Sun SCSI Theory of Operation

W. M. Bradley

ABSTRACT

This manual describes the circuitry of the Sun Microsystems SCSI board. The SCSI board is an IEEE P796 board which contains an interface to the ANSI standard Small Computer Systems Interface (SCSI) bus, as well as four serial lines with full modem control.

1. Manual Overview

The SCSI board will be described in 3 sections: the SCSI section, the serial line or UART section, and the P796 bus interface circuitry which is shared by the other two sections.

Definitions and Conventions

Host Bus — The P796 bus or the internal bus which is just a buffered version of the P796 bus. The P796 bus signals names are prefixed by "P1." (e.g. - P1.D06\), while the internal bus signal names are not (e.g. - D06).

Asserted — An active-low signal is asserted when it is low. An active-high signal is asserted when it is high.

Negated — A signal is negated when it is not asserted.

Signal Names — Active high signals have names which do not end with a backslash ("\") character (e.g. - A03). Active-low signals have names which end with a backslash ("\") character (e.g. - SEL\). Signals which are related to the SCSI circuitry only generally are prefixed with "S." (e.g. - S.DMAREQ). Signals which directly connect to one of the connectors along the top of the board end with "|".

Component Designators — IC's are numbered Uxxx, where xxx is a number. The hundreds digit of the number tells which page of the schematic the part is on. If the part has multiple independent sections, sections are numbered starting at 0. In the case of 74LS74 of 74F74 flip-flops, section 0 is the section whose clock input is pin 3, and section 1 is the one whose clock is pin 11. In general, the lower the pin numbers, the lower the section number.

2. SCSI Interface

The SCSI bus is an ANSI standard interface bus which is capable of connecting processors to peripherals and to other processors. It is intended to provide an interface protocol which is largely independent of the manufacturer of the peripheral device attched. SCSI peripheral devices may include disk drives, tape drives, printers, network interfaces, and others. A *Host Adapter* is needed to interface between the SCSI bus and the processor which wants to use the SCSI peripherals. The Sun SCSI Host Adapter comes in two versions: as a P796 card, and as part of of the Sun-2 single board. Except for minor (and inevitable) differences relating to the fact that the P796 bus is not the same as the internal bus on the Sun-2 single board, the two versions are identical. For further information about the SCSI bus, consult the *SCSI Buyers' Guide,* published by ADES. The *Controller/System Interface Specification* by Adaptec, Inc is also

useful.

The Sun SCSI Host Adapter conforms to the SCSI spec. Arbitration is not supported. DMA is used for the data transfer phase of an SCSI transaction. Programmed I/O is used for the command, status, and message phases.

IMPORTANT This manual assumes that you understand the SCSI spec. If you don't, you may need to first read one of the documents referenced above.

The SCSI interface has 2 main sections: the SCSI bus interface and the DMA controller. The SCSI bus interface is the same for both this board (for the P796 bus) and the Sun-2 Single Board. The DMA controller is largely the same for both boards, but the circuit implementation details differ.

2.1. SCSI Bus Interface

The SCSI Bus Interface consists of the Data Register, the Command/Status Register, the Interface Control Register, the Parity circuit, and some control circuitry.

2.1.1. Register Addresses

In the following table, "base" is the base address of the SCSI board, as configured by dipswitch U305 (see section 4.2).

Address	Size	Register	Mode
base+ 0x00	16	Data	read/write
base+ 0x02	8	Command/Status	read/write
base+ 0x04	16	Interface Control	read/write
base+ 0x06		reserved	
base+ 0x08	16	DMA Address High Word	write
base+ 0x0a	16	DMA Address Low Word	write
base+ 0x0c	16	DMA Count	read/write
base+ 0x0e		reserved	

2.1.2. Connector

J3, a 50 pin header connector, is the SCSI connector. It is mechanically and electrically compatible with the SCSI spec. Consult the SCSI spec or the SCSI board schematic for the pinout.

2.1.3. Interface Control Register

The Interface Control Register is used to control the SCSI host adapter by enabling/disabling various modes and by asserting or testing control lines on the SCSI bus. It consists of U110, U115, and U119. When the CPU writes to the Interface Control Register, the data present on the lower half of the host data bus will be latched into U115. This latch is automatically set to all zeroes whenever the system Init signal is asserted. Bits 4 and 5 connected to control lines on the SCSI bus via open-collector drivers in U100. If a 1 is written to either of these bits, the corresponding SCSI control line will be asserted (active low). Other bits in the lower byte control various modes of the Host Adapter, as described in the Sun SCSI Programmers Manual.

When the CPU reads the Interface Control Register, buffers U110 and U119 drive the host data bus. Some of the bits reflect the state of the host adapter. Others sense the SCSI control lines. U116 receives the SCSI control lines, providing hysteresis for noise immunity and inverting the active-low SCSI lines to the active-high logic convention used on the host adapter.

2.1.4. Data Register

The Data Register is used for transferring data from the host bus to the SCSI bus during the Data Phase of an SCSI transaction. It is 16 bits wide on the host bus side and 8 bits wide on the SCSI bus side. When it is read or written from the CPU, all 16 bits are accessed. When it is accessed from the SCSI side, one byte at a time is written, since the SCSI data bus is 8 bits wide. For information about how bytes are sequenced into and out of the Data Register, see the Byte Sequencing section elsewhere in this manual.

In addition, the Data Register is used to hold the selection ID's during the Selection Phase. The ID bits of the Host Adapter and the desired Target are written into the upper byte of the Data Register at the start of the selection procedure. The Data Register is used for this because during the Selection Phase the SCSI I/O, C/D, and MSG lines are in the same states as they are in for a "Data Out" transfer. Decoding is simplified by using the Data Register to hold the selection ID's, because the hardware need not treat the selection condition as being different from the Data Out condition. It also makes it possible to cope with controllers which either require the host adapter's ID to bit to be asserted (e.g. - the ADES Series 1 Disk Link) or do not allow the host adapter's ID bit to be asserted (contrary to the SCSI spec, e.g. - the Sysgen SC4000).

Note that due to the use of Am2952 bidirectional latches (U111, U113) for the Data Register, it is not possible to read back what has been written to the Data Register.

2.1.5. Command/Status Register

The Command/Status Register is used to send Commands and Messages and to receive Status and Messages. It is only 8 bits wide, appearing on the upper half of the host data bus. This register may only be accessed with programmed I/O. For writing, the data on the upper half of the host data bus will be latched into the register on the trailing edge of the write strobe. A byte that is written to this register will appear on the SCSI data bus when the SCSI C/D line is active (Command) and the SCSI I/O line is inactive (Output).

For transfers from the SCSI bus to the CPU (Status or Message In), the data on the SCSI data bus is latched into Command/Status Register on the leading edge of the SCSI REQ line, so long as the SCSI C/D line is active (Command) and the SCSI I/O line is active (Input). As for the Data Register, it is not possible to read back a value that has been written to the Command/Status Register.

2.1.6. Parity

If the Parity Enable bit in the Interface Control Register is set, U114 will generate and check odd parity on the SCSI data bus. During output operations, parity is asserted onto the SCSI parity line through an open-collector driver in U100. During input, the parity of the SCSI bus will be checked. If a parity error occurs on input, it will be latched by the "SCSI" control PAL, U108. The latching action is accomplished by internal feedback within the PAL. The indication is visible as bit 17 of the Interface Control Register. The only way to clear the indication is by momentarily clearing the Parity Enable bit.

2.1.7. SCSI Control Circuitry

The control circuitry of the SCSI Bus Interface performs these functions: SCSI Request sensing, SCSI Acknowledge generation, DMA Request generation, Interrupt generation, and Byte Sequencing. These functions interact with each other, and depend on the settings of two modes.

2.1.7.1. SCSI Modes

There are two modes which influence the behavior of the SCSI control logic. DMA Enable controls whether or not data transfers are done with DMA. Word Mode, when set, causes SCSI data bytes to be packed/unpacked in the Data Register, so that transfers over the host bus can transfer 2 bytes at a time. If Word Mode is off, a separate host bus transfer must occur for each byte that is transferred over the SCSI bus. The normal case is to turn on both DMA Enable and Word Mode.

When Word Mode is on, a flip-flop (section 1 of U106) keeps track of whether the last byte that was transferred over the SCSI bus was an odd numbered byte or an even-numbered one. The output of this flip-flop is called SecondByte. SecondByte is initially off, and changes state after each data byte is transferred over the SCSI bus.

2.1.7.2. Byte Sequencing

In order to take advantage of the full width of the host data bus, 2 bytes from the SCSI data bus are packed into the Data Register. The 2 bytes are transferred over the host data bus in one operation, but only one byte at a time over the SCSI bus. The control section takes care of sequencing the bytes so that they go on the SCSI bus in the proper order.

For incoming data, the first byte goes into the upper half of the Data Register, and the second onto the lower half. At this point the Data Register is full and must be read from the host side before more transfers can take place. The SCSI REQuest is not acknowledged until the host has read the Data Register (either with DMA or programmed I/O) and thus emptied it. The cycle then repeats.

For outgoing data, the first request finds the Data Register empty, so the Data Register must be written with data from the host before the request may be acknowledged. When the second request comes in, it may be satisfied immediately, since there is still one byte in the lower half of the Data Register which hasn't been sent.

If Word Mode is off, the byte sequencing is turned off so that each individual request from the SCSI bus must be satisfied with a separate host access to the Data Register.

2.1.7.3. SCSI Request sensing

When the SCSI REQ line is asserted, the control circuitry decides whether or not the New Request bit in the Interface Control Register should be set. In general, New Request will be set only when the CPU must intervene to accomplish the transfer. It will not be set if the byte sequencing circuit or the DMA circuit can satisfy the REQuest without involving a programmed I/O operation.

2.1.7.4. SCSI Acknowledge generation.

Each byte transferred over the SCSI bus is accompanied by a REquest/ Acknowledge handshake. The SCSI Target device (controller) asserts REQuest, the host adapter either accepts or provides a data byte, the host adapter asserts ACKnowledge, the Target negates REQuest, and the host adapter negates ACKnowledge. The Sun host adapter asserts ACKnowledge whenever either the Data Register or the Command/Status Register is accessed from the host bus. The access may come from either a DMA operation or a programmed I/O operation in the case of the Data Register. The Command/Status Register is only accessible with programmed I/O. When either register is accessed, S.REGACC will be asserted by the "Decode" PAL, U121. The trailing edge of S.REGACC is remembered by U106 section 0, whose output signal S.REGACCL causes the "SCSI Control" PAL U108 to assert ACKnowledge. The S.REGACCL latch is cleared when the Target negates REQuest.

In Word Mode, half of the Data bytes are transferred without any transfer over the host bus, and thus without any host access of the registers. In this case, the SCSI Control PAL generates an automatic acknowledge as soon as the REQuest comes in.

Request/Acknowledge/DMA Request Conditions							
Command/ Data*	Dma Enable	Input/ Output*	Word Mode	Second Byte	New Request	Auto Acknowledge	DMA Request
1	x	x	x	x	Yes	no	no
0	0	0	0	x	Yes	no	no
0	0	0	1	0	Yes	no	no
0	0	0	1	1	no	Yes	no
0	0	1	0	x	Yes	no	no
0	0	1	1	0	no	Yes	no
0	0	1	1	1	Yes	no	no
0	1	0	0	x	no	no	Yes
0	1	0	1	0	no	no	Yes
0	1	0	1	1	no	Yes	no
0	1	1	0	x	no	no	Yes
0	1	1	1	0	no	Yes	no
0	1	1	1	1	no	no	Yes

2.1.7.5. Interrupt Requests

An interrupt is generated during REQuests for Status, Messages, or Data with DMA Enable Off. Data does not cause an interrupt if the byte sequencing circuit takes care of the request without a host access to the Data Register. A bus error (described in the DMA Control section) will also cause an interrupt. If one of these conditions exists, and the Interrupt Enable bit is on, the CPU will be interrupted at a level determined by switch U312. As soon as the request is acknowledged, the interrupt request goes away. The SCSI Control PAL takes care of this.

2.2. DMA Control

The DMA controller generates 24-bit DMA addresses and timing strobes. A count register may be used to set a maximum DMA count and ensure that that count is not exceeded. DMA cycles that start but do not finish within a predetermined time cause a bus error interrupt.

2.2.1. DMA Address Register

The DMA Address Register (U205,U206,U207) is a 24-bit wide counter. It is written from the host bus in 2 parts - the higher 8 bits and the lower 16 bits. The DMA Address Register may not be read from the host bus, because the outputs connect to the address bus and readback would necessitate arbitrating for the address bus, which is too complicated. The DMA Address Register counts bytes. The control circuitry is responsible for incrementing the DMA Address Counter twice if a word is transferred. Circuit Detail: Incrementing the counters is done by driving the function select lines on the 'LS461 counters rather than by driving the Carry In to the lowest counter. The reason is that the propagation delay from Carry In to Carry Out is too long for reliable operation at 10 MHz, considering other circuit delays. Only the lower 20 bits of the DMA Address are used, because the P796 Bus has only 20 address lines.

2.2.2. DMA Count Register

The DMA Count Register is a 16-bit counter which may be both read and written from the host bus. It is used both to determine how many bytes were transferred with DMA, and to enforce a maximum count. The DMA Count Register counts in tandem with the DMA Address Register. When one of them increments, the other one does too. The DMA Count Register generates the signal S.OVERRUN, which is just Carry Out, when the count reaches -1 (or 65535, depending on your point-of-view). This signal is used by the control circuit to disable further DMA cycles. If the DMA Count Register is loaded with - MaxCount - 1 (ones

complement of MaxCount), then the DMA Count Register will read -1 when the maximum count has been reached, and no more DMA bytes may be transferred.

2.2.3. DMA Cycle Generator (control circuitry)

The control circuit is responsible for running the DMA cycle in response to a DMA request. This involves detecting the DMA Request, arbitrating for the use of the host bus, generating the timing strobes, and incrementing the counters. This circuitry is necessarily different between the P796 Bus implementation and the Sun-2 single board, due to different bus protocols used.

2.2.3.1. P796 DMA control implementation

The DMA controller is implemented as a state machine clocked by BCLK from the P796 bus. Since the bus arbitration circuit is synchronized to BCLK, the controller doesn't have to externally synchronize its communication with the arbitrater chip.

2.2.3.2. Request Synchronization

A new DMA request is initiated by a rising edge on S.XREQ. If there is no bus error condition, this edge will set flip-flop U105, driving S.DMAREQ active. S.DMAREQ is then synchronized to BCLK by U203 Section 1, which generates S.BR\ on the first clock edge after S.DMAREQ is asserted. S.BR\ clears U105. S.BR\ starts the arbitration for the P796 bus.

2.2.3.3. Arbitration

Arbitration is performed by an 8289 P796 bus arbitration chip. The S.BR\ signal informs the 8289 that the SCSI board wants to use the bus. When the 8289 gains control of the bus, it asserts S.SMAEN\.

2.2.3.4. Timing Generation

The timing strobes are generated by U204, a PAL16R4. See Appendix **** for a listing of the microcode. Appendix **** shows a timing diagram.

If the cycle is a transfer from the SCSI board to P796 memory, as indicated by S.I/O\ being high (Input), then S.RD.DMA\ and P1.MWTC\ are generated. S.RD.DMA\ is asserted as soon as S.DMAEN\ is asserted. One clock later, P1.MWTC\ is asserted. S.RD.DMA\ enables the Data Register to drive the internal data bus. P1.MWTC\ is the P796 write strobe. The one clock delay between S.RD.DMA\ and P1.MWTC\ guarantees the data setup time to the start of P1.MWTC\. P1.MWTC\ remains asserted until XACK is received. The P796 bus signal P1.XACK\ is externally synchronized by U203 section 0 before being sent to the DMA control PAL (because the flip-flops inside the registered PALS are not fast enough to be used as synchronizers at 10 MHz). The average time between the assertion of P1.XACK\ and the negation of P1.MWTC\ is thus 'average synchronization time + 1 clock' = 1 1/2 clocks = 150 nsec, assuming a 10 MHz BCLK. Once S.RD.DMA\ is asserted, it latches itself on and stays on until quite late in the cycle, after P1.MWTC\ has been negated, thus guaranteeing data hold time after the end of P1.MWTC\.

If the cycle is a transfer from P796 memory to the SCSI board, as indicated by S.I/O\ being low (Output), then S.WR.DMA\ and P1.MRDC\ are generated. S.WR.DMA\ and P1.MRDC\ are asserted as soon as S.DMAEN\ is asserted. Data is clocked into the Data Register on the rising (trailing) edge of S.WR.DMA\. S.WR.DMA\ is driven by a non-registered output of the DMA control PAL, and is negated (high) as soon as the XACK input to the PAL goes active. P1.MRDC\ is driven by a registered output, and is not negated until the next clock after the XACK input goes active. Since XACK is externally synchronized, this gives 1 clock (100 nsec) delay between the trailing edge of S.WR.DMA\ and the end of P1.MRDC\, providing plenty of data hold time for the Data Register.

Regardless of the direction of the transfer, the last part of the DMA cycle is always the same. During the last clock period of the P1.MRDC\ or the P1.MWTC\ strobe, when XACK is asserted but P1.MRDC\ or P1.MWTC\ has not yet been negated, the DMA request flip-flop (U203 section 1) is cleared. This is done by asserting S.BRCLR\. The bus arbitration chip requires that the trailing edge of S.BR\ (Bus Request) be synchronized to the falling edge of BCLK, rather than the rising edge. For this reason, S.BRCLR\ is only asserted when BCLK is low.

At the end of the cycle, the DMA counters are incremented. If WordMode is on, the counters are incremented twice. Incrementing occurs during the one or two clock periods immediately following the negation of P1.MWTC\ or P1.MRDC\. State0 (internal to the PAL) remembers that the cycle just ended, so that it is time to increment the counters. State1 occurs one clock after State0, and is used to count to two so that the counters may receive two increment pulses if necessary. The counters require that the increment signal be valid for at least 35 nsec before a rising edge of BCLK.

2.2.3.5. Bus Errors

If a DMA cycle starts but does not complete within a reasonable time, a Bus Error condition results. Timing starts after the arbitration for the bus has been won. After 128 P1.BCLK\ periods, if P1.XACK\ has not been received, S.BERR will go high. When S.BERR goes high, XACK is asserted to the DMA control PAL so that it will complete its cycle. The rising edge of S.BERR clocks flip-flop U105 section 0, latching the S.BUSERR signal. If S.BUSERR is active, the DMA edge-triggered latch U105 section 1 is prevented from registering any more SCSI DMA requests. This prevents repeated faulty DMA cycles from loading down the host bus. S.BUSERR also causes an interrupt request, subject to the Interrupt Enable bit being set. S.BUSERR is readable by the the processor as the Bus Error bit in the Interface Control Register. The SCSI board must be reset in order to clear the Bus Error condition. The obvious signal to use for enabling the DMA timeout counter (U211) is S.DMAEN\, which is the bus grant signal. This doesn't work, since S.DMAEN\ is not necessarily negated between DMA cycles. Instead, S.REGACC\ is used. S.REGACC\ is active when any of the SCSI registers are being accessed. The only time this happens for long enough for a timeout is when a DMA cycle is not acknowledged.

2.2.3.6. DMA Overrun

If the SCSI Target devices tries to transfer more bytes than the maximum number that the host software set up in the DMA count register, the hardware will prevent the extra bytes from causing DMA transfers. To set a maximum count, the software writes (-MaxCount - 1) to the DMA Count Register. When the maximum count has been reached, the DMA Count Register will have counted to -1, causing the carry out signal from the counters to be active. This signal is labelled S.OVERRUN. If exactly the maximum count is transferred, nothing bad will happen. However, if another transfer is attempted while the S.OVERRUN signal is asserted, U212 section 3 will prevent P1.MWTC\ or P1.MRDC\ from driving the bus. This is done by taking away the output enable for the drivers inside the DMA control PAL. Only the P1.MWTC\ and P1.MRDC\ outputs are affected by this; the other outputs are enabled internally. Since the read or write strobe will not drive the bus, no slave device will respond to the DMA cycle, causing a Bus Error as described above. The Bus Error will complete the DMA control PAL's cycle, incrementing the DMA count register past -1 to either 0 or 1, depending on whether Word Mode is on or off. No further DMA cycles will be requested, due to the Bus Error condition. The Overrun condition may be distinguished from a "normal" Bus Error by the fact that the DMA count register reads 0 or 1.

3. Serial I/O Channels

The SCSI board has 4 serial I/O channels, each with full modeom control. The channels are implemented with two Zilog Z8530 dual UART chips. The data sheet for the Z8530 should be consulted for programming information.

3.1. Register Addresses

Address	Register
base + 0x0800	Channel B Control
base + 0x0802	Channel B Data
base + 0x0804	Channel A Control
base + 0x0806	Channel A Data
base + 0x1000	Channel D Control
base + 0x1002	Channel D Data
base + 0x1004	Channel C Control
base + 0x1006	Channel C Data

Registers are 1 byte wide. If accessed as words, only the upper 8 bits is significant.

3.2. Connectors

Channels A and B appear on 50-pin connector J1. Channels C and D appear on 50-pin connector J2. The pin assignments are:

J1		J2	
Pin	Signal	Pin	Signal
3	TxD A	3	TxD C
4	DB A	4	DB C
5	RxD A	5	RxD C
7	RTS A	7	RTS C
8	DD A	8	DD C
9	CTS A	9	CTS C
11	DSR A	11	DSR C
13	GND A	13	GND C
14	DTR A	14	DTR C
22	DA A	22	DA C
28	TxD B	28	TxD D
29	DB B	29	DB D
30	RxD B	30	RxD D
32	RTS B	32	RTS D
33	DD B	33	DD D
34	CTS B	34	CTS D
36	DSR B	36	DSR D
38	GND B	38	GND D
39	DTR B	39	DTR D
47	DA B	47	DA D

Connector pins not mentioned are not connected to anything.

4. P796 Bus Interface

4.1. Address and Data Transceivers

U301, U303, and half of U313 are address transceivers. The P796 Address Bus is driven only when S.DMAEN is active, otherwise it is received. P1.A00\ is handled separately, as described later. U309, U310, and U311 are data transceivers. U309 and U310 are used for word transfers and lower byte transfers. U311 is used for upper byte transfers, since the P796 bus uses the lower half of the bus for byte transfers regardless of which byte is being transferred.

4.2. Decoding

The SCSI board responds to a 16 Kbyte bank of addresses. The first 2 K is used for the SCSI registers. The second 2 K is used for one UART chip (2 channels), and the third 2 K is used for the other UART chip (2 channels). The SCSI registers and the UARTS are on separate 2 K boundaries so that the memory protection hardware on the CPU can protect the devices separately (protection applies to 2 K pages). The base address for the SCSI board is set with Dip Switch U305. When the six high order address bits P1.A14\ - P1.A19\ match the address selected with sections 1-6 of U305, SEL\ will be asserted. A switch on (closed) corresponds to an address bit asserted (1 for the software or electrical low on the P796 bus). U314 and U307 decode address bits A11 - A13 to provide separate read and write strobes for the SCSI and the UARTS. U200 and U210 further decode A00 - A03 to separately address the different SCSI registers.

4.3. Transfer Acknowledge

The SCSI registers require less than 100 nsec for reading or writing, whereas the UART's take about 500 nsec to read. The XACK generation circuit has to cope with this difference and delay XACK by different amounts of time depending on which device is being accessed. U302 provides time delays from 0 nsec up to 700 nsec at 100 nsec intervals (assuming BCLK is 10 MHz). The time starts from when either P1.MRDC\ or P1.MWTC\ is asserted. U304 selects one of the time delays, depending on address bits A11 - A13, which are the same bits used to select between the SCSI section or the UART's. The output signal from U304 drives P1.XACK\ through U316 section 2, which is enabled by the board select line SEL\.

4.4. Interrupt Level

Dip Switch U312 selects the interrupt priority level for SCSI interrupts. Dip Switch U315 selects the interrupt priority level for UART interrupts. Only one of the switches in each Dip Switch should be on (closed) at a time.

4.5. Bus Priority

If serial priority resolution is used and the SCSI board is at the high priority end of the bus, section 8 of Dip Switch U305 should be closed to ground the Bus Priority In line. If parallel priority resolution is used, or the SCSI board is not the highest priority master, the switch should be left open.

4.6. Control

The MB Interface PAL (U300), generates signals for controlling the data transceivers, the time delay circuit, and the P796 byte qualifier lines. CE.WORD\ is active when the board is selected and the transfer is either an entire word or just the lower byte. CE.BYTE\ is active when the board is selected and the transfer is the upper byte only. MB.IN\ is active when the direction of transfer is from the P796 bus to the SCSI board. MB.DS\ is active when the board is selected and one of the data strobes (P1.MWTC\ or P1.MRDC\) is active. P1.BHEN\ is active during a DMA transfer when the transfer is an entire word (Word Mode is on). P1.A00\ is active during a DMA transfer when the transfer is for the lower byte. **** If DMA is performed one byte at a time, the byte order is 68000 byte order (Big Endian), which means that the upper byte of a word is transferred first, or, equivalently, that an even DMA address refers

to the upper byte of a word. ****