SLAC-TN-79-7
December 1979

THE BERMUDA TRIANGLE

A subsystem of the 168/E interfacing scheme
used by Group B at SLAC*

Gerard J. Oxoby, Lorne J. Levinson and Quang H. Trang
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

591

SYSTEM OVERVIEW

The Bermuda Triangle system is our method of interfacing several 168/E microprocessors to a central system for control of the processors and overlaying their memories. The system is a three way interface with I/O ports to a large buffer memory, a PDP11 Unibus, and a bus to the 168/E processors. Data may be transferred bidirectionally between any two ports. Two Bermuda Triangles are used, one for the program memory and one for the data memory. The program buffer memory stores the overlay programs for the 168/E and the data buffer memory the incoming raw data, the data portion of the overlays, and the outgoing processed events. This buffering is necessary since the 168/E microprocessors' memories are small compared to the main program and the amount of data being processed.

The link to the computer facility is via a Unibus to IBM channel interface. A PDP11/Ø4 controls the data flow.

This note assumes familiarity with the overview of the 168/E system given in [1]. Figure 1 is a block diagram of the system. The 168/E microprocessors are described in [2,3].

MODE OF OPERATIONS

Data coming from the IBM computer via the Unibus is first transferred to the buffer memory. When a 168/E is available, the PDP11 sets up a transfer from the buffer memory to the 168/E memory by setting the appropriate addresses and then loading the word counter. Upon exhaustion of the word count the PDP11 is interrupted. The PDP then loads the 168/E's program counter and starts the processor by loading its status.

Once the 168/E's program is executed, the PDP11 is interrupted by the completion interrupt module. It then initiates a transfer from the processor memory to the buffer memory and from the buffer memory to IBM via the Unibus.

* Work supported by the Department of Energy under contract DE-AC03-76SF00515.

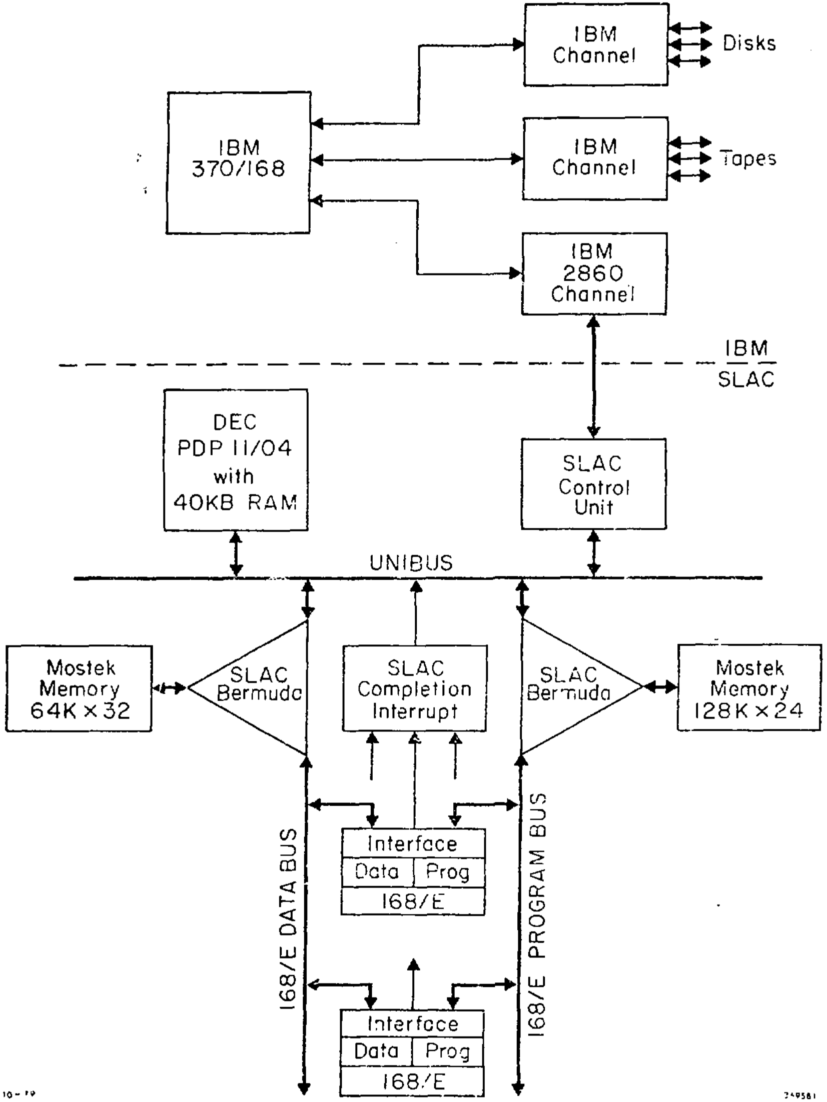


Figure 1: 168/E System Block Diagram

DESIGN CRITERIA

The design criteria were:

A three way interface between a Unibus, a large buffer memory and the 168/E interface bus.

All data paths to be bidirectional.

Only minor variations in the design between the 24 and 32 bit data path versions.

Restriction to a 5Ø line bus cable to the 168/E interface for reliability.

HARDWARE IMPLEMENTATION

-1- Buffer memories

The Buffer Memories were implemented using general purpose memory cards purchased from MOSTEK Memory Systems. The MK 8ØØØ card offers up to 128K 24 bit words with half word (12 bit) addressing capability. This feature is necessary in this application because of the Unibus 16 bit data path.

The program memory therefore is implemented with one card and stores 128K microinstructions. The data buffer memory is made of two cards, each depopulated to 16 bits. The memory cycle time is 5ØØ nsec with an access time of 375 nsec.

-2- Processor interface bus

The Processor Interface Bus was chosen to have a 24 bit wide address field in order to address all the 168/E's. The restriction of a 5Ø line cable required time multiplexing address and data. That is, the address is first presented and when it has been recognized the data transfer is started. The bus drivers are TTL Tristate. The transfer rate is one word per 7ØØ nsec.

The bus protocol is essentially identical to the one developed by the NIM FASTBUS Committee [4]. See Appendix A for the bus cable pin assignments.

-3- Interface

A universal interface board, MOSTEK MK85ØØ, was used. This card is the same size and has the same voltage pin assignments as the memory cards. To ease the mechanical Unibus interface another module was implemented (Bermuni). This module is plugged into a modified CAMAC crate whose backplane carries the Unibus signals. This "Unicrate", designed by SLAC Group G, is widely used by several experimental groups. See Appendix B for the Bermuda Triangle to Bermuni cable pin assignments.

-4- Chassis, power supply and backplane

The power supply and chassis are standard MOSTEK parts, MK8503. The backplane is a motherboard for a PDP11/70 compatible memory system sold by MOSTEK. There are 8 available connectors for MK8000 memory cards and one for an I/O board.

To accommodate the program and data subsystems all signal printed-circuit traces were cut between the fourth and fifth connectors (figure 2). Wires were wrapped to provide the 32 bit data path to the data interface. See Appendix C for backplane modifications and additions.

```
==========   ========== I/O connector       Not used


==========   ========== J1    Data Bermuda interface

==========   ========== J2                  Not used

==========   ========== J3    Data memory LO 16 bits

==========   ========== J4    Data memory HI 16 bits
-----------CUT----------
==========   ========== J5  Program memory expansion

==========   ========== J6            Program memory

==========   ========== J7                  Not used

==========   ========== J8 Program Bermuda interface
```

Figure 2: Backplane

SYSTEM DESIGN FEATURES

For clarity, byte addressing is used throughout the system.

Data transfers between the buffer memories and the Processor Interface Bus use direct memory access (DMA) to provide high throughput and minimum supervision by the PDP11.

-1- Unibus--Buffer Memory Transfers

Several schemes for addressing a large memory space from the Unibus were investigated. We needed to address up to 1/2 Mega-byte of data buffer memory and 1/2 Mega-byte of program buffer memory. This requires a 19 bit address for each, which exceeds the 16 bit address field of the PDP11.

The chosen scheme uses a window addressing method and an 8K byte range of Unibus address space. It is shown in figure 3:

```
        15        Page register        00
        ---------------------------------
        |e|x x x|p p p p p p p p p p p p|
        ---------------------------------
                   |
                   |        Unibus address lines
                   |17      13       08                00
                    -------------------------------------
                    |w w w w w|u u u u u|d d d d d d|h|0|
                    -------------------------------------
                   |               |           |
                   |               |           |
                   v               v           v
            ----------------------------
            |       Adder: A=p+u       |
            ----------------------------
                        |                      |
                        |                      |
                19      v        08            v       00
                ---------------------------------------
                |A A A A A A A A A A A A|d d d d d d|h|0|
                ---------------------------------------
                       20 bit Buffer Memory address
```

Figure 3: Buffer Memory Address Generation

The page register is loaded by the PDP11 via the Unibus. The most significant bit (15) determines if the buffer memory can be accessed. The 12 least significant bits, denoted p (in figure 3), are added to the Unibus address lines denoted u to generate the buffer memory most significant address bits. The Unibus bits denoted d and h are passed directly to make the least significant portion of the address field. The resulting 20 bit address is gated when the bits denoted w specify the 8K byte portion of Unibus address space occupied by the window. The bit denoted h selects the high or low half word of buffer memory.

The main advantage of this addressing structure is to allow placement of the 8K byte window anywhere in buffer memory, subject only to alignment on 256 byte multiples. It also permits any Unibus master device to initiate 8K byte block-transfers to and from the buffer memory. Since both windows occupy the same range of Unibus addresses, one has to be cautious: only one buffer memory may be enabled, by setting bit 15, at a time.

-2- Processor bus transfers

The Processor Interface Bus transfers data either between buffer memory and the processors, or between the Unibus and the procesors. Single word mode is used for transfers to or from the Unibus and block mode for transfers to or from the buffer memory.

During a DMA operation between the Processor Interface Bus and the buffer memory, the buffer memory address is provided by the Buffer Memory Address Register. This register is initially loaded by the PDP11. The Processor Interface Bus address is provided from a register loaded by the PDP11 and is incremented after each data transfer.

On the Processor Interface Bus the address and data are multiplexed as mentioned earlier under Hardware Implementation. Figures 4 and 5 are timing diagrams for a word transfer and a block transfer respectively.

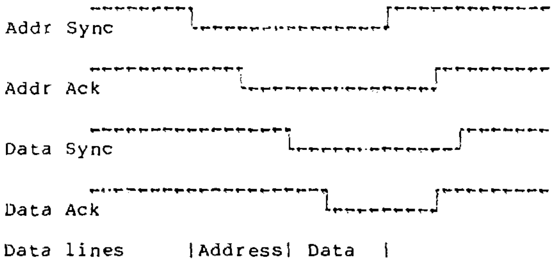


Figure 4: Processor Interface Bus: Single Word Transfer

From figure 5, we see that when a block transfer is taking place it should in no case be interrupted by a different source since the address information would be lost. For this reason any Unibus access to the Processor Interface Bus is blocked during a DMA operation and results in a Unibus time out.

The Unibus access to a processor can be either through: 1) one word in PDP11 I/O space, or 2) an 8K byte window. The latter provides direct transfer of data between the 168/E memory and a Unibus master device in up to 8K byte blocks.

Half words cannot be transferred on the Processor Interface Bus. Thus for transferring to or from the Unibus, holding registers were implemented.

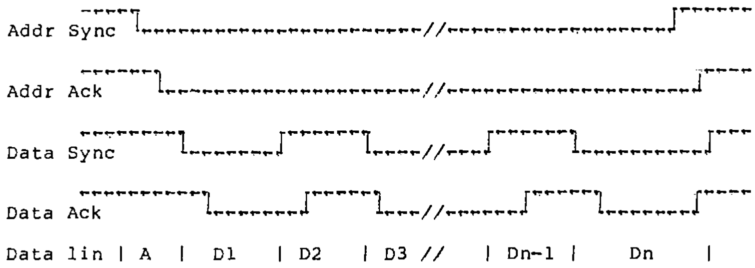


Figure 5: Processor Interface Bus: Block Transfer

-3- Buffer memory access

As seen previously the buffer memories can be accessed from the Unibus and the Processor Interface Bus. In the case of an access via the Unibus while a DMA with the Processor Interface Bus is in progress some interleaving has to be provided. This situation occurs when an asynchronous device such as a control unit becomes Unibus master. In this case, all the Processor Interface Bus control and data lines are held fixed after the current data word has been transferred.

-4- Unibus interface, Bermuni

This is a very basic Unibus interface containing the necessary receivers and drivers for the two ports, to the Unibus and to the triangle interface. It handles the bulk address decoding (A17-AØ4) and the PDP11 interrupt logic.

-5- Completion interrupt module (Ky Mui)

This Unicrate module generates an interrupt to the PDP11 when a signal is applied to any one of the 16 front panel Lemo connectors. Sixteen different vectors are presented according to the active input. They are consecutive, starting on a jumper selectable address of 1ØØ, 2ØØ or 3ØØ.

A latch for each input and a priority encoder permit successive servicing of interrupt requests.

-6- Design

Figure 6 is a block diagram of the Bermuda Triangle. The number within each block indicates which logic sheet to refer to. The SLAC drawing numbers are: LD 735-1ØØ-5Ø for the Bermuni (3 sheets), LD 735-1ØØ-51 for the Completion Interrupt Module (1 sheet), and LD 735-1ØØ-52 for the Bermuda Triangle (1Ø sheets).

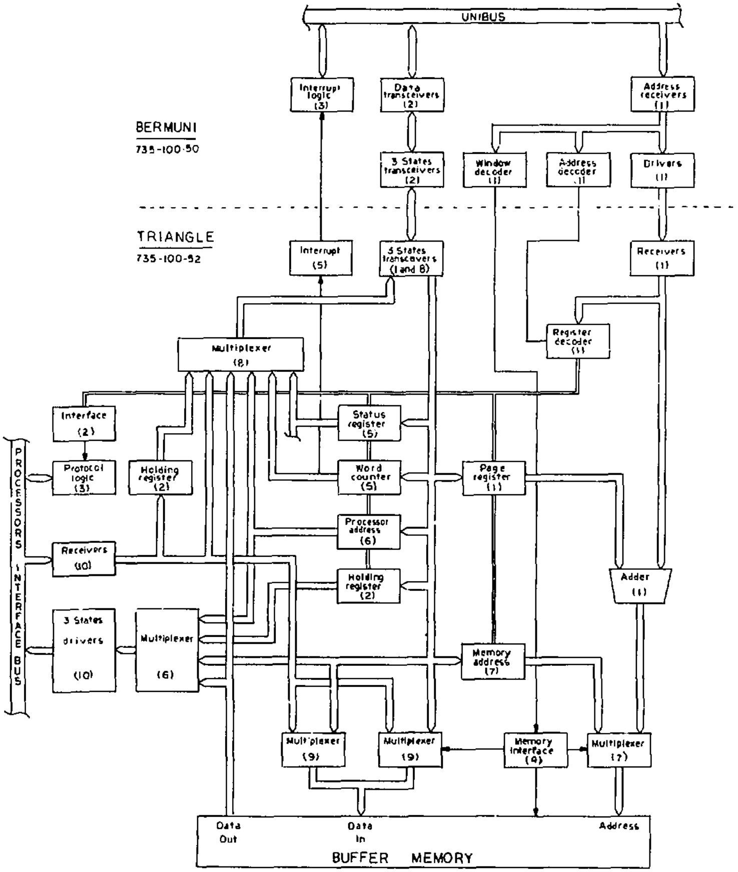


Figure 6: Bermuda Triangle Block Diagram

REGISTERS

The least significant address bits on the Unibus (AØ3-AØ1) are decoded to select one of 8 I/O addresses when the bulk decoding done by Bermuni indicates a selection.

The address range is from 16XXØØ to 16XX16 where X is jumper selectable.

The eight registers selected by the least significant bits are as follows:

ØØ	Control and Status Register
Ø2	Buffer memory address 8 MSB, R/O
Ø4	Buffer memory address 16 LSB, R/O
Ø6	Processor Interface Bus address 8 MSB
1Ø	Processor Interface Bus address 16 LSB
12	Word counter
14	Processor Interface Bus data
16	Page register, R/O

-1- Control and Status Register (BTDCSR and BTPCSR)

Figure 7 shows the function of each bit.

```
--------------------------------------------------------------
|15|                      |Ø7|Ø6|Ø5|   |Ø3|Ø2|Ø1|ØØ|
--------------------------------------------------------------
  |                         |  |  |      |  |  |  |
  |                         |  |  |      |  |  |  |
   --- Error                |  |  |      |  |  |  |
                            |  |  |      |  |  |  |
        Interrupt Pending --   |  |      |  |  |  |
                               |  |      |  |  |  |
        Interrupt enable R/W --   |      |  |  |  |
                                  |      |  |  |  |
        Window enable -------------      |  |  |  |
                                         |  |  |  |
        Time out -------------------------  |  |  |
                                            |  |  |
        Software error ----------------------  |  |
                                               |  |
        Window to processor R/W ----------------  |
                                                  |
        Transfer to buffer memory R/W -------------
```

Figure 7: CSR Bit Assignments

Bit 15 (BTERR): The OR of bits 02 and 03.

Bit 07 (BTDONE): Set by completion of a DMA operation or an error, ie. bit 15 set. Reset by Unibus Init, servicing the interrupt, or loading the word counter.

Bit 06 (BTINT): When set, by the PDP11, enables bit 07 to generate an interrupt.

Bit 05 (BTWEN): Reflects the most significant bit of the page register.

Bit 03 (BTTIMO): This bit is set when attempting to access the Processor Interface Bus and no response is received. It is also set if Address Acknowledge is dropped before the end of a transfer. Reset by Unibus Init or accessing the word counter.

Bit 02 (BTSERR): Set whenever an access from the Unibus to the Processor Interface Bus is attempted while a DMA is in progress. The DMA operation is not affected. Reset by Unibus Init or accessing the word counter.

Bit 01 (BTFBWX): When set, enables data transfers between the Unibus and the Processor Interface Bus via any address within an 8K byte address window. Since the same 8K address range is used to access the buffer memory, the most significant bit of both page registers must be zeroed. Reset by Unibus Init.

Bit 00 (BT2BM): Selects the direction of the transfers between the buffer memory and the Processor Interface Bus. When set, the data flow is from the processor to the memory.

-2- Buffer memory address (BTDBMA and BTPBMA)

This register holds the buffer memory address for transfers to or from the Processor Interface Bus. It is 24 bits wide and occupies two Unibus words. The most significant byte is loaded from the least significant data byte of address 16XX02. The 16 least significant bits are loaded via the Unibus address 16XX04.

It is incremented by 4 after each memory cycle caused by the DMA operation. The two least significant bits are always reset. Always read as 177777 by the PDP11.

-3- Processor Interface Bus address (BTDFBA and BTPFBA)

Similar in format to the buffer memory address, this register occupies Unibus addresses 16XX06 and 16XX04. The least significant 20 bits are incremented by 4 after each bus cycle. The 4 most significant bits are a specific 168/E address and are therefore not incremented.

-4- Word counter (BTDWC and BTPWC)

The two's complement of the number of 32 bit words to be transferred, by DMA, between the buffer memory and a processor is loaded in this register. It is incremented after each transfer.

Overflow of the counter sets bit 7 of the status register.

The current count can be read by the PDP11. Examining or loading it also resets the error bits of the status register

-5- Processor data (BTDDAT and BTPDAT)

Access to the Processor Interface Bus can also be via the Unibus address 16XX14. Two successive references are necessary to send or receive 32 bits of information. The Processor Interface Bus cycle is initiated on the first read or on the second write to this address. The Processor Interface Bus address is specified by the Processor Interface Bus Address register.

-6- Page register (BTDPAG and BTPPAG)

We have seen the use of this register in the section "Buffer Memory Addressing" above. Bit 15 enables access to the buffer memory from the Unibus and can be read back as bit Ø5 of the status register. This register is always read as 177777 by the PDP11.

The page register is not reset by Unibus Init. Therefore, if the Bermuda Triangle is used with an operating system, such as RSX11, which sizes the available PDP11 memory upon initiation, the enable bit should be moved to the status register. See sheet 5 of the logic diagrams.

REFERENCES

1. Paul F. Kunz et. al., Experience Using the 168/E Microprocessor for Offline Data Analysis, Proceedings of the 1979 IEEE Nuclear Science Symposium (Feb 1980), also SLAC-PUB-2418 (Oct 1979)

2. Paul F. Kunz, The LASS Hardware Processor, Nuc. Instr. Meth. 9 (1976) p. 435.

3. Paul F. Kunz et. al., The LASS Hardware Processor, Proc. 11th Annual Microprogramming Workshop, SIGMICRO Newsletter 9 (1978) p. 25.

4. B. Wadsworth, FASTBUS - An Emerging Laboratory Standard, Proceedings of the 1979 IEEE Nuclear Science Symposium (Feb 1980),

APPENDIX A: BERMUNI TO BERMUDA CABLE SIGNALS

1 - Ground	26- UD14
2 - USSYN	27- UD13
3 - UINTERRUPT	28- UD12
4 - Ground	29- Ground
5 - UAØ1	3Ø- UD11
6 - UAØ2	31- UD1Ø
7 - UAØ3	32- UDØ9
8 - Ground	33- UDØ8
9 - UAØ4	34- Ground
1Ø- UAØ5	35- UDØ7
11- UAØ6	36- UDØ6
12- UAØ7	37- UDØ5
13- Ground	38- UDØ4
14- UAØ8	39- Ground
15- UAØ9	4Ø- UDØ3
16- UA1Ø	41- UDØ2
17- UA11	42- UDØ1
18- UA12	43- UDØØ
19- Ground	44- Ground
2Ø- UCØ	45- URSERVICED
21- UREAD	46- SELECT REG.
22- Ground	47- Ground
23- URESET	48- PAGE DECODE
24- Ground	49- UMSYN
25- UD15	5Ø- Ground

APPENDIX B: PROCESSOR INTERFACE BUS SIGNALS

1 - Ground	26- Ground
2 - Data Sync.	27- D15
3 - Ground	28- D14
4 - Address Sync.	29- D13
5 - Ground	3Ø- D12
6 - Read cmd.	31- Ground
7 - D31	32- D11
8 - D3Ø	33- D1Ø
9 - D29	34- DØ9
1Ø- D28	35- DØ8
11- Ground	36- Ground
12- D27	37- DØ7
13- D26	38- DØ6
14- D25	39- DØ5
15- D24	4Ø- DØ4
16- Ground	41- Ground
17- D23	42- DØ3
18- D22	43- DØ2
19- D21	44- DØ1
2Ø- D2Ø	45- DØØ
21- Ground	46- Ground
22- D19	47- Address Ack.
23- D18	48- Ground
24- D17	49- Data Ack.
25- D16	5Ø- Ground

APPENDIX C: BACKPLANE WIRING FOR DATA VERSION OF BERMUDA

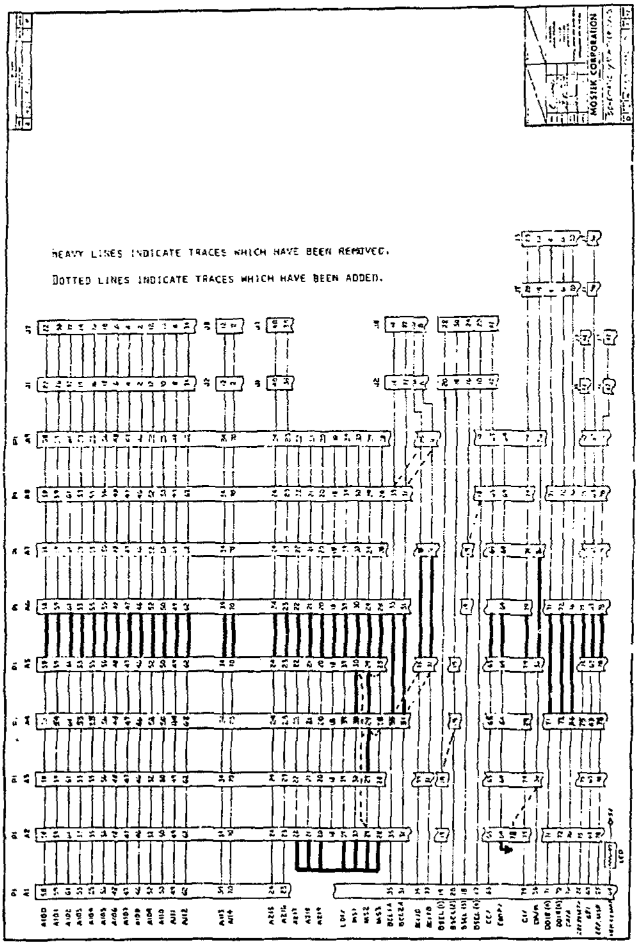
Connect the following:

	Data out	Data in
Bit 00	A2 P2-49 to A4 P2-49	A2 P2-43 to A4 P2-43
Bit 01	A2 P2-48 to A4 P2-48	A2 P2-42 to A4 P2-42
Bit 16	A2 P2-53 to A5 P2-49	A2 P2-52 to A5 P2-43
Bit 17	A2 P2-51 to A5 P2-48	A2 P2-45 to A5 P2-42
Bit 18	A2 P1-71 to A5 P2-47	A2 P1-68 to A5 P2-11
Bit 19	A2 P1-72 to A5 P2-46	A2 P1-69 to A5 P2-10
Bit 20	A2 P2-26 to A5 P2-09	A2 P2-33 to A5 P2-05
Bit 21	A2 P2-28 to A5 P2-08	A2 P2-34 to A5 P2-04
Bit 22	A2 P2-23 to A5 P2-06	A2 P2-22 to A5 P2-02
Bit 23	A2 P2-21 to A5 P2-07	A2 P2-20 to A5 P2-03
Bit 24	A2 P1-18 to A5 P2-50	A2 P1-17 to A5 P2-44
Bit 25	A2 P2-29 to A5 P2-73	A2 P2-32 to A5 P2-74
Bit 26	A2 P2-27 to A5 P2-72	A2 P2-31 to A5 P2-71
Bit 27	A2 P2-25 to A5 P2-69	A2 P2-30 to A5 P2-70
Bit 28	A2 P2-24 to A5 P2-65	A2 P2-18 to A5 P2-62
Bit 29	A2 P2-15 to A5 P2-59	A2 P2-16 to A5 P2-61
Bit 30	A2 P2-14 to A5 P2-58	A2 P2-12 to A5 P2-57
Bit 31	A2 P1-25 to A5 P2-54	A2 P1-26 to A5 P2-56

BUFFER MEMORY INTERFACE
CONNECTOR PIN LIST

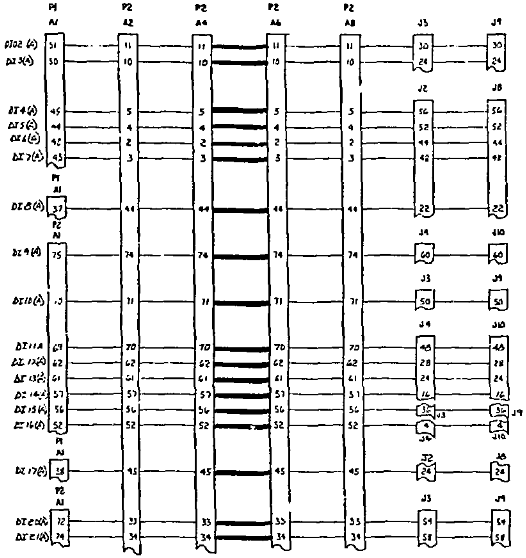
PIN	CONNECTOR		PIN	CONNECTOR	
NUMBER	J1	J2	NUMBER	J1	J2
01	OV	OV	41	OV	OV
02	+5V	DI06	42	+5V	DI01
03		DI07	43		DI00
04	OV	DI05	44	OV	DI08
05	+12V	DI04	45	+12V	DI17
06		DO06	46	AI08	DO03
07		DO07	47	AI07	DO02
08		DO05	48	AI06	DO01
09		DO04	49	AI11	DO00
10		DI03	50	AI10	DO08
11		DI02	51	OV	DO17
12		(DI30)	52	AI09	DI16
13		DOEN/	53	AI03	DO16
14		(DO30)	54	OV	DO15
15		(DO29)	55	AI04	OV
16		(DI29)	56	AI05	DI15
17	(DI24)	MCLK/	57	OV	DI14
18	(DO24)	(DI28)	58	AI00	DO14
19	BSEL	XREN/	59	AI01	DO13
20	AI19	DI23	60	OV	OV
21	AI18	DO23	61	AI02	DI13
22	AI17	DI22	62	AI12	DI12
23	AI16	DO22	63	GR/	DO12
24	AI15	(DO28)	64	COOP/	OV
25	(DO31)	(DO27)	65	CC/	OV
26	(DI31)	DO20	66	OV	XCLK/
27	SC/	(DO26)	67	OV	OV
28	MS3	DO21	68	DI18	OV
29	MS2	(DO25)	69	DI19	DO11
30	MS1	(DI27)	70	AI14	DI11
31	BCL2	(DI26)	71	DO18	DI10
32	ADOP	(DI25)	72	DO19	DO10
33	MB/	DI20	73	XA1	DO09
34	AI13	DI21	74	XA1/	DI09
35	BCL1	OV	75	XA2	
36	DA/	+12V	76	XA2/	+12V
37	TIOP/	OV	77	XA3/	OV
38			78	XA3	
39	LDI/	+5V	79	CI/	+5V
40	OV	OV	80	OV	OV

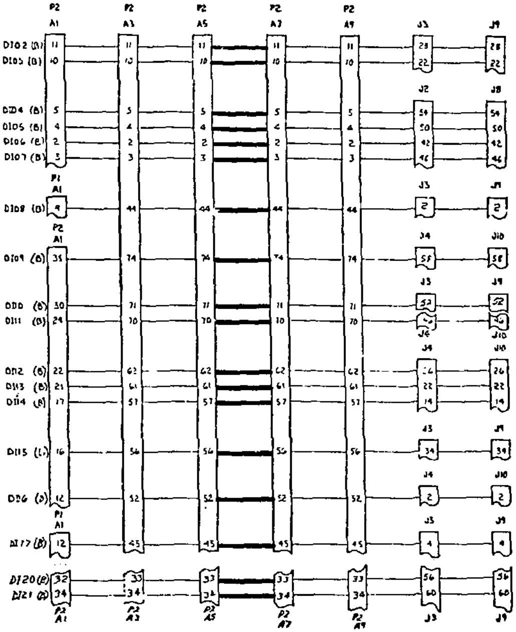
( ) indicates Data side only

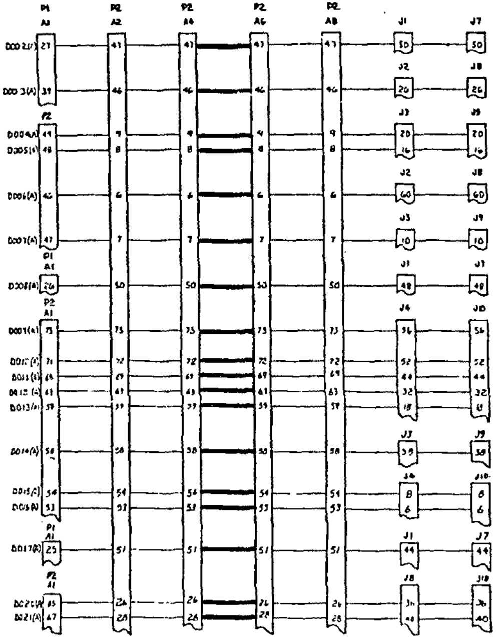


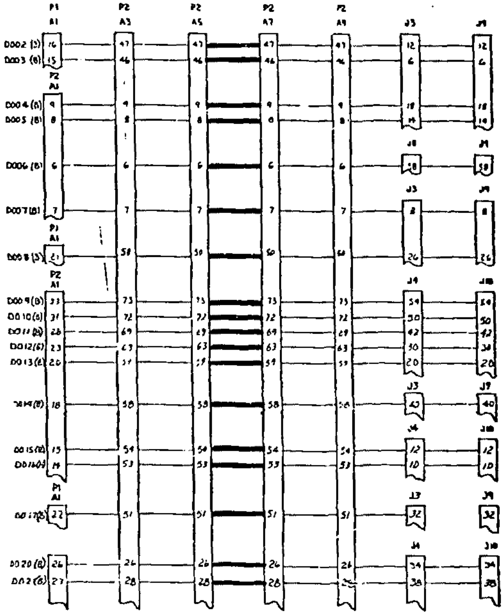
HEAVY LINES INDICATE TRACES WHICH HAVE BEEN REMOVED.

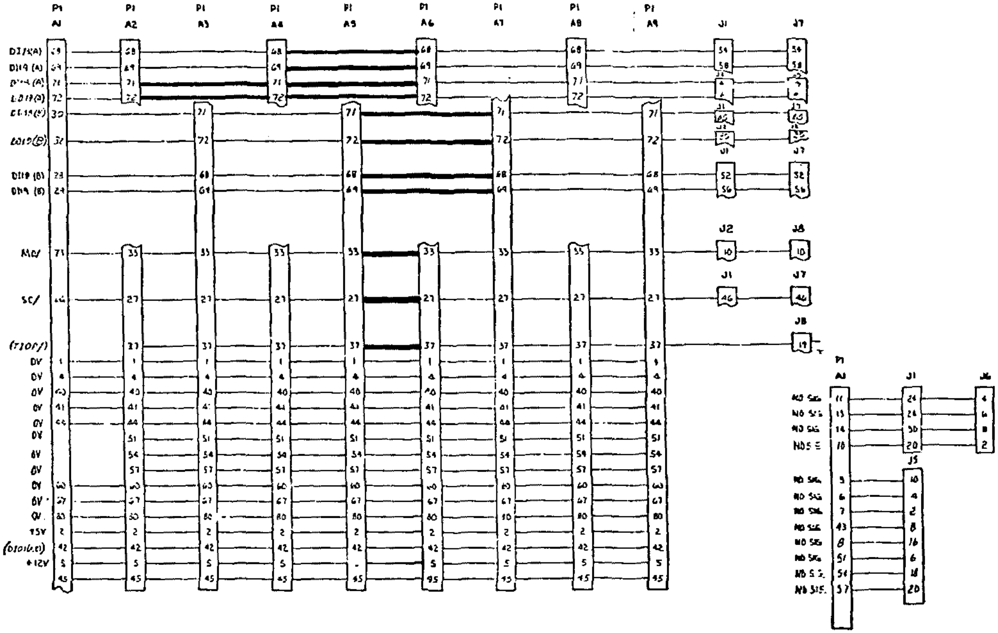
DOTTED LINES INDICATE TRACES WHICH HAVE BEEN ADDED.

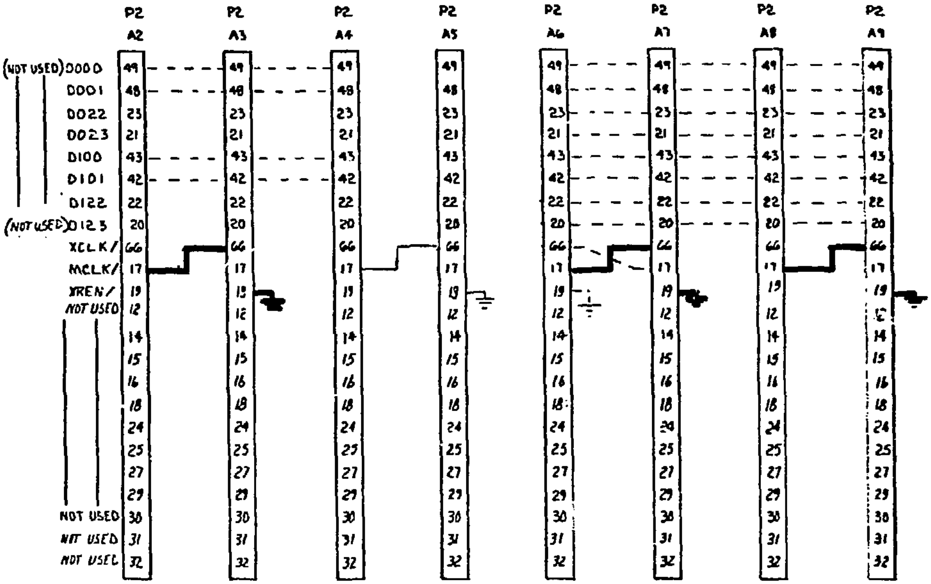

P1 A1
P2 A2
P2 A4
P2 A6
P2 A8
J3
J9
J2
J8
J4
J10
DI02 (A)
DI3(A)
DI4(A)
DI5(A)
DI6(A)
DI7(A)
DI8(A)
DI9(A)
DI10(A)
DI11A
DI12(A)
DI13(A)
DI14(A)
DI15(A)
DI16(A)
DI17(A)
DI20(A)
DI21(A)

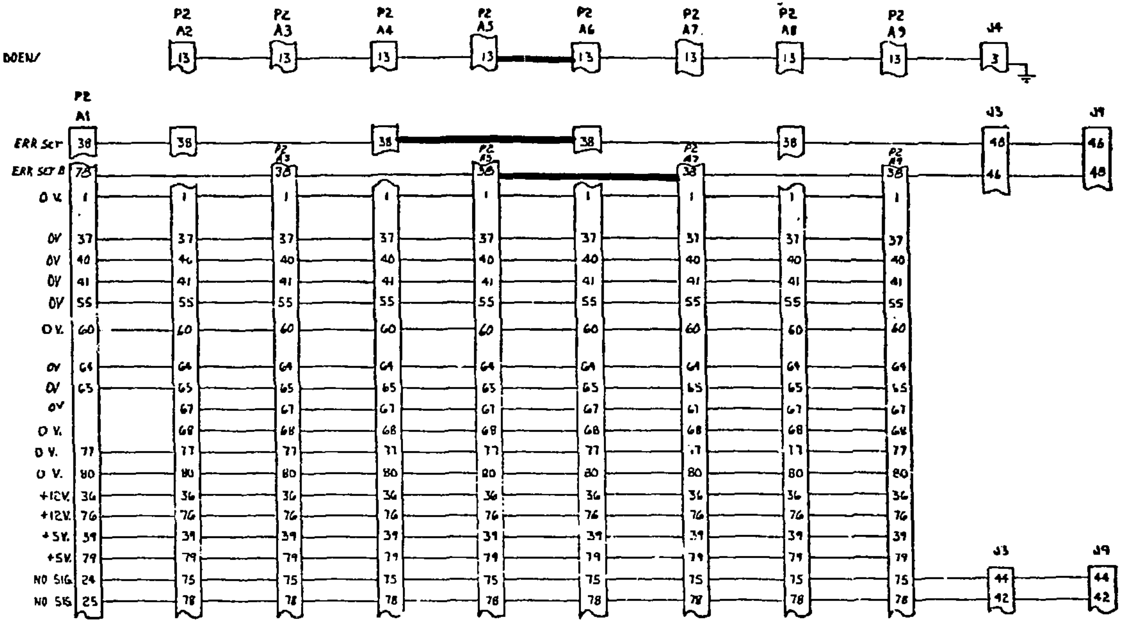

P2 A1
P2 A3
P2 A5
P2 A7
P2 A9
J3
J9
J2
J8
J4
J10
DI02 (B)
DI03 (B)
DI04 (B)
DI05 (B)
DI06 (B)
DI07 (B)
DI08 (B)
DI09 (B)
DI10 (B)
DI11 (B)
DI12 (B)
DI13 (B)
DI14 (B)
DI15 (B)
DI16 (B)
DI17 (B)
DI20(B)
DI21 (B)


P1
A1
P2
A2
P2
A4
P2
A6
P2
A8
J1
J7
J2
J8
J3
J9
J4
J10
DO02(A)
DO03(A)
DO04(A)
DO05(A)
DO06(A)
DO07(A)
DO08(A)
DO09(A)
DO10(A)
DO11(A)
DO12(A)
DO13(A)
DO14(A)
DO15(A)
DO16(A)
DO17(A)
DO20(A)
DO21(A)


P1
A1
P2
A3
P2
A5
P2
A7
P2
A9
J3
J9
J8
J4
J10
DO02(B)
DO03(B)
DO04(B)
DO05(B)
DO06(B)
DO07(B)
DO08(B)
DO09(B)
DO10(B)
DO11(B)
DO12(B)
DO13(B)
DO14(B)
DO15(B)
DO16(B)
DO17(B)
DO20(B)
DO21(B)


P1
A1
P1
A2
P1
A3
P1
A4
P1
A5
P1
A6
P1
A7
P1
A8
P1
A9
J1
J7
DI14(A)
DI19 (A)
DI14 (A)
LDI19(A)
LDI14(B)
LDI19(B)
DI19 (B)
DI14 (B)
J2
J8
MD/
J1
J7
SC/
J8
(TIOF/)
0V
0V
0V
0V
0V
0V
0V
0V
0V
0V
0V
+5V
(DIOIL(0))
+12V
P1
A1
J1
J6
NO SIG
NO SIG
NO SIG
NO S E
J5
NO SIG
NO SIG
NO SIG
NO SIG
NO SIG
NO SIG
NO S S.
NO SIG.


P2 A2
P2 A3
P2 A4
P2 A5
P2 A6
P2 A7
P2 A8
P2 A9
(NOT USED) DO00
DO01
DO22
DO23
DI00
DI01
DI22
(NOT USED) DI23
XCLK/
MCLK/
XREN/
NOT USED
NOT USED
NOT USED
NOT USED
49
48
23
21
43
42
22
20
66
17
19
12
14
15
16
18
24
25
27
29
30
31
32


P2 A1
P2 A2
P2 A3
P2 A4
P2 A5
P2 A6
P2 A7
P2 A8
P2 A9
J4
J3
J9
DOEN/
ERR SET
ERR SET B
0 V.
0V
0V
0V
0V
0 V.
0V
0V
0V
0 V.
0 V.
0 V.
+12V.
+12V.
+5V.
+5V.
NO SIG.
NO SIG.
13
3
38
78
1
37
40
41
55
60
64
65
67
68
77
80
36
76
39
79
24
25
75
78
46
48
44
42