VME-68K20
Hardware Reference Manual
PRELIMINARY

INTEGRATED SOLUTIONS
1140 Ringwood Court
San Jose, California 95131
(408) 943-1902

490103 Rev. A

January 1986 Preliminary

PREFACE

This manual describes the Integrated Solutions VME-68K20 processor board. The text provided in this manual includes a product overview, specifications, configuration information, and programming information. The manual is divided into four sections and two appendixes:

SECTION 1: This section describes the general features and architecture of the VME-68K20.

SECTION 2: This section lists the VME-68K20 specifications.

SECTION 3: This section provides the VME-68K20 switch and jumper configuration options.

SECTION 4: This section provides VME-68K20 programming and operations information.

APPENDIX A: This appendix provides VME-68K20 compatibility information.

APPENDIX B: This appendix provides application notes for the Signetic 2681 DUART.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1: INTRODUCTION

The VME-68K20 CPU board is a high-performance systems-oriented CPU on a standard dual-size VME board. Maximizing the performance available from the Motorola MC68020, the VME-68K20 combines the processing speed, large instruction space, and powerful instruction set of the MC68020 processor with the second-generation industry standard VME bus.

Integrated into the VME-68K20 board is the concept of a dual-bus architecture for maximum performance in both single processor and multiprocessor environments. The VME-68K20 also provides the additional features necessary to support multiple processors on a single VME bus.

1.1 Features

Using the latest in VLSI technology, the VME-68K20 offers the following features:

- 16.67 MHz 68020 CPU
- Dual 32-bit buses (VME bus and HSMEM bus)
- VME bus four-level arbiter
- HSMEM bus arbiter
- Support for interprocessor interrupts
- Memory Management Unit featuring an ultra-high-speed translation buffer
- Supports 256-Mbyte virtual address space and 256-Mbyte physical address space
- Support for an MC68881 floating point processor with independent clock
- Two asynchronous serial ports

1.2 Performance

The VME-68K20 maximizes the performance available from the Motorola MC68020 processor. Several architectural features contribute to this optimized performance:

- Use of the HSMEM bus — Maximum performance is realized when all memory is present on the HSMEM bus, and the VME bus is used strictly for I/O operations. At 16.67 MHz the 68020 operates with one wait state on reads and no wait states on writes. In addition, no VME bus cycles occur when the 68020 operates over the HSMEM bus, leaving the entire VME bus bandwidth available for peripheral operations.
- Ultra-high-speed translation buffer — The 68020 processor board uses a very high-speed single level translation buffer that translates virtual to physical addresses in less than 35 nanoseconds (ns).
- The Motorola MC68020 is a full 32-bit processor with 32-bit registers, data paths, and addressing. The processor's design enables the VME-68K20 to approach maximum performance without exotic hardware. The MC68020 always executes 32-bit instruction fetches resulting in almost a 50 percent decrease in the number of fetches required by 16-bit microprocessors. In addition, the MC68020 has a small internal instruction cache which further reduces the required number of external instruction fetches. The combination of these factors means that the one wait state on reads and no wait states on writes version of the 68020 operating at 16.67 MHz achieves between 90 and 92 percent of the maximum no wait state performance possible.

1.3 Architecture

As illustrated in Figure 1-1, the major functional elements that compose the VME-68K20 are

- MC68020 processor
- VME bus and HSMEM bus interface logic
- Memory Management Unit
- Serial ports
- EPROM
- MC68881 Floating point processor

1.3.1 MC68020 Processor

The VME-68K20 uses the Motorola MC68020 processor as the CPU. The MC68020 is a 32-bit processor that offers virtual memory support, 18 addressing modes, a flexible co-processor interface, floating point processor support, memory mapped I/O, internal instruction cache, and pipeline instruction execution. As discussed previously, the processor's advanced design enables the VME-68K20 to approach maximum performance without the need for exotic support hardware.

1.3.2 Bus Interfaces

The VME-68K20 supports two buses, the VME bus and the HSMEM bus. The VME-68K20 dual-bus-architecture provides maximum performance in both single processor and multiprocessor environments.

1.3.2.1 VME Bus

The VME-68K20 has a 32-bit VME interface. The interface supports byte, word, and longword transfers. In 28-bit mode, the VME-68K20 conforms to VME 32-bit addressing with the top four address bits driven to 0. The VME bus interface includes an interrupt handler for all seven interrupt levels and a four-level prioritized bus arbiter.

Interrupt Handler

The VME-68K20 can receive interrupts at any of the seven VME bus levels. In addition, any level can be independently disabled so that the 68020 does not respond to it. This is useful in multiprocessor environments where each processor handles a portion of the seven interrupt levels.

Support for interprocessor interrupts is required for multiprocessing. The VME-68K20 provides an on-board I/O port into which the 68020 can write to generate an interrupt at any of the seven VME bus levels. A programmable vector will then be driven onto the bus during the VME bus interrupt acknowledge cycle.

VME Bus Arbiter

The arbiter can be configured as a Release on Request or a Release When Done arbiter, and the 68020 processor can be configured to request the bus at any one of the four levels. Additionally, in multiprocessor environments where another processor is handling bus arbitration, the arbiter can be completely disabled.

1.3.2.2 HSMEM Bus

The HSMEM bus is a 32-bit bus implemented on the user defined pins of the VME-68K20 P2 connector in conjunction with connector J2, which supports HSMEM bus address Bits 27–24. The HSMEM bus provides access to a maximum of 256 Mbytes of physical memory with one wait state on reads and no wait states on writes at 16.67 MHz. No VME bus cycles occur when the 68020 operates over the HSMEM bus. The HSMEM bus signal lines are provided in Table 1-1.

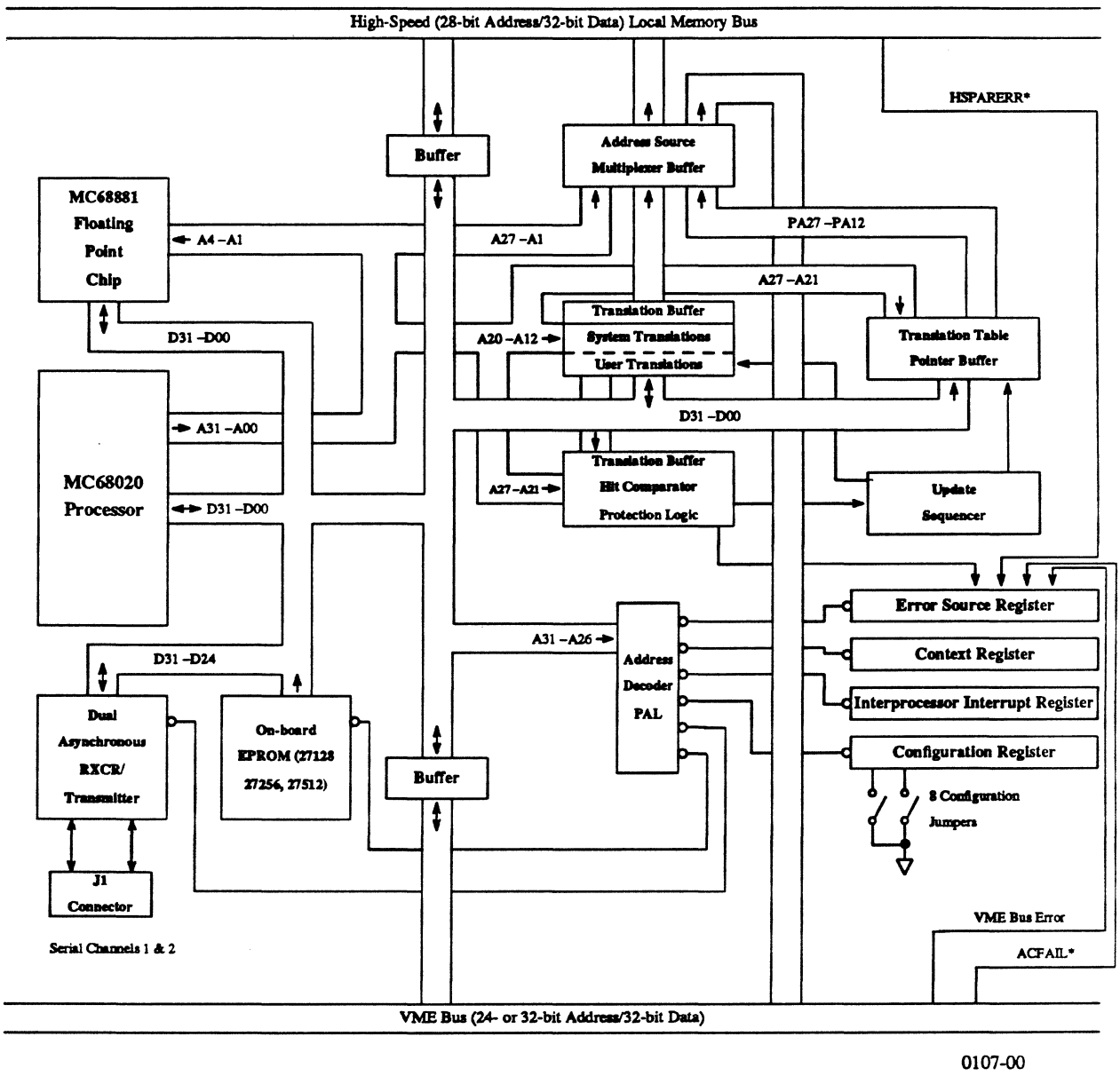


Figure 1-1. VME-68K20 Block Diagram

Table 1-1. HSMEM Bus Signals

27	Address Lines:	HSADD27–HSADD1
32	Data Lines:	HSDAT31–HSDAT0
9	Control Lines:	
	HSCYC*	Used to start an HSMEM cycle
	HSUDS* HSLDS*	Used as upper and lower data strobes on the HSMEM bus
	HSRW*	The read/write control line
	HSLW*	Used to signal 32-bit transfers
	HSREF*	Used to signal the start of a refresh cycle
	HSVME*	Used to signal the start of a VME access cycle
	HSPARERR*	The parity error signal returned by the memory board to the CPU if parity error occurs
	HSDTACK*	Used as a cache hit signal to eliminate the one wait state normally inserted on all high-speed memory cycles.

NOTE

The use of an asterisk (*) following the signal name indicates that the signal is true when it is low.

HSMEM Bus Arbiter

The VME-68K20 is designed around the concept of sharing HSMEM bus bandwidth (total bandwidth 16.67 Mbytes/second) among the 68020 processor, the VME bus DMA requests, and the memory refresh circuitry. Few DMA devices can use the entire 16.67 Mbytes/second bandwidth of the HSMEM bus. Therefore, the HSMEM bus arbiter gives DMA devices highest priority, i.e., whatever percentage of the memory cycles they need, and allocates the rest of the cycles to the 68020. The memory refresh circuitry utilizes one memory cycle every 16 microseconds. This is a much more efficient strategy than other bus arbitration schemes where the DMA device retains control of the bus for a period of time, although it cannot perform transfers at a rate even approaching the bus bandwidth. Bus arbitration occurs every 60 ns. In order to avoid synchronizing delays in the 68020 access path, the entire arbiter operates synchronously with the 68020 clock.

1.3.3 Memory Management Unit

The VME-68K20 has a demand paged Memory Management Unit (MMU) that supports a 256-Mbyte virtual address space and a 256-Mbyte physical address space. The VME-68K20 MMU performs three functions that are necessary for efficient multiprogramming operating systems:

- Translates from the 256-Mbyte virtual address space to the 256-Mbyte physical address space — The VME-68K20 demand paging subsystem is based on four-Kbyte pages. Consequently, 65,536 translations are needed to map the 256-Mbyte virtual address space. These translations are kept in system physical memory in page translation tables (PTTs). The high-speed translation buffer caches the most recently used translations. The hardware sequencer pulls required translations from the PTT area into the translation buffer without software intervention.

- Protects one user program in a multiprogramming environment from corrupting another user program or the operating system itself — The memory management unit has two bits in the PTT entry (PTTE) which are protection bits. These protection bits can be encoded for no access, read only access, or read/write access. If a user program attempts to execute a memory access that is not permitted by the protection bits, the MMU prevents the cycle from occurring and a bus error notifies the processor of the attempted access.
- Provides demand-paged virtual memory support — In this role, the MMU supports two additional bits in the PTTE. These bits are the page accessed bit and the page modified bit. The two bits are automatically modified by the hardware under the following conditions:
  - — Both bits should initially be 0 when the PTTE is created in main memory by the operating system. This indicates that the user program has not modified the page from what was present on the disk. It also indicates that the page has not yet been accessed by the user program.
  - — When a page is loaded into the translation buffer by the hardware sequencer, the page accessed and page modified bits are checked. If the page accessed bit is 0, it is updated to 1 by the hardware as is the memory-based PTTE. (Central to the design concept of the VME-68K20 MMU is the concept that the memory-based PTTEs are always updated by the hardware in order to remain identical to the translation buffer copy of the PTTEs. Thus, any change in the translation buffer page accessed or page modified bits is immediately followed by an update cycle in which the memory-based PTTE is updated.)
  - — When a write operation occurs to a page which has a page modified bit value of 0, the page modified bit is updated to 1 both in the translation buffer and in the memory-based PTTE.

The major elements that compose the MMU include a page table pointer buffer, context register, hardware sequencer, and an ultra-high-speed translation buffer.

1.3.3.1 Page Table Pointer Buffer

The Page Table Pointer buffer (PTP buffer) contains eight contexts (0–7), each context consisting of 128 twelve-bit Page Table Pointers (PTPs), see Figure 1-2. The PTP buffer can be thought of as an MMU that is used solely by the hardware sequencer. The PTP buffer is programmed by the system software for use by the hardware sequencer and it performs two functions:

- It translates virtual addresses of PTTs into physical addresses.
- It provides an abort mechanism for references to user virtual spaces for which no PTTEs exist. This allows better overall system performance by allowing large sections of virtual address spaces to be disabled very quickly and easily.

Virtual Address Translations

As mentioned previously, the PTP buffer contains eight contexts, the system context (0) and seven user contexts (1–7). As shown in Figure 1-2, it is the function of Bits 2–0 of the Context Register to provide an index into the PTP buffer to select one of the eight contexts.

Within a particular context, one of the 128 PTPs is selected by Bits 27–21 of the system/user virtual address. Context Register Bits 2–0 plus virtual address Bits 27–21 form an index into the PTP buffer (selecting a particular PTP) as illustrated in Figure 1-3.

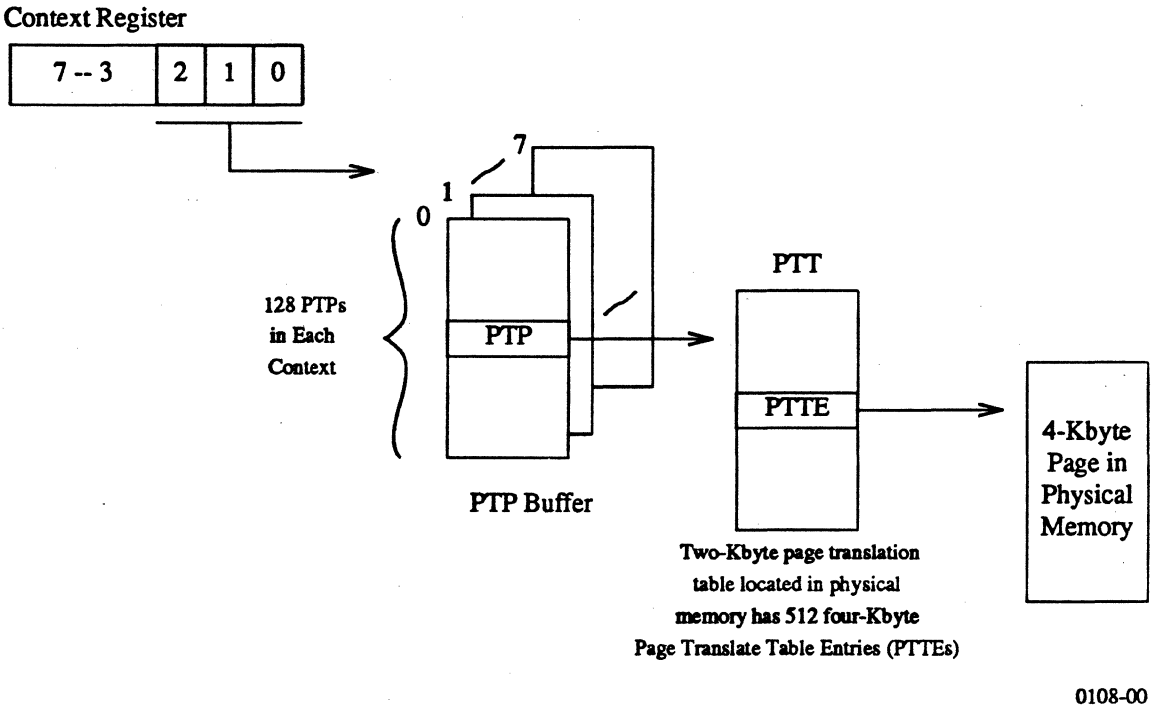


Figure 1-2. Memory Management Overview

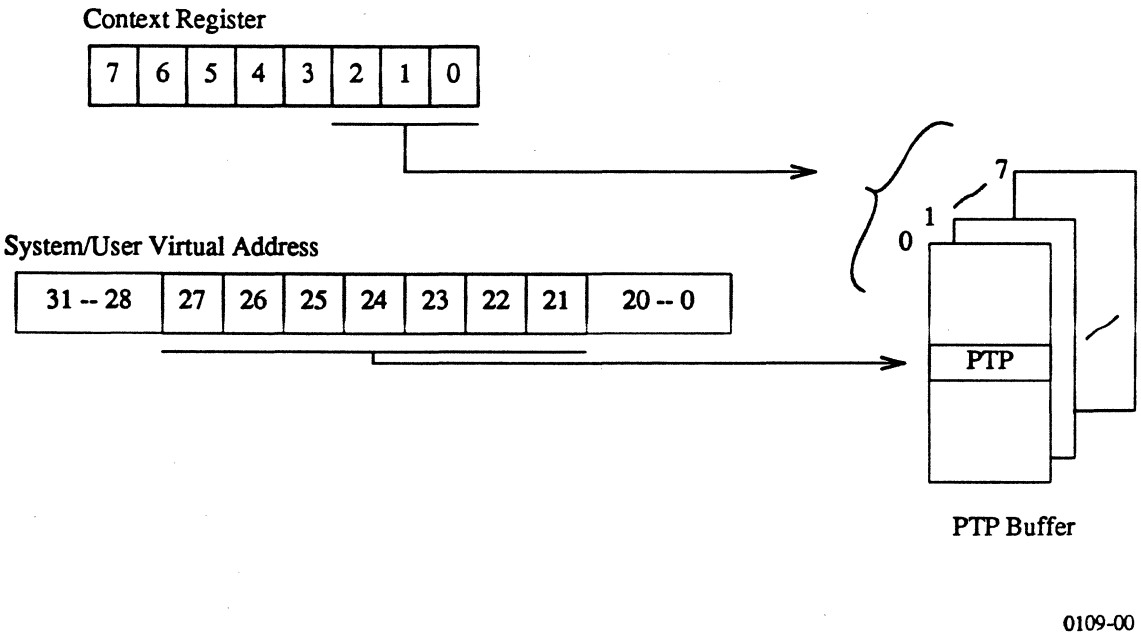


Figure 1-3. PTP Index

Each 12-bit PTP points to a two-Kbyte block in physical memory called a Page Translation Table (PTT). As shown in Figure 1-2, each PTT contains 512 four-byte Page Table Entries (PTTEs). Virtual address Bits 20-12 form an index into the PTT selecting one of the 512 PTTEs. The 12-bit PTP appended to virtual address Bits 20–12 forms the physical address of a particular PTTE as illustrated in Figure 1-4.

NOTE

PTTE physical addresses must be in the lower seven Mbytes on the HSMEM bus. The high order nine bits and low order two bits of the 32-bit physical address generated by the PTP are forced to 0 by the hardware.

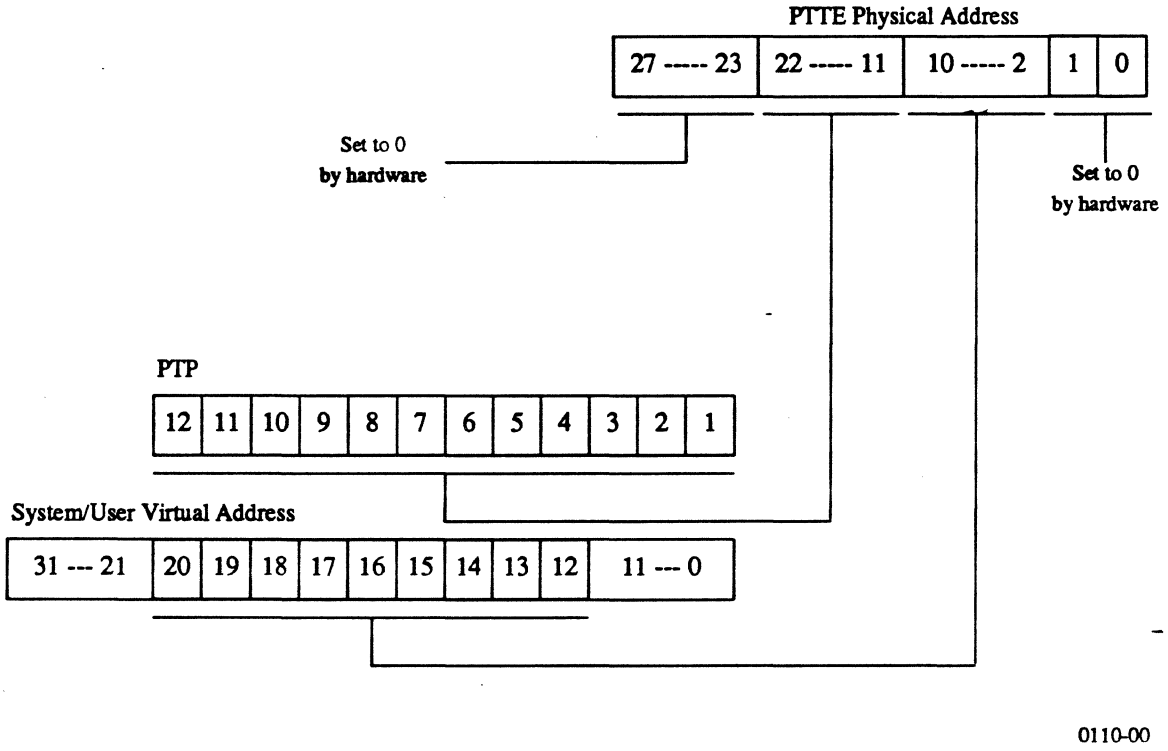


Figure 1-4. PTTE Physical Address

Each PTTE is one longword (32 bits) in length. The PTTE format is shown in Figure 1-5 and Table 1-2 provides bit definitions. PTTE Bits 15–0 specify a page number in physical memory; the complete physical address is formed by appending virtual address Bits 11–0 which provide the offset into the page number specified by PTTE Bits 15–0.

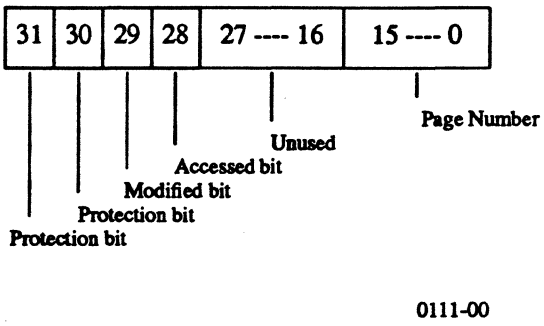


Figure 1-5. PTTE Format

Table 1-2. PTTE Bit Definitions

Bit	Name	Description
Bit 31	Protection Bit	When 0, this bit indicates that the PTTE is invalid. When 1, the bit indicates a valid entry. When a translation through an invalid entry in the translation buffer is attempted, the cycle is blocked and the cycle is terminated in a bus error as a protection violation.
Bit 30	Protection Bit	If Bit 30 is a 1, the accessed page is read only. If Bit 30 is 0, the accessed page is read/write. An attempt to write on a read only page results in a protection violation. A protection violation results in the cycle being blocked and terminated with a bus error to notify the processor of the violation.
Bit 29	Page Modified Bit	This bit is changed from 0 to 1 on the first write access to this page. The hardware sequencer is responsible for keeping the memory PTTE synchronized with the translation buffer entry.
Bit 28	Page Accessed Bit	This bit is changed from 0 to 1 on the first access to this page. This first access normally occurs the first time the page is pulled from the PTTs into the translation buffer. Again, the hardware sequencer is responsible for keeping the memory-based PTTE synchronized with the translation buffer entry.
Bits 27–16	—	Unused by hardware.
Bits 15-0	Page Number	The least significant twelve bits of the virtual address (page offset) are appended to these sixteen bits (page) to form the 28-bit physical memory address.

Figure 1-6 presents a summary view of the virtual-to-physical address translation. System/user virtual address Bits 31–28 determine which context in the PTP buffer is selected:

- 0000 is user mode and the context is Context Register Bits 2–0.
- 0001 is system mode and the context is 0 regardless of the Context Register contents.
- 0010 is system mode and the context is Context Register Bits 2–0.

The context with virtual address Bits 27–21 appended provide an index into the PTP buffer selecting a particular 12-bit PTP. The 12-bit PTP appended to virtual address Bits 20–12 produce the physical address of the PTTE in physical memory. PTTE Bits 15–0 appended to virtual address Bits 11–0 form the 28-bit physical address of the four-Kbyte page in physical memory.

NOTE

The high order four bits of the 32-bit VME address are zeroes.

Abort Mechanism

The PTP buffer performs a special function to assist the operating system. Since most user processes are considerably less than 256 Mbytes in size, it would be inefficient for the operating system to maintain all 65,536 translations necessary to translate a 256-Mbyte virtual address space. With most processes, almost all of the translations would contain the no-access protection code. Therefore, the PTP buffer provides an abort mechanism for detecting references to user virtual space for which no PTTEs exist. If all of the upper four bits of the PTP are 1s, it indicates that the PTT for this range of the virtual address space is not allocated. The hardware sequencer, upon encountering this situation, blocks the translation buffer from being updated and terminates the cycle in a bus error. The operating system, upon seeing the bus error, can determine whether the reference was a valid reference that should be paged in from disk or an illegal user access.

It is worth noting how PTPs are written and read by the processor as compared to their use by the hardware sequencer. As shown in Figure 1-7, when the processor is writing or reading a PTP, Data Bits 0–10 correspond to the 11 most significant bits (MSB) of the 12-bit PTP. The least significant bit (LSB) of the PTP corresponds to Data Bit 20. When the PTP is used to produce the physical address of the PTTE, the LSB of the PTP corresponds to the PTTE Physical Address Bit 11. The MSB of the PTP corresponds to PTTE Physical Address Bit 22.

1.3.3.2 Hardware Sequencer

The hardware sequencer provides two main functions. It fetches PTTEs from the memory-based PTTs whenever the necessary PTTE is not present in the translation buffer and it updates the contents of the memory-based PTTEs so they always agree with their cached counterparts. The hardware sequencer runs synchronously with the 68020 clock, taking five clock cycles (300 nanoseconds) to fetch a PTTE from physical memory or to update a PTTE in physical memory.

It is important to note with regard to MMU operation that in system mode it is possible to directly access physical memory without going through the translation buffer. The non-translated address select logic detects this range of virtual address which is to pass around the MMU and inhibits the MMU during these cycles.

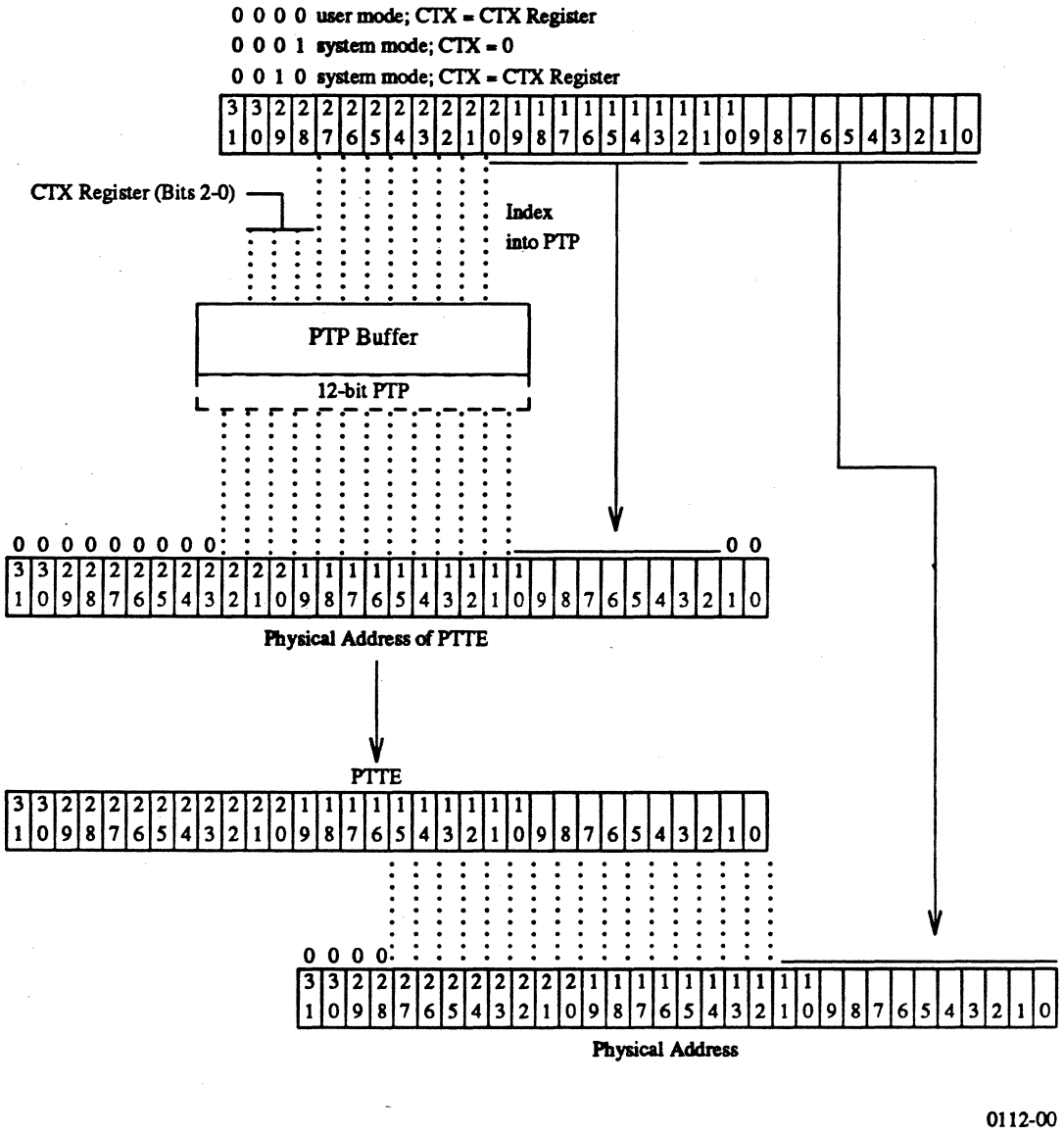


Figure 1-6. Virtual-to-Physical Address Translation Summary

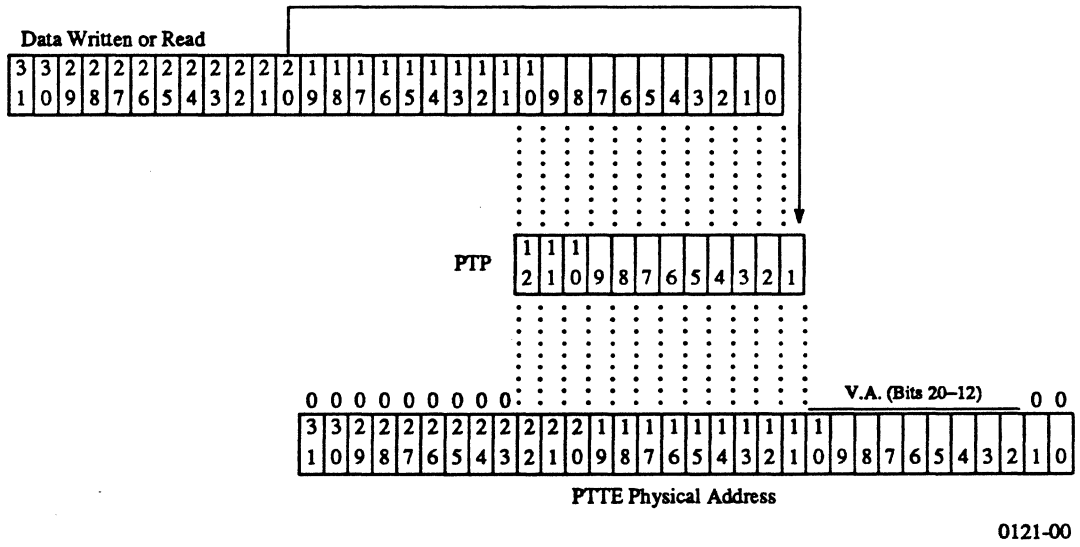


Figure 1-7. Reading and Writing PTPs

1.3.3.3 Translation Buffer

The translation buffer is made up of 1,024 by 28 bits of very high-speed static RAM. In the translation buffer, 512 entries are allocated to caching system translations and 512 entries to caching user translations. The hardware sequencer fetches the necessary PTTEs from the PTT area into the translation buffer. However, if translation buffer Bits 26–20 match virtual address Bits 27–21 and the valid bit is 1 (translation buffer Bit 31), a cache hit has occurred and no hardware sequencer assistance is required. Each entry in the translation buffer consists of 28 bits as shown in Figure 1-8.

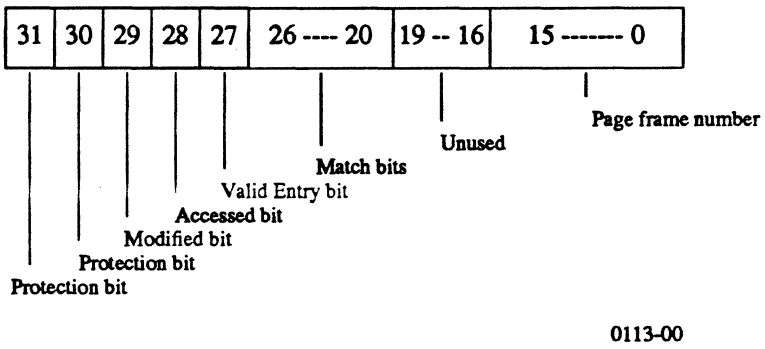


Figure 1-8. Translation Buffer Entry Format

1.3.4 Serial Ports

The VME-68K20 supports two asynchronous serial ports with software controlled baud rate. The serial ports may operate at speeds from 300 baud to 38.4 kilobaud.

1.3.5 EPROM

The EPROM contains self-test diagnostics, MMU initialization logic, and the UNIX bootstrap loader.

1.3.6 Floating Point Processor

The VME-68K20 has an optional on-board floating point processor (FPP). The FPP (an MC68881) operates as a co-processor. The FPP has its own oscillator so it can operate at a clock rate independent of the CPU.

1.4 Interrupts

The VME-68K20 uses the seven-level interrupt scheme of the 68020 to service three general categories of interrupts. These three categories of interrupts include

- Off-board interrupts
- On-board interrupts
- Interprocessor interrupts

1.4.1 Off-Board Interrupts

The VME-68K20 interrupt handler receives interrupt requests of up to seven different priority levels over the VME bus (see Figure 1-9). As the figure shows, the interrupt handler may receive all, some, or none of the VME INTERRUPT REQUEST signals as a function of configuration jumpers.

In a single processor environment, the interrupt handler would typically be configured to receive all seven levels of interrupts which it would handle in accordance with VME specifications. However, in multiprocessor environments, each VME-68K20 may be configured to handle only a portion of the interrupt requests on the VME bus, with the only restriction being that no two VME-68K20s may be configured to handle the same interrupt level.

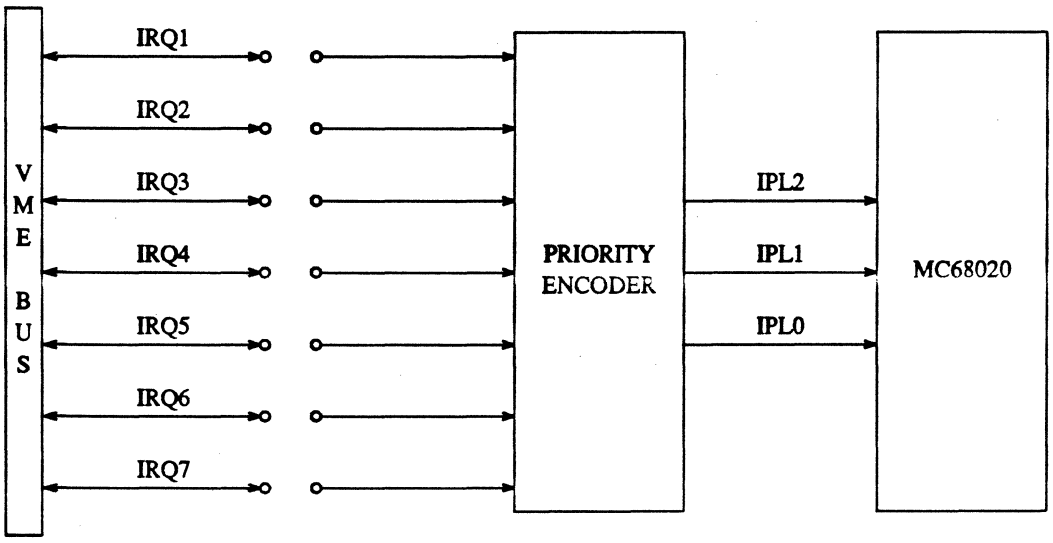


Figure 1-9. VME-68K20 Interrupt Handler

1.4.2 On-Board Interrupts

The VME-68K20 on-board interrupts include DARTINT* (on-board serial ports), CTCINT* (60 Hz clock generator), and NMIINT (non-maskable interrupt is generated when either an AC power failure or high-speed memory parity error occurs). The serial ports interrupt at Level 5, while the 60 Hz clock generator interrupts at Level 6.

1.4.3 Interprocessor Interrupts

In a multiprocessor environment, there are multiple processor boards, each of which is capable of receiving and handling interrupts from other processor boards, as well as generating and sending interrupts to other processor boards.

There are two methods for one processor to interrupt another. The first method is for the interrupting processor to place a particular address (interprocessor interrupt address) on the VME bus. In a multiprocessor environment, each VME-68K20 must be configured to have a unique interprocessor interrupt address. When this address is placed on the VME bus, an on-board interrupt at Level 4 or Level 7 (jumper controlled) is forced on the processor board that was addressed. The on-board interrupt results in an autovector to an interrupt handling routine. This first type of interprocessor interrupt is the *auto-vectored interprocessor interrupt.*

The second type of interprocessor interrupt is accomplished by sending an interrupt out on the VME bus over one of the INTERRUPT REQUEST lines IRQ3–IRQ6. When the interrupt acknowledge cycle occurs on the VME bus, the interrupting processor drives a programmable vector (the contents of the Interprocessor Interrupt Vector Register, an 8-bit on-board I/O register located at address 5C000000) on the bus to acknowledge the interrupt. This is the *vectored interprocessor interrupt.*

1.5 Software

The VME-68K20 is available with UNIX 4.2BSD (Berkeley Software Distribution). For those users that desire a full virtual memory implementation and a high level of file system performance, Berkeley UNIX is the software system of choice. Berkeley UNIX 4.2 now runs on almost all the Digital Equipment Corporation VAX systems that run UNIX. The Integrated Solutions' implementation provides the same VAX operating environment.

SECTION 2: SPECIFICATIONS

This section provides performance specifications and operating requirements for the VME-68K20 processor board.

2.1 Processor

The VME-68K20 employs the Motorola MC68020 processor operating at 16.67 MHz and supports an optional 12.5 MHz MC68881 floating point processor.

2.2 Form Factor

The VME-68K20 form factor is standard VME double-size — 160mm x 233.33mm.

2.3 System Bus

The system bus of the VME-68K20 is the VME bus as defined in the *VMEbus Specification*, Motorola part number MVMEBS/D1. The VME bus is implemented on VME bus connector P1 and the center row of pins on connector P2.

Connector P1 provides full support for the VME bus including all seven interrupt levels and full four level bus arbitration logic. The VME-68K20 can generate either 24-bit or 32-bit addresses on the VME bus. The VME-68K20 can generate either 24-bit or 32-bit addresses by using the extended addressing pins on the center row of the P2 connector. The P1 connector pin assignments and signal mnemonics are given in Table 2-1.

NOTE

With 32-bit addresses, the four high-order address bits are driven to 0.

2.4 High-Speed Memory Bus

The High-Speed Memory bus (HSMEM bus) is implemented on the user-defined I/O pins of the VME connector P2. Table 2-2 provides the HSMEM bus signal mnemonics and pin assignments for connector P2.

The concept of the HSMEM bus is central to realizing the full performance of the 68020 processor. The bus is 32-bit, semi-synchronous, with a very simple handshake sequence. All, part, or none of the high-speed memory is accessible from the VME bus depending on the configuration of a series of on-board jumpers. The HSMEM bus enables the 68020 processor to operate with one wait state on reads and no wait states on writes at 16.67 MHz.

The HSMEM bus has the following three groups of signal lines:

- Address lines HSADD1 through HSADD27
- Data lines HSDAT0 through HSDAT31
- Nine control lines

In addition to the P2 connector, which supports most of the HSMEM bus signals, VME-68K20 connector J2 supports high-speed address Bits 27–24 to enable the VME-68K20 to support up to 256 Mbytes of physical memory on the HSMEM bus. The J2 connector pin assignments and signal mnemonics are given in Table 2-3.

2.5 Bus Arbitration Timing

The bus arbitration latency is defined as the amount of time it takes to be granted the bus when the arbiter is in the idle state. For the VME-68K20, this amount of time varies from a minimum of 60 ns to a maximum of 120 ns due to the asynchronous nature of bus requests with respect to the arbiter which is a synchronous implementation. The average latency is 90 ns. These times are for the 16.67 MHz VME-68K20. P2 implements the HSMEM bus on the user-defined I/O pins (A and C rows) supporting a 16-Mbyte address space. The B row of pins implements 32-bit VME bus data transfers and the extended addressing bits discussed already.

Table 2-1. VME Bus Connector P1 Pin Assignments

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	VMED0	BBUSY*	VMED8
2	VMED1	BCLR*†	VMED9
3	VMED2	ACFAIL*	VMED10
4	VMED3	BG0IN*	VMED11
5	VMED4	BG0OUT*	VMED12
6	VMED5	BG1IN*	VMED13
7	VMED6	BG1OUT*	VMED14
8	VMED7	BG2IN*	VMED15
9	GND	BG2OUT*	GND
10	SYSCLK	BG3IN*	SYSFAIL*†
11	GND	BG3OUT*	VMEBERR*
12	VMEDS1*	BR0*	SYSRESET*
13	VMEDS0*	BR1*	LWRD*
14	VMEWR*	BR2*	VMEAM5
15	GND	BR3*	VADD23
16	VMEDTACK*	VMEAM0	VADD22
17	GND	VMEAM1	VADD21
18	VMEAS*	VMEAM2†	VADD20
19	GND	VMEAM3†	VADD19
20	VMEIACK*	GND	VADD18
21	IACKIN*	SERCLK†	VADD17
22	IACKOUT*	SERDAT†	VADD16
23	VMEAM4	GND	VADD15
24	VADD7	IRQ7*	VADD14
25	VADD6	IRQ6*	VADD13
26	VADD5	IRQ5*	VADD12
27	VADD4	IRQ4*	VADD11
28	VADD3	IRQ3*	VADD10
29	VADD2	IRQ2*	VADD9
30	VADD1	IRQ1*	VADD8
31	-12V	+5V STDBY†	+12V
32	+5V	+5V	+5V

† Not implemented on the VME-68K20.

Table 2-2. HSMEM Bus Connector P2 Pin Assignments

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	HSDAT7	+5 Volts	HSDAT15
2	HSDAT6	GND	HSDAT14
3	HSDAT5	RESERVED	HSDAT13
4	HSDAT4	VADD31	HSDAT12
5	HSDAT3	VADD30	HSDAT11
6	HSDAT2	VADD29	HSDAT10
7	HSDAT1	VADD28	HSDAT9
8	HSDAT0	VADD27	HSDAT8
9	VMECYC*	VADD26	HSADD23
10	HSADD1	VADD25	HSADD22
11	HSADD2	VADD24	HSADD21
12	HSADD3	GND	HSADD20
13	HSADD4	+5 Volts	HSADD19
14	HSADD5	VMED16	HSADD18
15	HSADD6	VMED17	HSADD17
16	HSADD7	VMED18	HSADD16
17	HSADD8	VMED19	HSADD15
18	HSADD9	VMED20	HSADD14
19	HSADD10	VMED21	HSADD13
20	HSADD11	VMED22	HSADD12
21	HSDAT31	VMED23	HSCYC*
22	HSDAT30	GND	HSLDS*
23	HSDAT29	VMED24	HSLW*
24	HSDAT28	VMED25	HSUDS*
25	HSDAT27	VMED26	HSDTACK*
26	HSDAT26	VMED27	HSPARERR*
27	HSDAT25	VMED28	HSDAT19
28	HSDAT24	VMED29	HSDAT18
29	HSDAT23	VMED30	HSDAT17
30	HSDAT22	VMED31	HSREF*
31	HSDAT21	GND	HSWRITE*
32	HSDAT20	+5 Volts	HSDAT16

Table 2-3. Connector J2 Pin Assignments

Pin Number	Signal Mnemonic	Signal Name
1	—	Unassigned
2	GND	Ground
3	HSADD27	High-Speed Bus Address 27
4	GND	Ground
5	HSADD26	High-Speed Bus Address 26
6	GND	Ground
7	HSADD25	High-Speed Bus Address 25
8	GND	Ground
9	HSADD24	High-Speed Bus Address 24
10	GND	Ground

2.6 Serial Ports

The two serial ports on the VME-68K20 board both support asynchronous communication with programmable baud rates of 300 to 19.2 kilobaud. Both serial ports exit the VME-68K20 board through a single ten-pin ribbon cable connector, J1, which is broken into two ports with an optional breakout panel. Table 2-4 provides the serial port signal mnemonics and J1 connector pin assignments.

Table 2-4. Serial Port Connector J1 Pin Assignments

Pin Number	Signal Mnemonic	Signal Name	Direction †
1	TXD0	Transmit data (Channel 0)	Out
2	RXD0	Receive data (Channel 0)	In
3	—	Unassigned	
4	—	Unassigned	
5	GND	Ground	
6	GND	Ground	
7	TXD1	Transmit data (Channel 1)	Out
8	RXD1	Receive data (Channel 1)	In
9	—	Unassigned	
10	—	Unassigned	

2.7 EPROM Socket

The EPROM socket is jumper configurable (jumpers E40–E42) for a single 2764, 27128, 27256, or 27512 EPROM.

† Out means the line is driven by the VME-68K20. In means the line is driven into the VME-68K20 by some external device.

2.8 Indicators

There are three LED indicators on the VME-68K20 board: DS1, DS2, and DS3. Table 2-5 provides a brief functional description of these indicators.

Table 2-5. VME-68K20 Indicators

Indicator	Description
DS1	DS1 is the VME bus Control Indicator. The LED lights when the VME-68K20 has control of the VME bus. DS1 extinguishes when another DMA device takes control of the bus or when the bus arbiter is in the idle state with no device in control of the VME bus.
DS2	DS2 is the Run Indicator. The LED lights when the 68020 on the VME-68K20 board is executing instructions.
DS3	DS3 is reserved for diagnostic use. DS3 corresponds to context register Bit 7. When Bit 7 is 1 the LED is off, when Bit 7 is 0 the LED is on. The LED is on during system power-up and is extinguished by the bootstrap EPROM after the power-on diagnostics have been successfully completed.

2.9 Power Requirements

The VME-68K20 power requirements are

- +5 volts at 6.4 amps
- +12 volts at 0.05 amp
- -12 volts at 0.05 amp

2.10 Environmental Requirements

The VME-68K20 environmental requirements are as follows

- Temperature: 0°C to 50°C (operating), -40°C to 65°C (non-operating).
- Humidity: 10% to 85% (non-condensing).

SECTION 3: CONFIGURATION

The VME-68K20 board has 45 sets of jumpers and one 10-bit switch bank (S1) which allow considerable configuration versatility. The jumpers are arranged into a number of logical groupings in accordance with the functions they perform. The physical locations of these functional groups of jumpers and S1 are shown in Figure 3-1. The jumper configuration descriptions and switch settings are provided in the paragraphs that follow.

3.1 MC68881 Floating Point Co-processor Clock (E5)

Jumper E5 controls the MC68881 floating point co-processor clock frequency. The jumper configurations are given in Table 3-1. The default clock frequency is 12.5 MHz.

Table 3-1. MC68881 Clock Frequency

Clock Frequency	Jumper Configuration
16.67 MHz	E5-2 to E5-3
12.5 MHz	E5-1 to E5-2

3.2 VME Bus Arbitration Jumpers

The VME-68K20 on-board arbiter has 22 jumper posts associated with it. These 22 jumper positions can be divided into three logical groups. The groups include the following:

- VME-68K20 bus request level (E7–E10 and E12)
- Bus arbiter enable/disable and bus grant control (E1–E4 and E6)
- Bus master type (E11)

3.2.1 Request Level (E7–E10 and E12)

Jumpers E7–E10 and E12 control the VME bus request level of the on-board requester. The VME-68K20 board contains full four-level bus arbitration logic, and the on-board 68020 must request the bus at one of these four levels. E12 is the common jumper and must be connected to one of the other four jumpers according to Table 3-2.

Note that the function of these jumpers is the same whether the on-board bus arbiter is used or an external off-board arbiter is used.

Table 3-2. VME-68K20 Bus Request Level Jumper Configurations

Jumpers	Bus Request Level
E12-E7	Level 0
E12-E9	Level 1
E12-E8	Level 2
E12-E10	Level 3

The processor *must* be connected to request the bus at only one level at any one time for reliable operation. In addition, this level must be the same level at which the VME-68K20 is inserted into the Bus Grant daisy chain. The factory default is E7-E12.

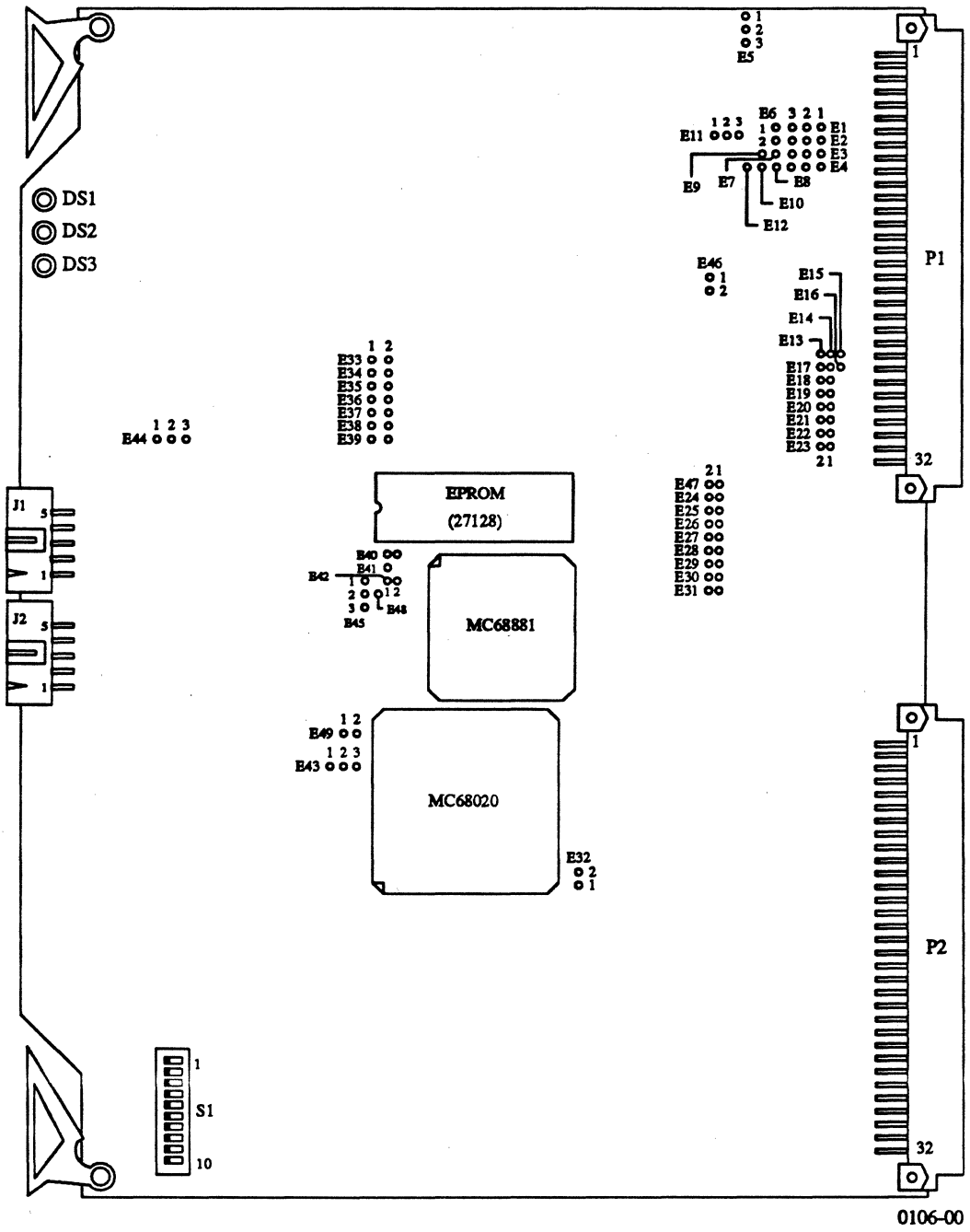


Figure 3-1. VME-68K20 Board Layout

3.2.2 Arbitration (E1–E4 and E6)

Jumpers E1–E4 and E6 are used in conjunction with the E7–E10 and E12 jumper configuration to control the bus grant daisy chains and to turn the bus arbiter on or off. The VME bus is a multiprocessor bus, but it has centralized bus arbitration. This means that when multiple VME-68K20 or other VME processor boards are simultaneously present on the bus, only one bus arbiter can be enabled. To explain the proper configuration of the jumpers, the signal present on each jumper pin must be considered. These pin assignments are given in Table 3-3.

Table 3-3. VME Bus Arbitration Jumper Pin Signals

Pin	Signal Name
E1-1	VME BUS GRANT IN Level 0
E2-1	VME BUS GRANT IN Level 1
E3-1	VME BUS GRANT IN Level 2
E4-1	VME BUS GRANT IN Level 3
E1-2	VME BUS GRANT OUT Level 0
E2-2	VME BUS GRANT OUT Level 1
E3-2	VME BUS GRANT OUT Level 2
E4-2	VME BUS GRANT OUT Level 3
E1-3	Arbiter BUS GRANT OUT Level 0
E2-3	Arbiter BUS GRANT OUT Level 1
E3-3	Arbiter BUS GRANT OUT Level 2
E4-3	Arbiter BUS GRANT OUT Level 3
E6-1	68020 BUS GRANT OUT
E6-2	68020 BUS GRANT IN

The VME BUS GRANT IN Levels 0-3 are inputs from the bus grant daisy chain when an external arbiter is used. These jumpers are not used when the on-board arbiter is used because the on-board arbiter serves as the start of the daisy-chain.

The VME BUS GRANT OUT Levels 0-3 are the bus grant daisy chain outputs that go on to the next VME board in the daisy chain. In the case where an external arbiter is used, three of these lines are tied straight through from VME BUS GRANT IN to VME BUS GRANT OUT. The one line which is not tied through passes through the on-board requester. In the case where the on-board arbiter is used, three of these lines are tied straight through from Arbiter BUS GRANT OUT to VME BUS GRANT OUT to start the daisy chain. Again the Arbiter BUS GRANT OUT/VME BUS GRANT OUT pair (which is not tied straight through) passes through the on-board requester and is the pair corresponding to the level at which the requester is currently jumpered to request the bus.

The Arbiter BUS GRANT OUT Levels 0-3 are the four outputs of the VME on-board arbiter and start the bus grant daisy chain. Not connecting these four lines effectively disables the on-board arbiter.

The explanation of the bus arbitration signals has given indications of the proper way to connect the jumpers for various possible arbiter/requester configurations. The following two examples further clarify the jumper configurations.

Example One

In the factory default configuration for the VME-68K20 board, the on-board arbiter is active and the on-board requester is requesting and being granted use of the VME bus at Level 0, the lowest of the four priority levels. Jumpers E2-3 and E2-2, E3-3 and E3-2, and E4-3 and E4-2 are connected together to originate the bus grant daisy chains for Levels 1, 2, and 3 respectively (see Figure 3-2). Jumper E1-3, which originates the daisy chain for Level 0 is tied to E6-2 which is the on-board requester BUS GRANT IN pin. Jumper E6-1, the on-board requester BUS GRANT OUT pin, is tied to E1-2, the VME BUS GRANT OUT pin. In this manner, the Level 0 bus grant originates on the VME-68K20 board, passes through the on-board requester, and then is driven out onto the VME bus.

NOTE

This configuration is used for the server node in a clustered system or workstation.

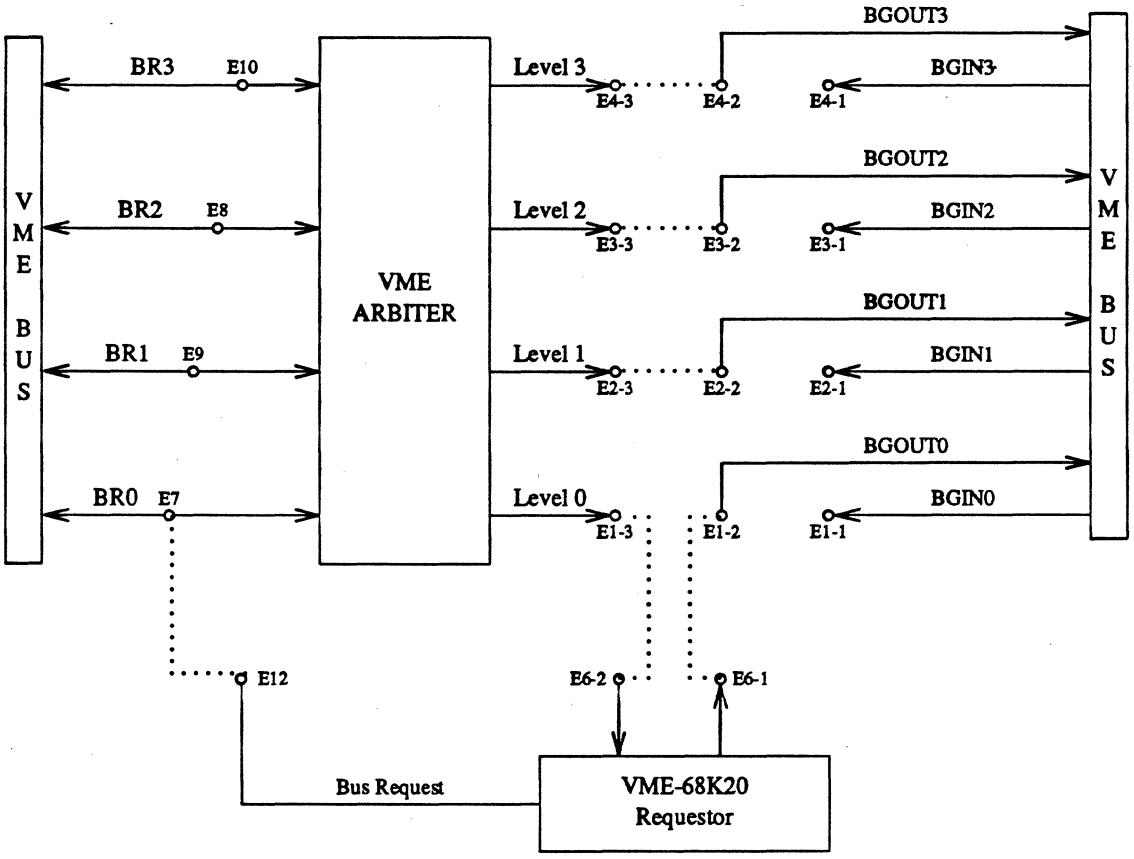


Figure 3-2. Example One: On-Board Arbiter Active

Example Two

In this case, an off-board arbiter is used and the on-board requester is requesting and being granted use of the VME bus at Level 3, the highest level. In this configuration, jumpers E1-2 and E1-1, E2-2 and E2-1, and E3-2 and E3-1 are connected to pass the bus grant daisy chains at Levels 0, 1 and 2 straight through the board (see Figure 3-3). Jumper E4-1, the VME BUS GRANT IN at Level 3 is tied to E6-2, the on-board requester BUS GRANT IN pin and jumper E6-1, the on-board requester BUS GRANT OUT pin is tied to E4-2, the VME BUS GRANT OUT pin at Level 3. In this manner, the Level 3 bus grant daisy chain passes through the on-board requester and the other 3 levels of bus grant pass straight through the board.

NOTE

This configuration is used for the various cluster nodes in a clustered system or workstation.

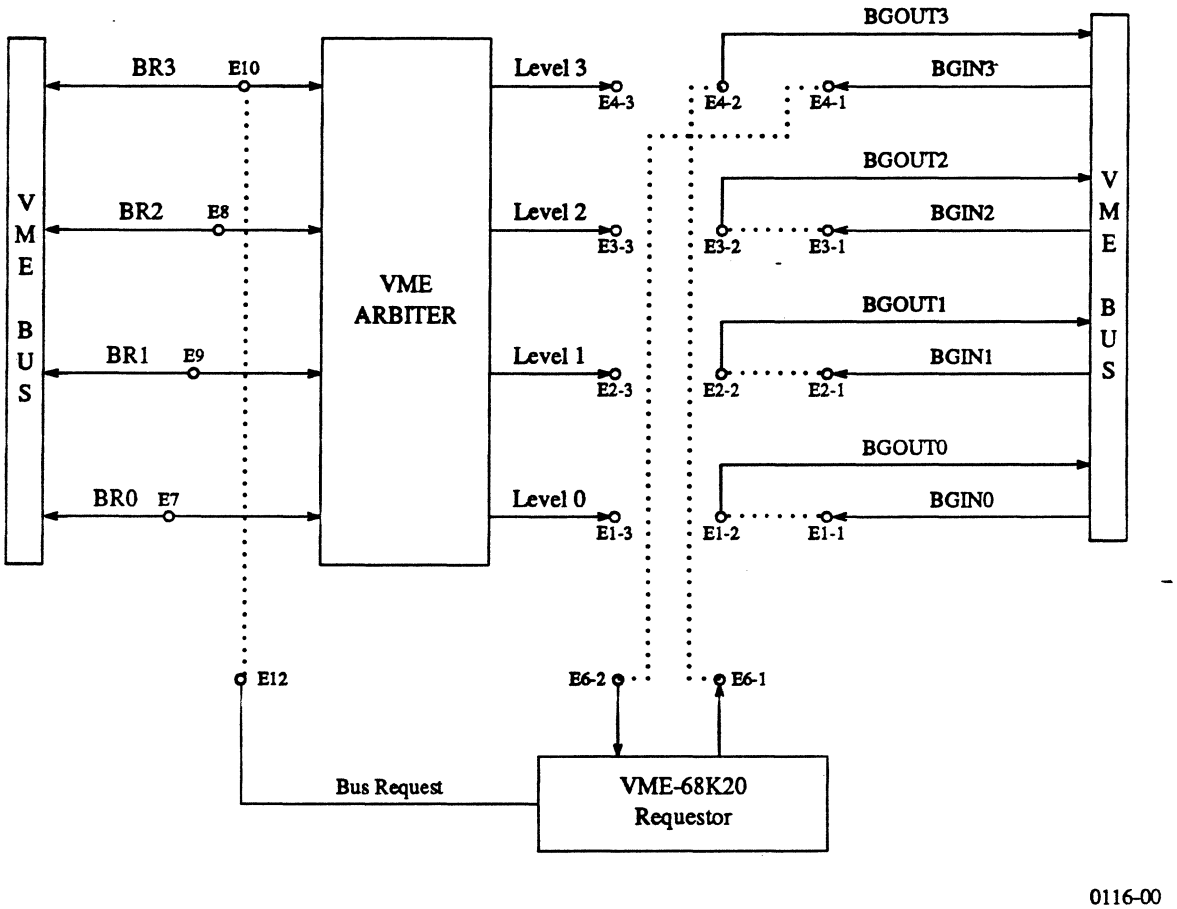


Figure 3-3. Example Two: Off-Board Arbiter Active

3.2.3 Bus Master Type (E11)

The jumper E11 controls whether the VME-68K20 is a Release on Request (ROR) or a Release When Done (RWD) bus master. If the jumper is inserted from E11-1 to E11-2, the VME-68K20 becomes a ROR bus master (factory default). This means that once it becomes bus master, it remains bus master until another bus request is present on the bus. Note that any level of bus request causes any level of bus master to release the bus. This is the appropriate jumper positioning for most single processor applications or for the server node in a clustered system or workstation.

If the jumper is inserted from E11-1 to E11-3, then the VME-68K20 becomes a RWD bus master. This means that the processor gets off the bus after completing each VME bus cycle and must again request the bus before it can do another VME bus transfer. This may be the appropriate configuration for a multiprocessor situation where the processors execute mostly from the HSMEM bus and only occasionally require access to the VME bus (cluster node in a clustered system or workstation).

Note that both the ROR and the RWD configurations are completely implemented in hardware and are completely transparent to the 68020. No code is required to either acquire or relinquish the bus in either case.

3.3 EPROM Socket Configuration (E40–E42)

Jumpers E40–E42 configure the VME-68K20 EPROM socket (board location U48, see Figure 3-1) for either a 2764, 27128, 27256, or a 27512 EPROM. The jumper configurations are given in Table 3-4.

Table 3-4. EPROM Socket Configuration

EPROM	Jumper Configuration
2764	E40-1 to E41 E41 to E42-1
27128	E40-1 to E41 E41 to E42-1
27256	E40-1 to E41 E42-1 to E42-2
27512	E40-1 to E40-2 E42-1 to E42-2

The factory default configuration is set for a 27128.

3.4 MC68020 Instruction Cache Enable/Disable (E32)

Jumper E32 determines whether or not the MC68020 internal instruction cache is enabled. The cache should always be enabled (no jumper inserted).

3.5 RAS Precharge Time Compensation (E34)

Jumper E34 determines the number of dead cycles that are inserted between high-speed memory cycles to compensate for RAS precharge time (current memory technology requires 90 to 100ns RAS precharge time). If E34-1 and E34-2 are not jumpered, one dead cycle is inserted between high-speed memory cycles. With a jumper installed between E34-1 and E34-2 (the factory default), two dead cycles are inserted (E34-1 and E34-2 should always remain jumpered).

3.6 Interrupts

The VME-68K20 has three groups of jumpers that are associated with three broad categories of interrupts. The three categories of interrupts include the following:

- Off-board interrupts
- On-board interrupts
- Interprocessor interrupts

3.6.1 Off-Board Interrupts (E17–E23)

The VME-68K20's interrupt handler may handle all, some, or none of the seven levels of VME Interrupt Requests (IRQ1–IRQ7). Jumpers E17–E23 determine which VME bus (or off-board) interrupt requests will be handled by a VME-68K20's interrupt handler.

In a single-processor environment, the single VME-68K20 will have the only interrupt handler on the VME bus, and consequently, it will handle all levels of off-board interrupts. In this case, jumpers E17–E23 will be installed (see Figure 3-4).

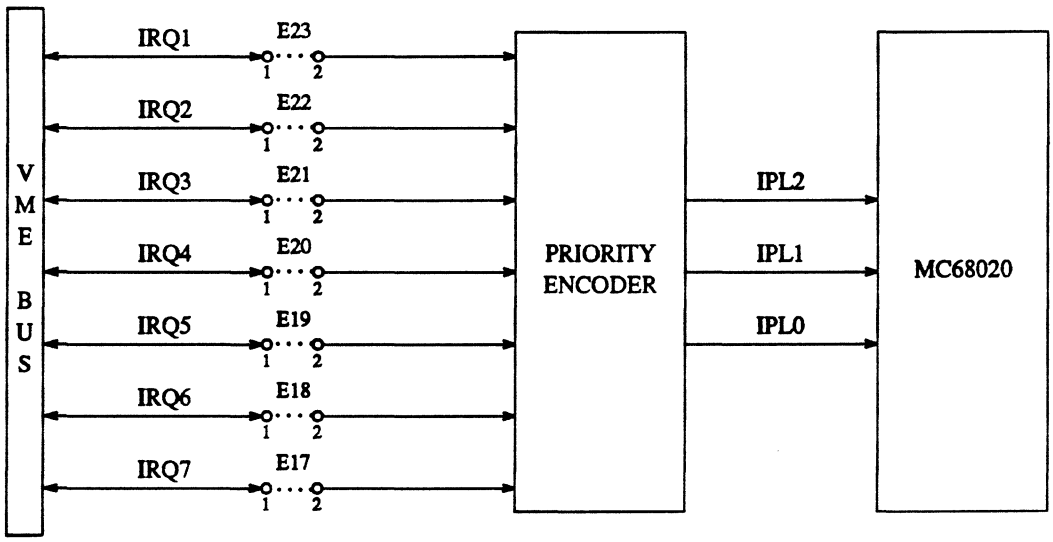


Figure 3-4. Off-Board Interrupt Jumper Configuration (Single-processor)

In multiprocessor environments, interrupt handlers on multiple processor boards may be configured to handle any portion of the seven levels of off-board interrupts, although no two interrupt handlers may handle identical interrupt levels. In Figure 3-5 for example, processor A's interrupt handler is configured to handle interrupts Levels 4–7, while processor B handles interrupt Levels 1–3.

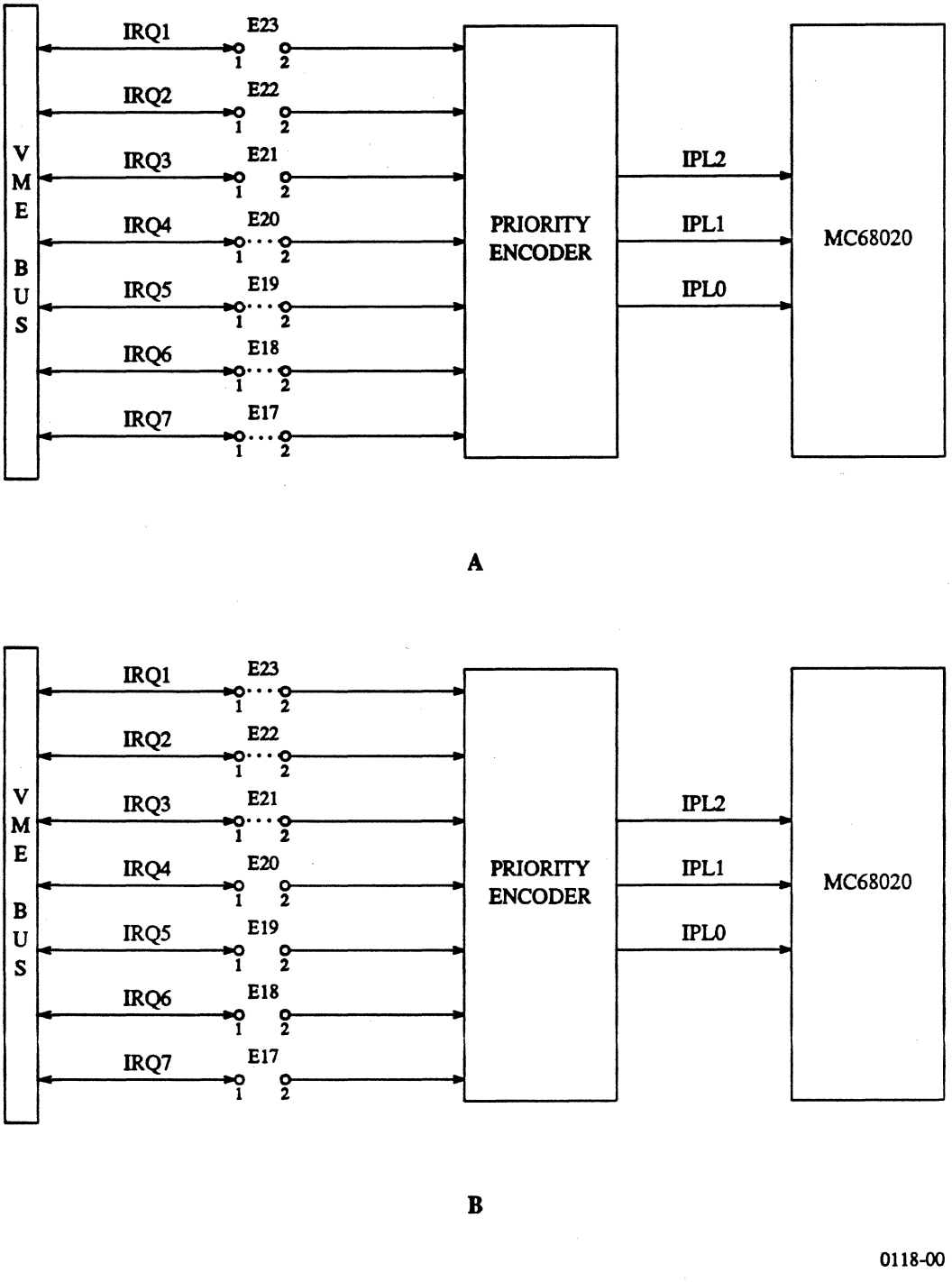


Figure 3-5. Off-Board Interrupt Jumper Configurations (Multiprocessor)

3.6.2 On-Board Interrupts (E13 and E14)

There are two on-board interrupt lines, DARTINT* and CTCINT*, that have associated configuration jumpers (E13 and E14, respectively). E13 must be tied to E19-2 (interrupt Level 5) and E14 must be tied to E18-2 (interrupt Level 6).

3.6.3 Interprocessor Interrupts (E15, E16, E33, and E35–E37)

In a multiprocessor environment, there are multiple processor boards, each of which is capable of receiving and handling interrupts from other processor boards, as well as generating and sending interrupts to other processor boards. The proper configuration of jumpers E15, E16, E33, and E35–E37 enables this interprocessor interrupt scheme to operate properly.

There are two methods for one processor to interrupt another. The first method is for the interrupting processor to place a particular address (interprocessor interrupt address) on the VME bus. In a multiprocessor environment, each VME-68K20 must be configured to have a unique interprocessor interrupt address. When this address is placed on the VME bus, an on-board interrupt is generated on the processor board that was addressed. The on-board interrupt results in an autovector to an interrupt handling routine. This first type of interprocessor interrupt is the *auto-vectored interprocessor interrupt.*

The second type of interprocessor interrupt is accomplished by sending an interrupt out on the VME bus over one of the INTERRUPT REQUEST lines. When the interrupt acknowledge cycle occurs on the VME bus, the interrupting processor drives a programmable vector on the bus to acknowledge the interrupt. This is the *vectored interprocessor interrupt.*

3.6.3.1 Auto-Vectored Interprocessor Interrupt Address (E35)

One processor board may interrupt another by writing to a particular address located in the interprocessor interrupt space (short I/O space). Since there may be as many as eight different processors on the VME bus, each must have a unique address within the interprocessor interrupt space in order to recognize which processor is being interrupted. Establishing this unique address is the function of jumper E35 and two programmable bits in the Context Register (Bits 6 and 5 which are Lev1 and Lev2, respectively). The E35, Lev1, and Lev2 bit configurations are provided in Table 3-5.

Table 3-5. Auto-Vectored Interprocessor Interrupt Addresses

Interprocessor Interrupt Address	E35	Lev1	Lev2	Processor
C000	in	in	in	Server
C002	in	in	out	Cluster 1
C004	in	out	in	Cluster 2
C006	in	out	out	Cluster 3
C008	out	in	in	Cluster 4
C00A	out	in	out	
C00C	out	out	in	
C00E	out	out	out	

3.6.3.2 Auto-Vectored Interprocessor Interrupt Priority (E15 and E33)

When one VME-68K20 places the auto-vectored interprocessor interrupt address of another processor on the VME bus, an on-board interprocessor interrupt is generated on the board that received the interrupt, i.e., the ONBDINT* signal is asserted. The priority of this interrupt is controlled by jumpers E15 and E33. The interrupt priority may be set at either Level 4 or Level 7 (see Table 3-6).

Table 3-6. Auto-Vectored Interprocessor Interrupt Priority

Priority Level	Jumper
4	E15 to E20-2 with E33 installed
7	E15 to E17-2 with E33 out

3.6.3.3 Vectored Interprocessor Interrupts (E16, E36, and E37)

When using vectored interprocessor interrupts, the interrupting VME-68K20 must send an interprocessor interrupt out over the VME bus on one of the INTERRUPT REQUEST lines IRQ3–IRQ6. Jumper E16 may be tied to Pin 1 of any one of these INTERRUPT REQUEST line jumpers (E18–E21) in order to send interprocessor interrupts to another interrupt handler at a selected priority level. For example, in Figure 3-6 the vectored interprocessor interrupt is sent out at priority Level 6. In addition to jumper E16, jumpers E36 and E37 must be configured as shown in Table 3-7.

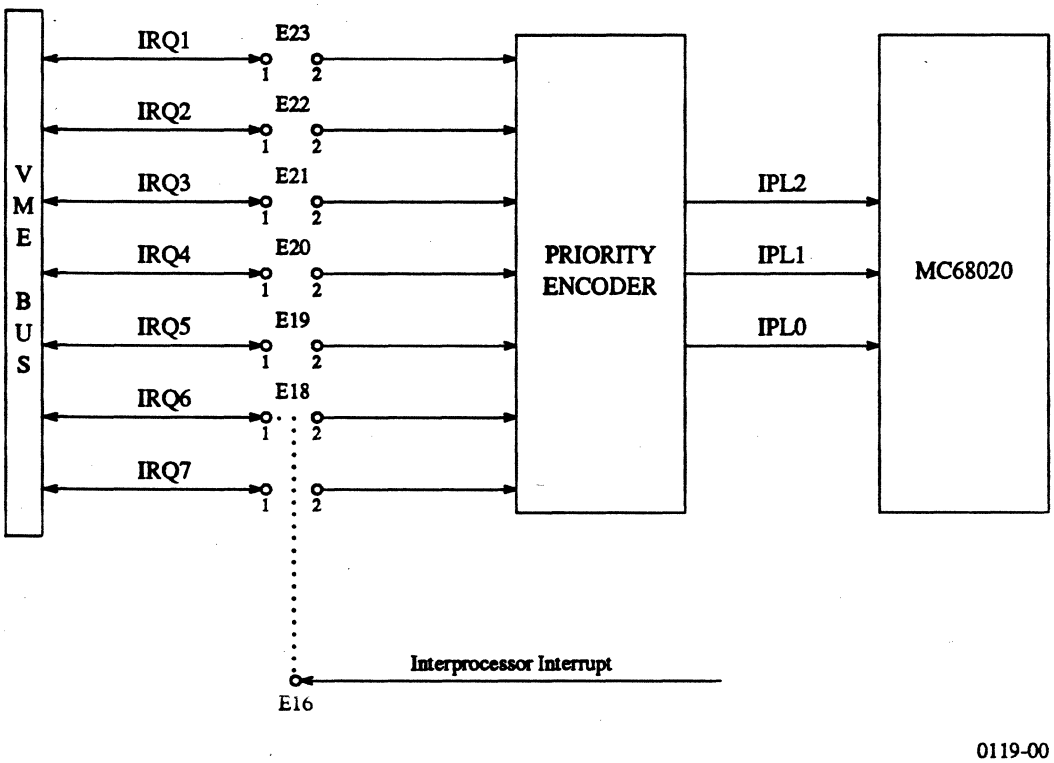


Figure 3-6. Off-Board Interprocessor Interrupt Jumpers

Table 3-7. Vectored Interprocessor Interrupt Jumpers

Interrupt Level	Jumper E16 to....	E36	E37
3	E21-1	in	in
4	E20-1	in	out
5	E19-1	out	in
6	E18-1	out	out

3.7 Board Status Register (E24–E31)

Jumpers E24–E31 set values in a byte register, the Board Status Register (BSR), which are associated with on-board PROM or operating system functions. Jumpers E24 and E25 correspond to BSR Bits 7 and 6, respectively. These jumpers determine the power-up baud rate for the on-board serial ports as shown in Table 3-8.

Table 3-8. Baud Rate Jumpers

Baud Rate	E24	E25
300	in	in
1200	in	out
19200	out	in
9600	out	out

NOTE

These jumpers are read by the 68020. The baud rate setting is a software function.

Jumpers E28–E31 (BSR Bits 3–0, respectively) set the processor number for a particular VME-68K20 processor board. The processor number for the VME-68K20 in a single-processor system is 0. In multiprocessor clustered systems and workstations, the server node (master) VME-68K20 is processor 0, while each additional cluster node (slave) is incremented by one. The jumpers provide a four-bit binary coded processor number with jumper E31 corresponding to the least significant bit and E28 the most significant bit. Table 3-9 provides the currently valid processor numbers and their corresponding jumper configurations. Jumpers E26 and E27 (BSR Bits 5 and 4, respectively) are reserved for future use.

Table 3-9. Processor Number Jumpers

Processor Number	E28	E29	E30	E31	Processor Implementation
0†	in	in	in	in	Single-processor or server node
1	out	out	out	in	Cluster 1
2	out	out	in	out	Cluster 2
3	out	out	in	in	Cluster 3
4	out	in	out	out	Cluster 4

† The binary value 15 is mapped to a value of 0.

3.8 On-Board I/O Cycle Length (E44)

Jumper E44 determines the length of on-board I/O cycles for the DUART, context register, error register, and PROMs. The jumper configurations and corresponding I/O cycle lengths for VME-68K20s running at 16.67 MHz are given in Table 3-10.

Table 3-10. I/O Cycle Length Configuration

I/O Cycle Length	Jumper Configuration
500 nanoseconds	E44-1 to E44-3
1 microsecond	E44-1 to E44-2†

3.9 VME-68K20 Access to VME Bus Memory (S1)

Using regular addressing, the VME-68K20 supports up to 16 Mbytes of physical memory. This memory may be present on either the VME bus or the HSMEM bus. Switch S1 (ten-bit switch) is used to set the address boundary below which all references to memory are to the HSMEM bus and above which all memory references are to the VME bus.

NOTE

S1 is functional only when using translated addresses.

Using regular addressing, S1 bit switches 1 through 4 must always remain open. Bit switches 5 through 10 are then set to determine the address boundary between VME bus and HSMEM bus memory references (see Table 3-11).

Table 3-11. S1 Switch Settings (Regular Addressing)

VME Bus Memory Address Range	HSMEM Bus Memory Address Range	S1 Bit Settings†						
		1–4	5	6	7	8	9	10
FC0000–FFFFFF	0–FBFFFF	O	C	C	C	C	C	C
F80000–FFFFFF	0–F7FFFF	O	C	C	C	C	O	C
F00000–FFFFFF	**0–EFFFFF**	**O**	**C**	**C**	**C**	**O**	**O**	**C**
E00000–FFFFFF	0–DFFFFF	O	C	C	O	O	O	C
C00000–FFFFFF	0–BFFFFF	O	C	O	O	O	O	C
800000–FFFFFF	0–7FFFFF	O	O	O	O	O	O	C
0–FFFFFF	None	O	O	O	O	O	O	O

NOTE

The factory default is printed in bold-face type.

† This jumper configuration is reserved for future use with faster versions of the processor. With increased clock speed, the I/O cycle length will change from the currently specified value.

When using extended addressing, S1 bit switches 1, 2, 3, and 4 correspond to address bits 27, 26, 25, and 24, respectively. The S1 bit settings for extended addressing are shown in Table 3-12.

Table 3-12. S1 Switch Settings (Extended Addressing)

VME Bus Memory Address Range	HSMEM Bus Memory Address Range	S1 Bit Settings 1	2	3	4	5	6	7	8	9	10
FFC0000–FFFFFFF	0–FFBFFFF	C	C	C	C	C	C	C	C	C	C
FF80000–FFFFFFF	0–FF7FFFF	C	C	C	C	C	C	C	C	O	C
FF00000–FFFFFFF	0–FEFFFFF	C	C	C	C	C	C	C	O	O	C
FE00000–FFFFFFF	0–FDFFFFF	C	C	C	C	C	C	O	O	O	C
FC00000–FFFFFFF	0–FBFFFFF	C	C	C	C	C	O	O	O	O	C
F800000–FFFFFFF	0–F7FFFFF	C	C	C	C	O	O	O	O	O	C
F000000–FFFFFFF	0–EFFFFFF	C	C	C	C	O	O	O	O	O	O
E000000–FFFFFFF	0–DFFFFFF	C	C	C	O	O	O	O	O	O	O
C000000–FFFFFFF	0–BFFFFFF	C	C	O	O	O	O	O	O	O	O
8000000–FFFFFFF	0–7FFFFFF	C	O	O	O	O	O	O	O	O	O
0–FFFFFFF	None	O	O	O	O	O	O	O	O	O	O

3.10 VME Bus Access to HSMEM Bus Memory (E38 and E39)

Jumpers E38 and E39 control the VME bus window in HSMEM bus memory. There are four possible settings which include the following:

- For use with the Optimum V Systems (both with and without graphics capabilities)
- For use with a cluster file server (master) processor
- For use with a cluster node (slave) processor
- For use with extended memory systems (greater than 16 Mbytes of high-speed memory)

The jumper configurations for E38 and E39 are provided in Table 3-13.

Table 3-13. Jumper E38 and E39 Configurations

VME Bus Window Into HSMEM Memory	E39	E38	Product Supported
0–DFFFFE	in	out	Optimum V Systems and WorkStations
0–7FFFFE	out	out	Cluster server (master)
no VME bus window	in	in	Cluster node (slave)
0–EFFFFFE	out	in	Extended memory systems

† In this table the bit switch settings are represented by the letters C or O, which mean closed (on) or open (off), respectively.

3.11 Non-Translated Address Boundary (E43)

In non-translated virtual address space, a different mechanism is used to control the boundary between HSMEM bus accesses and VME bus accesses. Currently, a single jumper (E43) is used to control this boundary. Table 3-14 provides the E43 jumper configurations.

Table 3-14. Non-Translated Address Boundary Jumper

Jumper	VME Bus Memory Address Range	HSMEM Bus Memory Address Range	Products
E43-1 to E43-2	E00000–FFFFFE	0–DFFFFE	Optimum V Systems and WorkStations
E43-2 to E43-3	800000–FFFFFF	0–7FFFFE	Cluster server (master) Cluster node (slave)

NOTE

With the current PAL (U37), all extended accesses (above 16 Mbytes) in non-translated virtual address space always go out on the VME bus. Contact Integrated Solutions if you have special requirements.

3.12 Static VME Bus I/O Sizing (E49)

Jumper E49 controls whether the VME bus I/O space (E00000–FFFFFE) is treated as a 16-bit addressable entity (E49 is in) or a 32-bit addressable entity (E49 is out).

SECTION 4: OPERATION/PROGRAMMING

The VME-68K20 logical address space from 30000000_{16} to $6FFFFFFF_{16}$ is dedicated to on-board I/O. The on-board I/O addresses do not pass through the MMU nor are they protected by it. However, on-board I/O space is only accessible when the MC68020 is in system mode. An attempt to access on-board I/O space while in user mode results in a bus-trap error. The on-board I/O address space is subdivided into four address areas which are allocated as shown in Table 4-1.

Table 4-1. On-Board I/O Address Space

Address Area	Base Address (hex)
System Translation Buffer	30000000
User Translation Buffer	40000000
Byte I/O Space	50000000
PTP Buffer	60000000

4.1 Translation Buffer

The translation buffer is made up of 1,024 by 28 bits of very high-speed static RAM. There are 512 entries for system translations (base address 30000000) and 512 entries for user translations (base address 40000000).

The translation buffer increments in quantities of four Kbytes (1000_{16}). Since the system and user translation buffer address areas each have 512 entries, the space actually used for system and user translations is indicated by the address ranges given in Table 4-2.

Table 4-2. System and User Translation Buffer Address Ranges

Buffer	Address Range (hex)
System Translation Buffer	30000000–301FF000
User Translation Buffer	40000000–401FF000

The translation buffer entry format is discussed in Section 1.

4.2 Byte I/O Space

The byte I/O space is a small portion of space accessible in system mode used for access to or programming of the following:

- EPROM
- DUART (serial port chip)
- Error Source Register (ESR)
- Interprocessor Interrupt Vector Register (IVR)
- Board Status Register (BSR)
- Context Register (CTX)

The base addresses for the members resident in the byte I/O space are provided in Table 4-3.

Table 4-3. Byte I/O Space Address Areas

Register/Chip	Base Address (hex)
EPROM	54000000
DUART	58000000
ESR (read only)	5C000000
IVR (write only)	5C000000
BSR (read only)	50000000
CTX (write only)	50000000

4.2.1 DUART (Serial Ports)

The VME-68K20 has two asynchronous serial ports with software controlled baud rate. The serial ports are implemented with a Signetics SCN2681 dual asynchronous receiver/transmitter (DUART).

The operation of the DUART is controlled by writing control words into appropriate control registers. The CPU monitors status by reading the channel status registers. The control and status register addresses are provided in Table 4-4. For detailed programming information, refer to Appendix B DUART Application Notes.

Table 4-4. DUART Register Addresses

Address	Register	Type	Description
58000000	MR1A/MR2A	Read/Write	Mode Registers A1 and A2
58000002	SRA	Read	Status Register A
58000002	CSRA	Write	Clock Select Register A
58000004	CRA	Write	Command Register A
58000006	RHRA	Read	Receiver Holding Register A
58000006	THRA	Write	Transmitter Holding Register A
58000008	IPCR	Read	Input Port Change Register
58000008	ACR	Write	Auxiliary Control Register
5800000A	ISR	Read	Interrupt Status Register
5800000A	IMR	Write	Interrupt Mask Register
5800000C	CTU	Read	Counter/Timer Upper
5800000E	CTL	Read	Counter/Timer Lower
58000010	MR1B/MR2B	Read/Write	Mode Registers B1 and B2
58000012	SRB	Read	Status Register B
58000012	CSRB	Write	Clock Select Register B
58000014	CRB	Write	Command Register B
58000016	RHRB	Read	Receiver Holding Register B
58000016	THRB	Write	Transmitter Holding Register B
58000018	OPCR	Write	Output Port Configuration Register
58000018	INP	Read	Input Port
5800001C	SOPBC	Write	Set Output Port Bits Command
5800001C	STARTC	Read	Start Counter Command
5800001E	ROPBC	Write	Reset Output Port Bits Command
5800001E	STOPC	Read	Stop Counter Command

The DUART is typically initialized with a short program that resets the control registers. The VME-68K20 EPROM-resident SIO port initialization program is as follows:

```
/*
 * Initialize SIO ports
 */
    movb    #CR_CMD_RST_MR,S2681_W_CR_A
    movb    #CR_CMD_RST_MR,S2681_W_CR_B
    movb    #CR_CMD_RST_RX,S2681_W_CR_A
    movb    #CR_CMD_RST_RX,S2681_W_CR_B
    movb    #CR_CMD_RST_TX,S2681_W_CR_A
    movb    #CR_CMD_RST_TX,S2681_W_CR_B
    movb    #CR_CMD_RST_ERR,S2681_W_CR_A
    movb    #CR_CMD_RST_ERR,S2681_W_CR_B
    movb    #CR_CMD_RST_BRK,S2681_W_CR_A
    movb    #CR_CMD_RST_BRK,S2681_W_CR_B
    movb    #(MR1_PAR_NONE|MR1_8_BIT),S2681_MR1_A
    movb    #(MR1_PAR_NONE|MR1_8_BIT),S2681_MR1_B
    movb    #MR2_1_STOP,S2681_MR2_A
    movb    #MR2_1_STOP,S2681_MR2_B
    movb    #0xBB,S2681_W_CSR_A          /* baud rates 9600*/
    movb    #0xBB,S2681_W_CSR_B
    movb    #(CR_ENABLE_TX|CR_ENABLE_RX),S2681_W_CR_A
    movb    #(CR_ENABLE_TX|CR_ENABLE_RX),S2681_W_CR_B
```

4.2.2 Error Source Register

The ESR is a read-only byte register located at address 5C000000. The bit definitions for the ESR are given in Table 4-5.

Table 4-5. ESR Bit Definitions

Bit	Mnemonic	Description
0	ABORT	In the case of a bus error, if Bit 0 is 0 and Bit 3 (PROTERR) is 0, the PTP was aborted. If Bit 0 is 1 and Bit 3 is 0 then a protection violation occurred.
1	VMEBERR	In the case of a bus error, if Bit 1 is 0, a VME bus error occurred.
2	ILLADDR	In the case of a bus error, if Bit 2 is 0, an illegal address error occurred.
3	PROTERR	In the case of a bus error, if Bit 3 is 0 and Bit 0 is 1, a protection violation occurred.
4	ACFAIL	In the case of an NMI, if Bit 4 is 0, an AC power failure occurred.
5	PARITY	In the case of an NMI, if Bit 5 is 0, a parity error occurred on the HSMEM bus.

4.2.3 Board Status Register

The BSR is a read-only byte register located at address 50000000. The bit values in the BSR are set by VME-68K20 board jumpers E24–E31. Refer to Section 3 for the BSR jumper configurations and corresponding bit definitions.

4.2.4 Context Register

The CTX is a write-only byte register located at address 50000000. The bit functions of the CTX are described in Table 4-6.

Table 4-6. CTX Bit Definitions

Bit	Description
7	Bit 7 is used for diagnostic purposes. Bit 7 is 0 during system power-up. Upon successful completion of the power-on diagnostics, the bootstrap EPROM sets Bit 7 to 1. The condition of Bit 7 may be monitored by watching VME-68K20 board indicator (LED) DS3. When Bit 7 is 1 DS3 is off, when Bit 7 is 0 DS3 is on.
6–5	CTX Bits 6 and 5 in conjunction with jumper E35 determine the auto-vectored interprocessor interrupt address in short I/O space for a particular VME-68K20 processor. Refer to Section 3 for the jumper and bit configurations and their corresponding auto-vectored interprocessor interrupt addresses.
4	If a 1 is written to Bit 4, the 60 Hz clock interrupt is enabled. A 0 value for Bit 4 disables the 60 Hz clock interrupt.
3	If a 1 is written to Bit 3, parity checking is enabled. A 0 value for Bit 3 disables parity checking.
2–0	Bits 2–0 determines the context in the PTP buffer. A binary 0 value for the three bits is the system context, while binary bit codes of one through seven correspond to the seven user contexts. Bit 0 is the low order bit for the context number.

4.3 Page Table Pointer Buffer

The PTP buffer stores the current context of 128 twelve-bit Page Table Pointer (PTPs). The PTP buffer's base address is 60000000. Since the PTP buffer is incremented in two-Mbyte increments, the 128 PTPs are stored within the address range of 60000000 to 6FE00000. The format of the PTPs is discussed in Section 1.

APPENDIX A: VME-68K20 BOARD COMPATIBILITY

The VME-68K20 processor board may be installed into any of the Optimum V family products. In order to ensure compatibility with other printed circuit boards in the system or workstation, Table A-1 provides the part numbers and revision levels for the printed circuit boards that are compatible with the VME-68K20.

Table A-1. Compatible Printed Circuit Boards

Product	Part Number	Description
VME-SCSI	590026-03 Rev. D	Disk controller
VME-HSMEM	590031 Rev. D	High-speed memory
VME-ICP8	590042-01 Rev. B	Intelligent Communications Processor (8 ports)
VME-ICP16	590042-02 Rev. A	Intelligent Communications Processor (16 ports)
VME-QIC2/RTC	590045-03 Rev. A	Cartridge Tape Controller
Monochrome display memory board	590082 Rev. B	Monochrome display memory
VME-GIP (monochrome)	590081 Rev. B	Monochome display controller
Color display memory board	590086	Color display memory
VME-GIP (color)	590077 Rev. B	Color display controller
VME-TC50	590064 Rev. A	Half-inch tape controller
VME-EC	630027	Ethernet controller
VME-SMD	630034	SMD disk controller

APPENDIX B: DUART APPLICATION NOTES

The two asynchronous serial ports on the VME-68K20 processor board are implemented with Signetics' SCN2681 DUART. The pages that follow provide detailed technical and programming information regarding the 2681. Section 2 of this manual provides the serial port connector pin assignment, while Section 4 provides the DUART control register addresses.

MICROPROCESSOR DIVISION JANUARY 1983

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

DESCRIPTION

The Signetics SCN2681 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single chip MOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It interfaces directly with microprocessors and may be used in a polled or interrupt driven system.

The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16x clock derived from a programmable counter/timer, or an external 1x or 16x clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

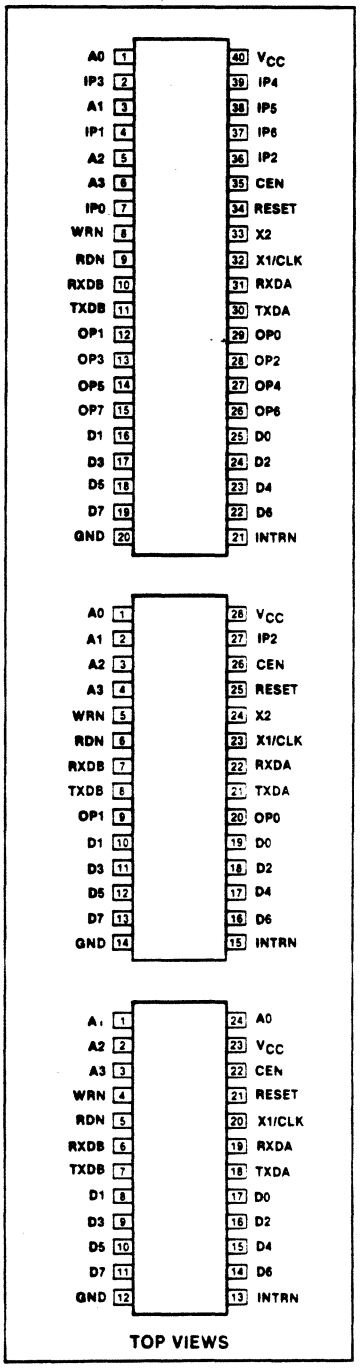
Also provided on the SCN2681 are a multipurpose 7-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

The SCN2681 is available in three package versions to satisfy various system requirements: 40-pin and 28-pin, both 0.6" wide DIPs, and a compact 24-pin, 0.4" wide, DIP.

FEATURES

- **Dual full-duplex asynchronous receiver/transmiter**
- **Quadruple buffered receiver data registers**
- **Programmable data format**
  - **—5 to 8 data bits plus parity**
  - **—Odd, even, no parity or force parity**
  - **—1, 1.5 or 2 stop bits programmable in 1/16 bit increments**
- **Programmable baud rate for each receiver and transmiter selectable from:**
  - **—18 fixed rates: 50 to 38.4K baud**
  - **—One user defined rate derived from programmable timer/counter**
  - **—External 1x or 16x clock**
- **Parity, framing, and overrun error detection**
- **False start bit detection**
- **Line break detection and generation**
- **Programmable channel mode**
  - **—Normal (full duplex)**
  - **—Automatic echo**
  - **—Local loopback**
  - **—Remote loopback**
- **Multi-function programmable 16-bit counter/timer**
- **Multi-function 7-bit input port**
  - **—Can serve as clock or control inputs**
  - **—Change of state detection on four inputs**
- **Multi-function 8-bit output port**
  - **—Individual bit set/reset capability**
  - **—Outputs can be programmed to be status/interrupt signals**
- **Versatile interrupt system**
  - **—Single interrupt output with eight maskable interrupting conditions**
  - **—Output port can be configured to provide a total of up to six separate wire-OR'able interrupt outputs**
- **Maximum data transfer: 1X — 1MB/sec, 16X — 125KB/sec**
- **Automatic wake-up mode for multidrop applications**
- **Start-end break interrupt/status**
- **Detects break which originates in the middle of a character**
- **On-chip crystal oscillator**
- **TTL compatible**
- **Single +5V power supply**

PIN CONFIGURATION



TOP VIEWS

ORDERING CODE

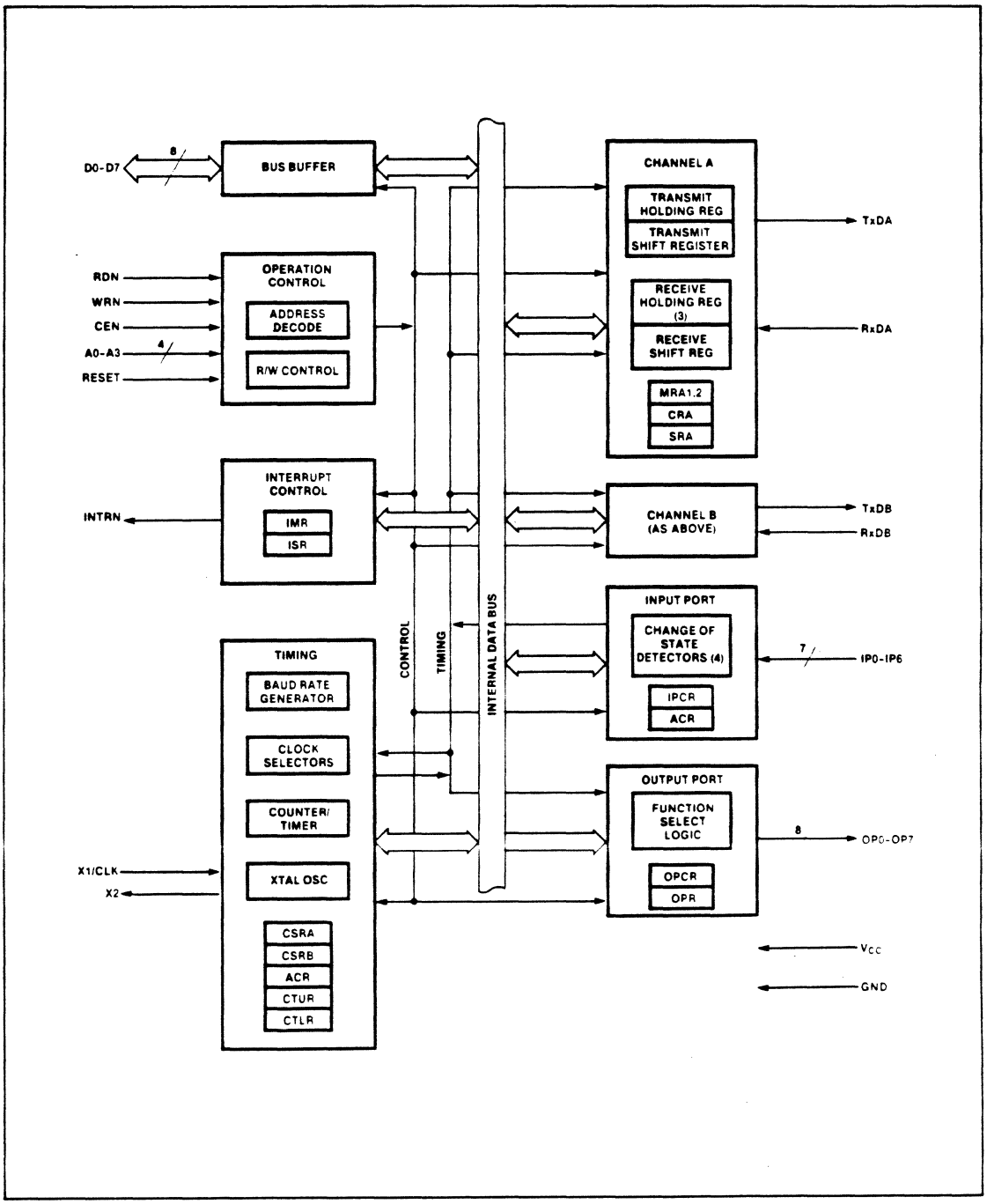
PACKAGES	$V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to 70°C		
	24 Pin[1]	28 Pin[2]	40 Pin[2]
Ceramic DIP	Not available	SCN2681AC1I28	SCN2681AC1I40
Plastic DIP	SCN2681AC1N24	SCN2681AC1N28	SCN2681AC1N40

[1]400 mil wide DIP
[2]600 mil wide DIP

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

BLOCK DIAGRAM



DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

PIN DESIGNATION

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
D0-D7	X	X	X	I/O	**Data Bus:** Bidirectional 3-state data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CEN	X	X	X	I	**Chip Enable:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0-D7 as controlled by the WRN, RDN and A0-A3 inputs. When high, places the D0-D7 lines in the 3-state condition.
WRN	X	X	X	I	**Write Strobe:** When low and CEN is also low, the contents of the data bus is loaded into the addressed register. The transfer occurs on the rising edge of the signal.
RDN	X	X	X	I	**Read Strobe:** When low and CEN is also low, causes the contents of the addressed register to be presented on the data bus. The read cycle begins on the falling edge of RDN.
A0-A3	X	X	X	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESET	X	X	X	I	**Reset:** A high level clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), puts OP0-OP7 in the high state, stops the counter/timer, and puts channels A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
INTRN	X	X	X	O	**Interrupt Request:** Active low, open drain, output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
X1/CLK	X	X	X	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. When a crystal is used, a capacitor must be connected from this pin to ground (see figure 5).
X2	X	X		O	**Crystal 2:** Connection for other side of the crystal. Should be connected to ground if a crystal is not used. When a crystal is used, a capacitor must be connected from this pin to ground (see figure 5).
RxDA	X	X	X	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
RxDB	X	X	X	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	X	X	X	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	X	X	X	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	X	X		O	**Output 0:** General purpose output, or channel A request to send (RTSAN, active low). Can be deactivated on receive or transmit.
OP1	X	X		O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low). Can be deactivated on receive or transmit.
OP2	X			O	**Output 2:** General purpose output, or channel A transmitter 1X or 16X clock output, or channel A receiver 1X clock output.
OP3	X			O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1X clock output, or channel B receiver 1X clock output.
OP4	X			O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.
OP5	X			O	**Output 5:** General purpose output, or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	X			O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output.
OP7	X			O	**Output 7:** General purpose output, or channel B open drain, active low, TxRDYB output.
IP0	X			I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN).
IP1	X			I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN).
IP2	X	X		I	**Input 2:** General purpose input, or counter/timer external clock input.
IP3	X			I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

PIN DESIGNATION (Continued)

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
IP4	X			I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	X			I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP6	X			I	**Input 6:** General purpose input or channel B receiver external clock input (RxCB). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
V_{CC}	X	X	X	I	**Power Supply:** +5V supply input
GND	X	X	X	I	**Ground**

BLOCK DIAGRAM

The 2681 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions.

Outputs OP3-OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4K baud. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection, for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Communications Channels A and B

Each communications channel of the 2681 comprises a full duplex asynchronous receiver/transmitter (UART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The inputs to this unlatched 7-bit port can be read by the CPU by performing a read operation at address D_{16}. A high input results in a logic 1 while a low input results in a logic 0. D_7 will always be read as a logic 1. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs lasting longer than 25-50μs will set the corresponding bit in the input port will change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice-versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E_{16} with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F_{16} with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The 2681 is conditioned to transmit data when the transmitter is enabled through the command register. The 2681 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted. If it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted, if the transmitter has been disabled.

Receiver

The 2681 is conditioned to receive data when enabled through the command register. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least sigificant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set upon receipt of the start bit of the new (overruning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to '11' for channels A and B respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, with receivers that are normally disabled, examine the received data stream and 'wake-up' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x switches the pointer to MR2x. The pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to table 2 for register bit descriptions.

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select — This bit selects either the channel A receiver ready status (RXRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select — This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the ac-

Table 1 2681 REGISTER ADDRESSING

A3	A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	RX Holding Register A (RHRA)	TX Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	RX Holding Register B (RHRB)	TX Holding Register B (THRB)
1	1	0	0	*Reserved*	*Reserved*
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

Table 2 REGISTER BIT FORMATS

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RX RTS CONTROL	RX INT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHAR.	
MR1A MR1B	0 = no 1 = yes	0 = RXRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = multi-drop mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	CHANNEL MODE		Tx RTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A MR2B	00 = Normal 01 = Auto echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

*Add 0.5 to values shown for 0-7 if channel is programmed for 5 bits/char.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CSRA CSRB	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
	See text				See text			

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CRA CRB	not used— must be 0	MISCELLANEOUS COMMANDS			DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
		See text			0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA SRB	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits 4:0. These bits are cleared by a 'reset error status' command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1 = TxRDYB	0 = OPR[6] 1 = TxRDYA	0 = OPR[5] 1 = RxRDY/ FFULLB	0 = OPR[4] 1 = RxRDY/ FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB (1X) 11 = RxCB (1X)		00 = OPR[2] 01 = TxCA (16X) 10 = TxCA (1X) 11 = RxCA (1X)	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	DELTA IP3	DELTA IP2	DELTA IP1	DELTA IP0	IP3	IP2	IP1	IP0
IPCR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

Table 2 REGISTER BIT FORMATS (Continued)

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ISR	INPUT PORT CHANGE	DELTA BREAK B	RxRDY/ FFULLB	TxRDYB	COUNTER READY	DELTA BREAK A	RxRDY/ FFULLA	TxRDYA
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IMR	IN. PORT CHANGE INT	DELTA BREAK B INT	RxRDY/ FFULLB INT	TxRDYB INT	COUNTER READY INT	DELTA BREAK A INT	RxRDY/ FFULLA INT	TxRDYA INT
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTUR	C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTLR	C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

cumulation (logical OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select — If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select — This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits per Character Select — This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select — Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is relocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-Send Control — If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select — This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1-9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1-1/16 to 2 stop bits can be programmed in increments of 1/16 bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1X clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select — This field selects the baud rate clock for the channel A receiver as follows:

	Baud Rate CLOCK = 3.6864MHz	
CSRA[7:4]	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4K.	19.2K
1 1 0 1	Timer	Timer
1 1 1 0	IP4—16X	IP4—16X
1 1 1 1	IP4—1X	IP4—1X

The receiver clock is always a 16X clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select — This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRA[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP3—16X	IP3—16X
1 1 1 1	IP3—1X	IP3—1X

The transmitter clock is always a 16X clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select — This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[7:4]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP6—16X	IP6—16X
1 1 1 1	IP6—1X	IP6—1X

The receiver clock is always a 16X clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select — This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP5—16X	IP5—16X
1 1 1 1	IP5—1X	IP5—1X

The transmitter clock is always a 16X clock except for CSRB[3:0] = 1111.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands — The encoded value of this field may be used to specify a single command as follows:

CRA[6:4]	COMMAND
0 0 0	No command.
0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.
0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
1 1 0	Start break. Forces the TXDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
1 1 1	Stop Break. The TXDA line will go high (marking) within two bit times. TXDA will remain high for one bit time before the next character, if any, is transmitted.

CRA[3] — Disable Channel A Transmitter — This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter — Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver — This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver — Enables operation of the channel A receiver. If not in the special wakeup mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break — This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received: further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1x clock). When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error — This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error — This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error — This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA) — This bit will be set when the channel A transmitter underruns, i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA) — This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

SRA[1] — Channel A FIFO Full (FFULLA) — This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA) — This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, if after this read there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select — This bit programs the OP7 output to provide one of the following:

- The complement of OPR[7]
- The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select — This bit programs the OP6 output to provide one of the followng:

- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select — This bit programs the OP5 output to provide one of the following:

- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select — This bit programs the OP4 output to provide one of the following:

- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select — This field programs the OP3 output to provide one of the following:

- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open collector output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1X clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

OPCR[1:0] — OP2 Output Select — This field programs the OP2 output to provide one of the following:

- The complement of OPR[2]
- The 16X clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be a 1X clock if CSRA[3:0] = 1111.
- The 1X clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select — This bit selects one of two sets of baud rates to be generated by the BRG:

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05K, 1.2K, 2.4K, 4.8K, 7.2K, 9.6K, and 38.4K baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2K, 1.8K, 2.0K, 2.4K, 4.8K, 9.6K, and 19.2K baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in table 3.

Table 3 BAUD RATE GENERATOR CHARACTERISTICS CRYSTAL OR CLOCK = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16X CLOCK (KHz)	ERROR (PERCENT)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2K	307.2	0
38.4K	614.4	0

NOTE
Duty cycle of 16X clock is 50% ± 1%

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

ACR[6:4]—Counter/Timer Mode and Clock Source Select — This field selects the operating mode of the counter/timer and its clock source as shown in table 4.

ACR[3:0] — IP3, IP2, IP1, IPO Change of State Interrupt Enable — This field selects which bits of the Input Port Change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7]=1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

Table 4 ACR [6:4] FIELD DEFINITION

ACR[6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)
0 0 1	Counter	TXCA — 1X clock of channel A transmitter
0 1 0	Counter	TXCB — 1X clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)
1 0 1	Timer	External (IP2) divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change of State — These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State — These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initialized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status — This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

ISR[6] — Channel B Change in Break — This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B 'reset break change interrupt' command.

ISR[5] — Channel B Receiver Ready or FIFO Full — The function of this bit is programmed by MR1B[6]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[4] — Channel B Transmitter Ready — This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready — In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change in Break — This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready or FIFO Full — The function of this bit is programmed by MR1A[6]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready — This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3-OP7 or the reading of the ISR.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs respectively of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 0002_{16}. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A3-A0 = 1110) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR.

The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A3-A0 = 1111). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8-bits to the upper 8-bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2.0			V
V_{IH}	Input high voltage (X1/CLK)		4.0			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{OC}	Open collector output leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{CC}	Power supply current				150	mA

NOTES:
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6,7]

PARAMETER	TENTATIVE LIMITS			UNIT
	Min	Typ	Max	
Reset Timing (figure 1)				
t_{RES} RESET pulse width	1.0			µs
Bus Timing (figure 2)[8]				
t_{AS} A0–A3 setup time to RDN, WRN low	10			ns
t_{AH} A0–A3 hold time from RDN, WRN high	0			ns
t_{CS} CEN setup time to RDN, WRN low	0			ns
t_{CH} CEN hold time from RDN, WRN high	0			ns
t_{RW} WRN, RDN pulse width	225			ns
t_{DD} Data valid after RDN low			175	ns
t_{DF} Data bus floating after RDN high			100	ns
t_{DS} Data setup time before WRN high	100			ns
t_{DH} Data hold time after WRN high	20			ns
t_{RWD} High time between READs and/or WRITEs[9,10]	200			ns
Port Timing (figure 3)[8]				
t_{PS} Port input setup time before RDN low	0			ns
t_{PH} Port input hold time after RDN high	0			ns
t_{PD} Port output valid after WRN high			400	ns
Interrupt Timing (figure 4)				
t_{IR} INTRN (or OP3–OP7 when used as interrupts) high from:				
Read RHR (RXRDY/FFULL interrupt)			300	ns
Write THR (TXRDY interrupt)			300	ns
Reset command (delta break interrupt)			300	ns
Stop C/T command (counter interrupt)			300	ns
Read IPCR (input port change interrupt)			300	ns
Write IMR (clear of interrupt mask bit)			300	ns
Clock Timing (figure 5)				
t_{CLK} X1/CLK high or low time	100			ns
f_{CLK} X1/CLK frequency	2.0	3.6864	4.0	MHz
t_{CTC} CTCLK (IP2) high or low time	100			ns
f_{CTC} CTCLK (IP2) frequency	0		4.0	MHz
t_{RX} RxC high or low time	220			ns
f_{RX} RxC frequency (16X)	0		2.0	MHz
(1X)	0		1.0	MHz
t_{TX} TxC high or low time	220			ns
f_{TX} TxC frequency (16X)	0		2.0	MHz
(1X)	0		1.0	MHz
Transmitter Timing (figure 6)				
t_{TXD} TxD output delay from TxC low			350	ns
t_{TCS} TxC output skew from TxD output data	0		150	ns
Receiver Timing (figure 7)				
t_{RXS} RxD data setup time to RXC high	240			ns
t_{RXH} RxD data hold time from RXC high	200			ns

NOTES

4. Parameters are valid over specified temperature range
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters
7. Test condition for outputs $C_L = 150pF$, except interrupt outputs. Test condition for interrupt outputs $C_L = 50pF$, $R_L = 2.7K$ ohm to V_{CC}
8. Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the 'strobing' input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN.
9. If CEN is used as the 'strobing' input, this parameter defines the minimum high time between one CEN and the next.
10. Consecutive write operations to the same command register require at least three edges of the X1 clock between writes.