SANDIA REPORT

SAND87—2617 • UC—32
Unlimited Release
Printed November 1987

RS-8232-2/66555

8232-2//066555

cy1

00000001 -

SANDAC V Programmer's Reference Manual

C. R. Borgman, Editor

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

SF2900Q(8-81)

1132098

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors or subcontractors.

SAND87-2617
Unlimited Release
Printed November 1987

Distribution
Category UC-32

SANDAC V Programmer's Reference Manual

C. R. Borgman, Editor
Guidance and Control Division II

Sandia National Laboratories
Albuquerque, NM 87185

Abstract

This manual describes from a programmer's viewpoint the SANDAC V embedded computer designed and built at Sandia National Labs. System information is given in sufficient detail to allow a systems programmer to add to or replace software for SANDAC V.

Acknowledgment

The editor wishes to acknowledge the members of the SANDAC V hardware development team, whose efforts have resulted in the hardware described herein. This team has been led by Larry Dalton, and the members listed alphabetically have been Gerald Grafe, Don Johnson, Ken Kimball, K. K. Ma, Brent Meyer, Harold Ortiz, Steve Reynolds, Glenn Russell, H. Shen, and Tom Widney.

This document has been prepared by Tech. Reps., Inc., under contract to Sandia National Laboratories. My thanks go to the many technical writers, editors, typists, and illustrators on their staff who have made contributions to this document. Special thanks go to Terry Cameron, who has made the major contribution to writing this document.

Motorola, MC68020, MC68881, MC68901, and MC68450 are trademarks of Motorola, Inc.
Signetics and SCN68562 are trademarks of Signetics Corporation.
Hitachi and HD63450 are trademarks of Hitachi America.
BUS-63600 and BUS-65300 II are trademarks of ILC Data Device Corporation.

CONTENTS

1. SYSTEM ARCHITECTURE 1-1

1.1 ARCHITECTURAL OVERVIEW 1-1
1.1.1 Bus System 1-1
1.1.2 Processor Module 1-3
1.1.3 Utility Module 1-3
1.1.4 Memory Modules 1-4
1.1.4.1 Global Memory Module 1-4
1.1.4.2 Local Memory Expansion Module 1-4
1.1.5 System I/O Module 1-4
1.1.6 MIL-STD-1553B Module 1-5

1.2 CONFIGURATION OPTIONS 1-5

2. SYSTEM ADDRESS MAP 2-1

2.1 SYSTEM ADDRESS SPACE 2-1

2.2 ADDRESS MAP 2-1
2.2.1 Utility Module Address Space 2-3
2.2.2 Global Memory Address Space 2-3
2.2.3 Processor Module Address Space 2-3
2.2.3.1 Resident RAM and RAM Space 2-8
2.2.3.2 Local Memory Blocks 2-8
2.2.3.3 Extended I/O 2-8
2.2.3.4 68901 MFP 2-8
2.2.3.5 Configuration Registers 2-8
2.2.3.6 512-Byte FIFO 2-9
2.2.3.7 Command/Status Register 2-9
2.2.3.8 Hardware ID Register 2-9
2.2.3.9 Software Interrupts 2-9
2.2.3.10 System I/O Module 2-9
2.2.4 Local Processor Module Alias Space 2-9

3. PROCESSOR MODULE 3-1

3.1 68020 MPU 3-1
3.1.1 User-Level Programming Model 3-1
3.1.1.1 Data Registers (D0-D7) 3-4
3.1.1.2 Address Registers (A0-A6) 3-4

3.1.1.3 User Stack Pointer (A7 or USP) 3-4
3.1.1.4 Condition Code Register (CCR) 3-4
3.1.2 Supervisor-Level Programming Model 3-5
3.1.2.1 Status Register (SR) 3-5
3.1.2.2 Interrupt Stack Pointer (ISP) 3-7
3.1.2.3 Master Stack Pointer (MSP) 3-7
3.1.2.4 Vector Base Register (VBR) 3-7
3.1.2.5 Source Function Code (SFC) Register 3-7
3.1.2.6 Destination Function Code (DFC) Reg. 3-8
3.1.2.7 Cache Control Register (CACR) 3-8
3.1.2.8 Cache Address Register (CAAR) 3-9
3.1.3 68020 Initial State 3-10

3.2 68881 FLOATING-POINT COPROCESSOR 3-10
3.2.1 Programming Model 3-10
3.2.1.1 Floating-Point Data Registers (FP0-FP7) 3-12
3.2.1.2 Floating-Point Control Register (FPCR) 3-12
3.2.1.3 Floating-Point Status Register (FPSR) 3-13
3.2.1.4 Floating-Point Inst. Addr. Reg. (FPIAR) 3-15
3.2.2 68881 Initial State 3-15

3.3 HARDWARE IDENTIFICATION (HWID) REGISTER 3-15

3.4 CONFIGURATION REGISTERS 3-16
3.4.1 Bus Arbitration Mode Control 3-18
3.4.2 Address Monitor 3-18

3.5 RESIDENT RAM AND RAM SPACE 3-22

3.6 512-BYTE FIFO 3-22

3.7 COMMAND/STATUS REGISTER 3-23

3.8 68901 MFP 3-25
3.8.1 General-Purpose Timers 3-26
3.8.2 Global Serial Bus Interface 3-27
3.8.3 Programmable I/O Lines 3-28
3.8.4 Interrupt Controller 3-28
3.8.5 Programming Model 3-29

3.9 INTERRUPT SYSTEM 3-41
3.9.1 Interrupts to the 68020 3-41
3.9.1.1 512-Byte FIFO Interrupt 3-41
3.9.1.2 Local Expansion Bus Interrupts 3-43
3.9.1.3 Global Bus Interrupts 3-43

3.9.1.4 Software Interrupts 3-43
3.9.1.5 68901 Interrupt 3-44
3.9.2 Interrupts Directed to the 10x7 Crossbar 3-46
3.9.2.1 Autovector Sources 3-49
3.9.2.2 Vectored Sources 3-49
3.9.3 Interrupts Generated by the Processor Module 3-49

3.10 THE LOCAL EXPANSION BUS INTERFACE 3-50

4. UTILITY MODULE 4-1

4.1 EPROM 4-1

4.2 RAM 4-3

4.3 32-BIT TIMER 4-3

4.4 STATUS REGISTER 4-5

5. MEMORY MODULES 5-1

5.1 GLOBAL MEMORY MODULE 5-1
5.1.1 RAM Storage 5-1
5.1.2 Access Monitor 5-1

5.2 LOCAL MEMORY EXPANSION MODULE 5-6

6. SYSTEM I/O MODULE 6-1

6.1 LOCAL BUS INTERFACE 6-1

6.2 PARALLEL PORT 6-3

6.3 68562 DUSCCs 6-3

6.4 68901 MFP 6-24
6.4.1 Timers 6-24
6.4.2 USART 6-24
6.4.3 Programmable I/O Lines 6-24
6.4.4 Interrupt Controller 6-25
6.4.5 Programming Model 6-26

6.5 63450 DMAC 6-39
6.5.1 Implementation 6-39
6.5.2 Addressing Modes 6-39
6.5.3 Restrictions 6-40
6.5.4 Programming Model 6-40

6.6 INTERRUPTS TO THE PROCESSOR MODULE 6-54

7. MIL-STD-1553B MODULE 7-1

7.1 ARCHITECTURAL OVERVIEW 7-1
7.1.1 Local Expansion Bus Interface 7-3
7.1.2 4K x 16-Bit RAM 7-3
7.1.3 BUS-66300 II 7-4
7.1.3.1 Configuration Register 7-6
7.1.3.2 Start/Reset Register 7-8
7.1.3.3 Interrupt Mask Register 7-9
7.1.3.4 Remote Terminal Address Register 7-10
7.1.3.5 Prescale/Counter Enable Register 7-10
7.1.3.6 Tag/Count Register 7-11
7.1.3.7 Interrupt Vector Register 7-11
7.1.4 BUS-65600 7-12

7.2 OPERATION AS A BUS CONTROLLER 7-12
7.2.1 BC Mode RAM Partitioning 7-12
7.2.2 BC Mode RAM Initialization 7-15
7.2.2.1 Stack Pointers A, B 7-15
7.2.2.2 Message Counters A, B 7-15
7.2.2.3 Descriptor Stacks A, B 7-15
7.2.2.4 Message Blocks 7-17
7.2.3 BC Mode BUS-66300 II Initialization 7-21

7.3 OPERATION AS A REMOTE TERMINAL UNIT 7-22
7.3.1 RTU Mode RAM Partitioning 7-22
7.3.2 RTU Mode RAM Initialization 7-25
7.3.2.1 Look-up Tables 7-25
7.3.2.2 Descriptor Stacks A, B 7-25
7.3.2.3 Data Blocks 7-26
7.3.3 RTU Mode BUS-66300 II Initialization 7-26
7.3.4 Illegal Mode Codes 7-26

7.4 OPERATION AS A BUS MONITOR 7-27
7.4.1 MT Mode RAM Partitioning 7-27
7.4.2 MT Mode RAM Initialization 7-27

7.4.3 MT Mode BUS-66300 II Initialization 7-29

7.5 INTERRUPTS TO THE PROCESSOR MODULE 7-29

8. BUS ERRORS 8-1

8.1 FIFO WRITE WHEN FULL 8-1

8.2 MONITORED LOCAL MEMORY ADDRESS 8-1

8.3 MONITORED GLOBAL MEMORY BLOCK 8-3

8.4 BREAKPOINT INSTRUCTION 8-3

8.5 WATCHDOG TIMER TIMEOUT 8-3

8.6 WRITE TO EPROM 8-3

8.7 WRITE TO UTILITY MODULE STATUS REGISTER 8-3

8.8 USER-LEVEL WRITE TO 32-BIT TIMER 8-4

9. OPERATIONAL CONCEPTS 9-1

9.1 POWERUP SEQUENCE 9-1

9.2 APPLICATION ENVIRONMENT 9-1
9.2.1 68020 ISP Initialization 9-3
9.2.2 Extant Processor Table 9-3
9.2.3 Global RAM Bitmap 9-4
9.2.4 Sleeping Processor Modules 9-4

9.3 EXCEPTIONS DURING POWERUP 9-4

9.4 ACCESSING RESOURCES 9-4
9.4.1 Global Bus Arbitration 9-5
9.4.2 Bus Mastership 9-6
9.4.3 Access Time 9-6

APPENDIX: GATAR AND HAWK RESOURCE UTILIZATION A-1

A.1 INTRODUCTION A-1

A.1.1 GATAR Software Debugger A-1
A.1.2 Hawk Operating System Kernel A-1

A.2 RESOURCES USED BY GATAR AND HAWK A-1
A.2.1 GATAR Memory Utilization A-1
A.2.2 Hawk Memory Utilization A-2
A.2.5 Hawk Local Resource Utilization A-9
A.2.6 Hawk Interrupt Utilization A-9
A.2.7 Exception Vectors (GATAR and Hawk) A-9

LIST OF FIGURES

Figure 1-1. SANDAC V Architecture 1-2

Figure 2-1. SANDAC V Address Map 2-2
Figure 2-2. Utility Module Address Map 2-4
Figure 2-3. Global Memory Address Map 2-5
Figure 2-4. Processor Module Address Map 2-6
Figure 2-5. Local Resource Address Map 2-7
Figure 2-6. System I/O Module Address Map 2-11

Figure 3-1. Processor Module Architecture 3-2
Figure 3-2. 68020 User-Level Programming Model 3-3
Figure 3-3. 68020 Supervisor-Level Programming Model 3-6
Figure 3-4. 68881 Floating-Point Coproc. Programming Model 3-11
Figure 3-5. General-Purpose Timer Hardware Configuration 3-26
Figure 3-6. Global Serial Bus Interface 3-27
Figure 3-7. Processor Module 68901 Programming Model 3-31
Figure 3-8. Processor Module Interrupt System 3-42

Figure 4-1. Utility Module Architecture 4-2
Figure 4-2. 32-Bit Timer 4-4

Figure 5-1. Global Memory Module 5-2

Figure 6-1. System I/O Module Architecture 6-2
Figure 6-2. System I/O Module 68901 MFP Programming Model 6-28
Figure 6-3. DMA Addressing Modes 6-41
Figure 6-4. 63450 DMAC Programming Model 6-42
Figure 6-5. System I/O Module Interrupt System 6-55

Figure 7-1. MIL-STD-1553B Module Architecture 7-2
Figure 7-2. BUS-66300 II Programming Model 7-5
Figure 7-3. BC Mode Memory Management 7-14
Figure 7-4. BC Mode Message Block Formats 7-18
Figure 7-5. RTU Mode Memory Management 7-24

Figure 9-1. SANDAC V Powerup Sequence 9-2

Figure A-1. Utility Module EPROM Utilization A-3
Figure A-2. Utility Module RAM Utilization A-4
Figure A-3. Processor Module RAM Utilization A-5
Figure A-4. Interrupt System as Initialized by GATAR A-8

LIST OF TABLES

TABLE 3-1. CONFIGURATION REGISTER SUMMARY 3-17
TABLE 3-2. PROCESSOR MODULE 68901 MFP ADDRESS MAP 3-30

TABLE 4-1. UTILITY MODULE ADDRESSES 4-1

TABLE 5-1. GLOBAL MEMORY ADDRESSES 5-3
TABLE 5-2. ACCESS MONITOR COVERAGE 5-4
TABLE 5-3. LOCAL MEMORY EXPANSION ADDRESSES 5-6

TABLE 6-1. SYSTEM I/O MODULE ADDRESSES 6-3
TABLE 6-2. 68562 DUSCC ADDRESS MAP 6-5
TABLE 6-3. 68562 DUSCC REGISTER ADDRESSING 6-6
TABLE 6-4. SYSTEM I/O MODULE 68901 MFP ADDRESS MAP 6-27
TABLE 6-5. 63450 DMAC REGISTER ADDRESSING 6-43
TABLE 6-6. SYSTEM I/O MODULE INTERRUPT STRUCTURE 6-54

TABLE 7-1. MIL-STD-1553B MODULE ADDRESSES 7-3
TABLE 7-2. BUS-66300 II ADDRESS MAP 7-4
TABLE 7-3. 4K X 16-BIT RAM PARTITIONING FOR BC MODE 7-13
TABLE 7-4. 4K X 16-BIT RAM PARTITIONING FOR RTU MODE 7-23
TABLE 7-5. 4K X 16-BIT RAM PARTITIONING FOR MT MODE 7-28

TABLE 8-1. SANDAC V BUS ERROR SUMMARY 8-2

TABLE 9-1. WAIT STATE SUMMARY 9-7
TABLE 9-2. SUMMARY OF RESET ACTIVITY 9-9

TABLE A-1. SANDAC V EXCEPTION VECTORS A-11

1. SYSTEM ARCHITECTURE

1.1 ARCHITECTURAL OVERVIEW

The SANDAC V modular architecture lets the user configure a target system with the appropriate type and number of hardware modules for a specific application. The types of modules available for system configuration include the following:

- Processor
- Utility
- Global Memory
- Local Memory Expansion
- System I/O
- MIL-STD-1553B

Within a target system, Processor Modules, the one required Utility Module, and Global Memory Modules reside on the global bus. Local Memory Expansion Modules, System I/O Modules, and MIL-STD-1553B Modules reside on the local bus associated with a Processor Module. These modules physically connect to a Processor Module's local bus, thus extending or expanding it. The modules are referred to generally as expansion modules, and the extended local bus is designated the local expansion bus. A third bus, the global serial bus, connects all Processor Modules to an external diagnostic port located on the Utility Module.

The SANDAC V general architecture is illustrated in Figure 1-1. The bus system and the hardware modules are described in the following paragraphs.

1.1.1 Bus System

The SANDAC V bus system contains the following components:

- Global bus
- Local buses associated with each Processor Module
- Global serial bus

The global bus interconnects all Processor Modules, the Utility Module, and any Global Memory Modules. To avoid contention problems when more than one Processor Module at a time tries to use this bus, an arbitration system gives bus control to the module with the highest priority, which is determined by hardware strapping (refer to Subsection 3.3).

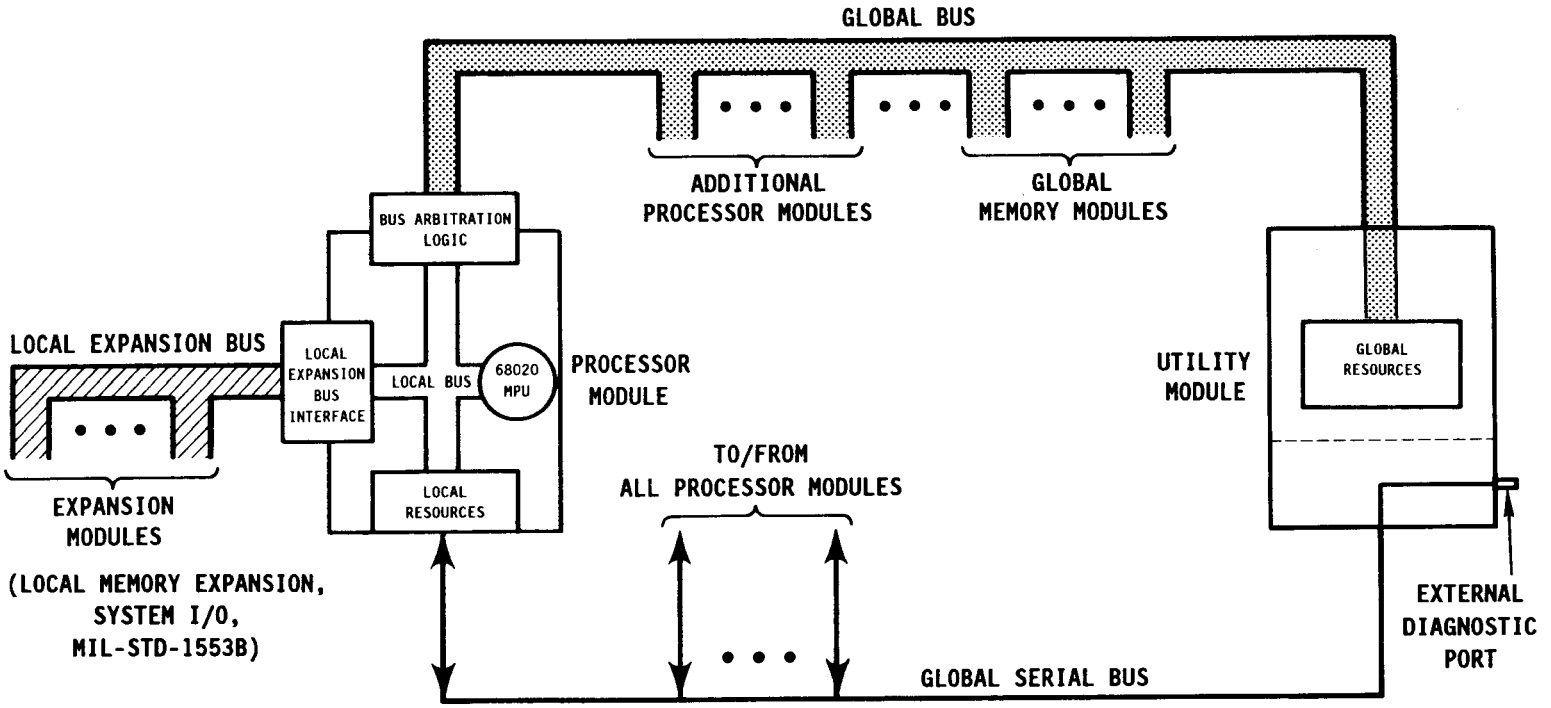


Figure 1-1. SANDAC V Architecture

The local bus on each Processor Module provides a Processor Module's 68020 MPU with access to memory and I/O resources resident on the module. Additional memory and I/O can be added to the local bus by attaching SANDAC V Local Memory Expansion, System I/O, or MIL-STD-1553B Modules to the Processor Module. The resources provided by any attached modules are said to reside on the local expansion bus.

The global serial bus connects all Processor Modules to an external diagnostic port located on the Utility Module. This port is a path for loading application code into the SANDAC V and also serves as a window into the system for debug operations. The SANDAC V software development system's Gateway terminal interfaces with the target system via this port. Software running on the Gateway communicates with individual Processor Modules over the global serial bus and loads application code or performs debug operations.

1.1.2 Processor Module

The Processor Module is an independent processing unit that employs the Motorola MC68020 Microprocessor Unit (MPU) complemented with the Motorola MC68881 Floating-Point Coprocessor. Processing activity is supported with a variety of local resources supplied by the following:

- 128K of RAM
- 1 VLSI Gate Array
- 1 Motorola MC68901 Multifunction Peripheral

The module resides on the SANDAC V global bus. Access to global resources is through the global bus interface, which includes bus arbitration logic. Additionally, the local memory and I/O resources can be expanded by attaching as many as four additional SANDAC V modules to the local expansion bus, including Local Memory Expansion, System I/O, or MIL-STD-1553B Modules. Section 3 describes the Processor Module.

1.1.3 Utility Module

One Utility Module is required for a SANDAC V. The module provides the system with power and clocks, as well as the following global resources:

- 256K of EPROM
- 128K of RAM
- 32-bit timer
- Status register

Section 4 describes the Utility Module.

1.1.4 Memory Modules

SANDAC V has two types of memory modules: the Global Memory Module that attaches to the global bus and the Local Memory Expansion Module that attaches to a Processor Module's local expansion bus.

1.1.4.1 Global Memory Module. The Global Memory Module provides 512K of battery-backed RAM for global bus storage. The module includes an access monitor that provides notification if an overwrite occurs in a user-selected area of the RAM. The size of the monitored area is selectable from 128 bytes to 256K bytes. SANDAC V has address space for 30 Global Memory Modules; however, the number actually installed in a system is subject to the limitations discussed in Subsection 1.2. Section 5 describes the Global Memory Module.

1.1.4.2 Local Memory Expansion Module. The Local Memory Expansion Module provides 2M of battery-backed RAM for local bus storage. The access monitor included on the host Processor Module provides monitoring for attached Local Memory Expansion Modules. A Processor Module has address space reserved for six of these modules, but because of electrical drive limitations, only four modules can be supported. Section 5 describes the Local Memory Expansion Module.

1.1.5 System I/O Module

The System I/O Module (Section 6) attaches to a Processor Module's local expansion and provides the following I/O resources:

- 1 parallel port
- 5 async/sync serial channels
- 4 16-bit counter/timers
- 4 8-bit counter/timers
- 2 baud rate generators
- 8 input/output/interrupt lines
- 8 input lines
- 8 output lines
- 4 DMA channels

To support these resources, the module features the following peripheral devices:

- 2 Signetics SCN68562 Dual Universal Serial Communications Controllers
- 1 Motorola MC68901 Multifunction Peripheral
- 1 Hitachi HD63450 Direct Memory Access Controller

1.1.6 MIL-STD-1553B Module

The MIL-STD-1553B Module attaches to a Processor Module's local expansion bus and provides a dual-redundant interface for the MIL-STD-1553 command/response data bus. To support this interface, the module features the following devices:

- BUS-66300 II MIL-STD-1553-to-Microprocessor Interface Unit
- BUS-65600 MIL-STD-1553 Bus Controller, Remote Terminal, Bus Monitor
- 4K x 16-bit RAM

Section 7 describes the MIL-STD-1553B Module.

1.2 CONFIGURATION OPTIONS

Because of signal propagation delays, a SANDAC V target system is limited to a maximum of 20 interconnected hardware modules. This total includes all global bus modules (Utility Module, Processor Modules, and Global Memory Modules) and all modules connected to Processor Module local expansion buses (Local Memory Expansion Modules, System I/O Modules, or MIL-STD-1553B Modules).

In configuring a system, the following considerations apply:

- One Utility Module is required for any configuration.

- The global bus supports a maximum of 15 Processor Modules.

- A Processor Module can have a maximum of four expansion modules attached to its local expansion bus. The modules can include the following module types in the quantity indicated:

    - 1 System I/O Module
    - As many as 4 Local Memory Expansion Modules
    - As many as 4 MIL-STD-1553B Modules

NOTES

2. SYSTEM ADDRESS MAP

2.1 SYSTEM ADDRESS SPACE

SANDAC V has total address space of 4 gigabytes, which is the addressing capability (32 address bits) of the Processor Module's 68020 MPU. Although the address space contains global resources on the global bus and local resources on Processor Module local buses, *the address space for the entire system is linear.* By issuing the proper address, any Processor Module in the system can access any global or local resource, including any resource on the global bus, its own local resources, or the local resources of any other Processor Module.

2.2 ADDRESS MAP

The SANDAC V address map is shown in Figure 2-1. The map is divided into the following major address blocks:

- Utility Module address space
- Global memory address space
- Processor Module address space
- Local Processor Module alias space

The address partitioning within these areas is described in the following paragraphs.

NOTE

SANDAC V's 8-digit hex addresses are specified throughout this document as follows:

XX,XXX,XXX

This convention visually aids in reading the addresses and also has a logical connection to the memory mapping of the hardware, i.e., Processor Module 1 address space starts at 01,000,000 hex; Processor Module 2 address space starts at 02,000,000 hex; etc. This is a convention for the document only; the commas in the format are not to be input to software tools.

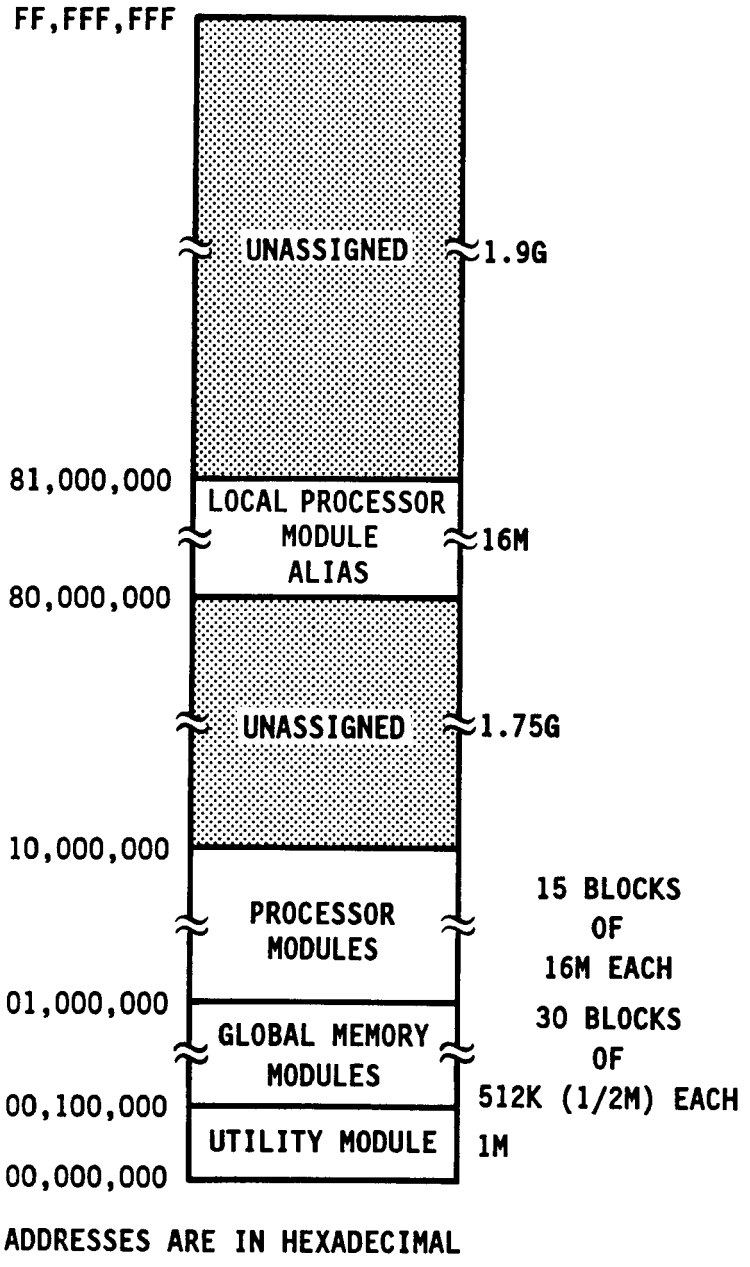


Figure 2-1. SANDAC V Address Map

2.2.1 Utility Module Address Space

The global resources provided by the Utility Module reside in the first megabyte of SANDAC V address space. The address map for this space is shown in Figure 2-2.

In the 256K space reserved for EPROM, either a 128K or 256K device can be used. When a 128K device is used, addresses to the upper 128K locations access the corresponding location in the lower 128K space.

Utility Module address space also includes a 128K block of RAM 32 bits wide.

Spaces of 64K each are reserved for both the 32-bit timer and the Utility Module status register. Both of these resources consist of only a single register, but any address issued within their specified spaces accesses them. Detailed descriptions of these resources are given in Section 4.

2.2.2 Global Memory Address Space

Thirty blocks of 512K each are reserved as global memory space. The size of the individual blocks (1/2 megabyte) corresponds to the storage capacity of a single Global Memory Module. The addresses that a Global Memory Module responds to are determined by hardware jumpers located on the module; thus, these modules can be placed in any of the reserved blocks. The address map for global memory is shown in Figure 2-3.

2.2.3 Processor Module Address Space

SANDAC V can be configured with as many as 15 Processor Modules. A 16 megabyte block of address space is reserved for each of the 15 possible modules. The particular block used by a module is determined by the Processor Module identification (ID) number. The four-bit ID, from 0001 (1) through 1111 (15), is set with four hardware jumpers. The four bits that form the ID also correspond to bits 28 through 25 of the module's address space. Therefore, the 16M space assigned to each Processor Module is defined by starting address 0n,000,000 hex and ending address 0n,FFF,FFF hex, where "n" is the Processor Module ID (Processor ID) in hex. The address map for Processor Modules is shown in Figure 2-4.

A Processor Module's local resources reside within its defined address space. The local resource address map, which is the same for all Processor Modules, is shown in Figure 2-5. The address assignments are briefly described in the paragraphs that follow.

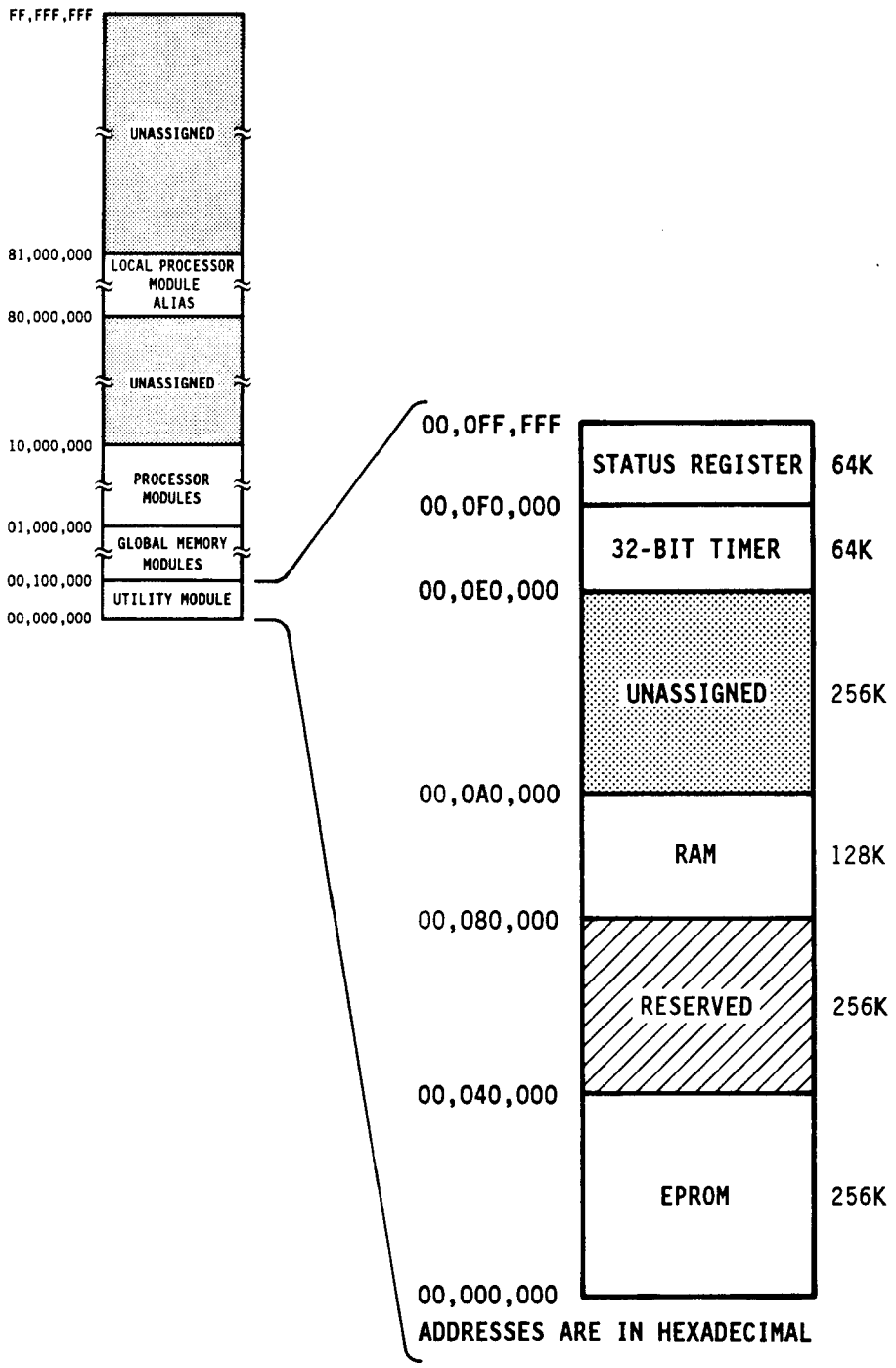


Figure 2-2. Utility Module Address Map

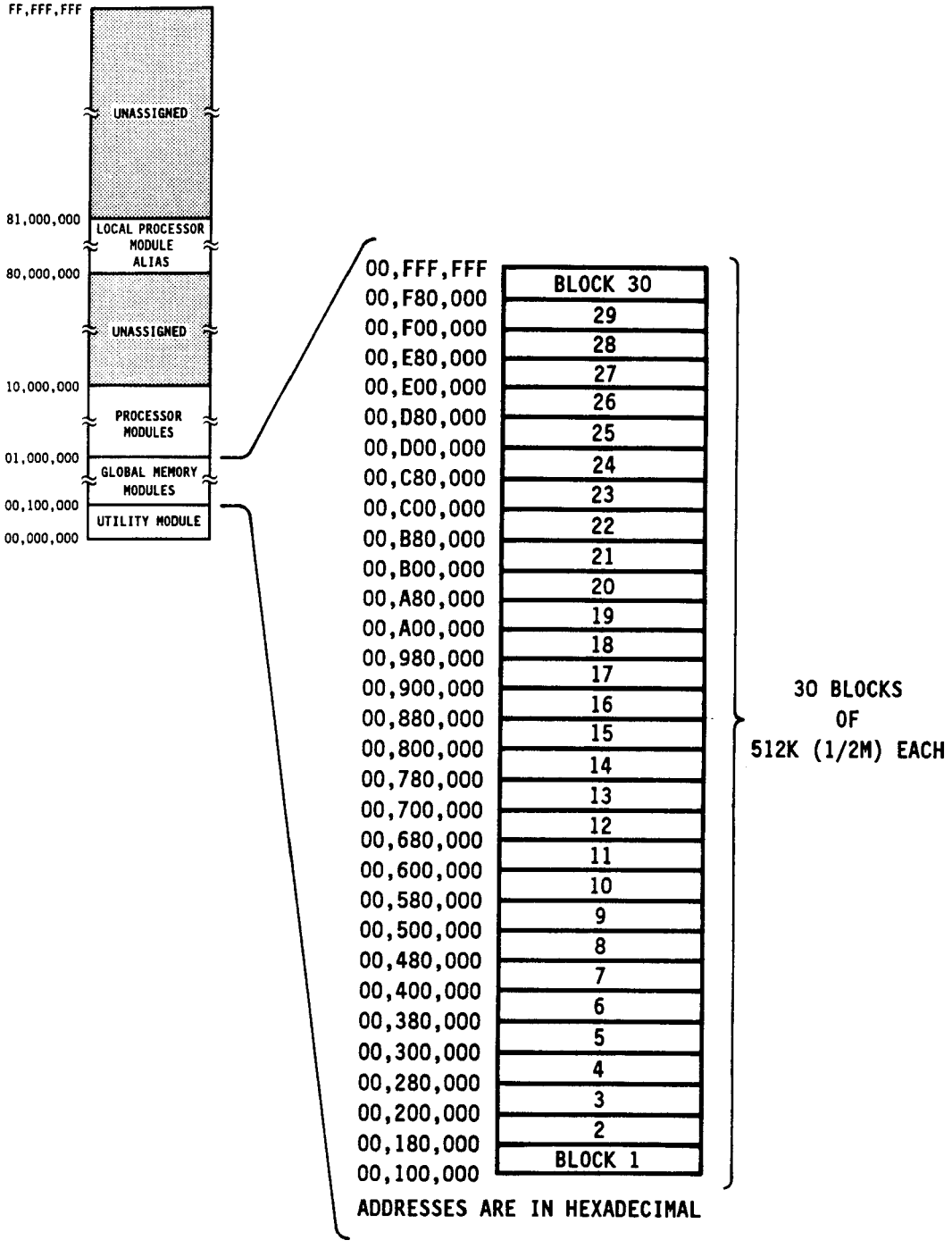


Figure 2-3. Global Memory Address Map

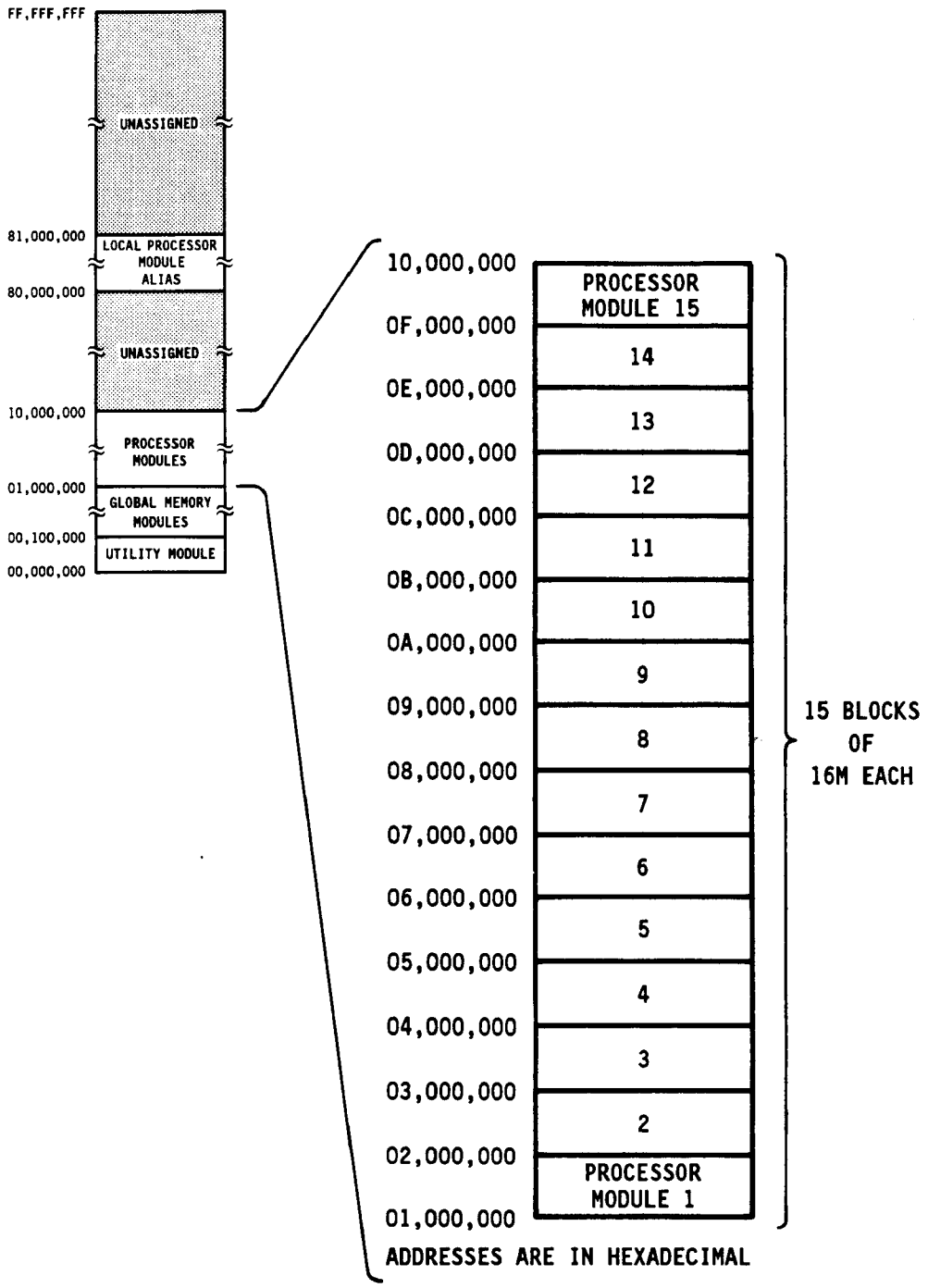


Figure 2-4. Processor Module Address Map

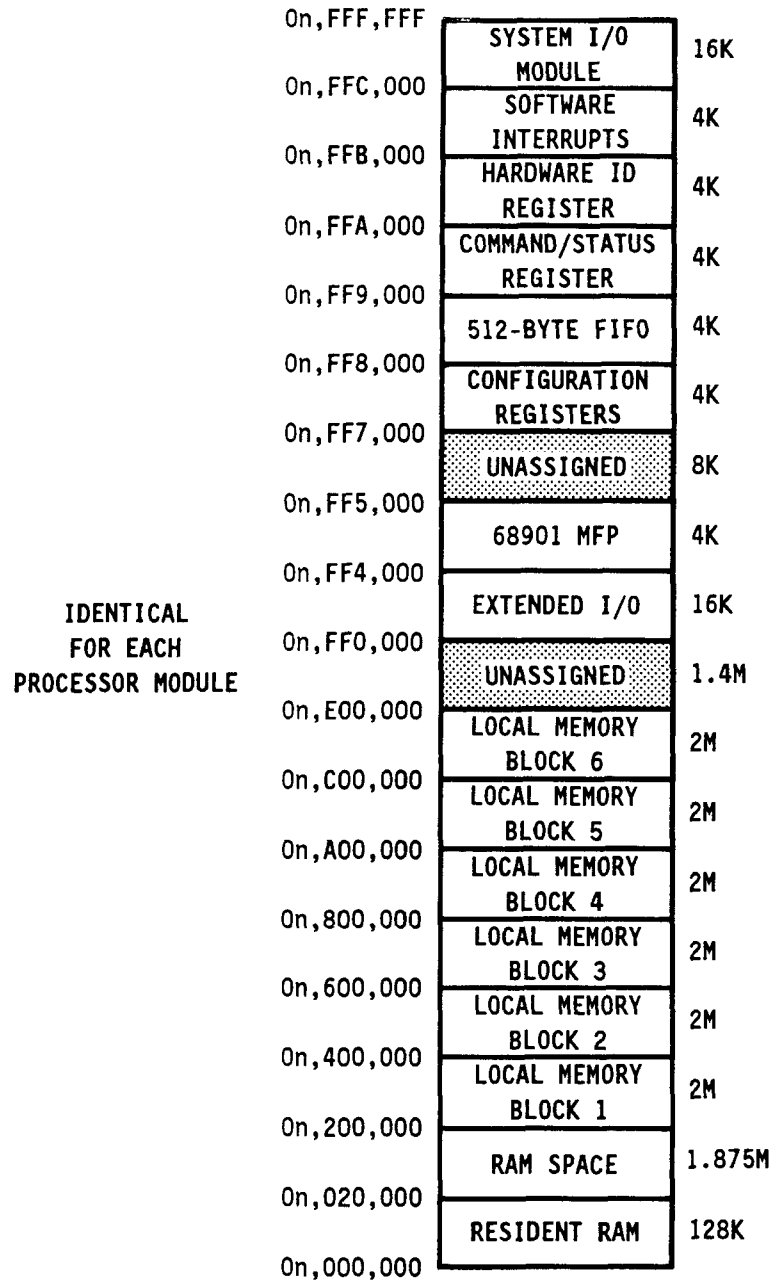


Figure 2-5. Local Resource Address Map

2.2.3.1 Resident RAM and RAM Space. The first 128K of Processor Module (local resource) address space is occupied by RAM resident on the Processor Module. The next 1.875M of space is reserved as RAM space, providing expansion capability for the resident RAM. The RAM space is logically divided into 128K blocks. Addresses issued to locations in the RAM space blocks access the corresponding locations in the 128K resident RAM block.

2.2.3.2 Local Memory Blocks. Six blocks of 2M each are reserved for Local Memory Expansion Modules, which attach to the local expansion bus. The size of the individual blocks corresponds to the storage capacity (2M) of a single module.

In configuring a system, as many as four Local Memory Expansion Modules can be attached to a Processor Module. The addresses that a module responds to are determined by hardware jumpers; thus, these modules can be placed in any of the reserved blocks.

2.2.3.3 Extended I/O. The 16K extended I/O space is an area that I/O modules may be designed into. When this space is accessed, a "chip select" is generated. Modules in this space need only provide address decoding for the lower 14 bits of address. The MIL-STD-1553B Module (refer to Section 7) is located in this address space.

2.2.3.4 68901 MFP. A 4K space is reserved for the 68901 Multifunction Peripheral (MFP) resident on the Processor Module. The 68901 itself contains only 24 addressable registers. The base address for the 68901 is 0n,FF4,000 hex, where "n" is the Processor ID in hex. Specific addresses for the 68901 register set are given in Section 3.

2.2.3.5 Configuration Registers. A 4K space is reserved for the group of 15 registers in the Processor Module's gate array. Collectively, these registers are referred to as the configuration registers because they establish Processor Module operating conventions. Individually or in groups, the configuration registers support the following Processor Module features:

- Bus arbitration
- Address monitoring
- 68020 interrupt priorities
- 68901 input selection

The base address for the register group is 0n,FF7,000 hex, where "n" is the Processor ID in hex. The specific addresses for the registers within the group are given in Section 3.

2.2.3.6 512-Byte FIFO. A 4K space is reserved for the 512-byte FIFO. The FIFO represents only one addressable location but responds to any address within the space, beginning with 0n,FF8,000 hex, where "n" is the Processor ID in hex. The 512-byte FIFO is described in Section 3.

2.2.3.7 Command/Status Register. A 4K space is reserved for the Command/Status Register. The register represents only one addressable location but responds to any address within the space, beginning with 0n,FF9,000 hex, where "n" is the Processor ID in hex. The Command/Status Register is described in Section 3.

2.2.3.8 Hardware ID Register. A 4K space is reserved for the Hardware ID Register. The register represents only one addressable location but responds to any address within the space, beginning with 0n,FFA,000 hex, where "n" is the Processor ID in hex. The Hardware ID Register is described in Section 3.

2.2.3.9 Software Interrupts. A 4K space is reserved for the Processor Module's software-generated interrupt requests. Eight separate requests can be generated by writing to eight addresses in this address space. The write operation itself generates the interrupt request; there are no registers associated with this function. The software interrupts are described in Section 3.

2.2.3.10 System I/O Module. A 16K area is reserved for a System I/O Module, which attaches to the local expansion bus. The resources provided by the System I/O Module are assigned to specific addresses within the 16K as shown in Figure 2-6. The System I/O Module is described in Section 6.

2.2.4 Local Processor Module Alias Space

The 16 megabyte local Processor Module alias at 80,000,000 hex is a logical entity only; no physical memory is present. It is included in the SANDAC V address map to support software development in a multiple processor environment.

When a Processor Module issues an address that is within the 16M alias space, the bus cycles generated access the corresponding address in that Processor Module's 16M local resource address space. For example, if the Processor Module with the ID of 1 issues an address for location 80,000,1FE hex, the actual access is of location 01,000,1FE hex; a location in its own address space. By the same token, if the Processor Module with the ID of 2 issues an address for location 80,000,1FE hex, the actual access is of location 02,000,1FE hex, a location in *its* own address space.

The local Processor Module alias allows application programs for individual Processor Modules to be written without regard to what specific IDs they will be given when the system is implemented; all code is simply linked within the alias space (80,000,000+ hex). Additionally, at powerup, all Processor Modules execute the same code out of the Utility Module's global ROM. Implementing this code with data references in the alias space (80,000,000+ hex) provides each Processor Module with a copy of the data variables. This allows each to execute the same code simultaneously without conflicts.

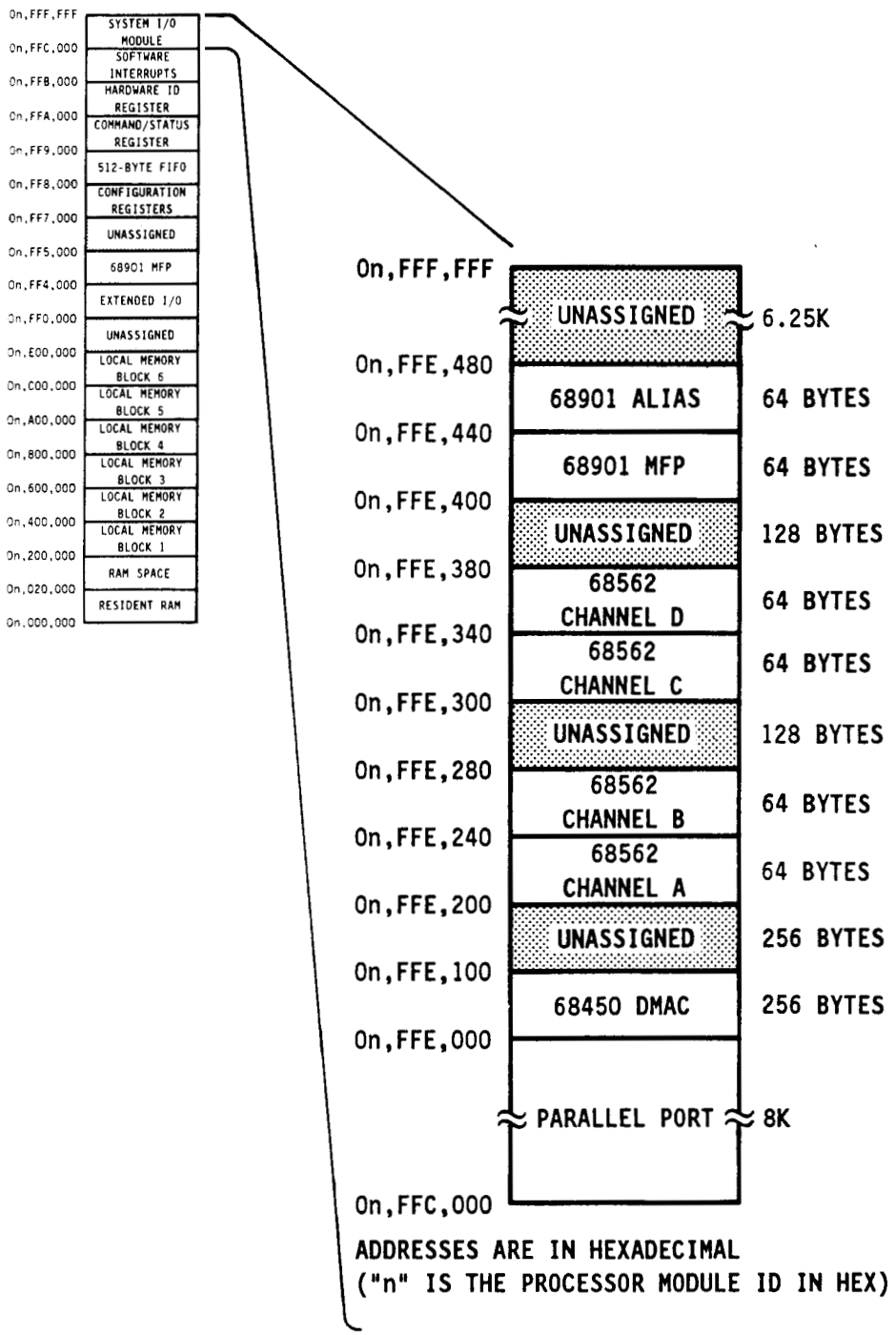


Figure 2-6. System I/O Module Address Map

NOTES

3. PROCESSOR MODULE

The architecture of the Processor Module is shown in Figure 3-1. This module provides the following resources:

- MC68020 Microprocessor Unit (MPU)
- MC68881 Floating-Point Coprocessor
- Hardware Identification (HWID) Register
- Configuration registers (15)
- 128K of RAM (expandable)
- 512-byte First-In-First-Out (FIFO) Buffer
- Command/Status Register
- MC68901 Multi-Function Peripheral
- Programmable interrupt system
- Local expansion bus interface

The 15 configuration registers support bus arbitration, an address monitor, and the programmable interrupt system. The 68901 provides two general-purpose timers, supports the global serial bus interface, and provides an interrupt controller that is used with the interrupt system. Procesor Module resources are described in the following paragraphs.

3.1 68020 MPU

The 68020 MPU is a Motorola 68000-family processor with a 32-bit address bus, data bus, and register set. Complete 68020 reference information is contained in the Motorola publication, *MC68020 32-Bit Microprocessor User's Manual.* A summary of the 68020's features is given in the following paragraphs.

3.1.1 User-Level Programming Model

The 68020 user-level programming model contains the following registers:

- Eight 32-bit data registers (D0-D7)
- Seven 32-bit address registers (A0-A6)
- 32-bit User Stack Pointer (A7 or USP)
- 32-bit Program Counter (PC)
- 8-bit Condition Code Register (CCR)

The user-level programming model is shown in Figure 3-2. Its registers are described in the following paragraphs.

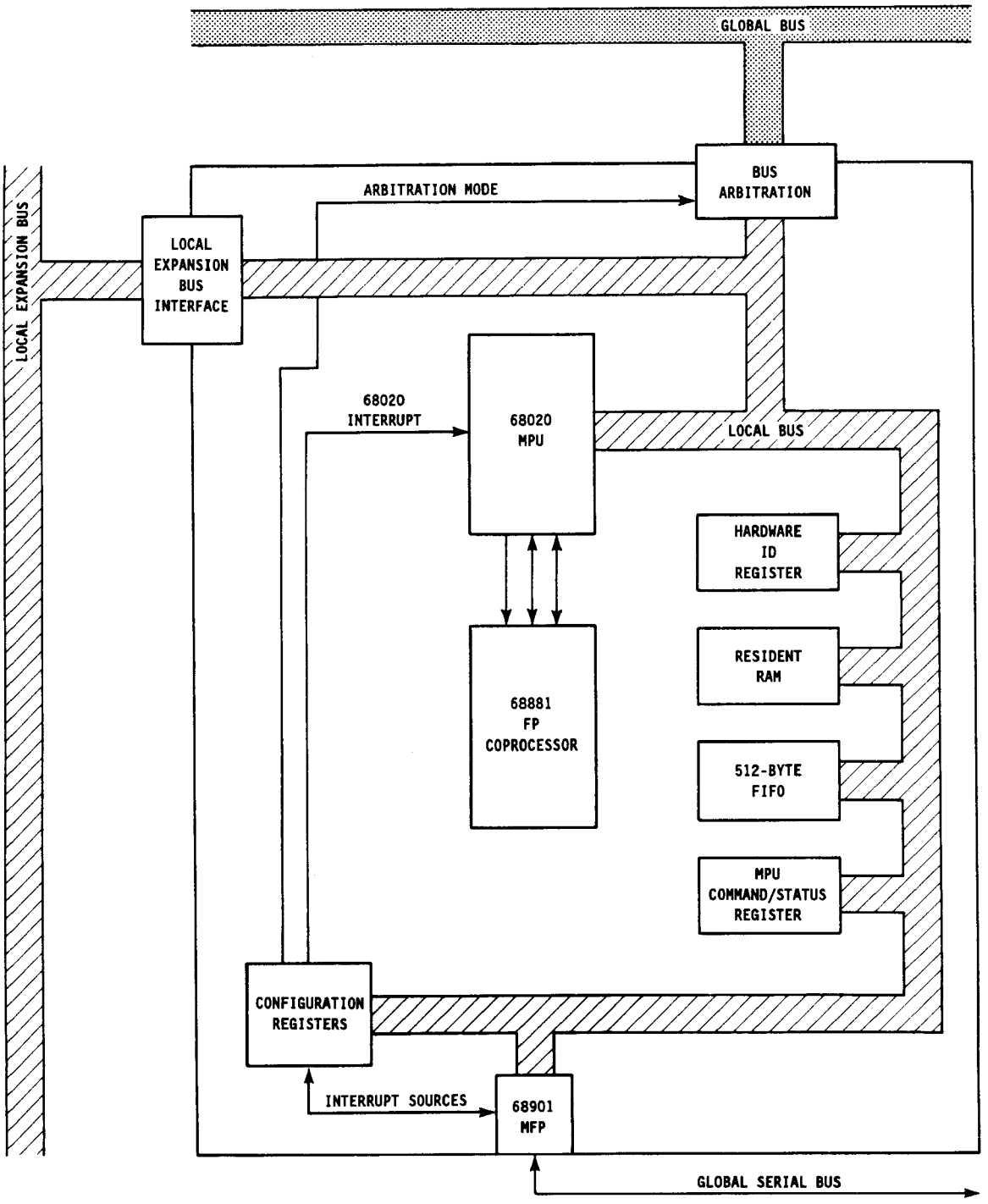


Figure 3-1. Processor Module Architecture

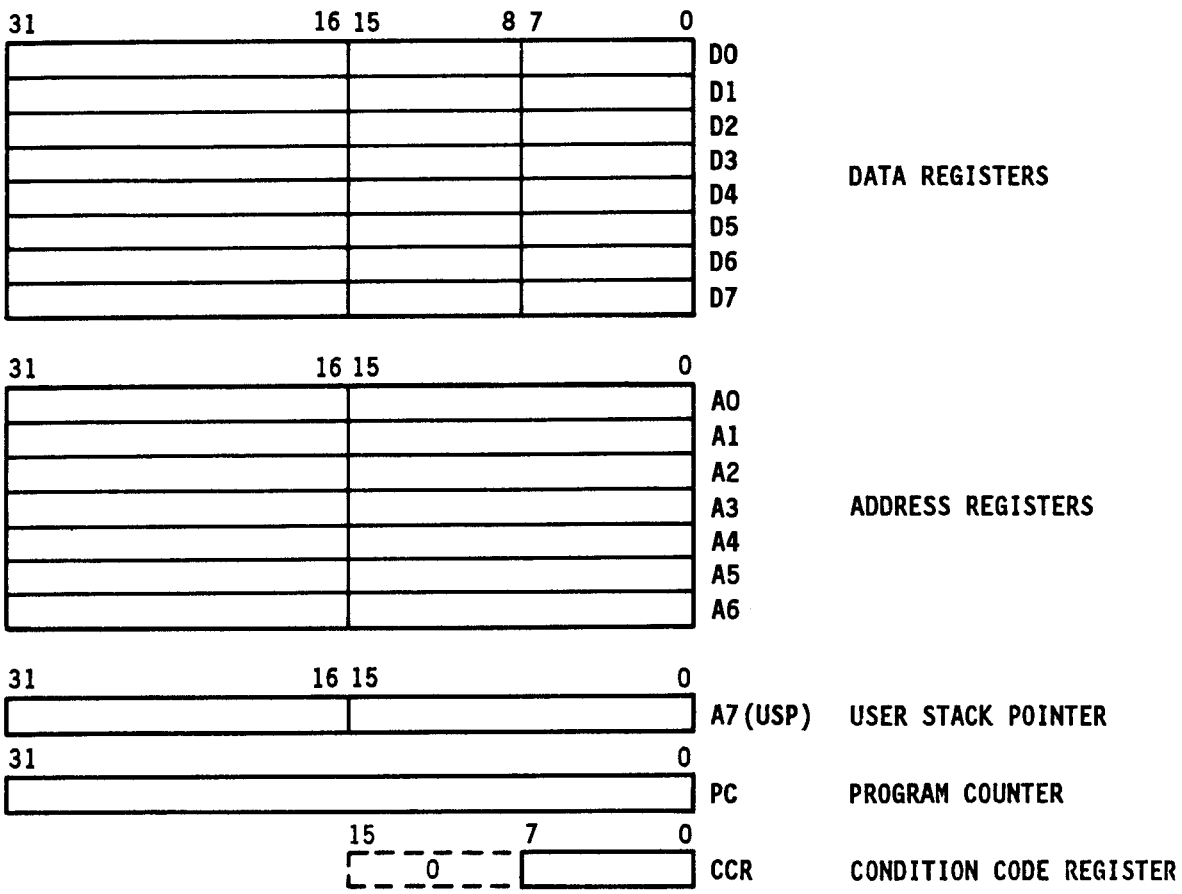


Figure 3-2. 68020 User-Level Programming Model

3.1.1.1 Data Registers (D0-D7). Data registers D0 through D7 are general-purpose registers that can be used for byte (8-bit), word (16-bit), or long-word (32-bit) operations. All data registers can be used as index registers.

3.1.1.2 Address Registers (A0-A6). Address registers A0 through A6 can be used as software stack pointers, base address registers, or index registers. They can also be used as data registers for word and long-word operations.

3.1.1.3 User Stack Pointer (A7 or USP). The USP contains the 32-bit address of the current top of the user-level stack. During user-level operation, instructions that implicitly access the stack use the address contained in USP.

3.1.1.4 Condition Code Register (CCR). The CCR provides information about MPU results with five separate indicator bits. The format of the CCR is shown below:

CCR

7	6	5	4	3	2	1	0
*	*	*	X	N	Z	V	C

***Not Used**

C CARRY - Set if a carry is generated out of the most-significant bit of the result during an addition operation or if a borrow is generated out of the minuend during a subtraction operation. Cleared otherwise.

V OVERFLOW - Set if the result of an arithmetic operation is not representable in the operand size. Cleared otherwise.

Z ZERO - Set if the result equals zero. Cleared otherwise.

N NEGATIVE - Set if the most-significant bit of the result (sign bit) is set. Cleared otherwise.

X EXTEND - This bit functions the same as the C bit but is used for multiprecision operations.

3.1.2 Supervisor-Level Programming Model

The 68020 supervisor-level programming model, shown in Figure 3-3, contains the following registers:

- Eight 32-bit data registers (D0-D7)
- Seven 32-bit address registers (A0-A6)
- 32-bit User Stack Pointer (A7 or USP)
- 32-bit Program Counter (PC)
- 16-bit Status Register (SR)
- 32-bit Interrupt Stack Pointer (A7' or ISP)
- 32-bit Master Stack Pointer (A7" or MSP)
- 32-bit Vector Base Register (VBR)
- 32-bit Source Function Code (SFC) Register
- 32-bit Destination Function Code (DFC) Register
- 32-bit Cache Control Register (CACR)
- 32-bit Cache Address Register (CAAR)

Registers D0-D7, A0-A6, and the USP (A7) function the same as the corresponding registers in the user-level programming model. The other registers are described in the following paragraphs.

3.1.2.1 Status Register (SR). The SR provides control bits for establishing the MPU's privilege level, supervisor-level operating mode, tracing mode, and interrupt level. SR bits can be modified only when the MPU is operating at the supervisor level. The low-order eight bits of the SR are identical to the user-level CCR. The format of the SR is shown below:

SR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T1	T0	S	M	*	IL2	IL1	IL0	*	*	*	X	N	Z	V	C

*Not Used

Tn TRACE MODE - These bits determine the mode of tracing as shown below:

T1	T0	Tracing Mode
0	0	No tracing
0	1	Trace on change of flow, i.e., branch, etc.
1	0	Trace on any instruction execution
1	1	Not used

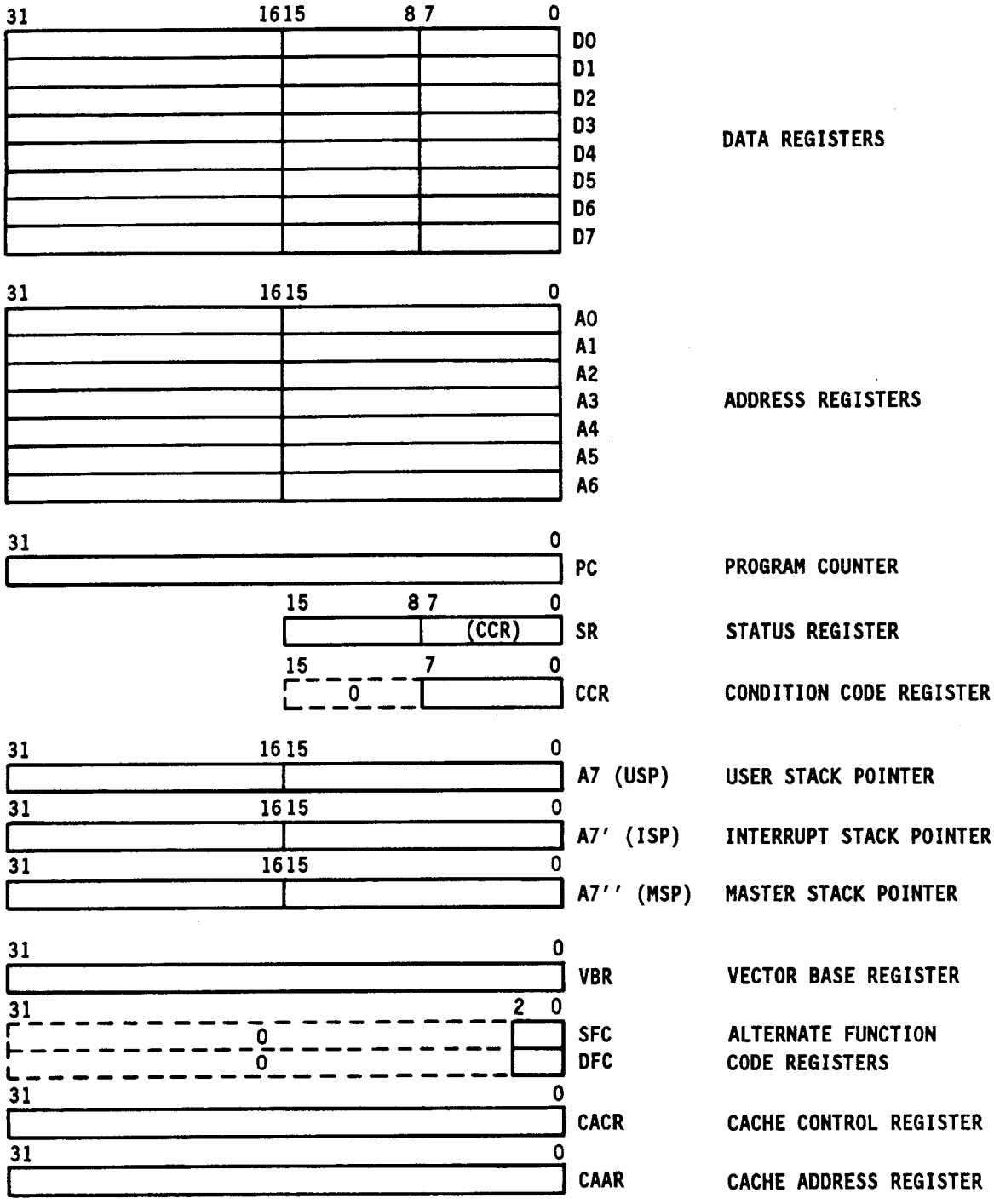


Figure 3-3. 68020 Supervisor-Level Programming Model

S SUPERVISOR - When this bit is set, the MPU operates at the supervisor level. When cleared, the MPU operates at the user level.

M MASTER - When this bit is set and the MPU is at the supervisor level, the MPU uses the Master Stack Pointer (MSP). When this bit is cleared (supervisor level), the MPU uses the Interrupt Stack Pointer (ISP).

ILn INTERRUPT LEVEL - These bits indicate the current interrupt priority level. The priority levels are from one (001) through seven (111), with level seven being the highest.

3.1.2.2 Interrupt Stack Pointer (ISP). The ISP contains the 32-bit address of the current top of the supervisor-level interrupt stack. During supervisor-level operation when the M bit of the SR is cleared, instructions that implicitly access the stack use the address in the ISP.

3.1.2.3 Master Stack Pointer (MSP). The MSP contains the 32-bit address of the current top of the supervisor-level master stack. During supervisor-level operation when the M bit of the SR is set, instructions that implicitly access the stack use the address in the MSP.

3.1.2.4 Vector Base Register (VBR). The VBR contains the 32-bit address that points to the base of the exception vector table.

3.1.2.5 Source Function Code (SFC) Register. The three least-significant bits of the SFC register specify the source of a MOVES read. The format of the SFC register is shown below:

SFC Register

31 thru 3	2	1	0
*	FC2	FC1	FC0

*Not Used

FCn FUNCTION CODE - The encoding of the three function code bits to indicate the source of a MOVES read is as follows:

FC2	FC1	FC0	Address Space
0	0	0	Unassigned
0	0	1	User Data
0	1	0	User Program
0	1	1	Unassigned
1	0	0	Unassigned
1	0	1	Supervisor Data
1	1	0	Supervisor Program
1	1	1	CPU Space

3.1.2.6 Destination Function Code (DFC) Register. The three least-significant bits of the DFC register specify the destination of a MOVES write. The format of the DFC register is shown below:

DFC Register

31 thru 3	2	1	0
*	FC2	FC1	FC0

***Not Used**

FCn FUNCTION CODE - The encoding of the three function code bits to indicate the destination of a MOVES write is as follows:

FC2	FC1	FC0	Address Space
0	0	0	Unassigned
0	0	1	User Data
0	1	0	User Program
0	1	1	Unassigned
1	0	0	Unassigned
1	0	1	Supervisor Data
1	1	0	Supervisor Program
1	1	1	CPU Space

3.1.2.7 Cache Control Register (CACR). The CACR provides four bits for controlling the 68020 internal cache memory. The format of the CACR is shown below:

CACR

31 thru 4	3	2	1	0
*	C	CE	F	E

*Not Used

E ENABLE CACHE - When this bit is set, the 68020 internal instruction cache is enabled. When the bit is cleared, the cache is disabled, forcing continuous cache misses, suppressing fills, and causing the MPU always to access external memory.

F FREEZE CACHE - When this bit is set and the instruction cache is enabled, cache misses are not allowed to replace valid cache data.

CE CLEAR ENTRY - When this bit is set, the MPU invalidates (clears the Valid bit) the cache entry associated with the index in the Cache Address Register (CAAR). This operation is independent of the states of the E or F bits or the external Cache Disable (CDIS-) pin. This bit always reads as a zero.

C CLEAR CACHE - When this bit is set, it causes all the Valid bits in the instruction cache to be cleared, invalidating all entries. This bit always reads as a zero.

3.1.2.8 Cache Address Register (CAAR). The CACR specifies the cache entry affected by setting the CE bit in the CACR. The format of the CAAR is shown below:

CAAR

31 thru 8	7 thru 2	1	0
*	Index	*	*

*Not Used

INDEX The 6 bits of this field select 1 of the 64 entries in the 68020 internal instruction cache. The selected entry is affected by a CACR Clear Entry operation.

3.1.3 68020 Initial State

When the 68020 is initialized, the S bit in the Status Register is set, causing the MPU to function at the supervisor level. The initial state of the supervisor-level programming model is summarized below:

Register	Contents
D0-D7	00000000H
A0-A6	00000000H
A7 (USP)	00000000H
PC	Determined by Reset Exception Vector
SR	27XXH
A7' (ISP)	00000000H
A7" (MSP)	00000000H
VBR	00000000H
SFC	Indeterminate
DFC	Indeterminate
CACR	00000000H
CAAR	Indeterminate

3.2 68881 FLOATING-POINT COPROCESSOR

The 68881 (Motorola MC68881) floating-point coprocessor is a full implementation of the IEEE standard for binary floating-point arithmetic (IEEE Std 754). On the Processor Module, the 68881 interfaces with the 68020 over a standard coprocessor interface and forms a logical extension of the MPU register and instruction sets. Complete reference information on the 68881 is contained in the Motorola publication, *MC68881 Floating-Point Coprocessor User's Manual*. A summary of the 68881 features is provided in the following paragraphs.

3.2.1 Programming Model

The 68881 programming model includes the following registers:

- Eight 80-bit floating-point data registers (FP0-FP7)
- 32-bit Floating-Point Control Register (FPCR)
- 32-bit Floating-Point Status Register (FPSR)
- 32-bit Floating-Point Instruction Address Register (FPIAR)

The programming model is shown in Figure 3-4. The registers are described in the paragraphs that follow.

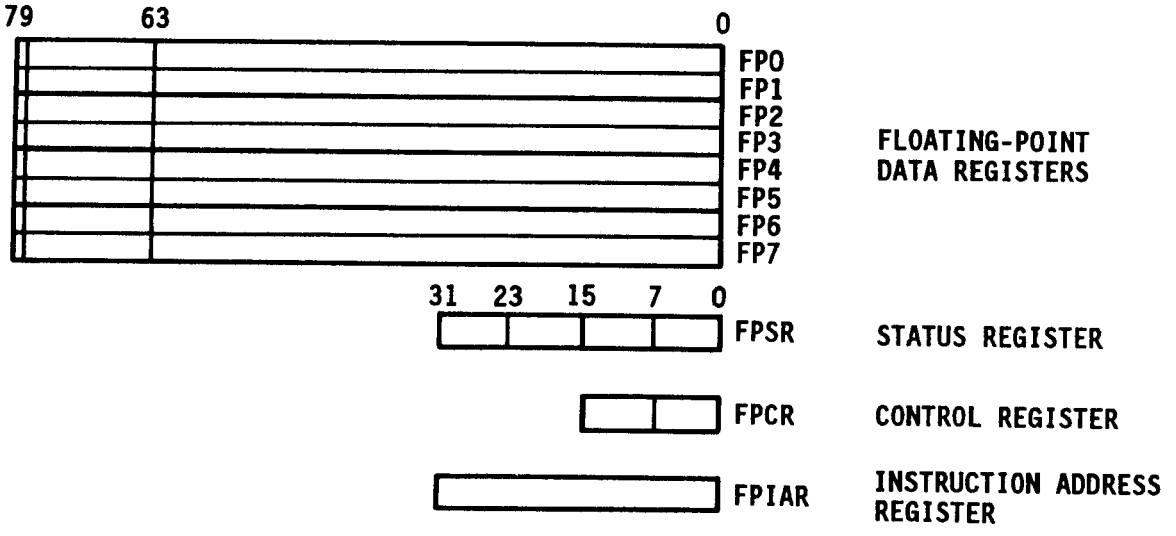


Figure 3-4. 68881 Floating-Point Coprocessor Programming Model

3.2.1.1 Floating-Point Data Registers (FP0-FP7). The 80-bit floating-point registers, FP0 through FP7, are general-purpose data registers analogous to the 68020 registers D0 through D7. The contents of these registers are always in extended precision format.

3.2.1.2 Floating-Point Control Register (FPCR). The FPCR provides bits for enabling floating-point exception traps and for selecting the rounding mode and precision for inexact results. The format of the FPCR is shown below:

FPCR

31 thru 16	15 thru 8	7	6	5	4	3	2	1	0
*	EX TRAP	RP		RM		*			

*Not Used

EX TRAP EXCEPTION TRAP ENABLE - This field contains one bit for each floating-point exception class. When a bit is set, the occurrence of the corresponding condition causes an exception. The format of this field is shown below:

FPCR Bit	Bit Mnemonic	Trap Condition
15	BSUN	Branch/set on unordered
14	SNaN	Signaling Not-a-Number
13	OPERR	Operand error
12	OVFL	Overflow
11	UNFL	Underflow
10	DZ	Divide-by-zero
9	INEX2	Inexact results
8	INEX1	Inexact conversion

RP ROUNDING PRECISION - These bits specify where rounding is to occur in the mantissa for results of different precisions as shown below:

FPCR Bit 7	FPCR Bit 6	Precision
0	0	Extended
0	1	Single
1	0	Double
1	1	Undefined

RM ROUNDING MODE - These bits specify how inexact results are to be rounded as shown below:

FPCR Bit 5	FPCR Bit 4	Rounding Mode
0	0	Round to nearest (RTN)
0	1	Round to zero (RTZ)
1	0	Round to minus infinity (RTMI)
1	1	Round to plus infinity (RTPI)

3.2.1.3 Floating-Point Status Register (FPSR). The FPSR provides the following information:

- Condition codes
- Quotient bits
- Current exceptions
- Accrued exceptions

The format of the FPSR is shown below:

FPSR

31 thru 28	27 thru 24	23 thru 16	15 thru 8	7 thru 3	2 thru 0
*	CC	QUOTIENT	EX IND	ACC EX	*

*Not Used

CC CONDITION CODE - These bits are modified following the execution of arithmetic instructions involving FP0-FP7. The format of this field is shown below:

FPSR Bit	27	26	25	24
Condition Code	N	Z	I	NaN

N NEGATIVE - Set if the result was negative. Cleared otherwise.

Z ZERO - Set if the result was zero. Cleared otherwise.

I INFINITY - Set if the result was infinity. Cleared otherwise.

NaN NOT a NUMBER - Set if the result was NaN. Cleared otherwise.

QUO. The quotient byte of the FPSR is filled following a modulo (FMOD) or IEEE (FREM) remainder instruction. The format of the byte is shown below:

FPSR Bit	23	22	21	20	19	18	17	16
Quotient	S	Quotient Bits						

S SIGN - This bit indicates the sign of the entire quotient.

Quotient Bits - This field contains the seven least-significant bits of the quotient. The field is unsigned.

EX IND EXCEPTION INDICATORS - This field contains one bit for each floating-point exception condition. When an exception is detected, the corresponding bit is set. All instructions except those that do not cause exceptions clear this field before starting. The format of the field is shown below:

FPSR Bit	Bit Mnemonic	Exception Condition
15	BSUN	Branch/set on unordered
14	SNaN	Signaling Not-a-Number
13	OPERR	Operand error
12	OVFL	Overflow
11	UNFL	Underflow
10	DZ	Divide-by-zero
9	INEX2	Inexact result
8	INEX1	Inexact conversion

ACC EX ACCRUED EXCEPTION - This field contains a history of all floating-point exceptions that have occurred since the field was last cleared. The bits are formed using logical combinations of the bits in the EX IND field as shown below:

FPSR Bit	Bit Mnemonic	Exception Conditions
7	IOP	BSUN or SNaN or OPERR
6	OVFL	OVFL
5	UNFL	UNFL and INEX2
4	DZ	DZ
3	INX	INEX1 or INEX2 or OVFL

3.2.1.4 Floating-Point Instruction Address Register (FPIAR). The FPIAR contains the 32-bit address of the last floating-point instruction executed.

3.2.2 68881 Initial State

The state of the 68881 programming model following a powerup or reset operation is summarized below:

Register	Content
FP0-FP7	Positive non-signaling NaN, mantissa = all 1s
FPCR	00000000H
FPSR	00000000H
FPIAR	00000000H

3.3 HARDWARE IDENTIFICATION (HWID) REGISTER

The Hardware Identification (HWID) Register is an 8-bit register that contains the revision level of the Processor Module's gate array VLSI and also reflects the setting of the hardware jumpers that form the Processor Module identification (ID) number. The four-bit ID, from 0001 (1) through 1111 (15), is unique for each Processor Module resident in a SANDAC V. The ID defines a Processor Module's global address and also specifies the module's priority for global bus arbitration. The format of the HWID Register is shown below:

HWID Register

7	6	5	4	3	2	1	0
R3	R2	R1	R0	ID3	ID2	ID1	ID0

Address: On,FFA,000 (n = Processor ID in hex)
Access Level: Supervisor (Read only)

Rn REVISION - This field contains the four-bit revison level of the gate array VLSI. Bit R3 is the most-significant bit.

IDn IDENTIFICATION - This field contains the four-bit Processor Module identification (ID) number. The ID defines a Processor Module's global address and also specifies the module's global bus arbitration priority as shown below:

ID3	ID2	ID1	ID0	ID	Address Space (hex)	Priority
0	0	0	1	1	01,000,000-01,FFF,FFF	Lowest
0	0	1	0	2	02,000,000-02,FFF,FFF	.
0	0	1	1	3	03,000,000-03,FFF,FFF	.
0	1	0	0	4	04,000,000-04,FFF,FFF	.
0	1	0	1	5	05,000,000-05,FFF,FFF	.
0	1	1	0	6	06,000,000-06,FFF,FFF	.
0	1	1	1	7	07,000,000-07,FFF,FFF	.
1	0	0	0	8	08,000,000-08,FFF,FFF	.
1	0	0	1	9	09,000,000-09,FFF,FFF	.
1	0	1	0	10	0A,000,000-0A,FFF,FFF	.
1	0	1	1	11	0B,000,000-0B,FFF,FFF	.
1	1	0	0	12	0C,000,000-0C,FFF,FFF	.
1	1	0	1	13	0D,000,000-0D,FFF,FFF	.
1	1	1	0	14	0E,000,000-0E,FFF,FFF	.
1	1	1	1	15	0F,000,000-0F,FFF,FFF	Highest

3.4 CONFIGURATION REGISTERS

The Processor Module's gate array contains a group of 15 registers known collectively as the configuration registers, which support the following functions:

- Bus arbitration
- Address monitoring
- Interrupt selection

Table 3-1 summarizes the gate array configuration registers. Detailed descriptions of the bus arbitration and address monitor registers are given in the following paragraphs. The interrupt select registers are described in Subsection 3.9.

TABLE 3-1. CONFIGURATION REGISTER SUMMARY

Register	Function	Address*
CONFIG1	Bus arbitration mode select, address monitor mode select	0n,FF7,001
CONFIG2	Address monitor upper address most-significant byte	0n,FF7,002
CONFIG3	Address monitor upper address middle byte	0n,FF7,003
CONFIG4	Address monitor upper address least-significant byte	0n,FF7,004
CONFIG5	Address monitor lower address most-significant byte	0n,FF7,005
CONFIG6	Address monitor lower address middle byte	0n,FF7,006
CONFIG7	Address monitor lower address least-significant byte	0n,FF7,007
CONFIG8	FIFO, 68901 interrupt priority select	0n,FF7,008
CONFIG9	Local interrupt W, X priority select	0n,FF7,009
CONFIG10	Local interrupt Y, Z priority select	0n,FF7,00A
CONFIG11	Global interrupt W, X priority select	0n,FF7,00B
CONFIG12	Global interrupt Y, Z priority select	0n,FF7,00C
CONFIG13	Software interrupt 0, 1 input select	0n,FF7,00D
CONFIG14	Software interrupt 2, 3 input select	0n,FF7,00E
CONFIG15	Software interrupt 4, 5 priority select	0n,FF7,00F

*Addresses are in hex; "n" is the Processor ID.

3.4.1 Bus Arbitration Mode Control

Half of CONFIG1 controls the two modes of arbitration for the SANDAC V global bus. The two modes are described below:

Priority Mode The Processor Module with the highest priority (highest ID number) controls the bus for as long as it requests it.

Fair Mode After a Processor Module has been the bus master, it is prohibited from making another request until the bus has been free for at least one cycle. Therefore, when two or more modules contend for the bus, the highest priority module is served first, but it is prevented from acquiring the bus again until the lowest priority module has been allowed access.

The mode of bus arbitration is selected by the setting of one bit (bit 4) in CONFIG1. The format of the register is shown below:

CONFIG1

7	6	5	4	3	2	1	0
*	*	*	PRI	**			

*Not Used
**Used by Address Monitor

Address: On,FF7,001 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

PRI PRIORITY - When this bit is set, the Processor Module uses the priority mode of arbitration; when cleared, the fair mode is used.

3.4.2 Address Monitor

Half of CONFIG1 and registers CONFIG2 through CONFIG7 make up the programmable address monitor that covers the entire 16M of Processor Module address space, including all memory, I/O, and register locations. The block of address space and the types of accesses to be monitored are specified with the seven configuration registers. When a specified access is detected within the monitored address space, a bus error is signaled. A bit in the Command/Status Register (Subsection 3.7) indicates an address monitor bus error.

Access types are specified with CONFIG1. The format of this register is shown below:

CONFIG1

7	6	5	4	3	2	1	0
*	*	*	**	U	S	R	W

*Not Used
**Used for Arbitration Mode

Address: On,FF7,001 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

U USER-LEVEL ACCESS - When this bit is set, the address monitor forces a bus error when a user-level access of the selected address block occurs. This bit can be set concurrently with the S bit.

S SUPERVISOR-LEVEL ACCESS - When this bit is set, the address monitor forces a bus error when a supervisor-level access of the selected address block occurs. This bit can be set concurrently with the U bit.

R READ ACCESS - When this bit is set, the address monitor forces a bus error when a read access of the selected address block occurs. This bit can be set concurrently with the W bit.

W WRITE ACCESS - When this bit is set, the address monitor forces a bus error when a write access of the selected address block occurs. This bit can be set concurrently with the R bit.

NOTE

The address monitor is disabled when R=0 and W=0 or when U=0 and S=0. This is the initial state for both bit pairs.

The boundaries of the address space to be monitored are set with CONFIG2 through CONFIG7. Three of the registers hold the 24 bits that specify the upper address, and three hold the 24 bits of the lower address.

CAUTION

It is advisable to set the lower address boundary one instruction higher than the desired lower limit because a Branch Always (BRA) instruction may be at the last location before the monitored area. In this case, the 68020 would perform a prefetch operation into the monitored area prior to executing the BRA instruction. This activity would cause an erroneous address monitor bus error to be signaled.

The formats of the address monitor boundary registers are shown below:

CONFIG2 (Upper Address Most-Significant Byte)

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

Address: On,FF7,002 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

CONFIG3 (Upper Address Middle Byte)

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Address: On,FF7,003 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

CONFIG4 (Upper Address Least-Significant Byte)

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Address: On,FF7,004 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

CONFIG5 (Lower Address Most-Significant Byte)

7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16

Address: On,FF7,005 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: FF hex

CONFIG6 (Lower Address Middle Byte)

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Address: On,FF7,006 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: FF hex

CONFIG7 (Lower Address Least-Significant Byte)

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Address: On,FF7,007 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: FF hex

3.5 RESIDENT RAM AND RAM SPACE

The first 0.5M of Processor Module address space is reserved for resident RAM. This space is divided into four 128K blocks. Only the first 128K block contains actual RAM. This RAM is 32 bits wide and is accessible at either the user or the supervisor level. The addresses for the 128K of actual resident RAM are "aliased" in the other three 128K blocks: addresses issued to locations in these blocks access the corresponding location in the lower 128K block.

Directly above the four 128K blocks reserved for resident RAM is a 1.5M space that is not accessible. Any access of this area causes a bus error to be signaled.

Directly above the inaccessible space are six blocks of 2M each reserved for SANDAC V Local Memory Expansion Modules (Section 5), which attach to the local expansion bus. The size of the individual blocks corresponds to the storage capacity (2M) of a single module. In configuring a system, as many as four of these modules can be attached to a Processor Module.

The following is a summary of Processor Module RAM space:

Address (hex)	Location	Implementation
0n,000,000-0n,01F,FFF	Processor Module	Actual RAM (128K)
0n,020,000-0n,07F,FFF	Processor Module	RAM Alias (3 times)
0n,080,000-0n,1FF,FFF	Processor Module	Inaccessible (Bus Error)
0n,200,000-0n,3FF,FFF	Expansion Bus	Local Mem. Exp. Module
0n,400,000-0n,5FF,FFF	Expansion Bus	Local Mem. Exp. Module
0n,600,000-0n,7FF,FFF	Expansion Bus	Local Mem. Exp. Module
0n,800,000-0n,9FF,FFF	Expansion Bus	Local Mem. Exp. Module
0n,A00,000-0n,BFF,FFF	Expansion Bus	Local Mem. Exp. Module
0n,C00,000-0n,DFF,FFF	Expansion Bus	Local Mem. Exp. Module

n = Processor ID in hex

3.6 512-BYTE FIFO

The 512-byte first-in-first-out (FIFO) buffer is 8 bits wide and 512 locations deep. This 8-bit port is writable and readable at the supervisor level, storing and returning the most-significant byte of the data bus (D31 - D24). Any time the FIFO contains data, an interrupt request is generated.

Writing to the FIFO when it is full causes a bus error and the data are not written. If the local MPU is responsible for the write, a bit in the Command/Status Register (Subsection 3.7) is set to indicate the cause of the bus error. If a global bus master is responsible for the write, the bit is not set, and there is no indication of the cause of the

bus error. FIFO "empty" and "half-full" status is reflected by two bits in the Command/Status Register. A summary of FIFO access is given below:

Address: 0n,FF8,000 hex (n = Processor ID in hex)
Access level: Supervisor
Initial state: Indeterminate

3.7 COMMAND/STATUS REGISTER

The Command/Status Register is a single register that serves two functions: when bits in the register are set with a write operation, commands are issued to the MPU (Command Register); when the register is read, various status conditions are returned (Status Register). The format of the register is shown below, separated into command and status functions:

Command Register

7	6	5	4	3	2	1	0
RST	HLT	TBE	ABE	FBE	CD	*	*

*No Command Function

Address: 0n,FF9,000 hex (n = Processor ID in hex)
Access Level: Supervisor (Write only)
Initial State: 00 hex

RST RESET - When this bit is set, the local MPU is held in the reset state. It is removed from the reset state only when another device accesses (read or write, any location) its memory space after a minimum of 256 clock cycles has elapsed since the bit was set. An access prior to 256 clocks causes a bus error on the originating MPU. Once the MPU is removed from the reset state, it proceeds with the normal powerup reset sequence; thus, there is no need to clear this bit. The reset command implemented through this register is intended for software diagnostic purposes only.

HLT HALT - When this bit is set, the MPU enters the halted processing state. Another Processor Module must clear the bit to remove the MPU from the halted state.

TBE TIMEOUT BUS ERROR - Clearing this bit clears the corresponding TBE flag in the status portion of the register.

ABE ADDRESS MONITOR BUS ERROR - Clearing this bit clears the corresponding ABE flag in the status portion of the register.

FBE FIFO BUS ERROR - Clearing this bit clears the corresponding FBE flag in the status portion of the register.

CD CACHE DISABLE - Setting this bit disables the 68020 internal instruction cache. This is a hardware disable and causes the cache to be bypassed regardless of the state of the 68020 CACR. Clearing the bit enables the cache, assuming first it is enabled through the CACR.

Status Register

7	6	5	4	3	2	1	0
HTD	HLT	TBE	ABE	FBE	CD	FHF	FMT

Address: On,FF9,000 hex (n = Processor ID in hex)
Access Level: Supervisor (Read only)
Initial State: 01 hex

HTD HALTED - When this bit is set, it indicates that the MPU is in the halted processing state. The MPU may have entered this state as a result of writing to the HLT bit in the Command Register or because of a double bus error. To determine the cause, the user should examine the Status Register HLT bit (see below). If HLT is set, then the HTD condition is the result of the Command Register; if HLT is cleared, the HTD condition is the result of a double bus error.

HLT HALT - This bit echoes the state of the HLT bit in the command portion of the register.

TBE TIMEOUT BUS ERROR - When this bit is set, it indicates that a bus error was caused because there was no response to an MPU address within 512 clock cycles. This flag is cleared by writing a 0 to the corresponding bit in the command portion of the register.

ABE ADDRESS MONITOR BUS ERROR - When this bit is set, it indicates that a bus error was caused by the address monitor (Subsection 3.5.1) detecting an illegal memory access by the local MPU. An illegal access by a global bus master is not flagged. This flag is cleared by writing a 0 to the corresponding bit in the command portion of the register.

FBE FIFO BUS ERROR - When this bit is set, it indicates that a bus error was caused by the local MPU writing to the FIFO when it was full. A write-when-full by a global bus master is not flagged. This flag is cleared by writing a 0 to the corresponding bit in the command portion of the register.

CD CACHE DISABLE - This bit echoes the state of the CD bit in the command portion of the register.

FHF FIFO HALF FULL - When this bit is set, it indicates that the 512-byte FIFO (Subsection 3.6) contains at least 257 bytes.

FMT FIFO EMPTY - When this bit is set, it indicates that the 512-byte FIFO (Subsection 3.6) is empty.

3.8 68901 MFP

The 68901 Multi-Function Peripheral (MFP) is a Motorola 68000-family device that generically provides the following resources for MPU support:

- 4 timers (Timers A, B, C, D)
- 1 Universal Synchronous/Asynchronous Receiver-Transmitter (USART)
- 8 programmable I/O lines with interrupt capability
- 1 16-source interrupt controller

On the Processor Module, these resources are utilized as follows:

- Timers A and B are general-purpose
- The USART and Timers C and D form the global serial bus interface
- The 8 I/O lines are used as inputs to the interrupt controller
- The interrupt controller handles as many as 16 requests for the module's programmable interrupt system

These Processor Module resources and a 68901 programming model are described in the following paragraphs. Complete information on the 68901 is in the Motorola publication, *MC68901 Multi-Function Peripheral.*

3.8.1 General-Purpose Timers

Only two of the four timers in the 68901, Timers A and B, are available for general-purpose use. The hardware configuration of the timers is shown in Figure 3-5. Each timer is an 8-bit prescaler/counter-type timer. As shown, the timers are interconnected to form a single 16-bit timer. Timer A can also function independently as an 8-bit timer. The input clock frequency at Timer A is 2.4576 MHz.

The timers can be programmed to operate in any of the following modes:

- Delay
- Pulse width measurement
- Event count

Refer to the 68901 manual for a description of these modes.

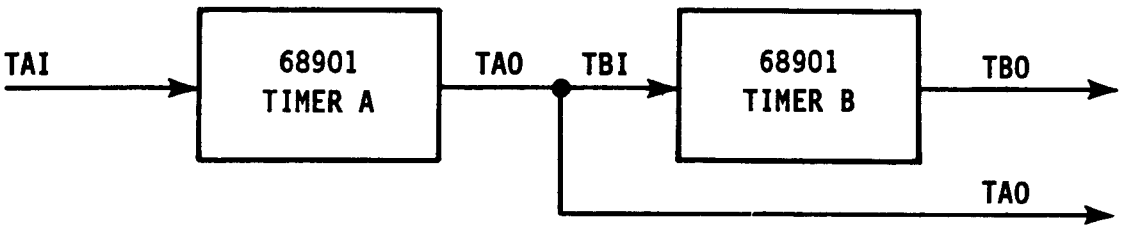


Figure 3-5. General-Purpose Timer Hardware Configuration

3.8.2 Global Serial Bus Interface

The global serial bus interface is implemented with the 68901 USART and Timers C and D, which are hardwired to supply the USART receive and transmit clocks, respectively. The USART and the timers are specifically dedicated to the global serial bus interface. The USART must support the interface as specified by the GATAR development software (Appendix). As GATAR requires asynchronous operation, descriptions of the USART Synchronous Character Register (SCR) and other features that apply only to synchronous operation are not included in the programming model description.

A block diagram of the interface is shown in Figure 3-6.

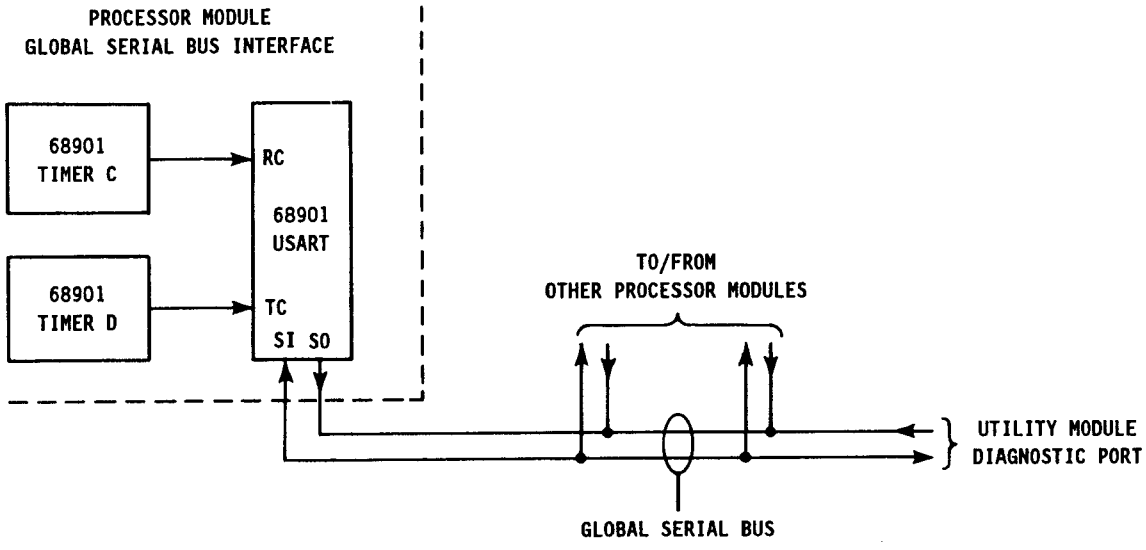


Figure 3-6. Global Serial Bus Interface

3.8.3 Programmable I/O Lines

The eight programmable I/O lines are used only to connect external interrupt requests to the 68901 interrupt controller and therefore cannot be used for general-purpose I/O operations. For this reason, a description of the General-Purpose Input/Output Data Register (GPIP) is not included in the programming model section.

3.8.4 Interrupt Controller

The 68901 interrupt controller has 16 prioritized channels for handling interrupt requests. The requests are generated by sources internal to the 68901 and by external sources. The internal sources include the 68901 timers and USART. The external sources are routed to the controller on the eight programmable I/O lines, I0 through I7, and the Timer A external input, TAI.

The 16 interrupt controller channels are prioritized, and concurrent requests are handled highest priority first. The channel assignments for the interrupt controller and their priorities are as follows:

Channel	Interrupt Request	Priority
1111	Input 7 (I7)	Highest
1110	Input 6 (I6)	.
1101	Timer A	.
1100	USART (receive buffer full)	.
1011	USART (receiver error)	.
1010	USART (transmit buffer empty)	.
1001	USART (transmitter error)	.
1000	Timer B	.
0111	Input 5 (I5)	.
0110	Input 4 (I4)	.
0101	Timer C	.
0100	Timer D	.
0011	Input 3 (I3)	.
0010	Input 2 (I2)	.
0001	Input 1 (I1)	.
0000	Input 0 (I0)	Lowest

When interrupt requests are presented to the interrupt controller, the controller responds to the highest priority source and causes the 68901 to generate a single vectored interrupt request. During the 68020 interrupt acknowledge cycle, the 68901 furnishes an interrupt vector with the format shown below:

68901 Interrupt Vector Format

7	6	5	4	3	2	1	0
V3	V2	V1	V0	CH3	CH2	CH1	CH0

Vn VECTOR - The four bits in this field are copied from the 68901 Vector Register (VR), (refer to the programming model). The bits are loaded into the VR when the 68901 is initialized to designate the interrupt vector the device will use. The SANDAC V powerup software initializes the VR to 40 hex (64 decimal); thus, the 68901 uses interrupt vector numbers 40 through 5F (64 through 79).

CHn CHANNEL - The four bits in this field designate the interrupt controller channel that made the interrupt request.

3.8.5 Programming Model

The address map for the 68901 MFP is given in Table 3-2. The programming model is shown in Figure 3-7 and followed by a register summary.

TABLE 3-2. PROCESSOR MODULE 68901 MFP ADDRESS MAP

Register	Acronym	Offset*
General-Purpose Input/Output	GPIP	01
Active Edge	AER	03
Data Direction	DDR	05
Interrupt Enable A	IERA	07
Interrupt Enable B	IERB	09
Interrupt Pending A	IPRA	0B
Interrupt Pending B	IPRB	0D
Interrupt In-Service A	ISRA	0F
Interrupt In-Service B	ISRB	11
Interrupt Mask A	IMRA	13
Interrupt Mask B	IMRB	15
Vector	VR	17
Timer A Control	TACR	19
Timer B Control	TBCR	1B
Timer C and D Control	TCDCR	1D
Timer A Data	TADR	1F
Timer B Data	TBDR	21
Timer C Data	TCDR	23
Timer D Data	TDDR	25
Synchronous Character	SCR	27
USART Control	UCR	29
Receiver Status	RSR	2B
Transmitter Status	TSR	2D
USART Data	UDR	2F

*Offset in hex from base address 0n,FF4,000.

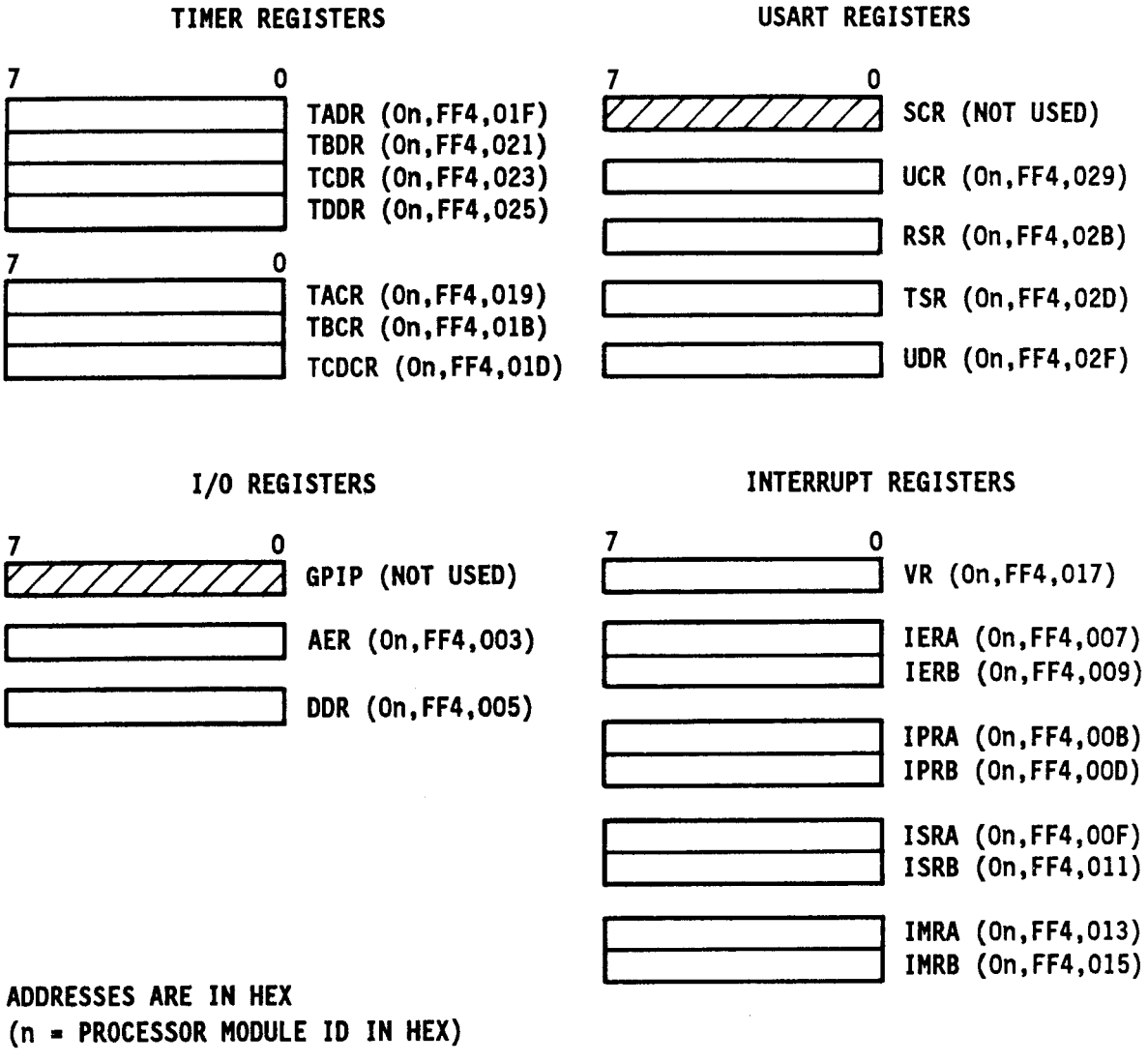


Figure 3-7. Processor Module 68901 Programming Model

TIMER DATA REGISTERS (TADR, TBDR, TCDR, TDDR)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Addresses

TADR: On,FF4,01F hex (n = Processor ID in hex)
TBDR: On,FF4,021 hex
TCDR: On,FF4,023 hex
TDDR: On,FF4,025 hex

Access Level: Supervisor
Initial State: Indeterminate

TIMERS A AND B CONTROL REGISTERS (TACR, TBCR)

7	6	5	4	3	2	1	0
0	0	0	TACR RESET TA0 TBCR RESET TB0	TIMER A/B CONTROL BITS 0000 - TIMER STOPPED 0001 - DELAY MODE, ÷4 PRESCALER 0010 - DELAY MODE, ÷10 PRESCALER 0011 - DELAY MODE, ÷16 PRESCALER 0100 - DELAY MODE, ÷50 PRESCALER 0101 - DELAY MODE, ÷64 PRESCALER 0110 - DELAY MODE, ÷100 PRESCALER 0111 - DELAY MODE, ÷200 PRESCALER 1000 - EVENT COUNT MODE 1001 - PULSE WIDTH MODE, ÷4 PSCLR 1010 - PULSE WIDTH MODE, ÷10 PSCLR 1011 - PULSE WIDTH MODE, ÷16 PSCLR 1100 - PULSE WIDTH MODE, ÷50 PSCLR 1101 - PULSE WIDTH MODE, ÷64 PSCLR 1110 - PULSE WIDTH MODE, ÷100 PSCLR 1111 - PULSE WIDTH MODE, ÷200 PSCLR			

Addresses

TACR: On,FF4,019 hex (n = Processor ID in hex)
TBCR: On,FF4,01B hex

Access Level: Supervisor
Initial State: 00 hex

TIMERS C AND D CONTROL REGISTER (TCDCR)

7	6	5	4	3	2	1	0
0	TIMER C CONTROL BITS			0	TIMER D CONTROL BITS		
	000-TIMER STOPPED 001-DELAY MODE, ÷4 PSCLR 010-DELAY MODE, ÷10 PSCLR 011-DELAY MODE, ÷16 PSCLR 100-DELAY MODE, ÷50 PSCLR 101-DELAY MODE, ÷64 PSCLR 110-DELAY MODE, ÷100 PSCLR 111-DELAY MODE, ÷200 PSCLR				000-TIMER STOPPED 001-DELAY MODE, ÷4 PSCLR 010-DELAY MODE, ÷10 PSCLR 011-DELAY MODE, ÷16 PSCLR 100-DELAY MODE, ÷50 PSCLR 101-DELAY MODE, ÷64 PSCLR 110-DELAY MODE, ÷100 PSCLR 111-DELAY MODE, ÷200 PSCLR		

Address: On,FF4,01D hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

USART CONTROL REGISTER (UCR)

7	6	5	4	3	2	1	0
CLOCK	WORD LENGTH		START/STOP BIT, FORMAT CONTROL		PARITY ENABLE	EVEN/ODD PARITY	0
0 - CLK X1 1 - CLK ÷16	00 - 8 BITS 01 - 7 BITS 10 - 6 BITS 11 - 5 BITS		00 - 0 START 0 STOP SYNCHRONOUS 01 - 1 START 1 STOP ASYNCHRONOUS 10 - 1 START 1.5 STOP ASYNCHRONOUS (USED ONLY IF BIT 7=1) 11 - 1 START 2 STOP ASYNCHRONOUS		0-DISABL 1-ENABLE	0 - ODD 1 - EVEN	

Address: On,FF4,029 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

RECEIVER STATUS REGISTER (RSR)

7	6	5	4	3	2	1	0
BUFFER FULL	OVERRUN ERROR	PARITY ERROR (PE)	FRAME ERROR (FE)	FOUND/ SEARCH OR BREAK DETECT	MATCH/ CHAR. IN PROGRESS	SYNCRO. STRIP ENABLE	RECEIVER ENABLED
1-WORD XFERED TO RCV BUFFER 0-RCV BUFFER READ	1-WORD RCVD & RCV BUFFER FULL 0-RSR READ	1-RCVD WORD HAS PE 0-RCVD WORD WITH NO PE	1-RCVD WORD HAS FE 0-RCVD WORD WITH NO FE	SYNCHRO. MODE 1-RCVD WORD MATCH SCR 0-RCVD WORD NO MATCH ASYNC. MODE 1-RCVD WORD IS BREAK 0-BREAK TERMI-NATES, RCV BUFFER READ	SYNCHRO. MODE 1-RCVD WORD MATCH SCR 0-RCVD WORD NO MATCH ASYNC. MODE 1-START BIT DETECT 0-END OF WORD DETECT	1-DISABL 0-ENABLE	1-ENABLE 0-DISABL

Address: On,FF4,02B hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

TRANSMITTER STATUS REGISTER (TSR)

7	6	5	4	3	2	1	0
BUFFER EMPTY	UNDERRUN ERROR	AUTO-TRNARND	END OF XMIT	BREAK	OUTPUT STATE TX DISABLED		XMTR ENABLE
1-XMIT BUFFER EMPTY 0-XMIT BUFFER FULL	1-XMIT SR XMTD BEFORE XMIT BUFFER WRITE 0-TSR READ	1-RCVR ENABLE AFTER XMIT (MPU WRT) 0-XMTR DISABL	1-WHEN TX DISABL AND XMIT DONE	ASYNC. ONLY 1-XMT BREAK 0-STOP BREAK	00 - HIGH IMPED. 01 - LOW 10 - HIGH 11 - LOOPBACK		1-ENABLE 0-DISABL

Address: On,FF4,02D hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

USART DATA REGISTER (UDR)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Address: On,FF4,02F hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: Indeterminate

ACTIVE EDGE REGISTER (AER)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - INTERRUPTS GENERATED ON THE FALLING EDGE OF THE INPUT SIGNAL 1 - INTERRUPTS GENERATED ON THE RISING EDGE OF THE INPUT SIGNAL							

Address: On,FF4,003 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

DATA DIRECTION REGISTER (DDR)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - I/O LINE IS AN INPUT 1 - I/O LINE IS AN OUTPUT							

Address: On,FF4,005 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

VECTOR REGISTER (VR)

7	6	5	4	3	2	1	0
V7	V6	V5	V4	S	0	0	0
MOST-SIGNIFICANT BITS OF INTERRUPT VECTOR NUMBER				0-AUTO. EOI 1-SOFTW. EOI			

Address: On,FF4,017 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT ENABLE REGISTER A (IERA)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	TIMER A	RCV BUFFER FULL	RCV ERROR	XMIT BUFFER EMPTY	XMIT ERROR	TIMER B
FOR ALL BITS: 0 - INTERRUPT CHANNEL DISABLED 1 - INTERRUPT CHANNEL ENABLED							

Address: On,FF4,007 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT ENABLE REGISTER B (IERB)

7	6	5	4	3	2	1	0
GPIP5	GPIP4	TIMER C	TIMER D	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - INTERRUPT CHANNEL DISABLED 1 - INTERRUPT CHANNEL ENABLED							

Address: On,FF4,009 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT PENDING REGISTER A (IPRA)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	TIMER A	RCV BUFFER FULL	RCV ERROR	XMIT BUFFER EMPTY	XMIT ERROR	TIMER B
FOR ALL BITS: 0 - NO INTERRUPT PENDING 1 - INTERRUPT PENDING							

Address: On,FF4,00B hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT PENDING REGISTER B (IPRB)

7	6	5	4	3	2	1	0
GPIP5	GPIP4	TIMER C	TIMER D	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - NO INTERRUPT PENDING 1 - INTERRUPT PENDING							

Address: On,FF4,00D hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT IN-SERVICE REGISTER A (ISRA)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	TIMER A	RCV BUFFER FULL	RCV ERROR	XMIT BUFFER EMPTY	XMIT ERROR	TIMER B
FOR ALL BITS: 0 - NO INTERRUPT PROCESSING IN PROGRESS FOR THE CHANNEL 1 - INTERRUPT PROCESSING IN PROGRESS FOR THE CHANNEL							

Address: On,FF4,00F hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT IN-SERVICE REGISTER B (ISRB)

7	6	5	4	3	2	1	0
GPIP5	GPIP4	TIMER C	TIMER D	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - NO INTERRUPT PROCESSING IN PROGRESS FOR THE CHANNEL 1 - INTERRUPT PROCESSING IN PROGRESS FOR THE CHANNEL							

Address: On,FF4,011 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT MASK REGISTER A (IMRA)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	TIMER A	RCV BUFFER FULL	RCV ERROR	XMIT BUFFER EMPTY	XMIT ERROR	TIMER B
FOR ALL BITS: 0 - INTERRUPTS ARE MASKED FOR THE CHANNEL 1 - INTERRUPTS ARE NOT MASKED FOR THE CHANNEL							

Address: On,FF4,013 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT MASK REGISTER B (IMRB)

7	6	5	4	3	2	1	0
GPIP5	GPIP4	TIMER C	TIMER D	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - INTERRUPTS ARE MASKED FOR THE CHANNEL 1 - INTERRUPTS ARE NOT MASKED FOR THE CHANNEL							

Address: On,FF4,015 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

3.9 INTERRUPT SYSTEM

The Processor Module interrupt system serves the following functions:

- Selecting and prioritizing interrupt requests to the resident 68020 MPU
- Generating interrupt requests to other Processor Modules

A block diagram of the interrupt system is shown in Figure 3-8. The operation of the system is described in the following paragraphs.

3.9.1 Interrupts to the 68020

The following are sources of interrupt requests to the resident 68020 MPU:

- 512-byte FIFO
- Local expansion bus interrupt W, X, Y, or Z
- Global bus interrupt W, X, Y, or Z
- Software interrupts
- 68901 MFP

As shown in Figure 3-8, some of the sources present their requests to the input of a programmable 10x7 crossbar switch. The switch is controlled by configuration registers CONFIG8 through CONFIG11 and CONFIG15. These registers are user programmable and assign a 68020 interrupt priority level to the requests.

The interrupt sources that are not presented to the 10x7 crossbar switch are presented to a 6x9 crossbar switch that is controlled by configuration registers CONFIG12 through CONFIG14. These registers are user programmable and connect the interrupt sources to the external inputs of the 68901 interrupt controller. Interrupt requests generated by these sources are presented to the 10x7 crossbar switch as the composite 68901 interrupt request.

The following paragraphs describe the interrupt sources and the associated crossbar switches.

3.9.1.1 512-Byte FIFO Interrupt. The 512-byte FIFO (Subsection 3.6) generates an interrupt request whenever it contains data. This interrupt request is directed to the 10x7 crossbar switch. The FIFO interrupt service routine must read the FIFO to clear the interrupt request. Additionally, the routine can make use of the FIFO Empty (FMT) flag in the status portion of the Command/Status Register; after reading the FIFO, the FMT flag can be examined to see whether there are more data in the FIFO.

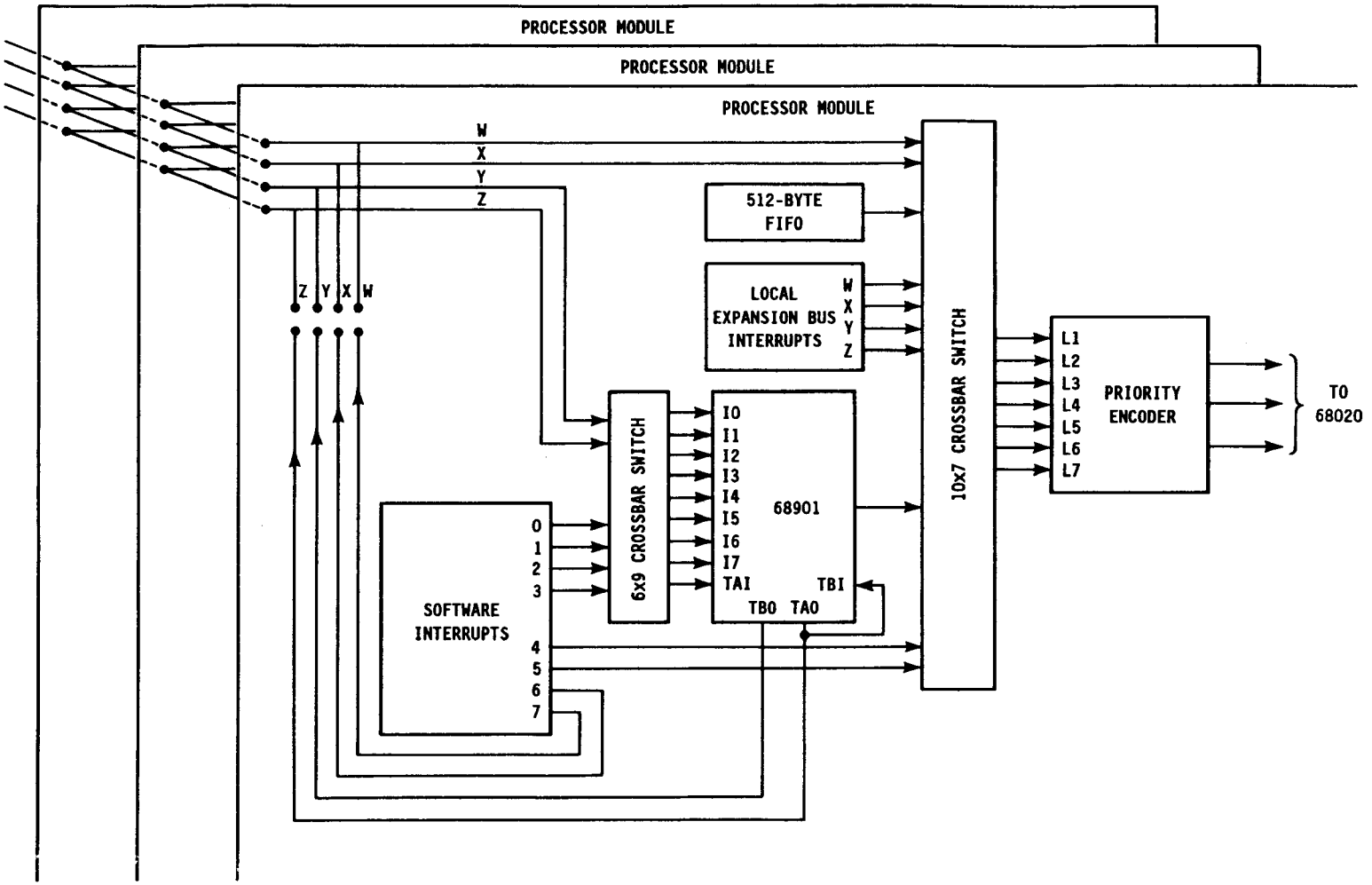


Figure 3-8. Processor Module Interrupt System

3.9.1.2 Local Expansion Bus Interrupts. The local expansion bus includes four interrupt request lines, W, X, Y, and Z, which send requests from local expansion bus modules to the Processor Module. These interrupts are directed to the 10x7 crossbar switch.

On the System I/O Module and the MIL-STD-1553B Module, some of the interrupt sources are hardwired to specific request lines; others may be connected to any line required with hardware jumpers. The following is a summary of local expansion bus interrupt request line assignments. They are applicable only when an expansion module(s) is attached to the local expansion bus.

Expansion Module	Interrupt Source	Request Line
System I/O	Serial Ports A, B	W
System I/O	Serial Ports C, D	W
System I/O	Serial Port E	X
System I/O	DMA Controller	Hardware selectable
System I/O	Parallel Port	Hardware selectable
MIL-STD-1553B	1553 Interface	Hardware selectable

3.9.1.3 Global Bus Interrupts. The global bus includes four interrupt request lines, W, X, Y, and Z. These lines send interrupt requests from one Processor Module to another. Global bus interrupt sources are connected to the request lines with hardware jumpers. Interrupts W and X are directed to the 10x7 crossbar switch; interrupts Y and Z are routed through the 6x9 crossbar switch to the 68901 interrupt controller external inputs.

3.9.1.4 Software Interrupts. Eight separate interrupt requests can be generated by writing to eight addresses in Processor Module address space. The write operation itself, to the designated address and with any data pattern, generates the interrupt request; there are no registers or control bits associated with this function. The write addresses for generating software interrupts are as follows:

Software Interrupt	Address (hex)
0	0n,FFB,000
1	0n,FFB,001
2	0n,FFB,002
3	0n,FFB,003
4	0n,FFB,004
5	0n,FFB,005
6	0n,FFB,006
7	0n,FFB,007

n = Processor ID in hex
Access at these addresses is supervisor level.

Software interrupts 0 through 3 are routed through the 6x9 crossbar switch to the 68901's interrupt controller external inputs; interrupts 4 and 5 are directed to the 10x7 crossbar switch. Software interrupts 6 and 7 generate interrupt requests that can be connected to the global interrupt lines.

3.9.1.5 68901 Interrupt. The 68901 MFP generates a single interrupt request from its interrupt controller. The request is initially generated by 1 of the 16 sources monitored by the controller. These sources include those internal to the 68901 and external sources. The internal sources include the following:

- Timer A, B, C, D
- USART (receive buffer full)
- USART (receiver error)
- USART (transmit buffer empty)
- USART (transmitter error)

The external sources are directed to the controller on the eight programmable I/O lines, I0 through I7, and the Timer A external input, TAI. External requests can be received from the following sources:

- Global bus interrupt Y
- Global bus interrupt Z
- Software interrupt 0
- Software interrupt 1
- Software interrupt 2
- Software interrupt 3

External requests are initially presented to the input of a programmable 6x9 crossbar switch that is controlled by configuration registers CONFIG12 through CONFIG14 (Table 3-1). These registers are user programmable and connect a specific source to a specific I/O line based on application requirements. The format of the registers is shown below:

CONFIG12 (6x9 Global Bus Y, Z Input Select)

7	6	5	4	3	2	1	0
IL3	IL2	IL1	IL0	IL3	IL2	IL1	IL0
Global Bus Y				Global Bus Z			

Address: On,FF7,00C hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: FF hex

CONFIG13 (6x9 Software Interrupt 0, 1 Select)

7	6	5	4	3	2	1	0
IL3	IL2	IL1	IL0	IL3	IL2	IL1	IL0
Software Int. 0				Software Int. 1			

Address: On,FF7,00D hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: FF hex

CONFIG14 (6x9 Software Interrupt 2, 3 Select)

7	6	5	4	3	2	1	0
IL3	IL2	IL1	IL0	IL3	IL2	IL1	IL0
Software Int. 2				Software Int. 3			

Address: On,FF7,00E hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: FF hex

ILn INPUT LINE - These bits select the 68901 external input line as shown below:

IL3	IL2	IL1	IL0	68901 External Input
0	0	0	0	I0
0	0	0	1	I1
0	0	1	0	I2
0	0	1	1	I3
0	1	0	0	I4
0	1	0	1	I5
0	1	1	0	I6
0	1	1	1	I7
1	0	0	0	TAI
1	0	0	1	Source not selected
1	0	1	0	.
1	0	1	1	.
1	1	0	0	.
1	1	0	1	.
1	1	1	0	.
1	1	1	1	Source not selected

For any software interrupts directed to the 68901 interrupt controller, the corresponding input line should be programmed to recognize the request as "active high." Additionally, the user should allow for one instruction time between the generation of the request and its recognition by the 68020.

3.9.2 Interrupts Directed to the 10x7 Crossbar

As shown in Figure 3-8, the following interrupt requests are presented to the 10x7 crossbar switch:

- 512-byte FIFO
- Local expansion bus interrupts W, X, Y, Z
- Global bus interrupts W and X
- Software interrupts 4 and 5
- 68901 MFP

The 68020 interrupt priority level (IPL) for each of the sources is set with configuration registers CONFIG8 through CONFIG11 and CONFIG15 (Table 3-1). When a request is received, the switch outputs the programmed priority to an encoder that places the level on the 68020 IPL lines. As there are 10 interrupt requests present at the switch and only 7 68020 interrupt priority levels, some requests may be programmed to the same priority, or some may not be selected. The format of the registers is shown below:

CONFIG8 (10x7 FIFO/68901 IPL)

7	6	5	4	3	2	1	0
*	PL2	PL1	PL0	*	PL2	PL1	PL0
FIFO				68901			

*Not Used

Address: On,FF7,008 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

CONFIG9 (10x7 Local Expansion Bus W, X IPL)

7	6	5	4	3	2	1	0
*	PL2	PL1	PL0	*	PL2	PL1	PL0
Local Exp. W				Local Exp. X			

*Not Used

Address: On,FF7,009 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

CONFIG10 (10x7 Local Expansion Bus Y, Z IPL)

7	6	5	4	3	2	1	0
*	PL2	PL1	PL0	*	PL2	PL1	P10
Local Exp. Y				Local Exp. Z			

*Not Used

Address: On,FF7,00A hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

CONFIG11 (10x7 Global Bus W, X IPL)

7	6	5	4	3	2	1	0
*	PL2	PL1	PL0	*	PL2	PL1	PL0
Global Bus W				Global Bus X			

*Not Used

Address: On,FF7,00B hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

CONFIG15 (10x7 Software Interrupts 4, 5 IPL)

7	6	5	4	3	2	1	0
*	PL2	PL1	PL0	*	PL2	PL1	PL0
Software Int. 4				Software Int. 5			

*Not Used

Address: On,FF7,00F hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

PLn PRIORITY LEVEL - These bits specify the interrupt priority level as shown below:

PL2	PL1	PL0	Interrupt Priority Level
0	0	0	Not Selected
0	0	1	1 (Lowest)
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7 (Highest)

3.9.2.1 Autovector Sources. The following sources request autovectors:

- 512-byte FIFO
- Global bus interrupts W, X
- Software interrupts 4, 5

NOTE

The 68020 generates one autovector for each priority level; therefore, assigning any of the autovector sources the same priority level can cause the 68020 to use the same interrupt handling routine for different interrupt types.

3.9.2.2 Vectored Sources. The local expansion bus interrupts and the 68901 MFP each provide a vectored response to an interrupt acknowledge. In the event that any of these are programmed to the same level, a daisy chain within the 10x7 crossbar switch ensures that only one source provides a vector. The order of the daisy chain is as follows:

1. 68901 MFP
2. Local expansion bus interrupt W
3. Local expansion bus interrupt X
4. Local expansion bus interrupt Y
5. Local expansion bus interrupt Z

NOTE

No autovector source should have the same priority level as any of the vectored sources.

3.9.3 Interrupts Generated by the Processor Module

A Processor Module can direct interrupt requests to other Processor Modules using global interrupt lines W, X, Y, and Z. The following Processor Module resources can be used to generate the requests:

- Software Interrupt 6
- Software Interrupt 7
- Timer A External Timeout (TAO)
- Timer B External Timeout (TBO)

An interrupt source is connected to the required global request line with a hardware jumper on the Processor Module acting as the source.

3.10 THE LOCAL EXPANSION BUS INTERFACE

The local expansion bus interface allows the addition of one or more of the following SANDAC V modules to a Processor Module's local bus:

- Local Memory Expansion Module (Section 5)
- System I/O Module (Section 6)
- MIL-STD-1553B Module (Section 7)

A maximum of four modules can be attached to one Processor Module. Module types can be combined, but the interface supports only the following quantities:

- 1 System I/O Module
- As many as 4 Local Memory Expansion Modules
  (memory space exists for 6 but only 4 can be used)
- As many as 4 MIL-STD-1553B Modules

The resources provided by these modules are mapped in the address space of the Processor Module to which they are attached as shown below:

Address (hex)	Assignment
0n,200,000-0n,3FF,FFF	Local Memory Expansion Module 1
0n,400,000-0n,5FF,FFF	Local Memory Expansion Module 2
0n,600,000-0n,7FF,FFF	Local Memory Expansion Module 3
0n,800,000-0n,9FF,FFF	Local Memory Expansion Module 4
0n,A00,000-0n,BFF,FFF	Local Memory Expansion Module 5
0n,C00,000-0n,DFF,FFF	Local Memory Expansion Module 6
0n,FF0,000-0n,FF3,FFF	Expansion I/O (MIL-STD-1553B Module)
0n,FF5,000-0n,FF6,FFF	Unassigned
0n,FFC,000-0n,FFF,FFF	System I/O Module

n = Processor ID in hex

4. UTILITY MODULE

The SANDAC V Utility Module resides on the global bus and provides the system with the following global resources:

- 256K EPROM
- 128K RAM
- 32-bit Timer
- Status Register

The architecture of the Utility Module is shown in Figure 4-1; the location of the global resources in global address space is summarized in Table 4-1. Resource descriptions are provided in the paragraphs that follow.

TABLE 4-1. UTILITY MODULE ADDRESSES

Assignment	Start Address*	End Address*
EPROM (256K)	00,000,000	00,03F,FFF
Reserved (256K)	00,040,000	00,07F,FFF
RAM (128K)	00,080,000	00,09F,FFF
Unassigned (256K)	00,0A0,000	00,0DF,FFF
32-Bit Timer (64K)	00,0E0,000	00,0EF,FFF
Status Register (64K)	00,0F0,000	00,0FF,FFF

*Addresses are in hexadecimal.

4.1 EPROM

The 256K of 16-bit global EPROM contains the following program modules:

- SANDAC V startup code
- Interrupt-driven portion of GATAR
- Hawk operating system

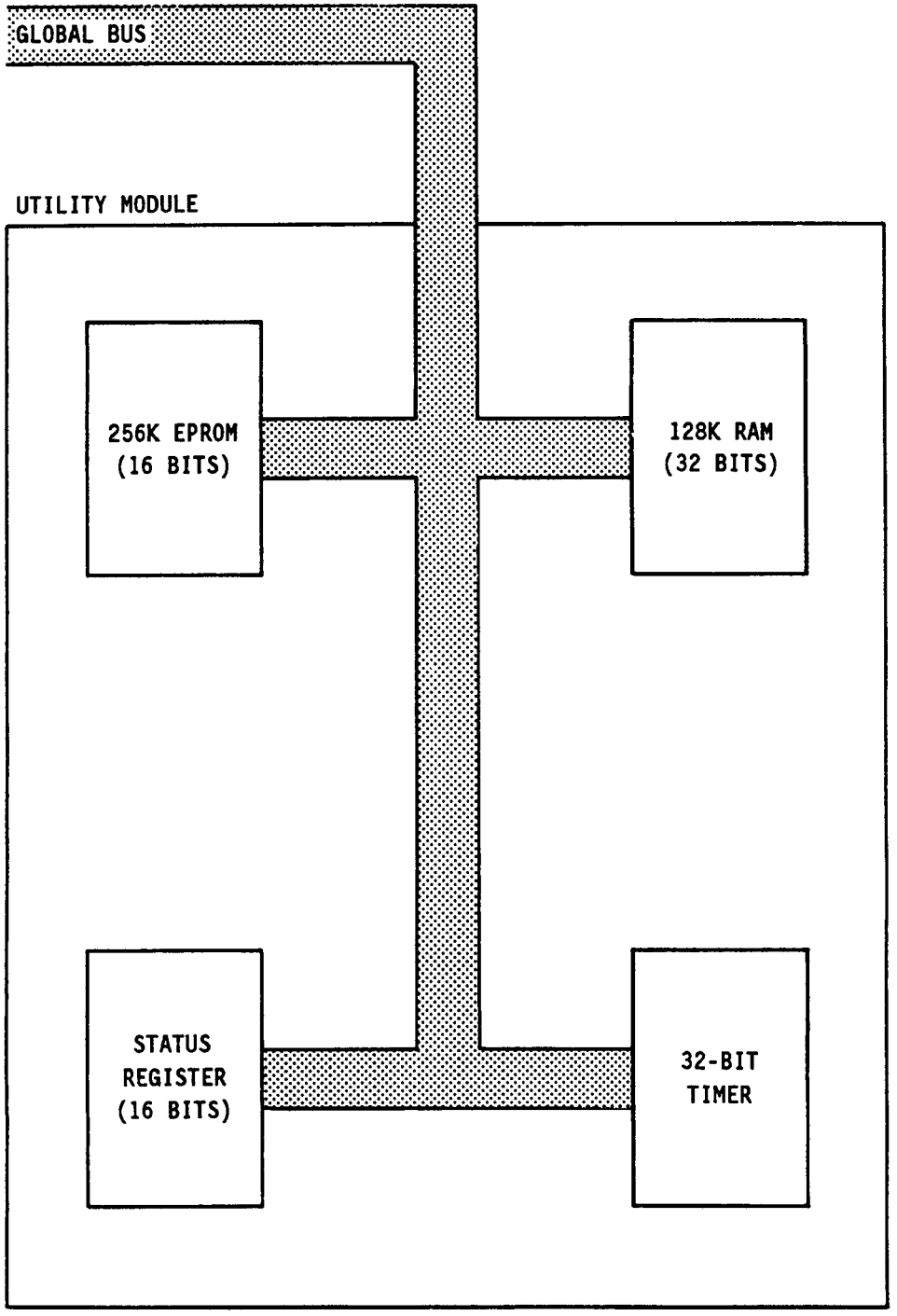


Figure 4-1. Utility Module Architecture

At powerup or reset, all Processor Modules execute the startup code stored in the EPROM. A description of the startup operation is given in Section 9.

When the EPROM is read, the Utility Module wait state generator introduces from one to eight wait states, selectable with hardware jumpers. The default selection is four wait states. A write operation to the EPROM area causes a bus error.

4.2 RAM

The 128K of 32-bit wide global RAM can be accessed by all Processor Modules at either the supervisor or user level. When RAM is accessed, the Utility Module wait state generator introduces from one to eight wait states, selectable with hardware jumpers. The default selection is two wait states. This RAM supports read-modify-write operations.

4.3 32-BIT TIMER

The 32-bit timer is implemented with a 32-bit up-counter clocked at 1 MHz. At this frequency, the counter value is incremented every microsecond; thus, microsecond resolution is attainable for general-purpose timing functions.

The timer is cleared by a supervisor-level write operation. No specific data value is required; the write access itself clears the timer. When the timer is read, its value is placed in a holding register, thus allowing the counter to continue running. For maximum accuracy, the entire 32 bits are read with one access. The read can be done at either the supervisor or the user level. A functional diagram of the 32-bit timer is shown in Figure 4-2.

When the timer is accessed, the Utility Module wait state generator introduces from one to eight wait states, selectable with hardware jumpers. The default selection is two wait states.

The following is a 32-bit timer programming summary:

Address: 00,0EX,XXX hex (X = don't care)
<u>Access Level</u>
Read: User or Supervisor (Read count)
Write: Supervisor only (Clear timer)
Initial State: 00000000 hex

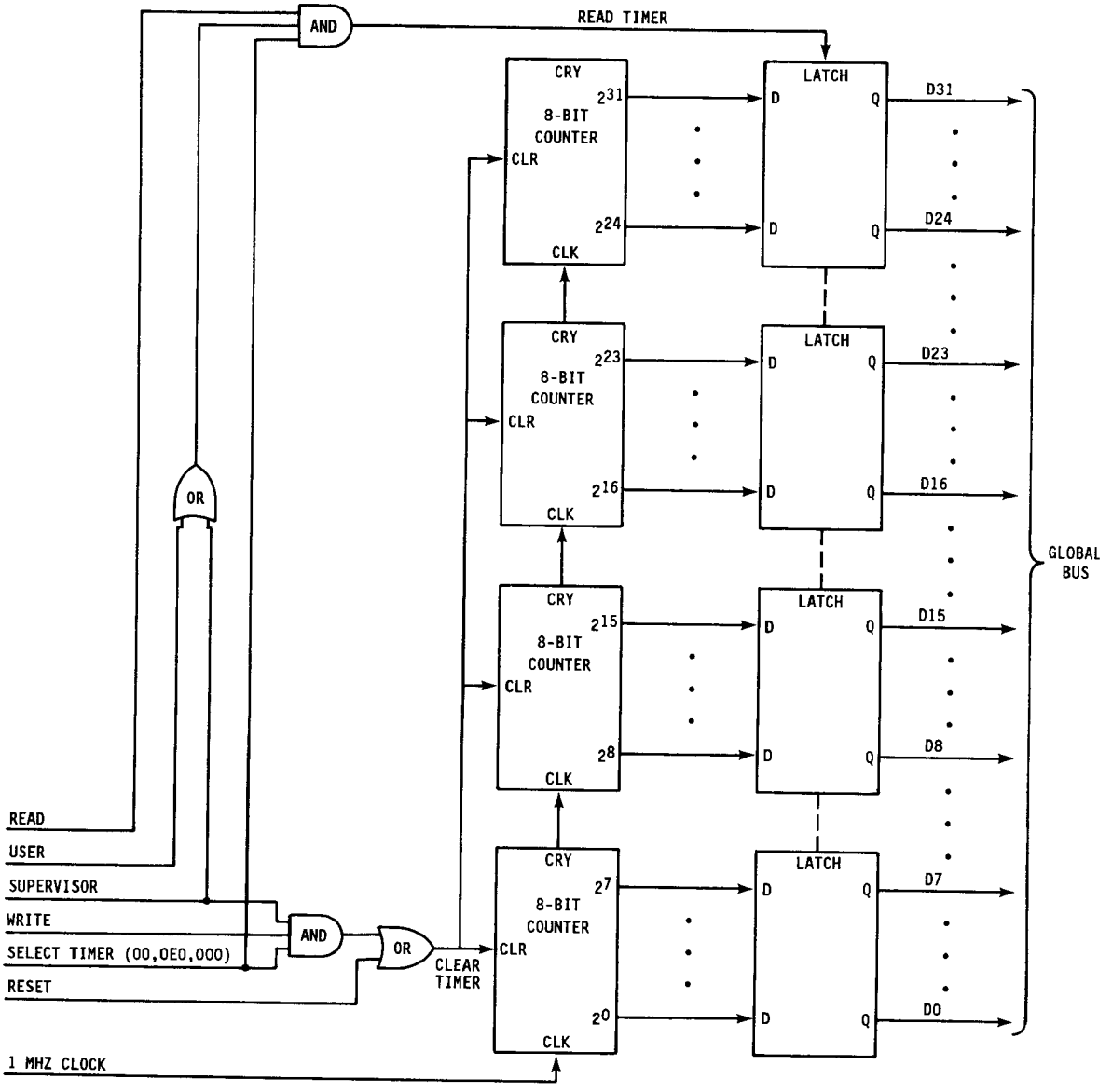


Figure 4-2. 32-Bit Timer

4.4 STATUS REGISTER

The 32-bit Status Register provides information about the battery backup for SANDAC V RAM, reset conditions, and a general-purpose external signal. When the status register is read, the Utility Module wait state generator introduces from one to eight wait states, selectable with hardware jumpers. The default selection is two wait states. The format of the Status Register is shown below:

Utility Module Status Register

31	30	29	28	27 thru 0
*	EXT	RST	BAK	*

***Not Used**

Address: 00,0FX,XXX hex (X = don't care)
Access Level: User or Supervisor (Read only)
Initial State: Refer to bit descriptions

EXT EXTERNAL INPUT - This bit monitors the output of a general-purpose opto-isolator that may be used to buffer an external signal, such as a switch setting.

RST RESET CONDITION - This bit differentiates between a powerup and a software reset. When the bit is set, it indicates that the reset was the result of a powerup. When cleared, it indicates a software reset. This bit must be read as soon as possible after a reset; it is guaranteed to be valid for only 50 milliseconds.

BAK RAM BACKUP - This bit indicates the status of the RAM backup battery. When the bit is set, it indicates that the battery voltage is above 3.1 volts and that the contents of RAM will be held intact if the main power source is turned off. When cleared, it indicates a low battery. This bit must be read as soon as possible after a reset; it is guaranteed to be valid for only 100 milliseconds.

NOTES

5. MEMORY MODULES

Two types of memory modules are available for SANDAC V:

- Global Memory
- Local Memory Expansion

Global Memory Modules reside on the global bus; Local Memory Expansion Modules attach to a Processor Module and reside on the local expansion bus. The modules are described in the following paragraphs.

5.1 GLOBAL MEMORY MODULE

Figure 5-1 is a block diagram of the Global Memory Module. As shown, the module provides 512K of RAM storage and an access monitor.

5.1.1 RAM Storage

The Global Memory Module provides 512K of battery-backed RAM storage on the global bus. Within the SANDAC V address map, 30 blocks of 512K each are reserved for Global Memory Modules. Modules may be placed in any of the blocks; hardware jumpers provide proper address decoding. Table 5-1 summarizes Global Memory Module address space.

5.1.2 Access Monitor

The Global Memory Module includes an access monitor that, when enabled, generates a bus error if an overwrite occurs in a user-selected block (or blocks) of module RAM.

NOTE

The access monitor does not inhibit the write operation. An overwrite occurs, and a bus error is indicated.

From 2 to 4,096 blocks of module RAM can be monitored, depending on the block size selected. The size can be from 128 to 256K bytes, selectable with hardware jumpers.

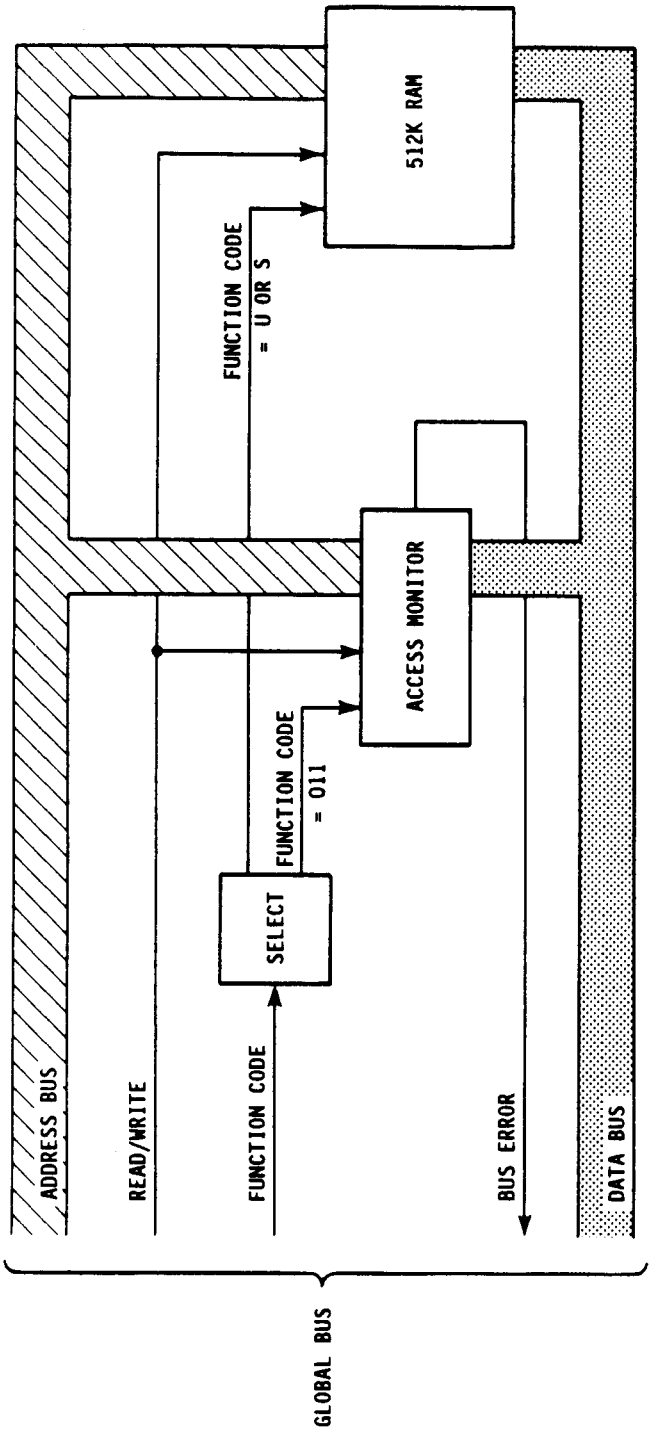


Figure 5-1. Global Memory Module

TABLE 5-1. GLOBAL MEMORY ADDRESSES

MEMORY BLOCK	START ADDRESS*	END ADDRESS*
1	00,100,000	00,17F,FFF
2	00,180,000	00,1FF,FFF
3	00,200,000	00,27F,FFF
4	00,280,000	00,2FF,FFF
5	00,300,000	00,37F,FFF
6	00,380,000	00,3FF,FFF
7	00,400,000	00,47F,FFF
8	00,480,000	00,4FF,FFF
9	00,500,000	00,57F,FFF
10	00,580,000	00,5FF,FFF
11	00,600,000	00,67F,FFF
12	00,680,000	00,6FF,FFF
13	00,700,000	00,77F,FFF
14	00,780,000	00,7FF,FFF
15	00,800,000	00,87F,FFF
16	00,880,000	00,8FF,FFF
17	00,900,000	00,97F,FFF
18	00,980,000	00,9FF,FFF
19	00,A00,000	00,A7F,FFF
20	00,A80,000	00,AFF,FFF
21	00,B00,000	00,B7F,FFF
22	00,B80,000	00,BFF,FFF
23	00,C00,000	00,C7F,FFF
24	00,C80,000	00,CFF,FFF
25	00,D00,000	00,D7F,FFF
26	00,D80,000	00,DFF,FFF
27	00,E00,000	00,E7F,FFF
28	00,E80,000	00,EFF,FFF
29	00,F00,000	00,F7F,FFF
30	00,F80,000	00,FFF,FFF

*Addresses are in hexadecimal.

As block size increases, the number of blocks that can be monitored decreases. The choice of block sizes and the corresponding number of blocks that can be monitored are summarized in Table 5-2.

TABLE 5-2. ACCESS MONITOR COVERAGE

Size of Monitored Block (in bytes)	Number of Blocks Covered (decimal)
128	4096
256	2048
512	1024
1K	512
2K	256
4K	128
8K	64
16K	32
32K	16
64K	8
128K	4
256K	2

Monitoring for any block of module RAM is enabled by writing to the access monitor. Additionally, the "status" of a block (monitoring enabled or disabled) can be determined by reading the access monitor.

Logically, the access monitor is a 512K block of address space with the same addresses as the 512K block of RAM on the module. The address spaces for the monitor and the RAM are actually separate because they require different 68020 function codes for access. RAM is accessed with any user-level or supervisor-level function code, which is generated automatically during a normal bus cycle. To write to or read from the access monitor, the user must program the 68020's DFC or SFC register with the function code 011. (The DFC register is programmed to write to the access monitor, and the SFC is programmed to read from the monitor. The registers are programmed with the MOVEC instruction.) The access monitor is then accessed using a MOVES instruction, which performs a bus cycle using the function code in the DFC or the SFC register.

Monitoring for a block of module RAM is enabled by writing all 1s to the starting address of the block in access monitor space. The write operation to enable monitoring can be byte, word, or long word in length (FF, FFFF, or FFFFFFFF, respectively) but must be on a long-word boundary. Once monitoring is enabled, a write operation to any location in the specified block causes a bus error. However, an overwrite of the location occurs, and the original data cannot be recovered.

Monitoring is disabled, or not selected, for a block by writing 0s to the starting address of the block in access monitor space. The enabled or disabled status for a block can be determined by reading the block starting address in the access monitor space and examining the most-significant bit of the data returned. The read can be a byte, word, or long-word access but must be on a long-word boundary. A 1 indicates monitoring is enabled; 0 indicates that it is disabled.

At powerup, monitoring is randomly enabled and disabled; therefore, during system initialization, the access monitor should be programmed as required by the application or cleared (monitoring disabled for all blocks). Sample code segments for clearing and initializing the access monitor are given in the following example:

A Global Memory Module is located in global memory block 7, addresses 00,400,000 through 00,47F,FFF hex. The access monitor is strapped for monitoring 4K-size blocks.

```
* clear the monitor:  disable monitoring for all blocks

            movea.l #00400000,a0     * pointer to block
            clr.l d0                 * d0 = all 0s (value to disable monitor)
            moveq.l #3,d1            * access monitor function code to d1
clr_next:   movec dfc,d2             * save normal function code
            movec d1,dfc             * access monitor function code to dfc
            moves.b d0,(a0)          * write to monitor to disable
            movec d2,dfc             * restore normal function code
            adda.l #1000,a0          * add block size (4K) to pointer
            cmpa.l #00480000,a0      * start of next global block (8) ?
            blt clr_next             * if still block 7, disable next 4K block
```

Enable monitoring for the first 64K of the memory block (16 4K blocks):

```
* enable monitoring for first 64K

            movea.l #00400000,a0     * pointer to block
            moveq.l #FFFFFFFF,d0     * d0 = all 1s (value to enable monitor)
            clr.l d1                 * initialize block counter
            moveq.l #3,d2            * access monitor function code to d2
en_next:    movec dfc,d3             * save normal function code
            movec d2,dfc             * access monitor function code to dfc
            moves.b d0,(a0)          * write to monitor to enable
            movec d3,dfc             * restore normal function code
            addq.l #1,d1             * increment block counter
            adda.l #1000,a0          * add block size (4K) to pointer
            cmpa.l #10,d1            * 16 (0x10) blocks done ?
            bne en_next              * if < 16, enable next block
```

5.2 LOCAL MEMORY EXPANSION MODULE

The Local Memory Expansion Module provides 2M of battery-backed RAM storage on a Processor Module local expansion bus. The address monitor on the host Processor Module can be used for monitoring accesses to this memory space if required.

Within Processor Module address space, six blocks of 2M each are reserved for Local Memory Expansion Modules. Modules may be placed in any of the blocks; hardware jumpers provide proper address decoding. Although there is address space for six Local Memory Expansion Modules, the electrical drive capability of a Processor Module supports a maximum of four modules.

Table 5-3 summarizes this Local Memory Expansion Module address space.

TABLE 5-3. LOCAL MEMORY EXPANSION ADDRESSES

MEMORY BLOCK	START ADDRESS*	END ADDRESS*
1	0n,200,000	0n,3FF,FFF
2	0n,400,000	0n,5FF,FFF
3	0n,600,000	0n,7FF,FFF
4	0n,800,000	0n,9FF,FFF
5	0n,A00,000	0n,BFF,FFF
6	0n,C00,000	0n,DFF,FFF

*Addresses are in hex; n = Processor ID.
These modules can also be addressed by the host Processor Module with the local Processor Module alias (Subsection 2.2.4), i.e., 80,200,000 through 80,3FF,FFF, etc.

6. SYSTEM I/O MODULE

The SANDAC V System I/O Module attaches to a Processor Module's local expansion bus and provides the following resources:

- 1 parallel port
- 5 async/sync serial ports
- 4 16-bit counter/timers
- 4 8-bit counter/timers
- 2 baud rate generators
- 8 input discretes
- 8 output discretes
- 8 input/output/interrupt discretes
- 4 DMA channels
- 4 digital phase-locked loops (DPLLs)
- 4 general-purpose isolated input buffers
- 14 general-purpose nonisolated input/output buffers

The parallel port is implemented by buffering 12 address lines, the module's 16-bit data bus, and the appropriate control signals to an external connector. The serial ports, counter/timers, DPLLs, and I/O discretes are components of two 68562 Dual Universal Serial Communications Controllers (DUSCC) and a 68901 Multi-Function Peripheral (MFP). The DMA channels are implemented with a 63450 DMA Controller (DMAC).

The architecture of the System I/O Module is shown in Figure 6-1. The local expansion bus interface and the major functional components are described in the paragraphs that follow.

6.1 LOCAL BUS INTERFACE

The System I/O Module is viewed by the Processor Module as a 16-bit data port on the local expansion bus. The 16-bit System I/O Module data bus connects to data bits 16 through 31 of the local expansion bus as required for 68020 dynamic bus sizing.

The module is mapped into a 16K byte area within the Processor Module's address space. Table 6-1 summarizes the address assignments for System I/O Module resources.

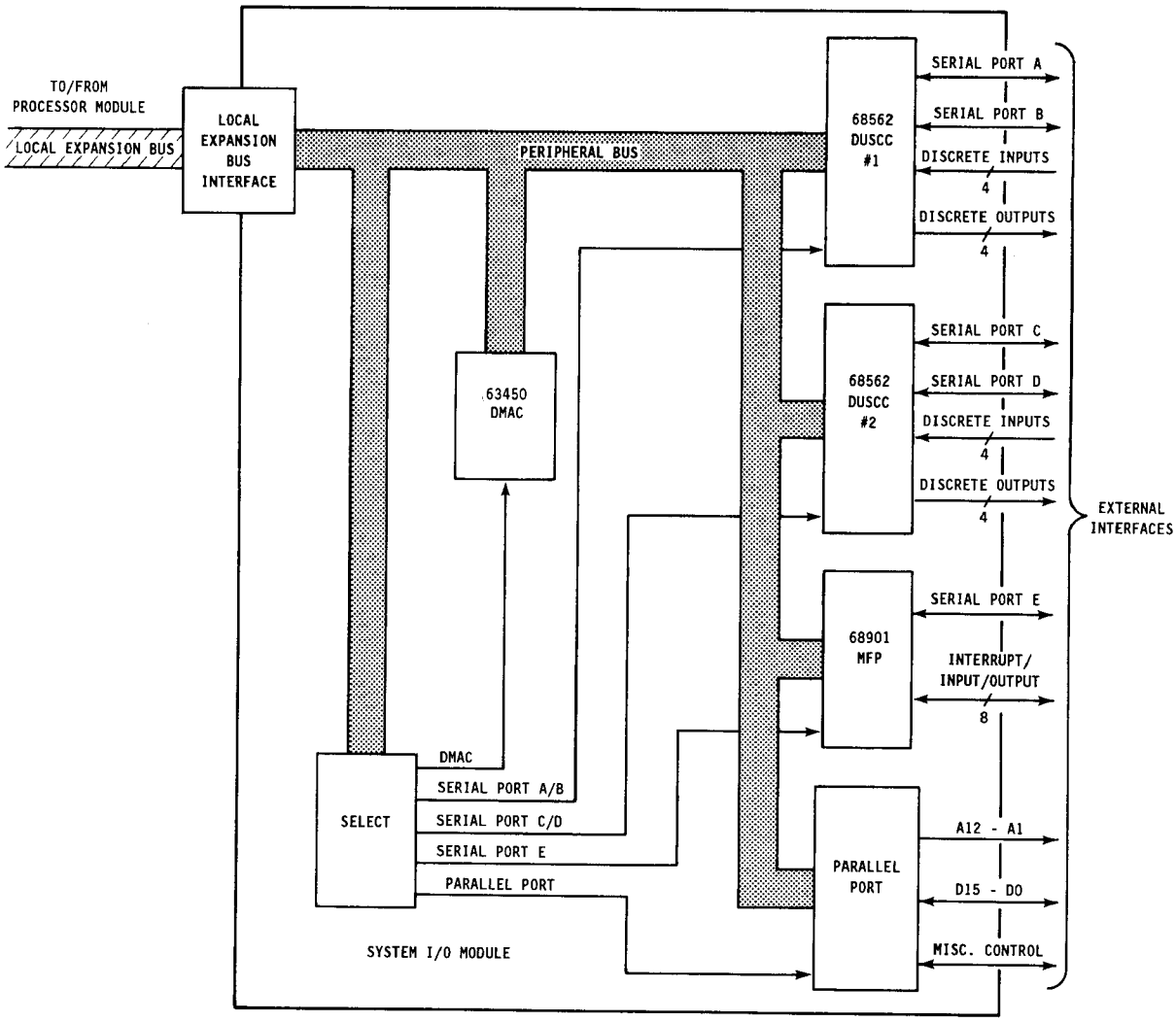


Figure 6-1. System I/O Module Architecture

TABLE 6-1. SYSTEM I/O MODULE ADDRESSES

ASSIGNMENT	START ADDRESS*	END ADDRESS*
Parallel Port	0n,FFC,000	0n,FFD,FFF
63450 DMAC	0n,FFE,000	0n,FFE,0FF
Unassigned	0n,FFE,100	0n,FFE,1FF
68562 Port A	0n,FFE,200	0n,FFE,23F
68562 Port B	0n,FFE,240	0n,FFE,27F
Unassigned	0n,FFE,280	0n,FFE,2FF
68562 Port C	0n,FFE,300	0n,FFE,33F
68562 Port D	0n,FFE,340	0n,FFE,37F
Unassigned	0n,FFE,380	0n,FFE,3FF
68901 MFP	0n,FFE,400	0n,FFE,43F
68901 Alias**	0n,FFE,440	0n,FFE,47F
Unassigned	0n,FFE,480	0n,FFF,FFF

*Addresses are in hexadecimal; "n" is the Processor ID.
**68901 registers are also selected in this address space.

6.2 PARALLEL PORT

The parallel port is a wordwide data port for memory-mapped access to external devices. The port's 16 data lines, 12 address lines, and associated control lines allow the local MPU, any global MPU, or the System I/O Module's DMAC to perform word-length or byte-length read and write operations at any addresses within the 8K byte address space. The uses of the memory locations within parallel port address space are defined by the device connected to the port.

6.3 68562 DUSCCs

The System I/O Module contains two Signetics SCN68562 Dual Universal Serial Communications Controllers (DUSCCs). These devices, DUSCC #1 and DUSCC #2, provide the module with the following resources:

DUSCC #1:	Serial Ports A, B
	2 16-bit Counter/Timers
	4 General-Purpose Input Lines
	4 General-Purpose Output Lines
	2 Digital Phase-Locked Loops
DUSCC #2:	Serial Ports C, D
	2 16-bit Counter/Timers
	4 General-Purpose Input Lines
	4 General-Purpose Output Lines
	2 Digital Phase-Locked Loops

The serial ports (System I/O Module ports A, B, C, and D) spport the following protocols:

- Asynchronous (ASYNC)
- Synchronous character-oriented protocol (COP)
- Bit-oriented protocol (BOP)

A port's transmit and receive clocks are independent and may be provided by the same or different sources. Available sources are on-chip bit rate generators, digital phase-locked loops, counter/timers, or an external source.

Each DUSCC contains two 16-bit counter/timers. As mentioned above, a counter/timer can supply the transmit and/or receive clocks for either channel or for external devices. If it is not used this way, it can be used in any of the following capacities:

- Event counter
- Character counter
- Delay generator
- Bit length measurement

Each of the two communication channels within a DUSCC has four dual-purpose input lines and four dual-purpose output lines. When a channel is used without DMA the input and output lines can be used to sense or control discrete signals. With DMA, the input and output lines provide the request and acknowledge handshake signals for implicit DMA transfers.

Complete information on the 68562 is in the Signetics publication, *SCN68562 Dual Universal Serial Communications Controller (DUSCC).* DUSCC register addressing on the System I/O Module is summarized in Tables 6-2 and 6-3.

TABLE 6-2. 68562 DUSCC ADDRESS MAP

Register	Acronym	Offset*	Mode**
Channel Mode Reg. 1	CMR1	01	R/W
Channel Mode Reg. 2	CMR2	03	R/W
SYN 1/Secondary Addr. 1 Reg.	S1R	05	R/W
SYN 2/Secondary Addr. 2 Reg.	S2R	07	R/W
Transmitter Parameter Reg.	TPR	09	R/W
Transmitter Timing Reg.	TTR	0B	R/W
Receiver Parameter Reg.	RPR	0D	R/W
Receiver Timing Reg.	RTR	0F	R/W
Counter/Timer Preset Reg. High	CTPRH	11	R/W
Counter/Timer Preset Reg. Low	CTPRL	13	R/W
Counter/Timer Control Reg.	CTCR	15	R/W
Output/ Misc. Reg.	OMR	17	R/W
Counter/Timer High	CTH	19	R/W
Counter/Timer Low	CTL	1B	R/W
Pin Configuration Reg.	PCR	1D	R/W
Channel Command Reg.	CCR	1F	R/W
Transmitter FIFO	TXFIFO	21-27	W
Receiver FIFO	RXFIFO	29-2F	R
Receiver Status Reg.	RSR	31	R/W***
Transmitter/Receiver Stat. Reg.	TRSR	33	R/W***
Input/Counter/Timer Stat. Reg.	ICTSR	35	R/W***
General Status Reg.	GSR#	37	R/W***
Interrupt Enable Reg.	IER	39	R/W
Not Used	-	3B	
Interrupt Vect. Reg. (Unmodified)	IVR	3D##	R/W
Interrupt Vect. Reg. (Modified)	IVRM	7D##	R
Interrupt Control Reg.	ICR	3F##	R/W

*Offset in hex from serial port (channel) base address:

Port A: 0n,FFE,200 hex (n is the Processor ID)
Port B: 0n,FFE,240
Port C: 0n,FFE,300
Port D: 0n,FFE,340

**Read (R), Write (W)

***A write to this register may reset status.

#The GSR may be accessed as either DUSCC channel.

##Offset is from device base address (0n,FFE,200 for Ch. A, B; 0n,FFE,300 for Ch. C, D)

TABLE 6-3. 68562 DUSCC REGISTER ADDRESSING

68562 DUSCC Register	Address* by System I/O Module Serial Port (DUSCC # & Ch.) Port A (#1, Ch. A)	Port B (#1, Ch. B)	Port C (#2, Ch. A)	Port D (#2, Ch. B)
CMR1	0n,FFE,201	0n,FFE,241	0n,FFE,301	0n,FFE,341
CMR2	0n,FFE,203	0n,FFE,243	0n,FFE,303	0n,FFE,343
S1R	0n,FFE,205	0n,FFE,245	0n,FFE,305	0n,FFE,345
S2R	0n,FFE,207	0n,FFE,247	0n,FFE,307	0n,FFE,347
TPR	0n,FFE,209	0n,FFE,249	0n,FFE,309	0n,FFE,349
TTR	0n,FFE,20B	0n,FFE,24B	0n,FFE,30B	0n,FFE,34B
RPR	0n,FFE,20D	0n,FFE,24D	0n,FFE,30D	0n,FFE,34D
RTR	0n,FFE,20F	0n,FFE,24F	0n,FFE,30F	0n,FFE,34F
CTPRH	0n,FFE,211	0n,FFE,251	0n,FFE,311	0n,FFE,351
CTPRL	0n,FFE,213	0n,FFE,253	0n,FFE,313	0n,FFE,353
CTCR	0n,FFE,215	0n,FFE,255	0n,FFE,315	0n,FFE,355
OMR	0n,FFE,217	0n,FFE,257	0n,FFE,317	0n,FFE,357
CTH	0n,FFE,219	0n,FFE,259	0n,FFE,319	0n,FFE,359
CTL	0n,FFE,21B	0n,FFE,25B	0n,FFE,31B	0n,FFE,35B
PCR	0n,FFE,21D	0n,FFE,25D	0n,FFE,31D	0n,FFE,35D
CCR	0n,FFE,21F	0n,FFE,25F	0n,FFE,31F	0n,FFE,35F
TXFIFO	0n,FFE,221**	0n,FFE,261**	0n,FFE,321**	0n,FFE,361**
RXFIFO	0n,FFE,229**	0n,FFE,269**	0n,FFE,329**	0n,FFE,369**
RSR	0n,FFE,231	0n,FFE,271	0n,FFE,331	0n,FFE,371
TRSR	0n,FFE,233	0n,FFE,273	0n,FFE,333	0n,FFE,373
ICTSR	0n,FFE,235	0n,FFE,275	0n,FFE,335	0n,FFE,375
GSR	0n,FFE,237	0n,FFE,277	0n,FFE,337	0n,FFE,377
IER	0n,FFE,239	0n,FFE,279	0n,FFE,339	0n,FFE,379
IVR	0n,FFE,23D	0n,FFE,23D	0n,FFE,33D	0n,FFE,33D
IVRM	0n,FFE,27D	0n,FFE,27D	0n,FFE,37D	0n,FFE,37D
ICR	0n,FFE,23F	0n,FFE,23F	0n,FFE,33F	0n,FFE,33F

*Addresses are in hexadecimal; n is the Processor ID.
**FIFOs are four consecutive odd address locations.

CHANNEL MODE REGISTER 1
(DUSCC #1: CMR1A, CMR1B; DUSCC #2: CMR1C, CMR1D)

7	6	5	4	3	2	1	0
DATA ENCODING		EXTND CONTROL	ADDRESS MODE (BOP)		CHANNEL PROTOCOL MODE		
00 - NRZ/MANCHSTR 01 - NRZI 10 - FM0 11 - FM1		BOP ONLY 0 - NO 1 - YES	00 - 8 BIT 01 - EXTND ADDR. 10 - 16 BIT 11 - 16 BIT W/GRP		000 - BOP PRIMARY 001 - BOP SECONDARY 010 - BOP LOOP 011 - BOP LOOP (NO ADDR. COMP.) 100 - COP DUAL SYN 101 - COP DUAL SYN (BISYNC) 110 - COP SINGLE SYN 111 - ASYNCHRONOUS		
		PARITY*	PARITY MODE (COP/ASYNC)				
		0 - EVEN 1 - ODD	00 - NO PARITY 01 - RESERVED 10 - WITH PARITY 11 - FORCE PARITY				

*For BISYNC: 0 = EBCDIC, 1 = 8-bit ASCII

Addresses
Port A: On,FFE,201 hex (n = Processor ID in hex)
Port B: On,FFE,241
Port C: On,FFE,301
Port D: On,FFE,341

Access Level: Supervisor
Initial State: 00 hex

CHANNEL MODE REGISTER 2
(DUSCC #1: CMR2A, CMR2B; DUSCC #2: CMR2C, CMR2D)

7	6	5	4	3	2	1	0
CHANL CONNECT		DATA XFER INTERFACE			FRAME CHECK SEQUENCE SEL.		
00 - NORMAL 01 - AUTO ECHO 10 - LOCAL LOOP 11 - RESERVED		000 - HALF DUPLEX SINGLE ADDRESS DMA 001 - HALF DUPLEX DUAL ADDRESS DMA 010 - FULL DUPLEX SINGLE ADDRESS DMA 011 - FULL DUPLEX DUAL ADDRESS DMA 100 - WAIT ON RX ONLY 101 - WAIT ON TX ONLY 110 - WAIT ON RX OR TX 111 - POLLED OR INTERRUPT			000 - NONE 001 - RESERVED 010 - LRC 8 PRESET 0s 011 - LRC 8 PRESET 1s 100 - CRC 16 PRESET 0s 101 - CRC 16 PRESET 1s 110 - CRC CCITT PRESET 0s 111 - CRC CCITT PRESET 1s		

Addresses
Port A: On,FFE,203 hex (n = Processor ID in hex)
Port B: On,FFE,243
Port C: On,FFE,303
Port D: On,FFE,343

Access Level: Supervisor
Initial State: 00 hex

SYN1/SECONDARY ADDRESS REGISTER 1
(DUSCC #1: S1RA, S1RB; DUSCC #2: S1RC, S1RD)

7	6	5	4	3	2	1	0
ASYNC - CHARACTER COMPARE (5 - 8 BITS) COP - SYN1 (5 - 8 BITS) BOP - FIRST ADDRESS OCTET							

Addresses
Port A: On,FFE,205 hex (n = Processor ID in hex)
Port B: On,FFE,245
Port C: On,FFE,305
Port D: On,FFE,345

Access Level: Supervisor
Initial State: Indeterminate

SYN2/SECONDARY ADDRESS REGISTER 2
(DUSCC #1: S2RA, S2RB; DUSCC #2: S2RC, S2RD)

7	6	5	4	3	2	1	0
ASYNC - NOT USED COP - SYN2 (5 - 8 BITS) BOP - SECOND ADDRESS OCTET							

Addresses
Port A: On,FFE,207 hex (n = Processor ID in hex)
Port B: On,FFE,247
Port C: On,FFE,307
Port D: On,FFE,347

Access Level: Supervisor
Initial State: Indeterminate

PIN CONFIGURATION REGISTER
(DUSCC #1: PCRA, PCRB; DUSCC #2: PCRC, PCRD)

7	6	5	4	3	2	1	0
X2/IDC	GPO2/RTS	SYNOUT/ RTS	RTXC PIN		TRXC PIN		
0 - X2 1 - IDC (PCRA, PCRC ONLY)	0 - GPO2 1 - RTS	0 - SYNO 1 - RTS	00 - INPUT 01 - C/T 10 - TXCLK 1X 11 - RXCLK 1X		000 - INPUT 001 - XTAL/2 010 - DPLL 011 - C/T 100 - TXCLK 16X 101 - RXCLK 16X 110 - TXCLK 1X 111 - RXCLK 1X		

Addresses
Port A: On,FFE,21D hex (n = Processor ID in hex)
Port B: On,FFE,25D
Port C: On,FFE,31D
Port D: On,FFE,35D

Access Level: Supervisor
Initial State: 00 hex

TRANSMITTER PARAMETER REGISTER
(DUSCC #1: TPRA, TPRB; DUSCC #2: TPRC, TPRD)

7	6	5	4	3	2	1	0
UNDERRUN CONTROL		IDLE	TEOM ON ZERO CNT OR DONE	TX RTS CONTROL	CTS ENABLE TX	TX CHARACTER LENGTH	
COP 00 - FCS-IDLE 01 - RESERVED 10 - MARKs 11 - SYNs BOP 00 - FCS-FLAG-IDLE 01 - RESERVED 10 - ABORT-MARKs 11 - ABORT-FLAGs		COP 0 - MARK 1 - SYN BOP 0 - MARK 1 - FLAG	COP 0 - NO 1 - YES BOP 0 - NO 1 - YES	0 - NO 1 - YES	0 - NO 1 - YES	00 - 5 BITS 01 - 6 BITS 10 - 7 BITS 11 - 8 BITS	

ASYNC STOP BITS PER CHARACTER:

	5 BITS/ CHAR	6, 7, OR 8 BITS/ CHAR
0000	1.063	0.563
0001	1.125	0.625
0010	1.188	0.688
0011	1.250	0.750
0100	1.375	0.875
0110	1.438	0.938
0111	1.500	1.000
1000	1.563	1.563
1001	1.625	1.625
1010	1.688	1.688
1011	1.750	1.750
1100	1.813	1.813
1101	1.875	1.875
1110	1.938	1.938
1111	2.000	2.000

Addresses
Port A: On,FFE,209 hex (n = Processor ID in hex)
Port B: On,FFE,249
Port C: On,FFE,309
Port D: On,FFE,349
Access Level: Supervisor; Initial State: 00 hex

TRANSMITTER TIMING REGISTER
(DUSCC #1: TTRA, TTRB; DUSCC #2: TTRC, TTRD)

7	6	5	4	3	2	1	0
EXTERNAL SOURCE	TRANSMITTER CLOCK SELECT			BIT RATE SELECT			

0 - RTXC	000 - 1X EXTERNAL		BIT RATE
1 - TRXC	001 - 16X EXTERNAL		
	010 - DPLL	0000	50
	011 - BRG	0001	75
	100 - 2X OTHER CHANNEL C/T	0010	110
	101 - 32X OTHER CHNL C/T	0011	134.5
	110 - 2X OWN CHANNEL C/T	0100	150
	111 - 32X OWN CHANNEL C/T	0101	200
		0110	300
		0111	600
		1000	1050
		1001	1200
		1010	2000
		1011	2400
		1100	4800
		1101	9600
		1110	19.2K
		1111	38.4K

Addresses
Port A: On,FFE,20B hex (n = Processor ID in hex)
Port B: On,FFE,24B
Port C: On,FFE,30B
Port D: On,FFE,34B

Access Level: Supervisor
Initial State: Indeterminate

RECEIVER PARAMETER REGISTER
(DUSCC #1: RPRA, RPRB; DUSCC #2: RPRC, RPRD)

7	6	5	4	3	2	1	0
SYN STRP	FCS TO FIFO				DCD ENABLE RX	RX CHARACTER LENGTH	
COP 0 - NO 1 - YES	COP 0 - NO 1 - YES	COP AUTO HUNT & PAD CHK 0 - NO 1 - YES	COP EXT SYNC 0 - NO 1 - YES	COP STRIP PARITY 0 - NO 1 - YES	0 - NO 1 - YES	00 - 5 BITS 01 - 6 BITS 10 - 7 BITS 11 - 8 BITS	
BOP NOT USED	BOP 0 - NO 1 - YES	BOP OVERRUN MODE 0 - HUNT 1 - CONT	BOP NOT USED	BOP ALL PARITY ADDRESS 0 - NO 1 - YES			
ASYNC NOT USED	ASYNC NOT USED	ASYNC NOT USED	ASYNC RX RTS CONTROL 0 - NO 1 - YES	ASYNC STRIP PARITY 0 - NO 1 - YES			

Addresses
Port A: On,FFE,20D hex (n = Processor ID in hex)
Port B: On,FFE,24D
Port C: On,FFE,30D
Port D: On,FFE,34D

Access Level: Supervisor
Initial State: 00 hex

RECEIVER TIMING REGISTER
(DUSCC #1: RTRA, RTRB; DUSCC #2: RTRC, RTRD)

7	6	5	4	3	2	1	0
EXTERNAL SOURCE	RECEIVER CLOCK SELECT			BIT RATE SELECT			
0 - RTXC 1 - TRXC	000 - 1X EXTERNAL 001 - 16X EXTERNAL 010 - BRG 011 - C/T OF CHANNEL 100 - DPLL SOURCE = 64X X1/CLK 101 - DPLL SOURCE = 32X EXTRNL 110 - DPLL SOURCE = 32X BRG 111 - DPLL SOURCE = 32X C/T			0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111		BIT RATE 50 75 110 134.5 150 200 300 600 1050 1200 2000 2400 4800 9600 19.2K 38.4K	

Addresses
Port A: On,FFE,20F hex (n = Processor ID in hex)
Port B: On,FFE,24F
Port C: On,FFE,30F
Port D: On,FFE,34F

Access Level: Supervisor
Initial State: Indeterminate

OUTPUT AND MISC. REGISTER
(DUSCC #1: OMRA, OMRB; DUSCC #2: OMRC, OMRD)

7	6	5	4	3	2	1	0
TX RESIDUAL CHAR. LENGTH			TXRDY ACTIVATE	RXRDY ACTIVATE	OUT 2	OUT 1	RTS
000 - 1 BIT 001 - 2 BITS 010 - 3 BITS 011 - 4 BITS 100 - 5 BITS 101 - 6 BITS 110 - 7 BITS 111 - LENGTH SPECIFIED BY BITS 1 & 0 OF TPRA OR TPRB			0-FIFO NOT FULL 1-FIFO EMPTY	0-FIFO NOT EMPTY 1-FIFO FULL	0-OUTPT HIGH 1-OUTPT LOW	0-OUTPT HIGH 1-OUTPT LOW	0-RTSN NEGATE (HIGH) 1-RTSN ASSERT (LOW)

Addresses
Port A: On,FFE,217 hex (n = Processor ID in hex)
Port B: On,FFE,257
Port C: On,FFE,317
Port D: On,FFE,357

Access Level: Supervisor
Initial State: 00 hex

TRANSMIT FIFO (TXFIFO)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
FIFO CONSISTS OF 4 REGISTERS							

Addresses (4 odd addresses access one FIFO register)
Port A: On,FFE,221 through On,FFE,227 hex (n = Processor ID in hex)
Port B: On,FFE,261 through On,FFE,267
Port C: On,FFE,321 through On,FFE,327
Port D: On,FFE,361 through On,FFE,367

Access Level: Supervisor
Initial State: Indeterminate

RECEIVE FIFO (RXFIFO)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
FIFO CONSISTS OF 4 REGISTERS							

Addresses (4 odd addresses access one FIFO register)
Port A: On,FFE,229 through On,FFE,22F hex (n = Processor ID in hex)
Port B: On,FFE,269 through On,FFE,26F
Port C: On,FFE,329 through On,FFE,32F
Port D: On,FFE,369 through On,FFE,36F

Access Level: Supervisor
Initial State: Indeterminate

COUNTER/TIMER CONTROL REGISTER
(DUSCC #1: CTCRA, CTCRB; DUSCC #2: CTCRC, CTCRD)

7	6	5	4	3	2	1	0
ZERO DETECT INTRPT	ZERO DETECT CONTROL	OUTPUT CONTROL	PRESCALER		CLOCK SOURCE		
0-DISABL 1-ENABLE	0-PRESET 1-CONT.	0-SQUARE 1-PULSE	00 - 1 01 - 16 10 - 32 11 - 64		000 - RTXC PIN 001 - TRXC PIN 010 - X1/CLK DIVIDED BY 4 011 - X1/CLK DIVIDED BY 4 GATED BY RXD 100 - RX BRG 101 - TX BRG 110 - RX CHARACTERS 111 - TX CHARACTERS		

Addresses
Port A: On,FFE,215 hex (n = Processor ID in hex)
Port B: On,FFE,255
Port C: On,FFE,315
Port D: On,FFE,355

Access Level: Supervisor
Initial State: Indeterminate

COUNTER/TIMER PRESET HIGH REGISTER
(DUSCC #1: CTPRHA, CTPRHB; DUSCC #2: CTPRHC, CTPRHD)

7	6	5	4	3	2	1	0
MOST-SIGNIFICANT BITS OF COUNTER/TIMER PRESET VALUE							

Addresses
Port A: On,FFE,211 hex (n = Processor ID in hex)
Port B: On,FFE,251
Port C: On,FFE,311
Port D: On,FFE,351

Access Level: Supervisor
Initial State: Indeterminate

COUNTER/TIMER PRESET LOW REGISTER
(DUSCC #1: CTPRLA, CTPRLB; DUSCC #2: CTPRLC, CTPRLD)

7	6	5	4	3	2	1	0
LEAST SIGNIFICANT BITS OF COUNTER/TIMER PRESET VALUE							

Addresses
Port A: On,FFE,213 hex (n = Processor ID in hex)
Port B: On,FFE,253
Port C: On,FFE,313
Port D: On,FFE,353

Access Level: Supervisor
Initial State: Indeterminate

COUNTER/TIMER HIGH
(DUSCC #1: CTHA, CTHB; DUSCC #2: CTHC, CTHD)

7	6	5	4	3	2	1	0
MOST-SIGNIFICANT BITS OF COUNTER/TIMER							

Addresses
Port A: On,FFE,219 hex (n = Processor ID in hex)
Port B: On,FFE,259
Port C: On,FFE,319
Port D: On,FFE,359

Access Level: Supervisor
Initial State: Indeterminate

COUNTER/TIMER LOW (CTLA, CTLB)
(DUSCC #1: CTLA, CTLB; DUSCC #2: CTLC, CTLD)

7	6	5	4	3	2	1	0
LEAST SIGNIFICANT BITS OF COUNTER/TIMER							

Addresses
Port A: On,FFE,21B hex (n = Processor ID in hex)
Port B: On,FFE,25B
Port C: On,FFE,31B
Port D: On,FFE,35B

Access Level: Supervisor
Initial State: Indeterminate

TRANSMITTER AND RECEIVER STATUS REGISTER
(DUSCC #1: TRSRA, TRSRB; DUSCC #2: TRSRC, TRSRD)

7	6	5	4	3	2	1	0
XMTR EMPTY	CTS UNDERRUN	FRAME COMPLETE		DPLL ERROR			
		ASYNC	ASYNC		ASYNC	ASYNC	ASYNC
		NOT USED	SEND BREAK ACK		NOT USED	NOT USED	NOT USED
			COP		COP	COP	COP
			SEND SOM ACK		NOT USED	RX HUNT MODE	RX XPNT MODE
			BOP		BOP		
			SEND SOM/ ABORT ACK		RX CHARACTER LENGTH 000 - 0 BITS 001 - 1 BIT 010 - 2 BITS 011 - 3 BITS 100 - 4 BITS 101 - 5 BITS 110 - 6 BITS 111 - 7 BITS		

Addresses
Port A: On,FFE,233 hex (n = Processor ID in hex)
Port B: On,FFE,273
Port C: On,FFE,333
Port D: On,FFE,373

Access Level: Supervisor
Initial State: 00 hex

RECEIVER STATUS REGISTER
(DUSCC #1: RSRA, RSRB; DUSCC #2: RSRC, RSRD)

7	6	5	4	3	2	1	0
ASYNC* CHAR COMPARE%	ASYNC RTS NEGATED	ASYNC OVERRUN ERROR	ASYNC NOT USED	ASYNC BRK END DETECT	ASYNC BRK START DETECT	ASYNC FRAMING ERROR%	ASYNC PARITY ERROR%
COP* EOM DETECT%+	COP PAD ERROR+	COP OVERRUN ERRROR	COP NOT USED	COP NOT USED	COP SYN DETECT	COP CRC ERROR%	COP PARITY ERROR%
BOP* EOM DETECT%	BOP ABORT DETECT	BOP OVERRUN ERROR	BOP SHORT FRAME DETECT	BOP IDLE DETECT	BOP FLAG DETECT	BOP CRC ERROR%	BOP RCL NOT ZERO%
LOOP* EOM DETECT%	LOOP ABORT/ EOP DETECT	LOOP OVERRUN ERROR	LOOP SHORT FRAME DETECT	LOOP TURN-AROUND DETECT	LOOP FLAG DETECT	LOOP CRC ERROR%	LOOP RCL NOT ZERO%

%Status bit is FIFOed
+BISYNC mode only
*All modes indicate character count complete

Addresses
Port A: On,FFE,231 hex (n = Processor ID in hex)
Port B: On,FFE,271
Port C: On,FFE,331
Port D: On,FFE,371

Access Level: Supervisor
Initial State: 00 hex

INPUT AND COUNTER/TIMER STATUS REGISTER
(DUSCC #1: ICTSRA, ICTSRB; DUSCC #2: ICTSRC, ICTSRD)

7	6	5	4	3	2	1	0
C/T RUNNING	C/T ZERO COUNT	DELTA DCD	DELTA CTS/LC	DCD	CTS/LC	GP12	GP11
0-C/T STOP'D 1-C/T STRT'D	1 AT ZERO CNT	1 AT DCD STATE CHANGE	1 AT CTS CHANGE BOP LOOP 1 AT LC CHANGE	CURRENT STATE OF INPUT PINS			

Addresses
Port A: On,FFE,235 hex (n = Processor ID in hex)
Port B: On,FFE,275
Port C: On,FFE,335
Port D: On,FFE,375

Access Level: Supervisor
Initial State: 00 hex

INTERRUPT ENABLE REGISTER
(DUSCC #1: IERA, IERB; DUSCC #2: IERC, IERD)

7	6	5	4	3	2	1	0
DCD/CTS	TXRDY	TRSR [7:3]	RXRDY	RSR [7:6]	RSR [5:4]	RSR [3:2]	RSR [1:0]
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

Addresses
Port A: On,FFE,239 hex (n = Processor ID in hex)
Port B: On,FFE,279
Port C: On,FFE,339
Port D: On,FFE,379

Access Level: Supervisor
Initial State: 00 hex

INTERRUPT VECTOR REGISTER (IVR)

7	6	5	4	3	2	1	0
INTERRUPT VECTOR							

Addresses
Port A: On,FFE,23D hex (n = Processor ID in hex)
Port B: On,FFE,23D
Port C: On,FFE,33D
Port D: On,FFE,33D

Access Level: Supervisor
Initial State: 0F hex

INTERRUPT VECTOR REGISTER, MODIFIED (IVRM)

7	6	5	4	3	2	1	0
INTERRUPT VECTOR, MODIFIED							

Addresses
Port A: On,FFE,27D hex (n = Processor ID in hex)
Port B: On,FFE,27D
Port C: On,FFE,37D
Port D: On,FFE,37D

Access Level: Supervisor
Initial State: 0F hex

GENERAL STATUS REGISTER (GSR)

7	6	5	4	3	2	1	0
CHANNELS B/D				CHANNELS A/C			
EXTERNAL OR C/T STATUS	RX/TX STATUS	TXRDY	RXRDY	EXTERNAL OR C/T STATUS	RX/TX STATUS	TXRDY	RXRDY

Addresses
Port A: On,FFE,237 hex (n = Processor ID in hex)
Port B: On,FFE,277
Port C: On,FFE,337
Port D: On,FFE,377
(The same register is accessed as either channel.)

Access Level: Supervisor
Initial State: 00 hex

INTERRUPT CONTROL REGISTER (ICR)

7	6	5	4	3	2	1	0
CHANNEL INTERRUPT PRIORITY		VECTOR MODE		BITS TO MODIFY	VECTOR INCLUDES STATUS	CHNL A/C MASTER INT ENBL	CHNL B/D MASTER INT ENBL
00 - CHANNEL A/C 01 - CHANNEL B/D 10 - INTERLEAVED 11 - INTERLEAVED		00 - VECTORED 01 - VECTORED 10 - VECTORED 11 - NON VECTORED		0 - 2:0 1 - 4:2	0 - NO 1 - YES	0 - NO 1 - YES	0 - NO 1 - YES

Addresses
Port A: On,FFE,23F hex (n = Processor ID in hex)
Port B: On,FFE,23F
Port C: On,FFE,33F
Port D: On,FFE,33F

Access Level: Supervisor
Initial State: 00 hex

CHANNEL COMMAND REGISTER
(DUSCC #1: CCRA, CCRB; DUSCC #2: CCRC, CCRD)

7	6	5	4	3	2	1	0
00 = TRANSMITTER CMD		DON'T CARE	DON'T CARE	TRANSMITTER COMMAND 0000 - RESET TX 0001 - RESET TX CRC* 0010 - ENABLE TX 0011 - DISABLE TX 0100 - TRANSMIT SOM (TSOM) 0101 - TRANSMIT SOM WITH PAD(TSOMP) 0110 - TRANSMIT EOM (TEOM)* 0111 - TRANSMIT ABORT/BREAK (TABRK) 1000 - TRANSMIT DLE (TDLE)* 1001 - GO ACTIVE ON POLL 1010 - RESET GO ACTIVE ON POLL 1011 - GO ON-LOOP 1100 - GO OFF-LOOP 1101 - EXCLUDE FROM CRC*			
01 = RECEIVER CMD		DON'T CARE	DON'T CARE	RECEIVER COMMAND 0000 - RESET RX 0001 - RESERVED 0010 - ENABLE RX 0011 - DISABLE RX			
10 = C/T CMD		DON'T CARE	DON'T CARE	COUNTER/TIMER COMMAND 0000 - START 0001 - STOP 0010 - PRESET TO FFFF 0011 - PRESET FROM CTPRH/CTPRL			
11 = DPLL CMD		DON'T CARE	DON'T CARE	DPLL COMMAND 0000 - ENTER SEARCH MODE 0001 - DISABLE DPLL 0010 - SET FM MODE 0011 - SET NRZI MODE 0100 - RESERVED FOR TEST 0101 - RESERVED FOR TEST			

*FIFOed Command

Addresses

Port A: On,FFE,21F hex (n = Processor ID in hex)
Port B: On,FFE,25F; Port C: On,FFE,31F; Port D: On,FFE,35F
Access Level: Supervisor; Initial State: Indeterminate

6.4 68901 MFP

The 68901 Multi-Function Peripheral (MFP) is a Motorola 68000-family device that provides the following resources for MPU support:

- 4 timers (Timers A, B, C, D)
- 1 Universal Synchronous/Asynchronous Receiver-Transmitter (USART)
- 8 programmable I/O lines with interrupt capability
- 1 16-source interrupt controller

These resources and a 68901 programming model are described in the following paragraphs. Complete information on the 68901 is in the Motorola publication, *MC68901 Multi-Function Peripheral.*

6.4.1 Timers

Each of the four 68901 timers is an 8-bit counter with a programmable prescaler. They can be used for general-purpose functions or can supply the transmit and/or receive clocks for the 68901 USART. The Timers A and B can be programmed to operate in any of the following modes:

- Delay
- Pulse width measurement
- Event count

Timers C and D operate only in the delay mode. Refer to the 68901 manual for a description of these modes.

6.4.2 USART

The USART is a full-duplex receiver and transmitter that supports asynchronous and limited synchronous protocols. The transmit and receive clocks can be provided by one or two of the 68901 counter/timers or an external source.

The 68901 USART supports System I/O Module Serial Port E.

6.4.3 Programmable I/O Lines

The programmable I/O lines can function as eight individually programmable I/O lines with interrupt capability. This allows for the configuration of an 8-bit I/O port or 8 discrete input/output/interrupt signals.

6.4.4 Interrupt Controller

The 68901 interrupt controller has 16 prioritized channels for handling interrupt requests. The requests presented to the controller are generated by sources internal to the 68901 and by external sources. The internal sources include the 68901 timers and USART. The external sources are routed to the controller on the eight programmable I/O lines, I0 through I7, and the Timer A external input, TAI.

The 16 interrupt controller channels are prioritized. Concurrent requests are handled highest priority first. The channel assignments for the interrupt controller and their priorities are as follows:

Channel	Interrupt Request	Priority
1111	Input 7 (I7)	Highest
1110	Input 6 (I6)	.
1101	Timer A	.
1100	USART (receive buffer full)	.
1011	USART (receiver error)	.
1010	USART (transmit buffer empty)	.
1001	USART (transmitter error)	.
1000	Timer B	.
0111	Input 5 (I5)	.
0110	Input 4 (I4)	.
0101	Timer C	.
0100	Timer D	.
0011	Input 3 (I3)	.
0010	Input 2 (I2)	.
0001	Input 1 (I1)	.
0000	Input 0 (I0)	Lowest

When interrupt requests are presented to the interrupt controller, the controller responds to the highest priority source and causes the 68901 to generate a single vectored interrupt request. During the 68020 interrupt acknowledge cycle, the 68901 furnishes an interrupt vector with the following format:

68901 Interrupt Vector Format

7	6	5	4	3	2	1	0
V3	V2	V1	V0	CH3	CH2	CH1	CH0

Vn VECTOR - The four bits in this field are copied form the 68901 Vector Register (VR) (refer to the programming model). The bits are loaded into the VR when the 68901 is initialized to designate the interrupt vector the device will use.

CHn CHANNEL - The four bits in this field designate the interrupt controller channel that made the interrupt request.

6.4.5 Programming Model

The address map for the System I/O Module 68901 MFP is given in Table 6-4. The programming model is shown in Figure 6-2 and followed by a register summary.

TABLE 6-4. SYSTEM I/O MODULE 68901 MFP ADDRESS MAP

Register	Acronym	Offset*
General-Purpose Input/Output	GPIP	01
Active Edge	AER	03
Data Direction	DDR	05
Interrupt Enable A	IERA	07
Interrupt Enable B	IERB	09
Interrupt Pending A	IPRA	0B
Interrupt Pending B	IPRB	0D
Interrupt In-Service A	ISRA	0F
Interrupt In-Service B	ISRB	11
Interrupt Mask A	IMRA	13
Interrupt Mask B	IMRB	15
Vector	VR	17
Timer A Control	TACR	19
Timer B Control	TBCR	1B
Timer C and D Control	TCDCR	1D
Timer A Data	TADR	1F
Timer B Data	TBDR	21
Timer C Data	TCDR	23
Timer D Data	TDDR	25
Synchronous Character	SCR	27
USART Control	UCR	29
Receiver Status	RSR	2B
Transmitter Status	TSR	2D
USART Data	UDR	2F

*Offset in hex from Channel E base address 0n,FFE,400.

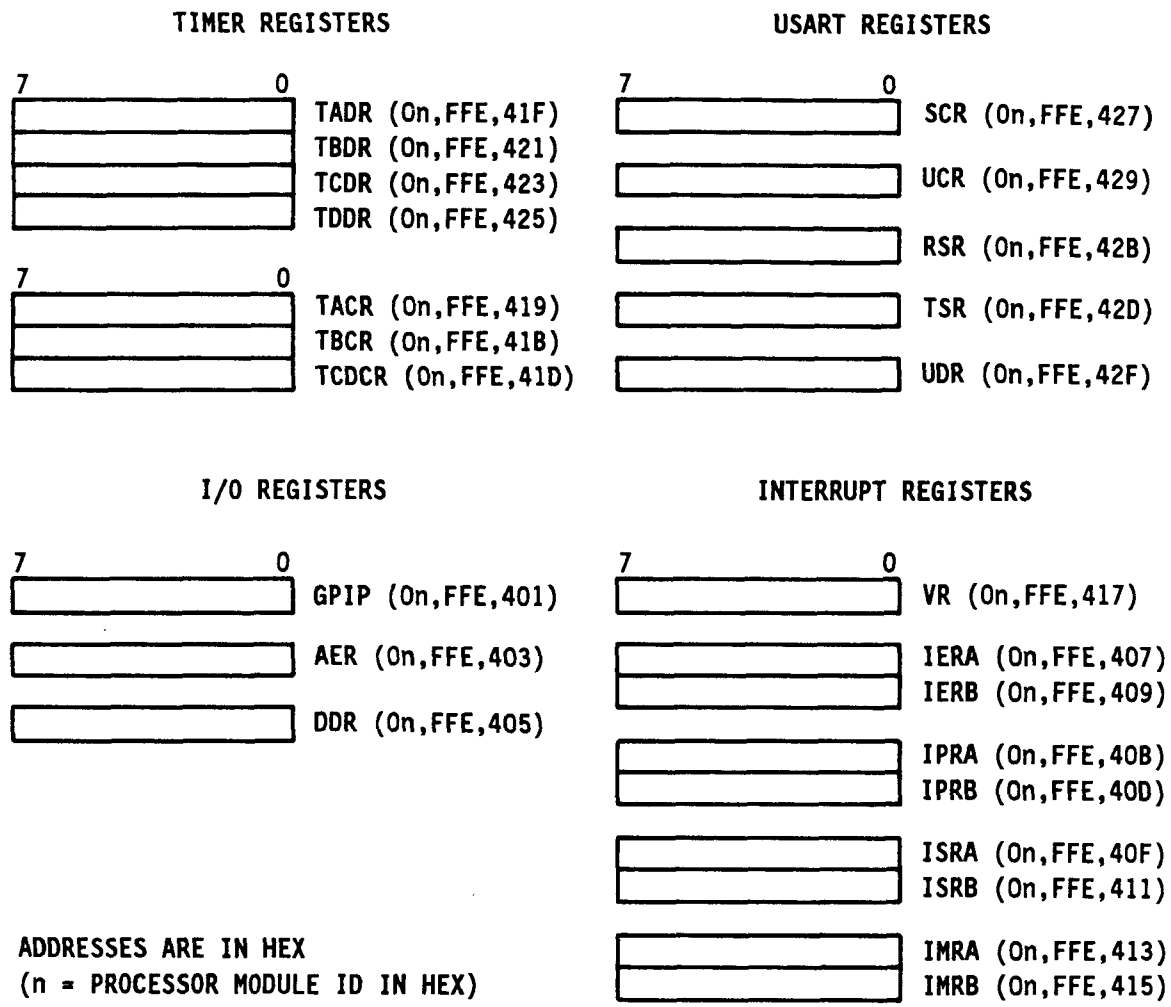


Figure 6-2. System I/O Module 68901 MFP Programming Model

TIMER DATA REGISTERS (TADR, TBDR, TCDR, TDDR)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Addresses

TADR: On,FFE,41F hex (n = Processor Module ID in hex)
TBDR: On,FFE,421 hex
TCDR: On,FFE,423 hex
TDDR: On,FFE,425 hex

Access Level: Supervisor
Initial State: Indeterminate

TIMERS A AND B CONTROL REGISTERS (TACR, TBCR)

7	6	5	4	3	2	1	0
0	0	0	TACR RESET TA0 TBCR RESET TB0	TIMER A/B CONTROL BITS			
				0000 - TIMER STOPPED 0001 - DELAY MODE, ÷4 PRESCALER 0010 - DELAY MODE, ÷10 PRESCALER 0011 - DELAY MODE, ÷16 PRESCALER 0100 - DELAY MODE, ÷50 PRESCALER 0101 - DELAY MODE, ÷64 PRESCALER 0110 - DELAY MODE, ÷100 PRESCALER 0111 - DELAY MODE, ÷200 PRESCALER 1000 - EVENT COUNT MODE 1001 - PULSE WIDTH MODE, ÷4 PSCLR 1010 - PULSE WIDTH MODE, ÷10 PSCLR 1011 - PULSE WIDTH MODE, ÷16 PSCLR 1100 - PULSE WIDTH MODE, ÷50 PSCLR 1101 - PULSE WIDTH MODE, ÷64 PSCLR 1110 - PULSE WIDTH MODE, ÷100 PSCLR 1111 - PULSE WIDTH MODE, ÷200 PSCLR			

Addresses

TACR: On,FFE,419 hex (n = Processor Module ID in hex)
TBCR: On,FFE,41B hex

Access Level: Supervisor
Initial State: 00 hex

TIMERS C AND D CONTROL REGISTER (TCDCR)

7	6	5	4	3	2	1	0
0	TIMER C CONTROL BITS			0	TIMER D CONTROL BITS		
	000-TIMER STOPPED 001-DELAY MODE, ÷4 PSCLR 010-DELAY MODE, ÷10 PSCLR 011-DELAY MODE, ÷16 PSCLR 100-DELAY MODE, ÷50 PSCLR 101-DELAY MODE, ÷64 PSCLR 110-DELAY MODE, ÷100 PSCLR 111-DELAY MODE, ÷200 PSCLR				000-TIMER STOPPED 001-DELAY MODE, ÷4 PSCLR 010-DELAY MODE, ÷10 PSCLR 011-DELAY MODE, ÷16 PSCLR 100-DELAY MODE, ÷50 PSCLR 101-DELAY MODE, ÷64 PSCLR 110-DELAY MODE, ÷100 PSCLR 111-DELAY MODE, ÷200 PSCLR		

Address: On,FFE,41D hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

SYNCHRONOUS CHARACTER REGISTER (SCR)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
SYNCHRONOUS CHARACTER							

Address: On,FFE,427 hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: Indeterminate

USART CONTROL REGISTER (UCR)

7	6	5	4	3	2	1	0
CLOCK	WORD LENGTH		START/STOP BIT, FORMAT CONTROL		PARITY ENABLE	EVEN/ODD PARITY	0
0 - CLK X1 1 - CLK ÷16	00 - 8 BITS 01 - 7 BITS 10 - 6 BITS 11 - 5 BITS		00 - 0 START 0 STOP SYNCHRONOUS 01 - 1 START 1 STOP ASYNCHRONOUS 10 - 1 START 1.5 STOP ASYNCHRONOUS (USED ONLY IF BIT 7=1) 11 - 1 START 2 STOP ASYNCHRONOUS		0-DISABL 1-ENABLE	0 - ODD 1 - EVEN	

Address: On,FFE,429 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

RECEIVER STATUS REGISTER (RSR)

7	6	5	4	3	2	1	0
BUFFER FULL	OVERRUN ERROR	PARITY ERROR (PE)	FRAME ERROR (FE)	FOUND/ SEARCH OR BREAK DETECT	MATCH/ CHAR. IN PROGRESS	SYNCRO. STRIP ENABLE	RECEIVER ENABLED
1-WORD XFERED TO RCV BUFFER 0-RCV BUFFER READ	1-WORD RCVD & RCV BUFFER FULL 0-RSR READ	1-RCVD WORD HAS PE 0-RCVD WORD WITH NO PE	1-RCVD WORD HAS FE 0-RCVD WORD WITH NO FE	SYNCHRO. MODE 1-RCVD WORD MATCH SCR 0-RCVD WORD NO MATCH ASYNC. MODE 1-RCVD WORD IS BREAK 0-BREAK TERMI-NATES, RCV BUFFER READ	SYNCHRO. MODE 1-RCVD WORD MATCH SCR 0-RCVD WORD NO MATCH ASYNC. MODE 1-START BIT DETECT 0-END OF WORD DETECT	1-DISABL 0-ENABLE	1-ENABLE 0-DISABL

Address: On,FFE,42B hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

TRANSMITTER STATUS REGISTER (TSR)

7	6	5	4	3	2	1	0
BUFFER EMPTY	UNDERRUN ERROR	AUTO-TRNARND	END OF XMIT	BREAK	OUTPUT STATE TX DISABLED		XMTR ENABLE
1-XMIT BUFFER EMPTY 0-XMIT BUFFER FULL	1-XMIT SR XMTD BEFORE XMIT BUFFER WRITE 0-TSR READ	1-RCVR ENABLE AFTER XMIT (MPU WRT) 0-XMTR DISABL	1-WHEN TX DISABL AND XMIT DONE	ASYNC. ONLY 1-XMT BREAK 0-STOP BREAK	00 - HIGH IMPED. 01 - LOW 10 - HIGH 11 - LOOPBACK		1-ENABLE 0-DISABL

Address: On,FFE,42D hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

USART DATA REGISTER (UDR)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Address: On,FFE,42F hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: Indeterminate

GPIP DATA REGISTER (GPIP)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Address: On,FFE,401 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: Indeterminate

ACTIVE EDGE REGISTER (AER)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - INTERRUPTS GENERATED ON THE FALLING EDGE OF THE INPUT SIGNAL 1 - INTERRUPTS GENERATED ON THE RISING EDGE OF THE INPUT SIGNAL							

Address: On,FFE,403 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

DATA DIRECTION REGISTER (DDR)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - I/O LINE IS AN INPUT 1 - I/O LINE IS AN OUTPUT							

Address: On,FFE,405 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

VECTOR REGISTER (VR)

7	6	5	4	3	2	1	0
V7	V6	V5	V4	S	0	0	0
MOST-SIGNIFICANT BITS OF INTERRUPT VECTOR NUMBER				0-AUTO. EOI 1-SOFTW. EOI			

Address: On,FFE,417 hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT ENABLE REGISTER A (IERA)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	TIMER A	RCV BUFFER FULL	RCV ERROR	XMIT BUFFER EMPTY	XMIT ERROR	TIMER B
FOR ALL BITS: 0 - INTERRUPT CHANNEL DISABLED 1 - INTERRUPT CHANNEL ENABLED							

Address: On,FFE,407 hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT ENABLE REGISTER B (IERB)

7	6	5	4	3	2	1	0
GPIP5	GPIP4	TIMER C	TIMER D	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - INTERRUPT CHANNEL DISABLED 1 - INTERRUPT CHANNEL ENABLED							

Address: On,FFE,409 hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT PENDING REGISTER A (IPRA)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	TIMER A	RCV BUFFER FULL	RCV ERROR	XMIT BUFFER EMPTY	XMIT ERROR	TIMER B
FOR ALL BITS: 0 - NO INTERRUPT PENDING 1 - INTERRUPT PENDING							

Address: On,FFE,40B hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT PENDING REGISTER B (IPRB)

7	6	5	4	3	2	1	0
GPIP5	GPIP4	TIMER C	TIMER D	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - NO INTERRUPT PENDING 1 - INTERRUPT PENDING							

Address: On,FFE,40D hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT IN-SERVICE REGISTER A (ISRA)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	TIMER A	RCV BUFFER FULL	RCV ERROR	XMIT BUFFER EMPTY	XMIT ERROR	TIMER B
FOR ALL BITS: 0 - NO INTERRUPT PROCESSING IN PROGRESS FOR THE CHANNEL 1 - INTERRUPT PROCESSING IN PROGRESS FOR THE CHANNEL							

Address: On,FFE,40F hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT IN-SERVICE REGISTER B (ISRB)

7	6	5	4	3	2	1	0
GPIP5	GPIP4	TIMER C	TIMER D	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - NO INTERRUPT PROCESSING IN PROGRESS FOR THE CHANNEL 1 - INTERRUPT PROCESSING IN PROGRESS FOR THE CHANNEL							

Address: On,FFE,411 hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT MASK REGISTER A (IMRA)

7	6	5	4	3	2	1	0
GPIP7	GPIP6	TIMER A	RCV BUFFER FULL	RCV ERROR	XMIT BUFFER EMPTY	XMIT ERROR	TIMER B
FOR ALL BITS: 0 - INTERRUPTS ARE MASKED FOR THE CHANNEL 1 - INTERRUPTS ARE NOT MASKED FOR THE CHANNEL							

Address: On,FFE,413 hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

INTERRUPT MASK REGISTER B (IMRB)

7	6	5	4	3	2	1	0
GPIP5	GPIP4	TIMER C	TIMER D	GPIP3	GPIP2	GPIP1	GPIP0
FOR ALL BITS: 0 - INTERRUPTS ARE MASKED FOR THE CHANNEL 1 - INTERRUPTS ARE NOT MASKED FOR THE CHANNEL							

Address: On,FFE,415 hex (n = Processor Module ID in hex)
Access Level: Supervisor
Initial State: 00 hex

6.5 63450 DMAC

The System I/O Module contains one Hitachi HD63450 Direct Memory Access Controller (DMAC). The HD63450 is functionally equivalent to the Motorola MC68450. Complete information on the device is in the Hitachi data sheet or in the Motorola publication, *MC68450 Direct Memory Access Controller*. The following paragraphs describe the DMAC implementation of the System I/O Module and the 63450 programming model.

6.5.1 Implementation

The purpose of the DMAC is to reduce the interrupt load on the MPU servicing I/O ports. The overhead associated with interrupts from I/O ports is replaced by the much shorter time required to gain control of the local bus and make a data transfer.

The 63450 has four DMA channels, each of which provides for the following types of data transfers:

- Memory-to-memory
- Memory-to-device
- Device-to-memory

(In the SANDAC V, memory-to-memory transfer can be accomplished faster by the Processor Module MPU.)

The DMAC performs DMA cycles only on the local bus; it cannot use global memory as a source or destination. During DMA operations, the 63450 provides the lower 24 bits of the source or destination address. The upper 8 bits of the address are a hardwired 80 hex (local Processor Module alias).

6.5.2 Addressing Modes

The 63450 DMAC operates in either the implicit (single address) or the explicit (dual address) transfer mode. On the System I/O Module, the two 68562 DUSCCs are the only devices that support the implicit transfer mode.

Implicitly addressed devices do not require the generation of a device data register address for the data transfer. Devices that support this mode are controlled by the DMAC's device control interface. During a transfer, the DMAC enables a device on the system data bus and provides the required memory address and completes the transfer. Only a single address is required. Implicit addressing is illustrated in Figure 6-3a.

Explicitly addressed devices require that a data register within the peripheral device be addressed. Because the address bus is used to access the device, the data cannot be directly transferred to/from memory, since memory also requires addressing. Therefore,

the data are transferred to an internal holding register in the DMAC and then transferred to the destination during a second bus transfer. Explicit addressing is illustrated in Figure 6-3b.

6.5.3 Restrictions

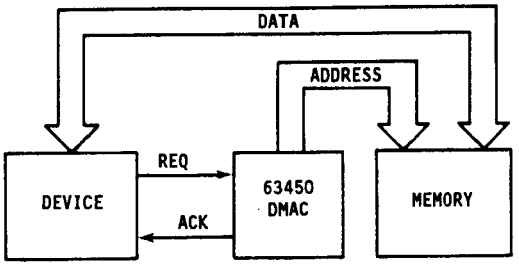
The DMAC should be programmed to operate in the burst mode when used with early 68562 DUSCC devices having no revision level. These devices may generate a spurious pulse on the DMA service request line at the end of a transfer. If the DMAC is in cycle steal mode, this may be recognized as a valid request since the request input is edge sensitive. In burst mode, the input is level sensitive and the pulse will not be recognized since it is negated before the DMAC can arbitrate for the bus. This anomoly has been fixed in 68562 revision A devices.

Array-chained and linked-array-chained transfers are not available in implicit mode because no provision has been made for a READY- input when reading the chaining information from memory.

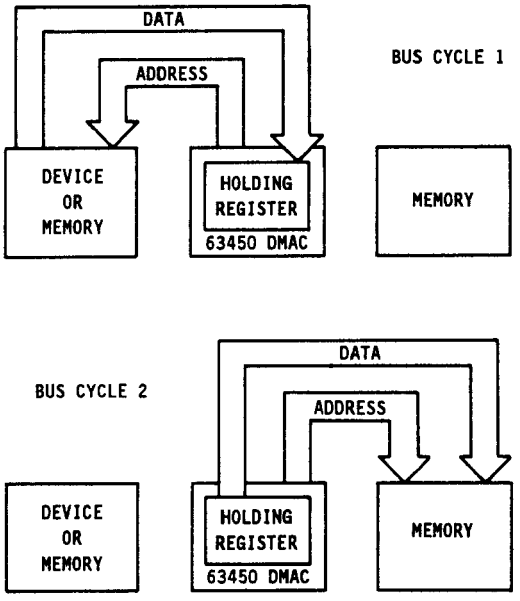
The peripheral control line (PCL) pins should never be programmed as outputs.

6.5.4 Programming Model

The programming model for the 63450 DMAC is shown in Figure 6-4 followed by a register address table (Table 6-5) and register summary.



a. Implicit DMA Addressing

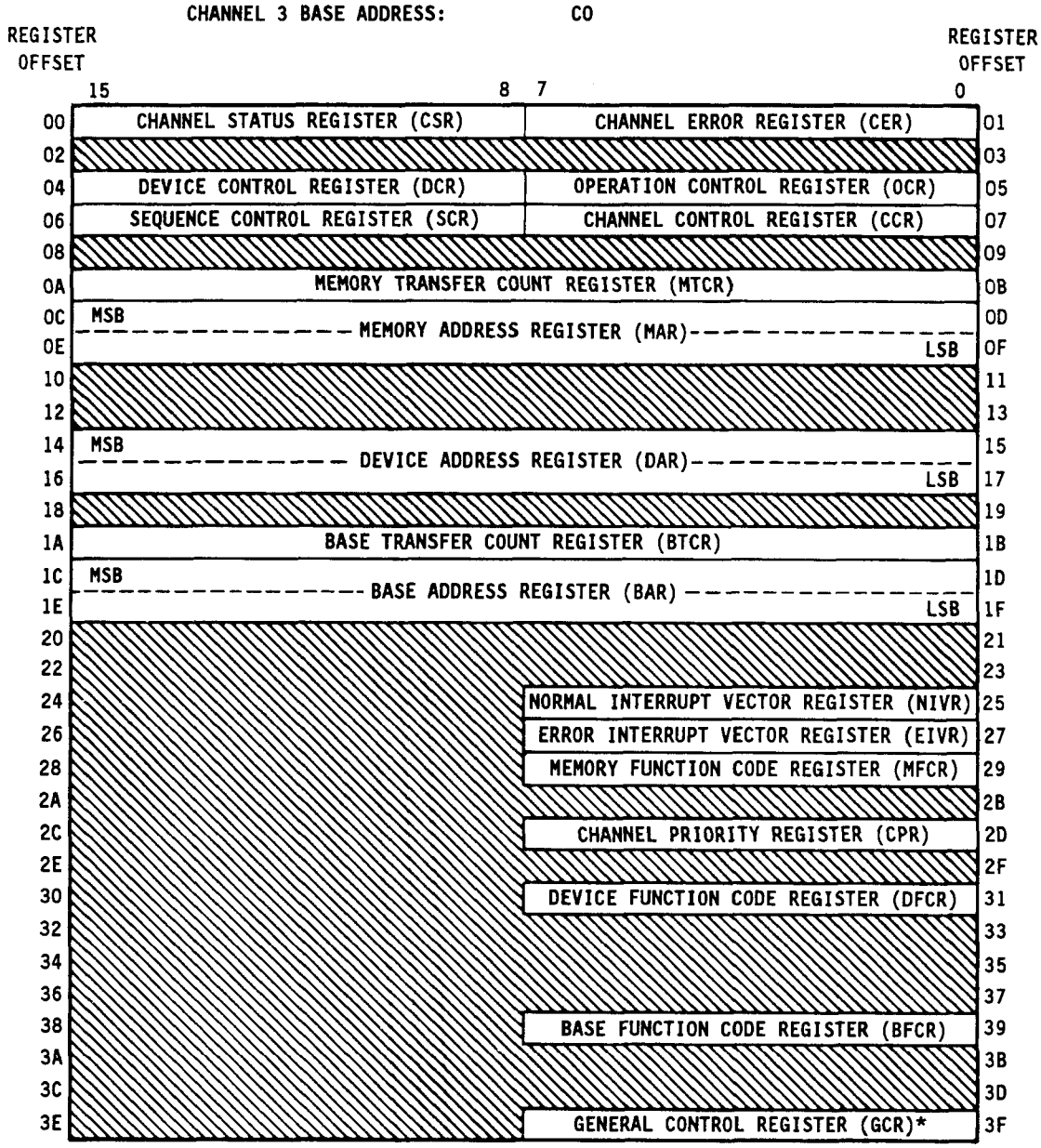


b. Explicit DMA Addressing

Figure 6-3. DMA Addressing Modes

63450 DMAC BASE ADDRESS: On,FFE,000 HEX (n = PROCESSOR ID)
(80,FFE,000)

CHANNEL 0 BASE ADDRESS: 00
CHANNEL 1 BASE ADDRESS: 40
CHANNEL 2 BASE ADDRESS: 80
CHANNEL 3 BASE ADDRESS: C0



*GCR IS LOCATED AT FF ONLY.

Figure 6-4. 63450 DMAC Programming Model

TABLE 6-5. 63450 DMAC REGISTER ADDRESSING

63450 DMAC Register	Size in Bits	Address* by DMAC Channel			
		Channel 0	Channel 1	Channel 2	Channel 3
CSR	8	0n,FFE,000	0n,FFE,040	0n,FFE,080	0n,FFE,0C0
CER	8	0n,FFE,001	0n,FFE,041	0n,FFE,081	0n,FFE,0C1
DCR	8	0n,FFE,004	0n,FFE,044	0n,FFE,084	0n,FFE,0C4
OCR	8	0n,FFE,005	0n,FFE,045	0n,FFE,085	0n,FFE,0C5
SCR	8	0n,FFE,006	0n,FFE,046	0n,FFE,086	0n,FFE,0C6
CCR	8	0n,FFE,007	0n,FFE,047	0n,FFE,087	0n,FFE,0C7
MTCR	16	0n,FFE,00A	0n,FFE,04A	0n,FFE,08A	0n,FFE,0CA
MAR	32	0n,FFE,00C	0n,FFE,04C	0n,FFE,08C	0n,FFE,0CC
DAR	32	0n,FFE,014	0n,FFE,054	0n,FFE,094	0n,FFE,0D4
BTCR	16	0n,FFE,01A	0n,FFE,05A	0n,FFE,09A	0n,FFE,0DA
BAR	32	0n,FFE,01C	0n,FFE,05C	0n,FFE,09C	0n,FFE,0DC
NIVR	8	0n,FFE,025	0n,FFE,065	0n,FFE,0A5	0n,FFE,0E5
EIVR	8	0n,FFE,027	0n,FFE,067	0n,FFE,0A7	0n,FFE,0E7
MFCR	8	0n,FFE,029	0n,FFE,069	0n,FFE,0A9	0n,FFE,0E9
CPR	8	0n,FFE,02D	0n,FFE,06D	0n,FFE,0AD	0n,FFE,0ED
DFCR	8	0n,FFE,031	0n,FFE,071	0n,FFE,0B1	0n,FFE,0F1
BFCR	8	0n,FFE,039	0n,FFE,079	0n,FFE,0B9	0n,FFE,0F9
GCR	8	0n,FFE,0FF	0n,FFE,0FF	0n,FFE,0FF	0n,FFE,0FF

*Addresses are in hexadecimal; "n" is the Processor ID.

DEVICE CONTROL REGISTER (DCR)

7	6	5	4	3	2	1	0
XRM*		DTYP**		DPS	PCL***		
00 - BURST 01 - (UNDEFINED) 10 - CYCLE STEAL WITHOUT HOLD 11 - CYCLE STEAL WITH HOLD		00 - EXPLICIT M68000 01 - EXPLICIT M6800 10 - IMPLICIT w/ACK 11 - IMPLICIT w/ACK & RDY		0-8-BIT 1-16-BIT	000 - STATUS 001 - INTERRUPT 010 - START PULSE 011 - ABORT 100 - UNDEFINED 101 - UNDEFINED 110 - UNDEFINED 111 - UNDEFINED		

*DUSCCs (Rev 0) require XRM = 00 to prevent spurious requests.
**DUSCC implicit mode is with DTYP = 11.
***PCL must be 000 for this implementation.

Addresses
Channel 1: On,FFE,004 hex (n = Processor ID in hex)
Channel 2: On,FFE,044
Channel 3: On,FFE,084
Channel 4: On,FFE,0C4
Access Level: Supervisor
Initial State: Indeterminate

OPERATION CONTROL REGISTER (OCR)

7	6	5	4	3	2	1	0
DIR	BTD	SIZE		CHAIN		REOG	
0-MEMORY TO DEVICE 1-DEVICE TO MEMORY	0-TERMINATE AFTER CURRENT CYCLE 1-RESTART NEXT BLOCK	00 - BYTE 01 - WORD 10 - LONG WORD 11 - BYTE, NO PACKING		00 - DISABLED 01 - (UNDEFINED) 10 - ARRAY CHAINING 11 - LINKED ARRAY CHAINING (CHAINING CANNOT BE USED WITH IMPLICIT ADDR.)		00 - INTERNAL LIMITED RATE 01 - INTERNAL MAXIMUM RATE 10 - EXTERNAL 11 - AUTO START, EXTERNAL	

Addresses
Channel 1: On,FFE,005 hex (n = Processor ID in hex)
Channel 2: On,FFE,045
Channel 3: On,FFE,085
Channel 4: On,FFE,0C5
Access Level: Supervisor; Initial State: 00 hex

SEQUENCE CONTROL REGISTER (SCR)

7	6	5	4	3	2	1	0
0	0	0	0	MAC		DAC	
				00 - NO COUNT 01 - COUNT UP 10 - COUNT DOWN 11 - (UNDEFINED)		00 - NO COUNT 01 - COUNT UP 10 - COUNT DOWN 11 - (UNDEFINED)	

Addresses

Channel 1: On,FFE,006 hex (n = Processor ID in hex)
Channel 2: On,FFE,046
Channel 3: On,FFE,086
Channel 4: On,FFE,0C6

Access Level: Supervisor
Initial State: 00 hex

CHANNEL CONTROL REGISTER (CCR)

7	6	5	4	3	2	1	0
STR	CNT	HLT	SAB	INT	0	0	0
START CHANNEL	CONTINUE OPER.	SOFTWARE HALT	SOFTWARE ABORT	INTERRUP ENABLE			

Addresses

Channel 1: On,FFE,007 hex (n = Processor ID in hex)
Channel 2: On,FFE,047
Channel 3: On,FFE,087
Channel 4: On,FFE,0C7

Access Level: Supervisor
Initial State: 00 hex

CHANNEL STATUS REGISTER (CSR)

7	6	5	4	3	2	1	0
COC	BTC	NDT	ERR	ACT	DIT	PCT	PCS
CHANNEL OPER. COMPLETE	BLOCK TRANSF. COMPLETE	NORMAL DEVICE TERM.	ERROR	CHANNEL ACTIVE	0-NO DONE TRAN-SITION 1-DONE TRANS.	PCL TRAN-SITION	PCL LEVEL

Addresses

Channel 1: On,FFE,000 hex (n = Processor ID in hex)
Channel 2: On,FFE,040
Channel 3: On,FFE,080
Channel 4: On,FFE,0C0

Access Level: Supervisor
Initial State: 00 or 01 hex (Dependent on PCL)

CHANNEL PRIORITY REGISTER (CPR)

7	6	5	4	3	2	1	0
0	0	0	0	0	0	CP	
						00 - PRIORITY 0 (HIGHEST) 01 - PRIORITY 1 10 - PRIORITY 2 11 - PRIORITY 3 (LOWEST)	

Addresses

Channel 1: On,FFE,02D hex (n = Processor ID in hex)
Channel 2: On,FFE,06D
Channel 3: On,FFE,0AD
Channel 4: On,FFE,0ED

Access Level: Supervisor
Initial State: 00 hex

GENERAL CONTROL REGISTER (GCR)

7	6	5	4	3	2	1	0
0	0	0	0	BT		BR	
				00 - 16 CLOCKS 01 - 32 CLOCKS 10 - 64 CLOCKS 11 - 128 CLOCKS		00 - 50.00% 01 - 25.00% 10 - 12.50% 11 - 6.25%	

Address: On,FFE,OFF hex (n = Processor ID in hex)
Access Level: Supervisor
Initial State: 00 hex

CHANNEL ERROR REGISTER (CER)

7	6	5	4	3	2	1	0
0	0	0	ERROR CODE				
			00000 - NO ERROR 00001 - CONFIGURATION ERROR 00010 - OPERATION TIMING ERROR 00011 - UNDEFINED 00100 - UNDEFINED 00101 - ADDRESS ERROR IN MAR 00110 - ADDRESS ERROR IN DAR 00111 - ADDRESS ERROR IN BAR 01000 - UNDEFINED 01001 - BUS ERROR IN MAR 01010 - BUS ERROR IN DAR 01011 - BUS ERROR IN BAR 01100 - UNDEFINED 01101 - COUNT ERROR IN MTCR 01110 - UNDEFINED 01111 - COUNT ERROR IN BTCR 10000 - EXTERNAL ABORT 10001 - SOFTWARE ABORT 10010 - UNDEFINED . - UNDEFINED 11111 - UNDEFINED				

Addresses

Channel 1: On,FFE,001 hex (n = Processor ID in hex)
Channel 2: On,FFE,041
Channel 3: On,FFE,081
Channel 4: On,FFE,0C1

Access Level: Supervisor
Initial State: 00 hex

NORMAL INTERRUPT VECTOR REGISTER (NIVR)

7	6	5	4	3	2	1	0
NORMAL INTERRUPT VECTOR							

Addresses

Channel 1: On,FFE,025 hex (n = Processor ID in hex)
Channel 2: On,FFE,065
Channel 3: On,FFE,0A5
Channel 4: On,FFE,0E5

Access Level: Supervisor
Initial State: 0F hex

ERROR INTERRUPT VECTOR REGISTER (EIVR)

7	6	5	4	3	2	1	0
ERROR INTERRUPT VECTOR							

Addresses

Channel 1: On,FFE,027 hex (n = Processor ID in hex)
Channel 2: On,FFE,067
Channel 3: On,FFE,0A7
Channel 4: On,FFE,0E7

Access Level: Supervisor
Initial State: 0F hex

DEVICE FUNCTION CODE REGISTER (DFCR)

7	6	5	4	3	2	1	0
0	0	0	0	DEVICE FUNCTION CODE 000 - RESERVED 001 - USER DATA SPACE 010 - USER PROGRAM SPACE 011 - RESERVED FOR USER 100 - RESERVED BY MOTOROLA 101 - SUPERVISOR DATA SPACE 110 - SUPERVISOR PROGRAM SPACE 111 - CPU SPACE			

Addresses
Channel 1: On,FFE,031 hex (n = Processor ID in hex)
Channel 2: On,FFE,071
Channel 3: On,FFE,0B1
Channel 4: On,FFE,0F1

Access Level: Supervisor
Initial State: Indeterminate

MEMORY FUNCTION CODE REGISTER (MFCR)

7	6	5	4	3	2	1	0
0	0	0	0	MEMORY FUNCTION CODE 000 - RESERVED 001 - USER DATA SPACE 010 - USER PROGRAM SPACE 011 - RESERVED FOR USER 100 - RESERVED BY MOTOROLA 101 - SUPERVISOR DATA SPACE 110 - SUPERVISOR PROGRAM SPACE 111 - CPU SPACE			

Addresses
Channel 1: On,FFE,029 hex (n = Processor ID in hex)
Channel 2: On,FFE,069
Channel 3: On,FFE,0A9
Channel 4: On,FFE,0E9

Access Level: Supervisor
Initial State: Indeterminate

BASE FUNCTION CODE REGISTER (BFCR)

7	6	5	4	3	2	1	0
0	0	0	0	BASE FUNCTION CODE			
				000 - RESERVED 001 - USER DATA SPACE 010 - USER PROGRAM SPACE 011 - RESERVED FOR USER 100 - RESERVED BY MOTOROLA 101 - SUPERVISOR DATA SPACE 110 - SUPERVISOR PROGRAM SPACE 111 - CPU SPACE			

Addresses

Channel 1: On,FFE,039 hex (n = Processor ID in hex)
Channel 2: On,FFE,079
Channel 3: On,FFE,0B9
Channel 4: On,FFE,0F9

Access Level: Supervisor
Initial State: Indeterminate

MEMORY TRANSFER COUNT REGISTER (MTCR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEMORY TRANSFER COUNT															

Addresses

Channel 1: On,FFE,00A hex (n = Processor ID in hex)
Channel 2: On,FFE,04A
Channel 3: On,FFE,08A
Channel 4: On,FFE,0CA

Access Level: Supervisor
Initial State: Indeterminate

BASE TRANSFER COUNT REGISTER (BTCR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE TRANSFER COUNT															

Addresses

Channel 1: On,FFE,01A hex (n = Processor ID in hex)
Channel 2: On,FFE,05A
Channel 3: On,FFE,09A
Channel 4: On,FFE,0DA

Access Level: Supervisor
Initial State: Indeterminate

DEVICE ADDRESS REGISTER (DAR)

31	BITS 30 THRU 1	0
DEVICE ADDRESS*		0

*Bits 31 through 24 in the register have no effect on the address output by the System I/O Module; they are hardwired to 80 hex always to access the local address space.

Addresses

Channel 1: On,FFE,014 hex (n = Processor ID in hex)
Channel 2: On,FFE,054
Channel 3: On,FFE,094
Channel 4: On,FFE,0D4

Access Level: Supervisor
Initial State: Indeterminate

MEMORY ADDRESS REGISTER (MAR)

31	BITS 30 THRU 1	0
	MEMORY ADDRESS*	0

*Bits 31 through 24 in the register have no effect on the address output by the System I/O Module; they are hardwired to 80 hex always to access the local address space.

Addresses

Channel 1: On,FFE,00C hex (n = Processor ID in hex)
Channel 2: On,FFE,04C
Channel 3: On,FFE,08C
Channel 4: On,FFE,0CC

Access Level: Supervisor
Initial State: Indeterminate

BASE ADDRESS REGISTER (BAR)

31	BITS 30 THRU 1	0
	BASE ADDRESS*	0

*Bits 31 through 24 in the register have no effect on the address output by the System I/O Module; they are hardwired to 80 hex always to access the local address space.

Addresses

Channel 1: On,FFE,01C hex (n = Processor ID in hex)
Channel 2: On,FFE,05C
Channel 3: On,FFE,09C
Channel 4: On,FFE,0DC

Access Level: Supervisor
Initial State: Indeterminate

6.6 INTERRUPTS TO THE PROCESSOR MODULE

Figure 6-5 illustrates the System I/O Module interrupt system. A summary of this structure is in Table 6-6.

TABLE 6-6. SYSTEM I/O MODULE INTERRUPT STRUCTURE

Device	Request	Acknowledge
68562 DUSCC #1 (Serial Ports A,B)	Local Exp. Bus W	Local Exp. Bus W
68562 DUSCC #2 (Serial Ports C,D)	Local Exp. Bus W	Daisy chain out of DUSCC #1
68901 MFP (Serial Port E)	Local Exp. Bus X	Local Exp. Bus X
63450 DMAC	Hardware selectable	Hardware selectable
Parallel Port	Hardware selectable	Hardware selectable

Both of the 68562 DUSCCs are hardwired to the Local Expansion Bus W request line. A request from additional devices can be wire-ORed to this line with hardware jumpers. The interrupt acknowledge for this request is returned on the Local Expansion Bus W acknowledge line. The acknowledge is first brought to DUSCC #1 (serial ports A, B). If no interrupts are pending, the device propagates the acknowledge to DUSCC #2 (serial ports C, D). If additional devices were connected to local request line W, the acknowledge would have to be daisy chained from DUSCC #2 to the devices. DUSCC #2 would propagate the acknowledge if it had no interrupts pending.

The 68901 MFP is hardwired to the Local Expansion Bus X request line. A request from additional devices can be wire-ORed to this line with hardware jumpers. The interrupt acknowledge for this request is returned on the Local Expansion Bus X acknowledge line. If additional devices were connected to local request line X, the acknowledge would have to be daisy chained from the MFP to the device. The MFP would propagate the acknowledge if it had no interrupts pending.

The paths for interrupt requests from the 63450 DMAC and the parallel port are selectable with hardware jumpers.

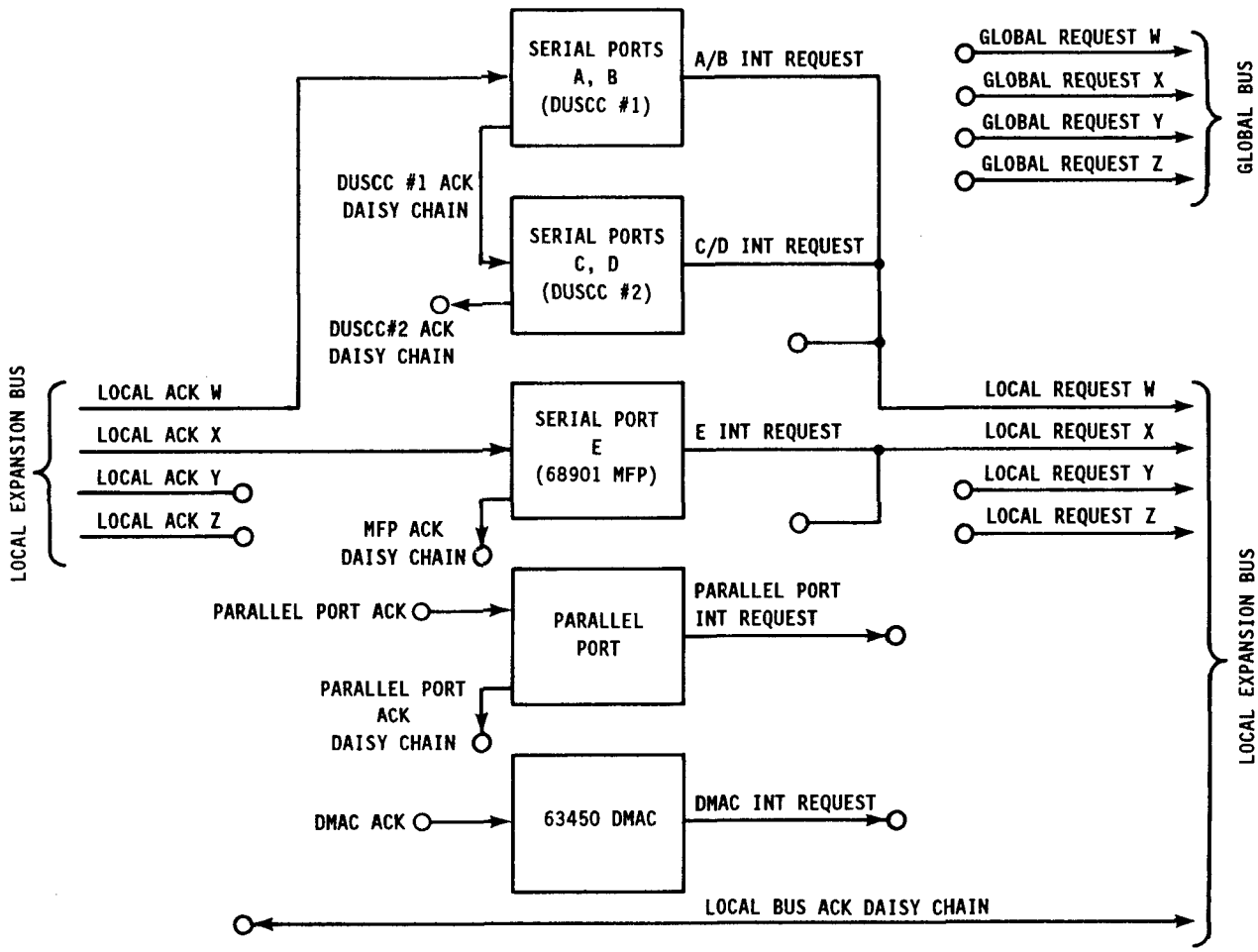


Figure 6-5. System I/O Module Interrupt System

If a device is wire-ORed on a request line with another device, it must be daisy chained appropriately on the acknowledge line. The 63450 DMAC cannot propagate an acknowledge signal; therefore, it must always be last in a daisy chain.

Global Bus W, X, Y, and Z request lines are each available for one device interrupt request. These requests must be auto-vector requests because no mechanism exists for returning an interrupt acknowledge to the requesting device.

The interrupt system also includes an interrupt acknowledge daisy chain line that interconnects all modules on the local expansion bus. This allows devices on other modules to be included in a particular interrupt acknowledge daisy chain. For example, the acknowledge signal from a propagating device can be connected to the daisy chain line on the System I/O Module; the daisy chain line is then connected to the acknowledge line into a device on another module.

7. MIL-STD-1553B MODULE

The MIL-STD-1553B Module attaches to a Processor Module local expansion bus and provides a dual-redundant MIL-STD-1553B bus interface. It can function as a bus controller (BC), remote terminal unit (RTU), or a bus monitor (MT).

7.1 ARCHITECTURAL OVERVIEW

The MIL-STD-1553B Module contains the following major components:

- 4K x 16-bit RAM
- BUS-66300 II MIL-STD-1553-to-Microprocessor Interface Unit
- BUS-65600 MIL-STD-1553 BC, RTU, and MT
- Dual Transceiver

The architecture of the module is shown in Figure 7-1.

The 4K x 16-bit RAM is accessible by the local MPU or global bus master and by the BUS-66300 II device. All 1553 transfers are accomplished through the RAM. To transmit data out on the 1553 bus, an MPU (local or global bus master) stores a specially formatted block of data in the 1553 module RAM and commands the BUS-66300 II to initiate the transfer. The 1553 interface components then read the RAM and transmit the data out on the 1553 bus. Data received on the 1553 bus are written to the RAM by the 1553 interface components. When the reception is complete, the 66300 device interrupts the local MPU; the MPU can then read the data from the RAM.

In addition to the 66300 device registers, the following registers have been implemented:

- Remote Terminal Address Register
- Tag Register/Counter
- Prescale/Count Enable Register
- Interrupt Vector Register

Although these registers are external to the 66300 device, they are included as part of its programming model.

The MIL-STD-1553B Module's local expansion bus interface and the major components of the module are described in the following paragraphs.

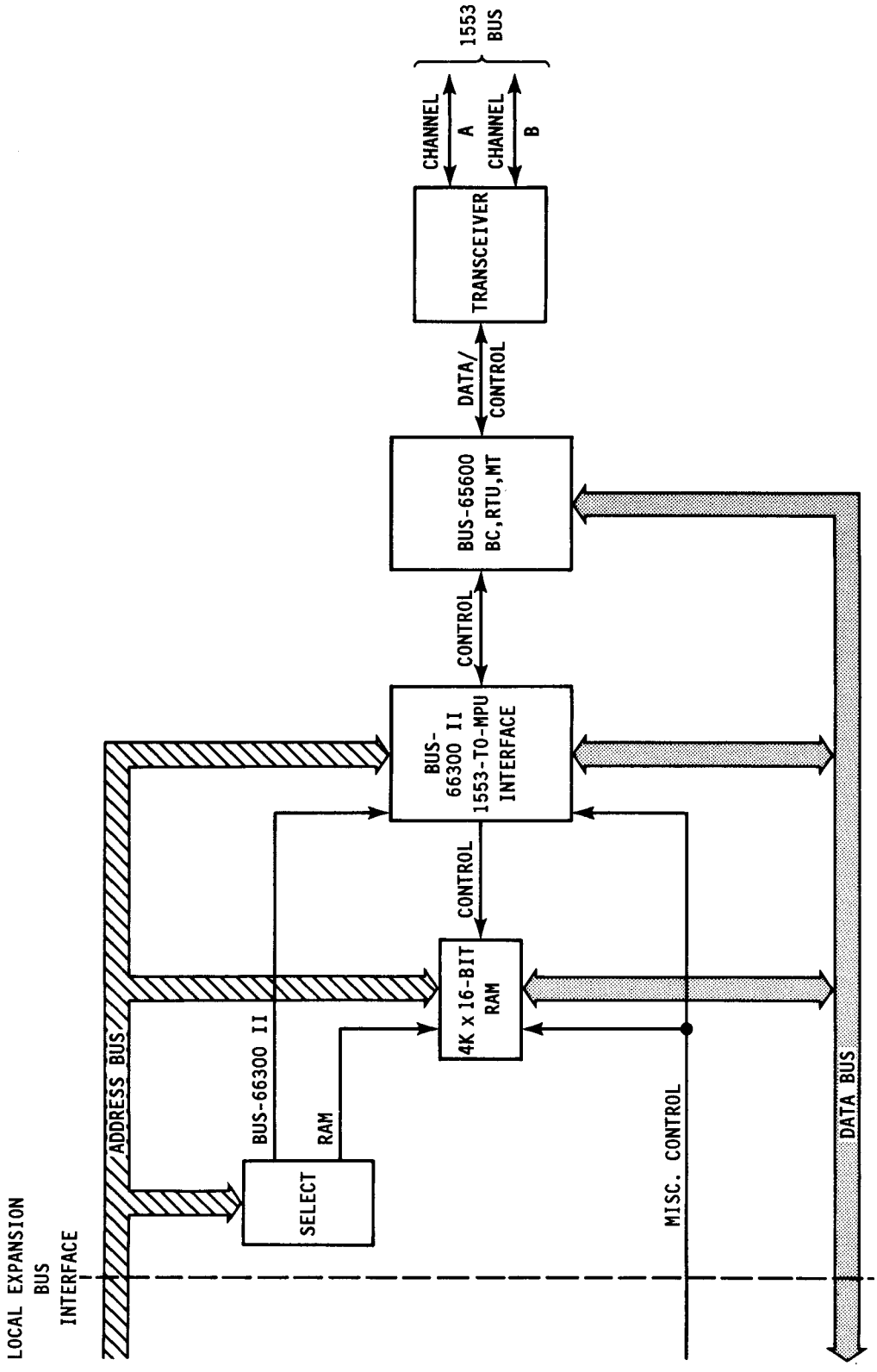


Figure 7-1. MIL-STD-1553B Module Architecture

7.1.1 Local Expansion Bus Interface

On the local expansion bus, the MIL-STD-1553B Module is implemented as a 32-bit data port for accesses of the 4K x 16-bit RAM and as a 16-bit data port for accesses of registers. The module is mapped into a 16K area within Processor Module address space. Table 7-1 summarizes the address assignments for MIL-STD-1553B Module resources.

TABLE 7-1. MIL-STD-1553B MODULE ADDRESSES

Assignment	Address*
2K x 32-bit RAM	0n,FF0,000-0n,FF1,FFF
BUS-66300 II Int. Mask Reg.	0n,FF2,000
BUS-66300 II Config. Reg.	0n,FF2,002
Not Used	0n,FF2,004
BUS-66300 II Start/Reset Reg.	0n,FF2,006
Interrupt Vector Reg.	0n,FF2,008
Remote Term. Addr. Reg.	0n,FF2,00A
Prescale/Count Enable Reg.	0n,FF2,00C
Tag Register/Counter	0n,FF2,00E

*Addresses are in hexadecimal; "n" is the Processor ID.

(The register addresses in the block from 0n,FF2,000 through 0n,FF2,00F are replicated throughout the space 0n,FF2,010 through 0n,FF3,FFF.)

7.1.2 4K x 16-Bit RAM

The 4K x 16-bit RAM resident on the MIL-STD-1553B Module is a buffer accessible by both the MPU and the 1553 interface components. All 1553 transfers are accomplished through this shared buffer. Data to be transmitted over the 1553 bus are written to the RAM by an MPU (local or global bus master). The 1553 interface components then read the RAM and transmit the data out on the 1553 bus. Data received on the 1553 bus are written to the RAM by the 1553 interface components. The local MPU then reads the RAM to get the received data.

The MPU views the RAM as a byte-addressable data port at addresses 0n,FF0,000 through 0n,FF1,FFF hex, where "n" is the Processor ID. The BUS-66300 II views the RAM as a word-addressable data port addressed from 0000 through 0FFF hex. Understanding the difference in viewpoints is crucial because the MPU must establish pointers for the BUS-66300 II at addresses in the RAM where the 66300 device expects to find them. Also, the pointers must be in the set of 66300 device addresses and must direct the device to the correct RAM locations. Additionally, the MPU must issue the

correct addresses to read data from the RAM that was written by the 66300 device using its set of addresses and pointers.

The RAM can be divided into two areas to support the double buffering scheme available with the BUS-66300 II. The two areas, designated A and B, are alternately used as "current" and "noncurrent" areas. Under MPU control, the 66300 device is directed to the current area. While the device is processing 1553 messages using the current area, the MPU can be can be working with the noncurrent area.

7.1.3 BUS-66300 II

The BUS-66300 II MIL-STD-1553-to-Microprocessor Interface Unit functions as the intermediary between the 1553 module's shared RAM and the BUS-65600 MIL-STD-1553 BC, RTU, and MT. The operating features of 66300 device are established by the MPU via three internal registers, which are memory-mapped in Processor Module address space. The address map for the 66300 is given in Table 7-2.

TABLE 7-2. BUS-66300 II ADDRESS MAP

Register	Acronym	Offset*
Interrupt Mask	IMR	00
Configuration	CR	02
Start/Reset	S/RR	06
Interrupt Vector	IVR	08
Remote Terminal Address	RTAR	0A
Prescale/Count Enable	P/CER	0C
Tag/Count	T/CR	0E

*Offset is in hex from base address 0n,FF2,000.

The programming model for the 66300, including the four external support registers, is shown in Figure 7-2. Descriptions of the registers follow the programming model diagram. Complete information on the BUS-66300 II is in the ILC Data Device Corportion publication, *BUS-66300 II MIL-STD-1553 to Microprocessor Interface Unit.*

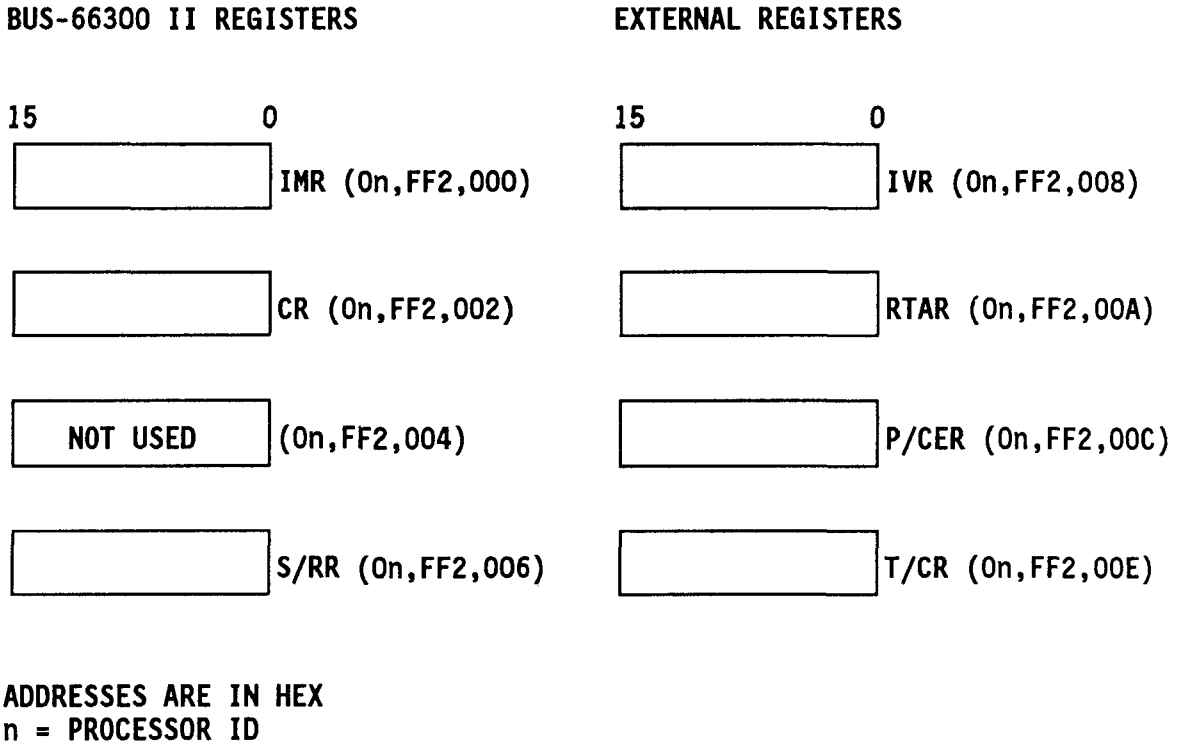


Figure 7-2. BUS-66300 II Programming Model

7.1.3.1 Configuration Register. The Configuration Register (CR) contains four bits that establish the 1553 operating mode (BC, RTU, or MT) and two additional operating parameters. The register controls 4 bits in the 1553 status word in RT mode only. To assert these bits on the 1553 bus, the corresponding register bits must contain zeros. The format of the CR is shown below:

Configuration Register (CR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE		$\overline{A}$/B	SOE	$\overline{DBA}$	$\overline{BSY}$	$\overline{SRQ}$	$\overline{SSF}$	Not Used (= 1)							

Address: On,FF2,002 hex (n = Processor ID)
Access Level: Supervisor
Initial State: 00FF hex

MODE 1553 Operating Mode - These two bits establish the 1553 operating mode as shown below:

Bit 15	Bit 14	Operating Mode
0	0	Bus Controller (BC)
0	1	Bus Monitor (MT)
1	0	Remote Terminal Unit (RTU)
1	1	Illegal

$\overline{A}$/B RAM Buffer A or B - The 1553 interface can be implemented with the common RAM divided into two buffer areas, current and noncurrent. This is to prevent partially updated message blocks from being read by the MPU or transferred onto the 1553 bus. The designations A and B are given to these two areas, and either may be the current area. The area that the 66300 accesses is determined by the state of the A/B bit: 0 - Area A; 1 - Area B.

SOE Stop On Error - When this bit is set, the 66300 will stop at the end of the current data block transfer if an error has been detected. When the bit is cleared, the 66300 will continue if there are more messages to send after the current block is transferred.

$\overline{\text{DBA}}$ Dynamic Bus Control Acceptance - This bit corresponds to the bit in the 1553 status word that occurs at bit time 18. The function of this bit in MIL-STD-1553B (page 13) is as follows:

> This bit shall be used if the RT implements the optional dynamic bus control function. This bit, when used, shall indicate the acceptance or rejection of a dynamic bus control offer. . . . A logic one shall indicate acceptance of control, and a logic zero shall indicate rejection of control. If this function is not used, this bit [in the 1553 status word] shall be set to logic zero.

A zero in bit position 11 of the CR asserts this bit in the bus status word. A one negates it.

$\overline{\text{BSY}}$ Busy - This bit corresponds to the bit in the 1553 status word that occurs at bit time 16. The function of this bit in MIL-STD-1553B (page 13) is as follows:

> This bit, when used, shall indicate that the RT or subsystem is unable to move data to or from the subsystem in compliance with the bus controller's command. A logic one shall indicate the presence of a busy condition, and a logic zero its absence. In the event the busy bit is set in response to a transmit command, then the RT shall transmit its status word only. If this function is not implemented, the bit [in the 1553 status word] shall be set to logic zero.

A zero in bit position 10 of the CR asserts this bit in the bus status word. A one negates it.

$\overline{\text{SRQ}}$ Service Request - This bit corresponds to the bit in the 1553 status word that occurs at bit time 11. The function of this bit in MIL-STD-1553B (page 12) is as follows:

> This bit, when used, shall indicate the need for the bus controller to take specific predefined actions relative to either the RT or associated subsystem. Multiple subsystems, interfaced to a single RT, which individually require a service request signal shall logically OR their individual signals into the single status word bit. In the event this logical OR is performed, then the designer must make provisions in a separate data word to identify the specific requesting subsystem.

> The service request bit is intended to be used only to trigger data transfer operations which take place on an exception rather than a periodic basis. A logic one shall indicate the presence of a service request, and a logic zero its absence. If this function is not implemented, the bit [in the 1553 status word] shall be set to zero.

A zero in bit position 9 of the CR asserts this bit in the bus status word. A one negates it.

$\overline{\text{SSF}}$ Subsystem Flag - This bit corresponds to the bit in the 1553 status word that occurs at bit time 17. The function of this bit in MIL-STD-1553B (page 13) is as follows:

> This bit, when used, shall flag a subsystem fault condition, and alert the bus controller to potentially invalid data. Multiple subsystems, interfaced to a single RT, which individually require a subsystem flag bit signal shall logically OR their individual signals into the single status word bit. In the event this logical OR is performed, then the designer must make provisions in a separate data word to identify the specific reporting subsystem. A logic one shall indicate the presence of the flag, and a logic zero its absence. If not used, this bit [in the 1553 status word] shall be set to logic zero.

A zero in bit position 8 of the CR asserts this bit in the bus status word. A one negates it.

7.1.3.2 Start/Reset Register. The Start/Reset Register (S/RR) is a write-only register that contains two bits for initiating data transfers and for resetting all the registers in the 66300. The format of the S/RR is shown below:

Start/Reset Register (S/RR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not Used														CS	RST

Address: On,FF2,006 hex (n = Processor ID)
Access Level: Supervisor (Write only)
Initial State: Write Only

CS Controller Start - In BC mode, setting this bit causes the 66300 to start message transmission. In MT mode, setting this bit enables monitor operation.

RST Reset - Setting this bit clears all 66300 registers.

7.1.3.3 Interrupt Mask Register. The Interrupt Mask Register (IMR) contains three bits used for enabling or disabling interrupt requests on certain data transfer conditions. The format of the IMR is shown below:

Interrupt Mask Register (IMR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not Used (= 1)								Not Used				BEM	F/S	*	EOM

*Not Used

Address: On,FF2,000 hex (n = Processor ID)
Access Level: Supervisor
Initial State: FF00 hex

BEM Bus Controller End of Message - When this bit is set, the 66300 generates an interrupt request after all BC messages have been transferred. When cleared, no interrupt is generated for this condition.

F/S Format Error/Status Set - When this bit is set, the 66300 generates an interrupt request under the following conditions:

Loop Test Failure: Received loopback word did not match the transmitted word.

Message Error: Received message contained an address error, a status word with one of the eight 1553 status bits set, or a 1553 specification violation such as a parity error or a Manchester encoding error.

Time Out: Transmission was not received within the allotted time.

Status Set: Received status word contained status bits set or an address error.

When the bit is cleared, no interrupt is generated for these conditions.

EOM End of Message - When this bit is set, the 66300 generates an interrupt request after every message transfer. When cleared, no interrupt is generated for this condition.

7.1.3.4 Remote Terminal Address Register. The Remote Terminal Address Register (RTAR) is a read-only register that reflects the RTU address present on connector J607. The format of the RTAR is shown below:

Remote Terminal Address Register (RTAR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not Used = 1								0	0	P	RTU Address				

Address: On,FF2,00A hex (n = Processor ID)
Access Level: Supervisor
Initial State: Determined by address on J607

RTU Address - 5-bit RTU address.

P Parity - Odd parity for the 5 bits in positions 4 through 0.

7.1.3.5 Prescale/Counter Enable Register. The Prescaler/Counter enable Register (P/CER) contains eight bits of prescale count value and one bit that enables the Tag Register counter. The function of the P/CER is directly related to the Tag Register, described in the following paragraph. Writing any value to the P/CER clears the Tag Register.

The format of the P/CER is shown below:

Prescale/Count Enable Register (P/CER)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not Used							TCE	Prescale Count							

Address: On,FF2,00C hex (n = Processor ID)
Access Level: Supervisor
Initial State: Indeterminate (all except TCE are write only)

TCE Tag Register Count Enable - When this bit is set, the tag counter is enabled, and the value in the Tag Register is incremented by one each time the prescale counter rolls over. When this bit is cleared, the tag counter is disabled. This bit is both readable and writable.

Prescale Count - Bits 7 through 0 are write-only and form a prescale value for the tag counter. The prescaler divides a 2 MHz input clock by the divisor value established by these bits. The required divisor is loaded into the register in 2s complement form. A divisor of 1 (counter value of FF in 2s complement) is illegal; a value of 00 gives a divisor of 256.

7.1.3.6 **Tag/Count Register.** This 16-bit register stores a fixed value or accumulates a count of the prescaler output as described in paragraph 7.1.3.5. The T/CR is cleared by writing to the P/CER. The value in the T/CR is written to the Descriptor Block in RAM at the beginning and the end of every message transfer. The format of the T/CR is shown below:

Tag/Count Register (T/CR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Tag Value															

Address: On,FF2,00E hex (n = Processor ID)
Access Level: Supervisor
Initial State: 0000 hex

7.1.3.7 **Interrupt Vector Register.** The Interrupt Vector Register (IVR) stores the 8-bit interrupt vector returned to the MPU during an interrupt acknowledge cycle. The desired vector is written to the register by the user. The register is also readable. The format of the IVR is shown below:

Interrupt Vector Register (IVR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not Used = 1								Interrupt Vector							

Address: On,FF2,008 hex (n = Processor ID)
Access Level: Supervisor
Initial State: Indetermmate

7.1.4 BUS-65600

The BUS-65600 BC, RTU, and MT interfaces the BUS-66300 II and the 1553 bus transceivers. The 65600 device is controlled exclusively by handshaking from the 66300 device and has no user-accessible locations. Complete information on the BUS-65600 is in the ILC Data Device Corportion publication, *BUS-65600 MIL-STD-1553 Bus Controller, Remote Terminal, and Bus Monitor.*

7.2 OPERATION AS A BUS CONTROLLER

For the MIL-STD-1553B Module to function as a bus controller (BC), the following steps are required:

1. The 4K x 16-bit RAM is partitioned for BC operation.

2. RAM is initialized with the required data values.

3. The BUS-66300 II is reset and programmed for BC mode. Other desired 66300 device operational features, e.g., interrupt conditions, are also established.

4. A Controller Start command is issued to the BUS-66300 II.

Once the 66300 device has been started, it works autonomously with the RAM and the BUS-65600 to complete the required 1553 transfer. Detailed descriptions of the steps listed above are given in the following paragraphs.

7.2.1 BC Mode RAM Partitioning

The 4K x 16-bit RAM is partioned for BC operation as shown in Table 7-3. To support double buffering, locations are specified for both A area and B area stack pointers, message counters, internal pointers, descriptor stacks, and message. An overview of how these components are used for BC mode memory management is shown in Figure 7-3.

TABLE 7-3. 4K X 16-BIT RAM PARTITIONING FOR BC MODE

Processor Module Address*	BUS-66300 II Address*	Function
0n,FF0,000-0n,FF0,1FF	0000-00FF	User Defined**
0n,FF0,200-0n,FF0,201	0100	Stack Pointer A
0n,FF0,202-0n,FF0,203	0101	Message Counter A
0n,FF0,204-0n,FF0,207	0102-0103	Internal Pointer A
0n,FF0,208-0n,FF0,209	0104	Stack Pointer B
0n,FF0,20A-0n,FF0,20B	0105	Message Counter B
0n,FF0,20C-0n,FF0,20F	0106-0107	Internal Pointer B
0n,FF0,210-0n,FF1,FFF	0108-0FFF	User Defined**

*Addresses are in hex; "n" is the Processor ID.

**User-defined areas include Descriptor Stacks A and B and message blocks.

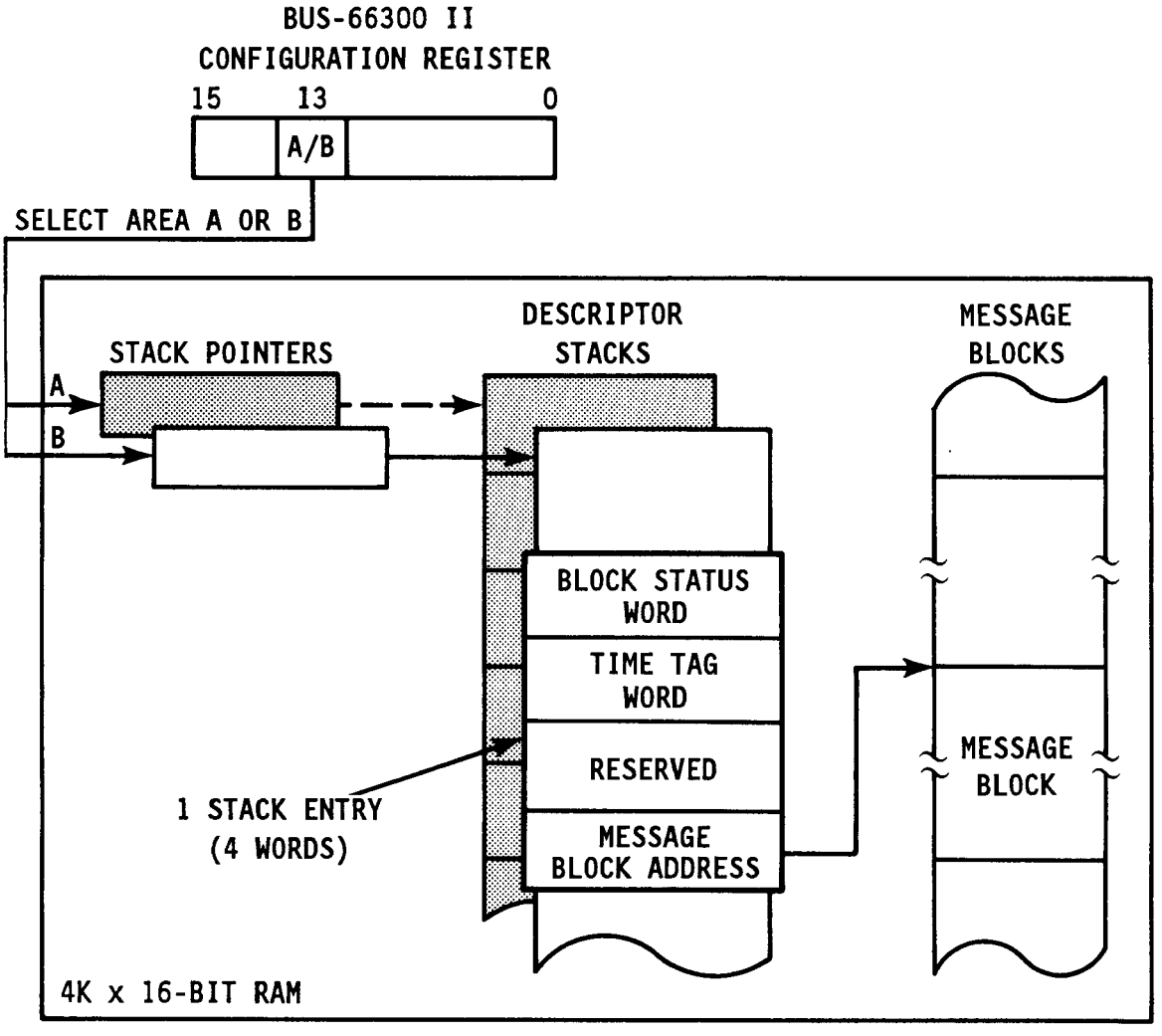


Figure 7-3. BC Mode Memory Management

7.2.2 BC Mode RAM Initialization

The following steps are required to initialize the 4K x 16-bit RAM for BC mode operation:

1. Initialize stack pointers.
2. Load message counter.
3. Initialize descriptor stack with message block addresses.
4. Load message blocks.

Descriptions of the components to be loaded into RAM are given in the following paragraphs.

7.2.2.1 Stack Pointers A, B. The stack pointer locations each contain the 16-bit address of a descriptor stack. Stack Pointer A has the address of Descriptor Stack A; Stack Pointer B has the address of Descriptor Stack B. A Descriptor Stack contains 64 four-word entries and automatically wraps around (last entry is followed by the first). The current area stack pointer determines the Descriptor Stack entry that applies to the current message. The BUS-66300 II automatically increments the current area stack pointer by four upon completion of each message even if an error is detected.

7.2.2.2 Message Counters A, B. The message counter locations each contain the 16-bit count specifying the number of messages that will be included in the 1553 transfer. The count (number of messages) must be specified in 1's complement form.

7.2.2.3 Descriptor Stacks A, B. The descriptor stack contains one 4-word entry for each message in the 1553 transfer. The format of a stack entry is shown below:

WORD 1	BLOCK STATUS
WORD 2	TIME TAG
WORD 3	RESERVED
WORD 4	MESSAGE BLOCK ADDRESS

7.2.2.3.1 Block Status. The Block Status word is initialized by the BUS-66300 II when it starts a message transfer and is updated at the conclusion of a transfer. The format of the Block Status word is shown below:

BLOCK STATUS WORD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOM	SOM	$\overline{A}$/B	ERR	SS	FE	RTO	LTF	Not Used							

EOM End of Message - Set to indicate a message transfer is complete.

SOM Start of Message - Set to indicate that a message transfer has been started and is in progress. When the transfer is complete, this bit is cleared and EOM is set.

$\overline{A}$/B In BC mode, this bit is always 0. In RTU mode, this bit is 0 if 1553 channel A is used; it is 1 if channel B is used.

ERR Error - Set to indicate that an error was detected during a message transfer.

SS Status Set - Set to indicate that the status word received for the 1553 transfer contained one or more status bits set or an address error (BC mode only).

FE Format Error - Set to indicate that a transfer violated a 1553 specification (parity error, Manchester encoding error, etc.).

RTO Response Timeout - Set to indicate the expected transmission was not received in the allotted time (BC mode only).

LTF Loop Test Fail - Set to indicate that the word received on loop back did not match the transmitted word.

7.2.2.3.2 Time Tag. The Time Tag word is a 16-bit location that stores the value provided by the Tag/Count Register (T/CR). This location is initialized with the value from the T/CR at the start of a message transfer and updated at the end of the transfer.

7.2.2.3.3 Message Block Address. The Message Block Address is the 16-bit word address of the message for the 1553 transfer.

7.2.2.4 Message Blocks. The message blocks in the 4K x 16-bit RAM consist of a group of consecutive words that are loaded with the control, command, data, and status words associated with a 1553 message. The formats of the different 1553 message types used in the BC mode of operation are shown in Figure 7-4.

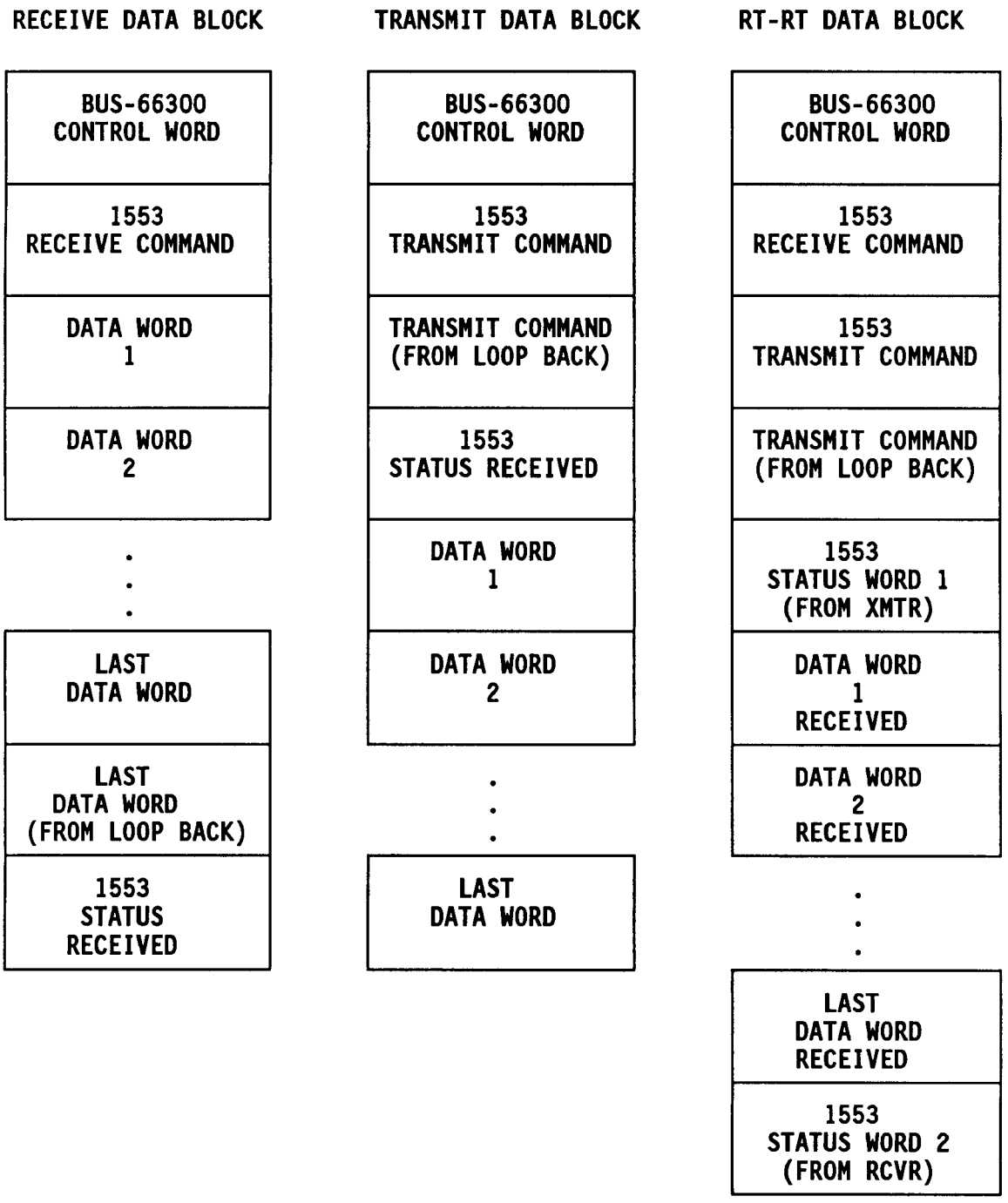


Figure 7-4 (1 of 3). BC Mode Message Block Formats

MODE CODE WITHOUT DATA	MODE CODE WITH DATA RECEIVE DATA	MODE CODE WITH DATA TRANSMIT DATA
BUS-66300 CONTROL WORD	BUS-66300 CONTROL WORD	BUS-66300 CONTROL WORD
1553 MODE COMMAND	1553 MODE COMMAND	1553 MODE COMMAND
MODE COMMAND (FROM LOOP BACK)	DATA WORD	MODE COMMAND (FROM LOOP BACK)
1553 STATUS WORD	DATA WORD (FROM LOOP BACK)	1553 STATUS WORD
	1553 STATUS WORD	DATA WORD RECEIVED

Figure 7-4 (2 of 3). BC Mode Message Block Formats

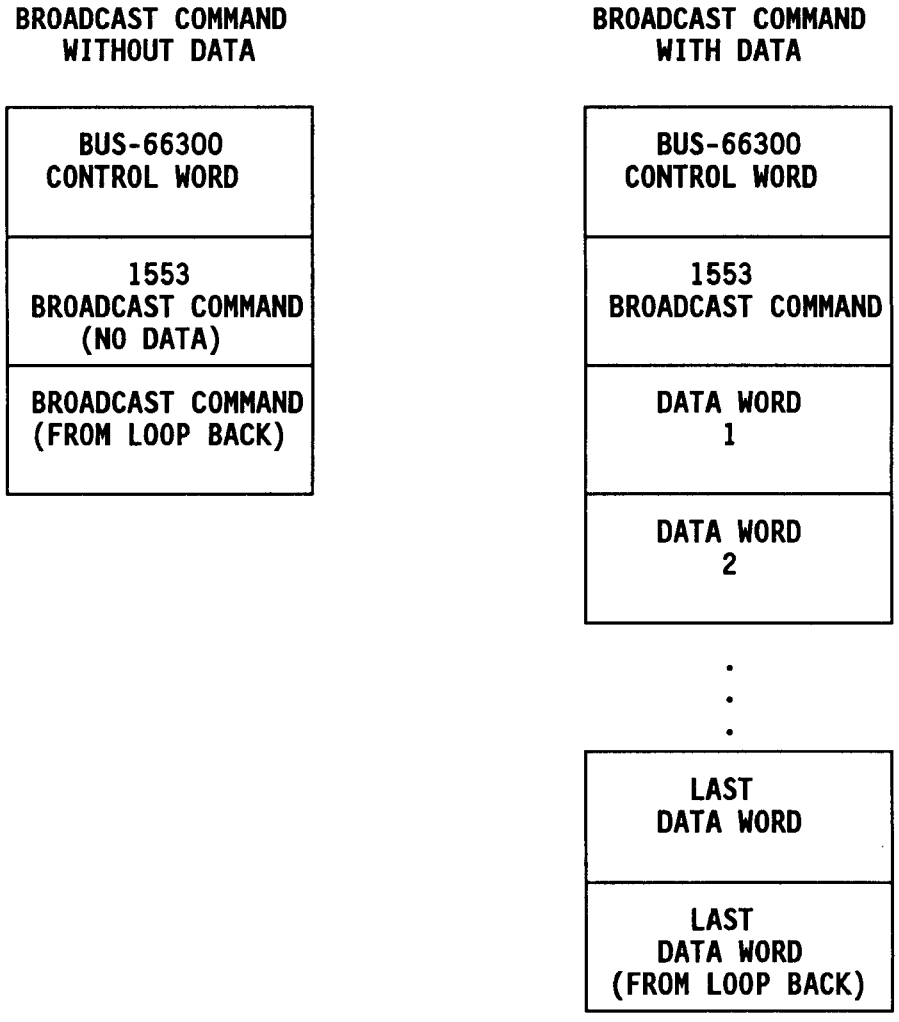


Figure 7-4 (3 of 3). BC Mode Message Block Formats

For a description of the 1553 command, data, and status words included in the message blocks, refer to MIL-STD-1553B. The format of the BUS-66300 II control word included in the message blocks is described below.

BUS-66300 II CONTROL WORD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								A/$\overline{B}$	*	MBB	*		MC	B	R-R

*Not Used

A/$\overline{B}$ Bus Channel A/B - Setting this bit causes transmission over 1553 bus A. Clearing the bit causes transmission over 1553 bus B.

MBB Mask Broadcast Bit - Setting this bit prevents the Broadcast RCVD bit of the status word response from signaling a status error during a Broadcast command.

MC Mode Code - When this bit is set it signifies that the 1553 command word count field signifies a mode code type. When cleared, the command word field contains a count.

B Broadcast - When this bit is set it indicates a Broadcast command (multiple RTUs addressed, no status word expected).

R-R RTU-to-RTU - When this bit is set, the BC initiates an RTU-to-RTU transfer.

7.2.3 BC Mode BUS-66300 II Initialization

After RAM has been initialized for BC mode, the BUS-66300 II is initialized as follows:

1. A Reset command is issued to the 66300 device.
2. The BC mode of operation is set in the 66300's Configuration Register.
3. The required interrupt masks are set via the Interrupt Mask Register.

Once the device is initialized, a Controller Start command is issued to start the 1553 transfer. The BUS-66300 II then works autonomously with the BUS-65600 and the

RAM to complete the transfer. One or more of the interrupts described in 7.1.3.3 may be used to notify the MPU of completion or errors.

7.3 OPERATION AS A REMOTE TERMINAL UNIT

For the MIL-STD-1553B Module to function as a remote terminal unit (RTU), the following steps are required:

1. The 4K x 16-bit RAM is partitioned for RTU operation.

2. RAM is initialized with the required data values.

3. The BUS-66300 II is reset and programmed for RTU mode. Other desired 66300 device operational features, e.g., interrupt conditions, are also established.

4. The 66300 device waits for a 1553 command.

Once the BUS-66300 II receives a 1553 command, it works autonomously with the RAM and the BUS-65600 to complete the required 1553 transfer. Detailed descriptions of the steps listed above are given in the following paragraphs.

7.3.1 RTU Mode RAM Partitioning

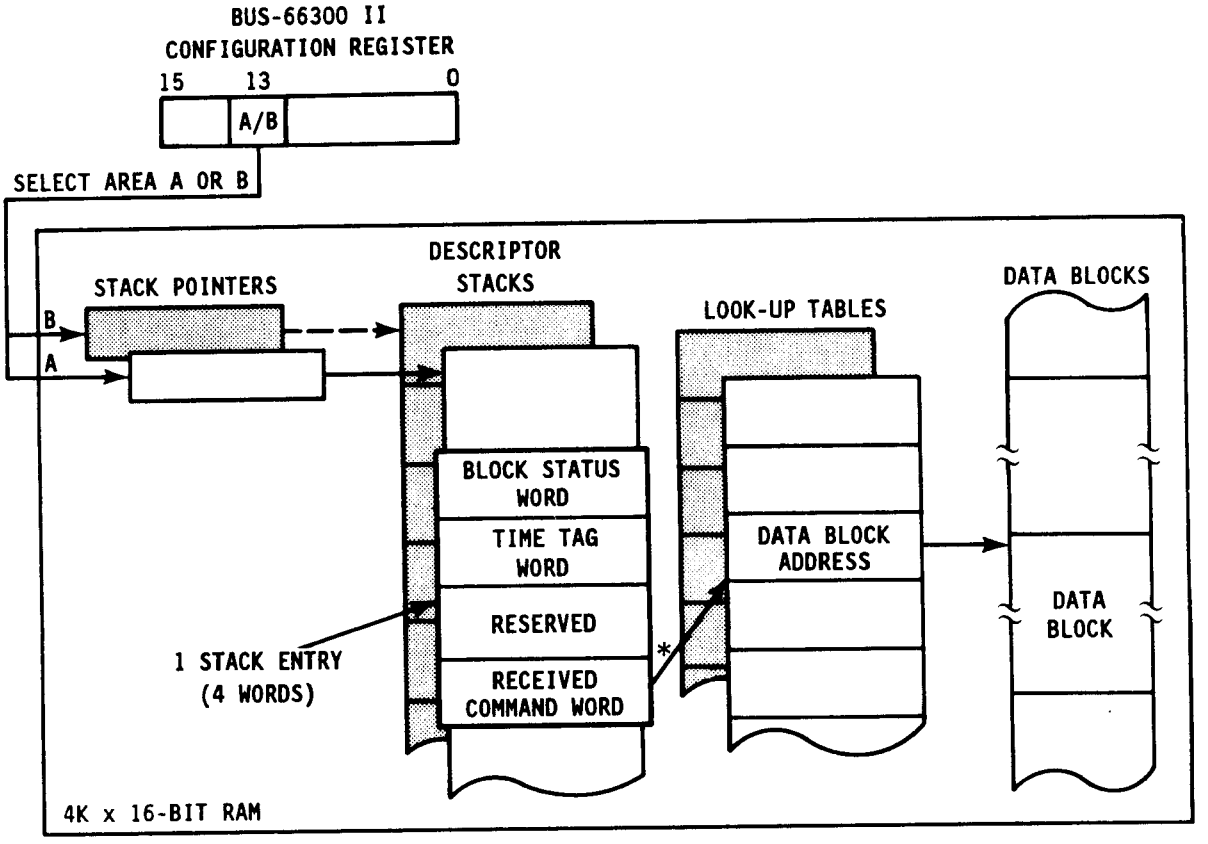
The 4K x 16-bit RAM is partitioned for RTU operation as shown in Table 7-4. To support double buffering, locations are specified for both A area and B area stack pointers, internal pointers, look-up tables, descriptor stacks, and data blocks. An overview of how these components are used for RTU mode memory management is shown in Figure 7-5.

TABLE 7-4. 4K X 16-BIT RAM PARTITIONING FOR RTU MODE

Processor Module Address*	BUS-66300 II Address*	Function
0n,FF0,000-0n,FF0,1FF	0000-00FF	User Defined**
0n,FF0,200-0n,FF0,201	0100	Stack Pointer A
0n,FF0,202-0n,FF0,203	0101	Reserved
0n,FF0,204-0n,FF0,207	0102-0103	Internal Pointer A
0n,FF0,208-0n,FF0,209	0104	Stack Pointer B
0n,FF0,20A-0n,FF0,20B	0105	Reserved
0n,FF0,20C-0n,FF0,20F	0106-0107	Internal Pointer B
0n,FF0,210-0n,FF0,27F	0108-013F	User Defined**
0n,FF0,280-0n,FF0,2FF	0140-017F	Look-up Table A
0n,FF0,300-0n,FF0,37F	0180-01BF	User Defined**
0n,FF0,380-0n,FF0,3FF	01C0-01FF	Look-up Table B
0n,FF0,400-0n,FF1,FFF	0200-0FFF	User Defined**

*Addresses are in hex; "n" is the Processor ID.

**User-defined areas include Descriptor Stacks A and B and data blocks.



*POINTER IS FORMED WITH T/$\bar{R}$ BIT AND SUBADDRESS FIELD OF COMMAND WORD AND BIT 13 OF CR.

Figure 7-5. RTU Mode Memory Management

7.3.2 RTU Mode RAM Initialization

The following steps are required to initialize the 4K x 16-bit RAM for RTU mode operation:

1. Initialize stack pointers.
2. Load look-up table with data block addresses.
3. Load data into transmit data blocks

The stack pointers function the same as those used in BC mode; refer to 7.2.2.1 for a description. The look-up tables and descriptor stacks are described in the following paragraphs.

7.3.2.1 Look-up Tables. The first 32 words of the look-up table contain the addresses of the data blocks to be used when receiving data from 1553 subaddresses 0 through 31, respectively. The next 32 words of the table contain the addresses of the data blocks to be used when the RTU is directed to transmit data from 1553 subaddresses 0 through 31, respectively.

7.3.2.2 Descriptor Stacks A, B. The descriptor stack contains one four-word entry for each message in the 1553 transfer. The format of a stack entry is shown below:

WORD 1	BLOCK STATUS
WORD 2	TIME TAG
WORD 3	RESERVED
WORD 4	RECEIVED COMMAND WORD

The Block Status and Time Tag words are the same as those used for BC mode; refer to 7.2.2.3.1 and 7.2.2.3.2 for a description. The Received Command word is described below.

7.3.2.2.1 Received Command Word. The format of the Received Command word in the descriptor stack is shown below:

RECEIVED COMMAND WORD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTU Address					T/R	RTU Subaddress					Word Cnt/Mode Code				

This word is an exact copy of the 1553 command that was sent to the RTU. For a description of the fields refer to MIL-STD-1553B.

The BUS-66300 II uses the T/R and the RTU Subaddress fields of the command word and the $\bar{A}$/B bit in the Configuration Register to formulate a look-up table address. This step is illustrated in Figure 7-5.

The format of the address is shown below:

LOOKUP TABLE ADDRESS CALCULATION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	$\bar{A}$/B	1	T/R	RTU Subaddress				

7.3.2.3 Data Blocks. Data blocks are reserved in RAM for storing data received from 1553 subaddresses 0 through 31 and for holding data to be transmitted to 1553 subaddressses 0 through 31. A data block may be any length sufficient to contain the particular message as long as it does not cross a 256-word boundary. Data blocks may be shared by the Look-up Tables A and B.

7.3.3 RTU Mode BUS-66300 II Initialization

After RAM has been initialized for RTU mode, the BUS-66300 II is initialized as follows:

1. A Reset command is issued to the 66300 device.

2. The RTU mode of operation is set in the 66300's Configuration Register.

3. The required interrupt masks are set via the Interrupt Mask Register.

Once the device is initialized, it waits for a 1553 command. When a 1553 command is received, the BUS-66300 II works autonomously with the BUS-65600 and the RAM to complete the transfer. One or more of the interrupts discribed in 7.1.3.3 may be used to notify the MPU of completion or errors.

7.3.4 Illegal Mode Codes

When operating as an RTU, a 1553 command containing a "mode code" may be received. Refer to *MIL-STD-1553B* for a description of mode codes. The MIL-STD-1553B Module contains hardware that will allow selected mode codes to be deemed illegal. The

circuit consists of an erasable programmable logic device (EPLD) that has for its inputs the word count, mode code, transmit/receive, and broadcast outputs of the BUS-65600 device. The EPLD is programmed to recognize the selected illegal mode codes. When an illegal mode code is received the message error bit is set in the returned 1553 bus status word.

With the standard hardware, all mode codes assigned by *MIL-STD-1553B* are allowed; all others are illegal. The EPLD may be reprogrammed to illegalize additional mode codes.

7.4 OPERATION AS A BUS MONITOR

For the MIL-STD-1553B Module to function as a bus monitor (MT), the following steps are required:

1. The 4K x 16-bit RAM is partitioned for MT operation.

2. RAM is cleared and the stack pointers are loaded.

3. The BUS-66300 II is reset and programmed for MT mode. Other desired 66300 device operational features, e.g., interrupt conditions, are also established.

4. A Controller Start command is issued to the 66300 device.

Once the BUS-66300 II has been started, it works autonomously with the RAM and the BUS-65600. Detailed descriptions of the steps listed above are given in the following paragraphs.

7.4.1 MT Mode RAM Partitioning

The 4K x 16-bit RAM is partitioned for MT operation as shown in Table 7-5. To support double buffering, locations are specified for both A area and B area stack pointers.

7.4.2 MT Mode RAM Initialization

The following steps initialize the 4K x 16-bit RAM for MT mode operation:

1. Clear entire RAM.
2. Initialize stack pointers.

TABLE 7-5. 4K X 16-BIT RAM PARTITIONING FOR MT MODE

Processor Module Address*	BUS-66300 II Address*	Function
0n,FF0,000-0n,FF0,001	0000	1553 Word
0n,FF0,002-0n,FF0,003	0001	Identification (ID) Word for preceding 1553 word
0n,FF0,004-0n,FF0,1FF	0002-0FFF	1553 word/ID word pairs
0n,FF0,200-0n,FF0,201	0100	Stack Pointer A**
0n,FF0,208-0n,FF0,209	0104	Stack Pointer B**

*Addresses are in hex; "n" is the Processor ID.
**These areas are overwritten once data collection begins.

In MT mode of operation, the entire RAM area is used as a stack to store valid 1553 words and corresponding identification words. The stack pointers contain the address in RAM where the first received word will be stored. The value is loaded into an internal register in the BUS-66300 II. Once MT operation begins, the stack pointer location in RAM will be overwritten with data.

The format of the identification word stored with each 1553 word is shown below:

MT IDENTIFICATION WORD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Gap Time								1	1	1	ERR	CS	$\overline{A}$/B	CD	0

Gap Time - This field contains eight bits that are used as a counter. The field is reset for each new transmission and incremented every 0.5 μs. If the field reaches a count of FF hex (128 μs), it remains static.

Bits 7,6,5 Word Transfer - These bits are set to 1. They may be used to indicate that a word was successfully transferred from the 1553 bus. This bit may be cleared in memory by the MPU at the time it reads the corresponding data word; this provides a method of keeping track of words that have been processed.

ERR Error - When this bit is cleared it indicates that the associated 1553 word was valid. When set, it indicates the word had a Manchester error, parity error, low bit count, or status/sync field.

CS Command Sync - When this bit is set, it indicates a command or status sync field. When cleared, it indicates a data sync field.

$\overline{A}$/B Channel $\overline{A}$/B - When this bit is set, it indicates that the transmission was received on channel B. When cleared, it indicates channel A.

CD Contiguous Data - This bit is set if the current word was received at least 2 μs after receipt of the previous word (mid-parity to mid-sync).

7.4.3 MT Mode BUS-66300 II Initialization

After RAM has been initialized for MT mode, the BUS-66300 II is initialized as follows:

1. A Reset command is issued to the BUS-66300.
2. The MT mode of operation is set in the 66300's Configuration Register.
3. Initialize the stack pointer.

Once the device is initialized, a Controller Start command is issued to start monitor operation. The BUS-66300 II then works autonomously with the BUS-65600 and the RAM logging 1553 transfers into memory.

7.5 INTERRUPTS TO THE PROCESSOR MODULE

The MIL-STD-1553B Module can be strapped to use local expansion bus interrupt request lines W, X, Y, or Z. The module contains an interrupt vector register (7.1.3.7) and can supply a user-programmed vector during an interrupt acknowledge cycle.

NOTES

8. BUS ERRORS

SANDAC V bus errors result from the following conditions:

- Writing to the 512-byte FIFO when full
- Accessing a monitored local memory address
- Accessing a monitored global memory address
- Executing a Breakpoint (BKPT) instruction
- Timing out of the watchdog timer
- Writing to the Utility Module EPROM
- Writing to the Utility Module Status Register
- Writing to the Utility Module 32-bit timer at the user level

These conditions are described in the following paragraphs. Table 8-1 summarizes SANDAC V bus errors.

8.1 FIFO WRITE WHEN FULL

Writing to the 512-byte FIFO when it is full causes a bus error to be signaled and the bus error exception to be taken. The FBE bit (bit 3) of the local Command/Status Register (Subsection 3.7) is set if the local processor executed the write. The routine that handles the exception should include reading of this bit. If the bit is set, it should be cleared by writing a 0 to the corresponding bit in the command portion of the Command/Status Register. A write to a full FIFO by a global master causes a bus error to be signaled to the global master but does not set the FBE bit of the local Command/Status Register.

8.2 MONITORED LOCAL MEMORY ADDRESS

When a local memory access matches the conditions (address block, access level, and access type) programmed into the Processor Module address monitor, a bus error is signaled and the bus error exception is taken. The ABE bit (bit 4) of the local Command/Status Register (Subsection 3.7) is set if the access was by the local processor.

TABLE 8-1. SANDAC V BUS ERROR SUMMARY

Bus Error Source	Condition	Indicator	Exception Taken
FIFO	Write to FIFO when full	Command/Status Register bit 3 (FBE) set	Bus Error
Processor Module Address Monitor	Access of monitored block	Command/Status Register bit 4 (ABE) set	Bus Error
Global Mem. Module Access Monitor	Access of monitored block	None	Bus Error
BKPT Instruction	Execution of 68020 BKPT (Breakpoint) instruction	None	Illegal Instruction
Watchdog Timer	No response within 512 clock cycles	Command/Status Register bit 5 (TBE) set	Bus Error
	No response to interrupt ack.	Command/Status Register bit 5 (TBE) set	Spurious Interrupt
Utility Module EPROM	Write access	None	Bus Error
Utility Module Status Register	Write access	None	Bus Error
Utility Module 32-Bit Timer	Write access at user level	None	Bus Error

The exception handler should include reading of this bit. If the bit is set, it should be cleared by writing a 0 to the corresponding bit in the command portion of the Command/Status Register. An access by the global master does not set the ABE bit of the local Command/Status Register.

8.3 MONITORED GLOBAL MEMORY BLOCK

When a global memory access writes into a block monitored by the Global Memory Module access monitor, a bus error is signaled and the bus error exception is taken. In this case, there is no direct indication that this is the cause of the bus error.

8.4 BREAKPOINT INSTRUCTION

Execution of the 68020 Breakpoint (BKPT) instruction causes a bus error to be signaled, and hardware causes the illegal instruction exception to be taken. Any breakpoint number specified in the instruction is ignored.

8.5 WATCHDOG TIMER TIMEOUT

When there is no response to an address within 512 clock cycles (watchdog timer time period), a bus error is signaled. The TBE bit (bit 5) of the local Command/Status Register (Subsection 3.7) is set if the watchdog timer times out during an access such as reading a software interrupt address or writing to a nonexisting location. Under these conditions the bus error exception is taken. The bit is also set if the timer times out while waiting for a response during an interrupt acknowledge cycle. In this case, the spurious interrupt exception is taken. Both exception handlers should include reading of this bit. If the bit is set, it should be cleared by writing a 0 to the corresponding bit in the command portion of the Command/Status Register.

8.6 WRITE TO EPROM

When a write to the EPROM on the Utility Module is attempted, a bus error is signaled and the bus error exception is taken. In this case, there is no direct indication that this is the cause of the bus error.

8.7 WRITE TO UTILITY MODULE STATUS REGISTER

When a write to the Status Register on the Utility Module is attempted, a bus error is signaled and the bus error exception is taken. In this case, there is no direct indication, such as a status bit, that this is the cause of the bus error.

8.8 USER-LEVEL WRITE TO 32-BIT TIMER

When a write to the 32-bit timer on the Utility Module is attempted at the user level of privilege, a bus error is signaled and the bus error exception is taken. In this case, there is no direct indication, such as a status bit, that this is the cause of the bus error.

9. OPERATIONAL CONCEPTS

9.1 POWERUP SEQUENCE

SANDAC V application programs are stored in and executed from RAM. Normally, the code executed by a Processor Module is stored in its own local memory. Before execution, application code must be downloaded into RAM with a portable computer known as the Gateway, which interfaces with the SANDAC V global serial bus via the Utility Module external diagnostic port.

Downloading with the Gateway is controlled by the GAteway-to-TARget (GATAR) software package, a SANDAC V monitor and software debugger. GATAR consists of two cooperating programs, one running in the Gateway under MS-DOS and one running in SANDAC V. The SANDAC V portion of GATAR is resident in the global ROM on the Utility Module, and at powerup, all SANDAC V Processor Modules simultaneously execute this code. The code initializes all Processor Modules for possible GATAR activity with the Gateway. Once all Processor Modules are initialized, the one with the highest Processor ID number becomes the "master" processor, and all others "go to sleep."

Initially, the master processor checks for a Gateway running the GATAR support program S5. If detected, communication is established, and SANDAC V responds to GATAR commands issued via the Gateway. If a Gateway running S5 is not detected, the master processor calls a segment of code associated with SANDAC V's Hawk operating system and Hawkeye debugger. Hawkeye is another development tool used for task-level debugging of applications that use the Hawk operating system. Hawkeye is run on a Gateway terminal under MS-DOS, and the Gateway is connected to serial port D of any System I/O Module. If a Gateway running Hawkeye is detected, SANDAC V responds to Hawkeye commands issued via the Gateway for Hawk debugging. Hawkeye also has a binary loading capability that may be invoked when Hawkeye is activated. If a Gateway running Hawkeye is not detected, the master processor jumps to the default start location, where an application program resides if it has been previously loaded by GATAR. The flowchart in Figure 9-1 illustrates the SANDAC V powerup sequence.

9.2 APPLICATION ENVIRONMENT

During software development, GATAR loads, activates, and debugs application code. Before the start of an actual mission, GATAR downloads the final application code into SANDAC V RAM. For the download operation, SANDAC V is powered on, and the Gateway is detected by the master processor. Communication is established, and downloading proceeds under Gateway control. Once the application is loaded, the Gateway is disconnected and the SANDAC V is powered off. Battery-backed RAM keeps the application code intact.

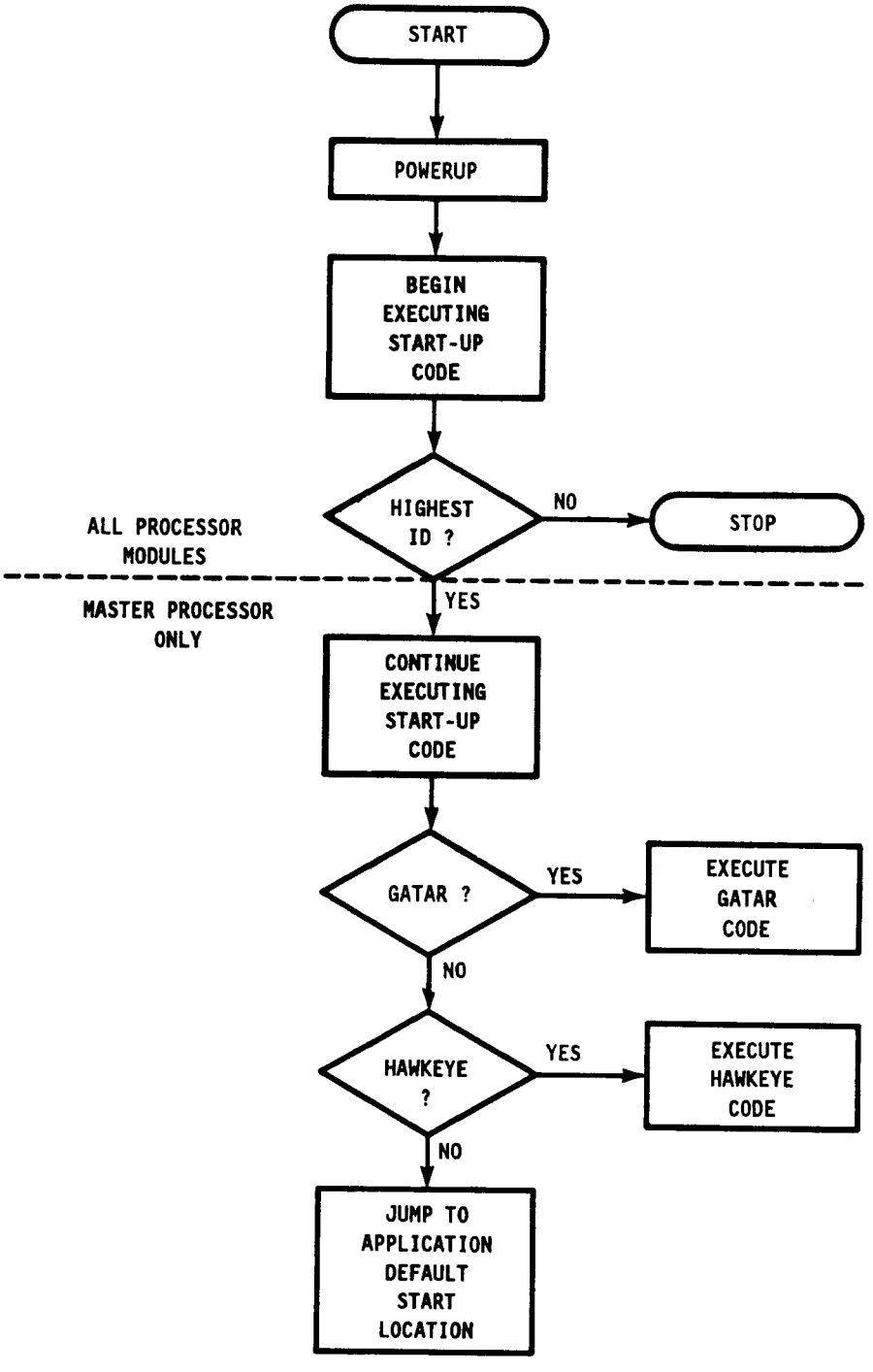


Figure 9-1. SANDAC V Powerup Sequence

When the mission starts and power is applied to SANDAC V, all Processor Modules execute the code in Utility Module ROM. All Processor Modules are initialized, and the one with the highest priority becomes the master, which looks for the Gateway running GATAR and then looks for the Gateway running Hawkeye. After these two checks, the master processor jumps to the default location (80,000,000 hex, the alias for the base address for its own local RAM) and begins executing the previously loaded application program.

During the powerup sequence, several SANDAC V resources are initialized for possible use by GATAR and/or the Hawk operating system. The appendix to this document describes these resources and how they are initialized.

In addition to the resources initialized for GATAR and Hawk, the powerup sequence accomplishes the following:

- On all Processor Modules, 68020 Interrupt Stack Pointer is initialized.
- Extant Processor Table is established in Utility Module RAM.
- Global RAM bit map is established in Utility Module RAM.
- All Processor Modules except master are put to sleep.

These operations are described in the following paragraphs.

9.2.1 68020 ISP Initialization

At powerup, the 68020 on all Processor Modules is reset. When the 68020 accesses the Reset Exception Vector from the EPROM-based exception vector table, the ISP is initialized with the value 80,07F,BE0 hex.

9.2.2 Extant Processor Table

The Extant Processor Table indicates which of the 15 possible SANDAC V Processor Modules are in the system. The table is located in the first 16 bytes of Utility Module RAM (00,080,000 through 00,080,00F hex). Each of the 16 bytes (numbered 0 through 15) in the table corresponds to one of the possible Processor Modules in SANDAC V; that is, byte 1 is for Processor Module 1 (ID = 1), byte 2 is for Processor Module 2 (ID = 2), etc. During powerup, the ID numbers of those Processor Modules actually in the system are loaded into the corresponding table locations. A 00 hex is loaded into locations corresponding to nonexistent Processor Modules. Table entry 0, which corresponds to the nonexistent Processor Module 0, is always loaded with FF hex. The Processor ID number occupies bits 3 through 0 of a table entry. Bit 4 is set to 1 if that Processor Module has a System I/O Module attached to its local expansion bus.

9.2.3 Global RAM Bitmap

The Global RAM bitmap is a single long word in Utility Module RAM at address 00,080,010 hex. It is built during powerup and indicates which, if any, of the 30 possible 512k global memory blocks contain Global Memory Modules. The blocks are numbered 1 through 30, and bits 1 through 30 of the bitmap correspond to these blocks. A 1 in a bitmap position indicates a Global Memory Module is present in the corresponding global memory block; a 0 indicates no module is present. Bits 0 and 31 are always 0.

9.2.4 Sleeping Processor Modules

During the powerup sequence, all Processor Modules compare their Processor IDs with the contents of the Extant Processor Table. The module with the highest ID (highest priority) becomes the master processor and continues to run; all the other modules put themselves to sleep.

When a module determines that it is not the highest priority, its 68020 is directed to an address where it executes the following instruction:

```
stop #2000
```

The immediate operand for the instruction (2000 hex) is moved into the 68020 Status Register, the program counter is advanced to point to the next instruction, and the MPU stops fetching instructions. Instruction execution will resume when a trace, a reset, or an interrupt exception occurs.

9.3 EXCEPTIONS DURING POWERUP

During the powerup sequence, each Processor Module builds an exception vector table in its own local RAM. This table is set up primarily for use by GATAR and is described in the appendix to this document. Before this table is built, however, all Processor Modules use the exception vector table stored in the Utility Module EPROM at addresses 00,000,000 through 00,000,3FF hex. The EPROM table halts a Processor Module with a double bus error if an exception is taken.

9.4 ACCESSING RESOURCES

Although SANDAC V contains global resources on the global bus and local resources on Processor Module local buses, *the address space for the entire system is linear.* The 32-bit addressing capability of the 68020 MPU gives each Processor Module an address range of 4 gigabytes, and the global bus provides access to all SANDAC V address space.

By issuing the proper address, any Processor Module can access any global or local resource, including its own local resources (local access), any resource on the global bus (global access), or the local resource of any other Processor Module (global-to-local access). The following summarizes local and global accesses:

Local — Processor Module's 68020 MPU issues address within local address space assigned to that module. Access of location is via module's local bus or local expansion bus.

Global — Processor Module's 68020 MPU issues address assigned to global module (Utility Module or a Global Memory Module). Access of location is via global bus.

Global-to-Local — Processor Module's 68020 MPU issues address within address space of another Processor Module. Access of location is via global bus and local bus of addressed module.

9.4.1 Global Bus Arbitration

With multiple Processor Modules the possibility exists that two or more will try to use the global bus at the same time, so each SANDAC V Processor Module has a priority to determine which can use the bus first. A module's priority is related to its ID as shown below:

ID	Address Space (hex)	Priority
1	01,000,000-01,FFF,FFF	Lowest
2	02,000,000-02,FFF,FFF	.
3	03,000,000-03,FFF,FFF	.
4	04,000,000-04,FFF,FFF	.
5	05,000,000-05,FFF,FFF	.
6	06,000,000-06,FFF,FFF	.
7	07,000,000-07,FFF,FFF	.
8	08,000,000-08,FFF,FFF	.
9	09,000,000-09,FFF,FFF	.
10	0A,000,000-0A,FFF,FFF	.
11	0B,000,000-0B,FFF,FFF	.
12	0C,000,000-0C,FFF,FFF	.
13	0D,000,000-0D,FFF,FFF	.
14	0E,000,000-0E,FFF,FFF	.
15	0F,000,000-0F,FFF,FFF	Highest

Before using the global bus, a Processor Module initiates bus arbitration, comparing its priority to the priority of other modules requesting the global bus. If the bus is not busy, the module with the highest priority becomes the bus master.

SANDAC V has two modes of global bus arbitration, selectable with software:

Priority Mode Processor Module with highest priority (highest ID number) controls bus for as long as it requests it.

Fair Mode After Processor Module has been bus master, it cannot make another request until the bus has been free for at least one cycle. Therefore, when two or more modules contend for the bus, the highest priority module is served first, but it is prevented from acquiring the bus again until the lowest priority module has been allowed access.

9.4.2 Bus Mastership

After a Processor Module is given global bus mastership through bus arbitration, it has exclusive use of the bus for a single access cycle. If the resource accessed is in the address space of another Processor Module, the accessing module, after gaining global bus mastership, must gain control of the accessed Processor Module's local bus. The accessing module then has exclusive use of the accessed module's address space for a single access cycle.

9.4.3 Access Time

If an access is within the Processor Module's local space, the 68020 MPU can run at maximum speed. Generally, no wait states are required when accessing a local resource. If an access is of a global location, three wait states are required for global bus arbitration. Additional wait states may be required by the resource being accessed. This assumes the global resource is available and there are no bus requests from other Processor Modules pending. If the access is of a location in another Processor Module's address space, three wait states are required for global bus arbitration and another six wait states are required to gain control of the local bus. This assumes the local MPU is idling. Additional wait states may be required by the resource being accessed.

Table 9-1 summarizes the number of wait states required when accessing SANDAC V resources.

TABLE 9-1. WAIT STATE SUMMARY

Resource	Wait States When Accessed By Local MPU	Wait States When Accessed By Remote MPU
Utility Module (UM) EPROM	Not applicable	3 + # selected by UM jumper
UM RAM	Not applicable	3 + # selected by UM jumper
UM 32-Bit Timer	Not applicable	3 + # selected by UM jumper
UM Status Register	Not applicable	3 + # selected by UM jumper
Global Memory Module (GMM)	Not applicable	3 + # selected by GMM jumper
Processor Module (PM) RAM	0 to 3 (by jumper)	10 + # selected by PM jumper
PM 68901 MFP	Inserted by device	10 + # inserted by device
PM FIFO	3	13
PM Configuration Register	3	13
PM Command/Status Register	3	13
PM Software Interrupts	3	13
System I/O Module	Inserted by device	10 + # inserted by device
MIL-STD-1553B Module	Inserted by device	10 + # inserted by device
Local Exp. Memory Module	1-8 module jumper	10 + # selected by PM jumper

9.5 PERFORMANCE CONSIDERATIONS

The maximum SANDAC V global bus bandwidth is 10 megabytes (Mb) per second. For maximum performance, programs and data should be stored in the local memory space of the Processor Module executing the program. Data sharing is practical via global memory and/or between Processor Modules as long as its use is consistent with the bus bandwidth.

Normally, the global bus bandwidth is high enough that the execution time of a specific task in a single-processor SANDAC V is not significantly different from the execution time of the same task in a 15-processor system. However, in certain pathological cases where Processor Modules are accessing global resources in a very tight loop, the global bus will reach saturation with three Processor Modules. In such a case, the fourth or any additional Processor Modules will either have limited access to the global bus or will significantly reduce the throughput of the first three Processor Modules.

9.6 RESET CONDITIONS

Reset occurs under any of the following conditions:

- When power is applied, i.e., powerup
- When the RST bit in the Processor Module Command/Status Register is set (a detailed description of the RST bit's function is given in 3.7)
- When a RESET instruction is executed by the Processor Module 68020

Table 9-2 summarizes the activity from the viewpoint of the Processor Module during the different reset conditions.

TABLE 9-2. SUMMARY OF RESET ACTIVITY

Activity	At Powerup	By RST Bit	By RESET Instruction
Reset 68020	yes	yes	no
Reset 68881	yes	yes	yes
Reset 68901	yes	yes	yes
Clear 512-byte FIFO	yes	yes	yes
Reset Command/Status Register	yes	yes	yes
Clear pending 68020 interrupt requests	yes	yes	yes
Disable bus arbitration for 256 clock cycles	no	yes	no
Reset bus error and halt logic	yes	yes	yes
Reset Configuration Registers to default conditions	yes	yes	yes
Send Reset signal to local expansion bus	yes	yes	yes

NOTES

APPENDIX: GATAR AND HAWK RESOURCE UTILIZATION

A.1 INTRODUCTION

A.1.1 GATAR Software Debugger

The GATAR (GAteway-to-TARget) software package provides machine level debugging for each Processor Module in the SANDAC V along with limited I/O and a memory monitor feature. GATAR includes the EPROM-based code initially executed at powerup. This startup code initializes the Processor Module's gate array and 68901 MFP and builds an exception vector table to allow the GATAR software to communicate with the external Gateway computer. GATAR also initializes other SANDAC V hardware that it does not use.

A.1.2 Hawk Operating System Kernel

The Hawk operating system kernel provides services to an application at a level of control above the hardware level. Hawk supports the "virtual machine" concept with each task in a virtual machine and with message queues between tasks.

Hawk has an associated debugger called Hawkeye. During the SANDAC V startup sequence, the GATAR code calls a segment of code associated with Hawk and Hawkeye. This code initializes certain SANDAC V resources in the event that the Hawk operating system kernel or the Hawkeye debugger will be used.

A.2 RESOURCES USED BY GATAR AND HAWK

Both GATAR and Hawk use some of the following SANDAC V resources:

- Memory (Global and Local)
- Local resources (Processor Module and System I/O Module)
- Interrupt Requests
- Exception Vectors

How these resources are used is described in the following subsections.

A.2.1 GATAR Memory Utilization

GATAR reserves the following SANDAC V memory resources:

Global EPROM (Utility Module)	00,000,000 - 00,007,FFF hex
Global RAM (Utility Module)	00,080,000 - 00,080,07F hex
Local RAM (Each Processor Module)	0n,07C,000 - 0n,07F,FFF hex (n is the Processor ID in hex. This is also local Processor Module alias 80,07C,000 through 80,07F,FFF.)

The space in Utility Module EPROM includes the SANDAC V startup code and a startup exception vector table. Twenty bytes in Utility Module RAM contain the Extant Processor Table (9.2.2) and the Global RAM Bitmap (9.2.4).

The top 16K of Processor Module RAM is used by GATAR for internal data structures. The space is defined from addresses 0n,07C,000 through 0n, 07F,FFF hex (n = Processor ID), the top of the fourth 128K block of local RAM space where no RAM is physically present. Local Processor Module aliasing puts the actual code at locations 0n,01C,000 through 0n,01F,FFF. GATAR builds an exception vector table, which replaces the EPROM-based exception vector table, in this space starting at location 0n,07F,C00 hex. This exception vector table is described in Subsection A.2.7.

See Figures A-1 through A-3 for memory maps showing GATAR memory utilization.

A.2.2 Hawk Memory Utilization

Hawk uses the following SANDAC V memory resources:

Global EPROM (Utility Module)	00,008,000 - 00,01F,FFF hex
Global RAM (Utility Module)	00,080,080 - 00,09E,6EC hex
Local RAM (Processor Module)	0n,07B,800 - 0n,07F,C00 hex (n is the Processor ID in hex. This is also local Processor Module alias 80,07B,800 through 80,07F,C00.)

The space in Utility Module EPROM holds the operating system kernel. If the kernel is used by an application, it is copied into the local RAM of the Processor Module that will use it. Address 00,008,000, the start address for Hawk, contains an RTS instruction if HAWK is not in the EPROM. Additionally, approximately 121K of Utility Module RAM is required for Hawk to function. Local pool memory for Hawk is not included in the Local RAM usage shown avove.

Memory maps of Hawk memory utilization are shown in Figures A-1 through A-3.

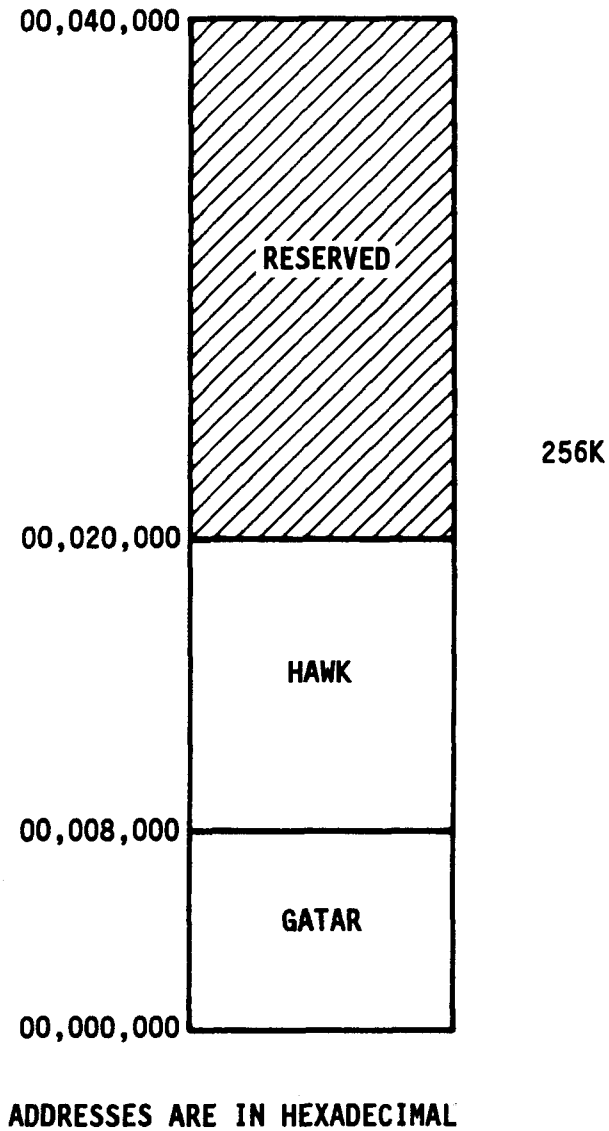


Figure A-1. Utility Module EPROM Utilization

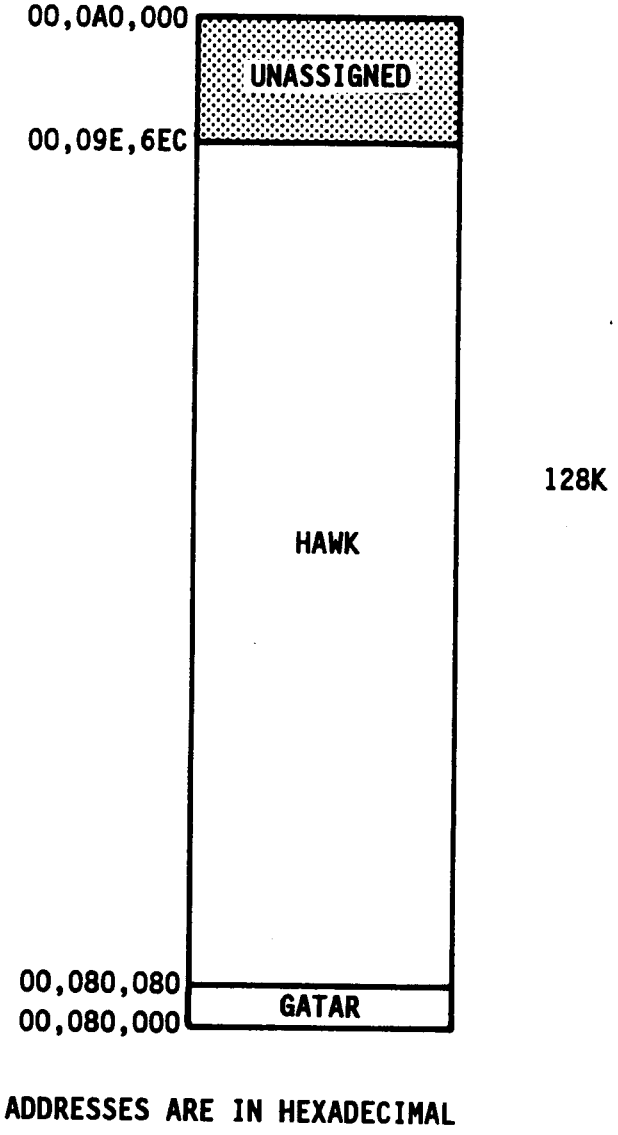
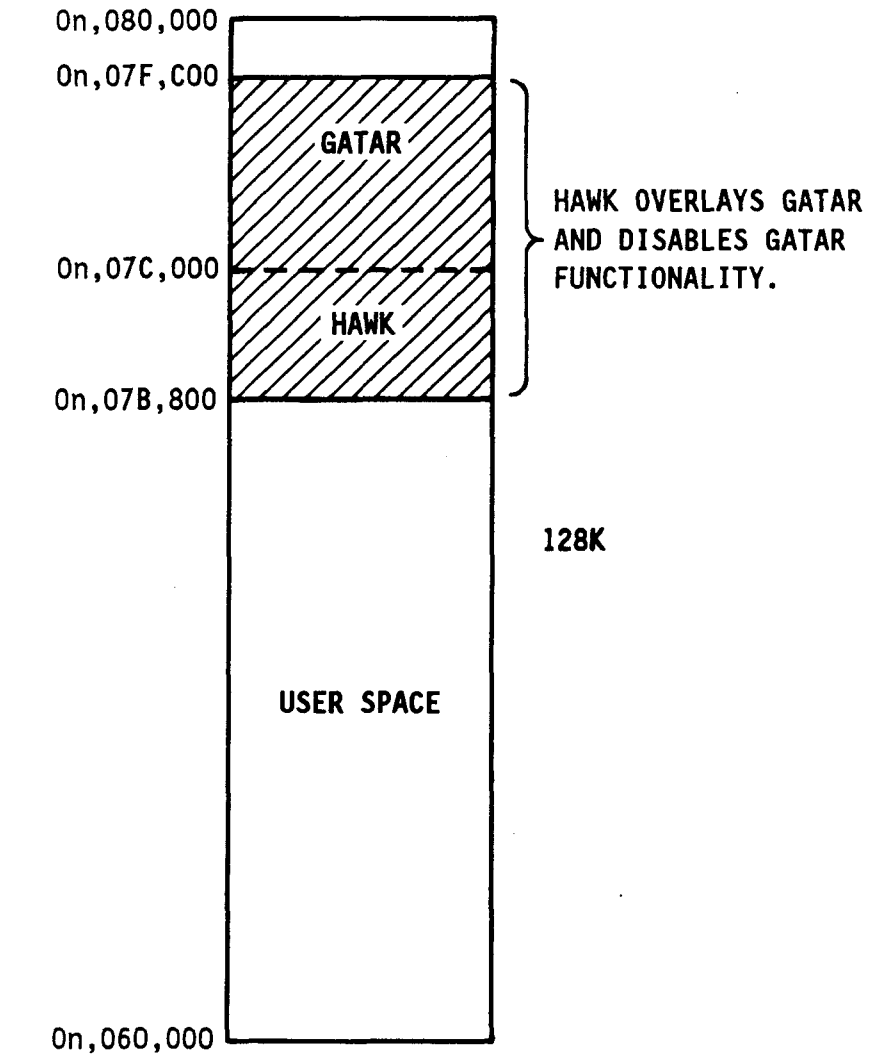


Figure A-2. Utility Module RAM Utilization



ADDRESSES ARE IN HEXADECIMAL
(n IS THE PROCESSOR ID IN HEX)

Figure A-3. Processor Module RAM Utilization

A.2.3 GATAR Local Resource Utilization

GATAR communicates with each Processor Module in SANDAC V via the global serial bus. The Processor Module 68901 MFP's Timers C and D and USART form the global serial bus interface.

GATAR initializes the Processor Module 68901 USART and timer registers as follows:

Register*	Initialized To (hex)	Comments
TACR	-	Not initialized
TBCR	-	Not initialized
TCDCR	11	Delay mode, ÷4 prescaler
TADR	-	Not initialized
TBDR	-	Not initialized
TCDR	01	Receive clock count-down value
TDDR	01	Transmit clock count-down value
UCR	88	USART 16x, 8-bit data, 1 stop bit, no parity
TSR	03	Xmtr enabled, output low when disabled
RSR	01	Rcvr enabled

*68901 registers are described in 3.8.5.

A.2.4 GATAR Interrupt Utilization

GATAR uses the following Processor Module interrupts:

- 68901 USART Receive Buffer Full
- 68901 USART Transmit Buffer Empty
- Software Interrupt 3

To utilize these interrupts, GATAR initializes the Processor Module 68901 MFP general-purpose I/O and interrupt controller registers as follows:

Register*	Initialized To (hex)	Comments
DDR	00	I0-I7 programmed as inputs
AER	-	Not initialized (default is falling edge)
IERA	14	Enable RCV Buffer Full, XMIT Buffer interrupts
IERB	08	Enable I3 interrupt (Software Interrupt 3 directed to I3)
IMRA	14	Unmask RCV Buffer Full, XMIT Buffer interrupts

Register*	Initialized To (hex)	Comments
IMRB	08	Unmask I3 interrupt (Software Interrupt 3 directed to I3)
IPRA	-	Not initialized
IPRB	-	Not initialized
ISRA	-	Not initialized
ISRB	-	Not initialized
VR	40	68020 interrupt vectors 40 - 4F hex (64 - 79 decimal)

*68901 registers are described in 3.8.5.

GATAR requires that the vectored 68901 interrupt request be connected to the interrupt priority level (IPL) encoder for a 68020 IPL of 3; it also requires that Software Interrupt 3 be connected to the 68901's I3 input. These connections are made by programming the 6x9 and 10x7 interrupt crossbar switches via the configuration registers in the Processor Module gate array. During the powerup sequence, GATAR programs the configuration registers as shown below:

Register*	Initialized To (hex)	Comments
CONFIG1 thru CONFIG7	-	Not initialized
CONFIG8	23	FIFO = IPL2; 68901 = IPL3
CONFIG9	56	Local IRQ W = IPL5; Local IRQ X = IPL6
CONFIG10	40	Local IRQ Y = IPL4; Local IRQ Z not connected
CONFIG11	-	Not initialized
CONFIG12	F5	Global IRQ Y not connected; Global IRQ Z to 68901 I5
CONFIG13	01	Software Interrupt 0 to 68901 I0; Software Interrupt 1 to 68901 I1
CONFIG14	23	Software Interrupt 2 to 68901 I2; Software Interrupt 3 to 68901 I3
CONFIG15	10	Software Interrupt 4 = IPL1; Software Interrupt 5 not connected

*Configuration registers are described in 3.9.1 and 3.9.2.

Some of the configuration register initialization accommodates interrupts used by Hawk. Figure A-4 shows the Processor Module interrupt system as it is set up by GATAR.

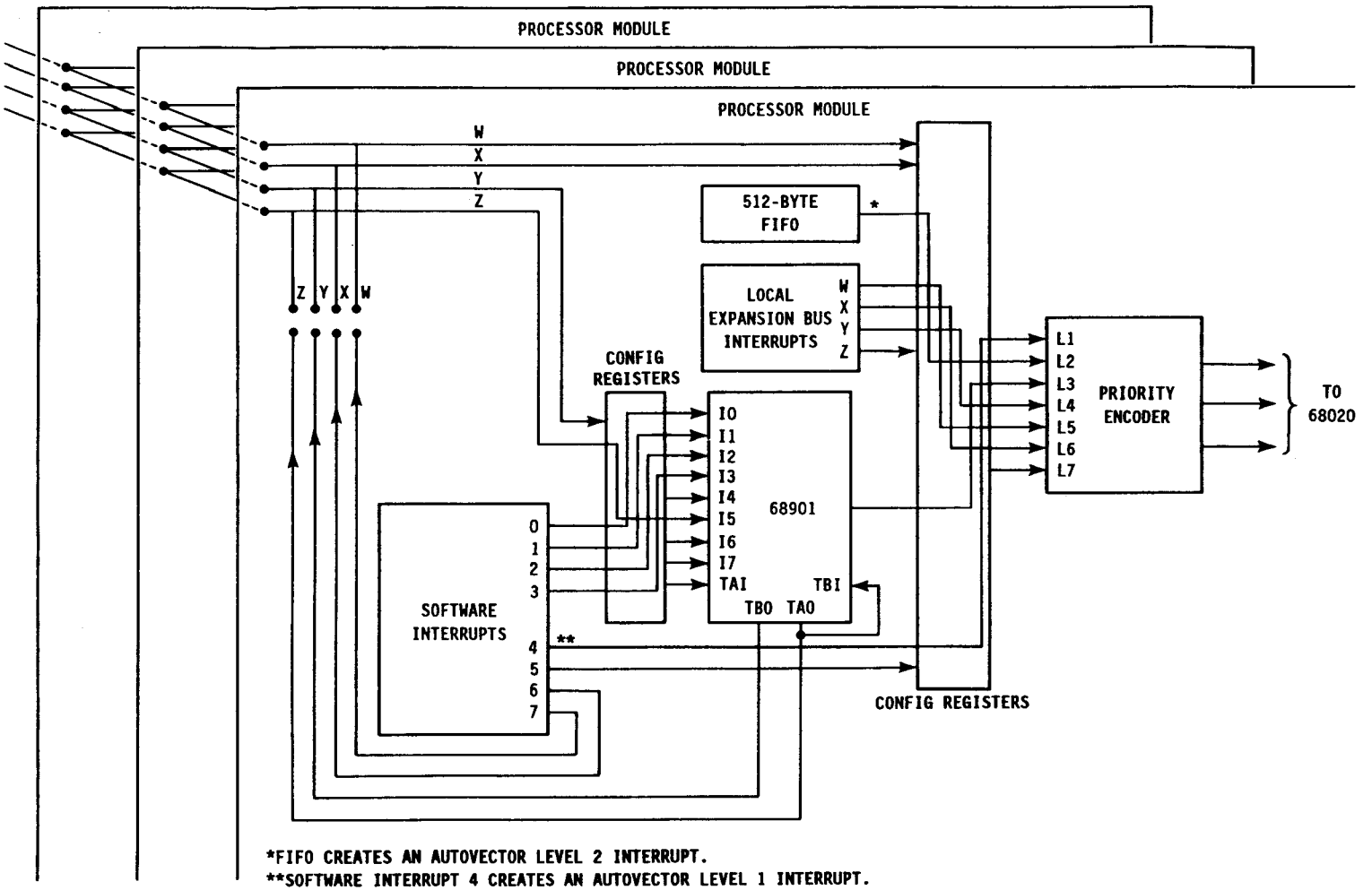


Figure A-4. Interrupt System as Initialized by GATAR

A.2.5 Hawk Local Resource Utilization

Hawk uses the following local resources:

Processor Module:	68901 MFP Timers A and B
System I/O Module:	Serial port D (DUSCC #2 channel B)

In using the Processor Module 68901 timers, Hawk controls the 68901's TACR, TCBR, TADR, and TBDR registers. In using the System I/O Module's serial port D, Hawk controls all the registers associated with DUSCC #2 channel B. Additionally, the Pin Configuration Register (PCR) of DUSCC #1 must be programmed to provide an interrupt daisy chain to DUSCC #2.

A.2.6 Hawk Interrupt Utilization

Hawk uses the following Processor Module interrupt requests:

- Processor Module 68901 Timer A
- Processor Module 68901 Timer B
- Software Interrupt 4
- Local IRQ W

The 68901's interrupt controller handles the timer interrupts; the 68901 interrupt is configured by the GATAR initialization routine as a vectored interrupt at an interrupt priority level (IPL) of 3. GATAR also configures Software Interrupt 4 at IPL 1 and Local IRQ W at IPL 5.

Figure A-4 shows the Processor Module interrupt system, including Hawk interrupts, as set up by GATAR.

A.2.7 Exception Vectors (GATAR and Hawk)

At powerup, SANDAC V initially uses an EPROM-based exception vector table until GATAR builds the Default Exception Vector Table in each Processor Module's local RAM at base address 80,07F,C00 hex. (The local 68020 Vector Base Register is initialized with this base address.)

The Default Exception Vector Table provides handlers for all possible exceptions in the GATAR environment. The only vectors required for GATAR operation are the ones related to serial communication (vectors 74 and 76), interprocessor communication (67), 68020 Traps 4 and 5 (36 and 37), the 68020 illegal instruction (4), and the 68020 Trace (9). GATAR initializes all other vectors to direct the MPU to either a fatal or a

nonfatal exception handler. A fatal exception handler (FEH) reports the exception and stops the MPU from running. A nonfatal exception handler (NFH) reports the exception but leaves the MPU running.

Hawk also uses the exception vector table built in local RAM, but it redefines most of the GATAR exceptions except Trap 4.

A composite GATAR and Hawk exception vector table is given in Table A-1. In the table, the FEH or NFH notation indicates the type of handler provided by GATAR.

NOTE

If software is being developed without the aid of Hawk or another embedded operating system kernel, the exceptions used by Hawk and other software packages should be avoided as much as possible to allow for inclusion of these packages as they mature or system requirements allow their use. A critical application, such as flight software for a missle system, should establish its own exception handler for all exceptions, including those not used, so that a single exception during a flight does not cause the processor to halt for the duration of the flight.

TABLE A-1. SANDAC V EXCEPTION VECTORS

Vector Number (decimal)	VBR Offset (Hex)	Assignment
0	000	68020 Reset: Initial Interrupt Stack Pointer
1	004	68020 Reset: Initial Program Counter
2	008	68020 Bus Error (FEH)
3	00C	68020 Address Error (FEH)
4	010	68020 Illegal Instruction--Used by GATAR for breakpoint
5	014	68020 Zero Divide (FEH)
6	018	68020 CHK, CHK2 Instruction (FEH)
7	01C	cpTRAPcc, 68020 TRAPcc, TRAPV Instructions (FEH)
8	020	68020 Privilege Violation (FEH)
9	024	68020 Trace--Used by GATAR for single step
10	028	68020 Line 1010 Emulator (FEH)
11	02C	68020 Line 1111 Emulator (FEH)
12	030	68020 Reserved (FEH)
13	034	Coprocessor Protocol Violation (FEH)
14	038	68020 Format Error--(FEH); used by VRTX to point to configuration table when VRTX is used in the system
15	03C	68020 Uninitialized Interrupt (FEH)
16-23	040-5C	68020 Reserved (FEH)
24	060	68020 Spurious Interrupt (FEH)

TABLE A-1. CONTINUED

Vector Number (decimal)	VBR Offset (Hex)	Assignment
25	064	68020 Level 1 Interrupt Auto Vector--GATAR NFH; Software Interrupt 4 used by Hawk
26	068	68020 Level 2 Interrupt Auto Vector--GATAR NFH; Processor Module FIFO
27	06C	68020 Level 3 Interrupt Auto Vector (NFH)
28	070	68020 Level 4 Interrupt Auto Vector (NFH)
29	074	68020 Level 5 Interrupt Auto Vector (NFH)
30	078	68020 Level 6 Interrupt Auto Vector (NFH)
31	07C	68020 Level 7 Interrupt Auto Vector (NFH)
32	080	68020 TRAP #0--GATAR NFH; used by Hawk; used by VRTX to make user calls when VRTX is used in the system
33	084	68020 TRAP #1 (NFH)
34	088	68020 TRAP #2 (NFH)
35	08C	68020 TRAP #3 (NFH)
36	090	68020 TRAP #4--Used by GATAR
37	094	68020 TRAP #5--Used by GATAR
38	098	68020 TRAP #6 (NFH)
39	09C	68020 TRAP #7 (NFH)
40	0A0	68020 TRAP #8 (NFH)
41	0A4	68020 TRAP #9 (NFH)
42	0A8	68020 TRAP #10 (NFH)

TABLE A-1. CONTINUED

Vector Number (decimal)	VBR Offset (Hex)	Assignment
43	0AC	68020 TRAP #11 (NFH)
44	0B0	68020 TRAP #12 (NFH)
45	0B4	68020 TRAP #13 (NFH)
46	0B8	68020 TRAP #14 (NFH)
47	0BC	68020 TRAP #15--GATAR NFH; used by Hawk
48	0C0	68881 Branch or Set on Unordered Condition (FEH)
49	0C4	68881 Inexact Result (FEH)
50	0C8	68881 Divide by Zero (FEH)
51	0CC	68881 Underflow (FEH)
52	0D0	68881 Operand Error (FEH)
53	0D4	68881 Overflow (FEH)
54	0D8	68881 Signaling NAN (FEH)
55	0DC	68020 Reserved (FEH)
56	0E0	68851 Configuration (FEH)
57	0E4	68851 Illegal Operation (FEH)
58	0E8	68851 Access Level Violation (FEH)
59-63	0EC-0FC	68020 Reserved (FEH)
64	100	Processor Module 68901 Input I0--Initialized by GATAR to handle Software Interrupt 0 (NFH)
65	104	Processor Module 68901 Input I1--Initialized by GATAR to handle Software Interrupt 1 (NFH)

TABLE A-1. CONTINUED

Vector Number (decimal)	VBR Offset (Hex)	Assignment
66	108	Processor Module 68901 Input I2--Initialized by GATAR to handle Software Interrupt 2 (NFH)
67	10C	Processor Module 68901 Input I3--Initialized by GATAR to handle Software Interrupt 3; used by GATAR for interprocessor communication.
68	110	Processor Module 68901 Timer D--Not Used (NFH)
69	114	Processor Module 68901 Timer C--Not Used (NFH)
70	118	Processor Module 68901 Input I4 (NFH)
71	11C	Processor Module 68901 Input I5--Initialized by GATAR to handle Global IRQ Z (NFH)
72	120	Processor Module 68901 Timer B--GATAR NFH; used by Hawk
73	124	Processor Module 68901 USART Transmitter Error (NFH)
74	128	Processor Module 68901 USART Transmit Buffer Empty--Used by GATAR
75	12C	Processor Module 68901 USART Receive Error (NFH)
76	130	Processor Module 68901 USART Receiver Buffer Full--Used by GATAR
77	134	Processor Module 68901 Timer A--GATAR NFH; used by Hawk
78	138	Processor Module 68901 Input I6 (NFH)
79	13C	Processor Module 68901 Input I7 (NFH)
80	140	System I/O Module 68901 Input I0 (NFH)
81	144	System I/O Module 68901 Input I1 (NFH)

TABLE A-1. CONTINUED

Vector Number (decimal)	VBR Offset (Hex)	Assignment
82	148	System I/O Module 68901 Input I2 (NFH)
83	14C	System I/O Module 68901 Input I3 (NFH)
84	150	System I/O Module 68901 Timer D (NFH)
85	154	System I/O Module 68901 Timer C (NFH)
86	158	System I/O Module 68901 Input I4 (NFH)
87	15C	System I/O Module 68901 Input I5 (NFH)
88	160	System I/O Module 68901 Timer B (NFH)
89	164	System I/O Module Serial Port E (68901 USART) Transmit Error (NFH)
90	168	System I/O Module Serial Port E (68901 USART) Transmit Buffer Empty (NFH)
91	16C	System I/O Module Serial Port E (68901 USART) Receive Error (NFH)
92	170	System I/O Module Serial Port E (68901 USART) Receiver Buffer Full (NFH)
93	174	System I/O Module 68901 Timer A (NFH)
94	178	System I/O Module 68901 Input I6 (NFH)
95	17C	System I/O Module 68901 Input I7 (NFH)
96	180	System I/O Module Serial Port A (DUSCC #1 channel A) Receiver Ready (NFH)
97	184	System I/O Module Serial Port A (DUSCC #1 channel A) Transmitter Ready (NFH)

TABLE A-1. CONTINUED

Vector Number (decimal)	VBR Offset (Hex)	Assignment
98	188	System I/O Module Serial Port A (DUSCC #1 channel A) Rx/Tx Status (NFH)
99	18C	System I/O Module Serial Port A (DUSCC #1 channel A) Ext. or C/T Status (NFH)
100	190	System I/O Module Serial Port B (DUSCC #1 channel B) Receiver Ready (NFH)
101	194	System I/O Module Serial Port B (DUSCC #1 channel B) Transmitter Ready (NFH)
102	198	System I/O Module Serial Port B (DUSCC #1 channel B) Rx/Tx Status (NFH)
103	19C	System I/O Module Serial Port B (DUSCC #1 channel B) Ext. or C/T Status (NFH)
104	1A0	System I/O Module Serial Port C (DUSCC #2 channel A) Receiver Ready (NFH)
105	1A4	System I/O Module Serial Port C (DUSCC #2 channel A) Transmitter Ready (NFH)
106	1A8	System I/O Module Serial Port C (DUSCC #2 channel A) Rx/Tx Status (NFH)
107	1AC	System I/O Module Serial Port C (DUSCC #2 channel A) Ext. or C/T Status (NFH)
108	1B0	System I/O Module Serial Port D (DUSCC #2 channel B) Receiver Ready; GATAR NFH; used by Hawk
109	1B4	System I/O Module Serial Port D (DUSCC #2 channel B) Transmitter Ready; GATAR NFH; used by Hawk
110	1B8	System I/O Module Serial Port D (DUSCC #2 channel B) Rx/Tx Status; GATAR NFH; used by Hawk

TABLE A-1. CONTINUED

Vector Number (decimal)	VBR Offset (Hex)	Assignment
111	1BC	System I/O Module Serial Port D (DUSCC #2 channel B) Ext. or C/T Status; GATAR NFH; used by Hawk
112	1C0	System I/O Module 63450 Channel 0 Normal Interrupt (NFH)
113	1C4	System I/O Module 63450 Channel 0 Error Interrupt (NFH)
114	1C8	System I/O Module 63450 Channel 1 Normal Interrupt (NFH)
115	1CC	System I/O Module 63450 Channel 1 Error Interrupt (NFH)
116	1D0	System I/O Module 63450 Channel 2 Normal Interrupt (NFH)
117	1D4	System I/O Module 63450 Channel 2 Error Interrupt (NFH)
118	1D8	System I/O Module 63450 Channel 3 Normal Interrupt (NFH)
119	1DC	System I/O Module 63450 Channel 3 Error Interrupt (NFH)
120-143	1E0-23C	Reserved for System I/O Module to vector devices on parallel port (NFH)
144-151	240-25C	Reserved for VME interface under development (NFH)
152-159	260-27C	Reserved for MIL-STD-1553B Module or other devices assigned to extended I/O memory space (NFH)
160-255	280-3FC	Unused (NFH)

The following notes apply to Table A-1:

1. All unused exceptions can be modified and used by the application software. This will not affect the system software beyond the normal effects of stealing time from lower priority interrupts. (Each hardware interrupt has a priority level associated with it as defined above, and each processor created exception has a priority as described in the 68020 user's manual.)

2. All 256 exception vectors must be treated the same. Some sort of default handler is necessary on every exception, even those reserved for future use because vectored hardware devices are not restricted from returning reserved vector numbers.

3. The handlers for exception vectors 2 and 3 (bus error and address error) may need to be jointly defined by all coresident software because of the overloaded use of these exceptions. (See Section 8.)

4. Exception vector 4 (illegal instruction) may be modified only if users want to implement their own breakpoint debugging capability. The same comments can be made regarding exception vector 9 (Trace).

5. Exception vectors 48 through 54 are disabled by the startup code and will not occur unless the user enables them by writing to the 68881 Floating-Point Coprocessor Control Register.

6. Exception vector 11 should never occur while a valid coprocessor is installed on each Processor Module.

7. To avoid future conflicts, all the 68020 reserved exceptions should be avoided. No catastrophic results will occur if they are used at present, but their use may preclude utilization of new parts available in the future.

8. VRTX notes: Hunter & Ready's VRTX uses vector 14 to point to the configuration table, and typically, trap #0 provides the user interface to VRTX.

DISTRIBUTION:

Honeywell, Inc. (5)
Attn: Scott Lapham
John Abernathy
Ed Householder (2)
Charlie Blaine
Avionics Division
13350 U.S. Highway 19
Clearwater, FL 33516

ERIM
Attn: Monique L. Boucher
P.O. Box 8618
Ann Arbor, MI 48107-8616

Hughes Aircraft Company
Missile Systems Group
Attn: Ralph Rodriguez
Canoga Park, CA 91304-0445

AFWAL/FIBL
Attn: Ray Bortner
WPAFB, OH 45433-6553

AVRADA (2)
Attn: Dominick Pondaco
Hexagon Building
Fort Monmouth, NJ 07703

1411 W.M. Davidson
1411 M.M. Moya
1411 L.P. Ray
1412 S.J. Weissman
1412 G.S. Davidson
1412 G.S. Davidson (for University of Indiana)
1412 M.L. Yee
1413 V.G. Grafe
1413 V.P. Holmes

2335 J.W. Bartholomew
2335 J.T. Cummings
2335 D.L. Eilers
2335 T.P. Koepp
2335 K.K. Ma
2335 J.A. McCoy
2335 P.E. Pierce

2336 S.M. Becker
2336 C.R. Borgman (25)
2336 L.J. Dalton (25)
2336 D.L. Harris
2336 D.A. Johnson
2336 K.B. Kimball
2336 J.F. Kobs
2336 N.R. Kolb
2336 N.R. Kolb (for University of Arizona)
2336 B.T. Meyer
2336 E.J. Nava
2336 K.M. Piorkowski
2336 S.W. Reynolds
2336 S.C. Richards
2336 G.A. Russell
2336 H. Shen

8131 W.R. Bolton
8131 S.E. Faas
8131 G.A. Hawley

9131	D.D. Boozer
9131	J.D. Bradley
9131	E.E. Creel
9131	J.R. Fellerhoff
9131	J.A. Hollowell
9131	M.L. Pedroncelli
9131	R. Wardlaw
9132	C.L. Cook
9132	R.S. Edmunds
9132	J.R. Etter
9132	F.V. James
9132	A.M. Oczon
9132	J.R. Phelan
9132	B.A. Sims
9132	A.C. Watts
9132	J.E. White
9133	G.W. Donohoe
9133	R.J. Fogler
9133	C.M. Hart
9133	L.D. Hostetler
9133	S.D. Martinez
9133	M.S. Murray
3141	S.A. Landenberger (5)
3151	W.L. Garner (3)
3154-1	C.H. Dalin (28) for DOE (OSTI)
8524	P.W. Dean