(10)

PRELIMINARY

DOLPHIN MASSBUS ADAPTER FUNCTIONAL SPECIFICATION

30 OCTOBER 1978

MIKE AUGERI

REVISION STATUS

REVISION 0 - 30 OCTOBER 1978 - PRELIMINARY

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 General Description

The RH30 Massbus Adapter (MBA) provides an interface between the Dolphin Bus and the DEC Standard Massbus. The MBA will be functionally compatible with TOPS10, TOPS20 and VAX/VMS operating systems. The existing versions of these operating systems will require modifications to support features which are not common to the various operating systems.

Up to eight devices (disks, tapes) may be attached to the Massbus, but only one device can be transferring data at any one time. From a programming viewpoint, the MBA will provide access to two (2) sets of registers, internal and external. Internal registers are located in the MBA itself. External registers are located in the devices and the CPU has read or write access to them via the asynchronous control bus portion of the Massbus. The CPU can read or write these registers for purposes of determining interface status or initiating data transfers between the device and system memory. The registers can be accessed while a data transfer is in progress.

As part of the internal register structure, the MBA will provide primary and secondary device command registers and a channel command buffer. The device command registers will contain the Massbus function code and desired sector/surface (track) or frame count. Having a secondary device command register eliminates the necessity of the program having to respond to device interrupts within the sector gap, essentially providing a full sector time to respond to interrupts for the next device command. The channel command buffer will contain the channel opcode, word count and data buffer address. By storing this information internally, the interface will permit changing data buffer locations on non-sector boundaries, which is required to maintain compatibility with TOPS10.

The MBA will provide 8 words (36 or 32 bit words) of data buffering for purposes of averaging the Dolphin Bus transfer rate, including bus latency, to the device transfer rate. Memory data will be transferred to or from the device via the synchronous data bus portion of the Massbus. Read reverse will be implemented to facilitate efficient handling of tapes.

For purposes of clarification, units on the Dolphin Bus will be referred to as controllers and units on the Massbus will be referred to as devices (disks, tapes, etc.). The function of the controller is to provide an interface between the CPU and/or memory and the device. Furthermore, the term channel is used loosely and really means channel functionality, since the channel is integrated into the controller and does not exist as a seperate entity. See Figure 1.1.1 Basic System Block Diagram and Figure 1.1.2 Simplified Functional Block Diagram.

The option number for the Massbus Adapter is the RH30 Dolphin Massbus Adapter. The board numbers and tentative descriptions are the ?

1.2 Goals

1.2.1 Performance

The goal will be to have the MBA accommodate Massbus devices with transfer rates of 2 M word/second over the Massbus (18 or 16 bit words). This translates into 1 M word/second over the Dolphin Bus (36 or 32 bit words) or a device transfer rate of 4.5 M byte/second with 36 bit words or 4 M byte/second with 32 bit words (8 bit bytes). The commitment will be to handle the next generation of disk products which are anticipated to have transfer rates of 3.0 M byte/second or less.

1.2.2 Diagnostics

The design will permit a clear distinction between controller or device failures. This will be accomplished by providing the ability to test the MBA with no device attached to the Massbus (utilizing loop back or wraparound techniques) and by allowing program simulation of the required control signals. If the Massbus has 8 devices attached, one can be disabled and its address used. To assist in design verification the program will be able to force as many of the detectable error conditions as is practical. To enhance servicing hardware failures the design will attempt to allow isolation to the board level, however, this approach may be compromised by space limitations.

1.2.3 Design and Construction

The goal will be to implement the MBA on three (3) extended HEX (12") multilayer modules with 150 equivalent IC positions per board. The proposed three boards will be partitioned as a System Bus Interface (SBI), Internal Control Logic (ICL) and Massbus Interface (MBI). The IC technology used will be ECL Macro Cell Arrays (MCA) for the system bus interface chips and ECL and/or Schottky TTL SSI/MSI parts for the rest of the logic. Where it is practical the use of PROMs or FPLAs will be used to reduce random logic chip counts. The breadboard version of the MBA will be built using multiwire boards. The prototype will be built with etched boards.

The MBA will occupy one (1) of the eight (8) I/O controller sections of the new press-pin backplane. Attachment to the Massbus will be made via three (3) forty (40) conductor flat

cables which connect directly to the backplane pins of the slot containing the Massbus Interface board and go out to the standard Massbus transition connector. A Massbus terminator board will be designed that also connects to the backplane.

1.2.4 Cost

The cost of the MBA, including components, assembly and test, is estimated to be $2000.

1.2.5 Power Requirements

A rough estimate of the power requirements for the MBA is shown in the following table. This estimate is based on one board with 11 MCA chips and 40 ECL SSI chips and the other two boards with 150 TTL chips each. Each TTL chip is assumed to require an average of 75 mA and each ECL chip is assumed to require an average of 65 mA. The calculation of the power for the MCA is based on 4W per package or about 0.77A per package. The -2.0 volt figure is based on 470 terminators at 19 mA each (for ECL signal lines based on 3.5 term/SSI chip, 30 term/MCA chip). The fourth board is the Massbus terminator board.

V		+5	-2.0	-5.2	Total watts/board
1.	A	---	8.93	11.06	---
	W	---	17.86	57.51	75.37
2.	A	11.25	---	---	---
	W	56.25	---	---	56.25
3.	A	11.25	---	1.0	---
	W	56.25	---	5.2	61.45
4.	A	3.69	---	---	---
	W	19.37	---	---	19.37
Total per supply					
	A	26.19	8.93	12.06	---
	W	131.87	17.86	62.71	---

Total power dissipation for the MBA is 212.44 watts.

1.2.6 Environmental Requirements

Refer to DEC Standard 102.

1.3 Non Goals

No attempt will be made to permit isolation of faults to the chip level. Also, no provision for self-diagnosis is included in the design.

2.0 REFERENCES

Dolphin Bus Specification

Bus Interface Chip Set Specification

SBI Specification

VAX MBA Specification

DEC Standard 102 - Environmental Standard for Computers and Peripherals

DEC Standard 159 - Massbus Specification

RH20 Manual

RP04 Manual

RP05/RP06 Manual

RS04 Manual

TU16/TM02 Manual

3.0 System Specifications

3.1 Introduction

The signals that the MBA interfaces with are shown in figure 3.1.1 Massbus Adapter Input/Output signals. Section 3.2 briefly describes the signals on the Dolphin Bus and section 3.3 briefly describes the signals on the Massbus.

3.2 Dolphin Bus Interface

There are a total of 63 signals on the Dolphin Bus. All signals are transmitted or received using the MCA bus interface chip set. This chip set is comprised of 4 different chip types and requires 11 chips total. All signals on the bus are asserted (low/high ?).

3.2.1 Dolphin Bus Signals

SIGNAL NAME	# of SIGNALS	DIRECTION
ECC	7	S <--> C
TAG	4	S <--> C
ID	6	S <--> C
DATA	36	S <--> C
FAULT	1	S <--> C
CNF	2	S <--> C
CNF PARITY	1	S <--> C
PI SYNC	1	S ---> C
PI REQUEST	1	S <--- C
CR	1	S <--- C
CG	1	S ---> C
~~RESERVE~~	1	S <--> C
CLOCK	1	S ---> C
Total	63	

S = System
C = Controller

3.2.2 Dolphin Bus Signal Definitions

The following defines the signals on the Dolphin Bus. As the design progresses, a more elaborate description of how the MBA uses these signals will be added. For more detail see the Dolphin Bus Specification and the Bus Interface Chip Set Specification.

ECC. These seven lines contain the error correction code for the 36 DATA lines, 4 TAG lines and 6 ID lines. The

seven bit code is capable of single bit error correction and double bit error detection.

TAG. These four lines are used to indicate what type of message is being sent on the bus.

ID. These six lines are used to indicate the ID number of the transmitter or receiver of a message.

DATA. These 36 lines are used to send data, masks or commands.

FAULT. This line is used to indicate that a fault occured on the bus.

CNF. These two lines are used to confirm receipt of a message on the bus.

CNF PARITY. This line is used to transmit the parity of the CNF code. This line can also be used as a substitute line if one of the CNF, PI REQ,PI SYNC, or RESERVE lines fails.

PI SYNC. This line is used to synchronize the sending of PI requests.

PI REQ. This line is used to transmit PI request to the processor. The level of the request is determined by the time relationship between this line and the PI SYNC line.

CR. This line is used to request cycles on the bus.

CG. This line is used to grant cycles on the bus.

RESERVE. This line is used to insure proper priority use of resources when high contention occurs.

CLOCK. This line is used to transmit the system clock. The clock is currently specified at 30 Mhz (33.33ns/bus cycle).

DIAGNOSTIC BUS SIGNALS. To be supplied

CROBAR. To be supplied

3.2.3 Dolphin Bus Protocol and Timing

To be supplied

3.3 Massbus Interface

There are a total of 56 signals on the Massbus. All signals are differential signals, except for MASS FAIL. The transmitters used are DEC 75113 Dual Differential Line Drivers (Part Number

19-11341) and the receivers used are DEC 75107B Dual Line Receivers (Part Number 19-10268). The Massbus can logically be divided into two sections, the asynchronous section referred to as the control bus and the synchronous section referred to as the data bus. The control bus has a 16 bit data path with an odd parity bit and 14 control signals, for a total of 31 signals. The data bus has an 18 bit data path with an odd parity bit and 6 control signals, for a total of 25 signals. Each of these sections are described below.

3.3.1 Massbus Signals

SIGNAL NAME	# OF SIGNALS	DIRECTION
MASS DS2-DS0	3	C--->D
MASS RS4-RS0	5	C--->D
MASS CTOD	1	C--->D
MASS DEM	1	C--->D
MASS TRA	1	C<---D
MASS C15-C00	16	C<-->D
MASS CPA	1	C<-->D
MASS INIT	1	C--->D
MASS FAIL	1	C--->D
MASS ATTN	1	C<---D
MASS RUN	1	C--->D
MASS OCC	1	C<---D
MASS SCLK	1	C<---D
MASS WCLK	1	C--->D
MASS D17-D00	18	C<-->D
MASS DPA	1	C<-->D
MASS EBL	1	C<---D
MASS EXC	1	C<-->D
TOTAL	56	

C = Controller

D = Device

NOTE: All signals are differential except MASS FAIL.

3.3.2 Massbus Control Bus Signal Definition

MASS DS2-DS0. Device select lines (3) for selecting one of eight possible devices on the Massbus.

MASS RS4-RS0. Register select lines (5) for selecting one of thirty-two possible registers in each device.

MASS CTOD. This signal is asserted for a control bus write and negated for a control bus read.

MASS DEM. The DEMAND signal from the controller to the device indicating the controller wants to transfer control bus data to or from the device.

MASS TRA. The TRANSFER signal from the selected device to the controller acknowledging DEMAND.

MASS C15-C00, MASS CPA. The 16 bit bidirectional control bus data path, with odd parity, between the controller and the device.

MASS INIT. The INITIALIZE signal from the controller to the device, when asserted, performs a device reset/initialize function.

MASS FAIL. This signal is asserted when the controller has a power failure.

MASS ATTN. The ATTENTION signal is asserted by the device when it requests service from the controller. The controller determines which device (or devices) is (are) driving the attention line by reading the attention summary (pseudo-) register.

3.3.3 Massbus Control Bus Protocol and Timing

To be supplied.

3.3.4 Massbus Data Bus Signal Definition

MASS RUN. The RUN signal from the controller to the device, when first asserted, indicates the controller is ready to begin a data bus transfer. Thereafter, the device samples the RUN signal at the trailing edge of each EBL pulse and if it is still asserted, the operation normally continues.

MASS OCC. The OCCUPIED signal from the device to the controller, when asserted, indicates the device is ready to begin a data bus transfer. The assertion of OCC is not

dependent on receiving RUN. If RUN has been negated, on the trailing edge of EBL, OCC will be negated.

MASS SCLK. The SYNC CLOCK signal from the device to the controller. This signal indicates to the the controller that data is available on the data bus for a device read or requests data from the controller for a device write.

MASS WCLK. The WRITE CLOCK signal from the controller to the device returns the SCLK received from the device, telling the device data is available on the data bus for a device write.

MASS D17-D00, MASS DPA. The 18 bit bidirectional data path with odd parity between the controller and the device.

MASS EBL. The END OF BLOCK signal from the device to the controller, when asserted, indicates that the device has completed transferring as many data words as it was requested to transfer by the last read/write command.

MASS EXC. A bidirectional signal, when asserted by either the controller or the device, indicates an error condition was detected during the data bus transfer.

3.3.5 Massbus Data Bus Protocol and Timing

To be supplied.

3.4 Current and Future Massbus Devices

See Figure 3.4.1 for transfer rates of current and future Massbus devices.

3.5 Description of Channel Operation

This section will attempt to describe the channel related operations of the TEN and VAX systems. It is most important that the appropriate software people review this section.

3.5.1 TEN Channels

In the following discussion, the term TEN is used to denote either a DEC10/TOPS10 or DEC20/TOPS20 system. The DEC10 is

assumed to have internal channels, which makes the hardware configurations similar. Refer to figure 3.5.1.

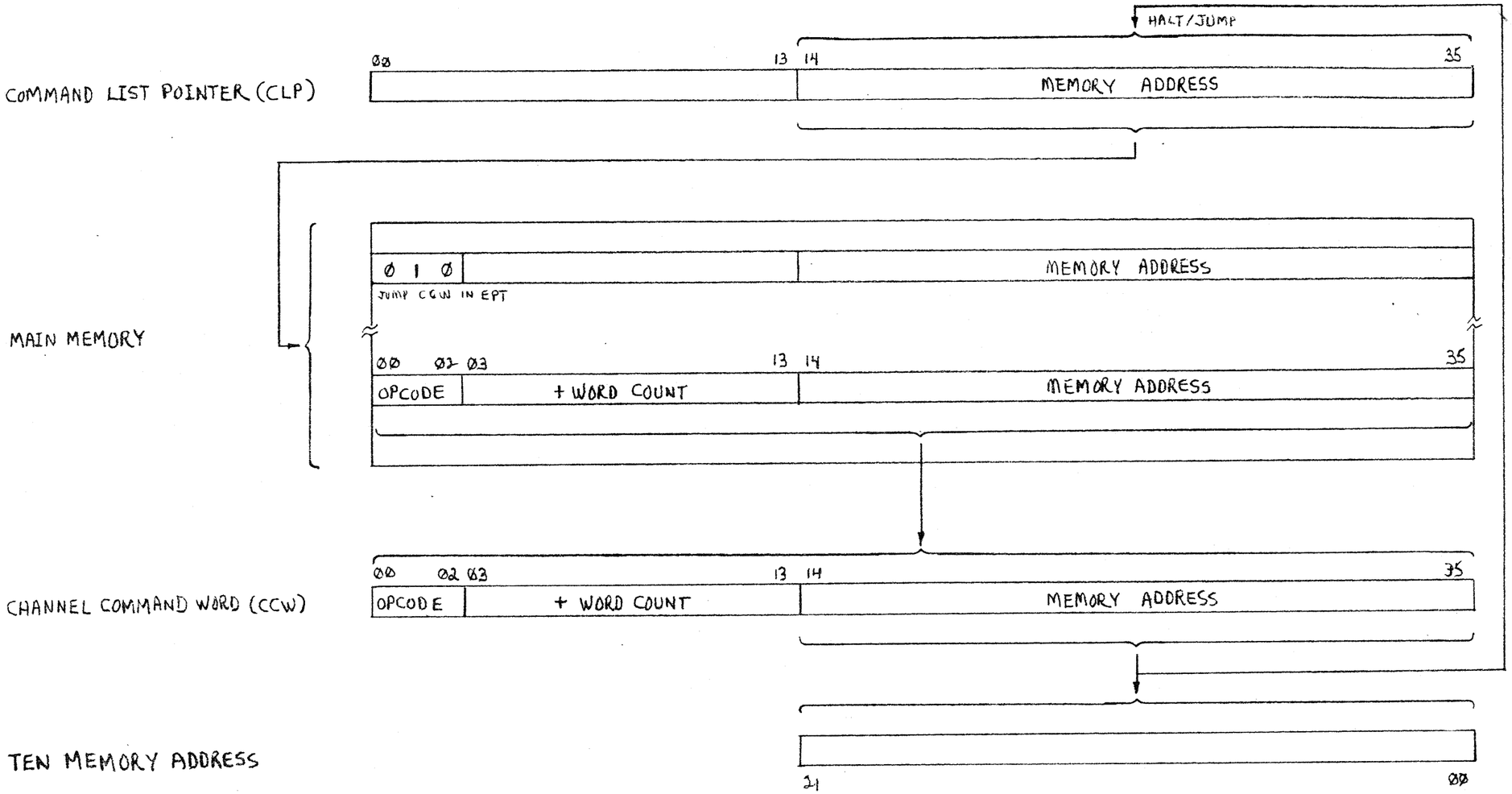
The functionality of the channel is divided between the RH20 and the MBOX. In simplified terms, the RH20 provides device control and the MBOX provides memory control. In this regard, only the block address register (BAR) and transfer control register (TCR) of the RH20 and the command list pointer (CLP) and channel command word (CCW) of the MBOX will be discribed. Refer to figure 3.5.2.

The BAR is loaded with the selected surface and sector for disks or the frame count for tapes. The TCR is loaded with the two's complement of the number of blocks to be transferred and the function code to be loaded into the device control register. After the TCR is loaded, the RH20 will transfer the contents of the BAR to the device, if the BAR has been loaded (loading the BAR is optional for disks). It then will transfer the function code of the TCR to the device and load the block counter in the RH20 with the block count. The RH20 also stores the direction of transfer (device read or device write) and signals the MBOX over the CBUS. The TCR also contains a bit which can reset the command list pointer in the MBOX (see CLP below).

Note on block count: In TEN systems the disk format (RPs) has 128 36-bit words per sector. The block count keeps track of the number of end of block (EBL) signals received from the device. In the case of disk transfers, to properly terminate, the RH20 requires that the total number of words transferred must equal 128 times the block count, i.e., WC = 128 x BC (see SKIP function below). In the case of tape transfers, the block count is always loaded with a one (1) since only one record can be transferred per device command.

The CCW contains a 3-bit operation code, space for an 11-bit positive word count, and a 22-bit memory address. The operation codes defined are HALT, JUMP, FORWARD DATA TRANSFER (FDT), LAST FORWARD DATA TRANSFER (LFDT), REVERSE DATA TRANSFER (RDT) and LAST REVERSE DATA TRANSFER (LRDT). The difference between FORWARD and REVERSE is that the memory address is incremented or decremented respectively. Data transfer op codes will be abbreviated XDT or LXDT, where "X" can be either "F" or "R". The word count specifies the number of 36-bit words to be transferred and only has meaning for XDT or LXDT. The memory address specifies the starting address of the buffer for (L)XDTs or the address of the next CCW for HALT or JUMP op codes.

The CLP contains a 22-bit memory address which specifies the address of the next CCW. The CLP can be incremented by one (+1) when the word count of the current CCW decrements to zero and point to the next CCW, it can be loaded from a CCW in the case of a HALT or JUMP op code, or it can be (p)reset to point to a location in the executive process table (EPT)



TEN DATA WORD

00	17 18	35
1		2

	00 05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20 29	30 35
S/P BAR	70/72	L_R	X	X	X	X	X	X	X	X	DEV SEL			X	X	SURFACE & SECTOR (DISK) OR FRAME COUNT (TAPE)	
S/P TCR	71/73	L_R	R_{CLP}	X	X	$STOR_E$	X	X	X	X	DEV SEL			X	DTE_S	- BLOCK COUNT	FUNCTION CODE

TEN CHANNEL OPERATION

FIGURE 3,5.2

corresponding to the respective channel number. The latter is only done at the start of a transfer and is controlled by the reset bit in the TCR. The CCW stored in the EPT is (generally) a JUMP CCW. When the CCW is fetched and decoded as a JUMP, the address field of the CCW is loaded into the CLP and the next CCW is fetched from this address. This process continues until an (L)XDT CCW is fetched. When an (L)XDT CCW is fetched, the actual data transfer is started and continues until the word count decrements to zero. At this point the CLP is incremented and if the current CCW was not decoded as an LXDT, the next CCW is fetched. This process continues until an LXDT CCW is fetched and the word count reaches zero or a HALT is fetched, at which time the transfer is terminated. If the last CCW fetched is a HALT, the address field of the CCW will be loaded into the CLP. The JUMP CCW is provided so that a contiguous block of memory does not have to be allocated for a particular command list.

As mentioned above, in order to terminate a disk transfer properly, the total word count must be equivalent to an integral multiple of sectors. This works fine for TOPS20 since the minimun transfer size is a page which is equivalent to four sectors (blocks) or 512 words. However, for TOPS10, which was developed prior to paging, the size of a transfer can be an arbitrary number of words, i.e., WC not equal to 128 x BC. To handle this case, a SKIP capability is provided. A SKIP operation will be performed when an (L)XDT CCW is decoded and the memory address field is all zeros. On a device read, the number of words specified by the word count are read from the device but not transferred into memory (dropped into the bit bucket). Similarly, on a device write, a number of words are fetched from the EPT (address 60) and transferred to the device. In this way, the total word count can be made equal to an intergral number of sectors, eg, if 120 "real" words are required, two CCWs are generated, one XDT specifying 120 words and another LXDT specifying 8 words and a zero address field.

A complication that arises because of the SKIP function is described as follows. Suppose a controller has only a one word data buffer sufficient to handle two device transfers and doesn't do any prefetching of CCWs (a'la DF10). If the controller decodes a data transfer and fetches the first word from memory (assume a device write) and starts the device and the device requests the first data word immediately and the controller finds out it has only a one word word count! In this case, the controller may have to request use of the system bus three times, each time contending for the bus and a memory access time. The three requests can occur because the first request may yield a JUMP CCW, the next request a data transfer CCW, the third request the data word -- and all this has to be accomplished within the time of two device requests! At maximum Massbus speed this time would only be 1 microsecond. Clearly, one

must provide more data buffering and either prefetch CCWs or provide a mechanism that eliminates the contention and access time in fetching CCWs (i.e., providing local storage of CCWs). You don't want to have an arbitrarily large data buffer since this has ramifications in the case where you want to read one sector and write the next, or vice versa. If only one data buffer is used you will have to empty and refill the buffer within the sector gap time. In any event, an analysis should be performed to determine the size of the data buffer, taking into account the latency (contention plus access) of a heavily loaded system.

The chief disadvantages of the TEN format is that it doesn't provide byte offsetting and it requires some fudging to get around the block count (as part of the SKIP problem the ending status is no longer valid since the address field no longer reflects the last address read or written).

3.5.2 VAX Channels

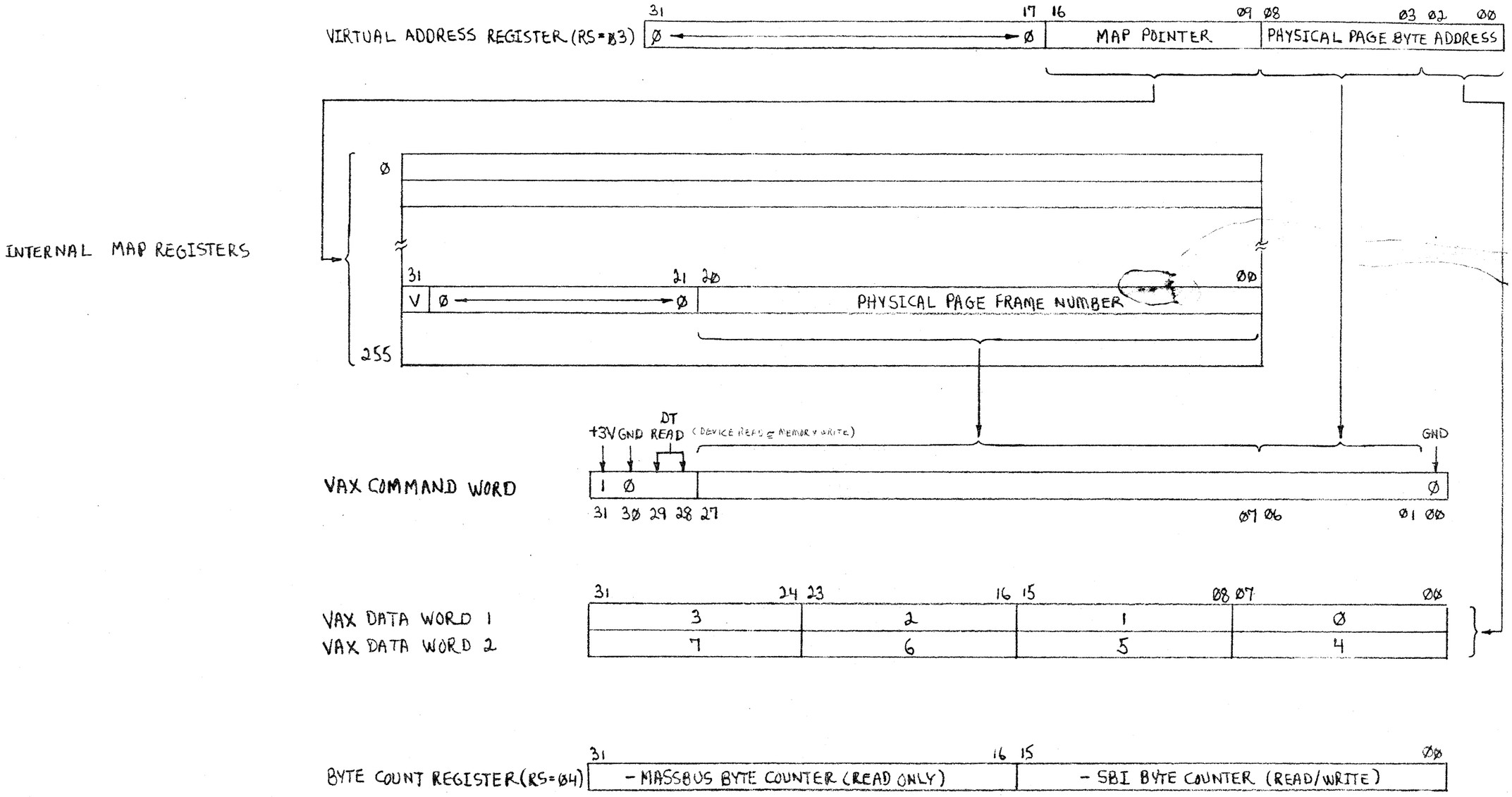
In the following discussion, the term VAX is used to denote a VAX/VMS system. The functionality of the channel is entirely provided by the RH780. Only the Virtual Address Register (VAR), Map Registers (MR) and Byte Count Register (BCR) will be described. See Figures 3.5.3 and 3.5.4.

Before proceeding, some terms will be defined. A "word" is a 16 bit quantity, a "longword" is a 32 bit quantity and a "quadword" is a 64 bit quantity. A quadword is synonomous with two longwords, four words or eight bytes. In the VAX, there are 64 quadwords/page, 128 longwords/page or 512 bytes/page.

The Virtual Address Register (VAR) consists of two fields, a 9 bit Physical Page Byte Address (PPBA) that points to a particular byte within a page and an 8 bit Map Pointer (MP) that selects one of 256 Map Registers (MR). The least significant three bits of the PPBA form a counter that is used to point to a particular byte within a quadword. This three bit field will be referred to as the Byte Offset (BO). The remaining 6 bits of the PPBA can be thought of as a "Physical Page Quadword Address" (PPQA). The MP and PPQA are concatenated to form a 14 bit counter.

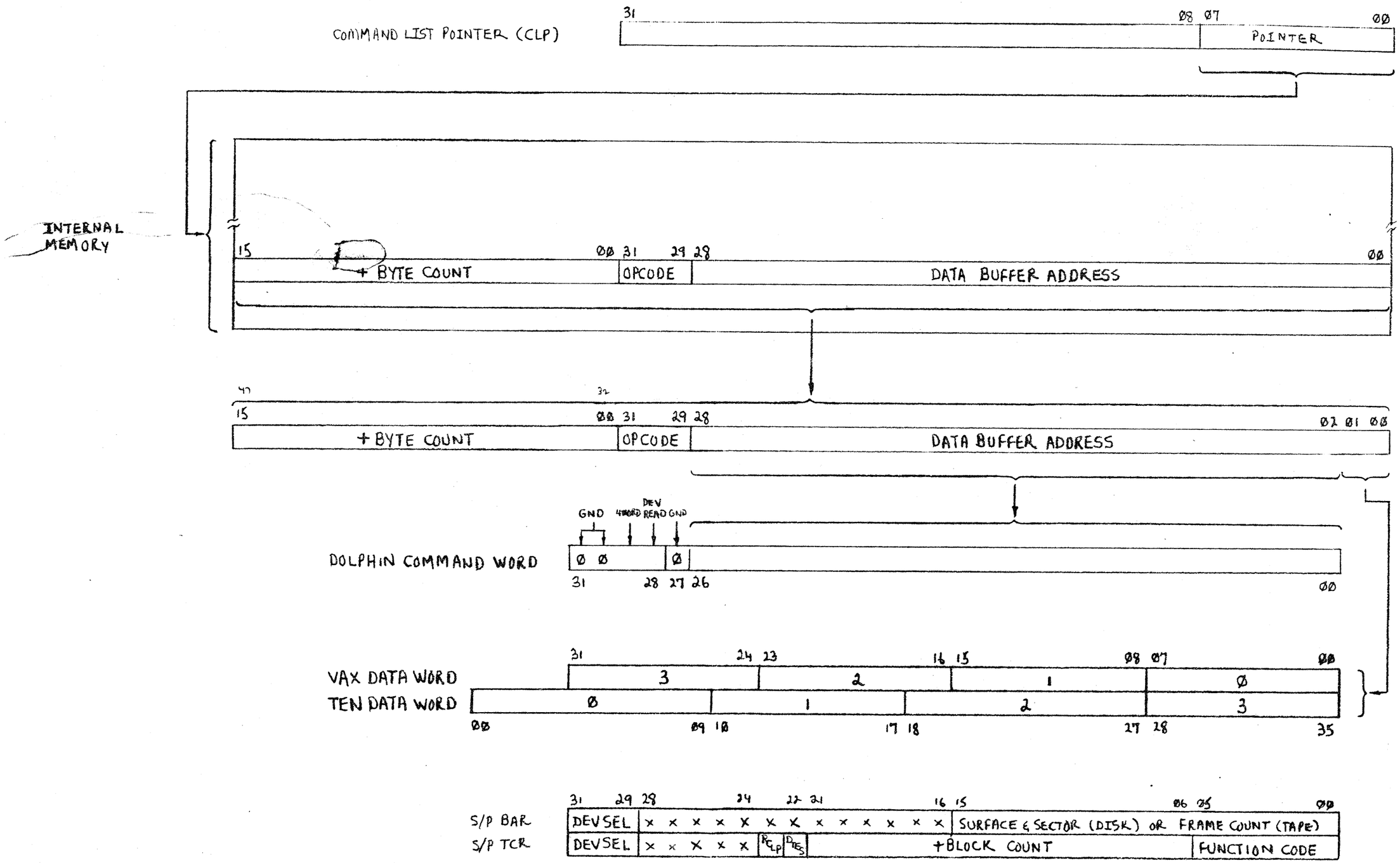
The Map Registers (MR) contain a 21 bit Physical Page Frame Number (PPFN) that specifies a particular page of memory to be referenced. A bit is also present marking the MR as valid. There are 256 Map Registers.

The Byte Count Register (BCR) consists of two 16 bit counters, a Massbus Byte Counter (MBC) and an SBI Byte Counter (SBC). The program loads the SBC with the 2s complement of the total number of bytes to be transferred for a given transfer. The RH780 will load the same value



VAX CHANNEL OPERATION

FIGURE 3.5.4



PROPOSED DOLPHIN CHANNEL OPERATION
FIGURE 3.5.5

into the MBC. The MBC is incremented twice for each SCLK, once on the leading edge of SCLK and once on the trailing edge of SCLK. The SBC is incremented each time a byte within a quadword is accessed. The overflow of both counters terminates the transfer successfully.

The physical address on the SBI references a longword address. Since the RH780 only performs quadword transfers over the SBI that are aligned on even address boundaries, it does not have to assert the least significant bit of the physical address. Bits 08-03 of the VAR are used directly to form bits 06-01 of the physical address and bits 20-00 of the MR are used directly to form bits 27-07 of the physical address. The appropriate MR is selected by the MP.

The RH780 contains a byte-wide silo in the data path. The initial state of the BO points to the first byte of the quadword to be accessed. Each time a byte is accessed the BO is incremented by 1 for forward transfers or decremented by 1 for reverse transfers and the SBC is incremented. Each time the BO reaches a count of 7 (forward) or 0 (reverse) a command address is issued to the SBI followed by the quadword of data (device read) or requesting a quadword of data (device write). Termination will be handled differently since all bytes of a quadword may not have been accessed. The BO is also used to generate the appropriate mask bits. At the completion of the current SBI transfer the VAR is incremented or decremented (excluding the BO) and will therefore point to the next quadword. This process will continue until the SBC overflows. If the transfer is larger than 512 bytes, the map pointer field will be incremented or decremented and a new map register will be referenced, i.e., the transfer will continue with a new page.

The chief disadvantages of the VAX format is that the page size is "hard-wired" into the VAR and there is no provision for having a "backup" device command, i.e., when the current transfer completes, the RH780 must wait for the next command to be issued.

3.5.3 Proposed Channel For the Dolphin MBA

The proposal is that the byte handling capability of the VAX MBA be merged with the TEN channel command format and the backup register facility. The TEN channel command format will change to a byte count and byte address. A more definitive explanation will be forthcoming. See Figure 3.5.5.

4.0 PROGRAMMING AND OPERATION

4.1 Introduction

The Massbus Adapter registers are shown in the following table. The table gives the Massbus Adapter Register Address (MARA) in octal, the type of access, the name and mnemonic. Register addresses are based on the least significant 10 bits of the register address received during the COMMAND TAG cycle and are interpreted as word addresses.

Register Address	Access	Name (Mnemonic)
MARA 0000	R/W	Configuration and Status Register (CSR)
MARA 0001	R/W	Control Register (CR)
MARA 0002	R/W*	Status Register (SR)
MARA 0003	R/W	Diagnostic Register (DR)
MARA 0004	R/W	Secondary Block Address Register (SBAR)
MARA 0005	R/W	Secondary Transfer Control Register(STCR)
MARA 0006	R/W*	Primary Block Address Register (PBAR)
MARA 0007	R/W*	Primary Transfer Control Register (PTCR)
MARA 0010	R/W*	Current Channel Command Word 0 (CCCW0)
MARA 0011	R/W*	Current Channel Command Word 1 (CCCW1)
MARA 0012 - 0377	---	Not Used
MARA 0400 - 0777	R*/W	Control Bus Transmit Data Register(CBTDR)
MARA 0400 - 0777	R/W*	Control Bus Receive Data Register (CBRDR)
MARA 1000 - 1377	R/W	Channel Command Word 000 (CCW000) through Channel Command Word 377 (CCW377)
MARA 1400 - 1777	---	Not Used

An asterisk (*) means that access is provided under diagnostic control.

In order to access these registers the CPU must issue a COMMAND TAG (TAG = 01) to the MBA. The generalized format of the data transmitted to the MBA during the COMMAND TAG is described below and shown in figure 4.1.1. The figure shows the correspondence between TEN and VAX bit numbering of the data lines. As a convention, all bit descriptions will only use VAX bit numbering.

Description of COMMAND TAG:

Bits X3 - X0:

MASK 3 - 0 (M3 - M0). The MASK bits should all be zero for the MBA, ie, only one word operations will be performed. If necessary, the MBA can check the MASK bits and signal an error confirmation if they are not all zero. Otherwise the MASK bits will be ignored.

Bits 31 - 28:

Function Code 3 - 0 (FC3 - FC0). The function code describes the operation to be performed by the MBA. The function codes, in octal , recognized by the MBA are:

00 Read Masked One Word (all mask bits should be zero)

01 Write Masked One Word (all mask bits should be zero)

04 Read Masked Lock (treated as Read Masked One Word)

05 Write Masked Unlock (treated as Write Masked One Word)

All other function codes are invalid and will result in an error confirmation.

Bit 27:

IO. This bit must be asserted to address the I/O address space.

Bits 26 - 21:

Controller Address 5 - 0 (CA5 - CA0). The controller address is the address of the intended receiver of the command. This address must match the ID assignment of the controller to initiate the operation specified by the function code.

Bits 20 - 00:

Register Address 20 - 00 (RA20 - RA00). The register address selects a particular register accessible by the controller and is interpreted as a word address. Bits RA09 and RA08 divide the register address into four groups as follows:

0000000 - 0000377 MBA Internal Registers

0000400 - 0000777 MBA External Registers

0001000 - 0001377 MBA Channel Command Registers

0001400 - 7777777 Non-existant Registers

Not all of the 256 available internal registers will be implemented (only 10 are currently defined). The 256 external registers allow direct addressing of up to 32 registers in up to 8 devices on the Massbus. The 256 channel command registers allow storage of up to 128 2-word channel commands. Addressing non-existant registers will result in an error confirmation.

A simplified MBA Register Layout is shown in figure 4.1.2 and an MBA Functional Block Diagram is shown in figure 1.1.2. Detailed

	TEN BIT NUMBERING	04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35				
	VAX BIT NUMBERING	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00				
	MARA 0000 : CSR	FLAGS	ECC SYNDROME	PIL	DEVICE CODE	R/W
	MARA 0001 : CR	NOT USED			CONTROL	R/W
	MARA 0002 : SR	NOT USED			STATUS	R/W*
	MARA 0003 : DR	NOT USED			DIAGNOSTIC	R/W
	MARA 0004 : SBAR	DEVSEL	NOT USED		SURFACE/SECTOR (DISK) OR FRAME COUNT (TAPE)	R/W
	MARA 0005 : STCR	DEVSEL	NOT USED	BLOCK COUNT	FUNCTION CODE	R/W
	MARA 0006 : PBAR	DEVSEL	NOT USED		SURFACE/SECTOR (DISK) OR FRAME COUNT (TAPE)	R/W*
	MARA 0007 : PTCR	DEVSEL	NOT USED	BLOCK COUNT	FUNCTION CODE	R/W*
	MARA 0010 : CCCW0	OPCODE	DATA BUFFER ADDRESS			R/W*
	MARA 0011 : CCCW1	NOT USED			BYTE COUNT	R/W*
	MARA 0400 - 0777 : CBTDR	NOT USED			CONTROL BUS DATA	R*/W
	MARA 0400 - 0777 : CBRDR	DEVSEL	REGSEL	NOT USED	CONTROL BUS DATA	R/W*
EVEN	MARA 1000 - 1377 : CCW000 - CCW376	OPCODE	DATA BUFFER ADDRESS			R/W
ODD	MARA 1000 - 1377 : CCW001 - CCW377	NOT USED			BYTE COUNT	R/W

SIMPLIFIED REGISTER LAYOUT

FIGURE 4.1.2

descriptions of the registers and functional blocks are given in the following sections.

All registers and pointers in the MBA are initialized to the appropriate state on power-up and by writing a "1" into bit 02 (RHINIT) of the Control Register (CR). A Massbus initialize is generated by writing a "1" into bit 03 (MBINIT) of the Control Register (CR). All unused bits of registers will be read as "0" and writing them will have no effect except for parity checking. This applies to unused registers as well. In general, bits will be cleared by writing "0" and set by writing "1". This applies mostly to the Status Register (SR) since no "clear errors" bit is provided, ie, individual error bits must be cleared by writing a "0" or RHINIT. All registers contain a maximum of 32 bits of data.

4.2 MARA 0000: Configuration and Status Register (CSR).

This register contains information which is common to all devices that attach to the Dolphin Bus. The information described is only tentative and subject to change. Bits 07-00 are read only since they are "hard-wired" into each device. The remaining bits will be cleared by writing zero (0) or RHINIT. See Figure 4.2.1.

Bits 07-00:

Device Code 7-0 (DC7-DC0). This code is provided to uniquely identify each type of device attached to the Dolphin Bus in order to auto-configure the system. The types of devices being considered and their device codes are:

DEVICE TYPE	DEVICE CODE
"VAX" CPU	To be supplied.
"TEN" CPU	"
Memory (MEM)	"
Console (CSL)	"
Dolphin Bus Repeater (DBR)	"
Unibus Adapter (UBA)	"
Massbus Adapter (MBA)	"
Channel Bus Adapter (CBA)	"
HSC-50 Adapter (HSCA)	"

Bits 12-08:

Priority Interrupt Level 4-0 (PIL4-PIL0). This five bit field is the encoded priority interrupt level assigned to the controller. If this field is all zeroes, interrupts will be inhibited.

Bits 19-13:

ECC Error Syndrome (C40-C0). This seven bit field latches the error syndrome when a Single Bit Error (SBE) is detected off the Dolphin Bus so that the failing bit can be identified. The information will remain latched until SBE is cleared by the software.

Bit 20:

Lost Error Bit (LEB). This bit will set if a previous error has latched an "error bit or field" and another error within the field is detected.

Bit 21:

Read Data With Error Tag Received (RDWETR). This bit will set if a tag of Read Data With Error is received. This will occur, e.g., on a memory read when the memory controller detects an uncorrectable double bit error (DBE) when the data is retrieved from the memory array.

Bit 22:

Invalid Function Code Received (IFCR). This bit could be used under the following conditions: unused function code received; a write unlock received without previously receiving a read lock; receiving a function code not supported by the controller; a mask bit configuration not supported by the controller; etc. The use of this bit is not fully defined for the MBA or any other controller.

Bit 23:

Invalid Tag Received (ITR). This bit could be used under the following conditions: unused tag received; a read data tag received without having previously issued a "read" command tag (unexpected read data); write data tag not received on cycle following a "write" command tag; too many/too few data tags received, e.g., the number of data tags received does not correspond to the number indicated by the original command tag function code (detected by responder on write, by commander on read); etc. Again, this bit is not fully defined.

Bit 24:

Timeout on Read Data (TORD). The setting of this bit indicates that read data is not received within some specified period of time. This is somewhat device dependant since the question arises as to how long the device can wait? In this sense it is

related to "data overrun" condition. However, a maximum time may want to be determined for the system.

Bit 25:

Timeout on Command Tag (TOCT). The setting of this bit indicates that a command tag did not get accepted within some specified amount of time due to repeated busy confirmations. Same comment applies here as on TORD.

Bit 26:

Error Confirmation Received on Command Tag (ECROCT). No further explanation required. The question is what do you do when this happens? If in the middle of a data transfer do you abort? Hardware retry is not implemented.

Bit 27:

Null Confirmation Received on Command Tag (NCROCT). Same comment applies here as on ECROCT.

Bit 28:

Single Bit Error (SBE). This bit will be set when a correctable single bit error is detected in a received message. It will assert FAULT if an interrupt is enabled for this condition and will latch the syndrome. FAULT will stay asserted (if enabled) and the syndrome latched, until the software clears SBE. An enable, should be provided, so one of the other flags will have to be eliminated.

Bit 29:

Double Bit Error (DBE). This bit will set if an uncorrectable double bit error is detected in the received message. It unconditionally asserts FAULT and generates an interrupt. FAULT will stay asserted until the software clears DBE.

Bit 30:

Receive Fault (RFAULT). The setting of this bit indicates the controller detected FAULT assertion while it was transmitting a message on the bus. Cleared by software.

Bit 31:

Transmit Fault (TFAULT). The setting of this bit indicates the

controller detected a condition that causes it to assert FAULT on the bus. Cleared by software.

4.3 MARA 0001: Control Register (CR)

This register contains bits that control the operation of the MBA. See figure 4.3.1.

Bit 00:

Attention Interrupt Enable (AIEN). This bit is set to enable Massbus Attention (MBATTN) to cause an interrupt. Cleared by RHINIT.

Bit 01:

~~Disable~~ Register Access Error Interrupt (DRAEI). If this bit is set it will prevent an RAE from generating an interrupt. DRAEI is cleared by RHINIT.

It is intended that this bit perform a function similar to the Disable Register Access Error Stop (DRAES) bit in the RH20. The following describes the difference between the RH20 implementation and what is proposed for the RH30.

In the RH20, if DRAES is set it will prevent an interrupt when an RAE occurs and also permits changing the device select and register select by allowing the preparation register to be loaded. Conversely, if DRAES is cleared, an RAE will generate an interrupt and will inhibit loading of the preparation register (or any internal register as well!). For the RH30 it is proposed that if DRAEI is set it will prevent an interrupt when an RAE occurs and if it (DRAEI) is cleared an RAE will generate an interrupt, but regardless of the state of DRAEI, the device select and register select information is latched. Note that the device select and register select information is stored in the RH30 and in that sense can be thought of as a "preparation register". The difference is that in the RH30 this register is not directly addressable as it is in the RH20. In the RH30, the "preparation register" stores the information received from the Dolphin Bus during the command cycle.

The reason for disabling the interrupt is so that you can enter a "RETRY LOOP" where you can read or write an external register "N" times without generating an interrupt each time an RAE occurs. This is the same in both the RH20 and RH30. The reason for latching the device select and register select information in the event of an RAE is so that a program can set up a sequence of read or write instructions to a group of external registers, execute the sequence (probably in an interrupt service routine)

and check for an RAE at the end of the sequence. If an RAE occurred, the program can then read the CBRDR to determine the failing address and the cause of the failure. In the RH20 approach, the same sequence of instructions would be executed and if an RAE occurred, the failing address and the state of CBTO and CBPE would be lost (unless the RAE occurred on the last instruction).

Bit 02:

RH Initialize (RHINIT). Writing a one (1) in this bit will initialize the MBA but will not generate a Massbus initialize. This signal will be generated on powering up the MBA. Will always be read as a zero (0).

Bit 03:

Massbus Initialize (MBINIT). Writing a one (1) in this bit will generate a Massbus Initialze but will not initialize the MBA. This signal will be generated on powering up the MBA. The MBINIT pulse will be at least 400 microseconds. Will always be read as zero (0).

Bit 04:

Format (FORMAT). The format bit selects the format to be used by the MBA during data transfers. FORMAT = 0 selects 32-bit formatting and FORMAT = 1 selects 36-bit formatting. This bit will reverse the order that bytes are accessed, ie, with FORMAT = 0, bytes are accessed right to left and with FORMAT = 1, bytes are accessed left to right. This bit will be in the zero (0) state on power up and will be cleared by RHINIT. See figure 4.3.2.

Bits 31-05

Not used. Read as zero (0).

4.4 MARA 0002: Status Register (SR)

This register contains all the status and error bits specifically related to internal and device operations. See Figure 4.4.1.

The following description gives an overview of the operation of some of the bits in this register. For a definition of these bits, see below. The Command Done (COMDON) bit generates a non-maskable interrupt. This bit signifies that a command has

been completed, with or without error. Massbus Attention (MBATTN) will generate an interrupt if Attention Interrupt Enable (AIEN) is set. If AIEN is not set an interrupt will not be generated but the state of the MBATTN bit can still be read. A Register Access Error (RAE) will generate an interrupt if Disable Register Access Error Interrupt (DRAEI) is not set. If DRAEI is set, an interrupt will not be generated but the state of the RAE bit can still be read. An RAE is caused by either a Control Bus Parity Error (CBPE) or a Control Bus Timeout (CBTO).

Bits 14-05 of this register are grouped together as Transfer Errors (XFERER). Some of these errors are defined as fatal errors and some are defined as non-fatal errors. The difference is in the handling of Command Done (COMDON). A fatal error will cause the controller to negate RUN when the error is detected. The device will then negate OCC on the trailing edge of the next EBL (the end of the current block) and allow COMDON to set. For a non-fatal error the program has an option as to when it wants to terminate the transfer. This is controlled by the state of the Disable Transfer Error Stop (DTES) bit in the primary transfer control register. If DTES is cleared, a non-fatal error is treated as a fatal error. However, if DTES is set, the controller will log the error but will not negate RUN until all the blocks specified by the device command have been transferred. The motivation for doing this is to allow the program to perform a number of retries and if not successful, the last retry will set DTES to allow the program to transfer as much data as possible before terminating. The program can then decide what to do with the data. The errors considered fatal are: Device Response Error (DRE), Data Overrun (DAOVRN), Long Word Count Error (LWCE), Short Word Count Error (SWCE), Non-existent Memory (NXM) and Channel Command Word Parity Error (CCWPE). Transfer errors that are considered non-fatal are: Massbus Exception (MBEXC), Data Bus Parity Error (DBPE), Data RAM Parity Error (DRPE), Write Check Error (WRCKER) and Read Data With Error Tag Received (RDWETR) (see section 4.2). If a DBPE or DRPE is detected on a device read, the bad data will be written into memory with correct ECC generated on the bad data. If an RDWETR or DRPE is detected on a device write, the bad data will be sent to the Massbus with no adjustment in parity. This is because parity is not generated when sending data to the Massbus. It is also planned that the parity checker on the Massbus data lines will be enabled for device writes, so that a DRPE should also cause a DBPE.

The Massbus Spec lists a few other error conditions that can be checked. It recommends that if SCLKs are not received from the device within 250 milliseconds after the controller asserts RUN, an error should be flagged. Also,if EBL is not received within 250 milliseconds after the controller negates RUN, an error should be flagged. Both of these timeouts would be disabled if OCC is still asserted. How about a timeout on OCC after loading the device command register? The spec is unclear!

Bit 00:

Command Done (COMDON). This bit indicates that a data transfer command has terminated, with or without error. Setting of this bit causes an unconditional interrupt. This bit is cleared by writing a zero (0) or by RHINIT.

Bit 01:

Massbus Attention (MBATTN). Indicates that a device on the Massbus has asserted the Attention line. This bit will cause an interrupt if Attention Interrupt Enable (AIEN) is set. It is cleared by clearing the attention condition in the device.

Bit 02:

Register Access Error (RAE). If this bit sets it indicates either a Control Bus Parity Error (CBPE) occurred when reading an external register or a Control Bus Timeout (CBTO) occurred when reading or writing an external register. If an RAE occures while writing an external register it is a result of a CBTO. However, if an RAE occurs while reading an external register it could be the result of a CBTO or a CBPE. In order to allow the program to distinguish between the two errors, the state of CBTO and CBPE are returned in the Control Bus Receive Data Register (see Section 4.3). An RAE will cause an interrupt if Disable Register Access Error Interrupt (DRAEI) is not set (see section 4.3). Furthermore, the device select, register select, CBTO and CBPE will be latched until RAE is cleared. This will allow the program to read the CBRDR to determine the failing address and the cause of the failure. RAE is cleared by writing a zero (0) or by RHINIT.

Bit 03:

Secondary Transfer Control Register Full (STCRF). This bit indicates that the Secondary Transfer Control Register (STCR) has been loaded with a data transfer command. This bit will be cleared when the Secondary is loaded into the Primary, for transfer to the device, indicating the Secondary is available. Cleared by RHINIT. Can be cleared by writing a zero (0) to disable transfer of the secondary to the primary (?).

Bit 04:

Primary Transfer Control Register Full (PTCRF). This bit indicates the Primary Transfer Control Register (PTCR) has been loaded and that a data transfer command is in progress. This bit is normally cleared by Command Done (COMDON) setting, enabling the transfer of the secondary to the primary if STCRF is set. Clearing this bit will be inhibited if the command terminates

with an error so the contents can be preserved.

Bit 05:

Massbus Exception (MBEXC). When this bit sets it indicates a Massbus device detected an error condition during a data transfer. Is there any reason why we want to drive this bit? Cleared by writing a zero (0) or by RHINIT.

Bit 06:

Data Bus Parity Error (DBPE). When this bit sets it indicates that the controller detected even parity over the data bus lines during a read data transfer. Cleared by writing a zero (0) or by RHINIT.

Bit 07:

Data RAM Parity Error (DRPE). Indicates that the controller detected a parity error while trying to read the data buffer (either a read or write data transfer). Cleared by writing a zero or RHINIT.

Bit 08:

Write Check Error (WRCKER). Indicates that the controller detected a difference between the data received from the data bus and the data read from the Data Buffer during a write check data transfer command. Cleared by writing a zero (0) or by RHINIT.

Bit 09:

Device Response Error (DRE). If this bit sets it indicates that a Control Bus Timeout (CBTO) occurred when the controller tried to transfer the device command from the PBAR or PTCR (see Section 4.8 and 4.9) to the device to initiate a data transfer operation. It seems desirable to distinguish between an RAE and a DRE since the first occurs under software control and the latter under hardware control. A DRE sets COMDON which in turn generates an interrupt. DRE is cleared by writing a zero (0) or by RHINIT.

Bit 10:

Data Overrun (DAOVRN). When this bit sets it indicates the data buffer in the adapter was either full for a device read or empty for a device write when an SCLK is received from the device. Cleared by writing a zero (0) or by RHINIT.

Bit 11:

Long Word Count Error (LWCE). Set on a device read if last transfer is asserted (see section 4.10) and the word count is not zero and the block count is zero and the data buffer is empty. Set on a device write if last transfer is asserted and the word count is not zero and the block count is zero and the data buffer is not empty. LWCE also occurs on a read (with last transfer) if the word count is not zero and the block count is zero and the data buffer is not empty, but empties before the word count goes to zero. Also, on a write (with last transfer) if the word count is zero and the block count is not zero and the data buffer is not empty, but an EBL is received that makes the block count go to zero before the buffer empties. Cleared by writing a zero (0) or RHINIT.

Bit 12:

Short Word Count Error (SWCE). Set on a device read if last transfer asserted and the word count is zero and the data buffer is empty. SWCE also occurs on a device read (with last transfer) if the word count is not zero and the block count is zero and the data buffer is not empty, but the word count goes to zero before the data buffer empties. Also, on a write (with last transfer) if the word count is zero and the block count is not zero and the data buffer is not empty, but the data buffer empties and the block count does not go to zero with the next EBL (we could also have the case where the data buffer empties and another SCLK is received, in which case the buffer would "underflow"). Another example of SWCE is if we are doing a read the word count is zero and the block count is not zero and the data buffer is empty, but an SCLK is received, or an EBL that does not cause the block count to go to zero. The final case of SWCE is if we are doing a write and the word count is zero and block count is not zero and the data buffer is empty, but another SCLK is received, or an EBL is received that does not cause the block count to go to zero. Cleared by writing a zero (0) or RHINIT.

The following table summarizes LWCE and SWCE.

					ACTION
LAST TRANSFER:	WC>0	BC>0	FC=0	RD	CONTINUE
	WC>0	BC>0	FC=0	WR	CONTINUE
	WC>0	BC>0	FC>0	RD	CONTINUE
	WC>0	BC>0	FC>0	WR	CONTINUE
	WC>0	BC=0	FC=0	RD	LWCE
	WC>0	BC=0	FC=0	WR	LWCE
	WC>0	BC=0	FC>0	RD	LWCE, SWCE, NORM
	WC>0	BC=0	FC>0	WR	LWCE
	WC=0	BC>0	FC=0	RD	SWCE, NORM
	WC=0	BC>0	FC=0	WR	SWCE, NORM
	WC=0	BC>0	FC>0	RD	SWCE
	WC=0	BC>0	FC>0	WR	LWCE, SWCE, NORM
	WC=0	BC=0	FC=0	RD	NORM
	WC=0	BC=0	FC=0	WR	NORM
	WC=0	BC=0	FC>0	RD	SWCE
	WC=0	BC=0	FC>0	WR	LWCE

The error conditions indicated must be reviewed for validity in the case of a write check function. Furthermore, it should be noted that the controller will not attempt to handle an odd number of words over the Massbus.

Bit 13:

Non-Existent Memory (NXM). When this bit sets it indicates that memory did not respond to a request within a predetermined time (?). Cleared by writing a zero (0) or by RHINIT.

Bit 14:

Channel Command Word Parity Error (CCWPE). Indicates that the controller detected a parity error while trying to read the CCW buffer. Cleared by writing a zero (0) or by RHINIT.

Bit 31-15:

Not used. Read as zero (0).

4.5 MARA 0003: Diagnostic Register (DR).

This register contains various control bits to allow a program to verify the functionality of the Massbus Adapter. Not all bits have been competely defined but the intent is to provide the capability of simulating all operations, including data transfers, without a device attached to the Massbus. Whether the simulation will be controlled by setting specific bits in some

sequence or whether certain bits define an operation that is controlled by a single step clock has not been decided. This decision will be dependent on how the control logic will be implemented. All bits cleared by RHINIT. See Figure 4.5.1.

Bit 00:

Diagnostic Mode (DM). This bit will enable all of the diagnostic functions to be performed.

Bit 01:

Loopback (LPBK). This bit enables both the Massbus transmitters and receivers on both the control bus and data bus so that data can be wrapped around from the transmitters to the receivers.

Bit 02:

Control Bus Even Parity (CBEP). Setting this bit will cause the controller to generate even parity over the control bus data lines in order to check out the parity detection logic in the device. Further, if loopback is set, the control bus parity checker in the controller can be checked out.

Bit 03:

Attention (ATTN). Setting this bit will simulate the operation of this signal, normally asserted by a device on the Massbus.

Bit 04:

Occupied (OCC). Setting this bit will simulate the operation of this signal, normally asserted by a device on the Massbus.

Bit 05:

Synchronous Clock (SCLK). Setting this bit simulates the operation of this signal, normally asserted by a device on the Massbus.

Bit 06:

End of Block (EBL). Setting this bit simulates the operation of this signal, normally asserted by a device on the Massbus.

Bit 07:

Exception (EXC). Setting this bit simulates the operation of this signal, normally asserted by a device on the Massbus. There is some question as to whether this signal should be generated by the controller under certain error conditions in order to terminate a device during a data transfer.

Bit 08:

Complement Dolphin Bus Parity (CDBP). Setting this bit will cause the controller to complement the received parity bit(s) from the Dolphin Bus. This will permit writing "bad" data into the channel command word buffers and the data buffers in order to check out internal parity checkers on these buffers. Additionally, allowing bad parity to be written into the data buffer will provide a means of checking out the parity network on the data bus in the device, and if loopback is set, the parity network in the controller on the data bus.

Bit 09:

Load Current Channel Command Word (LDCCW). Setting this bit will cause the Current Channel Command Word register (CCCW0 and CCCW1) to be loaded from the Channel Command Word Register (CCWR) pointed to by the Channel Command Word Pointer (CCWP).

Bit 10:

Load Primary Transfer Control Register (LDPTCR). Setting this bit will load the Primary Transfer Control Register (PTCR) from the Secondary Transfer Control Register (STCR).

Bit 11:

Transmit Primary Transfer Control Register (TPTCR). Setting this bit will simulate the transmission of the PTCR to the control bus.

Bit 12:

Write Check (WRTCHK). Not clear how this bit will be used or even if it is necessary!

Bit 31-13:

Not used (or defined yet). Read as zero (0).

4.6 MARA 0004: Secondary Block Address Register (SBAR).

This register is loaded by the software with the "block address" information to be used with the next data transfer command to be transfered to the device. This information will be loaded into the Primary Block Address Register (PBAR) when the PBAR becomes available. This register is cleared by RHINIT. See figure 4.6.1.

Bits 15-00:

This 16 bit field contains the desired surface/desired sector information for disk type devices or the frame count (2s complement) for tape type devices. For disks, the Desired Surface 7-0 (DSF7-DSF0) is in bits 15-08 and the Desired Sector 7-0 (DSC7-DSC0) is in bits 07-00. For tapes the Frame Count is in bits 15-00.

Bits 28-16:

Not used. Read as zero (0).

Bits 31-29:

Device Select 2-0 (DS2-DS0). This three bit field selects the Massbus device to which the "block address" is to be transferred when the command is initiated.

4.7 MARA 0005: Secondary Transfer Control Register (STCR).

This register is loaded by the software with the next data transfer command to be transferred to the device and a block count used by the MBA to terminate the transfer. This information will be loaded into the Primary Transfer Control Register (PTCR) when the PTCR becomes available. Only data transfer commands (read, write, write check) should be loaded into the STCR, Non-data transfer commands (seek, space, etc.) are loaded into the device directly (see section 4.12). This register is cleared by RHINIT. See Figure 4.7.1.

Bits 05-00:

Function Code 5-0 (FCD5-FCD0). This six bit field contains the function code of the data transfer command to be performed by the device. The following table lists the function codes that can be loaded into this register:

LSB = GO Bit
MSB = 1 for data transfer
MSB = 0 for non-data transfer

OCTAL CODE	FUNCTION
51	Write Check Data
53	Write Check Header and Data (treated as Write Check Data)
57	Write Check Reverse
61	Write Data
63	Write Header and Data (treated as Write Data)
71	Read Data
73	Read Header and Data (treated as Read Data)
77	Read Reverse

All Write Check commands perform in a way that appears to be a device write from the memory side of the controller and a device read from the device side of the controller. That is, data is read from memory and compared with the data read from the device. This command is normally preceded with a device write command. It is intended that the same set of CCWs can be used with both commands. There must be a correspondence between these functions and the channel operation code. In other words, if the CCW specifies a reverse data transfer the function code should specify either a write check reverse or read reverse. (See Sections 4.10 and 4.11).

Bit 21-06:

Block Count 15-00 (BLC15-BLC00). This 16 bit field contains the positive block count of the number of blocks to be transferred to/from the device. For a disk type device, the block count represents the number of sectors to be transferred to/from the disk. In this case, the block count should equal the word count divided by the blocking factor of the device. For a tape type device, the block count represents the number of records to be transferred to/from the tape. Since only one record can be transferred to/from tape with a given command, the block count is always loaded with a one (1) for tape transfers.

Bit 22:

Disable Transfer Error Stop (DTES). If this bit is set the controller will ignore the following errors: MBEXC, DBPE, DRPE, RDWETR and WRCKER. The setting of the error bits is not affected

by this bit. If DTES is cleared and these errors occur, transfer will be aborted at the end of the current block. DTES is set it is recommended that error correction in the de (if it exists) be inhibited at the same time. The reason is if error correction is allowed, the error correction information may be invalidated if an error occurs in the next block.

Bit 23

Reset Command List Pointer (RCLP). This bit will reset the command list pointer when a data transfer is initiated. It is mainly intended to be used with a Write Check command so that the previous channel command list can be reused. However, it assumes the CCL begins at location 0!

Bits 28-24:

Not used. Read as zero (0).

Bits 31-29:

Device Select 2-0 (DS2-DS0). This three bit field selects the Massbus device to which the data transfer command is to be transferred when the command is initiated.

4.8 MARA 0006: Primary Block Address Register (PBAR).

This register is loaded by the hardware from the SBAR if the SBAR is full and the last data transfer is completed. This register is a read only register except under diagnostic control and is cleared by RHINIT. See figure 4.6.1.

4.9 MARA 0007: Primary Transfer Control Register (PTCR).

This register is loaded by the hardware from the STCR if the STCR is full and the last data transfer is completed. It is a read only register except under diagnostic control and is cleared by RHINIT. See Figure 4.7.1.

Description of SBAR, STCR, PBAR and PTCR.

According to the Massbus Specification, if the controller wants to access the next sector after completing the current transfer, it has 5 microseconds after the trailing edge of EBL to load the next device command (reference Sector Decision Point). This could take two (2) control bus cycles. Also, the controller

should assert RUN within 100 microseconds after loading a valid data transfer command. The specification is confusing here since in some places it says 10 milliseconds for a write (Section 4.3.3.1), 100 microseconds for a read (Section 4.3.2.1), 10 milliseconds for both (Figures 4.3.3.2 and 4.3.2.3), and 100 microseconds for both (Figures 4.3.3.3 and 4.3.2.3)! To perform a read/write next, or vice-versa, we have the ability to negate RUN and reassert RUN in less than 5 microseconds.

4.10 MARA 0010: Current Channel Command Word 0 (CCCW0).

This register contains the current state of word 0 of the channel command utilized by the controller during data transfer commands. It contains a channel operation code and data buffer address counter. The register is loaded by the hardware from the Channel Command Word Registers (see section 4.14) when a data transfer is initiated and each time the byte count reaches zero providing the current CCW is not an LXDT. Upon termination of the current data transfer, the data buffer address will point to the last byte of memory read or written. If an error condition occurs, loading of this register will be inhibited until the program has a chance to go through its recovery routines. The register is cleared by RHINIT. See Figure 4.10.1.

Bits 28-00:

Data Buffer Address 28-00 (DBA28-DBA00). This 29 bit register is a counter that specifies the address of the data buffer in memory. Bits DBA28-DBA02 form bits 26-00 of the physical address used when sending a COMMAND TAG to the Dolphin Bus. Bits DBA28-DBA02 are incremented (FDT) or decremented (RDT) by 1 (one word transfer) or 4 (four word transfer) at the completion of the current Dolphin Bus cycle. Bits DBA01-DBA00 form a byte offset into the word currently being disassembled (device write) or assembled (device read). As a word is being assembled, the appropriate mask bits are also being generated. Bits DBA01-DBA00 are incremented (FDT) or decremented (RDT) by 1 each time a byte is accessed. Bits DBA01-DBA00 should always have an initial value of 0 (FDT) or 3 (RDT) for a TEN system.

Bits 31-29:

Operation Code 2-0 (OC2-OC0). This three bit field specifies the operation to be performed by the channel logic during a data transfer command. The decoding of these bits is as follows:

OC2	OC1	OC0	OPERATION
0	0	0	Forward Data Transfer (FDT)
0	0	1	Last Forward Data Transfer (LFDT)
0	1	0	Reverse Data Transfer (RDT)
0	1	1	Last Reverse Data Transfer (LRDT)
1	0	0	Skip Last Forward Data Transfer (SLFDT)
1	0	1	Skip Last Reverse Data Transfer (SLRDT)
1	1	0	Reserved
1	1	1	Reserved

Forward Data Transfer means that the Data Buffer Address is incremented with each memory reference and reverse data transfer means that the data buffer address is decremented with each memory reference. A Last Data Transfer (Forward or Reverse) means that the channel will terminate upon the byte count reaching zero. The direction of the transfer (device read or device write) is specified by the contents of the function code field in the PTCR (see Section 4.9).

A "SKIP" means the controller will either read "N" words from the device without transferring them to memory or it will send "N" words of zeroes (0) (with correct parity) to the device. For a write, the data is not obtained from memory. In other words, for a read or a write, no memory references are generated. In all other respects (i.e., termination) a skip works the same as the other operations.

4.11 MARA 0011: Current Channel Command Word 1 (CCW1)

This register contains the current state of word 1 of the channel command utilized by the controller during data transfer commands. It contains the byte counter. The register is loaded by the hardware from the Channel Command Word Registers (see section 4.14) when a data transfer is initiated and each time the byte count reaches zero providing the current CCW is not an LXDT. The register is cleared by RHINIT. See figure 4.11.1.

Bits 15-00:

Byte Count 15-00 (BC15-BC00). This 16-bit register is a counter that specifies the number of bytes to be transferred. It is decremented by 1 each time a byte is accessed in the disassembly or assembly registers. Bits BC01-BC00 should always have an initial value of zero for a TEN system (except for tapes?).

4.12 MARA 0400-0777: Control Bus Transmit Data Register (CBTDR)

This register contains the data for external control bus write operations. A write to this register will initiate a control bus write cycle at the completion of the current control bus cycle, if one is in progress. This register can also be read for diagnostic tests. Register is cleared by RHINIT. See Figure 4.12.1. The register address bits of the command cycle (RA09-RA00) are used for selecting the device and register to be written and are interpreted as follows:

RA09	RA08	RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00
0	1	DS2	DS1	DS0	RS4	RS3	RS2	RS1	RS0

Bits 15-00:

Control Bus Data 15-00 (C15-C00). This 16 bit field contains the data to be written into the device register.

Bits 31-16:

Not used. Read as zero (0).

4.13 MARA 0400-0777: Control Bus Receive Data Register (CBRDR).

This register contains the data received from external control bus read operations. A read of this register will initiate a control bus read cycle at the completion of the current control bus cycle, if one is in progress. The register is cleared by RHINIT. The device and register to be read is selected in the manner described in Section 4.12 and will be returned to the program with bits 31-29 containing the device select and bits 28-24 containing the register select information used to address the control bus. See Figure 4.13.1.

NOTE: The primary reason for having a separate read register (CBRDR) and write register (CBTDR) is to provide a diagnostic loopback path for the control bus.

Bits 15-00:

Control Bus Data 15-00 (C15-C00). This 16 bit field contains the data read from the device register selected by the device select and register select fields of the command cycle.

Bit 16:

Control Bus Parity (CPA). This bit contains the parity bit read

from the control bus during control bus read cycles.

Bit 17:

Control Bus Parity Error (CBPE). The setting of this bit indicates the adapter detected even parity over the control bus data lines during a control bus read operation. A CBPE will cause Register Access Error (RAE) to set in the Status Register (See Section 4.4). (CBPE is disabled when reading the Attention Summary Register.

Bit 18:

Control Bus Timeout (CBTO). The setting of this bit indicates that a device on the Massbus did not respond to Demand (DEM) with Transfer (TRA) within a predetermined time (1.67 microsecond ?). The timeout will be disabled when reading the Attention Summary (Pseudo-) Register in Massbus devices. A CBTO will cause Register Access Error (RAE) to set in the Status Register (See Section 4.4). If CBTO occurs on a read the data in C15-C00 is suspect.

Bits 23-19:

Not used. Read as zero (0).

Bits 28-24:

Register Select 4-0 (RS4-RS0). This five bit field contains the register select bits used to address the Massbus.

Bits 31-29:

Device Select 2-0 (DS2-DS0). This three bit field contains the device select bits used to address the Massbus.

4.14 MARA 1000 - MARA 1377: Channel Command Word Registers 000-377 (CCW000-CCW377).

These registers provide buffering for up to 128 2-word channel commands (CCW). The format is identical to the CCCW0 and CCCW1 (see Section 4.10, 4.11 and Figure 4.14.1). They are loaded under program control before a data transfer is initiated, ie, before loading the STCR (see Section 4.7). The buffer can be accessed by the program while the current transfer is in progress so that the program can be setting up a channel command list (CCL) for the next transfer. A channel command list is a set of

channel command words corresponding to a single device command. A pointer is maintained by the hardware that points to the next CCW to be retrieved. For loading, the software must keep track of which locations can be loaded.

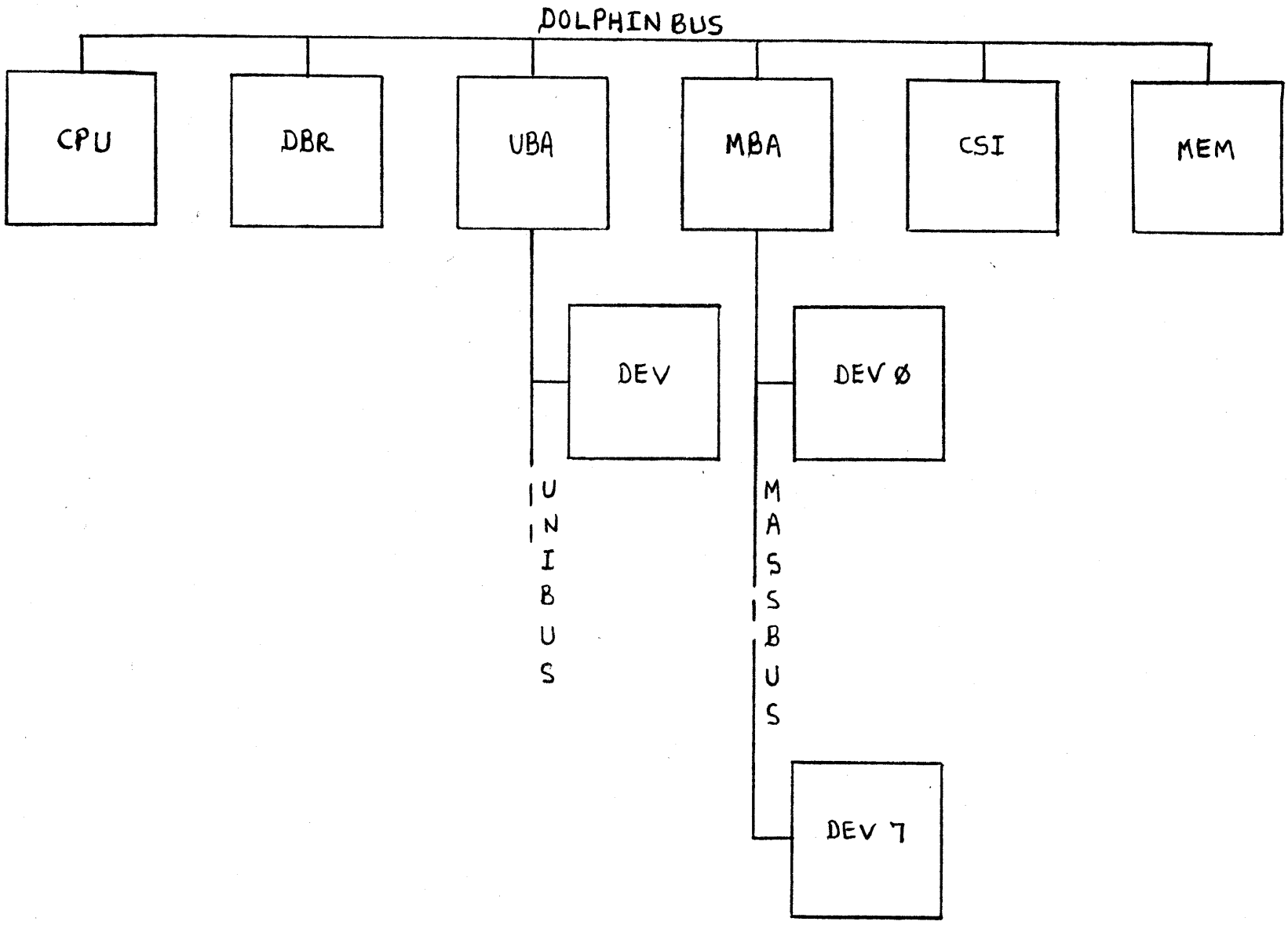
Note: A bit may be added to signal an error if the software tries to write the CCWRs during a data transfer. It is felt that this might aid in program debugging. Comments please!

4.15 Data Transfer Operation

The Function Codes that the MBA is able to generate as part of a COMMAND TAG are:

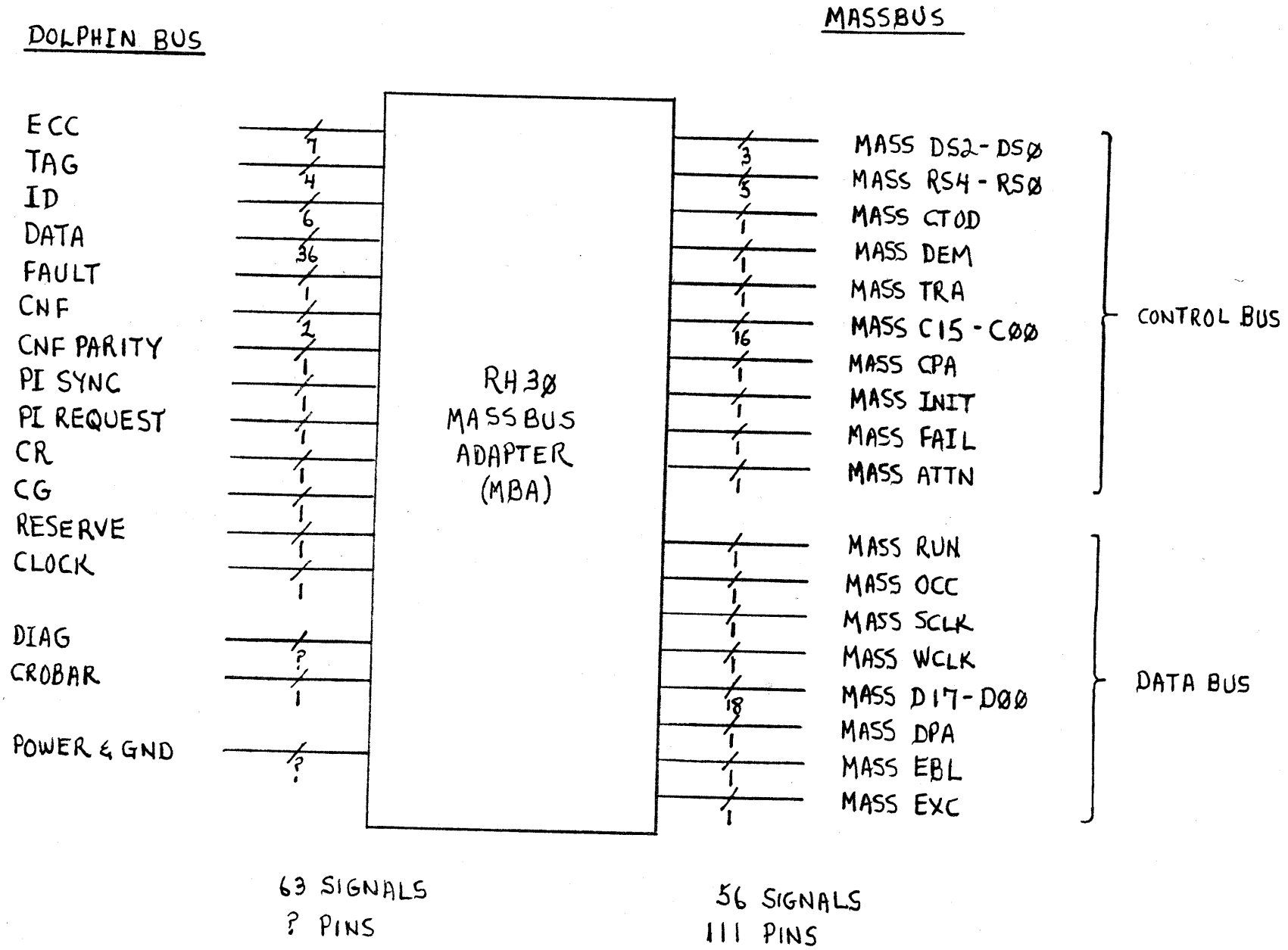
00	Read Masked One Word (all mask bits will be zero)
01	Write Masked One Word (mask bits transmitted with COMMAND)
02	Read Four Words
03	Write Four Words
07	Write Masked Four Words (mask bits transmitted with each DATA word)

Function Codes 00 and 01 are either 32 or 36 bit transfers, 02 and 03 are 36 bit transfers and 07 is a 32 bit transfer.



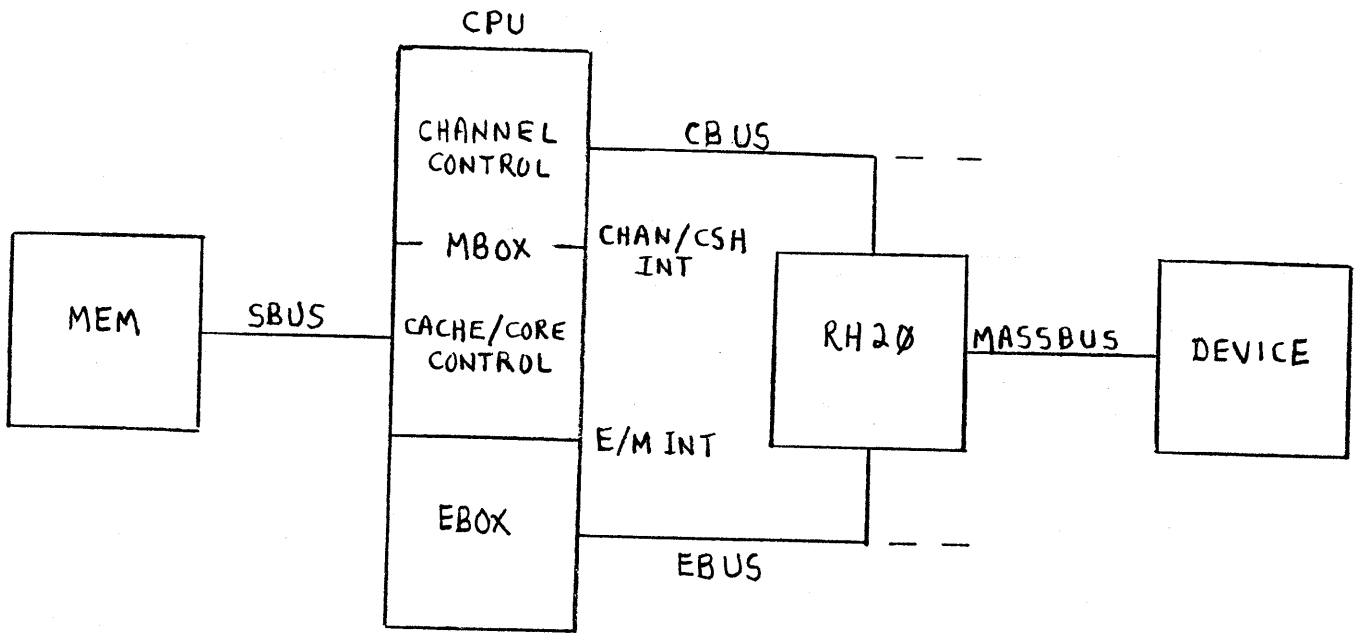
SIMPLIFIED SYSTEM BLOCK DIAGRAM

FIGURE 1.1.1



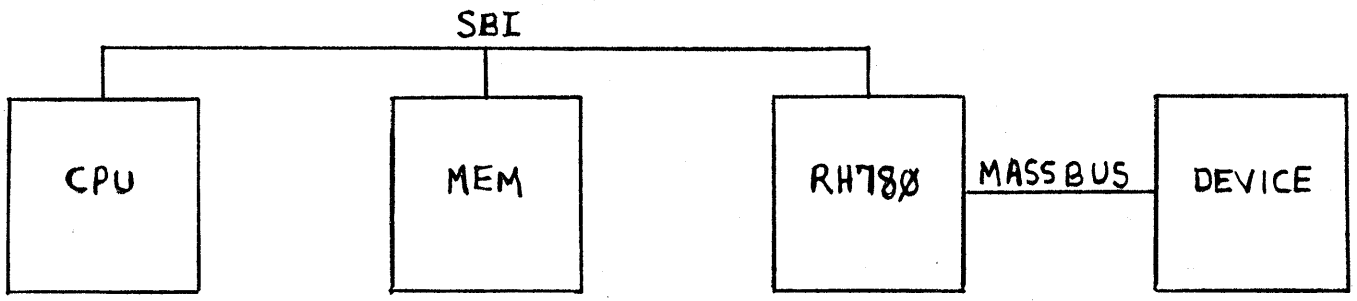
RH3Ø MASSBUS ADAPTER INPUT/OUTPUT SIGNALS

FIGURE 3.1.1



TEN SIMPLIFIED SYSTEM

FIGURE 3.5.1



VAX SIMPLIFIED SYSTEM

FIGURE 3.5.3

TEN BIT NUMBERING

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35

VAX BIT NUMBERING

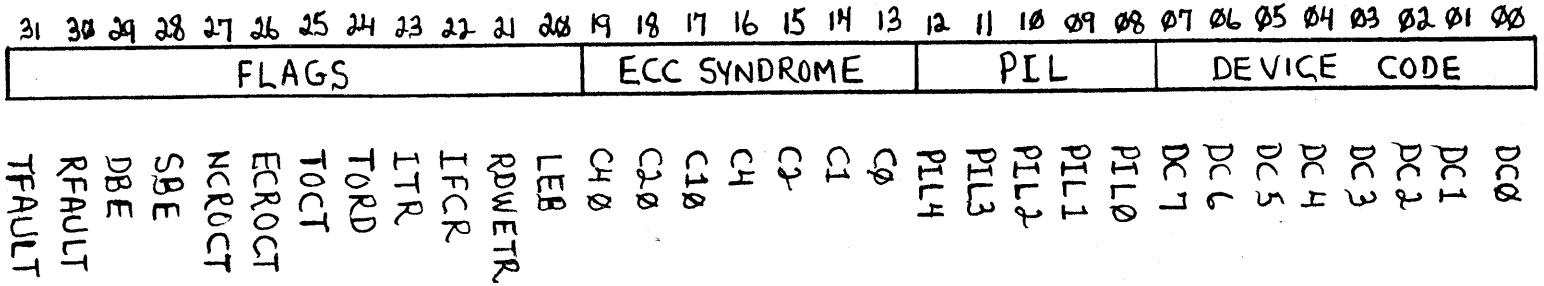
X3 X2 X1 X0 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

MASK	FUNCTION	1	CONT. ADDRESS	REGISTER ADDRESS

M3 M2 M1 M0 FC3 FC2 FC1 FC0 IO CA5 CA4 CA3 CA2 CA1 CA0 RA 20 RA 19 RA 18 RA 17 RA 16 RA 15 RA 14 RA 13 RA 12 RA 11 RA 10 RA 09 RA 08 RA 07 RA 06 RA 05 RA 04 RA 03 RA 02 RA 01 RA 00

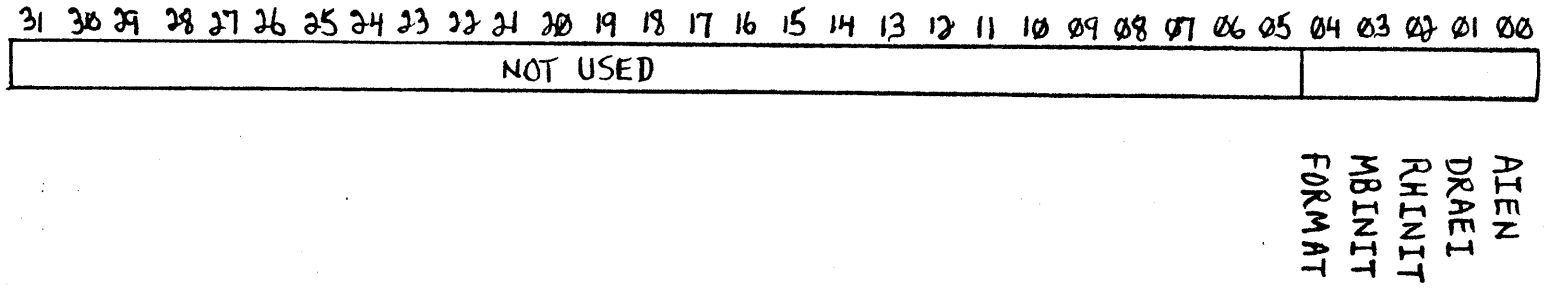
COMMAND TAG FORMAT

FIGURE 4.1.1



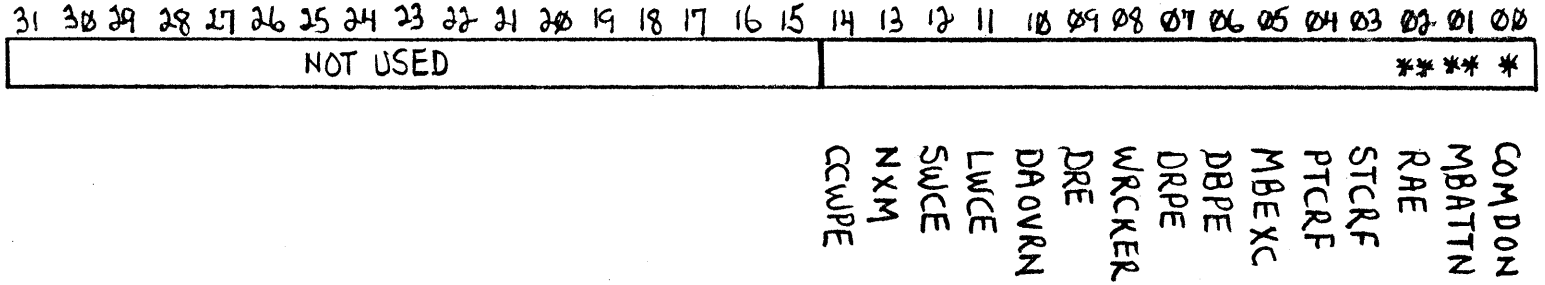
MARA 0000 : CONFIGURATION AND STATUS REGISTER (CSR)

FIGURE 4.2.1



MARA 0001: CONTROL REGISTER (CR)

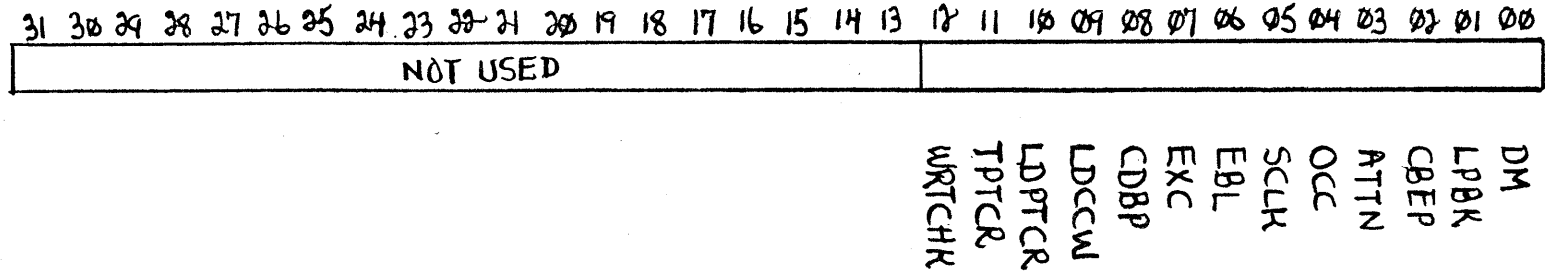
FIGURE 4.3.1



* UNCONDITIONAL INTERRUPT
** CONDITIONAL INTERRUPT

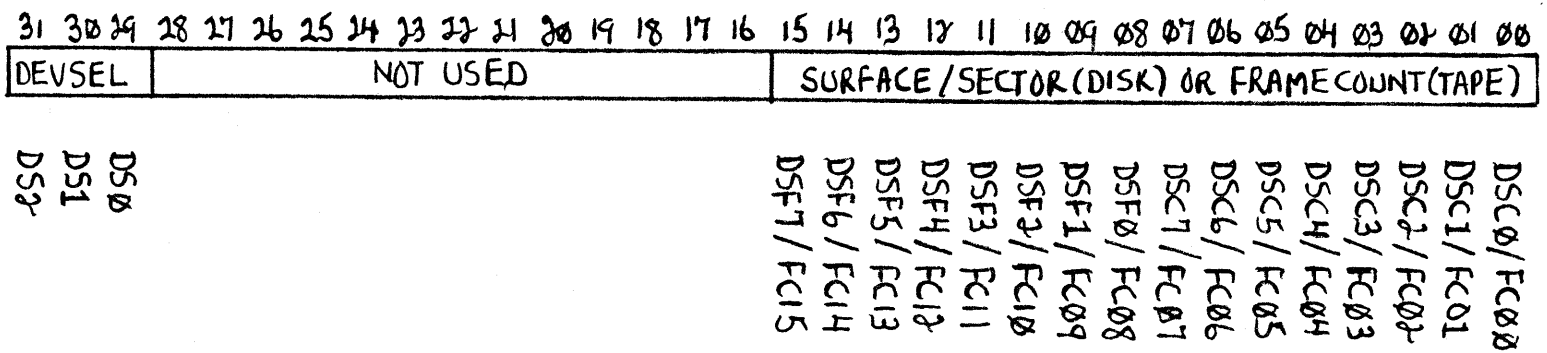
MARA 0002: STATUS REGISTER (SR)

FIGURE 4.4.1



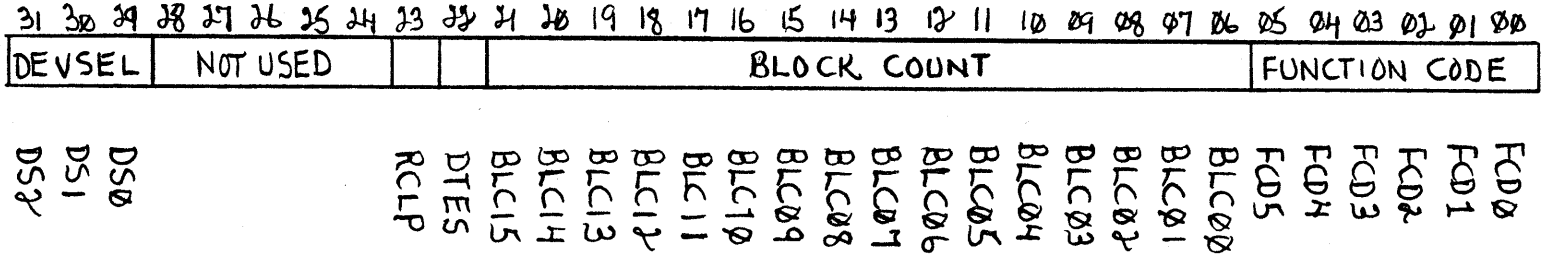
MARA 0003: DIAGNOSTIC REGISTER (DR)

FIGURE 4.5.1



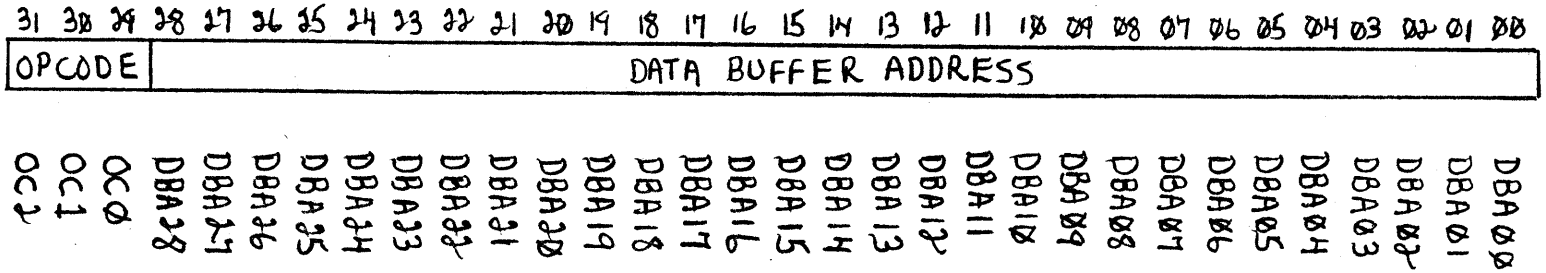
MARA 0004: SECONDARY BLOCK ADDRESS REGISTER (SBAR)
MARA 0006: PRIMARY BLOCK ADDRESS REGISTER (PBAR)

FIGURE 4.6.1



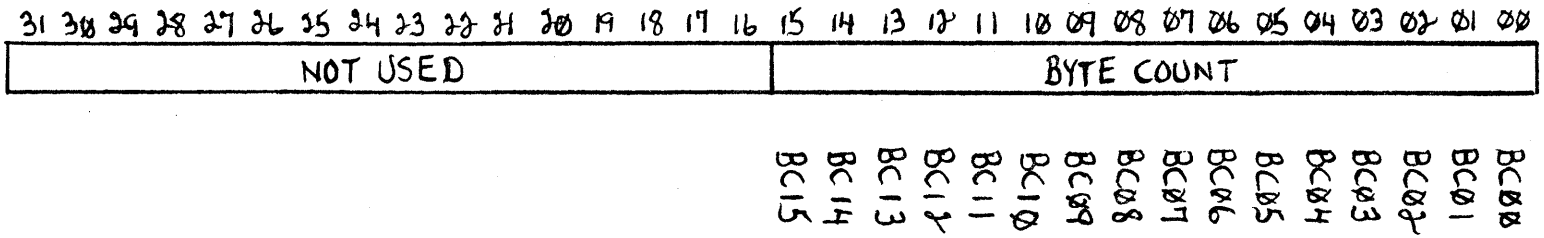
MARA 0005 : SECONDARY TRANSFER CONTROL REGISTER (STCR)
MARA 0007 : PRIMARY TRANSFER CONTROL REGISTER (PTCR)

FIGURE 4.7.1



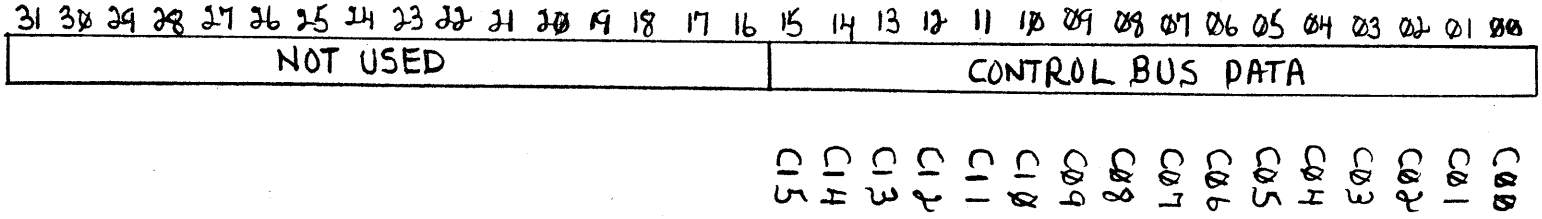
MARA 0010 : CURRENT CHANNEL COMMAND WORD 0 (CCCW0)

FIGURE 4.10.1



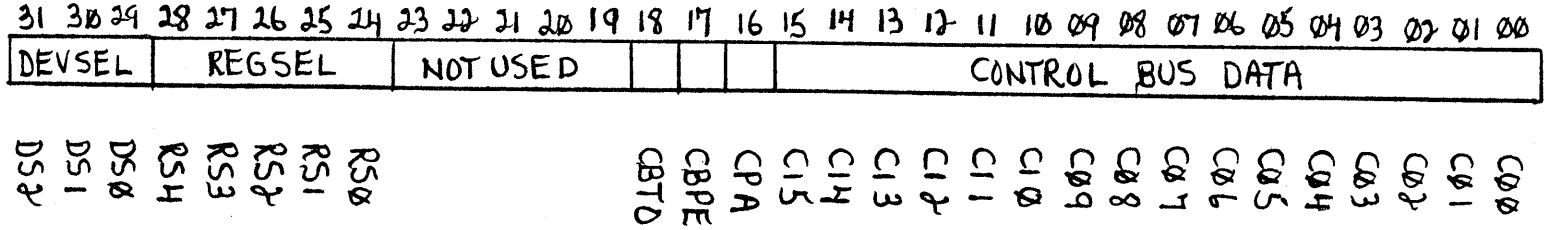
MARA 0011 : CURRENT CHANNEL COMMAND WORD 1 (CCCW1)

FIGURE 4.11.1



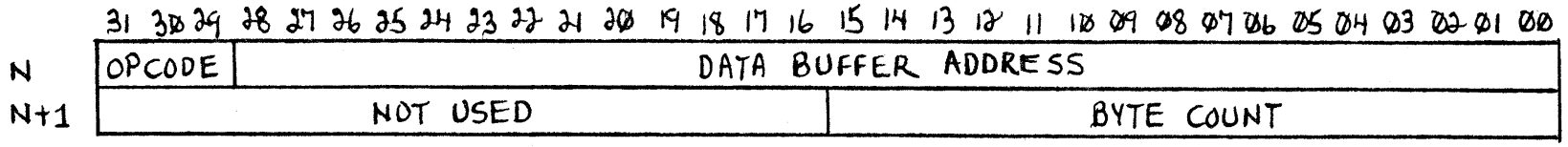
MARA 0400 - 0777 : CONTROL BUS TRANSMIT DATA REGISTER (CBTDR)

FIGURE 4.12.1



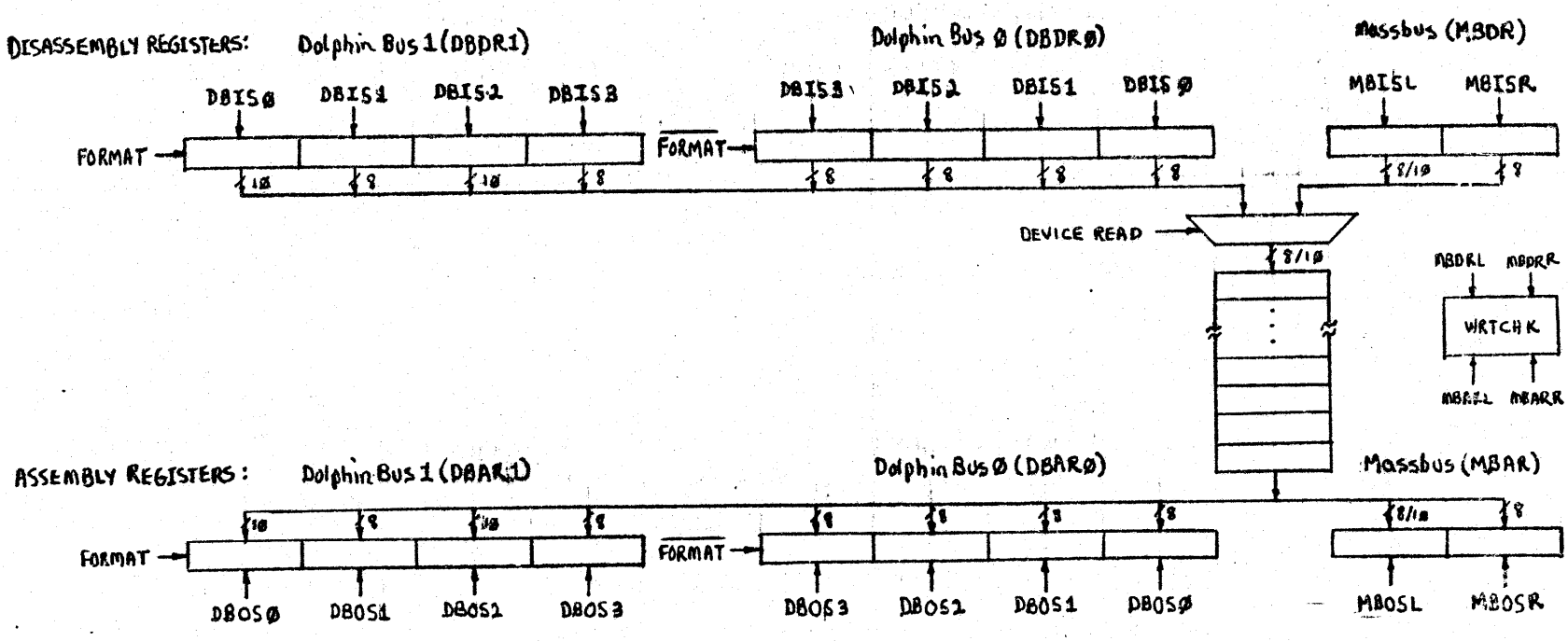
MARA 0400-0777 : CONTROL BUS RECEIVE DATA REGISTER (CBRDR)

FIGURE 4.13.1



MARA 1000 - 1377: CHANNEL COMMAND WORD REGISTERS 000 - 377 (CCW000 - CCW377)

FIGURE 4.14.1



DEFINITIONS:

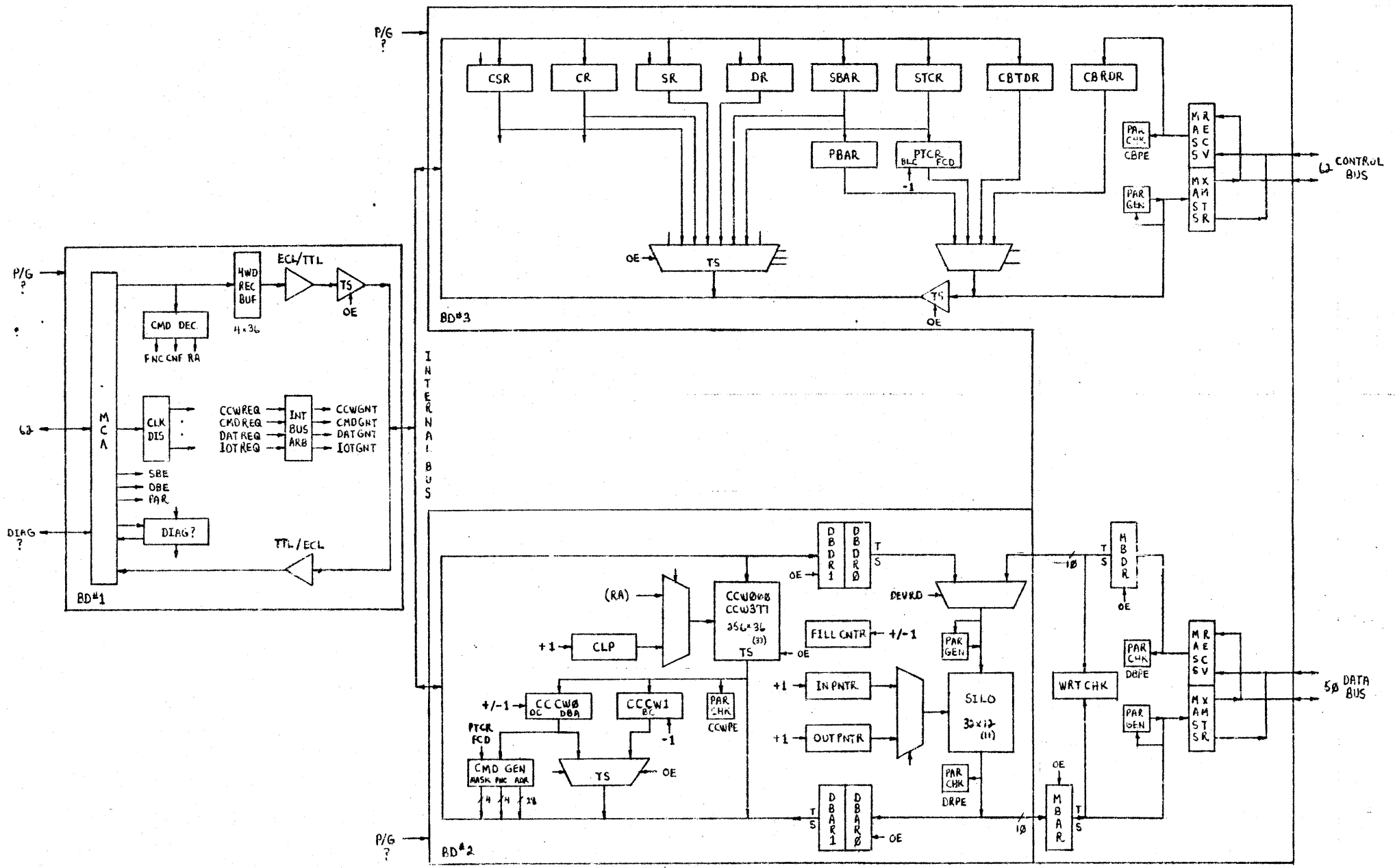
DBIS X = Dolphin Bus Input Select (Byte X)
MBIS X = Massbus Input Select (Left/Right)
DBOS X = Dolphin Bus Output Select (Byte X)
MBOS X = Massbus Output Select (Left/Right)

INPUT refers to the byte selected to transmit data to the silo.
OUTPUT refers to the byte selected to receive data from the silo.

FORMAT (partially) selects DBDR1 for device write and DBAR1 for device read.
$\overline{\text{FORMAT}}$ selects DBDRØ for device write and DBARØ for device read.

DBISX and DBOSX are a function of the byte offset.

Doesn't show data inputs to DRs or data outputs of ARs.



SIMPLIFIED BLOCK DIAGRAM

FIGURE 1.1.2