digital .interoffice memorandum

To: Distribution

Date: February 25, 1979
From: Dan Sullivan
Dept: L.C.E.G.
Tel: 231-6519
Loc: MR1-2/E47

Subject: MA30 Functional Specification Review

A review of the MA30 Memory Subsystem Functional Specification is tentatively scheduled to be held during the Dolphin CPU Meeting on March 8, 1979 (in the KL10 Conference Room, 3:30 - 5:00). If this date is to be changed it will be to a later one and you will be notified prior to Friday, March 2.

Attached is a copy of the MA30 specification. Rather than having an agenda of interesting items, it will be reviewed on a page by page basis.

The are some area of particular interest. The functions provided by using the IIL interface are not specified. Unless someone sees it to be a serious difficulty I will spec' this as the design progresses.

It has been suggested that when the memory becomes poisoned it should send valid data tags (even if the data is questionable) to at least disk controllers reading the memory so that they may complete their transfer and not leave a directory incomplete. The MA30 is not spec'd to do this because I believe that this problem has to be solved in the disk controller anyhow. This problem exists anytime a disk controller accesses a location with an uncorrectable error or receives data with an uncorrectable bus error. The only way to solve this is to have the controller pass through bad data and I believe this is already done. This should be discussed as if my argument is wrong I can add functionality that would allow selected ID's to unconditionally receive valid data tags (however I don't think that this will solve the bus error problem).

The MA30 does not have a timeout spec'd for the READ LOCK - WRITE UNLOCK sequence. I have done this because there is a good chance that, because of MCA pin limitations, I will not be able to implement it anyway. It also it has been proposed that this timeout would reset the lock. I do not believe the MA30 has the intelligence to know when to unlock a data base.

The best this could cause is the loss of information necessary to analyze the error and at worst it could corrupt a data base.

The MA3Ø provides flags so that a processor or the console can tell if a memory is active and if it is processing WRITE UNLOCKS. This, I believe, is not only sufficient but a better way to solve the problem of missing WRITE UNLOCKS. Note that if a timer reset the lock unconditionally, that everytime software got a lock it would have to check to see if a timeout previously occurred (if it could figure out which MA3Ø the lock was in). If interrupts generally touch data bases then their completion time would be affected.

MA3Ø MEMORY SUBSYSTEM
FUNCTIONAL SPECIFICATION

REV Ø FEBRUARY 25, 1979

Maintained By: Dan Sullivan
Dept: L.C.E.G.
DTN: 231-6519
Loc: MR1-2/E47

1.Ø NOMENCLATURE

The following nomenclature is used to refer to Dolphin Bus transactions and parts of the system. In addition familiarity with the Dolphin Bus Specification is necessary to understand this specification.

accept -

A memory request is accepted if a REQUEST ACCEPTED confirmation code has been sent for the first bus word of the memory request.

Bus-

Bus always refers to the Dolphin Bus

bus device -

A bus device is anything that interfaces to the Dolphin Bus in accordance with the Dolphin Bus Specification

bus word -

A bus word is all the bits transfered from one bus device to another in one clock period. The interpretations of the data lines is dependent on the tag (as defined in the Dolphin Bus Specification) associated with it. Figures 1 and 2 show the interpretations of the data lines when a Command Tag is used. Figure 3 shows the interpretations of the data lines when a Data Send Tag is used.

TEN BIT NUMBERING

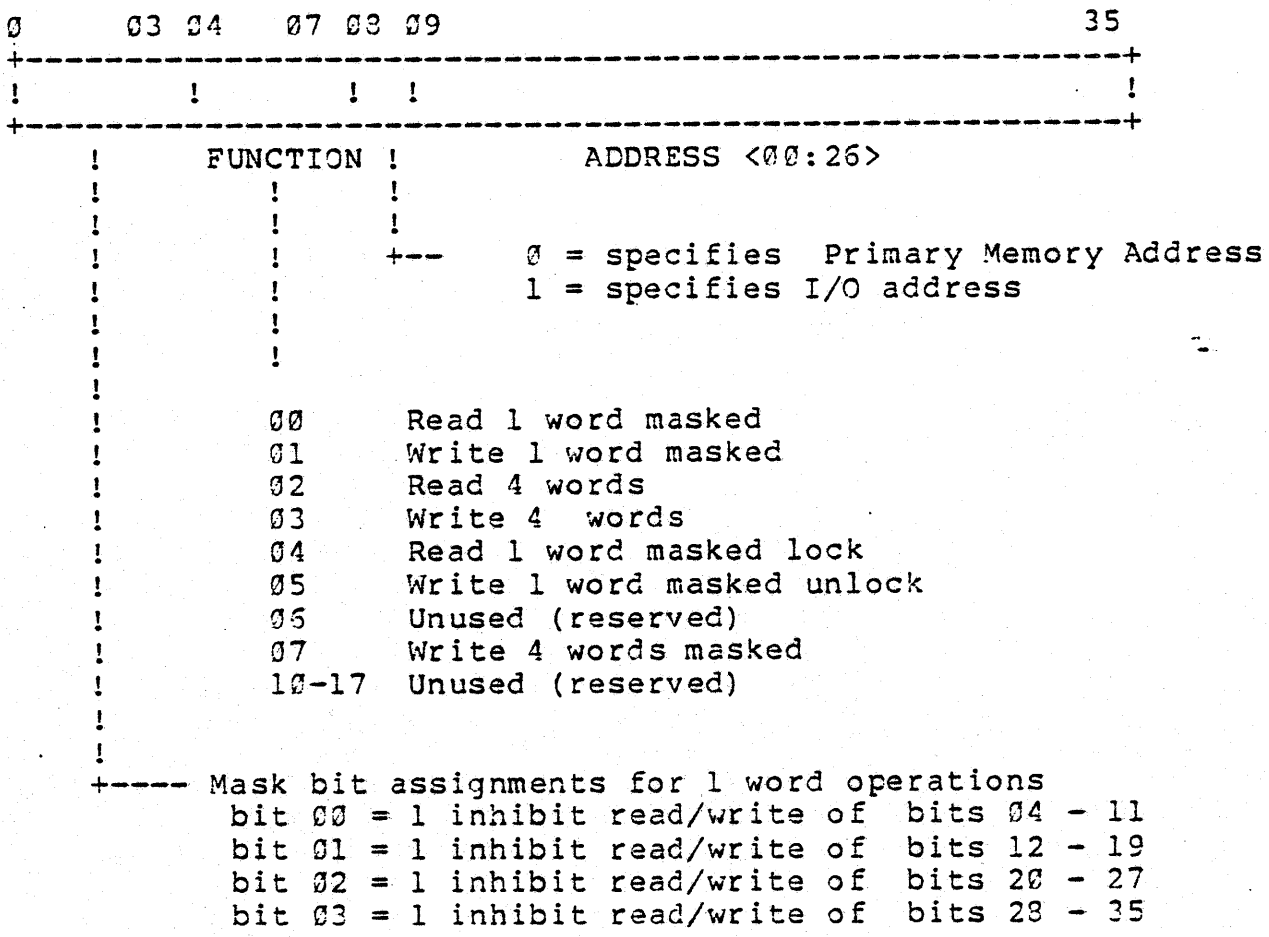


Figure 1
Data Line Interpretations for Command Tag (Dolphin)

VAX BIT NUMBERING

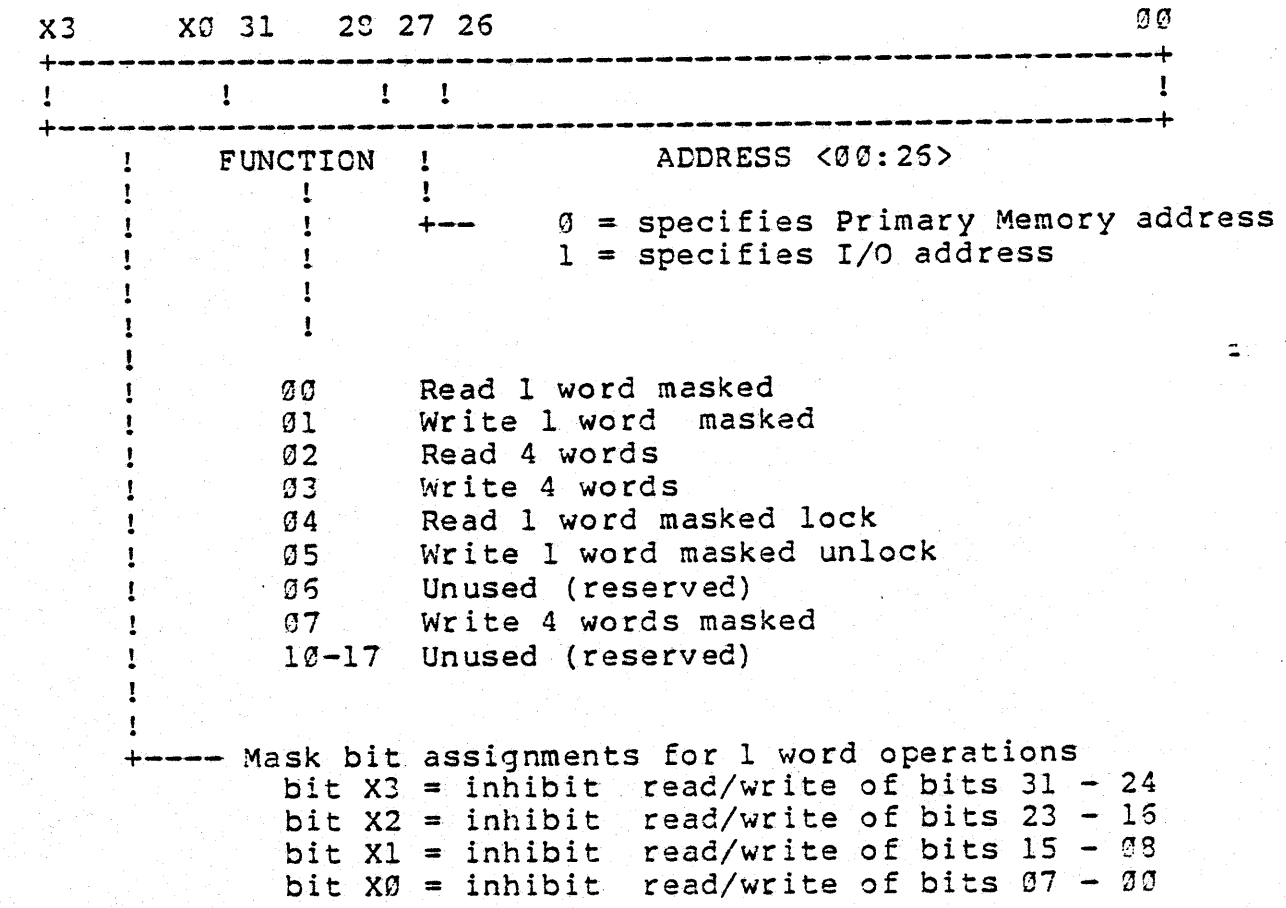
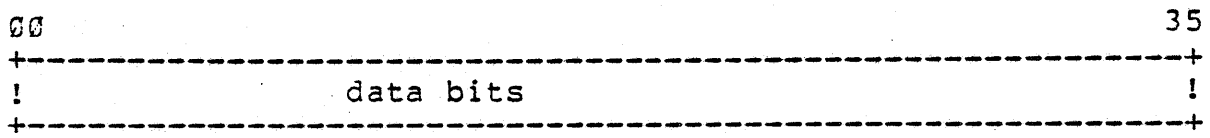
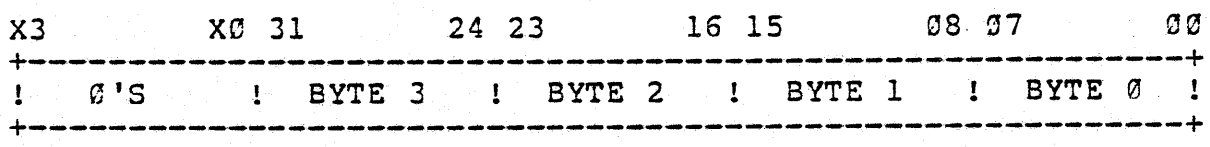


Figure 2
Data Line Interpretations for Command Tag (VAX)

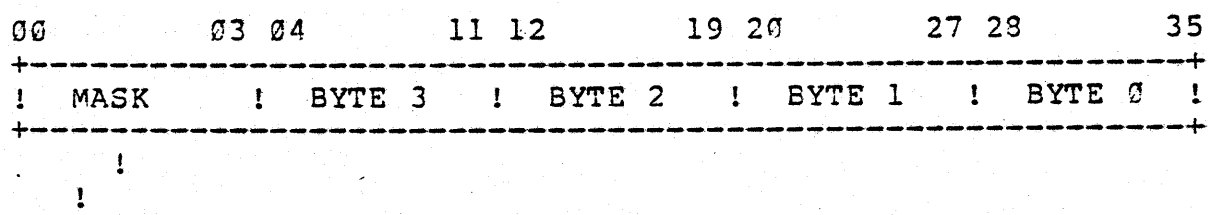
TEN BIT NUMBERING - except WRITE 4 WORDS MASKED



VAX BIT NUMBERING - except WRITE 4 WORDS MASKED

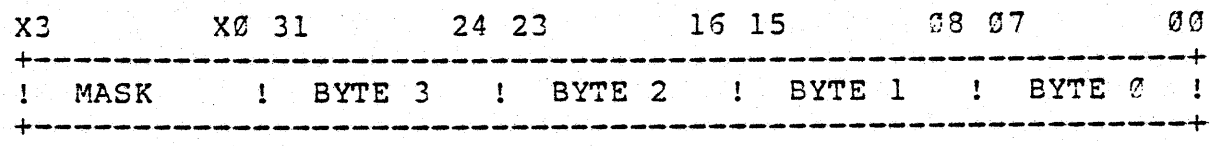


TEN BIT NUMBERING - for WRITE 4 WORDS MASKED



Mask bits for write
- bit ØØ = Ø write bits Ø4-11
- bit Ø1 = Ø write bits 12-19
- bit Ø2 = Ø write bits 2Ø-27
- bit Ø3 = Ø write bits 28-35

VAX BIT NUMBERING - for WRITE 4 WORDS MASKED



Mask bits for write
- bit X3 = Ø write bits 31 - 24
- bit X2 = Ø write bits 23 - 16
- bit X1 = Ø write bits 15 - Ø8
- bit XØ = Ø write bits Ø7 - ØØ

Figure 3
Interpretations of Bus Lines for DATA SEND Tag

confirm -
To confirm a bus word is to send a confirmation other than NULL STATE for that word. To confirm a memory request is to confirm the first word of that memory request.

field -

A field is a collection of bits numbered 0 through N, where there are N+1 bits in the field. The number contained in a field is to be interpreted as a binary number whose low-order bit is the high-order bit of the field. A flag is a one bit field, that is the flag FLAG is the same as the field FLAG <0:0>.

bus message -
A bus message is collection of contiguous bus words that are sent from one device to another.

memory operation -
A memory operation is memory request, all the actions taken by the memory as a result of that memory request, and a memory response if necessary.

memory request -
A memory request is a bus message sent from a device to the MA30. A command tag is always put in the first bus word of a memory request. The type of memory request is the type of function specified in the first bus word.

memory response -
A memory response is a bus message sent from the MA30 to a device as a result of a memory request from that device. For example when a read is done a memory response returns the data.

source-
A source is a device that sends a memory request on the Bus

2.0 OVERALL DESCRIPTION

The MA30 is a memory subsystem that interfaces to the Dolphin Bus. It can provide up to 4 Mwords of memory. The words may be interpreted as being 32 or 36 bits wide. Functionality for writting any combination of eight-bit bytes in a 32-bit word is provided. Data in the memory is protected by a single error correcting and double error detecting code.

In the implementation of the MA30 there will at times be some design tradeoffs which will favor one performance factor and degrade another. When this happens the tradeoff will be chosen that optimizes the following performance factors:

1. Four word read access time, most important,
2. Four word masked write throughput,

3. One word masked write throughput,
4. Dolphin performance.

Because so much of the hardware in the MA3Ø will be configured by software, even though it will be possible to configure it to run on Dolphin or VAX, it may also be possible to configure it in such a way that performance is better optimized for one of the systems to the loss of the other or even be incompatible with the other.

3.Ø FUNCTIONALITY

The MA3Ø performs all memory reference types of commands mentioned in the Dolphin Bus Specification. It is always a server on the Bus, that is it sends only memory responses on the Bus. It protects the integrity of data, address, and some control in itself and logs information on errors it detects in them.

3.1 Memory Commands

The MA3Ø will not accept memory requests except when the following function codes are in the first bus word of the memory requests: Read Masked, Write Masked, Read 4 Words, Write Masked 4 Words, Write 4 Words, Read Masked Lock, and Write Masked Unlock. The details of the processing of these memory requests are in the Memory Operation section of this specification.

3.2 Data Integrity

The MA3Ø never regenerates check bits for data and therefore not only are the arrays protected by a single error correcting, double error detecting code, but also are the data paths in the memory.

3.3 Address Integrity

Most of the address path in the MA3Ø is protected by parity. The parity is never regenerated from the address but is derived from the the check bits received from the Bus.

3.4 Error Handling

Bus errors will be handled by the Bus chip set. An error detected within the MA30 will cause appropriate information to be loaded into CSR('s) and flags to be set. The information and flag(s) are retained until explicitly cleared or, in some cases, until another error occurs.

Errors within the MA30 are divided into two classes, correctable and uncorrectable. Correctable errors are those errors detected by the MA30 and then corrected so that the memory operation is completed with the correct result. Uncorrectable errors are those errors which the MA30 detects but that results in bus operation being completed with an incorrect result.

In the case of a correctable error occurring while information is being retained on a previous error, the information on the uncorrectable error is lost and the CORRER LOST flag is set to show that this has occurred at least once.

When an uncorrectable error occurs and information is being retained on a previous error that was correctable then the information on the correctable error is lost and the information on the correctable error is retained. The CORRER LOST flag is set to show that this has happened. Note that in this case it is possible that erroneous information is collected on the correctable error.

When an uncorrectable error occurs and information is being retained on an uncorrectable error then the information on the second uncorrectable error is lost and the UNCERR LOST is set to show that this occurred at least once.

Whenever an error is detected could have poisoned memory (changed it in some unknown location) the POISONED MEM flag is set. As long as this flag is set all of the memory responses sent as a result of a memory request that specified Primary Memory Address will have an invalid data tag.

The details of how to clear error flags is contained in the Control and Status Registers of this specification.

4.0 MEMORY ORGANIZATION

An MA30 Memory Subsystem may contain up to 4 Mwords which is 22 bits of address. All words in memory have 36 bits and are interpreted by the source of a memory request as having a 32 or 36 bit format.

In addition, the MA3Ø contains a number of control and status registers (CSR's). CSR's are located in the I/O address space. These CSR's contain fields of bits. The allocation of fields to CSR's is given in the Control and Status Registers section of this specification.

The MA3Ø has a queue that can hold up to two memory requests.

The MA3Ø will terminate the Dolphin Bus.

4.1 Physical Addressing Of Memory

The MA3Ø will confirm a memory request when the first bus word of it has no uncorrectable errors and specifies a Primary Memory Address and the number contained in the ADDRESS field (ADDRESS <ØØ:26>) of the Command Tags contained within the range of Primary Memory Addresses for which the memory is configured. The MA3Ø can be configured by software to have to up to 4M contiguous Primary Memory Addresses in increments of 256K.

The MEM SIZE field (MEM SIZE <Ø:8>) is used to configure the size of the MA3Ø. The number contained there plus one is the number of 256K increments of Primary Memory Addresses which the MA3Ø will contain. MEM SIZE <Ø:3> can be read or written. On an MA3Ø MEM SIZE<4:8> are read only and are always Ø.

The MEM FIRST field (MEM FIRST <Ø:3>) is used to configure the first Primary Memory Address which the MA3Ø contains. The number contained in this field times 2**22 is the smallest Primary Memory Address that the MA3Ø will contain. MEM FIRST<Ø:1> can be read or written. On a MA3Ø MEM START <2:3> are read only and are always Ø.

4.2 Physical Location Of Memory

The physical memory address of a location refers to the specific piece(s) of hardware that implement that location. This linking is permanent, a given physical memory address always refers to the same piece(s) of hardware. The linking can be changed only by changing hardware. The Primary Memory Address, which is in the ADDRESS field of a bus word that has a command tag, is not a physical address because the actual piece(s) of hardware to which it refers can be changed by software.

Physical Address Space of the MA3Ø is divided into 16 blocks numbered Ø to 15. Each block contains 256K locations. The Primary Memory Address is mapped onto a

physical memory address. The number contained in ADDRESS <00:17> is the address of the location within a block. The block number is derived from ADDRESS <7:10> and the BLOCK SWAP field (BLOCK SWAP <0:1>) which is located in a CSR. BLOCK SWAP <0:1> is XOR'd with the two low-order bits of the number contained in ADDRESS <7:10> to produce the block number that the Primary Memory Address is mapped onto. Figure 5 shows this mapping.

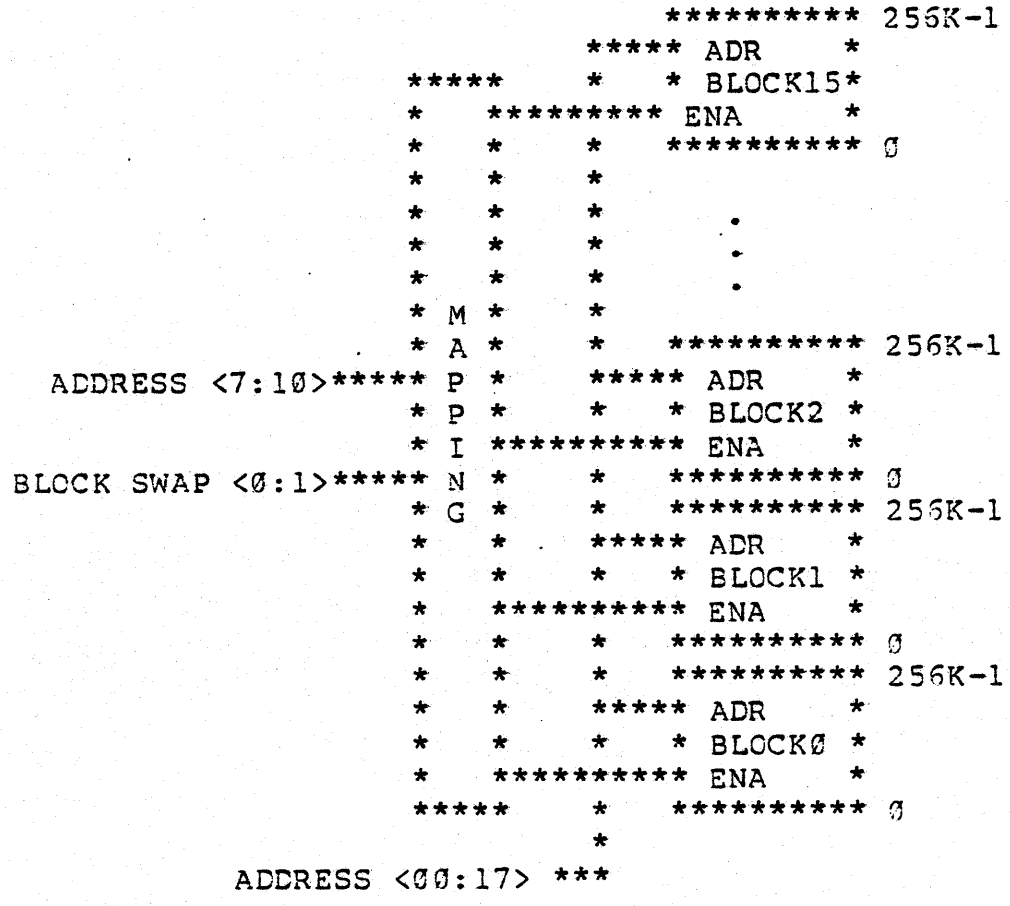


FIGURE 5
MAPPING OF PRIMARY MEMORY ADDRESS ONTO PHYSICAL MEMORY

Various pieces of software and require that certain sections of the first 256K of Primary Memory Address space have no uncorrectable errors. This feature allows up to four different sets of memory IC's to be chosen among to implement the first 256K of the Primary Memory Address space.

4.3 Control And Status Registers

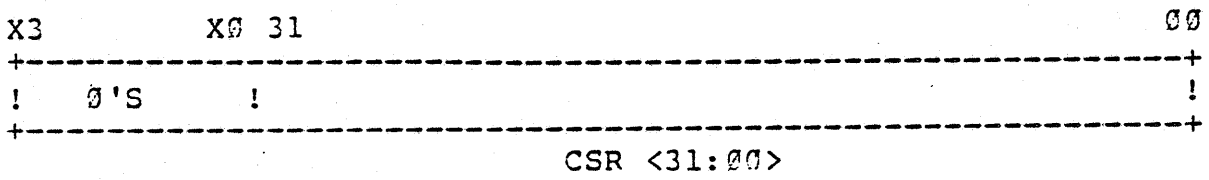
The MA3Ø contains a number of Control and Status Registers (CSR's) which are used to initialize, modify, and observe it. The CSR's are accessible in the I/O address space.

Each CSR is given an I/O address. The MA3Ø will confirm a memory request that has no detected uncorrectable errors and specifies an I/O Address in its first bus word, if the number contained in ADDRESS <ØØ:26> is the address of a CSR in the MA3Ø. The Bus Chip Set makes this determination. Note that per the Dolphin Bus Specification all command functions that refer to I/O addresses are interpreted as one word operations.

All CSR's contain 36 bits. Figure 6 shows the format of a CSR. Each CSR contains a number of fields. When not all bits of the CSR are used by fields then they are read-only and always contain a Ø. The allocation of fields to CSR's is given in Table 1. This allocation is subject to change both in the position of a field in a CSR and the CSR in which the field is located.

All CSR's can be read and written unless specified otherwise. CSR3 contains error flags. These flags can be cleared by writting zeros into them, however, if a flag was set by an error between the time CSR3 is read and written that error will go undetected. CSR4 contains the CLR ERRORS flag. If set this flag will clear all the error flags in CSR3 that were set the last time CSR3 was read. Note that reading CSR3 when this flag is set will cause CSR3 to be cleared.

VAX BIT NUMBERING



TEN BIT NUMBERING

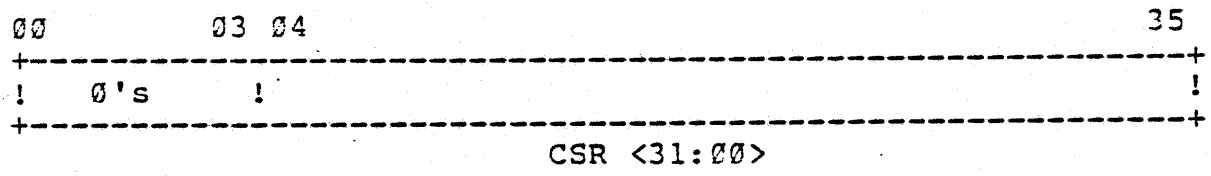


Figure 6
Format of a CSR

```
CSRØ <Ø:Ø> = OPR MODE<Ø:Ø>
CSRØ <1:2> = BLOCK SWAP <Ø:1>
CSRØ <3:6> = MEM FIRST <Ø:3>
CSRØ <7:14> = MEM SIZE <Ø:8>
CSRØ <15> = ON LINE
CSRØ <16> = IGNR CORR
CSRØ <17:19> = REFR PER <Ø:2>
CSRØ <17:2Ø> = Ø

CSR1 <ØØ:23> = ERR LOC <ØØ:23>
CSR1 <24:35> = Ø

CSR2 <Ø:6> = ERR SYN <Ø:6>
CSR2 <7:35> = Ø

CSR3 <Ø> = CTL ERR
CSR3 <1> = ADDR ERR
CSR3 <2> = DATA ERR
CSR3 <3> = POISONED MEM
CSR3 <4> = UNCERR LOST
CSR3 <5> = CORERR LOST
CSR3 <6:35> = Ø

CSR4 <Ø> = CLR ERRORS
CSR4 <1> = HIGH P ONLY
CSR4 <2> = ILLEGAL BUS WORD
CSR4 <3> = MEM LOCKED
CSR4 <4:8> = LOCK OWNER
CSR4 <9> = WRITE LOCK RCVD
CSR4 <1Ø:13> = ID SEL <Ø:3>
CSR4 <14> = MEM ACTIVE
CSR4 <15> = CLR EXTRA BIT
CSR4 <16:19> = EXTRA BIT ADDR <18:21>
CSR4 <2Ø> = EXTRA BIT ADDR <Ø>
CSR4 <21:35> = Ø

CSRn <Ø> = EXTRA BIT,  n = 5 mod 16
CSRn <1:35> = Ø,   n = 5 mod 16

CSR7 <Ø:7> = ID INFO Ø <Ø:7>
CSR7 <7:35> = Ø

CSR8 <Ø:7> = ID INFO 1 <Ø:7>
CSR8 <8:35> = Ø
       .
       .
       .
CSR2Ø <Ø:7> = ID INFO 13 <Ø:7>
CSR2Ø <8:35> = Ø
CSR22 <Ø:7> = ID INFO 14 <Ø:7>
CSR22 <8:35> = Ø
CSR23 <Ø:7> = ID INFO 15 <Ø:7>
CSR23 <3:35> = Ø
```

Table 1
Allocation of Fields and Flags to CSR's

4.4 Extra Bit

Every location in memory has associated with it an extra bit is not covered by ECC. This bit may be read or written and can be used to cause various events to happen when the location it is associated with is accessed.

The extra bits are accessed by reading or writting a CSRn <Ø>, where n=5 mod 16. The physical address of the location associated with the extra bit accessed in CSRn <Ø> is

(n - 7 + EXTRA BIT ADDR <18:21> + (EXTRA BIT ADDR <Ø> * (2**21)))
XOR ((2**18) * BLOCK SWAP <Ø:1>).

All numbers in the above expression should be interpreted as binary so that the XOR function makes sense. EXTRA BIT ADDR <18:21> and EXTRA BIT ADDR <Ø> are located in CSR's. EXTRA BIT ADDR <1:17> and EXTRA BIT ADDR <22:35> do not exist. Figure 9 shows an alternate way to show the physical address by concatenating fields.

PHYSICAL ADDRESS OF LOCATION ASSOCIATED
WITH EXTRA BIT IN CSRn <Ø> =

EXTRA BIT ADDR	(17 BIT BINARY) (REPRESENTATION) (of n - 7)	EXTRA BIT ADDR <18:21>

XOR

BLOCK SWAP <Ø:1> ! (ØØØØØØØØØØØØØØØØ)

Figure 9
Extra Bit Addressing

4.5 Array ID

There are up to 16 hex boards of hardware that implement the memory arrays of the MA30. There are 128 bits of information used to identify uniquely each board manufactured. The definition of these bits has not yet been determined. These bits are contained in 16 fields (ID INFO 2 <0:7>, ID INFO 1 <0:7>, through ID INFO 17 <0:7>) which are located in CSR's. The ID SEL field (ID SEL <0:3>) determine which boards' information is in the ID INFO fields. The ID SEL field is in a CSR. The relationship between the number contained in the ID SEL field and backplane slot will be given at a later date. The ID INFO fields are read only. Writting them will not change their contents.

5.0 MEMORY OPERATION

The MA30 is always a server, that is it only sends memory responses on the Bus. It will send one memory response for every accepted memory request that requires one. If a memory request is accepted an error detected in the MA30 will not prevent a memory response from being sent. Whenever the MA30 does not accept a memory request (e.g. error, busy) it will never send a memory response to that memory request. The memory requests which require responses are enumerated later in this specification.

The MA30 requires a backplane signal to tell it if it running on battery backed power or not. This signal is not yet specified but will be included in the Dolphin Bus Specification. The MA30 will this signal to determine whether it should ignore Bus signals or not.

The ON LINE flag, which is in a CSR, is used to enable the Primary Memory Address space of the MA30. When this flag is clear the MA30 will confirm all memory requests with Primary Memory Addresses with BUSY. When this flag is set the MA30 will confirm as described elsewhere in this specification.

The MEM ACTIVE flag is set every time the MA30 begins to process a memory request. This flag is located in a CSR.

The MA30 checks every bus word for errors. If it detects an uncorrectable error it interprets the ID to not be its own and the Tag to not be a Command. It will confirm the bus word with error in accordance with the Dolphin Bus Specification.

If no uncorrectable errors are found then it decodes the bus word to see if it should otherwise confirm it (note: in the context of this specification sending NULL STATE on

the confirmation lines is the same as not confirming a bus word). In this case if the bus word contains a COMMAND Tag it is considered to be the beginning of a memory request and terminates a previous memory request if one is in progress. The Physical Addressing Of Memory, and Control And Status sections of this specification explain when memory requests are confirmed.

If the MA30 confirms a memory request it will do so with ERROR if the function specified is not Read Masked, Write Masked, Read 4 Words, Write 4 Words, Write Masked 4 Words, Read Masked Lock, or Write Masked Unlock. If one of the above functions is specified it will confirm with BUSY if its queue is full and, if ID is that of a high priority device, sets the HIGH P ONLY flag which is in a CSR. If the queue is not full and the HIGH P ONLY flag is set and the ID is that of a high priority device it confirms with REQUEST ACCEPTED and clears the HIGH P ONLY flag. If the HIGH P ONLY flag is set and the ID is not that of a high priority device it confirms with BUSY. If the queue is not full and the HIGH P ONLY flag is clear it confirms with REQUESTED ACCEPTED.

A bus word that has no uncorrectable errors and has the MA30's ID and does not have a Data Send or Invalid Data Tag will be confirmed with ERROR. This will also cause the ILLEGAL BUS WORD flag, which is in a CSR, to be set.

A bus word that has the MA30's ID and has a Data Send or Invalid Data Tag will be confirmed either with ERROR and the ILLEGAL BUS WORD flag set, if it is not part of a bus message, or DATA ACCEPTED if it is.

Whenever the ILLEGAL BUS WORD flag is set the Bus Fault line is asserted in accordance with the Dolphin Bus Specification.

Once the MA30 accepts a memory request it queues it, and processes it immediately if it can. If it cannot it queues it to be processed when it is able. Memory requests are always processed in the order they are accepted.

There are two classes of memory requests, reads and writes. A read is a memory request that requires a memory response and a write is one that does not.

Memory requests may be processed with or without detecting an error. Processing without detecting an error is normal processing, if an error is detected it is erroneous processing. In normal processing some bits in the MA30 (CSR's or arrays) are read or written. In erroneous processing a particular flag is set and some information may be logged. Other actions taken will depend on the type of error and the state of the MA30 when it occurred.

The processing that occurs because of an accepted memory request depends on the function specified by it (the function is specified by the first bus word of the memory request) and the operating mode of the memory. The operating mode is specified by the operating mode field (OPR MCDE <Ø:Ø>) which is located in a CSR. Normal operating mode is indicated by OPR MODE <Ø:Ø>=Ø. Maintenance operating mode is indicated by OPR MODE <Ø:Ø>=1.

Each block in the MA3Ø is part of a group. Blocks Ø and 1 form Group Ø, Blocks 2 and 3 form Group 1, and so on. Each group has a data register associated with it called a group buffer. The group buffer associated with Group N is called Group Buffer N. Figure 7 shows the groups in the MA3Ø. A group buffer can hold four 36-bit words and it is not used in the normal operating mode of the MA3Ø. In maintenance operating mode all writes, in addition to what they do in normal operating mode, also write the group buffer associated with the group the location being written is in. All reads read the group buffer associated with the group the location being read is in.

This feature is provided so that the MA3Ø can provide some valid data when the Bus is single ticked, as in this condition the RAM's have lost their contents due to lack of refresh.

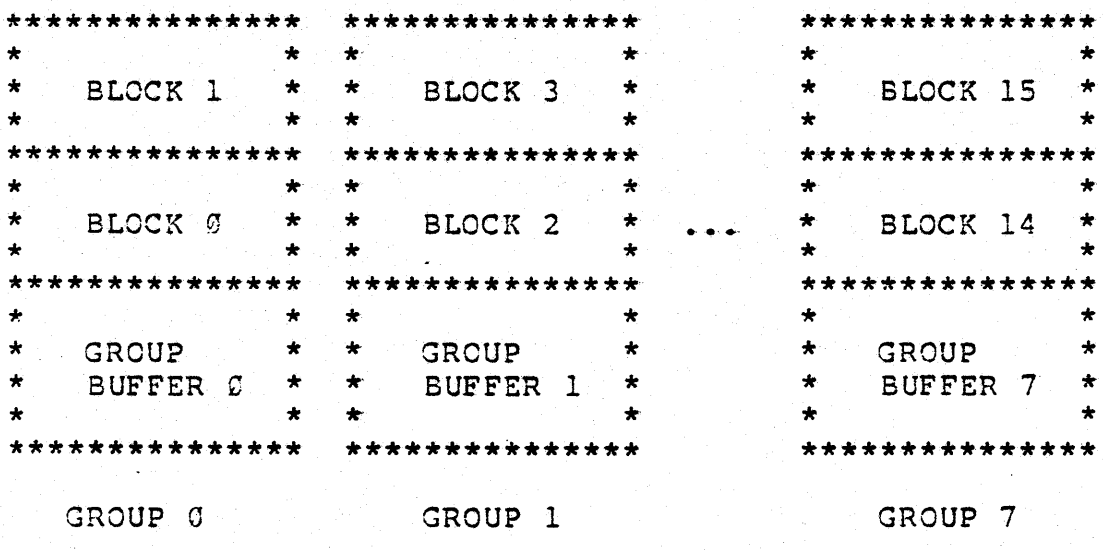


FIGURE 7
Group Buffers in MA3Ø

5.1 Description Of Processing Of Memory Requests

The MA30 processes one memory request at a time. In addition it must process refresh requests that are generated within the MA30 itself. There is no overlapping of processing refresh requests and memory requests. Whenever there is a queued memory request ready for processing and a refresh request pending, the refresh request is processed first.

Because of the technology used by the memory it must be periodically refreshed. The MA30 generated a refresh request once every refresh period. The refresh period is determined by the refresh period field (REFR PER <0:2>), which is located in a CSR. For a system clock with a 33ns period, the refresh period is equal to 8.45usec plus, 2.11usec times the number contained in REFR PER <0:2>. If the clock period for the system is different, whether by intent, drift, or whatever, the refresh period will be affected proportionally.

5.1.1 Errors While Processing Memory Requests -

During the processing of a memory request there are a number of types of errors that may be detected. They are addressing errors, control errors, and data errors. When they are detected they cause a flag to be set, an addressing error the ADDR ERR flag, a control error the CTL ERR flag, and a data error the DATA ERR flag. These flags are located in CSR's.

Addressing errors are always uncorrectable errors (see the Error Handling section of this specification for the implications of this). In addition to setting the ADDR ERR flag, ADDRESS <00:21> is retained in ERR LOC <00:21> and BLOCK SWAP <0:1> is saved in ERR LOC <22:23>. The Error Location field (ERR LOC <00:23>) is in a CSR.

Control errors are always uncorrectable errors. As with addressing errors, they also retain ADDRESS <00:21> and BLOCK SWAP <0:1>.

Data errors may be correctable or uncorrectable. If the data error is correctable and the extra bit associated with the location in error is set, and the IGNR CORR flag (which is in a CSR) is set then the error is ignored, that is the MA30 behaves as though the error was never detected. Uncorrectable and correctable errors that are not ignored set the DATA ERR flag and they retain ADDRESS <00:21> and BLOCK SWAP<0:1> in ERR LOC <00:23>. In addition they retain the syndrome of the error in the Error Syndrome field (ERR SYN <0:6>).

5.1.2 Read Memory Requests -

A read memory request requires a memory response. Read Masked, Read 4 Words, and Read Masked Lock are read memory requests. The bus words in the memory response may have NULL, RETURN DATA, or DATA INVALID tags. If four data words must be returned they may not be in contiguous bus words, however, the MA30 will retain ownership of the Bus between returned data words and transmit NULL tags. If no errors were detected in the data retrieved from memory then 4 word reads will transmit their data in 4 contiguous bus words. In any case the last data word will be transmitted in no more than 15 bus ticks after the first.

READ MASKED LOCK memory requests, in addition to returning data, set the MEM LOCKED flag, which is in a CSR. Also the ID of source of the memory request is saved in the LOCK OWNER field, which is in a CSR. The MA30 will always confirm BUSY to a READ MASKED memory request if the MEM LOCKED flag is set.

5.1.3 Write Memory Requests -

A write memory request does not require a memory response. Write Masked, Write Masked 4 Words, Write 4 Words, and Write Masked Unlock are write memory requests.

Before a write masked unlock memory request actually writes a location it compares the ID of the source of the memory request with the LOCK OWNER field. If they are not equal the LOCK ERR flag ,which is in a CSR, is set, the location is not written, and the Bus Fault line is asserted. If they are equal and the LOCK ERR flag is not set the location is written and the MEM LOCKED flag is cleared and the WRITE LOCK RCVD flag, which is in a CSR, is set. If they are equal and the LOCK ERR flag is set then the location is written and the MEM LOCKED flag cleared.

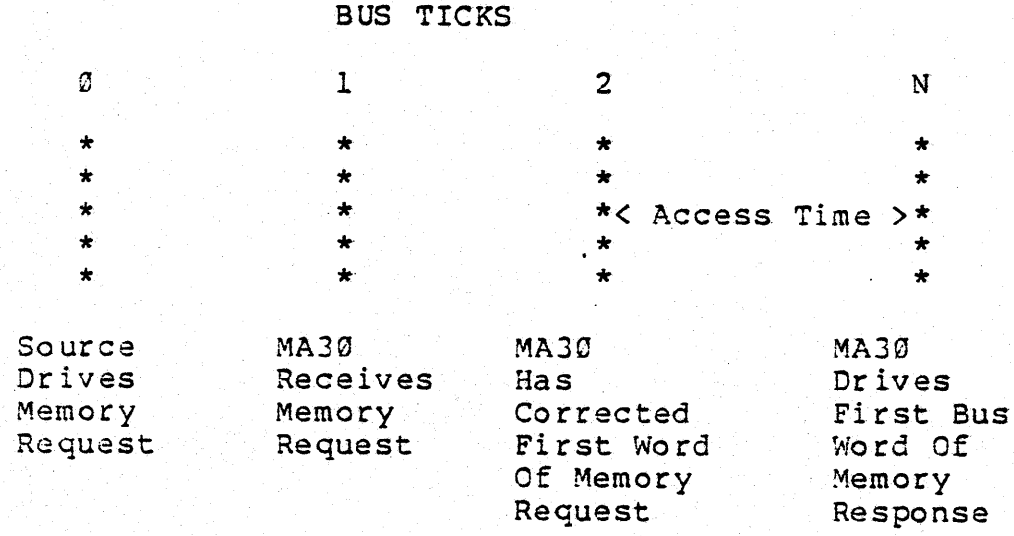
6.0 IIL BUS INTERFACE

The MA30 will interface to the IIL bus. Besides using it to configure and control the Bus Chip Set it will also be used to configure and control the rest of the MA30. Exactly what functions it provides will be determined as the design progresses.

7.0 PERFORMANCE

Two types of performance measures are specified, access time and throughput. Both assume the Bus is available to the MA30 whenever it requires it.

Access time is defined as the maximum time from when the MA30 has the first corrected bus word of a memory request until it drives the first bus word of the memory response on the Bus. It assumes that none of the data read from memory had detected errors. Access time is meaningless for a write type of memory request. Figure 8 shows the period measured to determine access time.



Access Time = N - 2 Bus Ticks

Figure 7
Access Time

The throughput of a type of memory request is the maximum rate at which it can be repeated. This performance measure also assumes that no errors are detected during a memory operation.

Table 2 shows the performance for the MA30.

Memory Operation	Access Time @33ns clock	Throughput @33ns clock
Read 1 Word Masked	429ns	2.5MHz
Read 4 Words	429ns	2.5MHz
Write 1 Word Masked	-	1.6MHz
Write 4 Words Masked	-	1.2MHz
Write 4 Words	-	2.5MHz
Write 4 words Masked and All Mask Bits Clear	-	2.5MHz

Table 2
MA30 Performance

3.0 POWER REQUIREMENTS

For MA30 power requirements see the memo "MA30 Memory Power Requirements" by D. J. Chin, 19 January 1979. The pertinent information in this memo will be included in this specification at a later date.