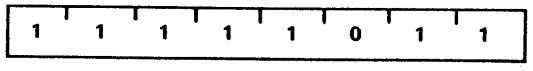
The content of register A is placed on the 8-bit bidirectional data bus for transmission to the specified port.



Cycles: 3
States: 10
Addressing: direct
Flags: none

EI (Enable interrupt)

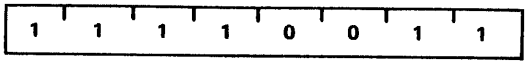
The interrupt system is enabled following the execution of the next instruction.



Cycles: 1
States: 4
Flags: none

DI (Disable interrupts)

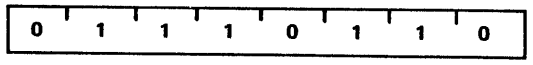
The interrupt system is disabled immediately following the execution of the DI instruction.



Cycles: 1
States: 4
Flags: none

HLT (Halt)

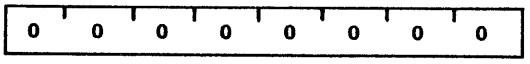
The processor is stopped. The registers and flags are unaffected.



Cycles: 1
States: 7
Flags: none

NOP (No op)

No operation is performed. The registers and flags are unaffected.



Cycles: 1
States: 4
Flags: none

INSTRUCTION SET

Summary of Processor Instructions

MNEMONIC	DESCRIPTION	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	CLOCK[2] CYCLES
$MOV_{r1,r2}$	Move register to register	0	1	D	D	D	S	S	S	5
MOV M,r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r,M	Move memory to register	0	1	D	D	D	1	1	0	7
HLT	Halt	0	1	1	1	0	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
INR r	Increment register	0	0	D	D	D	1	0	0	5
DCR r	Decrement register	0	0	D	D	D	1	0	1	5
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	10
JZ	Jump on zero	1	1	0	0	1	0	1	0	10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	10
JP	Jump on positive	1	1	1	1	0	0	1	0	10
JM	Jump on minus	1	1	1	1	1	0	1	0	10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	10
CALL	Call unconditional	1	1	0	0	1	1	0	1	17
CC	Call on carry	1	1	0	1	1	1	0	0	11/17
CNC	Call on no carry	1	1	0	1	0	1	0	0	11/17
CZ	Call on zero	1	1	0	0	1	1	0	0	11/17
CNZ	Call on no zero	1	1	0	0	0	1	0	0	11/17
CP	Call on positive	1	1	1	1	0	1	0	0	11/17
CM	Call on minus	1	1	1	1	1	1	0	0	11/17
CPE	Call on parity even	1	1	1	0	1	1	0	0	11/17
CPO	Call on parity odd	1	1	1	0	0	1	0	0	11/17
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	5/11
RNC	Return on no carry	1	1	0	1	0	0	0	0	5/11
RZ	Return on zero	1	1	0	0	1	0	0	0	5/11
RNZ	Return on no zero	1	1	0	0	0	0	0	0	5/11
RP	Return on positive	1	1	1	1	0	0	0	0	5/11
RM	Return on minus	1	1	1	1	1	0	0	0	5/11
RPE	Return on parity even	1	1	1	0	1	0	0	0	5/11
RPO	Return on parity odd	1	1	1	0	0	0	0	0	5/11
RST	Restart	1	1	A	A	A	1	1	1	11
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	11
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	11
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	11
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	11
POP B	Pop register pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
XTHL	Exchange top of stock H & L	1	1	1	0	0	0	1	1	18
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	5
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	5
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	5
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	5
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	5
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	5
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	5
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	5
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4

NOTES: 1. DDD or SSS - 000 B - 001 C - 010 D - 011 E - 100 H - 101 L - 110 Memory - 111 A.
2. Two possible cycle times, (5/11) indicate instruction cycles dependent on condition flags.

APPENDIX C

SBC-901, SBC-902 SCHEMATICS

Schematic drawings for the SBC-901 and SBC-902 are provided in this appendix. Information and diagrams in this section are subject to change without notice. References should be made to schematics shipped with this module.

RESISTOR NETWORK SPECIFICATIONS:

RESISTANCE VALUES:
±2% (MAX)

OPERATING TEMPERATURE:
0°C TO +70°C

TEMPERATURE COEFFICIENT:
±200 PPM/°C OVER TEMPERATURE RANGE OF 0°C TO +70°C

OPERATING VOLTAGE:
6.0 VDC (MAX)

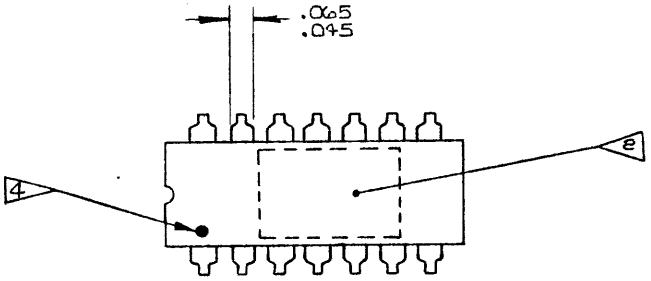
POWER RATING:
AT 70°C, 0.7 WATT PER PACK

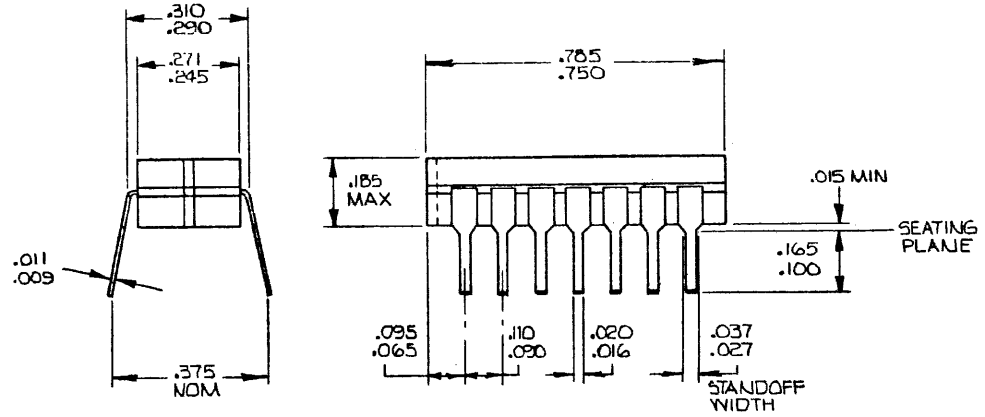
TRACKING RESISTANCE RATIO:
±1.0% (MAX)

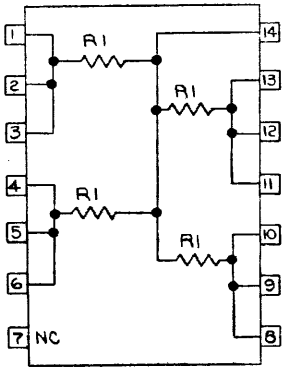
STABILITY:
±1% YEAR (MAX)

LOAD LIFE:
±1% (ΔR) OVER 1000 HOURS

PACKAGE:
DUAL IN LINE - CERAMIC OR PLASTIC







C-2

NOTES:
UNLESS OTHERWISE SPECIFIED,

1. PART NO IS 4500645-01.

2. INK STAMP PRODUCT CODE, RESISTOR VALUE, PART NO AND DASH NUMBER WITH CONTRASTING COLOR AND MIN .12 HIGH CHARACTERS. NO OTHER MARKINGS ARE PERMITTED EXCEPT FOR MANUF. BATCH NO.

E.G.) SBC-902
R 1K
4500645-01

3. FOR PROCUREMENT SEE LV4500645

4. IDENTIFY PIN ONE CLEARLY ON TOP OF PACKAGE.

DESCRIPTION			
PARTS LIST			
intel® 3065 BOWERS AVE. SANTA CLARA CALIF. 95051			
TITLE TERMINATING PACK PULL UP			
SIZE	DEPT	DRAWING NO.	REV
C	410	4500645	B

RESISTOR NETWORK SPECIFICATIONS:

RESISTANCE VALUES:
±2% (MAX)

OPERATING TEMPERATURE:
0°C TO +70°C

TEMPERATURE COEFFICIENT:
±200 PPM/°C OVER TEMPERATURE RANGE OF 0°C TO +70°C

OPERATING VOLTAGE:
6.0 VDC (MAX)

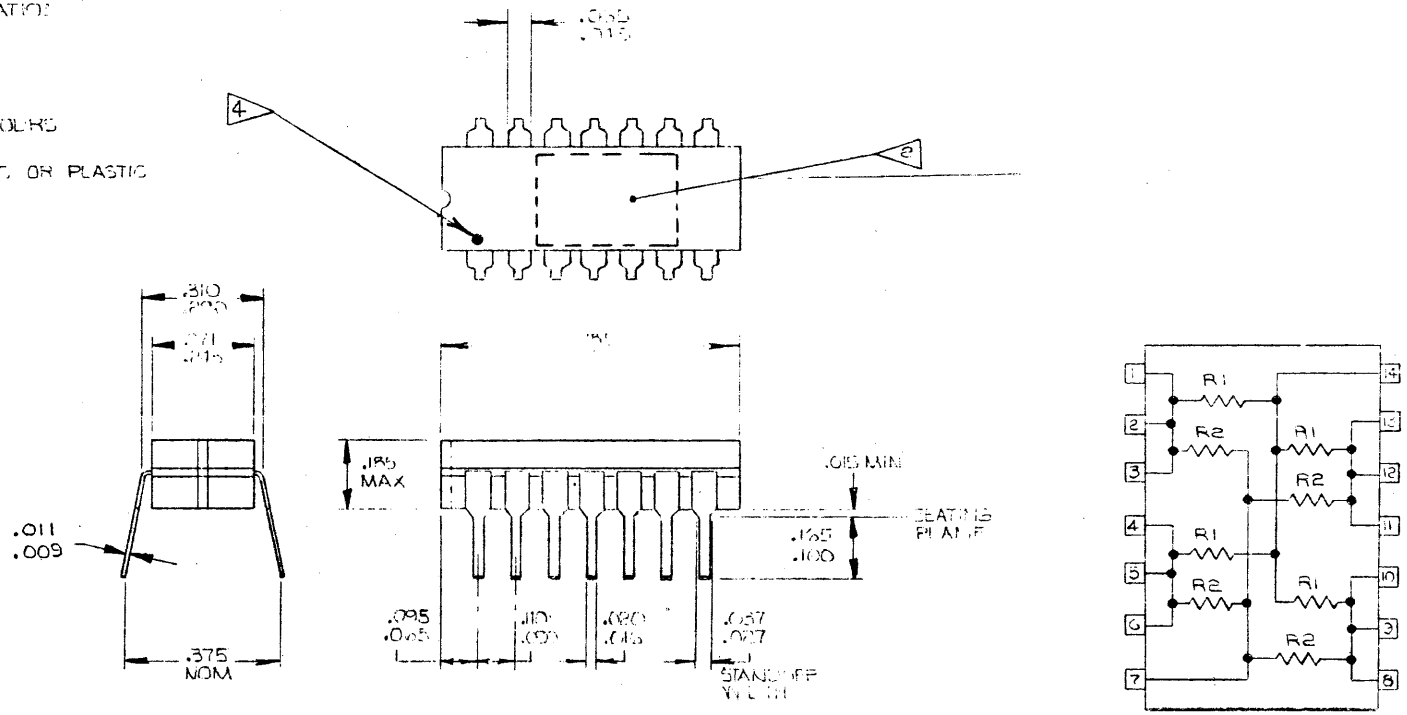
POWER RATING:
AT 70°C, 0.7 WATT PER PACK

TRACKING RESISTANCE RATIO:
±1.0% (MAX)

STABILITY:
±1% YEAR (MAX)

LOAD LIFE:
±1% (ΔR) OVER 1000 HOURS

PACKAGE:
DUAL IN-LINE - CERAMIC OR PLASTIC



NOTES:
UNLESS OTHERWISE SPECIFIED,

1. PART NO IS 4500644-01.

2. INK STAMP PRODUCT CODE, RESIST R VALUE, PART NO, AND DASH NUMBER WITH CONTRASTING COLOR INK USING MIN .05 HIGH CHARACTERS. NO OTHER MARKINGS PERMITTED EXCEPT MANUF BATCH NO.

   E.G.) LBC-301
   R220/330
   4500644-01

3. FOR PROCUREMENT SEE LV 4500644-01.

4. IDENTIFY PIN ONE CLEARLY ON TOP OF PACKAGE.

C-3

DESCRIPTION			
PARTS LIST			
intel® 3065 BOWERS AVE. SANTA CLARA CALIF. 95051			
TITLE: TERMINATING PACK PULL UP / PULL DOWN			
SIZE	DEPT	DRAWING NO.	REV
C	410	4500644	B

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, CA 95051

Printed in U.S.A. MCS 194-0676/10K