CHAPTER 3

HOST INTERFACE SUBSYSTEM

The Host Interface Subsystem is responsible for providing the Micro Channel Architecture (MCA) bus interface for the MGR adapter. It also provides a Local bus interface to the Geometry Subsystem which is compatible with the SGI private bus. The MCA interface allows the MGR card to operate in any system that has the MCA bus. The performance of the adapter will depend on the level of MCA supported on the system.

The major function of this subsystem is to perform the appropriate MCA bus data transfers and to route the data from the MCA bus to the Local bus which distributes this information to the other subsystems. All data transfers will be initiated by the host CPU via a command to the Host Interface Subsystem or to the Geometry Subsystem. When a command requesting DMA service is received, the Host Interface Subsystem arbitrates for the Micro Channel and transfers the data by performing a Bus Master DMA Cycle. This cycle will conform to the Data Streaming Architecture if it is supported by the Host system. Refer to the host system's technical reference manual for a description of the DMA modes supported.

The following sections will describe the external interfaces+ the major components, the registers. the interrupts, the basic operations and the programming considerations of the Host Interface Subsystem.

External Interfaces

The Host Interface Subsystem has external interfaces to the host system via the MCA bus. It also has an external interface to the Geometry Subsystem via the Local bus. The following sections will describe these two interfaces.

MCA Bus Interface

The MCA bus interface has the following features:

- A 32-bit Address bus
- A 32-bit Data bus
- A 4-bit Arbitration bus
- Control Signals for:
  - Address and Data Validation
  - Arbitration and Bus Ownership
  - Bus Cycle Sequencing
  - Pacing Control
- Level Sensitive Interrupt Signals

The Host Interface Subsystem takes advantage of the fastest means of transfer supported in the MCA. The following data transfer functions are supported:

- 32-bit Memory Transfers
- 32-bit Bus Master DMA
- 32-bit Data Streaming (Master Only)

To preserve data integrity, when performing Bus Master DMA, the Host Interface Subsystem implements Data and Address Panty.

Local Bus Interface

The Local bus is a private interface which is used to communicate with the Geometry Subsystem. This bus is compatible with the SGI private bus used on the Personal IRIS workstation. Access to the Local bus can be accomplished via memory mapped I/O or burst DMA transfers. The Local bus consists of:

- A 32-bit Address/Data bus
- An-Address strobe and a burst signal
- A Read/Write signal
- Interrupt signals to host
- Pacing control signals
- Clocks

Major Components

The major components of the Host Interface Subsystem are shown in the block diagram of Figure 3.1. The following paragraphs describe these components.

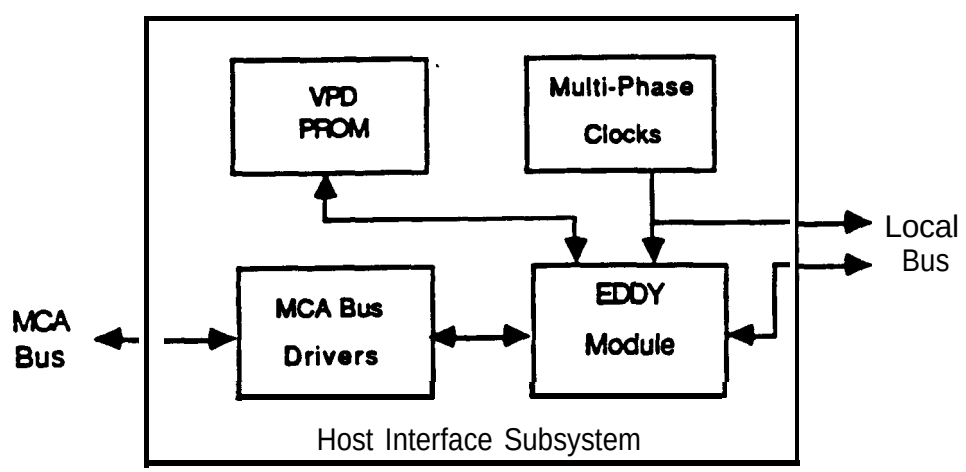


Figure 3.1 Host Interface Subsystem Block Diagram

Eddy module

The Eddy module is the main component of the Host Interface Subsystem. It is responsible for providing the MCA bus interface to the host system. It is also responsible for providing a Local bus interface to the Geometry Subsystem which is compatible with the SGI private bus. The Eddy module is a Bus Master DMA / Memory Slave device for the Micro Channel Architecture.

The Eddy module will support a variety of modes of data transfer on the MCA bus, which include:

- Setup Cycles
- 32 bit Memory Slave Cycles
- 32 bit Normal Bus Master Cycles
- 32 bit Bus Master Data Streaming Cycles

The Eddy module only responds to 32-bit Masters when operating as a Memory Slave device. All data transfers to the local bus are 32-bit transfers. The Bus Master DMA feature of the Eddy module will always perform a 32-bit transfer and therefore should always be programmed to transfer data to or from system memory (i.e. 32-bit Slave Memory). The following paragraphs describe the address decoding performed by the Eddy Module.

Address Decode

The Address Decode sectlon is responsible for the control of read and write operations to the POS registers, the memory mapped I/O registers and the memory mapped control / status registers. The Eddy module will respond to one 32K byte address range. A map of this address range is shown in Figure 3.2. The address range is divided into two 16K sub-ranges. The first 16K range is reserved for the Geometry Subsystem and the second 16K rang8 is reserved for the Eddy module.

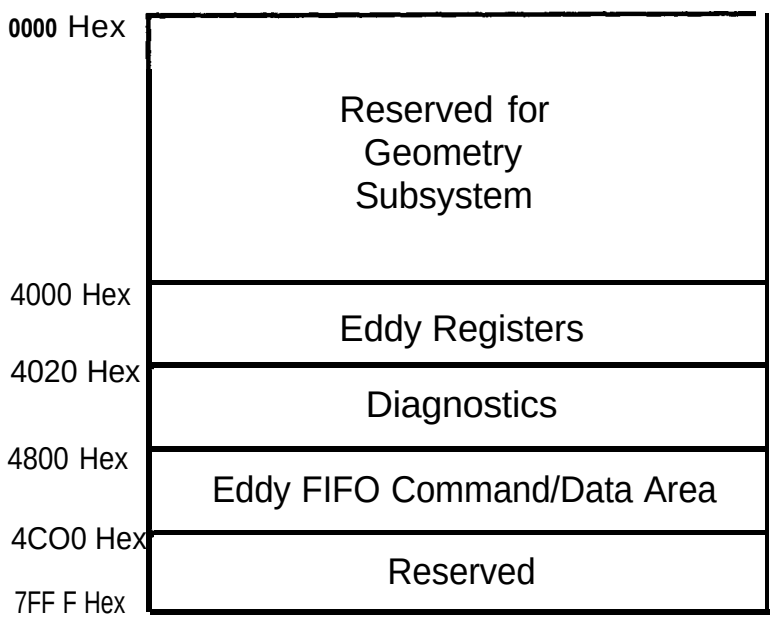


Figure 3.2 MGR Memory Map

For memory cycles within the first 16K address range the lower 14 bits of the MCA address bus and the 32 bits of the MCA data bus are passed to the Local bus as single cycle operations. The HQ1 chip in the Geometry Subsystem decodes these 14 address bits and performs the desired operation. The decoding and use of these 14 address bits are described in the Geometry Subsystem chapter.

The Eddy module uses the second 16K range to store it's control / status registers and a Diagnostics Communications area. Included in this memory range is a Local FIFO area to allow host software to perform ***store*** multiple instructions to the Geometry Subsystem without the Local bus being incremented.

For addresses in the second 16K range the lower four bits of the MCA address bus are used to determine the source or destination register. Data is then latched into the destination register for write operations or put on the MCA bus for read operations. These operations are used to initiate Bus Master DMA transfers, enable and reset interrupts and for diagnostics.

Programmable Option Select Registers

The Programmable Option Select (POS) registers contain information to identify and configure ***the*** adapter thus eliminating the need for hardware jumpers. There are eight directly addressable 8 bit POS registers, as shown in Figure 3.3. Four sub-data registers are accessible through the use of a sub-addressing facility. One of the POS registers indicates channel check status information which is only valid is after an error has caused the Eddy module to Channel Check the host system.

POS Register	Reset Value	Description
0	E6 Hex	Card ID Low Byte
1	8E Hex	Card ID High Byte
2	00 Hex	CEN, INT_LVL, & Config
3	00 Hex	Vital Product Data
4	X3 Hex	Function Enables 8 Config
5	00 Hex	Channel Check Register
6	00 Hex	VPD PROM Address / Status
7	00 Hex	VPD Card Address

Figure 3.3 POS Register Map

The POS registers are defined in the registers section of this chapter and their use is described in the basic operations and programming considerations sections of this chapter.

Control / Status Registers

The Control / Status section of the Eddy module contains the seven **32-bit** registers as shown in Figure 3.4. Some of the 32 bit registers are accessible as multiple 8 or 16 bit registers. The byte ordering of the host machine will affect the address required for accessing the registers when

accessing 8 or 16 bit portions of the 32 bit registers. These registers are defined *in* the registers section of this chapter.

The Eddy module contains two DMA channels. Each channel will arbitrate for the Micro Channel on the same arbitration level. These channels will operate in a serial manner so that a transfer on one channel will complete before a transfer on the other channel can begin. If an error occurs during a DMA transfer, that transfer will be terminated. The other channel will not be able to begin a transfer until the one that was terminated is allowed to complete.

REG	RT/2 Address	PS/2 Address	Width	Oper	Description
0	4000 Hex	4000 Hex	32 bits	R/W	DMA 0 Source / Destination Address
1	4004 Hex	4004 Hex	32 bits	R/W	DMA 0 Length Count
2	400B Hex	4008 Hex	8 bits	R/W	DMA 0 Control Register
	400A Hex	4009 Hex	8 bits	R/W	Interrupt Mask Register
	4009 Hex	400A Hex	8 bits	R/W	Interrupt Status Register
	4008 Hex	400B Hex	8 bits	R/W	Reset Register
3	400E Hex	400C Hex	16 bits	R/W	Command FIFO Address
	400C Hex	400E Hex	16 bits	R/W	Diagnostics Control Register
4	4010 Hex	4010 Hex	32 bits	R/W	DMA 1 Source / Destination Address
5	4014 Hex	4014 Hex	32 bits	R/W	DMA 1 Length Count
6	4018 Hex	4018 Hex	8 bits	R/W	DMA 1 Control Register

Figure 3.4 Control / Status Register Map

The following paragraphs describe the remaining components of the Host Interface Subsystem.

VPD PROM

Only the identification and location of the adapter can be determined from the POS registers. The **Vital** Product Data (VPD) PROM provides additional information about the adapter. It is accessible through the use of the POS Subaddressing Extension which is described later in the section on the POS registers.

The **MGR** adapter has a VPD PROM on each of the five cards which are used to form the base and enhanced configurations. Each VPD PROM is 256 bytes and contains the following data:

- VPD Header
  - VPD ASCII characters
  - VPD total length
  - CRC on data within the VPD PROM

- VPD data fields
  - Card Description
  - Engineering Change Level
  - Part Number
  - Manufacturer'
  - Field Replaceable Unit Number
  - Next VPD Address
  - Loadable Microcode Level
  - Serial Number - unique for each card

The VPD is read by placing the card number in POS 7, placing the sub-address of the data in POS 6, and reading POS 3. These registers are defined later in the registers section. The format of the VPD PROMs is shown In Figure 3.5.

POS 7	POS 6	Description of VPD read via POS 3
0	0	NOT ACCESSIBLE via POS 3
0	1	VPD in ASCII
0	4	Total Length of VPD 2 bytes
0	6	CRC 2 bytes
0	8	. (Delimiter)
0	9	Keyword 1 (2 bytes)
0	B	Keyword 1 Length (inclusive)
0	C	Keyword 1 Data
0	N	.
0	N + 1	Keyword 2 (2 bytes)
0	N + 3	Keyword 2 Length (inclusive)
0	N + 4	Keyword 2 Data

Figure 3.5 Vital Product Data PROM Map

As shown in Figure 3.5, the VPD data format consist of an . delimiter followed by a 2 byte keyword value. This is followed by a data length byte and the data. The number of delimiter, keywords, length bytes and data bytes is dependent on the amount of information to be placed in the VPD PROM. The programming considerations for accessing the VPD PROM are described later in this chapter.

Multi-Phase Clocks

The Multi-Phase Clocks section of the Host Interface Subsystem provides the necessary clock signals to the Eddy module. It also provides the various 10 MHz clocks to the Local Bus with the necessary phase relationships.

MCA Bus Drivers

The MCA Bus drivers provide the necessary drive levels as required by the Micro Channel specification. The Eddy module has the ability to control the direction and tri-state nature of these devices.

Registers

The Eddy module contains the Programmable Option Select (POS) registers and the control / status registers. The POS registers are used to configure the MGR adapter. The control / status registers are used to control the DMA channels and to control interrupts. They also provide status information about the MGR adapter.

Programmable Option *Select* (POS) Registers

The POS registers allow the MGR adapter to be identified and configured. The POS registers conform to the MCA specification and are programmed using the IBM reference disk using the adapter description file provided with the MGR adapter. The reference disk places the configuration data in the battery backed CMOS RAM which is then accessed during the Power On Self Test (POST) at system power-on or reset. The host application then reads the POS registers to determine the configuration data placed in the registers by the POST program. The host software can also directly access the registers to change the configuration bits dynamically. The host software also reads the POS registers to determine the type of hardware error which has occurred after a Channel Check error has occurred. The POS registers are also used to read the contents of the VPD PROMs. The POS registers are accessed by performing a SETUP configuration cycle on the MCA bus. The programming considerations section of this chapter will describe the method for performing a SETUP cycle under the DOS, OS/2 and UNIX operating systems. The following pages define the Programmable Option Select registers.

MGR Identification Register (POS 0 and POS 1)

The MGR adapter is uniquely identified by the value in POS registers 0 and 1. These registers are used at Power-on or Reset to identify the MGR adapter so that the other POS registers can be configured using the data from the CMOS RAM. The value in these registers is hard wired to a value of 8EE6h and are read only. Each register is active when it's address is placed on the MCA address bus and the SETUP signal is 0. The two registers are shown in Figure 3.6.

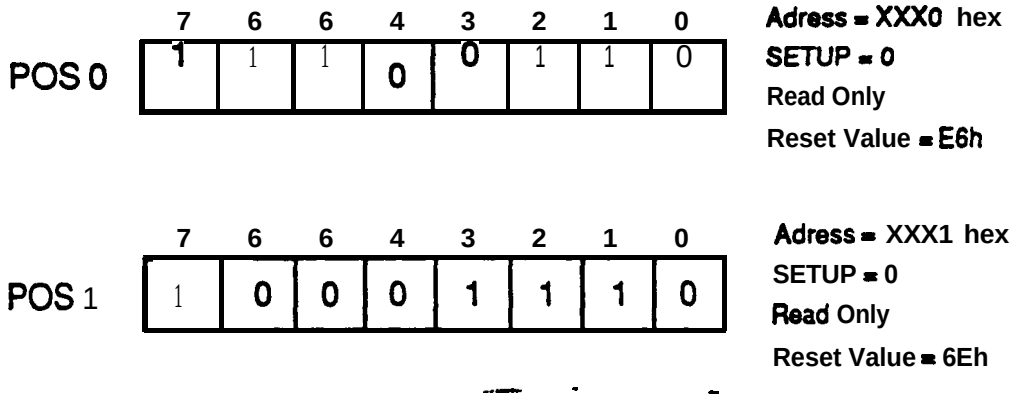


Figure 3.6 MGR Card Identification Register

POS 0

Bits 7 - 0 : ID low byte. These bits are hard wired to **E6h.**

POS 1

. . Bits 7 - 0 : **ID high byte.** These bit6 are hard wired to 8Eh.

Card Enable and Device Configuration Register (POS 2)

The card enable and device configuration register is the POS 2 register and is shown in Figure 3.7. This register contains device configuration bits as well as the Card Enable bit.

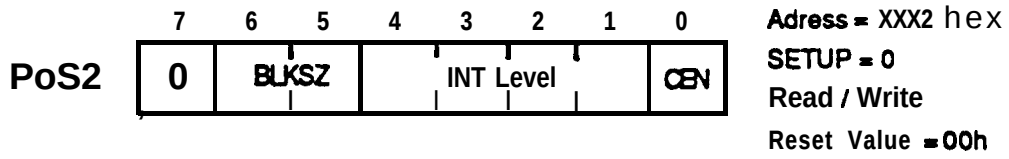


Figure 3.7 Card Enable and Device Configuration Register

These bits are defined as:

Bit 7 : **Reserved.**

Bit 6-5 : **Block Transfer Size (BLKSZ).** These bits determine the maximum block transfer size that will be attempted by the Eddy module during a Bus Master DMA. These bits are encoded to select one of three block sizes or allow the adapter to continue transfering data until preempted or tne DMA is complete. These bits are used to tune the performance of a particular system. The bits have the following definitions:

00 - No Restriction (recommended for RT/2 when Data Streaming is disabled)
01 - 16 Transfers (recommended for RT/2 when Data Streaming is enabled)
10 - 32 Transfers
11 - 64 Transfers (recommended for PS/2 systems)

Bit 4-1 : **Interrupt Level (INT Level).** These bits determine the interrupt level which will be used by the MGR adapter to signal the host. Only eight of the possible sixteen values are valid. Setting this field to an invalid value will disable all interrupts from the MGR adapter. The bits have the following definitions:

0010 - Interrupt Level 2
0011 - Interrupt Level 3
0100 - Interrupt Level 4
0101 - Interrupt Level 5
0110 - Interrupt Level 6
1001 - Interrupt Level 9
1010 - Interrupt Level 10
1011 - Interrupt Level 11
1100 - Interrupt Level 12
All other combinations Invalid - Interrupts will be disabled

Bit 0 : **Card Enable (CEN).** This bit is used to enable or disable the MGR adapter. When this bit is inactive (Low), the MGR adapter will only respond to the Setup Registers.

0 - Card Disabled
1 - Card Enabled

Vital Product Data (VPD) Register (POS 3)

The Vital Product Data **Register is** the POS 3 register and is shown in Figure 3.8. The **VPD** Register serves as a data port for reading the **Vital** Product Data **PROMs.** The sub-address of the data byte in the selected VPD PROM to be read is formed from the various bit settings in POS registers 4-7 as explained later.

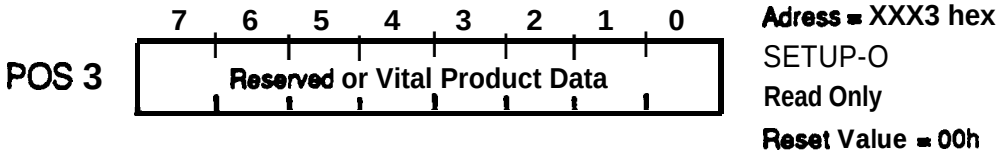


Figure 3.8 Vital Product Data Register

-These bits are defined as:

Sub-Address = **00h**

Bits 7 - 0 : **Undefined.** Sub-address 0 is undefined in the **VPD** PROMs.

Sub-Address = **01h - FFh**

Bits 7 - 0 : **Vital Product Data.** This byte will contain the VPD data read from the selected **VPD PROM.**

Functional Enables and Configuration Register (POS 4)

The Functional Enables and Configuration Register is the POS 4 Register and is used to access the four sub-address registers. The desired sub-address register address is placed in the POS 6 register and the POS 4 register is then read or written. The sub-address registers contain information on which functions of the **MGR adapter** are to be enabled and the Memory Address where the **MGR** adapter is located. The bit definitions for the four sub-address **registers** are **shown** in Figure 3.9

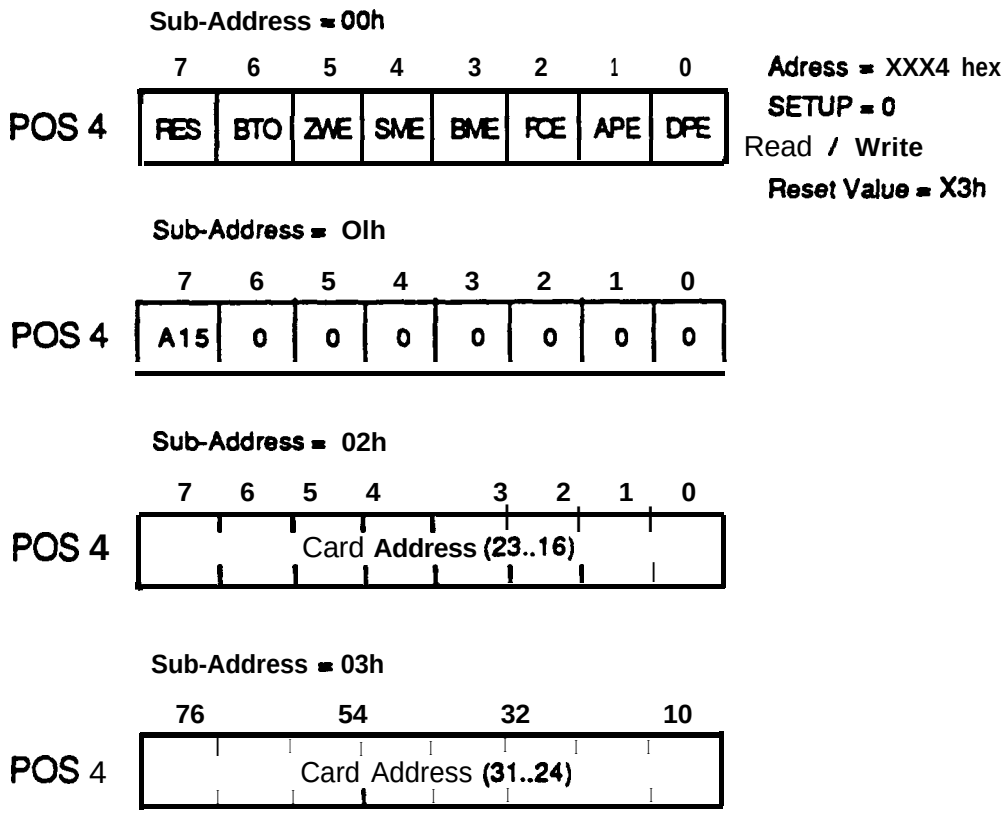


Figure 3.9 Function Enables 8 Card Memory Address Register

These bits are defined as:

Sub-Address = 00h

Bit 7 : Reserved.

Bit 6 : **Byte Ordering (BTO).** This bit is used to inform the adapter of it's environment. If this bit is **1,** the byte ordering conforms to the Big **Endian** format with the bytes arranged MSB to LSB in the data word. If the bit is 0, the byte ordering conforms to the Little **Endian** format with the bytes arranged LSB to MSB in the data word. The **PS/2** Architecture conforms to the Little **Endian** format.

0 - Big **Endian** format **(RT/2)**
Byte 0 = Most Significant Byte

Byte 3 = Leas! Significant Byte

1 - Little **Endian** format **(PS/2)**
Byte 0 = Leas! Significant Byte
Byte 3 = Most Significant Byte

Bit 5 : **Zero Wait State Enable (ZWE).** This bit is used to perform Zero Wait State Write cycles to the memory mapped **I/O** registers. When this bit is se! to 0 then 1 wait state is added to the normal cycle. When this bit is se! to 1 then no wait state is added to the cycle.

0 - One Wait State Write Cycle
1 - Zero Wait State Write Cycle

Bit 4 : Data Stre[redacted]e Enable (SME). This bit is used to enable/disable the use of Data Streaming by the MGR adapter. If **this** bit is 0, Data Streaming a! 10 Mhz will be used during Bus Master DMA on the Micro Channel upon request. Setting this bit to 1 will disable the Data Streaming function.

0 - Data Streaming Enabled
1 - Data Streaming Disabled

Bit 3 : Bus Master DMA Enable (BME). This bit is used to enable/disable the use of Bus Master DMA on the Micro Channel. If this bit is 1, the Eddy module will perform a Bus Master DMA on the Micro Channel upon request. Setting this bit to 0 will disable the use of Bus Master DMA thus making the MGR adapter a memory slave.

0 - Bus Master DMA Disabled
1 - Bus Master DMA Enabled

Bit **2 : Enable FIFO Overflow Channel Check (FOE).** This bit is used to enable/disable Channel Checking on a Command FIFO **Overflow.** If **this** bit is se! to 0, a Channel Check will no! occur if the Command FIFO overflows. Setting this bit to 1 will enable a synchronous Channel Check to occur if the Command FIFO is overflowed by the host software. This function should only be enabled in systems which support synchronous error recovery.

0 - Channel Checking on FIFO Overflow Disabled
1 - Channel Checking on **FIFO** Overflow Enabled

Bit 1 : **Address Parity Enable (APE).** This bit is used to enable/disable the checking and generation of Address Parity. If this bit is se! to 0, Address **Parity** checking and generation is disabled. If this bit is se! to 1, Address **Parity** is checked if the Enable Signal on the Micro Channel is active and is generated for all Bus Master DMA transfers. This function should only be enabled for host systems which support Address Parity and synchronous error recovery.

0 - Address Parity **Gen/Check** Disabled
1 - Address Parity **Gen/Check** Enabled

Bit 0 : **Data Parity Enable (DPE).** This bit is used **to** enable/disable the checking and generation of Data Parity. If this bit is se! to 0, Data Parity checking and generation is disabled. If this bit is se! to **1,** Data **Parity** is generated on Write Cycles and checked on Read Cycles. If Bad **Parity** is detected during a Read Cycle, a synchronous Channel Check is generated.

It is recommended that this bit be set to 0 unless Data Parity is supported by the host system.

0 - Data Parity Gen/Check Disabled
1 - Data Parity Gen/Check Enabled

Sub-Address = 01h

Bit 7 : Address 15 (A15). This bit is used in conjunction with sub-registers two and three to decode a 32K byte address range for the MGR's memory mapped I/O registers.

Bits 6-O : Reserved.

Sub-Address = 02h

Bits 7-O : Address (23-16). This register is used in conjunction with sub-registers one and three to decode a 32K byte address range for the MGR's memory mapped I/O registers.

Sub-Address = 03h

Bits 7-O : Address 131-24). This register is used in conjunction with sub-registers one and two to decode a 32K byte address range for the MGR's memory mapped I/O registers.

Channel Check and Arbitration Level Register (POS 5)

The Channel Check and Arbitration Level Register is the POS **5** register and is shown in Figure 3.10. This register is used to report hardware errors to the host system. This register also determines the Arbitration Level to be used for Bus Master DMA and if Fairness will be supported for DMA.

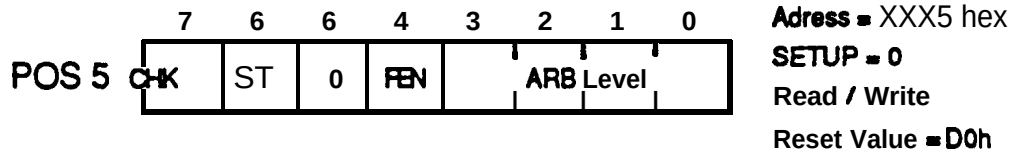


Figure 3.10 Channel Check and Arbitration Level Register

These bits are defined as:

Bit 7 : **Channel Check Bit (CHK).** When this bit is set to 0, the **MGR** adapter has signaled the host of a hardware error. This bit is set to 1 on any write to this register. This bit is read only.

0 - Channel check active, a hardware error has occurred
1 - No errors have occurred

Bit 6 : **Channel Check Status Bit (ST).** This bit indicates the presence of Status information in the POS 6 register about the pending Channel Check. This bit is set to 1 on any write to this register. This bit also controls access to the sub-address registers. This bit is read only.

0 - POS 6 contains status information
1 - POS 6 contains a sub-address

Bit 5 : **Reserved.**

Bit 4 : **Fairness Enable (FEN).** If this bit is 0, the Bus Master DMA section of the **MGR** adapter will conform to the Fairness Algorithm for the Micro Channel. That is, if a Bus Master Transfer on behalf of the **MGR** adapter is preempted prior to completion, the MGR will not arbitrate for the bus until all preempting devices on the channel have been serviced. If this bit is **1,** no restrictions are placed on when the MGR adapter can arbitrate for the Micro Channel.

0 - Arbitration Fairness Disabled
1 - Arbitration Fairness Enabled

Bits 3-O : **Arbitration Level (ARB Level).** These bits define the Arbitration Level the MGR adapter will use for Bus Master DMA.

Channel Check Status / Sub-Address Register (POS 6)

The Channel Check Status / Sub-Address Register is the POS 6 register and is shown in Figure 3.11. This register has two functions which are controlled by the status bit (bit 6) of the POS 5 register. If the status bit is 1, this register is defined as a sub-address register. If the status bit is a 0, this register contains status information about the pending Channel Check.

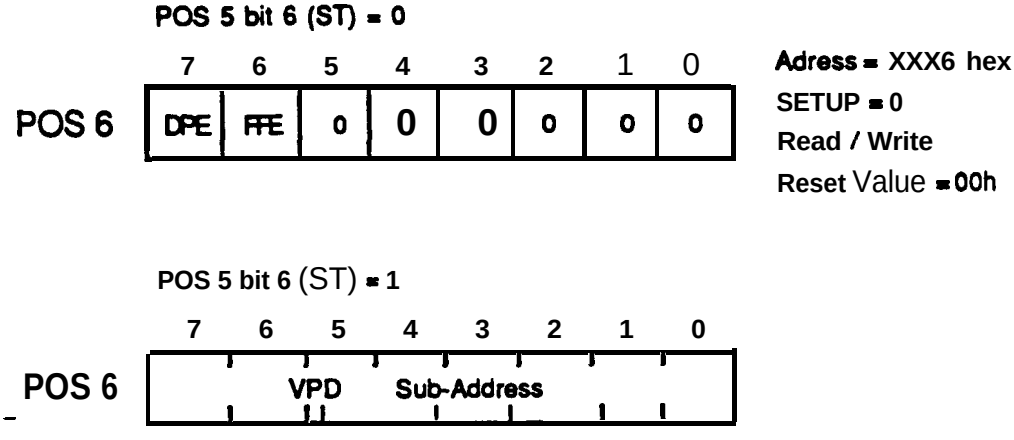


Figure 3.11 Channel Check / Sub-Address Register

These bits are defined as:

Status Bit = 0

Bit 7 : **Data Parity Error (DPE).** This bit indicates a parity error has occurred on the Micro Channel. It is reset to 0 when POS register 6 is written.

0 - No Error Detected
1 - Error Detected on Micro Channel Data Bus

Bit 6 : **FIFO Overflow Error (FFE).** This bit indicates that the Command FIFO has encountered an Overrun condition. It is reset to 0 when POS register 6 is written.

0 - No Error Detected
1 - FIFO Overflowed

Bits 5-O : **Reserved.**

Status Bit = 1

Bits 7-O : **VPD PROM sub-address** When Channel Check status is not pending, these bits are used to address the VPD PROM.

Vital Product Data Card Address Register (POS 7)

The Vital Product Data Card Address Register is the POS 7 register and is used to select which card will be enabled for VPD PROM reads. The MGR adapter has two configurations which are composed of either three or four cards and each card contains a VPD PROM. This register is shown in Figure 3.12.

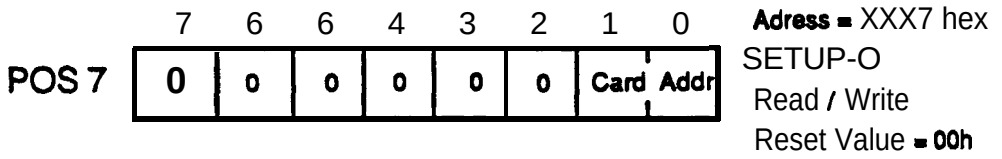


Figure 3.12 Vital Product Data Card Address Register

These bits are defined as:

Bits 7-2 : Reserved.

Bits 1-O : VPD Card Address. These bits are used to decode which card or VPD PROM is to be accessed through the POS 3 register.

00 - MGE2 card
01 - MRV2 card
10 - MDE1 card or MEV1 card depending on Configuration
11 - MZB1 card

Control / Status Registers

The Control / Status registers include the source/destination address register, the length count register and the control register for both DMA **channels 0** and 1. There are also an interrupt mask register, an interrupt status register, a reset register, a command FIFO register and a diagnostics register. These registers allow the host software to set up two separate DMA transfers on the two DMA channels. When the first channel completes it's transfer the second channel can begin it's transfer. This allows the host software to set up a DMA transfer while one is in progress.

These registers also allow the host software to control the interrupt masks which selectively enable or disable the various interrupts. The POS registers are used to set the Interrupt Request (IRQ) level used by the MGR adapter. The various interrupts generated on the MGR adapter are multiplexed onto a single IRQ line to the host. Once an interrupt has been generated the host software can read the interrupt status register to determine which interrupt has occurred.

There are also a reset register, a local command address register and a diagnostics register. These registers as well as the DMA and interrupt registers are defined on the following pages.

Channel 0 DMA Source/Destination Address Register

The contents of this register points to the next word to be transferred when the Eddy module acquires ownership of the Micro Channel for a DMA transfer. This register must be initialized to the first word of a contiguous block of *memory* before the Eddy module is given permission to start the transfer. Once the DMA has been started, the register can not be written until the DMA transfer has completed. Writing to this register during a DMA transfer is pending or is in progress will produce unpredictable results. All DMA transfers must start on a word boundary, therefore the lower two bits of this register are read **only** and will always return zero. This register is shown in Figure 3.13.

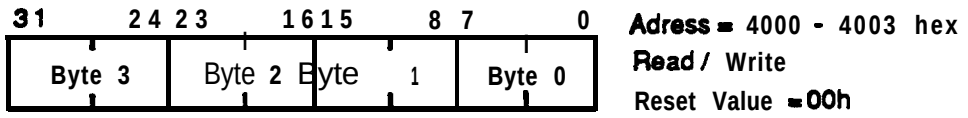


Figure 3.13 Channel 0 DMA Source/Destination Address Register

These bits are defined as:

Bits 31-O : Source or Destination Address. -This register contains the source or destination address for the DMA transfer. This register is Read Only if a DMA transfer is in progress or is pending.

Channel 0 DMA Transfer Length Count Register

This register contains the number of words contained in the contiguous block of memory to be transferred via DMA. This register, like the address register, must be initialized before permission is given to the Eddy module to start the transfer. Setting the Most Significant Bit to One (1) will cause the present byte ordering to be overridden. Writing to this register while a DMA is pending or is in progress will produce unpredictable results. This register is shown in Figure 3.14.

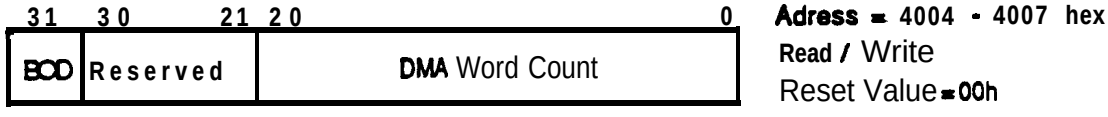


Figure 3.14 Channel 0 DMA Transfer Length Count Register

These bits are defined as:

Bit 31 : **Byte Ordering Override (BOD).** This bit when set causes the selected byte ordering to be reversed during this DMA transfer.

0 - Normal Byte drdering
1 - Switch Ordering for this DMA

Bit 30 - 21 : **Reserved.**

Bits 20-O : **DMA Word Count.** This field contains the number of words to transfer.

Channel 0 DMA Control/Status Register

This register is used to initiate a DMA transfer and to check on it's progress. Information about the status of the **MGR's** logic is also provided in this register for software polling. This register is shown in Figure 3.15.

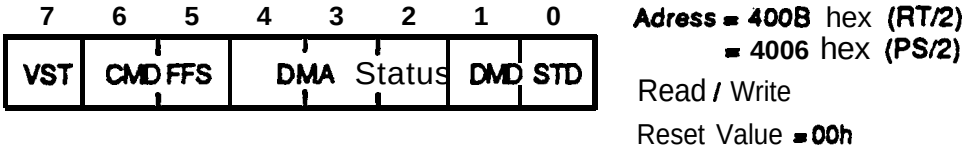


Figure 3.15 Channel 0 DMA Control / Status Register

These bits are defined as:

Bit 7 : **Vertical Status (VST).** This bit is status of the monitor's vertical retrace Interval. It will be set high when the vertical retrace is started and will be reset at the end of the vertical retrace. This vertical retrace time is approximately 505 **usec.**

0 - Updating Screen Information
1- Vertical Retrace Time

Bit 6 - 5 : **Command FIFO Status (CMD FFS).** These bits indicate the status of the External command FIFO. They are encoded to indicate that the FIFO is less than half full, more than half full (i.e. **>=256** words), and completely full. These bits are intended to be used for pacing of commands sent by the host.

00 - FIFO < Half Full
01 - FIFO >= Half Full
10 - FIFO Full
11 = Reserved

Bits 4 - 2 : **DMA Status.** These bits are encoded to indicate the status of the DMA transfer. They are also used to report any errors encountered during a transfer.

000 - DMA Complete **with** no 'errors
001 - DMA Pending
010 - Reserved
011 - DMA in Progress
100 - DMA Cancelled by host system
101 - DMA Terminated by Channel Check
110 - DMA Terminated, Slave Response Error
111 - DMA Terminated, Data Parity Error

Bit 1 : **DMA D i r -.** This bit determines the direction of the impending DMA transfer. Setting this bit low (0) will cause data to be read from system memory and written to the **MGR's** microcode Data RAM or to the Frame Buffer bitplanes. If this bit is set high **(1),** data will be read from the microcode Data **RAM** or the Frame Buffer bitplanes and written into the host memory.

0 - From Host memory to MGR adapter
1 - From MGR adapter to Host memory

Bit 0 : Start DMA (STD). Setting this bit to a one will cause the Eddy module to arbitrate for MCA bus ownership. Once ownership is obtained then the DMA transfer will be started. When the DMA has been completed or has been terminated because of an error this bit will be reset to zero. Writing a zero to this bit while a DMA is pending or in progress will cause the DMA to be terminated with Status Code 100 in bits 4-2.

0 - DMA Not Started
1 - DMA Started

Interrupt Mask Register

This register allows the host software to individually select which types of interrupts it wishes to be notified about. This register is shown in Figure 3.16.

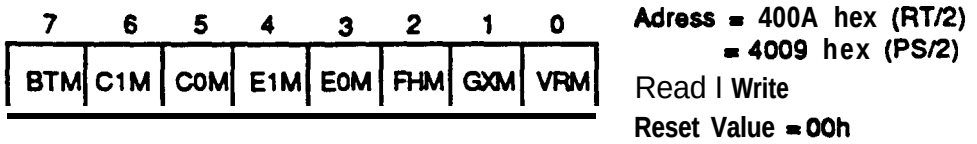


Figure 3.16 Interrupt Mask Register

These bits are defined as:

Bit 7 : **Bus Time Out Mask (BTM).** Setting this bit to a one will enable an interrupt to be generated when a timeout condition is encountered on the Local Bus or on the Micro Channel during a Bus Master DMA cycle. The least significant bits of the Reset register Identifies which bus caused the timeout, A zero in this bit will not allow an interrupt to be generated when this condition occurs.

0 - Interrupt Disabled
1 - Interrupt Enabled

Bit 6 : **Channel 1 DMA Complete Mask (C1M).** Setting this bit to a one will enable an interrupt to be generated on the completion of a DMA transfer. A zero in this bit will not allow an interrupt to be generated when this condition occurs.

0 - Interrupt Disabled
1 - Interrupt Enabled

Bit 5 : **Channel 0 DMA Complete Mask (C0M).** Setting this bit to a one will enable an interrupt to be generated on the completion of a DMA transfer. A zero in this bit will not allow an interrupt to be generated when this condition occurs.

0 - Interrupt Disabled
1 - Interrupt Enabled

Bit 4 : Channel 1 DMA Error Mask (E1M). Setting this bit to a one will enable an interrupt to be generated when an Error is detected during a DMA transfer. A zero in this bit will not allow an interrupt to be generated when this condition occurs.

0 - Interrupt Disabled
1 - Interrupt Enabled

Bit 3 : **Channel 0 DMA Error Mask (E0M).** Setting this bit to a one will enable an interrupt to be generated when an Error is detected during a DMA transfer. A zero in this bit will not allow an interrupt to be generated when this condition occurs.

0 - Interrupt Disabled
1 - Interrupt Enabled

Bit 2 : Comman(FHM)) Half Full Mask . Setting this bit to a one will enable an interrupt to be generated when the external Command FIFO reaches the Half Full Mark. This interrupt can be used for software command pacing. A zero in this bit will not allow an interrupt to be generated when this condition occurs.

0 - Interrupt Disabled
1 - Interrupt Enabled

Bit 1 : Graphics Interrupt Mask (GXM). Setting this bit to a one will enable interrupts generated by the Geometry Subsystem to be seen by the host. A zero in this bit will not allow an interrupt to be generated when this condition occurs.

0 - Interrupt Disabled
1 - Interrupt Enabled

Bit 0 : Vertical Retrace Interrupt Mask (VRM). Setting this bit to a one will enable an interrupt to be generated approximately 505 usec before the actual vertical retrace begins. A zero in this bit will not allow an interrupt to be generated when this condition occurs.

0 - interrupt Disabled
1 - Interrupt Enabled

Interrupt Status Register

This register is used to report which condition caused the system interrupt to occur. The system interrupt is cleared when this register contains zero. Each interrupt is cleared individually by writing a one to the corresponding bit location. This register is shown in Figure 3.17.

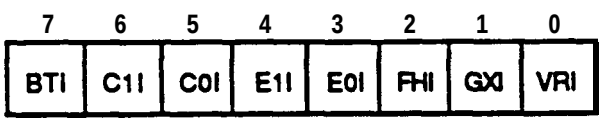


Figure 3.17 interrupt Status Register

These bits are defined as:

Bit 7 : **Bus Ti[redacted]terrupt (BTI).** If this bit is set to a one, the Local Bus or the Micro Channel has timed out. The least significant two bits of the Reset register identifies which bus caused the timeout. A zero in this bit indicates that this condition has not occurred. Writing a one to this bit will reset it to a zero.

0 - Interrupt Not Pending
1 - interrupt Pending

Bit 6 : **Channel 1 DMA Complete Interrupt (C1I).** If this bit is set to a one, DMA Channel One has completed it's DMA transfer with no errors. A zero in this bit indicates that this condition has not occurred. Writing a one to this bit will reset it to a zero.

0 - interrupt Not Pending
1 - Interrupt Pending

Bit 5: **Channel 0 DMA Complete Interrupt (C0I).** If this bit is set to a one, DMA Channel Zero has completed it's DMA transfer with no errors. A zero in this bit indicates that this condition has not occurred. Writing a one to this bit will reset it to a zero.

0 - Interrupt Not Pending
1 - Interrupt Pending

Bit 4 : **Channel 1 DMA Error Interrupt (E1I).** If this bit is set to a one, DMA Channel One has terminated due to an error condition. The exact cause of this interrupt can be determined by reading the DMA Status in the DMA 1 Control / Status register. A zero in this bit indicates that this condition has not occurred. Writing a one to this bit will reset it to a zero.

0 - Interrupt Not Pending
1 - Interrupt Pending

Bit 3 : **Channel 0 DMA Error Interrupt (E0I).** If this bit is set to a one, DMA Channel Zero has terminated due to an error condition. The exact cause of this interrupt can be determined by reading the DMA Status in the DMA 0 Control / Status register. A zero in

this bit indicates that this condition has not occurred. Writing a one to this bit will **reset** it to a zero.

0 - Interrupt Not Pending
1 - Interrupt Pending

Bit 2 : **Comm(FHI)FIFO Half Full Interrupt** . If this bit is set to a one, the external Command **FIFO** has reached the Half Full Mark. This interrupt can be used for software command pacing. A **zero in this bit indiites that** the FIFO is less than half full. Writing a one to this bit will reset it to a zero.

0 - Interrupt Not Pending
1 - Interrupt Pending

Bit 1 : **Graphics Interrupt Interrupt (GXI).** If this bit is **set** to a one, **the** Geometry Subsystem has generated a graphics interrupt to the host. A zero In this **bit** indicates that a graphics interrupt has not **been** generated. Writing a one to this bit will reset it to a zero.

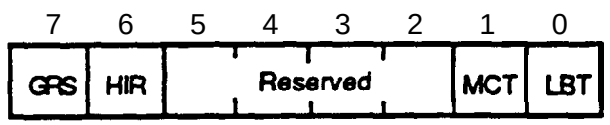
0 - Interrupt Not Pending
1 - Interrupt Pending

Bit 0 : **Vertical Retrace Interrupt (VRI). if this bii** is **set** to a one, **the vertical retrace time is approaching.** The actual start of the vertical retrace can be determined by reading the **Vertical Retrace** Status bit in the DMA 0 Control **/** Status register. A **zero** in this bit **indicates** that this condition has not occurred. Writing a one to this bit will reset it to a zero.

0 - Interrupt Not Pending
1 - Interrupt Pending

Reset Register

This register is used to reset the Graphics Subsystem and the DMA sections of the Eddy module. This register is shown in Figure 3.18.



Adress = 4008 hex (RT/2)
= 4008 hex (PS/2)
Read / Write
Reset Value = 43h

Figure 3.18 Reset Register

These bits are defined as:

Bit 7 : Graphics Subsystem Reset (GRS). Setting this bit to a zero will cause the Graphics Subsystems to be reset. This bit must remain zero for 10 msec. Setting this bit to one puts the Graphics Subsystems in an operational state.

0 = Graphics Subsystems in Reset
1 = Graphics Subsystems operational

Bit 6 : Host Interface Reset (HIR). Setting this bit to a zero will cause the DMA sections of the Eddy module to be reset. The POS setup registers will not be affected by this bit. Setting this bit to a one puts the DMA sections into an operational state.

0 = Host Interface Subsystem in Reset
1 = Host Interface Subsystem operational

Bit 5 - 2 : Reserved.

Bit 1 : Micro Channel Timeout (MCT). If this bit is set to a one, the Slave memory on the Micro Channel held Channel Ready too bng.

0 = Micro Channel did not Timeout
1 = Micro Channel Timed Out

Bit 0 : Local Rus Timeout (LBT). If this bit is set to a one, the Local Bus is locked up.

0 = Local Bus did not Timeout
1 = Local Bus Timed Out

Local Command FIFO Address

This register contains the Address to be supplied to the Local Bus when an access is made to the FIFO data transfer area. This is normally used to point to the Command FIFO to enable Command Data to be transferred via a Store Multiple instruction. This register is shown in Figure 3.19.

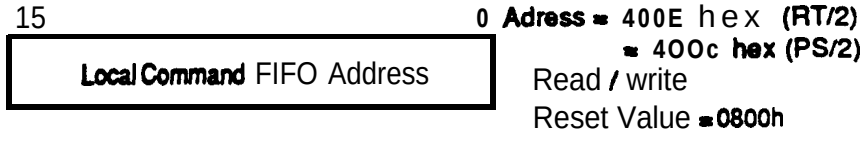


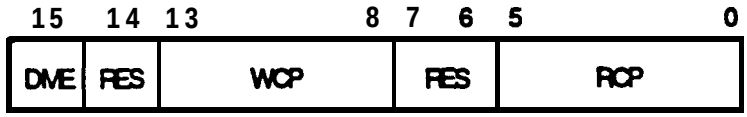
Figure 3.19 Local Command FIFO Address Register

These bits are defined as:

Bit 15 - 0 : Local Command FIFO Address. This register contains the address of the graphios subsystem's component that will receive the data written into the local FIFO data transfer area. This address is normally the external command FIFO address which is **800h.**

Diagnostic Control Register

This register is used to put the MGR adapter into a diagnostics mode, select the type of diagnostics to perform and to start the diagnostics. This register is shown in **Figure** 3.20.



Adress = 400C hex (RT/2)
= 400E hex (PS/2)
Read I Write
Reset Value = 1020h

Figure 3.20 Diagnostics Control Register

These bits are defined as:

Bit 15 : **Diagnostics Mode Enable (DME).** Setting this bit to **a one will** put the MGR adapter into a diagnostics mode of operation. Setting this bii to zero puts the MGR adapter in an operational state.

0 - Normal mode
1 - Diagnostics mode

Bit 14 : **Reserved (RES).**

Bit 13 - 8 : **Write Commit Pointer (WCP).** This field determines the point in the internal Master FIFO which the Eddy module will begin requesting the Micro Channel once the Internal FIFO has been filled.

Bit 7 - 6 : Reserved (RES).

Bit 5 - 0 : **Read Commit Pointer (RCP).** This field determines the point in the internal Master FIFO which the Eddy module will begin requesting the Micro Channel once the Internal **FIFO** has been emptied.

Channel 1 DMA Source/Destination Address Register

The contents of this register points to the next word to be transferred when the Eddy module acquires ownership of the Micro Channel for a DMA transfer. This register must be initialized to the first word of a contiguous block of memory before the Eddy module is given permission to start the transfer. Once the DMA has been started, the register can not be written until the DMA transfer has completed. Writing to this register during a DMA transfer is pending or is in progress will produce unpredictable results. All DMA transfers must start on a word boundary, therefore the lower two bits of this register are read only and will always return zero. This register is shown in Figure 3.21.

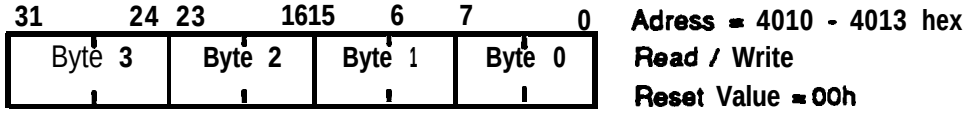


Figure 3.21 Channel 1 DMA Source/Destination Address Register

These bits are defined as:

Bits 31-O : Source or Destination Address. This register contains the source or destination address for the DMA transfer. This register is Read Only if a DMA transfer is in progress or is pending.

Channel 1 DMA Transfer Length Count Register

This register contains the number of words contained in the contiguous block of memory to be transferred via DMA. This register, like the address register, must be initialized before permission is given *to the* Eddy *module* to start the transfer. Setting the Most Significant Bit to One (1) will cause the present byte ordering to be overridden. Writing to this register while a DMA is pending or is in progress will produce unpredictable results. This register is shown in Figure 3.22.

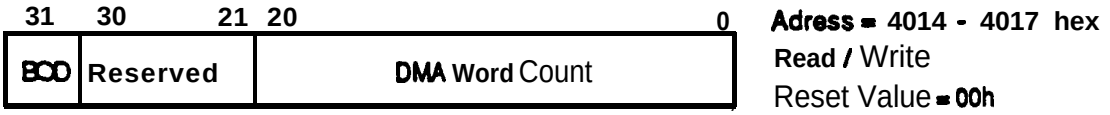


Figure 3.22 Channel 1 DMA Transfer Length Count Register

These bits are defined as:

Bit 31 : Byte Ordering Override (BOD). This bit when set causes the selected byte ordering to be reversed during this DMA transfer.

0 --Normal Byte Ordering
1 - Switch Ordering for this DMA

Bit 30 - 21 : Reserved.

Bits 20-O : DMA Word Count. This field contains the number of words to transfer.

Channel 1 DMA Control/Status Register

This register is used to initiate a DMA transfer and to check on it's progress. Information about the status of the MGR's logic is also provided in this register for software polling. This register is shown in Figure 3.23.

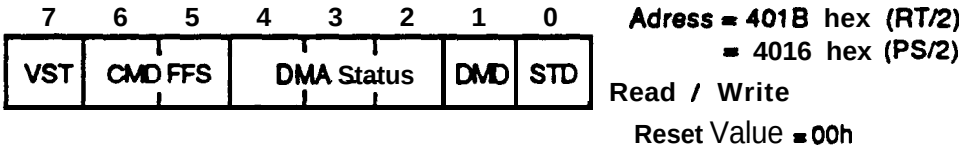


Figure 3.23 Channel 1 DMA Control / Status Register

These bits are defined as:

Bit 7 : Vertical Status (VST). This bit is status of the monitor's vertical retrace interval. It will be se? high when the vertical retrace is started and will be reset at the end of the vertical retrace. This vertical retrace time is approximately 505 usec.

0' - Updating Screen Information
1 - Vertical Retrace Time

Bit 6 - 5 : Command FIFO Status (CMD FFS). These bits indicate the status of the External command FIFO. They are encoded to indicate that the FIFO is less than half full, more than half full (i.e. >=256 words), and completely full. These bits are intended to be used for pacing of commands sent by the host.

00 - FIFO < Half Full
01 - FIFO >= Half Full
10 - FIFO Full
11 = Reserved

Bits 4 - 2 : DMA Status. These bits are encoded to indicate the status of the DMA transfer. They are also used to report any errors encountered during a transfer.

000 - DMA Complete with. no errors
001 - DMA Pending
010 - Reserved
011 - DMA in Progress
100 - DMA Cancelled by host system
101 - DMA Terminated by Channel Check
110 - DMA Terminated, Slave Response Error
111 - DMA Terminated, Data Parity Error

Bit 1 : DMA Direction (DMD). This bit determines the direction of the impending DMA transfer. Setting this bit low (0) will cause data to be read from system memory and written to the MGR's microcode Data RAM or to the Frame Buffer bitplanes. If this bit'is set high (1), data will be read from the microcode Data RAM or the Frame Buffer bitplanes and written into the host memory.

0 - From Host memory to MGR adapter
1- From MGR adapter to Host memory

Bit 0 : **Start DMA (STD).** Setting this bit to a one will cause the Eddy module to arbitrate for MCA bus ownership. Once ownership is obtained then the DMA transfer will be started. When the DMA has been completed or has been terminated because of an error this bit will be reset to zero. Writing a zero to this bit while a DMA is pending or in progress will cause the DMA to be terminated with Status Code 100 in bits 4-2.

0 - DMA Not Started
1 - DMA Started

Interrupts

The Host Interface Subsystem can generate an interrupt to the host system on the Interrupt Request (IRQ) Level selected in POS register 2. The Eddy module multiplexes eight possible interrupts from the adapter into the single IRQ line to the host system. The eight possible interrupts include:

- Bus Time Out Interrupt
- Channel 1 DMA Complete Interrupt
- Channel 0 DMA Complete Interrupt
- Channel 1 DMA Error Interrupt
- Channel 0 DMA Error Interrupt
- FIFO Half Full Interrupt
- Graphics Interrupt
- Vertical Retrace Interrupt

As described in the Registers section above, the Interrupt Mask Register is used to selectively enable or disable any of the eight interrupts. The Interrupt Status Register is used to determine which of the eight interrupts caused the host system to be interrupted. This register can also be polled to determine if any of the interrupt conditions has become active in an environment in which interrupts are not enabled.

The Bus Time Out Interrupt is generated if either the MCA bus or the Local Bus does not respond within an acceptable time period. This interrupt would tend to indicate a hardware failure has occurred. A possible corrective action might be to do a complete Reset of the MGR adapter. If this did not correct the problem then a hardware repair is in order.

The Channel 1 DMA Complete Interrupts are generated after a DMA operation has successfully completed on DMA channel 1. The Channel 0 DMA Complete Interrupts are generated after a DMA operation has successfully completed on DMA channel 0.

The Channel 1 DMA Error Interrupt is generated if an error occurs during a DMA transfer on Channel 1. The host software can check bits 4 - 2 in the Channel 1 DMA Control / Status Register to determine the type of error and then attempt the appropriate corrective action.

The Channel 0 DMA Error Interrupt is generated if an error occurs during a DMA transfer on Channel 0. The host software can check bits 4 - 2 in the Channel 0 DMA Control / Status Register to determine the type of error and then attempt the appropriate corrective action.

The FIFO Half Full Interrupt is generated by the FIFO hardware when the FIFO reaches the half full condition. This interrupt is provided to allow the host software to pace the writing of command tokens into the FIFO. This interrupt is described in greater detail in the Geometry Subsystem chapter.

The Graphics Interrupt is generated by the Geometry Engine under the control of the microcode and is used to signal the host software that the microcode has reached a condition that needs the host

software to take action. An example would be when the Feedback buffer is full and needs to be read by the host. This interrupt is described in greater detail in the Geometry Subsystem chapter.

The Vertical Retrace Interrupt is generated by the Display Subsystem hardware before a vertical retrace is about to begin. This allows the host software to transfer control to a vertical retrace interrupt service routine by the time the vertical retrace has actually begun. The host updates the Display Subsystem hardware during vertical retrace time to prevent any visible screen glitches from occurring. This interrupt is described in greater detail in the Display Subsystem chapter.

Basic Operations

The Host Interface Subsystem provides the MCA bus interface signals and registers required by the MCA specification. The Eddy module provides a bus translation from the MCA bus to the Local Bus, the Programmable Option Select (POS) registers and the Control / Status registers. It also provides a hardware reset capability and a limited diagnostics capability. The following paragraphs describe these capabilities.

MCA Bus to Local Bus Translation

From the host software perspective the bus translation is totally transparent when reads or writes are performed to one of the *other* three subsystems on the MGR adapter. The Eddy module provides the necessary signals required by both the MCA bus and the Local bus. The Eddy module also provides a special FIFO command/data area that can accept multiple command/data parameters written by the host using a store multiple words type of instruction. The Eddy can then write them out to a single Local bus address. This Local bus address is usually the external FIFO address in the Geometry Subsystem. The Local bus address is specified in the Local Command FIFO Address Register.

POS Register Operations

The POS registers are used to- configure the MGR adapter's MCA hus hardware interface. These registers contain the following hardware configuration information:

- MGR adapter ID
- Interrupt Request (IRQ) Level
- Card enable/disable bit
- Byte Order Selection bit
- Wait State Selection bit
- DMA block transfer size and DMA Bus Master enable/disable bit
- Data Streaming Mode enable/disable bit
- Error checking enable/disable bits
- Memory address
- Arbitration Level
- Fairness enable/disable bit
- Channel Check Status
- VPD subaddress registers

The Reference Disk is used to store the POS configuration data into the battery backed-up CMOS RAM. The Reference disk uses the configuration data contained in the adapter description file to create the POS register configuration data which it places into the CMOS RAM. At system poweron

or reset the POS registers are normally programmed by the Power On Self Test (POST) routines using the configuration information from the CMOS RAM. The POST reads the CMOS RAM to get the POS configuration data and verifies the adapter ID matches the ID on the card plugged into the corresponding slot. If the ID does not match an error is reported and the user must use the reference disk to reconfigure the system. Normally the ID matches so the POS registers are programmed as specified in the configuration data. This arrangement prevents conflicts between the various different adapters plugged into the system. It also means that the various hardware configuration parameters written into the POS registers may vary depending on the system configuration.

The host software that will access the MGR adapter must read the POS registers to determine the current configuration. The host software can write out changes to the configuration as needed. The only limitation is that the configuration parameters such as card IRQ level and memory address should not be changed. If they are changed a conflict with another card may result. The host software also read and writes the POS registers to access the VPD PROMs Located on the MGR adapter cards. The POS registers 5 and 6 are also read to determine the channel check status if a Channel Check condition is activated for the MGR adapter.

Interrupt Operations

The MGR adapter can be programmed to receive an interrupt on the IRQ level programmed into the POS 2 register. The appropriate interrupt vector should be programmed with the Interrupt Service Routine (ISR) address. The ISR should be able to service any interrupts which are enabled using the Interrupt Mask Register. The Interrupt Status Register is read to determine which interrupt caused the interrupt. Refer to the Registers and the Interrupts sections above for additional details on the types of interrupts.

DMA Operations

The Eddy module contains two DMA channels which can be programmed to perform Bus Master DMA cycles. Only one channel can be active at a time, so the alternate channel can be programmed and started while the other channel is completing it's DMA transfer. When the active channel completes it's transfer the alternate channel which had been pending will begin it's transfer. Each channel has a Source/Destination Address Register, a Length Register and a Control/Status Register. The source or destination address of the host memory buffer is programmed into the Source/Destination Register. The word count is programmed into the Length Register and then the Control/Status Register is programmed to set the direction of the transfer and to start the transfer.

If the DMA complete or DMA error interrupts are going to be enabled then the ISR should be able to handle these two types of interrupts. For the DMA completion interrupt, the host software may notify the user program that the data is available or it may want to start another DMA operation. The action to be taken is dependent on the needs of the host software. In the case of the DMA error interrupt the host will probably want to retry the failed DMA operation. Once again the host software must determine what actions it will take in the event of an error.

Hardware Reset Operations

The Eddy module provides a Reset Register which can be used to reset the Eddy module and the other three graphics subsystems. This register should be used to perform a complete MGR adapter reset at power-on or whenever a hardware condition arises which makes a reset appropriate. After the MGR adapter is reset, the microcode must be downloaded and all the Display Subsystem components must be reprogrammed. The bitplanes will be undefined after a reset so they should be cleared. The reset does not cause the POS registers to be affected.

Diagnostics Operations

The Eddv module contains a **Diagnostics** Reaister which can be used to test the Eddv module internal FIFO operation in the event the **external** command FIFO is overflowed.

Programming Considerations

The programming considerations for the Host interface Subsystem concern the programming of the POS registers and the Control / Status registers. The following paragraphs describe the programming requirements for these registers.

POS Register Programming Considerations

The Programmable Option Select (POS) eliminates the need for switches on the adapter cards by replacing their function with programmable registers. The System Configuration utilities, on the Reference Disk provided by the system manufacturer, automatically create configuration data for the system board and each adapter installed in the system. The configuration utility reads the unique adapter ID number from each adapter and matches it with an adapter description file **(ADF).** The configuration information in the ADF files are used to create the system configuration which is stored in battery-backed CMOS RAM. Refer to the host system technical reference manuals for the specific POS information.

The configuration data is used by the power-on self-test (POST) routines to automatically -- configure the system whenever the system is powered-on or reset. The POST verifies that the configuration stored in the CMOS RAM matches the **ID** numbers of the installed adapters. if the configuration has changed then the configuration utility must be used to reconfigure the system. Once the configuration check passes the configuration data is programmed into the POS registers from the data stored in the CMOS RAM. The folbwing example shows an example adapter description file for the **MGR** adapter.

MGR Adapter Description File Example

```
Adapterid 08EE6h

AdapterName     "SGI Micro-Channel Graphics Adapter"

NumBytes    4

Namedttem

Prompt    "Interrupt Level'

choice    " Level 2 .    pos[0]=xxx001 Oxb  int    2
choice    . Level 3 .    pos[0]=xxx0011xb   int    3
choice    " Level 4 "    pos[0]=xxx0 1 00xb int    4
choice    . Level 5 "    pos[0]=xxx0101xb   int    5
choice    " Level 6 .    pos[0]=xxx01 1 Oxb int    6
choice    " Level 9 "    pos[0]=xxx1001xb   int    9
choice    . Level 10 "   pos[0]=xxx1 01 Oxb int    10
choice    " Level 11 "   pos[0]=xxx1011xb   int    11
choice    " Level 12 "   pos[0]=xxx1 1 00xb int    12

Help
```

'Determines the interrupt level used by the MGR Graphics adapter."

NamedItem

Prompt 'Memory Mapped I/O Address Range'

choice	" OCOOO to 0C7FFF .	pos[0]=x00xxxxxb mem 0C000h-0C7FFFh
choice	. 0C800 to OCFFFF "	pos[0]=x01 xxxxxb mem 0C800h-0CFFFFh
choice	" 00000 to 0D7FFF .	pos[0]=x1 Oxxxxxb mem 0D000h-0D7FFFh
choice	" 00800 to ODFFFF .	pos[0]=x1 1 xxxxxb mem 0D800h-0DFFFFh

Help

"Determines the range of Memory Mapped I/O addresses used by the MGR Graphics adapter. The addresses in the selected range can not be used by any other device.

NamedItem

Prompt "Arbitration Level"

choice	" Arb Level 1"	pos 3]=xxxx0001b	arb	1
choice	" Arb Level 0 "	pos 3]=xxxx0000b	arb	0
choice	. Arb Level 2 "	pos 3]=xxxx001 Ob	arb	2
choice	" Arb Level 3 .	pos 3]=xxxx0011b	arb	3
choice	" Arb Level 4 .	pos 3]=xxxx0100b	arb	4
choice	. Arb Level 5 .	pos 3]=xxxx0101b	arb	5
choice	" Arb Level 8 "	pos 3]=xxxx011 Ob	arb	6
choice	" Arb Level 7 .	pos 3]=xxxx0111b	arb	7
choice	" Arb Level 8 "	pos 3]=xxxx1000b	arb	8
choice	" Arb Level 9 "	pos 3]=xxxx1001b	arb	9
choice	. Arb Level 10 .	pos 3]=xxxx101 Ob	arb	10
choice	" Arb Level 11 .	pos[3]=xxxx1011b	arb	11
choice	. Arb Level 12 .	pos[3]=xxxx1100b	arb	12
choice	. Arb Level 13 "	pos[3]=xxxx1101b	arb	13
choice	. Arb Level 14 .	pos[3]=xxxx111 Ob	arb	14

Help

'Determines the bus arbitration level used.'

The above example only sets the configuration parameters which may cause a conflict with other system components. The memory address range choices are the valid choices on a PS/2 under DOS. If OS/2 or Unix were used the choices might be different. Also the configuration utilities on the Reference Disk can not access the sub-address registers where the physical memory address is stored. Therefore the BLKSIZ bits in POS register 2 are used to store a 2 bit field which is used by the low level device software to program the sub-address registers with the appropriate memory address. The BLKSIZ parameter and the other parameters in POS registers 2 through 5 are also initialized by the low level device software.

The programming requirements for accessing the POS registers depends on the type of host system and on the operating system. The system technical reference manuals should be consulted to determine the exact POS register programming requirements.

MGR Memory Mapped I/O Prdgramming Considerations

The MGR adapter occupies a 32K byte range of memory. The physical memory base address of the adapter is programmed into the POS sub-address registers 1-3. When the low level device software is performing the adapter initialization it will read these registers to get the base address of the adapter. This address is stored in the variable GRP which is used in all the various adapter hardware access macros described in this document. All accesses to the adapter hardware represent an offset from the base address stored in the POS sub-address registers. The lower 16K bytes of the address range are reserved for the other three graphics subsystems and will be described in the following chapters. The second 16K range of addresses are reserved for the Eddy module Control / Status registers, the Local FIFO data area and the Diagnostic data area. The following macros defined in the file mgr.h allow the Control/Status registers to be accessed.

```
/* The GRP variable is defined as a global variable and is initialized with the physical address of
the MGR adapter which is contained in the POS sub-address registers 1-3.*/

unsigned long     GRP;

/* The following macro is placed in the variius software modules that use the MGR hardware
access macros. */

#define  GRPsetup     extem unsigned long GRP;

/* Control / Status Register base address offset and physical address definitions */

#define  MGR_REGS_OFF          0x00004000

#define  MGR_CSR_PHYS          (GRP | MGR_REGS_OFF)

/* Control / Status Register address offset definitions for the RT/2 */

#define  MGR_DMA_ADDR0         0x00    r DMA 0 src/dest addr reg */
#define  MGR_DMA_LEN0          0x04    /* DMA 0 word length count reg */
#define  MGR_CTRL0             0x0B    /* DMA 0 control/status reg */
#define  MGR_IMASK             0x0A    /* Interrupt mask reg */
#define  MGR_ISR               0x09    /* Interrupt status reg */
#define  MGR_GFX_RESET         0x08    /* Graphics reset reg ● I
#define  MGR_CMD_FIFO          0x0E    /* Command FIFO address */
#define  MGR_DIAG_CTRL         oxoc    r Diagnostic Control reg */
#define  MGR_DMA_ADDR1         0x10    r DMA 1 src/dest addr reg */
#define  MGR_DMA_LEN1          0x14    r DMA 1 word length count reg */
#define  MGR_CTRL1             0x1 B   r DMA 1 control/status reg */

/* Control / Status Register address offset definitions for the PS/2 */

#define  MGR_DMA_ADDR0         0x00    r DMA 0 src/dest addr reg */
#define  MGR_DMA_LEN0          0x04    r DMA 0 word length count reg */
#define  MGR_CTRL0             0x08    r DMA 0 control/status reg */
#define  MGR_IMASK             0x09    /* Interrupt mask reg */
#define  MGR_ISR               0x0A    /* Interrupt status reg */
#define  MGR_GFX_RESET         0x0B    r Graphics reset reg */
#define  MGR_CMD_FIFO          oxoc    /* Command FIFO address */
#define  MGR_DIAG_CTRL         0x0E    /* Diagnostic Control reg */
```

```
#define   MGR_DMA_ADDR1     0x10    I' DMA 1 src/dest addr reg */
#define   MGR_DMA_LEN1      0x14    /* DMA 1 word length count reg */
#define   MGR_CTRL1         0x18    /* DMA 1 control/status reg */
```

Adapter Reset Programming Considerations

The MGR adapter must be reset before it is ready to be programmed. The Reset register has two bits which allow the overall adapter to be reset and the Eddy module to be reset. Bit 7 must be reset to zero to initiate an adapter reset operation. This bit must be left zero for at least one millisecond. When it is set to one the adapter will be reset and is ready to be programmed. Bit 6 of the Reset register allows the host software to reset the Eddy module without affecting the rest of the adapter. This bit must be toggled to zero and then back to one again to perform the Eddy module reset. The POS registers are not affected by either of the reset operations. The macro **RESET_GFX** defined in the file mgr.h is provided to perform the reset operation. This macro performs both an adapter reset and an Eddy module reset.

```
/* Graphics Reset Register definitions */

#define   GFX_RESET_ADDR        (GRP | MGR_REGS_OFF | MGR_GFX_RESET)
#define   GFX_RESET_BIT         0x80
#define   GFX_DMA_RESET_BIT     0x40

#define   RESET_GFX(X) \
    if (x) \
      '(volatile char ●  )GFX_RESET_ADDR &= ~(GFX_RESET_BIT | GFX_DMA_RESET_BIT); \
    else \
      '(volatile char ●  )GFX_RESET_ADDR |= (GFX_RESET_BIT | GFX_DMA_RESET_BIT); \
```

Example Usage:

```
RESET_GFX(0);       r Start reset operation */

DELAY();            /* Delay for at least 1 millisecond ● /

RESET_GFX(1);       r Set to one for normal operation */
```

DMA Programming Considerations

The **MGR** adapter has two DMA channels which can be programmed via the Eddy modules registers. Only one DMA channel can be active at any'given time. The currently active channel must also complete it's transfer before the other channel can begin. The inactive channel can be programmed while the other channel is transferring data. When the inactive channel's start bit is set it will go into a pending state and will start when the other channel completes it's transfer. The POS registers contain several bits which must be programmed before any DMA operations are initiated. The DMA data block size must be placed in the **BLKSIZ** field of the POS 2 register. The Bus Master DMA Enable bit in POS 4 subaddress register 0 must be set *to* enable any DMA operations. Finally, if the host system supports Data Streaming mode then the Data Streaming Mode Enable bit in POS 4 sub-address register 0 should be reset to zero to enable data streaming mode. The POS register bits are normally programmed as part of the adapter initialization.

The individual DMA transfers are programmed using the Eddy module control I status registers. Each channel has a host system buffer address register, a word count register and a DMA control/ status register. The physical address of the DMA buffer in the host system's memory is placed in the source / destination address register for the desired channel. The word count of the data to be

transferred is placed in the transfer length count register. The direction bit is set depending on the direction of the data transfer and when the everything is ready the start DMA bit is set to begin the transfer. The completion of the transfer can be handled with interrupts or by polling the DMA status. When a DMA transfer completes, the interrupt service routine reads the interrupt status register to determine the cause of the interrupt. If the interrupt is due to a DMA completion the interrupt service routine would continue with normal DMA processing. If the interrupt is due to a DMA error then the interrupt service routine would check the DMA status field in the DMA control / status register for the channel that caused the error. The status field indicates the cause of the error and allows the host software to take the appropriate corrective action. The following macros in the file mgr.h are provided for accessing the DMA control / status registers.

```
/* DMA control / status registers */

#define GFX_CTRL0_ADDR      (GRP|MGR_REGS_OFF|MGR_CTRL0)
#define GFX_CTRL1_ADDR      (GRP|MGR_REGS_OFF|MGR_CTRL1)
#define GFX_DMA_STATUS      0x1C        /* Status mask - 3 bits */
#define GFX_DMA_DIRECT      0x02        /* DMA direction bit */
#define GFX_DMA_START       0x01        /* DMA start bit */
```

Interrupt Programming Considerations

The Eddy module contains an interrupt mask register and an interrupt status register for controlling host system interrupts from the MGR adapter. The Eddy module multiplexes eiaht possible interrupts onto a single interrupt level to the host system. The interrupt level is selected by the value programmed into the POS 2 register bits 4-1. This value is set by the POST routines during power-on or a system reset using the configuration data stored in the CMOS RAM by the set configuration program on the Reference disk. The host software gets the interrupt level by reading the POS 2 register and setting the appropriate interrupt vector to transfer control to an interrupt service routine.

Once the interrupt service routine's vector is installed the host software can then enable any of the eight interrupts by setting the appropriate bits in the interrupt mask register. After an interrupt occurs the interrupt service routine can read the interrupt status register to determine the cause of the interrupt. The interrupt vector location and the mechanism for disabling and enabling host system interrupts is system dependent. Refer to the system technical reference manuals for information on how to perform these tasks, The following macros are provided in the file mgr.h for accessing the bits in the interrupt mask register and the interrupt status register.

```
/* Interrupt Mask Register Definitions */

#define GFX_IMASK_ADDR      (GRP|MGR_REGS_OFF|MGR_IMASK)

#define GFX_VR_MASK         0x01        /* Vert retrace int mask */
#define GFX_GE_MASK         0x02        /* Graphics int mask */
#define GFX_FIFO_MASK       0x04        /* FIFO half-full int mask */
#define GFX_DMA0_ERR_MASK   0x08        /* DMA 0 Error int mask */
#define GFX_DMA1_ERR_MASK   0x10        /* DMA 1 Error int mask */
#define GFX_DMA0_FIN_MASK   0x20        /* DMA 0 Finish int mask */
#define GFX_DMA1_FIN_MASK   0x40        /* DMA 1 Finish int mask */

/* Interrupt Status Register Definitions */

#define GFX_ISR_ADDR        (GRP|MGR_REGS_OFF|MGR_ISR)
```

```
#define GFX_INT_VR          0x01    /* Vert retrace int */
#define GFX_INT_GE          0x02    /* Graphics int */
#define GFX_INT_FIFO        0x04    /* FIFO half-full int */
#define GFX_INT_DMA0_ERR    0x08    /* DMA 0 Error int */
#define GFX_INT_DMA1_ERR    0x10    /* DMA 1 Error int */
#define GFX_INT_DMA0_FIN    0x20    /* DMA 0 Finish int */
#define GFX_INT_DMA1_FIN    0x40    /* DMA 1 Finish int */

/* GFXIMASK returns the contents of the interrupt mask register */

#define GFXIMASK \
    (*(volatile char *) GFX_IMASK_ADDR)

/* GFXINTR returns the contents of the interrupt status register */

#define GFXINTR \
    (*(volatile char *) GFX_ISR_ADDR)

/* MASK_GFXINTR sets the specified interrupt mask register bit to zero */

#define MASK_GFXINTR(x) \
    (*(volatile char *)( GFX_IMASK_ADDR) &= ~(x))

/* GFXIMASK_WR writes the specified value to the interrupt mask register */

#define GFXIMASK_WR(x) \
    *(volatile char * ) (GFX_IMASK_ADDR) = (x)

/* CLR_GFXINTR resets the specified interrupt bit by writing a one to the specified bit in the
interrupt status register */

#define CLR_GFXINTR(x) \
    (*(volatile char *) (GFX_ISR_ADDR) |= (x))

/* CLEAR_VRI resets the vertical retrace interrupt by writing a one to the vertical retrace bit
in the interrupt status register */

#define CLEAR_VRI() \
    *(volatile char *) (GFX_ISR_ADDR) = GFX_INT_VR;
```

Status Polling Programming Considerations

Instead of using interrupts the host software may poll the status registers in the Eddy module to determine when an event has occurred. The interrupt status register can be polled to determine when a condition has occurred that would cause an interrupt to be generated if the corresponding interrupt bit were set in the interrupt mask register. The polling software would then take the appropriate action. The host software can also poll the DMA control status registers to determine the status of the vertical retrace and the status of the FIFO half full flag. The following macros in the file mgr.h are provided.

```
/* DMA control / status registers */

#define GFX_CTRL0_ADDR          (GRP | MGR_REGS_OFF | MGR_CTRL0)
```

```
#define GFX_CTRL1_ADDR          (GRP|MGR_REGS_OFF|MGR_CTRL1)
#define GFX_VRSTAT              0x80        /* vertical retrace status bit */
#define GFX_FIFOSTAT            0x60        /* FIFO half-full status */
/* FIFO half-full polling macro waits until the FIFO is < half full */
#define GEWAIT \
    while (((*(volatile char *)GFX_CTRL0_ADDR) & GFX_FIFOSTAT));
```