Reviewed 12/2
An update
exists

CHAPTER 1

CONSOLE HARDWARE OVERVIEW

1.1 INTRODUCTION

The system console is the point of interface between the operator, field engineer, or development engineer (during debug) and the JUPITER computer system. Its function in life is to support the diagnostic philosophy, boot the system, and monitor the system environment of the JUPITER System.

The console hardware consists of a DEC PDP-11 compatible instruction set microprocessor, 64k words of RAM, 6k words of PROM, six serial asynchronous I/O devices, a time of year clock (with battery back-up) that includes an interval timmer, system clock control hooks (system clock start, stop, or half-step), diagnostic paths, some mass storage device interface path, environmental monitoring path for monitoring both the power sub-system and the thermal environment, and 256 words of non-volatile CMOS RAM (with on board battery backup).

The Console also provides a path to the JUPITER SYSTEM specific logic housed on the I/O Box module. The specific functions available are described in the I/O-Box subsystem specification but are described, briefly, here for ease of understanding. The I/O-Box houses a TTL-Bus Monitor, Diagnostic Path Logic, and the Master Oscillator Control Logic.

The following diagram (FIGURE 1) is a block diagram which shows the Console interconnections to the JUPITER System.


DIAGNOSTIC PATH
THRU CPU/MEMORY
M-BOX
MBUS (ECL)
I/O-BOX
I/O-BUS (TTL)
CONSOLE
PORT
ADAPTER
CONSOLE LOAD DEVICE
ENVIRONMENTAL MONITOR
REMOTE DIAGNOSIS LINK
CONSOLE TERMINAL
LOCAL TERMINALS
CONTROL PANEL

FIGURE 1 JUPITER SYSTEM CONSOLE BLOCK DIAGRAM

1.2 GOALS

The console subsystem goals are:

1. Meet JUPITER SYSTEM RAMP goals.
2. Support Remote Diagnosis.
3. ~~Allow for the implementation of a true, operator-less system.~~
4. Provide On-Line diagnosis capability.
5. Provide a console load device consistent with the system RAMP requirements and philosophy.

The following sections describe the specifications that must be met for the console subsystem to meet its goals.

1.3 POWER REQUIREMENTS

The console subsystem (the L0063 module) requires the following power:

+5 volts @ 15 amp. (or less)
+12 volts @ 1 amp. (or less)
-12 volts @ 1 amp. (or less)

The console subsystem load device (Mass Storage Unit) requires the following power:

BA11-NE:
115 Vac @ 12 amp. (max)

BA11-NF:
230 Vac @ 6 amp. (max)

RL02:

RLV11 (RL02 CONTROLLER - M8013 & M8014):
+5 VOLTS @ 6.5 amp.
+12 volts @ 1 amp.

1.4 ENVIRONMENTAL REQUIREMENTS

The console subsystem is designed as an integral part of the JUPITER System, and as such, is intended to operate in a DEC STD Class A environment. The console relies on the system packaging to assure that the applicable RFI/EMI standards are assured.

1.5 OPTIONS

There are NO Options available for the Console subsystem . The cables and terminal devices attached to the console communications ports are not part of the console subsystem.

1.6 PRE-REQUISITES

The following items are required for proper console subsystem operation:

* VT100 or equivalent video display terminal
* LA38 or equivalent hard-copy terminal
* Sufficient power and cooling
* Two (2) RL02 Disk Drives

Why are they required? What if they break?

1.7 NEGATIVE SPECIFICATIONS

The following items are not part of the console subsystem goals/design criteria:

1. Software compatibility with existing PDP-11 based systems.

2. Compatibility with existing, off the shelf, pre-built operating systems.

   Both of these items are precluded because of the optimization of hardware based addressing for the communications and other I/O devices used to achieve a single module console subsystem.

3. Use of "QBUS" devices on the console RLB Interface is restricted to an address range of 774400-774576, a single interrupt vector address of 160 (octal), and a level four interrupt.

1.8 CONSOLE RAMP

The following data lists the RAMP goals for the JUPITER console subsystem. These goals reflect the overall system RAMP goals.

1. Error detection and Error Logging
2. Remote Diagnostic
3. I/O-Bus monitor support (I/O Box resident transaction monitor with an historical ring buffer)
4. Console Memory Error Correction (soft error detection)
5. Console Memory Parity (classical PDP-11 parity)
6. Loopback Diagnostic capability for Console Communications lines
7. Temperature and Airflow sensing
8. Power Supply Monitoring
9. Module Isolation Diagnostics
10. Software support of the above listed features
11. Error Recovery

1.9 IMPLEMENTATION

The console subsystem RAMP goals are intimately tied to the overall JUPITER System RAMP goals and RAS structure. In fact, the console is the focal point for implementing the various RAMP features that must be included in order to meet both customer and Digital Equipment Corporation expectations.

1.9.1 CPU Error Detection And Error Logging

The internal structure of the JUPITER CPU is a pipelined architecture. This structure lends itself to the detection of single bit errors through the use of odd parity on each nine bit portion of the data path. The partitioning of the data path is such that input has a parity check performed before an operation is accomplished. This method of partitioning allows the pipe to be stopped and the error occurrence to be recorded. Additionally, this allows the input to be properly recovered and

retried.

The various logic functions within the JUPITER CPU are tasked with informing the console subsystem of the occurrence of a parity error. A soft error, i.e. a transient parity error - one that does not occur every time data is fed into a function, can be corrected through the recovery of a correct copy of the erroneous input data. There are various intermediate level copies of Accumulator (AC) data including a Master AC copy, that are available for just this purpose.

The clock input for the failing logic function (perhaps the entire CPU - with the exception of the M-Box) is stopped in an orderly and proper fashion upon detection of a parity error. The console subsystem is tasked with restarting the clock in an orderly and proper manner.

The fact that an error has occurred is logged by the console subsystem, stored in the console load device, and, upon proper restart of the clocks, the console informs the CPU that there is error information that the operating system must forward to the system error log. This provides a redundant log of system error conditions but this is extremely beneficial, as will be shown in the next section.

1.9.2 JUPITER System Remote Diagnosis

The console subsystem provides an asynchronous communications channel that allows the use of remote system diagnosis. This communications channel allows Field Support to call the JUPITER system and access the console. The remote diagnosis channel can be used to run device diagnostic programs that will determine if a module is in need of replacement. Through the use of this diagnostic tool, the remote diagnostic path, there will be a higher than previously experienced degree of certainty that the modules the Field Support Engineer brings to the ailing machine are the proper modules.

1.9.3 JUPITER I/O-Bus Monitor

The I/O Bus Monitor resides on the I/O Box module and records the TTL BUS Data and transactions in a Ramdom Access Memory. The Console can access this recorded data via the console I/O Box interface. The I/O Bus Monitor appears to the Console as registers in its I/O address space. For a detailed description of the I/O Bus Monitor refer to the I/O Box Engineering Specification.

1.9.4 Console Memory Parity

The console subsystem local memory has memory parity implemented. This allows the detection of any single bit error. Memory transfers automaticly generate and check parity - odd parity of course. This error detection scheme coupled with the soft error correction method , provide a console memory that is capable of single bit error detection and correction.

1.9.5 Console Memory Error Correction

The console subsystem local memory has an error correction mechanism built into the console resident software. This software algorithm consists of a series of check words that when coupled with a CRC type routine provide the isolation to the single bit error level. The method of backing up this error correction scheme is the console reboot from the console load device. The console reboot sequence does NOT require JUPITER CPU intervention to be accomplished. This means that the console can recover from soft errors with out impacting the system performance. It is possible for the console to reload a bad block of memory (perhaps 256 word block) with out reloading the whole console memory.

1.9.6 Loopback Capability

All console devices have loopback capability. The disk interface, the communications channels, and the I/O-Box interface can all be looped back on themselves to turn typical Write Only or Read Only paths into Read/Write paths. This Loopback capability is extended in the I/O-Box to include the TTL-Bus, the Ports (under RAM loadable control) and the M-Box path.

1.9.7 Environmental Monitoring

The console subsystem can monitor the physical environment to determine the presence of over temperature or out of voltage specification conditions. The enviromental monitoring is accomplished by the Enviromental Monitoring modules. The JUPITER System uses two EMMs (one in I/O cabinet and one in CPU cabinet) and interface to the the Console module via a single RS422 type communication line.

1.9.8 Diagnostics

The JUPITER System diagnostics provide true module failure isolation. The basic architecture of the CPU is such that fault conditions can be detected and with some moderate support from the console subsystem, the failing unit can be determined.

All of the features described in 2.1 are supported by the console monitor software supplied by the diagnostic engineering group. This software is resident on the console load device.

CONTROL

F11 MICRO (DAT/CTL) 2.2

MIB 15:0 micro instruction 24 bus

MMU 2.3

DAL 17:16

BUFF (TRI-ST) PWR-ON ADR 2.6

BUFF (TRI-ST) SER-VICE 2.7

BUFF (TRI-ST) INT-VECT 2.8

REG CPI console panel interface

BUFF CPI

KALV LED, KLINIK LED

UMD SW, CTYMD SW, LOCK SW, STS START SW, CSL START SW

BUFFER

CSL PWR LED

+5V

+3V

CONSOLE PANEL INTERFACE

2.9

DAL 17:16

REG (TRI-ST) ADR 17:0

OE

DMA (0)

DAL 15:0 2.5 data address line

ADR 17:0 2.10

REF ADR

MIX

MA 7:0

RAM BPAR OUT

RAM BPAR IN

RAM 64KX16 2.11

PROM 6K X 16 [4K MEM 2K I/O] 2.11

NV RAM 256X16 2.11

ADDRESS

BUFF

BUFF

TRNVER (TRI-ST)

REG (TRI-ST) 2.13.1 MTER0 Maint. Error Reg.

REG (TRI-ST) 2.13.2 MTER1

LATCH (TRI-ST) DMA ADR

RLB ADR 17:16

DMA (1)

REG (TRI-ST) DIAG CR 2.13.3

BUFF (TRI-ST) DAL

BUFF (TRI-ST) DATA

DATA 15:0

TOY 2.17

TRNVER (TRI-ST) SLU 2.16

SD 7:0

2.16

BAUD RATE GENS

B60
TX B61
RX B61
TX B62
RX B62
B63
B64
B65

2X USART 280-SIO

EMM (RS422)

CTY (RS232)

KLINIK (RS232)

VT (RS232)

DN+ (RS232)

INTERCONSOLE (RS422/3)

REG (TRI-ST) 2.14.2 DIAG SCAN PORT SEL

BUFF OR

2.14.1 PS 7:0 (TO PORTS)

PDOUT 7:0 (FR PORT

SDR IN (TO I/O BO

RLB ADR 17:16

TRNVER 2908 2.15 RLB DMA - memory + RLΦ2

RLB INTERFACE

to RLΦ2 (QBUS protocol)

TRNVER 2908 IOB 2.18

PAR CK/GEN

XOR BYTE PAR CK/GEN

RAM BPAR OUT

RAM BPAR IN

I/O BOX INTERFACE

to ports

REVISIONS		
CHK	CHANGE NO.	REV

FIGURE 2

digital

DRN. T. Toomey
CHK'D.

DATE 07-NOV-80
DATE

ENG.
BOARD LOCATION:
SHEET OF
NEXT HIGHER ASSEMBLY:

DATE

TITLE: JUPITER Console BLOCK DIAGRAM

SIZE D
CODE CS
NUMBER

VULCAN:<2888.L0863>CONBLK.DRW 07-NOV-80 11:55
FIRST USED ON OPTION/MODEL:

CHAPTER 2

SYSTEM CONSOLE DESCRIPTION - HARDWARE

2.1 INTRODUCTION

A block diagram of the JUPITER Console is shown in Figure 2. The JUPITER console subsystem is based on the Digital Equipment Corporation F-11 chip set. This chip set microprocessor implements the full PDP-11/34 instruction set. As implemented, the F-11 has a maximum addressing capability of 124K words of memory and 4K words of I/O space. This addressing capability is implemented as 64K words of RAM, 4K words of PROM, and 4K words of PROM-I/O. The subsystem includes a DMA disk type of mass storage unit (RLB) interface, six programmable asynchronous communications channels, a time of year clock with an interval timer feature (on module battery backup),non-volatile RAM (on module battery backup), a DEC MMU - memory management unit, a parallel I/O channel to the I/O Box, a console panel interface, and Maintenance/Diagnostic registers.

how is it limited?

This subsystem can be viewed as a simple, bounded PDP-11 computer system. The console subsystem executes the PDP-11 instruction set, handles interrupts, addresses I/O in the classical PDP-11 manner, and has the typical PDP-11 programmable/ jumperable features hardwired. Interrupt vectors and priorities are hard wired in this subsystem because there are no variations to the standard console subsystem.

The heart of the console processor is contained on three MOS/LSI chips. They are the data chip, the control chip and the memory management unit (MMU). The data and control chips are combined in a single 40-pin package. The MMU is packaged as one 40-pin chip. Rerference APPENDIX B for a detailed functionality description of the F-11 chip set.

2.2 F-11 MICRO DESCRIPTION

The F-11 MICRO (Dat/Ctl) chip recognizes and executes the PDP-11/34 instruction set, supports four priority vectored interrupt levels, DMA, addresses up to 28K of memory, and supports the classical PDP-11 I/O page.

2.2.1 Data Chip

The data chip contains the PDP-11 general registers, the processor status word (PS), several working registers, the arithmetic and logic unit (ALU), and conditional branching logic. The data chip does the following.

1. Performs all arithmetic and locical functions.
2. Handles all data and address transfers with the DATA bus with the exception of relocation, which is handled by the MMU.
3. Generates most of the signals used for interchip communication and external system control.

2.2.2 Control Chip

The control chip contains the microprogram sequence logic and 552 words of microprogram storage in programmmable logic arrays (PLA) and read-only memory (ROM) arrays.

During the course of a normal microinstruction cycle, the control chip accesses the appropriate microinstruction in the PLA or ROM, sends it along the MIB to the data and MMU chips for execution, and then generates the address for the next microinstruction to be accessed.

2.3 MMU DESCRIPTION

The MMU chip provides the memory management function and storage for status registers and the FP11 floating point accumulators (not used in JUPITER console). This chip provides dual mode (user and Kernel) address relocation of 18 bits. Sixteen-bit virtual addresses are received from the data chip via the data address lines (DAL), relocated to the appropriate 18-bit physical address, and the sent on the DAL to replace the original virtual address. The MMU chip contains the status registers and

active page registers (PAR/PDR register pairs) as well as access protection and error detection capability. Refer to for progaming information.

The MMU or Memory Management Unit chip expands the addressing space in the same manner as the PDP-11/34 MMU but the addressing is limitted to the on board memory: 64K words of RAM, 4K words of PROM in the memory address area; 2K words of PROM in the I/O address area, and the I/O device addresses implemented in the upper 2K words of the I/O area.

2.4 MICROINSTRUCTION BUS (MIB)

The 16-bit time-multiplexed, microinstruction bus is common to the data and control chip. A subset of the MIB is routed to the MMU because it does not need access to all MIB control signals. A different subset of the MIB controls the console board logic.

2.5 DATA ADDRESS LINES (DAL)

The DAL is a time-multiplexed bus which transfers data from the data chip to the other MOS chips or between the console panel interface or DATA bus and the MOS chips. The DAL also transfers service information, power-up mode/bootstrap address and interrupt vector address to the MOS chips.

DAL 17 and DAL 16 are only genterated by the MMU chip during address relocation.

2.6 POWER-ON ADDRESS (PWR-ON ADR)

The PWR-ON ADR buffer contains option data which is used during power up and in the emulation of the PDP-11 HALT macroinstruction. The PWR-ON ADR buffer generates the power-up mode, halt/trap option and bootstrap address which is hardwired . The gating of this data on the DAL is controlled by MIB 3 from the F-11 MICRO. A power-on mode 2 is used. This mode causes the F-11 MICRO to internally generate a bootstrap address by looking at the PWR-ON ADR buffer (DAL 15:8 = 160000). This address is loaded into the PC. The processor sets the PS to 340 (octal, PS<07:05> = 7) to inhibit interrupts before the processor is ready for them. The processor begins execution by fetching an instruction from the location pointed to by the PC. The halt/trap option is wired to trap to 10 (octal) on a HALT instruction. The following conditions will cause the F-11 MICRO

to read the PWR-ON ADR buffer.

1. Power-up sequence for the console which is initiated by the negation of the DCLO signal from the Envioronmental Monitoring module (EMM). The normal power-up sequence is initiated by the power-on condition of the console panel switch/es.

2. Console start which is initiated by depressing the console start switch on the console panel.

3. System start which is initiated by depressing the system start switch on the console panel.

4. The electronic finger which is generated by the JUPITER processor via the I/O Box interface.

The format of the PWR-ON ADR buffer is as follows:

15	14	13	12	11	10	09	08
1	1	1	0	0	0	0	0

07	06	05	04	03	02	01	00
POK					1	1	0

bits <1:0> - indicate one of four power up options.

00: power up to vector 24
01: power up to micro-ODT
10: power up to a user boot macrocode
11: power up to user microcode

bit <2> - is halt option, it determines the action taken when a HALT is encountered in Kernel mode.

0: go to micro-ODT on HALT
1: trap to 10 (octal) on HALT

bits <6:3> - unused, read as don't cares

bit <7> - indicates the condition of the power level (ACLO)

0: power level bad
1: power level okay

Does this mean when we reload the console, it will so through all its tests? I'd like a mode where we just resynch. There's a KCCON command

bit <8> - is used to specify the user macro boot address.

0: get boot address of bits <15:9>
1: use 173000 as the macro boot address

bits <15:9> - upper 7 bits of the user macro boot address. bits <8:0> are set to 0 by the F-11 MICRO.

2.7 SERVICE

The SERVICE buffer places service information on the DAL every F-11 MICRO clock low time with the exception of an address relocation by the MMU chip. The F-11 MICRO (control chip) uses this service information to determine whether to interrupt or fetch the next instruction. The format of the SERVICE buffer is as follows in the order of priority in which they are handled, (highest to lowest):

15	14	13	12	11	10	09	08
0	0	0	Event	INTRQ4:	INTRQ5:	INTRQ6:	INTRQ7:

07	06	05	04	03	02	01	00
PWRF	1	0	CTLERR:	MMUERR:	PARERR:	BUSERR:	DCOK

DAL Input	Service Definition		Active State
<0>	*DCOK:	Power-up signal	1
<4>	*CTLERR:	No active Control chip	1
<3>	*MMUERR:	Memory Management abort	0
<1>	*BUSERR:	System Bus timeout error	1
<2>	*PARERR:	PARITY ERROR	1
<6>	*SPARE:	Causes an Unconditional Jump to Control chip 3 locoation 76	0
<7>	PWRF:	Power-fail signal	1
<8>	INTRQ7:	Priority Level 7 interrupt request	1
<12>	EVENT:	Event line interrupt request (priority level 6)	1
<9>	INTRQ6:	Priority Level 6	1

		interrupt request	
<10>	INTRQ5:	Priority Level 5	1
		interrupt request	
<11>	INTRQ4:	Priority Level 4	1
		interrupt request	
<5>	HALT:	Halt request	1

* Asynchronous abort-causing service inputs, i.e. these conditions generate RESET (section A.2.5).

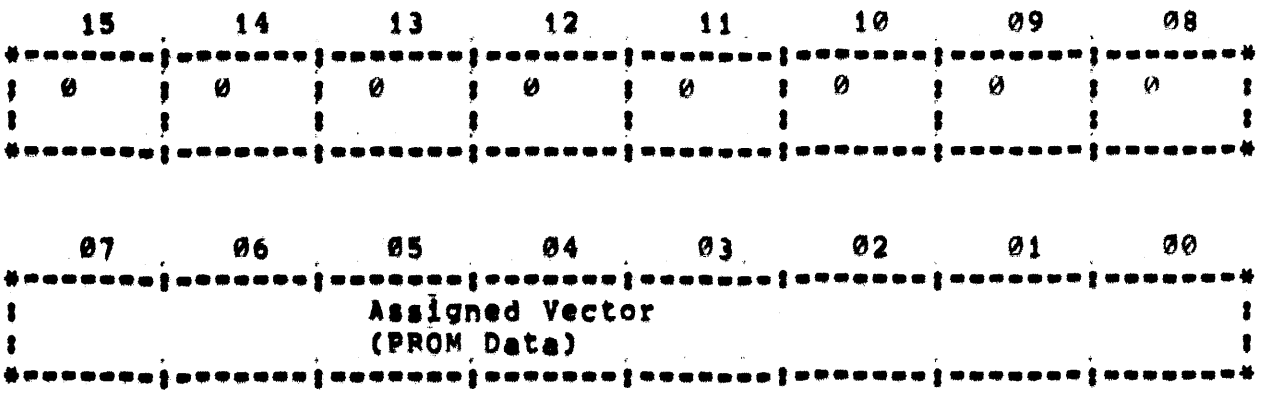
2.8 INTERRUPT VECTOR (INT VECT)

Interrupt requests are initiated from the console internal devices, I/O Box and RL02 Disk Drive controller and passed to the F11-MICRO for processing.

------NOTICE------

With the exception of the RL02 Disk Drive, all console interrupts must be program dismissed (i.e. The condition causing the interrupt must be cleared by software.)

The INT VECT buffer generates the interrupt vector address during the interrupt acknowledge sequence from the F-11 MICRO. If an interrupt is being serviced, the F-11 MICRO asserts MIB 13 (IAK) which is used to control the gating of the INT VECT buffer onto the DAL. All interrupt vector addresses (including the RL02) are hardwired/PROM generated. The format of the INT VECT buffer is as follows.



bits <15:8> - Interrupt Vector (INT VECT)

These bits are hardwired to zero to generate an address range of 000377-000000.

bits <7:0> - Interrupt Vector (INT VECT)

These bits are programed in PROM and defined as follows:

bits <7:0>Octal	Assigned Device
160	RL02
270	All unasigned PROM locations
300	EMM and CTY (SIO 0)
310	KLINIK and VT (SIO 1)
320	DN and Interconsole (SIO 2)
340	IOB ERR (I/O Box)
344	IOB ATN (I/O Box)
350	TSD Flag (TOY Shift Done)

2.9 CONSOLE PANEL INTERFACE

The Console Panel Interface consists of a console panel, an addressable register and interface lines between the console panel and console module.

2.9.1 Console Panel

The JUPITER system Console Panel consists of five switches and three LED indicators. The control panel is used to select which of the local terminals is the CTY, enable KLINIK, control system On/Off, and provide control for starting the console or system.
The following switches are resident on the console panel:

1. POWER ON - This is the master power switch that is used to enable all DC power on or to turn off all DC power. This switch is used in conjunction with position 1 and postion 2 of the CONTROL switch.

   ON Postion: all DC power is enabled, Battery Backup is enabled - provided that the ON position was achieved while the CONTROL switch was in position 2 through 4 - and +5 volts, +12 volts, -12 volts are on to the console module and the enviormental monitoring module. The BA-11N also has power applied. The console software commands the EMM to turn on all other DC power - MBOX, EBOX, IBOX, FPA, I/O BOX, and PORT Logic.

   OFF Position: All DC power is disabled, Battery Back Up is disable. All DC power is shut down regardless of the position of any other Console Panel switch.

2. CONTROL - This is a four (4) position key switch.

   Position 1: Power Disable - This switch position is used in conjunction with the POWER ON switch. When in this position, POWER ON can be used to turn OFF system DC power. System DC power can not be restored until the control switch is changed to positions 2 through 4.

   Position 2: Power Enable - This switch position is used in conjunction with the master POWER ON switch. The CONTROL switch must be in positions 2 through 4 in order for the POWER switch to enable system DC power.

   Position 3: KLINIK User - This switch position is used to inform the console software that those features associated with a KLINIK link in USER mode can be enabled.

   Position 4: KLINIK CTY - This switch position is used to inform the console software that those features associated with a KLINIK link in CTY mode can be enabled.

3. CONSOLE LOCK - This is a two (2) position key switch. When in the lock "ON" position, the SYSTEM START and CONSOLE START switches are not functional. The console software reads the state of this switch to envoke software features associated with the CONSOLE LOCK switch.

4. SYSTEM START - This is a monemtary contact switch that is used to initiate a "cold" system start. The CONSOLE LOCK switch must be in the OFF position for this switch to be functional. Activation of the SYSTEM START causes the console processor PC to be forced to 160000 - the first word location in the console I/O PROM.

5. CONSOLE START - This is a momemtary contact switch that is used to initiate a console start (not including a system start). The CONSOLE LOCK switch must be in the OFF position for this switch to be functional. Activation of the CONSOLE START causes the console processor PC to be forced to 160000 - the first word location in the console I/O PROM.

how are they different.

How are 4 & 5 related to power-on address) p. 2-3?

software decides what to do!

The following LED's are resident on the control panel:

1. TTL ON - This LED indicates that there is TTL power available to this LED and to the Console.

2. KLINIK - This LED is console software dependent and used to indicate that the KLINIK link is enabled/operational. — which one?

3. KEEP ALIVE -This LED is console software dependent and used to indicate reoccurring communication between the JUPITER software and console software.

2.9.2 Interface Lines

The interface lines between the console panel and console module are TTL level compatiable. Connections are made from the I/O backplane - console module slot - to the console panel circuit board. The following eight (8) interface signal lines transmit information between the console panel and console module.

1. CSL START L - This line is asserted low as long as the CONSOLE START switch is depressed.

2. SYS START L - This line is asserted low as long as the SYSTEM START switch is depressed.

3. LOCK L - This line is asserted low when the CONSOLE LOCK switch is in the OFF position.

4. KLINIK USER L - asserted high when the CONTROL switch is in position four (4).

5. KLINIK CTY L - asserted high when the CONTROL switch is in position three (3).

6. KALV LED L - asserted low by the console module to turn on the KEEP ALIVE LED.

7. KLINIK LED L - asserted low by the console module to turn on the KLINIK LED.

8. TTL ON L - asserted low by the console module to turn on the TTL ON LED.

2.9.3 Console Panel Interface Register

The console panel interface register is a 16-bit register addressable in the console I/O space. There are five (5) Light emitting Diodes (LED) on the console module which are used to encode a failing diagnostic test. These LEDs are driven by the console panel interface register. The format for this register is as follows.

Address: 771200

15	14	13	12	11	10	09	08
KALV LED	KLINIK LED		LED 4	LED 3	LED 2	LED 1	LED 0

07	06	05	04	03	02	01	00
	KLINIK USER	KLINIK CTY	LOCK	PWR START	SYS START	CSL START	SWLTCH CLR

bit <15> - KALV LED (write only bit)

0: Turn KEEP ALIVE LED off.
1: Turn KEEP ALIVE LED on.

bit <14> - KLINIK LED (write only)

0: KLINIK LED is OFF.
1: KLINIK LED is ON.

bit <13> - Not Used

bits <12:8> - LED <4:0>

These are the five LEDs residing on the console module. (write only bits)
0: Turn LED on.
1: Turn LED off.

bit <7> - Not Used

bit <6> - KLINIK USER (read only bit)

0: KLINIK USER mode is not enabled.
1: KLINIK USER mode is enabled.

bit <5> - KLINIK CTY (read only bit)

0: KLINIK CTY mode is not enabled.
1: KLINIK CTY mode is enabled.

bit <4> - LOCK (read only bit)

0: The console panel CONSOLE LOCK switch is in the OFF position.
1: The console panel CONSOLE LOCK switch is in the ON position.

bit ~~<3>~~ 3 - PWR START (read only bit)

0: DCLO from the EMM has not been asserted or SWLTCH CLR (bit <0>) has been generated.
1: A power on condition (DCLO assertion) has been encountered. No access to the NV RAM is allowed until this bit is clear.

bit ~~<2>~~ 2 - SYS START (read only bit)

0: A system "cold" start has not be initiated from the console panel or SWLTCH CLR (bit <0>) has been generated.
1: A system "cold" start has been initiated from the console panel.

bit <1> - CSL START (read only bit)

0: A console start has not been initiated from the console panel or SWLTCH CLR (bit <0>) has been generated.
1: A console start has been initiated from the console panel.

bit <0> - SWLTCH CLR (write only bit)

0: No function.
1: Writting this bit clears the PWR START, SYS START, and CSL START latches (bits <3:1>).

2.10 ADDRESS (ADR)

The address bus (ADR 17:0) is a Tri-State bus used to drive memory address lines. Memory, register, and device selection is also decoded from the address bus to control gating of the desired read or write data. The address bus is driven by the ADR 17:0 register with the exception of DMA transfers. In the case of a DMA transfer the ADR 17:0 register is Tri-Stated and the DMA ADR latches drive the address bus.

All addressing is implemented on word (16-bits) boundaries and byte instruction addressing supported.

The ADR 17:0 register is clocked for every data input or data output address cycle generated by the F-11 MICRO. It contains the physical address for immediately following or last, data input or data output cycle.

The DMA ADR latches (bits <17:0) receives address bits via the RLB Interface data lines and latches upon the receipt of SYNC from the RLB Interface control line. DMA addresses/transfers are only supported by the console RAM.

2.11 CONSOLE MEMORY

The memory mapping of the 2080 Console Subsystem provides for accessing the 70K Words of on board memory (68K Words in the Memory Page, and 2K Words in the I/O Page) partitioned as shown in the following physical address map: (Reference Appendix A, ADDRESS REFERECNE CHART)

PAGE	F-11 PHYSICAL ADDRESS (OCTAL)	FUNCTION
M	000000-377776 (word boundary)	RAM (addressable)
E	400000-737776 (word boundary)	NON-EXISTENT
M	740000-757776 (word boundary)	4K PROM
	* * * * * * * * * * * * * * * * * * * *	
I	760000-767776 (word boundary)	2K PROM
	770000-770776 (word boundary)	256 words NV RAM
O	771000-777776 (word boundary)	I/O Devices

Notice that the PROM memory overlaps the memory and I/O space so that the top 4K words of memory (4K ROM) are accessible only when the MMU is enabled. In this manner, the necessary power-up routines can always be accessed while the routines in the 4K PROM can be accessed only when the system MMU is enabled. Additionally, although the MMU provides the capabilities to address substantially more memory than is implemented on the console processor, there are no provisions for addressing memory that is not on-board. what does this imply?

The RAM (64K x 18 Dynamic Random Access Memory) includes memory byte parity. Byte parity is computed on data being written and checked on data being read from memory. The parity circuit is intended to assure data integrity in the RAM Memory. It is not intended to provide bus transaction parity between the F-11 CPU and I/O devices.

There is no parity implemented for the PROM (6K x 16 Read Only Memory) or the NV RAM (256 x 16 Non-Volatile Random Access Memory). The only parity implemented in the I/O space is on the I/O Box Interface.

As noted the specific address space assignment for the PROM in the memory space is 740000-757776; however, the hardware decoding does not look at address bits 16 and 15 which allows this PROM to recognize addresses in the range of 440000-757776. This allow the memory PROM addressing to be tolerable of address failures associtated with address bits 16 and 15.

2.12 DATA BUS

The DATA bus (DATA 15:0) is a Tri-State bus used to transfer:

1. data between the DAL (F-11 MICRO) and memory, Time of Year Clock, Maintenance registers, Serial Line Units, Diagnostic Scan Port Select register, RLB Interface, and IOB Interface.

2. address bits to the RLB Interface and IOB Interface from the DAL (F-11 MICRO).

3. data between the RAM and RLB Interface for DMA transfers.

4. address bits to the DMA ADR latches from the RLB Interface for DMA transfers.

The DATA bus is an exact copy of the DAL bus at all times with the exception of a F-11 MICRO data input cycle or a DMA transfer.

2.13 MAINTENANCE REGISTERS

2.13.1 MAINTENANCE ERROR REGISTER 0 (MTERR0)

With the exception of DMA transfers and MMU abort conditions, the MTERR0 register will contain the physical error address for the RAM or IOB Interface read parity error, or immeditate trap error condition. The MTERR0 does not capture the error address for DMA read transfer from RAM. In the case of an MMU abort the contains of the MTERR0 are indetermient.

The MTERRØ register is clocked to the address bus Data (ADR 15:Ø) on read cycles of the RAM or IOB Interface. It is also clocked to ADR 15:Ø when RESET signal (immediate trap condition) is generated. If a console bus parity error (CSLB PERR) is detected, clocking of this register on RAM and IOB Interface read cycles are prohibited until the CSLB PERR Flag is cleared by console software. The assertion of all RESETs clock the MTERRØ register. The following immediate trap conditions generate RESET (refer to 2.7 SERVICE).

1. Power up
2. Control error (CTL ERR)
3. Bus error (BUS ERR)
4. Parity error (PARERR)
5. MMU abort (MMUERR)

The format for the MTERRØ register which is READ ONLY is as follows.

Address: 771100

15	14	13	12	11	10	Ø9	Ø8
ERROR ADDRESS 15:Ø8							

Ø7	Ø6	Ø5	Ø4	Ø3	Ø2	Ø1	Ø0
ERROR ADDRESS Ø7:ØØ							

2.13.2 MAINTENANCE ERROR REGISTER 1 (MTERR1)

MTERR1 contains the two most significant bits (<17:16>) of the error address and control/status bits for console software and diagnostic test purposes. The format for the MTERR1 register is as follows.

Address: 771102

15	14	13	12	11	10	09	08
CSLB PERR	DMA RPER	0	RMPLB	RMPHB	WPDI XOR	BPERR CLR / RLB RDCOK	DMA PCLR / RLB RPOK

07	06	05	04	03	02	01	00
BPER TRP	IRESET	IINT EN	RLBLP	WE PLB	WE PHB	ERADR 17	ERADR 16

bit <15> - CSLB PERR - Console Bus Parity Error (read Only)

0: Power on initialization or writing bit <9> (BPERR CLR)
1: a parity erorr (low or high byte) has been detected on F-11 MICRO input cycle directed at the RAM or IOB Interface. Also prohibits further clocking of MTERR0.

bit <14> - DMA RPER - Direct Memory Access Read Parity Error (read only)

0: Power on initialization or writing bit <10> (DMA PCLR)
1: a parity error (low or high byte) has been detected on a DMA data transfer from RAM. This bit will not initiate an interrupt or trap. It's use is intened soley as a flag.

bit <13> - Not Used - read as zero.

bit <12> - RMPLB - Read Memory parity Low Byte (read only)

0: RAM parity bit for the low byte data was a zero on the last RAM read which was initiated by the F-11 MICRO. Power on initialization or writing bit <9> (BPERR CLR)
1: RAM parity bit for the low byte data was a one on the last RAM read which was initiated by the F-11 MICRO.

bit <11> - RMPHB - Read Memory parity High Byte

(read only)

0: RAM parity bit for the high byte data was a zero on the last RAM read which was initiated by the F-11 MICRO. Power on initialization or writing bit <9> (BPERR CLR)
1: RAM parity bit for the high byte data was a one on the last RAM read which was initiated by the F-11 MICRO.

bit <10> - WPDI XOR - Write Parity Datain XOR
This is a flip-flop which exclusively or's itself with the low or high byte parity bit on all byte write instructions; or exclusively or's itself with the exclusive or of the low and high byte parity bits on all word write instructions. (read/write)

0: the WPDI XOR flop is read as a zero; or clear to a zero by writing bit <10> to a zero.
1: the WPDI XOR flop is read as a one; or set to a one by writing bit <10> to a one.

bit <9> - BPERR CLR - Bus Parity Error Clear (write only)

0: No function.
1: clear the CSLB PERR, RMPLB, and RMPHB flops to zeros.

- RLB RDCOK -RL Bus Receive DCOK (read only)

0: DC power in the BA-11n cabinet is okay.
1: DC power in the BA-11N cabinet is NOT okay.

bit <8> - DMA PCLR - DMA Parity Error Clear (write only)

0: No function.
1: clear the DMA RPER flag.

- RLB RPOK -RL Bus Receive POK (read only)

0: AC power in the BA-11n cabinet is okay.
1: AC power in the BA-11N cabinet is NOT okay.

Bit <7> - BPER TRP - Bus Parity Error Trap (read/write)

0: BPER TRP is read as zero; or cleared to a zero by writing bit <7> to a zero. This bit is also cleared on power up initialization.

1: BPER TRP is read as a one; or set to a one by writing bit <7> to a one. When BPER TRP is a one, a Console Bus Parity error (CSLB PERR) will cause a trap to location 114 (octal).

bit <6> - IRESET - IOB Interface Reset (read/write)

0: IRESET is read as a zero; or cleared to a zero by writing bit <6> to a zero. This bit is also cleared on power up initialization.
1: IRESET is read as a one; or set to a one by writing bit <6> to a one. As long as IRESET is a one the IOB Interface RESET is asserted to the I/O Box module.

bit <5> - IINT EN - I/O Box Interrupt Enable (read/write)
This bit is used to enable interrupts from the I/O Box - IOB Interface interrupts IOB ERR, IOB ATN, and IOB EFIN.

0: IINT EN is read as a zero; or cleared to a zero by writing bit <5> to a zero. Power up initialization also clears this bit.
1: IINT EN is read as a one; or set to a one by writing bit <5> to a one. When IINT EN is one, all interrupts from the I/O Box are enabled.

bit <4> - RLBLP - RLB Interface Loopback (read/write)
This bit enables loopback mode on the RLB Interface data. In loopback mode the data (RLB D15:D00) is latched in the interface receivers on a write operation to the RLB Interface and remains latched until the next write operation to the RLB Interface. A read operation to the RLB Interface reads the latched data. The RLB Interface data lines require termination at both ends of the line and, therefore; DC POWER MUST BE ON TO THE BA11-N CABINET FOR PROPER OPERATION OF LOOPBACK MODE.
Valid addresses to the RLB Interface in loopback mode are 774400-774576.

0: RLBLP is read as a zero; or cleared to a zero by writing bit <4> to a zero. Power on initialization clears this bit.
1: RLBLP is read as a one; or set to a one by writing bit <4> to a one. When RLBLP is a one, the RLB Interface loopback mode is enabled.

bit <3> - WE PLB - Write Even Parity Low Byte (read/write)
This bit enables the generation of even parity for the low byte data (bits 7:0) for write

operations to the RAM or I/O Box.

0: WE PLB is read as a zero; or cleared to a zero by writing bit <3> to a zero. Power on initialization also clears this bit. When WE PLB is a zero, odd parity is generated for the low byte data.
1: WE PLB is read as a one; or set to a one by writing bit <3> to a one. When WE PLB is a one, even parity is generated for the low byte data.

bit <2> - WE PHB - Write Even Parity High Byte (read/write)
This bit enables the generation of even parity for the high byte data (bits 15:8) for write operations to the RAM or I/O Box.

0: WE PHB is read as a zero; or cleared to a zero by writing bit <3> to a zero. Power on initialization also clears this bit. When WE PHB is a zero, odd parity is generated for the low byte data.
1: WE PHB is read as a one; or set to a one by writing bit <3> to a one. When WE PHB is a one, even parity is generated for the low byte data.

bit <1> - ERADR 17 - Error Address bit 17 (read only)
This bit reflects the state of physical address bit 17 at the time of an error condition. See 2. Maintenance Error Register 0 (MTERR0).

0: ERADR 17 is read as a zero.
1: ERADR 17 is read as a one.

bit <0> - ERADR 16 - Error Address bit 16 (read only)
This bit reflects the state of physical address bit 16 at the time of an error condition. See 2. Maintenance Error Register 0 (MTERR0).

0: ERADR 16 is read as a zero.
1: ERADR 16 is read as a one.

2.13.3 DIAGNOSTIC CONTROL REGISTER (DIAG CR)

The DIAG CR is used to control the enabling of loopback features on the console module. This register resides in the console address space at 771104 (octal). The DIAG CR is not initialized on power up. The format for the DIAG CR is as follows.

Address: 771104

15	14	13	12	11	10	09	08
NOT USED							

07	06	05	04	03	02	01	00
IOBXLP	SIODSL	SIO2BL	SIO2AL	SIO1BL	SIO1AL	SIO0BL	SIO0AL

bits <15:8> - Not Used - These bits are not implemented and are read and written as don't care.

bit <7> - IOBXLP - IOB Interface Loopback (read/write)
This bit enables loopback mode on the IOB Interface data and byte (low and high) parity bits. In loopback mode the data (IOB D15:D00) and byte parity bits are latched in the interface receivers on a write operation to the IOB Interface and remains latched until the next write operation to the IOB Interface. A read operation to the RLB Interface reads the latched data and byte parity is checked. The IOB Interface data and parity lines require termination at both ends of the line and, therefore, DC POWER MUST BE ON TO THE I/O BOX MODULE FOR PROPER OPERATION OF LOOPBACK MODE.
Valid addresses to the IOB Interface in loopback mode are 776200-776776.

0: IOBXLP is read as a zero; or cleared to a zero by writing bit <4> to a zero.
1: IOBXLP is read as a one; or set to a one by writing bit <4> to a one. When IOBXLP is a one, the RLB Interface loopback mode is enabled.

bit <6> - SIODSL -Serial Input Output Data Set Loopback (read/write)
This bit enables loopback mode on the Data Set signals for SIO 0, Chanel A (KLINIK line). Loopback mode loops DTR (Date Terminal Ready) to DSRDY (Data Set Ready); RTS (Ready to Send) to CTS (Clear to Send); and DTR (Data Terminal Ready) to DCD (Data Carrier Detect). All Data Set signals are looped at the output of the SIO chip (before the interface line transmitter)

This page has been changed

to the input of the SIO chip (after the interface line receiver.

0: SIODSL is read as a zero; or cleared to a zero by writing bit <6> to a zero.
1: SIODSL is read as a one; or set to a one by writing bit <6> to a one. When SIODSL is a one, the Data set loopback mode is enabled.

bit <5> - SIO0AL -Serial Input Output 0, Chanel A Loopback (read/write)
This bit enable loopback on transmit data to receive data for SIO0, Chanel A (EMM).

0: SIO0AL is read as a zero; or cleared to a zero by writing bit <5> to a zero.
1: SIO0AL is read as a one; or set to a one by writing bit <5> to a one. When SIO0AL is a one, loopback mode is enabled and the transmit interface line is held in a "marking" condition.

bit <4> - SIO0BL -Serial Input Output 0, Chanel B Loopback (read/write)
This bit enable loopback on transmit data to receive data for SIO0, Chanel B (CTY).

0: SIO0BL is read as a zero; or cleared to a zero by writing bit <4> to a zero.
1: SIO0BL is read as a one; or set to a one by writing bit <4> to a one. When SIO0BL is a one, loopback mode is enabled and the transmit interface line is held in a "marking" condition.

bit <3> - SIO1AL -Serial Input Output 1, Chanel A Loopback (read/write)
This bit enable loopback on transmit data to receive data for SIO1, Chanel A (KLINIK).

0: SIO1AL is read as a zero; or cleared to a zero by writing bit <3> to a zero.
1: SIO1AL is read as a one; or set to a one by writing bit <3> to a one. When SIO1AL is a one, loopback mode is enabled and the transmit interface line is held in a "marking" condition.

bit <2> - SIO1BL -Serial Input Output 1, Chanel B Loopback (read/write)
This bit enable loopback on transmit data to receive data for SIO1, Chanel B (VT).

0: SIO1BL is read as a zero; or cleared to a zero by writing bit <2> to a zero.

1: SIO1BL is read as a one; or set to a one by writing bit <2> to a one. When SIO1BL is a one, loopback mode is enabled and the transmit interface line is held in a "marking" condition.

bit <1> - SIO2AL -Serial Input Output 2, Chanel A Loopback (read/write)
This bit enable loopback on transmit data to receive data for SIO2, Chanel A (DNxx).

0: SIO2AL is read as a zero; or cleared to a zero by writing bit <1> to a zero.
1: SIO2AL is read as a one; or set to a one by writing bit <1> to a one. When SIO2AL is a one, loopback mode is enabled and the transmit interface line is held in a "marking" condition.

bit <0> - SIO2BL -Serial Input Output 2, Chanel B Loopback (read/write)
This bit enable loopback on transmit data to receive data for SIO0, Chanel B (INTERCONSOLE).

0: SIO2BL is read as a zero; or cleared to a zero by writing bit <0> to a zero.
1: SIO2BL is read as a one; or set to a one by writing bit <0> to a one. When SIO2BL is a one, loopback mode is enabled and the transmit interface line is held in a "marking" condition.

2.14 DIAGNOSTIC SCAN PORT SELECT (DIAG SCAN PORT SEL)

The Diagnostic Scan Port Selects are used in conjunction with the Diagnostic Scan path associated with the Ports. The Port Diagnostic Scan path is used to load Port Microcode, and for diagnostic visibility into the Port logic.

A Diagnostic Scan Data Register - 16 bit parallel load/output shift register - is located on the I/O Box module. Each Port (mamimum of seven) also has a Diagnostic Scan Data Register. The Diagnostic Scan Data Register on the I/O Box module forms a serial data loop to each Port's Diagnositc Scan Data register. The serial data stream from the I/O Box Diagnostic Scan Data is passed thru the IOB Interface cable to all Ports. The serial data stream leaving each Port's Diagnostic Scan Data register is routed to the console module. Depending upon DIAG SCAN PORT SEL, the console module inclusive or's the serial data stream from the Ports into a single data stream which is passed through the IOB Interface cable to the I/O Box Diagnostic Scan Data Register. In addition to control of the or'ing of Port Scan serial data, the DIAG SCAN PORT SEL register sends a Port select line to each Port to allow selective Port

participation in the Diagnostic Scan path.

2.14.1 Port Select Interface

The Console module and Ports 0-3 are in the I/O Backplane. Ports 4-7 are in the I/O Expansion Backplane. The Port Select (PS0 - PS7 SEL) are routed along both backplanes. PS4 - PS7 SEL on the I/O Backplane are connect by a Flex Print cable to PS0 - PS3 SEL on the I/O Expansion Backplane. The eight (8) interface lines are TTL compatible and as follows.

1. PS0 SEL - asserted low when port 0 is selected.
2. PS1 SEL - asserted low when port 1 is selected.
3. PS2 SEL - asserted low when port 2 is selected.
4. PS3 SEL - asserted low when port 3 is selected.
5. PS4 SEL - asserted low when port 4 is selected.
6. PS5 SEL - asserted low when port 5 is selected.
7. PS6 SEL - asserted low when port 6 is selected.
8. PS7 SEL - asserted low when port 7 is selected.

2.14.2 DIAG SCAN PORT SEL Register

The format of the DIAG SCAN PORT SEL register is as follows.

Address: 771400

15	14	13	12	11	10	09	08
NOT USED							

07	06	05	04	03	02	01	00
PS0 SEL	PS1 SEL	PS2 SEL	PS3 SEL	PS4 SEL	PS5 SEL	PS6 SEL	PS7 SEL

bits <15:8> - Not Used - These bits are not implemented and

read and written as don't care.

bit <7> - PS0 SEL - Port Select 0 (read/write)

0: PS0 SEL is read as a zero; or cleared to a zero by writing bit <7> to a zero.
1: PS0 SEL is read as a one; or set to a one by writing bit <7> to a one. When PS0 SEL is a one, Port 0 is active in the diagnostic scan path and its serial data stream to the I/O box is being inclusive or'ed with other selected Ports.

bit <6> - PS1 SEL - Port Select 1 (read/write)

0: PS1 SEL is read as a zero; or cleared to a zero by writing bit <6> to a zero.
1: PS1 SEL is read as a one; or set to a one by writing bit <6> to a one. When PS1 SEL is a one, Port 1 is active in the diagnostic scan path and its serial data stream to the I/O box is being inclusive or'ed with other selected Ports.

bit <5> - PS2 SEL - Port Select 2 (read/write)

0: PS2 SEL is read as a zero; or cleared to a zero by writing bit <5> to a zero.
1: PS2 SEL is read as a one; or set to a one by writing bit <5> to a one. When PS2 SEL is a one, Port 2 is active in the diagnostic scan path and its serial data stream to the I/O box is being inclusive or'ed with other selected Ports.

bit <4> - PS3 SEL - Port Select 3 (read/write)

0: PS3 SEL is read as a zero; or cleared to a zero by writing bit <4> to a zero.
1: PS3 SEL is read as a one; or set to a one by writing bit <4> to a one. When PS3 SEL is a one, Port 3 is active in the diagnostic scan path and its serial data stream to the I/O box is being inclusive or'ed with other selected Ports.

bit <3> - PS4 SEL - Port Select 4 (read/write)

0: PS4 SEL is read as a zero; or cleared to a zero by writing bit <3> to a zero.
1: PS4 SEL is read as a one; or set to a one by writing bit <3> to a one. When PS4 SEL is a one, Port 4 is active in the diagnostic scan path and its serial data stream to the I/O box is being inclusive or'ed with other selected Ports.

bit <2> - PS5 SEL - Port Select 5 (read/write)

0: PS5 SEL is read as a zero; or cleared to a zero by writing bit <2> to a zero.
1: PS5 SEL is read as a one; or set to a one by writing bit <2> to a one. When PS5 SEL is a one, Port 5 is active in the diagnostic scan path and its serial data stream to the I/O box is being inclusive or'ed with other selected Ports.

bit <1> - PS6 SEL - Port Select 6 (read/write)

0: PS6 SEL is read as a zero; or cleared to a zero by writing bit <1> to a zero.
1: PS6 SEL is read as a one; or set to a one by writing bit <1> to a one. When PS6 SEL is a one, Port 6 is active in the diagnostic scan path and its serial data stream to the I/O box is being inclusive or'ed with other selected Ports.

bit <0> - PS7 SEL - Port Select 7 (read/write)

0: PS7 SEL is read as a zero; or cleared to a zero by writing bit <0> to a zero.
1: PS7 SEL is read as a one; or set to a one by writing bit <0> to a one. When PS7 SEL is a one, Port 7 is active in the diagnostic scan path and its serial data stream to the I/O box is being inclusive or'ed with other selected Ports.

2.15 RLB INTERFACE DESCRIPTION

The RLB Interface provides a DMA (Direct Memory Access) path between the MSU device (RL02 Disk Drive) and the Console subsystem resident memory. The DMA operations are enabled by the console processor software. The Console subsystem MSU device contains all the required microcode, system boot code, diagnostics, etc. that is necessary to bring up the hardware portion of the system. The RLB Interface connects to the RLV-11 (RL02 Controller) which resides in the BA11-N Mounting Box. The BA11-N along with Two (2) RL02 Disk Drives reside in the JUPITER External cabint. Connection to the BA11-N is made with Two BC06 cables from the I/O Backplane console module slot.

The RLV-11 consists of Two (2) quad height modules (M8013 and M8014) which plug into the BA11-N backplane. These two modules controls DMA transfers between the console RAM and RL02 Disk Drive. The DMA arbitration is done by the console module.

The RLB Interface lines implements the "QBUS" Protocol and equivalent signals with the exception of BHALT and BREF. The RLB Interface is also restricted to addresses in the range 774400-774576, and a level 4 interrupt vector at address 160.

The following information characterizes the console subsystem RL02 Disk Drive unit operational capabilities:

1. DMA path to the console memory

2. Full 18 bit memory address support.

The RL02 disk interface consists of the following sixteen bit registers:

Mnemonic	Function
Address	Register Mnemonic and Key
774400	CSR - Control/Status R/W
774402	BAR - Bus Address R/W
774404	DAR - Disk Address R/W
774406	MPR - Mulitpurpose R/W

2.16 SLU - SERIAL LINE UNIT - DESCRIPTION

The ZILOG Z80-SIO chip is used to implement the Serial Line Units for the JUPITER. The JUPITER Console subsystem implements six serial line units. These SLU's are serial asynchronous EIA RS232C and RS422/3 lines. The line speed is programmable - WRITE ONLY - for speeds between 50 baud and 19.2 kilobaud. The character format for the transmitter and receiver must be the same. The functional allocation is as follows (The addresses below are restricted to the low byte address exactly as shown):

1. EMM - Bus Address 776000 (channel A of SIO 0)

   Interrupt Priority 5

   Interrupt Vector 300

   This is used as the communication link between the EMM/s and Console module. The line speed is programmable for the transmitter/receiver. Split line speed is not supported (i.e. the transmitter and receiver speeds are programmed the same.

2. CTY LINE - Bus Address 776010 (channel B of SIO 0)

   Interrupt Priority 5

   Interrupt Vector 300

This is a RS422 line and used for console CTY, line speed is programmable and split line speed operation is supported (i.e. the transmitter and receiver speeds are independently programable).

3. KLINIK LINE - Bus Address 776020 (channel A of SIO 1)

   Interrupt Priority 5

   Interrupt Vector 310

   This is a RS232 a line and used for KLINIK applications. The line speed is programmable and split speed operation is supported. All Modem signals are supported.

4. VTxxx LINE - Bus Address 776030 (channel B of SIO 1)

   Interrupt Priority 5

   Interrupt Vector 310

   This is a RS232 line and used for the console VTxxx. The line speed is programmable and split speed operation is not supported.

5. DNxxx LINE - Bus Address 776040 (channel A of SIO 2)

   Interrupt Priority 5

   Interrupt Vector 320

   This is a RS232 line and reserved for DNxxx applications. The line speed is programmable and split speed operation is not supported.

6. INTERCONSOLE LINE - Bus Address 776050 (channel B of SIO 2)

   Interrupt Priority 5

   Interrupt Vector 320

   This is a RS422/3 line and reserved for local applications. The line speed is programmable.

The following table details the programmable baud rate generator addresses:

Bus Address	Function
776004	EMM TX/RX BAUD RATE SEL (SIO0 A) - (4 bit register)
776014	CTY LINE TX/RX BAUD RATE SEL (SIO0 B) - Split baud rate (8 bit register)
776024	KLINIK LINE TX/RX BAUD RATE SEL (SIO1 A) - Split baud rate (8 bit register)
776034	VT LINE TX/RX BAUD RATE SEL (SIO1 B) - (4 bit register)
776044	DNxx LINE TX/RX BAUD RATE SEL (SIO2 A) - (4 bit register)
776054	INTERCONSOLE LINE TX/RX BAUD RATE SEL (SIO2 E - (4 bit register)

The Remote Line supports the following modem control signals:

1. Protective ground
2. Transmit Data (Output)
3. Receive Data (Input)
4. Request To Send (Output)
5. Clear To Send (Input)
6. Data Terminal Ready (Output)
7. Data Carrier Detect (Input)
8. Signal Ground
9. Data Set Ready (Input)

2.17 TIME OF YEAR CLOCK

The console Interval Timer is a programmable interrupt timer and Time of Day clock. This logic circuitry has an on board (on module) battery backup. The function consists of an oscillator, Time of Year (TOY) chip with an interval timer feature (Time Pulse output), a 16-bit parallel load shift register and control register. The clock circuit with its internal registers is found in addresses:

The TOY chip is powered by the +2.8 volt battery to retain and continue updating the time of year when console module power is off. The battery is diode isolated from the console module +5.0 volts which supplies power to the TOY chip when power is on.

The TOY chip has an internal forty (40) bit shift register which is used to set or read the time of year. The shift register form a continues serial path with the TOY DATA register. On a time read command the TOY chip parallel loads the time of year into its 40-bit shift register. A shift go (TOY GO) is then issued to the TOY logic which cause 16-bits of the time of year to be shifted from the TOY chip to the TOY DATA register and shift done (TSD FLAG) is set and the shift operation stops. The TOY DATA register now contains the first 16-bits of time data and a read TOY DATA registers is performed. The TOY GO and read TOY DATA register is repeated until the entire time of year is read.

For a set time command operation the procedure is reversed using writes to the TOY DATA register. The time of year data is shifted in the following order - LSB first.

1. Seconds (binary coded Decimal)
2. Minutes (binary coded decimal)
3. Hour (binary coded decimal)
4. Date (binary coded decimal)
5. Month (hexa-decimal code)

The TOY chip also has a programed Time Pulse Output (TP) which is implemented as an interval timmer.
771000 To 771077 Reserved for the TOY logic.

2.17.1 TOY COMMAND/STATUS REGISTER (TSC)

The format for the TSC is as follows.

Adress: 771000

15	14	13	12	11	10	09	08
TP FLAG	TSD FLAG	TOY GO					TCMD STRB

07	06	05	04	03	02	01	00
TP IEN	TSD IEN				TPSEL	CMDSEL 1	CMDSEL 0

tab stop 5,10

bit <15> - TP FLAG - Time Pulse Flag (read/write)
This bit is the source for the interval timmer.

0: TP FLAG is read as a zero; or cleared to a zero by writing bit <15> to a zero. Power on initialization also clears TP FLAG.
1: TP FLAG is read as a one; or set to a one by writing bit <15> to a one. TP FLAG is also set on the interval frequency of the Time Pulse Output from the TOY chip. (See bits <2:0>)
When TP FLAG is a one and TP IEN (bit <7>) is a

one, a EVENT interrupt is initiated which causes a trap to location 100 (octal).

bit <14> - TSD FLAG - Toy Shift Done Flag (read/write)
This bit is an enable to the TOY interrupt request. The hardware sets this bit when 16-bits have been shifted in the TOY DATA register.

0: TSD FLAG is read as a zero; or cleared to a zero by writing bit <14> to a zero.
1: TSD FLAG is read as a one; or set to a one by writing bit <14> to a one. When TSD FLAG is a one and TSD IEN (bit <6>) is a one, a vector interrupt will occurr at location 350 (octal).

bit <13> - TOY GO - Start Shifting (write only)
This bit enables a 16 bit displacement shift of the TOY chip and TOY DATA registers in a "round robin" fashion.

0: No function. read as don't care.
1: TOY GO is set to a one by writing bit <13> to a one.

bits <12:9> - Not Used - These bits are not implemented and read and write as don't care.

bit <8> - TCMD STRB - TOY Command Strobe (read/write)
This bit initiates the command as specified by TPSEL, CMDSEL 1, and CMDSEL 0 (bits <2:0>).

0: TCMD STRB is read as a zero; or cleared to a zero by writing bit <8> to a zero. This bit is also cleared by DCLO from the EMM.
1: TCMD STRB is read as a one; or set to a one by writing bit <8> to a one. On the transistion of TCMD STRB to a one, the command specified by TPSEL, CMDSEL 1, and CMDSEL 0 is latched in the TOY chip and remain latched as long as TCMD STRB is a one. TCMD STRB must be a zero for two (2) microseconds (minimum) prior to the initialization of a command.

bit <7> - TP IEN - Time Pulse Interrupt Enable (read/write)

0: TP IEN is read as a zero; or cleared to a zero by writing bit <7> to a zero. This bit is also cleared on power on initialization.
1: TP IEN is read as a one; or set to a one by by writing bit <7> to a one. When TP IEN is a one and TP FLAG (bit <15>) is a one, an EVENT interrupt is initiated which causes a trap to

location 100 (octal).

bit <6> - TSD IEN - Toy Shift Done Interrupt Enable (read/write)
This bit is an enable to the TOY interrupt request.

0: TSD IEN is read as a zero; or cleared to a zero by writing bit <6> to a zero.
1: TSD IEN is read as a one; or set to a one by writing bit <6> to a one. When TSD FLAG is a one and TSD FLAG (bit <14>) is a one, a vector interrupt will occurr at location 350 (octal).

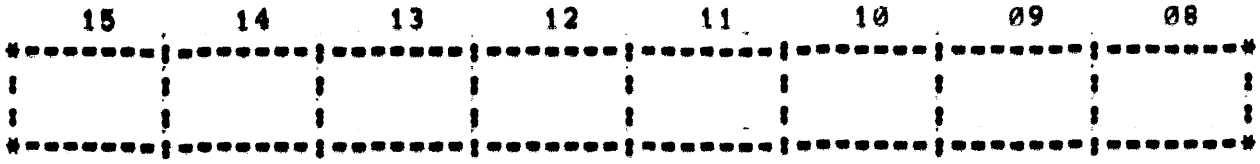
bits <5:3> - Not Used - These bits are not implemented and read and write as don't care.

bits <2:0> - TP SEL, CMDSEL 1, CMDSEL 0 (read/write)
These bits are used to drive the command inputs of the TOY chip. The decoding is as follows.

TP SEL	CMDSEL 1	CMDSEL 0	FUNCTION
0	0	0	Register Hold - Holds the 40-bit shift register.
0	0	1	Register Shift - Shifting the 40-bit shift register is enabled.
0	1	0	Time Set - Presets the data of the 40-bit shift register in the time counter. Shifting is disabled.
0	1	1	Time Read - Read into the 40-bit shift register the data of the time counter. Shifting is disabled.
1	0	0	TP = 64Hz - The TP FLAG is clocked at 64Hz rate.
1	0	1	TP = 256Hz - The TP FLAG is clocked at 256Hz rate.
1	1	0	TP = 2048Hz - The TP FLAG is clocked at a 2048Hz rate.
1	1	1	Test Mode

The format for the TOY DATA register is as follows.

Address: 771002



2.18 I/O BOX INTERFACE DESCRIPTION

Communications between the console module and the I/O Box module is performed over the IOB interface. The I/0 Box appears to the console as 192 16-bit registers in the console I/0 address space. All data sent over this interface is checked for odd byte parity. The console IOB transceiver chips has internal odd parity generation and checking logic which is used for both the RAM (64K x 18 Dynamic Randon Access Memory) and I/O Box transactions. The IOB Interface uses open collector bus drivers and termination at both ends of the bus for bi-directional signals. The uni-directional lines are terminated at the receiving end.

The IOB Interface Bus signals are defined as follows.

1. IOB D15:D00 L - Bi-directional data lines asserted low for a one. These lines transfer data between the console module and I/O Box module. A 16-bit address is aslo transfered to the I/0 Box to select one of 192, 16-bit registers.

2. IOB PARL L - Bi-directional parity line for the low byte data (IOB D07:D00). Asserted low for a one.

3. IOB PARH L - Bi-directional parity line for the high byte data (IOB D15:08). Asserted low for a one.

4. IOB SEL L - Uni-directional line to the I/O Box asserted low to indicate that an address for the I/O box is on the bus and valid to decode and a read or/and write cycle/s to follow. IOB SEL L remains asserted after IOB READ L is deasserted to indicate that a IOB WRITE L will follow (read-pause-write operation)/

5. IOB READ L - Uni-directional line to the I/O Box asserted low and indicates a read operation from the the I/O Box.

6. IOB WRITE L - Uni-directional line to the I/O Box asserted low and indicates a write operation to the I/O Box.

7. IOB BTWT L - Uni-directional line to the I/O Box asserted low during the address valid time to indicate a write only operation to follow; or asserted low during the assertion of IOB WRITE L to indicate a byte write operation.

8. IOB RESET L - Uni-directional line to the I/O Box asserted low.

9. -IOB ACLO L - Uni-directional line to the I/O Box asserted low when AC power is okay.

10. DCLO L - Uni-directional line to the I/O Box asserted low when the System DC power is bad. This line is passed through the I/O Box to the MBOX for JUPITER Memory battery back-up purposes.

11. IOB ACK L - Unidirectional line to the console module asserted low by the I/O Box to indicate that it has valid data on the bus or accepted data from the console.

12. IOB ERR L - Uni-directional line to the console module asserted low by the I/O Box to indicate that it has detected an error condition in the JUPITER CPU, PORT/s, I/O Box.

13. IOB ATN L - Uni-directional line to the console module asserted low by the I/O Box to indicate That the console's attention is required for a transaction.

14. IOB EFIN L - Uni-direction line to the console module asserted low by the I/O Box which causes the console processor to be "jam" started to location 160000 (octal). This line is initialized by JUPITER software when communication to the console is losted.

15. SDR IN H - Uni-directional line to the I/O Box asserted high for a one. This signal line is the inclusive or of selected Ports' diagnostic scan data. (refer to 2.x DIAG SCAN PORT SEL)

776200-776776 I/O-Box internal registers

2.19 ENVIRONMENTAL MONITOR

The system environmental monitor consists of various thermal sensing mechanisms located in the machine (not on the actual JUPITER logic modules) and various sensors located in the power subsystem.

CHAPTER 3

CONSOLE HARDWARE PROGRAMMING

3.1 INTRODUCTION

The system console interfaces to the JUPITER-TTL bus as an invisible type of port, that is, it uses the TTL bus as the path from the console to the CPU. Using what is, essentially, a subset of the basic I/O port protocol, the console can request interrupt service, pass information to the CPU, access memory at the discretion of the memory controller, etc.

The console can also use the TTL bus to pass information to, and receive information from, any of the ports. The console also has access to the I/O-bus arbitration logic, and, with such access it can initiate a memory to port transfer. This can be done through the simulation of the doorbell function provided by the E-BOX. The console can ring the bell of a given port after the console has loaded the console memory registers in the I/O Box with the proper transaction data. The E-BOX path to the ports is through the I/O-Box and so is the console path to the rest of the system. This allows the I/O-Box to be designed in such a manner as to allow the console to access all of the functions available to a port, E-BOX, or M-BOX. A certain amount of caution will be exercised in the design of this area of the console and I/O-Box in order to preclude the possibility of a paranoid console (console has access to the machine - so - take over the busses).

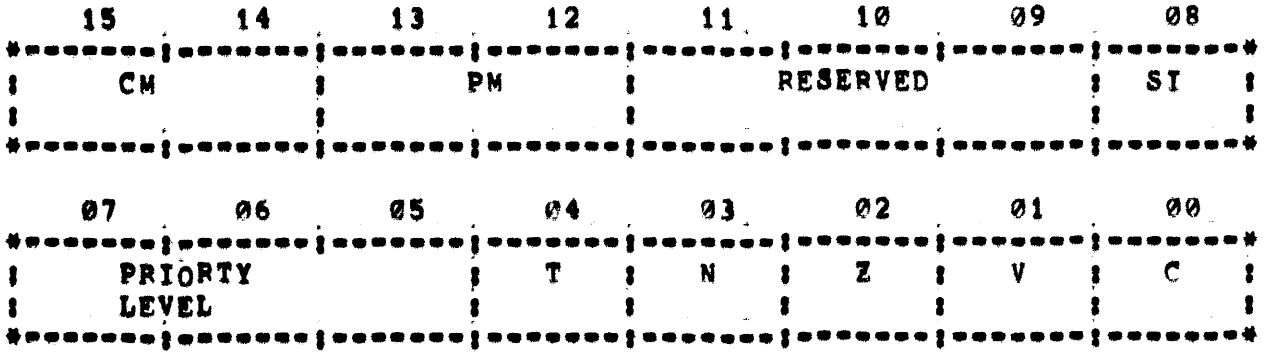
The console is notified of the fact that the JUPITER CPU has data for it by the use of the IOB ATN interface line.

3.2 F-11 - CONSOLE PROCESSOR

The F-11 implementation used on the console subsystem does not have the capability of running existing DEC software without modification. The standard processor tests, memory tests and so on will run but the I/O routines for passing information from the operator to the cpu or vice versa will not work because of the fact that there is no DL-11 type device present at the classical DL-11 address.

3.2.1 Processor Status Word (PSW)

The processor Status Word contains current status information about the operation of the CPU. Stack pointer selection is determined by the "current mode" information in the PS. In exception to this are the macroinstructions which move information to or from the previous data or instruction space. In these instances stack pointer selection is determined by the "previous mode" information. The PS fromat is described below.



bits <15:14> - Current Mode (CM)

These bits indicate what the present memory management modes is. (read/write bits)
PS<15:14>=00: Kernel Mode
PS<15:14>=01: Supervisor Mode
PS<15:14>=10: Unused **
PS<15:14>=11: User Mode

bits <13:12> - Previous Mode (PM)

These bits are used with memory management to indicate what the last memory management mode was. (read/write bits)
PS<13:12>=00: Kernel Mode
PS<13:12>=01: Supervisor Mode
PS<13:12>=10: Unused **

PS<13:12>=11: User Mode

** Unused mode is treated as User mode except during read/write access to a Stack Pointer register (Unused mode does NOT select ANY SP).

bits <11:9> - Reserved

bit <8> - Instruction Suspension bit (IS)

This bit is cleared on power up. When a potentially suspendable instruction is executed, PS<8> cleared means that instruction execution has NOT been suspended; set means that instruction execution has been suspended.

bits <7:5> - Priority Level

These bits are used by software to determine which interrupts will be processed. (read/write bits)

PS <7:5>Octal	Interrupt Level Acknowledged *
7	none
6	7,
5	7,6,
4	7,6,5,
3	7,6,5,4
2	7,6,5,4
1	7,6,5,4
0	7,6,5,4

*Higher levels acknowledged first.

bit <4> -Trace bit (T)

The Trace bit is used in debugging programs. It allows programs to be single-instruction stepped. If bit is set at the end of instruction execution, the processor traps to location 14 (octal).

bits <3:0> - Condition Codes

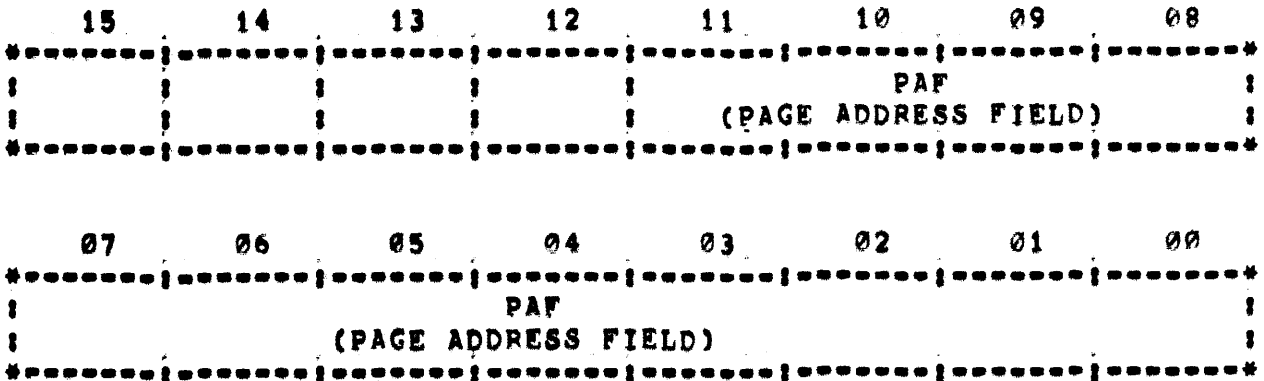
The condition codes contain information on the result of the last CPU operation. The bits are set after execution of all arithmetic or logical single-operand or double-operand instructions. The bits are set as follows.

N=1 if the result was negative.
Z=1 if the result was 0.

V=1 if the operation resulted in an arithmetic overflow.

C=1 if the operation resulted in a carry from the MSB (most significant bit) or a 1 was shifted from MSB or LSB (least significant bit).

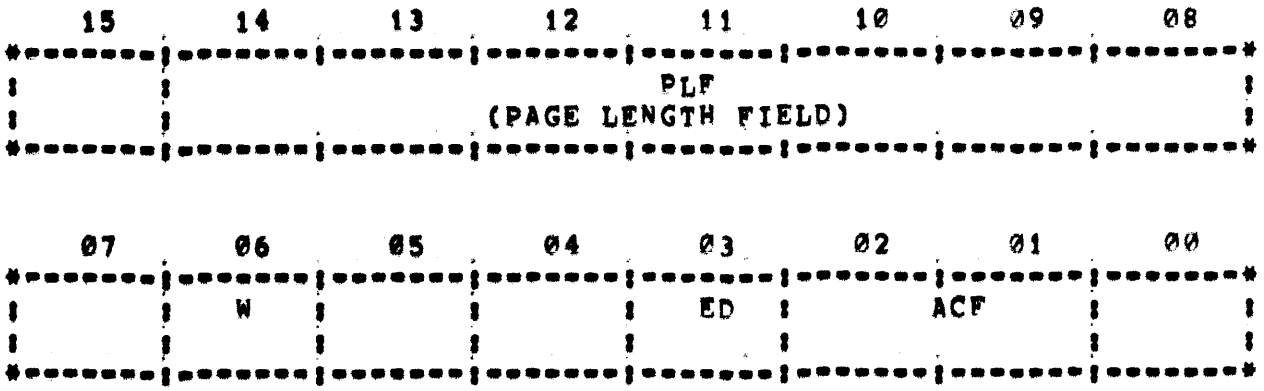
3.2.2 MMU PAGE ADDRESS REGISTER (PAR)



bits <11:00> - Page Address Field (PAF)

The PAF is added to bits <12:6> of the virtual address received from the DAL to create part of the 18 bit relocated physical address.

3.2.3 MMU PAGE DESCRIPTOR REGISTER (PDR)



bits <14:8> - Page Length Field (PLF)

The seven-bit PLF specifies the number of addressable blocks (block = 32 words) in the accessed page (it defines the page boundary). Bits <12:6> of the virtual address (desired block number) are compared with the PLF to detect illegal references outside this page boundary.

bit <6> = Written Into (W)

This bit is set if any location in the accessed page is wittten into after the page is initially loaded into memory (unless the relocated destination is one of the internal memory management registers). It can only be set during internally controlled relocation operations. It is cleared when the PDR or PAR of that page is explicityly written.

bit <3> = Expansion Direction (ED)

This bit controls the expansion direction of the page boundary and is used in checking for page lenght violations.

ED=0: Upward expansion
ED=1: Downward expansion

bits <2:1> = Access Control Field (ACF)

The two-bit ACF describes the access rights to the page. Any attempt to perform an operation not allowed by the ACF causes an MMU abort. The ACF codes are as follows:

ACF <2:1>	FUNCTION
00	Abort any access to this page
01	Abort any write access to this page.
10	Abort any access to this page.
11	No abort = allow read or write.

3.3 RL02 DISK DRIVE

The following define the RL02 Controller/Disk Drive registers.

3.3.1 CONTROL/STATUS REGISTER (CSR)

Address: 774400

15	14	13	12	11	10	09	08
ERR	DE	NXM	DLT/ HNF	DCRC/ HCRC	OPI	DS1	DS0

07	06	05	04	03	02	01	00
CRDY	IE	BA17	BA16	F2	F1	F0	DRDY

3.3.2 BUS ADDRESS REGISTER (BAR)

Address: 774402

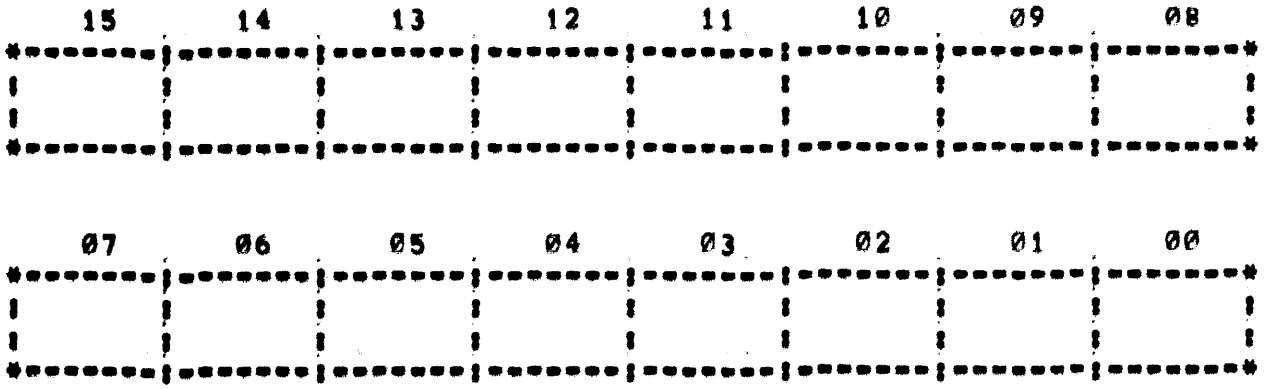
15	14	13	12	11	10	09	08
BUS ADDRESS 15:08							

07	06	05	04	03	02	01	00
BUS ADDRESS 07:01							0

3.3.3 DISK ADDRESS REGISTER (DAR)

The disk address register is a 16-bit, read/write, word-addressable register with and address of 774404. Its contents can have one of three meanings, depending on the function being performed. Clearing of this register is accomplished by executing a BUS INIT.

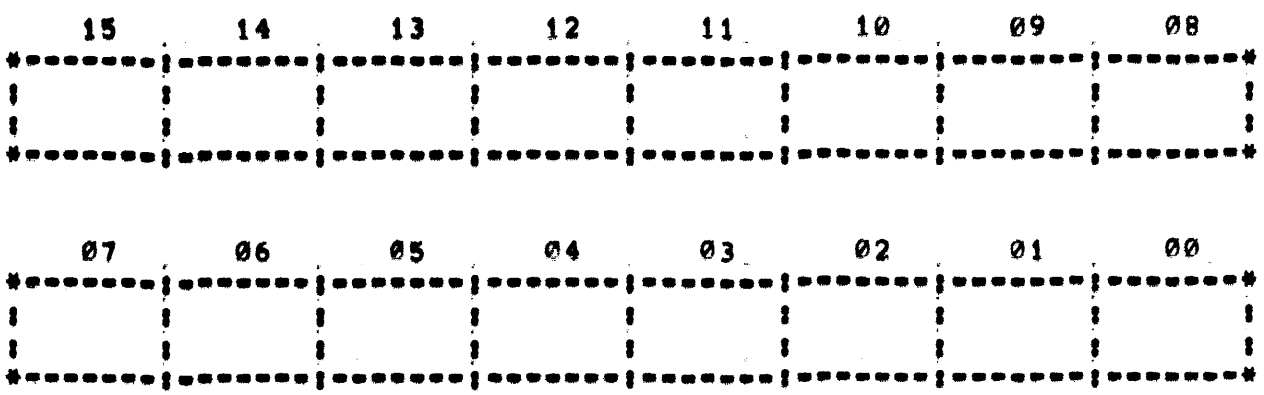
Address: 774404



3.3.4 MULTIPURPOSE REGISTER (MPR)

The MPR is two registers bearing the same base address - 774406. When writing into that location, the word counter accepts the data. When reading from that location, the FIFO output buffer provides the data.

Address: 774406



3.4 SLU - SERIAL LINE UNITS

The SLU's are implemented using the ZILOG Z80-SIO dual channel communications device. The Z80-SIO's are implemented as word addressable devices. Any attempt to write into the high byte of the SLU addresses is ignored while the low byte is accepted in word or byte mode operations. The SIO device is similar to some typical communications devices available on the Unibus (DQ-11, DMC-11, or KMC-11) in that there are actually registers hidden inside the device. The device appears to have two registers

Since I don't program this stuff... is there any significant impact on functionality in what I am saying here?

resident on the bus while in actuality the first address or CSR (Control and Status Register) is used as a pointer register.

The SIO register addressing scheme allows the internal registers to be selected by setting up the CSR register with the internal register pointer. The SIO internal register is then read by accessing the SLU Data register.

Each SLU's CSR (register addresses 776002, 776012, 776022, and 776032) or Control and Status Register has the following bit definitions:

D<2:0> REGISTER SELECT POINTER address (0 to 7 octal) bits

D<5:3> Command Select bits

D<7:6> Extended Command bits

The read and write functions appear to access different combinations of status information lumped together in a register. Write commands access up to 8 registers (including Reg 0) while the read commands access up to 3 registers.

3.4.1 Register Descriptions

The registers are defined as follows:

Write Reg	Function
0	Pointer and command
1	Interrupt control register
2	(REMLIN and LOCLIN2 only) Interrupt Vector register
3	Receive Mode Reg
4	Clock Mode Plus reg
5	Transmit Mode reg
6	Channel A sync reg (CTY or LOCLIN1 - not used)
7	Channel B sync reg (REMLIN or LOCLIN2 - not used)

Read Register	Function
0	Flag Register
1	Extended Flag Register
2	(REMLIN and LOCLIN2 only) Interrupt Vector register

The following information describes the internal SLU registers:

Write Reg 0

Bit	Title	Function
0 1 2	PNT 0 PNT 1 PNT 2	These bits are used to select the destination register for the next action; for a write, the next byte is written into the register selected by the pointer; for a read, the next read access is to the register selected by the pointer.
3 4 5	CMD 0 CMD 1 CMD 2	These bits are encoded to provide 8 commands to the SIO: 0 - null command, 1 - Send Abort (SDLC mode only not used by console), 2 - Reset external/status Interrupts. 3 - channel reset, 4 - reset receive interrupt on first receive character, 5 - reset transmitter interrupt pending, 6 - error reset, 7 - return from interrupt.
6 7	CRC 0 CRC 1	These bits are used to encode four CRC logic reset commands; 0 - Null command, 1 - reset RX CRC checker, 2 - reset TX CRC generator, 3 - reset Sending CRC/Sync latch. Only the null command is used by the console.

WRITE REG 1

0	EIE	External Interrupt Enable allows modem control status changes to cause an interrupt.
1	TIE	Transmit Interrupt Enable allows TBMT interrupts
2	SAV	Status Affects Vector allows the Interrupt Status to be encoded into the interrupt vector. This allows the various interrupt conditions to point to the proper service routine through the use of a dispatch table.
3 4	RIM 0 RIM 1	Rx Interrupt Mode bits provide and encoded command that controls the action taken on a Receiver Interrupt; 0 - Rx Interrupts Disabled. 1 - Rx interrupt on first character only. 2 - Interrupt on all received characters, parity error affects vector. 3 - Interrupt on all received characters, Parity Error does not affect Vector.
5	WROT	?
6	RFN	?
7	W/RE	?

WRITE REGISTER 2

This is the internal SIO interrupt vector register. This is a programmable register that can be used hold a basic pointer for a dispatch table. Coupled with the assertion of the SAV bit (bit 2 of Write Reg 1), the various interrupt causing conditions can be directed to the proper subroutine. For example, if a dispatch table were constructed with the following information:

XXX X00 Base address - pointer for Remlin TBMT service routine
XXX X02 Base address +2 - Pointer for Remlin Modem control service routine
XXX X04 Base address +4 - pointer for Remlin Receive data available routine
XXX X06 Base address +6 - pointer for Remlin special receive condition routine
XXX X10 Base address +10 - pointer for CTY TBMT service routine
XXX X12 Base address +12 - pointer for CTY Modem control routine (not used)
XXX X14 Base address +14 - pointer for CTY receive data avail routine
XXX X16 Base address +16 - pointer for CTY special receive condition routine

The same scenario would apply to LOCLIN1 and LOCLIN2. This is a programmable 8 bit register that does not generate the PDP-11 type interrupt vector but rather provides a handy dandy register to store the low byte pointer for the dispatch table (a programmers imagination can be counted on to achieve other uses for this mechanism).

WRITE REGISTER 3

0	RECEN	Receiver Enable.
1	SYNLDIN	Not used
2	ASM	Not used
3	RCRCE	Not used
4	EHM	Not used
5	AUTO	Automatic mode enable, DCD and CTS become the enable signals for RX and TX respectively.
6	RBC 0	Receiver Bits per Char select: 0-5 bits, 1-7 bits,
7	RBC 1	2 - 6 bits, 3 - 8 bits per character.

WRITE REGISTER 4

0	PAREN	Parity Enable
1	PE/-O	Parity Even (when set)/ Parity Odd (when clear)
2	SB 0	Stop Bit select: 0 - not used, 1 - 1 stop bit per
3	SB 1	character. 2 - 1.5 stop bits per character. 3 - 2 stop bits per character.
4	SYNO 0	Not used.
5	SYNO 1	
6	CLCK 0	Clock divisor select. Always set to 1 (01) for
7	CLCK 1	JUPITER use. 0 - X1 clock, 1 - X16 clock, 2 - X32 clock, 3 - X64 clock.

WRITE REGISTER 5

0	TCRCE	Tx CRC Enable - not used. Set to 0.
1	RTS	Request To Send. Set = ON, Clear = OFF.
2	-SDLC	Not Used. Setting does not matter
3	TX EN	Transmitter Enable.
4	BREAK	Send BREAK character. The Break character is sent for as long as this bit is true.
5	TBC 0	Tx Bit Count: 0 - 5 bits (or less), 1 - 7 bits,
6	TBC 1	2 - 6 bits, 3 - 8 bits per character.
7	DTR	Data Terminal Ready. Set = ON, Clear = OFF.

WRITE REGISTER 6 Not used.

WRITE REGISTER 7 Not used.

READ REGISTER 0

0	RCA	Rx Character Available. Commonly called Receive Data Available.
1	IP	Interrupt Pending. Available only on CTY and LOCLIN1. Always set to 0 in REMLIN and LOCLIN2.
2	TBMT	Transmitter Buffer eMpTy.
3	DCD	Shows the state of DCD signal at time of last external interrupt or following execution of command 2.
4	S/H	Not used in JUPITER. State indeterminate, must be ignored.
5	CTS	Shows state of CTS as in DCD above.
6	SC/S	Not used in JUPITER. State indeterminate, must be ignored.
7	B/A	Break character detected.

READ REGISTER 1

0	ALSENT	Used for diagnostic purposes to determine that the Transmit shift register is empty. (TBMT true coupled with this bit indicates that the Transmitter is DONE).
1	RC 0	
2	RC 1	
3	RC 2	
4	PE	
5	ROE	
6	C/FE	
7	EOF	

READ REGISTER 2
((More Data will be provided))

3.5 SYSTEM CONSOLE - I/O BOX INTERFACE PROGRAMMING

The I/O-BOX interface consists of up to 192 16-bit registers that can be read or written by the console processor. The actual register descriptions and addresses are described in the I/O Box Engineering Specification. The I/O-Box includes the JUPITER Diagnostic paths, the console to CPU path through the I/O box, the console to TTL-Bus PORT path, and the Environmental monitoring logic. The interrupt request level presented to the console sub-system from the I/O Box is:

Level 4 - CPU to Console interrupt request.

APPENDIX A

ADDRESS REFERENCE CHART

The following data attempts to provide a quick reference for the addressing, vector address and interrupt priority of each device on the internal console subsystem bus.

BUS ADDRESS	FUNCTION (READ/WRITE)	COMMON MNEMONIC	INTERRUPT PRIORITY	INTERRUPT VECTOR
777 776	PROCESSOR STATUS WORD			
777 774	STACK LIMIT			
777 772	PROGRAM INTERRUPT REQUEST			
777 770	} DEC RESERVED			
. .				
777 720	}			
777 717	USER	R6 (SP)		
777 716	SUPERVISOR	R6 (SP)		
777 707	}	R7 (PC)		
777 706	KERNEL	R6 (SP)		
777 705	}	R5		
777 704	}	R4		
777 703	} GENERAL	R3		
777 702	} REGISTER	R2		
777 701	} SET 0	R1		
777 700	}	R0		
777 656	}			
777 654	}			
777 652	}			
777 650	} USER ACTIVE PAGE			
777 646	} ADDRESS REGISTERS			
777 644	}			
777 642	}			
777 640	}			

777 616	|				
777 614	|				
777 612	|				
777 610	) USER ACTIVE PAGE				
777 606	) DESCRIPTOR REGISTERS				
777 604	|				
777 602	|				
777 600	|				
777 576	) STATUS REGISTER 2				
777 574	) STATUS REGISTER 1				
777 572	) STATUS REGISTER 0				
776 776	I/O Box	R/W			344 (IOB ATTN)
776 200	Interface Registers	R/W	IOB	4	340 (IOB ERR)
776 056	reserved				
776 054	Baud Rate Gen	W	(INTER-		
776 052	Cont/Stat Register	R/W	CONSOLE)		
776 050	Data Register	R/W	LOCLIN2	5	320 (SLU 2 Chan B)
776 046	reserved				
776 044	Baud Rate Gen	W			
776 042	Cont/Stat Register	R/W	(DN20)		
776 040	Data Register	R/W	LOCLIN1	5	320 (SLU 2 Chan A)
776 036	reserved				
776 034	Baud Rate Gen	W			
776 032	Cont/Stat Register	R/W			
776 030	Data Register	R/W	CSLVT	5	310 (SLU 1 Chan B)
776 026	reserved				
776 024	Baud Rate Gen	W			
776 022	Cont/Stat Register	R/W			
776 020	Data Register	R/W	KLINIK	5	310 (SLU 1 Chan A)
776 016	reserved				
776 014	Baud Rate Gen	W			
776 012	Cont/Stat Register	R/W			
776 010	Data Register	R/W	CTY	5	300 (SLU 0 Chan B)
776 006	reserved				
776 004	Baud Rate Gen	W			
776 002	Cont/Stat Register	R/W			
776 000	Data Register	R/W	EMM	5	300 (SLU 0 Chan A)
774 406	Multipurpose Reg.	R/W			
774 404	Disk Addr Register	R/W			
774 402	Bus Addr Register	R/W	(RL02)		
774 400	Cont/Stat Register	R/W	RLB	4	160

```
772 356 |
772 354 |
772 352 |
772 350 )  USER ACTIVE PAGE
772 346 )  ADDRESS REGISTERS
772 344 |
772 342 |
772 340 |
772 316 |
772 314 |
772 312 |
772 310 )  USER ACTIVE PAGE
772 306 )  DESCRIPTOR REGISTERS
772 304 |
772 302 |
772 300 |


771 400    Port Scan Selects      R/W      EMM

771 200    CSL Panel Interface    R/W      CPI

771 104    Diag Cont Register     R/W      DIAGCR
771 102    Maint Error Reg 1      R/W      MTER1
771 100    Maint Error Reg 0      R        MTER0

771 002 )  TOY Data Register      R/W
           )                                        6        100 (Interval Timer)
771 000 )  TOY Cmd/Stat Reg.      R/W      TOY     5        350 (Shift Done)

770 777 )  Non-Volatile
 |   |  )  (Battery-Backed)
770 000 )  RAM                    R/W      NVRAM

767 776 )  I/O Page
 |   |  )  Resident
760 000 )  ROM                    READ ONLY

------     )  Power Fail          READ ONLY        ?        024
------     )  Bus Error                                     004
```

APPENDIX B

F-11 FUNCTIONAL DESCRIPTION

B.1 OVERVIEW

The F-11 chip is a 16-bit LSI microprocessor with PDP-11/34 functionality. It can also execute FP11 floating point instructions and the DEC standard 168 Commercial Instruction SET (CIS). The chip set is composed of three different types of chipes. The Date chip (DC302) is the primary data path. It contains PDP-11 registers, scratchpad registers, and the general purpose ALU. The Control chip (DC303), of which there are several difference microprogrammed versions, contains the control store in PLA and ROM terms and the microprogram sequencing logic necessary for macroinstruction emulation. Note that in its native language the F-11 chip set is not a PDP-11. The chip set emulates PDP-11 macroinstructions by executing an appropriate string of microinstructions. The Memory Management chip (DC 304) not only contains the necessary registers and arithmetic logic to implement 18 and 22-bit adressing, but also contains the FP11 floating point register file.

B.2 DATA CHIP

The Data chip contains the PDP-11 general registers, several scratch pad registers, the processor status word (PS), the ALU, and the conditional branching logic. This chip performs all arithmetic and logical functions, handles data and address transfers with the external work (except relocated addresses), and coordinate most interchip communication.

B.3 CONTROL CHIP

The Control chip contains the microprogram sequence logic and 552 words of microprogram storage in PLA and ROM arrays. The difference microprogrammed versions of Control chips implement difference macroinstruction sets. The base Control chip (303-x01)* emulates the basis PDP-11 instruction set and provides the console function (ODT). Two additional Control chips (303-x02,-x03) implement the 46 FP11 floating point instructions. Six other chips (303-x04, -x05, -x06, -x07, -x08, -x09) eumulate the 24 instructions in the Commercial Instruction Set (CIS).

The Control chip numbers consist of several field. The "303" indicates the generic type of F-11 chip, in this case the Control chip. The "x" indicates the latest shippable revision. The final two numbers designate the microcode contents.

B.4 MMU CHIP

The Memory Management Unit (MMU) performs two unrelated activities; not only does it provide the memory management function, but also contains the necessary registers for FP11 floating point execution. In the first capacity, the chip provides dual mode capability (User Instruction Space, Kernel Instruction Space), for address relocatin of a 16 bit virtual address to a 18 or 22 bit physical address. The MMU chip contains the required error detection logic to provide sophisticated memory protection features such as read/write access control, and page length limits. In addition all the necessary memory management data registers are in this chip. In the second capacity, the MMU chip merely provides the 366 16-bit registers needed for operand storage, status information storage, and scratchpad areas during floating point operations.

B.5 CHIP SYSTEM ARCHITECTURE

The signals that the chips use for communication among themselves and to the external world can be broken into three categories; DAL, MIB, and discrete signals. The DAL(Data Address Line) and MIB (Microinstruction Bus) busses are time-multiplexed, bi-directional interconnects. They are time-multiplexed in that they each have two independent definitions during the clock cycle. During the first half of a cylcle, clock-high time, the DALs are used to transfer data to or from the chips. This data could be a PDP-11 macroinstruction, a 16 bit physical or virtual addrress, or some form of numerical data. During the last half of a cycle, clock-low time, the DALs are used to transfer a relocated address or service information.

The relocated address is the 18 or 22 bit translation of the 16-bit virtual address which was driven by the Data chip during the preceding clock-high time. Service is both synchronous and asynchronous system level status information which is drive by the system interface; it is composed of interrupts, error conditions, power conditions and halt. The service information is latched in the Control chip where it is used to direct microcode flow. The MIB is similarly time-multiplexed. During clock-low time, a new microinstruction travels along the MIB from the active Control chip to the Data chip and all other Control chips. During the next clock-high time, the Data chip generates control information based on this microinstruction and transmits control information on the MIB. The discrete signals are dedicated lines, each with a specific meaning; that is, they are not time-multiplexed.

TABLE 1.1

FUNCTIONAL SUMMARY OF F-11 COMMUNICATION BUSSES

	Clock-high Time	Clock-low Time
DALs:	a macroinstruction, a 16 bit physical or virtual address, or numberical data	a relocated address, or service data
MIB:	control information	a microinstruction

B.6 F-11 CHIP SET INTERCONNECT

The following sections give a chip by chip description of the F-11 chip set interconnect.

B.7 DATA CHIP PINOUT

B.7.1 Clock (CLK)

The data chip uses a single high voltage clock to control internal functions and synchronize them to thee external environment.

B.7.2 Data/Address Lines (DAL<15:0>

The DAL's are used during clock-high time to transfer data-address information to or fromthe data chip. This is achieved by the execution of an Input, Output, or Address microinstruction. During address relocation cycles, the DALs are used during clock-low time to input the relocated address to decode for a possible explicit reference to the PS (bus address equal to 777776 in 18-bit mode or 17777776 in 22-bit mode).

B.7.3 Microinstruction Bus (MIB)

The MIB receives microinstructions from the active Control chip on lines <15.0>. The microinstructions are valid during clock-low time and are latched on the rising edge of clock.

During clock-high time, MIBs <15.0> convey the following control information.

a. -MME (MIB <15>)
-MME is latched on the falling edge of clock. It is the only MIB conotrol signal not generated by the Data chip, It is asserted low by the MMU or external logic to indicate that the virtual address being transmitted by the data chip will be relocated during the last half (clock-low time) of the address cycle. Accordingly, the data chip turns on its DAL receivers during clock-low time to input the externally generated address. The new address is evaluated by the PS explicit addressing logic to check for an explicit reference to the PS. The -MME input is ignored during non-address cycles.

b. -INITF (MIB <14>)
The low assertion ofthis signal is intended to be used for system initialization. Its activation is microcode controlled. The current microcode asserts -INITF during the power up sequence and when executing the PDP-11 RESET macroinstruction in Kernel mode. The length of its assertion is 42 x (clock period). It is then cleared for 366 x (clock period).

c. Interrupt Acknowledge (IAK) (MIB <13>)
When this signal is asserted high, it indicates that the data chip is servicing an interrupt. External logic should place a vector in the DALs before the end of the clock-high cycle in which this signal is asserted. Under microcode control the Data chip then pushes the old PC and PS onto the stock and reads the new PC and PS at the vector location.

d. Branch Allow (-BA) (MIB <11>)
When low, this signal indicates to the control chip(s) that the specified branch conditions for a conditional Jump microinstruction have been satisfied. This signal is generated by comparing the branch conditions specified in the microintruction to the PS or ALU status. The set of status flags chosen is determined by the particular Jump microinstruction executed (Data chip spec section 3.3.9.6). Note that the logic which generates this signal compares the existing branch conditions to the microinstruction every cycle. This is done no matter what particular microinstruction is being executed. Hence, the control chip(s) must decode when a conditional Jump is being executed to know when to sample -BA.

e. -SYNC (MIB <7>)
This signal goes low at the beginning of an I/O cycle and high at the conclusion of the cycle. That is, it goes low when an address is on the DALs, during the execution of an Address microinstruction. It then returns high after the data transfer has taken place, during the execution of the concluding Input or Output microinstruction. This signal is needed by the MMU and can be used externally for bus control.

f. Address/Input/Out (AIO<2:0>) (MIB <12,9,8>)
This 3-bit encoded Output gives the following information about the present microinstruction cycle:

TABLE 5.1

AIO <2:0> CONTROL CODES

AIO <2:0>	Name	Microinstruction Invoked by	Indicates that:
0	AWO	Address	a data write only operation to follow. (During this cycle an address is transmitted by the Data Chip.)
1	ARW	Address	data read/modify/write operation to follow. (During this cycle an address is transmitted by the Data Chip.)
2	-	Unused	
3	ARO	Address	a data read only operation to follow. (During this cycle an address is transmitted by the Data chip.)
*4	DOUTB	Output	the Data Chip is performing a byte output operation. (During this cycle a byte of data is transmitted by the Data chip.)
*5	DOUT	Output	the Data Chip is performing a word output operation. (During this cycle a word of data is transmitted by the Data Chip.)
*6	DIN	Input	the data chip is performing an input operation. (During this cycle a word of data is read by the Data Chip.)
7	NOP	all except Address, Input, or Output	no Data chip I/O operation is performed.

*These control signals are forced to NOP if an Input or Output microinstruction is executed when the explicit PS address mode is active (section 2.4), i.e., if the PS is being referenced by its I/O address, the Input and Output AIO codes are overriden to the NOP AIO code. This information is needed by the Control and MMU chips, and can be used for external bus control.

g. Next Address Test (-NATEST) (MIB<10>)
This signal is for test purposes only. The Data chip holds it continually disabled (-NATEST=1) to ensure normal operation of the Control chip(s).

h. X-Control (X<2:0>) (MIB<6:4>)
The 3-bit X-Control output encodes the following information for use by the control chip(s) and the MMU:

TABLE 5.2

X Control Codes

X <2>	X <1>	X <0>	CODE	NAME	FUNCTION
0	0	0	0	XLRTS	Load Subroutine Return Information
0	0	1	1	ZLIR	Load Instruction Register
0	0	0	2	XSOV	Stack Overflow
0	1	1	3	XUMDE	Not Kernel
1	0	0	4	XLPSPT	Load PS <7:4>
1	0	1	5	XLPSP	Load PS <7:5>
1	1	0	6	XLPST	Load PS <4>
1	1	1	7	-	No operation.

i. General Purpose Outputs (GPO <3:0>) (MIB <3:0>)
The general purpose outputs are for use in external system control. They are activated by bits in the control field, bits <7:4>, of the INPI and OUTI microinstructions.

microinstruction bit <4> set GPO 0 (MIB <0>) = 1

microinstruction bit <5> set GPO 1 (MIB <1>) = 1

microinstruction bit <6> set GPO 2 (MIB <2>) = 1

microinstruction bit <7> set GPO 3 (MIB <3>) = 1

The following definitions describe GPO codes used in the current microcode.

TABLE 5.3

Currently Defined GPO Codes

GPO 3 2 1 0
(INPI activated codes)

1 0 0 0 - input power up and HALT instruction option;(microcode acronym- OPTION)

0 0 0 1 - turn on runlight after instruction fetch and after echoing character in micro-ODT; (microcode acronym- RUNLIGHT)

(OUTI activated codes)

0 0 1 1 - toggle halt line flip-flop on CPU board during micro-ODT, activated by the ODT "H" command; (microcode acronym- HALTTOG)

0 1 0 1 - clear the event line clock interrupt; (microcode acronym- CLRCLK)

0 1 1 0 - clear the power fail interrupt;

(microcode acronym- CLRPWR)

0 1 1 1 - latch upper 2 bits of 18-bit address during micro-ODT, these 2 bits come from the Data chip (A,2,9); (microcode acronym-ADDR18)

B.7.4 Bank Select I/O (BSIO)

BSIO is driven every clock-high time. Its state is determined by "ANDing" DALs <15:13>. Thus during address cycles, an active signal on this line indicates a reference to the I/O page, the top 4K words of the memory address space. When an address relocation is occuring, the MMU will assert this line, if appropriate, during clock-low time. It is used to check for an explicit reference to the PS. BSIO can be used externally for bus control, however, it should only be examined during address microcycles since the BSIO output driver is turned on indiscriminately every time the DAL drivers are enabled.

B.7.5 Disable Drivers (-DD)

This active-low input is used for testing purposes only. For normal operatinos it is tied inactive high with an internal pull-up resistor.

B.8 CONTROL CHIP PINOUT

B.8.1 Clock (CLK)

The Control chip uses a single high voltage clock to control internal functions and to synchronize them to the external environment.

B.8.2 Data/Address Lines (DAL <15:0>)

The DALs are the inputs to the PIR and the Control chip PS data. During appropriate clock-high intervals, they used to load a macroinstruction into the PIR or data into the Interrupt Priority and T-bit latches (sectin 3.4), during appropriate clock-low times, they are used to load service information. Following are the definitions of the service inputs in the order of priority in which they are handled, (highest to lowest):

DAL Input	Service Definition		Active State
<0>	*DCOK:	Power-up signal	1
<4>	*CTLERR:	No active Control chip	1
<3>	*MMUERR:	Memory Management abort	0
<1>	*BUSERR:	System Bus timeout error	1
<2>	*PARERR:	PARITY ERROR	1
<6>	*SPARE:	Causes an Unconditional Jump to Control chip 3 locoation 76	0
**	TBIT:	T-bit set	
**	STKOVF:	Stack overflow	
<7>	PWRF:	Power-fail signal	1
<8>	INTRQ7:	Priority Level 7 interrupt request	1
<12>	EVENT:	Event line interrupt request (priority level 6)	1
<9>	INTRQ6:	Priority Level 6 interrupt request	1
<10>	INTRQ5:	Priority Level 5 interrupt request	1
<11>	INTRQ4:	Priority Level 4	1

		interrupt request	
<5>	HALT:	Halt request	1
**	WAIT:	Wait request	

* Asynchronous abort-causing service inputs, i.e. these conditions must externally generate RESET (section A.2.5).

** These service bits are generated inside the Control chips. They are the outputs of the respectively named internal flip-flops.

B.8.3 Microinstruction Bus (MIB <15.0>)

During clocklow time the active control chip drives the MIB with a microinstruction. All Conrol chips, active and inactive, concurrently use the MIB to input and latch the microinstruction. The latched microinstruction is used in all Control chips to initiate various internal activities. These activities inclIude loading the PIR with a macroinstruction, updating the internal PS information, or in the case of Unconditional Jump microinstructions, enabling the Chip Select Logic.

During clockhigh time control information is received on the MIB from the Data chip. It is latched on the falling edge of clock. The signals used by the Control chips are Interrupt Acknowledge (IAK) (MIB <13>), Branch Allow (-BA) (MIB <11>), Next Address Test (-NATEST) (MIB <10>), Address/Input/Output (AIO <1>) (MIB <9>), and X-Control (X <2:0>) (MIB <6:4>). The control information is defined in the Data chip pinout (section 5.1.3).

B.8.4 RESET

This initialization input is used by power up logic and abort logic to start the microprogram at a known point. RESET activates the chip select logic of control chip "0" and disables all other Control chips. In addition, it causes service information to be loaded into all Control chips. The microcode routine associated wtih the highest priority active service bit is then executed.

B.8.5 Disable MIB Drivers (-DMIB)

The - DMIB input is used for chip testing only, it is tied inactive high for normal operation with an internal pull up resistor.

B.8.6 Chip Select (-CSEL)

This control signal is held low by the active Control chip to indicate that it is currently selected and controlling the microprogram sequence. -CSEL is a common line drive by each Control chip with an open drain buffer, the non-assertion of this signal indicates that no chip is currently selected; in other words, a Control chip error (CTLERR). This condition should be used by external logic to generate the RESET signal which will restart the microprogram at known point (section A.2.6).

B.9 MMU CHIP PINOUT

B.9.1 Clock (CLK)

The MMU uses a single high voltage clock to control internal functions and to synchronize them to the external environment.

B.9.2 Data/Address Lines (DAL <21:0>)

The DALs are the input/output to the register files. For floating point operations and explicit PAR/PDR references, data and addresses are transmitted and received on these lines during clock-high time. For address relocatin cycles, the virtual address is received on the DALs <15:0> during clock-high time. The physical address is then transmitted during the following clock-low time on DALs <17:0> for 18-bit addressing and <21:0> for 22-bit addressing. DALs which are unused during an output operation are forced to zero. For example DALs <21:17> would be zero for the reading of a floating point register or an explicit PAR/PDR reference.

B.9.3 Microinstruction Bus (MIB <15,12,9:4>)

During clock-low time the MMU receives microinstruction bits <12,9:4> from the active Control chip. They are latched on the rising edge of CLK. These bits are used for internal control. During clock-high time control signals are received from the date

chip and latched on the falling edge of CLK. The signals used by the MMU are Address/Input/Output (AIO <2:0>) (MIB <12,9,8>), Sync Flip-Flop (-SYNCF) (MIB <7>), and X-Control (X <2:0>) (MIB <6:4>). In addition the MMU asserts Memory Management Enable (-MME) (MIB <15>) during clock-high time.

B.9.4 Bank Select I/O (BSIO)

The assertion of BSIO indicates a reference to the I/O page, the top 4K words of the address space. During clock-high time this line is an input. It is needed to decode for explicit I/O bus references to the internal MMU registers. During clock-low time it is an output. The assertion of BSIO during relocated address cycles again indicates a reference to the I/O page. It is used during clock-low time by the MMU and the Data chip to check for an explicit reference to an internal register.

B.9.5 Disable MMU Slave (-DMMUS)

Assertion of this active low input when the clock is high disables the DALs <15:0>, BSIO, and -ABORT output drivers for the remainder of the cycle. External logic cann use this line to block internal register accesses roo to suppress relocation outputs. It also informs the MME logic that any ensuing relocation operation is being externally controlled, this is necessary when 18-bit ODT is implemented (section A.2.9)> In order to insure normal operation when such external logic is not active, the -DMMUS input is internally driven inactive high when the clock is low, and then sustained by an internal pullup-register.

B.9.6 ABORT

This output becomes active low during clock-low time of an internally active relocation cycle if an abort condition occurs as a result of a page violation, a write violation, or a page length error. The purpose of this signal is to reset the Control chip(s) and to generate an -MMUERR service signal. This signal can also be used to reset external bus control logic. Note, however, that this output is blocked when the -DMMUS input is asserted. The -ABORT output is perecharged to its inactive high state at the beginning of the next cycle (when the clock goes high), and the sustained by an internal resistor.

B.9.7 MMU Reply (-MREPLY)

The MMU chip asserts the active-low - MREPLY output during I/O cycles (when the clock is high) in response to DIN, DOUT, our DOUTB signals (see AIO <2:0> inputs) if a PAR/PDR or Status Register in the MMU chip is explicitly addressed by its physical bus address. It can be used to satisfy system bus I/O protocol. Note in section A.3.1 that the maximum specified access time for -MREPLY is greater than the minimum specified clock-high time. Accordingly, external logic may have to extend the clock-high period during bus cycles until -MREPLY is received. The -MREPLY output is internally precharged to its inactive high state when the clock goes low, and then sustained by an internal resistor.

B.9.8 -External Bus Map Enable (-UBMAP)

The -UBMAP output signal is derived from control bit <5> in Status Register 3. The open-drain output buffer pulls the -UBMAP signal low if SR3 (5) is set. An external resistor must be supplied to pull this signal back up.

B.10 CHIP INTERACTION

B.10.1 F-11 Chip Set Configurations

The intent of this section is to clarify how the three different types of F-11 chips work together and what their combined abilities yield. First the basis combination of Data chip and Control chip will be presented. Then memory management will be examined. Finally, floating point and CIS operation will be discussed.

B.10.2 Data Chip/Base Control Chip Interaction

The Data chip/Base Control chip combination relizes a multichip PDP-11 microprocessor with expandable microstore. The data chip/Control chip relationship is basically that of ALU and sequencer. The Data chip is an ALU and controller. The Control chip is a microsequencer and microstore. Figure 6.1 illustrates the overall operation of the Data/Control chip combination. The following sequence gives an overview of their operation.

1. Fetch macroinstruction: both the Data and Control chips receive and latch a macroinstruction.

2. Execute microsequence: the Control chip transmits a specific sequence of microinstructions to emulate or service the latched macro-information (a macroinstruction or service data); simultaneously the Data chip receives the microinstructions and performs the appropriate arithmetic, logical, and control functions.

3. Service: if a service condition has been asserted, the Control chip loads the service data and the chip set returns to step (2). If no service is pending the chip set returns to step (1).

In the present discussion just the Base Control chip is being considered; however, all the Control chips interact wtih the Data chip in the same manner. They differ only in what macro-level information they can decode.

The following detailed emulation of the CLR R0 macroinstruction will make the preceding process clearer.

CLK	Data Chip	Control Chip
	PART 1: MACROINSTRUCTION FETCH	
*low		transmit ADR2 PC micro-instruction on the MIB, (this instruction causes memory to be addressed by the value in the PC and the PC to be incremented by 2)
*	latch ADR2 PC micro-instruction from the MIB	
high	execute ADR2; put PC on the DALs, increment PC by 2, output control signals (AIO, X, -SYNCF, -BA, BSIO)	
v		latch control information from the MIB
low		transmit INPI micro-instruction on the MIB
*	latch INPI microin-struction from the MIB	

high	execute INPI microinstruction; input the CLR R0 macroinstruction from the DALs and output control signals (AIO, X, GPO, -SYNCF, -BA)	
v	latch macroinstruction from the DALs	latch macroinstruction from the DALs and control information from the MIB

*The terms in this column define the state of the clock line; low = clock voltage at a low, high = clock voltage at a high, ^ = a rising edge, v = a falling edge.

PART 2: MACROINSTRUCTION EMULATION

low		transmit CLR microinstruction on the MIB
^	latch CLR microinstruction from the MIB	
high	execute CLR microinstruction; clear the register which is specified in the latched macroinstruction and transmit control signals (AIO, X, -SYNCF, -BA) on the MIB	
v		latch the control information from the MIB

The chip set now goes to service. On the next rising edge the macro level service data is clocked into the Control chip from the DALs. If no service is being requested, the machine falls through to the macroinstruction fetch sequence. If service is being requested, the appropriate trap sequence is executed.

B.10.3 Memory Management Interaction

When the mmu is used for memory management, it supplies the resources for 18 and 22 bit addressing. When it is added to the Data chip/Control chip(s) system, one additional communication situtation arises. This situation is when an address needs to be relocated. The interaction is depicted in the following sequence:

Relocated Address Interchip Communication

CLK	Data Chip	MMU Chip
high (during execution of an address microinstruction)	Output the address on the DALs and BSIO, output the proper AIO code (as well as all other codes) on the MIB	detect an address cycle by looking at the AIO codes, assert -MME (MIB 15) to indicate a relocated address cycle and input the address from the DALs and BSIO
v	latch -MME (this tells the Data chip that the MMU is relocating an address)	
low		generate the relocated address and transmit it on the DALs and BSIO
^	latch the relocated address from the DALs and BSIO to check for an explicit reference to the PSW	latch the internally relocated address to check for an explicit reference to a PAR, PDR, or status register

If memory management is disabled, address cycles proceed as all othe microcycles. If it is enabled, it adds three changes to the address cycles of the basic chip interaction. First of all, it changes when the address on the DALs is valid. Instead of being valid towards the end of clock-high time, the address must be examined towards the end of clock-low time. Secondly, clock-low time must be lengthened from the minimum spec value for nonrelocated cycles. This is required to allow the MMU time to operate. Thirdly, service can't be on the DALs at the same time as a relocated address. Due to the way in which it is microprogrammed, the chip set will not load service data during a relocated cycle. In addition, the system interface must not drive the DALs with service data at this time.