Model 2917-Z1A

VMEbus Interface w/DMA

INSTRUCTION MANUAL

May 22, 2001

CONTENTS

FIGURES

Schematic Drawing # 086101-C-6033 See Reply Card Following Warranty

Warranty

VME Interface for the 3922 Controller

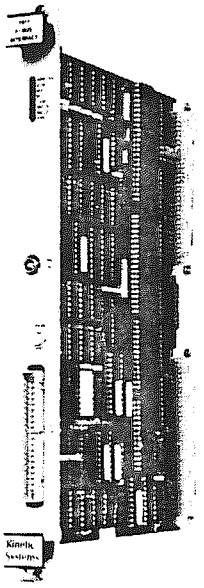
Allows a computer VME bus to host up to eight 3922s **2917**

Features

- **Provides dedicated VMEbus interface**
- **Used with 3922 parallel-bus crate controllers**
- **Supports DMA transfers**
- **Controls up to eight CAMAC crates**
- **Bus lengths up to 90 meters (300 feet)**
- **RS-485 balanced-line signaling between the 2917 and 3922s for high noise immunity**
- **Throughput up to 1 megabyte per second**
- **16 kilobyte command list memory**
- **Full VMEbus master capabilities**
- **Supports 16-bit VMEbus transfers**
- **Flexible interrupt structure**

Typical Applications

- **Integrate CAMAC with VME systems**
- **General-purpose data acquisition and control**
- **Laboratory automation**
- **Industrial process control**
- **Flight simulation systems**



General Description *(Product specifications and descriptions subject to change without notice.)*

The 2917 is a computer-bus adapter for use with computers incorporating the VMEbus. The 2917 provides the interface between the VMEbus and as many as eight CAMAC crates using the 3922 Parallel Bus Crate Controller. Connection is made via a parallel bus that is a 40-conductor twisted-pair ribbon cable. Bus lengths must be ordered separately. Signaling on the parallel bus is accomplished using RS-485 balanced line drivers and receivers, which gives high noise immunity and allows an overall cable distance between the 2917 and the last 3922 of up to 90 meters (300 feet). The last 3922 on the parallel bus is terminated with a termination card provided with the 3922.

Data throughput rates for a 16-bit data transfer using a 5 meter cable between the 2917 and a 3922 is 920 kbytes per second for CAMAC double-buffer read operations. 890 kbytes can be achieved with CAMAC write operations.

CAMAC operations are executed from a 16-bit x 8192 word on-board command-list memory. CAMAC data to and from the VMEbus can be transferred with the 2917 acting as either bus master or slave. All data transfers between the 2917 and the VMEbus are 16 bits wide. 32-bit addresses are generated by the 2917 during DMA transfers. Block transfers based on the CAMAC Q-response are supported.

The 2917 can generate an interrupt on the VMEbus when a CAMAC Look-At-Me (LAM) occurs or when the entire command list has been executed. The request level at which the interrupt is made is programmable from VMEbus, as is the generated interrupt vector.

Power Requirements

+5 volts:	4020 mA

Ordering Information

Model 2917-Z1A VMEbus Interface to the model 3922 crate controller, with DMA

Associated Products

Model 3922-Z1B	Parallel Bus Crate Controller
Model 5843-Txyz	Bus Interface Cable (one required for each 3922)

UNPACKING AND INSTALLATION

The Model 2917 is shipped in a antistatic bag within a Styrofoam packing container. Carefully remove the module from its static-proof bag and set the various options to conform to the operating environment. The default settings on the 2917 are as follows: base address set to $FF00 HEX, VME Bus Request set level three, and Data Storage Format set to LOW/HI. Figure 3 shows the location of components affected by option choices.

The 2917 is a VME "B" size module that requires both P1 and P2 VME backplane connectors. The selected VME slot that the 2917 will reside in requires the Bus Request (BR) and Interrupt Acknowledge (IACK) straps be removed for proper VME bus operation. Any empty slots between the 2917 and slot one should have their BR and IACK straps loaded. Connect the 2917 front panel K-Bus connector to the 3922 crate controllers with the last 3922 having a termination card connected. Refer to the 3922 Crate Controller section for more information on the 3922.

SWITCH AND STRAP SELECTION

Device Address Selection

The 2917 resides in the short I/O address space of the VME bus and responds to both short I/O Address Modifier Codes - 29 HEX (Short Non-Privileged Access) and 2D HEX (Short Supervisory Access). The switches SA15 through SA08 determine the base address of the 2917. The 2917 must allocate the 256 word location in the short I/O address space. The default base address setting is FF00 HEX.

WHEN SELECTING A BASE ADDRESS, CARE SHOULD BE TAKEN TO AVOID ADDRESSES ASSIGNED TO OTHER DEVICES IN THE SHORT I/O ADDRESS SPACE.

The bit pattern for the base address is shown below:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SA 15	SA 14	SA 13	SA 12	SA 11	SA 10	SA 09	SA 08	0	0	0	0	0	0	0	0

Bits 15 through 08 are user selectable via the address switches SA15-SA08.
Bit 07 is set to zero.
Bits 06 through 00 are set to "0" to indicate a block of 256 words.

Refer to Figure 3 on page 38 for switch location and switch settings.

VME Bus Grant Selection

The 2917 can be selected for all four Bus Request (BR) settings. The 2917 uses the BR signals to become a VME master for DMA operations. The BR straps on the VME backplane need to be removed from the slot that the 2917 occupies. The default setting is BR #3. Refer to Figure 3 on page 38 for location of Bus Request straps and settings.

Data Storage Format Selection

The 2917 has an option to select the order in which a 24-bit CAMAC word is fetched/stored. A strap on the 2917 selects this option with the default set to LOW/HI. This setting will first access the lower 16-bit CAMAC word then the upper 8-bit CAMAC word. Refer to the DMA section for more operating details, also refer to Figure 3 on page 37 for location and strap selection.

FRONT PANEL LEDS

ADD_REC: This LED indicates when the 2917 is being addressed from the VME bus.

RUN: This LED indicates that the 2917 is running a command list.

M2,M1: These two LEDs indicate the current CAMAC operating mode. They reflect bits 4 and 3 in the current Mode word of the Command List.

LAM: This LED indicates that one or more crates connected to the 2917 has a LAM set and is need of service.

NO-X: This LED indicates the status of the CAMAC X-response from the last CAMAC command that was executed.

NO-Q: This LED indicates the status of the CAMAC Q-response from the last CAMAC command that was executed.

ERR: This bit indicates that the last CAMAC operation had ended in an error condition.

External Trigger In

The 2917 has a front panel Trigger-In input to start the execution of the Command List. This is the same thing as setting the GO bit in the CSR. The trigger pulse is a low-true signal requiring a minimum of 50 ns pulse width. The DMA bit and/or DIR bit in the CSR must be set to the desired setting prior to the Trigger-In input.

Reset Button

The front panel Reset button will reset the 2917 to a power-up condition. The command memory data will retain the previous stored data.

2917 REGISTER DESCRIPTION

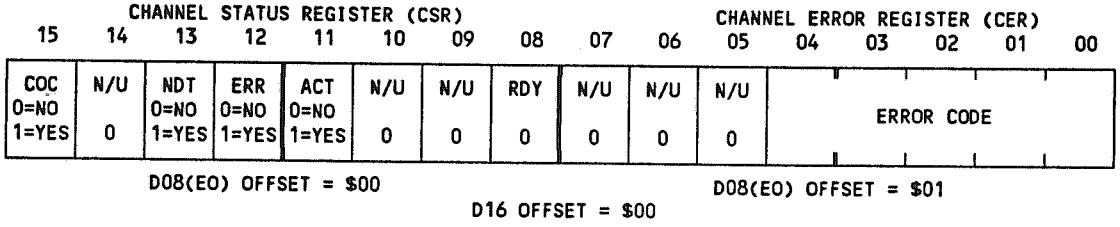
DMA CONTROLLER REGISTER

The DMA Controller has 12 8-bit registers that need to be programmed prior to any DMA transfer on the VMEbus. These 12 registers can be grouped as six 16-bit word registers for ease of programming. Programming information on the DMA controller can be found below in the

register descriptions and also in the Technical Specifications included in Appendix A of this manual.

Channel Status/Error Register

The Channel Status/Error Register (CSER) is used to monitor current status and error conditions of the DMA controller. Prior to starting a DMA transfer, bits #15, #13 and #12 must be cleared to zero. A register layout and description of the Channel Status/Error Register is given below:



BIT #	MNEMONIC	DESCRIPTION
15	COC	**Channel Operation Complete.** When this bit is set, a DMA transfer has completed whether successful or not. This bit must be cleared before another transfer is started.
14	N/U	**Not used.** Read as zero.
13	NDT	**Normal Device Terminate.** When this bit is set, the command list has aborted. Check the 2917 on-board CSR register for the abort condition. This bit must be cleared before another transfer is started.
12	ERR	**Error.** Error is set when the transfer termination was due to an error. Read the Channel Error Register to determine the cause. This bit must be cleared before another transfer is started.
11	ACT	**Channel Active.** This bit is set when the channel has been started and remains set until the operation is terminated.
10, 9	N/U	**Not used.** Read as zeros.
8	RDY	**Ready Input State.** This bit indicates the 2917 is ready. This bit is always read as a zero.
7, 6, 5	N/U	**Not used.** Read as zeros.

4-0	ERR CODE	**Error Code.** This Error Code indicates the source of the error when indicated by CSR #12. These Error Code bits are clear when the CSR #12 is cleared.

ERROR CODES

```
00000 = No Error
01001 = Bus Error
10001 = Software Abort
```

Device/Operation Control Register

The Device/Operation Control Register (DOCR) is used to select operating conditions of the DMA Controller. This register must be written to, before a transfer is started. A register layout and description of the Device/Operation Control Register is given below:

DEVICE CONTROL REGISTER (DCR) — OPERATION CONTROL REGISTER (OCR)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
EXT REQ MODE	N/U	N/U	N/U	OCR BITS (5:4)	N/U	N/U	N/U	DIR	N/U	OPERAND SIZE		N/U	N/U	N/U	N/U
0	0	1	1		0	0	0		0			0	0	1	0

D08(EO) OFFSET = $04 — D08(EO) OFFSET = $05

D16 OFFSET = $04

BIT #	MNEMONIC	DESCRIPTION
15	EXT REQ MODE	**External Request Mode.** This bit must be written with a one, the 2917 only can operate in Cycle Steal mode. A value of zero equals Burst mode. This is not a valid operating mode.
14	N/U	**Not used.** Read as zero.
13, 12	N/U	**Not used.** Read as ones.
11	OCR(5:4)	This bit is set if either bit #4 or #5 in the Operation Control Register is set to a one.
10-8	N/U	**Not used.** Read as zeros.
7	DIR	**Direction:** 0 = Memory to 2917 (CAMAC writes) 1 = 2917 to Memory (CAMAC reads)
6	N/U	**Not used.** Read as zero.

5, 4	SIZE	**Operand Size.** Operand size must be set 01. The 2917 only transfers words.

Bits 5,4	Size
0 0 =	Byte
0 1 =	Word
1 0 =	Long Word
1 1 =	Double Word

3, 2	N/U	**Not used.** Read as zeros.
1	N/U	**Not used.** Read as one.
0	N/U	**Not used.** Read as zero.

Sequence/Channel Control Register

The Sequence/Channel Control Register (SCCR) is used to initiate the start of a DMA operation or to abort a DMA operation. A register layout and description of the Sequence/Channel Control Register is given below:

SEQUENCE CONTROL REGISTER (SCR) — CHANNEL CONTROL REGISTER (CCR)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
NOT USED								START	N/U	N/U	SOFT ABORT	INT ENA	N/U	N/U	N/U
0	0	0	0	0	1	0	0		0	0			0	0	0

D08(EO) OFFSET = $06 — D08(EO) OFFSET = $07

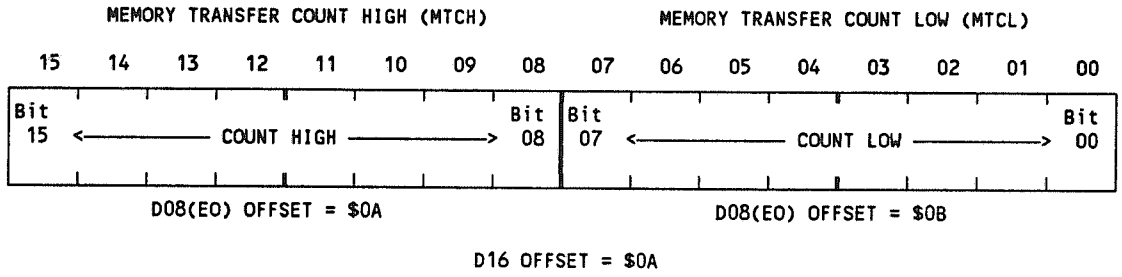
D16 OFFSET = $06

BIT #	MNEMONIC	DESCRIPTION
15-11	N/U	**Not used.** Read as zeros.
10	N/U	**Not used.** Read as one.
9, 8	N/U	**Not used.** Read as zeros.
7	START	**Start operation.** Setting this bit to a one will initiate the DMA controller and set the Active bit in the CSR. A pending start can only be reset by a software abort being set in the CCR.
6 ,5	N/U	**Not used.** Read as zero.
4	SOFT ABORT	**Software Abort.** Setting this bit stops a DMA transfer.

3	INT ENA	**Interrupt Enable.** This bit must be set to zero. The 2917 Bus Interrupter is used for DONE interrupts.
2-0	N/U	**Not used.** Read as zeros.

Memory Transfer Count Register

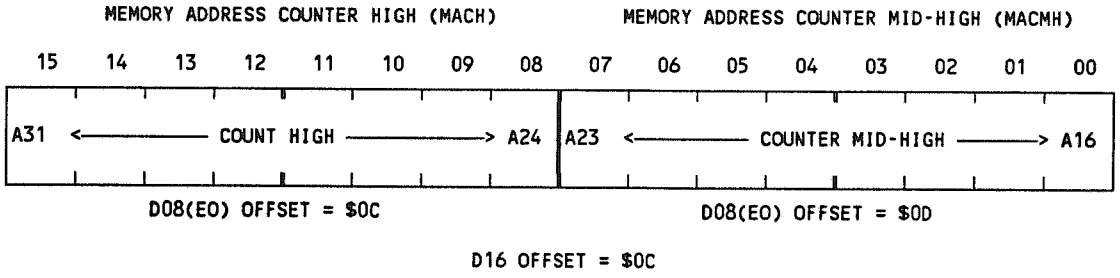
The Memory Transfer Count (MTC) Register is a 16-bit counter which counts the number of words transferred by the DMA controller. A register layout of the Memory Transfer Count Register is given below:



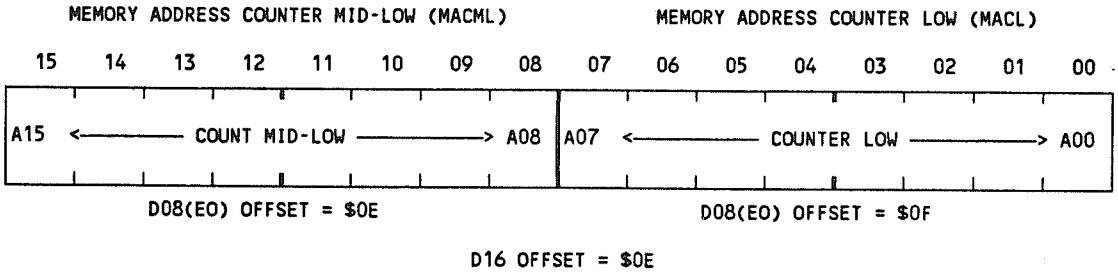
Memory Address Counter Register

The Memory Address Counter Register is a 32-bit register which programs the starting address of a DMA transfer. Only the least significant 24-bits of the counter are implemented in the DMA controller. If 32 operations are desired, the upper eight address bits can be programmed in the 2917 Address Modifier Register. Two 16-bit Memory Address Counters must be loaded before the start of a DMA transfer. A register layout of the Memory Address Counter Register is given below:

Memory Address Counter High (MACHI)



Memory Address Counter Low (MACLO)

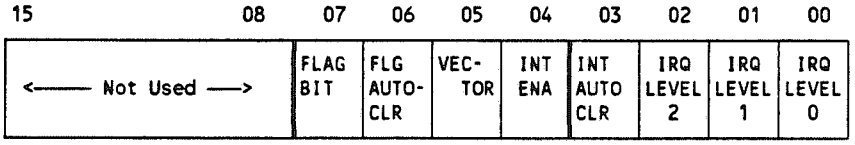


BUS INTERRUPT REGISTERS

The Model 2917 uses the MC68153 Bus Interrupter to generate four separate interrupts. The four conditions that can cause an interrupt are: 1) a LAM, 2) DONE bit in the CSR set, 3) DMA Buffer Empty and 4) List Abort condition. If any two or more interrupts are set to the same interrupt request level, MC68153 gives preference to the highest number requester; that is, List Abort has the highest priority and LAM has the lowest. Programming information on the Bus Interrupter can be found below in the register descriptions and also in the Technical Specifications included in APPENDIX B of this manual.

Interrupt Control Registers

The Interrupt Control Registers are used to enable and select the interrupt request level. A register layout and description of the Interrupt Control register is given below:



Interrupt Control Register OFFSET

LAM	**D16 = $40**	D08(EO) = $41
DONE	**D16 = $42**	D08(EO) = $43
DMA BUFFER EMPTY	**D16 = $44**	D08(EO) = $45
LIST ABORT	**D16 = $46**	D08(EO) = $47

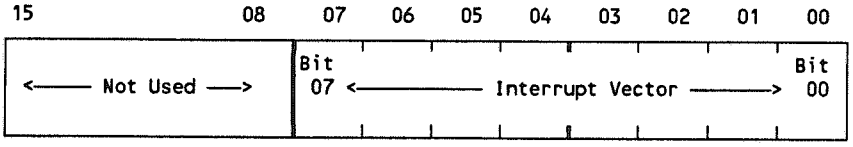
BIT #	MNEMONIC	DESCRIPTION
15-8	N/U	**Not used.** These bits are shown only for clarity when D16 operations are preformed. These bits are read as ones.
7	FLAG	**FLAG.** This bit can be used by software in conjunction with test and set instructions.
6	FLG AUTO-CLR	**Flag Auto-Clear.** When this bit is set, the FLAG bit is automatically cleared during an interrupt acknowledge cycle.
5	VECTOR	**Vector.** This bit must be set to zero, the 2917 uses internal vectors.
4	INT ENA	**Interrupt Enable.** Setting this bit to a one will enable interrupts.
3	INT AUTO-CLR	**Interrupt Auto-Clear.** Setting this bit to a one will clear bit #4 (INT ENA) during an interrupt acknowledge cycle

responding to this request. To re-enable, bit #4 must be written with a one.

2-0 **IRQ LEVEL** **Interrupt Request Level.** This three bit field determines the interrupt level, one through seven, to be generated. A value of zero disables the interrupt.

Interrupt Vector Registers

Each Interrupt Control Register has its own associated Interrupt Vector Register. Each register is 8-bits wide and supplies a data byte during an interrupt acknowledge cycle. A register layout of the Interrupt Vector Register is given below:



Interrupt Vector Register	OFFSET	
LAM	**D16 = $48**	D08(EO) = $49
DONE	**D16 = $4A**	D08(EO) = $4B
DMA BUFFER EMPTY	**D16 = $4C**	D08(EO) = $4D
LIST ABORT	**D16 = $4E**	D08(EO) = $4F

2917 ON-BOARD REGISTERS

All on-board registers are 16-bit wide. Only D16 word transfers to or from these are allowed. These eight registers are described in the following sections.

Address Modifier Register

The Address Modifier Register (AMR) is a write only register. Information written into the AMR is used to drive the VMEbus Address Modifier signals and the upper eight address signals (A31-A24) when the 2917 transfers data as a VMEbus master. The upper eight address signals cannot increment during a DMA transfer. This is only an 8-bit latch register that is enabled when the 2917 becomes a VMEbus master. A register layout and a description of the AMR is given below:

ADDRESS MODIFIER REGISTER

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
A31	A30	A29	A28	A27	A26	A25	A24	N/U	LWD	AM5	AM4	AM3	AM2	AM1	AM0

D16 OFFSET = $60

BIT #	MNEMONIC	DESCRIPTION
15-8	**A31-A24**	**VME Address Lines.** These added address bits are to make use of the full VME address range when the 2917 becomes a VMEbus master. These bits do not increment during a DMA transfer.
7	**N/U**	**Not used.** Read as a one.
6	**LWD**	**Longword.** This bit must be written to a one because the 2917 is unable to transfer 32-bit data words.
5-0	**AM5-AM0**	**Address Modifier Codes.** These bits must be written with the proper AM Code before a DMA transfer is started.

Command Memory Register

The 2917 Command Memory is an 8K x 16-bit RAM memory. All CAMAC commands are executed from this memory. Data for one or more command lists is written or read as if the memory was a single register. The Memory Address is incremented automatically after each write or read operation. If multiple lists are present, the list to be executed is selected by first setting the Command Memory Address to that list. All lists **must** be CAMAC **"Read"** or **"Write"**, with the following exceptions:

1. Read or write lists can contain dataless CAMAC commands, using Function Codes F(8) through F(15) and F(24) through F(31).

2. Read lists can contain Inline Write operations.

The instructions for the Command List (written as MEM DATA) can be selected from the following:

(a) **Single CAMAC Transfer** – Read, Write, or Control

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
W(1)	0	0	0	0	0	C4	C2	C1	CM 0	TM2 0	TM1 0	QM2 0	QM1 0	WS2	WS1	AD
W(2)	0	0	N16	N8	N4	N2	N1	A8	A4	A2	A1	F16	F8	F4	F2	F1

D16 OFFSET = $62

(b) **Single CAMAC Transfer** – Inline Write (usually in Read list)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
W(1)	0	0	0	0	0	C4	C2	C1	CM 0	TM2 1	TM1 1	QM2 0	QM1 0	WS2	WS1	AD
W(2)	0	0	N16	N8	N4	N2	N1	A8	A4	A2	A1	F16	F8	F4	F2	F1
W(3)	W16	W15	W14	W13	W12	W11	W10	W9	W8	W7	W6	W5	W4	W3	W1	AD
W(4)	X	X	X	X	X	X	X	X	W24	W23	W22	W21	W20	W19	W18	W17

Note: Word #4 is a dummy word for 16-bit Inline Write operation.

D16 OFFSET = $62

(c) **Block CAMAC Transfer**

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
W(1)	0	0	0	0	0	C4	C2	C1	CM 0	TM2 0	TM1 1	QM2 X	QM1 X	WS2	WS1	AD
W(2)	0	0	N16	N8	N4	N2	N1	A8	A4	A2	A1	F16	F8	F4	F2	F1
W(3)	WORD COUNT LOW (The 2's Complement of the number of 16-bit data word to be transferred to or from VMEbus.)															
W(4)	WORD COUNT HIGH (A dummy register, usually set to all ones.)															

D16 OFFSET = $62

(d) **JUMP instruction** – Jump to another command list.

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
W(1)	0	0	0	0	0	0	0	0	CM 1	TM2 1	TM1 0	0	0	0	0	0
W(2)	0	0	0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

D16 OFFSET = $62

(e) **HALT instruction** – End of Command list.

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
W(1)	0	0	0	0	0	0	0	0	CM 1	TM2 0	TM1 0	0	0	0	0	0

D16 OFFSET = $62

Any list can contain a combination of the above instructions, terminated by the HALT instruction.

The Mode word (word #1) in the list instructions is partitioned as follows:

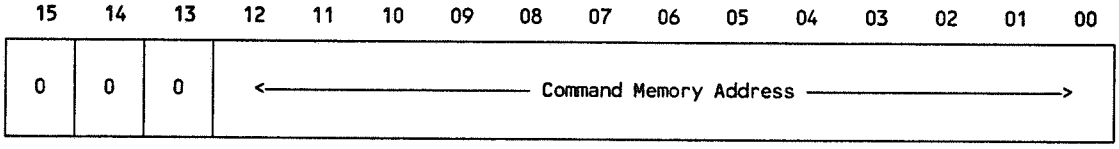
BIT #	MNEMONIC	DESCRIPTION
15-11	N/U	**Not Used.** Don't care.
10-8	C4,C2,C1	**Crate.** This 3-bit field selects one of eight 3922 Crate Controllers.
7	CM	**Command Mode:** 0 = CAMAC transfer 1 = Non-CAMAC instruction. (HALT, etc.)
6,5	TM2,TM1	**Transfer Mode: CM = 0** 0 0 = Single CAMAC transfer 0 1 = Block CAMAC transfer 1 0 = Not Used 1 1 = Single Inline write transfer **Transfer Mode: CM = 1** 0 0 = HALT instruction 0 1 = Not Used 1 0 = JUMP instruction 1 1 = Not Used
4,3	QM2,QM1	**Q-Mode Transfer Type:** 0 0 = Q-Stop, Single, Inline Write 0 1 = Q-Ignore 1 0 = Q-Repeat 1 1 = Q-Scan
2,1	WS2,WS1	**Word Size:** 0 0 = 24-bit CAMAC transfer

0 1 = 16-bit CAMAC transfer
1 0 = Not Used
1 1 = Not Used

0 **AD** **Abort Disable.** Data of zero will Abort a transfer on an error (refer to the Operation Section of this manual for a description of error for each mode). Data of one disables Abort.

Command Memory Address Register

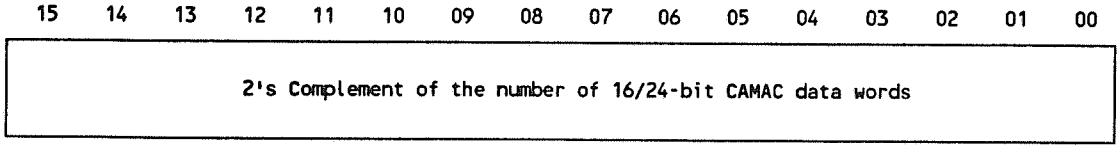
The Command Memory Address (CMA) Register is a 16-bit Write/Read Counter. When loading a new list, this register is loaded with an address that points to the initial word location for that command list. When writing or reading the Command Memory, the address register is auto-incremented after each read or write operation. After a HALT instruction is encountered or the list is aborted, the Command Memory Address Register points to the next list address. A layout of the Command Memory Address Register is given below:



D16 OFFSET = $64

Command Word Count Register

The Command Word Count (CWC) Register is a 16-bit read-only counter which contains the 2's complement of the number of words not yet executed in a CAMAC block transfer read or write operation. When a CAMAC block transfer is executed, this register is loaded by the Command Memory Sequencer with the initial word count from the command list and incremented as transfers occur for that block. When a block transfer is aborted, the "current" word count can be determined by reading this register. Note that the data in this register monitors the last block operation executed not the total word count for all the list operations. A register layout for the Command Word Count Register is given below:

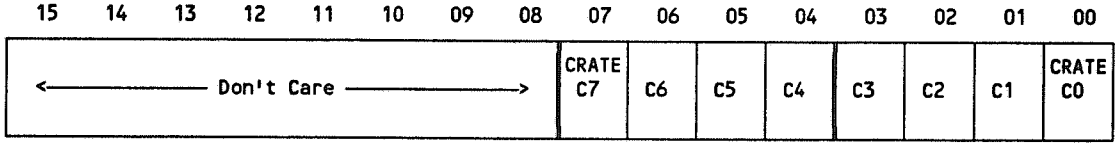


D16 OFFSET = $66

Service Request Register

The Service Request Register (SRR) provides information regarding LAMs in a multicrate system. When this register is read, the 2917 issues a Parallel Poll to all 3922s connected to the bus.

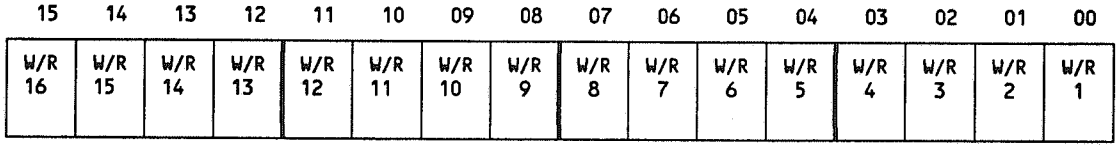
The 8-bits represent the eight possible crate addresses. A true bit (1) in the SRR indicates that its associated crate is asserting its LAM line. If any 3922 is requesting service, the LAM signal (bit #9 of the CSR) is set. A register layout of the Service Request Register is given below:



D16 OFFSET = $68

Data Low Register

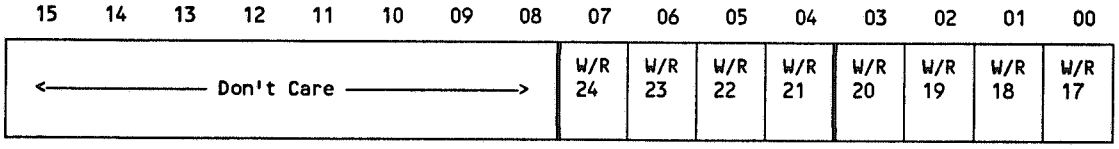
The Data Low Register (DLR) is used to access the lower 16-bits of the 24-bit CAMAC word. During a CAMAC write operation, F(16)-F(23), the contents of the DLR are used to generate the data signals on the CAMAC write data lines W16 through W1. During CAMAC read operations, F(0)-F(7), the DLR is loaded with the data signals present on the CAMAC read data lines R16 through R1. A register description of the Data Low Register is given below:



D16 OFFSET = $6A

Data High Register

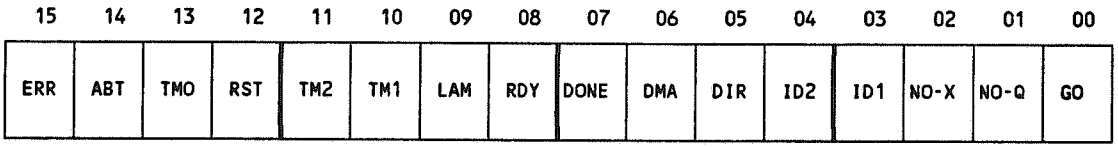
The Data High Register (DHR) is used to access the upper 8-bits of the 24-bit CAMAC word. During a CAMAC write operation, F(16)-F(23), the contents of the DHR are used to generate the data signals on the CAMAC write data lines W24 through W17. During CAMAC read operations, F(0)-F(7), the DHR is loaded with the data signals present on the CAMAC read data lines R24 through R17. A register description of the Data Low Register is given below:



D16 OFFSET = $6C

Control Status Register

The Control Status Register (CSR) is used to start the 2917 operations and monitor the results of those operations. A layout and description of the Control Status Register is given below:



D16 OFFSET = $6E

BIT #	MNEMONIC	DESCRIPTION
15	ERR	**ERROR.** This is a read-only bit which is set when an operation in the Command List terminates with a error. See the Operations section for a description of ERROR for each mode of operation.
14	ABT	**ABORT.** This is a read-only bit which is set when an operation terminates in ERROR and the Abort Disable bit (AD) of the operation was not set.
13	TMO	**TIME OUT.** This is a read-only bit which is set when a timeout occurs during a 2917/3922 bus transaction.
12	RST	**RESET INTERFACE.** This is a write-only bit which resets the 2917 to a power-up state. This bit is read as zero.
11,10	TM2,TM1	**TRANSFER MODE.** These bits indicate the last Transfer Mode type to be executed. Same bits as the Crate/Mode word in the Command List.
9	LAM	**LOOK-AT-ME.** When asserted, indicates that a LAM request for service is pending. Multiple 3922s may assert this bit. The SRR will indicate which 3922 is requesting service. LAM is a read-only bit.
8	RDY	**DATA READY.** This bit, when set to one, indicates that the DLR and DHR are ready to be written to or valid data is ready to be read.
7	DONE	**DONE.** This is a read-only bit which is set when the 2917 has completed or aborted a Command List. It is cleared while the 2917 is executing a list.
6	DMA	**DIRECT MEMORY ACCESS.** This is a write/read bit which, when set, directs the 2917 to transfer data to or from the VME global memory using an on-board DMA controller. The DMA controller will clear this bit when its transfer count is exhausted. When DMA is cleared, data is transferred with the 2917 acting as a slave. This bit is cleared on power-up.

5	**DIR**	**DIRECTION.** This bit directs the flow of data between VMEbus and the CAMAC bus. 0 = CAMAC to VME (CAMAC read) 1 = VME to CAMAC (CAMAC write) Note: This bit is don't care for a Inline Write command or a CAMAC Control command.
4,3	**ID2,ID1**	**INTERFACE ID.** These bits are read as zeros.
2	**NO-X**	**NO-X.** When set, indicates that a CAMAC operation results in a X-response of zero.
1	**NO-Q**	**NO-Q.** When set, indicates that a CAMAC operation results in a Q-response of zero.
0	**GO**	**GO** - when set instructs the 2917 to begin executing the Command List. This bit is reset by a HALT command or Abort Command List.

2917 OPERATING MODES

Slave Single Transfers

The 2917 provides a programmed transfer mode of operation. Under programmed transfer, all transfers to or from the 2917 are done via programmed Short I/O. Programmed transfers may be executed with either 16- or 24-bit CAMAC data words. The size of the data words transferred is determined by the setting of the Word Size bits (bits 2 and 1) in the Mode word in the command list. During a CAMAC write or read operation, the DONE bit in the CSR will not be set until data is written to or read from the data registers. For a 16-bit CAMAC operation, DHR doesn't need to be written or read. A Ready bit (RDY bit 8 in CSR) will indicate if read data is present or if the data registers, DLR and DHR, can be written to. The following describes the sequence necessary to execute CAMAC read, write and control functions using programmed transfers. Note: A write or read list can have one or more single write/read CAMAC commands. A CAMAC control command and/or Inline Write command can also be in a write/read list.

Programmed Transfer Read Operation (F16=0: F8=0)

1. Load Command Memory with single CAMAC command list with the Mode word set to all the proper options. Refer to the CMR for more details.

2. Load the CMA Register to the beginning of the list.

3. Load the CSR with:
   a. DIR (bit 5) to 0.
   b. GO (bit 0) to 1.

4. Poll CSR for the RDY bit (bit 8) until it is set to one or the ERR bit (bit 15) to be set.

5. Read **only** the DLR for 16-bit CAMAC reads. In 24-bit mode, DLR or the DHR, can be read first.

6. Check CSR for DONE to be set. If Done is not set, another read may be pending. Return to step four.

Programmed Transfer Write Operation (F16=1: F8=0)

1. Load Command Memory with single CAMAC command list with the Mode word set to all the proper options. Refer to the CMR for more details.

2. Load the CMA Register to the beginning of the list.

3. Load the CSR with:
   a. DIR (bit 5) to 1.
   b. GO (bit 0) to 1.

4. Poll CSR for the RDY bit (bit 8) to be set to a one.

5. Write **only** the DLR for 16-bit CAMAC writes. In 24-bit mode, DLR or the DHR, can be written first.

6. Check CSR for DONE and/or ERR to be set. If Done is not set, another write may be pending. Return to step four.

Programmed Transfer Control Operation (F16=X: F8=1)

1. Load Command Memory with Control CAMAC command list with the Mode word set to all the proper options. Refer to the CMR for more details.

2. Load the CMA Register to the beginning of the list.

3. Load the CSR with:
   a. DIR (bit 5) is don't care. The DIR bit would be set to the proper value if the command list included write or read commands.

   b. GO (bit 0) to 1.

4. Check CSR for DONE and/or ERR to be set.

The following are definitions for ERROR and ABORT when executing operations in the programmed single command transfer mode.

ERROR = NO-X + NO-Q
ABORT = /AD * ERROR

where

NO-Q is bit 1 of CSR.

NO-X is bit 2 of CSR.
AD is bit 0 of Mode word in Command List.

Slave Block Transfers

The 2917 supports four types of block transfers: Q-STOP, Q-IGNORE, Q-REPEAT, and Q-SCAN. The command list needs to be set-up with the Block mode command in each of these modes. The selection of the Q-Modes are selected in the Mode word (bits 3 and 4) of the Command list. See the CMR for selecting the various options. During a Block transfer, the RDY bit (bit 8) in the CSR, should be monitored. This bit will indicate if Read data is present or if the data registers, DLR and DHR, can written to. Refer to the section on Block Modes for selecting the proper Q-Modes.

The following describes the sequence necessary to execute a CAMAC Block write or read function using programmed transfers. Note: A Block write or read list can have more than one Block write/read CAMAC command. In addition to the Block command, a CAMAC Single, Control, or Inline Write command can also be in the same list.

Programmed Block Transfer Read Operation (F16=0: F8=0)

1. Load Command Memory with single CAMAC command list with the Mode word set to all the proper options. Refer to the CMR for more details.

2. Load the CMA Register to the beginning of the list.

3. Load the CSR with:
   a. DIR (bit 5) to 0.
   b. GO (bit 0) to 1.

4. Poll CSR for the RDY bit (bit 8) until it is set to one or the ERR bit (bit 15) to be set.

5. Read **only** the DLR for 16-bit CAMAC reads. In 24-bit mode, DLR or the DHR, can be read first.

6. Check CSR for DONE to be set. If Done is not set, another read may be pending Return to step four.

Programmed Block Transfer Write Operation (F16=1: F8=0)

1. Load Command Memory with single CAMAC command list with the Mode word set to all the proper options. Refer to the CMR for more details.

2. Load the CMA Register to the beginning of the list.

3. Load the CSR with:
   a. DIR (bit 5) to 1.
   b. GO (bit 0) to 1.

4. Poll CSR for the RDY bit (bit 8) to be set to a one.

5. Write **only** the DLR for 16-bit CAMAC writes. In 24-bit mode, DLR or the DHR, can be written first.

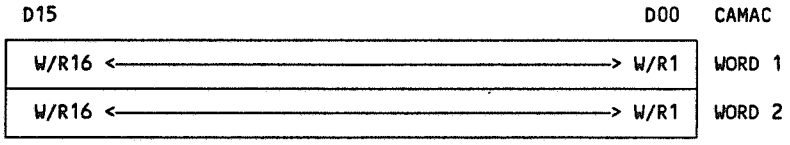
6. Check CSR for DONE and/or ERR to be set. If Done is not set, another write may be pending. Return to step four.

DMA Transfers

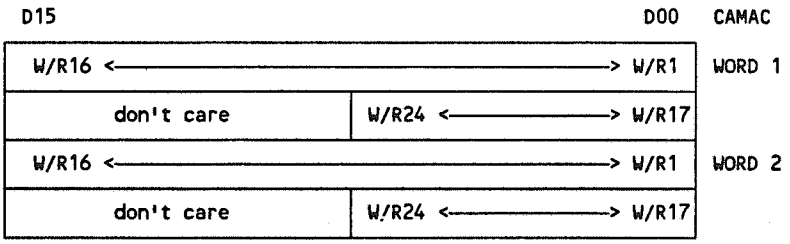
In DMA mode, the 2917 becomes a VME bus Master transferring a single CAMAC word (16 or 24-bit) at a time. DMA mode can be used on a Single and/or Block command list.

Three formats are available for storage of CAMAC data words into global memory. These transfers are only D16 words (D15-D00). Using 16-bit storage format requires one word of global memory for each CAMAC word since only the lower 16-bits of the CAMAC word are used. Using the 24-bit storage format, the full 24-bit CAMAC data word is stored in global memory, requiring two global memory words per CAMAC word. The 24-bit storage format has an option for the order in which the upper 8-bit and the lower 16-bit words are stored. A strap option on the 2917 selects this option with the default set to LOW/HI. This setting will access the lower 16-bit CAMAC word, then the upper 8-bit CAMAC word. The HI/LOW setting is the reverse of the LOW/HI selection. The following diagrams show the three storage formats.

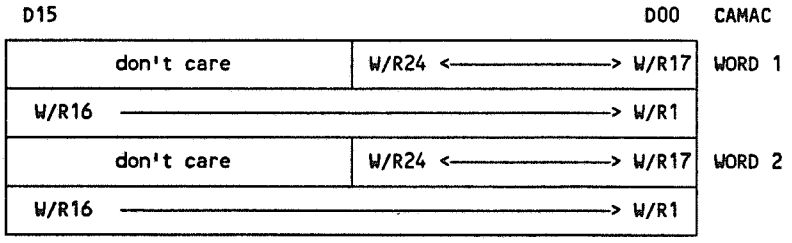
16-Bit Storage Format



24-BIT Storage Format (LOW/HI selected)



24-BIT Storage Format (HI/LOW selected)



The following table shows the various word size selections:

WS2	WS1	WORD SIZE
0	0	24-bits
0	1	16-bits
1	0	8-bits (not used)
1	1	RESERVED

When using the 2917 in a DMA operating mode, the user must load the DMA Transfer count in the DMA control to the number of VME data transfers. The Command list, if using Block mode, must contain the 2's complement of the number of CAMAC words (16- or 24-bits to be transferred. For CAMAC writes DMA transfers, load the MACHI and MACLO with the first address containing the CAMAC write data. For CAMAC read DMA transfers, load the MACHI and MACLO with the address reserved for CAMAC read data. Subsequent CAMAC data words are fetched/stored in the following address space.

The Memory Transfer Count (MTC) and the Command Word Count (CWC) are incremented by the following amounts for the given data word size:

Word Size	MTC	CWC
16-bits	1	1
24-bits	2	1

DMA Single Transfers

The DMA Single Transfer mode is the same thing as Slave Single Transfer mode except the user does not have to poll the RDY bit in the CSR. The DMA controller handles all the hand shaking required to transfer VME data to/from CAMAC. The following describes the sequence necessary to execute Single CAMAC read or write commands in DMA mode.

DMA Single Transfer Write/Read Operation

1. Load Command Memory with single CAMAC command list with the Mode word set to all the proper options. Refer to the CMR for more details.

2. Load the CMA Register to the beginning of the list.

3. Load the MACHI and MACLO with the starting address where data is to be fetched/stored in global memory.

4. Load the AMR register with the proper Address Modifier Codes and the upper eight VME address bits.

5. Load the MTC register with the number of VME data transfers.

6. Load the CSER with zero to clear any pending DMA transfers.

7. Load the DOCR with a data value of 8010 HEX for CAMAC writes and 8090 HEX for CAMAC reads. Set the Cycle Steal Mode and VME word size to D16 and DMA direction.

8. Load the SCCR with data 0080 HEX. This will start the DMA controller chip.

9. Load the CSR with:
   a. DMA (bit 6) to 1.
   b. DIR (bit 5) to 0 for CAMAC read, 1 for CAMAC write.
   c. GO (bit 0) to 1.

10. Poll CSR for the DMA bit (bit 6) to be cleared and the DONE (bit 7) to be set. If only the DMA bit is cleared and the DONE bit is not set, there may be more CAMAC commands in the Command List. If so, repeat steps three through nine. Steps 3 and 4 do not have to be reloaded if the transfer is to start where it ended its last transfer.

11. If Done is set in the CSR, the command list is terminated by a HALT instruction or by an ERROR condition. Check the CSR for any Errors.

    Note: When using the 2917 with interrupts enabled, ignore the step which includes polling for "DONE" and "DMA" since both of these bits generate an interrupt. The "DONE" bit will set the "DONE" interrupt and the "DMA" bit will set the "DMA BUFFER EMPTY" interrupt.

DMA Block Transfers

The DMA Block Transfer mode is the same thing as Slave Block Transfer mode except the user does not have to poll the RDY bit in the CSR. The DMA controller handles all the hand shaking required to transfer VME data to/from CAMAC. The DMA controller can only transfer 64k words at a time. If the user is going to transfer blocks larger than 64k, then the DMA controller will need to be reloaded when the "DMA" bit in the CSR is cleared or if a DMA BUFFER EMPTY interrupt is set. Once the DMA controller is loaded with new values, the CSR must have the DMA bit set to one again. The Command List will continue from where it had stopped. Refer to the section on Block Modes for selecting the proper Q-Modes.
The following describes the sequence necessary to execute Block CAMAC read or write commands in DMA mode.

DMA Block Transfer Write/Read Operation

1. Load Command Memory with Block CAMAC command list with the Mode word set to all the proper options. Refer to the CMR for more details.

2. Load the CMA Register to the beginning of the list.

3. Load the MACHI and MACLO with the starting address where data is to be fetched/stored in global memory.

4. Load the AMR register with the proper Address Modifier Codes and the upper eight VME address bits.

5. Load the MTC register with the number of VME data transfers.

6. Load the CSER with zero to clear any pending DMA transfers.

7. Load the DOCR with a data value of 8010 HEX for CAMAC writes and 8090 HEX for CAMAC reads. This sets the Cycle Steal Mode and VME word size to D16 and DMA direction.

8. Load the SCCR with data 0080 HEX. This will start the DMA controller chip.

9. Load the CSR with:
   a. DMA (bit 6) to 1.
   b. DIR (bit 5) to 0 for CAMAC read, 1 for CAMAC write.
   c. GO (bit 0) to 1.

10. Poll CSR for the DMA bit (bit 6) to be cleared and the DONE (bit 7) to be set. If only the DMA bit is cleared and the DONE bit is not set, there may be more CAMAC commands in the Command List. If so, repeat steps three through nine. Steps 3 and 4 do not have to be reloaded if the transfer is to start where it ended its last transfer.

11. If Done is set in the CSR, the command list has be terminated by a HALT instruction or by an ERROR condition. Check the CSR for any Errors.

    Note: When using the 2917 with interrupts enabled, ignore the steps which include polling for "DONE" and "DMA" since both of these bits generate an interrupt. The "DONE" bit will set the "DONE" interrupt and the "DMA" bit will set the "DMA BUFFER EMPTY" interrupt.

BLOCK MODES

Q-STOP Block Transfer

To select the Q-STOP block transfer mode, set QM2 to 0 and QM1 to 0 in the Mode word of the Command List. During a Q-STOP block transfer operation, the CAMAC command specified in the second Command List word is repeated until a Q-RESPONSE of zero is received or the command word count is exhausted. The Block transfer is also terminated if an "ERROR" (bit 15 in CSR) or "ABORT" (bit 14 in CSR) is generated. The following equations describe "ERROR" and "ABORT" for the Q-STOP block transfer mode.

ERROR = NO-X + NO-Q
ABORT = /AD * ERROR

Q-IGNORE Block Transfer

To select the Q-IGNORE block transfer mode, set QM2 to 0 and QM1 to 1 in the Mode word of the Command List. During a Q-IGNORE block transfer operation, the CAMAC command specified in the second Command List word is repeated until the command word count is

exhausted. The Block transfer is also terminated if an "ERROR" or "ABORT" is generated. The following equation describes "ERROR" and "ABORT" for the Q-IGNORE block transfer mode.

ERROR = NO-X
ABORT = /AD * ERROR

Q-REPEAT Block Transfer

To select the Q-REPEAT block transfer mode, set QM2 to 1 and QM1 to 0 in the Mode word of the Command List. During a Q-REPEAT block transfer operation, the CAMAC command specified in the second Command List word is repeated for each data word until a Q-RESPONSE of one is received. A Q-RESPONSE of one causes either new write data to be fetched or read data to be stored. The command is repeated for each data word until the command word count is exhausted. If Q=1 response is not obtained within 60 milliseconds, the 3922 crate controller will terminate block transfer and the "ERROR" or "ABORT" bit is generated. The following equations describe "ERROR" and "ABORT" for the Q-REPEAT block transfer mode.

ERROR = NO-X + NO-Q
ABORT = /AD * ERROR

Q-SCAN Block Transfer

To select the Q-SCAN block transfer mode, set QM2 and QM1 to 1 in the Mode word of the Command List. During a Q-SCAN block transfer, the 3922 uses the Q-RESPONSE from the previous command to determine the station number (N) and subaddress (A) for the next operation. A Q-RESPONSE of zero indicates that the last valid subaddress of the current station number has to be accessed. The 3922 responds to a Q=0 by resetting the subaddress, incrementing the station number, and continuing the scan. A Q-RESPONSE of one indicates that the last command was executed to a valid CAMAC address. The 2917/3922 responds to a Q=1 by either storing the read data or fetching new write data. When a Q=1 response is obtained, the 3922 updates the CAMAC address as follows: the subaddress is incremented or, if the subaddress is 15, it is reset to zero, and the station number is incremented by one. If due to programming error, the 3922 increments beyond station number 23, the block transfer will terminate with an "ERROR" or "ABORT". The following equation is the definition of "ERROR" and "ABORT" when using the Q-SCAN mode operation.

ERROR = NO-Q
ABORT = /AD * ERROR

2917/3922 INTERCONNECTION BUS

The bus interface is a 40-wire twisted pair ribbon cable from the rear of the 3922, wired in a loop-thru basis from the 2917, through the 3922's provided in a particular system. Balanced RS-485 signalling is used between the 2917 and 3922s to enhance performance.

The interconnection bus is a multiplexed 8-bit address/data bus. Before data transfers occur on the bus, the 2917 enables a header byte which enables one of eight possible 3922s to either receive or transmit data. Each 3922 is preset to a unique crate address via front-panel switch

which is used to compare with the incoming header byte. If a 3922 recognizes its crate address, it prepares for the data transfer portion of the cycle. An information bus timeout (INFO TMO) will occur if a header byte is asserted and the addressed 3922 does not respond within seven microseconds.

The protocol for single-CAMAC-word transfers, depending upon the word size selection in the header byte (8, 16 or 24 bit), is as follows:

8-bit CAMAC word size	ADDRESS/DATA LO
16-bit CAMAC word size	ADDRESS/DATA LO/DATA MID
24-bit CAMAC word size	ADDRESS/DATA LO/DATA MID/DATA HI

Note that the ADDRESS is transmitted by the 2917, while DATA is transmitted by either the 2917 or 3922, depending upon the data direction (read or write operation).

For multiple-CAMAC-word transfers, a single ADDRESS header is followed by the data as shown:

ADDRESS/CAMAC WORD 1/CAMAC WORD 2/. . ./CAMAC WORD N

Note that the CAMAC word contains one, two, or three bytes of data, depending upon the word-size selection in the ADDRESS header. Dataless CAMAC operations (F8=1) involve only the ADDRESS header, with no data following.

The header byte is shown below:

07	06	05	04	03	02	01	00
ADRS 4	ADRS 2	ADRS 1	WS2	WS1	M4	M2	M1

BITS **MEANING**

7-5 Binary 3922 Crate Address - Selects Crate Address 0-7

4-3 Word Size

WS2	**WS1**	**Word Size**
0	0	24-bit
0	1	16-bit
1	0	8-bit
1	1	RESERVED

2-0 Binary Operations Mode

M4	M2	M1	Operation
0	0	0	Q-STOP Operation
0	0	1	IGNORE-Q Operation
0	1	0	Q-REPEAT Operation
0	1	1	Q-SCAN Operation
1	0	0	SINGLE Operation
1	0	1	RESERVED
1	1	0	NAF1 (Lower byte of NAF)
1	1	1	MAF2 (Upper byte of NAF)

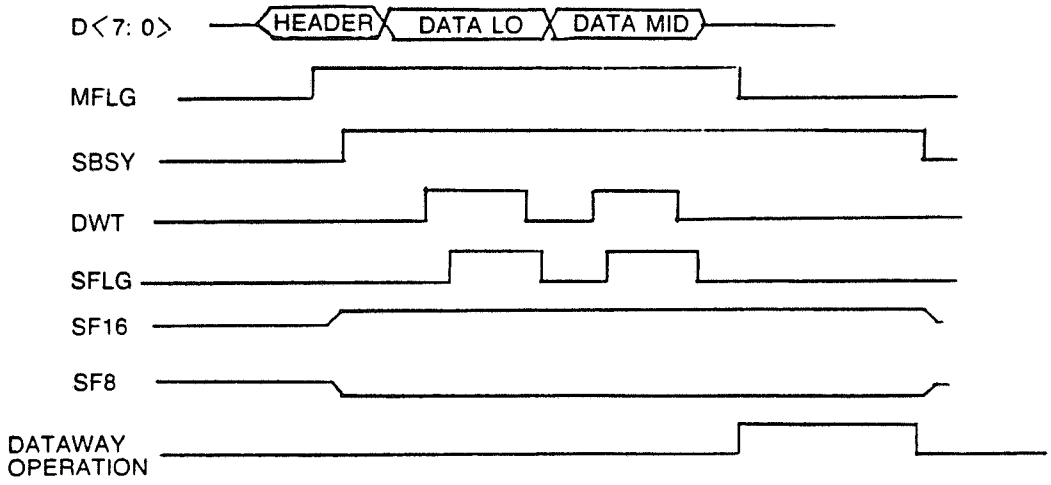
The following shows the signals present on the interconnection bus along with a brief description of their function:

Dataway Q	DWG	When asserted, indicates that the current Dataway operation gave a Q=1 response: Q=0 if negated.
Dataway X	DWX	When asserted, indicates that the current Dataway operation gave a X=1 response: X=0, if negated.
Data Read	DRD	When asserted, indicates that the 3922 should supply read data.
Data Write	DWT	When asserted, indicates that write data is available to the 3922.
Information Bus	D<7:0>	Used for address, R/W data, and parallel poll information.
Look-At-Me	LAM	When asserted, indicates that a LAM request for service is pending. Multiple 3922s may assert this line.
Master Busy	MBSY	When asserted, indicates that Address information can be strobed into the 3922. This is negated to indicate "End of Transfer" or to abort the present transfer.
Parallel Poll	Poll	When asserted, indicates that a Parallel Poll operation is in progress.
Slave Busy	SBSY	When asserted, indicates that the addressed 3922 has not yet completed a data transaction. When negated, indicated that the 3922 has completed the transaction. This line can be negated by the 3922 to abort the present transfer.
Slave Flag	SFLG	Asserted by a 3922 to indicate a Reply Response from either a DRD or DWT.

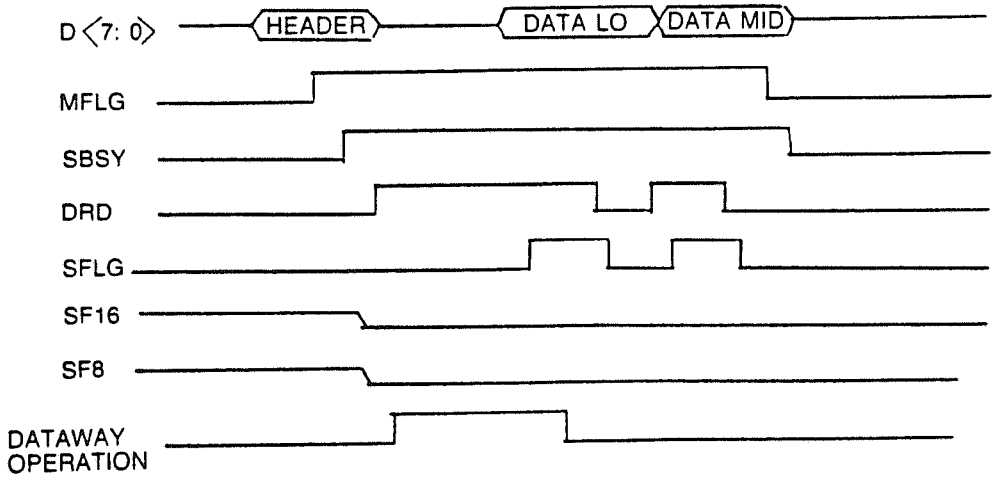
Slave F8	SF8	When asserted, indicates that F8=1 in the "CURRENT" 3922.
Slave F16	SF16	When asserted, indicates that F16-1 in the "CURRENT" 3922.

The following diagram shows a typical write and read sequence over the interconnection bus.

16-BIT CAMAC WRITE OPERATION



16-BIT CAMAC READ OPERATION



MODEL 3922 CRATE CONTROLLER

FEATURES AND OPERATION

The Model 3922 meets all the requirements of IEEE Standard 583 for CAMAC crate controllers. It forms the communications link between the 2917 VMEbus Interface and the I/O modules in the CAMAC crate. When used with the 2917, the 3922 performs a wide variety of CAMAC Commands to modules in the crate. The 2917/3922 combination supports program transfers as well as high-speed DMA operations. The 3922 also contains several internal registers that can be read or written at pseudo-address N(30). As a main crate controller, the 3922 supports auxiliary crate controllers as specified by IEEE Standard 675. The 3922 can also be easily field-changed to become an auxiliary crate controller.

One to eight 3922s can be connected to a single 2917 VMEbus Interface. Each 3922 includes a front-panel switch that allows address selection from 0 to 7. Multiple 3922s are interconnected on a loop-through basis and the last 3922 can be up to 500 feet from the 2917 PC Interface. The last 3922 on the bus will use the terminator card (provided with the 3922) in connector DB2. (See Figure 1 below)

To achieve higher speed DMA data transfers, the 3922's Read and Write Data Registers can be double-buffered. This allows a Dataway cycle to be requested and executed while data is transferred across the 2917/3922 bus. This buffering produces a side effect during Read operations. When the 2917 terminates a DMA transfer, the 3922 has already performed one or more Dataway operations to fill its buffer. This can cause a problem in some instances. Two operating states are provided for DMA Read operations. By clearing the bit in the 3922 Status Register, the single-buffer mode performs a Read operation only when the 2917 requests data. By setting this bit, double-buffer mode is selected. Double buffering results in some increase in throughput. Single-buffer mode should be used for Q-REPEAT and Q-SCAN operations or for other Read transfers where extra Dataway cycles result in lost data.

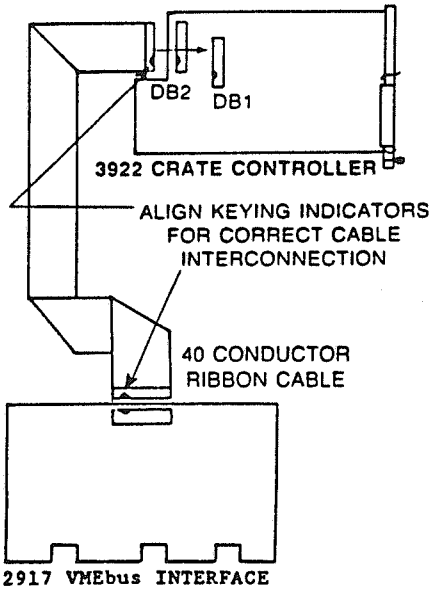


Figure 1 - 3922 K-Bus Connection

Model 2917-Z1A

Address Selection

A front-panel, thumb-wheel switch allows you to select each 3922's address. Any address from 0 to 7 can be selected. However, all 3922s connected to single 2917s should have this switch set to different addresses.

Registers

CAMAC Commands are performed to modules within the crate using Station address 1 through 23. Internal registers in the 3922 are accessed in the same manner as module registers by pseudo-address N(30). There are five 3922 internal commands. These are: Read Status, Read LAM Pattern, Read LAM Mask, Write Status, and Write LAM Mask. The N(30) Commands are shows here:

Command	Q-Response	Action
F(1)A(0)	ONLINE	Read Status
F(1)A(12)	ONLINE	Read LAM Pattern
F(1)A(13)	ONLINE	Read LAM Mask
F(17)A(0)	ONLINE	Write Status
F(17)A(13)	ONLINE	Write LAM Mask

Notes: 1. N=30 for all commands.
2. X=1 is returned for all valid commands.
3. Q=0 is returned if the front-panel on-line switch is in an "off-line" position. The only command that will be executed when the 3922 is off-line is N(30)F(1)A(0), Read Status Register.

Status Register N(30)F(17)A(0), N(30)F(1)A(0)

BIT	WRITE OPERATION	READ OPERATION
1	Generate Z	0
2	Generate C	0
3	Set Inhibit	Inhibit (Crate Controller Register State)
4	0	0
5	0	0
6	0	0
7	0	Dataway Inhibit
8	Double-Buffer Mode	Double-Buffer Mode
9	Enable Service Request	Service Request Enabled
10	Set Internal L24	Internal L24 Set
11	0	0
12	0	0
13	0	0
14	0	Front-Panel Switch Off-line
15	0	Write Buffer Full
16	0	Selected LAM Present
17-24	0	0

A detailed description of the bits in the 3922 Status Register is provided here:

MODEL 3922 CRATE CONTROLLER

BIT	DESCRIPTION
1	By setting this bit with the 3922 on-line, the crate controller executes a CAMAC Initialize (Z) operation. This bit is always read back as zero.
2	By setting this bit with the 3922 on-line, the crate controller executes a CAMAC Clear (C) operation. This bit is always read back as zero.
3	By setting this bit, the 3922 asserts the Dataway Inhibit line. With the 3922 asserting the Inhibit line, both bits 3 and 7 will return a value of **ONE** when the Status Register is read.
7	This Read-only bit indicates the state of the Inhibit line on the CAMAC Dataway. Note that other modules such as the 3655 can assert the Inhibit line.
8	By setting this bit, all DMA Read operations are executed in double-buffer mode. In this mode, a Dataway Read operation is executed and transferred to the 3922 buffer as soon as a DMA transfer request is made. This double-buffering allows extra Dataway operations to occur when the block is terminated. If this bit is cleared (single-buffer mode), a DMA Dataway Read operation is executed only after the 2917 VMEbus Interface requests data. Single-buffered mode should be used for Q-REPEAT and Q-SCAN operations or other transfers where extra Dataway operations will result in lost data. This bit is Read/Write.
9	By setting this bit, a Service Request is made to the 2917 whenever a module Look-At-Me (LAM) is pending **AND** that LAM is enabled in the LAM Mask Register. This bit is Read/Write.
10	This bit sets the internal L24 signal. The L24 signal can be used for software and hardware testing associated with the Service Request. This bit is Read/Write.
14	This Read-only bit indicates the status of the front-panel, on-line switch. With the 3922 in the off-line state, only the Status Register can be read. No other commands will be executed by the 3922.
15	This a Read-only bit which is set when a Q-STOP operation is terminated due to a Q=0 response and the 3922 contains a CAMAC write data word in its buffer.
16	This Read-only indicates if a selected LAM is present in the CAMAC crate. This "SLP" condition is **TRUE** only when one or more LAM requests (L1 to L24) are asserted **AND** the LAM Mask bit(s) associated with the LAM request(s) are **TRUE**.

LAM Register N(30)F(1)A(12)

This is a 24-bit register which indicates the present state of all LAM requests in the CAMAC crate. Each bit corresponds to the appropriate Station in the crate {L(9) is associated with N(9), etc.}.

LAM Mask Register N(30)F(1)A(13), N(30)F(17)A(13)

The LAM Mask Register is used to select those LAM requests from modules that cause a Request-for-Service (Selected LAMs Present) to be forwarded to the 2917. For example, if Bits 2, 3 and 8 are **TRUE** in the LAM Mask Register and all other bits are **FALSE**. only L(2), L(3) or L(8) causes a Service Request.

3922 FRONT PANEL

Switches

Crate Address	A thumb-wheel switch which selects the crate address for the 3922. The address ranges from 0 to 7.
On-Line	Manually places the 3922 on-line or off-line. No Dataway operations are executed when the 3922 is off-line.
Z/C	Performs a manual Dataway Z or C to the CAMAC crate. Note that the 3922 must be off-line for this switch to have any effect.

LEDs

BUSY	Flashes whenever the 3922 is performing a Dataway operation.
NO-Q	The last CAMAC operation, including an N(30) command, sets this LED "ON" when an Q=0 condition occurs.
NO-X	The last CAMAC operation, including an N(30) command, sets this LED "ON" when an X=0 condition occurs.
Inhibit	This LED is "ON" whenever the Dataway Inhibit line is asserted.
SLP	This LED is "ON" when a selected LAM is present.

LEMOs

BUSY	A low-true TTL signal when Dataway BUSY is TRUE.
Request	Part of the auxiliary crate controller (ACC) protocol. Note that Request MUST be patched to Grant In for a main crate controller (even when no ACCs are used).

Grant In	Part of the ACC protocol. Patched to Request or Grant Out from a higher priority crate controller.
Grant Out	Part of the ACC protocol. Patched to Grant In from a lower priority crate controller.

INSTALLING THE CRATE CONTROLLER

Operating Option Selections

Determine if the 3922 is to be used as a main crate controller (in a stand-alone system) or as an auxiliary crate controller (with another computer being the main host for the crate). The 3922 is shipped from the factory as a main crate controller. If it is to be used as such, proceed to **Installing the 3922**.

If the 3922 is to be used as an auxiliary crate controller, remove the resistor packs and the resistors from both the left and right (A and B) boards of the 3922. Otherwise, proceed to **Installing the 3922**. To convert the 3922 from a main to an auxiliary crate controller, perform the following operations:

1. Remove the rear panel from the 3922. Retain the screws and PC finger insulation.

2. Remove the two screws that hold the A board to the front panel (the left board when viewed from the front panel).

3. Gently separate the A board from the B board. It might be helpful to loosen **slightly** the two screws that hold the B board to the front panel.

4. On the B board, remove the nine resistor packs located by the Dataway fingers. Also, remove the 100 Ω resistor located between IC locations J and K.

5. On the A board, remove the six resistor packs that are loaded in sockets located on the left-hand side of the PC board. These are marked RN1-RN6.

6. Store the resistor packs for future use.

7. Reassemble 3922 crate controller.

Installing the 3922

Locate the 3922 Crate Controller. Locate the free end of the bus cable from the 2917. All main crate controllers must be inserted into the right-hand stations of the crate (slots 24 and 25 of a full-size crate). Auxiliary crate controllers can be placed in any available stations except the main station. Starting from the rear of the crate, "thread" the bus cable through the rear I/O opening in the area where the 3922 will be placed. **BE SURE THAT POWER TO THE CRATE IS "OFF".**

Referring to Figure 2, note the location of the data-bus connectors, DB1 and DB2, on the A board and the Auxiliary Controller Bus (ACB) connector on the B board. With the 3922 facing left side up, connect the bus cable from the 2917 to Connector DB1. Match the key indicator (arrow) on the 3922 connector to the cable connector. This key indicates Contact 1 and should be facing toward the bottom of the module. If using only one 3922, connect the terminator to Connector DB2. Note that the terminator should also have its key indicator toward the bottom of the module. If two or more 3922s are used, find the bus cable for interconnecting a 3922 with the next one on the bus. Thread this cable through the rear I/O opening in the crate and connect it to Connector DB2 (with the key toward the bottom of the module). Repeat this operation for all crates on the bus. The last 3922 on the bus will use the terminator in Connector DB2.

Slide the 3922 into the appropriate slots in the crate. Be sure that the bus cable does not "snag". When the 3922 is almost fully inserted, tighten the jackscrew until it is fully seated.

Front Panel Set-up

Set the front-panel, thumb-wheel, address switch to the desired address. This is generally "1" for a single-crate system. In multicrate systems, verify that no other 3922 has the same address. The order of addresses need not bear any relationship to the physical order of the crates on the bus. Connect the Request, Grant In, and Grant Out LEMO connectors as appropriate for the system.

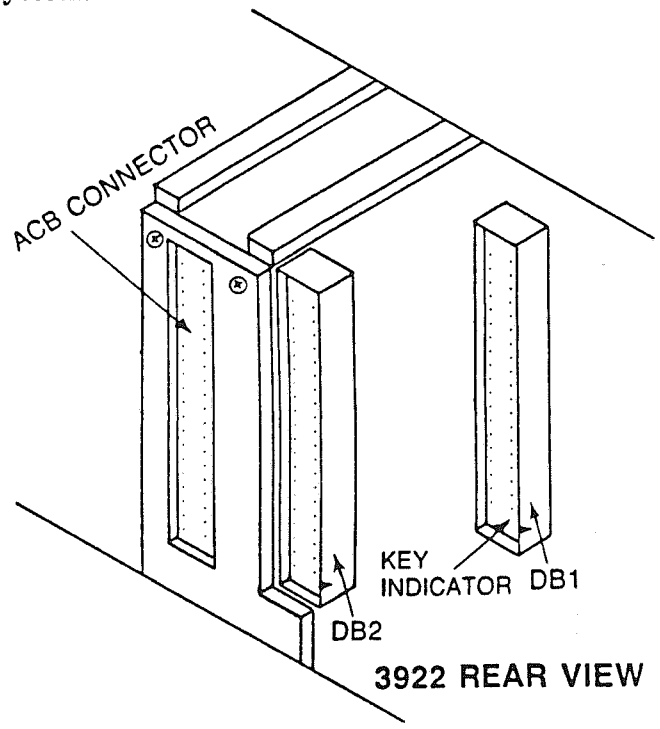


Figure 2 - Connector Locations on the 3922

Note that, in a single-controller system, Request MUST be connected to Grant In before the 3922 can function. Power cannot be applied to the CAMAC crate(s).

2917 REGISTER LAYOUT

Channel Status/Error Register (CSE) D16 OFFSET = $00

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
COC 0=NO 1=YES	N/U	NDT 0=NO 1=YES	ERR 0=NO 1=YES	ACT 0=NO 1=YES	N/U	N/U	RDY	N/U	N/U	N/U	ERROR CODE				
	0				0	0	1	0	0	0					

Device/Operation Control Register (DOC) D16 OFFSET = $04

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
EXT REQ MODE	N/U	N/U	N/U	OCR BITS (5:4)	N/U	N/U	N/U	DIR	N/U	OPERAND SIZE		N/U	N/U	N/U	N/U
	0	1	1		0	0	0		0			0	0	1	0

Sequence/Channel Control Register (SCC) D16 OFFSET = $06

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
NOT USED								START	N/U	N/U	SOFT ABORT	INT ENA	N/U	N/U	N/U
0	0	0	0	0	1	0	0		0	0			0	0	0

Memory Transfer Count Register (MTC) D16 OFFSET = $0A

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Bit 15 <— COUNT HIGH —> Bit 08								Bit 07 <— COUNT LOW —> Bit 00							

Memory Address Counter High (MACHI) D16 OFFSET = $0C

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
A31 <— COUNT HIGH —> A24								A23 <— COUNTER MID-HIGH —> A16							

Memory Address Counter Low (MACLO) D16 OFFSET = $0E

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
A15 <— COUNT MID-LOW —> A08								A07 <— COUNTER LOW —> A00							

LAM Interrupt Control Register D16 OFFSET = $40
Done Interrupt Control Register D16 OFFSET = $42
DMA Buffer Empty Interrupt Control Register D16 OFFSET = $44
List Abort Interrupt Control Register D16 OFFSET = $46

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
<— Not Used —>								FLAG BIT	FLG AUTO-CLR	VEC-TOR	INT ENA	INT AUTO CLR	IRQ LEVEL 2	IRQ LEVEL 1	IRQ LEVEL 0

LAM Interrupt Vector Register D16 OFFSET = $48
Done Interrupt Vector Register D16 OFFSET = $4A
DMA Buffer Empty Interrupt Vector Register D16 OFFSET = $4C
List Abort Interrupt Vector Register D16 OFFSET = $4E

15 ← 08	07	06	05	04	03	02	01	00
← Not Used →	Bit 07 ←	Interrupt Vector						→ Bit 00

Address Modifier Register (AMR) D16 OFFSET = $60

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
A31	A30	A29	A28	A27	A26	A25	A24	N/U	LWD	AM5	AM4	AM3	AM2	AM1	AM0

Command Memory Register (CMR) D16 OFFSET = $62

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1

Command Memory Address Register (CMA) D16 OFFSET = $64

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	← Command Memory Address →												

Command Word Count Register (CWC) D16 OFFSET = $66

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
2's Complement of the number of 16/24-bit CAMAC data words															

Service Request Register (SRR) D16 OFFSET = $68

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
← Don't Care →								CRATE C7	C6	C5	C4	C3	C2	C1	CRATE C0

Data Low Register (DLR) D16 OFFSET = $6A

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
W/R 16	W/R 15	W/R 14	W/R 13	W/R 12	W/R 11	W/R 10	W/R 9	W/R 8	W/R 7	W/R 6	W/R 5	W/R 4	W/R 3	W/R 2	W/R 1

Data High Register (DHR) D16 OFFSET = $6C

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
← Don't Care →								W/R 24	W/R 23	W/R 22	W/R 21	W/R 20	W/R 19	W/R 18	W/R 17

Control Status Register (CSR) D16 OFFSET = $6E

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ERR	ABT	TMO	RST	TM2	TM1	LAM	RDY	DONE	DMA	DIR	ID2	ID1	NO-X	NO-Q	GO

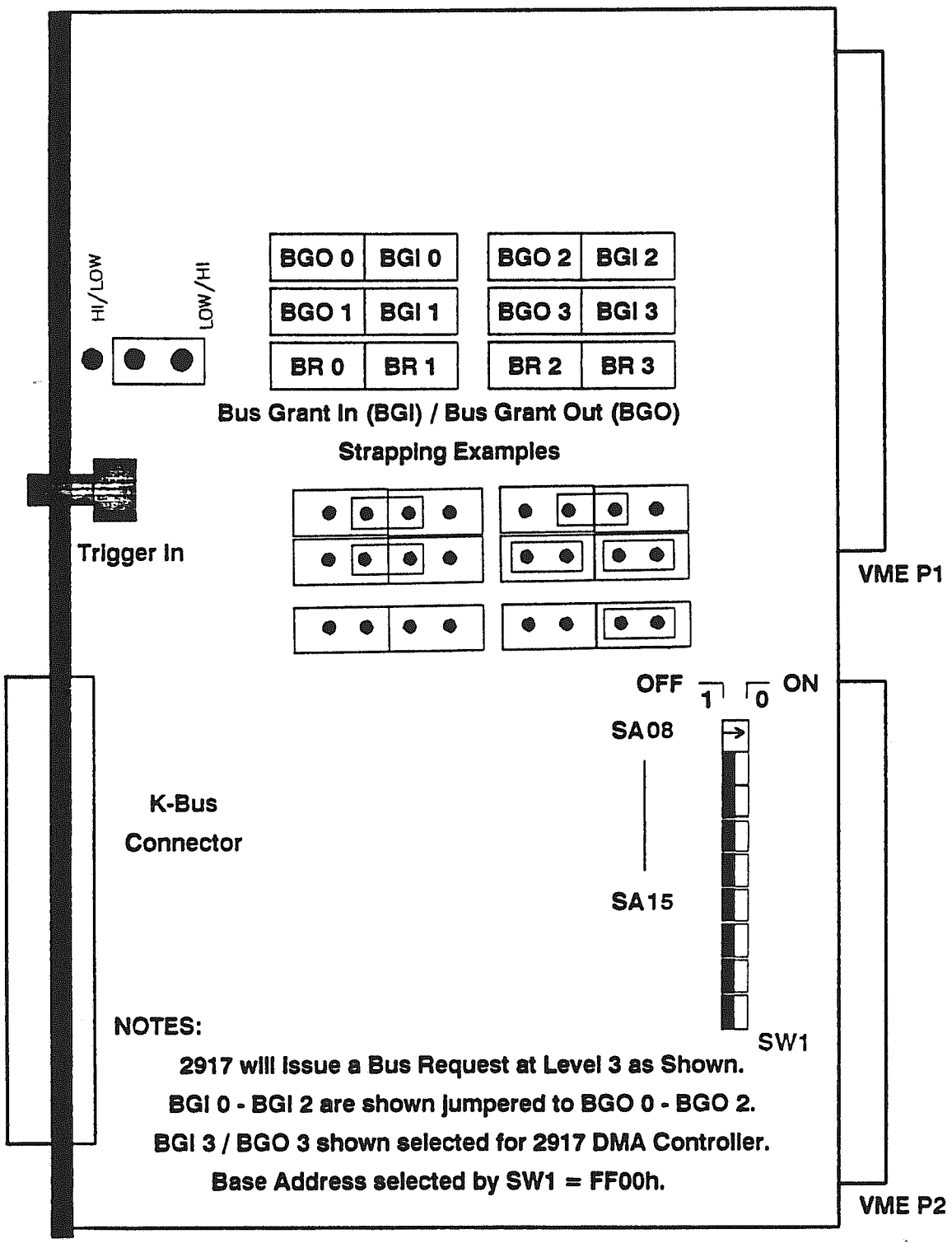


FIGURE 3 - 2917 STRAP AND SWITCH LOCATIONS

APPENDIX A

MOTOROLA MC68153 BUS INTERRUPTER CHIP

MOTOROLA
SEMICONDUCTORS
P.O. BOX 20912 • PHOENIX, ARIZONA 85036

MC68153

Advance Information

BUS INTERRUPTER MODULE

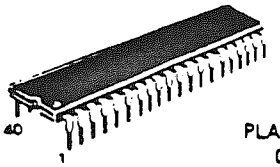
The bipolar LSI MC68153 Bus Interrupter interfaces a microcomputer system bus to multiple slave devices requiring interrupt capabilities. It handles up to 4 independent sources of interrupt requests and is fully programmable.

- VERSAbus/VMEbus Compatible
- MC68000 Compatible
- Handles 4 Independent Interrupt Sources
- 8 Programmable Read/Write Registers
- Programmable Interrupt Request Levels
- Programmable Interrupt Vectors
- Supports Interrupt Acknowledge Daisy Chain
- Control Registers Contain Flag Bits
- Single +5.0 Volt Supply
- Total Power Dissipation = 1.5 W Typical
- Temperature Range of 0°C to 70°C
- Chip Access Time = 200 ns Typical with 16 MHz Clock
- 40-Pin Dual-In-Line Package

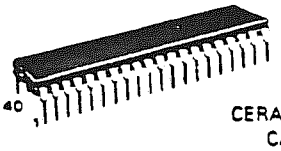
TTL

BUS INTERRUPTER MODULE

ADVANCED LOW POWER SCHOTTKY

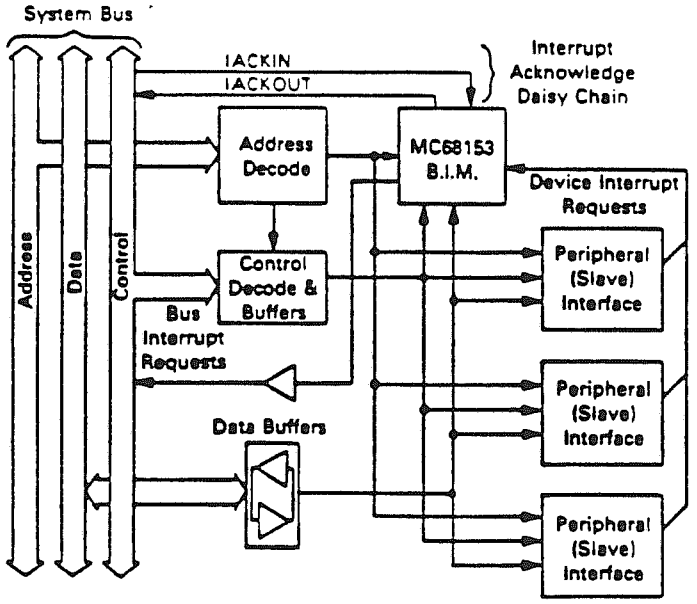


P SUFFIX
PLASTIC PACKAGE
CASE 711-03



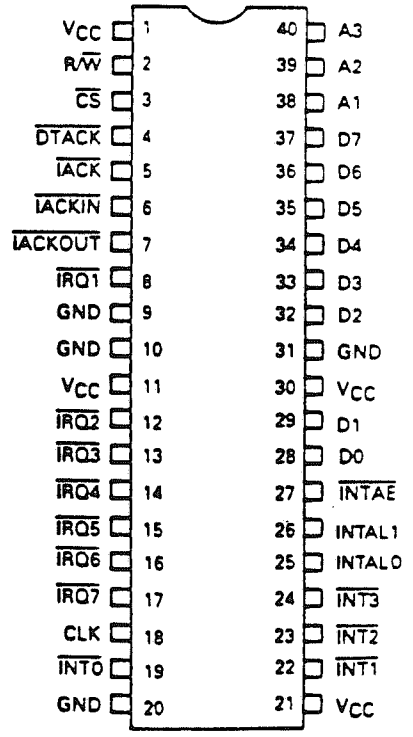
L SUFFIX
CERAMIC PACKAGE
CASE 734-04

FIGURE 1 — MC68153 SYSTEM BLOCK DIAGRAM



VERSAbus is a trademark of Motorola.

PIN ASSIGNMENTS



ADI-1057
(Replacing NP-151)

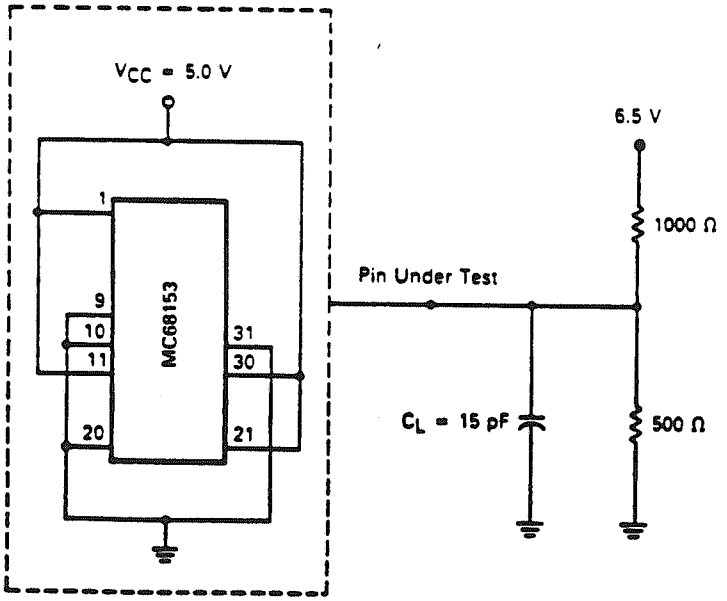
ABSOLUTE MAXIMUM RATINGS (Beyond which useful life may be impaired.)

Parameter	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.5 to +7.0	V
Input Voltage	V_{in}	−0.5 to +7.0	V
Input Current	I_{in}	−30 to +5.0	mA
Output Voltage	V_{out}	−0.5 to +5.5	V
Output Current	I_{OL}	Twice Rated I_{OL}	mA
Storage Temperature	T_{stg}	−65 to +150	°C
Junction Operating Temperature	T_J	−55 to +175	°C

DC ELECTRICAL SPECIFICATIONS (V_{CC} = 5.0 V ±5%, T_A = 0°C to 70°C)

Parameter	Symbol	Min	Max	Unit	Test Conditions
High Level Input Voltage	V_{IH}	2.0	—	V	
Low Level Input Voltage	V_{IL}	—	0.8	V	
Input Clamp Voltage	V_{IK}	—	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA
High Level Output Voltage(1)	V_{OH}	2.7	—	V	V_{CC} = MIN, I_{OH} = −400 μA
Low Level Output Voltage	V_{OL}	—	0.4	V	V_{CC} = MIN, I_{OL} = 8.0 mA
Output Short Circuit Current(2)	I_{OS}	−15	−130	mA	V_{CC} = MAX, V_{OUT} = 0 V
High Level Input Current	I_{IH}	—	20	μA	V_{CC} = MAX, V_{IN} = 2.7 V
Low Level Input Current	I_{IL}	—	−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V
Supply Current	I_{CC}	225	385	mA	V_{CC} = MAX
Output Off Current (High)	I_{OZH}	—	20	μA	V_{CC} = MAX, V_{OUT} = 2.4 V
Output Off Current (Low)	I_{OZL}	—	−20	μA	V_{CC} = MAX, V_{OUT} = 0.4 V

AC TEST CIRCUIT — AC Testing of All Outputs



NOTES:
1. Not applicable to open-collector outputs.
2. Not more than one output should be shorted at a time for longer than one second.
3. $\overline{CS}$ Low to CLK High (Setup Time) of 15 ns Min must be observed.
4. IACK Low to CLK High and $\overline{IACKIN}$ Low to CLK High (Setup Times) of 15 ns Min must be observed.
5. See Table 1 for additional performance guidelines.

AC ELECTRICAL SPECIFICATIONS (V_{CC} = 5.0 V ±5%, T_A = 0°C to 70°C)

Parameter	Test Number[5]	Max (ns)
CLK High to Data Out Valid (Delay)[3]	1	55
CLK High to $\overline{DTACK}$ Low (Delay)[3]	2	40
$\overline{CS}$ High to $\overline{DTACK}$ High (Delay)	3	35
CLK High to Data Out Valid (Delay)[4]	4	55
CLK High to $\overline{INTAE}$ Low (Delay)[4]	5	40
$\overline{IACK}$ High to Data Out High Impedance (Delay)	6	60
$\overline{IACK}$ High to $\overline{DTACK}$ High (Delay)	7	45
$\overline{CS}$ High to Data Out High (Delay)	8	45
$\overline{CS}$ High to $\overline{IRQ}$ High (Delay)	9	60
$\overline{IACK}$ High to $\overline{INTAE}$ High (Delay)	10	35

GENERAL DESCRIPTION

The MC68153 Bus Interrupter Module (BIM) is designed to serve as an interrupt requester for peripheral devices in a microcomputer system. Up to 4 independent devices can be interfaced to the system bus by the MC68153. Intended for asynchronous master/slave bus operation, the BIM is compatible with VERSAbus, VMEbus, MC68000 device bus, and other system buses. Figure 1 shows a block diagram of a typical configuration. In this example, three peripheral devices (bus slaves) are connected to the system data bus. Each of these devices could be parallel I/O, serial I/O, or some other function. An interrupt request from any device is routed to the MC68153, and the BIM handles all interface to the system bus. It generates a bus interrupt request as a result of the device interrupt request. When the system interrupt handler or processor responds with an interrupt acknowledge cycle, the MC68153 can answer supplying an interrupt vector and handling all timing.

The functional block diagram of the MC68153 is shown in Figure 2. The device contains circuitry to accept four separate interrupt sources ($\overline{INT0}$ – $\overline{INT3}$). Interface to the system bus includes generation of bus interrupt requests ($\overline{IRQ1}$ – $\overline{IRQ7}$), response to a bus interrupt acknowledge cycle (either supplying a vector or passing on a daisy chain signal), and releasing the bus interrupt request signal at the proper time. The BIM has flexibility provided by eight programmable read/write registers. Four 8-bit vector registers (VR0 – VR3) contain status/address information and supply a byte vector in response to an interrupt acknowledge cycle for the corresponding interrupt source. Four other 8-bit control registers (CR0 – CR3) contain information that oversees operation of the interrupt circuitry. The control information is programmable and includes interrupt request level and interrupt enable and disable. Also contained in the control registers are flag-bits. These flags are useful for task coordination, resource management, and interprocessor communication.

SIGNAL DESCRIPTION

Throughout the data sheet, signals are presented using the terms asserted and negated independent of whether the signal is asserted in the high voltage or low voltage state. Active low signals are denoted by a superscript bar.

BIDIRECTIONAL DATA BUS — D0 – D7

Pins D0 – D7 form an 8-bit bidirectional data bus to/from the system bus. These are active high, 3-state pins.

ADDRESS INPUTS — A1 – A3

These active high inputs serve two functions. One function is to select one of the eight possible registers during a read or write cycle. Secondly, during an interrupt acknowledge A1 – A3 show the level of interrupt being acknowledged, and the BIM uses these to determine if a match exists with an internal level.

CHIP SELECT — $\overline{CS}$

$\overline{CS}$ is an active low input used to select the BIM's registers for the current bus cycle. Address strobe, data strobe, and appropriate address bits must be included in the chip select equation.

READ/WRITE — R/$\overline{W}$

The R/$\overline{W}$ input is a signal from the system bus used to determine if the current bus cycle is a read (high) or write (low).

DATA TRANSFER ACKNOWLEDGE — $\overline{DTACK}$

$\overline{DTACK}$ is an open-collector, active low output that signals the completion of a read, write, or interrupt acknowledge cycle. During read or interrupt acknowledge cycles, $\overline{DTACK}$ is asserted by the MC68153 after data has been provided on the data bus; during write cycles it is asserted after data has been accepted from the data bus. A pullup resistor is required to maintain $\overline{DTACK}$ high between bus cycles.

FIGURE 2 — MC68153 FUNCTIONAL BLOCK DIAGRAM

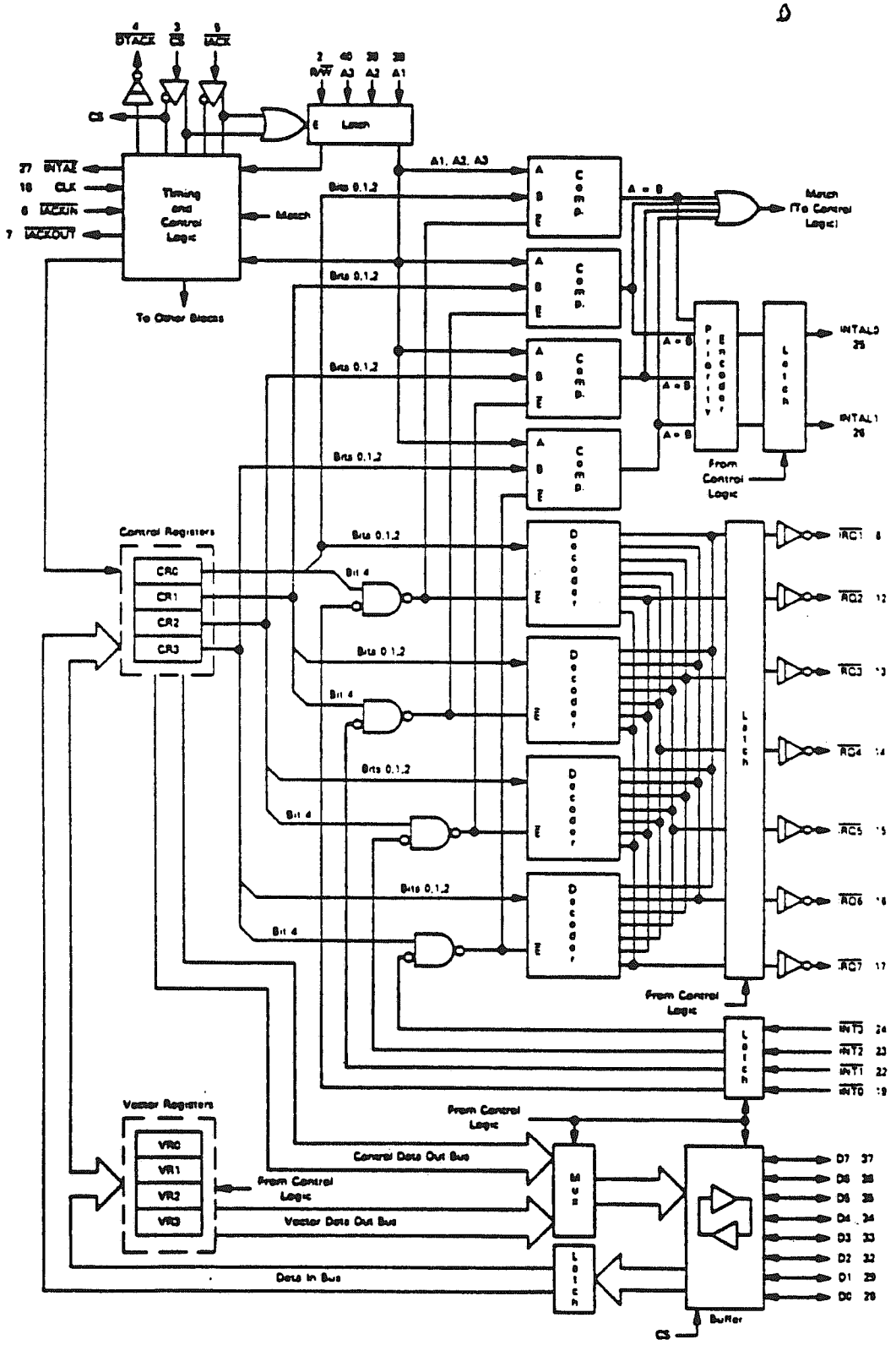
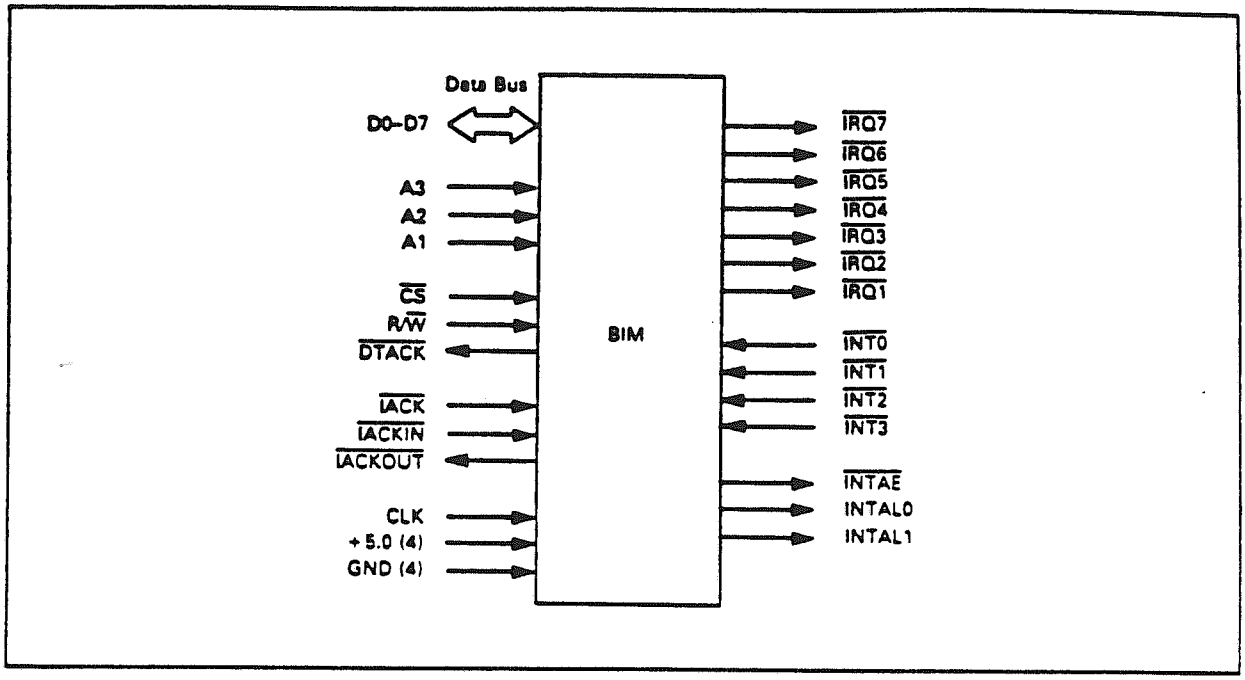


FIGURE 3 — LOGICAL PIN ASSIGNMENT



INTERRUPT ACKNOWLEDGE SIGNALS — $\overline{IACK}$, $\overline{IACKIN}$, $\overline{IACKOUT}$

These three pins support the interrupt acknowledge cycle. A low level on the $\overline{IACK}$ input indicates an interrupt acknowledge cycle has been initiated. This signal is conditioned externally with Address Strobe and the lower data strobe of an MC68000 type bus. After $\overline{IACK}$ is asserted the BIM compares the interrupt level presented on address lines A1, A2, and A3 with the current levels generated internally and determines if a match exists. Then, if input $\overline{IACKIN}$ is asserted (driven low), the BIM will either complete the interrupt acknowledge cycle if a match exists or assert output $\overline{IACKOUT}$ if *no* match exists.

$\overline{IACKIN}$ and $\overline{IACKOUT}$ form part of a prioritized interrupt acknowledge daisy chain. The daisy chain prioritizes interrupters and guarantees that two or more devices requesting an interrupt on the same level will not respond to the same cycle. The requesting device (or interrupter) must wait until $\overline{IACKIN}$ is asserted and not pass the signal on (assert $\overline{IACKOUT}$) if it is to complete the interrupt acknowledge cycle.

BUS INTERRUPT REQUEST SIGNALS — $\overline{IRQ1}$ - $\overline{IRQ7}$

These open-collector outputs are low when asserted, indicating a bus interrupt is requested at the corresponding level. An open-collector buffer is normally required for sufficient drive when interfacing to a system bus. A pullup resistor is required to maintain $\overline{IRQ1}$ - $\overline{IRQ7}$ high between interrupt requests.

DEVICE INTERRUPT REQUEST SIGNALS — $\overline{INT0}$ - $\overline{INT3}$

$\overline{INT0}$ - $\overline{INT3}$ are active low inputs used to indicate to the BIM that a device wants a bus interrupt.

INTERRUPT ACKNOWLEDGE ENABLE — $\overline{INTAE}$

During an interrupt acknowledge cycle, this output pin is asserted low to indicate that outputs INTAL0 and INTAL1 are valid. These two outputs contain an encoded number (x) corresponding to the interrupt ($\overline{INTx}$) being acknowledged. This feature can be used to signal interrupting devices, which supply their own vector, when to respond to the interrupt acknowledge cycle with the vector and a $\overline{DTACK}$ signal.

INTERRUPT ACKNOWLEDGE LEVEL — INTAL0, INTAL1

These active high outputs contain an encoded number corresponding to the interrupt level being acknowledged. They are valid only when $\overline{INTAE}$ is asserted low.

CLOCK — CLK

The CLK input is used to supply the clock for internal operations of the MC68153.

RESET — $\overline{CS}$, $\overline{IACK}$

Although a reset input is not supplied, an on-board reset is performed if $\overline{CS}$ and $\overline{IACK}$ are asserted simultaneously.

FIGURE 4 — MC68153 REGISTER MODEL

ADDRESS BIT A3	A2	A1	FLAG	FLAG AUTO-CLEAR	EXTERNAL/INTERNAL	INTERRUPT ENABLE	INTERRUPT AUTO-CLEAR	INTERRUPT LEVEL			REGISTER NAME
0	0	0	F	FAC	X/$\overline{IN}$	IRE	IRAC	L2	L1	L0	CONTROL REGISTER 0
0	0	1	F	FAC	X/$\overline{IN}$	IRE	IRAC	L2	L1	L0	CONTROL REGISTER 1
0	1	0	F	FAC	X/$\overline{IN}$	IRE	IRAC	L2	L1	L0	CONTROL REGISTER 2
1	1	1	F	FAC	X/$\overline{IN}$	IRE	IRAC	L2	L1	L0	CONTROL REGISTER 3
1	0	0	V7	V6	V5	V4	V3	V2	V1	V0	VECTOR REGISTER 0
1	0	1	V7	V6	V5	V4	V3	V2	V1	V0	VECTOR REGISTER 1
1	1	0	V7	V6	V5	V4	V3	V2	V1	V0	VECTOR REGISTER 2
1	1	1	V7	V6	V5	V4	V3	V2	V1	V0	VECTOR REGISTER 3
			7	6	5	4	3	2	1	0	REGISTER NAME
			REGISTER BIT								

REGISTER DESCRIPTION

The MC68153 contains 8 programmable read/write registers. There are four control registers (CR0 – CR3) that govern operation of the device. The other four (VR0 – VR3) are vector registers that contain the vector data used during an interrupt acknowledge cycle. Figure 4 illustrates the device register model.

CONTROL REGISTERS

There is a control register for each interrupt source, i.e., CR0 controls $\overline{INT0}$, CR1 controls $\overline{INT1}$, etc. The control registers are divided into several fields:

1. Interrupt level (L2, L1, L0) — The least significant 3-bit field of the register determines the level at which an interrupt will be generated:

L2	L1	L0	IRQ LEVEL
0	0	0	DISABLED
0	0	1	IRQ1
0	1	0	IRQ2
0	1	1	IRQ3
1	0	0	IRQ4
1	0	1	IRQ5
1	1	0	IRQ6
1	1	1	IRQ7

A value of zero in the field disables the interrupt.

2. Interrupt Enable (IRE) — This field (Bit 4) must be set (high level) to enable the bus interrupt request associated with the control register. Thus, if the $\overline{INTX}$ line is asserted and IRE is cleared, *no* interrupt request ($\overline{IRQX}$) will be asserted.
3. Interrupt Auto-Clear (IRAC) — If the IRAC is set (Bit 3), IRE (Bit 4) is cleared during an interrupt acknowledge cycle responding to this request. This action of clearing IRE disables the interrupt request. To re-enable the interrupt associated with this register, IRE must be set again by writing to the control register.
4. External/Internal (X/$\overline{IN}$) — Bit 5 of the control register determines the response of the MC68153 during an interrupt acknowledge cycle. If the X/$\overline{IN}$ bit is clear (low level) the BIM will respond with vector data and a $\overline{DTACK}$ signal, i.e., an internal response. If X/$\overline{IN}$ is set, the vector is not supplied and no $\overline{DTACK}$ is given by the BIM, i.e., an external device should respond.
5. Flag (F) — Bit 7 is a flag that can be used in conjunction with the test and set instruction of the MC68000. It can be changed without affecting chip operation. It is useful for processor-to-processor communication and resource allocation.
6. Flag Auto-Clear (FAC) — If FAC (Bit 6) is set, the Flag bit is automatically cleared during an interrupt acknowledge cycle.

VECTOR REGISTERS

Each interrupt input has its own associated vector register. Each register is 8 bits wide and supplies a data byte during its interrupt acknowledge cycle if the associated External/Internal (X/$\overline{IN}$) control register bit is clear. This data can be status, identification, or address information depending on system usage. The information is programmed by the system user.

DEVICE RESET

When the MC68153 is reset, the registers are set to a known condition. The control registers are set to all zeros (low). The vector registers are set to $0F. This value is the MC68000 vector for an uninitialized interrupt vector.

FUNCTIONAL DESCRIPTION

SYSTEM OVERVIEW

The MC68153 is compatible with many system buses, however, it is primarily intended for VMEbus, VERSAbus and MC68000 applications. Figure 5 shows a system configuration similar to VMEbus. In the figure only one system Data Transfer Bus (DTB) master is used. The Priority Interrupt structure provides a means for peripheral slave devices to ask for an interrupt of other processor (DTB master) activity and receive service from the processor. The MC68153 BIM acts as an interface device requesting and responding to interrupt acknowledge cycles for up to 4 independent slaves.

In Figure 5, functional modules are identified as Interrupters and an Interrupt Handler. An Interrupter (such as the MC68153) receives slave requests for an interrupt and handles all interface to the system bus required to ask for and respond to interrupt requests. The Interrupt Handler receives the bus interrupt requests, determines when an interrupt acknowledge will occur and at which level, and finally either performs the interrupt acknowledge (IACK) cycle or tells the DTB master to execute the IACK cycle.

The signal lines in the Priority Interrupt structure include (* — indicates active low):

1. IRQ1*–IRQ7* — seven prioritized interrupt request lines.
2. IACK* — signal line that indicates an interrupt acknowledge cycle is occurring.
3. IACKIN*/IACKOUT* — two signals that form part of a daisy chain that prioritizes interrupters.

In addition Data Transfer Bus control signals are involved in the IACK bus cycle:

1. AS* — the Address Strobe asserted low indicates a valid address is on the bus.
2. DSO* — the lower Data Strobe asserted low indicates a data transfer will occur on bus bits D00–D07.
3. WRITE* — the Read/Write is negated indicating the data is to be read from the Interrupter.
4. A01–A03 — Address lines A01–A03 contain the encoded priority level of the IACK cycle.
5. D00–D07 — Data bus lines D00–D07 are used to pass the interrupt vector from the responding Interrupter to the Interrupt Handler.
6. DTACK* — Data Transfer Acknowledge asserted low signals that the Interrupter has put the vector on the data bus.

FIGURE 5 — SIMPLE VMEbus CONFIGURATION

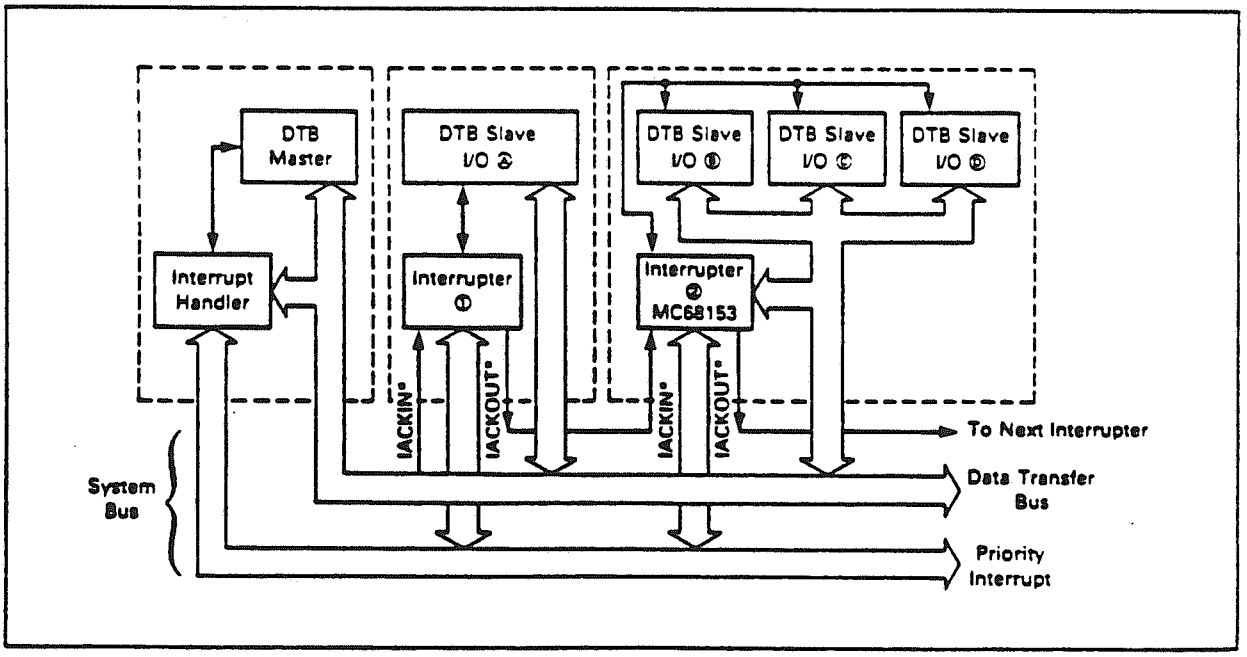
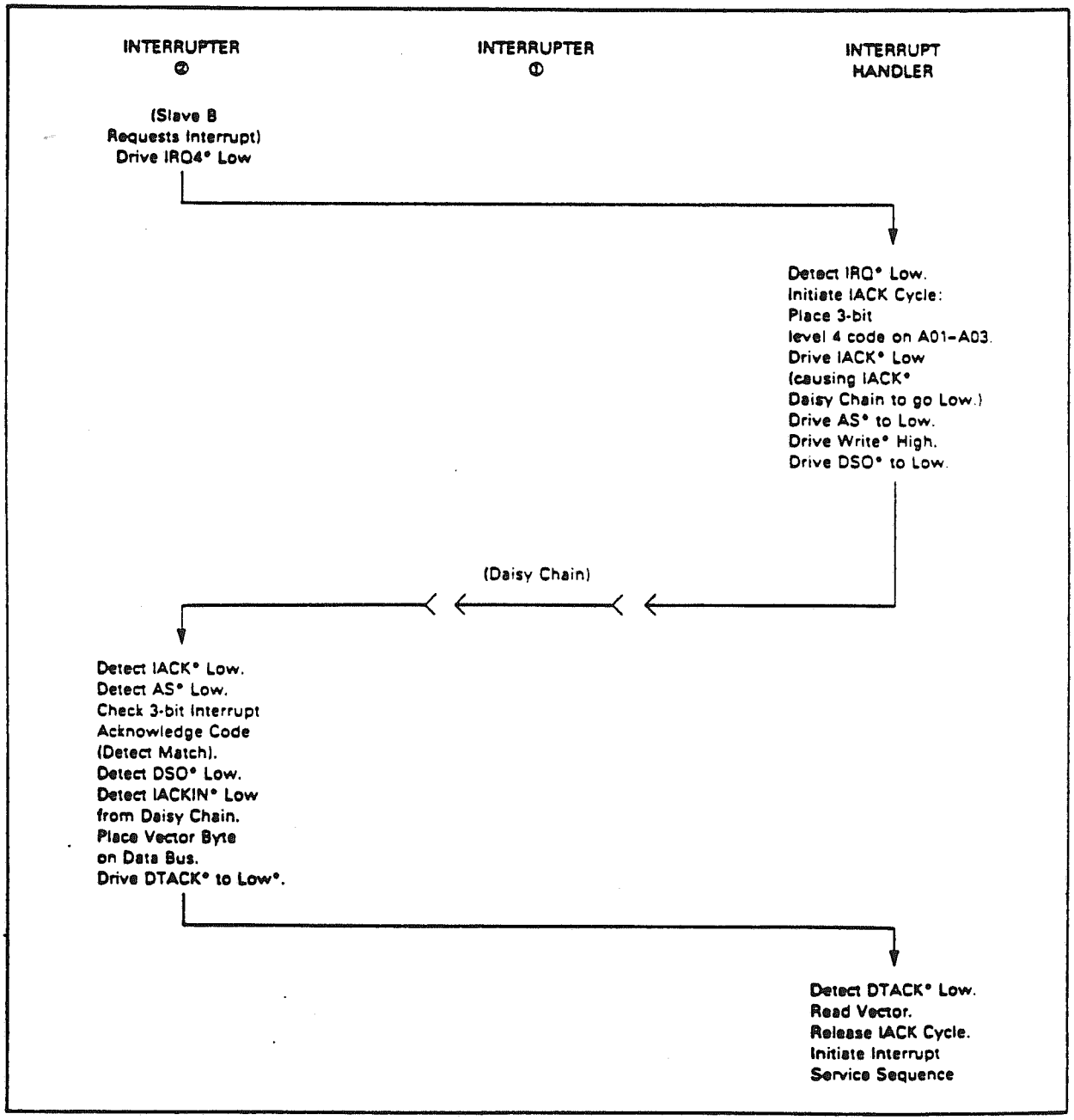


Figure 6 shows a flow diagram of a typical interrupt request and acknowledge operation. Briefly, the sequence of events is first, an Interrupter makes a request, next the Handler responds with an IACK cycle, then the Interrupter passes a vector to the Handler completing the IACK cycle, and finally the Handler uses the vector to determine additional action. Typically, an interrupt service routine is stored in software and the vector points to its starting address.

Note the daisy chain operation. If the IACK level (on A01–A03) does not match the Interrupter's request level or if no request is pending, the Interrupter passes the IACKIN* signal on and asserts IACKOUT*. This sequential action automatically prioritizes Requesters on the same level (first one in line with a request pending gets serviced) and prevents two or more Interrupters from responding simultaneously.

FIGURE 6 — INTERRUPT REQUEST AND ACKNOWLEDGE OPERATION FLOW DIAGRAM



This discussion is a very cursory look at the bus operation. For more details including situations with multiple bus masters, the user is directed to the VMEbus Specification MVMEBS or VERSAbus Specification M68KVBS. Also, the MC68153 can be used with other buses having similar interrupt structures.

BIM BUS INTERFACE

Figure 7 shows a simplified block diagram of the MC68153 interface to VERSAbus or VMEbus. Address Decode and Control Decode are dependent on the application and must be designed to guarantee BIM ac specifications. It is possible in most cases that the decode logic can be shared with the slave devices. Buffers are provided where shown to comply with bus loading and drive specifications. It is also possible that buffers can be shared with the slave bus interface.

READ/WRITE OPERATION

All eight BIM registers can be accessed from the system bus in both read and write modes. The BIM has an asynchronous bus interface, primarily designed for MC68000-like buses. The following signals generate read and write cycles: Chip Select ($\overline{CS}$), Read/Write (R/$\overline{W}$), Address Inputs (A1–A3), Data Bus (D0–D7), and Data Transfer Acknowledge ($\overline{DTACK}$). During read and write cycles the internal registers are selected by A1, A2, and A3 in compliance with the Figure 4 Truth Table.

Figure 8 shows the device timing for a read cycle. R/$\overline{W}$ and A1–A3 are latched on the falling edge of $\overline{CS}$ and must meet specified setup and hold times. Chip access time for valid data and $\overline{DTACK}$ are dependent on the clock frequency as shown in the figure.

Figure 9 shows the device timing for a write cycle. R/$\overline{W}$, A1–A3, and D0–D7 are latched on the falling edge of $\overline{CS}$ and must meet specified setup and hold times. Chip access time for $\overline{DTACK}$ is dependent on the clock frequency as shown in the figure.

FIGURE 7 — VMEbus/VERSAbus INTERFACE BLOCK DIAGRAM

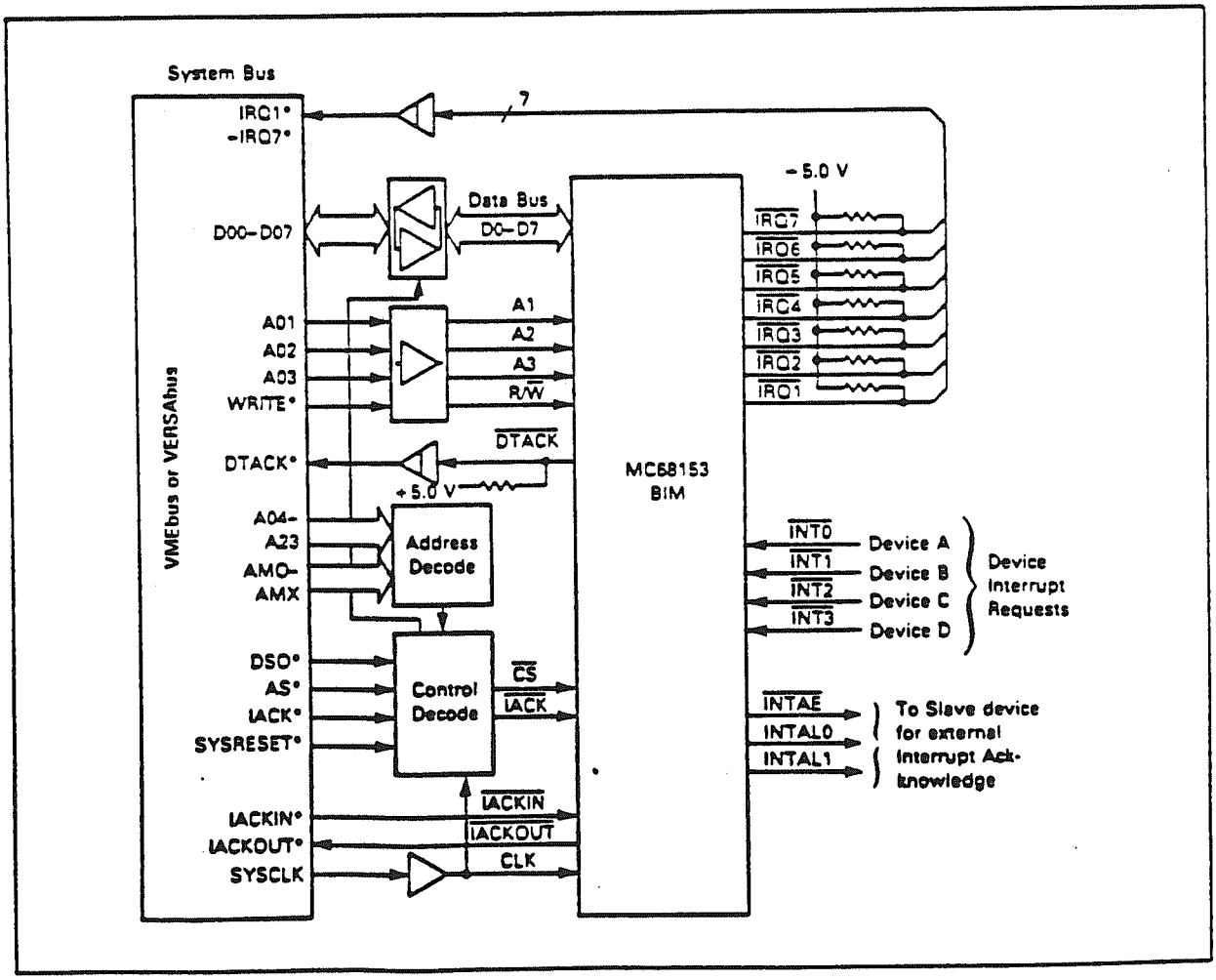


FIGURE 8 — READ CYCLE

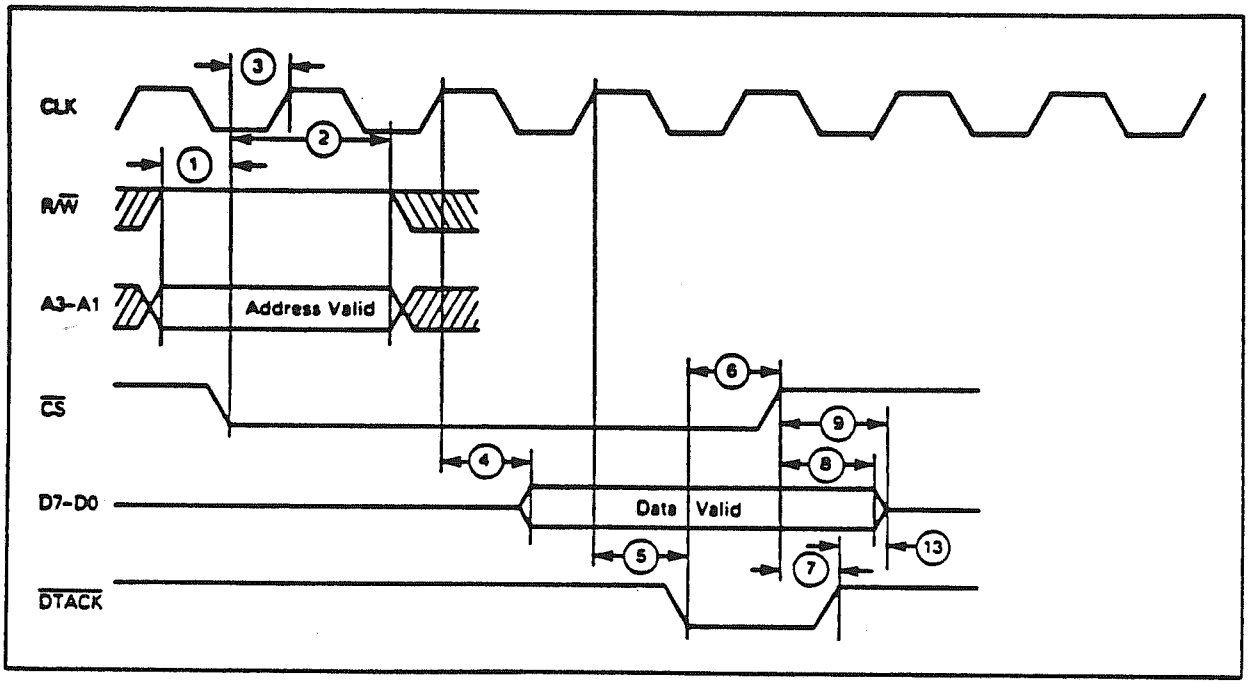
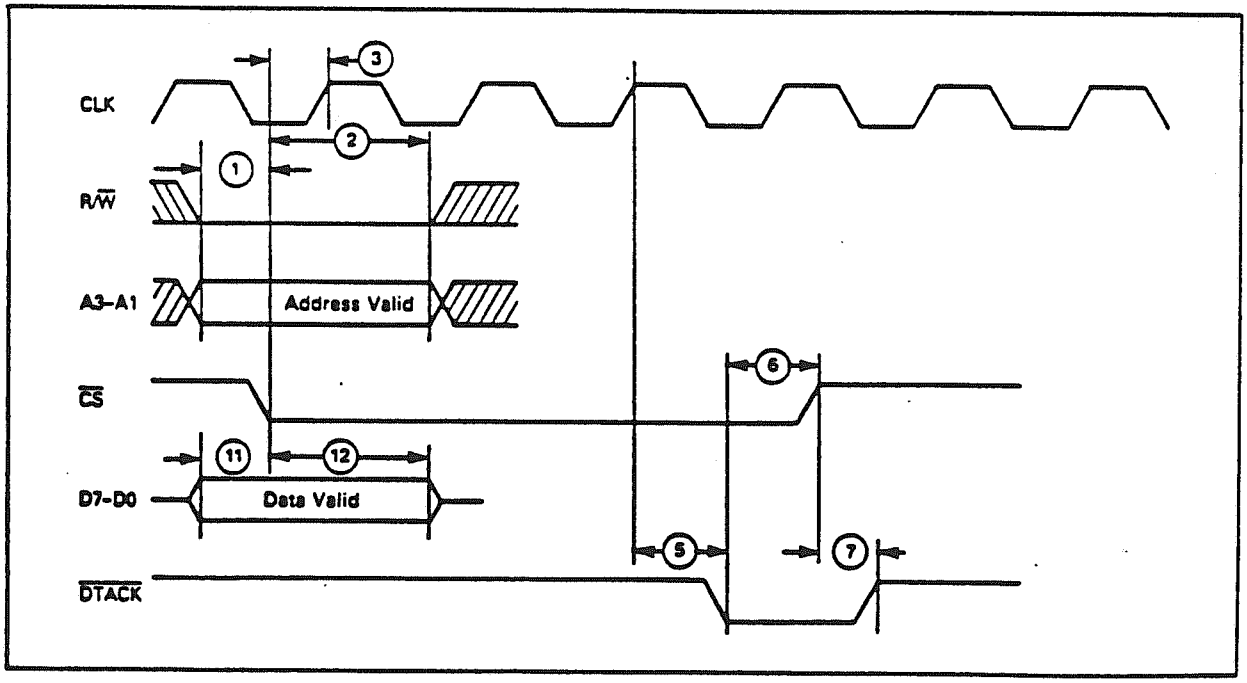


FIGURE 9 — WRITE CYCLE



INTERRUPT REQUESTS

The MC68153 accepts device interrupt requests on inputs $\overline{INT0}$, $\overline{INT1}$, $\overline{INT2}$, and $\overline{INT3}$. Each input is regulated by Bit 4 (IRE) of the associated control register (CR0 controls $\overline{INT0}$, CR1 controls $\overline{INT1}$, etc). If IRE (Interrupt Enable) is set and a device input is asserted, an Interrupt Request open-collector output ($\overline{IRQ1}$–$\overline{IRQ7}$) is asserted. The asserted $\overline{IRQX}$ output is selected by the value programmed in Bits 0, 1, and 2 of the control register (L0, L1, and L2). This 3-bit field determines the interrupt request level as set by software.

Two or more interrupt sources can be programmed to the same request level. That $\overline{IRQX}$ output will remain asserted until multiple interrupt acknowledge cycles respond to all requests.

If the interrupt request level is set to zero, the interrupt is disabled because there is no corresponding IRQ output.

INTERRUPT ACKNOWLEDGE

The response of an Interrupt Handler to a bus interrupt request is an interrupt acknowledge cycle. The IACK cycle is initiated in the MC68153 by receiving $\overline{IACK}$ low. R/$\overline{W}$, A1, A2, A3 are latched, and the interrupt level on line A1–A3 is compared with any interrupt requests pending in the chip. Further activity can be one of four cases:

1. No further action required — This occurs if $\overline{IACKIN}$ is not asserted. Asserting $\overline{IACK}$ only starts the BIM activity. If the daisy chain signal never reaches the MC68153 ($\overline{IACKIN}$ is not asserted), another Interrupter has responded to the $\overline{IACK}$ cycle. The cycle will end, the chip $\overline{IACK}$ is negated, and no additional action is required.

2. Pass on the interrupt acknowledge daisy chain — For this case, $\overline{IACKIN}$ input is asserted by the preceding daisy chain Interrupter, and $\overline{IACKOUT}$ output is in turn asserted. The daisy chain signal is passed on when no interrupts are pending on a matching level or when any possible interrupts are disabled. The Interrupt Enable (IRE) bit of a control register can disable any interrupt requests, and in turn, any possible matches.

3. Respond internally — For this case, $\overline{IACKIN}$ is asserted and a match is found. The MC68153 completes the IACK cycle by supplying an interrupt vector from the proper vector register followed by a $\overline{DTACK}$ signal asserted. $\overline{IACKOUT}$ is not asserted because the interrupt acknowledge cycle is completed by this device.

   For the MC68153 to respond in this mode of operation, the EXTERNAL/$\overline{INTERNAL}$ control register bit (X/$\overline{IN}$) must be zero. For each source of interrupt request, the associated control register determines the BIM response to an IACK cycle, and the X/$\overline{IN}$ bit sets this response either internally (X/$\overline{IN}$ = 0) or externally (X/$\overline{IN}$ = 1).

4. Respond externally — For the final case, $\overline{IACKIN}$ is also asserted, a match is found and the associated control register has X/$\overline{IN}$ bit set to one. The MC68153 does not assert $\overline{IACKOUT}$ and does assert $\overline{INTAE}$ low. $\overline{INTAE}$ signals that the requesting device must complete the IACK cycle (supplying a vector and $\overline{DTACK}$) and that the 2-bit code contained on outputs INTAL0 and INTAL1 shows which interrupt source is being acknowledged.

These cases are discussed in more detail in the following paragraphs.

Internal Interrupt Acknowledge

For an internal interrupt acknowledge to occur, the following conditions must be met:

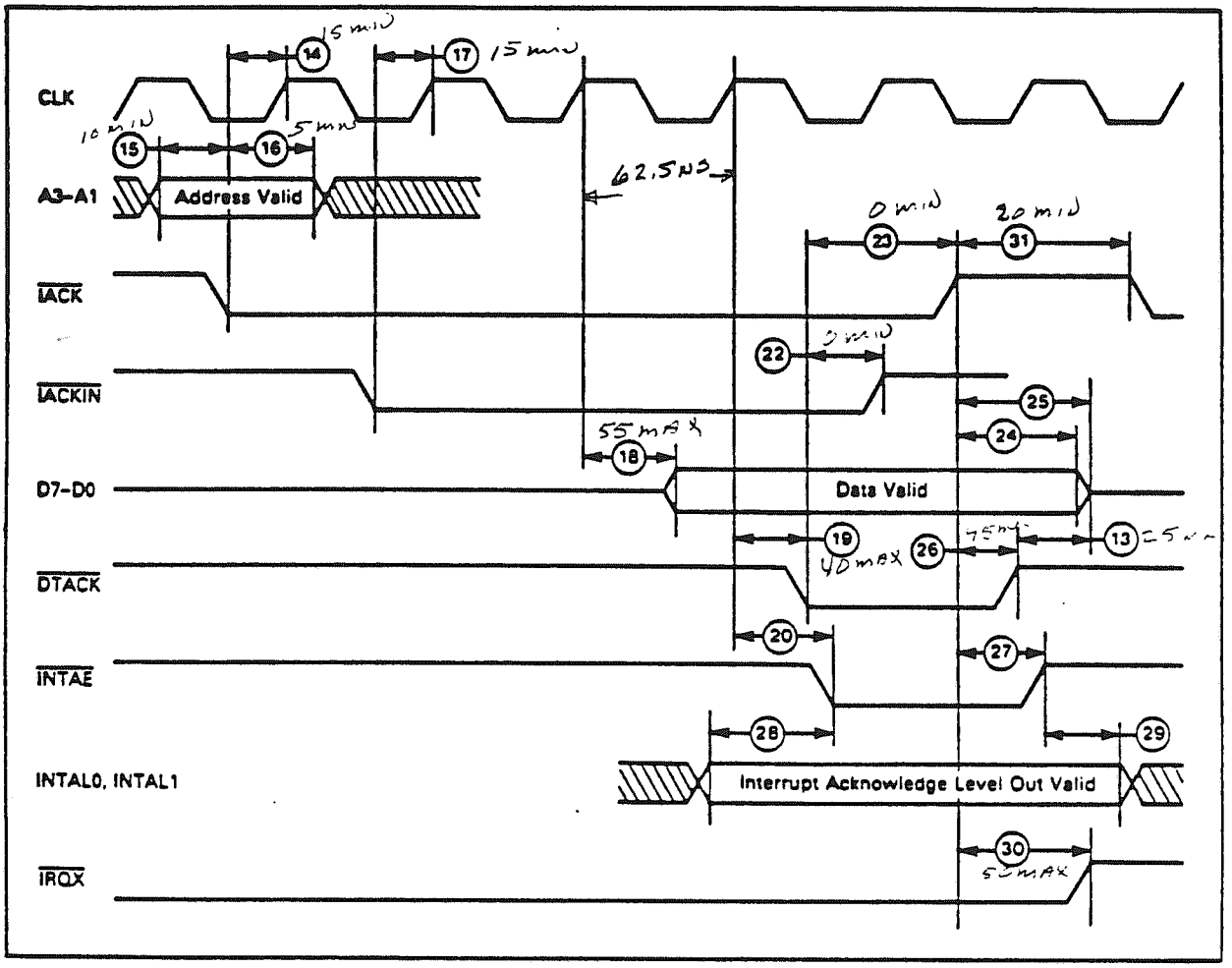
1. One or more device interrupt inputs ($\overline{INT0}$–$\overline{INT3}$) has been asserted and corresponding control bit IRE value is one.
2. $\overline{IACK}$ asserted.
3. A match exists between (A3, A2, A1) and the (L2, L1, L0) field of an enabled, requesting control register. If two or more devices are requesting at the same interrupt level, preference is given to the highest number requester, that is, $\overline{INT3}$ has highest priority and $\overline{INT0}$ has lowest.
4. Control register bit X/$\overline{IN}$ of matching interrupt source must be zero.
5. $\overline{IACKIN}$ asserted.

The internal interrupt acknowledge cycle timing is shown in Figure 10. The 8-bit interrupt acknowledge vector is presented to the data bus and $\overline{DTACK}$ is asserted. Note also that INTAL0 and INTAL1 are valid and $\overline{INTAE}$ is asserted during this cycle although they would normally not be used. The cycle is terminated (data and $\overline{DTACK}$ released) after $\overline{IACK}$ is negated.

During the IACK cycle, the INTERRUPT AUTO-CLEAR control bit (IRAC) comes into play. If the IRAC = one for the responding interrupt source, the INTERRUPT ENABLE (IRE) bit is automatically cleared during the IACK cycle, thus disabling the associated interrupt input and any $\overline{IRQX}$ output asserted due to this interrupt input. Before another interrupt can be requested from this source, IRE must be set to one by writing to the control register.

Note that $\overline{IACKOUT}$ is not asserted because this device is responding to the IACK and does not pass the daisy chain signal on. Also, new device interrupt requests occurring on $\overline{INT0}$–$\overline{INT3}$ after $\overline{IACK}$ is asserted are locked out to prevent any race conditions on the daisy chain.

FIGURE 10 — INTERRUPT ACKNOWLEDGE CYCLE — INTERNAL VECTOR



External Interrupt Acknowledge

For an external interrupt acknowledge, the same conditions as listed above are met with one exception. Control register bit X/$\overline{IN}$ of matching interrupt source must be set to one. The timing is shown in Figure 11. For this cycle, the interrupt vector and $\overline{DTACK}$ must be supplied by an external device. $\overline{INTAE}$ is asserted indicating that INTAL0 and INTAL1 are valid. The external device can use these signals to enable the vector and $\overline{DTACK}$. The cycle is terminated after $\overline{IACK}$ is negated.

The IRAC control bit acts in the external interrupt acknowledge the same as described for the internal response (see above). Also, $\overline{IACKOUT}$ is not asserted and new device interrupts are disabled for reasons discussed above.

Pass On IACK Daisy Chain

If the MC68153 has no interrupt request pending at the same level as the interrupt acknowledge, the IACK daisy chain signal is passed on to the next device if $\overline{IACKIN}$ is asserted. The following conditions are thus met:

1. $\overline{IACK}$ asserted.
2. No match exists between [A3, A2, A1] and the [L2, L1, L0] field of an enabled, requesting control register.
3. $\overline{IACKIN}$ is asserted.

$\overline{IACKOUT}$ is asserted if these conditions are valid. This output drives $\overline{IACKIN}$ of the next Interrupter on the daisy chain, passing the signal along. Figure 12 shows the timing for this case. $\overline{IACKOUT}$ is negated after $\overline{IACK}$ is negated.

FIGURE 11 — INTERRUPT ACKNOWLEDGE CYCLE — EXTERNAL VECTOR

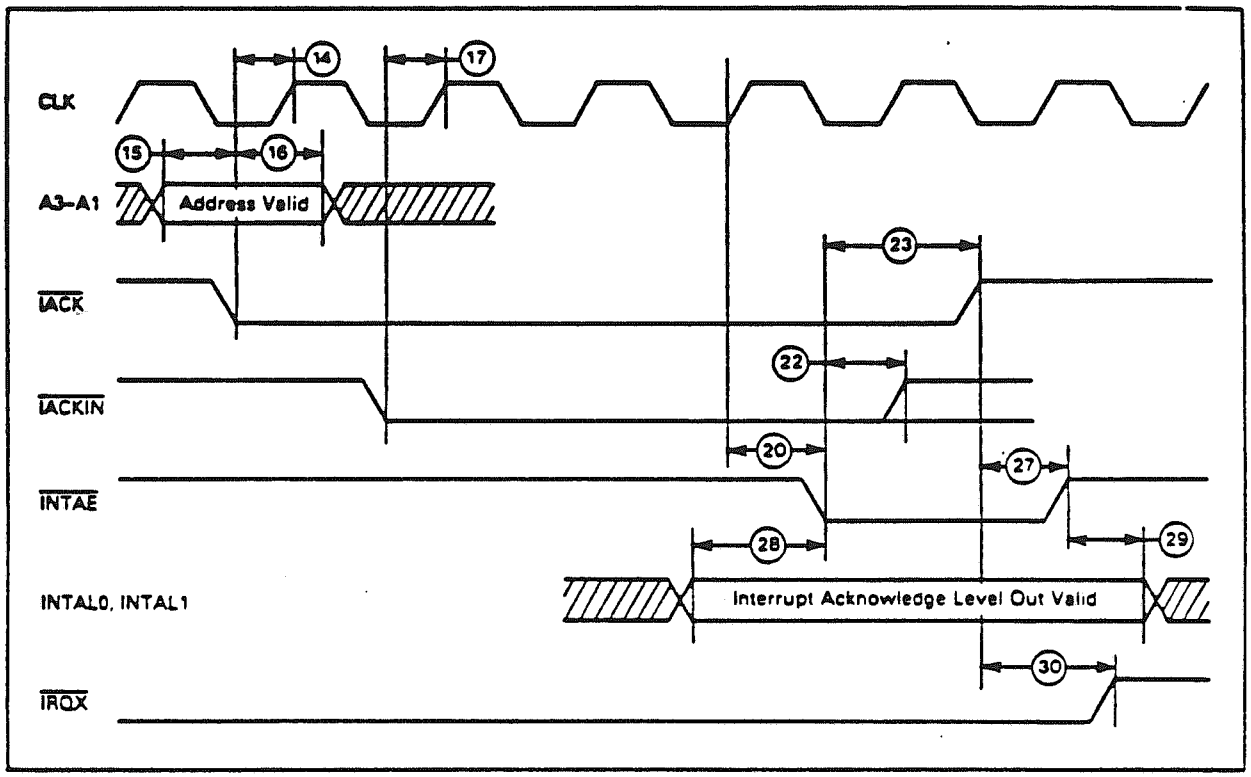
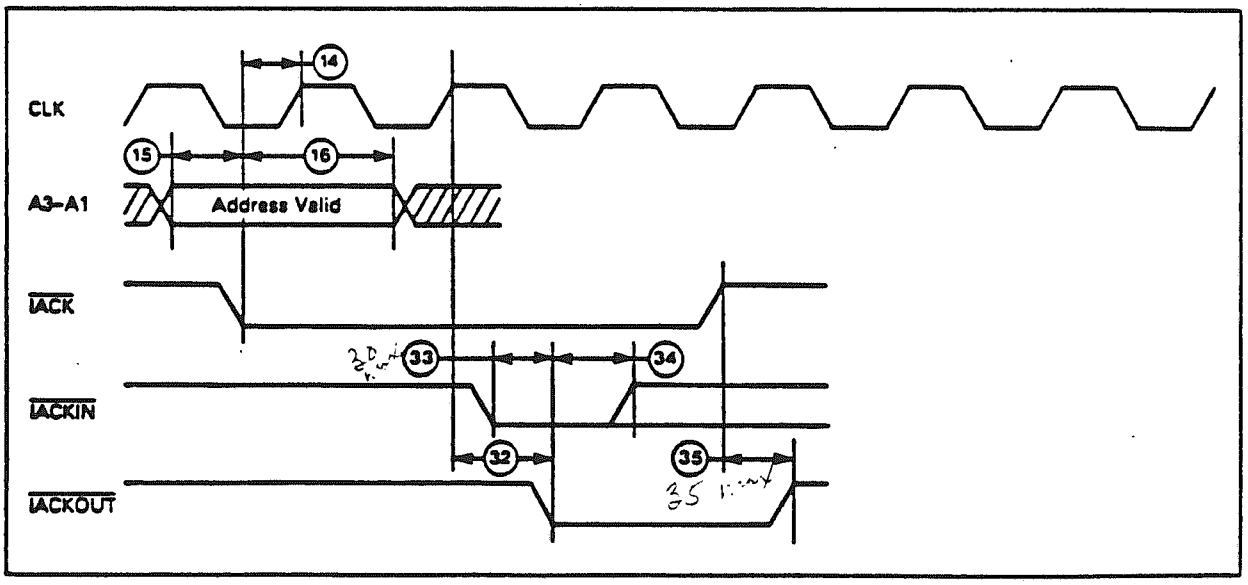


FIGURE 12 — INTERRUPT ACKNOWLEDGE CYCLE — $\overline{\text{IACKOUT}}$



CONTROL REGISTER FLAGS

Each control register contains a Flag bit (F) and a Flag Auto-Clear bit (FAC). Both bits can be read or altered via a register write without affecting the interrupt operation of the device. The Flag is useful as a status indicator for resource management and as a semaphor in multitasking or multiprocessor systems. Flag (F) is located in bit position 7 and can be used with the MC68000 Test and Set (TAS) instruction.

The Flag Auto-Clear (FAC) is used to manipulate the Flag bit. If the Flag is set to one and the FAC is also one, an interrupt acknowledge cycle to the associated interrupt source clears the Flag bit. This feature is useful in determining the interrupt status and passing messages.

RESET

There is no reset input, however, a chip reset is activated by asserting both $\overline{CS}$ and $\overline{IACK}$ simultaneously (Figure 13). These inputs should be held low for a minimum of two clock cycles for a full reset function. The control registers are reset to all zeroes and the Vector Registers are set to a value of $0F. This vector value is the uninitialized vector for the MC68000. See the MC68000 Users Manual for more details on this vector.

CLOCK

The chip clock is required for internal operation to occur. Typical frequency is 16 MHz in VMEbus and VERSAbus applications derived from the system clock. Any frequency can be used, however, up to 25 MHz (Figure 14).

FIGURE 13 — RESET

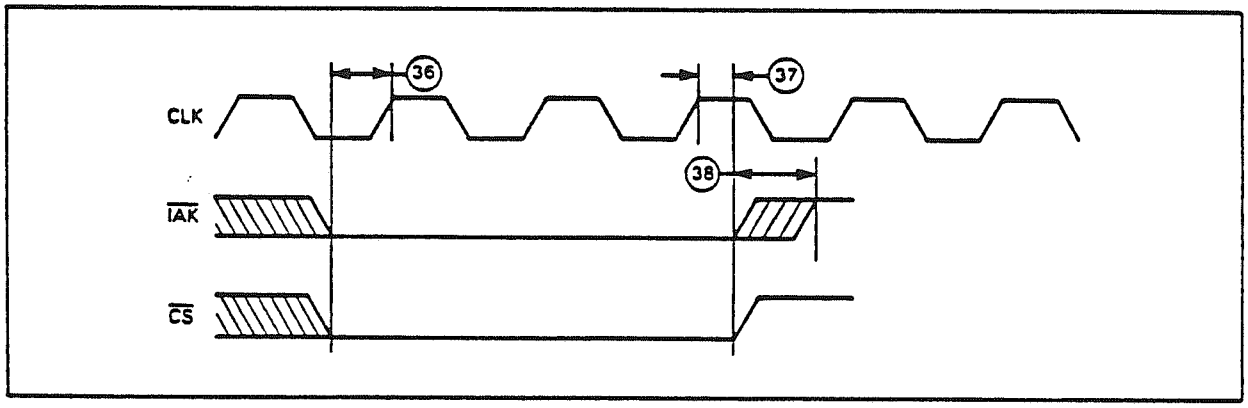


FIGURE 14 — CLOCK WAVEFORM

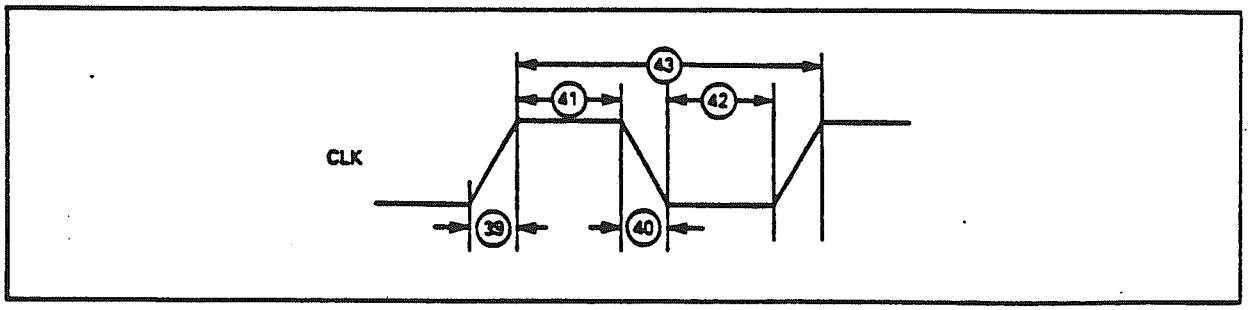


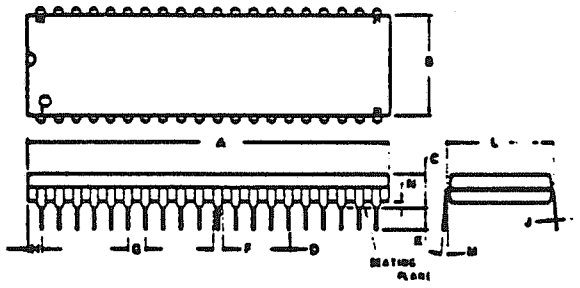
TABLE 1
AC PERFORMANCE SPECIFICATIONS
(V_{CC} = 5.0 V ±5%, T_A = 0°C to 70°C)

Number	Characteristic	Min	Max	Units	Notes
1	R/$\overline{W}$, A1–A3 Valid to $\overline{CS}$ Low (Setup Time)	10	—	ns	1
2	$\overline{CS}$ Low to R/$\overline{W}$, A1–A3 Invalid (Hold Time)	5.0	—	ns	
3	$\overline{CS}$ Low to CLK High (Setup Time)	15	—	ns	1
4	CLK High to Data Out Valid (Delay)	—	55	ns	2
5	CLK High to $\overline{DTACK}$ Low (Delay)	—	40	ns	2
6	$\overline{DTACK}$ Low to $\overline{CS}$ High	0		ns	
7	$\overline{CS}$ High to $\overline{DTACK}$ High (Delay)	—	35	ns	10
8	$\overline{CS}$ High to Data Out Invalid (Hold Time)	0	—	ns	
9	$\overline{CS}$ High to Data Out High-Impedance (Hold Time)	—	50	ns	
10	$\overline{CS}$ High to $\overline{CS}$ or $\overline{IACK}$ Low	20	—	ns	
11	Data In Valid to $\overline{CS}$ Low (Setup Time)	10	—	ns	
12	$\overline{CS}$ Low to Data In Invalid (Hold Time)	5.0	—	ns	
13	$\overline{DTACK}$ High to Data Out High-Impedance	—	25	ns	10
14	$\overline{IACK}$ Low to CLK High (Setup Time)	15	—	ns	1
15	A1–A3 Valid to $\overline{IACK}$ Low (Setup Time)	10	—	ns	
16	$\overline{IACK}$ Low to A1–A3 Invalid (Hold Time)	5.0	—	ns	
17	$\overline{IACKIN}$ Low to CLK High (Setup Time)	15	—	ns	1, 8
18	CLK High to Data Out Valid (Delay)	—	55	ns	3
19	CLK High to $\overline{DTACK}$ Low (Delay)	—	40	ns	3
20	CLK High to $\overline{INTAE}$ Low (Delay)	—	40	ns	3
22	$\overline{DTACK}$ Low to $\overline{IACKIN}$ High	0	—	ns	8
23	$\overline{DTACK}$ Low to $\overline{IACK}$ High	0	—	ns	
24	$\overline{IACK}$ High to Data Out Invalid (Hold Time)	0	—	ns	
25	$\overline{IACK}$ High to Data Out High Impedance (Delay)	—	60	ns	
26	$\overline{IACK}$ High to $\overline{DTACK}$ High (Delay)	—	45	ns	10
27	$\overline{IACK}$ High to $\overline{INTAE}$ High (Delay)	—	35	ns	
28	INTAL0, INTAL1 Valid to $\overline{INTAE}$ Low (Setup Time)	1.0	2.0	CLK Per	
29	$\overline{INTAE}$ High to INTAL0, INTAL1 Invalid (Hold Time)	1.0	2.0	CLK Per	
30	$\overline{IACK}$ High to $\overline{IRQx}$ High (Delay)	—	50	ns	7, 10
31	$\overline{IACK}$ High to $\overline{IACK}$ or $\overline{CS}$ Low	20	—	ns	
32	CLK High to $\overline{IACKOUT}$ Low (Delay)	—	40	ns	5
33	$\overline{IACKIN}$ Low to $\overline{IACKOUT}$ Low (Delay)	—	30	ns	4, 8
34	$\overline{IACKOUT}$ Low to $\overline{IACKIN}$, $\overline{IACK}$ High	0	—	ns	8
35	$\overline{IACK}$ High to $\overline{IACKOUT}$ High (Delay)	—	35	ns	
36	$\overline{IACK}$ and $\overline{CS}$ both Low to CLK High (Setup Time)	15	—	ns	9
37	CLK High to $\overline{IACK}$ or $\overline{CS}$ High (Hold Time)	0	—	ns	
38	$\overline{IACK}$ or $\overline{CS}$ High to $\overline{IACK}$ and $\overline{CS}$ High (Skew)	—	1.0	CLK Per	6
39	Clock Rise Time	—	10	ns	
40	Clock Fall Time	—	10	ns	
41	Clock High Time	20	—	ns	
42	Clock Low Time	20	—	ns	
43	Clock Period	40	—	ns	

NOTES:

1. This specification only applies if the VBIM had completed all operations initiated by the previous bus cycle when $\overline{CS}$ or $\overline{IACK}$ was asserted. Following a normal bus cycle, all operations are completed within 2 clock cycles after $\overline{CS}$ or $\overline{IACK}$ have been negated. If $\overline{IACK}$ or $\overline{CS}$ is asserted prior to completion of these operations, the new cycle, and hence, $\overline{DTACK}$ is postponed.
   If the $\overline{IACK}$, $\overline{IACKIN}$ or $\overline{CS}$ setup time is violated, $\overline{DTACK}$ may be asserted as shown, or may be asserted one clock cycle later (i.e. $\overline{IACK}$ will not be recognized until the next rising edge of the clock).
2. Assumes that 3 has been met.
3. Assumes that 14 and 17 have both been met.
4. Assumes that 14 has been met. ($\overline{IACKOUT}$ cannot go low prior to $\overline{IACKIN}$ going low).
5. Assumes that 14 has been met and $\overline{IACKIN}$ has been low for at least the amount of time specified by 33.
6. 38 is the minimum skew between the last moment when both $\overline{IACK}$ and $\overline{CS}$ are asserted to when both are negated, to insure that an access cycle is not unintentionally started.
7. Assumes no other $\overline{INTx}$ input is causing $\overline{IRQx}$ to be driven low.
8. In non-daisy chain systems, $\overline{IACKIN}$ may be tied low.
9. Failure to meet this spec. causes RESET to be ignored for 1 clock period. It is then necessary to keep these signals low for 3 clock periods instead of 2.
10. Delay time is specified from input signal to Open-Collector Output pulled High thru 1.0 kΩ resistor to +5.5 V.

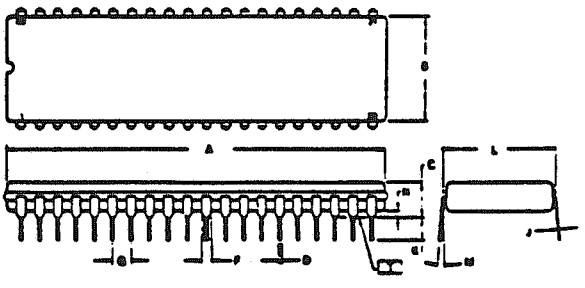
OUTLINE DIMENSIONS



DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

NOTES
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

CASE 711-03
PLASTIC PACKAGE



DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	51.31	53.34	2.020	2.100
B	12.70	15.49	0.500	0.610
C	4.06	5.84	0.160	0.230
D	0.38	0.56	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	5°	15°	5°	15°
N	0.51	1.27	0.020	0.050

NOTES
1. DIM A IS DATUM.
2. POSITIONAL TOLERANCE FOR LEADS: ⌀ 0.25 (0.010) M T A M B M
3. T IS SEATING PLANE.
4. DIM L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONS A AND B INCLUDE MENISCUS.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

CASE 734-04
CERAMIC PACKAGE

TYPICAL THERMAL CHARACTERISTICS

Package	θJA (Junction to Ambient) Still Air	Junction Temperature Still Air @ 70°C Ambient
L Suffix	40°C/W	147°C
P Suffix	35°C/W	137°C

MOTOROLA Semiconductor Products Inc.
BOX 20912 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.

APPENDIX B

SIGNETICS SCB68430 DIRECT MEMEORY ACCESS INTERFACE CHIP

Signetics

SCB68430
Direct Memory Access Interface (DMAI)

Preliminary Specification

Microprocessor Products

DESCRIPTION

The SCB68430 Direct Memory Access Interface (DMAI) is a single channel interface circuit which is intended to complement the performance and architectural capabilities of the SCN68000 microprocessor. The DMAI functions by transferring a series of operands (data) between memory and a device: operand sizes may be byte, word, or long word. A block is a sequence of operands: the number of operands in the block is determined by a transfer count stored within the DMAI. The SCB68430 can be programmed to utilize single cycle (cycle stealing) or burst data transfers.

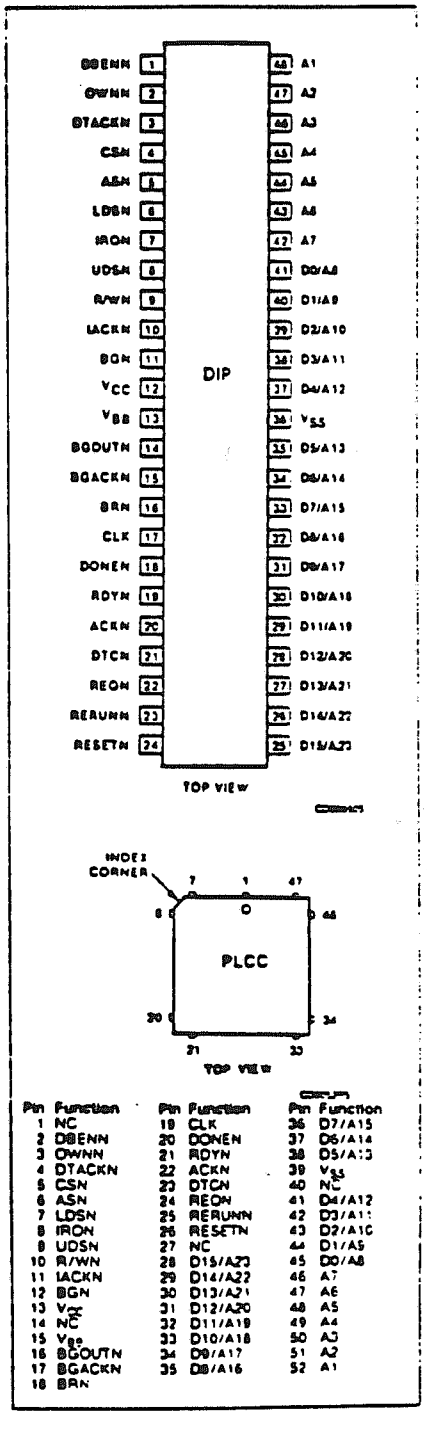
The DMAI provides two interfaces. The microprocessor interface is fully compatible with the SCN68000 microprocessor. The device interface includes lines for requesting, acknowledging, controlling, and timing the data transfers. The DMAI is a single-channel subset of the other 68000 family DMA controllers (68440 and 68450). It is software compatible with these devices and provides similar interfacing signals to both the system bus and the device.

The SCB68430 is constructed using Signetics ISL bipolar technology.

FEATURES

- Bus compatible with SCN68000 microprocessor
- Software compatible with other 68K family DMA controllers
- Single address transfers
- Cycle steal and burst mode operation
- Bus arbitration daisy chain
- Automatic rerun on bus error
- Supports 32-bit transfers for VME bus
- Supports SCN68000 vectored interrupts
- 24-bit address counter
- 16-bit transfer counter
- Maximum transfer rate of 5 Mbytes per second
- Signetics ISL bipolar technology

PIN CONFIGURATION



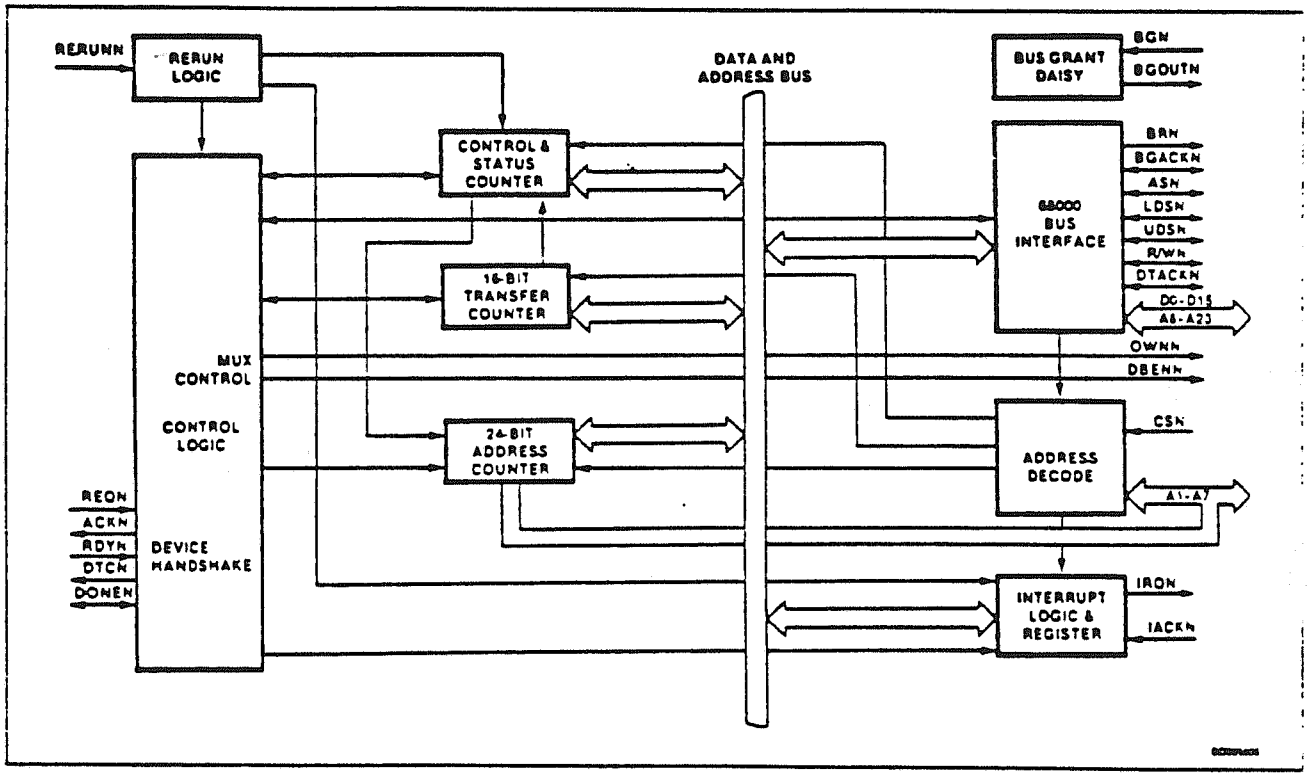
Pin	Function	Pin	Function	Pin	Function
1	NC	18	CLK	36	D7/A15
2	DBENN	20	DONEN	37	D6/A14
3	OWNN	21	RDYN	38	D5/A13
4	DTACKN	22	ACKN	39	V_{SS}
5	CSN	23	DTCN	40	NC
6	ASN	24	REQN	41	D4/A12
7	LDSN	25	RERUNN	42	D3/A11
8	IRQN	26	RESETN	43	D2/A10
9	UDSN	27	NC	44	D1/A9
10	R/WN	28	D15/A23	45	D0/A8
11	IACKN	29	D14/A22	46	A7
12	BGN	30	D13/A21	47	A6
13	V_{CC}	31	D12/A20	48	A5
14	NC	32	D11/A19	49	A4
15	V_{SS}	33	D10/A18	50	A3
16	BGOUTN	34	D9/A17	51	A2
17	BGACKN	35	D8/A16	52	A1
18	BRN				

Direct Memory Access Interface (DMAI) SCB68430

ORDERING CODE

PACKAGES	V_{CC} = 5V ± 5%, T_A = 0°C to 70°C	
	10MHz	12.5MHz
Ceramic DIP	SCB68430CAI48	SCB68430CCI48
Plastic DIP	SCB68430CAN48	SCB68430CCN48
Plastic LCC	SCB68430CAA52	SCB68430CCA52

BLOCK DIAGRAM



PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	DESCRIPTION
A1 – A7	48 – 42	I/O	Address Lines: Active high, three-state. In the MPU mode, these low order address lines specify which internal register of the DMAI is being accessed. In DMA mode, A1 – A7 are ouputs which provide the low order address bits of the location being accessed. Three-stated in IDLE Mode.
A8 – A23/ D0 – D15	41 – 37 35 – 25	I/O	Address/Data Lines: Active high, three-state. These lines are time multiplexed for data and address leads The lines OWNN, RWN, CSN, and DBENN are used to control the demultiplexing of the address and data using external circuitry. In MPU mode, the bidirectional data lines (D0 – D15) are used to transfer data between the MPU and the DMAI. In the DMA mode, A8 – A23 provide the high order address bits of the location being accessed. Three-stated in IDLE mode.
ASN	5	I/O	Address Strobe: Active low, three-state. In MPU and IDLE modes, ASN is an input which indicates that the current bus master has placed a valid address on the bus. It is monitored by the DMAI during bus arbitration to ascertain that the previous bus master has completed the current bus cycle. In DMA mode, it is an output indicating that the DMAI has placed a valid address on the bus.
UDSN	8	I/O	Upper Data Strobe: Active low, three-state. In MPU and IDLE modes, UDSN is an input which indicates that the upper data byte of the addressed word is being addressed. In DMA mode, it is an output with the same meaning.
LDSN	6	I/O	Lower Data Strobe: Active low, three-state. In MPU and IDLE modes, LDSN is an input which indicates that the lower data byte of the addressed word is being addressed. In DMA mode, it is an output with the same meaning.
R/WN	9	I/O	Read/Write: Active high for read, low for write, three-state. In MPU mode, R/WN is an input which controls the direction of data flow through the DMAI's input/output data bus interface and, if required, through an external data bus buffer. R/WN high causes the DMAI to place the data from the addressed register on the data bus, while R/WN low causes the DMAI to accept data from the data bus. In DMA mode, R/WN is an output to memory and I/O controllers indicating the type of bus cycle. It is held three-stated during IDLE mode.
CSN	4	I	Chip Select: Active low. When low, places the DMAI into the MPU mode. This input signal is used to select the DMAI for programmed data transfers. These transfers take place over D0 – D15 as controlled by the R/WN and A1 – A7 inputs. The DMAI is deselected when CSN is high. CSN is ignored during DMA mode.
DTACKN	3	I/O	Data Transfer Acknowledge: Active low, three-state. In MPU mode, DTACKN is asserted on a write cycle to indicate that the data on the bus has been latched, and on a read cycle or interrupt acknowledge cycle to indicate that valid data is present on the bus. The signal is negated (driven high) when completion of the cycle is indicated by negation of the CSN or IACKN input, and returns to the inactive third state a short time after it is negated. In DMA Mode, DTACKN is an input monitored by the DMAI to determine when the addressed device (memory) has latched the data (write cycle) or put valid data on the bus (read cycle).
RESETN	24	I	Master Reset: Active low. Assertion of this pin clears internal control registers (See table 1), initializes the interrupt vector register to H'0F', and sets the status register to the default value B'0000 000X', where X is the state of RDYN. All bidirectional I/O lines are three-stated and the DMAI is placed in the IDLE mode.
CLK	17	I	Clock: Active high. Usually the system clock, but may be any clock meeting the electrical specifications. Used by the DMAI to synchronize device functions and external control lines, and may not be gated off at any time.
IRQN	7	O	Interrupt Request: Active low, open collector. This output is asserted, if interrupts are enabled, upon end of transfer, on occurrence of a bus error, and on receipt of an abort from the MPU. The CPU can read the status register to determine the interrupting condition(s), or can respond with an interrupt acknowledge cycle to cause the DMAI to output an interrupt vector on the data bus.
IACKN	10	I	Interrupt Acknowledge: Active low. When asserted, indicates that the current cycle is an interrupt acknowledge cycle. The DMAI normally responds by placing the contents of the interrupt vector register on the data bus and asserting DTACKN. IACKN is not serviced if the DMAI has not generated an interrupt request.
BRN	16	O	Bus Request: Active low, open collector. BRN is asserted by the DMAI to request ownership of the bus after a DMA request is sensed on the REQN input from the I/O device. It is negated when the bus has been granted (BGN low) and BGACKN has been asserted, or, in burst DMA request mode, if the I/O device negates its request at least one clock cycle before BGACKN is asserted.
BGN	11	I	Bus Grant: Active low. BGN indicates to the DMAI that it is to be the next bus master. This signal is originated by the MPU and propagated via a daisy chain or other prioritization mechanism. After BGN is asserted, the DMAI waits until DTACKN, ASN, and BGACKN have become inactive before assuming ownership of the bus by asserting BGACKN.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.	TYPE	DESCRIPTION
BGOUTN	14	O	**Bus Grant Output:** Active low. Daisy chain output which is asserted by the DMAI when BGN is asserted and the DMAI does not have a bus request pending.
BGACKN	15	I/O	**Bus Grant Acknowledge:** Active low, three-state. As an input, BGACKN is monitored by the DMAI during the bus arbitration cycle to determine when it can assume ownership of the bus (BGACKN negated). In DMA mode, it is asserted by the DMAI to indicate that it is the bus master. Three-stated in MPU and IDLE modes.
RERUNN	23	I	**Rerun:** Active low. This input is asserted by external error detect logic to indicate a bus error. In DMA mode, the DMAI stops operation and three-states the data, address, and control lines, except BGACKN. It remains halted until RERUNN becomes inactive, and then re-tries the last bus cycle. If RERUNN is asserted again, the DMAI sets the ERR bit in the status register, stops DMA operation, releases the bus, and interrupts the CPU, if interrupts are enabled, responding with a special interrupt vector when IACKN is asserted. Not monitored in MPU and IDLE modes.
REQN	22	I	**DMA Request:** Active low. This input from the I/O device requests service from the DMAI and causes the DMAI to request control of the bus. In burst mode, the input is level sensitive, and the DMAI releases the bus after REQN is negated and the current DMA cycle is completed. In cycle steal mode, the REQN input is negative edge triggered. A negative going edge must occur at least one clock cycle before DTCN is asserted to accomplish continuous transfer cycles but not earlier than beginning of master cycle.
ACKN	20	O	**DMA Request Acknowledge:** Active low. ACKN is asserted by the DMAI to indicate that it has gained the bus and the requested bus cycle is now beginning. It is asserted at the beginning of every bus cycle after ASN has been asserted, and is negated at the end of every bus cycle.
RDYN	19	I	**Device Ready:** Active low. RDYN is asserted by the requesting device to indicate to the DMAI that valid data has either been stored or put on the bus. If negated, it indicates that the data has not been stored or presented, causing the DMAI to enter wait states. RDYN can be held low continuously if the device is fast enough so that wait states are not required.
DTCN	21	O	**Device Transfer Complete:** Active low. In DMA mode, DTCN is asserted by the DMAI to indicate to the device that the requested data transfer is complete. On a write to memory operation, it indicates that the data provided by the device has been successfully stored. On a read from memory operation, it indicates to the device that the data from memory is present on the data bus and should be latched.
DONEN	18	I/O	**Done:** Active low, open collector. As an output, DONEN is asserted by the DMAI concurrent with the ACKN output to indicate to the device that the transfer count is exhausted and that the DMAI's operation is completed as a result of that transfer. As an input, if asserted by the device before the transfer count became zero, it causes the DMAI to abort service and generate an interrupt request, if interrupts are enabled.
OWNN	2	O	**Own:** Active low, open collector. This output is asserted by DMAI during the DMA mode to indicate bus mastership. It can be used to enable external address/data and control buffers. Inactive in MPU and IDLE modes.
DBENN	1	O	**Data Bus Enable:** Active low, open collector. Asserted by the DMAI when CSN is asserted or when IACKN is asserted and the DMAI has an interrupt request pending. Can be used to enable bidirectional data buffers for D0 - D15. Inactive in IDLE and DMA mode.
V_{CC}	12	I	**Power Supply:** +5 volt power input.
V_{BB}	13	I	**Power Supply:** +1.5 volt power input.
V_{SS}	36	I	**Ground:** Signal and power ground input.

PIN DESCRIPTION

The Pin Description table describes the function of each of the pins of the DMAI. Signal names ending in 'N' are active low. All other signals are active high. In the descriptions, 'MPU mode' refers to the state when the DMAI is chip selected. The term 'DMA mode' refers to the state when the DMAI assumes ownership of the bus. The DMAI is in the 'IDLE mode' at all other times.

In this data sheet signals are discussed using the terms 'active' and 'inactive' or 'asserted' and 'negated' independent of whether the signal is active in the high (logic one) state or the low (logic zero) state. Refer to the individual pin descriptions for the definition of the active level of each signal.

REGISTERS AND COUNTERS

Register Map

The internal accessible register organization of the DMAI is shown in table 1. The following rules apply to all registers:

1. A read from a reserved location in the map results in a read from the 'null register'. The null register returns all ones for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle but no write occurs.
2. Unused bits of a defined register are read as indicated in the register descriptions.
3. All registers are addressable as 8-bit quantities. To facilitate operation with the 68K MOVEP instruction, addresses are ordered such that certain sets of registers may also be accessed as words or long words.

The operation of the DMAI is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The contents of certain control and status registers are initialized on RESET.

To provide compatibility with the other 68K family DMA controllers, control and status bits are mapped in bit positions equivalent to where they are located in the register map of the other devices. Bits which are used in the other devices but not in the DMAI are assigned default values. If upward compatibility to the other controllers is required, the programmer should use these default values when writing the control words to the registers, although they have no effect in the DMAI. When a register is read, the default value is returned regardless of the value used when the register is programmed. The default value is indicated by '(x)' in unused bit positions in the register formats, which are illustrated in table 2.

Table 1. DMAI ADDRESS MAP

ADDRESS BITS[1,2] 7 6 5 4 3 2 1 0	ACRONYM	REGISTER NAME	MODE	AFFECTED BY RESET
d d 0 0 0 0 0 0	CSR	Channel Status Register	R/W[3]	Yes
d d 0 0 0 0 0 1	CER	Channel Error Register	R	Yes
d d 0 0 0 0 1 0		Reserved		
d d 0 0 0 0 1 1		Reserved		
d d 0 0 0 1 0 0	DCR	Device Control Register	R/W	Yes
d d 0 0 0 1 0 1	OCR	Operation Control Register	R/W	Yes
d d 0 0 0 1 1 0	SCR	Sequence Control Register	R/W[4]	No
d d 0 0 0 1 1 1	CCR	Channel Control Register	R/W	Yes
d d 0 0 1 0 0 0		Reserved		
d d 0 0 1 0 0 1		Reserved		
d d 0 0 1 0 1 0	MTCH	Memory Transfer Counter High	R/W	No
d d 0 0 1 0 1 1	MTCL	Memory Transfer Counter Low	R/W	No
d d 0 0 1 1 0 0	MACH	Memory Address Counter High	R/W[4]	No
d d 0 0 1 1 0 1	MACMH	Memory Address Counter Middle High	R/W	No
d d 0 0 1 1 1 0	MACML	Memory Address Counter Middle Low	R/W	No
d d 0 0 1 1 1 1	MACL	Memory Address Counter Low	R/W	No
d d 0 1 d d d d		Reserved		
d d 1 0 0 0 d d		Reserved		
d d 1 0 0 1 0 0		Reserved		
d d 1 0 0 1 0 1	IVR	Interrupt Vector Register – Normal	R/W	Yes
d d 1 0 0 1 1 0		Reserved		
d d 1 0 0 1 1 1	IVR	Interrupt Vector Register – Error	R/W	Yes
d d 1 0 1 0 d d		Reserved		
d d 1 0 1 1 0 0		Reserved		
d d 1 0 1 1 0 1	CPR	Channel Priority Register	R/W[4]	No
d d 1 0 1 1 1 0		Reserved		
d d 1 0 1 1 1 1		Reserved		
d d 1 1 d d d d		Reserved		

NOTES:
1. A0 = 0 for UDSN asserted. A0 = 1 for LDSN asserted.
2. 'd' designates don't care
3. A write to this register may perform a status resetting operation
4. This register is a dummy register present only to provide compatibility with other 68K family DMA controllers. A write to this register has no effect on the DMAI

Table 2. REGISTER BIT FORMATS

DEVICE CONTROL REGISTER

	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
DCR	EXTERNAL REQUEST MODE 0 = BURST 1 = CYCLE STEAL	NOT USED (0)	NOT USED (1)	NOT USED (1)	NOT USED (*)	NOT USED (0)	NOT USED (0)	NOT USED (0)

*Should be programmed as '0' for SIZE (OCR[5:4]) = 00 and as '1' otherwise. When read, the value of this bit is OCR[5]. .OR.OCR[4].

OPERATION CONTROL REGISTER (OCR)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
OCR	DIRECTION 0 = MEM TO DEV 1 = DEV TO MEM	NOT USED (0)	OPERAND SIZE 00 = BYTE 01 = WORD (16 BIT) 10 = LONG WORD* 11 = WORD (32-BIT)*		NOT USED (0)	NOT USED (0)	NOT USED (1)	NOT USED (0)

*Long word and 32-bit word modes are not supported by 68440. 32-bit word mode is not supported by 68450.

SEQUENCE CONTROL REGISTER (SCR)

	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
SCR	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (1)	NOT USED (0)	NOT USED (0)

CHANNEL CONTROL REGISTER (CCR)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
CCR	START 0 = NO 1 = YES	NOT USED (0)	NOT USED (0)	SOFTWARE ABORT 0 = NO 1 = YES	INTERRUPT ENABLE 0 = NO 1 = YES	NOT USED (0)	NOT USED (0)	NOT USED (0)

CHANNEL STATUS REGISTER (CSR)

	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
CSR	CHANNEL OPERATION COMPLETE 0 = NO 1 = YES	NOT USED (0)	NORMAL DEVICE TERMINATE 0 = NO 1 = YES	ERROR 0 = NO 1 = YES	CHANNEL ACTIVE 0 = NO 1 = YES	NOT USED (0)	NOT USED (0)	READY INPUT STATE 0 = LOW 1 = HIGH

CHANNEL ERROR REGISTER (CER)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
CER	NOT USED (0)	NOT USED (0)	NOT USED (0)	ERROR CODE 00000 = NO ERROR 01001 = BUS ERROR 10001 = SOFTWARE ABORT				

CHANNEL PRIORITY REGISTER (CPR)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
CPR	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)

Device Control Register (DCR)

[15] External Request Mode

This bit selects whether the DMAI operates in burst or cycle steal mode.

0 Burst mode. This mode allows a device to request the transfer of multiple operands using consecutive bus cycles. In this mode the request (REQN) line is an active low input which is asserted by the device to request an operand transfer. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. If the request line is active when the DMAI asserts ACKN, and remains active at least until the DMAI asserts device transfer complete (DTCN), the DMAI recognizes a valid request for another operand, which will be transferred during the next bus cycle. If the request line is negated before the DMAI asserts DTCN, the DMAI relinquishes the bus and waits for the next request, but the current transfer will be completed

1 Cycle steal mode. In this mode, the device requests an operand transfer by generating a falling edge on the request (REQN) line. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN)

output. The request line must be in the inactive state for at least one clock cycle before a request is made. After a request has been asserted, it must remain at the assertion level for at least one clock cycle. If another request is received before the first operand part of a former request is acknowledged, the second request is not recognized. Normally, the DMAI will relinquish the bus after servicing a valid request. However, if the device generates a new request before the DMAI asserts DTCN for the last operand part, the DMAI will retain ownership of the bus and that request will be serviced before the bus is relinquished.

Operation Control Register (OCR)

[7] Direction

0 Transfer is from memory to device.

1 Transfer is from device to memory.

[5:4] Operand Size

The programming of these bits determine whether UDSN, LDSN, or both are generated during the transfer cycle and the increment by which the memory address counter (MAC) is changed in each transfer cycle.

00 Byte. The operand size is 8 bits. The MAC is incremented by one after each operand transfer. If the LSB of the MAC is a '0', UDSN is asserted during the transfer. If the LSB of a MAC is a '1', LDSN is asserted during the transfer. The transfer counter decrements by one before each byte is transferred.

01 Word. The operand size is 16 bits. The MAC is incremented by two after each operand transfer. The value of the LSB of the MAC is ignored and both UDSN and LDSN are asserted during the transfer. The transfer counter decrements by one before each word is transferred.

10 Long word. The operand size is 32 bits. The operand is transferred as two 16-bit words. The MAC is incremented by two after each 16-bit word is transferred. The value of the LSB of the MAC is ignored and both UDSN and LDSN are asserted during the transfer. The transfer counter decrements by one before the entire long word is transferred. Note that this mode is not implemented in the 68440.

11 Double word. The operand size is 32 bits. The operand is transferred as a single 32-bit word. The MAC is incremented by four after each operand transfer. The value of the two LSBs of the MAC is ignored (the A1 output will always be a zero in this mode) and both UDSN and LDSN are asserted during the transfer. The transfer counter decrements by one before the double word is transferred. Note that this mode is not implemented in the 68440 or 68450; it is included in the DMAI to support VME bus operations.

Sequence Control Register (SCR)

This register serves no function in the DMAI. It is included only to provide compatibility with the programming for the 68440 and 68450 DMA controllers.

Channel Control Register (CCR)

[7] Start Operation

0 No start pending.

1 Start operation. The start bit is set to initiate operation of the DMAI. The memory address counter and the memory transfer counter should have been previously initialized, and all bits of the channel status register (CSR) should have previously been reset. The DMAI initiates operation by clearing any pending requests, clearing the start bit, and setting the channel active bit in the CSR. The DMAI is then ready to receive requests for an operation. The channel cannot be started if any of the internal status bits in the CSR (CSR[15:11]) have not been cleared.

A pending start cannot be reset by a write to the register. START can be cleared only by the DMAI when it starts operation or by setting the software abort bit (CCR[4]).

[4] Software Abort

0 Do not abort.

1 Abort operation. Setting this bit terminates the current operation of the DMAI and places it in the IDLE state. The channel operation complete and error bits in the CSR are set, the channel active bit in the CSR is reset, and an ABORT ERROR condition is signaled in the CER. Setting this bit causes a pending start to be reset.

[3] Interrupt Enable

0 Interrupts not enabled.

1 Enable interrupts. An interrupt request is generated if the channel operation complete bit in the CSR is set. When the IACKN input is asserted, the DMAI returns the normal interrupt vector if the error bit in the CSR is not set, or the error interrupt vector if error is set.

Channel Status Register (CSR)

A read of this register provides the status of the DMAI. The COC, NDT, and ERR bits can be cleared by writing a '1' to the bit positions of the register which are to be cleared. Those bit positions which are written with a '0' remain unaffected.

[15] Channel Operation Complete

This bit is set following the termination, whether successful or not, of any DMAI operation and indicates that the DMA transfer has completed. This bit must be cleared to start another channel operation.

[13] Normal Device Termination

This bit is set when the device terminates the DMAI operation by asserting the DONEN line while the device was being acknowledged. This bit must be cleared to start another channel operation.

[12] Error

This bit is used to report that the DMAI's operation was terminated due to the occurrence of an error. The condition which caused the error can be determined by reading the channel error register (CER). This bit must be cleared to start another channel operation. When this bit is cleared, the CER is also cleared.

[11] Channel Active

This bit is set after the channel has been started and remains set until the channel operation terminates. It is then automatically cleared by the DMAI. The bit is unaffected by the write operations.

[8] Ready Input State

This bit reflects the state of the RDYN input at the time the CSR is read. The bit is a '0' if RDYN is low and a '1' if RDYN is high. This bit is unaffected by write or reset operations.

Channel Error Register (CER)

[4:0] Error Code

This field indicates the source of error when an error is indicated in CER[12]. The contents of this register are cleared when CER[12] is cleared.

00000 No error.

01001 Bus error. A bus error occurred during the last bus cycle generated by the DMAI. See rerun description in OPERATION section.

10001 Software abort. The channel operation was terminated by a software abort command. See CCR[4].

Channel Priority Register (CPR)

This register serves no function in the DMAI. It is included only to provide compatibility with the programming for the other 68K family DMA controllers.

Memory Address Counter (MACH, MACMH, MACML, MACL)

The 32-bit memory address counter is used to program the memory location where the first operand to be transferred is located or is to be transferred to, depending on the direction of transfer. The counter must be initialized prior to beginning the transfer of a block of data and then increments automatically depending on the operand length, as de-

scribed in the Operation Control Register description.

Only the least significant 24 bits of the counter (MACMH, MACML, and MACL) are implemented in the DMAI. The most significant byte of the counter, MACH, is provided only to allow compatibility with programming of the 68440 and 68450. Writing to MACH has no effect on the DMAI operation. Reading MACH always returns H'00'.

Memory Transfer Counter (MTCH, MTCL)

The 16-bit memory transfer counter programs the number of operands to be transferred by the DMAI. The counter must be initialized prior to beginning the transfer of a block of data and then decrements once per operand transfer (regardless of operand size) until it reaches the terminal value of zero. Channel operation then terminates and the COC bit in the CSR will be asserted.

Interrupt Vector Register (IVR)

The IVR contains the value to be placed on the data bus upon receipt of an interrupt acknowledge from the MPU. Only the seven most significant bits of the programmed value are used by the DMAI. The output vector from the DMAI contains a zero in the least significant bit position if a normal termination occurred (error bit not set) and contains a one in the least significant bit position if termination was due to an error (error bit set).

The contents of this register are initialized to H'0F' by a reset. The value returned will be H'0F', regardless of the error state, until the register is programmed by the MPU.

To provide compatibility with the other 68K family DMA controllers, the IVR has two addresses (see table 1). If program compatibility is required, the value written at the normal IVR address should have a zero as its LSB, and the value written at the error IVR address should be the same but with the LSB equal to one.

OPERATION

A DMAI operation proceeds in three principal phases. During the initialization phase, the MPU configures the channel control registers, loads the initial memory address and transfer count, and starts the channel. During the transfer phase, the DMAI accepts requests for transfers from the device, arbitrates for and acquires ownership of the bus, and provides for addressing and bus controls for the transfers. The termination phase occurs after the operation is complete, when the DMAI reports the status of the operation.

Operation Initiation

After having programmed the control registers, the memory address counter, and the memory transfer counter, the MPU sets the start bit (CCR[7]). The DMAI initiates the operation by clearing any pending requests, clearing the start bit, and setting the channel active bit in the CSR. The DMAI is then ready to receive valid requests for an operation.

The channel cannot be started if any of the internal status bits in the CSR (CSR[15:11]) have not been cleared. An error is not signaled if this condition occurs. The only indication of this state is that the start bit remains set in the CCR. A pending start cannot be reset by a write to the register. START can be cleared only by the DMAI when it starts operation or by setting the software abort bit (CCR[4]).

Device/DMAI Communication

Communication between the peripheral device and the DMAI is accommodated by five signal lines:

Request (REQN)

The device makes a request for service by asserting the request line. The DMAI can operate in either the burst request mode or the cycle stealing request mode, as programmed by the external request mode bit (DCR[15]).

The burst mode allows a device to request the transfer of multiple operands using consecutive bus cycles. In this mode the request line is an active low input. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. If the request line is active when the DMAI asserts ACKN, and remains active at least until the DMAI asserts device transfer complete (DTCN), the DMAI recognizes a valid request for another operand, which will be transferred during the next bus cycle. If the request line is negated before the DMAI asserts DTCN, the DMAI relinquishes the bus and waits for the next request. For long word transfers (2 x 16), the request must be asserted at least until the acknowledge for the second part of the operand has been asserted.

In the cycle steal mode, the device requests an operand transfer by generating a falling edge on the request line. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. The request line must be in the inactive state for at least one clock cycle before a request is made. After a request has been asserted, it must remain at the assertion level for at least one clock cycle. If another request is received before the first operand part of a former request is acknowledged, the second request is not recognized. Normally, the DMAI will relinquish the bus after servicing a valid request. However, if the device generates a new request before the DMAI asserts DTCN for the last operand part, the DMAI will retain ownership of the bus and that request will be serviced before the bus is relinquished.

Acknowledge (ACKN)

The DMAI asserts the acknowledge line, which implicitly addresses the device making the request, during transfers to and from the device. The line may be used to control buffering circuits between the data bus and the MPU bus.

During burst mode, REQN must not be disasserted for less than one CLK period plus four RC time constants, where R is the value of the resistor used for the pullup on BRN and C has a typical value of 20pF.

Ready (RDYN)

Ready is an active low input which is asserted by the requesting device to indicate to the DMAI that valid data has either been stored or put on the bus. If negated, it indicates that the data has not been stored or presented, causing the DMAI to enter wait states until RDYN is asserted. RDYN can be held low continuously if the device is fast enough so that wait states are not required. The current state of the ready input is reflected in CSR[8].

Done (DONEN)

Done is a bidirectional active low signal. As an output, it is asserted and negated by the DMAI concurrent with the ACKN output of the last operand part to indicate to the device that the memory transfer count is exhausted and that the DMAI's operation is completed as a result of that transfer.

The DMAI also monitors the state of the line while acknowledging a device. If the device asserts DONEN, the DMAI will terminate operation after the transfer of the current operand. In this case the DMAI clears the channel active bit and sets the channel operation complete and normal device termination bits in the CSR. If both the DMAI and the device assert DONEN, the device termination is not recognized, but the operation does terminate.

Device Transfer Complete (DTCN)

DTCN is an active low output which is asserted by the DMAI to indicate to the device that the requested data transfer is complete. On a write to memory operation, it indicates that the data provided by the device has been successfully stored. On a read from memory operation, it indicates to the device that the data from memory is present on the data bus and should be latched. DTCN is not asserted if assertion of the RERUNN input terminates the bus cycle.

Bus Arbitration

Upon receiving a valid request for a transfer from the device, the DMAI will arbitrate for and obtain ownership of the system bus.

The DMAI indicates that it wishes to become the bus master by asserting its bus request (BRN) output. This is a wire-ORed signal that indicates to the MPU that some external device requires control of the bus. The processor is effectively at a lower priority level than external devices and will relinquish the bus after it has completed the last bus cycle it has started. The processor puts the bus up for external arbitration by asserting its bus grant (BGN) output. This signal may be routed through a daisy chain (such as provided by the DMAI) or through some other priority-encoded network. When the DMAI making the bus request receives the bus grant (indicated by its BGN input being asserted), it is to be the next bus master. It waits until address strobe (ASN), data transfer acknowledge (DTACKN) and bus grant acknowledge (BGACKN) become inactive and then assumes ownership of the bus by asserting its own BGACKN output. The DMAI then negates the BRN output and proceeds with the data transfer phase. After this phase is completed, the DMAI relinquishes bus ownership by negating the BGACKN output.

In burst DMA mode, detection of an active low request input after the DMAI operation has been started will begin the bus arbitration cycle. However, if the device negates its request at least one clock cycle before the DMAI asserts BGACKN, the DMAI will negate its bus request and will not assume ownership of the bus.

Data Transfers

The actual transfer of data between the memory and the device occurs during the data transfer phase. All transfers occur during a single cycle except in the case of long word operands, in which case two cycles are used to transfer the operand as two 16-bit words. The transfers take place using a 'single address' protocol; the DMAI addresses the memory via the bus address lines, while the device is implicitly addressed via the acknowledge output.

When a request is generated using the request method programmed in the control register, the DMAI obtains the bus and asserts acknowledge to notify the device that a transfer is to take place. The DMAI asserts all S68000 bus control signals needed for the transfer and holds them until the device responds with ready. The bus cycle then terminates normally. Ready may be tied low (asserted) if the device is fast enough.

When the transfer is from memory to the device, data is valid when DTACKN is asserted by the memory and remains valid until the data strobe(s) are negated. The assertion of DTCN from the DMAI can be used to latch the data, as the data strobes are not removed until one-half clock after the assertion of DTCN.

When the transfer is from device to memory, the data must be valid on the bus before the DMAI asserts the data strobe(s). The device indicates valid data by asserting ready. The DMAI then asserts the strobes and holds them asserted until the memory accepts the data, indicated by the assertion of DTACKN. The DMAI then asserts DTCN and negates the data strobes.

Flow charts for these operations are shown in figures 1 and 2. Refer to the timing section for the equivalent timing diagrams.

Operation Termination

Termination of the block transfer occurs under the conditions detailed below.

Terminal Count

As part of each transfer of an operand, the DMAI decrements the memory transfer counter. If this counter is decremented to zero, the operand is the last operand of the block. The DMAI operation is complete and it notifies the device of completion by asserting the DONEN output during the last operand transfer cycle. When the transfer has been completed, the channel active bit in the CSR is cleared and the COC bit is set, unless an error occurs.

Device Termination

The DMAI monitors the state of the DONEN line while acknowledging a device transfer request. If the device asserts DONEN, the DMAI will terminate operation after the transfer of the current operand. When the transfer has been completed, the DMAI clears the channel active bit and sets the COC and normal device termination bits in the CSR. If both the DMAI and the device assert DONEN, the device termination is not recognized, but the operation does terminate.

Software Abort

The software abort bit (CCR[4]) allows the MPU to abort the current operation of the DMAI. The COC and error bits in the CSR are set, the channel active bit in the CSR is cleared, and an abort error condition is signaled in the CER.

Rerun Error

The DMAI provides a rerun input (RERUNN) to indicate a bus exception condition. RERUNN must arrive prior to or in coincidence with DTACKN in order to be recognized, and the DMAI verifies that the line has been stable for two clock cycles before acting on it. The occurrence of a rerun during a DMAI bus cycle forces it to terminate the bus cycle in an orderly manner.

When the assertion of rerun is verified, the DMAI stops operation and three-states the data, address, and control lines, except BGACKN, so that it retains ownership of the bus. It remains halted until rerun becomes inactive, and then re-tries the last bus cycle. If rerun is asserted again, the DMAI stops DMA operation, releases the bus, sets the error and COC bits in the CSR, clears the active bit in the CSR, and sets the error code in the CER to indicate a bus error.

While stopped due to assertion of rerun, the DMAI does not generate any bus cycles and will not honor any requests until it is removed. However, the DMAI still recognizes requests.

Error Recovery Procedure

If an error occurs during a DMA transfer such that the DMAI stops the DMA operation, information is available to the operating system for an error recovery routine.

The information available to the operating system consists of the memory address counter, the memory transfer counter, and the control, status, and error registers. The DMAI decrements the memory transfer counter before attempting a DMA operation, so the register will contain the count minus one of the attempted transfer. The memory address counter will contain the address at which the DMA operation was attempted.

Reset

The reset input (RESETN) provides a means of resetting and initializing the DMAI from an external source. If the DMAI is a bus master when reset is received, the DMAI relinquishes the bus. Reset clears the control and error registers, sets all bits of the status register except CSR[8] to zero, and initializes the interrupt vector register to H'0'F.

Interrupts

The interrupt enable bit (CCR[3]) determines whether the DMAI generates interrupt requests. When the bit is set, an interrupt request is generated if the channel operation complete bit in the CSR is set. When the IACKN input is asserted, and the DMAI has an interrupt request pending, the DMAI returns an interrupt vector on the data bus.

The interrupt vector issued is the contents of the IVR. Only the seven most significant bits of the programmed value are used by the DMAI. The vector from the DMAI contains a zero in the LSB position if a normal termination occurred (error bit not set) and contains a one in the LSB position if termination was due to an error (error bit set).

The contents of this register are initialized to H'0F' by a reset. The value returned will be H'0F', regardless of the error state, until the register is programmed by the MPU.

To provide compatibility with the other 68K family DMA controllers, the IVR has two addresses (see table 1). If program compatibility is required, the value written at the

Direct Memory Access Interface (DMAI)

SCB68430

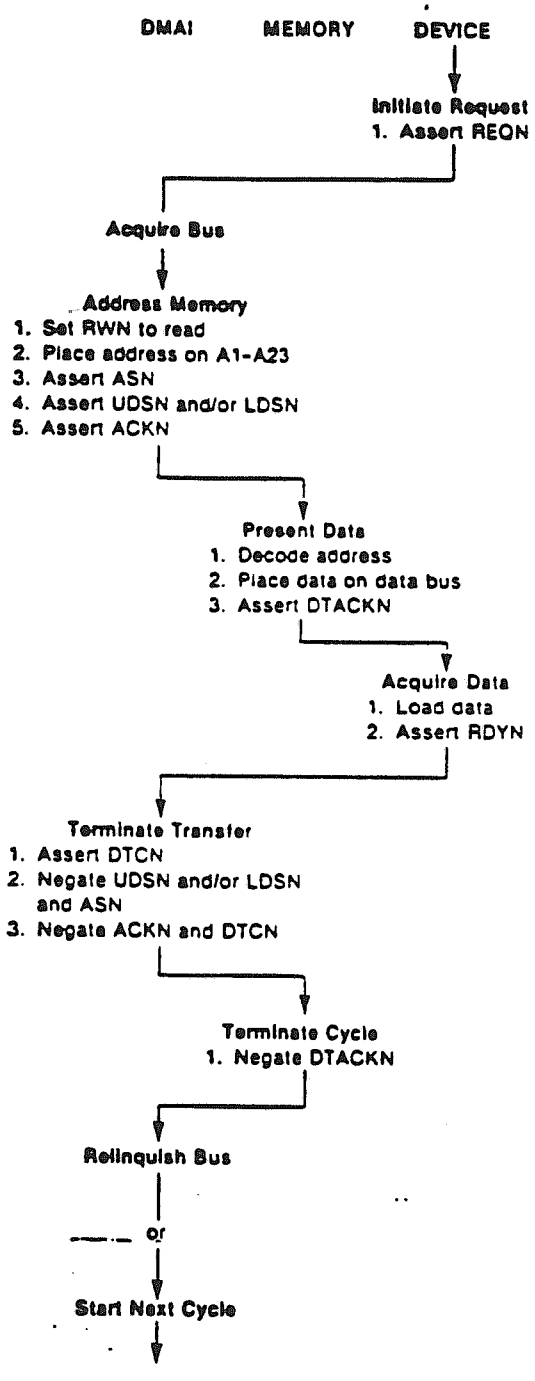


Figure 1. Transfer from Memory to Device Flowchart

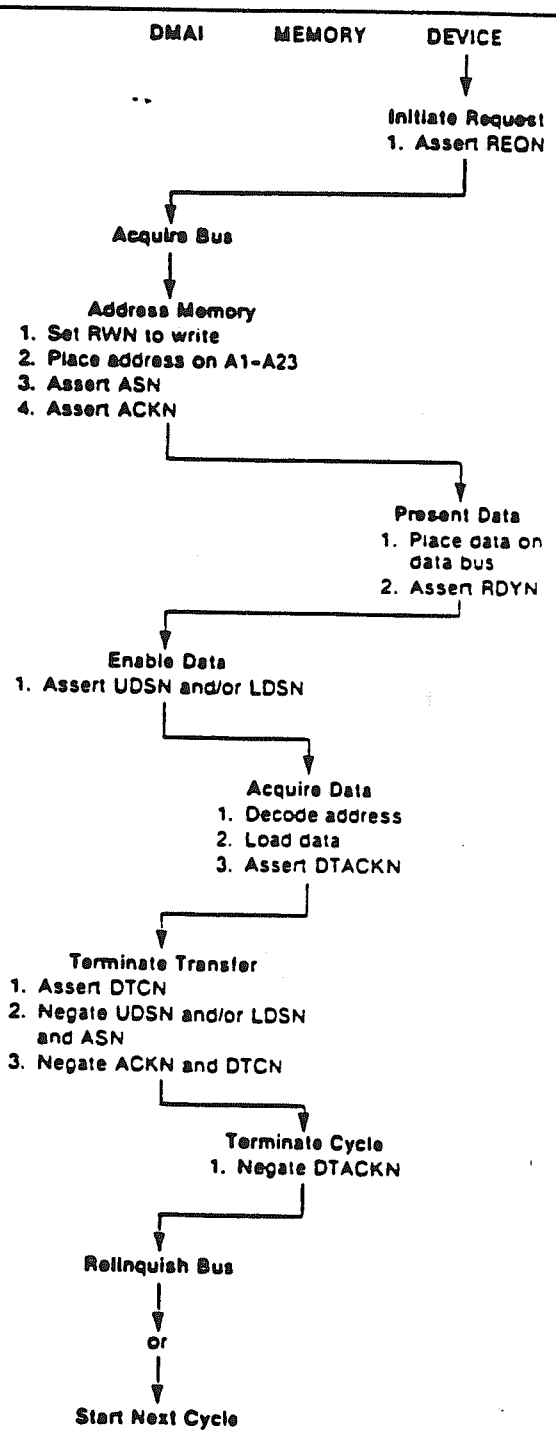


Figure 2. Transfer from Device to Memory Flowchart

normal IVR address should have a zero as its LSB, and the value written at the error IVR address should be the same but with the LSB equal to one.

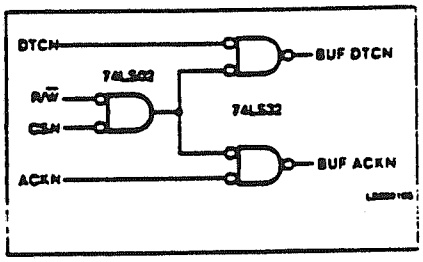
APPLICATIONS

Figure 3 illustrates a typical interconnection of the DMAI in a 68000 based system.

DESIGN NOTE

When clearing the error bit in CSR (bit 12) after a DMAI abort due to a double RERUNN, ACKN and DTCN will both go low concurrent with CSN and DTACKN for one CLK cycle.

To prevent the possibility that the device may misinterpret these signals, it is suggested that these signals be ANDed with CSN (see figure below).



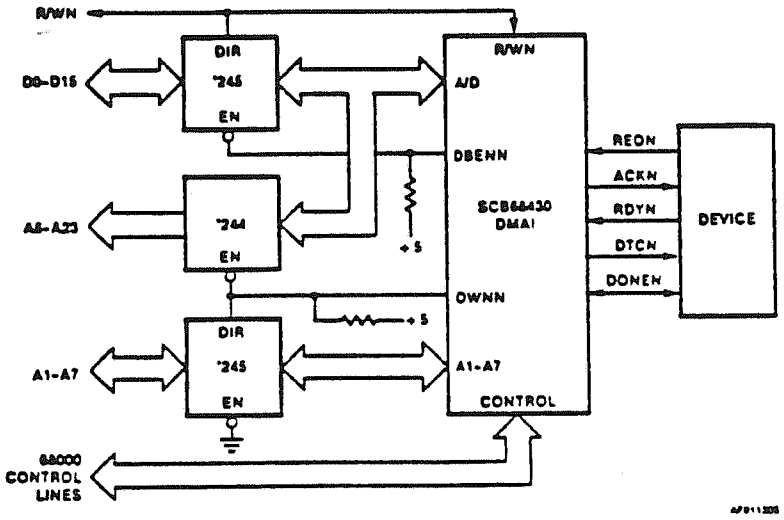


Figure 3. DMAI Application

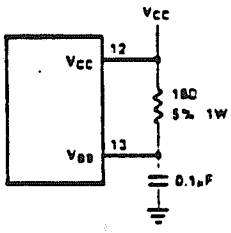


Figure 4. Recommended V_{BB} Test Circuit

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Supply voltages V_{CC} and V_{BB}	−0.5 to +7.0	V
Input voltage	−0.5 to +5.5	V
Operating temperature range[2]	0 to +70	°C
Storage temperature	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS V_{CC} = 5.0V ± 5%, V_{BB} = Figure 4, T_A = 0°C to +70°C[3,4,7]

PARAMETER		TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{IL}	Input low voltage			0.8	V
V_{IH}	Input high voltage		2.0		V
V_{OL}	Output low voltage	I_{OUT} = 5.3mA		0.5	V
V_{OH}	Output high voltage, all outputs except open collector outputs[5]	I_{OUT} = −400µA	2.5		V
I_{IL}	Input low current	V_{IN} = 0.4V		−400	µA
I_{IH}	Input high current	V_{IN} = 2.7V		20	µA
I_{OC}	Open collector off state current[5]	V_{OUT} = 2.4V		20	µA
I_{SC}	Output short circuit current[6]	V_{CC} = max	−40	−100	mA
I_{CC}	V_{CC} supply current	V_{CC} = max		130	mA
I_{BB}	V_{BB} supply current			275	mA

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other conditions other than those indicated in the Electrical Characteristics section of this data sheet is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. Parameters are valid over specified temperature range.
4. All voltage measurements are referenced to ground (V_{SS}). For testing, all signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V as appropriate.
5. IRQN, BRN, DONEN, and OWNN are open collector outputs.
6. No more than one output should be connected to ground at one time.
7. Capacitive test load is 100pF for all pins except DTCN which has a 35pF capacitive test load.

AC ELECTRICAL CHARACTERISTICS V_{CC} = 5.0V ± 5%, V_{BB} = Figure 4, T_A = 0°C to +70°C[3,4,7]

NO.	FIGURE	CHARACTERISTICS	TENTATIVE LIMITS				UNIT
			10MHz		12.5MHz		
			Min	Max	Min	Max	
1	5	A1 – A7, ASN, RWN, set-up to UDSN, LDSN low	0		0		ns
2	5	D0 – D15 3-state to invalid data from ASN, CSN, and UDSN or LDSN low	10		10		ns
3	5	DTACKN 3-state to high from ASN, CSN, and UDSN or LDSN low	10		10		ns
4	5	CSN low after UDSN or LDSN low		25		25	ns
5	5, 6	DBENN low after ASN and CSN low		60		60	ns
6	5	D0 – D15 valid data from ASN, CSN, and UDSN or LDSN low		100		100	ns
7	5	DTACKN low after D0 – D15 valid data	−15	30	−15	30	ns
8	5	A1 – A7, ASN, RWN or CSN hold after UDSN and LDSN high	0		0		ns
9	5, 6	DBENN high from either ASN or CSN high		45		45	ns
10	5	D0 – D15 to 3-state from UDSN and LDSN high		80		80	ns
11	5	D0 – D15 to invalid data from UDSN and LDSN high	10		10		ns
12	5, 6	DTACKN high from UDSN and LDSN high		55		55	ns
13	5, 6	DTACK 3-state from either CSN or ASN high		85		85	ns
14	6	A1 – A7, ASN, RWN set-up to UDSN, LDSN low	50		50		ns/s
15	6	CSN set-up before UDSN or LDSN low	20		20		ns
16	6	DTACKN 3-state to high after CSN and ASN low	10		10		ns
17	6	D0 – D15 valid after UDSN or LDSN low		0		0	ns
18	6	DTACKN low from UDSN or LDSN low		100		100	ns
19	6	UDSN and LDSN low time	115		100		ns
20	6	A1 – A7 hold after UDSN and LDSN high	0		0		ns
21	6	ASN, RWN and CSN hold after UDSN and LDSN high	0		0		ns

AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	FIGURE	CHARACTERISTICS	TENTATIVE LIMITS				UNIT
			10MHz		12.5MHz		
			Min	Max	Min	Max	
22	6	D0 – D15 hold after UDSN and LDSN high	0		0		ns
23	7	DBENN low from last low of ASN, IACKN, LDSN		65		65	ns
24	7	D0 – D7 valid after last low of ASN, IACKN, LDSN		105		105	ns
25	7	DTACKN 3-state to high after last low of ASN, IACKN, LDSN		100		100	ns
26	7	DTACKN low after last low of ASN, IACKN, LDSN		110		110	ns
27	7	DBENN high after first high of ASN, IACKN, LDSN		55		55	ns
28	7	D0 – D7 hold after first high of ASN, IACKN, LDSN		60		60	ns
29	7	D0 – D7 3-state after first high of ASN, IACKN, LDSN		80		80	ns
30	7	DTACKN high after first high of ASN, IACKN, LDSN		60		60	ns
31	7	DTACKN 3-state after first high of ASN, IACKN, LDSN		95		95	ns
32	8	BRN high from CLK high		65		65	ns
33	8, 11, 12	BGACKN low from CLK low		75		75	ns
34	8, 11, 12	OWNN low from CLK high		75		75	ns
35	8	BGACKN high from CLK low		75		75	ns
36	8,11,12	OWNN high from CLK high (load dependent)		50		50	ns
37	10	REQN set-up before CLK low	30		30		ns
38	10	REQN hold after CLK high	20		20		ns
39	10	BRN low from CLK high		80		80	ns
41	11, 12	ASN, UDSN, LDSN, RWN 3-state to high from CLK low		75		75	ns
43	11, 12	A1 – A23 to valid ASN	0		0		ns
44	11, 12	ASN low from CLK high		65		65	ns
45	11, 12	LDSN, UDSN low from CLK high		90		90	ns
46	11, 12	ACKN low from CLK high		65		65	ns
47	11, 12	DTACKN set-up to CLK high	30		30		ns
48	11, 12	RDYN set-up to CLK low	30		30		ns
49	11, 12	DTCN low from CLK high		70		70	ns
50	11, 12	ASN high from CLK high		75		75	ns
51	11, 12	LDSN, UDSN, high from CLK high		90		90	ns
52	11, 12	DTACKN, RDYN hold after CLK high	0		0		ns
–	11, 12	ASN, LDSN, UDSN, high from DTCN low	–20		–20		ns
53	11, 12	ACKN high from CLK high		50		50	ns
54	11, 12	DTCN high from CLK high		50		50	ns
55	11, 12	Address valid after CLK low	10		10		ns
–	11, 12	Address valid after ASN high	0		10		ns
56	11, 12	DONEN (output) low from CLK low		120		120	ns
57	11, 12	DONEN (output) high from CLK high		50		50	ns
58	11, 12	DONEN (input) set-up low before CLK low	30		30		ns
59	11, 12	DONEN (input) hold low after CLK high	0		0		ns
60	11, 12	BGACKN, ASN, UDSN, LDSN, RWN to 3-state from CLK low		75		75	ns
62	11, 12	A1 – A23 valid to 3-state from CLK high		100		100	ns
63	12	R/WN low from CLK high		65		65	ns
64	12	R/WN high from CLK high		75		75	ns
65	13	RERUNN set-up low before CLK high	30		30		ns
66	13	RERUNN hold low from CLK high	20		20		ns
67	13	A1 – A23 to idle state from CLK low		100		100	ns
68	13	A1 – A23 to valid after CLK low		85		85	ns

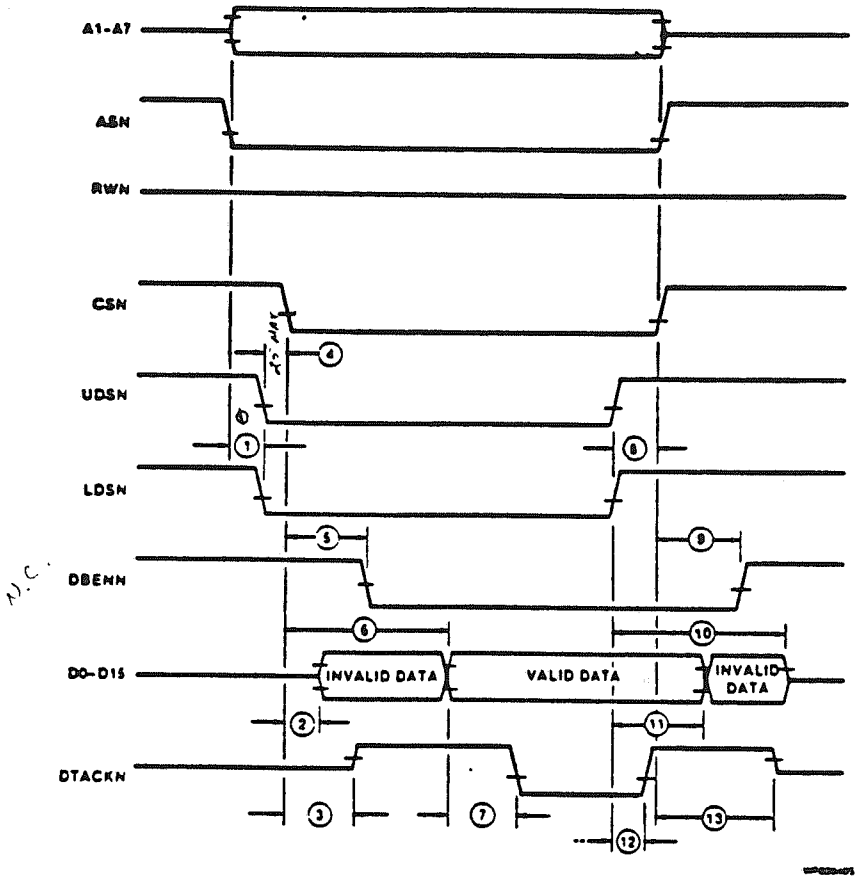


Figure 5. DMAI Read Timing

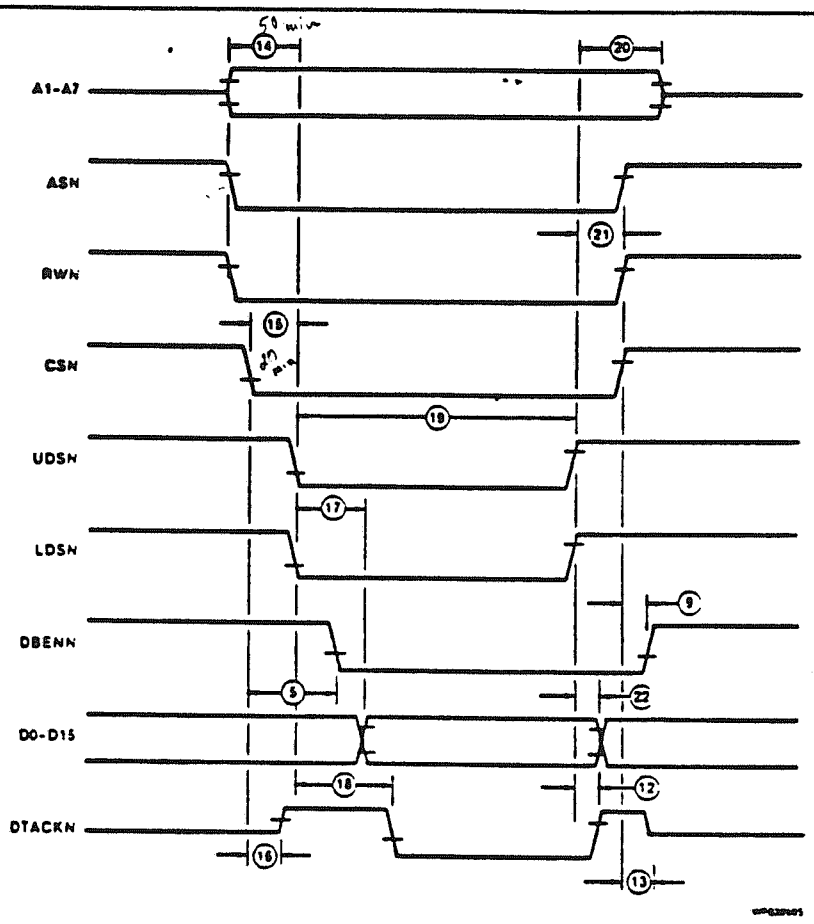


Figure 6. DMAI Write Timing

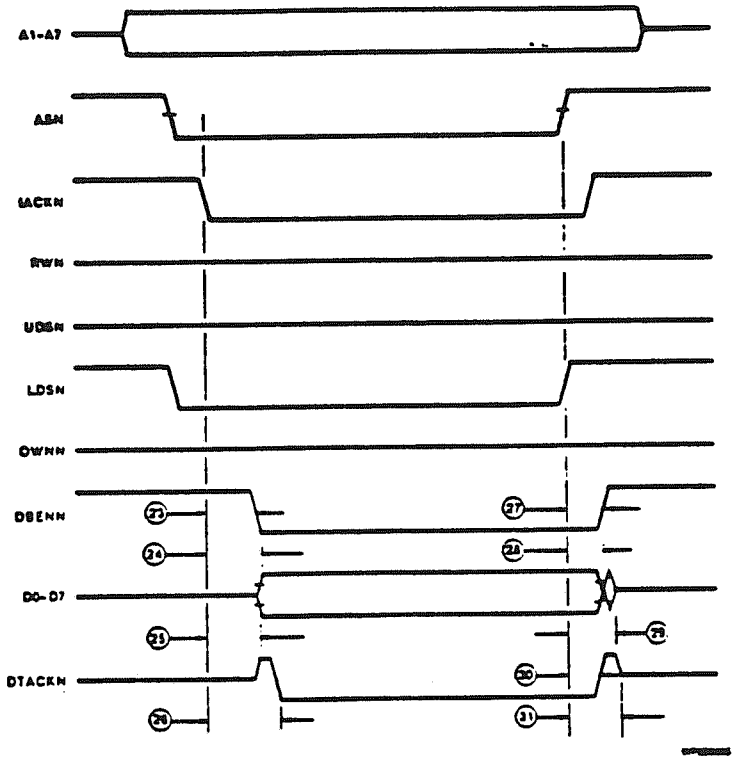
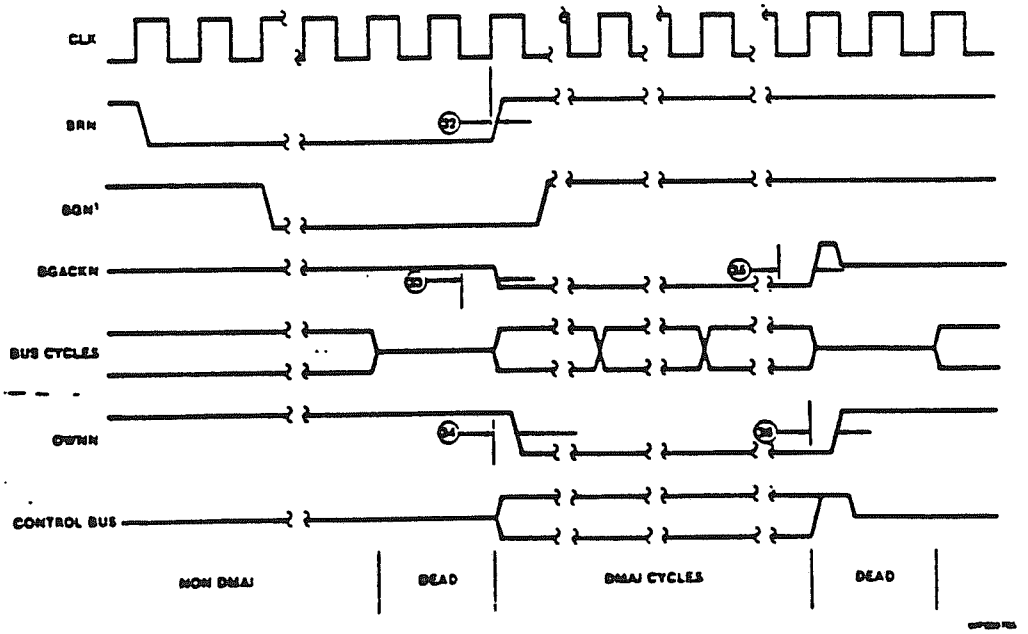


Figure 7. CPU IACK Cycle to DMAI



NOTES:
1. Device will become master if BGN is asserted concurrent with or later than REON (same clock edge or later).
2. ASN, DTACKN and BGACKN must be negated in order for DMAI to become master. Timing assumes all these happen concurrent with BGN — if not, it is from the latest signal which is negated.

Figure 8. DMAI Bus Arbitration Timing

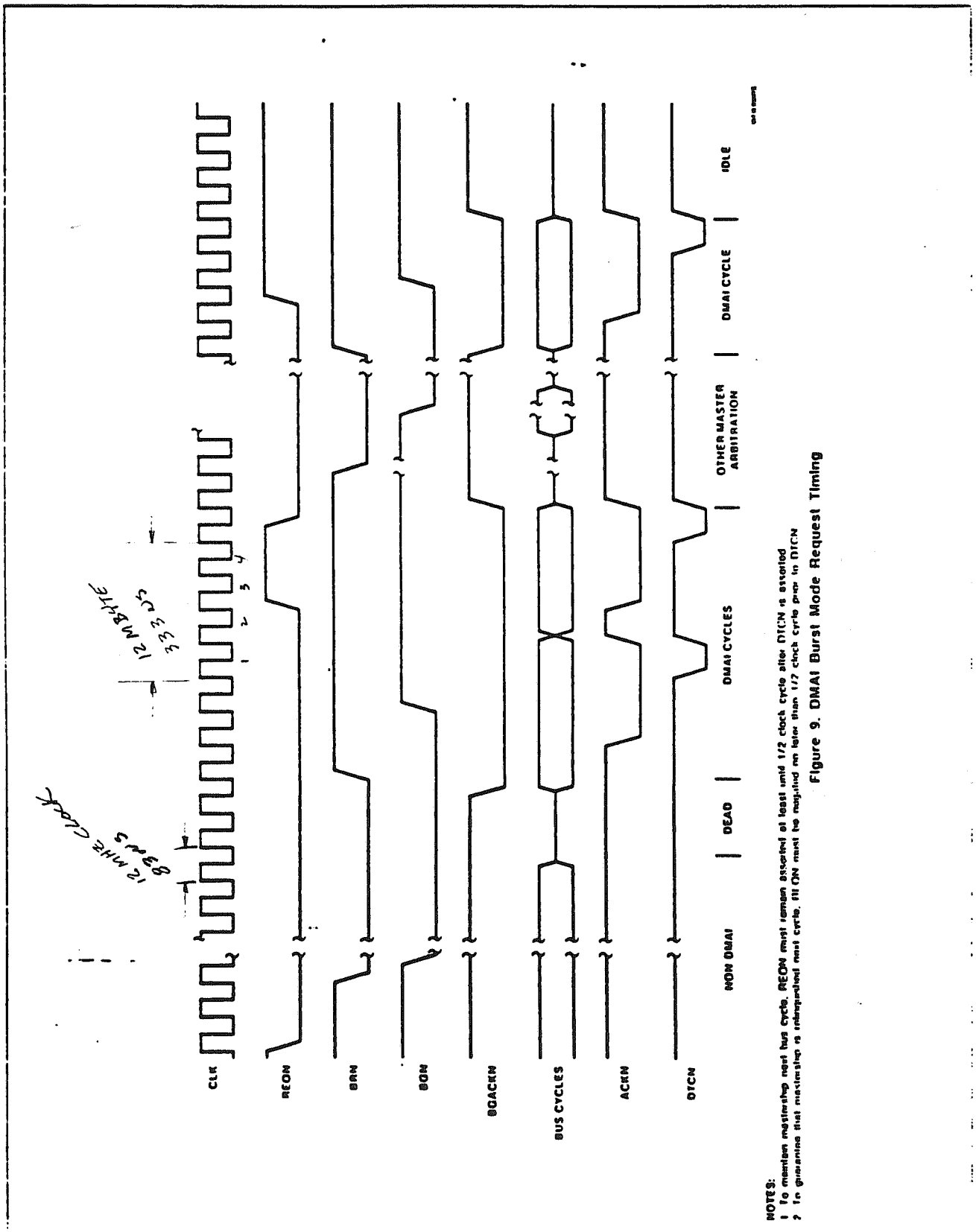


Figure 9. DMAI Burst Mode Request Timing

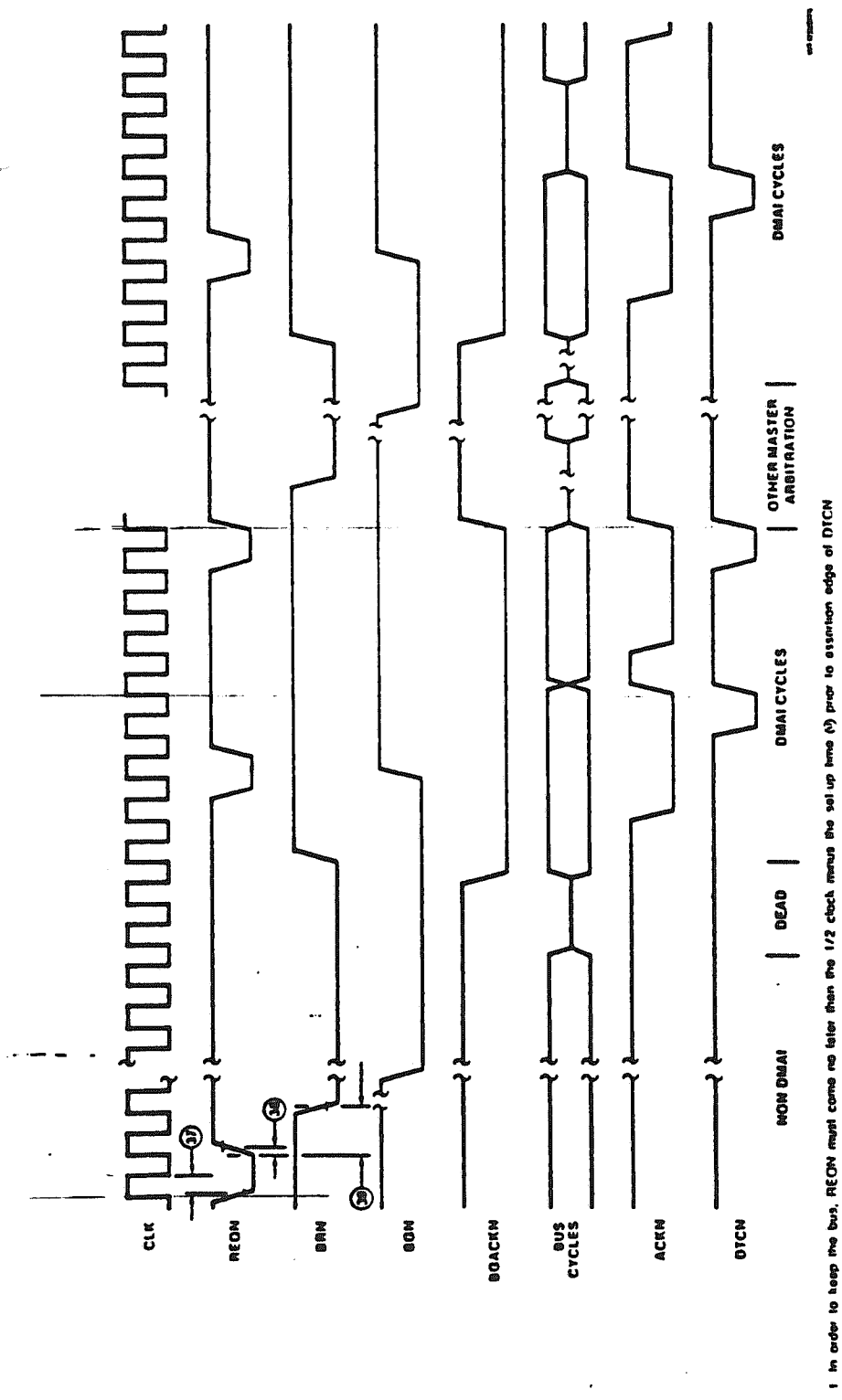
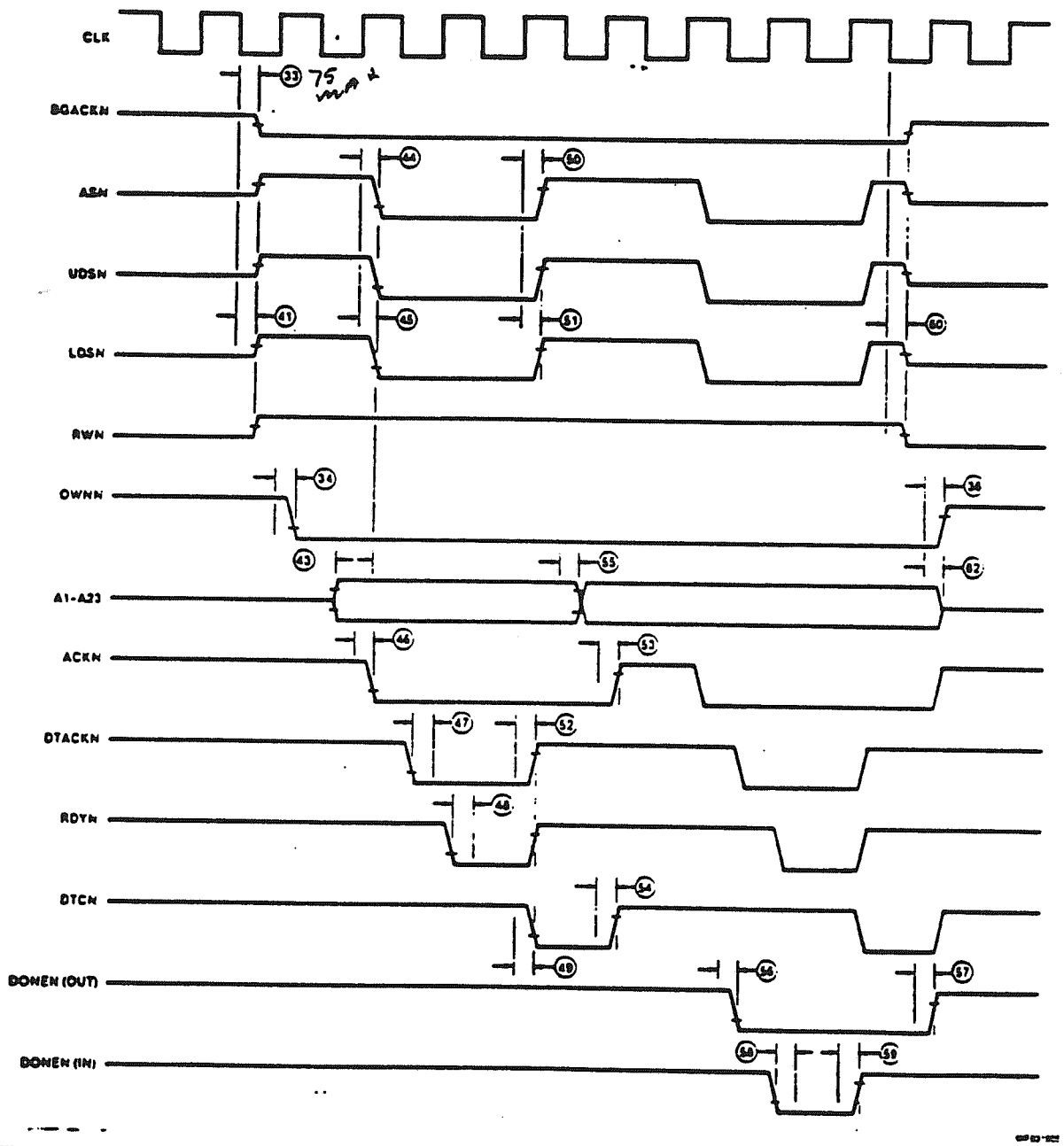
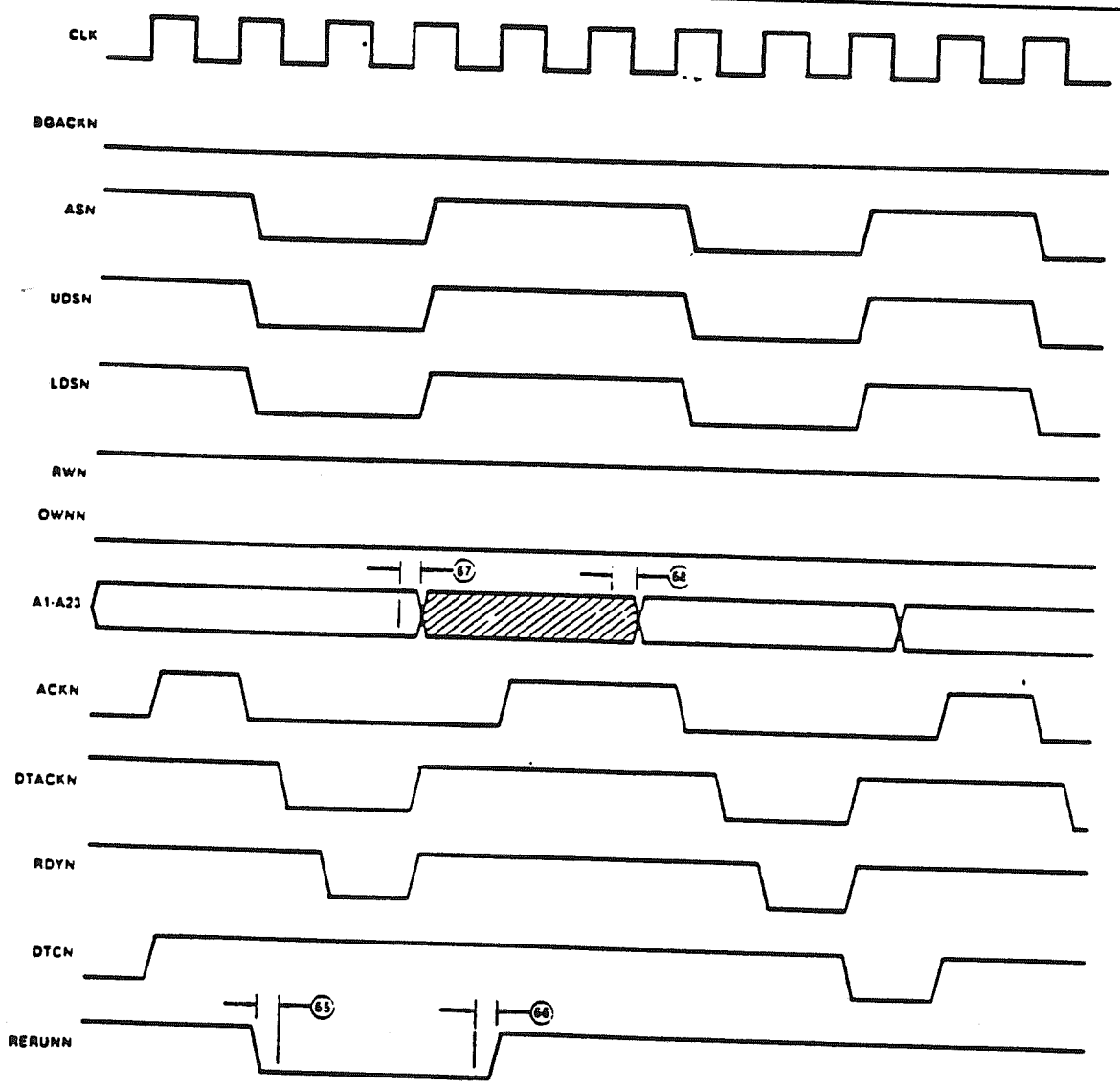


Figure 10. DMAI Cycle Steal Mode Request Timing



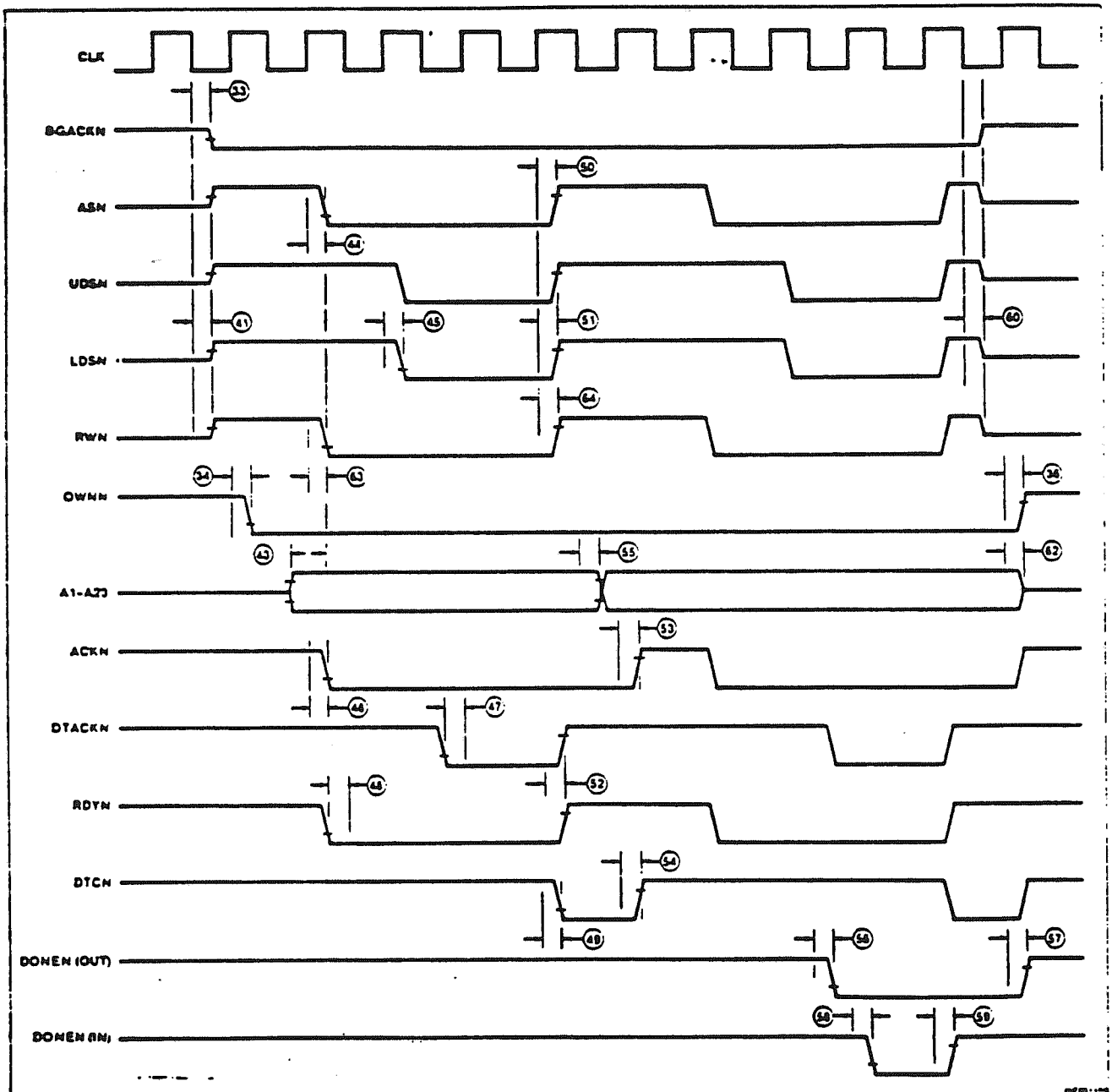
NOTE:
1. 16-bit transfer illustrated. For 8-bit transfer either LDSN or UDSN, but not both, will be asserted each cycle, depending on byte address.

Figure 11. Read from Memory, Write to Device



NOTES.
1. 16-bit transfer illustrated. For 8-bit transfer either LDSN or UDSN, but not both, will be asserted each cycle, depending on byte address.
2. DMAI will release the bus after a RERUNN if there is no valid request. The next request will then retry the cycle which was terminated by the RERUNN signal.
3. RERUNN must be asserted no later than DTACKN and RDYN.
4. If a cycle is terminated by RERUNN, the transfer count will be one less than the actual data transferred correctly. The double RERUNN signal on the same cycle will terminate the DMAI operation with a status bit set and an interrupt generated (if enabled).

Figure 13. Rerun Asserted During Read from Memory, Write to Device



NOTE:
1. 16-bit transfer illustrated. For 8-bit transfer either LDSN or UDSN, but not both, will be asserted each cycle, depending on byte address.

Figure 12. Write to Memory, Read from Device

WARRANTY

KineticSystems Company, LLC warrants its standard hardware products to be free of defects in workmanship and materials for a period of one year from the date of shipment to the original end user. Software products manufactured by KineticSystems are warranted to conform to the Software Product Description (SPD) applicable at the time of purchase for a period of ninety days from the date of shipment to the original end user. Products purchased for resale by KineticSystems carry the original equipment manufacturer's warranty.

KineticSystems will, at its option, either repair or replace products that prove to be defective in materials or workmanship during the warranty period.

Transportation charges for shipping products to KineticSystems shall be prepaid by the purchaser, while charges for returning the repaired warranty product to the purchaser, if located in the United States, shall be paid by KineticSystems. Return shipment will be made by UPS, where available, unless the purchaser requests a premium method of shipment at their expense. The selected carrier shall not be construed to be the agent of KineticSystems, nor will KineticSystems assume any liability in connection with the services provided by the carrier.

The product warranty may vary outside the United States and does not include shipping, customs clearance, or any other charges. Consult your local authorized representative or reseller for more information regarding specific warranty coverage and shipping details.

PRODUCT SPECIFICATIONS AND DESCRIPTIONS IN THIS DOCUMENT SUBJECT TO CHANGE WITHOUT NOTICE.

KINETICSYSTEMS SPECIFICALLY MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR ANY OTHER WARRANTY EITHER EXPRESSED OR IMPLIED, EXCEPT AS IS EXPRESSLY SET FORTH HEREIN. PRODUCT FAILURES CREATED BY UNAUTHORIZED MODIFICATIONS, PRODUCT MISUSE, OR IMPROPER INSTALLATION ARE NOT COVERED BY THIS WARRANTY.

THE WARRANTIES PROVIIDED HEREIN ARE THE PURCHASER'S SOLE AND EXCLUSIVE REMEDIES ON ANY CLAIM OF ANY KIND FOR ANY LOSS OR DAMAGE ARISING OUT OF, CONNECTED WITH, OR RESULTING FROM THE USE, PERFORMANCE OR BREACH THEREOF, OR FROM THE DESIGN, MANUFACTURE, SALE, DELIVERY, RESALE, OR REPAIR OR USE OF ANY PRODUCTS COVERED OR FURNISHED BY KINETICSYSTEMS INCLUDING BUT NOT LIMITED TO ANY CLAIM OF NEGLIGENCE OR OTHER TORTIOUS BREACH, SHALL BE THE REPAIR OR REPLACEMENT, FOB FACTORY, AS KINETICSYSTEMS MAY ELECT, OF THE PRODUCT OR PART THEREOF GIVING RISE TO SUCH CLAIM, EXCEPT THAT KINETICSYSTEMS' LIABILITY FOR SUCH REPAIR OR REPLACEMENT SHALL IN NO EVENT EXCEED THE CONTRACT PRICE ALLOCABLE TO THE PRODUCTS OR PART THEROF WHICH GIVES RISE TO THE CLAIM. IN NO EVENT SHALL KINETICSYSTEMS BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOSS OF PROFITS.

Products will not be accepted for credit or exchange without the prior written approval of KineticSystems. If it is necessary to return a product for repair, replacement or exchange, a Return Authorization (RA) Number must first be obtained from the Repair Service Center prior to shipping the product to KineticSystems. The following steps should be taken before returning any product:

1. Contact KineticSystems and discuss the problem with a Technical Service Engineer.
2. Obtain a Return Authorization (RA) Number.
3. Initiate a purchase order for the estimated repair charge if the product is out of warranty.
4. Include a description of the problem and your technical contact person with the product.
5. Ship the product prepaid with the RA Number marked on the outside of the package to:

KineticSystems Company, LLC
Repair Service Center
900 North State Street
Lockport, IL 60441

Telephone: (815) 838-0005
Facsimile: (815) 838-4424
Email: tech-serv@kscorp.com