BBN Communications

Pluribus Hardware Configuration Maintenance Manual

BBNCC Task Order #83-014

Revision B

Date: September, 1983

Submitted To:

Maryland Procurement Office
FT. Meade, MD
Contract No. MDA 904-83-C-0869

Prepared By:

BBN Communications Corporation
33 Moulton Street
Cambridge, MA 02238

Pluribus Board Configuration Manual
Version Number: B
September 1983

Printed in the United States of America.

Text by Pete Marti (1983), drawing upon Pluribus Document 7: Maintenance; Report No. 3004 (1976).

Edited by Eugene Chang and Shirley Taggart (1983).

First printing May, 1983.

Second printing September, 1983

BBN Communications Corporation, 33 Moulton Street, Cambridge, Mass. 02238

Table of Contents

Table of Contents (Cont.)

1 Introduction

1.1 Purpose

This manual describes Pluribus configuration aspects, and provides the necessary documentation to verify and/or to change Pluribus system configuration. In order to effectively utilize this manual, technicians require the System Release Notice (SRN) that is shipped with every BBN Computer Pluribus system.

In order to provide technicians with a useful and comprehensive document, the information in each chapter is divided into sections and subsections. Sections refer to board type, while subsections refer to a specific board. Sections on each board's "function in life" provide additional in-depth board information.

The following pertinent information appears at the end of each subsection:

1. A. A pictorial of the board (half-intensity)

   B. Relevant strappable fields (full-intensity).

   C. A coordinate system indexing fields and boards.

2. Exploded view of the strappable fields or representations of the switches.

3. Charts for strapping the field and the most probable switch setting.

4. A sample SRN for each board.

The manual contains the following items to increase its effectiveness and systemize its contents:

1. A Table of Contents
2. Page and Section Numbers
3. Page and Section Labels
4. Pictorial, Field/Charts, and Typical Example are connected via superscripts.
5. Appendices at the end of the manual to give supportive information for configuring, maintaining, and repairing Pluribus systems.

1.2 Description

The Pluribus has a modular design. Depending on system configuration, many boards have fields that require an initial adjustment. Since identical boards are strapped in different ways depending on their location, board information is particularly important.

A technician straps a board by adding or deleting wires between wire-wrap pins or by changing the postions (ON/OFF) of the switches.

The SRN is a series of configuration patterns (layouts) used by manufacturing to build a system, and by Field Service to verify the present configuration. The SRN, reflecting the current machine configuration, is invaluable to the technicians that repair and maintain the systems. Without this information, a technician may incorrectly interpret the diagnostics and, as a result, troubleshoot the wrong portion of the machine. In addition, misconfiguration may cause other intermittent problems.

Since the SRN provides a detailed and comprehensive diagram of a product, and as such, is the product's most valuable companion, a technician must ensure that a particular SRN and machine match. An up-to-date and correct SRN is the technician's first step toward solving a Pluribus problem.

2 CONTROL PANEL

2.1 PBI-PCB

PBI - 2102301G01
PCB - 2102298G01

2.1.1 Purpose

The control panel (or console) is an addressable system module that provides the operator with a method to interface with the Infibus, and therefore the system. As such, the control panel can communicate with any other addressable system or peripheral device.

From the control panel, an operator has input and output cabalities to the system, and can perform the following functions:

A. Analyze system status (Control panel operating in slave mode)

B. Communicate with the following system modules (Control panel operating in master mode):

1. processors
2. memories
3. any I/O device

2.1.2 Description

The control panel module consists of the switch panel, PCB card, PBI card, a small ribbon cable to interconnect the PCB and PBI cards, and 3 ribbon cables to connect the switch panel to the PCB and PBI cards.

The control panel uses +5 volts DC power. The control panel has 63 touch-response switches and indicators to regulate system interaction and visual display during program development and maintenance.

NOTE
The touch-response switches require

a slight amount of pressure
DO NOT overdo it.

The switch is located behind the markings and below the neon indicator associated with the switch.

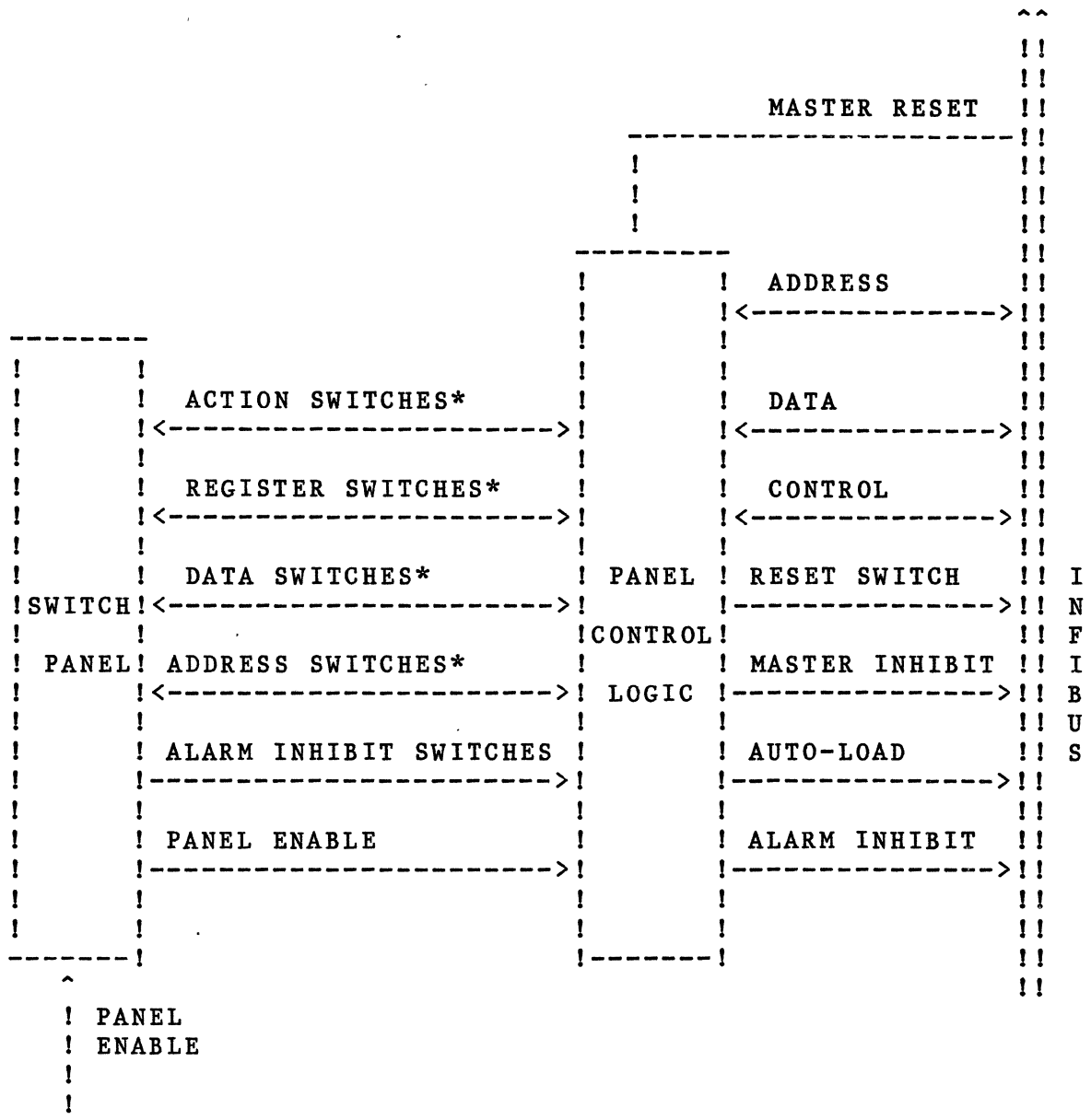
CAUTION

Only the operator's fingers should actuate the switches.

DO NOT use pencils, tools and other sharp objects.

NOTE

A touch-response function can be effected only when the panel is enabled.



* All switches have a corresponding indicator

2.1.3 Switches

RESET SWITCH

The switch marked **RESET** is located in the lower lett-hand corner of the switch panel. When touched, the **RESET** indicator lights, and the bus computer system is reset. During the reset the **HALT** indicator flashes once for about .1 second.

As a result of resetting the bus, the **LOAD** switch indicator, the **IDLE** indicator, and the **REGISTER 0** indicator remain lit until a new command is given. (NOTE: These indicators remain lit even though the **RESET** switch is no longer being touched.)

AUTO-LOAD SWITCH

The auto-load switch, marked **LOAD**, is located to the right of **RESET**.

To enable the bootstrap loader press **RESET**. The **LOAD** indicator lights, indicating the bootstrap loader is enabled.

Whenever AC power is applied, the **LOAD** switch is automatically enabled and lit. The lit **LOAD** indicator is extinguished when the panel is disabled, but its indicator is lit again, as soon as the panel is enabled.

INHIBIT SWITCH

The inhibit switch, marked **INH**, is the second switch from the right in the bottom row and, when activated, inhibits all system interrupts on priority levels 1 through 4. The **INH** switch is similar to a toggle switch. Therefore, if the first touch lights (turns on) the indicator, the second touch darkens (turns off) the indicator and vice versa. The **INH** indicator, when lit, indicates that all system interrupts are inhibited.

REGISTER SWITCH

The top row contains sixteen switches and indicators marked 0 through 15. The word **REGISTER.** appears directly above the switches. The **REGISTER** switches select any one of sixteen possible system processor registers (Refer to table 1). The selected register can be accessed by pushing the **READ** or **WRITE** switch located in the top row.

The switch indicator most recently touched is lit, while the previously lit indicator is darkened. The number on the lighted switch indicator is the selected register number.

Following a system reset or after AC power is turned on, **REGISTER 0** is always selected.

Table 1. Register Addresses

Register Switch	Register Address (Hex)	Register Name
0	FFx0	General Register 0 (PC)
1	FFx2	General Register 1
2	FFx4	General Register 2
3	FFx6	General Register 3
4	FFx8	General Register 4
5	FFxA	General Register 5
6	FFxC	General Register 6
7	FFxE	General Register 7
8	FFy0	Status Register
9	FFy2	Instruction Register
10	FFy4	Firmware A (address of last instruction)
11	FFy6	Firmware B
12	FFy8	Not used
13	FFyA	Not used
14	FFyC	Not used
15	FFyE	Halt/Run Control

x = 0, 2, 4, 6 for processors 0, 1, 2, 3 respectively.
y = 1, 3, 5, 7 for processors 0, 1, 2, 3 respectively.

ADDRESS SWITCHES

Sixteen **ADDRESS** bit switches marked 0 through 15 are located in the second row from the top, below the **REGISTER** switches. The word **ADDRESS** appears directly above the switches. The coded address switches select the address and when touched, the second row **READ** or **WRITE** switch acesses this address.

Lighted, the **ADDRESS** switch indicators represent binary ONES; extinguished, they represent binary ZEROS. Each switch has toggle-switch action. Therefore, if the first touch lights (turns on) the indicator, the second touch darkens (turns off) the indicator, and vice versa.

After touching **RESET** or after AC power is turned on, the address switch indicators are always cleared to hexadecimal 0000.

CLEAR (ADDRESS) SWITCH

The switch marked **CLEAR.** is located in the second row from the top on the right-hand side. When tocuhed it clears the **ADDRESS** switch indicators to hexadecimal 0000.

DATA SWITCHES

The third row from the top (below the **ADDRESS** switches) contains sixteen **DATA** bit switches marked 0 through 15. The word **DATA** appears directly above these switches.

The **DATA** switches code and generate the written data into the location selected by the **ADDRESS** or **REGISTER** switches.

The location can be in memory or a selected processor register. The **DATA** switches operate in the same manner as the **ADDRESS** switches. The **DATA** switch indicators also diplay the data read from any address or register location.

CLEAR (DATA) SWITCH

The switch marked **CLEAR** is located in the third row from the top on the right-hand side. When touched, it clears the **DATA** switch indicators to hexadecimal 0000.

BUSY INDICATOR

The indicator marked **BUSY** is located directly above the **RESET** indicator.

The BUSY indicator lights for at least .1 second whenever the Infibus is busy with at least one bus transfer.

IDLE INDICATOR

The indicator marked **IDLE** is located between the **RUN** and the **HALT** switches in the bottom row.

The **IDLE** indicator lights when the CPU or another system processor is waiting for an external interrupt.

2.1.4 Master Mode

The following touch-response switches and LED indicators either initiate or are related, by the control panel logic, to an actual Infibus access.

Such an Infibus access implies that the control panel is the master and the addressed module is the slave.

DONE INDICATOR

The **DONE** indicator is located on the right-hand side in the third row from the top. When lit, the **DONE** indicator acknowledges that a bus access (with the control panel as the master) has been successfully completed. The **DONE** indicator remains dark if the

action is aborted.

The **DONE** indicator functions only when coupled with the following switch actions:

a. **WRITE** (to register)
b. **READ** (from register)
c. **WRITE** (at address)
d. **READ** (at address)
e. **RUN**
f. **HALT**
g. **ADH**
h. **STEP**

The **ATTN** switch (which accesses the bus on external interrupt level 1 instead of on the direct data transfer level) does not use the **DONE** indicator.

WRITE (TO REGISTER)

The switch marked **WRITE** is located on the left-hand side of the top row.

The **WRITE** switch lights, and then writes the data displayed in the DATA indicators into the selected CPU or system processor register. If the **DONE** indicator is lit, access was successful.

READ (FROM REGISTER)

The switch, marked **READ** is located on the right-hand side of the top row.

When touched, the **READ** switch lights. If the **DONE** indicator is also lit, the **DATA** indicators display the contents of the addressed processor register.

WRITE (AT ADDRESS)

The switch marked **WRITE** is located on the left-hand side of the second row of switches.

The **WRITE** (at address) switch operates in the same manner as the **WRITE** (to register) switch with one exception. Depressing the **WRITE** (at address) switch

writes the **DATA** indicators settings into the address displayed by the **ADDRESS** indicators.

READ (AT ADDRESS)

The switch marked **READ** is located on the right-hand side of the second row of switches.

The **READ** (at address) switch operates in the same manner as the **READ** (from register) switch with one exception. Depressing the **READ** (at address) displays the **DATA** indicators at the address displayed by the **ADDRESS** indicators.

The first touch of the **READ** switch does not increment the **ADDRESS** indicators before accessing the address. However, each successive touch of the **READ** switch increments the **ADDRESS** indicators by 2 prior to access. Touching any switch except the **REGISTER** or **DATA** switches before touching the **READ** switch, prevents the **ADDRESS** indicators from being incremented prior to the access.

RUN

The switch marked **RUN** is located in the bottom row directly to the left of the **IDLE** indicator.

If the **DONE** indicator is lit, touching the **RUN** switch forces the selected processor to resume executing instructions. The **RUN** switch indicator remains lit if at least one processor is executing instructions. If the **IDLE** indicator is lit, the processor unit is waiting for an external interrupt. After all processors have halted program execution, the **HALT** indicator turns on.

The **RUN** and **HALT** indicators may appear to be on simultaneously when the Run and Halt conditions are interlaced at a high repetition rate. If this occurs the **HALT** indicator may be less intense than the **RUN** indicator since the **HALT** has a minimum delay time of .1 second.

HALT

The switch marked **HALT** is located to the right of the **IDLE** indicator in the bottom switch row.

If the **DONE** indicator is lit, touching the **HALT** switch forces the selected processor to stop executing instructions. After all processors have halted program execution, the **HALT** indicator is lit.

ADDRESS HALT

The switch marked **ADH** (Address Halt) is located directly to the right of **HALT** in the bottom row.

When the address on the Infibus compares exactly to the **ADDRESS** switch indicators of the switch panel the HALT indicator is lit. The selected processor is halted as long as the **DONE** indicator remains lit. The **ADH** switch is ineffective during the time the panel control logic, functioning as the master, accesses the Infibus.

The **ADH** switch-indicator has toggle action.

STEP

The switch marked **STEP** is located to the right of the **ADH** in the bottom switch row.

Touching the **STEP** switch lights its indicator and causes the addressed processor unit to execute the next instruction. At the completion of this instruction the contents of the selected processor register (**REGISTER 0-15**) are displayed on the **DATA** indicators. The **DONE** indicator must remain lit following this action.

ATTENTION

Touching the **ATTN** switch causes the generation of an "attention" interrupt. The **ATTN**, switch differs from all other switches since it does not actuate its function until the touch is released. The **ATTN** indicator may appear to remain off if the processor immediately services the attention interrupt. However,

if the processor was not interrupted upon release of the **ATTN** switch, then the **ATTN** switch inicator remains lit and the control panel is locked until the interrupt is serviced.

This condition occurs:

a. if all processors are halted
b. if the **INH** switch indicator was lit prior to releasing the **ATTN** switch
c. if the CPU is operating with all interrupts inhibited

The **ATTN** switch can always be switched off by touching and releasing the switch a second time, thereby unlocking the control panel.

The control panel address (device number) is issued for the **ATTN** interrupt on priority level 1.

2.1.5 Slave Mode

The control console **ADDRESS** and **DATA** switch-indicators are used either for visual display or for manual input of information during processor execution.

There are two 16-bit registers in the panel control logic, one for address and the other for data, that can be addressed by processors and other master system modules. Each of these registers is associated with its respective set of switch-indicators on the panel. The registers are assigned special addresses that can be accessed in the same manner as a location in memory. To display data on either the **ADDRESS** or **DATA** indicators, the program must include a register-to-memory class of instruction when the display data is written into the addressed panel register.

To sample the **ADDRESS** or **DATA** switch settings, the program must include a memory-to-register class of instruction where the switch contents are read from the panel register.

A maximum of four separate control panels may be operated on the Infibus. However, most Pluribus systems have only a single panel, and this panel is always designated as control panel number 1. Table 2 lists the possible register addresses for four control panels in a system.

Since a conflict between an operator and the program may occur during a write function, the contents of the register should be read back to verify what was written. If the contents do not compare, another attempt should be made to write the proper data.

Table 2. Addressable Registers (Switch-Indicators)

Panel No.	Panel Register Name	Hex Address
1	address	FF80
1	data	FF82
2	address	FF84
2	data	FF86
3	address	FF88
3	data	FF8A
4	address	FF8C
4	data	FF8E

INTERRUPT INHIBIT

Figure 3 shows the location of three toggle switches on the back side of the switch panel assembly. These switches permit manually enabling or inhibiting the power fail interrupt, power recover/autoload and line frequency interrupts.

POWER FAIL The PWR FAIL switch (S72) (PFIN) is located as shown in figure 3. Placing the switch OFF inhibits the power fail interrupt (PWST).

In the ON position the power fail interrupt is issued when the BCU detects a power fail condition.

POWER RECOVERY

The PWR RCVRY switch (S71) (PRIN) has three positions, OFF, IN, and AL, to inhibit or to enable the power recovery (PWST) interrupt and power recovery via auto-load (PRAL) interrupts.

In OFF position neither interrupt is issued upon recovery.

In IN position the power recovery interrupt is enabled and the auto-load interrupt is inhibited.

In AL positon, the power recovery via auto-load interrupt is enabled and power recovery interrupt is inhibited.

The auto-load interrupt can only be issued if the system contains an auto-load module (ALD).

2.1.6 Line Frequency

The LINE FREQ switch (S70) (LFIN) inhibits or enables the line frequency interrupt (LFRQ) (JIFFY CLOCK). Turning the switch OFF inhibits the line frequency interrupts. In the ON position, the line frequency interrupt is enabled and a level 4 interrupt service request is issued once during each line frequency cycle.

2.1.7 Optional Functions

Several optional functions are available as jumpers or switches on the switch panel.

The On switch position has the same effect as connected jumper; the OFF position the same effect as a disconnected jumper. The switches or jumpers are described in table 3.

Table 3. Optional Switch/Jumper Functions.

Switch or Jumper	Jumper Position	Function
BYTE (S66)	ON	Only bytes are read or written. Full word can be read or written.
	* OFF	
INCREMENT CONTINUOUS READ OR WRITE ADDRESS (S67)	ON	Address increments continuously during read mode if S68 (below) is ON; during write mode if S688 and S69 are ON. Address increment disabled during continuous read/write mode.
	* OFF	
CONTINUOUS READ/WRITE (S68)	ON	Enable continuous read/write mode.
	* OFF	Disable continuous read/write mode.
WRITE INCREMENT (S69)	ON	Allows address increment during write.
	* OFF	Address does not increment during write.

The * jumper positions are the standard settings

2.1.8 Configuration

The following pages explain the strapping necessary to properly configure the PBI and PCB for BBNCC applications.

3 PROCESSORS

3.1 Physical Description

Each processor(CPU)module consists of two printed circuit cards, the CPA and the CPC. These cards are interconnected by their backedge connectors (those opposite the Infibus end). An ICM module connects the first CPU on a bus (or primary processor) to the BCU. The ICM module contains three connectors that interconnect the CPA, CPC* and BCU.

In multiprocessor systems, the secondary (or buddy) processor CPA and CPC* are interconnected by an IDM interconnect module containing two connectors. The CPU is always located in the first slot lett of the BCU (viewing the chassis from the insertion end). The CPA is always left of the CPC*. Secondary processors are always located in the slot to the left of the primary CPU. Secondary processors are always located in any slots to the left of the CPU.

Edge connectors on the primary circuit cards are keyed to prevent improper insertion on the Infibus. However, circuit cards are not keyed to prevent improper relative slot locations.

* In older systems a CPB may be used instead of a CPC. Both cards are functionally equivalent.

3.1.1 CPA

CPA - 2102292G01

The CPA contains the ALU, ALU multiplexor (MUX), register file and Infibus interface logic.

3.1.2 CPC

CPC -2102291G01

The CPC contains the microcode control logic and ROM storage.

3.2 Specifications

3.2.1 Power Requirements

+5V, +15V and -15V
All voltages are plus or minus 1%

3.2.2 Performance

Instruction execution rate	250 instructions per second (average)
Arithmetic function process time	165 nanoseconds 2 parallel 16-bit operands
Logical function process time	145 nanoseconds 2 parallel 16-bit operands
Addressable registers	12
Program Counter (R0) 16-bits	1
General registers (R1-R7)16-bits	7
Status register 16-bits	1
Instruction register	1
Firmware register A 16-bits	1
Firmware register B 16-bits	1
Halt/Run Control	1
Interface registers	4
Receiver register	16-bit
Transmitter register	16-bit
Address register	16-bit
Protect key	2-bit

Internal control registers	3
E register	16-bit
S register	12-bit
M register	36-bit
Microcode memory	256-word, 36-bit ROM
Lookup table memory	256-word, 8-bit ROM

3.3 Instruction Set And Format Summary

The basic processor in the Pluribus system is a microprogrammed general purpose 16-bit minicomputer with 8 general registers (including a program counter) a status register, and a control register. These registers may be accessed externally by other devices. The multiprocessor architecture allows one processor to stop another, to examine and change its registers, and to restart the system.

Most of our Pluribus systems use a general purpose processor that executes a 65-mnemonic instruction set. This processor functions as a central processor unit (CPU) in a single processor system, or as either primary processor (CPU) or a secondary processor in a multiprocessor system. The Infibus can accomodate up to three secondary processors.

Every primary processor is connected with the BCU. This allows the processor to recieve system interrupts on priority levels 4 through 1, and to perform central processing functions. Secondary processors have no control over system interrupts.

There are 8 general instruction classes: MOVE, ADD, SUBTRACT, INCLUSIVE OR, EXCLUSIVE OR, AND, COMPARE, and TEST. Each of these instructions may use a variety of addressing modes including register-to-register, memory-to-register, register-to-memory, indexed, indirect, and auto-indexed. Rotate, shift, conditional branch, unconditional branch, and subroutine call instructions are also available.

The bits in the processor's status register test the branch conditions, and have tests for the following results:

result = zero
result = negative
carry on last arthmetic instruction
register value odd
overflow
value greater-than on last comparison
value equal on last comparison
loop completion

The branch can occur on either value TRUE or FALSE. Instructions contain either one or two words. The processor's status register contains 3 programmable flags. The software manuipulates these flags.

The processor recognizes and generates 16-bit addresses. In addition, it contains a 2-bit KEY register which is settable by the SKEY instruction in the processor. The contents of this register are appended to the most significant end of the 16-bit address to generate an 18-bit address. Every memory access by a processor has these two bits appended. Certain of these 18-bit addresses are mapped into 20-bit system addresses described later.

The processor operates on either 16-bit words or 8-bit bytes of data. Bit 0 is identified with the low order bit and bit 15 with the high order bit.

3.4 Processor States

The processors can be in one of three states: halted, running, or idle. Transitions between these states may be affected either by the processor itself or by external manipulation. The implications of each of these three states are as follows:

Halt: No instructions executed, interrupts disabled, registers externally accessible.

Run: Instructions executed, interrupts enabled, registers not externally accessible except control register.

Idle: No instructions executed, interrupts enabled, registers not externally accessible except control register.

External references to registers which are not accessible will result in a QUIT, as described in Chapter 12. The IDLE state is entered from the running state by executing a WAIT instruction; the HALT state, by a HALT instruction. The RUN state is entered from the Idle state by the occurrence of an interrupt.

External manipulation of these states operates as follows:

The processor control register is the only register accessible while the processor is running. To halt a processor, a one is written to its control register.

The processor normally halts when the instruction it is currently executing is completed. However, if the control register is read between the time that a zero is written to the control register and the time that the processor completes its current instruction, the halt signal will be lost. To guarantee that a processor does in fact halt, an algorithm such as the one in the following example is used: To guarantee an algorithm such as the one in the following example is used:

```
L:  Write 1 to the processor control register.
    Read some other processor register.
    If QUIT results go to L.
    At this point the processor is halted.
```

To start a processor, one must initialize all important registers to needed values and store the number two into the control register.

This is done as follows:

1. Assure that the processor is halted.
2. Initialize the program counter to the address of the program to be executed.
3. Load the general registers with any values to be passed to the program
4. Initialize the status register to specify the enabled interrupt levels, initial status flags, and programmable flags.
5. Set bit 11 (hexadecimal constant 0800) to activate the processor.
6. Write a 2 to the control register and to start the

processor.

The ability to single-step through an instruction sequence is one of the processor's advantageous features. The procedure for this function is identical to the procedure for starting a processor except that a 3 is written to the control register instead of a 2.

3.5 Quit Handling

Processors requesting access to memory locations on I/O registers do so by directly or indirectly placing the desired address on the appropriate bus along with the required operation (e.g., read, write) and any required data(for a write operation).

When the requested operation is complete, the processor receives a signal called DONE. If no device on the destination bus recognizes the relayed address, or if the device recognizing the address malfunctions, the DONE signal is not returned to the processor. Instead, after a fixed period of time, the BCU on the requesting processor bus will send a QUIT signal to the processor, causing a conventional interrupt. Requests by I/O devices for I/O busses yields an equivalent outcome.

To determine if known devices have disappeared or if new ones have appeared, a device discovery routine in the reliability software searches system address space, reads the device registers, and checks for resulting Quits. This mechanism requires an interrupt level routine which responds to Quits, and then passes the information back to the application program. A unique pattern surrounding the Quit instruction generates this routine.

For example, to check for a QUIT occurring when location ABC is referenced you may write the following:

```
        LDA         A2, ABC
        NOP
        BR          . + 4   ----------------------------!
   L:   QUIT BRANCH ADDRESS                             !
                                                        !
             If no QUIT, program continues here.<------!
                   !
```

The QUIT interrupt service, upon receiving control, would check to see if the two instructions following the one which caused the QUIT were NOP and BR . +4. If so, the QUIT service simply stores the two bytes starting at location L (the address of the instruction causing the QUIT plus 8) in the program counter and dismisses the interrupt.

If the two instructions at L-4 and L-2 do not match the NOP BR . +4 pattern, the interrupt service routine handles the QUIT in the usual manner.

Of course, references to ABC which do not cause QUITs cause the execution to continue at L+2 as indicated. Under the control of a microprogram stored in a Read Only Memory (ROM) on the processor module, each processor executes its respective instructions.

3.6 Functional Description

The microprogram is stored in one 1024-bit ROM microcircuit components. Each component is arranged in 256, four-bit words. Therefore, the nine components store 256 thirty-six-bit control words that compose the microprogram.

After each 36-bit control word is read out of the ROM (and sometimes modified), the word is decoded into 13 fields to supply enabling and timing signals to functional elements throughout the processor.

Figure 1 shows the general data flow between processor function elements. The functional elements can be grouped into three sections: Infibus interface, arithmetic-register, and microcode control.

The Infibus interface section logically interconnects the processor and the Infibus. The arithmetic register section performs arithmetic and logical functions under control of the microprogram. The microcode control section stores, reads out, assembles and decodes the microcode control words.

3.6.1 Infibus Interface Section

The Infibus interface section consists of drivers/receivers that interconnect the processor and Infibus. This section also contains address recognition and decoding logic, three 16-bit registers: address (A Reg), data receive (R Reg) and data transmit (T Reg), the run/halt control flip-flops and a protect-key register.

The A, R, and T registers are gated to the Arthimetic Logic Unit (ALU) through the ALU multiplexor (MUX) in the arthmetic-register section. The A register holds addresses placed on the Infibus when the processor is addressing a location in memory or a system register. The R register holds 16-bit data words received on the Infibus. The T register has left- and right-bit shift capability.

The halt/run flip-flop enable processor control functions, and the protect-key bits (address bits AB16 and AB17) enable Key 0 and Key 1 bits, respectively, on the Infibus.

3.6.2 Microcode Control Section

The processor microcode control section consists of three registers: E, S and M, the ROM control storage, a ROM data lookup table, and microcode selection logic.

E Register

The E register holds 16 bits of data used to modify the next microcode word.

Fields in this register correspond to the macro-instruction word format and specify the instruction code, addressing mode, general register (R1 through R7), index register and, occasionally, a literal operand. E register bits 3 through 0 also operate as a loop counter.

S Register

The S register is a 12-bit counter that sequences microsteps, addresses the ROM control storage, and is under control of the microcode word. The four most significant bits (bits 11 through 8) are always ZERO. Bits 7 through 0 are used to address the 256-word ROM.

M Register

The M register receives the 36-bit microcode word that specifies the action of the current microstep as well as controlling the next sequential microstep. M register outputs are decoded into enabling signals, and distributed with clock pulses throughout the processor as the processor control logic.

ROM Control Storage

Microcode control words are stored in the ROM control storage - a 36-bit by 256-word read-only memory consisting of nine 256 by 4-bit LSI microcircuits. The 36-bit microcode word contains 13 fields that are decoded to select operands and specify control functions within the processor.

Test and conditional skip or conditional branch microfunctions are provided to allow coding that minimizes the number of microsteps required per function. Functions of the 13 fields are summarized here:

a. **S Field** - Specifies type of microcommand. The type may be a normal sequential step, special command, branch, memory synchronize, or conditional jump command.

b. **T and A Fields** - Select one of 16-logical functions if T=ZERO, or one of 16-arithemetic functions if T=ONE, to be performed by the ALU.

c. **C Field** - Specifies option of adding or subtracting a Carry In to the ALU and also controls setting Carry and Overflow flip-flops.

d. **D Field** - Selects the option of skipping the next micro order either unconditionally or based on the ALU output being all ONEs or the jump condition being met.

e. **X Field** - Selects the X field of 12 registers in register file for an ALU input and/or an ALU destination.

f. **F Field** - Specifies the X field modification for the next micro-instruction.

g. **Y Field** - Selects R, A, or T register or literal fields L2 and L1 in the microcode (under control of the M field) for input to ALU.

h. **M Field** - Operates with other fields to provide different functions. When Y=3, it controls mapping of literals L2 and L1 fields to the ALU's Y input. When S=1 and L2=7, it specifies the type of shift to perform. When S=2, it is part of the branch address. When S=4 or 5, it specifies conditional jump codes. When S=6 or 7, it specifies a bit position in the T register to be tested.

i. **L2 Field** - When Y=3, it is part of the literal that is gated to the ALU Y input. When S=1, it is a special command. For S=2 through 7, L2 represents the most significant bits of the control storage branch or jump addresses. For Z=1 and L1=0,4,8 or 12; L2 represents the four most significant bits of the data lookup table address.

j. **L1 Field** - When Y=3, it is part of the literal that is gated to the ALU Y input. For S=2 through 7, it represents the 4 least significant bits of branch or jump addresses. When Z=1, the L1 field specifies bits to be used to generate a special literal source or which field of the E register is to be used as the least significant four bits of the lookup table address.

k. **Z Field** - This bit is used to enable special

interpretation of L2 and L1 fields to select a field from the E register or other sources for generation of special literals on the next micro instruction (by modifying the L1 and L2 or M fields).

1. W Field - Selects register file specified in X field and/or one of the A, T, or E registers or the loop counter to receive the ALU output. For W=5,6, or 7 both the register file and A, T, or E register are selected to receive ALU output, respectively. For W=0 or 4 the ALU1 signal is copied into the ONEs flip-flop. For W=3 the four least significant bits are copied into the loop counter.

ROM Data Lookup Table

The ROM data lookup table is an 8-bit, 256-word read-only-memory consisting of two 256 by 4-bit LSI microcircuits. The lookup table is enabled and addressed by the current microcontrol word in the M register and a 4-bit field from the E register.

When enabled, its output is ANDed with the literal fields of the next ROM control word. Other status conditions, when enabled by the current ROM control word, also may be ANDed into the literal fields of the next ROM control word.

The microcode selection logic forms the AND of the ROM control storage and the other enabled functions and places the control word into the M register. Through a look-ahead feature inherent in the processor, the next control word is read while the current word is executed.

3.6.3 Master-Slave Modes

The processors operate in either a master or slave mode. A processor operates in the master mode when addressing a system module; and in the slave mode when addressed by a system module. The master mode is the most common mode of operation when the processor executes programmed instructions and self interrupts.

If operating as a CPU, the processor allows the requesting device number to be received while in the slave mode, and thereby controls system interrupts on priority levels 4 through 1. Since the register file cannot be accessed by an external module (such as a control panel), the processor usually halts when addressed in the slave mode. However, the run/halt control register is accessible in either the halt or run modes and, as a result, a processor can be halted while running.

3.6.4 Programming Considerations

Several programming considerations are discussed in the following paragraphs: register addressing, ACTIVE-IDLE states, HALT-RUN control, interrupt control and multiple processors.

Register Addressing

Thirteen registers, addressable from the control panel, provide register display when the processor is in the HALT mode. Table 1 contains a list of the addressable registers and identifies their function in the processor.

Active-Idle State

Bit 11 of the status register specifies, under control of the firmware, the operation mode that the processor assumes after executing an instruction or servicing an interrupt request.

If bit 11 is a Zero, the processor assumes the IDLE state. If bit 11 is a One, the processor assumes the ACTIVE state.

The IDLE state is a simple loop operation with the Run flip-flop asserted. The processor interrogates the Halt flip-flop, and if the processor is a CPU, it looks for any enabled interrupt service requests.

In ACTIVE state, the RUN flip-flop is asserted and the processor executes the next instruction in a program sequence or services an enabled interrupt request.

Table 1. Register Addresses

Register Switch	Register Address (Hex)	Register Name
0	FFx0	General Register 0 (PC)
1	FFx2	General Register 1
2	FFx4	General Register 2
3	FFx6	General Register 3
4	FFx8	General Register 4
5	FFxA	General Register 5
6	FFxC	General Register 6
7	FFxE	General Register 7
8	FFy0	Status Register
9	FFy2	Instruction Register
10	FFy4	Firmware A)address of last instruction)
11	FFy6	Firmware B
12	FFy8	Not implemented
13	FFyA	Not implemented
14	FFyC	Not implemented
15	FFyE	Halt/Run Control

x = 0, 2, 4, 6 for processors 0, 1, 2, 3 respectively.
y = 1, 3, 5, 7 for processors 0, 1, 2, 3 respectively.

Halt-Run Control

The control register consists of two flip-flops that control HALT and RUN modes of operation. The Halt flip-flop is asserted or negated any time it is addressed. The Run flip-flop is asserted any time the processor is taken out of the HALT mode.

The flip-flops are addressable from a control panel. The control register (Run/Halt flip-flops) read out is 0000 if the processor is running and FFFF if the processor is halted.

NOTE
Since the CPU is incapable of servicing interrupts in the HALT mode, the programmer should ensure that a secondary processor does not place the CPU in this mode.

The control register functions in four possible states. Table 2 identifies these states and the processor mode for the three meaningful states. Table 2 is a list of four possible states for the control register, identifying the processor mode for the three meaningful states. Bit Zero of the control register controls the Halt flip-flop and bit one, the Run flip-flop.

The Halt flip-flop is interrogated at the end of each instruction execution and, if asserted, the processor is halted. If the Halt flip-flop is negated, the processor is in the RUN mode or is executing a WAIT instruction. If both Halt and Run flip-flops are asserted, the processor executes one instruction at a time. This single instruction execution can be initiated by pushing STEP on the control panel.

Table 2. HALT-RUN Control Codes

RUN	HALT	Mode	Interrupts
0	0	Not defined	
0	1	HALT	Cannot be serviced
1	0	RUN* or IDLE*	Allowed (if enabled)
1	1	STEP	Allowed (if enabled)

* RUN (ACTIVE) if bit 11 of status register is a one
RUN (IDLE) if bit 11 of status register is a ZERO

Interrupt Control

The CPU accepts a system interrupt only if a bit in the status register corresponding to the interrupt priority level is enabled, and the control panel's master interrupt inhibit signal (INH) is absent from the Infibus.

Bits 15 through 12, of the processor status register, control system interrupts 4 through 1, respectively. Zero enables and One disables the interrupt. When the CPU executes an interrupt, all system interrupt priority levels 4 through 1 are disabled.

The program is written to ensure that interrupts permitted during an interrupt subroutine are enabled by resetting corresponding status bits to Zero at execution time.

Executive memory is reserved for interrupt control. The programmer reserves addresses 0000 to 005F for system and self interrupt executive space, and provides the proper vector addresses for interrupt service routines.

Memory executive space and the memory addresses assigned to each interrupt level appear in Table 3.

Executive space for two self interrupts(level 5 and 6) are available for each processor, in additioin to the space for four system interrupts. (See below).

Table 3. System and Self Interrupt Executive Space

	ADDRESS THAT CAUSED ABORT	STATUS	ABORTED INSTRUCTION ADDRESS	SERVICE ROUTINE VECTOR
6	ADDR00 0028	ADDR00 002A	ADDR00 002C	ADDR00 002E
	ADDR01 0038	ADDR01 003A	ADDR01 003C	ADDR01 003E
	ADDR10 0048	ADDR10 004A	ADDR10 004C	ADDR10 004E
	ADDR11 0058	ADDR11 005A	ADDR11 005C	ADDR11 005E
	UNIMPLEMENTED INSTRUCTION	**STATUS**	**UNIMPLEMENTED INSTRUCTION ADDRESS**	**SERVICE ROUTINE VECTOR**
5	ADDR00 0020	ADDR00 0022	ADDR00 0024	ADDR00 0026
	ADDR01 0030	ADDR01 0032	ADDR01 0034	ADDR01 0036
	ADDR10 0040	ADDR10 0042	ADDR10 0044	ADDR10 0046
	ADDR11 0050	ADDR11 0052	ADDR11 0054	ADDR11 0056
	MODULE ADDRESS	**STATUS**	**PROGRAM COUNTER**	**SERVICE ROUTINE VECTOR**
4	0018	001A	001C	001E
	MODULE ADDRESS	**STATUS**	**PROGRAM COUNTER**	**SERVICE ROUTINE VECTOR**
3	0010	0012	0014	0016
	MODULE ADDRESS	**STATUS**	**PROGRAM COUNTER**	**SERVICE ROUTINE VECTOR**
2	0008	000A	000C	000E
	MODULE ADDRESS	**STATUS**	**PROGRAM COUNTER**	**SERVICE ROUTINE VECTOR**
1	0000	0002	0004	0006

Typical Interrupt Operation

When the CPU acknowledges an enabled interrupt request, the firmware accepts the module address and interrupt number, and stores the requesting module address, processor current status, and program counter contents (next instruction address) into system interrupt executive space.

The firmware then loads the service routine vector address into the program counter (R0), and vectors to the interrupt subroutine address with all interrupts disabled.

Once the interrupt subroutine has been serviced, and to return to a a program executed prior to the interrupt, activate a Return-From-Interrupt instruction (RETN). This instruction replaces the status register with the previous processor status and loads the program counter (R0) with the address of the interrupted program's next sequential instruction.

If the CPU acknowledges a second interrupt before RETN is executed, the CPU loads the new interrupt module address, status and program counter information in the executive space for the interrupt, and vectors to a new subroutine. Consequently, another interrupt should not be enabled on the same or previous level before either returning or saving the interrupt status in a return stock.

Self Interrupts

Each processor controls two self interrupts: unimplemented instruction and address abort, levels 5 and 6, respectively.

Unimplemented instruction (level 5) signals the system, and may be used for trapping to arithmetic or logic subroutines that simulate the operation of the unimplemented instruction. Address abort or trap (level 6) avoids Infibus tie-up and notifies the system of an erroneous address or a unresponsive slave module.

Since any one of four possible processors in a multiple processor system can execute self interrupts, four different sets of memory address locations are reserved within the system interrupt executive space for self interrupts (Refer to figure 3).

3.6.5 Multiple Processors

When a processing task is too large for the CPU, the work load is divided among additional processors.

Secondary processors may be added to do different tasks, subtasks to those required by the CPU, or a parallel task (the same task with different input data). The second, third and fourth processors are designated as 01, 02, 03, respectively. The CPU, electrically connected to the BCU, is automatically designated as 00.

Regardless of the secondary processors' functions, the CPU schedules the work load, and passes data to memory or to the other processors if input is from an interrupt-driven device.

If the CPU does not schedule tasks, the processor communications routine must be designed with great care so that the processors do not lock up. The status register of each processor establishes a communication link between processors, and thereby determines task status and passes information.

Instructions from within the processor control the status register's three flag bits (F1, F2 and F3). These flag bits may also be used as a branch test, and may be interrogated by halting the processor and reading its status register.

When more complex information is passed between processors, a number of core memory locations are dedicated to each processor. These information blocks are then placed in these locations to await processor interrogation. Either incoming or outgoing information blocks may be used.

All processors have unlimited access to memory. As a result, partitioning memory and using multiple processors on an Infibus with common memory requires caution.

Software is designed to allow multiple processors to access the same programs, thus greatly reducing core memory requirements.

3.6.6 Configuration

The following pages explain the necessary strapping to properly configure the CPA and CPC boards.

4 MEMORIES

4.1 Functional Description

Memory modules only operate in the slave mode. The address and mode control signals sent by a master module initiates memory operations.

The memories perform one of three operations: read-restore, clear-write, or read-modify-write. The mode control signals from the master module initiating the operation specifies the particular operation. A full word consisting of 16 bits, or a byte consisting of either 8 bits or 9 bits, for each respective model, can be transferred during any of these operations.

During read byte operations, the memory module transmits the selected byte (left or right) in bit positions 7 through 0 on the Infibus data lines. The remaining bits are transmitted as zeros.

During write byte operations, data bits 7 through 0 are transferred into either the left or right byte position of the addressed memory location, and the other byte position of the word remains unchanged.

Read-Restore Operation

In the read-restore operation, a full memory cycle is initiated when the memory module recognizes the address of an accessed location. Data at the accessed location is read out destructively during the read portion of the cycle, and transferred to the data register. The data is written back into the same location during the restore portion of the cycle.

Clear-Write Operation

In the clear-write operation, a full memory cycle is initiated. Data at the addressed location is read out destructively and discarded during the clear portion of the cycle. During the write portion of the cycle, the new data in the data register is transferred to the accessed location.

Read-Modify-Write Operation

In the read-modify-write operation, a split memory cycle occurs. The read portion of the cycle is the same as the first half of a read-restore operation. The word or byte from the accessed location is sent to

the master module. The memory cycle is temporarily suspended to allow the master module sufficient time to modify the data.

When modified, the data is returned to memory. The master module initiates the write half of the cycle, and the modified word is restored to the accessed location. The cycle time depends on the time the master module requires to modify the word and to start the write portion of the cycle.

4.2 Multiple Memory Modules

In systems containing two or more memory modules, memory addressing can be modified to enhance operation for specific applications. Jumpers and/or switch settings on each memory module determine module address and addressing mode: overlap, interleave, or interlock.

Overlap Mode

In the overlap mode, two memory modules are operated simultaneously within the same system, where the next cycle is started in the second memory module before the current cycle is completed in the first, or vice versa. Therefore, the time required on the Infibus for communication with the master module is minimized. This capability is inherent in computer systems that contain two or more memory modules.

Each memory module contains its own data and address registers so that each may operate asynchronously and independently of other system modules. The overlap mode can be used either by a single master module alternately reading data from, or writing data into, two memory modules; or by two or more master modules, where each operates with a separate memory module.

For example, a processor can be executing instructions from one memory module while an I/O controller with a block transfer adapter is transmitting data received from a peripheral device to a second memory module.

Interleave Mode

The interleave mode is an extension of th overlap mode. To enhance the overlap mode in configurations having one master module operating with two memory modules, the memory can be modified by jumper wires to operate in the interleave mode.

Interleaving is accomplished by interchanging address bit positions 1 and 13 for 4K memories, and 1 and 14 for 8K memories throughout the entire storage address decode logic. One memory module is made to contain even-numbered word addresses, the other bank odd-numbered word addresses. With this arrangement, a single proccessor receives or transmits alternately data addressed consecutively from, or to, the two memory modules.

Interlock Mode

The interlock mode inhibits the overlap mode. Threfore, a new memory cycle cannot begin in any memory module until the current cycle is completed.

To implement the interlock mode, the memory modules are provided with a special jumper to disable the inherent overlap mode. Moditication to this mode is used to minimize power consumption when system performance can be reduced.

4.3 Programming Considerations

Memory operations can be initiated only when the applied module address code matches the assigned, hardwired memory address code. Thereafter, memory operatins are controlled by cycle-initiation, byte- or word-mode, and other control signals that are stored in the control storage register.

Memory Addressing

All memory locations in a Pluribus Computer System are assigned a 16-bit address. Coding of the address is shown in figure 1 and is described as follows:

1. Address bits 13, 14 and 15 specify the 4K module

address; bits 14 and 15 specify the 8K module address.

2. Address bits 1 through 12 for 4K modules and bits 1 through 13 for 8K modules, specify the storage location within the selected memory module.
3. Address bit 0 specifies which of the two bytes in the selected storage location is to be accessed during the byte mode operations: ZERO specifies left byte; ONE specifies right byte.
4. Address bits 16 and 17 specify the KEY bits, which correspond to Infibus signals KEY0-N and KEY1-N, respectively. If this option is enabled, the KEY bits can be used by the program as a code to enable and disable access to portions of memory.

Memory Addressing
4K MEMORY

KEY 1	KEY 0	Module Address	Word Address	BYTE
17 16		15 13	12 1	0

8K MEMORY

KEY 1	KEY 0	Module Address	Word Address	BYTE
17 16		15 14	13 1	0

System Addressing

One memory module in any Pluribus system must be addressed ZERO so that the auto-load, interrupt, power fail and restart functions are operative. All other memory modules are assigned addresses consecutively, ONE, TWO, THREE and so forth.

The first 24 word addresses in module ZERO are reserved for executive space to store status, program counters, interrupt vectors and device interrupt identifiers. Also, the first 256 word addresses are addressable directly with some control class instructions that use an absolute address mode.

The 16-bits of the system address registers define the address space as 0000 through FFFF hex. This space represents 32K words or 64K bytes of unique memory storage locations.

However, address F000 through FFFF hex are reserved to uniquely identify the various I/O controllers and system hardware registers which are directly addressable. This 2K word address space is larger than required, so in conventional practice, I/O controllers and system modules are assigned addresses in the range F800 through FFFF hex, leaving F000 through F7FF hex for the customer, either to be used as memory address space or for addressing custom modules.

4.4 TAG,SID,EXY - Magnetic Core Memory

TAG - 2102304G01
SID - 2102299G01
EXY - 2102299G01

4.4.1 Purpose

The Magnetic Core Memory, the first memory used with a Pluribus system, provides storage space within the Pluribus for the programs' diagnostics and data.

Core memory is non-volatile. Consequently, all data is preserved during power failures up to and including forever.

4.4.2 Description

Each memory module is contained on three printed circuit (PC) cards. The core mats and the solid state circuitry required to implement the memory logic is contained on the circuit cards. The cards plug into the Infibus at one end and terminate at the other end into an interconnect module (ICM).

The ICM module is the cap that fits across the outside edge of all three cards and provides one-for-one bussing of intra-module signals between the three PC cards.

The cards are designated:

TAG - Timing and Gating
SID - Sense/Inhibit Data
EXY - Electronics X- and Y-Drive

Core mats contained on the EXY card are planar arrays using 18-mil lithium-ferrite cores. The Location of the three PC card slots is not fixed. However, the cards must be adjacent and consecutive, thereby eliminating any vacant slots between the memory modules and the processor. The 3 cards are normally inserted into the Infibus in the following way: the TAG on the right, the SID in the middle, and the EXY on the lett.

4.4.3 Power Requirements

+15 volts
+ 5 volts
-15 volts

4.4.4 Performance Criteria

Cycle Time	800 +/- 25	nanoseconds
	875 +/- 25	nanoseconds
Write-access Time	280	nanoseconds
Read-access Time	460	nanoseconds
	500	nanoseconds

4.4.5 Configuration

The following pages explain the strapping necessary to properly configure the TAG-SID-EXY Memory Boards for use on the local or common bus.

BRUCE ANDERSON #3149

4.5 SME - Semiconductor Memory

SME - 2102550G01

4.5.1 Purpose

The SME memory board was designed to increase the memory capacity of a Pluribus bus.

The SME memory is volatile, signifying that all data is not preserved during power failures up to and including forever. A battery pack is available to eliminate this problem during short power outages.

4.5.2 Description

The SME is a single-board with 16K words of memory. The SME's ability to replace 8K words of core memory in 3 bus slots with 16K words of memory in one bus slot establishes the SME as a desired replacement for the TAG-SID-EXY core memories.

4.5.3 Performance Criteria

Cycle Time	800 +/- 25	nanoseconds
	875 +/- 25	nanoseconds
Write-access Time	280	nanoseconds
Read-access Time	460	nanoseconds
	500	nanoseconds

4.5.4 Configuration

The following pages explain the strapping necessary to properly configure the SME memory for use on the local bus or common bus.

5 BUS COUPLERS

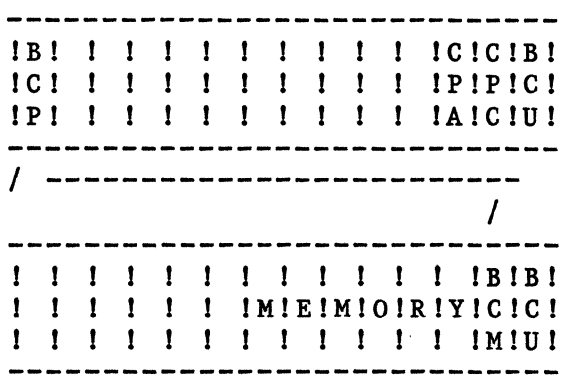
The bus coupler is a device which allows transactions on one bus to be transformed into transactions on another bus, as determined by the address. The bus coupler is composed of one card on each bus and connecting cables. There are three card types: BCP, BCM, and BCI. These cards make up the various coupler types.

Bus Type 1	Bus Type 2	Card Type 1	Card Type 2
P	M	BCP	BCM
P	I	BCP	BCM
I	M	BCI	BCM
P	M/I	BCP	BCM
M/I	M/I	BCI	BCM

P = processor bus
M = memory bus
I = I/O bus
M/I = memory and I/O bus

5.1 A Typical Scenario

Consider a processor reading a data word from common memory.



The processor first requests and then gains access to the Processor bus. The processor generates an address corresponding to the location it wishes to reference. The BCP recognizes this as a non-local address, and passes the request down the cable to the BCM. The BCM, in turn, decides that this address is relevant to its bus, and requests a cycle on the Memory bus.

When the cycle is obtained, the BCM passes the address. The memory recognizes this address, and delivers the fetched data and a DONE signal. The BCM accepts this data, passes it up the cable to the BCP, and releases the Memory bus for the next cycle.

The BCP, in turn, places the data on the Processor bus for the waiting processor, until the processor is free to release the Processor bus. To the processor, it appears as if it has addressed a slave with slow response on its own bus.

The memory similarly believes that it has been accessed by a local master. This is the model for all bus coupler transactions.

Since the bus coupler is in the critical path between the processor and the memory, it is a convenient place to perform address mapping and parity checking.

5.2 BCP

BCP - 2100528G01

Each BCP contains four 7-bit map registers for each of the four possible processors on the Infibus. The map registers are numbered 0-3, and are located in the address space of each processor at locations FC00-FC06.

Each processor has its own set of map registers, selected by bits 16 and 17 of the 18-bit address of data on the Infibus. These two bits are specified by the last execution of the SKEY of JKEY instruction in the particular processor.

Writing the new contents of the map to the corresponding address FC00, FC02, FC04, or FC06 modifies the map registers. The high order 7 bits of the word written becomes the new contents of the map register. The other bits are ignored. In general, bus coupler registers are written but not read. Reading a map register gives a result of zero and does not change the register. Infibus RESET does not affect the contents of the map register. The contents of the map registers are unpredictable at power-up.

During forward (normal) bus coupling the BCP is a Slave device on its bus, and the BCP transforms each 18-bit processor address into a 20-bit system address to be sent to the BCM.

Each processor's address space is divided up into 7 components:

Addresses	Description
0000-3FFF	References to Local Memory
4000-5FFF	Transform address using map 0
6000-7FFF	Transform address using map 1
8000-9FFF	Transform address using map 2
A000-BFFF	Transform address using map 3
C000-FBFF	Transform address to I/O space
FC00-FFFF	References to Processor Registers and Local I/O space

The BCU ignores addresses within 0000-3FFF or FC08-FFFF. The other 4k words ranges of the processor address space is transformed via an an address map. The BCP forms a 20 bit system address by preserving the low order 13 bits of the 16-bit processor-generated address while replacing the high order 3 bits by the 7-bit contents of the corresponding map register.

16-bit addresses in the segment C000-FBFF are referenced to Pluribus IO device registers. The system address for such a reference consists of appending four bits of 1's to the most significant portion of the address.

When the BCP transforms(maps)an address, this mapped address, any data, and one of the control operations (read, write, read-modify-write, byte) are

transmitted from the BCP to the attached BCM through a cable.

The BCM uses the control operations to generate a bus access on the target bus. The target bus is usually identical to the bus access on the source bus with the exception of the mapped address. The control operations will be used by the BCM to generate a bus access on the target bus, generally identical to the bus access on the source bus except for the transformation of the address.

Read operations using map 3, however, are changed into read-modify-write accesses on the target bus (where the write data is zero) to implement locks for multiprocess communications.

Write and read-modify-write operations through map 3 are unaffected.

When backwards bus coupling is enabled, the BCP acts as a Master on its bus and simply passes along the 18-bit references generated by the BCM at the other end of the cable.

5.3 Configuration

The following pages explain the necesary strapping to properly configure the BCP board for BBNCC applications.

5.4 BCM

BCM - 2101861G01

When a processor accesses a shared resourse on a memory or I/O bus, all of the BCP's on the source bus map the initial address and pass it along to the BCM end of the bus coupler.

Similarly, when an I/O device accesses a shared resource on a memory bus, all the BCI's on the source bus transmit the initial address to their BCM end. Each BCM then determines if the address sent to it is one to which it should respond. If it is not, the BCM simply ignores the request. If it is, the BCM requests access to its Infibus.

When the BCM receives control, it transfers the 20-bit address, any data, and all control signals to the bus and returns any responses received to the originating end(to the BCP or BCI) of the bus coupler pair. The Address Recognition Switches described below determine the addresses to which a BCM responds.

There are two important reasons for making the bus coupler perform address discrimination:

1. To reduce hardware contention.

   If each BCM simply passed all addresses to the containing bus, every processor reference to common memory would be in contention for each memory bus rather than just the single bus on which the referenced memory was located.

   A similar contention problem would exist for processor references to I/O busses and I/O references to common memory.

2. To eliminate multiple responses by the connected busses.

   Since a bus always responds either positively (by DONE) or negatively (by QUIT), one DONE and multiple QUITs would result from every access to common memory if address discrimination was not accomplished. The QUITs would, of course, confuse

the device that previously requested the access since it would already have taken actions based on the previous DONE.

This same issue is the motivation for configuring Pluribus systems so that BCMs connected to different busses recognize disjoint areas of system address space. In general, the addresses recognized by all BCMs connected to the same bus are identical.

5.4.1 Switches

The BCM contains several physical switch or jumper registers which must be manually set and a single 16-bit control register which may be referenced under program control. The switch registers along with the number of bits (switches) in each register are indicated below:

Switch Register	Number of Bits
MEMSW (Memory or I/O Bus)	1
BCM CONTROL REGISTER ADDRESS	6
BCM ADDRESS RECOGNITION:	
BASE	8
RELEVANCE	8

The algorithm used by the BCM for address discrimination is as follows:

If the BCU receives a 20-bit address less than FC000, then the high order 6 bits of the address are compared against the high order 6 bits in the BCM ADDRESS RECOGNITION switches.

The comparison is satisfied for a particular address bit, if either the corresponding RELEVANCE switch is OFF, or the RELEVANCE switch is ON and the address bit matches the corresponding switch (bit) in BASE. If all 6 high order bits satisfy the comparison,

then the BCM accepts and uses it to request a bus access.

Typically, the BASE and RELEVANCE switches are set to recognize a contiguous portion of system address space. Setting the high order 6 bits of BASE to some starting address and turning off some number of low order switches (within the high order 6) in RELEVANCE accomplishes this task. Of course, other settings of RELEVANCE switches implement more complicated memory access patterns.

If the 20-bit address passed to the BCM is greater than or equal to FC000, and MEMSW is on, the BCM does not recognize the address.

If the address is greater than or equal to FC000, and MEMSW is off, bits 11 and 12 of the address must satisfy the comparison test described above (with respect to the two low order bits of the BCM ADDRESS RECOGNITION switches) in order to ensure that the 20-bit address is recognized and placed on the I/O bus.

5.4.2 Control Register

The BCM contains one internal register, the BCM Control Register.

These control registers are located at the beginning of one of the address space segments recognized by the bus connected to the BCMs.

The 6-bit BCM CONTROL REGISTER ADDRESS switch specifies the precise location of a BCM Control Register. The number set in this switch supplies the BCM Control Register with displacement (in words) information from the starting address of the control register block.

To be more precise, the address of each BCM control register is:

Address Bits	From
14-19	High order 6 bits of BCM ADDRESS RECOGNITION BASE switches if MEMSW on If MEMSW off, 1
13	Negation of MEMSW
11,12	Low order 2 bits of ADDRESS RECOGNITION BASE switches
7-10	0
1-6	Contents of BCM CONTROL REGISTER ADDRESS switches
0	0

The control Registers for BCMs on a memory bus and those on an an I/O bus differ in one respect. Those on a memory bus share addresses with the memory devices on that bus, whereas those on an I/O bus do not share locations with any I/O device. Since devices referencing BCM control registers expect a single DONE signal upon completion of an access, the BCM works as follows:

If MEMSW is on, the BCM does not return a DONE since references to the memory device returns the DONE signal.

If MEMSW is off, the control register acts as a device in its own right. Since there are no "overlapping devices", the BCM generates and returns a DONE signal for references made to its control register.

BCM and memory devices share the same address when the BCM control is read or written. For BCMs attached to I/O busses, reading will return 2100 if the attached bus is up (operational) or 0 if the bus is in the process of going down due to a power failure (of course a QUIT is returned if the attached bus is completely

down).

If the BCM shares the address of its control register with a memory module, however, this 2100 or 0 is Inclusive-Or'ed with the contents of the associated memory word. For this reason, and to permit proper operation of the Pluribus system parity mechanism, any read of a BCM control register is normally preceded by a write of zero to the control register. This clears any "shadow" memory location. Since the password is incorrect, it does not affect the control register contents.

The BCM control register has two latches which respectively allow forward (BCM-to-BCP) and backward (BCP-to-BCM) requests. The control register also allows a reset of the processor bus on the attached BCP. To make any change in the state of the control register, the high order 13-bits of the data word written must also contain the DE78 hex pattern (the password). If the BCM control register is on a memory bus and if the key does not match, a DONE repsonse is returned. If BCM control register is on an I/O bus and the key does not matech, a QUIT response is returned. The lower order 3 bits have the following effects:

bit 2 (4) enables backwards data transfer(backwards bus coupling)

bit 1 (2) enables forward data transfers

NOTE: 1 = enabled 0 = disabled

Bit 0 (1) if zero causes a reset of the bus on the attached BCP; otherwise it does nothing.
NOTE: The state of the latches is changed every time (but only when) the high order bits match the password pattern. Furthermore, backwards coupling should be enabled only if MEMSW is off. On reset and power-up, forward coupling is enabled, and backward coupling is disabled. A read of the control register has no effect on its contents.

5.5 Backwards Bus Coupling (BBC)

BBC is used to initiate an access on the bus associated with the BCP, on behalf of an access on the BCM bus (usually an I/O bus). This path allows processors to access other processor's address space, and thus start and stop processors and inspect and change local memories. Before a BBC-access can take place, the individual BCM attached to the desired processor bus must be enabled for BBC as described above.

Since two BCMs cannot be enabled for the BBC simultaneously, a software lock is usually used to sequence BBC usage.

BBC uses a block of eight words in system space. This block is located at the beginning of the second segment of each I/O bus address space plus 80 hex. Thus the address of the first word of the block is as follows:

I/O bus 1	FE080
I/O bus 2	FE880
I/O bus 3	FF080
I/O bus 4	FF880

Each BCM uses address recognition switch A, positions 7 and 8, to determine which of these addresses it will recognize. The eighth word of the BBC block (described above) is the map register, and holds the high order 15 bits of the processor bus address to be accessed with BBC. The high order 13 bits (FFF8) of the data word written to the map specify the corresponding bits of the address within a processor's address space.

Bits 2 and 1 (6) select which processor's address space (of four on a bus) is used. The 0 bit (1) is ignored.

The map register may only be accessed when the desired BCM is enabled for BBC; otherwise a QUIT results. Contents of the map are not affected by any references other than to the map register. Reading the map gives a zero and also sets its contents to zero. Reset and power-up clears the map.

When a BCM is enabled for BBC, an access to one of the first four words of the BBC block is transformed into an access on the bus of the attached BCP. The value of the map register determines the address of this access, and the displacement of the originating access from the beginning of the BBC block. Thus, setting the map register causes a mapping of the first four words of the BBC block on the BCM bus (the "window") onto four words of the BCP bus.

Write, read, and byte operations are transformed into corresponding accesses. A read-modify-write access through the BBC window is not allowed. Since the request paths of the coupler are full-duplex, it is possible for forward and backward requests to happen simultaneously. Since the continuation of either must wait for the completion of the other, a potential deadlock exists. The deadlock is prevented by the bus controller (BCU) on the I/O bus. The I/O bus BCU times-out the BBC requests via a QUIT, aborting the backwards request. The I/O bus quit causes the BCP to abandon the BBC request initiated by the BCM, and the forward request continues to completion.

Review of MEMSW functions - when off:

1. allows recognition of I/O address
2. causes a DONE to be given on accesses of the control register
3. adds 2000 hex to address of control register
4. allows BBC to be enabled
5. ensures that the control register address is in I/O space

5.6 Configuration

The following pages explain the necessary strapping to properly configure the BCM board for BBNCC applications.

5.7 BCI

BCI - 2100168G01

The BCI serves in place of a BCP when coupling an I/O Infibus to a memory Infibus. Its relation to the BCM is identical to that of the BCP with the following exceptions:

(1) No address mapping is performed (devices on I/O busses generate 20-bit addresses).
(2) Any address less than FC000 (with any data and control signals) is passed directly through the BCI to the BCM.
(3) Any address greater than or equal to FC000 is ignored.
(4) Read, write, read-modify-write, and byte operations cause the corresponding accesses to be performed.

Devices on an I/O Infibus cannot directly communicate with any I/O devices on another I/O Infibus; any such communication must be done via common memory. The BCI-BCM bus coupler cannot be used for backwards bus coupling.

5.8 Configuration

The following pages explain the necessary strapping to properly configure the BCI board for BBNCC applications.

5.9 Power Sense

Each end of the coupler senses the logic supply voltage of its other end. If it falls below an internal reference, all control signals from the other (or remote) end are ignored.

References involving only the coupler card on the (local) bus where power remains up,(e.g. the BCP map when the memory bus is down) are unaffected. Furthermore, when a power-down sequence is detected by a (remote) BCM, the attached (local) BCP causes a level 1, device code 1 interrupt (external attention) on its (local) bus. The (remote) BCM does not assert data bits 13 and 8 (2100) on a read of its control register. The power-down sequence (the interrupt signal and blocking of the BCM data) occurs 2.5 ms before the power loss affects the remote bus.

5.10 Parity Generation/Checking

The calculated parity is called "Address XOR Data" parity or AXD parity for short.

AXD parity involves two parity bits for each 16-bit word, one associated with each 8-bit byte. Each parity bit is calculated as the exclusive-OR of the address parity and contents parity of the byte.

This parity function detects:

(1) one data bit in error (2) all data bits zero (3) all data bits one (4) one address bit in error

No parity checking is done for BBC transfers.

There are two basic parity schemes: **CLASSICAL** and **FEEDBACK**

When a processor accesses memory with classical parity (write source generated parity), the BCP on write accesses generates the parity bits. These bits are passed through the BCM and stored in memory along with the data. During a read access, the parity is

fetched from memory and passed through the BCM with the data to the BCP where the parity is checked.

In feedback parity, (write source checked parity) the parity always flows towards the BCP for both read or write access. Thus a read access is essentially similar to a read access with classical parity.

On a write access, however, a device on the memory or IO bus (the PAR for the Pluribus), computes the parity of the arriving data and address, stores it with the data, and sends it back to the BCP for checking.

The feedback parity scheme has the advantage of detecting some errors sooner, but at the cost of requiring a separate parity generator. In all cases, if the parity check discovers an error, the BCP issues a QUIT. Both halves of read-modify-write and lock access cycles are checked independently. If there is a parity error during the first half-cycle, a QUIT is issued immediately, and the write half-cycle does not take place.

The BCI performs exactly the same parity functions as a BCP. For couplers to memory busses that contain standard memories (from either processor or I/O busses) classical parity is usually used.

A bus coupler between the processor and I/O busses has an additional problem. Since the data read from a location on I/O bussess is different from that previously written (if any), the parity cannot be stored with the data. Hence another device (PAR) on the I/O bus, generates parity for all accesses on that bus.

With the coupler configured for feedback parity, all references are checked, at least through the coupler. An option jumper, provided on the BCM to delay the effective time of receipt of the DONE pulse, gives the PAR time for the parity computation. When a PAR is not used, the normal DONE may be used. Accesses to M/I busses involve several cases.

Small Pluribus systems may have both memory and I/O devices on the same bus (MI bus). A transfer between an I/O device and memory on an MI-bus can escape parity checking. To ensure parity checking MI

buses are configured with PAR cards. The usual MI-bus configuration involves setting the bus couplers (BCMs) for the feedback card, configuring them to generate parity on all access cycles except memory read cycles and then located on the M/I bus. As a result, a processor referencing I/O works in the same manner as if it were on a separate standard I/O bus; the memory looks to the processor as if it were a memory that supports feedback parity.

An option jumper is also provided on the BCP to disable parity checking. In this mode, all parity circuitry is enabled, but if an error is detected, it is ignored and a DONE returned.

Another jumper on the BCP allows a shorter access time if no parity checking is needed.

6 BCU - BUS CONTROL UNIT

BCU - 2102293G01

6.1 Purpose

The purpose of the BCU is to control who has use of the bus at any one time. When it is used on a processor bus it enables the connected processor, via a tophat, to receive system interrupts on priority levels 4 through 1, and to perform central processing functions.

6.2 Configuration

The following pages explain the strapping and options necessary to properly configure the BCU, depending on its application, I.E. processor bus, etc.

7 BXD/BXR

BXD - 2102295G01
BXR - 2102296G01

7.1 BUS EXTENDERS

7.1.1 Purpose

Bus Extenders are designed to increase the circuit card capacity of a bus by interconnecting two Infibusses. The following configurations are possible:

1. Using two 16-slot Infibusses, makes available 29 slots plus one slot for the BCU and two slots for the BXD/BXR combination.
2. Using one 16-slot and one 24-slot Inbibus, makes available 37 slots plus one slot for the BCU and two slots for the BXD/BXR combination.
3. Using two 24-slot Infibusses, makes available 45 slots plus one slot for the BCU and two slots for the BXD/BXR combination.

7.1.2 Description

A bus extension consists of one BXD card, one BXR card and two BEX cables that interconnect the BXD and BXR. The cables have a special design that suppresses noise. In an emergency one of the console cables may be used until the proper BEX cable arrives.

The Bus Extender Driver (BXD) is installed into the slot left of the last-occupied slot in the main chassis. The BXD is the last circuit card in the precedence chain of the main chassis.

The Bus Extender Receiver (BXR) is installed into the right-most slot in the auxiliary chassis. The BXR is the first circuit card in the precedence chain of the auxiliary chassis. Cards may be located in either chassis except in those for the BCU, Processors, and bus couplers, which are located in the main chassis.

Generally, devices that are accessed infrequently, such as the control panel, auto-load, and PSB are located in the auxiliary chassis.

The RTC/RTT is located in the auxiliary chassis in the last-occupied slot. This location is chosen to avoid the interference that would occur if the RTC was located on the main chassis.

7.1.3 Extended Lines

Extended bus lines are interconnected on a one-for-one basis. Signal designations at the cable drivers/receivers are prefixed by the letter C but are the same as those for the Intibus.

Most of the extender lines are bi-directional.A signal is driven and received on a single line by both the BXD and BXR. These include the address, data and mode control lines.

Three extender lines, Strobe (STRB-N), Half Cycle (HCYC-N), and Quit (QUIT-N) are driven and received on separate lines by both the BXD and BXR. Another twelve lines of the extender are driven only by the BXD and received by the BXR and ten lines are driven only by the BXR and received by the BXD. Power fail/restart line PRIN-N, PRAL-N, PRIN-N and LFIN-N are extended and contain no drivers or receivers in either the BXD or BXR.

7.1.4 Non-Extended Lines

Twelve lines on the Infibus are neither extended nor driven in the bus extender. These lines are not shown in Figure 1 and are listed here for reference only:

Infibus Line	Pin
BT1A-N	P1-A30
BT1B-N	P1-B30
BT2A-N	P1-A31
BT2B-N	P1-B31
BT3A-N	P1-A32
BT3B-N	P1-B32
BT4A-N	P1-A33
BT4B-N	P1-B33
PCDB-N	P1-B25
SELC-N	P1-B34
LFRQ-N	P1-B07
(Reserved)	P1-B50

7.1.5 Configuration

The following pages explain the necessary strapping to properly configure the BXD and the BXR boards for BBNCC applications.

8 LOAD DEVICES

8.1 ALD - AUTO-LOAD

ALD - 2102294G01

8.1.1 Purpose

The Auto-Load card is a module that provides automatic loading from the paper-tape reader or cassette. cassette.

The auto-load signal (ATLD-N) from the control panel causes the auto-load card to transfer the boot loader start address to the level one interrupt location (memory location 6). The ALD then generates a CPU interrupt to level 1. This interrupt places the device number in memory location 0000. The level one interrupt causes the CPU to execute the boot loader stored in ROM on the ALD.

8.1.2 Description

Figure 1 is a block diagram of the auto-load module.

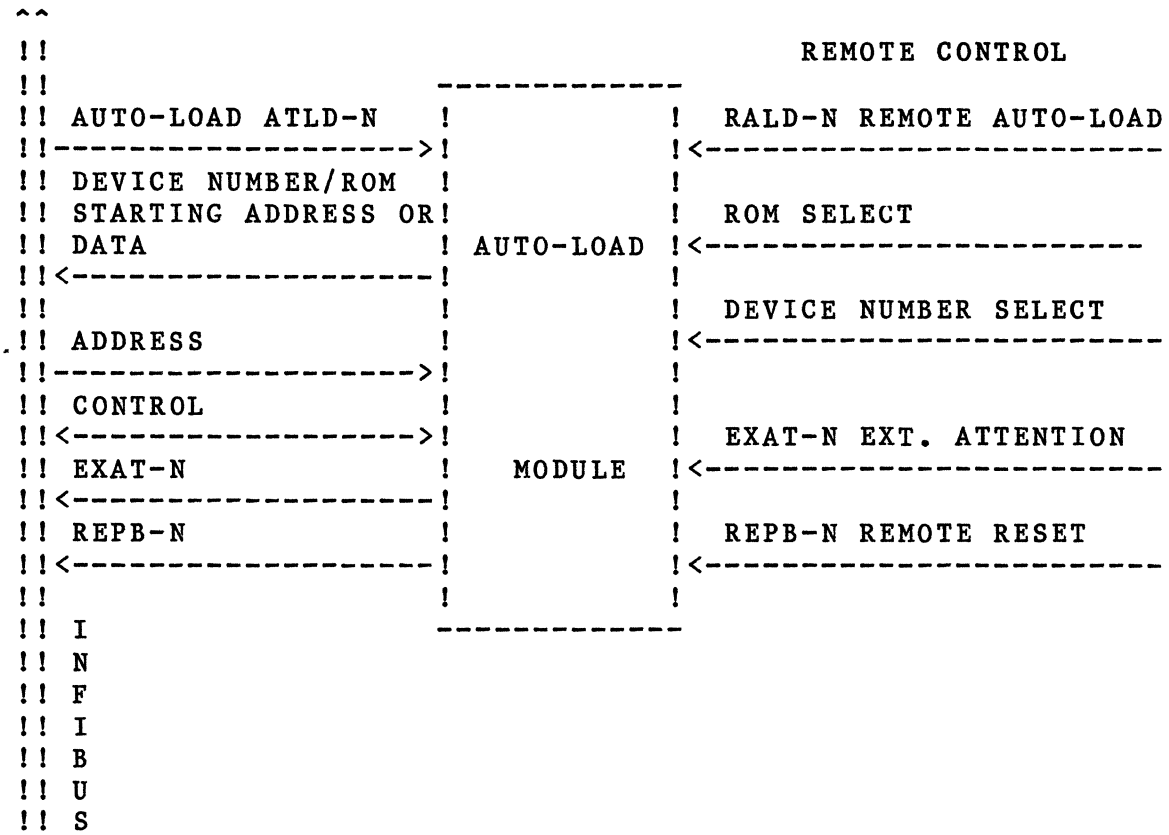


Figure 1. Auto-Load Module I/O Functions
Simplified Block Diagram

Two manual switches (S4 and S3) on the auto-load module can be set to select one of four device numbers (0, 2, 4, and 6) so that one ALD serves up to four devices of the same type.

Two other switches (S2 and S1) can be set to to select one of four possible bootstrap loaders (numbered 1, 2, 3, and 4). The auto-load module supports up to four different load devices (one bootstrap loader per device type).

8.1.3 ROM Configuration

The bootstrap loader routines are stored in ROM. The ROMs contain either 256 or 512 four-bit words (nibbles). Each nibble constitutes one-half of a byte with two bytes making a 16-bit word. A single loader program is stored in one ROM integrated circuit. Each 16-bit word occupies four consecutive nibble locations. The ALD logic unit makes four consecutive read operations to fetch each 16-bit word of the loader program. A 16-bit word is assembled in a register and transferred to the Infibus.

A read cycle in byte mode requires the transfer of two nibbles to the data register. The byte data is loaded into the least-significant into the least-significant eight bit locations (7 through 0) with the most-significant eight bits (15 through 8) set to zeros.

Either the left or right byte of a program word is fetched and stored in the least-significant eight bit locations of the data register and transferred on the Infibus. In all byte transfers, the most-significant eight bit locations of the word are all zeros.

8.1.4 External (Remote) Control

The contacts of the four manual switches are wired to an edge connector (J1) so that the device number and ROM can be selected from a remote control panel or other source. Other remote functions can be connected to J1 so that complete remote control is possible.

When the ROM or device is selected remotely, the associated switches on the auto-load module must be in the down (open) position.

8.1.5 Configuration

The following pages explain the strapping necessary to properly configure the ALD board.

8.2 PPB - PAPER TAPE CONTROLLER

PPB - 2102297G01

8.2.1 Purpose

The High Speed Paper Tape controller (Peripheral Parallel Buffer) allows the transfer of data from the paper-tape devices. The card can be configured to exchange data with several different types of paper-tape readers and punches. **Note:** This is an obsolete interface. Recently manufactured Pluribus use the serial interface and a cassette reader as a load device.

8.2.2 Description

Figure 1 is a general block diagram of the PPB. The seven blocks in the diagram represent functional logic groups that operate to receive and transmit input and output data between the Infibus and the I/O Devices.

8.2.3 Data Bus Drivers/Receivers & Multiplexer

The data bus drivers/receivers transmit and receive I/O data. Data received for output is applied directly to the output data register.

Input data is selected by the data bus multiplexer. One of four types of information can be selected for input:

(1) read data from the device
(2) control register contents
(3) device and controller status
(4) controller address (same as device number)

8.2.4 Input & Output Data Registers

The controller has two data registers: one for input, the other for output. Each register temporarily holds eight bits of I/O data for transfer between the Infibus and the I/O device.

Both registers have the same address. Therefore, the input data register can only be written. The data register address is FC32.

8.2.5 Control Register

The control register can hold up to six bits loaded from the six least-significant bits of the data bus. Three control bits are decoded, and used to activate read/write/interrupt control functions. The control register contents can be read through the data bus multiplexer by a Read Control Register operation. The Control Register address is FC36.

Bit 0 is the Start bit if writing and the Active bit if reading.
Bit 1 says output = 1
Bit 2, if set, allows interrupt.
Bit 3, if set, says Device Command and inhibits the Write Command.
Bits 4 and 5 are unused.

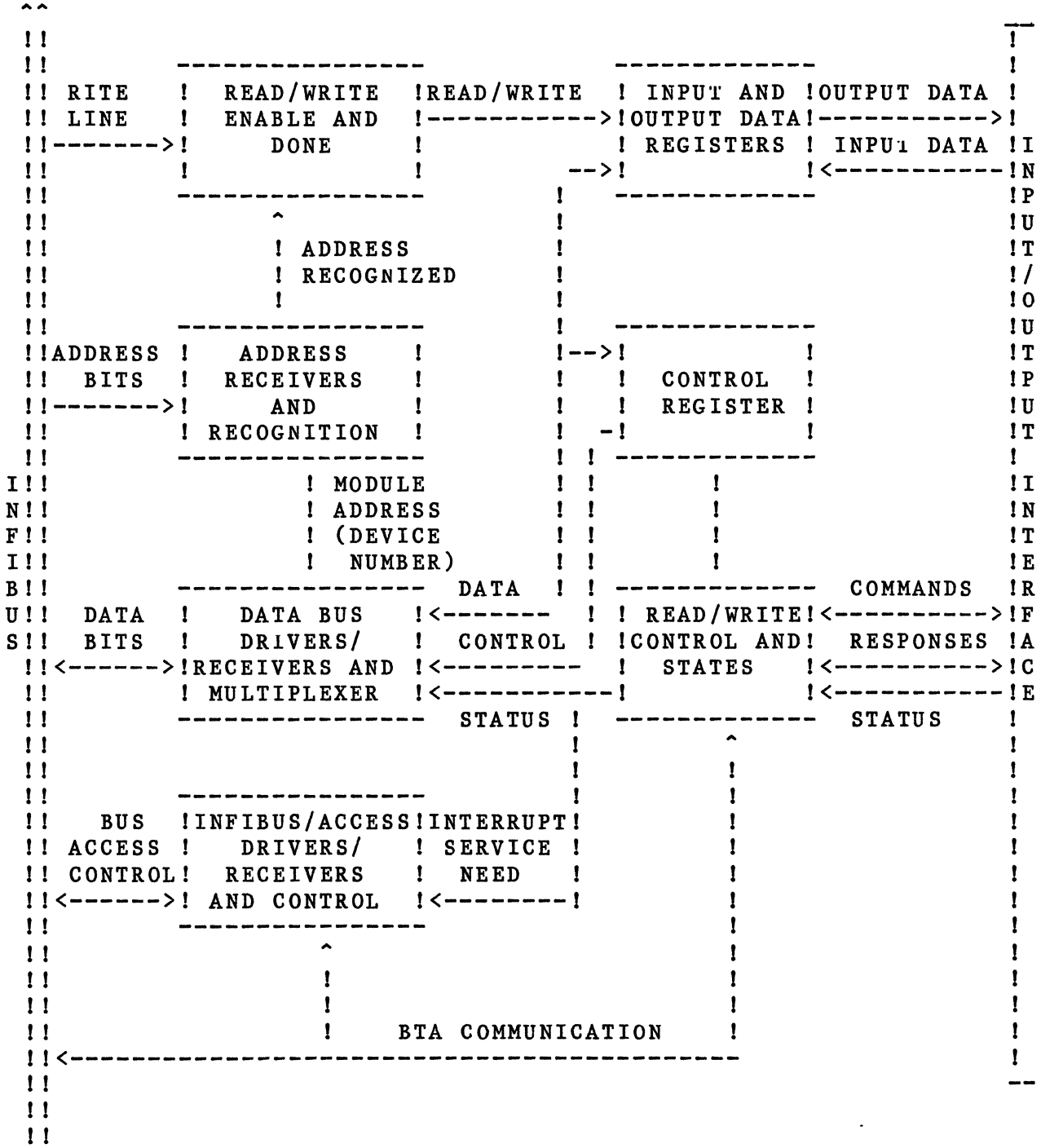
8.2.6 Read/Write Control & Status

The read/write control logic sends I/O commands to the device, receives device responses, and handles read or write control functions in the controller.

Status is read at FC30. One status bit (bit 2) is used for Read timing error or no data read (bit 0) after 300 milliseconds.

One device-not-ready (bit 1 for paper tape reader and bit 3 for a punch) status bit is used for each paper tape read or write device. Device-not-ready status bits are not stored in the controller but are DC signals that indicate operational condition. All

status is available for input by a Read Status Register operation.



8.2.7 Configuration

The following pages explain the strapping necessary to properly configure the PPB board.

8.3 PSB - PERIPHERAL SERIAL BUFFER

PSB - 2102300G01

8.3.1 Purpose

The PSB converts parallel to serial data and vice versa. This conversion takes place between the system and the cassette reader, or the system and Console Terminal, depending on which is connected to the system.

8.3.2 Description

The Peripheral Serial Buffer (PSB) is a single card that interfaces with an external terminal as the Console Terminal and/or the Cassette Reader. It can be configured to run on the Master processor bus (Cassette), I/O bus (Console Terminal) or on a single bus (Console Terminal).

8.3.3 Cabling

Located in the upper left hand corner, the cable from the terminal or the cassette connects to J1. The numbered pins can be accessed for scope or meter readings at the back of the board behind the connector. The following table lists the information located at each pin.

A1	SEND DATA (EIA)
B1	REQUEST TO SEND (EIA)
A2	
B2	
A3	
B3	SEND DATA + (CURRENT LOOP)
A4	
B4	SEND DATA - (CURRENT LOOP)
A5	TERMINAL READY (EIA)
B5	RECEIVE DATA IN + (CURRENT LOOP)
A6	
B6	RECEIVE DATA IN - (CURRENT LOOP)
A7	DATA SET RING INDICATOR (EIA)
B7	PTR BUFFER FULL + (CURRENT LOOP)
A8	RECEIVE DATA IN (EIA)
B8	PTR BUFFER FULL - (CURRENT LOOP)
A9	CLEAR TO SEND
B9	
A10	GROUND
B10	GROUND

8.3.4 Configuration

The following pages explain the necessary strapping to properly configure the PSB board.

8.4 RLD - RELOAD CARD

RLD - 2101529G09

8.4.1 Purpose

The ReLoaD (RLD) card allows a Pluribus system to be reloaded via a communications line. The RLD monitors input data streams of up to eight modem interfaces. The RLD supports any mix of modem interface or speeds from 4800 bits per second to 1.3 Megabits per second. type at any speed (4800 to 1.3 MegaBit).

The RLD scans this data stream for instructions which direct it to store data in the system. An activated RLD can write a 16-bit word anywhere in system address space.

8.4.2 Description

The RLD is a single card device that usually resides on an I/O bus. Write cycles use forward bus coupler access to common memory, and backward bus coupler access to the processor busses. Bus acceses are combined to build sequences that start or stop processors and load memory. The functions permit the remote loading of a program and begin program execution. The RLD should not be connected to a modem interface used as a VDH (very distant host) port.

8.4.3 Line Protocol

Commands to the RLD, consisting of a 20-bit address, a 16-bit data word, a 16-bit CRC checksum and four bits of padding, are located within the normal ARPANET IMP-to-IMP packet format.

Following the DLE STX packet beginning, the first bit of packet text (the low order bit of the first word in the buffer) is a 1 to indicate a reload command packet.

The rest of the command packet consists of twenty bits of address (the least significant bit occurs first); four bits of non-zero padding; 16 data bits (the least significant bits occur first); and a 16-bit checksum on the preceding elements using the IBM CRC-16 polynomial:

$$X^{16} + X^{15} + X^{2} + 1$$

The low order address bit must be a 0. The non-zero padding assures that an all-zero reload packet is not interpreted as a command by the RLD.

Except for the first bit of the command in a packet, each of these 56 bits is doubled to increase the uniqueness of the reload packet and to guarantee that the DLE character (020) does not occur in the reload data stream. If it does, it must detected and undoubled.) As a result, each individual reload command requires seven 16-bit words of storage at the source machine.

Any number of commands are included in a reload packet subject only to the packet size restriction of the IMP software. A command string may be cleanly terminated by the normal DLE ETX sequence.

8.4.4 Device Operation

Three signals are supplied by each modem interface connected to the reload card. Two signals, the receive clock and receive data, represent the actual data stream from the neighboring IMP. The third signal is a synchronization pulse indicating the DLE STX was received. This pulse occurs only once per packet and immediately before the text of the packet.

When the pulse is received, the reload card front end (one of eight) looks for the next bit to be a 1. If it is, the received packet is a reload command packet. If no other line is selected, the active latch for that line is set. The outputs of the eight active latches drive an asyncchronous priority encoding arbiter that permits only one of the eight inputs to be

active at a time.

First, the OR of these latches disables further input and then starts a timer. Meanwhile the outputs of the latches are applied to a priority encoder which selects one of the set latches.

The encoder outputs feed multiplexors to select the corresponding data, and clock lines for further processing. From this point, the input is processed by a finite state machine. Each second bit is checked to be sure that it is identical to the preceding bit (except for the first low-order address bit which is checked for zero). The finite state machine output is the undoubled bit stream.

The undoubled bit stream shifts into two registers, one 40-bit data register, and one 16-bit CRC register. The two shift registers shifts in the next 40 bits. Thereafter, the 40-bit register stops, while the CRC continues on for 16 more bits. During these sixteen bit times, the CRC feedback point is examined for all zeroes to be sure the checksum is good.

If no error conditions are present at the conclusion of the checksum, (see below) an INFIBUS cycle is requested, using the conventional method, in order to write the data (the high order 16 bits of the 40-bit register) into the specified address (the address requested by the low order low 20 bits of the 40-bit register).

The address and the data are double-buffered to maximize the time available to access the bus. Once the checksum has been checked, the process is repeated in order to examine the data stream for another command. This process repeats as long as no exception conditions are detected. Any one of the exception conditions below terminates command string processing, clears the priority encoder latches, and permits new activations from any of the modem data streams.

The exception conditions that cause termination are as follows:

1. Bad checksum - the present command is ignored, but the last good command is completed.

2. Padding not matching jumpered identifier on the card as above.

3. Non-doubled data - if the second bit of a pair differs from the first, the last good command will be completed before termination. This property can be used to bring about clean termination of a command string merely by ending the packet with the usual DLE ETX. The single 1 in the DLE will cause termination after the last command, which will access normally.

4. Incomplete bus access - if the bus cycle for the previous command is not completed prior to the end of the checksum for the following command, the command string will be terminated. Although a new activation is allowed prior to completion of the access for the previous command, the new command will terminate as above if the access has not yet completed.

5. Watch Dog Timer Timeout - if there has been no new line selection for about 1 second, the current command is ignored.

8.4.5 Modem Conection

Each modem supplies the three signals described above over a twisted pair by using the 9614/9615 differential driver/receiver pair.

The modem data line is high for a data 1, while the low to high transition of the modem clock line indicates good data. A positive going pulse indicates the beginning of a packet.

Clock and data signals are derived from the output of the interface receivers for those lines from the modem. These signals are carried to the device via one 50-pin edge connector using a special twisted pair cable (IRLD) that fans out to a 16-pin DIP plug for each modem connection.

An additional gate in the modem connects to the reload card, by means of a jumper in the DIP plug. Since all these signals are derived in the modem, any

speed line with any modem may be connected to the RLD. Plugging an IRLD connector into a modem turns on the high-order bit of that modem's device number.

8.4.6 Infibus Connection

Subject only to physical cabling constraints, more than one RLD can be connected to a given line or set of lines using multiple modem interfaces on the same line.

The RLD devices request a given access almost simultaneously, but the resulting conflicts are resolved by the arbiter on the target bus, and do not affect this device. Since duplicate accesses are always completed prior to any subsequent commands, no harm is done.

8.4.7 Indicators

Three lights signal device operation. The first latches an activation and remains lit until the next master is reset. The second remains lit only while a line is currently active. The third lights briefly following bus access completion.

8.4.8 Reload Connector Pin Assignments

J1-1	MCK00+	modem 0 receive clock
J2-1	MCK00-	
J1-2	MDT00+	modem 0 receive data
J2-2	MDT00-	
J1-3	MG000+	modem 0 start of packet
J2-3	MG000-	
J1-4	MCK01+	modem 1 receive clock
J2-4	MCK01-	
J1-5	MDT01+	modem 1 receive data
J2-5	MDT01-	

J1-6	MG001+	modem 1 start of packet
J2-6	MG001-	
J1-7	MCK02+	modem 2 receive clock
J2-7	MCK02-	
J1-8	MDT02+	modem 2 receive data
J2-8	MDT02-	
J1-9	MG002+	modem 2 start of packet
J2-9	MG002-	
J1-10	MCK03+	modem 3 receive clock
J2-10	MCK03-	
J1-11	MDT03+	modem 3 receive data
J2-11	MDT03-	
J1-12	MG003+	modem 3 start of packet
J2-12	MG003-	
J1-13	MCK04+	modem 4 receive clock
J2-13	MCK04-	
J1-14	MDT04+	modem 4 receive data
J2-14	MDT04-	
J1-15	MG004+	modem 4 start of packet
J2-15	MG004-	
J1-16	MCK05+	modem 5 receive clock
J2-16	MCK05-	
J1-17	MDT05+	modem 5 receive data
J2-17	MDT05-	
J1-18	MG005+	modem 5 start of packet
J2-18	MG005-	
J1-19	MCK06+	modem 6 receive clock
J2-19	MCK06-	
J1-20	MDT06+	modem 6 receive data
J2-20	MDT06-	
J1-21	MG006+	modem 6 start of packet
J2-21	MG006-	

J1-22	MCK07+	modem 7 receive clock
J2-22	MCK07-	
J1-23	MDT07+	modem 7 receive data
J2-23	MDT07-	
J1-24	MG007+	modem 7 start of packet
J2-24	MG007-	
J1-25		unused
J2-25		

8.4.9 Modem Conector Pin Assignments

16	MCK-	modem receiver clock
15	MCK+	
14	MDT-	modem receive data
13	MDT+	
12	MGO-	modem start of packet
11	MGO+	
10		
jumper		
9		

8.4.10 Configuration

The following pages explain the strapping necessary to properly configure the RLD board.

9 DMA - Direct Memory Access

DMA - 2100136G01

9.1 Purpose

The Direct Memory Access (DMA) module performs general full-duplex INFIBUS communication as part of a device interface (e.g., Host, modem). The usual configuration requires that one or more of the cards adjacent to the DMA card contains a specialized logic for the particular device interface (DI). Control signals to and from the DMA are passed via a back plane connector (IBM, ICM, IDM).

9.2 Infibus Interface

9.2.1 Address and Registers

The 8 registers associated with each device are as follows:

000 device type
001 receive begin
010 receive end
011 receive status
100 transmit begin
101 transmit end
110 transmit status
111 device dependent

The register formats are described individually as follows:

NOTE

All eight registers can be read or written; the designations in the individual description denote only the meaningful actions.

000 DEVICE TYPE - READ ONLY

The Device Type-Read Only register contains the contents of an 8-bit manual switch register on the DI in the low-order byte, and an 8-bit number denoting the

type of device interface in the high-order byte.

001, 100 - RECV/XMIT BEGIN - WRITE ONLY

The high-order 16 bits of the system word address are placed in this register at the beginning of the data buffer to be used. The 3 low-order bits of the begin word address are set via the status register (see below).

010, 101 - RECV/XMIT END - READ/WRITE

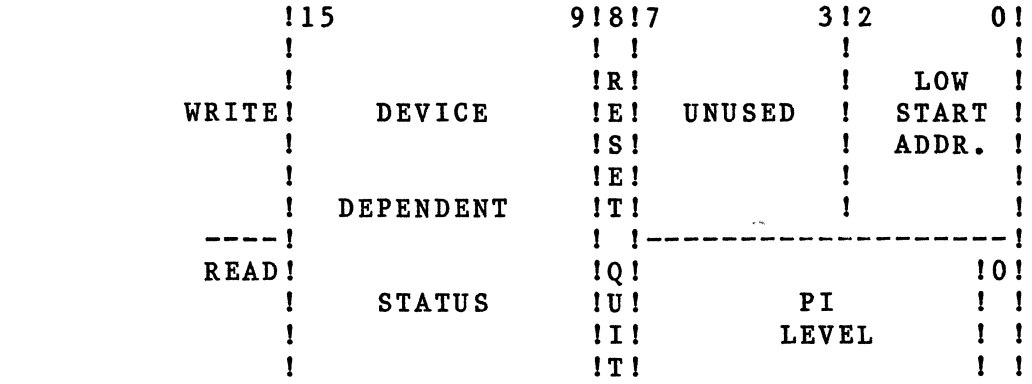
These registers are loaded with the 12 low-order bits of the system word address at the end of the buffer. The maximum buffer size is 4K words. The address should be given in bits 1-12 (hex 1FFE). The high order bit, set when writing the end pointer, indicates that this is the last packet of a message. Bits 0, 13, and 14 (hex 6001) are ignored.

Writing to this register initiates a buffer transfer. When the DMA completes the buffer transfer, a read obtains the address modulo 2**12 of the last transferred word using the same format described above.

The low-order bit serves as an error bit. This bit reports an error if the attached device interface is reporting an error, or if a QUIT occurred during the transfer of this block.

The high-order bit, if set, signifies that this is the last packet of a message, and that a QUIT has not occurred (indicating a good completion).

011, 110 - RECV/XMIT STATUS - READ/WRITE



(Note that there are two of these registers; these functions exist independently for receive and transmit.)

DEVICE DEPENDENT STATUS - READ/WRITE

These are used for direct two-way communication by a processor with the device interface, e.g., crosspatch, Host dead, loop.

RESET - WRITE

Setting this bit while writing resets that half of the DMA and DI.

LOW START ADDRESS - WRITE

The low-order three bits of the start pointer are loaded with this number each time the start pointer is written. Writing the status register prior to writing the begin pointer sets up this procedure.

QUIT - READ

This function is set when the DMA's last data access or interrupt request is terminated with a QUIT, indicating memory malfunction, non-existent address, etc. When the end pointer is read, error not completion is set.

PI LEVEL - READ

The 7-bit pseudo-interrupt code written to the PID on command from the device interface (upon buffer completion) is set by switches on the DI.

111 - DEVICE DEPENDENT (OPTIONAL) - READ/WRITE

This register can be used optionally by the DI for any appropriate device specific function. Assignment of data bits is arbitrary; the DI receives and transmits the data directly from and to the INFIBUS. The DMA merely supplies signals indicating when this register is read from or written into.

9.3 Device Interface

There are 61 signals by which the DMA communicates with the device interface (DI); 14 for transmit, 15 for receive, and 32 used for both transmit and receive.

All signals are TTL standard levels, positively asserted (i.e., HIGH = "1" = TRUE) unless otherwise stated. The fanout or fanin, as appropriate to each signal, is given in brackets. Pin assignments for the backplane connector are given in the attached table.

BACKPLANE (IxM) CONNECTOR ASSIGNMENTS

Pin	A	B
1	GROUND	GROUND
2	XMIT START+	RECV START+
3	XMIT READY+	RECV READY+
4	XMIT NOW+	RECV NOW+
5	XMIT DOINT+	RECV DOINT+
6	XMIT INTDN-	RECV INTDN-
7	XMIT EOB+	RECV EOB+
8	XMIT QUIT-	RECV QUIT-
9	XMIT STBST+	RECV STBST+
10	XMIT GVST+	RECV GVST+
11	XMIT RESET+	RECV RESET+
12	XMIT ERROR+	RECV ERROR+
13	XMIT RSTRT+	RECV RSTRT+
14	GVDT+	RECV LASTP+
15	ST08-	ST12-
16	ST09-	ST13-
17	ST10-	ST14-
18	ST11-	ST15-
19	ST00-	ST04-
20	ST01-	ST05-
21	ST02-	ST06-
22	ST03-	ST07-
23	XMIT GVPIL+	RECV GVPIL+
24	DDPRD+	DDPWR+
25	GROUND	GROUND
31	GROUND	GROUND
32	ADS04-	ADS09-
33	ADS05-	ADS10-
34	ADS06-	ADS11-
35	ADS07-	ADS12-
36	ADS08-	ADS13-
37	PIA11-	PIA12-
38	MRES+	unused
39	XMIT RBEG+	RECV RBEG+
55	GROUND	GROUND

The following lines are common to receive and transmit:

ST00-15 (DI>DMA) [-2]

These sixteen (16) lines are used to send status information from the DI upon command. They are open collector lines pulled up by 1K resistors in the DMA. They are negatively asserted; a low logic level will be reported as a one to the processor.

ADS04-ADS13 (DI>DMA) [-1]

The DI continuously broadcasts the sense of these signals, which determine the 10-bit device number used for address recognition. They are typically the output of a manual switch. These signals are negatively asserted; logic high implies an address bit of zero.

PIA11-PIA12 (DI>DMA) [-1]

The DI continuously broadcasts the sense of these signals to determine to which of 4 PIDs the pseudo-interrupts should be directed. These are typically determined by a manual switch. The signals are negatively asserted; logic high implies an address bit of zero.

GVDT (DMA>DI) [6]

This level instructs the DI to place on ST00-15 its device type code. Typically bits 08-15 are unique numbers assigned to each type of DI; bits 00-07 are the contents of a manual switch on the DI.

MRES (DMA>DI) [9]

A 500 microsecond pulse that occurs during system reset (e.g., startup).

DDPWR (DMA>DI) [6]

This pulse (500 ns) indicates that the device dependent register is being written. If the DI uses this option, it should sample the data directly from the INFIBUS on the leading edge of this signal.

DDPRD (DMA>DI) [6]

This level indicates that the device dependent register is being read. If the DI uses this option, the appropriate data should be gated onto the bus directly by this signal.

The transmit lines are as follows:

START (DMA>DI) [5]

The DMA sends this pulse (500 ns) when an end pointer has been written. The device should start and may request accesses.

READY (DI/DMA) [-3]

This is brought up by the DI to request a data access. It must remain true until the leading edge and, must be cleared before the trailing edge of NOW. If raised after DOINT and before INTDN, access order is not guaranteed.

NOW (DMA>DI) [10]

This pulse is used by the DI to strobe in the next data word from the data lines on the INFIBUS. On transmit it is 40-60 ns long; data should be strobed on the leading edge.

DOINT (DI>DMA) [-2]

This pulse requests that the DMA write the PI level to the PI locations, causing a pseudo-interrupt. It must remain true until the leading edge and, must be cleared before the trailing edge of INTDN. DOINT may not be raised while READY is true.

INTDN (DMA>DI) [10]

This negative going pulse (40-60 ns) signifies that the interrupt has been performed and the DI should now clear DOINT>

RBEG (DMA>DI) [10]

This pulse (500 ns) indicates that the begin register locations is being read. The DI may use this for its own purposes as it is ignored by the DMA.

EOB (DMA>DI) [10]

This level signifies that the current access references the last word in the buffer. It becomes true while READY is true, and remains true until a new begin pointer is written.

NOTE

On a one-word buffer (end=start), EOB becomes true as soon as the end pointer is written (START). On successive one-word transfers to the same location, EOB will never go false.

The DI should not request any further accesses after the current one (which will complete normally), or until the next START.

The DI should also use EOB as a stimulus to later request an interrupt (note that DOINT may not be raised while READY is true).

QUIT (DMA>DI) [9]

This negative going pulse (80-100 ns) indicates that the last data access or interrupt attempt referenced a non-existent location or was otherwise timed out.

The suggested protocol for a QUIT is as follows: If the QUIT is in response to a data request, READY should be cleared if set, and an interrupt requested (in that order).

If the QUIT is in response to an interrupt request, DOINT should be cleared and an interrupt is not requested. In either case, no other accesses should be requested until the next START.

STBST (DMA>DI) [4]

This pulse (500 ns) should be used by the DI to strobe in status information from the data lines on the INFIBUS. Data should be sampled on the leading edge.

GVST (DMA>DI) [4]

This level instructs the DI to place its device dependent status bits on ST09-15.

GVPIL (DMA>DI) [10]

Upon assertion of this level, the DI presents a 7-bit switch-selectable pseudo-interrupt level on ST01-07. GVPIL sometimes happens simultaneously with GVST, sometimes not.

RSTRT (DMA>DI) [5]

This 300 ns pulse informs the DI that a new begin pointer has been written.
The DI might use this to abort the current transfer with some error indication (e.g., append a bad checksum) and prepare for a new one. In that case, any current request for a data access or interrupt will be

aborted. If a transfer was in progress, an interrupt would be requested.

RESET (DMA>DI) [9]

Either a 500 ns or 500 microsecond pulse that should reset that half of the DI.

ERROR (DI>DMA) [-1]

A level asserted by the DI to indicate that some error condition (defined in a device dependent manner) has occurred. The DMA will report this by setting the low-order bit in the end pointer when read.
(This bit is also set by the DMA if a QUIT is received during the transfer of the block.)

The receive lines are identical except for the following:

1. NOW is used to indicate when the data word should be placed on the bus. To satisfy timing constraints, the data should be present at the BDR inputs when READY is brought up. NOW should go directly to the BDR enable inputs.

2. There is another signal LASTP (DI>DMA) [-1] which is a level returned by the DMA as the high-order bit of the end pointer when read to indicate that this was the last packet (End of Message). The DI should not assert this line if an error is detected, even if an end of message has been encountered.

   The high-order bit of the end pointer will similarly not be set (whatever the sense of LASTP) if the DMA has received a QUIT during the transfer of the block.

   The convention for denoting last packet on transmit is that the high-order bit of the end pointer is set when written. The DI should therefore sample the high-order data bit on the bus on the leading edge of transmit START to get this information.

9.4 Implementation

Typical circuits for pieces of the DI most intimately connected with the DMA are shown in Figures 1 and 2.

9.5 Switches (on DI)

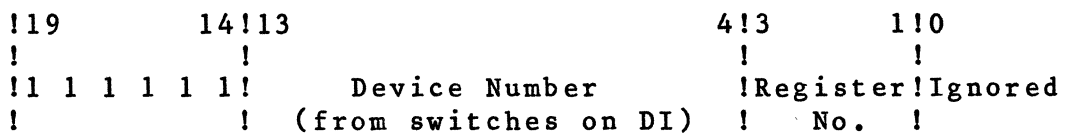
Device address (10) - selects address of device

Recv PI Level (7) }
} pseudo-interrupt codes
Xmit PI Level (7) }

PI Address (2) - selects which PID is being written to

Device type (8) - readable switch register

9.6 Addresses



9.7 DMA Logic Description

Address Recognition

The DMA recognizes a block of eight words. The high order 6 bits must be ones, and the next 10 bits are determined by switches on the device interface (DI). Lines from these switches (ADS04-ADS13) are EX-ORed with address lines 4-13 and the results wire-ANDed to form SWMAT+.

STRB+ is delayed by passing it through similar gating to form STRBD+. The latter is ANDed with HOLD- (to inhibit address recognition until the fall of HOLD) to form CLKAD which strobes the state of ADMAT into ME. The ME flop is operated inverted; ME+ is asserted when

the Q output is true. The rise of ME starts a 420 ns delay (MEPLS) before issuing a 50 ns DONE pulse (DONET). ME is cleared by the fall of undelayed STRB+.

The three low order address bits (not counting the byte bit, which is ignored) are sent to two demultiplexers (9334) to derive individual signals for writing and reading each of the individual registers. The data inputs are both senses of RITE, so one 9334 responds only to WRITEs and the other only to READs.

The WRITE 9334 operates as a strict demultiplexer, enabled by MEPLS. The READ 9334 also demultiplexes, but latches its state until the fall of ME+; thus its outputs are valid until the fall of STRB+.

Outputs from the 9334 are named:

```
                {          { DEVT  (device type)      } }
                {          { DDP   (device dependent) } }
{ R (read)  }{                                          }
{ W (write) }{                                          }
                { { R (recv.) }   { BEG (begin ptr.) }  }
                { { X (smit.) }   { END (end ptr.)   }  }
                {                 { ST  (status)     }  }
```

Several of these signals are passed directly to the DI as the appropriate control lines.

For example, writing the begin pointer (WRBEG, WXBEG) pulses the RSTRT lines. Writing the end pointer (WREND, EXEND) pulses the START lines. Writing or reading the status (WRST, WXST, RRST, RXST) causes the appropriate GVST or STBST line to be asserted.

If data bit 8 is set when writing the status (signifying a programmed reset) RSTRS or XSTRS will be asserted. These are ORed with MRES to form RRES and XRES. Accesses of the device dependent register (RDDP, WDDP) and reads of the begin pointer (RRBEG, RXBEG) are sent to the DI and otherwise are ignored by the DMA.

POINTERS

The pointers for receive and transmit sides are handled similarly. In the following, the receive half will be discussed; transmit is identical -- signal

names are obtained by substituting an X for an R.

The "current" pointer, which holds the word address of the location either previously or just being accessed by that half of the CMA, is stored in a 19-bit synchronous counter (RC01-RC19). The carry input to stage 13 is disabled; therefore the high order seven bits of the counter (13-19) act only as a latch and do not change when the counter is toggled. This counter is parallel-loaded when the BEGIN pointer is written (WRBEG).

The high order 16 bits come directly from the data bus (DB01-DB12). The low order 3 bits are specified by the low order 3 bits of the status word and are latched as RLC01-RLC03 by WRST. The counter is set for count-up operation.

The 12 bits of end pointer information are latched from the data bus by WREND. The outputs from the end pointer (RE01-RE12) are compared with the low order 12 bits of the current pointer (RC01-RC12). The output of the equals detect are wire-ANDed together to form the EOB signal.

CONTROL

.pg Bus access is controlled by two DBALs; one each for receive and transmit. In each DBAL, the A half controls data access and the B half handles pseudo-interrupt requests.

The command inputs are connected directly to the DI's READY and DOINT lines, which signify that it wants a data or interrupt request, respectively. The answering signal to READY, NOW, is given for the transmit side coincident with ACLK telling the DI to strobe-in the data which the DMA has read for it; on the receive side, NOW follows AONL, telling the DI to output its data to the bus for the write cycle the DMA is performing. Receive NOW is delayed until the fall of the current PCDA so that a quick fall of READY in response to NOW will not allow a little PCDA to leak out.

The response to DOINT< INTDN, is given at BCLK time. The DBALs are connected to the bus in the usual way, in parallel. QUIT and DONE are buffered to provide the necessary fanout to drive two DBALs. The

precedence pulse (PCDA-PCDAB-PCDB) is chained serially through the DBALs.

The DBAL is reset either by the reset signal for that half (RRES/XRES), or when the begin pointer is written. If the latter occurs in the normal course of events, viz., after an interrupt has occurred and the program is setting up for the next buffer, no bus transactions will be in progress and reset will do nothing.

However, if the begin pointer (to perform the restart function) is written while the device is active, the desirable feature of stopping any transaction in progress is accomplished.

The QUIT latches (RQUTL/XQUTL) remember that a QUIT has occurred during an access. They are reset by reset or the writing of the begin pointer for the next buffer. Writing the begin pointer also sets the corresponding "first" flop (RFRST/XFRST), thus preventing the first access from toggling RCLK/XCLK.

At its conclusion (the trailing edge of NOW/DOIT), the flop is reset and CLK follows READY. RITE is latched from the bus by STRB. RITE is asserted on the bus by the DMA (RITET) when on-line for either a receive access (RONLN) or an interrupt access (DOINT).

DATA BUS

The DMA contains a 16-line, open collector, negatively asserted bus (DAB00-DAB15). These wires are passed to the DI (ST00-ST15) and are used to pass non-data information to the DMA, e.g., device type, interrupt levels, status.

The other primary source is the pointer Multiplexer, which multiplexes the two current pointers and converts to open collector form. The multiplexer is enabled onto the DAB lines either when on-line for a data transfer (DATOL) or when reading either end pointer (RDEND) (only the low order 12 bits). In the former case we are broadcasting the address to be used for the data transfer.

The DAB lines in this case go through the address multiplexer to the address BDRs. Note that bits 16-19 go directly to the multiplexer since they are not used elsewhere.

If the end pointers are being read, the data BDRs are enabled (DDREN from SLVRD) and the address BDRs (ONLN) are not -- thus the information goes on the data lines.

A situation similar to the latter occurs when reading status or device type, except that the DI provides the source. During an interrupt access (DOINT), the address multiplexer switches so that the prewired (except for two bits from the DI) PID address is connected to the address BDRs. The DI supplies the PI level via the DAB lines which are broadcast on the data lines, since DOINT is a term in DDREN.

SPECIAL STATUS BITS

The error bits (RERR/XERR) are passed from the DI. The sense of the QUIT latch is ORed in and when the end pointer is read (RREND/RXEND) it is connected to bit 0. The last packet bit (LASTP) is also passed from the DI. If there is no QUIT on the receive side (RQUTL), completion will be reported (bit 15 set) when the end pointer is read.

The sense of the QUIT latches is reported as bit 8 when the status words are read (RRST/RXST).

The signal GVPI asking for the DI to present the appropriate PI levels on the ST lines is asserted when status is read (RRST/RXST), or when an interrupt access is on-line (RBONL/XBONL).

9.8 Configuration

The following pages explain the strapping necesary to properly configure the DMA board.

10 HOST PORTS

10.1 HLC - Local Host Interface

NOTE: This device is obsolete and is replaced by the HST card. The HST is backward compatible for the HLC.

10.1.1 Purpose

The Compatible Local Host (HLC) module, together with a DMA, interface a Pluribus system to a Local Host Interface using the protocal established in the BBN Report #1822, "Specifications for the Interconnection of a Host and an IMP."

The cable distance from HLC to Local Host Interface should be less than 50 feet. Interconnection to the Host is via a 26-pin Scotchflex connector. Jumpers select the use of the ready line and the type of input end of message padding. Interconnection to the DMA is via a standard IDM.

The required interconnection with the INFIBUS, including the input and output data paths, is directly through the bus connector. The module consists of a single card.

10.1.2 Host Connections

An interface must be compatible with the present local and distant Host interfaces. Under program control, for test purposes, the interface is loopable towards the IMP.

All control and data lines operate in a manner similar to the host ports of the early Honeywell IMPS. The IMP and Host Ready lines are the exceptions and do not operate in this similar way. Jumpering makes it possible to ignore ready lines entirely, or allows the IMP ready line to be on or off without a timing device.

The electrical characteristics of the drivers and receivers are compatible with drivers and receivers currently interfacing to Honeywell IMPs. Receivers are high impedence Schmitt trigger input inverters. Discrete drivers, designed for 68 ohm lines, allow two HLC boards to be connected to the same Local Host Interface. A timeout circuit provides for only one of these to be active at any given time.

10.1.3 DMA Connection

Communication with the DMA follows the rules in the DMA section. Transmit and receive pseudo-interrupts are set by:

1) end of message
2) writing a begin pointer while active (restart)
3) reset while active
4) QUIT while active
5) looping or unlooping while active.

If a QUIT occurs while attempting to pseudo-interrupt, the attempt is abandoned.

Internal looping (crosspatching) of the interface is done by setting the appropriate status bit. Both sides of the interface will reset and simultaneously change the state from a looped to an unlooped one. If either or both sides are active, an interrupt will occur on the busy side(s). Looped status indicates the interface is in fact looped internally. Master Reset or writing zero in that status bit will unloop the interface.

Busy (active) status, indicating that a transfer is in progress, is turned on by writing the end pointer to the DMA (start pulse from the DMA to the HLC). It is turned off by reset, master reset, restart, or end of message. IMP Ready status is true if the program has set IMP Ready previously, and if the driver timeout circuit has not timed out.

When the interface is appropriately jumpered (READY enabled), the IMP ready relay to the Host will be closed if the HLC is unlooped and IMP Ready is true. IMP Ready is cleared either by writing zero in that

status bit or by Master Reset. IMP Ready is also cleared when the driver timeout occurs. This timeout is temporarily prevented by writing to either the transmit or to receive status word at least once a second.

When the HLC is looped, Host Ready status is true if IMP Ready is true. If not looped, depending on the jumpering selected, it is true either when the interface is unlooped (READY disabled) or if the Host is ready (READY enabled). A Host error results when Host Ready goes false while the interface is not looped, or IMP Ready goes false, regardless of loop state. Host error status is reset when input is started. The receive DMA error line is true when a Host error has occurred.

A separate status is presented when a Host error with end-of-message and without end-of-message occurs. Each of the error lines is true while the corresponding busy status is true or if an interrupt is caused by any form of reset. Start clears the interrupt error condition.

10.1.4 Configuration

The HLC is obsolete and has been replaced by the HST card. If the HLC card is faulty, replace it with an HST. If an HLC is replaced with an HST, the end cap and cable must also be replaced.

10.2 HST - Host Interface

HST - 2100538G01

10.2.1 Purpose

The Host Interface (HST) module, together with a DMA, interface a Pluribus system to a Host Interface using the protocal established in BBN Report #1822, "Specifications for the Interconnection of a Host and an IMP". Either a local or distant host connection is used.

The cable distance from HST to a local Host Interface should be less than 50 feet. Cable runs of up to 2000 feet are possible using the distant host connection.

Connection to the Host is via a 26-pin Scotchflex connector. Jumpers select the use of the ready line, the type of input end-of-message padding, and local vs. distant drivers and receivers. Connection to the DMA is via a standard IDM. The required connection with the INFIBUS, including the input and output data paths, is directly through the bus connector. The module consists of a single card.

10.2.2 Host Connection

The interface must be compatible with present local and distant Hosts interfaces. Under program control, for test purposes, the interface is loopable towards the IMP.

All control and data lines operate in a manner similar to the to the early Honeywell Imps. The IMP and Host Ready lines are exceptions and do not operate in this similar way. Jumpering makes it possbile to ignore both ready lines entirely, and allows the IMP ready line to be on or off without a timing device.

The electrical characteristics of the drivers and receivers are compatible with drivers and receivers used with the early Honeywell IMPs. The drivers and receivers allow two HST boards to be connected to the same Special Host Interface. A timeout circuit allows only one of these to be active at any given time.

10.2.3 DMA Connection

Communication with the DMA follows the rules in the DMA section. Transmit and receive pseudo-interrupts are set by:

1) end of message
2) writing a begin pointer while active (restart)
3) reset while active
4) QUIT while active
5) looping or unlooping while active.

If a QUIT occurs while attempting to pseudo-interrupt, the attempt is abandoned.

Looping (crosspatching) of the interface is done by setting the appropriate status bit. Both sides of the interface will reset and simultaneously change the state from a looped to an unlooped one. If either or both sides are active, an interrupt will occur on the busy side(s). Looped status indicates the interface is in fact looped internally. Master Reset or writing zero in that status bit will unloop the interface.

Busy (active) status, indicating that a transfer is in progress, is turned on by writing the end pointer to the DMA (start pulse from the DMA to the HST). It is turned off by reset, master reset, restart, or end of message.

IMP Ready status is true if the program has set IMP Ready previously and if the driver timeout circuit has not timed out. When the interface is appropriately jumpered (READY enabled), the IMP ready relay to the Host will be closed if the HST is unlooped and IMP Ready is true. IMP Ready is cleared either by writing zero in that status bit or by Master Reset. IMP Ready is also cleared when the driver timeout occurs. This timeout is held off by writing to the transmit status word at least once a second.

When the HST is looped, Host Ready status is true if IMP Ready is true. If not looped, depending on the jumpering selected, it is true either when the interface is unlooped (READY disabled) or if the Host is ready (READY enabled).

A Host error results when Host Ready goes false while the interface is not looped, or IMP Ready goes false, regardless of loop state. Host error status is reset when input is started. The receive DMA error line is true when a Host error has occurred. Status is presented separately for the occurrence of a Host error with end of message and without end of message.

Each of the error lines is true while the corresponding busy status is true or if an interrupt is caused by any form of reset. Start clears the interrupt error condition.

There are a number of reset conditions: **Master Reset** Master Reset clears both halves of the interface, shuts off IMP Ready, and unloops the interface. **Transmit Reset** and

Transmit Restart clear the transmit side and set a pseudo-interrupt if it has been active. **Receive Reset** and **Receive Restart** are analogous. Looping or unlooping the interface will generate a reset to both sides. **Last Packet** is set on receive end of message only if error is not set.

10.2.4 INFIBUS Configuration

The input and output data paths are through the INFIBUS. Aside from the data drivers and receivers and power no other connection is made to the INFIBUS.

Data sampling and driving is according to the DMA rules. The high order bit is sampled on Transmit Start to capture last packet information.

The card device type is wired to be 02.

10.2.5 Configuration

The following pages explain the strapping necessary to properly configure the HST board.

11 MODEM PORTS

11.1 MLR-MLX Low Speed Modem Interface

NOTE:

This interface is obsolete, therefore the part numbers are not specified. The current modem interface uses the MHX-MUR circuit cards.

This interface consists of an MLR, MLX and a DMA, and permits communication between a Pluribus system and a Bell 303 modem. This module supports the synchronous line protocol used by ARPANET up to speeds of 250 Kb. The interface is programmed like any other DMA device; special features and unusual control sequences are discussed below.

Facilities are provided both to crosspatch the interface before the 303, so that it receives its own transmitted data, and to loop data back through the 303. This permits testing both the interface the the 303 under program control. An additional test mode forces the interface to send a constant all-zeroes checksum, and permits testing the error control logic under program control.

Data buffering is provided on the card to tolerate delays in response to either data access or interrupt requests of up to approximately 256 bit times on both input and output, without loss of data or line utilization. Additional delays of indefinite length can be tolerated on the output side by sending a line protocol idle sequence.

Multiple independent core buffers can be chained together to form a single line packet. Similarly, a single line packet can be broken into a number of different core buffers on receipt.

The lines to the 303 are driven by emitter-follower-type current switches, or received by high-impedance receivers, permitting connection of multiple interfaces to a single 303 for reliability. The termination resistors on the received lines can be removed to permit proper impedance matching in this case.

A timer is provided to disable the driving of any lines to the 303 if the transmit status word has not been written for roughly 1 second, so as not to interfere with the transmissions of the other interface. This timer does not affect operation when the interface is internally crosspatched.

A socket is provided on the MLR card for connection to a Reload card (RLD). This is a send-only connection, and operation of the interface is unaffected by the presence or absence of an RLD, with the exception that bit 15 of the device code is turned on when an RLD cable is plugged in.

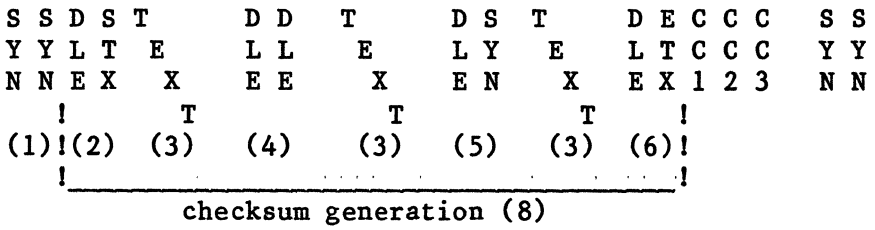
If an RLD cable is plugged into the interface, the card device type becomes 81. If an RLD cable is not plugged into the interface, the card device type is 01.

11.1.1 Line Protocol

All data on the line, both text and control, is organized as 8-bit bytes, and sent low-order bit first. There are four control bytes:

NAME	CODE (OCTAL)
SYN	026
DLE	020
STX	002
ETX	203

The protocol on the line is as follows:



NOTES

1. At least two SYN characters separate each two messages.

   An idle line is filled with SYN characters.

2. The beginning of a message is indicated by the sequence DLE, STX.

   The following characters are text; the text may be of any length.
3. The text is made up of characters taken from the buffer to be sent. The right half-word is sent first, then the left half-word. There is always an even number of text characters in a valid packet.

4. When the escape character DLE appears in the text, the hardware inserts an additional DLE.

5. If text is not available from core in time, the sequence DLE SYN is sent as an idle protocol until text becomes available.

6. The end of a packet is indicated by the sequence DLE ETX.

7. Each packet is followed by a 24-bit CRC checksum, sent as three 8-bit characters.

8. The 24-bit CRC checksum is generated and checked on all of the serial data from "DLE STX" to "DLE ETX" inclusive.

Note that doubled DLE's and the "DLE SYN" idle sequence are included, and that the checksum is generated on byte-serial data sent LSB first.

The polynomial used is:

$$(X^{12} + X^6 + X + 1)(X^{12} + X^{10} + X^4 + X^3 + X + 1)$$

On the receive side, the SYN pattern is used to detect character sync. During receipt of text, DLE is the escape character, and must be followed by:

1. another DLE, in which case one is discarded,

2. a SYN, in which case both are discarded, or

3. an ETX, in which case the end of text is detected.

Any other character indicates a line failure, and the packet is discarded.

11.1.2 Programming

The ML interface communicates with the DMA in the conventional fashion, and as such is programmed in a conventional way.

Points of interest are:

1. Error Indication (End pointer read)

   a. The transmit side will report error if interrogated while active, or during the time between transmit reset (if it hapened to be written) and writing the next transmit end pointer.

   b. The receive side will report error for the following conditions:

      (1) packet format error detected
      (2) internal buffer overflow
      (3) checksum error detected
      (4) internal buffer in a bad state
      (5) interrogation while active

      Interrogation after a receive reset will report error because of (4) above.

2. Format errors, checksum errors, or buffer overflow reset the receive finite state machine (FSM) to look for character sync and flag the data as error, but do not flush the buffer. Error-flagged buffers may lose the last byte of data (and will certainly lose some data if the error has been caused by buffer overflow).

3. The receive side will report "LAST PACKET" if there is no error, as defined above, and the end of the packet has been received and checked.

4. Buffer overrun is not considered an error, to permit receive chaining. If the program considers this an error, it should write a receive reset.

5. Writing a new begin pointer or a reset to either side while the side is active is recommended only as an infrequent practice, since there is a quite small, but finite, probability of "hanging" the bus as a result.

   The following describes normal operation:

   Writing a begin pointer while a transfer is in progress makes the related side of the interface go inactive and causes an interrupt.

   A "receive reset" while active flushes the input buffer, makes the receive side go inactive, forces the receive FSM to look for character sync, and generates an interrupt.

   A "transmit reset" while active flushes the output buffer, makes the transmit side inactive, and generates an interrupt.

6. Upon detection of a transmit data quit, the program must write transmit reset, which will cause a zero checksum followed by "end of message" format to be sent.

Failure to do this may result in the inclusion of a junk word in the middle of an otherwise valid packet. This error is possible only when the interface is connected to another Pluribus modem interface.

11.1.3 Configuration

A. There are two sets of jumpers on the MLR.

B. One set controls the crosspatch clock speed (normally set to 50 Kb) and the other set connects termination resistors to the receive lines from the 303, as described above.

11.2 MUR-MHX High Speed Modem Interface

MHX - 2100228G01
MUR - 2101529G08

This interface (MUR, MHX and a DMA) supports communication between a Pluribus and a variety of modems. It supports the Arpanet synchronous line protocol at speeds up to 1.344 Mb. The interface is programmed like any other DMA device; special features and unusual control sequences are discussed below.

Facilities are provided to crosspatch the interface before the modem so that it receives its own transmitted data, and to loop data back through the modem. This permits testing both the interface and the modem under program control.

Data buffering is provided on the card to tolerate delays in response to either data access or interrupt requests of up to approximately 512 bit times on both input and output, without loss of data or line throughput. Additional delays of indefinite length can be tolerated by the output side by sending a line protocol idle sequence.

Multiple independent memory buffers can be chained together to form a single line packet for transmission. Similarly, a single line packet can be received and broken into a number of different memory buffers.

The lines to the modem are driven by balanced line drivers, or received by high-impedance differential receivers, permitting connection of multiple interfaces to a single modem for reliability. The termination resistors on the receive and transmit lines can be removed to permit proper impedance matching when multiple interfaces are connected to a single modem.

A timer is provided to disable the output to the modem if the transmit status word has not been written every 1 second. This prevents a failed interface from interfering with the transmissions of the other (operational)interface. This timer does not affect operation on an internally crosspatched interface.

A socket is provided on the MUR board for connecting a Reload card (RLD). This is a send-only connection, and operation of the interface is unaffected by the presence or absence of an RLD. Bit 15 of the device type code is set to indicate an RLD connection.

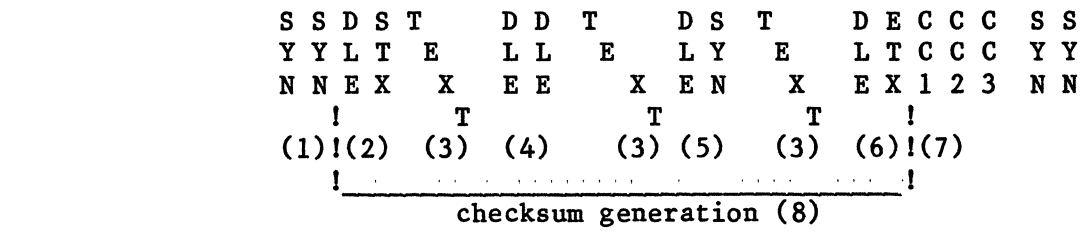
Device number selection, PI level selection and address, and 8 bits of device characteristics are switch-selectable. The card device type is wired to be 5. If an RLD cable is plugged into the interface, the card device type becomes 85.

11.2.1 Line Protocol

All data on the line, both text and control, is organized as 8-bit bytes, and sent low-order bit first. There are four control bytes:

NAME	CODE (OCTAL)
SYN	026
DLE	020
STX	002
ETX	203

The protocol on the ine is as follows:



At least two SYN characters separate each two messages. An idle line is filled with SYN characters. The beginning of a message is indicated by the sequence DLE, STX. The following characters are text; the text may be of any length. The text is made up of characters taken from the buffer to be sent. The right half-word is sent first, then the left half-word. There is always an even number of text characters in a valid packet.

When the escape character DLE appears in the text, the hardware inserts an additional DLE. If text is not available from core in time, the sequence DLE SYN is sent as an idle protocol until text becomes available.

The end of a packet is indicated by the sequence DLE ETX. Each packet is followed by a 24-bit CRC checksum, sent as three 8-bit characters. The 24-bit CRC checksum is generated and checked on all of the serial data from "DLB STX" to "DLE ETX" inclusive. Note that doubled DLE's and the "DLE SYN" idle sequence are included, and that the checksum is generated on byte-serial data sent LSB first.

The polynomial used is:

$$(X^{12} + X^{6} + X + 1)(X^{12} + X^{10} + X^{4} + X^{3} + X + 1)$$

On the receive side, the SYN pattern is used to detect character sync. During receipt of text, DLE is the escape character, and must be followed by:

1. another DLE, in which case one is discarded,

2. a SYN, in which case both are discarded, or

3. an ETX, in which case the end of text is detected.

Any other character indicates a line failure, and the packet is discarded.

11.2.2 Programming

The MUR interface communicates with the DMA in the conventional fashion, and as such is programmed in a conventional way.

Points of interest are:

1. Error Indication (End pointer read)

   a. The transmit side will report error if interrogated while active, or during the time between transmit reset (if it happened to be written) and writing the next transmit end pointer.

b. The receive side will report error for the following conditions:

(1) packet format error detected (2) internal buffer overflow (3) checksum error detected (4) internal buffer in a bad state (5) interrogation while active

Interrogation after a receive reset will report error because of (4) above.

2. Format errors, checksum errors, or buffer overflow reset the receive finite state machine (FSM) to look for character sync and flag the data as error, but do not flush the buffer.

Error-flagged buffers may lose the last byte of data (and will certainly lose some data if the error has been caused by buffer overflow).

3. The receive side will report "LAST PACKET" if there is no error, as defined above, and the end of the packet has been received and checked.

4. Buffer overrun is not considered an error, to permit receive chaining. If the program considers this an error, it should write a receive reset.

5. Writing a new begin pointer or a reset to either side while the side is active is recommended only as an infrequent practice, since there is quite small, but finite, probability of "hanging" the bus as a result.

The following describes normal operation:

Writing a begin pointer while a transfer is in progress makes the related side of the interface go inactive and causes an interrupt.

A "receive reset" while active flushes the input buffer, makes the receive side go inactive, forces the receive FSM to look for character sync, and generates an interrupt.

A "transmit reset" while active flushes the output buffer, makes the transmit side inactive, and generates an interrupt.

6. Upon detection of a transmit data quit, the program must write transmit reset, which will cause a zero checksum followed by "end of message" format to be sent. Failure to do this may result in the inclusion of a junk word in the middle of an otherwise valid packet.

11.2.3 Configuration

The following pages explain the strapping necessary to properly configure the MLX, MHX, and MUR boards.

12 MISCELLANEOUS BOARDS

12.1 PAR - Parity Card

PAR - 2101529G08

12.1.1 Purpose

The I/O Parity Card (PAR) computes parity from data and address on the I/O bus during accesses to devices on that bus. This parity is then normally passed through the bus coupler involved in the reference and checked against computed parity at the processor end. Two different implementations of parity checking are possible, depending on the choice of jumper wires on the card.

In the first scheme, the bus coupler (data source) generates parity for write data and checks parity for read data.

In the second scheme, the bus coupler always checks parity.

12.1.2 Write Source Gen. or Class. Parity

This method only works with data stored in memory. During write cycles the parity is computed and stored in memory with the data. When data is read, parity must be retrieved from memory with the data. The parity is then checked for errors.

Classical Parity provides shorter memory read and write cycles and is preferred for use with data stored in memory and will not work for I/O devices.

12.1.3 Write Source Check or Feedback Parity

In this scheme, parity must be computed for both read and write operations to allow for checking at the far end of the bus coupler. This method must be used for I/O devices, either feedback parity or classical parity can be used for memory.

During a write to the I/O bus, the access is suspended for a sufficiently long period to compute parity by asserting the HOLD line. When HOLD is turned off, the reference goes to completion. HOLD is again asserted on the trailing edge of STRB, for the next cycle. During a read, HOLD is turned off as quickly as possible by the leading edge of STRB.

Parity is constantly being computed, but the parity passed to the bus will not be good until a sufficiently long time after DONE to allow for the computation. The fall of STRB turns HOLD back on.

There are three options for further determining when parity should be computed based on address. These are again jumper selectable.

1. I/O only - Parity will be computed only for accesses with addresses within system I/O space (FC00O-FFC00). This is the normal mode for I/O busses.

2. except memory reads - Parity will be computed on all accesses except reads from addresses in system memory space (0-FC000). This is usually used on combined M/I busses. (Note that since we do not use parity cards on extended busses, no parity is computed for devices on extended busses.

3. all addresses - Parity will be computed on all accesses.

12.1.4 Parity Algorithm

One parity bit is associated with each of the two bytes of a word. The parity computation includes not only the 8 data bits of the byte but also the 20 address bits of that byte.

Note that the address of the left (high order) byte ends in a 0; the address of the right (low order) byte is odd. The parity bit is chosen such that the total number of one bits in the address, data, and parity bit is even.

Since the low order address bit is not relevant during a word (as opposed to byte) reference, the PAR ignores this bit and assigns the implicit addresses as above. This is because a processor may assert this bit when referencing memory during an indirect chain. Thus a direct reference might have different parity than an indirect one.

On a byte reference, this bit is relevant and therefore is included in the computation. On byte references, the data and its parity bit appear in the low order byte position regardless of which byte it actually is. Thus the high order bits are ignored and the high order parity bit output is not meaningful.

12.1.5 Implementation

Timing of control and data paths is shown below: When Schottky parity trees are used a delay in the write control path can be eliminated to allow for the shorter computation time. This is jumper selected on the card.

	Schottky	Standard
WRITE (feedback only)	140 ns	200 ns
READ (delay to be inserted in bus coupler)	130 ns	75 ns
reference which does not		

require parity computation with feedback parity	85 ns	85 ns

An objective in the design is to minimize the assertion of HOLD for off bus references. This is done by using a Schottky cross-coupled NAND latches for HOLD.

When using write souce generation, the control path delays do not exist since HOLD is never asserted. Therefore, there are no write delays, but the same data read delays as for write souce checking. The two parity lines, PBHI and PBLO, are driven to the bus only when an I/O bus address is detected.

12.1.6 Implications

It is important to the proper operation of wither scheme that the bus coupler insert delays where appropriate to compensate for parity computation delay in the PAR.

Therefore, on a read to the I/O bus the BCM on that bus must delay DONE to the BCP long enough to allow for the computation.

This does not apply to a write, since either no delay is needed or the required delay is built into the PAR; not does it apply to a memory bus, since presumably the correct delay is included in memories with parity.

12.1.7 Configuration

The following pages explain the strapping necessary to properly configure the PAR card.

12.2 PID - Pseudo-Interrupt

PID - 2100131G01

12.2.1 PURPOSE

The Pseudo-Interrupt Device (PID) is a one-card slave-only device. It has three addresses: read, write and clear. Its function is to store 7-bit data and report the highest value which has not been read since it was written and not cleared.

These levels are read from and written to data bits 1-7 (FE); all other bits are ignored. Each of the 128 possible 7-bit levels is represented by a latch in an array of addressable latches. The outputs of all the latches are input to a priority tree to determine the highest number stored. Master Reset causes a sequencer to clear each latch.

When a write is recognized at the write address, the latch addressed by the data lines is set.

When a read is recognized at the read address, the output of the priority tree (corresponding to the highest level set, 127 > 1) is output to the data lines, and the corresponding latch is cleared. Level zero is always set. If a number is written to the clear address, the appropriate latch is cleared.

A set of seven LED indicators at the edge of the card indicates the highest level currently set.

12.2.2 Addresses (XY=E0, E8, F0, F8)

FXY00 write address
R/W*

FXY02 read address
R only

FXY04 clear address
R/W*

Although FXY00 and FXY04 are functionally write only locations, a read returns the same number that would be obtained by reading the read address (FXY02), but without clearing that level.

12.2.3 Configuration

The folowing pages explain the strapping necessary to properly configure the PID.

12.3 RTC

RTC - 2101529G10

12.3.1 Purpose

The RTC is a one board device. The clock generates two varieties of pseudo-interrupts, one every 1.6 ms and the other every 25.6 ms. A 16-bit counter, which is incremented every 100 us, may be read by any other device. The Infibus signal CLKA (25 Mhz clock generated by the BCU) is divided by 2500 to drive the 16-bit counter.

A device may read the contents of the counter by issuing a read to the clock counter address. Reading the counter while it is incrementing is avoided by having the RTC perform a null bus operation (writing a zero to the PID) while it increments. Since it is bus master at that time, no other device can be reading the clock.

A one-second timeout disables the incrementing on the clock and all resulting bus accesses. The timeout is prevented by resetting the timer and keeping the timer reset with any clock access at least once a second. Master reset also stops the clock. The counter is tapped at the fourth and eighth bits to drive the interrupt logic.

When it is time to interrupt, a bus contention is initiated and the appropriate switch-selected pseudo-interrupt level is written as data to the PID address. The pseudo-interrupt levels are program readable.

Three other readable locations are on this device. The data is switch-selectable and can be used for system parameters, such as the IMP number.

12.3.2 Addresses (XY=E0,E8,F0,F8)

FXY06	clock counter
FXY08	pseudo-interrupt levels
	the high-order byte contains the 25.6 ms PID level (slow)
	the low-order byte contains the 1.6 ms PID level (fast)
FXY0A	Low Order Byte - IMP Number
FXY0C	not used
FXY0E	not used

NOTE: A write to these locations is a null operation.

12.3.3 Configuration

The following pages explain the strapping necessary to properly configure the RTC.

12.4 RTT

RTT - 2105513G01

12.4.1 Purpose

The purpose of the RTT with temperature sensing is two-fold. One, to provide a Real Time Clock with the same functionality as it predicessor discussed in the previous section, only in a PC version of the board versus the wire-wrap version. Two, to provide four remote temperature sensing channels for a Pluribus system cabinet.

12.4.2 Description

Four thermocouples are mounted strategically within the pluribus system and connected to four thermocouple channels on the RTT board. Each channel converts temperature measurements in the range of 0-64 degrees C (32-150 degrees F) into an 8-bit binary code. Previously unused switches on the RTT have been removed and their Infibus addresses, FXY0C and FXY0E are used to access each pair of channels over the 16-bit Infibus by the software. These registers are updated every 820usec.

If the temperature within a Pluribus system deviates from within acceptable limits appropriate traps are generated and reported to the NOC.

12.4.3 Addresses (XY=E0,E8,F0,F8)

FXY06 clock counter
FXY08 pseudo-interrupt levels

the high-order byte contains
the 25.6 ms PID level (slow)

the low-order byte contains
the 1.6 ms PID level (fast)

FXY0A	Low Order Byte	- IMP Number
FXY0C	High Order Byte	- Channel 0
	Low Order Byte	- Channel 1
FXY0E	High Order Byte	- Channel 2
	Low Order Byte	- Channel 3

NOTE: A write to these locations is a null operation.

12.4.4 Configuration

The following pages explain the strapping necessary to properly configure the RTT (with temperature sensing).

13 INFIBUS

The purpose of the Infibus is to provide the primary power and the communication path between devices. The term Infibus is often used in one of two ways:

1. The bus and bus protocol
2. The physical bus that makes up the backplane of a card cage.

The first usage implies the presence of the BCU.

Physically, the bus is a four-layer, printed wire panel containing 16 or 24 slots. The outer two layers provide the signal lines. The inner two layers are power buses.

Each line and power bus is connected to the same corresponding pin of each connector. Each device is inserted into one or more contiguous slots.

Power for the bus is provided by one of two possible power supply configurations:

1. A single power supply is plugged into the first 8 slots of the bus, leaving 16 slots for devices

2. A double power supply configuration is external to the bus, leaving 24 slots for devices.

The BCU module provides bus control and thus occupies the first slot of every infibus. The remaining 15 or 23 slots are available for other other devices.

A bus may be extended by adding another bus cabinet. This addition provides up to 23 more slots. The bus extender (a BXD card, a BXR card, and 2 cables) occupies one slot in each cabinet, leaving a total of 29, 37, or 45 slots for devices.

A low signal is active (asserted). A high signal is inactive (negated).

Only one device located on a bus has access to that bus at any one time. A device functions as both master and slave. A bus master (the device that has access to the bus) requests a data transfer with another(slave) device on the bus. The transfer is either a read or write.

The bus master communicates with the other devices located on the Infibus using the following information:

20-bit Address
Access type: Read, Write, or Read-Modify-Write
Data size: Word or byte
Data (write access only)
Parity

Each device on the bus continuously monitors the address transmitted by the bus Master. When a device recognizes its own address on the bus, the device becomes a slave (in terms of bus access). This slave device then performs the function designated by the address and control lines. When the function is complete, a DONE signal is returned.

The DONE signal notifies the master that:

1. Data from a read access is available

2. The bus master relinquishes control to the bus, and the BCU choses the next bus master from among those devices that have requested bus access.

13.1 Master Slave Modules

System modules that request and recieve bus access are called master modules. A master module instructs another system module(slave module) to receive or transmit data. Processors, DMA devices and the control panel are examples of master modules.

Memory modules always function as slaves since they cannot cause another module to execute an action. A master module functions as a slave if addressed by another master module.

If a device does not recognize the address that the master relays to the bus, or if the slave device malfunctions, the action is aborted. The BCU sends a QUIT signal (level 6 interrupt) to the bus master after a predetermined amount of time has elapsed (the particular bus determines this time period).

Once a QUIT signal is sent, the bus master relinquishes bus control, and the BCU grants bus access to the next requesting device. The BCU hardware determines the amount of elapsed time between the QUIT signal and bus access (usually between 3 and 800 microseconds).

Processor busses normally have the longest QUIT timeouts (300 t0 800 us), I/O busses, the next longest (30 to 300 us), and memory busses the shortest (3 to 30 us).

13.2 Addressing

It is possible that two different devices on a bus may recognize the same address. If both of these devices repond to the same bus access and initiate an action or a DONE, the system will malfunction. To prevent this occurrence, devices use an external criteria to resolve which device becomes the slave. The address recognition switches on each system device are normally set to recognize mutually exclusive portions of the system address space.

The BCU generates an initialization signal, called RESET (interrupt level 4), to each of the attached devices. Devices receive this signal 1.) when power is being restored, 2.) the bus is reset from the console or from another processor, or 3.) a bus transaction has not occured for over second.

When a device receives the RESET signal, it terminates its activity and initializes its registers and indicators.

13.3 Bus Access and Service

Data transfers between system devices occurs in two cycles: a select cycle and a service cycle. Based upon priority and precedence, the BCU selects a single system module from all requesting modules. During the service cycle, the selected module gains access to the bus and performs the required data transfer. The select and service cycles overlap in time.

A device may request bus access for data transfers while the current service cycle is still in operation. When bus access is available the device is notified. This overlap in selection and service maximize system efficiency.

Requests for bus access are received by the BCU on service request lines according to the access function required.

There are three types of access functions on the Infibus. The functions, the corresponding service request lines and the priorities are listed below.

Communication Function	Service Request Line	Priority
Module Service Request	SRLD	1
Interrupt Service Request	SRL4	2
	SRL3	3
	SRL2	4
	SRL1	5
Processor Service Request	SRLC	6

13.4 Selection and Service Control

The control sequence used during selection and service follows the same pattern for all modules. For each set of service lines (SRLx), there is a corresponding set of select lines (SELx). SELx, together with the precedence pulse, PCDA/B, is the first response by the BCU to a service request.

13.4.1 Module Service Request

A module service request occurs when a system module requires the bus to perform a direct data transfer.

These direct data transfers are independent of processor control and have the highest priority.

SELECT CYCLE

When select line SELD is "high", one or more master modules can activate service request line SRLD. If no master module is selected and waiting for the next service cycle (indicated by SACK being "low"), the BCU sets the SELD line "low" and, after a delay, the precedence pulse (PCDA/B).

SELD goes to all modules simultaneously and temporarily prevents additional requests for the Infibus. The precedence pulse is propagated (chained) through all modules until it reaches the first master module requesting the Infibus with SRLD. This requesting master module, highest in the precedence chain, blocks further propagation of this signal.

By receiving SELD and PCDA together, the requesting master module is selected for the next service cycle. This Selected Master module then sets select acknowledge (SACK) "low", and then removes service request SRLD. The BCU sees SACK "low" and removes the select (SELD) signal. By setting SACK "low", the waiting master module inhibits further select cycles.

SERVICE CYCLE

The waiting Selected Master module monitors the strobe line (STRB) to determine when the Infibus has been released by the Service Master module. Timing is derived from STRB. When STRB is removed, the Selected Master module becomes the Service Master. It sets address and control lines and the data lines if the operation is a write "low".

status, indicating that the addressed slave module either is not present in the system or is inoperable.

13.4.2 Interrupt Service Request

An interrupt service request can occur on any of four assigned lines when a system module requires the bus to interrupt the control processor instruction execution sequence.

SRL4 will initiate a level four interrupt which is reserved for system fault and RTC/RTT interrupts. The BCU also initiates a level 4 interrupt upon detecting a local power-failure, power-restart or a line-frequency (jiffy) interrupt.

SRL3 will initiate a level three interrupt which is reserved for high data rate, input-output devices such as disk, drums and magnetic tapes. These devices normally operate in a block transfer mode which allows direct communication with memory or another controller. None of the systems we maintain use a level three interrupt. The central processor is interrupted when the end of the block transfer occurs.

SRL2 will initiate a level one interrupt which is reserved for slow I/O input-output devices such as card and paper-tape readers and punches. None of the Pluribus systems use the level two interrupt.

The paper-tape reader and the cassette reader are both polled devices. Since they are only used by the bootstrap loader as polled device, their interface is not configured to use this interrupt.

SRL1 will initiate a level one interrupt which is reserved for slow I/O devices and is also shared by manual system interrupts. Pressing the "ATTN" on the control panel or asserting bus signal EXAT (external interrupt) causes a level one interrupt.

An example of an external interrupt is a remote power failure. Corresponding to each of these four interrupt service request lines are four interrupt select lines (SEL1-4) that are used by the BCU with the precedence pulse to select the interrupting device for

the next service cycle.

The bus access sequence for the interrupt function is similar to the access sequence for SRLD. Once a line interrupt request is selected on any of the four lines, all four interrupt levels are masked automatically by the processor. If a specific interrupt deserves priority over the current level, it is necessary for the current interrupt routine to enable the other level. When selected and granted service, the selected module places its device number on the data bus. This number is typically the address of the interrupting module and is stored in the executive space of memory. This device number is used by the processor when addressing the module in response to the interrupt Service Request

13.4.3 Internal Interrupts

Internal interrupts are a group of events recognized by the BCU which cause a processor response similar to a Line Interrupt. Power Failure, Restart and Line Frequency interrupts cause level four interrupts. The Infibus signal EXAT results in a level one interrupt.

To the processing system internal and line interrupts appear identical; however, the Infibus response differs. To a system user, this difference is unimportant since internal interrupts are not involved with any design. They exist as an internal function of the BCU with precedence higher than any system module which shares levels four or one.

13.4.4 External Interrupts

External interrupts are interrupt signals external to the Pluribus. The External Interrupt Module, interfaces these signals and initiates line interrupts. The module provides selective interrupt masking identivies each external interrupt signal with a unique module address.

13.4.5 Self Interrupts

Self interrupts are generated by each processor as a trap for the two recognizable and significant failure conditions - unimplemented instructions and Bus Cycle Transfer failure.

These interrupts are a function of processor microcode operation. They do not use unique Infibus operations nor do they share levels one through four. Interrupt levels five and six are provided for the Self Interrupts and each processor has unique memory locations assigned for these interrupts.

13.4.6 Line Interrupt Operation

The BCU continuously monitors the interrupt service request lines.

If an interrupt service request is received and interrupts are not inhibited by the MINH signal, and the particular interrupt line is not masked by a programmable mask bit, the BCU signals the processor that an interrupt condition has been detected and that the instruction sequence should be interrupted at the end of the current instruction.

When the current instruction ends, the processor signals the BCU which then transmits the corresponding select (SEL1-4) signal and, after a delay, issues the precedence pulse PCDA. An interrupt requesting master module, upon receiving an SELx corresponding to its SRLx and a precedence pulse becomes selected, inhibiting the further propagation of signal PCDA. This Selected Master issues SACK and then removes SRLx. The BCU removes the select signal SELx. When the current service cycle is completed, the waiting Selected Master becomes the Service Master. An interrupt Service Master does not drive the address bus but does place its module address on the data bus and, after a delay, sets the strobe signal "low" to indicate the data bus can be read by the processor. The processor issues DONE and the service cycle terminates.

During the interrupt service cycle, the BCU sends to the processor the number of the interrupt line serviced. The firmware accepts the interrupt line number and stores the module address, processor current status and program counter into the system interrupt executive space corresponding to the accepted line, then vectors to the interrupt routine.

The executive space allocation is shown in figure 1. This priority interrupt process avoids device polling because the interrupting module address is automatically stored in a location determined by the processor microcode.

The interrupt response time is 5.9 microseconds assuming an 850-nanosecond core memory. This time includes all of the content switching - module address, program counter, status register and interrupt vector - and the fetching of the first instruction of the interrupt service routine from core memory.

	MODULE ADDRESS	STATUS	PROGRAM COUNTER	SERVICE ROUTINE VECTOR
4	0018	001A	001C	001E
3	0010	0012	0014	0016
2	0008	000A	000C	000E
1	0000	0002	0004	0006

Figure 1. Interrupt Executive Space Allocation Processor Service Request

A processor service request occurs when any processor requires access to the bus to execute programmed instructions.

Since all processors use the same service request line, SRLC, processors are designed such that, once they have been selected for access, they will not assert the service request line again until all processors requesting access have been serviced (SRLC becomes inactive). Therefore, one processor cannot withhold access to the Infibus from all other processors. The processor function has the lowest priority.

PRIORITY

Simultaneous requests on two or more service request lines are resolved in the BCU by priority logic.

PRECEDENCE

If several modules request service simultaneously on the same service request level, the module physically nearest to the BCU gains access because the closest requesting module traps the precedence pulse.

The precedence pulse generated by the bus controller is chained from module to module with the nearest module receiving the pulse first. Due to this precedence pulse chain, it is imperative that all modules propagate the precedence pulse and all modules be installed adjacently on the Infibus.

13.5 Infibus Line Description

In this description, signal and control lines on the Infibus are described according to function and load.

Function - defines the signal or line and its use in the system.

Load - describes the source module where the signal

originates, and the terminating module where the signal terminates.

13.5.1 Select Cycle Lines

Fourteen lines are reserved for module selection.

PCDA/B-P - Precedence Pulse (pin 25A, B)

Function - PCDA-P is the precedence chain pulse received at the input to a given module. If the device is inactive on the level of the particular SELx currently asserted, PCDA is propagated to become PCDB as an output. PCDB is connected on the Infibus to become PCDA of the adjacent card.

Load - The signal has one originating source, the BCU, and is propagated right to left (from module insertion side) by each module if it is not requesting bus access on a selected line.

SACK-N - Select Acknowledge (Pin 26A)

Function - SACK is sent to the BCU by a device module to indicate that the module recognizes selection for the next service cycle. The selected module sets SACK-N "low" in response to receiving SELx and PCDA before setting its SRLx "high".

Load - SACK has one load, the BCU, and as many sources as there are master modules in the system.

SRLD-N - Service Request, Device (Pin 35A)

Function - Set "low" by a module to request bus access. SRLD-N initiates bus controller select cycle when the previous SACK is removed. SRLD must be set "high" after SACK goes "low".

Load - SRLD has one load, the BCU, and as many sources as there are master modules in the system.

SRL1-N, SRL2-N, SRL3-N, SRL4-N - Service Request, Interrupt (Pins 36-39A)

Function - Line set "low" by a module to indicate an

interrupt condition has occurred. SRLx-N initiates a BCU select cycle when the previous SACK is removed. SRLx must be set "high" after SACK goes "low".

Load - These lines have sources from each system module that can generate a system interrupt on that level. Each line has one load in the BCU.

SRLC-N - Service Request, Computer (Pin 34A)

Function - SRLC is set "low" by processors to request bus access. SRLC-N must be set "high" after SACK goes "low".

Load - SRLC has one load on the BCU, one load and one source on each processor.

SELD-N - Select, Device (Pin 35B)

Function - SELD is used along with the precedence pulse to select the module for bus access. SELD-N is returned by the BCU in response to SRLD. SELD is removed after SACK is received by the BCU.

Load - SELD has one source, the BCU, and as many loads as there are master modules in the system. There is no unique receiver for SELx and therefore it is used directly in the logic.

SEL1-N, SEL2-N, SEL3-N, SEL4-N - Select, Interrupt (Pins 36-39B)

Function - SELx is used along with the precedence pulse to select the module for bus access. SELx-N is returned by the BCU in response to SRLx. SELx is removed after SACK is received by the BCU.

Load - Each line has one source, the BCU, and as many loads as there are system modules that can generate a system interrupt on a defined level.

SELC-N - Select, Computer (Pin 34B)

Function - SELC is used along with the precedence pulse to select the highest precedence processor that is requesting access. SELC-N is returned by the BCU in response to SRLC. SELC is removed after SACK is received by the BCU.

Load - SELC has one source, the BCU, and as many loads as there are processors on the Infibus.

13.5.2 Service Cycle Timing

Three control signals are used to delineate the service cycle: STRB, DONE and QUIT.

STRB-N - Strobe (Pin 27B)

Function - The leading edge of STRB indicates the Infibus address, data (if write mode) and control signals are stable. The trailing edge indicates the service cycle is complete and the selected master may begin its service cycle.

Load - STRB is bi-directional and may have both a source and a load in every system module.

DONE-N - Done (Pin 27A)

Function - DONE is issued by the slave module to indicate that the bus service cycle is completed. In a READ cycle, the leading edge of DONE indicates valid data. In a WRITE cycle, the DONE indicates data acceptance. When DONE is received by the Master, it terminates the Service cycle, removing the Address, Control and Data signals from the bus.

Load - DONE is bi-directional and may have both a source and a load in every system module.

QUIT-N Quit (Abort) (Pin 26B)

Function - QUIT has several functions. Principally, it is used to signal the master module that the slave module did not transmit DONE within 5 microseconds after the start of the cycle. Alternatively, it is generated by a parity checking memory to indicate parity failure, a bus coupling module to indicate transfer failure, and a memory protection module to indicate private memory. The QUIT is used to cause a Master to terminate its Service Cycle and to indicate invalid data transfer. The master module may record the receipt of QUIT as a cycle-abort status and initiate a processor interrupt.

Load - This signal has multiple possible sources but principally the BCU, and as many loads as there are master modules.

13.5.3 Read/Write Control

Three control signals (RITE, HCYC, BYTE) are used to specify Read/Write operations.

RITE-N, HCYC-N, BYTE-N - Control Lines (Pins 23A, 24A, 23B)

Function - These lines are used by the Selected Master module to command a read or write function within the slave module. RITE, when "low", specifies a Write function. HCYC, when "low", specifies a half-cycle memory function. BYTE, when "low", specifies a byte-only transfer using data bits DB07-DB00. Combination functions of these three signals are listed in the below table.

RITE-N	HCYC-N	BYTE-N	FUNCTION
L	L	L	Write Only, Byte *
L	L	H	Write Only, Word *
L	H	L	Clear-Write, Byte
L	H	H	Clear-Write, Word
H	L	L	Read-Clear, Byte
H	L	H	Read-Clear, Word
H	H	L	Read and Restore, Byte
H	H	H	Read and Restore, Word

* A Read-Clear function must occur first in core memory.

Load - The lines are bi-directional and have a source in each master module using the lines and a load in the slave modules.

13.5.4 Address and Data

A set of 16 lines is used for address transmission and another set of 16 lines for data transmission.

AB15-AB00-N Address Lines (Pins 10-14A, 17-19A, 10-14B, 17-19B)

Function - Master system modules and addressable slave modules use these lines for addressing functions. AB15 is the most significant address line and AB00 is the least significant. AB00, when "low" in the byte mode, specifies the rightmost byte (odd numbered); when "high", specifies the leftmost byte (even numbered) of a 16-bit word. When byte mode is not indicated the the BYTE Infibus signal, AB00 is unused for addressing but rather may indicate multilevel indirect addressing.

Load - These lines are bi-directional. They have a source at any master module that can request bus service. Each line has a load at every slave module with an addressable function.

DB15-DB00-N - Data Lines (Pins 41-48A, 41-48B)

Function - All data transfers between system modules are these lines. DB15 is the most significant data line; DB00 is the least significant. Single-byte transfers use lines DB07-DB00 only, the rightmost byte. In transmitting a byte, lines DB15-DB08 should be high or zero. However, in receiving a byte, DB15-DB08 are indeterminate.

Load - These lines are bi-directional. They may have both a source and a load at all addressable system modules.

13.5.5 System Control

There are thirteen lines in the Infibus that are used for special system functions. They are related to switch functions that control system-reset, autoload and power-line interrupts.

REPB-N - Reset Push Button (Pin 07A)

Function - REPB is set "low" to cause the BCU to initiate a Master Reset operation.

Load - The Console Panels are the source and one load is in the BCU.

MRES-N - Master Reset (Pin 06A)

Function - MRES is set "low" by the BCU when it receives REPB, power-initiation (recovery) or power failure signals. MRES is used by all system modules to reset to an initial status condition. The activation of MRES is always delayed 2.2 to 3.0 milliseconds after receiving the initiation signal such as power failure or REPB. Prior to issuing MRES, the BCU terminates all bus transfers so that all operations will be stable prior to reset.

Load - A source in the BCU and a load in each module that can be reset.

PRAL-N - Power Restart Autoload (Pin 21A)

Function - PRAL being "low" enables the BCU to generate a pulse on ATLD instead of a level four interrupt when a power-restart occurs.

Load - The Control Panel is the source and the BCU is the load.

ATLD-N - Autoload (Pin 05B)

Function - ATLD is a pulse generated to initiate an autoload operation.

Load - Multiple sources are possible. One source is the Control Panel logic responding to the "LOAD" switch and another is the BCU optional response to power recovery. One load exists on the ALD module.

MINH-N - Master Interrupt Inhibit (Pin 05A)

Function - MINH, when "low", inhibits the BCU from responding to any system interrupt. Processor self-interrupts are never inhibited.

Load - The Control Panel is the source and the BCU is the load.

PFIN-N - Power Failure Interrupt Inhibit (Pin 21B)

Function - When PFIN is "low", it inhibits the BCU generation of an interrupt service request on level four when the power status line, PWST, indicates power failure.

Load - A source in each of the Control Panels and a load in the BCU.

PRIN-N - Power-Restart Interrupt Inhibit (Pin 22A)

Function - Setting PRIN "low" inhibits the BCU generation of a level four interrupt request whenever PWST indicates power restart.

Load - The control panels are the sources and the BCU is the load.

PWST-N - Power Status (Pin 06B)

Function - PWST is held high by the power supply to indicate regulated power is available. Power supply logic activates this signal 3.3 milliseconds before loss of regulation due to power failure. Only the BCU monitors this signal. If the function is not inhibited by PFIN, the BCU initiates a level four interrupt request upon PWST going "low". If PRIN is "high", the BCU will initiate a level four interrupt upon PWST going "high" after a delay to assure stable power conditions. Both the Power Fail and Power Restart interrupts have higher precedence than any module using SRL4.

Load - A source in the Power Supply and one load in the BCU.

LFIN-N - Line-Frequency Interrupt Inhibit (Pin 22B)

Function - Setting LFIN "low" inhibits the BCU generation of an interrupt service request on level four when the line frequency signal, LFRQ, is received from the power supply.

Load - The Control Panel is the source and the BCU is the load.

LFRQ-N - Line Frequency (Pin 07B)

Function - LFRQ is set "low" by the power supply once each line frequency cycle (normally 60 Hertz). Only the BCU monitors this line. If the function is not inhibited by LFIN, the BCU initiates a level four interrupt. The Line Frequency interrupt has higher precedence than any module using SRL4, although it may be serviced simultaneously with a Power Failure. LFRQ is used by the processor as jiffy clock where it compares it with the RTC/RTT clock signal to insure the RTC/RTT is working. Our diagnostics also use LFRQ.

Load - One source in the Power Supply and one load in the BCU.

EXAT - External Attention (Pin 08B)

Function - Assertion of this line causes the BCU to initiate a level one interrupt. This interrupt has higher precedence than any module using SRL1.

Load - Multiple sources are possible and one load in the BCU.

CLKA-N - Clock (Pin 50A)

Function - CLKA is generated by the BCU using a 25 MHz. oscillator. The signal represents the system clock and can be used by other system modules.

Load - One source in the BCU and one load in each system module requiring a system clock. The RTC uses this signal as the base for its clock signal. CLKA is used by the processor as its clock. When doing XPATCH, this signal is used as the clocking signal.

RUNN-N - Run (Pin 08A)

Function - RUNN is generated by all processors and is used by the Control Panel logic to light the halt and idle indicators. When RUNN is set "high", at least one processor is not halted - either executing instructions or is idle.

Load - One source in each processor and one load in the system Control Panel logic.

13.5.6 Block Transfer Adapter Lines (BTA)

A set of four Infibus lines is used to allow communication between any immediately adjacent modules. These lines are non-continuous. Signals appearing as output on BT1B (for example) are input at BT1A of the next higher numbered card location. These lines are primarily for special communication between a BTA and its slave I/O controller. They are also used by processor modules. Function described below apply only to communication between a block transfer adapter and its companion I/O controller. For other uses, the function can be uniquely defined. Any module that does not use these lines should propagate them by jumpering a A and B sides of the connector together. For example, BT1A (Pin 30A) should be jumpered to BT1B (Pin 30B).

BT1A, BT1B - Bus cycle Request (Pins 30A, B)

Function - The BT1A line is set "low" by the slave I/O controller to signal the BTA that a bus cycle is required. This signal is received on the BT1B connector pin of the BTA.

Load - One source in each I/O controller and one load in the companion block

received on the BT2A connector pin of the I/O controller.

Load - One source in each BTA and one load in the I/O controller.

BT3A, BT3B - Controller Error (Pins 32A, B)

Function - The I/O controller asserts the BT3A line to indicate an error condition to the BTA. This signal is received on the BT3B connector pin of the BTA.

Load - One source in each I/O controller and one load in the BTA.

BT4A, BT4B - Interrupt Request (Pins 33A, B)

Function - The BT4B line is set "low" by the BTA to signal the I/O controller to request a system

interrupt. The signal is received by the I/O controller on the BT4A connector pin. This signal remains "high" until the BTA block-length register counts down to zero, the controller error BT3B is set "low" or an abort condition occurs during a block transfer. When the BTA is installed but not being used for direct data transfers, bus cycle request BT1B is sent back to the slave I/O controller on this line.

Load - One source in each BTA and one or two loads in the I/O controller.

13.5.7 Memory Option Lines

The memory option lines provide system capability to expand the basic data and address Infibus control.

KEY0 (AB16)-N, KEY1(AB17)-N - Memory Protect Keys (Pins 20A, B)

Function - The KEY0/KEY1 lines are alternately defines as AB16/AB17. They are dual purposed to provide either memory protection key signals or a memory address expansion.

Load - Multiple sources are possible, typically, one source in each processor and one load in each memory or memory interface module.

KEY0-N (ALT), KEY1-N (ALT) - Memory Protect Keys (Pins 09A, B)

Function - The alternate KEY0, KEY1 lines provide common memory address expansion to 20-bits. These lines are not used on processor busses.

Load - Multiple sources are possible, typically, one source in each BCM, and one source in each I/O controller using 20-bit addressing. Each memory or memory interface module has one load.

HOLD-N - Memory Cycle Inhibit (Pin 24B)

Function - Setting HOLD "low" prevents any memory module from initiating a memory cycle. This signal is set "low" by a memory module when operating in the

interlocked mode. The BCU also asserts HOLD during an interrupt service cycle. A memory protect module asserts HOLD during the period it verifies protection keys.

Load - One source in each memory protect module and one in the BCU. One load in each memory.

PBLO, PBHI - Parity Bits (Pins 49A, B)

Function - These lines provide for parity.

Load - Multiple sources are possible, typically, one source in each processor and one load in each memory or memory interface module.

13.6 Pin Assignments

Pin Number	A Side	B Side
01	GND	GND
02	GND	GND
03	+15V	+15V
04	+15V	+15V
05	MINH-N	ATLD-N
06	MRES-N	PWST-N
07	REPB-N	LFRQ-N
08	RUNN-N	EXAT-N
09	KEY0(ALT)-N	KEY1(ALT)-N
10	AB00-N	AB08-N
11	AB01-N	AN09-N
12	AB02-N	AB10-N
13	AB03-N	AB11-N
14	AB04-N	AB12-N
15	GND	GND
16	+5V	+5V
17	AB05-N	AB13-N
18	AB06-N	AB14-N
19	AB07-N	AB15-N
20	KEY0/AB16-N	KEY1/AB17-N
21	PRAL-L	PFIN-N
22	PRIN-N	LFIN-N
23	RITE-N	BYTE-N
24	HCYC-N	HOLD-N
25	PCDA-P	PCDB-P
26	SACK-N	QUIT-N

27	DONE-N	STRB-N
28	+5V	+5V
29	+5V	+5V
30	BT1A	BT1B
31	BT2A	BT2B
32	BT3A	BT3B
33	BT4A	BT4B
34	SRLC-N	SELC-N
35	SRLD-N	SELD-N
36	SRL1-N	SEL1-N
37	SRL2-N	SEL2-N
38	SRL3-N	SEL3-N
39	SRL4-N	SEL4-N
40	GND	GND
41	DB00-N	DB08-N
42	DB01-N	DB09-N
43	DB02-N	DB10-N
44	DB03-N	DB11-N
45	DB04-N	DB12-N
46	DB05-N	DB13-N
47	DB06-N	DB14-N
48	DB07-N	DB15-N
49	PBLO-N	PBHI-N
50	CLKA-N	RESERVED
51	+5V	+5V
52	-15V	-15V
53	-15V	-15V
54	GND	GND
55	GND	GND

14 DEVICE HANDLING & I/O

Pluribus systems may be comprised of two types of I/O devices: pseudo-interrupt devices and priority interrupt devices.

Two primary distinctions exist between the devices. The pseudo- interrupt devices use the PID for interrupt signaling and interpret 20-bit addresses, while the the priority interrupt devices use traditional priority interrupt mechanisms and interpret 16-bit addresses.

This section is primarily concerned with the specifics of programming pseudo-interrupt devices. However, special considerations relevant to programming priority-interrupt I/O devices in a Pluribus envrionment are detailed at the end of this section.

14.1 Address Structure

As shown in figure 1, system addresses FC000 to FFBFF are reserved for Pluribus system I/O space. The detailed structure of this space depends on the allocation of addresses to I/O busses.

The total system I/O space is divided into four almost equal parts, two of which are assigned to each bus.

The high address segment for each bus is referred to as the primary I/O space and the low address segment as the auxiliary I/O space. Note that the primary address space of bus 1 (from address FF000 to FFBFF) is shorter than the other 3 segments by 512 words since these 512 addresses are allocated to individual processor maps, registers, and local I/O space.

Seventy-two words of address space are located at the beginning of each primary address space. These locations, associated with the clock (RTC/RTT) and PID on the bus, contain the bus coupler (BCM) control registers, and provide mapping for backwards bus coupling (using this bus).

The remainder of the system I/O space is divided into 8-word blocks. These blocks, called device register blocks, are associated with an I/O device (other than the clock and PID) and are attached to the bus.

A processor activates an I/O device by writing to a certain address within the device register block. The number of I/O busses and the allocation of system I/O addresses to these buses creates structures that may vary from that shown. The switches on the bus couplers determines this allocation.

The auxiliary and primary space allocations are identical with one exception. The highest address segment of the auxiliary is not reduced in size by 512 words, and therefore, is the same size as the rest of the segments.

The low 72 words of each primary segment is reserved on each bus.

14.2 Programming DMA I/O DEVICES

DMA (Direct Memory Access) devices provide the means for the automatic transfer of blocks of data to (from) memory from (to) I/O devices on the I/O busses.

Even though the DMA hardware and its associated device interface are on separate cards, the programmer regards them as a single unit.

In general, each data transfer involves sending or receiving a number of data buffers. Each data buffer consists of an integral number of words (an even number of bytes).

The programmer uses three main registers to control data flow operations such as "read" or "write". The registers are: 1) the begin memory (buffer) address register, the end memory (buffer) address register, and the status register.

These registers are contained in the 8-word device register blocks.

Each of these registers is described in detail below.

DEVICE TYPE

The high-order byte contains a number indicating the type of device interface involved (e.g. modem, host, etc.). The hardware in the device interface associated with the DMA determines the number.

In general, the low-order byte contains the value set in the device number switches in the device interface. The device type register is readable; writing to it will have no effect.

RECEIVE/TRANSMIT BEGIN ADDRESS

These registers contain the high-order 16 bits of the 20-bit system address that specifies the first location of the buffer to be read or written.

Bits 1-3 of the 20-bit starting address are contained in the receive or trnasmit status register (see below). Bit 0 of the 20-bit system address is always 0.

The beginning address registers may be either read or written. If read, the result returned is simply zero.

Normally, when writing to this location, no data transmission will be in progress in the direction corresponding to the register written (receive or transmit). The device will simply be initialized to transfer a buffer; actual data transfer does not commence until the buffer-end register is written.

If a transfer is in progress when the location is written, the transfer is aborted. The error bit (in the end address register - see below) will now be set, the PID written, and the corresponding half (receive or transmit) of the device initialized for transmission of a new buffer.

RECEIVE/TRANSMIT END ADDRESS

These registers may be read or written. Normally, bits 0-12 of these registers are written with the low-order 13 bits of the address at the end of the buffer. (Bit 0 is actually ignored and assumed to be zero.)

Writing to this address initiates the data transfer. Once the data transfer is completed, the programmer can read the registers to receive more specific information concerning the transfer.

Bit 15, if set, indicates that no error has been detected, and indicates that this is the last buffer of the transfer. (Bit 15 is set when the last buffer is transmitted correctly.)

Bit 0 serves as an error bit and is set if: (1) the device was reinitialized during the previous transmission (see above), (2) a QUIT occurred during transmission of the previous buffer, (3) the device is currently active (see RECEIVE/TRANSMIT STATUS below) or (4) the device itself is reporting an error.

Bits 1-12 of the end address register indicate the address, module 2 to the 12th power, of the last word actually transferred. The top 7 bits of the DMA pointer into the buffer come from the begin address (see below) and never change. Therefore, the buffer will "wrap around" on 8K byte boundaries in memory.

RECEIVE/TRANSMIT STATUS

The receive and transmit status registers may also be both read and and written.

Writing the RESET bit causes a particular half of the interface (receive or transmit) to reset itself.

If that portion of the interface is active when the reset is initiated, the operation in progress will be aborted. In addition, the error bit in the end-address register will be set, and the receive or transmit level for the device will be written to the PID.

Before writing the begin address, bits 1-3 of the buffer beginning location must be written to bits 0-2 of the corresponding status register.

Reading one of the status registers allows a processor to determine the PID level associated with the data transfer direction, and to interrogate the QUIT flag.

The PID is then written and the QUIT flag set if a QUIT occured during the previous data transfer. Such a result could indicate a parity error, non-existent address, etc. In this case, when the end pointer is read, the error bit will be set.

The interpretation of the device-dependent status bits varies from device to device, but generally, these bits provide direct two-way communication between a processor and a device interface.

If set, one of the device-dependent bits will become the ACTIVE bit, indicating that a transfer to or from the device is in progress.

More precisely, a DMA device is active from the time that its end pointer is written (which starts the device) until the time that it writes its level to the PID (indicating it is done).

DEVICE DEPENDENT

This register can be optionally used by the device interface for any appropriate function.

The assignment of data bits is arbitrary.

To initiate a transfer by a BBN DMA device, the program typically performs the following steps:

1. Write the STATUS REGISTER - This sets up the low-order 3 bits of the buffer start-word address and selects any desired options (e.g., looped modem). This procedure is normally done only once for a sequence of DMA transfers.

2. Write the BEGIN ADDRESS REGISTER - This sets up the 16 high order bits of the buffer start address.

3. Write the END ADDRESS REGISTER - This sets the end address of the buffer and initiates the DMA

transfer.

When the PID level, indicating device completion, is picked up by a processor, it will:

4. Read the END ADDRESS REGISTER and check bit-15 (completion). If set, the transfer was completed (i.e., no error occurred and this is the last buffer of the transfer). Bits 1-12 indicate buffer length.

5. If this bit is not set, bit 0 (error) is checked. If bit 0 is zero, then no error occurred but this buffer is not the last of the transfer. As above, bits 1-12 indicate buffer length.

6. If bit 0 is one, then an error has occurred. These are differentiated by examining the STATUS REGISTER. If bit 13 (active) is set, the device is still active and the PID value was spurious. If bit 8 (QUIT) is set, a QUIT occurred during the transfer. Device dependent status bits may further define the error.

In addition to the registers mentioned above, each BBN block transfer type of device has a number of manually settable switches. These switches, located on the device interface, are as follows (the number of switches reserved for each purpose is shown in parentheses):

1. Device Address Switch (10) - These switch settings define the address of the device register block in I/O space (see figure 6). The ten switches specify bits 4-13 of the address of the first register of the block. (Bits 14-19 of this address are all ones and bits 0-3 are all zeroes.)

2. Receive/Transmit PID levels (7) - These seven switches define the number written to the PID upon completion of a data transfer. For duplex devices there aree two sets of switches. For simplex devices only a single set is provided.

3. PID Address (2) - selects which of the 4 PIDS will be written to by the device. The selected PID must be on the same Infibus as the device itself.

4. Device Number (8) - In general, a set of 8 switches readable as the low-order byte of the first word in the device register block (see figure 8).

14.3 Non-DMA I/O DEVICES

Typically, non-DMA devices only contain a small amount of internal hardware buffering, and therefore, need to be frequently serviced by a processor(no slower than every few byte-times).

The mechanism by which such a device is serviced can take one or two forms in Pluribus systems.

One approach allows the device to be passive and puts the responsibility for servicing the device completely on the processors. For input, the processors have to poll the devices faster than the input rate so that no data is lost. For output, the processors have to deliver data to the devices at a rate sufficient to guarantee that no undesirable gaps within the data occur.

Although such an approach permits a relatively simple hardware interface implementation, it may require an undesirable amount of processor overhead.

An alternate approach is to make the device active with respect to notifying the processors when it requires service. In a Pluribus system this implies that the device will write its level to the PID when its internal buffers are ready. Checking whether the device needs service will, therefore, be done automatically as part of the main PID reading loop of the program.

Such an approach, of course, requires more hardware in the device interface than does implementation of the first approach mentioned above. Both DMA and non-DMA I/O devices which are addressed through system address space will have 16 byte device register blocks associated with them. In contrast to the DMA device register blocks which have a common format for all DMA devices, the structure of the non-DMA device register blocks will be device dependent.

15 BPK - BATTERY PACK

15.1 Purpose

The BPK battery pack sustains memory in the Pluribus system's 16K ram SME memories during short power outages.

15.2 Description

The battery pack and the memory bus share the same power supply. The battery pack requires 2 voltages: +8.5 volts (nominal) and -5.2 volts (regulated) to provide a back-up power supply. The pack obtains this power from the +15 and -15 volt supplies that drive the memory bus connected to the pack.

The pack is normally mounted on the front of the power supply. Designed to ensure that 6 SME module can be sustained for a 1/2 hour minimum, the pack has two 3-Amp Grasshopper fuses plugged into its front panel.

NOTE
The fuses MUST be removed
when the packs are in storage
or when power is not supplied
to the unit.

Leaving the fuses installed during these times will discharge the -10 volt battery and greatly reduce the life of the pack.

15.3 Indicator Lights

The battery pack has two indicator lights on its front panel.

The **CHARGE** indicator lights if the pack delivers a charge current to either battery that exceeds the amount required to maintain a charged battery.

The **DC IN** indicator indicates that both +15 and -15 volt supplies are getting to the BPK.

Loss of either voltage (or both) will cause the light to be off.

15.4 Voltage Adjustments

The pack contains three voltage adjustments:

(1) +8 V - Adjusted to 9.05 volts +/- .03 volts.

(2) -10 V - Adjusted to -11.35 volts +/- .03 volts.

(3) -VREL - Adjusted to -5.20 volts +/- .03 volts.

15.5 Preventive Maintenance

At each PM:

(1) Write down the voltage readings as found.

(2) Adjust voltage readings if out of spec.

(3) Do "Power Cycle".

(4) Recheck Voltages.

(5) Replace BPK if necessary.

Yearly: Replace all BPKs.

16 POWER SUPPLIES

16.1 Purpose

1. Provides +15V, -15V and +5V regulated dc voltages to the Infibus.

2. Provides a power status signal (PWST-N @ Pin 6B) to the Infibus.

3. Provides a line frequency signal (LFRQ-N @ Pin 7B) to the Infibus.

16.1.1 Power Failure (PWST-N)

NOTE : When the BCU and the console are configured in a normal manner, the following information is valid.

The PWST-N signal is normally "high", indicating that normal AC power exists. If the power supply's AC power drops below 105V AC, the system has approximately 3.3ms before a loss of regulation occurs.

Once the power supply drops below the level indicated above, the power supply's line-failure detection circuit generates 10KHz pulses to the Infibus using the power status signal line (PWST-N Pin 6B). The BCU detects these pulses and causes a level 4 power-fail interrupt. The BCU then starts a 2.2ms delay. The central processor (the processor electrically tied to the BCU, usually processor 0) processes this level 4 interrupt. After the 2.2ms delay, the BCU prevents further Infibus communications.

16.1.2 Power Recovery

NOTE : When the BCU and the console are configured in a normal manner, the following information is valid.

When the power goes above 105V AC, the power status line is set "high". The BCU sees PWST "high" and starts a 2.2ms delay to ensure stable power conditions. After 2.2ms, the BCU generates a level 4

power reset interrupt. The central processor handles this level 4 interrupt and brings the system up.

16.1.3 Line Frequency Signal (LFRQ-N)A

NOTE : When the BCU and the console are configured in a normal manner, the following information is valid.

LFRQ-N is low for 8.2ms each full cycle of the line voltage (about 60Hz). The BCU generates a level 4 interrupt each time a pulse occurs. The central processor handles this level four interrupt as a jiffy clock interrupt.

If this signal is incorrect or absent, a jiffy clock trap will be generated by the central processor.

16.1.4 Descriptions

16.1.4.1 Specifications

Voltages	Amperage	Order	Support Documents
+ 15	10	PS1	PS2A
-15,+5	12	PS2	PS2B
-15,+15,+5	12	PS3	PS1

16.1.4.2 PS1

Used in conjunction with the PS2, the PS1 provides the required Infibus power, and is housed with the PS2 in a separate cage that does not contain an Infibus. The PS1 and PS2 power suppies provide power to a 24-slot Infibus through cables that enter the cabinet from the rear. The PS1 provides the Infibus with +15V regulated DC power. See PS3 for specifications.

16.1.4.3 PS2

Used in conjunction with the PS1, the PS2 provides the required Infibus power. As described above, both the PSI and PS2 are housed in a cage that doesn't contain an Infibus. The PS1 provides the Infibus with +15V regulated DC power. See PS3 for specifications.

16.1.4.4 PS3

The PS3 plugs into a 16-slot Infibus. The input voltage is 105-125V AC, 47-63Hz. The regulated output is +15V, -15V and +5V plus or minus 1%. The maximum peak to peak riple and spikes is 150 mili-Volts.

Circuit Protection

1. Crowbar overvoltage protection shuts down the power supply at:

   +15V @ +19V
   +5V @ +7V
   -15V @ -19V

2. A thermostat on the sub chassis which monitors the ambient operating temperature and the case temperature of most pass transistors (critical circuits) provides overtemperature protection.

   Upon detection of overtemperature, the power supply shuts down. A built-in fan supplements the normal CMD cooling.

   NOTE: Upon discovery of a shut-down power supply (neon indicator on the front panel of P/S is on, but there is no voltage out), the power supply must be power-cycled to restart the circuit. **However,** even though the power supply may work after the power was cycled, it should be replaced and returned to Cambridge for rework.

3. Voltage protection diodes are positioned across each series regulator to prevent polarity reversal from exceeding approximately two volts.

4. Line input current is limited by a 10-amp fuse at

the front panel of the power supply.

Adjustments

Power supply adjustments can be made either through openings on the front panel, or, on older models, through the rear of the power supply that is located next to the Infibus connector. Older models returned for repair are also altered to reflect equipment changes and improvements. In this case, the adjustment pots on older power supply modules are moved to the front to facilitate easy access.

CAUTION : When adjusting the pots on the older models (from the rear), use an insulated long screwdriver(the blade should be wrapped with electrical tape).

NOTE: When voltage adjustment is required, measure the voltage at the right side of the Infibus (not at the test points) and adjust voltages to specifications at this point.

NOTE : All power supplies have a polarizing pin at the rear that prevents installation of the wrong power supply.

17 APPENDIX A - Master Pluribus Glossary

This glossary pertains to the BBNCC PLURIBUS and is unlimited in scope, and written to contain every possible word, phrase, or designation that has ever been used, is now used or is expected to be used in the future to describe the PLURIBUS.

It's main purpose is to reduce the "mystique" and confusion surrounding the Pluribus "world" and to provide clear definitions of the abundant nicknames unique to the Pluribus.

1822 - a BBN report; provides specifications for the interconnection of a Host and an IMP.

60Hz. interrupt - "Jiffy Clock". A classical interrupt occurring at the power line frequency on level 4, device number 1.

abort - QUIT.

access time - time from the initiation of the request (rise of STRB) to the presentation or acknowledgment of data (rise of DONE)

active - said of a DMA device while it is transferring data: from the writing of the end pointer to the setting of the PID level.

active I/O device - an I/O device which indicates its need for service directly, usually either by classical or pseudo interrupt; cf. passive I/O device.

ADCCP - Advanced Data Communication Control Procedures; a synchronous data communication protocol standard; promulgated by ANSI, American National Institute; functionally identical to HDLC, which is an international standard.

address halt - a feature of the control panel which halts a processor when a selected address is accessed on the bus.

address recognition - the process in which a module checks the Infibus address lines for an address which is in the range of those which pertain to that module.

address space - the set of locations accessible to (addressable by) a device; cf. memory space, I/O space, system address space, processor address space, etc.

AMLTST - a diagnostic which provides the basic testing of the AML card; requires the use of an AMT card and, therefore, is suited for use by Test City only and is not to be used on assembled machines.

amputate - to disconnect a bus (usually a processor bus) from the rest of the system by turning off the forward enable bit in all bus couplers coming from that bus.

ALD - AutoLoaD; a BBNCC/LEC card which implements the autoload function.

AMA - MLC cable terminator; drives, receives and terminates signals from the AML card; converts these signals to and from the levels required by the MLC.

AMB - see AMA

AMC - see AMA

AMD - see AMA

AML - MLC Adapter Module; a BBNCC module which allows the MLC to be connected as a polled slave I/O device in a Pluribus system.

arbitration - the act of choosing the next prospective user of a resource.

architecture - the overall hardware/software design of a computer.

ARPA - Advanced Research Projects Agency; part of the U.S. Department of Defense.

ARPA Network - a national network of heterogeneous computers linked to facilitate research; the first packet switching network and the original design environment for the Pluribus; built and maintained by BBN/BBNCC starting in 1969.

async - abbreviation for asynchronous.

asynchronous - Name of a class of communication methods over a single data wire; used by Teletype terminals, and many others; information bits are sent one at a time, at a rate agreed to by the sending and receiving hardware; data are sent in 8-bit characters, with parity bit, and start and stop bits to frame the character; no further structure of the data sent on the wire is defined -- characters merely arrive individually from time to time; most common method of connection to DEC equipment and various minicomputers, but is available on most mainframes.

attention - a classical interrupt on level 1, device number FF80, caused by pushing the "ATTN" button on the control panel.

autoload - a BBNCC/LEC module which contains some read only memory programmed to do loading from any of a number of I/O devices; when commanded by a bus signal, the autoload will initiate a classical interrupt, having first set the vector address to point to the ROM.

auto restart - see power recovery.

auxiliary I/O space - the portion of I/O space from FC000 through FDFFF.

auxiliary processor - a processor which is not number 0 and thus does not handle interrupts; buddy.

availability - ratio of actual uptime to total scheduled uptime.

AXD parity - a scheme wherein the parity bit(s) are derived from both the address and data; specifically, the parity bit of each byte is the exclusive-OR of the odd parity of the data and the odd parity of the byte address of that byte.

backbone - in a packet or circuit-switched network, the equipment which provides the basic long-distance transmission service; see also long-haul.

backplane - method used to interconnect component circuit boards, and distribute power and ground from the power supply.

backwards bus coupling - the process by which a master on a common (I/O or M/I) bus can access a slave on another bus (processor bus); used by processors to access other processors' address space.

bandwidth - the rate (bits/sec., kilobits/sec., megabits/sec.) at which information may be transferred or processed.

battery pack - BPK

bay - rack

BBC - Backwards Bus Coupling; data transfer across the bus coupler from a bus master that is on the slave side of the coupler.

BBC enable bit - bit 2 of the bus coupler control register; controls whether that coupler is selected for BBC.

BBC map - a register in the BCM which specifies the high-order address bits of a BBC reference.

BBC window - the four word region of system address space through which BBC references are performed.

BBN - Bolt Beranek and Newman Inc.; the developer of Pluribus.

BBNCC - BBN Communications Corporation; the subsidiary of BBN responsible for the manufacture, improvements and Field Service of Pluribus. Also the owner of LEC Sue Processor and its various equipment and BBNCC Hong Kong, its manufacturing facility.

BCI - Bus Coupler I/O end; the card which forms the I/O end of the I/O-to-memory bus coupler.

BCM - Bus Coupler Memory end; the card which forms the I/O end of a processor-to-I/O coupler and the memory end of processor-to-memory and I/O-to-memory bus couplers.

BCP - Bus Coupler Processor end; the card which forms the processor end of processor-to-memory and processor-to-I/O bus couplers.

BCU - Bus Control Unit; a BBNCC/LEC card which performs bus supervisory functions; it is chiefly responsible for arbitrating the use of the bus, but also arbitrates bus interrupts and other specialized functions.

BDR - Bus Driver/Receiver: a custom IC used in both BBNCC/LEC and BBNCC boards to interface with the Infibus; each IC contains four bus driver/receivers.

begin pointer - in a DMA device, the address of the first word of the buffer.

BEX - Bus EXtender; consists of BXD, BXR, and cables.

bezel - the decorative front of a bus unit which also contains an air filter.

binary synchronous - name of a class of synchronous communications protocols; messages of one or more characters are sent within data units called frames", together with error-checking and various protocol information; there are many different protocols that are known as binary synchronous; division of the data stream into characters requires synchronization of the receiver to the sender before the start of data frames, so that the receiver can properly extract characters for the incoming bit stream.

bisync - abbreviation of binary synchronous.

bit stuffing - name of a class of synchronous communication protocols; mutually exclusive with binary synchronous; HDLC (also ADCCP) is a sample of a bit-stuffing protocol; data stream is organized by framing characters known as "flags"; extra bits are "stuffed" into the data stream by the sender and removed by the receiver to avoid mistaken recognition of a flag within data; data need not be an integral number of characters.

black box - see silver box.

block transfer - the act of copying the contents of a series of contiguous memory locations to another place.

board - see card.

BPK - Battery PacK; a BBNCC module used to supply

refresh voltage to volatile SME memories during a power failure; mounts on front of power supply; charging power comes from the bus power supply; up to 6 connectors supply refresh voltage to each SME through the SME daughter board; connectors are wired in series and MUST be installed lowest number first.

BSO - a BBNCC hardware device used to implement Citinet (DDN) MSPM (modified X25); a single card providing 4 synchronous communication lines which implement the Citinet Host-IMP protocal, using a programable micro processor (National 6502).

BSOTST - see MSPMTST.

bug - a term used to signify a software or hardware problem; first coined by a U.S. Navy Lt. (female) during the 1940's when she was attempting to get a new system up and running - troubleshooting the problem, she located a real bug causing a short circuit.

buddy - the other processor(s) on the same bus; not the master; not processor 0.

buffer - a series of contiguous memory locations which holds a block of data.

bus - a common communication path between various devices (I/O controllers and processors) and memories; all devices and memories attached to the bus must obey a bus protocol, which controls how they access or respond to the bus; an abbreviation for Infibus.

bus arbiter - a BCU.

bus controller - a BCU.

bus coupler - a module which allows transactions on one bus to be transformed into transactions on another bus, depending upon address; composed of a BCM, either a BCP or BCI, and connecting cables; performs other special features such as parity generation and checking, mapping, power isolation, amputation, and backwards bus coupling.

bus extender - a BBNCC/LEC module which allows one logical bus to span two bus units; the extended bus looks just like one long bus; it consists of two cards,

BXD and BXR, and two connecting cables.

buskill - a software switch in the operational program which, when used, prevents the IMP from using a selected bus.

bus timer - a 1 second timer that is held off by bus accesses; upon timing out, resets the bus; guards against hung states.

bus unit - the basic mechanical module of the Pluribus; contains various combinations of Infibusses and power supplies, and has integral cooling.

BXD - Bus Extender Driver; a card which forms the Driver part of the bus extender; plugs into the same bus as the BCU, or main bus.

BXR - Bus Extender Receiver; a card which forms the Receiver part of the bus extender; plugs into the bus which does not have a BCU, or the extended part of the bus.

byte - 1/2 word; two bytes to a word.

cable - an assembly which electrically connects two or more modules and/or external equipment; each type has a four letter designation.

card - a logic board which plugs into the Infibus; each type has a three letter designation.

CASP - a Pluribus cassette reader module.

CBT (OBSOLETE TERM) - a no longer used BBN card.

CCITT checksum - a 16 bit checksum computed with polynomial

$$X^{16} + X^{12} + X^{5} + 1$$

CCP - Communications and Control Processor, a Pluribus application which involves the collection, limited processing and routing of seismic data.

central processor - the number 0 processor electrically connected to the BCU which handles all classical

interrupts.

checksum - a number of bits associated with a block of data computed via a fixed function from the data; the implicit redundancy can then be used to detect changes in the data.

checksum-block transfer (OBSOLETE TERM)

circuit-switched network - a communications network where each communication path is explicitly set up and torn down as needed; the "circuit" is a fixed alllocation of the network's resources for the duration of the path's existence; the telephone network is primarily a circuit-switched network.

classical parity - the parity scheme wherein the source generates on writes and the source checks on reads.

CLM - Controller, Line Module; part of MLC; contains logic to interface between LIM and AML

clock - usually refers to RTC/RTT Real Time clock.

CMB - the BBNCC/LEC printed circuit board which forms the actual Infibus; holds the edge connectors for the cards.

CMD - the newest mechanical module of the Pluribus; contains various combinations of Infibusses and power supplies, and uses a squirrel cage blower assemply for cooling; provides P/S voltage test jacks and local power on/off switch.

common bus - a bus which is not a processor bus: a memory, I/O, or M/I bus.

common memory - that memory which can be accessed by all processors, that is, all memory on memory or M/I busses.

configuration - the process by which a group of Pluribus modules are selected and an arrangement designed to create a machine for a particular application; a document associated with each BBNCC system describing the location and configuration of the system; also, the operational program has a software dynamic configuration process.

consensus - the agreement between processors that to take a particular action would be in their common interest; also refers to the process by which agreement is reached.

console - usually refers to the control panel; also can refer to the console terminal.

console terminal - an asynchronous I/O device connected through the PSB to allow communications with the Pluribus using DDT.

contention - the situation where multiple users are attempting to simultaneously use a resource; this usually causes delay.

continuous read/write - a feature of the control panel which, when set, repeatedly performs the access requested by depressing the "read" or "write" buttons; a switch located at the bottom center of the control panel.

control panel - a BBNCC/LEC module which allows manual reading and writing of addresses (typically memory locations), processor registers, starting or processors, and other special functions; consists of two cards, PCB and PBI, connected by a DIP connector cable, and a front panel, SWB, which connects to the other cards via three ribbon cables.

control register - a location associated with a module whose bits correspond to program settable functions; in a processor, register 15; in an LEC serial or parallel interface, the address of the device + 6; in a bus coupler, set by the jumpers on the BCM.

controller - any I/O device interface.

cooling module - the external shell of a bus unit which provides mechanical support for the Infibus chassis, blower/fan pack, bezel with filter, and airflow isolation and deflection.

CPA - a BBNCC/LEC card which forms part of the processor.

CPB - an early BBNCC/LEC card which is used to form part of the processor; superseded by the CPC, but still

used.

CPC - a BBNCC/LEC card which forms part of the processor.

CPZ - a BBNCC daughter board being installed on CPC cards to replace a chip no longer manufactured.

CPU - Central Processor Unit; more generally, but incorrectly, a processor.

crash - when the operational program is unable to start a processor, the program reports a crash; the crash number is a bit mask of which processors can't be started.

CRC-16 checksum - a 16 bit checksum computed with polynomial

$$X^{16} + X^{15} + X^{2} + 1$$

crosspatch - a debugging tool; describes the hardware and/or software procedure of connecting the Receiver and Transmitter of a Modem or Host interface together so that it talks to itself.

cycle time - the time from the beginning of a request until the device has completed all activity related to that request and is ready to start or accept another; usually longer than access time.

cyclic checksum - a checksum computed by dividing the data by a specific polynomial and taking the remainder; see CCITT checksum, CRC-16 checksum.

D-cable - a cable which connects a card plugged into an Infibus with the fantail.

DBAL - Dual Bus Access Logic, a custom IC that contains much of the logic necessary to be a bus master; provides a means for a device to access the Infibus.

DDT - Dynamic Debugging Technique; a program which allows the user to inspect and change memory locations and processor registers, start and stop processors, copy memory, field traps and other useful things; in many ways, can be thought of as a simple executive,

providing an environment for user programs; is part of all operational programs; must be loaded with some of the older diagnostics.

deadlock - a state in which two (or more) processes (hardware or software) are each waiting for a resource held by the other; each now waits indefinately for the other's resource to become available.

debugging - the procedure(s) used to identify and locate a hardware or software problem.

debugging tool - any tool, document or program used in the process of debugging a hardware problem.

device - usually a module that performs I/O functions; sometimes refers just to DMA devices.

device dependent - a register or bit whose interpretation or function is determined by the particular module with which it is associated.

device independent - a register or bit with a common interpretation or function over a range of different module types.

device register block - the eight word segment of address space which is associated with a DMA device.

DEVTST - a diagnostic which finds and tests the DMA devices.

device type - a number indicating the type of the associated module; usually program readable in the high order byte of the first register of the device.

device number - a number indicating the number of the associated module; usually program readable in the low order byte of the first register of the device; associated with each device which causes classical interrupts; when servicing an interrupt, this number can be read from the first word of the interrupt vector, indicating which device caused the interrupt.

discovery - the process whereby an operating system finds, through probing its environment, the configuration of the machine it is running on; also see staging.

Distant Host - a Host interconnected via a bit serial, handshook 1822 interface HST), usually up to distances of 2000 feet.

DMA - Direct Memory Access; a BBNCC card which performs the bus interaction and pointer management for I/O devices; the technique of allowing high-speed I/O devices access to system memory through a bus for highspeed transfers without processor intervention.

DMACHK - a diagnostic which checks one DMA device, or two connected DMA devices; can be used with a scope to enter a loop where the various parameters do not change.

DMA device - an I/O device which uses a DMA; it transacts directly with memory.

DMAFOO (OBSOLETE TERM) - an obsolete diagnostic.

DMATST (OBSOLETE TERM) - an obsolete diagnostic.

DONE - an Infibus signal that indicates successful completion of a bus access cycle; also serves as the strobe for data in a read access.

doubled cable - a cable which connects the two parts of a doubled interface with the fantail.

doubled interface - an interface which, for reliability considerations, consists of two modules on different busses, connected such that either one can serve the same external equipment.

elastic buffer - a buffer which allows input and output to proceed asynchronously, at different rates.

ELI - Eia Line Interface; a BBNCC/LEC card which provides two independent RS-232 interfaces on a single card in an MLC; it is the same as the ELIA card except, it cannot provide some control bits that may be required in some configurations; it is NOT interchangeable with the ELIA card in any system.

ELIA - Eia Line Interface; a BBNCC/LEC card which provides two independent RS-232 interfaces on a single card in an MLC; it is the same as the ELI except, more chips have been added to provide additional control

bits that may be required in some configurations; it is NOT interchangeable with the ELI card in any system.

ELIU - Eia Line Interface; provides one RS-232 interface in an MLC.

end pointer - in a DMA device, the address of the last word of a buffer.

executive core - usually refers to locations 0-5E; the area in which interrupt, QUIT, and ILLOP information is stored.

EXHIT - HIT EXecute; provides an interface (DDT) between the HIT diagnostic and the operator.

EXY - Eight X and Y; a BBNCC/LEC card which forms part of an 8K core memory; contains the core stack itself.

F-cable - a cable which connects external equipment to the fantail.

fan pack - a chassis containing six fans that provides the cooling for each bus unit on older Pluribus bus units.

fantail - a panel which contains connectors for cables from external equipment which interface to internal cables; used to facilitate the reconnection of external equipment.

feedback parity - the parity scheme wherein the destination generates and the source checks parity on all transfers.

Field Service Report - FSR.

flip-flop - a bi-stable multivibrator; a two-stage multivibrator circuit having two stable states; in one state, the first stage is conducting and the second is cut off; in the other state, the second stage is conducting the the first stage is cut off; a trigger signal changes the circuit from one stable state to the other, and the next trigger signal changes it back to the first state.

flop - flip-flop.

flow control - mechanisms which meter the flow of data into the nodes of a communication network; modern protocols, such as X.25 and HDLC, include flow control provisions.

force reload - a scheme by which memory locations may be loaded, processors started, etc., via special, heavily passworded messages on modem lines; used for remote start-up of machines; uses an RLD card.

F-stick - a hardware procedure to map an address in I/O space via the implicit fixed mapping; done on PID for single-bus machines (PLI or VDA).

FSR - Field Service Report; a document which, after filling in the blanks, identifies the site, it's problem, the solution, and the total hours, parts and other costs association with that trouble call; this history is further used to plan future operations and compute the budget; among other things, it creates a history.

full duplex - a communications path where transmission can take place in both directions simultaneously.

gateway - an interconnection between two packet-switching networks.

ground modem - a land line modem; not a satellite modem.

half duplex - a communications path wherein transmission can take place in either direction, but not both simultaneously.

halt(ed) - a state of the processor wherein instructions are not being executed, interrupts cannot be honored, and the registers are externally accessible.

HDLC - High-level Data Link Control; a bit-stuffing, synchronous data communication protocol; evolved from IBM's SDLC, HDLC is now an international standard promulgated by ISO; identical to ADCCP; a superset of X.25 using LAPB).

header - a socket resembling a chip socket except it has pins on the top and bottom; the top pins are

jumpered to provide configuration for the device and then installed in a chip socket on the card to provide connections to the circuit; also, describes the front portion of a data message or packet containing control information for sender and receiver.

hex - abbreviation for hexadecimal, base 16 numbering system; used extensively, but not exclusively, with the Pluribus.

high speed host - a BBNCC module which interfaces with certain high speed Hosts at NSA; provides up to 300KB throughput; uses a modified HST.

high speed modem - a BBNCC module which interfaces to a Bell 306 modem at speeds up to 1.5M baud; consists of MUR, MHX, and DMA.

HIT - the general Pluribus diagnostic; primarily tests common memory, I/O devices, and the bus couplers; sometimes mistakenly called EXHIT.

HITMSPM - HIT + MSPM diagnostic.

HLC - Local Compatible Host; a BBNCC card that forms part of a Local Host, low speed interface.

Host-IMP - The portion of the IMP program that handles input from Hosts.

Host - a computer which provides and uses the actual network services; connected into the network via an IMP; unlike a terminal, which typically communicates with the network in asynchronous protocol, a Host will use a packet-oriented interface protocol.

Host interface - a BBNCC module which interfaces to a local or distant Host; comprised of a DMA and an HLC or HST.

hot code - frequently executed code which is located in local memory.

HST - Local or Distant Host interface card; low or high speed; a BBNCC card which forms part of a Host interface; replaces the HLC.

IBM - four card interconnect module.

IBM checksum - CRC-16 checksum.

ICM - three card interconnect module.

ICMM - three card interconnect module used with a high speed modem; contains extended pins to connect D-cable and RLD cable.

IDM - two card interconnect module.

IDMH - two card interconnect module used with Host interface containing an HST; contains extended pins to connect D-cable.

I-cable - a cable which connects two internal cards.

idle - a state of the processor wherein no instructions are being executed, registers are not externally accessible, but interrupts may be honored.

ILIU - Current Loop Line Interface - a card which provides two independent
loop interfaces on a single card in an MLC.

illegal operation - trap caused by attempted execution of an instruction not in the repertoire of the processor.

ILLOP - illegal operation.

ILLOP vector - the four word block holding information pertinent to the current ILLOP; starts at 20 for processor 0, 30 for processor 1; contents are: illegal instruction, status, program counter, and address of service routine.

IMP - Interface Message Processor; the node computer of a Network, which performs the basic packet-switching functions.

IMP-Host - the portion of the IMP program that handles output to Hosts.

Infibus - the bus which physically and electrically connects the cards of a Pluribus system.

instruction set - the language understood by a processor; its instructions and the definition of how

it interprets them.

interface - a module which allows access and information flow to and from external equipment.

internal clock - a clock located on MLR and MUR cards; provides clocking signals for certain diagnostics during crosspatch testing.

internet - the technique, and the protocol, used in the ARPA community for communicating acrosss interconnected networks.

interrupt - the diversion of the control stream of a processor in response to an external event; the device number of the interrupting device, status and program counter at the time of interruption are saved and the processor jumps indirect through a fixed location; also refers to the bus transaction which causes the interrupt.

interrupt vector - the four word block holding information pertinent to a given interrupt level; for levels 1-4, starts at locations 0,8,10,18 respectively; contents are device number, status, program counter, and address of service routine.

INVENTORY - a diagnostic which finds the configuration of a Pluribus system.

I/O bus - a bus which contains primarly I/O devices.

I/O space - the part of system address space from FC000 to FFBFF; the area which may be accessed via fixed mapping from processor address space; also refers to the corresponding section of processor address space C000-FBFF).

IOKILL - A software table that controls discovery of I/O devices; each bit present in IOKILL causes one device address not to be checked.

ISI - Information Sciences Institute; University of Southern California, Marina Del Rey, Ca.; a PSAT site.

ISO - International Organization for Standardization; promulgates various standards throughout the world; HDLC is an ISO standard.

isochronous line - a serial communications scheme wherein bit timing is derived from a separate clock line, but characters may be sent at arbitrary intervals and are bounded by start and stop bits.

jiffy clock - a 60Hz. interrupt or 1/60th of a second.

JIG - a diagnostic which provides the basic bus coupler stand-alone test program.

K - 1024(decimal).

Katsuki - a term incorrectly found in some SRNs next to SME on the processor busses; also refers to Dave Katsuki, the namesake and formidable member of BBNCC's engineering department and a major contributor to the Pluribus design; the term in the SRNs translates to:

```
SME  0-8K  SA=0K  Key=0  KA=0  Split=Y  Mem.Blind=N
     8-16K SA=0K  Key=1  KA=0  Split=Y  Mem.Blind=N
```

kbps - KiloBits Per Second; a measure of throughput.

key bits - address bits 16 and 17, asserted on processor references according to the contents of a two bit register set by the SKEY instruction; used to differentiate among the various processors on a bus.

LEC - Lockheed Electronics Company; their SUE processor plant in Hong Kong is now owned by BBNCC.

level 1 interrupt - pressing the "ATTN" on the control panel or a remote (I/O, Memory or M/I bus) power failure; it has the lowest priority of the level 1 through level 4 interrupts.

level 2 interrupt - low data rate I/O devices such as card paper tape readers and punches use it; the paper tape reader and the new cassetts we use are both polled devices and therefore do not generate interrupts; has priority over level 1 interrupts.

level 3 interrupt - high data rate I/O devices such as disks, drums and magnetic tapes use it in a block transfer mode which allows DMA capabilities; none of the systems we maintain use this interrupt; has priority over level 1 and level 2 interrupts.

level 4 interrupt - local power fails, power restarts or jiffy clock interrupts use this interrupt; it has the highest priority of level 1 through level 4 interrupts.

level 5 interrupt - an ILLOP; illegal operation interrupt.

level 6 interrupt - a QUIT interrupt.

LL - Lincoln Labs; Hanscom Air Force Base, Lincoln, Ma.; a PSAT site.

local Host - a Host interconnected via a bit serial, handshook interface, usually over distances of less than 50 feet; longer runs are possible if a common ground can be insured.

local memory - memory on a processor bus, as opposed to common memory.

local network - a geographically-localized packet network, typically allowing communications of up to a few kilometers maximum.

lock - a data structure (usually a single word) used to interlock processes; also refers to the act of reading a lock with a read-clear cycle.

Lockheed Electronics Company - formerly the manufacturer of the SUE computer systems and several Pluribus parts.

long-haul network - non-local network, i.e. one covering distances greater than a few kilometers; packet-switching is an appropriate technology for long-haul networks; may consist of dedicated, point-to-point communications lines.

loop - see crosspatch.

looping plug - a debugging tool; comes in several flavors; used to provide an external loop on a Host or Modem interface.

low speed Host - refers to the HST card. Says that the capacitors are installed on the board.

low speed modem interface - a BBNCC module which interfaces to a Bell 303 modem at speeds up to 50 KB; consists of an MLX or MHX, MLR or MUR, and a DMA.

LSI - Large-Scale Integration; the technique of placing hundreds of electronic circuits on the same integrated circuit chip; as opposed to SSI (Small-Scale), MSI (Medium-Scale), or VLSI (Very Large-Scale), the differences are mainly in the number of circuits per chip.

Magic Modem - RLD.

mainframe - a large, general-purpose computer, which typically supports many programs running simultaneously under multiprogramming or timesharing control.

map value - the 7 bit number that determines which 4K regions of processor space is referred to by accesses in the associated map segment.

map segment - one of the four 4K regions of processor address space through which accesses are made to common memory.

mapping - the act of transforming an address in one address space to that in another.

master - the participant in a bus transaction which initiates the access; e.g. the processor, when it is accessing memory.

master processor - the lowest numbered processor in the system.

MB - multi-bus.

MBPM - second member of the MXXM family; never got beyond the prototype stage.

mbps - MegaBits Per Second; a measure of throughput.

MEMCHK - a diagnostic which finds all processors and memory pages and checks each page by all processors.

MEMCLR - a program which finds all memory and erases it; works on multi-bus machines only; used as a header on diagnostic and operational program tapes.

memory - generally used to describe a storage device; specifically, a BBNCC/LEC module, either 4K or 8K, by 16 bits (or 18 bits with parity) of random access core memory, consisting of three cards: TAG, SID, and EXY, the 4K is rarely used any more; a BBNCC card, 16K, by 16 bits (or 18 bits with parity) of random access semiconductor memory, the SME, with a battery pack option good for two hours backup during a power failure; a BBNCC card, 32K, in the R & D stage as of Feb. 82, to replace two SME's, called the SMR, will contain the battery pack on the card.

memory bus - a bus which contains primarily common memory.

memory space - the part of system address space from 0 to FBFFF.

message - the unit of data communicated between Hosts; messages are broken up by IMPs into one or more packets for transmission in the subnetwork.

message processor -

M/I bus - a common bus which contains both Memory and I/O.

MHR (OBSOLETE TERM) - a BBN card formerly used with an MHX to make a high speed Modem; no know MHR cards are now in field use; replaced by the MUR card.

MHX - a BBNCC card which forms the Transmit half of a High-speed ground Modem interface.

microprocessor - a computer processor implemented entirely on a single LSI or VLSI circuit chip.

MLC - MultiLine Controller;

MLCTST - a diagnostic which provides some testing of all the MLC hardware.

MLR - a BBNCC card which forms the Receive half of a Low-speed ground Modem interface.

MLX - a BBNCC card which forms the Transmit half of a Low-speed ground Modem interface.

Modem - a piece of external equipment which converts digital signals from the computer to analog signals for communication and vice versa; also refers to Modem interface.

Modem interface - the module which interfaces to a synchronous Modem, either ground or satellite, low or high speed.

module - a unit consisting of one or more cards which performs a unified function.

MSPM - the first member in the MXXM family; see BSO.

MSPMTST - a diagnostic which tests the MSPM (BSO card); used with DDT-41; contains the means to talk to the card's micro DDT.

MSR - the BBN card which forms the Receive part of the Satellite Modem interface.

MST - the BBN card which forms the Timing functions of the Satellite Modem interface.

MSX - the BBN card which forms the Transmit part of the Satellite Modem interface.

multi-bus - a Pluribus system containing two or more busses and having memory and I/O on bus(ses) separate from the processor bus(ses); has two or more processors; MB in a diagnostic title says the diagnostic is for testing a multi-bus Pluribus.

multiprocessor - a system which contains 2 or more processors with some common resources.

Multiwire - a technology for making cards, midway between printed circuit and wire wrap in the dimensions of cost and difficulty; consists of a printed circuit card which carries power and ground, covered by a sticky insulating layer, in which insulated wires are laid to form the signal paths.

MUR - a BBNCC card which forms the Receive half of a ground Modem interface; replaces MHR and MLR cards; provides higher speeds than older versions.

MXXM - a family of microprocessor based I/O devices; a

master/slave device: as a slave, it appears as 4 status/control words in I/O space; in master mode, it is capable of reading and writing all of common memory space; the microprocessor is capable of addressing 65K bytes of memory through windows; appears as an intelligent I/O device with DMA capabilities where DMA control information is specified by a software interface.

NCC - Network Control Center; see NOC

network - any system of intercommunicating computers and terminals.

Network Control Center - NCC

Network Operations Center - NOC

OPHELP - a function of the IMP DDT that provides easy access to commonly used data in the IMP.

OUTGUIDE - a guide containg OPHELPS, DDT commands, Trap listings, Staging descriptions and other useful data to aid in debugging a system using operational software; each version of software requires its own version of the outguide.

override - a state of IMP DDT in which potentially dangerous actions may be taken (when on) ; an attempt to prevent fat fingering by the untrained (when off).

P-cable - a cable which carries primarily power.

packet - a coherent unit of data, which is handled as a unit during its life time in a network; the unit of data communicated between IMPs on Modem lines; several packets may form a message; usually on the order of one or two thousand bits.

packet-switching - a communications scheme in which packets of data from many sources are forwarded to many destinations along the same line, multiplexing the use of the line at a high rate.

packet-switching network - a network wherein decisions to allocate network resources are made for each packet of information, rather than for the lifetime of a conversation (as in a circuit-switched network); packet

switching networks claim better overall resource utilization and fault- tolerance than circuit-switching networks, at the cost of occasional extra delays.

page - a 4K region of common memory, or more generally, system address space.

PAR - I/O PARity; a BBNCC card which generates parity for references to I/O devices.

parallel interface - a BBNCC/LEC module that can interface up to 20 parallel bits of information; can be polled or use classical interrupts; used primarily as the paper tape reader interface; card type PPB.

parity - the exclusive-or of the data and address bits; also refers to schemes which detect changes by generating and later checking the parity of a collection of bits.

parity memory - memory which is 18 bits wide, allowing a parity bit to be stored for each byte.

passive I/O device - a device which must be polled, does not interrupt; cf. active I/O device.

password - a specific combination of data bits which must be written in order for an action to take place; used for reliability considerations.

PBI - Panel Bus Interface - a BBNCC/LEC card which forms part of a control panel.

PBM - Processor Bus Memory; a possible future BBNCC module to provide improved memory and I/O capabilities on the processor busses.

PCB - Panel Control Board; a BBNCC/LEC card which forms part of a control panel.

PCD - PreCeDence passer; a BBNCC card which serves only to pass the precedence pulse by an empty slot; used for debugging.

PDU - Power Distribution Unit; usually refers to a BBNCC module which accepts site power and distributes it, with appropriate switches, circuit breakers and indicators; also refers to a BBNCC/LEC module which

provides two key switches, one for power, the other for processor selection.

peripheral - an I/O device or its controller.

PID - Pseudo Interrupt Device; a BBNCC card which serves as a hardware pending task queue.

PID level - the number that a device or a processor writes to the PID to signify that the associated task should be run.

PIDTST - a diagnostic which provides basic testing of the PID; works on single or multi-bus machines.

Pluribus - a line of modular, reliable, multiprocessor/minicomputer systems produced by BBNCC.

poll - the act of periodically checking a device to see if some event has occurred, as opposed to the device doing its own notification when a change in status occurs.

poller - see SSP.

port - an access interface to a part of a system; terminal port is an access point for a terminal to a computer.

power fail - a classical interrupt which occurs 2.5ms before bus operations are ceased preparatory to complete power loss; occurs on level 4, device number 2.

power restart - a classical interrupt which occurs on restoration of local bus power on level 4, device number 4.

power sense - an Infibus signal that indicates the condition of bus power, gives advance notice of a power failure; also refers to circuitry in the bus coupler that checks the status of power at each end of the coupler, allowing one end to disregard signals coming from a card with inadequate power.

power supply - a BBNCC/LEC module which supplies Infibus logic power and a 60Hz signal; comes in two styles: internal (plug-in, 5951 {PS3}) which takes 8 of

the 24 slots of an Infibus, and external (stand-alone, 5952 {PS1 and PS2}) which requires its own bus unit.

PPB - Peripheral Parallel Buffer; parallel interface; used to connect to paper tape reader; lower half can be used to connect to a paper tape punch.

precedence pulse - an Infibus signal which is daisy-chained between cards; used to resolve priority for the selection of the next bus master.

primary I/O space - the portion of I/O space from FE000 to FFBFF.

printed circuit - a technology for fabricating cards which involves etching away copper-clad epoxy boards to form the signal paths.

private memory - local processor memory.

processor - a BBNCC/LEC module which executes instructions; consists of two cards, CPA and either CPB or CPC; three microcode versions exist: standard, business, and scientific; also see SSP.

processor address space - the address space seen by an individual processor; includes available windows; 32K words long.

processor bus - a bus which contains processors and local memory.

processor bus address space - the aggregate of the four potential (only 2 used to date) processor address spaces on a processor bus; includes available windows; 128K words long; (64K words long for a two processor bus).

processor memory - local private processor memory; 8K maximum per processor.

PROKILL - a software table that controls discovery of processors in operational software; each bit prevents use of a processor.

PS1 - Power Supply; part of 5952 external (stand alone) power supply; supplies +15V.

PS2 - Power Supply; part of 5952 external (stand alone) power supply; supplies +5v & -15V.

PS3 - Power Supply; 5951 internal (plug-in); supplies +5v, +15v, and -15v.

PSAT - a Pluribus Satellite system; a pluribus containing an SMI and is part of the Wideband Satellite Network.

PSB - Peripheral Serial Buffer; serial interface to connect the Pluribus system to a console terminal or Pluribus cassette reader.

pseudo interrupt - the act of writing a number to the PID to indicate that a task associated with that number should be performed.

PSTOP - a program which halts all processors in a multi-bus Pluribus; combined with MEMCLR and used as a header for diagnostics and operational programs.

PTIP -generally, but incorrectly, used to identify a Pluribus system containing an MLC and/or SCANNER, called a Pluribus TIP; specifically, the one of a kind Pluribus system at 10 Molton St., Cambridge, Ma.; has it's own unique software, 6 bays, 7 MLCs, 6 processors, 160K of common memory, and 26 plus or minus a few interfaces, some of which are doubled; nicknamed "JON's anchor."

PTR - Paper Tape Reader.

PWRTST - a diagnostic which tests the processor bus power fail and reset interrupts, including the auto-reset timer facility and a rudimentary check of the 60 Hz (jiffy clock) interrupts.

QUIT - an Infibus signal that indicates abnormal completion; e.g. non-existant device, malfunctioning device, parity error, etc.; also refers to the trap that is taken when a processor-initiated access results in a QUIT.

QUIT timer - the timer on the bus arbiter (BCU) which regulates how long the bus will wait for a DONE before deciding that the intended slave will not respond and thus should issue a QUIT, terminating the access; these

timers have different values on different bus types.

QUIT vector - the four word block of memory which records information pertinent to the most recent processor-initiated QUIT; contents are: address referenced, status, program counter, and address of service routine; located at 28 for processor 0, at 38 for processor 1.

rack - the unit which houses bus units; up to five bus units, a PDU, and a fantail may be mounted in one rack.

read - an access in which data is transferred from slave to master.

read-clear - a read-modify-write access in which the written data is zero.

read-modify-write - an access in which data is read from memory and then potentially) different data is written back to the same address, all within one memory cycle.

real time clock - RTC/RTT.

refresh - the memory function of periodically rewriting each memory location so that data are not lost; certain semiconductor memory technologies, such as those used in the SME, require refreshing.

reload card - RLD.

remote buddy processor - processor 1 on a processor bus that does not contain the lowest numbered processor in the system; does not contain the console module.

remote master processor - processor 0 on a processor bus that does not contain the lowest numbered processor in the system; does not contain the console module.

remote power fail - a classical interrupt which indicates that a common bus's power is failing and about 2.5ms of usable power remains; occurs on level 1, device number 1.

remote processor - a processor on a remote processor bus.

remote processor bus - a processor bus not the master processor bus; not containing the lowest numbered processor in the system; does not contain the console module.

remote reset - the low-order bit of a bus coupler's control register; when cleared, causes a reset to occur on the bus which the BCP is plugged into.

reset timer - see bus timer.

resource - a part of a system needed by more than one of the parallel users and therefore a possible source of contention.

ribbon cable - a multiconductor cable made of several parallel wires bonded together in a flat shape.

ribbon - the part of an algorithm associated with a single PID level; may be one or more strips.

RLD - ReLoaD; a BBNCC module which allows remore forced reloads.

RLDTST - a diagnostic which tests the RLD's ability to force a reload into a Pluribus.

routing - a software subroutine which calculates which Modem or Host interface to communicate with; in networks, the function of choosing an appropriate path for packets, subject to available network resources.

RTCTST - a diagnostic which provides testing of the RTC/RTT; works on single or multi-bus.

run - a state of the processor in which intructions are being executed, interrupts may be honored, and registers are not externally accessible.

round robin - a feature of the bus arbitration scheme which enforces fairness of access to the bus; when a device has been granted a bus access that terminates normally, it is not allowed to request another access until all those devices on that bus which are currently requesting access have been granted same.

RTC - Real Time Clock; a BBN card which causes PID levels at intervals of 12.6ms and 1.6ms, has a program

readable 16 bit counter that increments each 100us, and three 16 bit readable switch registers; one of the readable switch registers is used to set the IMP number.

SACK timer - a timer on the bus arbiter (BCU) which guards against the case wherein a potential bus master requests an access, subsequently stops the precedence pulse, indicating that it will be the next master, but fails to assert SACK, acknowledging this fact.

Satellite modem interface - a BBN module which interfaces to a satellite ground station transmitter; has features to enable use of the satellite channel in broadcast mode, such as provision for switching the carrier and accurate timing of transmission and receipt of packets; consists of four cards: MSR, MST, MSX, and DMA.

scientific processor - the processor module with extended instruction set, including multiply, divide, double precision, etc.

select cycle - that part of an access which is concerned with selecting the next master.

self interrupt - a trap.

serial interface - a BBNCC/LEC module that interfaces asynchronous (start/stop) I/O devices; strappable for various speeds, character sizes, EIA vs. current loop, Modem options, etc.; is half duplex and can either be polled or use classical interrupts; card type PSB.

service cycle - that part of an access wherein the master actually transacts with the slave.

shared - common

SID - Sense and Inhibit Drivers; a BBNCC/LEC card which forms part of a 4K or 8K core memory.

silver box - a Bell 303 Modem simulator; provides cabled interface between two Modem interfaces less than 100 feet apart; provides external looping capabilities in both directions.

SIMP - Satellite Interface Message Processor; an IMP

which uses broadcast satellite channels as some of its communications links.

simplex - a communications path wherein communication can take place in only one direction.

single step - the act of causing a processor to execute a single instruction and then halt.

slave - the participant in a bus transaction which responds to the master's request; e.g. the memory, when the processor is accessing it.

SLI (OBSOLETE) - Synchronous Line Interface - a BBN card which interfaces two synchronous lines; a passive device which must be polled.

SME - Semiconductor MEmory; a BBNCC card, 16K by 16 bits (18 bits with parity) of random access semiconductor memory.

SMI - Satellite Modem Interface.

SMR - Semiconductor Memory; a BBNCC card, 32K by 16 bits (18 bits with parity) of random access semiconductor memory; now in the R & D stages; will contain a battery pack on the card.

SMS (OBSOLETE) - Synchronous Modem Simulator; a BBN card which interfaces two synchronous data sources, giving the appearance that the Pluribus is a Modem; a passive device which must be polled.

snapshot - an operational program mechanism which allows the current contents of the registers plus other selected status information to be stored in local memory each time a processor reports a trap.

SRI - SRI International, Palo Alto, CA.; a PSAT site.

SRN - System Release Notice; a document associated with each Pluribus system describing the location and configuration of each component.

SSP - Super Sue Processor; a BBN card residing on the I/O bus of PSATs; a polled processor.

stage - the portion of the operational IMP system that

discovers and configures all of the resources of a Pluribus; attemps recovery when a resource disappears or breaks.

start pointer - begin pointer

status register - a location associated with a module whose bits report various combinations of the module; in a processor, register 8; in a serial or parallel interface, the first register; in a DMA device, the fourth and seventh registers.

step - see single step.

store and forward - a technique for routing packets through a packet-switched network, where each network node acknowledges receipt of the packet to the sender, and then forwards it to the next node, based on its local routing data base.

STRB - an Infibus signal which indicates that a master is transacting with a slave; used to strobe address and, on a write, data.

strip - a set of instructions which are executed as a unit between references to the PID.

SUE - System User Engineered; the name of the line of BBNCC/LEC parts which make up part of a Pluribus.

SUPER-CONSOLE - a diagnostic which is the standard diagnostic/acceptance test for the console, consisting of PCB, PBI, SWB, and associated cabling.

SUPER-GINS - a diagnostic which is the principle test for SUE processors; GINS stands for General INstruction set.

Super Looper - a debugging tool; used to loop Modems and Local Hosts.

SWB - SWitch Board; the front panel of the control panel.

subnetwork - the collection of node computers (IMPs) and communication lines of a network which perform the actual routing and transmission of the data.

synchronous - a technique for communicating a stream of data, where individual bits of the data stream are referenced to the pulses on a separate clock connection (one each for transmit and receive directions); occuring at fixed time intervals.

synchronous line - a communications line where the timing information is derived from the transitions between data bits; data is usually sent in blocks.

synchronizer - a device which resolves two asynchronous time references.

system address space - the address space of common busses; seen directly by I/O devices and indirectly by processors after mapping; 512K words long.

system release notice - SRN.

TAC - Terminal Access Controller - a C/30 or H316 which is a host on the ARPA Network and provides access for up to 64 terminals.

TAG - Timing And Gating; a BBNCC/LEC card which forms part of the 4K or 8K core memory.

task - a programmer's term for a module that does packet forwarding inside the IMP.

Test City - The BBNCC Pluribus board repair facility located in Cambridge, Ma.

three phase wye - the type of AC power that the Pluribus PDU requires; there are five wires: one is for protective (green) ground; one is common for the other three, each of which has a 117 volt AC potential with the common, but the phase of the legs are staggered by 120 degrees.

throughput - the rate at which information can be processed.

TI - usually refers to Texas Instruments Silent 700 series terminals.

timer - a device, hardware or software, which watches over the activity of a part of the system; the timer is periodically reset by the occurrence of an event which

signifies correct operation and which should occur periodically; should a specified time interval elapse without a reset, the timer will "time out" initiating some remedial action.

TIP - Terminal Interface message Processor; an IMP with built-in simple Host capabilities which allows users at terminals access to the network, obviating (making unnecessary) an external Host computer.

TOD - Time of Day; a modification to a pair of PPB boards to interface a Systron-Donner clock.

TODEM - Tony's Modem; a BBN short-haul Modem.

TRANBOX - provides cable communications between two Modem interfaces, separated by more than 100 feet.

trap - either a QUIT or an ILLOP; in the IMP, an ILLOP causes its "number" (such as the low 13 bits of the instruction) to be reported to the NOC, along with register values at the time the ILLOP was executed; usually reports a harware or software failure; sometimes reports a suspiscious event which occasionally happens; NOC calls these reports "traps".

TTYTST - a diagnostic which tests the PSB and its associated terminal; also a test program located on some systems which can be used to exercise TIPs.

very distant Host adapter - VDA

very distant Host - VDH

VDA - Very Distant Host Adapter; contains a VDH and the electronics to provide a communications link between a local Host and an IMP, usually over a land line.

VDH - Very Distant Host; a Host connected to the IMP via a communication link, with associated error detection and retransmission protocols.

VISTAR - refers to an Infoton VISTAR crt terminal.

VISUAL 200 - refers to a DEC crt terminal.

VLSI - Very Large-Scale Integration; integrated circuit chips having several hundred to several thousand or

more circuits per chip.

watchdog timer - cf. timer.

Wideband Sattelite Network - The Pluribus Satellite Network.

window - one of the four 4K regions of processor address space which can be mapped onto a page of system address space.

wire-wrap - a technology for making boards wherein connections are made by wrapping a wire around a pin located adjacent to the component.

word - the basic element of data; has 16 bits and sometimes 2 parity bits; words between pluribus busses are 20 bits; there are two bytes in a word.

woven cable - a multiconductor cable constructed by weaving together several twisted pairs with a nylon thread.

write - a bus transaction where the data flow is from master to slave.

X.25 - shorthand for the interface described in Recommendation X.25 of the CCITT.

xpatch - see crosspatch.

ZCP - Pluribus Card ejector tool; "Zee Card Puller."

PLURIBUS GLOSSARY PARTS LIST- REVISED APRIL, 1983

NAME	BBNCC P/N	N/R	TYPE	DESCRIPTION
ALDWM	2102294G01	R	CIRC CD PC	Autoload card with BBN mod
B16C	1802402G01	N	CARD ENCL	16 LOC
B16NP	2202391G01	N	CARD ENCL	WITH BACKPLANE
B16NW	1802605G01	N	CARD ENCL	16 LOC MOD
B24C	1802403G01	N	CARD ENCL	24 LOC
B24NP	2202392G01	N	CARD ENCL	WITH BACKPLANE
B24NW	1802606G01	N	CARD ENCL	16 LOC MOD
BCI	2100168G01	R	CIRC CD PC	BUS COUPLER - I/O BUS
BCMPC	2101861G01	R	CIRC CD PC	Bus coupler - memory bus, PCB
BCPMW	2100528G01	R	CIRC CD MW	Bus coupler - processor bus, multiwire
BCUWM	2102293G01	R	CIRC CD PC	Bus controller with BBN mods
BEXCB	2401320G01	N	CABLE	CABLE BUS EXTENDER
BIO	ND02613G01	R	CIRC CD	Bus Interface Unit, not designed
BOX16	2201223G01	N	CHASSIS	16 LOC WITH CMB
BOX24	2201223G02	N	CARD ENCL	24 LOC
BOXPS	2300461G03	N	CARD ENCL	24 LOC LES/PS
BPK	2302400G01	R	PWR SUP	Battery Pack Module for SME
BSO	2102033G01	R	CIRC CD PC	4 channel bisync interface
BX16	1802397G01	N	CHASSIS	16 LOC PS/BXD/BXR
BX24	1802398G01	N	CHASSIS	24 LOC PS/BXD/BXR
BXDWM	2102295G01	R	CIRC CD PC	Bus extender driver with BBN mods
BXRWM	2102296G01	R	CIRC CD PC	Bus extender receiver with BBN mods

CASP	2302405G03	R	ASSY	CASSETTE MODULE
CMD16	2302407G02	N	ASSY	Cooling module - 16 slot cage
CMD24	2302407G03	N	ASSY	Cooling module - 24 slot cage
CMDPS	2302407G01	N	ASSY	Cooling module - power supply cage
CPAWM	2102292G01	R	CIRC CD PC	Processor card with BBN mod
CPCWM	2102291G01	R	CIRC CD PC	Processor card with BBN mod
DBNCA	2402411G01	N	CABLE	FTL/DBL
DBRSA	2402412G01	N	CABLE	KGB/FLTR BX (10X,11X = KGB/IBS)
DDR	2102413G01	R	CIRC CD PC	DISTANT DRIVER & RECEIVER H316
DHSDA	2402417G01	N	CABLE	FTL/HST/HST
DHSSA	2402418G01	N	CABLE	FTL/HST
DKGBA	2402421G01	N	CABLE	KGB/OPT I C BX (PLI)
DKGCA	2402422G01	N	CABLE	B FTL/B FTL/CONNF (PLI)
DMA	2100136G01	R	CIRC CD	Direct Memory Access card
DMBDA	2402429G02	N	CABLE	MUR/MUR/303
DMBSA	2402429G01	N	CABLE	MUR/303
DMEDA	2402430G01	N	CABLE	FTL/MER/MER
DMHDB	2402431G01	N	CABLE	FTL/MHR/MHR
DMHSA	2402431G02	N	CABLE	FTL/MHR
DMMDA	2402424G01	N	CABLE	MUR/MUR/FTL Double Mil spec modem cable
DMMSA	2402425G01	N	CABLE	Single Mil spec modem cable
DMRDA	2402435G01	N	CABLE	MUR/MUR/RS232
DMRSA	ND02612G01	N	CABLE	MUR/RS232
DPSBA	2402436G01	N	CABLE	FTL/PSB, 10'

DPSBB	2402437G01	N	CABLE	FTL/PSB, 2′
DPSC	2403650G01	N	CABLE	PSB/CASP
DPTRA	2402438G02	N	CABLE	FTL/PPB, 5′
DPTRB	2402438G03	N	CABLE	FTL/PPB, 10′
DPTRC	2402438G01	N	CABLE	FTL/PPB, 2′
DRBSA	2402439G01	N	CABLE	R PNL/CAP BX/R PNL (PLI), 2′
DRSCA	2402440G01	N	CABLE	FTL/SLI/DBL, 7′
DRSFA	2402441G01	N	CABLE	FTL/SLI/DBL/PSB, 9′
DRSPA	2402442G01	N	CABLE	SLI/FTL (PLI) (10X,11X = KGR/TBS),5′
DRSSA	2402443G01	N	CABLE	SMS/FTL (10X,11X = KGR KGB/TBS(PLI),5′
DSLD	2402444G01	N	CABLE	FTL/SLT/SLT, 10′
DSPLY	2501287-01	R	TERMINAL	INFOTON 200
DTOD	2402973G01	N	CABLE	FTL/PPB/PPB
DTTYA	2402446G01	N	CABLE	FTL/PSB, 5′
DTTYC	2402446G03	N	CABLE	FTL/PSB, 3′
EXYWM	2102299G01	R	CIRC CD PC	Core memory card - core stack
FBNCA	2402453G02	N	CABLE	FTL/TOD, 50′
FBNCB	2402453G01	N	CABLE	FTL/TOD, 30′
FDDRA	2402456G01	N	CABLE	FTL/DDR, 12′
FHDAA	2402742G01	N	CABLE	FTL/HOST, 50′
FHDAB	2402742G02	N	CABLE	FTL/HOST, 100′
FHDAC	2402742G03	N	CABLE	FTL/HOST,200′
FHSAA	2402457G01	N	CABLE	FTL/HOST with no connector 30′
FHSAB	2402457G02	N	CABLE	FTL/HOST with no connector 40′

FHSAC	2402457G03	N	CABLE	FTL/HOST with no connector 60′
FHSAD	2402457G04	N	CABLE	FTL/HOST with no connector 70′
FHSAE	2402457G05	N	CABLE	FTL/HOST with no connector 150′
FHSAF	2402457G06	N	CABLE	FTL/HOST with no connector 200′
FHSAG	2402457G07	N	CABLE	FTL/HOST with no connector 100′
FHSAH	2402457G08	N	CABLE	FTL/HOST with no connector 300′
FHSI	2402972G01	N	CABLE	H PLI FTL/516 LH
FHSSA	2402501G01	N	CABLE	FTL/HOST ext. w/conn., 30′
FHSSB	2402501G02	N	CABLE	FTL/HOST ext. w/conn., 40′
FHSSB	2402501G03	N	CABLE	FTL/HOST ext. w/conn., 60′
FHSSD	2402501G04	N	CABLE	FTL/HOST ext. w/conn., 70′
FHSSE	2402501G05	N	CABLE	FTL/HOST ext. w/conn., 100′
FHSSF	2402501G06	N	CABLE	FTL/HOST ext. w/conn., 150′
FHSSG	2402501G07	N	CABLE	FTL/HOST ext. w/conn., 200′
FHSSH	2402501G08	N	CABLE	FTL/HOST ext. w/conn., 300′
FKGAA	2402502G01	N	CABLE	B FTL/KG34-TB-4, 30′
FMHA	2402639G01	N	CABLE	FTL/306 MODEM
FMLAA	2402503G01	N	CABLE	FTL/303 ext. Bell 303 Modem cable, 30′
FSLAA	2402508G01	N	CABLE	FTL/LS SAT
FTOD	2402967G01	N	CABLE	FTL/TOD/TOD
FTS2A	2405043G01	N	CABLE	FTL/TTY
HA1T	2201921G01	N	HEADER	MUR HEADER-Mil spec protocol-terminated
HA1U	2201921G02	N	HEADER	MUR HEADER-Mil spec protocol-unterminated
HA2T	2201921G03	N	HEADER	MUR HEADER

HA2U	2201921G04	N	HEADER	MUR HEADER
HA3T	2201921G05	N	HEADER	MUR HEADER
HA3U	2201921G06	N	HEADER	MUR HEADER
HA4T	2201921G13	N	HEADER	MUR HEADER
HA4U	2201921G14	N	HEADER	MUR HEADER
HB1	2201921G07	N	HEADER	MUR HEADER
HB2	2201921G08	N	HEADER	MUR HEADER
HB3	2201921G09	N	HEADER	MUR HEADER
HC1	2201921G10	N	HEADER	MUR HEADER
HC2	2201921G11	N	HEADER	MUR HEADER
HC3	2201921G12	N	HEADER	MUR HEADER
HC4	2201921G15	N	HEADER	MUR HEADER
HST	2100538G01	N	CIRC CD MW	Host Interface card
IBCAA	2402513G01	N	CABLE	Bus coupler cable, 5′
IBCAB	2402513G02	N	CABLE	Bus coupler cable, 10′
IBCAC	2402513G03	N	CABLE	Bus coupler cable, 15′
IBCAD	2402513G04	N	CABLE	Bus coupler cable
IBPKA	2402514G01	N	CABLE	BPK/SME, 3 connector, 6′
IBPKB	2402514G02	N	CABLE	BPK/SME, 4 connector, 8′
IBPKC	2402514G03	N	CABLE	BPK/SME, 5 connector, 10′
IBPKD	2402514G04	N	CABLE	BPK/SME, 6 connector, 12′
IBPKE	2402514G05	N	CABLE	BPK/SME, 7 connector, 14′
IBPKF	2402514G06	N	CABLE	BPK/SME, 8 connector, 16′
ICB	2102517G01	R	CIRC CD WW	TOPHAT, 4 CARD INTERCONN. (SMI)

ICM	2102166G01	R	CIRC CD PC	TOPHAT, 3 CARD
ICMM	2101858G01	R	CIRC CD PC	TOPHAT, 3 CARD (RLD &D CABLE CONN)
IDDRA	2402518G01	N	CABLE	DDR/316 IMP
IDM	2102232G01	R	CIRC CD PC	TOPHAT, 2 CARD (PROCESSOR)
IDMH	2100541G01	R	CIRC CD PC	TOPHAT, 2 CARD (HST)
IPTPA	2402969G01	N	CABLE	PPB/REMEX PTR
IPTPB	2402969G02	N	CABLE	PPB/REMEX PTR
IPTRA	2402525G01	N	CABLE	PPB/REMEX PTR (INTERNAL) 10'
IPTRB	2402525G02	N	CABLE	PPB/REMEX PTR
IRLEA	2402527G01	N	CABLE	RLD/MUR/MUR
ITTC	2402970G01	N	CABLE	PSB/FTTYnoFTL
ITTR	2402971G01	N	CABLE	PSB/EIAnoFTL
KGB	2101529G03	R	CIRC CD WW	PLI BLACK
KGR	2101529G04	R	CIRC CD WW	PLI RED
MHX	2100228G01	R	CIRC CD PC	HIGH SPEED MODEM INTERFACE (TRANSM)
MUR	2101730G01	R	CIRC CD PC	MODEM CARD
OPPC	2201545G02	N	PANEL	CONTROL PANEL ASSY, LOCKHEED LOGO
PAR	2101529G08	R	CIRC CD WW	PARITY BOARD
PBIJC	2400836G01	N	CABLE	PBI/PCB (Old P4) 4"
PBIWM	2102301G01	R	CIRC CD PC	PARALLEL BUS INTERFACE
PCAS	4603634G01	N	PROM	Prog. IC, SCD-0085, used on CASP
PCBWM	2102298G01	R	CIRC CD PC	PARALLEL CONTROL BUFFER
PDB	2302539G01	R	ASSY	POWER DISTRIBUTION BOX
PDBS	2302540G01	R	ASSY	

PDBS2	ND02540G02	R	ASSY	220V
PDU	2302541G01	R	ASSY	POWER DISTRIBUTION UNIT FOR PLI
PID	2100131G01	R	CIRC CD PC	PSUEDO INTERRUPT DEVICE
PPBWM	2102297G01	R	CIRC CD PC	PARALLEL I/O CONTROLLER
PS1	2201251G03	R	PWR SPLY	BBFAN +15V 25A
PS2	2201206G03	R	PWR SPLY	BBN +5/-15V (P24BW)
PS3	2300430G06	R	PWR SPLY	BBFAN +5/-15V (P16WM)
PSBWM	2102300G01	R	CIRC CD PC	PARALLEL SERIAL BUFFER
PTR	2500247-01	R	ASSY	PAPER TAPE READER
PTRP	1802544G01	R	ASSY	READER/PUNCH RACK MTD
RFP	2202548G01	N	PANEL	FANTAIL RFP=Red Direct Path Board, PLI
RLD	2101529G09	R	CIRC CD WW	MODEM RELOAD DEVICE
RTC	2101529G10	R	CIRC CD WW	REAL TIME CLOCK
RTT	2105513G01	R	CIRC CD WW	REAL TIME CLOCK W/TEMP.SENSE
SBAT	2104399G01	R	PWR SUP	BATTERY PACK MOD FOR SMR (NOT RELEASED)
SBRSA	2402549G01	N	CABLE	BCPJ6/FLT PNL
SIDWM	2102302G01	R	CIRC CD PC	SENSE AND INHIBIT DRIVERS
SMEDB	2101596G01	R	CIRC CD PC	SME Daughter Board
SMEWM	2102550G01	R	CIRC CD PC	32K SER MEMORY SCD-0077
SMR	2103979G01	R	CIRC CD PC	SERIAL MEMORY (not released)
SWBP1	2201151G01	N	CABLE	TO CONTROL PANEL
SWBP2	2201151G02	N	CABLE	TO CONTROL PANEL
SWBP3	2201151G03	N	CABLE	TO CONTROL PANEL
SWBP4	2201151G04	N	CABLE	TO CONTROL PANEL
SWBWM	2102552G01	R	CIRC CD PC	CONTROL PANEL, BBNCC LOGO

TAGWM	2102304G01	R	CIRC CD PC	TIMING AND GATING BOARD
TDH	2202583G01	R	MODEM	Hardware Todem
TDL	2102157G01	R	CIRC CD	Analog Digital Circuit
TDP	2101448G01	R	CIRC CD PC	POWER SUPPLY BOARD
TDT	2102158G01	R	CIRC CD	Timing Board
TODEM	2302630G01	R	MODEM	MODEM ELIMINATOR MODEL 632
ZCP	8002586G01	N	KIT	CRD EJECTOR KIT
AMA	2100441G01	R	CIRC CD PC	MLC CABLE CARD/INTERFACE BD.
AMB	2100441G02	R	CIRC CD PC	MLC CABLE CARD/INTERFACE BD.
AMC	2100441G03	R	CIRC CD PC	MLC CABLE CARD/INTERFACE BD.
AMD	2100441G04	R	CIRC CD PC	MLC CABLE TERM/INTERFACE BD.
AML	2101529G01	R	CIRC CD PC	
BCMMW	2100527G01	R	CIRC CD MW	Multiwire bus-coupler memory
CBT	ND02659G01	R	CIRC CD WW	
CLMA	2302452G01	R	ASSY	(CLM=Control Logic Module) MLC logic
CPB	2102169G05	R	CIRC CD	PROCESSOR
DHD	2102414G01	R	CIRC CD PC	OBSOLETE (REPLACED BY DDR)
DHLDA	2402415G01	N	CABLE	HLC/HLC/FTL
DHLDB	2402415G01	N	CABLE	HLC/HLC/FTL
DHLDC	2402415G03	N	CABLE	HLC/HLC/FTL
DHLSA	2402416G01	N	CABLE	HLC/FTL
DILSA	2402420G01	N	CABLE	MLR/ILC (LCE)
DLCFA	2402423G01	N	CABLE	ELI/MODEM (MLC)
DLCFB	2402423G02	N	CABLE	ELI/MODEM (MLC)

DLCFC	2402423G03	N	CABLE	ELI/MODEM (MLC)
DLCMA	2402426G01	N	CABLE	ELI/MODEM (MLC)
DLCMB	2402426G02	N	CABLE	ELI/MODEM (MLC)
DLCMC	2402426G03	N	CABLE	ELI/MODEM (MLC)
DLTFA	2402427G01	N	CABLE	ELI/TERMINAL (MLC)
DLTFB	2402427G03	N	CABLE	ELI/TERMINAL (MLC)
DLTFC	2402427G04	N	CABLE	ELI/TERMINAL (MLC)
DLTFD	2402427G02	N	CABLE	ELI/TERMINAL (MLC)
DLTMA	2402428G01	N	CABLE	ELI/TERMINAL (MLC)
DLTMB	2402428G02	N	CABLE	ELI/TERMINAL (MLC)

MLC GLOSSARY

NAME	BBNCC P/N	N/R	TYPE	DESCRIPTION
DLTMC	2402428G03	N	CABLE	ELI/TERMINAL (MLC)
DMLDA	2402432G01	N	CABLE	MLR/MLR/FTL
DMLSA	2402433G01	N	CABLE	MLR/FTL
ELI	2101546G02	R	CIRC CD PC	MLC Interface Board
ELIA	2101546G01	R	CIRC CD PC	MLC Interface Board
ELIU	2100671G01	R	CIRC CD PC	MLC Interface Board
HLC	2101529G02	R	CIRC CD	host interface
IAMAA	2402511G02	N	CABLE	AML/CLM
IAMAB	2402511G01	N	CABLE	AML/CLM
IAMCA	2402512G02	N	CABLE	AML/CLM

IAMCB	2402512G01	N	CABLE	AML/CLM
ILIU	2100631G01	R	CIRC CD PC	(ILI=Curent Loop LIU Assy)
IRLDA	2402526G01	N	CABLE	MLR-MHR/RLD
LBT	2100955G01	R	CIRC CD PC	LIU Bus Terminator Assy MLC Cable
LEDA	2100854G01	R	CIRC CD PC	MLC LED assembly
LIMA	2302531G01	R	ASSY	(Line Interface Module) MLC LOGIC UNIT
LTA	2100936G01	R	CIRC CD PC	Line Driver PC Assy, MLC CABLE
MLR	2100137G01	R	CIRC CD PC	memory card
MLX	2100138G01	R	CIRC CD PC	memory card
OPCD	2201545G01	R	ASSY	CONTROL PANEL
SENAM	2100899G01	R	CIRC CD PC	(SEN=Sensing Amp) MLC LOGIC BOARD
SERME	2100704G01	R	CIRC CD PC	(SER=Single Memory PC Layout) MLC

18 APPENDIX B - Host Pinouts

1822 HOST CABLE FANTAIL PINOUTS

Signal Name	Abbrev.	Pin Number
Ready For Next Imp Bit	RFNIB+	1
There's Your Imp Bit	TYIMB	2
Last Imp Bit	LIBIT	3
Imp Data	IMDTA	4
Ready For Next Host Bit	RFNHB	5
There's Your Host Bit	TYHBX+	6
Last Host Bit	LHBIT+	7
Host Data	HSDTA+	8
Imp Master Ready	XMTOT	9
Imp Ready Test	XMTXN	10
Host Master Ready	HMRDY	11
Host Ready Test	HRDYT	12
Ready For Next Bit +	RFNBV+	13
There's Your Imp Bit +	TYIMV+	15
Last Imp Bit +	LIBIV+	17
Imp Data +	IMDTV +	19
Ready For Next Imp Bit -	RFNIB-	20
Ground	GND	21
Ground	GND	22
Ground	GND	23
Ground	GND	24
There's Your Host Bit -	TYHBX-	25
Last Host Bit -	LHBIT-	26
Host Data -	HSDTA-	27
Ready For Next Bit -	RFNBV-	32
There's Your Imp -	TYIMV-	33
Last Imp Bit -	LIBIV+	35
Imp Data -	IMDTV-	37

19 APPENDIX C - Looping Plugs

The following sections contain the wiring to produce local and distant 1822 looping plugs. The plugs are a male and female connector wired back upon themselves according to the charts, housed in a common case. This gives the capability of looping back in either direction.

19.1 1822 Distant (750589G02)

	JUMPER PIN	to PIN	
RFNIB+	1	13	RFNBV+
TYHBX+	6	15	TYIMB+
LHBIT+	7	17	LIBIV+
HSDTA+	8	19	IMDTV+
XMTOT+	9	11	HMRDY+
XMTXN+	10	12	HRDYT+
RFNIB-	20	32	RFNBV-
TYHBX-	25	33	TYIMV-
LHBIT-	26	35	LIBIV-
HSDTA-	27	37	IMDTV-

19.2 1822 Local (750589G01)

	JUMPER PIN	to PIN	
RFNIB	1	5	RFNBV
TYIMB	2	6	TYHBX
LIBIT	3	7	LHBIT
IMDTA	4	8	HSDTA
XMTOT	9	11	HMRDY
XMTXN	10	12	HRDYT

All slave modules examine the address and control lines. After a delay to allow for address deskewing, the Service Master module sets the strobe line "low" and then removes SACK to allow the BCU to determine the next Selected Master module. During the service cycle, the addressed slave module must respond with the service completion signal, DONE. If the data transfer is a write (RITE) to the addressed slave module, DONE indicates to the Service Master that the slave has accepted the data from the lines. Upon recognizing the DONE signal, the master removes strobe, address, data and control. The write operation for this master has been completed and the Infibus is either idle, or a waiting Selected Master can now set its address and control signals "low".

Overlapped selection of the next Service Master was enabled when the current Service Master removed its select acknowledge signal, SACK. If a read operation is performed during the service cycle, the Service Master sets only address and control lines "low" when the Infibus has been released by the previous master.

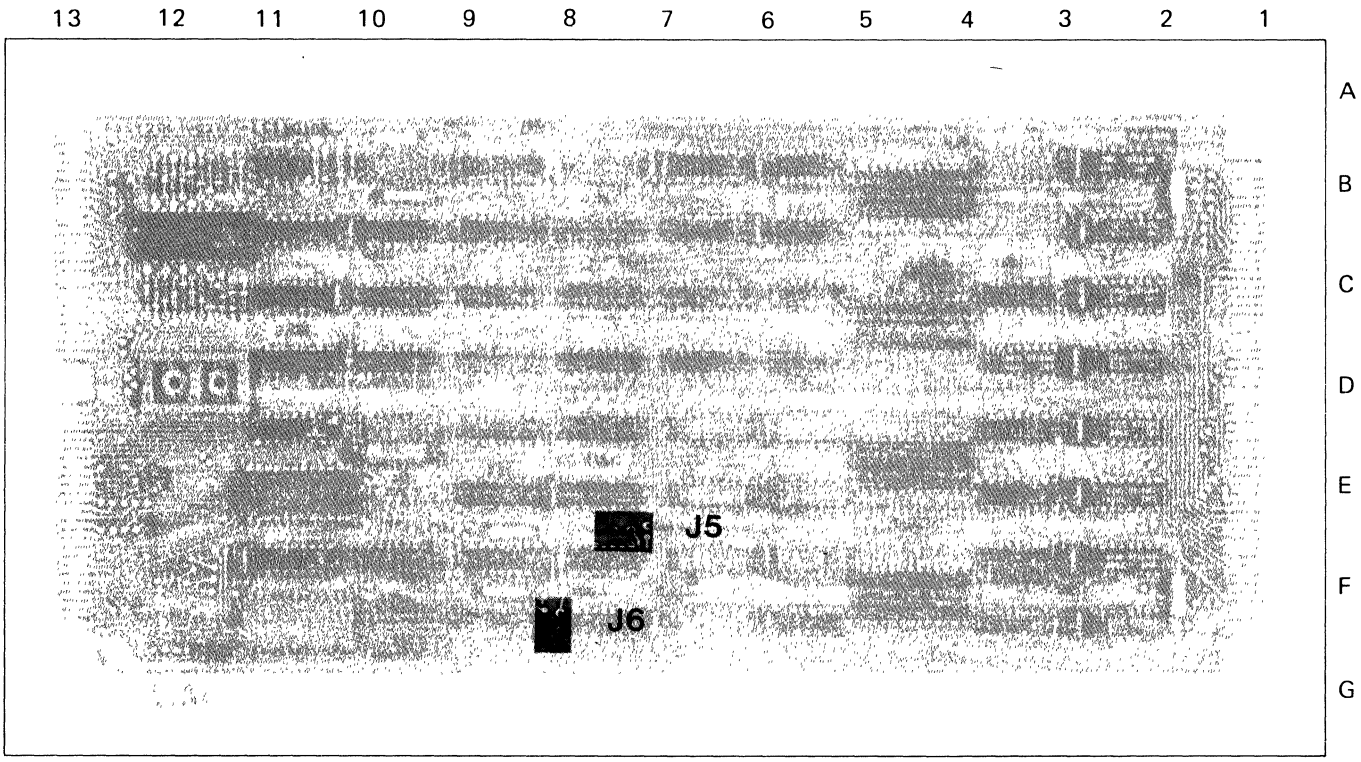
The Service Master waits for address deskewing and then sets STRB "low". Receiving STRB, the addressed slave places the data on the data lines and, after a delay, issues the DONE signal. Upon receiving DONE, the Service Master samples the read data and removes address, control and STRB. The slave removes the data as soon as STRB goes "high" to release the Infibus. This completes the read operation so that a newly selected Master can then assert address and control signals.

Overlapped selection of the next Selected Master was enabled when the current Service Master removed SACK. If no other master module is waiting, the Infibus is idle. Receipt of DONE by the master module indicates successful completion of the bus service. If DONE does not occur in approximately 5 microseconds, the BCU sets the QUIT line "low", indicating a cycle-abort condition. The occurrence of DONE or QUIT causes the master module to set "high" the lines that were set "low", including STRB.

Removal of STRB indicates to all other modules that another transfer may begin. A cycle-abort condition may be stored by the master module as a

PCB

2102298G01



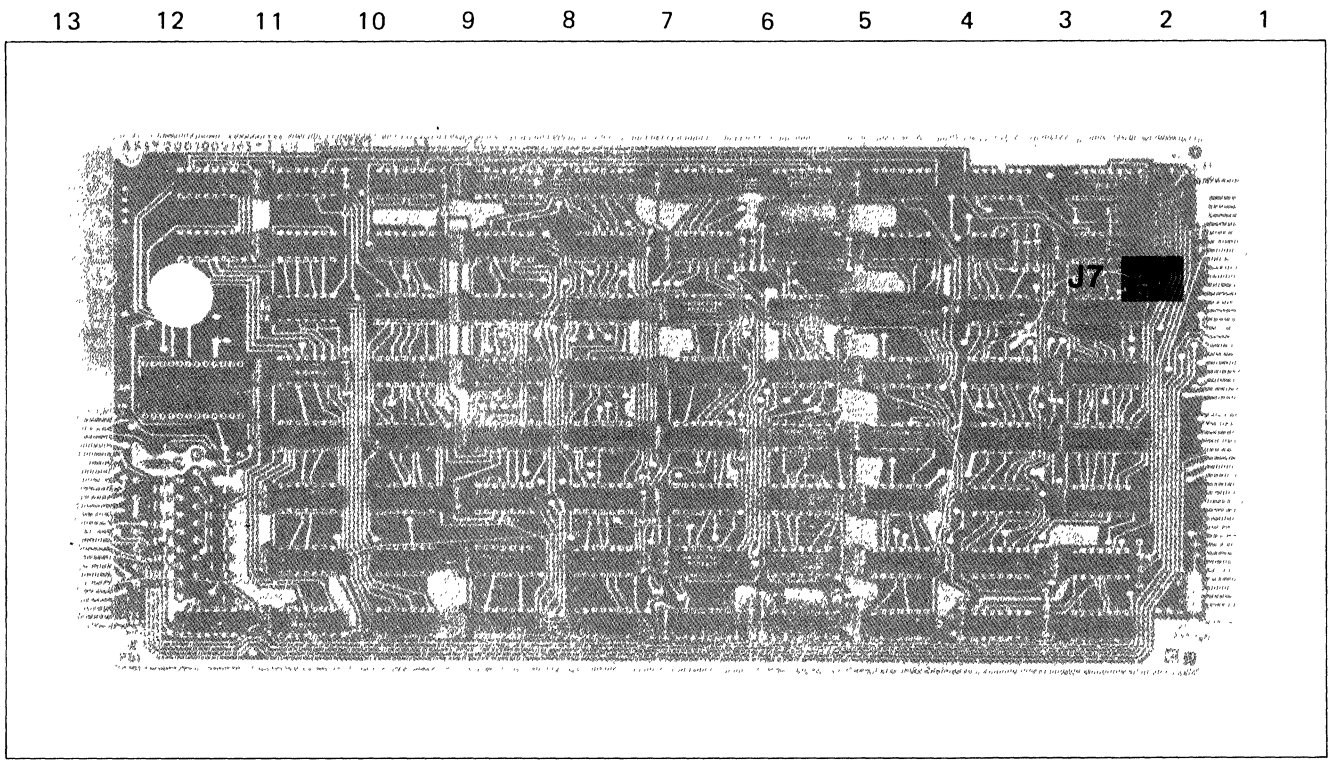
J5, J6 Control Panel Address

Panel No.	Connector J5	Connector J6
1 *	— —	— —
2	— —	1 to 2
3	1 to 2	— —
4	1 to 2	1 to 2

* Used for all applications by BBNCC

PBI

2102301G01



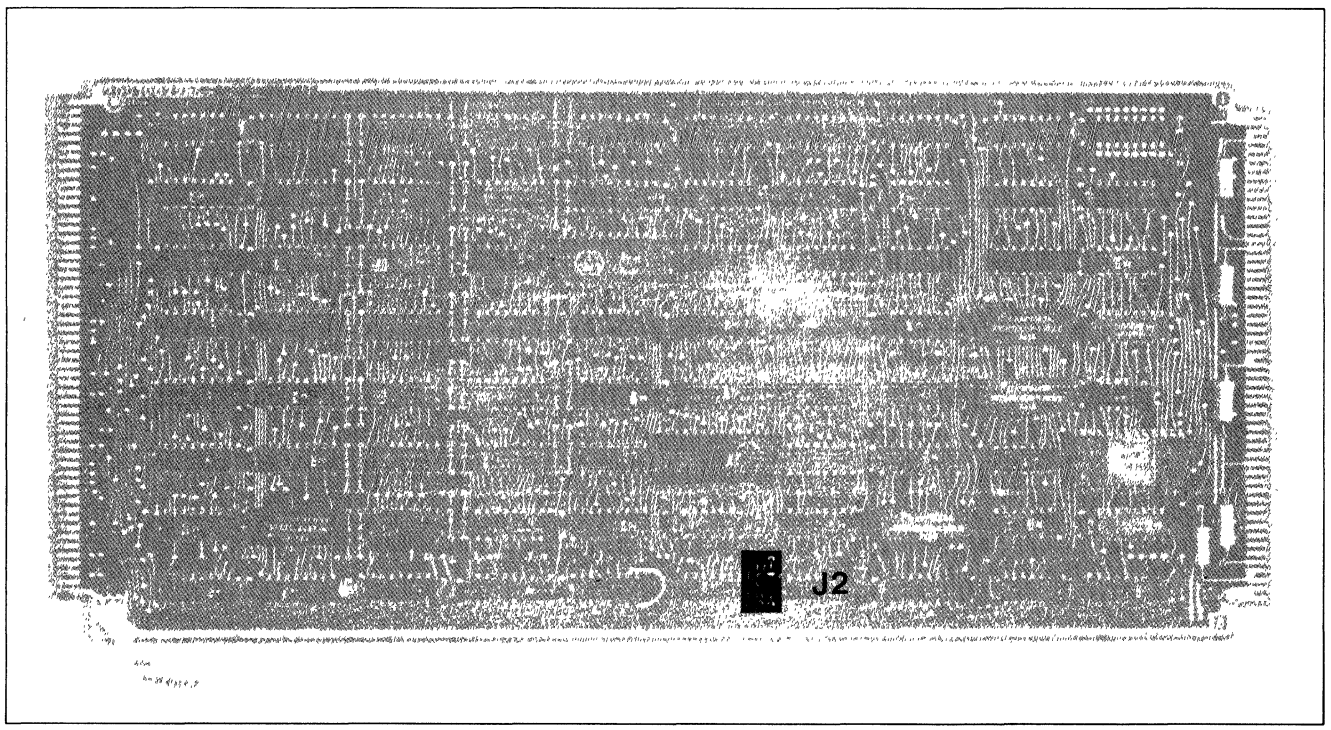
J7 Memory Protect Access

Bit	From	Connect to	
		1	0
16	J7-1	J7-2	—
17	J7-3	J7-4	—

* Used for all applications by BBNCC

CPB/C

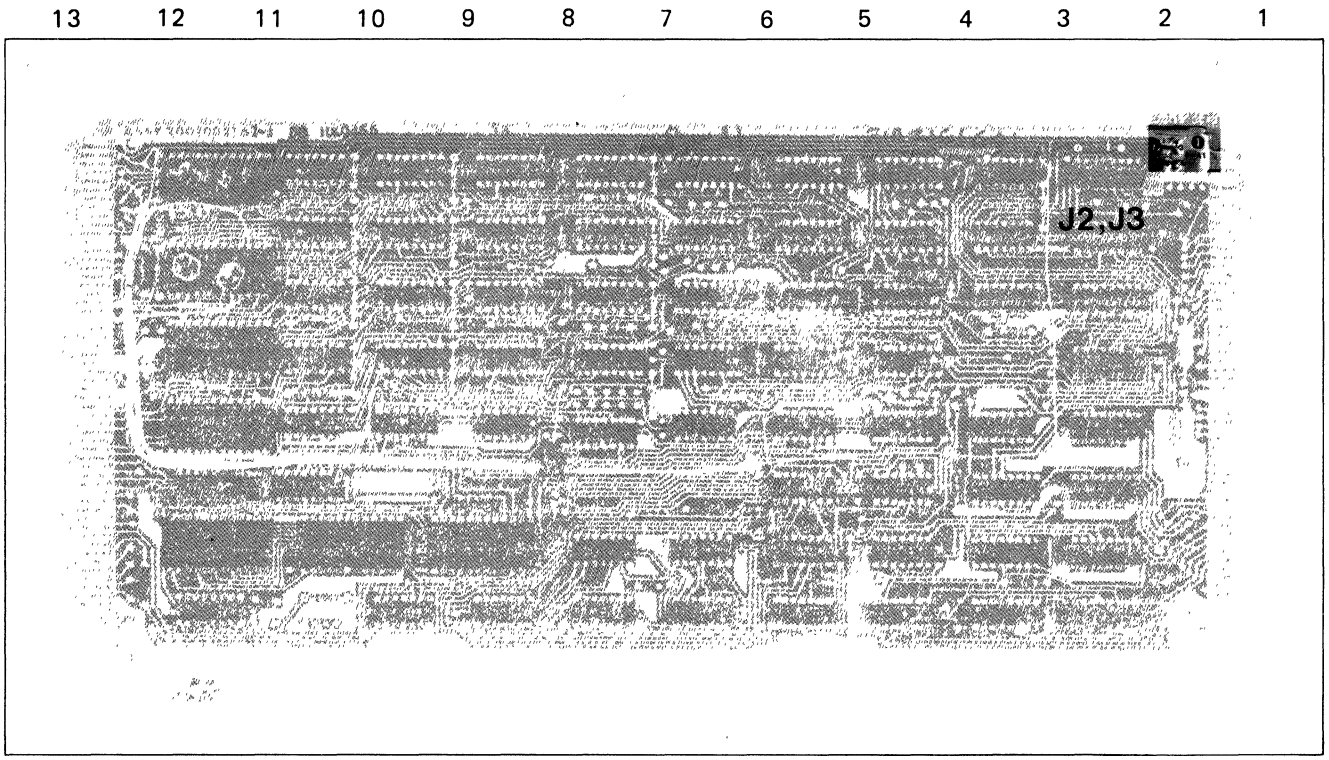
2102291G01



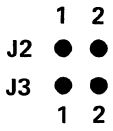
No jumpers are required
on the CPB or CPC boards
for BBNCC applications.
Field J2 should contain no jumpers

CPA

2102292G01



J2, J3 Processor Number



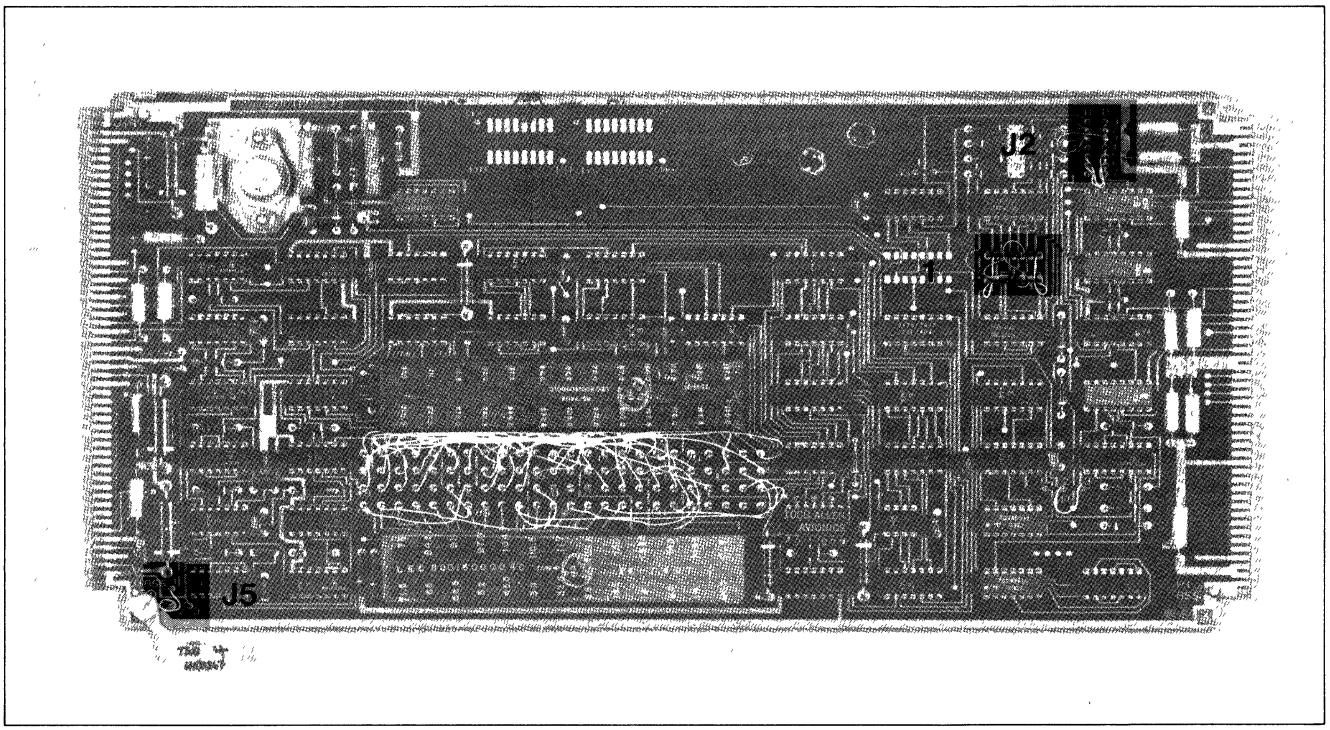
Number	J2	J3
0	1 to 2	1 to 2
*1	–	1 to 2
2	1 to 2	–
3	–	–

*All BBNCC applications

Typical Example: CPA (P20) $^{J2, J3}$ Procno = 1

TAG

2102304G01



1 Memory Address Select

Processor Local Memory

Address Range	Set These Switches						
	1	2	3	4	5	6	7
Master	X	X	X	X	X	X	X
Buddy	X	X	X	X	X	X	X

1 2 3 4 5 6 7

● ● ● ● ● ● ●

● ● ● ● ● ● ●

(Use Either Table)

Common Memory

Page	Address Range	Set These Switches							
		1	2	3	4	5	6	7	
0	0-8K	X	X	X	X	X	X	X	Key 0
200	8-16K	X	X		X	X	X	X	
400	16-24K	X	X	X	X	X		X	
600	24-32K	X	X		X	X		X	
800	32-40K	X	X	X	X		X	X	Key 1
P00	40-48K	X	X		X		X	X	
C00	48-56K	X	X	X	X			X	
E00	56-64K	X	X		X			X	
1200	64-72K	X	X	X		X	X	X	Key 2
1400	72-80K	X	X			X	X	X	
1600	80 88K	X	X	X		X		X	
1800	88-96K	X	X			X		X	
1A00	96-104K	X	X	X			X	X	Key 3
1C00	104-112K	X	X				X	X	
1E00	112-120K	X	X	X				X	
2200	120-128K	X	X					X	

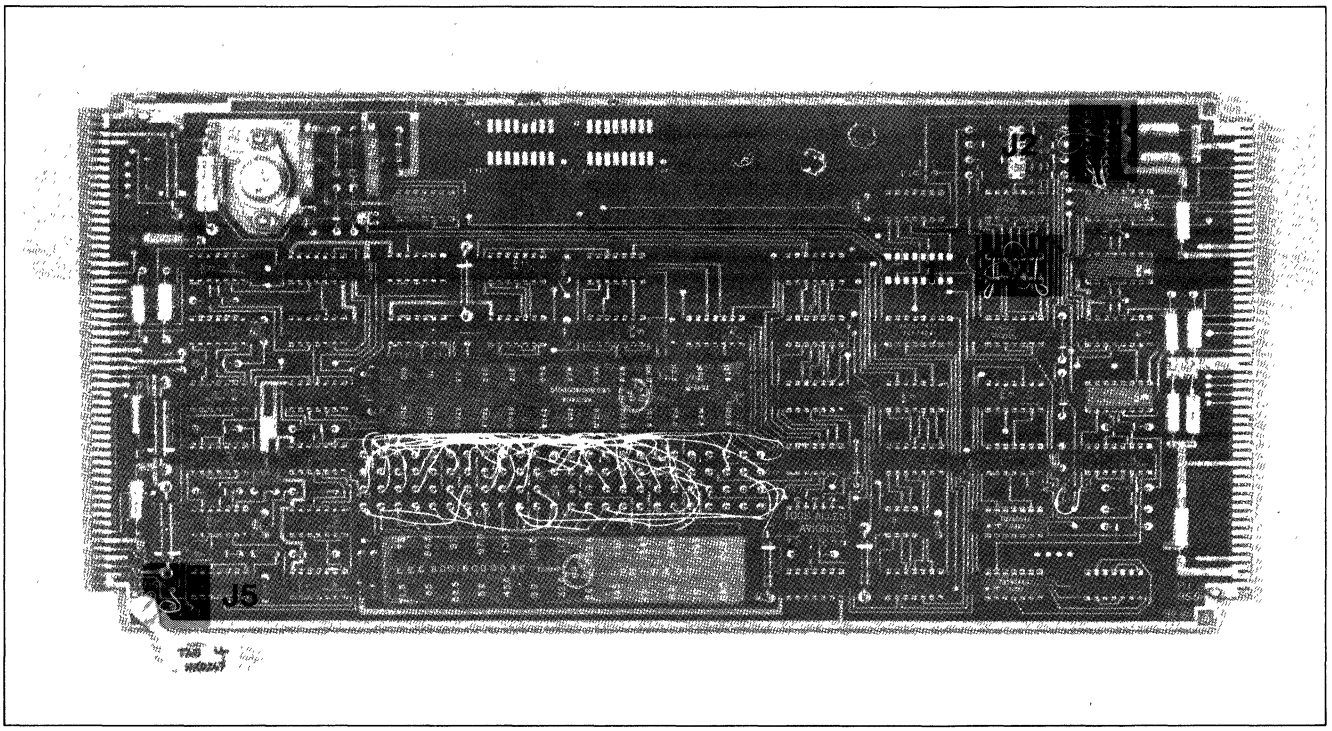
X = Jumper installed

Note On some cards these jumpers will be a bank of switches Look closely to make sure which switch is No 1

Typical Example: TAG [1] 8P [1] SA = 0 [1] Key = 1 [1] Adrec = 14-17

TAG

2102304G01



J2 Address Recognition

	A	B
1	X	X
2	●	●
3	●	●
4	X	X
5	●	●
6	●	●
7	●	●

X = Missing Pin

Address Bit	Connect		
	From	To	
17	J2-A2	J2-B2	*
16	J2-A3	J2-B3	*
15	J2-A5	J2-B5	*
14	J2-A6	J2-B6	*
13	J2-A7	J2-B7	
1	J2-A4	J2-B4	

* For 8K Memory Do Not Connect J2-A7 to J2-B7. Connect J2-A4 to J2-B4 Only for Interleave Mode

* Set for Standard Pluribus Applications

J2 Other Applications

	A	B
1	X	X
2	●	●
3	●	●
4	X	X
5	●	●
6	●	●
7	●	●

X = Missing Pin

	From	To
Interleave	J2-A4	J2-B4
* Interlock	—	—

* For BBNCC applications

High Core Blind

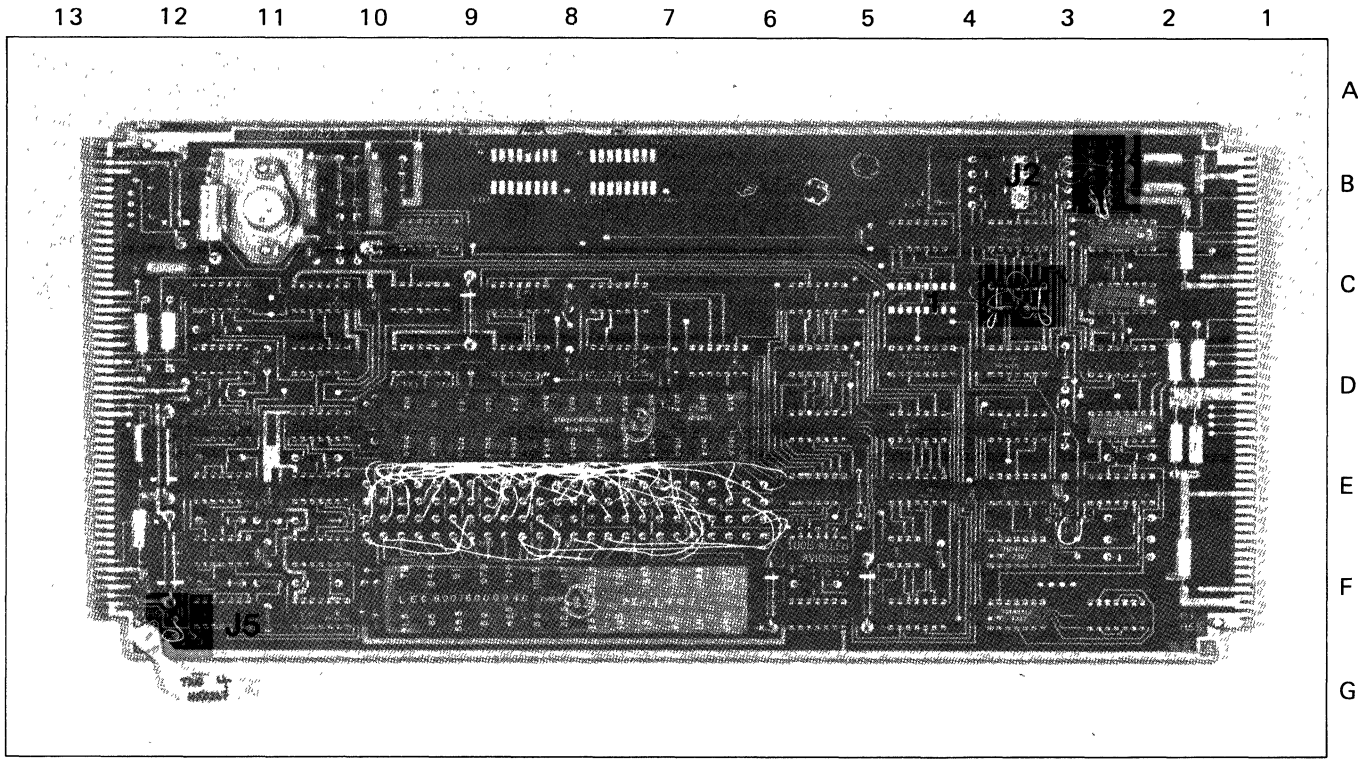
To Make 30K to 32K Blind Connect:

J2-A5 to J2-B5
J2-A6 to J2-B6
J2-A1 to J2-B1

Typical Example: TAG [1] 8P [1] SA = 0 [1] Key = 1 [1] Adrec = 14-17

TAG

2102304G01



J5 Memory Core Type

1 ●
2 ● X 3

X = Missing

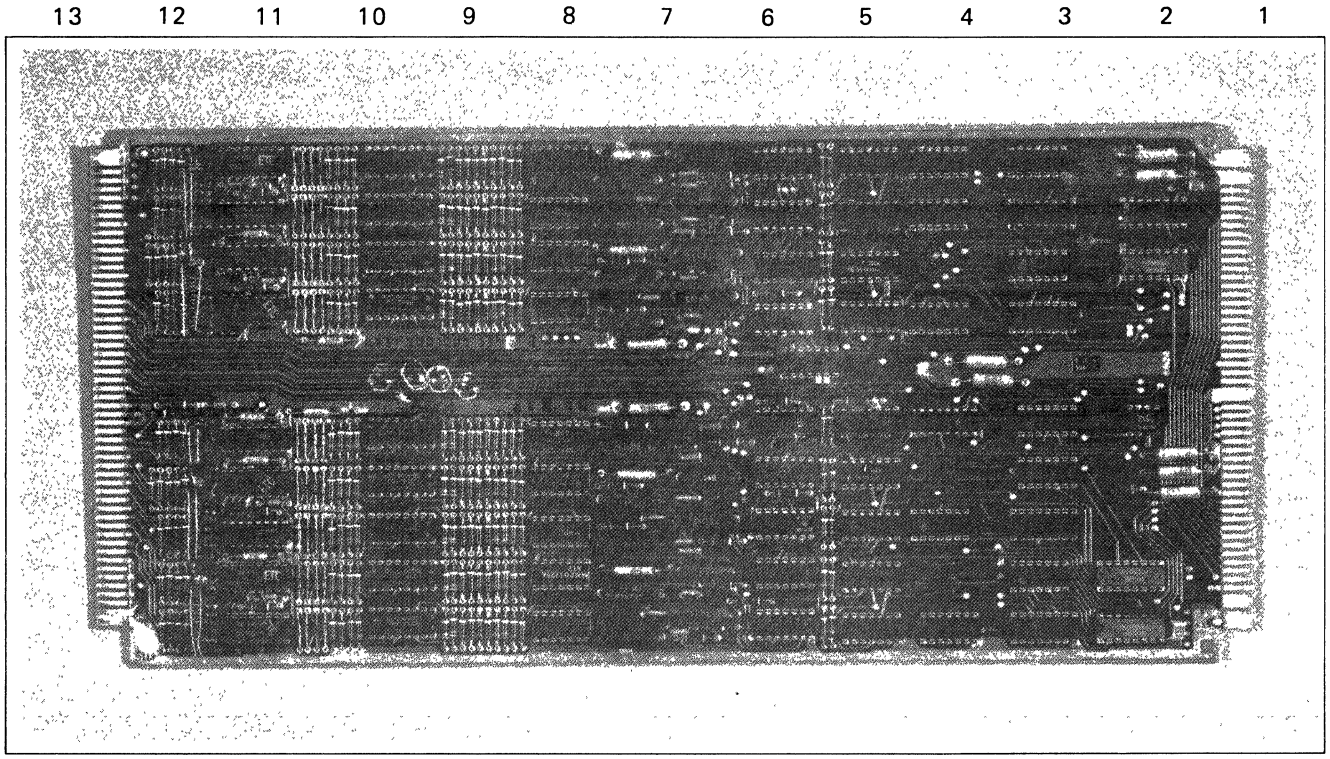
Card Type	Connect	
	From	To
EXY	J5-1	J5-2
MXY	J5-2	J5-3

Note: MXY is 4K only;
EXY is 4K or 8K

Typical Example: TAG [1] 8P [1] SA = 0 [1] Key = 1 [1] Adrec = 14-17

SID

2102302G01



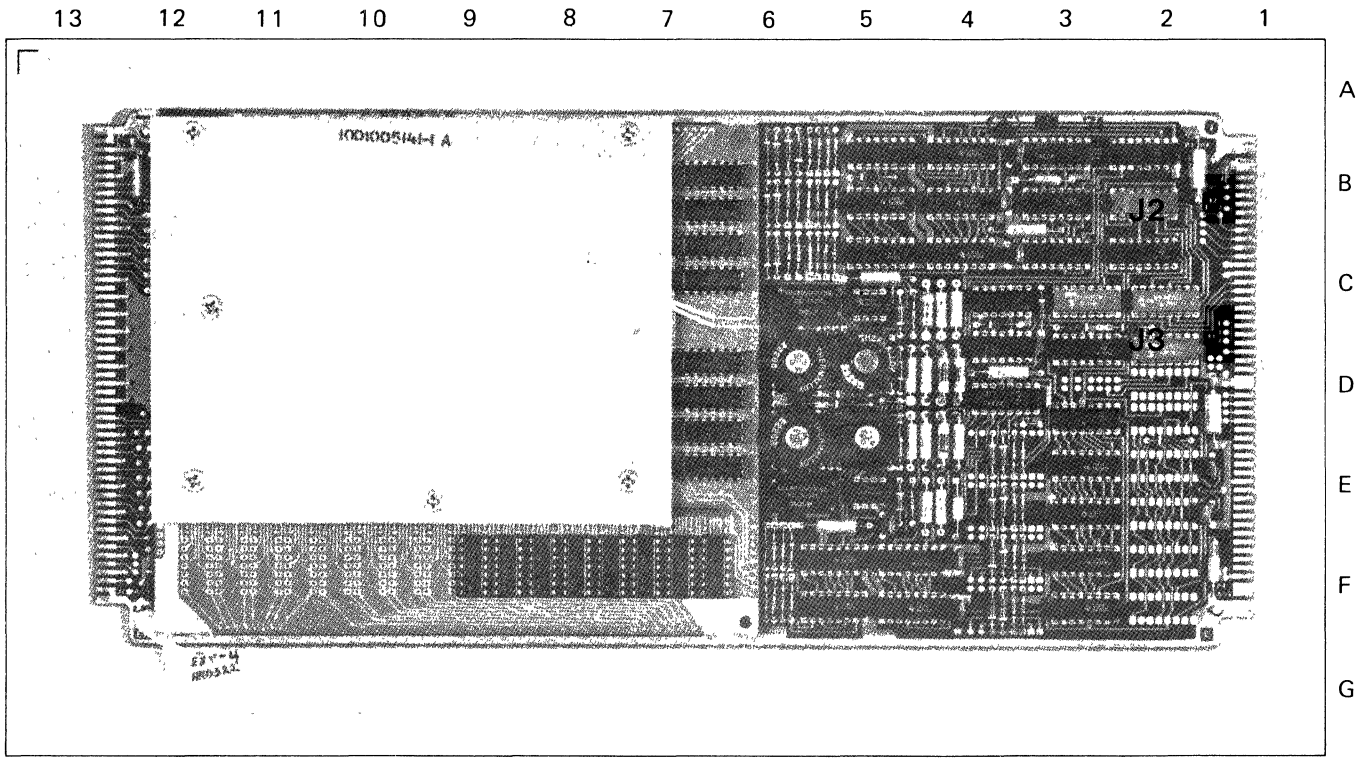
No jumpers are required

on the SID board

for BBNCC applications

EXY

2102299G01



J2, J3 Mode

J2 ● ●
1 2

J3 ● 1
● 2
● 3

		From	To
	Interleave	Remove J2-1	J2-2
		Add J3-1	J3-2
*	Interlock	Add J2-1	J2-2
		Remove J3-1	J3-2

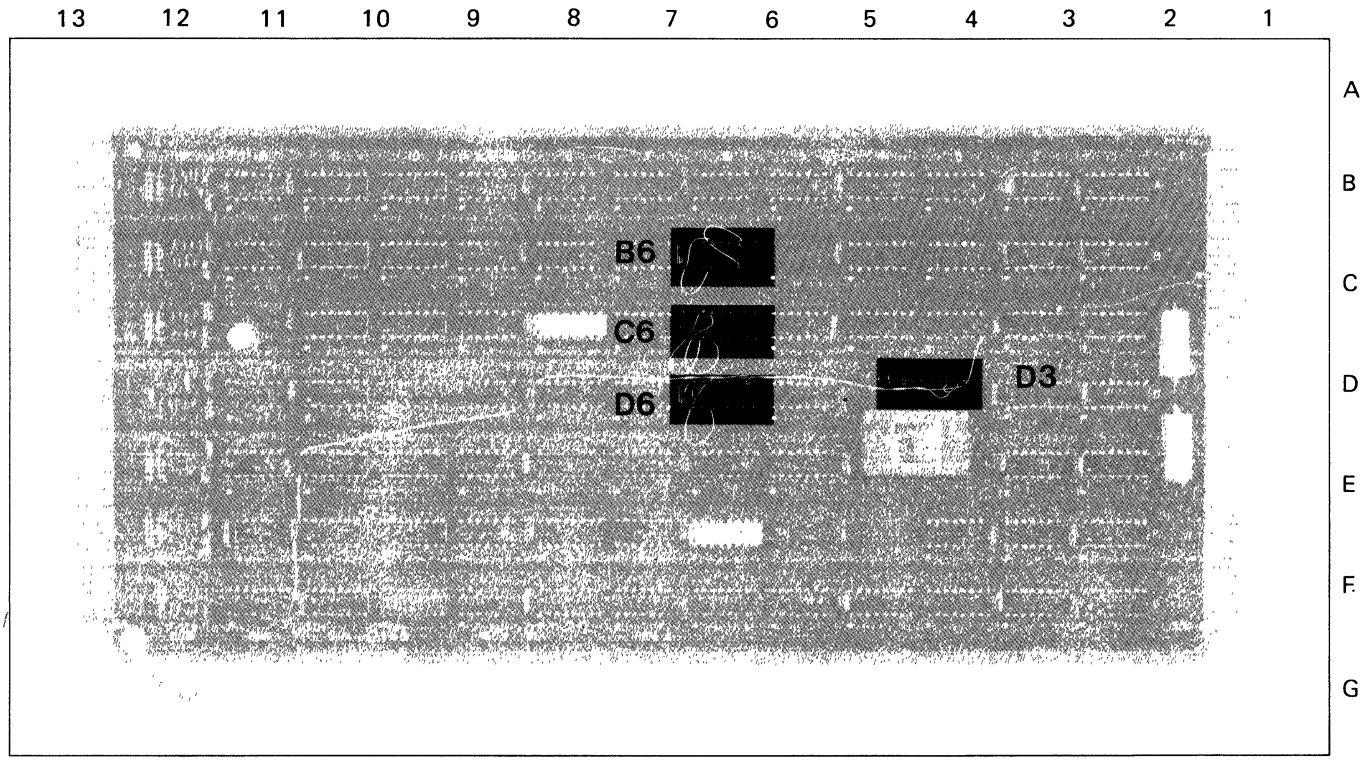
* All applications for BBNCC unless otherwise specified by SRN

Note: 8P in example denotes 8K memory (all sockets populated)

Typical Example: EXY $^{J2, J3}$ 8P

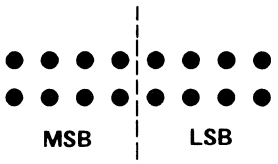
BCM

2101861G01

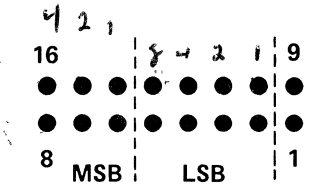


D6 Relevance

chk



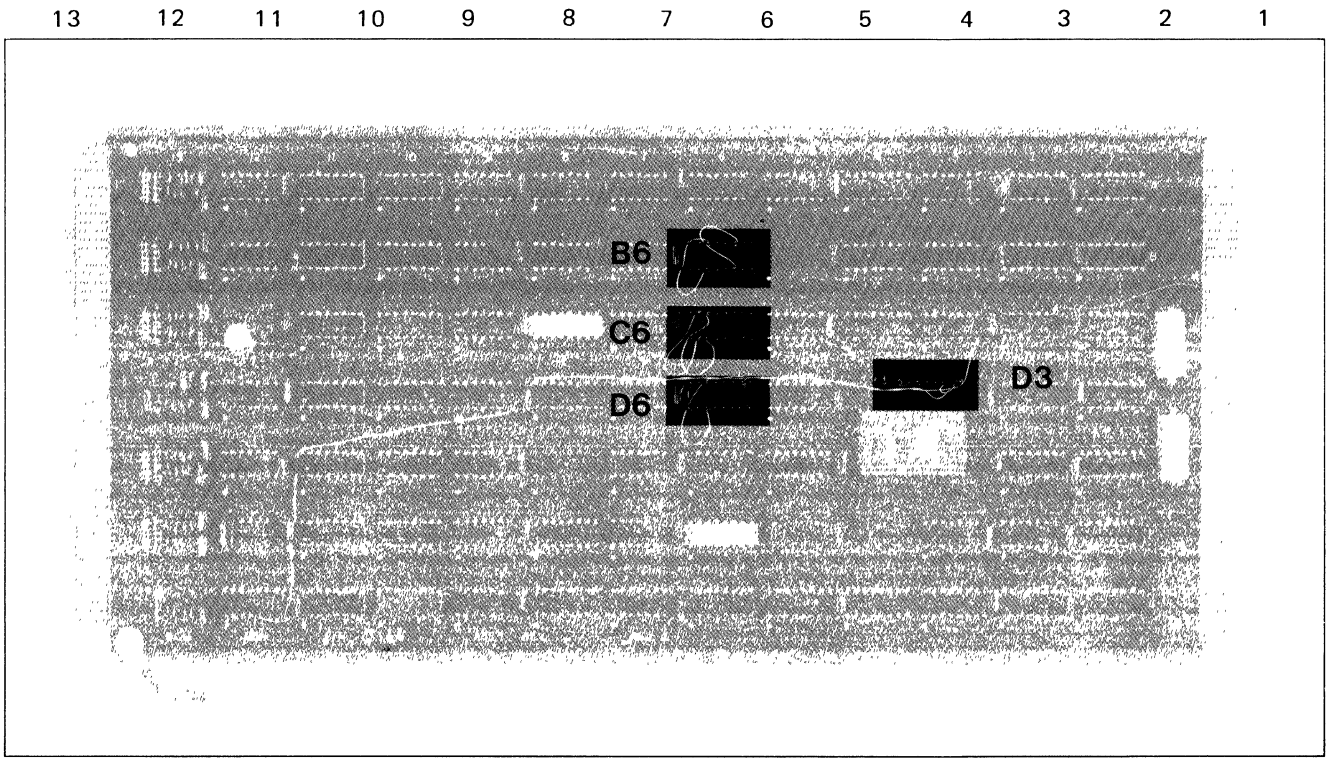
B6 Processor Number



Typical Example: BCM from PR # 12 00000-5FFFF & FF000 - FFFFF ^{C6}rec = 02, ^{D6}chk = C2
^{B6}Memsw = off ^{D3}Done = delayed ^{D3}PAR = Feedback ^{B6}NUM = 12

BCM

2101861G01



D3 Parity

when these are in the BCI must be checked to see that it is the same (CLASSICAL)

16 15 14 13 12 11 10 9

Parity	Connect	
	From	To
WRITE SOURCE GENERATE (Classical)	D3 - 15 D3 - 12	D3 - 16 D3 - 13
WRITE SOURCE	D3 - 14	D3 - 16

Note: PC version of this card has the above jumper rigid at G6 vs the above D3 location

WE USE FEEDBACK

D3 Done Pulse

16 15 14 13 12 11 10 9

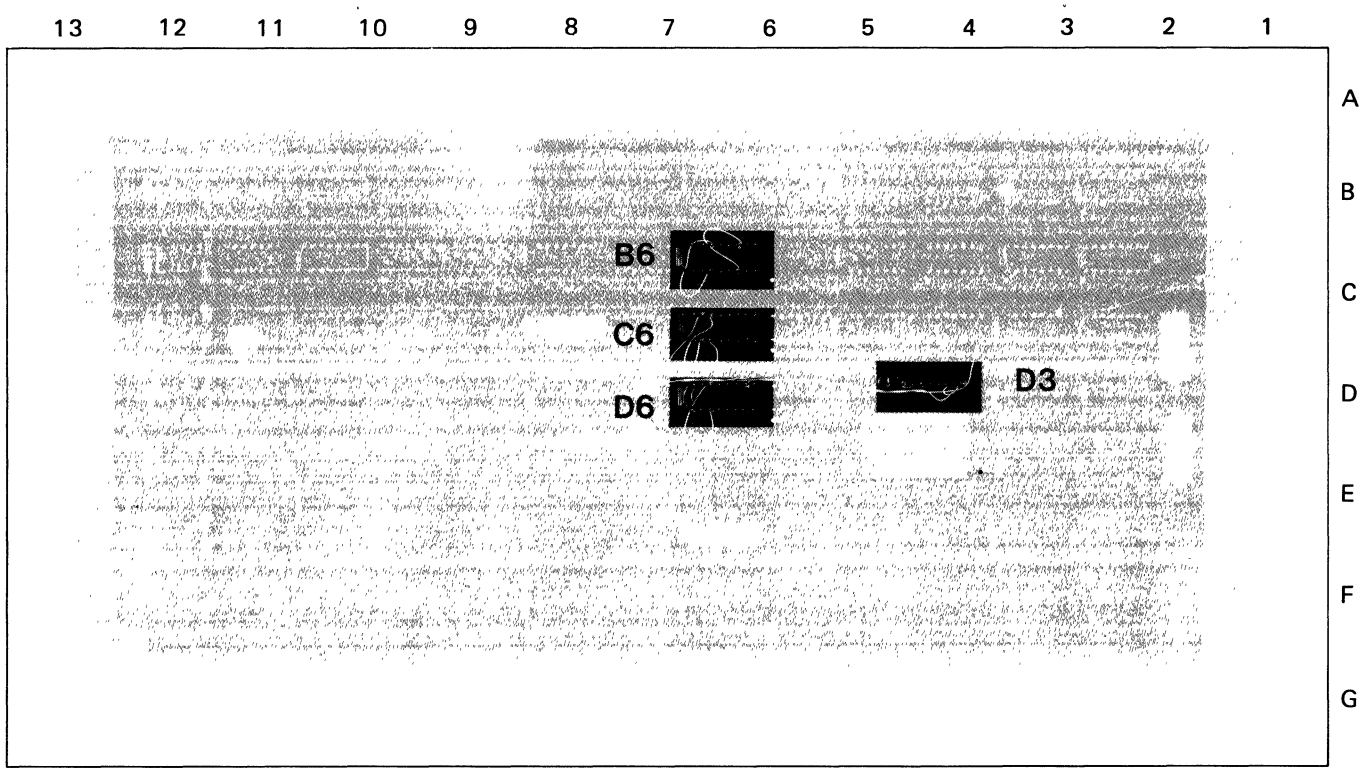
Done Pulse	Connect	
	From	To
Normal	D3 - 11	D3 - 10
Delayed*	D3 - 9	D3 - 10

* With the PAR card

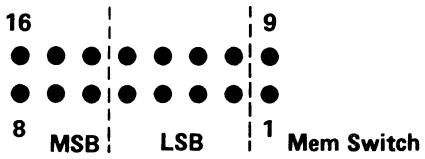
Typical Example: BCM from PR #12 00000-5FFFF & FF000 - FFFFF ^{C6}rec = 02, ^{D6}chk = C2 ^{B6}Memsw = off ^{D3}Done = delayed ^{D3}PAR = Feedback ^{B6}NUM = 12

BCM

2101861G01



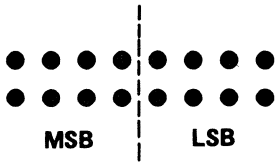
B6 Memswitch



	Jumper
Memory Bus	1–9
Dual M/I Bus on F-Bus	—
Dual M/I Bus on E-Bus (F-Bus BCI on Other End)	—

C6 Base

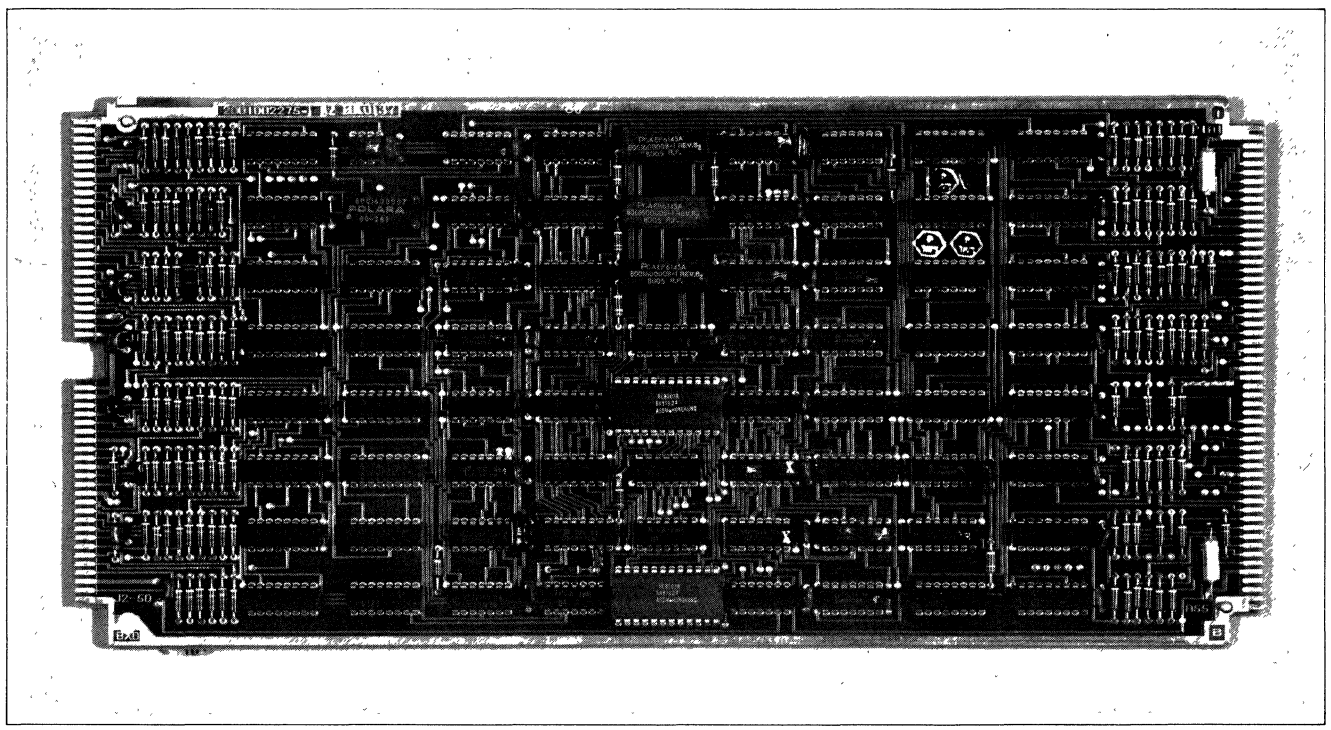
REC



Typical Example: BCM from PR #12 00000-5FFFF & FF000 - FFFFF ^{C6}rec = 02, ^{D6}chk = C2 ^{B6}Memsw = off ^{D3}Done = delayed ^{D3}PAR = Feedback ^{B6}NUM = 12

BXD

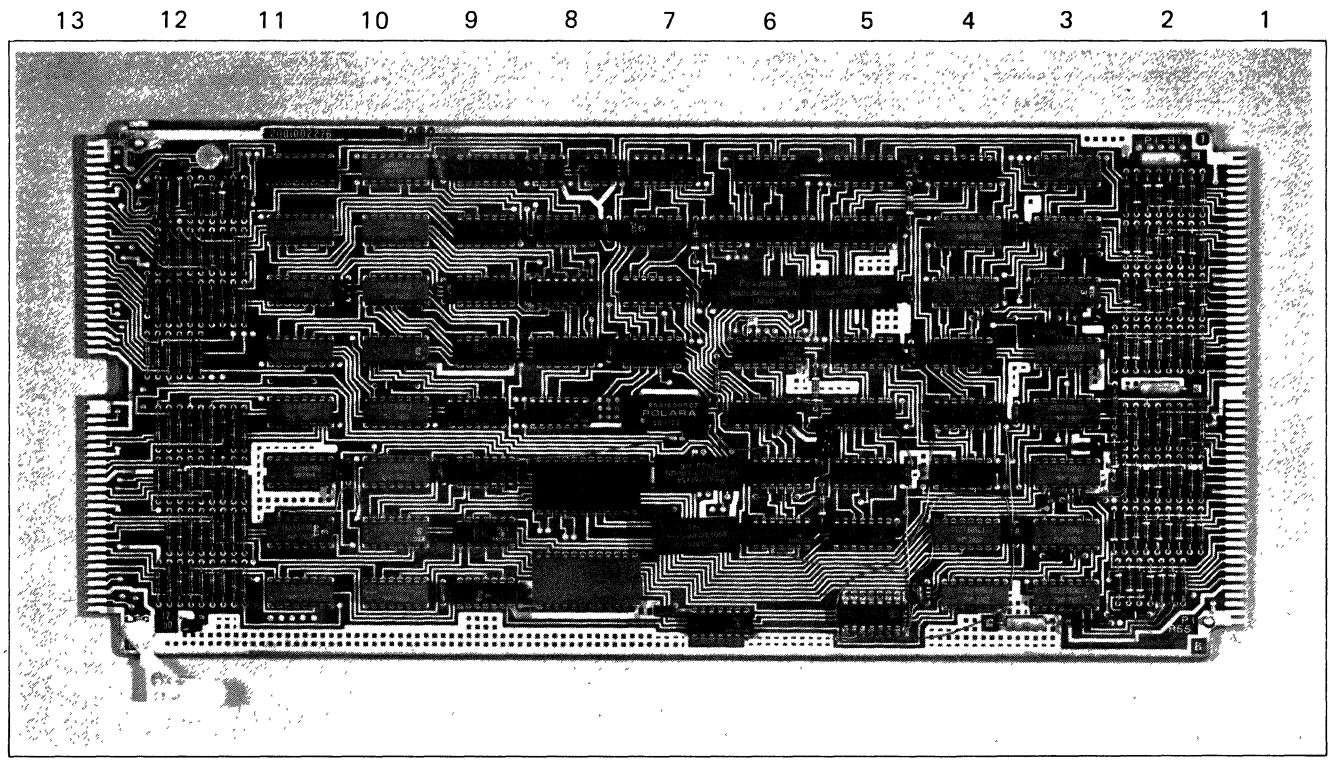
2102295G01



No jumpers are required
on the BXD board
for BBNCC applications

BXR

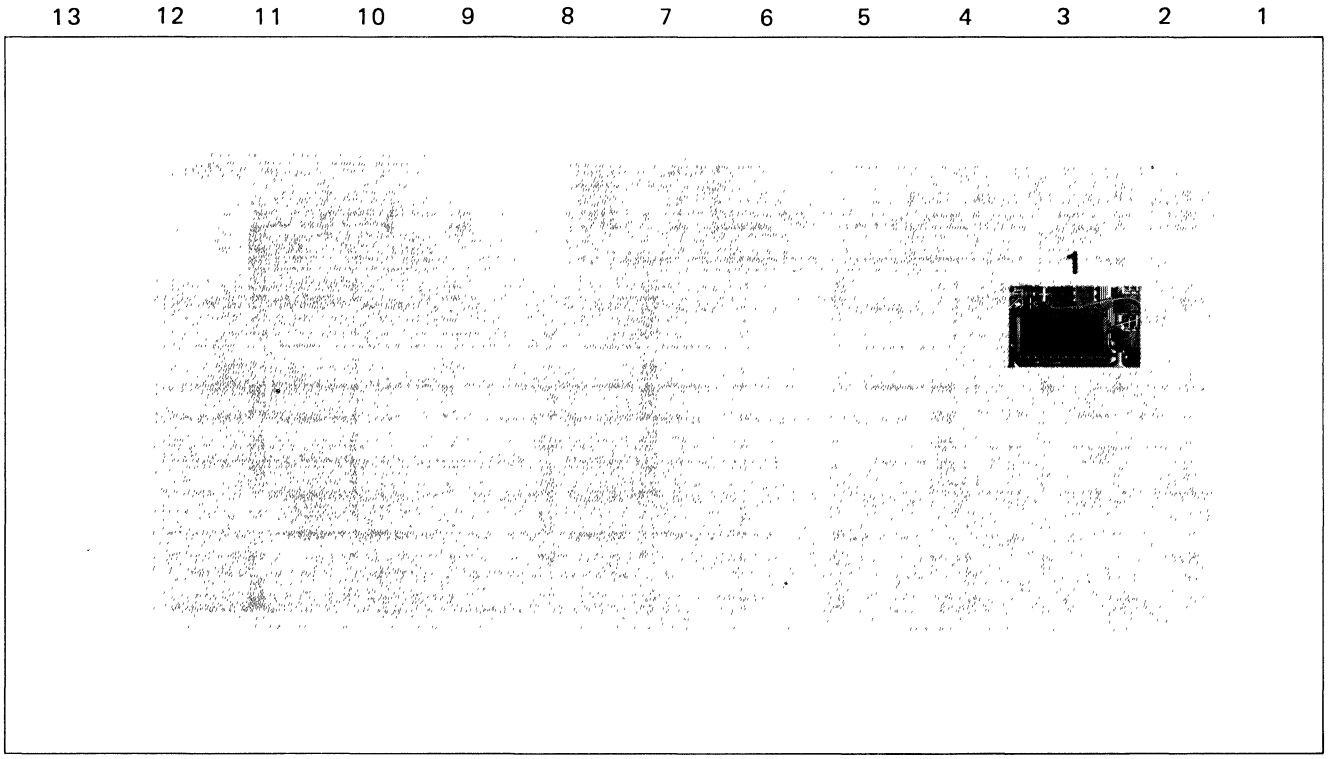
2102296G01



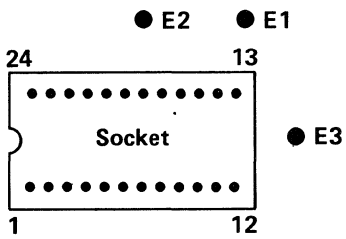
No jumpers are required
on the BXR board
for BBNCC applications

SME

2102550G01



1 Address Recognition - Local Memory



(Split 8K Local)

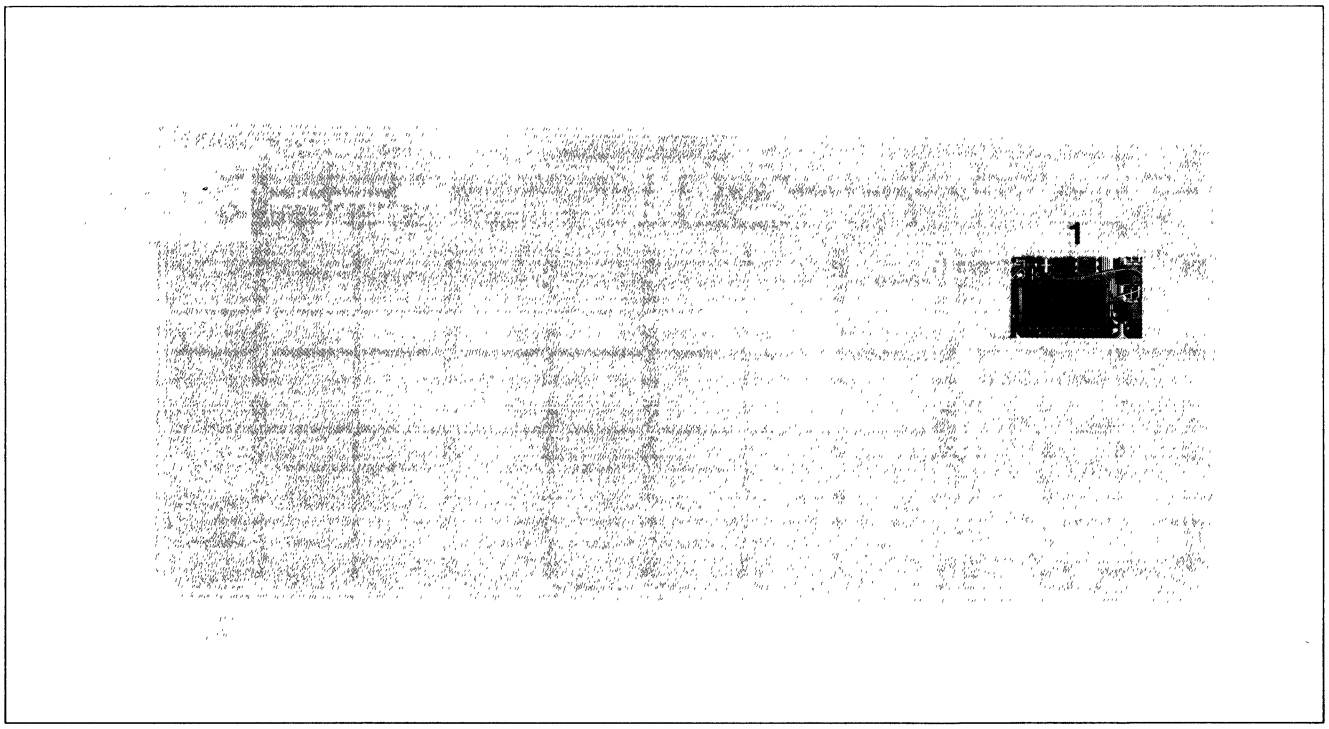
Jumper pins	3 - 20 - 19 - 18
	4 - 17
	6 - 13
	11 - 22
	24 - 21
	E1 - E3

Address Recognition is set by strapping the header and E-1 - E-3 as local or common memory per the shown charts.

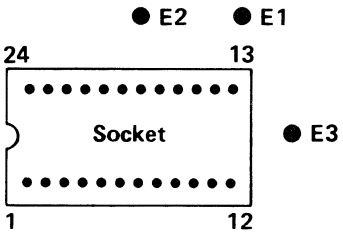
Typical Example: SME SA = 16K Key = 0 KA = 0 Split = N Mem. Blind = N

SME

2102550G01



1 Address Recognition - Common Memory



Add Range	Key Alt	Key	1st Page	Jumpers on Header					On BD
				13-22	11-21	12-19	6-24	4-17	E1-E3
0-16	0	0	0000					4-17	
16-32	0	0	0800					4-7	
32-48	0	1	1000					4-16	
48-64	0	1	1800					4-8	
64-80	0	2	2000					4-15	
80-96	0	2	2800					4-9	
96-112	0	3	3000					4-14	
112-128	0	3	3800					4-10	
128-144	1	0	4000			5-19		4-17	
144-160	1	0	4800					4-7	
160-176	1	1	5000					4-16	
176-192	1	1	5800					4-8	
192-208	1	2	6000					4-15	
208-224	1	2	6800					4-9	
224-240	1	3	7000					4-14	
240-256	1	3	7800					4-10	
256-272	2	0	8000			12-19		4-17	E1-E2
272 288	2	0	8800					4-7	
288-304	2	1	9000					4-17	
304-320	2	1	9800					4-8	
320-336	2	2	A000					4-15	
336-352	2	2	A800					4-9	
352 368	2	3	B000					4-14	
368-384	2	3	B800					4-10	
384-400	3	0	C000			5-19		4-17	
400-416	3	0	C800					4-7	
416 432	3	1	D000					4 16	
432 448	3	1	D800					4-8	
448 464	3	2	E000					4-15	
464-480	3	2	E800					4-9	
480 496	3	3	F000					4-14	
496 512	3	3	F800					4 10	

F Bus

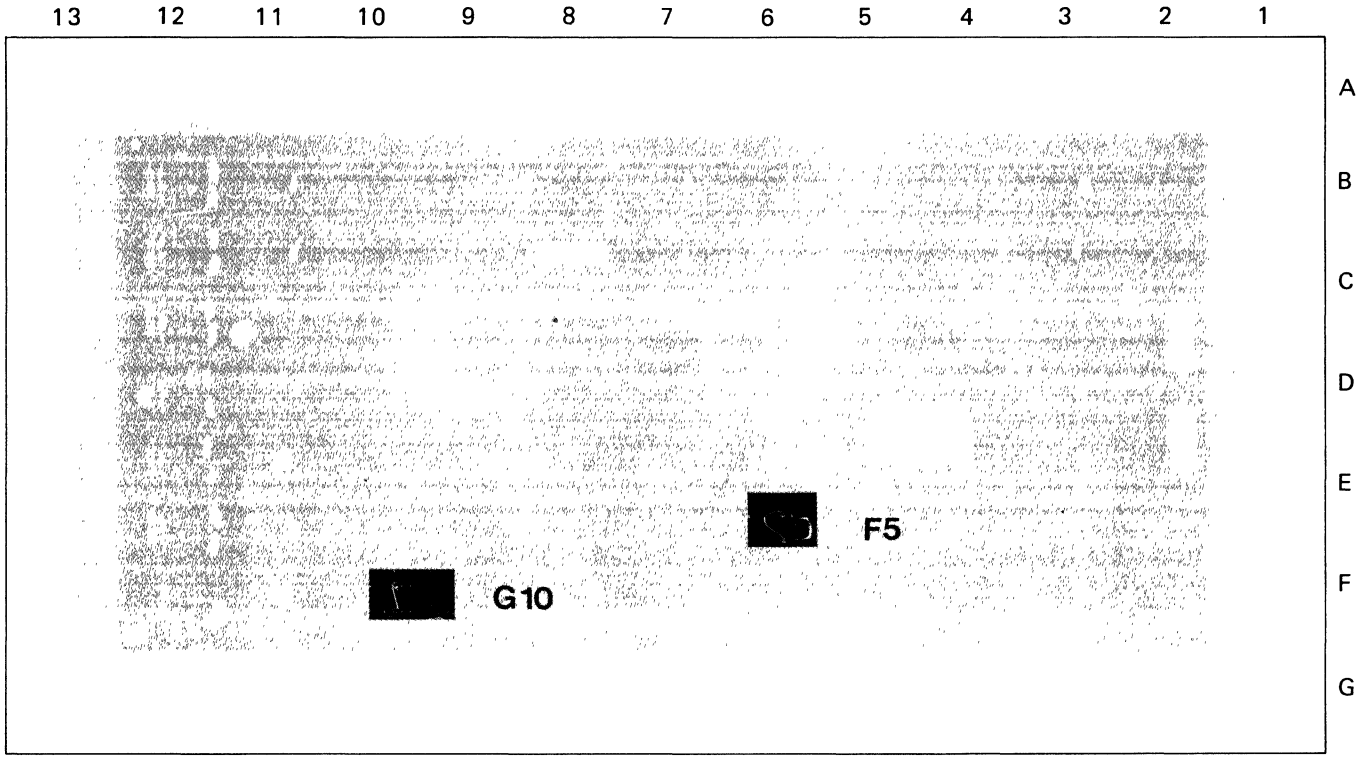
6 Bus

Address Recognition is set by strapping the header and E-1 - E-3 as local or common memory per the shown charts.

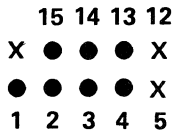
Typical Example: SME SA = 16K Key = 0 KA = 0 Split = N Mem. Blind = N

BCP

2100528G01

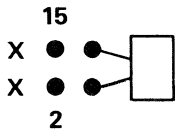


G10 Parity



Parity	Connect	
	From	To
Write Source Generate (Classical)	F5-2 G10-1 G10-3	F5-15 G10-15 G10-14
Write Source Check (Feedback)	G10-2 F5-2	G10-15 F5-15
None	G10-4	G10-13

F5 Parity

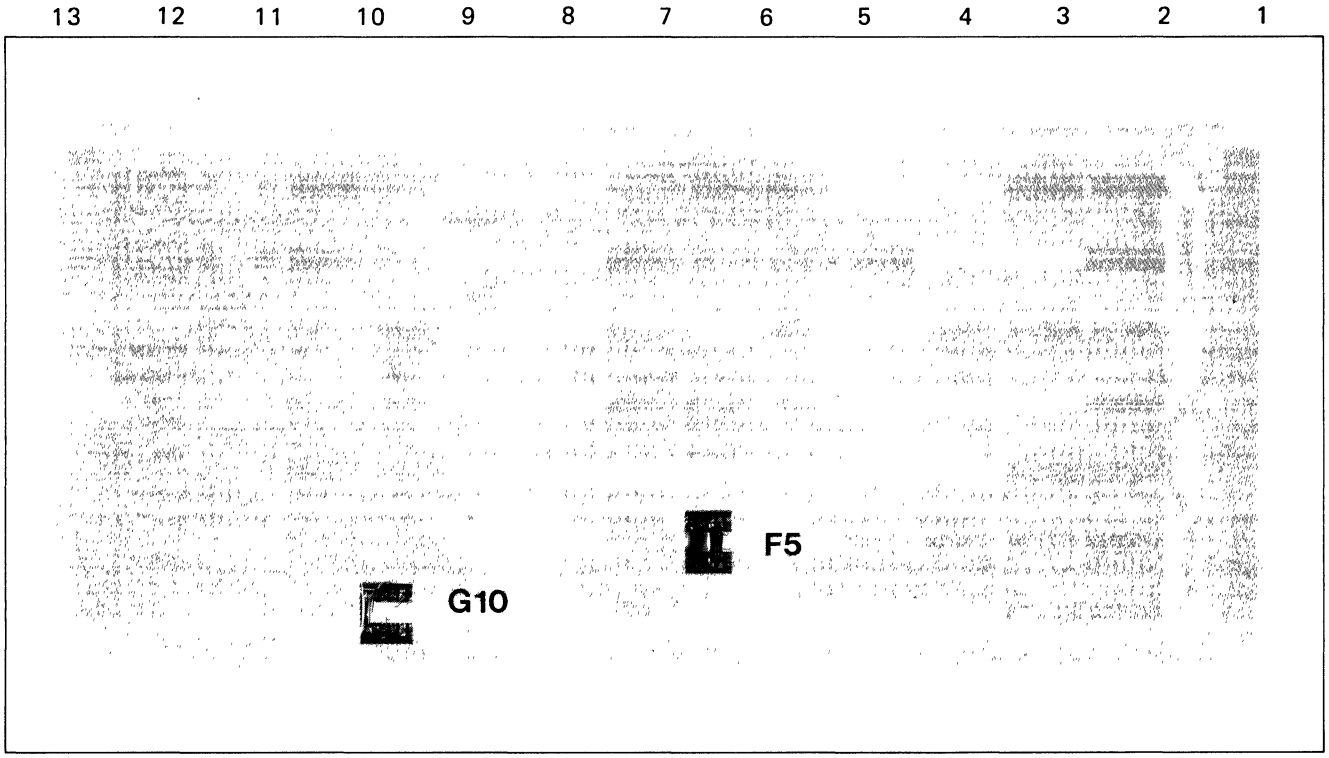


Parity	Connect	
	From	To
Write Source Generate (Classical)	F5-2 G10-1 G10-3	F5-15 G10-15 G10-14
Write Source Check (Feedback)	G10-2 F5-2	G10-15 F5-15
None	G10-4	G10-13

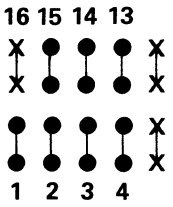
Typical Example: BCP to IO #E PAR = Feedback

BCI

2100168G01

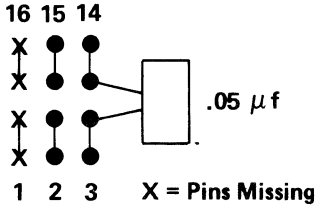


G10 Parity



Parity	Connect	
	From	To
Write Source Generate (Classical)	F5-2 G10-1 G10-3	F5-15 G10-15 G10-14
Write Source Check (Feedback)	G10-2 F5-2	G10-15 F5-15
None	G10-4	G10-13

F5 Parity

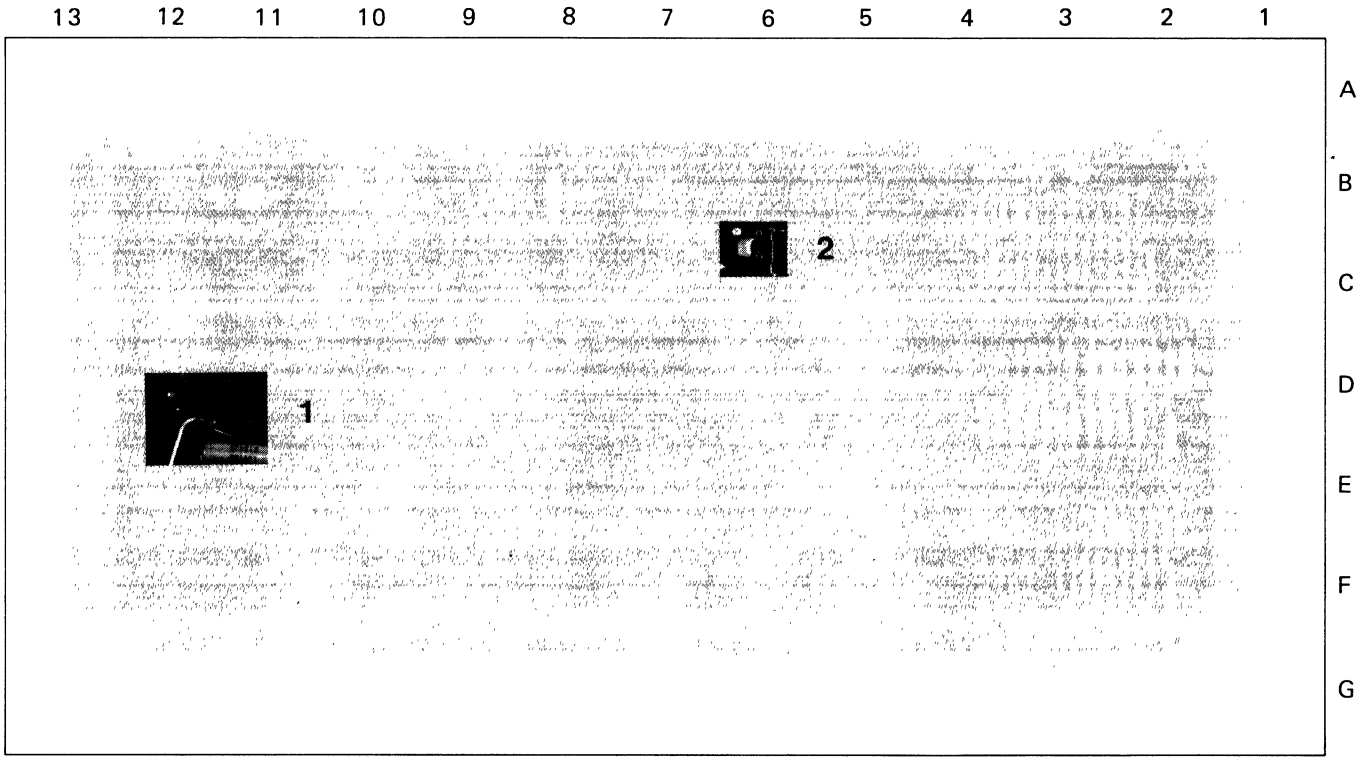


Parity	Connect	
	From	To
Write Source Generate (Classical)	F5-2 G10-1 G10-3	F5-15 G10-15 G10-14
Write Source Check (Feedback)	G10-2 F5-2	G10-15 F5-15
None	G10-4	G10-13

Typical Example: BCI to MEM# 40 G10, F5 PAR = Classical

BCU

2102293G01



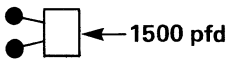
1 Enable/Disable Switch

Position	Result
Down	Auto Bus Reset
Up	No Auto Bus Reset

Note: Diagnostics Are to Be Run With the BCU Switch in Disable (Up) Position

---

2 Quit Delay



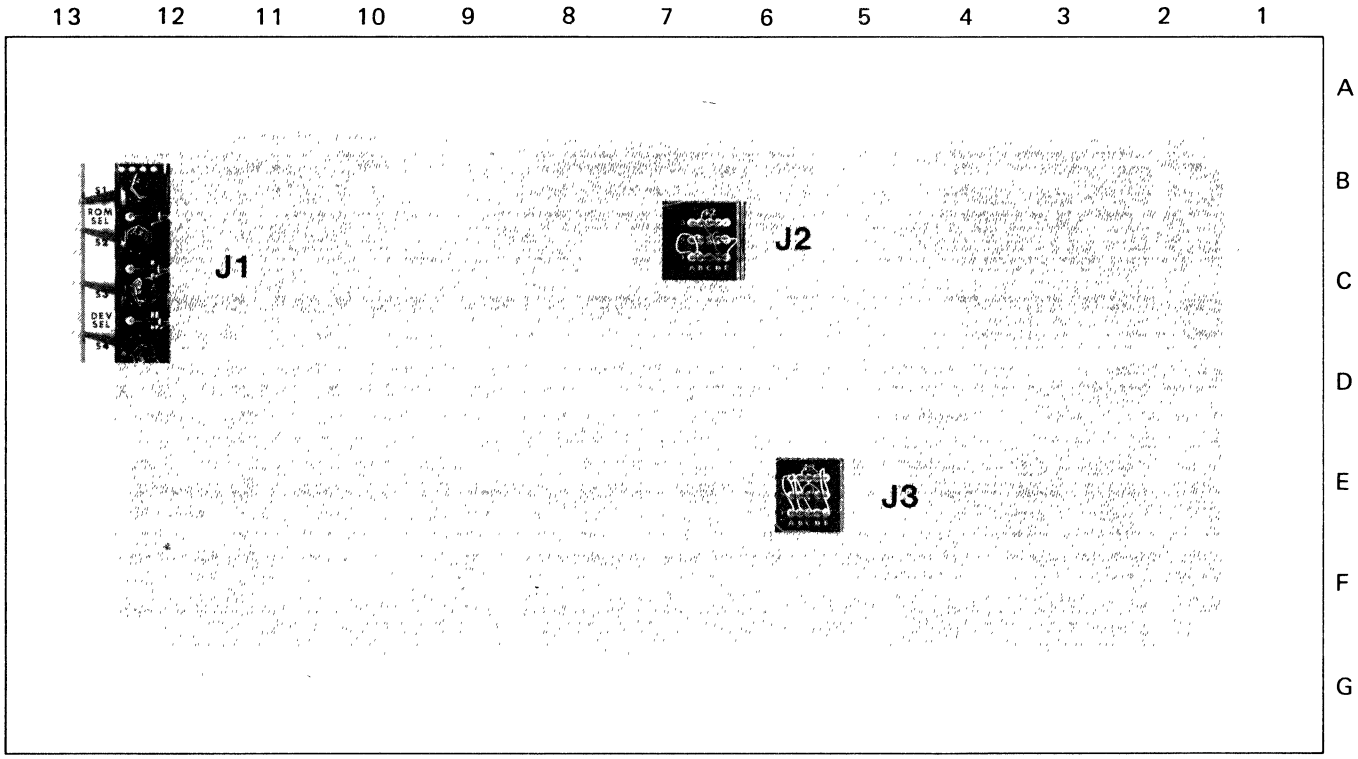
PSAT BOTH E + F BUS 30-70 NSEC

Bus	Added Capacitor	Quit Timing
Memory		3-7 μsec
VDA		3-7 μsec
PLI		3-7 μsec
E(M/I)		3-7 μsec
Processor	.1 μfd (104)	300-800 μsec
F(M/I)	.01 μfd (103)	30-70 μsec

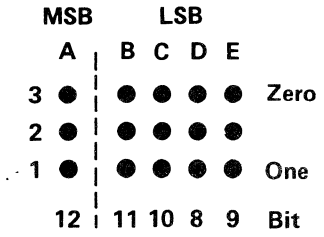
Typical Example: BCU Bus = ^{2}P

ALD

2102294G01



J3 Program Start Address



Address Bit	FEØØ	FAØØ	EAØØ
8	O	O	O
9	X	X	X
10	X	O	O
11	X	X	X
12	X	X	O

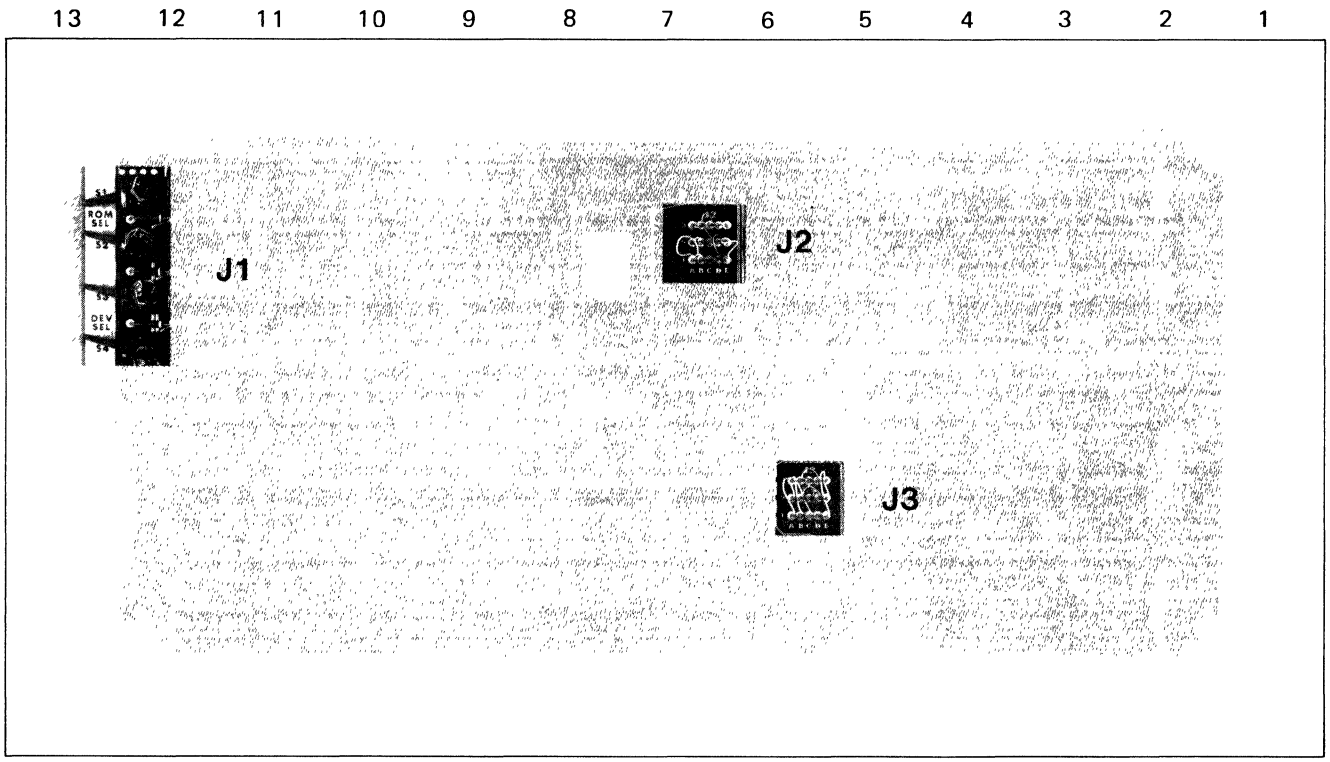
X = On O = Off

Typical Example: ALD $^{J2, J3}$FEØØ Device = FC30 PROM = 2

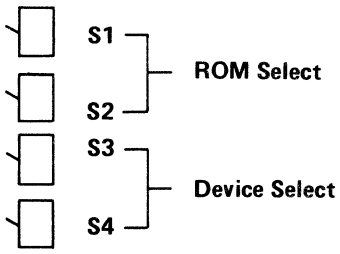
^{J1}Switches (Top-Bot) On Up Up Up

ALD

2102294G01



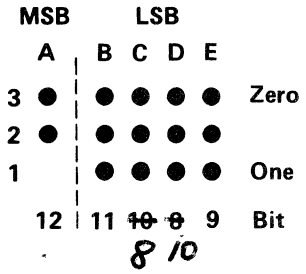
J1 Switches



	S1	S2	S3	S4
Papertape	U	U	U	U
Cassette	D	U	U	U
PSAT, F-Bus	D	U	U	D
PSAT, E-Bus	D	U	D	D

U = Up D = Down

J2 ALD Address



Address Bit	FEØØ	FAØØ	EAØØ
8	O	O	O
9	X	X	X
10	X	O	O
11	X	X	X
12	X	X	O

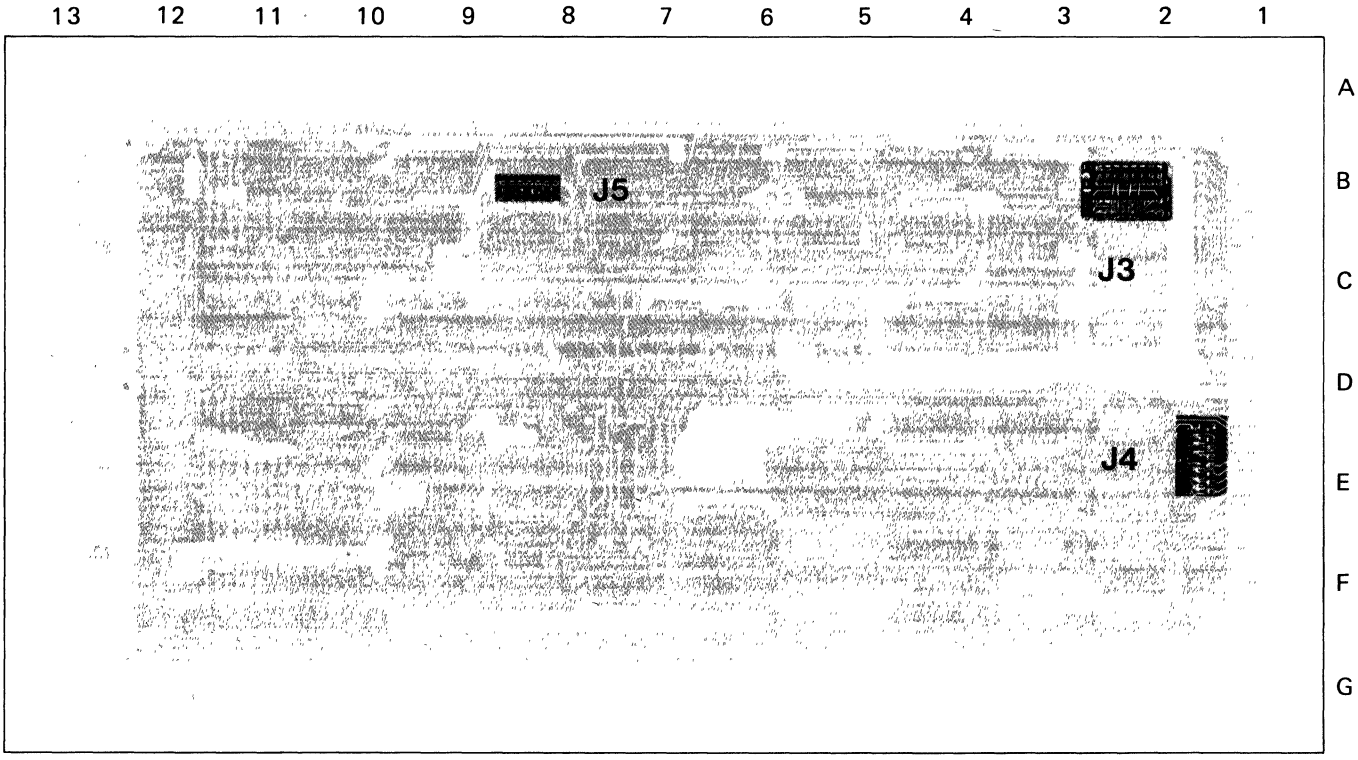
X = On O = Off

Typical Example: ALD $^{J2, J3}$FEØØ Device = FC30 PROM = 2

^{J1}Switches (Top-Bot) Dn Up Up Up

PPB

2102297G01



J3 Module Address

	MSB				LSB				
	A	B	C	D	E	F	G	H	
3	●	●	●	●	●	●	●	●	Zero
2	●	●	●	●	●	●	●	●	
1	●	●	●	●	●	●	●	●	One
	11	10	9	8	7	6	5	4	Bit

Address Bit	FC30
4	X
5	X
6	
7	
8	
9	
10	X
11	X

X = On

J4 Bus Service Level

	1	2
A	●	●
B	●	●
C	●	●
D	●	●
E	●	●
F	●	●
G	●	●
H	●	●

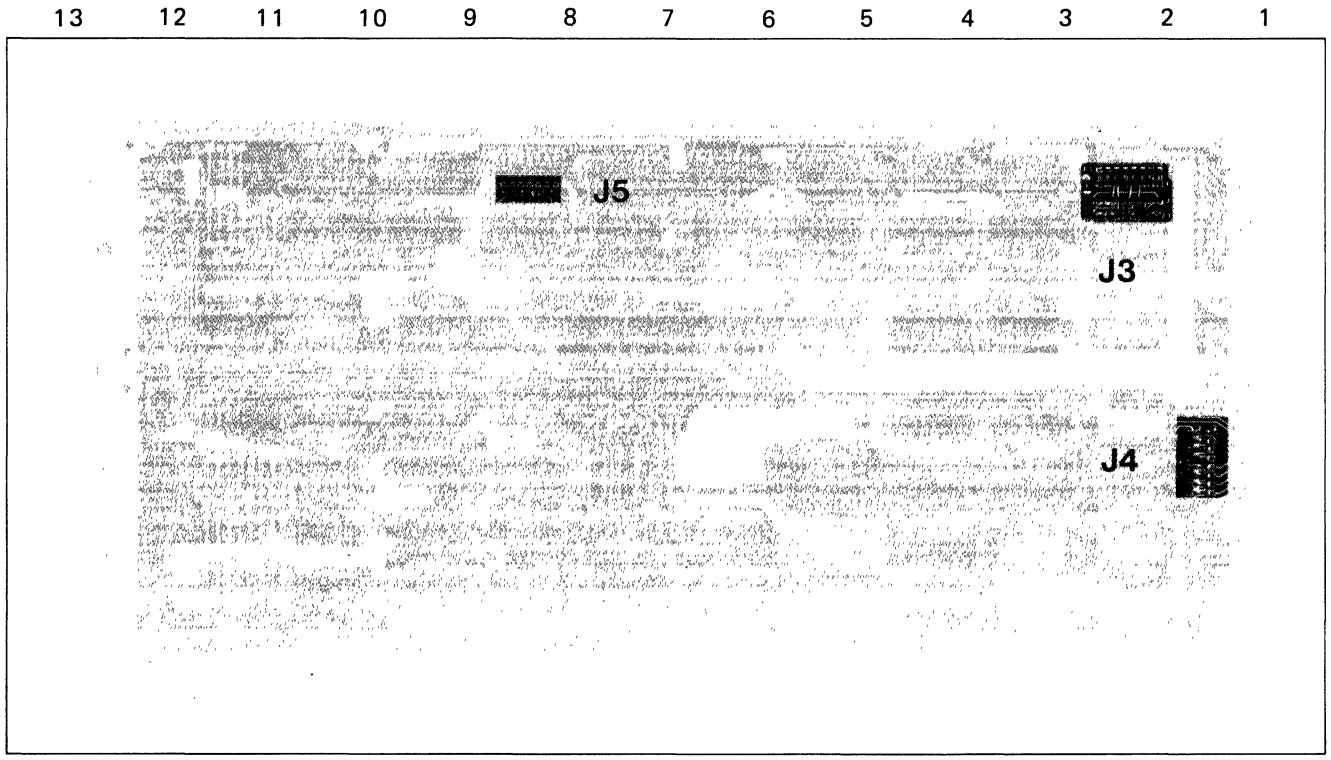
	Connect	
Level 2	1B - 2B	*
	1F - 2F	

* Normal

Typical Example: PPB ^{J3} FC30

PPB

2102297G01



J5 Block Transfer

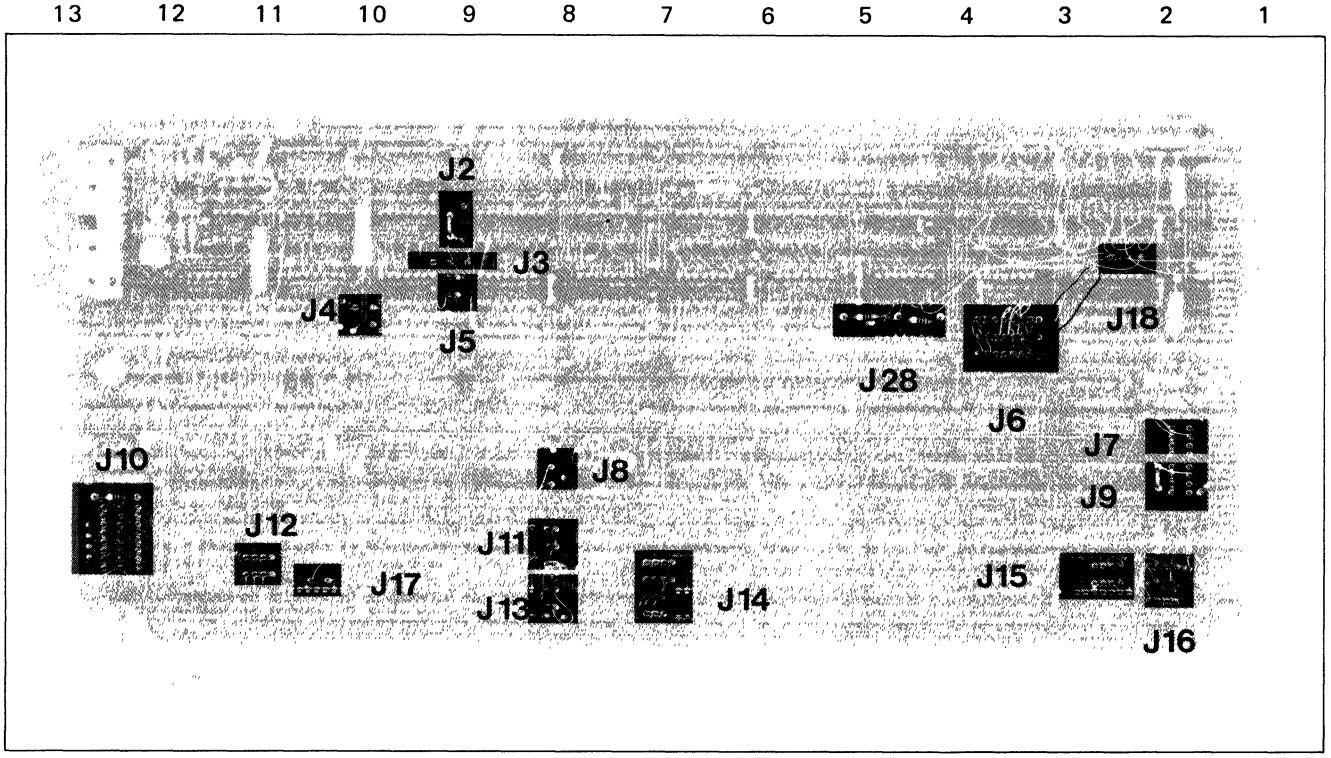
		Jumper
	With BTA	——
*	Without BTA	1 - 2

* Normal

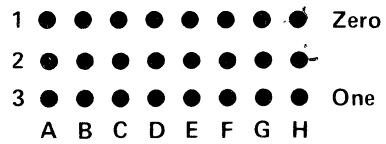
Typical Example: PPB ^{J3} FC30

PSB

2102300G01



J6 Address Recognition



	Address									Result Address
Column	A	B	C	D	E	F	G	H	*	
Bit	4	5	6	7	8	9	10	11	12	
Term I/O F-Bus						X		X	X	FA00
VDA-PLI	X						X	X	X	FC10
Term I/O E-Bus						X		X		EA00
CASP	X	X					X	X	X	FC30

X = One

* Saddlebacked Chips at A2, A3
One = J6, Pin 3H Jumpered to J18, Pin 1
Zero = J6, Pin 1H Jumpered to J18, Pin 1

J18

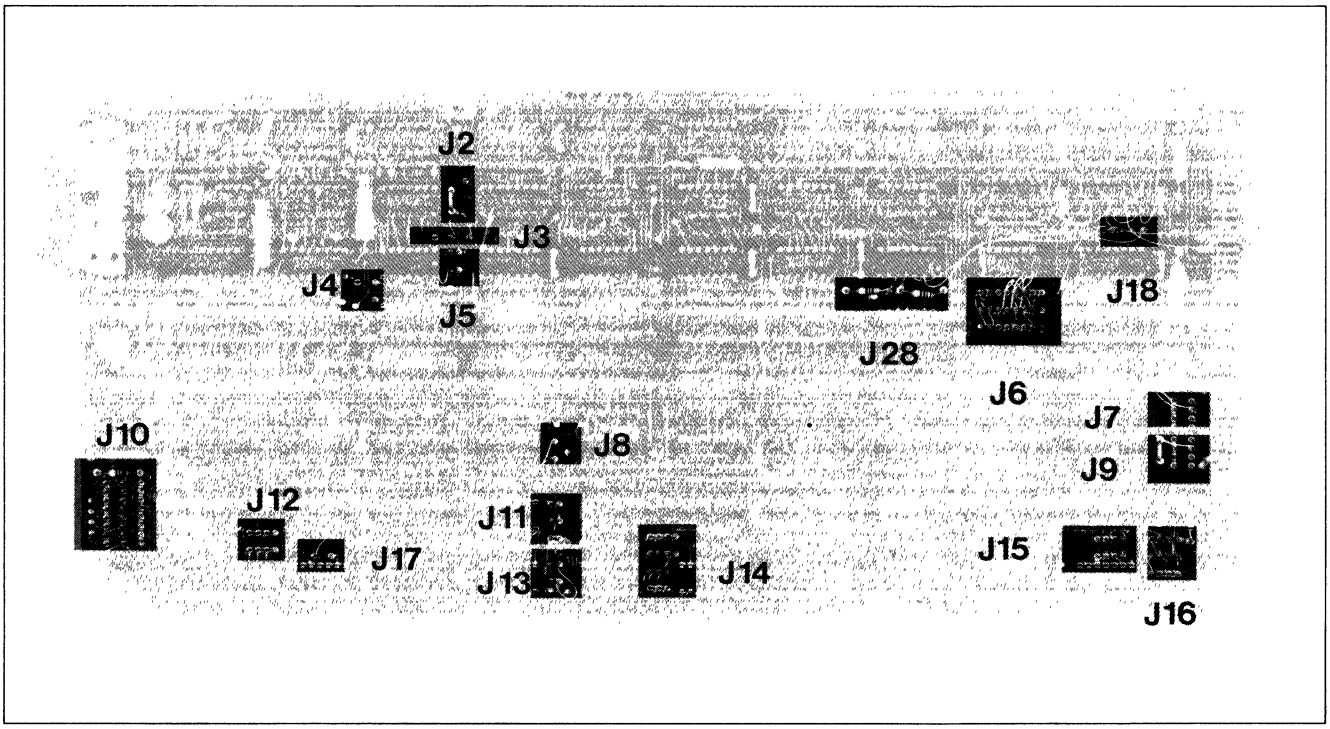
● X X ●
1 2 3 4 X = Missing

Typical Example: PSB ^{J6}FA00 Speed = 9600 intlev = none ^{3}stop = 2 ^{J14}data = 8 ^{J3}rcvdat = RS232 ^{6}Bus = 10 CRT = yes

---

PSB

2102300G01



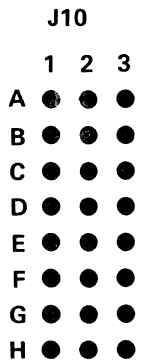
Speed (J10, J12, J15-J17)

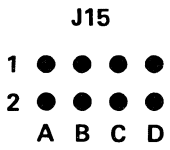
Data Rate		J10								J12				J17	J15				J16			
Baud	From	2A	2B	2C	2D	2E	2F	2G	2H	1A	1B	1C	1D	1	1A	1B	1C	1D	2A	2B	2C	2D
74.2	To	1A	3B	1C	3D	3E	1F	3G	3H	2A	—	—	—	—	—	2B	—	—	1A	3B	3C	1D
110	To	1A	1B	3C	1D	1E	1F	3G	1H	2A	—	—	—	—	—	—	2C	—	3A	3B	3C	1D
300	To	3A	1B	3C	1D	3E	3F	3G	1H	—	2B	—	—	—	—	—	2C	—	1A	3B	1C	—
600	To	3A	1B	3C	1D	3E	3F	3G	1H	—	2B	—	—	—	—	—	—	2D	1A	3B	1C	3D
1200	To	3A	3B	3C	1D	3E	1F	3G	3H	—	—	—	2D	—	—	—	2C	—	1A	3B	1C	1D
1800	To	3A	3B	3C	3D	1E	1F	3G	1H	—	—	—	—	2	—	—	2C	—	3A	1B	1C	1D
2400	To	3A	3B	3C	3D	1E	3F	1G	3H	—	—	—	—	2	—	—	2C	—	1A	1B	1C	3D
4800	To	3A	3B	3C	3D	1E	3F	3G	3H	—	—	—	—	2	—	—	—	2D	1A	1B	1C	3D
9600	To	3A	3B	3C	3D	3E	1F	3G	1H	—	—	—	—	*	—	—	—	2D	1A	1B	1C	1D
19200	To	3A	3B	3C	3D	3E	3F	1G	3H	—	—	—	—	**	—	—	—	2D	3A	1B	1C	1D

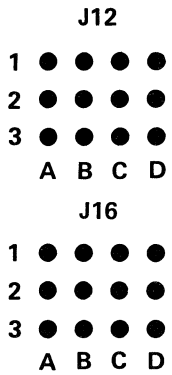
*J17-1 to U73 @ H8 12

**J17-1 to U73 @ H8 13

Normal Settings	
TTY	110
Terminal	9600
CASP	19200



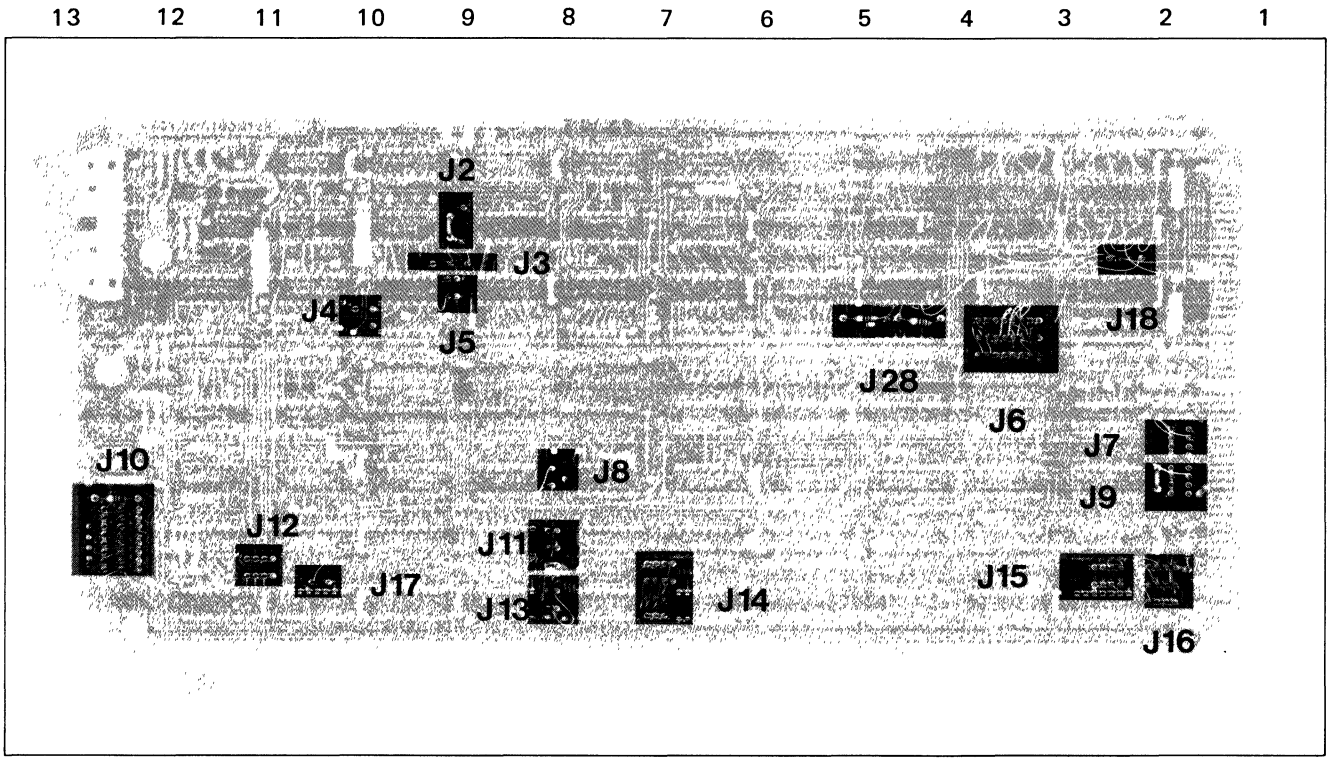




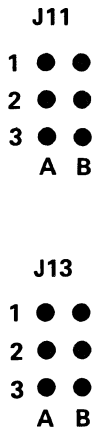
Typical Example: PSB ^{J6}FA00 Speed = 9600 intlev = none ^{3}stop = 2 ^{J14}data = 8 ^{J3}rcvdat = RS232 ^{6}Bus = 10 CRT = yes

PSB

2102300G01



3 Stop Bits



Number of Stop Bits

Number of Stop Bits		Terminal Connect From	Terminal Connect To	CASP Connect From	CASP Connect To
One		J13-2A	J13-1A		
		J13-2B	J13-3B		
		J11-2A	J11-3A		
		J11-2B	J11-1B		
Two	*	J13-2A	J13-1A	J13-1A	J13-2A
		J13-2B	J13-1B	J13-2B	J13-3B
		J11-2A	J11-1A	J11-1B	J11-2B
		J11-2B	J11-1B	J11-2A	J11-3A

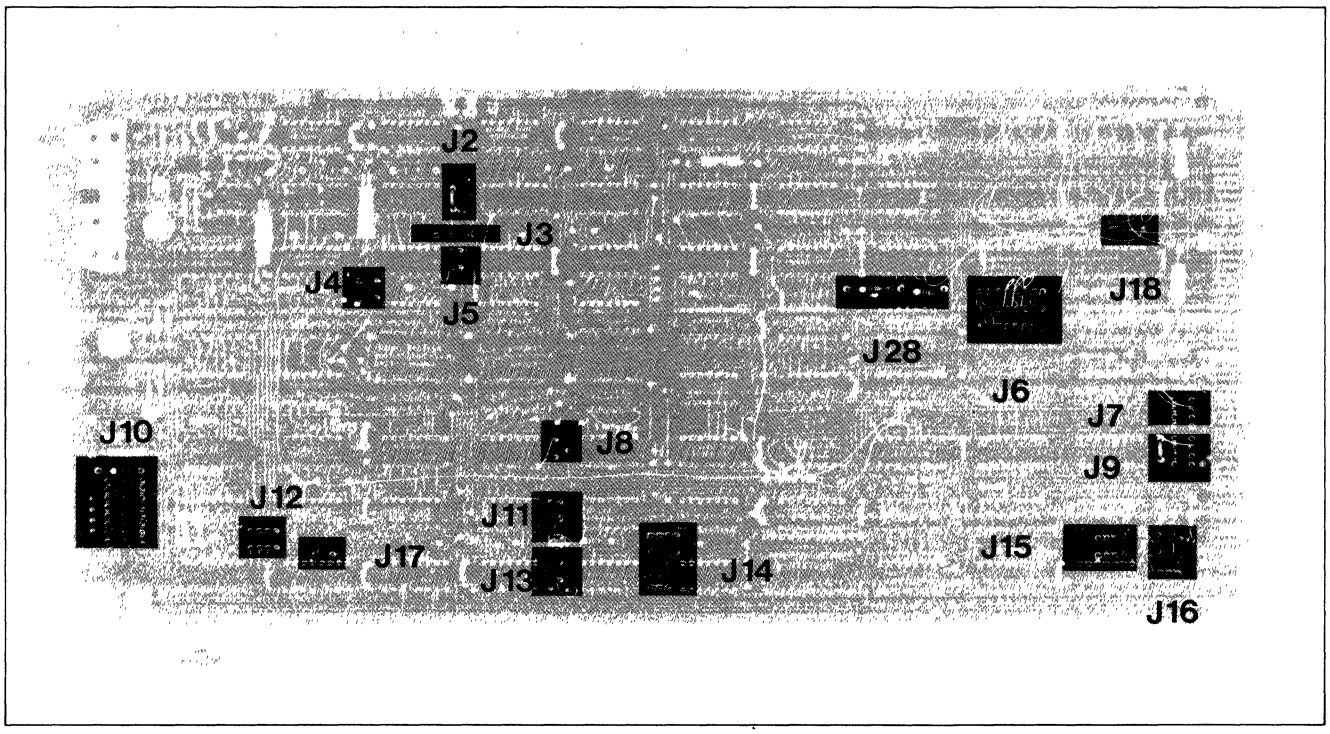
*Normal Setting

Typical Example: PSB ^{J6}FA00 Speed = 9600 intlev = none ^{3}stop = 2 ^{J14}data = 8 ^{J3}rcvdat = RS232 ^{6}Bus = 10 CRT = yes

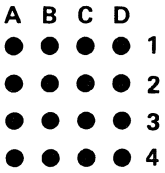
---

PSB

2102300G01



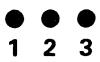
J14 Data Bits/Char.



	DATA BITS	CONNECT FROM	CONNECT TO
	FIVE	J14 - 2A	J14 - 1A
		J14 - 3A	J14 - 4A
		J14 - 3B	J14 - 4B
		J14 - 3C	J14 - 4C
	SIX	J14 - 2A	J14 - 3A
		J14 - 2B	J14 - 1B
		J14 - 3B	J14 - 4B
		J14 - 3C	J14 - 4C
	SEVEN	J14 - 2A	J14 - 3A
		J14 - 2B	J14 - 3B
		J14 - 2C	J14 - 1C
		J14 - 3C	J14 - 4C
*	EIGHT	J14 - 2A	J14 - 3A
		J14 - 2B	J14 - 3B
		J14 - 2C	J14 - 3C
		J14 - 2D	J14 - 1D

* Normal Setting

J3 Receive Data Mode



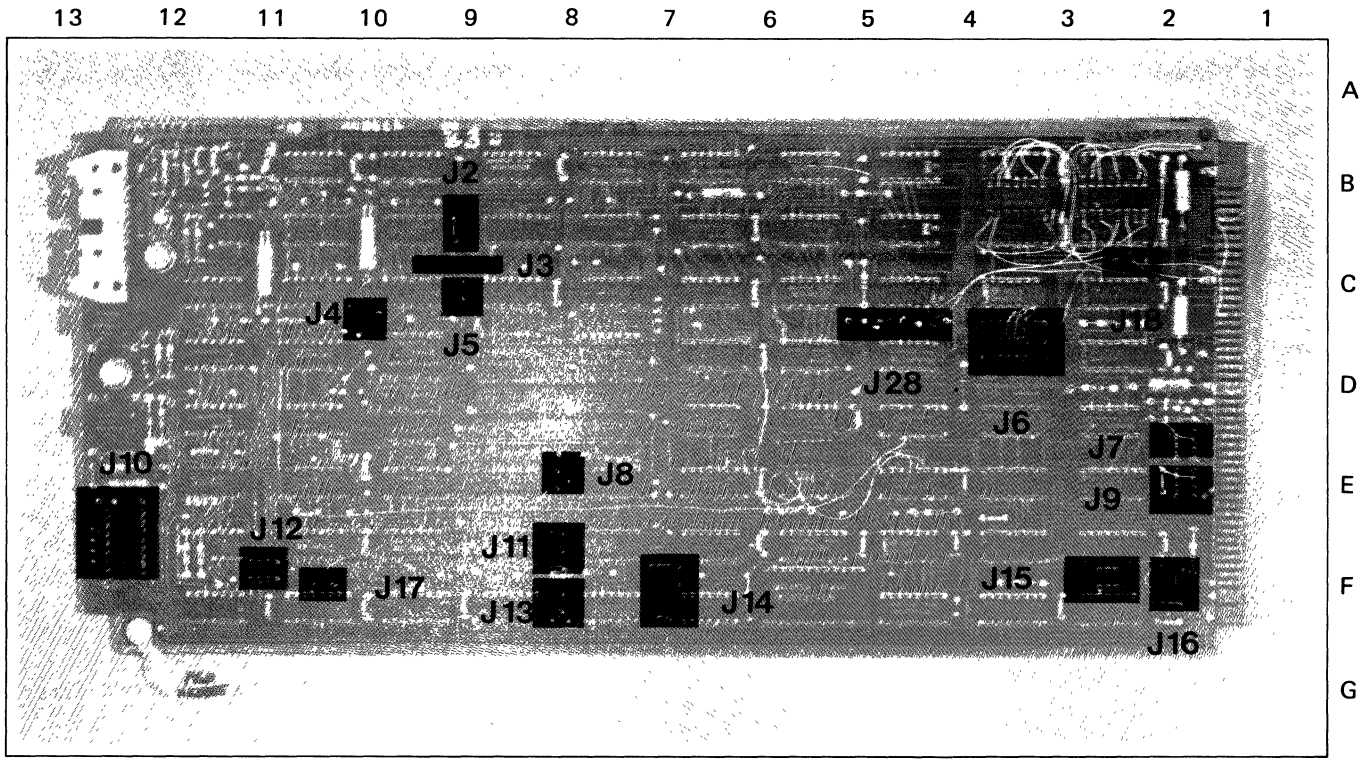
	Mode	Connect From	Connect To
	TTY	J3 - 2	J3 - 1
*	RS232	J3 - 2	J3 - 3

* Normal Setting

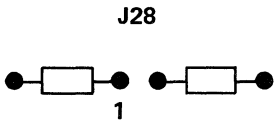
Typical Example: PSB ^{J6}FA00 Speed = 9600 intlev = none ^{3}stop = 2 ^{J14}data = 8 ^{J3}rcvdat = RS232 ^{6}Bus = 10 CRT = yes

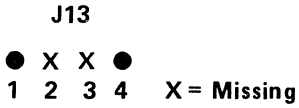
PSB

2102300G01



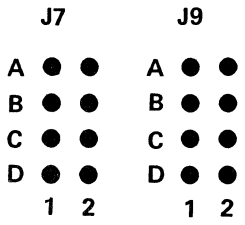
6 Bus Service





P	Processor Bus	Remove Jumper From J28 - 1 J18 - 4
I	M, I/O, or M/I Bus	Connect Jumper From J28 - 1 J18 - 4

7 Bus Service Level



CASP (J7 and J9)

From	To
1A	2B
1C	1D

Terminal

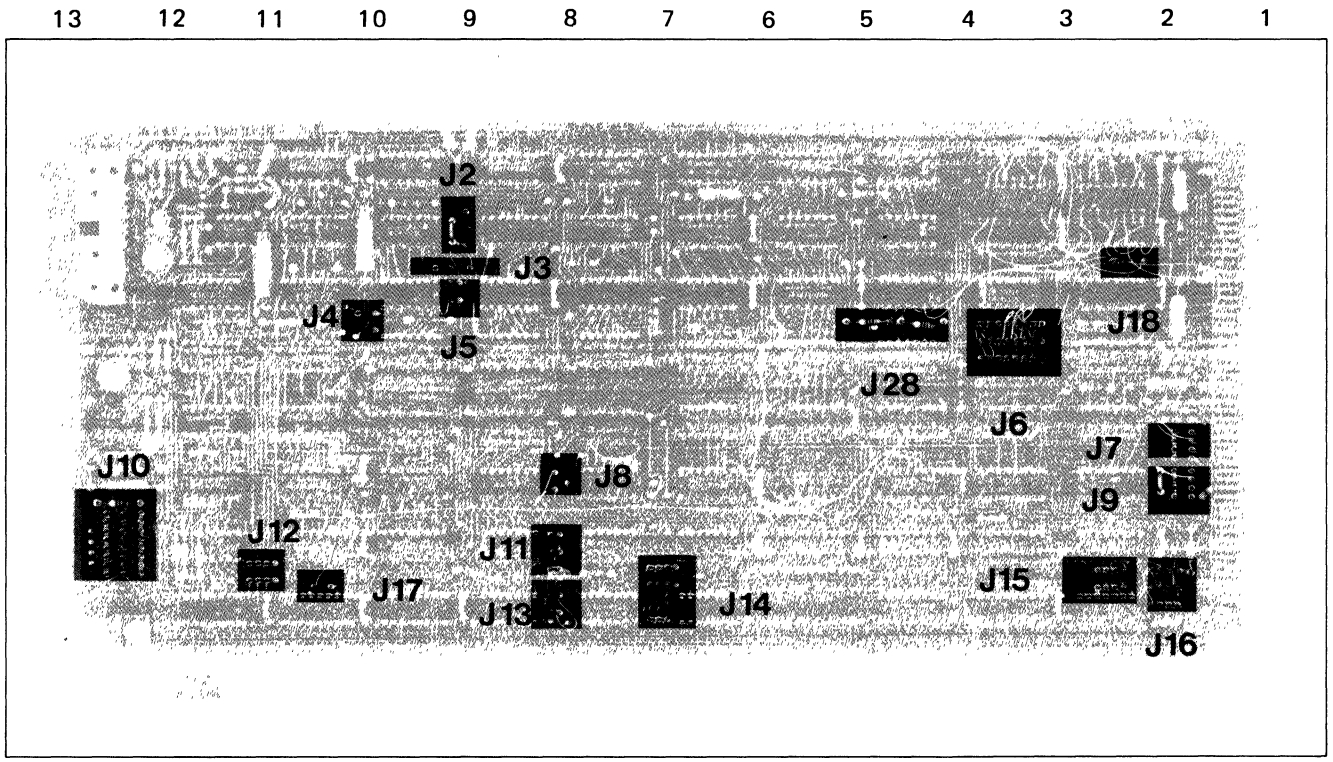
	Level	Connect J7 and J9	
		From	To
*	1	1A	2A
*	2	1B	2B
*	3	1C	2C
*	4	1D	2D
	None	J9 - 1A	J16 - 3A

*All applications for BBNCC

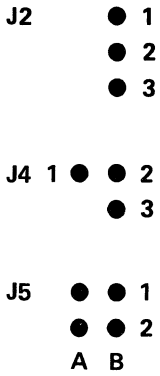
Typical Example: PSB ^{J6}FA00 Speed = 9600 intlev = none ^{3}stop = 2 ^{J14}data = 8 ^{J3}rcvdat = RS232 ^{6}Bus = IO CRT = yes

PSB

2102300G01



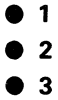
8 Auto Answer



Yes		No	
Connect		Disconnect	
From	To	From	To
J2-2	J2-1	J2-2	J2-1
J4-2	J4-3	J4-2	J4-3
J5-1B	J5-2B	J5-1B	J5-2B

* All applications for BBNCC

J2 CRT



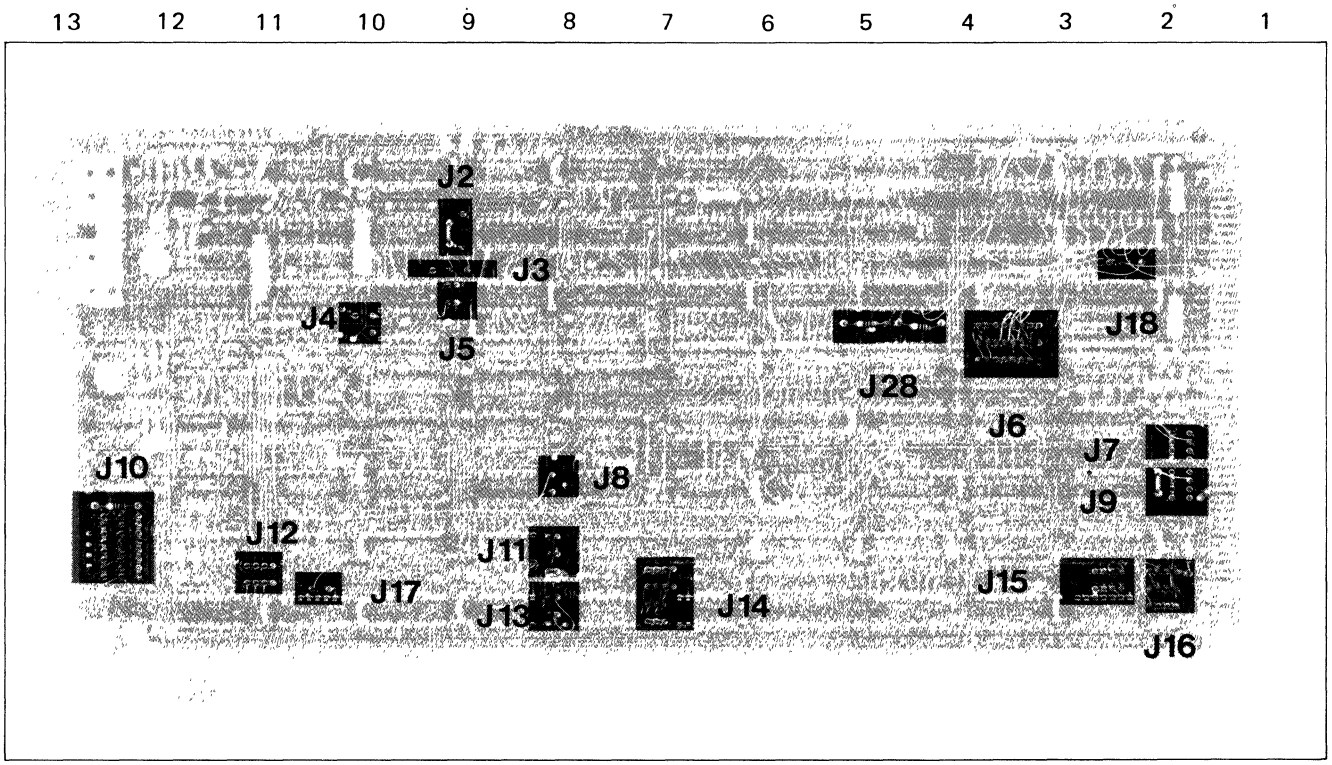
Yes		No	
Connect		Disconnect	
From	To	From	To
J2-2	J2-1	J2-2	J2-1

* All applications for BBNCC

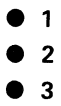
Typical Example: PSB ^{J6}FA00 Speed = 9600 intlev = none ^{3}stop = 2 ^{J14}data = 8 ^{J3}rcvdat = RS232 ^{6}Bus = 10 ^{J2}CRT = yes

PSB

2102300G01



J2 Device Ready

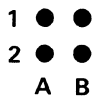


*

Yes		No	
Disconnect		Connect	
From	To	From	To
J2-2	J2-3	J2-2	J2-3

* All applications for BBNCC

J5 Data Set Ring Indicator



*

Yes		No	
Disconnect		Connect	
From	To	From	To
J5-1A	J5-2A	J5-1A	J5-2A

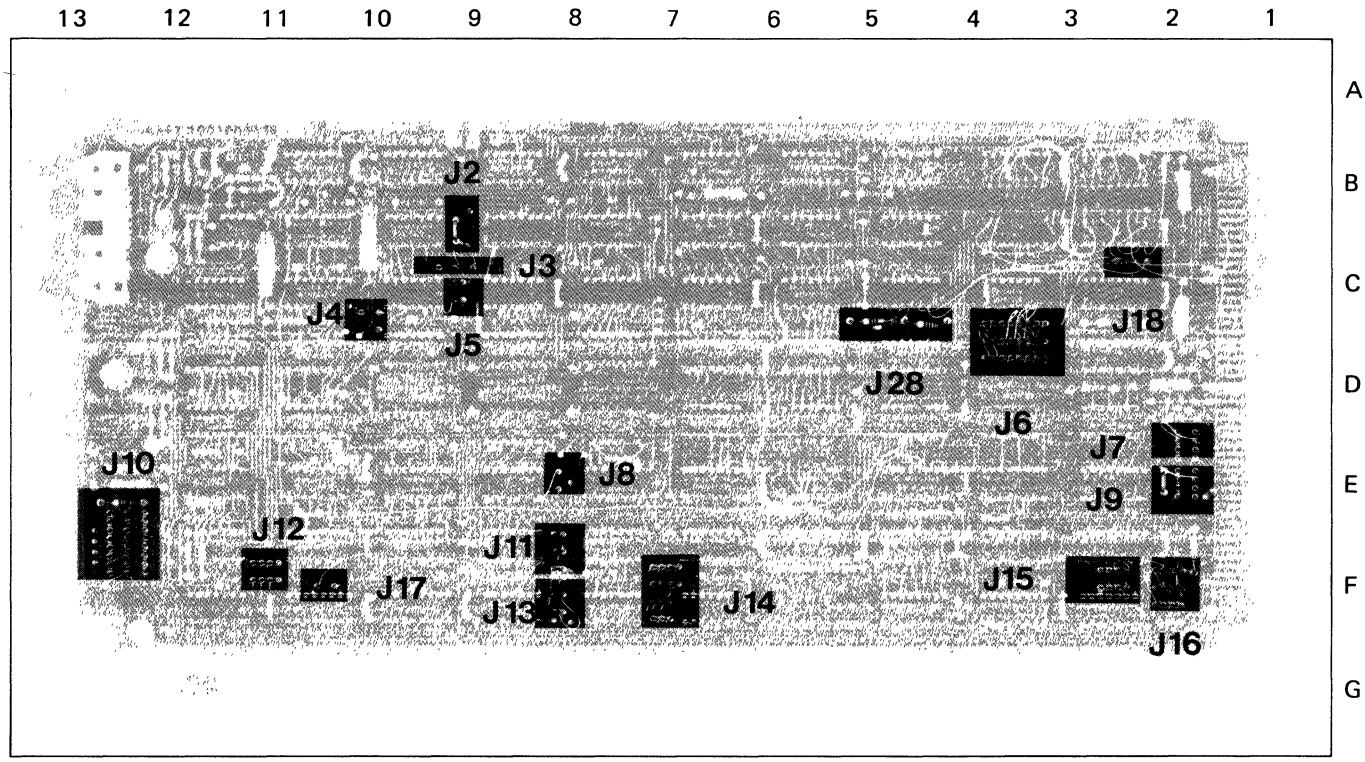
* All applications for BBNCC

Typical Example: PSB ^{J6}FA00 Speed = 9600 intlev = none ^{3}stop = 2 ^{J14}data = 8 ^{J3}rcvdat = RS232 ^{6}Bus = 10 CRT = yes

---

PSB

2102300G01



J8 BTA

● 1
● 2

*

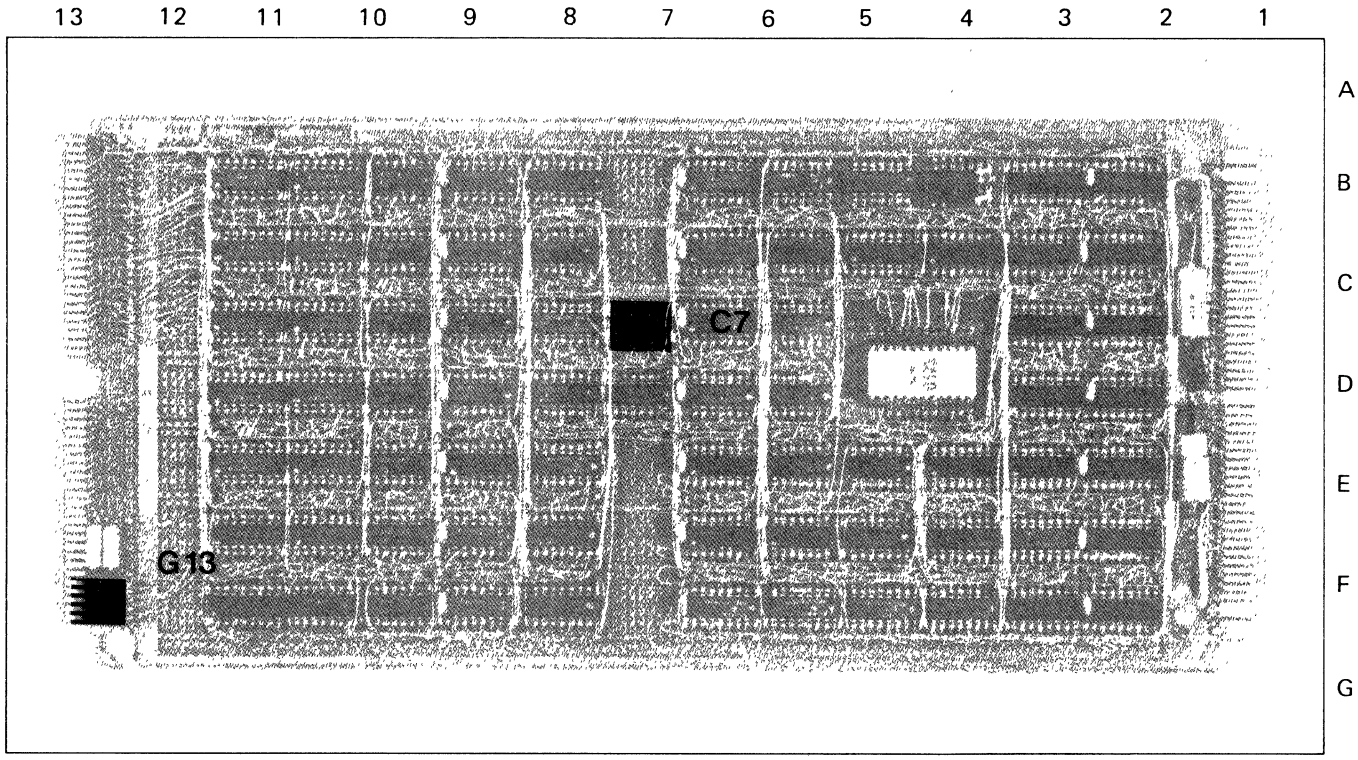
Yes		No	
Disconnect		Connect	
From	To	From	To
J8-1	J8-2	J8-1	J8-2

* All applications for BBNCC

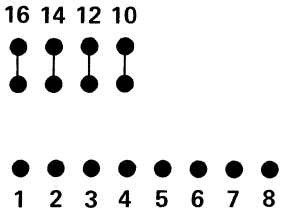
Typical Example: PSB ^{J6}FA00 Speed = 9600 intlev = none ^{3}stop = 2 ^{J14}data = 8 ^{J3}rcvdat = RS232 ^{6}Bus = 10 CRT = yes

RLD

2101529G09



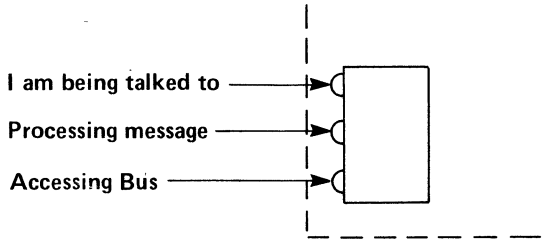
C7 Padding Identifier



PAD BIT	3	2	1	0
FROM	C7 - 16	C7 - 14	C7 - 12	C7 - 10
—	to	to	to	to
0	C7 - 1	C7 - 3	C7 - 5	C7 - 7
1	C7 - 2 •	C7 - 4 •	C7 - 6	C7 - 8

Note: If SRN has no entry beside RLD, then jumper C7 as PAD Bit "0"

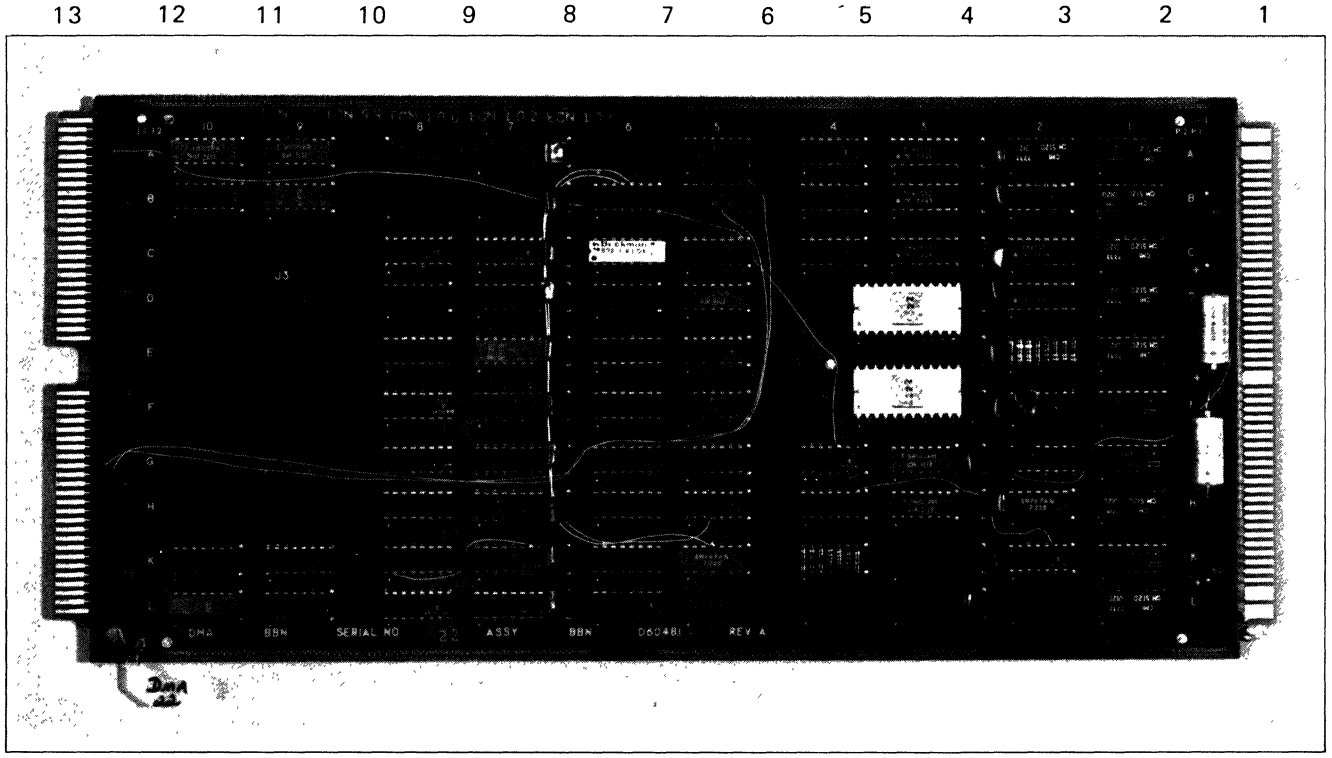
G13 Lights



Typical Example: RLD

DMA

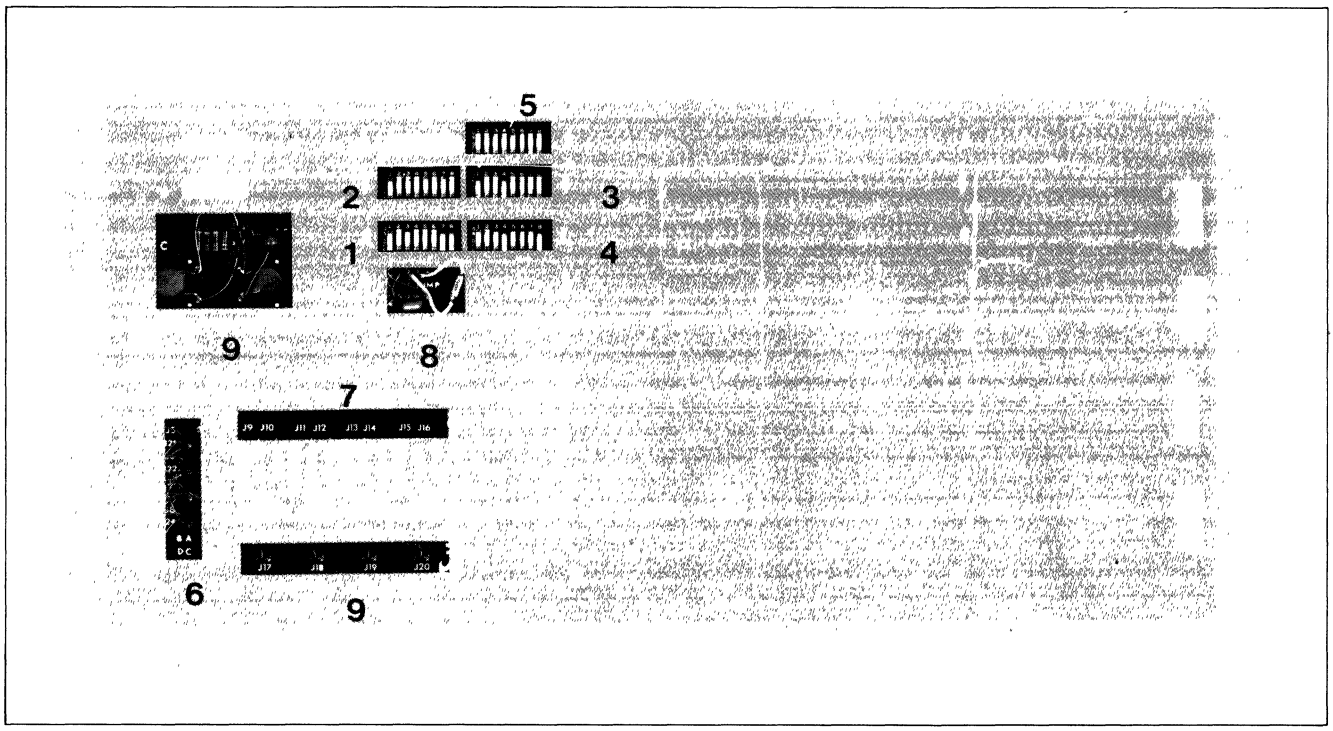
2100136G01



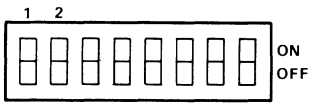
No jumpers are required
on the DMA board
for BBNCC applications

HST

2100538G01

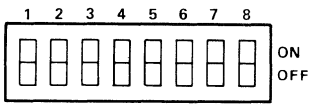


1 Pid Address



	SW1	SW2
E0	Off	Off
E8	Off	On
F0	On	Off
F8	On	On

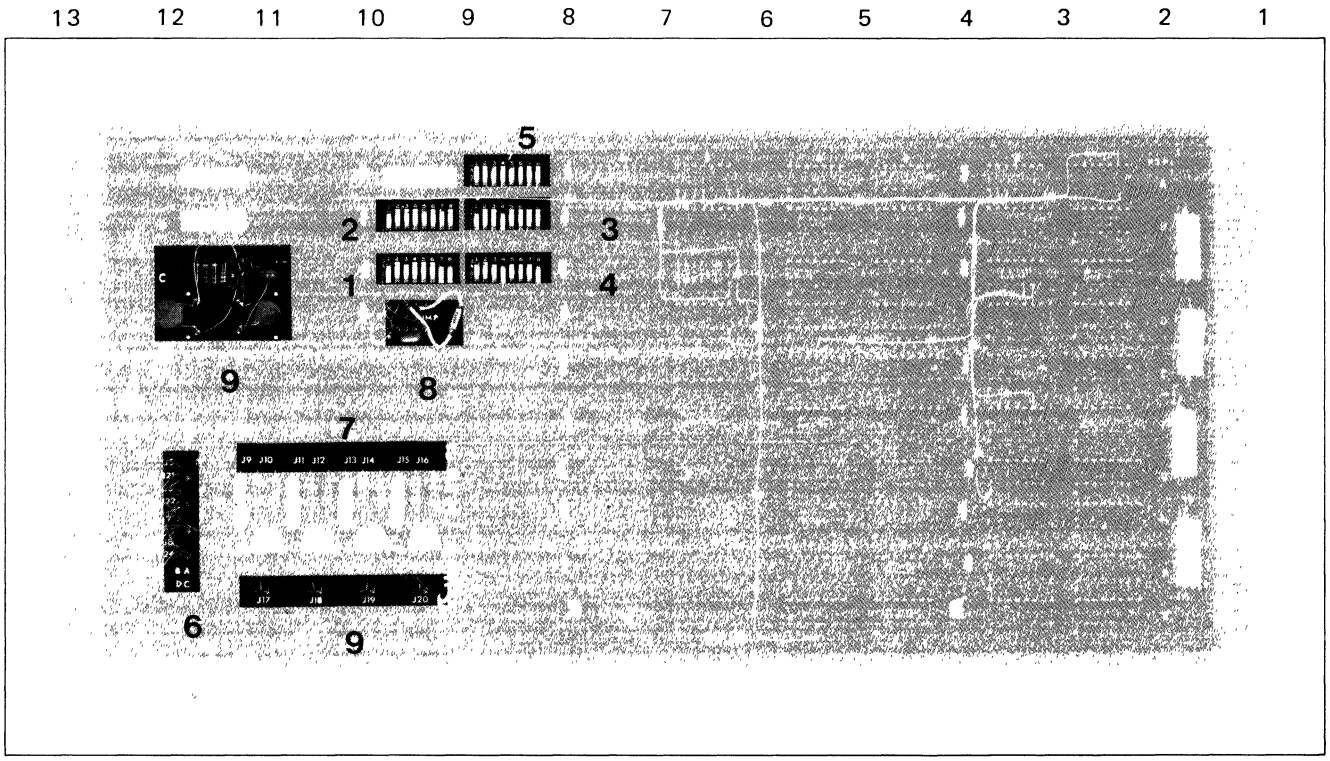
2 Transmit Pid



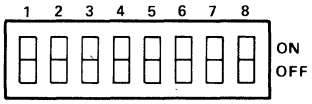
	MSB				LSB			
Switch	1	2	3	4	5	6	7	8

Typical Example: HST [4]E140 [1]PidAdd = E000 [2]PidTr = 90 [3]PidRcv = 80 [5]Dev = 00
[6]Rcv = Distant [7]Term = None [8]Rdy = E [8]Pad = P [9]Speed = L

HST

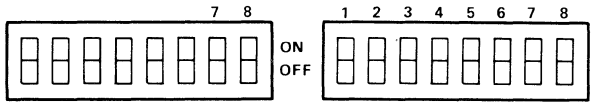


3 Receive Pid



	MSB				LSB			
Switch	1	2	3	4	5	6	7	8

1,4 Device Address



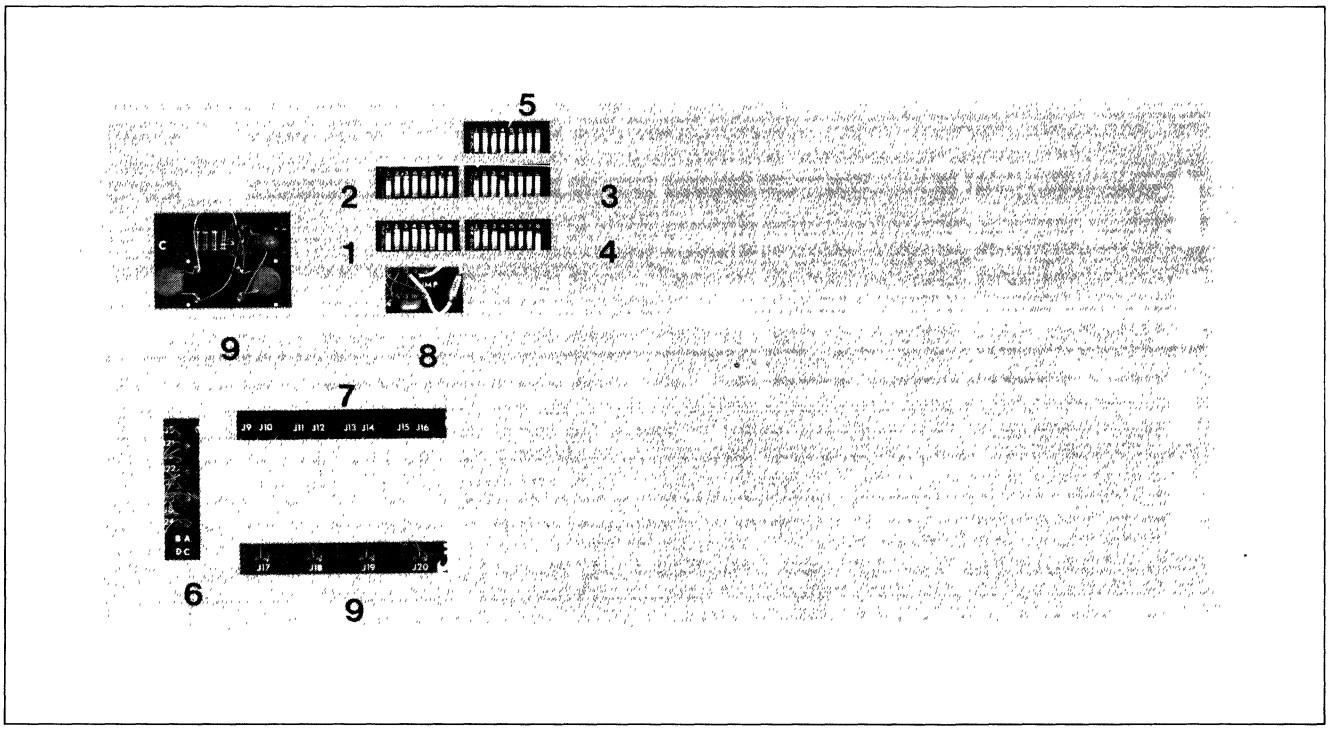
Switch	7	8	1	2	3	4	5	6	7	8
Bit	13	12	11	10	9	8	7	6	5	4

7 ON = E-8
7 8 ON = F-B

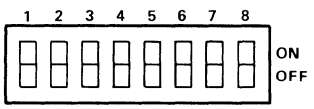
Typical Example: HST [4]E140 [1]PidAdd = E000 [2]PidTr = 90 [3]PidRcv = 80 [5]Dev = 00
[6]Rcv = Distant [7]Term = None [8]Rdy = E [8]Pad = P [9]Speed = L

HST

2100538G01

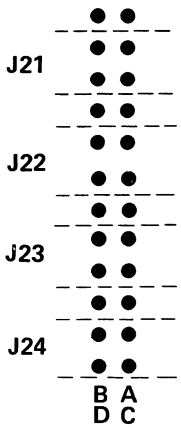


5 Device



	MSB				LSB			
Switch	1	2	3	4	5	6	7	8

6 Receivers



REFER TO 6 ON NEXT PAGE

Local

FROM	TO
J21B	J21C
J22B	J22C
J23B	J23C
J24B	J24C
J21D	J2 - 51
J22D	J2 - 52
J23D	J2 - 53
J24D	J2 - 54

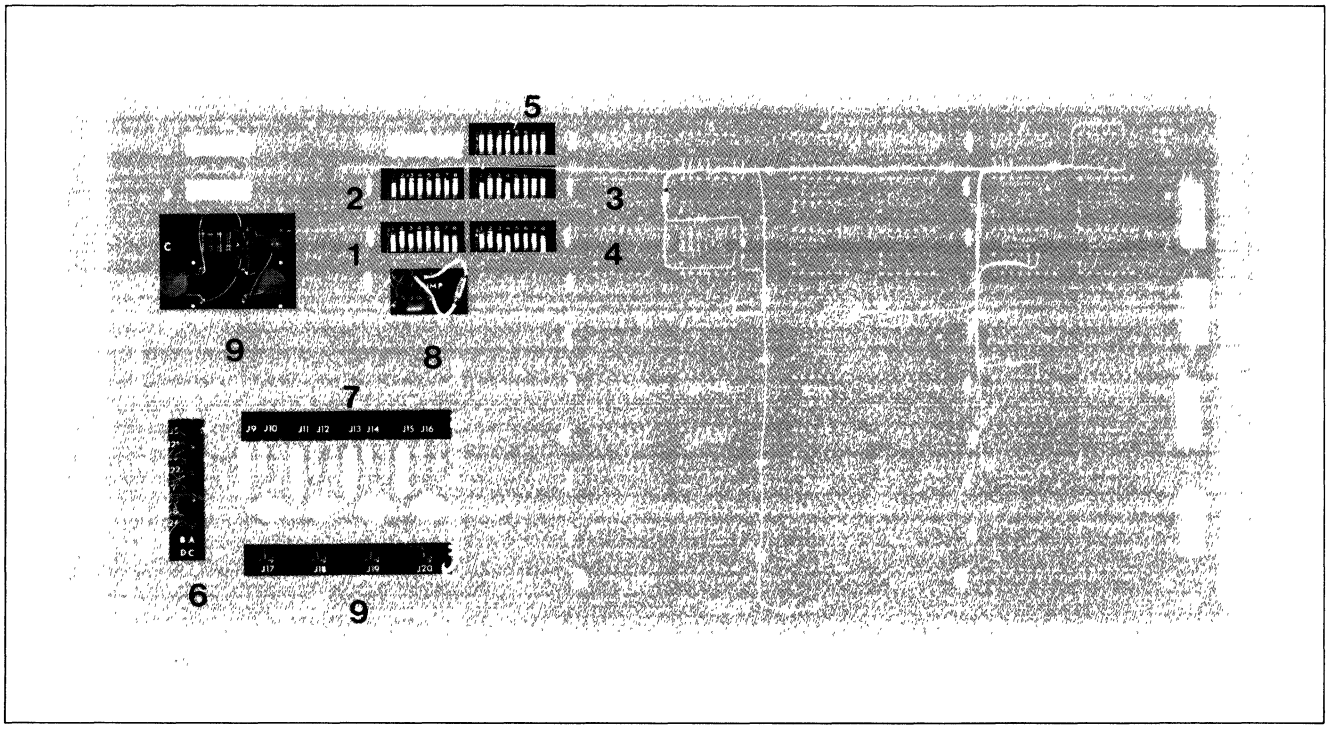
Distant

FROM	TO
J21B	J21D
J22B	J22D
J23B	J23D
J24B	J24D

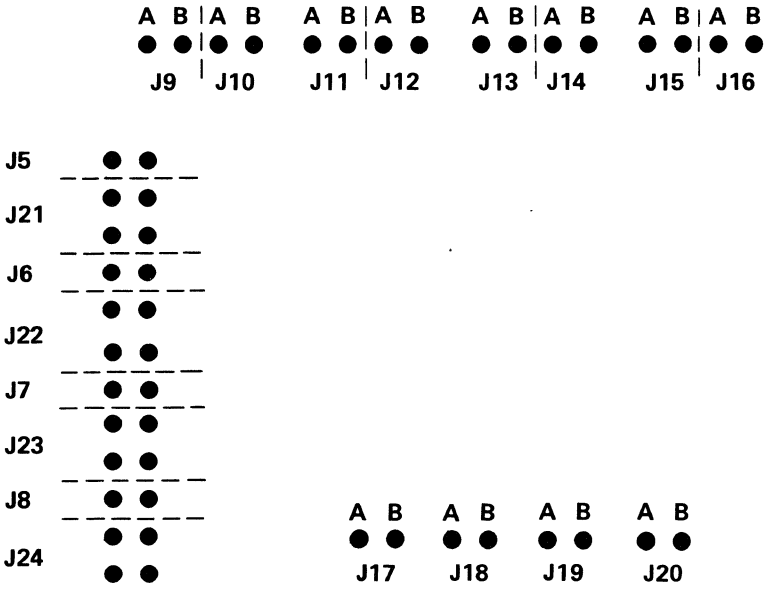
Typical Example: HST [4]E140 [1]PidAdd = E000 [2]PidTr = 90 [3]PidRcv = 80 [5]Dev = 00
[6]Rcv = Distant [7]Term = None [8]Rdy = E [8]Pad = P [9]Speed = L

HST

2100538G01



6,7,9 Termination Options



Distant Drivers (DD)

FROM	TO
J9A	J9B
J10A	J10B
J11A	J11B
J12A	J12B
J13A	J13B
J14A	J14B
J15A	J15B
J16A	J16B

Distant Receivers (DR)

FROM	TO
J5A	J5B
J6A	J6B
J7A	J7B
J8A	J8B
J21	J21B
J22A	J22B
J23A	J23B
J24A	J24B

Local Drivers (LD)

FROM	TO
J17A	J17B
J18A	J18B
J19A	J19B
J20A	J20B

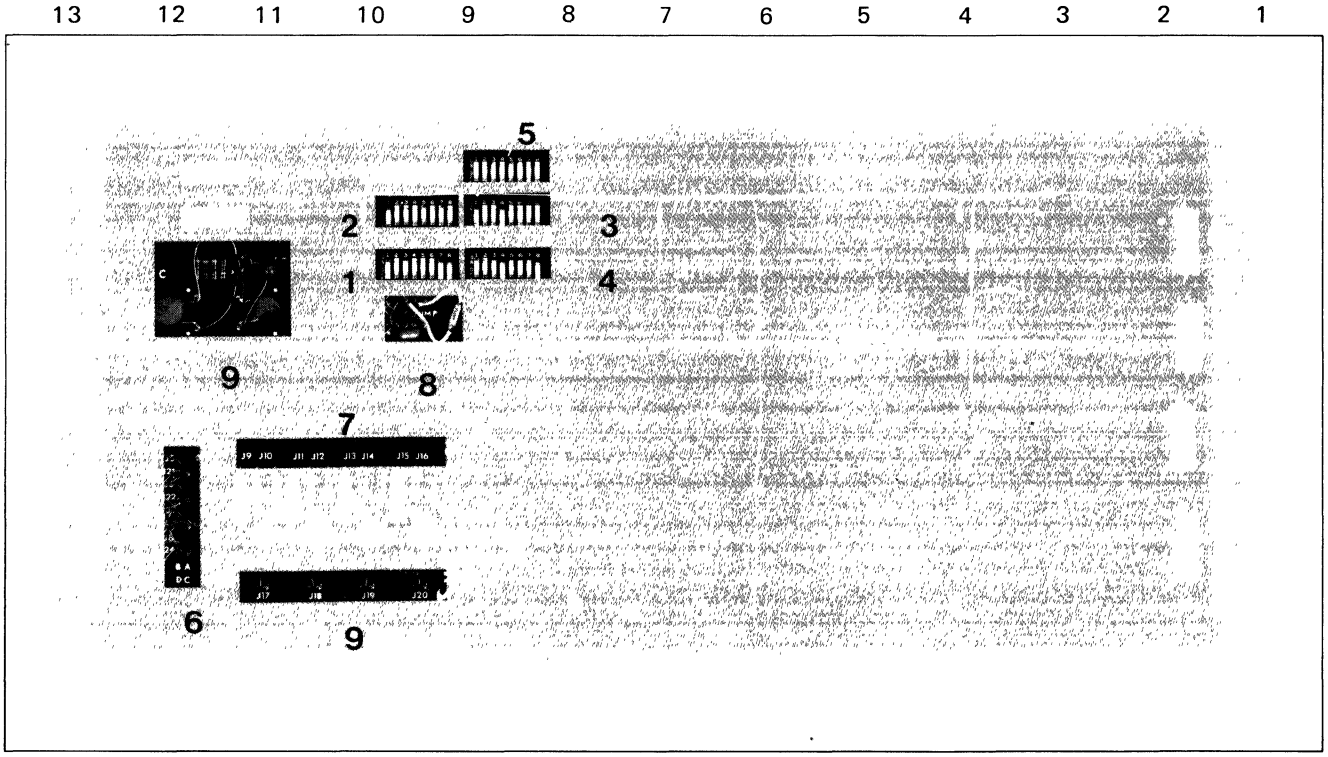
Local Receivers (LR)

FROM	TO
J5A	J5B
J6A	J6B
J7A	J7B
J8A	J8B

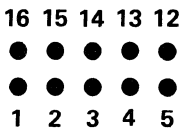
Typical Example: HST [4]E140 [1]PidAdd = E000 [2]PidTr = 90 [3]PidRcv = 80 [5]Dev = 00
[6]Rcv = Distant [7]Term = None [8]Rdy = E [8]Pad = P [9]Speed = L

HST

2100538G01



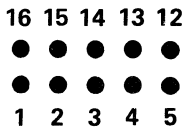
8 Ready Line



	Type	Connect	
		From	To
*	ENABLE	3	14
		4	13
		5	12
	DISABLE	13	14

*Normal

8 Padding



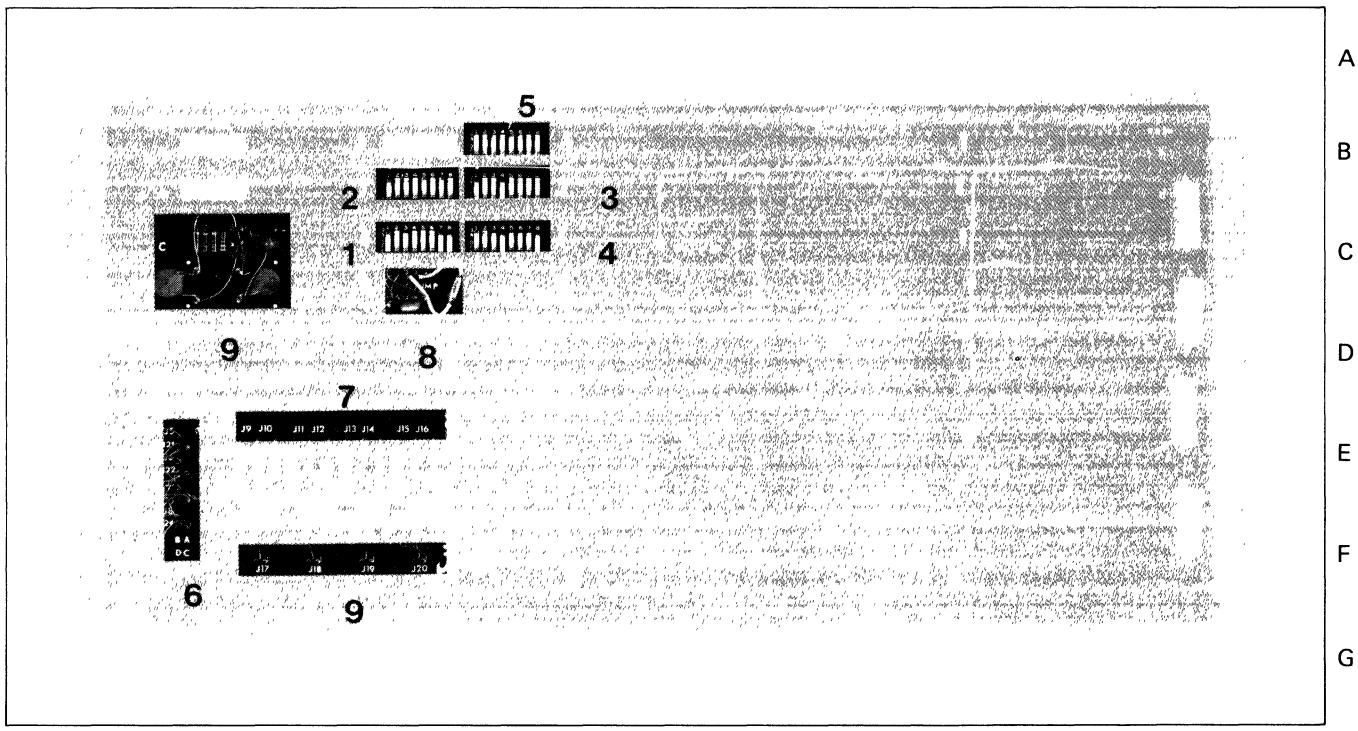
	Type	Connect	
		From	To
*	ONE	1	16
		2	15
	NO ONE	1	2
		15	16

* Normal

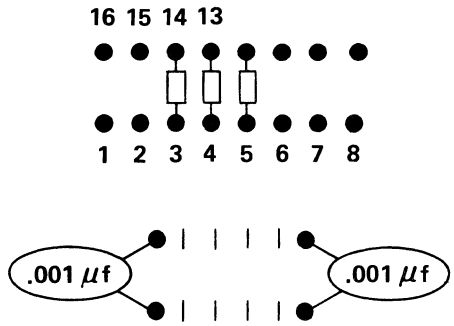
Typical Example: HST [4]E140 [1]PidAdd = E000 [2]PidTr = 90 [3]PidRcv = 80 [5]Dev = 00
[6]Rcv = Distant [7]Term = None [8]Rdy = E [8]Pad = P [9]Speed = L

HST

2100538G01



9 Speed



	High Speed (H)	Low Speed (L)
Transmit	Remove .001 μfd capacitor jumpered between C10-6 and C10-11	Jumper .001 μfd capacitor between C10-6 and C10-11
Receive	Remove .001 μfd capacitor jumpered between C10-7 and C10-10	Jumper .001 μfd capacitor between C10-7 and C10-10

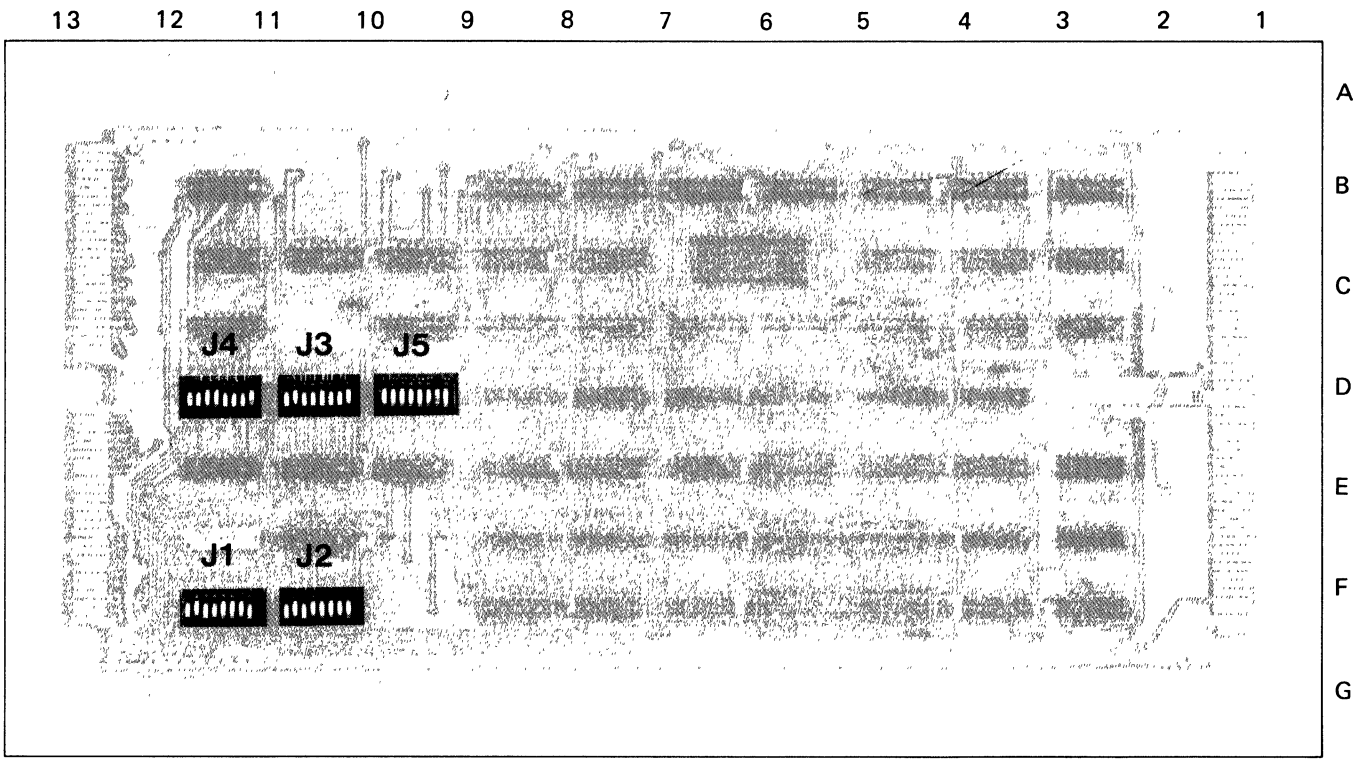
Notes

1. For 1822 local and distant levels only terminate drivers.
2. For V.35 option terminate both distant drivers and receivers.
3. In all cases, if two HSTs are paired for redundancy, only terminate at most one set of drivers and/or receivers.
4. Local receiver and distant receiver strappings are mutually exclusive.

Typical Example: HST [4]E140 [1]PidAdd = E000 [2]PidTr = 90 [3]PidRcv = 80 [5]Dev = 00
[6]Rcv = Distant [7]Term = None [8]Rdy = E [8]Pad = P [9]Speed = L

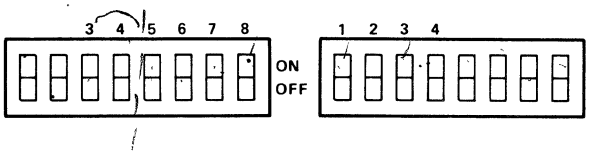
MLX/MHX

2100228G01



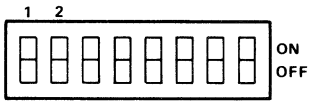
J 1,2 Device Address

1,3,4 = F-BUS
3 = E-BUS



Switch	3	4	5	6	7	8		1	2	3	4
Bit	13	12	11	10	9	8		7	6	5	4

J 1 Pid Address

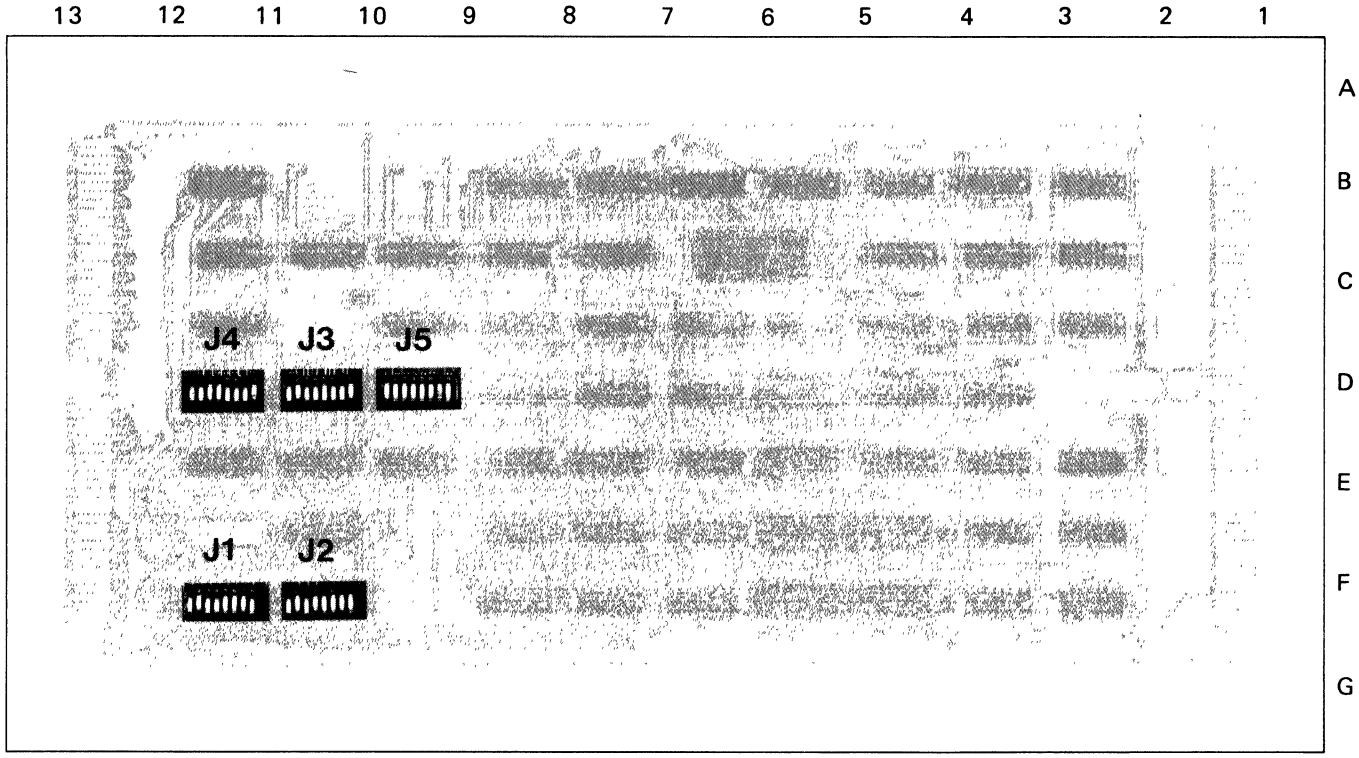


	SW1	SW2
E0	Off	Off
E8	Off	On
F0	On	Off
F8	On	On

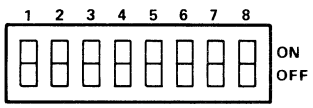
Typical Example: MHX $^{J1, J2}$F100 ^{J1}PidAdd = F000 ^{J3}PidTr = CE ^{J4}PidRcv = D6 ^{J5}Dev = 03

MLX/MHX

2100228G01

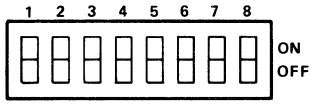


J3 Transmit Pid



Switch	1	2	3	4	5	6	7	8
Byte	MSB				LSB			

J4 Receive Pid

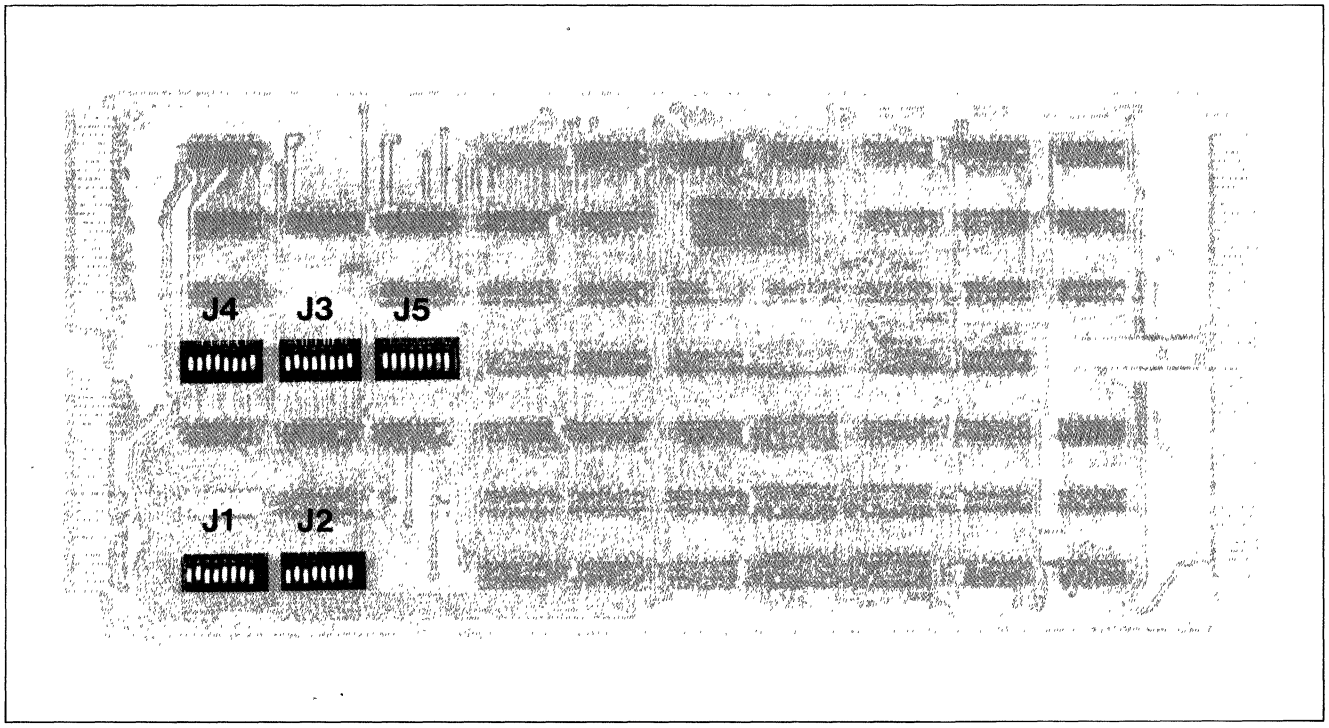


Switch	1	2	3	4	5	6	7	8
Byte	MSB				LSB			

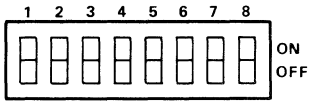
Typical Example: MHX $^{J1, J2}$F100 ^{J1}PidAdd = F000 ^{J3}PidTr = CE ^{J4}PidRcv = D6 ^{J5}Dev = 03

MLX/MHX

2100228G01



J5 Device No.

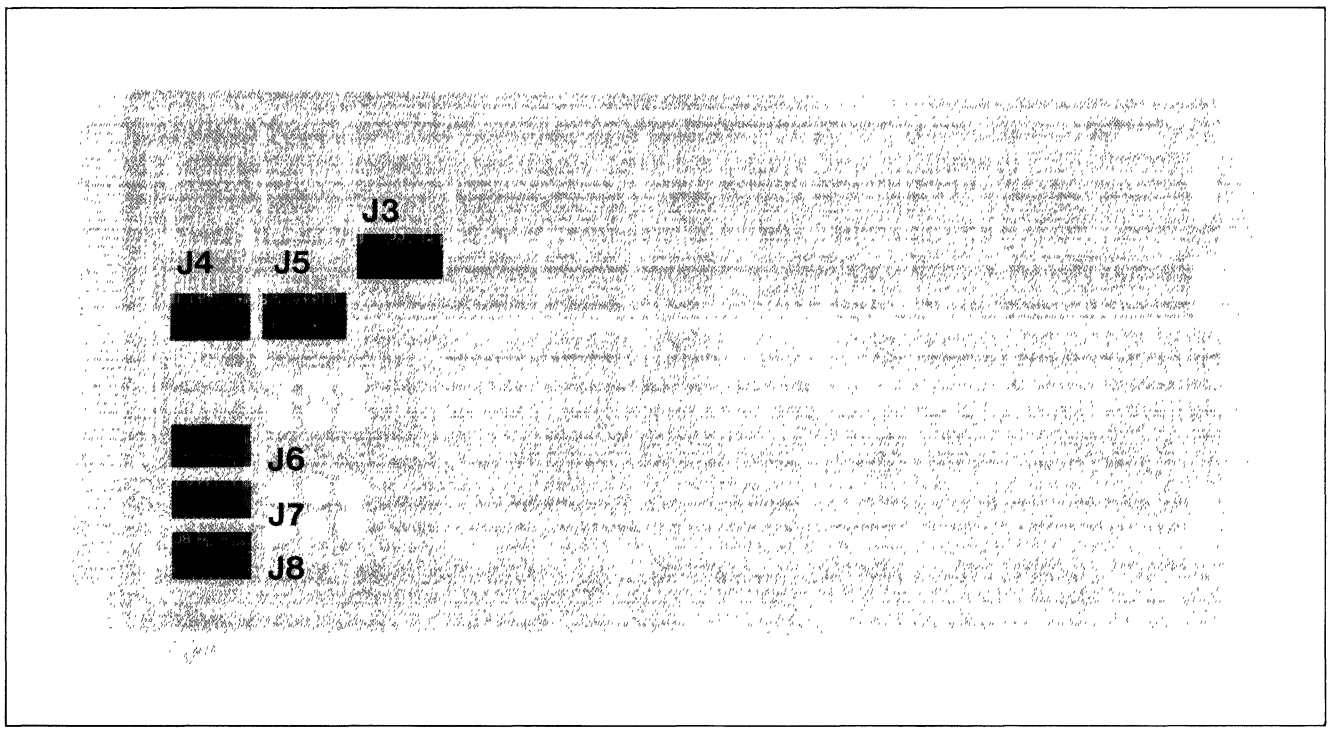


Switch	1	2	3	4	5	6	7	8
Byte	MSB				LSB			

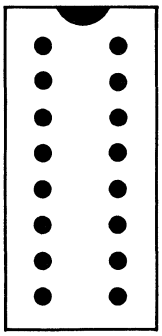
Typical Example: MHX $^{J1, J2}$F100 ^{J1}PidAdd = F000 ^{J3}PidTr = CE ^{J4}PidRcv = D6 ^{J5}Dev = 03

MUR

2101730G01



1 Protocol

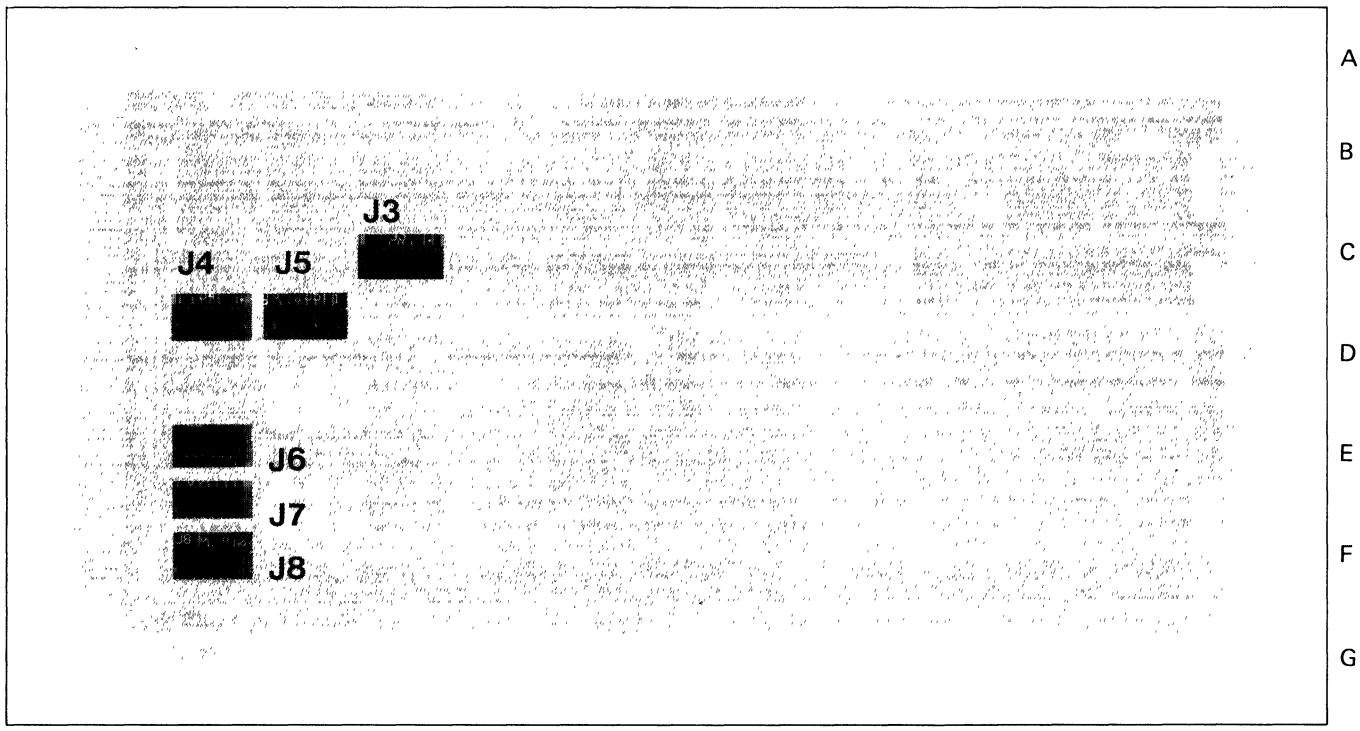


Protocol	J4	J6	J7	J8	J5
188-114	HA1-T or HA1-U	HA1-T or HA1-U	HA1-T or HA1-U	HA1-T or HA1-U	HBI
RS 422					HBI
188-100					HBI
					HBI
Bell 303	HA2-T or HA2-U	HA2-T or HA2-U	HA2-T or HA2-U	HA2-T or HA2-U	HB2
RS 232	HA3-T or HA3-U	HA3-T or HA3-U	HA3-T or HA3-U	HA3-T or HA3-U	HB3
RS232 Inverted Receive and Transmit Data	HA3-T or HA3-U	HA3-T or HA3-U	HA3-T or HA3-U	HA3-T or HA3-U	HB3
V.35	HA1-T or HA1-U	HA1-T or HA1-U	HA1-T or HA1-U	HA1-T or HA1-U	HB4
Bell 306 (Psat)	HAI-T or HAI-U	HA6-T	HA5-T	HA1-T or HA1-U	HB1
RS-449 (Psat)	—	HA7-T	HA8-T	HA9-T	HB5

Typical Example: MUR [1]Protocol = P3-T [2]RefClk = Internal [3]IntCl < =48KB

MUR

2101730G01



2 Reflected Clock Source Header

Reflected Clock Source	J3 Header
Internal	HC1
Transmit clock	HC2
Receive clock	HC3
Internal plus Honeywell compatible rec. data inversion *	HC4

* See Note - 3

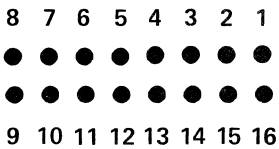
Notes: 1 - For additional information (config.), see mech. ref (Doc: MUR-05)

2 - For all jumpers (JMP) shown on headers, use 30 AWG – INSULATED wire

3 - For connection to Honeywell modem interfaces, use inverted receive data option

4 - For connection to Honeywell modem interfaces, use inverted driver data option, header J7

3 Internal Clock

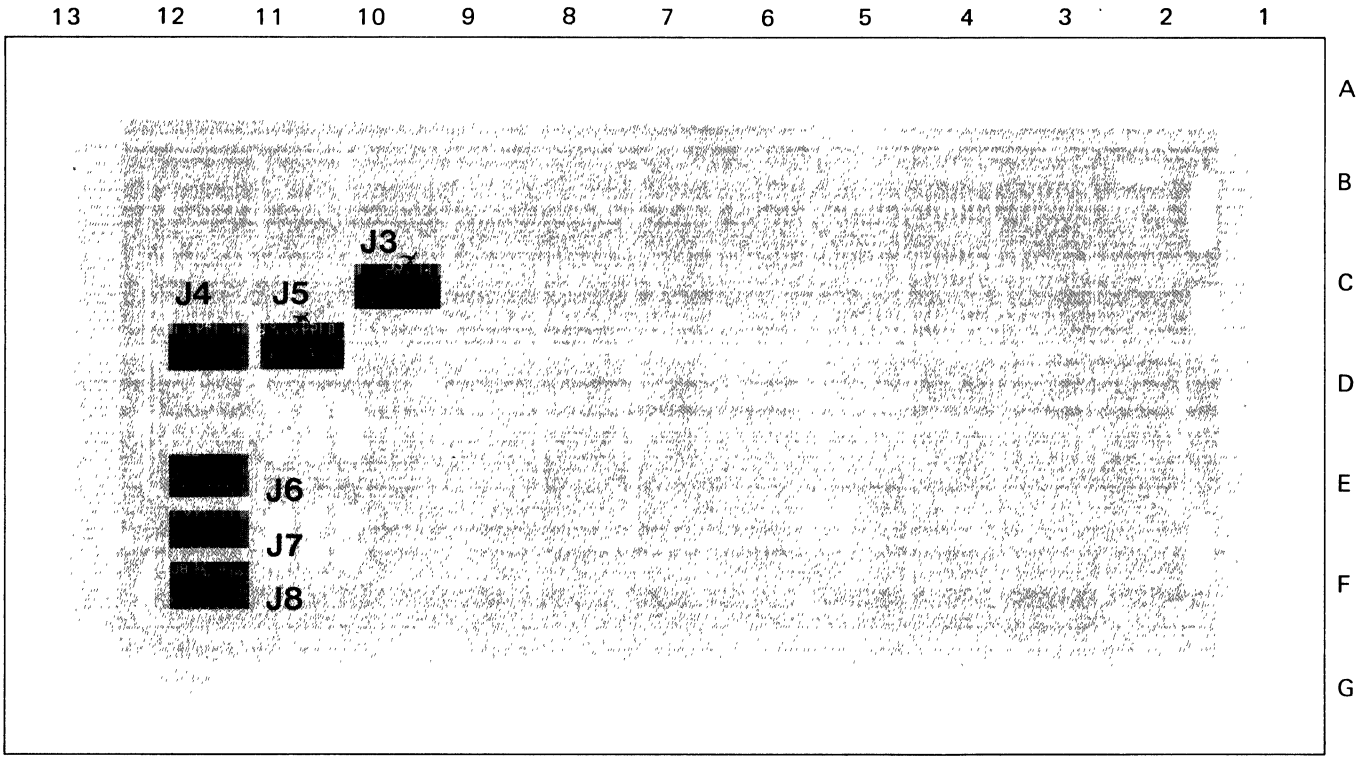


	Pin	Baud Rate
Pin # 16 to:	1	12.5 MB
	2	6.25 MB
	3	3.125 MB
	4	1.5 KB
	5	780 KB
	6	390 KB
	7	195 KB
	8	96 KB
	9	48 KB

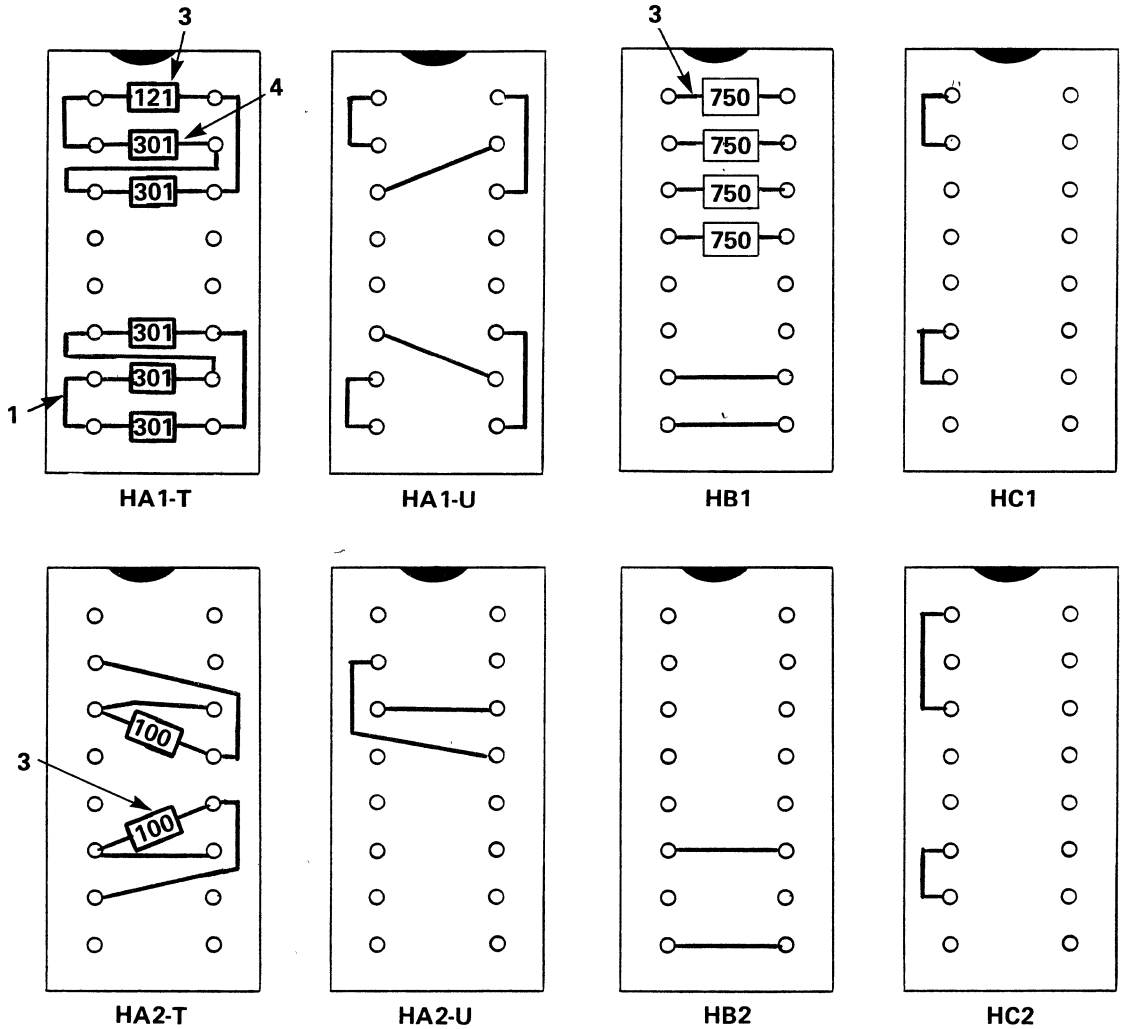
Typical Example: MUR [1]Protocol = P3-T [2]RefClk = Internal [3]IntCl < = 48KB

MUR

2101730G01

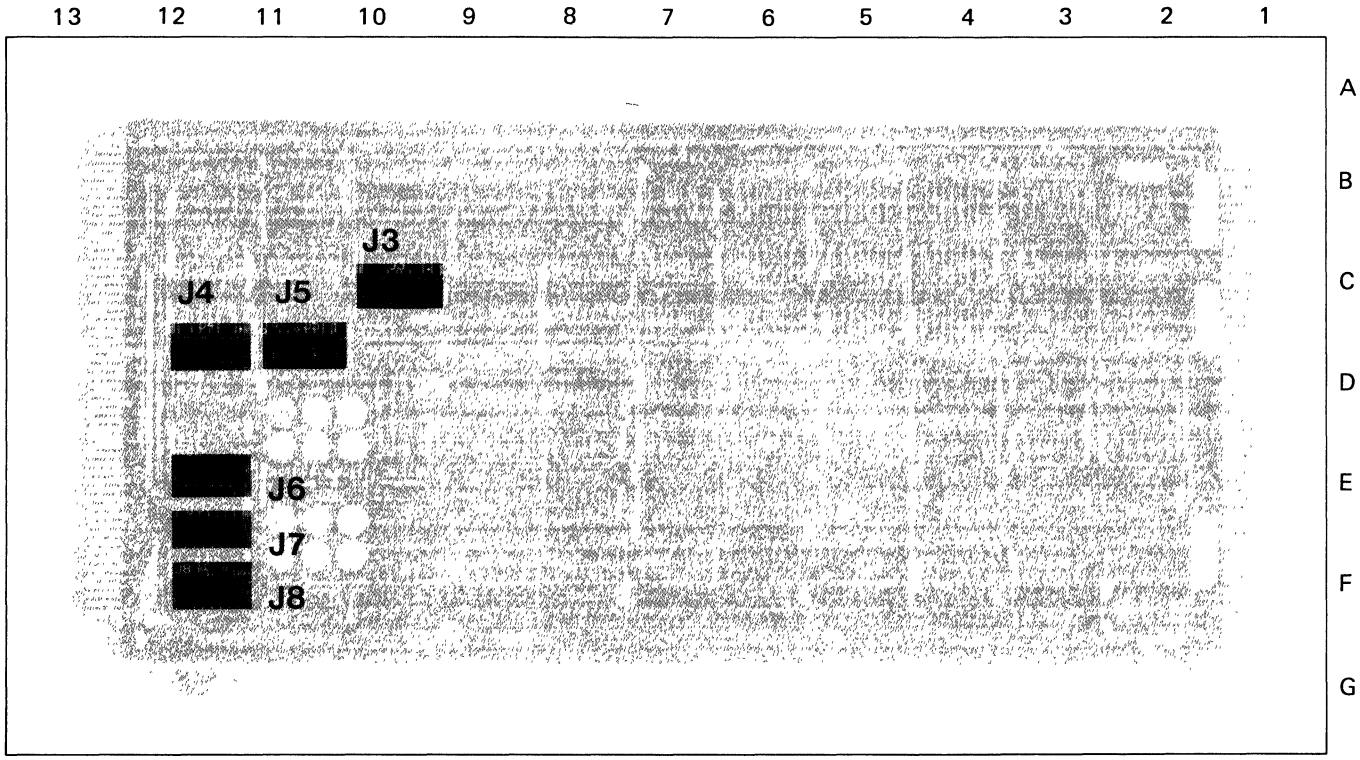


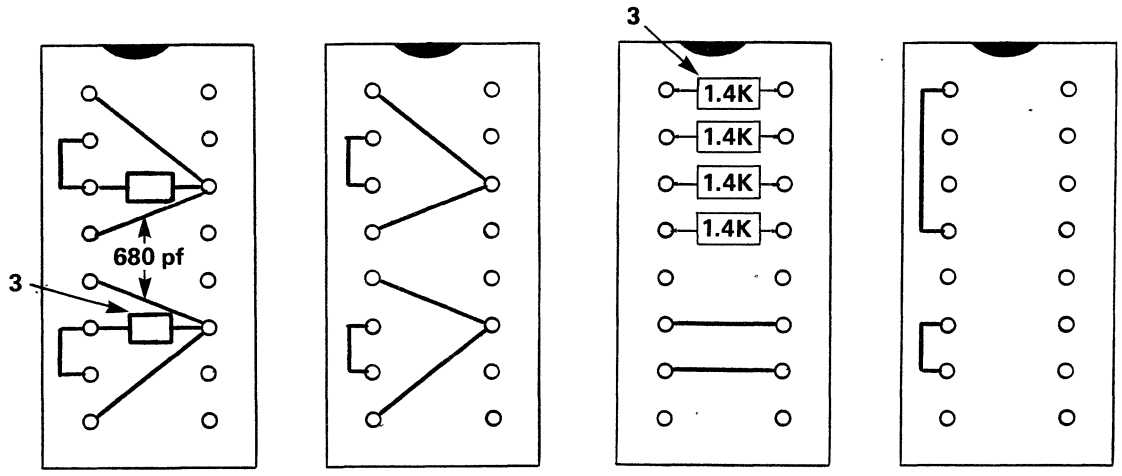
Header Assemblies

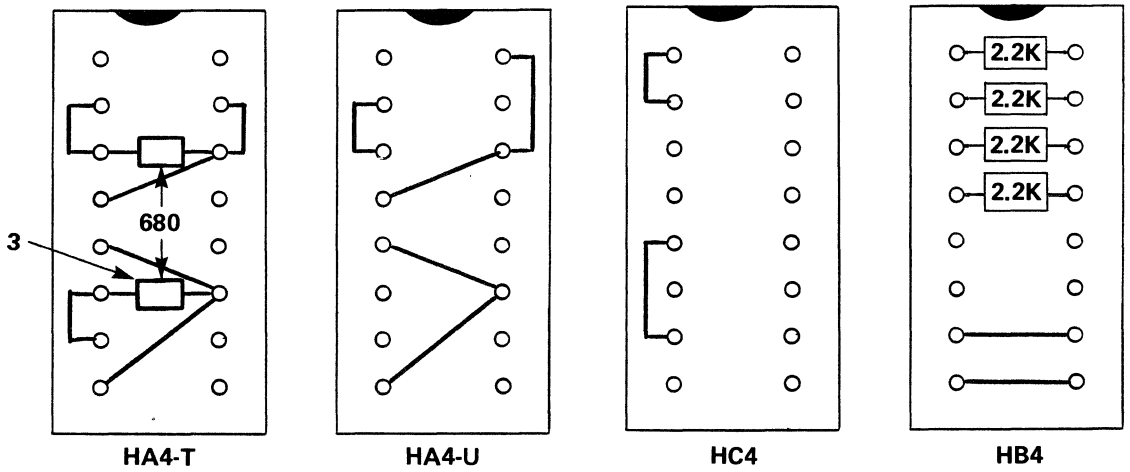


MUR

2101730G01

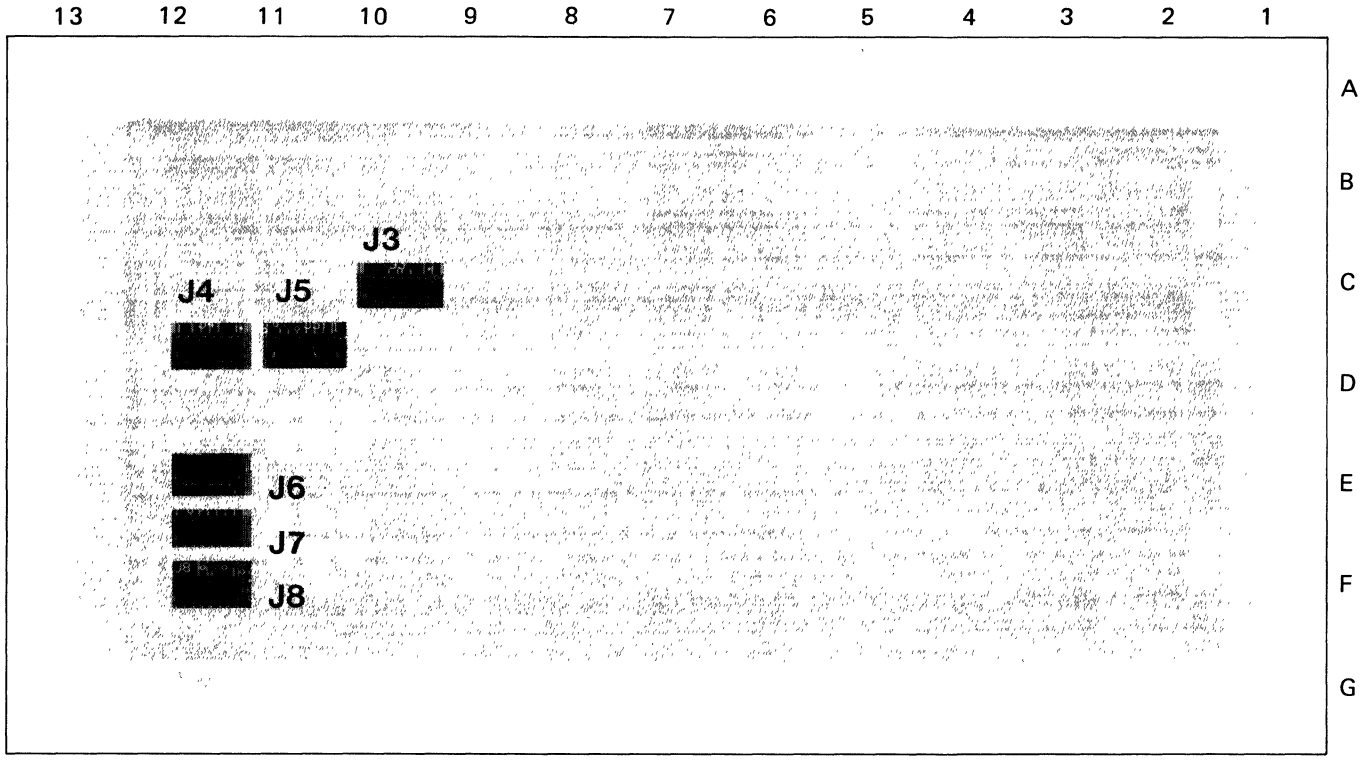


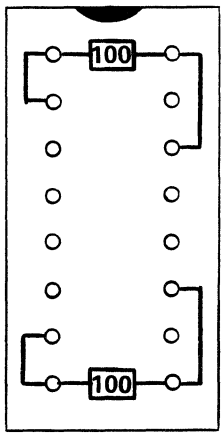




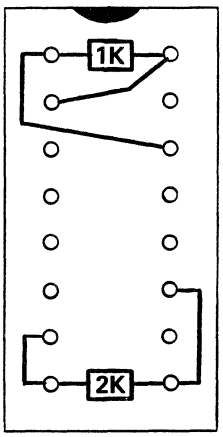
MUR

2101730G01

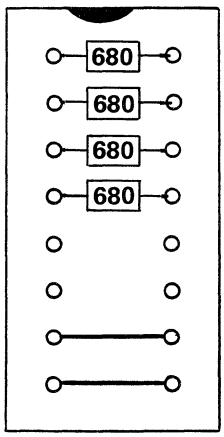




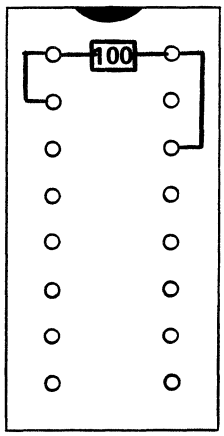
HA7-T



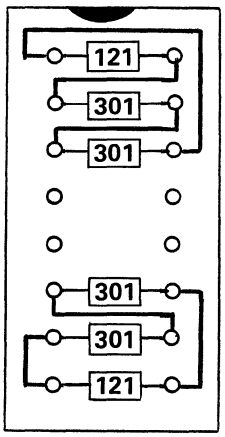
HA8-T



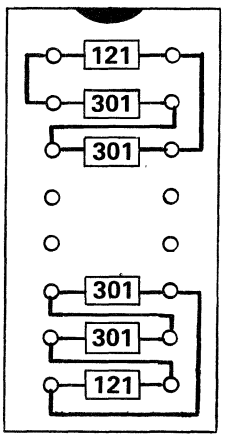
HB5



HA9-T



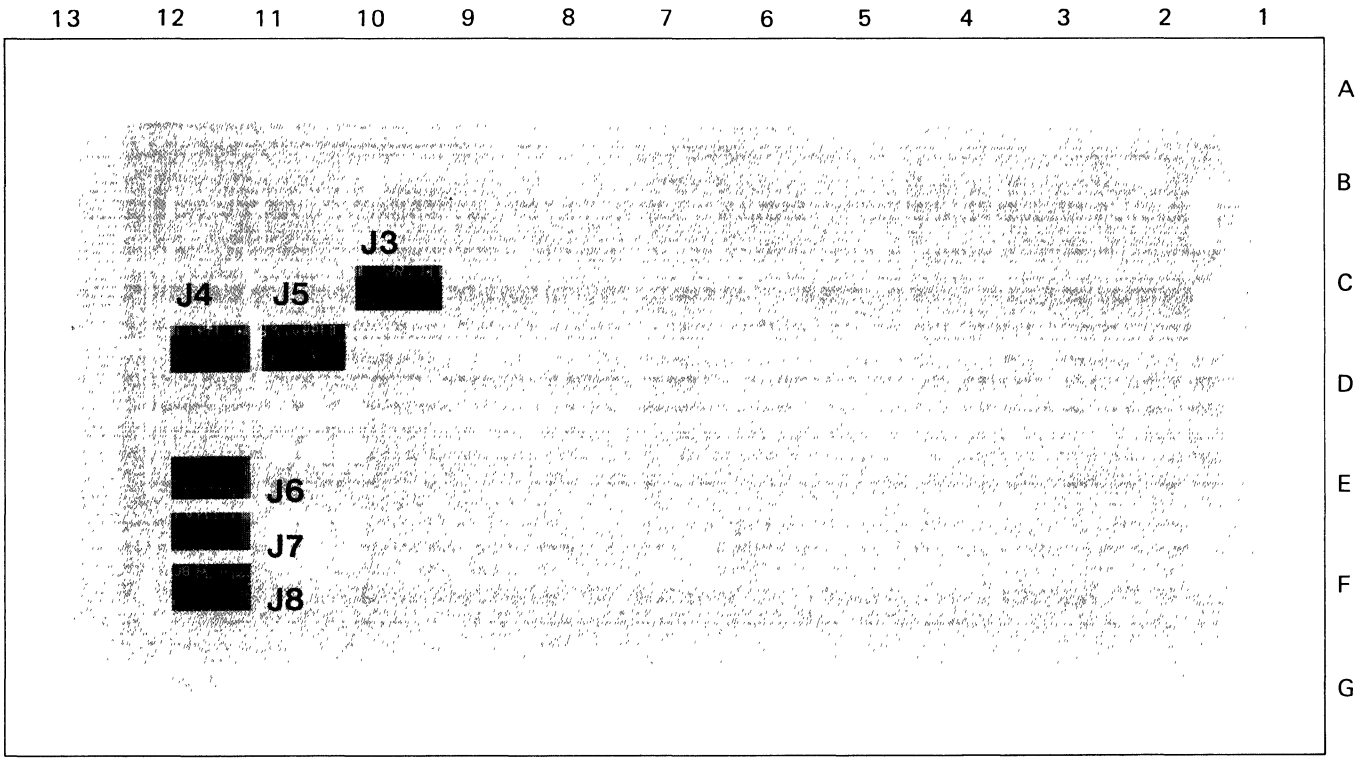
HA5-T



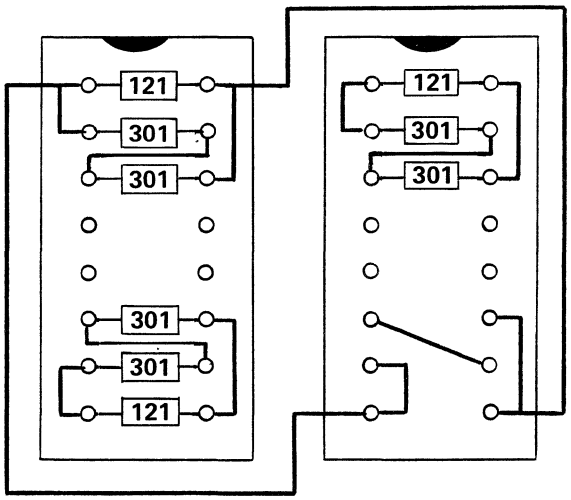
HA6-T

MUR

2101730G01

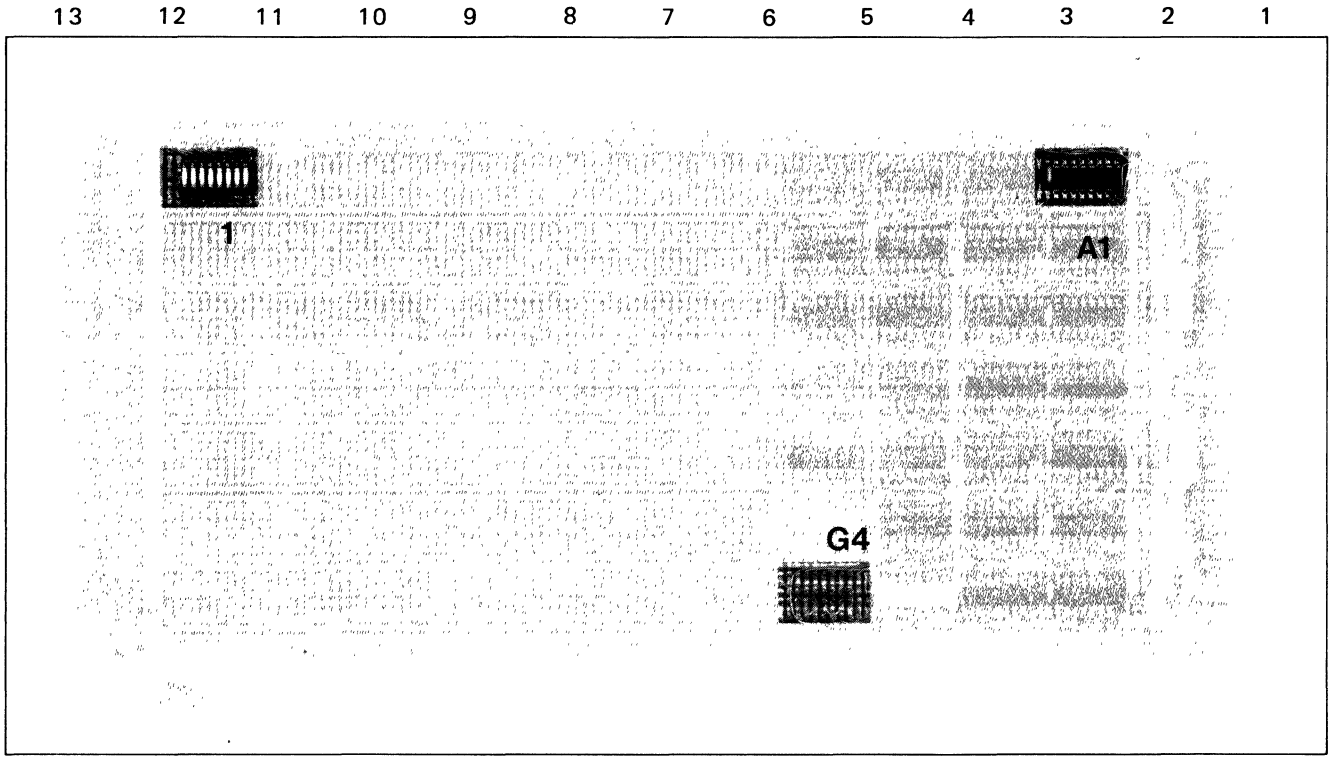


HAI-TMIC (J6 and J7) (NSA only)



PAR

2101529G08



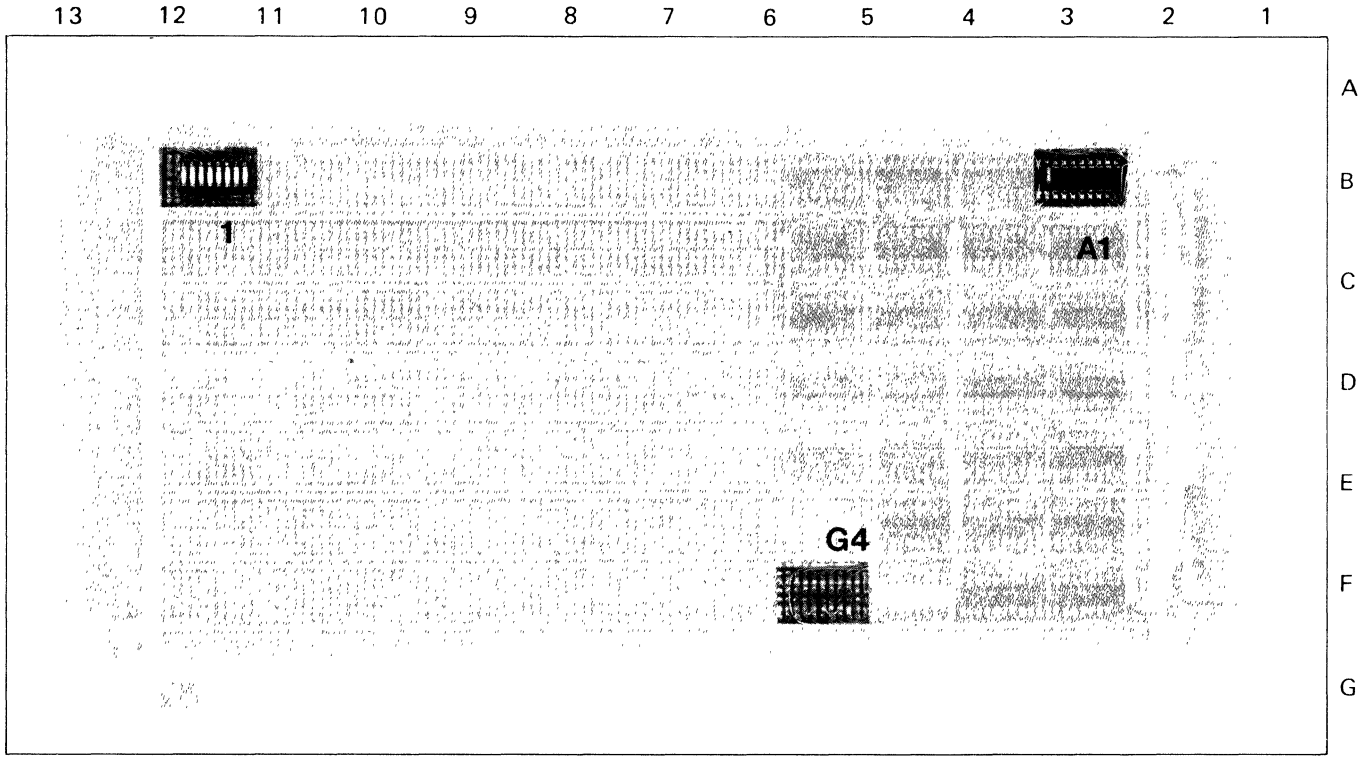
1 Switches

All switches are off
for all BBNCC applications

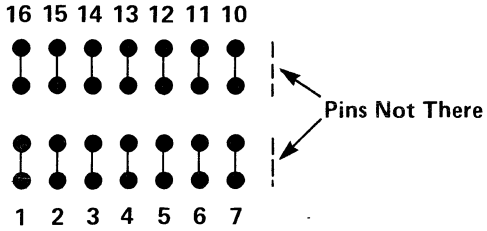
Typical Example: PAR ^{G4}PAR = Feedback ^{G4}All/IO/XMR = IO ^{A1}Busstick = 0

PAR

2101529G08

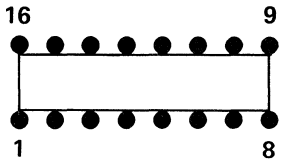


G4 Write Source Check



	Jumper
Classical	—
Feedback	G4-1 to G4-16

A1 Busstick

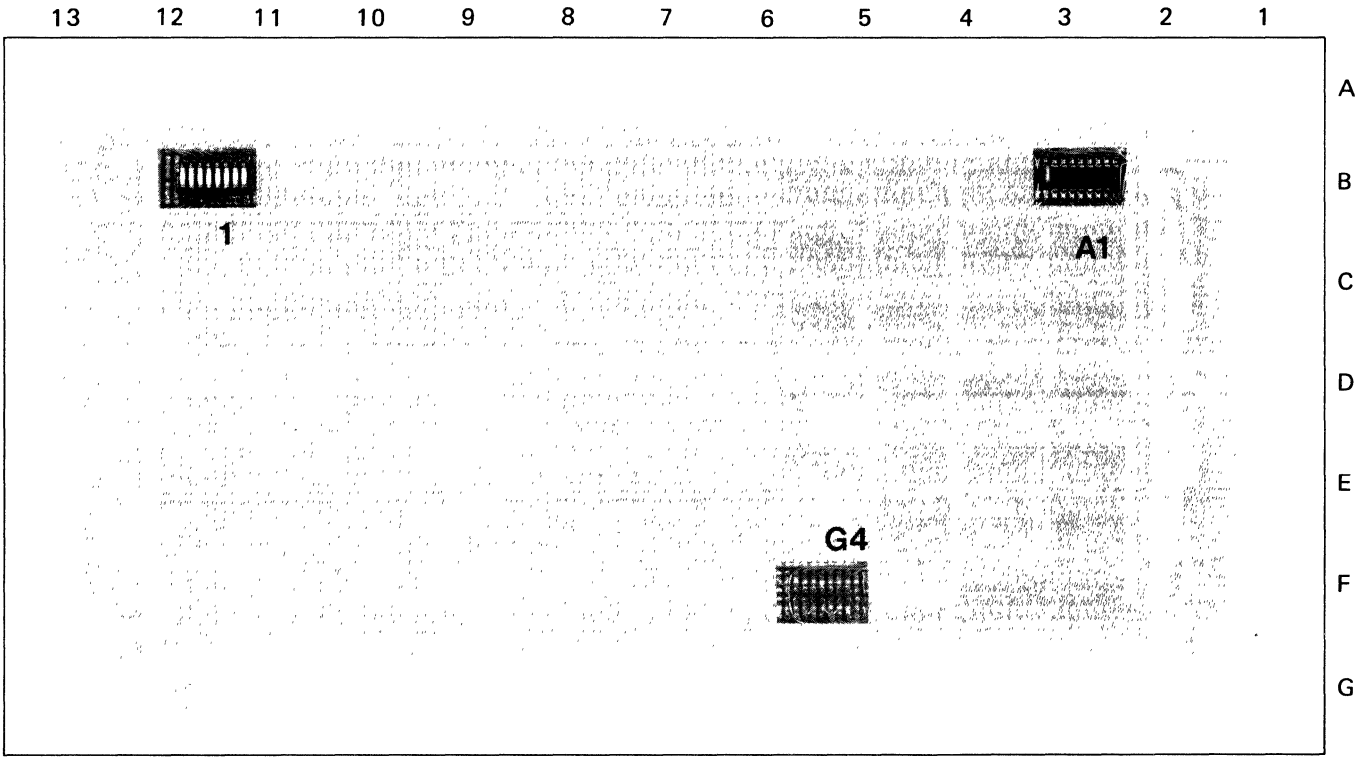


Busstick	Jumper Pins to GND
0	—
1	9 & 10

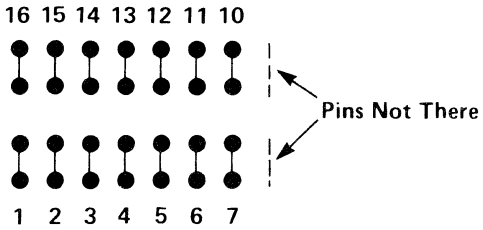
Typical Example: PAR ^{G4}PAR = Feedback ^{G4}AII/IO/XMR = IO ^{A1}Busstick = 0

PAR

2101529G08

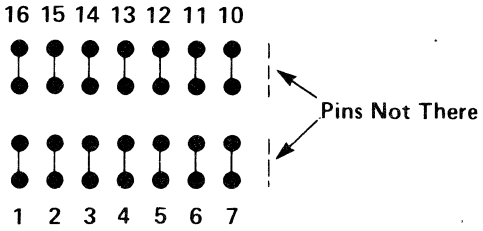


G4 Address Recognition



From	G4-5	G4-6	
	To	To	
I/0 Only	G4-12	G4-11	(Psats) I/0 Buss Separate From Mem Bus
All	—	G4-10	
All Exept Memory Read	G4-7	G4-10	(M/S) Mem and I/0 On Same Bus

G4 Parity Tree Type (B1, B3, E2, F2)

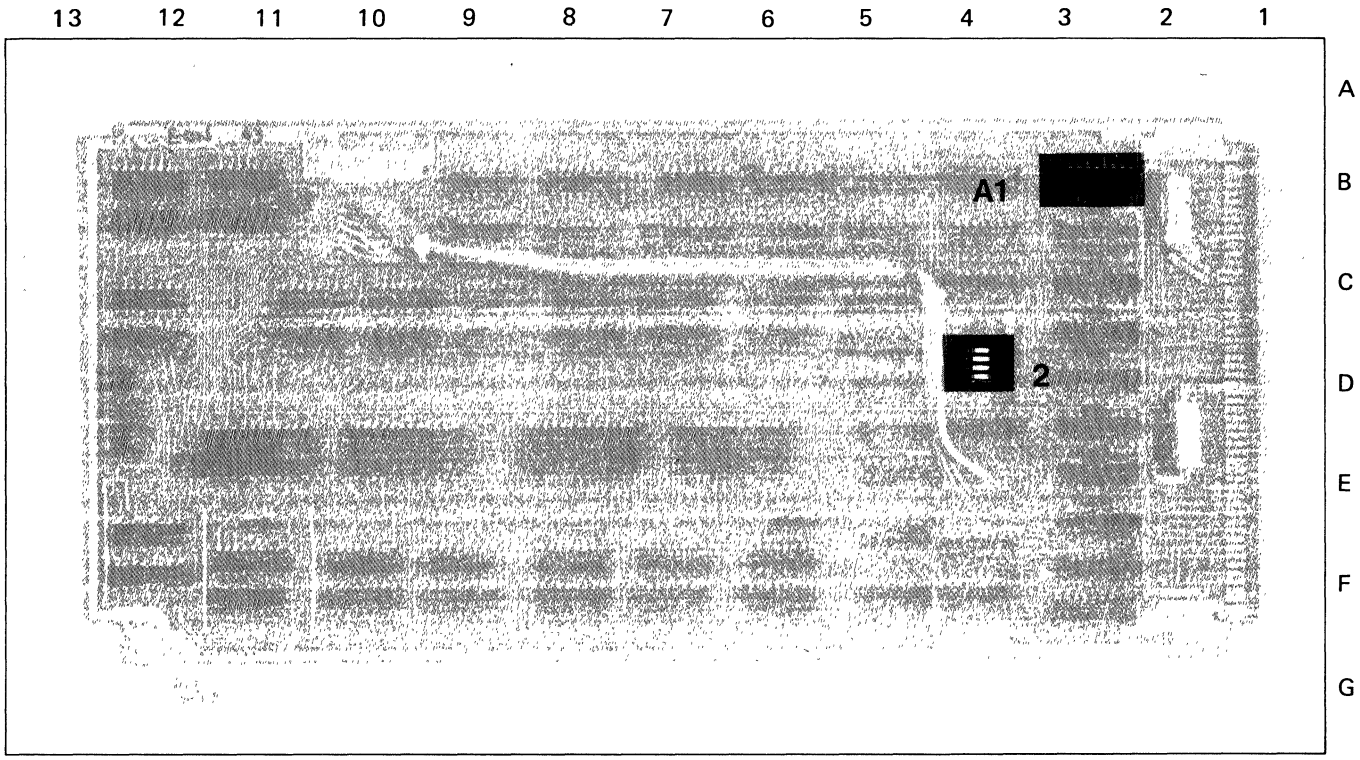


From	G4-3	G4-14	
Type	To	To	
74180	G4-2	G4-15	Factory Set
74S280	G4-4	G4-13	

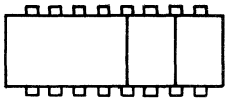
Typical Example: PAR ^{G4}PAR = Feedback ^{G4}All/IO/XMR = IO ^{A1}Busstick = 0

PID

2100131G01



A1 F-Stick



	Jumper to GND
PLI	9 and 10
VDA	9 and 10
Other	No Jumpers

2

S4		Bit 11
S3		Bit 12
S2		Not Used
S1		

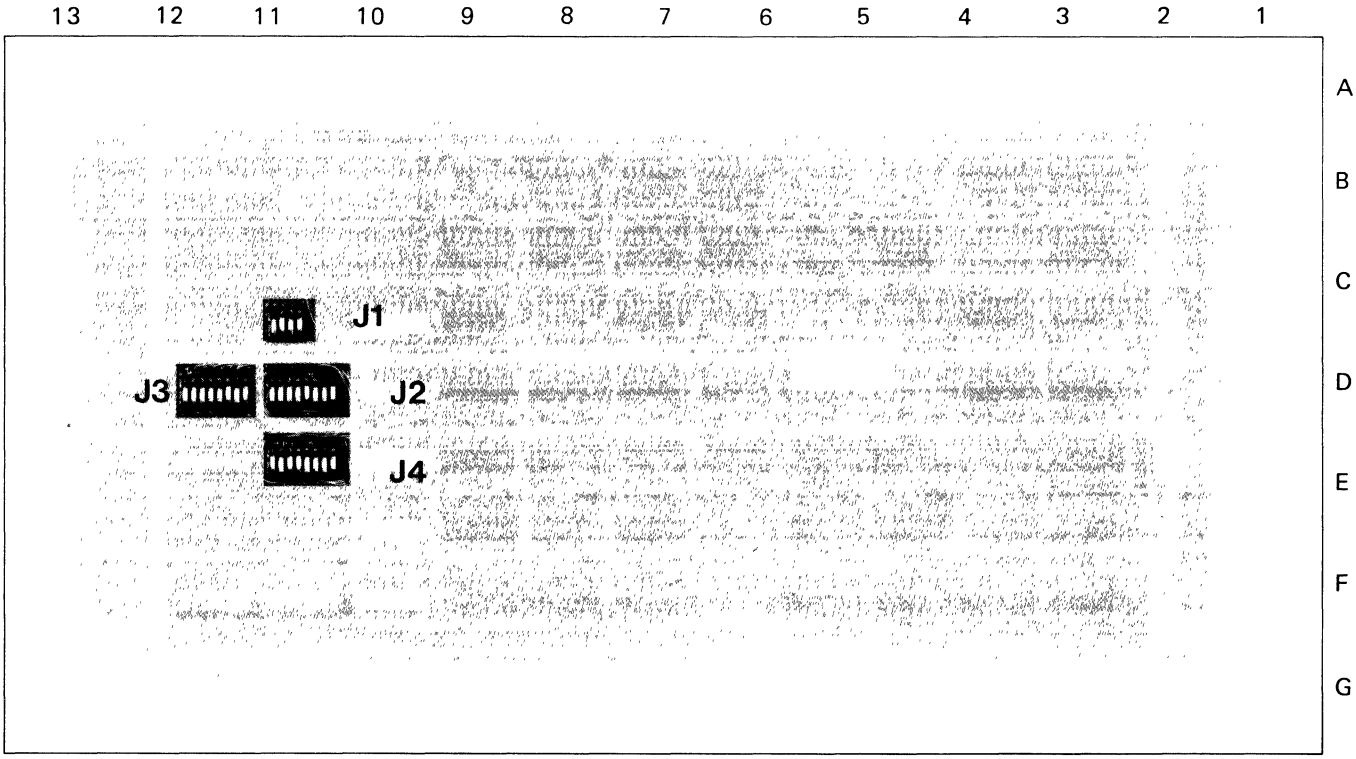
	Bit 12	Bit 11
PID Address	S3	S4
E0	Off	Off
E8	Off	On
F0	On	Off
F8	On	On

	S1
Single-Bus	On
Multiple Bus (PLI-VDA)	Off

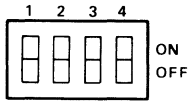
Typical Example: PID ^{2}E000 ^{2}S/M = Multiple

RTC

2101529G10



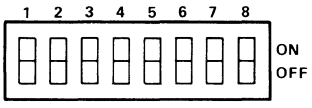
J1 Pid Address



	Pid Address		Clock Address		
	SW1	SW2	SW3	SW4	Result Address
					E0
		X		X	E8
	X		X		F0
	X	X	X	X	F8
Bit	12	11	12	11	

X - On

J2 Pid Fast



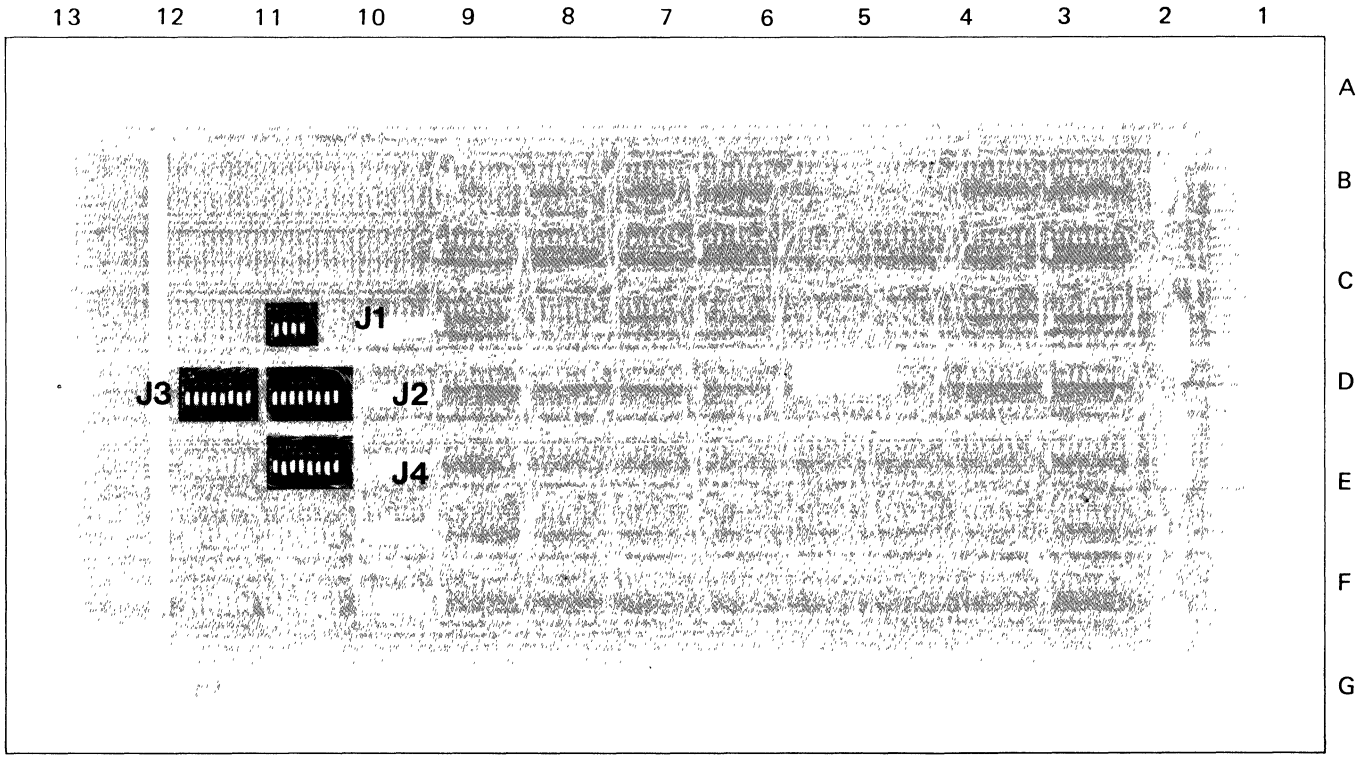
	MSB				LSB				
Switch	1	2	3	4	5	6	7	8	Result Address
	X	X	X	X	X	O	O	N	F8

O = Off
X = On
N = Not Used

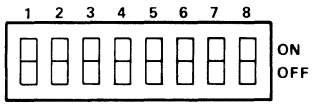
Typical Example: RTC ^{J1}E006 ^{J1}Pid Add = E000 ^{J2}PidFast = F8 ^{J3}PidSlow = FA ^{J4}SWs = 0,2,0

RTC

2101529G10



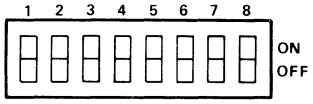
J3 Pid Slow



	MSB				LSB				
Switch	1	2	3	4	5	6	7	8	Result Address
	X	X	X	X	X	O	X	N	FA

O = Off
X = On
N = Not Used

J4 IMP Number



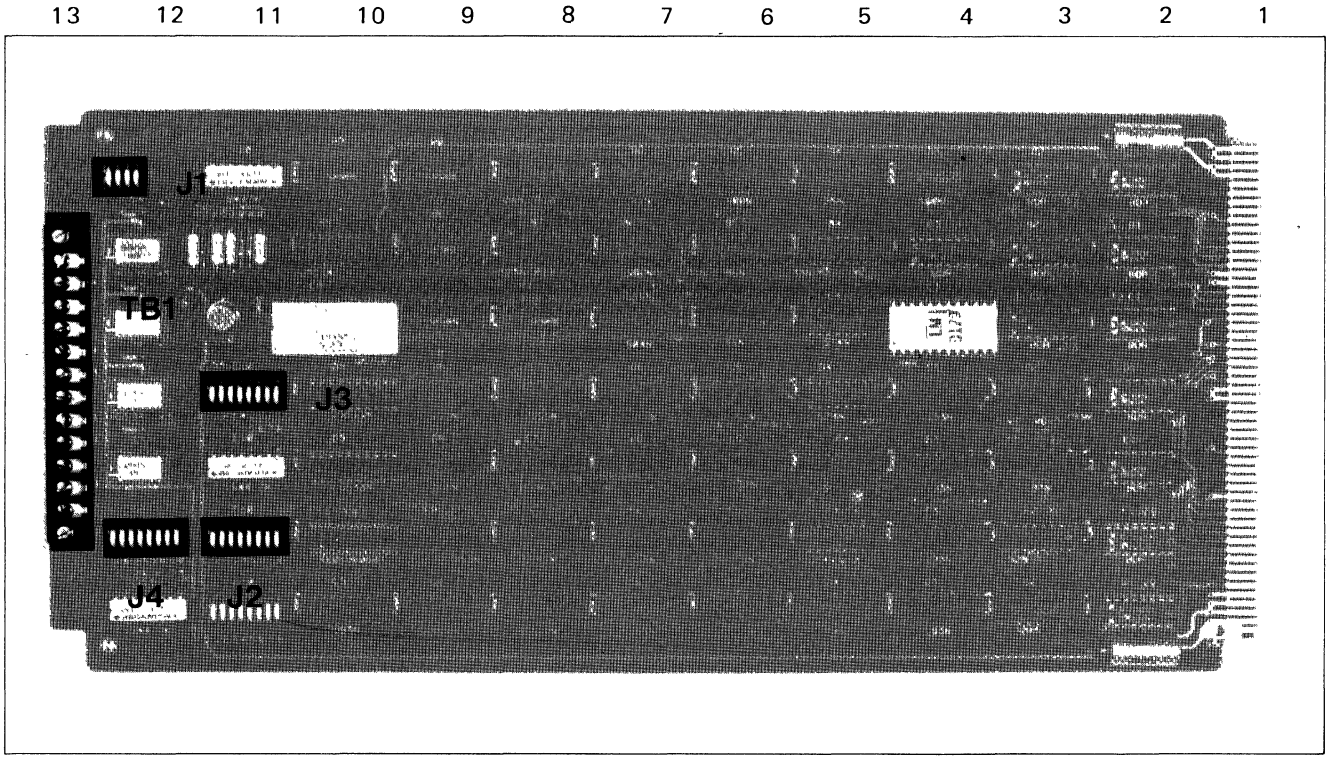
	MSB				LSB				
Switch	1	2	3	4	5	6	7	8	Result
	O	O	O	O	O	O	X	O	0, 2

O = Off
X = On
N = Not Used

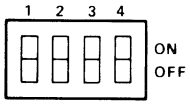
Typical Example: RTC [J1] E006 [J1] Pid Add = E000 [J2] PidFast = F8 [J3] PidSlow = FA [J4] SWs = 0,2,0

RTT

2105513G01



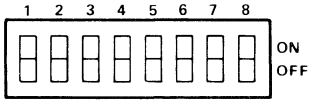
J1 Pid Address



	Pid Address		Clock Address		
	SW1	SW2	SW3	SW4	Result Address
					E0
		X		X	E8
	X		X		F0
	X	X	X	X	F8
Bit	12	11	12	11	

X - On

J2 Pid Fast



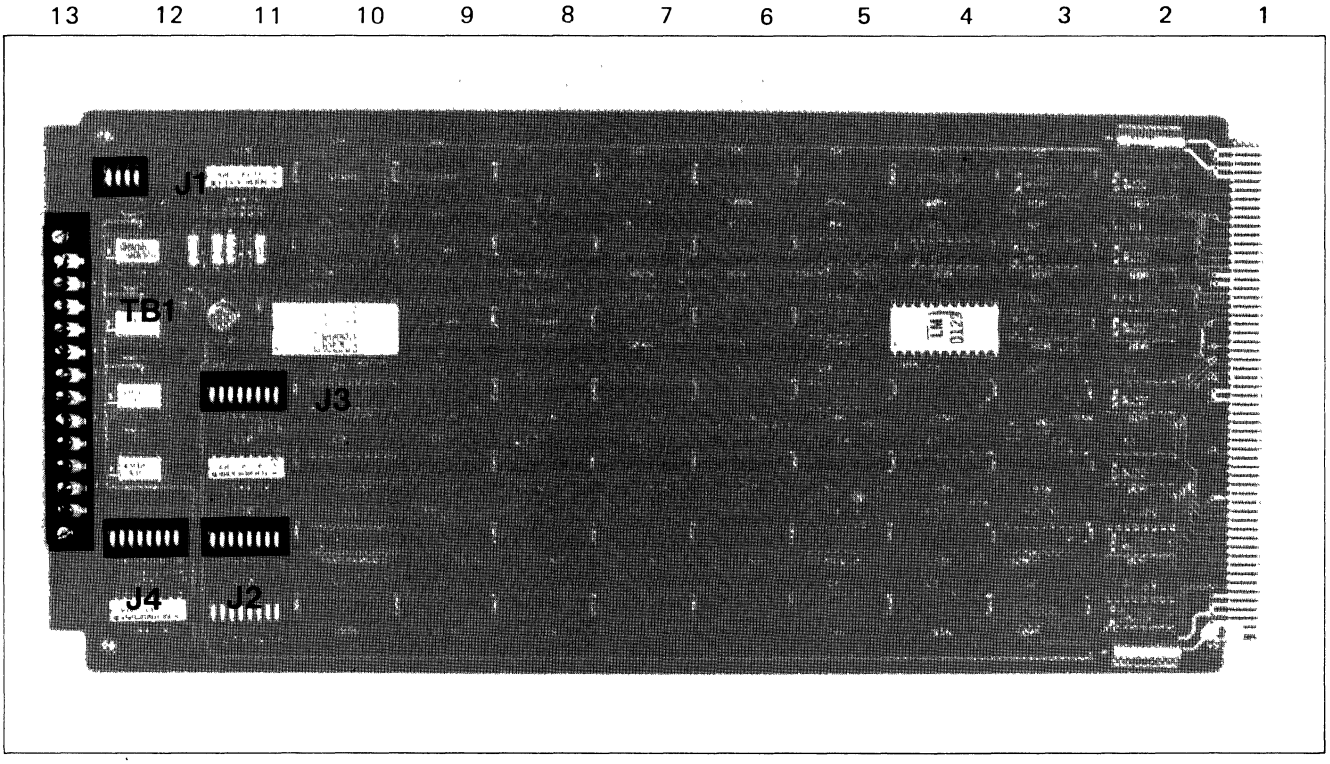
	MSB				LSB				
Switch	1	2	3	4	5	6	7	8	Result Address
	X	X	X	X	X	O	O	N	F8

O = Off
X = On
N = Not Used

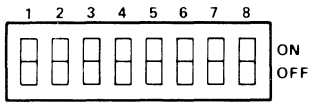
Typical Example: RTC ^{J1}E006 ^{J1}Pid Add = E000 ^{J2}PidFast = F8 ^{J3}PidSlow = FA ^{J4}SWs = 0,2,0

RTT

2105513G01



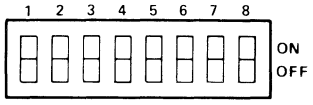
J3 Pid Slow



	MSB				LSB				
Switch	1	2	3	4	5	6	7	8	Result Address
	X	X	X	X	X	O	X	N	FA

O = Off
X = On
N = Not Used

J4 IMP Number



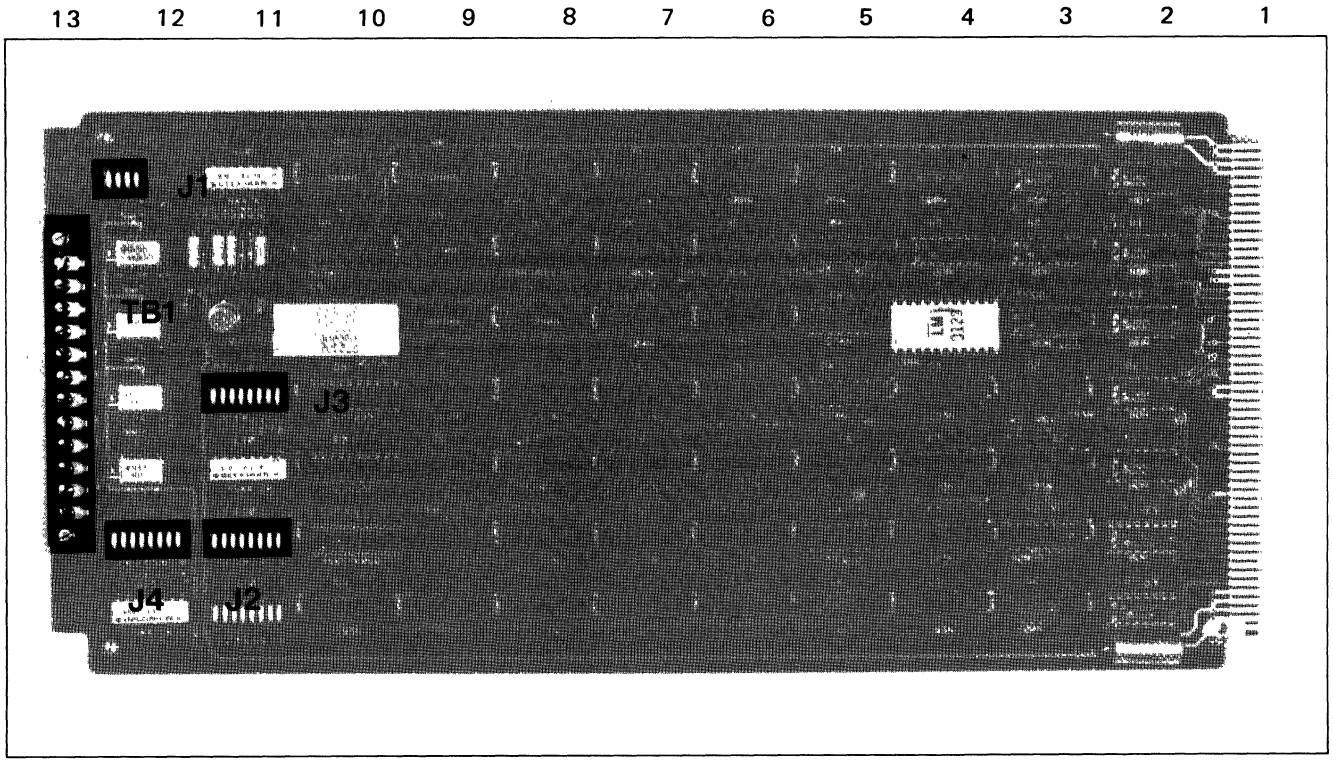
	MSB				LSB				
Switch	1	2	3	4	5	6	7	8	Result
	O	O	O	O	O	O	X	O	0, 2

O = Off
X = On
N = Not Used

Typical Example: RTC ^{J1}E006 ^{J1}Pid Add = E000 ^{J2}PidFast = F8 ^{J3}PidSlow = FA ^{J4}SWs = 0,2,0

RTT

2105513G01



TB1