2040 OS-9 Computer
4200-0152
Version 1.0

Table of Contents

SECTION 1 SYSTEM OVERVIEW
2040 OS9 COMPUTER FEATURES 7
SYSTEM MEMORY MAP 10
Board Address/Data Type Memory Map 10
SYSTEM INTERRUPT TABLE 10
EXTERNAL CONNECTORS 11
Testhead Connection 11
Emulator Connection 12
GPIB Connection 13
Keypad Connection 14
Printer Connection 14
EIA561 Connections (T1...T5, Terminal) 15
AUI Ethernet Connection 15
Twisted Pair 802.3 (Ethernet) 17
INTERNAL CABLES 17
GPIB 17
TBus/EMBus Data Cable 11
CPU I/O Cable 18
SCSI Cable 19
Terminal Cable 19
AUI Ethernet Cable 19
DC POWER DISTRIBUTION 20
DC Terminal Block 20
DC Harness A/B (9 Inch), C (13 Inch), D (17 Inch) 20
DC Harness E, +/12V (13 Inch) 21
AC HARNESS 22

SECTION 2 VME MOTHER BOARDS
VMEbus J1 BACKPLANE 23
Description 23
Placement of Jumpers 24
J1 Connector Pin Assignments 24
VMEbus J2 BACK PLANE 25
Description 25
J2 Connector Pin Assignments 26

SECTION 3 COMPUTER CARD
DESCRIPTION 27

SECTION 4 I/O CONTROLLER/MEMORY CARD
DESCRIPTION 29
2040 Serial/Flash/EPROM 29
VME BUS INTERFACE 29
General 29
Addressing 29
LOCAL ADDRESS MAP 30
EEPROM Memory 30
Flash Voltage Programming 30
Hardware Timer 30
Parallel Port #1 30
Parallel Port #2 (Keypad Operator Interface) 31
Serial Ports 31
EPROM Memory 31
Flash/EPROM Memory 31
P1 CONNECTOR PIN ASSIGNMENTS 32
P2 CONNECTOR PIN ASSIGNMENTS 33
INITIAL SWITCH SETTINGS 34
JUMPER SETTINGS 34

SECTION 5 TESTHEAD CONTROLLER/GPIB CONTROLLER CARD
DESCRIPTION 37
2040 Testhead Controller 37
VME BUS INTERFACE 37
General 37
Addressing 37
Local Address Map 38
Vector Registers 38
EEPROM 38
GPIB Controller 38
Testhead and Emulator Accesses 39
P1 Connector Pin Assignments 40
P2 Connector Pin Assignments 41
J1 Connector Pin Assignments (Front Panel GPIB) 42
JUMPER SETTINGS 42
JP1 through JP8 42

JP9 through JP15 42
JP16 through JP18 42
JP19 through JP21 43
JP22 through JP28 43

SECTION 6 J2 ADAPTER CARD
DESCRIPTION 45
2040 MVME147 J2 Breakout Assembly 45
CONNECTOR PIN ASSIGNMENTS 46
VME J2 Back Plane Connector P1 46
SCSI Port J6 47
Ethernet Port J7 47
Parallel (Printer) Port J3 48
Serial Port J5 (Port #1, /Term) 48
Serial Ports J4 (Port #3, /t5), J1 (Port #4, /t6), J2 (Port #2, /t7) 48

SECTION 7 MASS STORAGE DEVICES
DESCRIPTION 49
HARD DRIVE 50
Description 50
Replacement and Configuration 50
Disassembly 50
Assembly 51
FLOPPY DRIVE 51
Description 51
Replacement and Configuration 51
Disassembly 52
Assembly 52
TAPE DRIVE 53
Description 53
Replacement and Configuration 53
Disassembly 53
Assembly 54

SECTION 8 PERIPHERAL DEVICES
TERMINAL AND ASSOCIATED CABLING 55
PRINTER AND ASSOCIATED CABLING 56

SECTION 9 SOFTWARE
OS9 DRIVERS AND DESCRIPTORS 57
HARD DISK DIRECTORY STRUCTURE 58
/NV0 DIRECTORY STRUCTURE 59
OS9 BOOT SEQUENCE 59
GPIB SOFTWARE 60
IFC/REN 61
Hexadecimal Conversion 62
GPIB Commands 62
GBIP Address 63
Controller Protocol 63
Time Out 64
Terminator/EOI 64
XMode 64
Slave Operation 65
USING THE IEEE 488 FROM BASIC 66
Opening A “path” 66
Sending GPIB Commands: TALK, LISTEN 66
Sending, Receiving Data 67
Closing The Path 67
Handling Errors 67
System Configuration 69
Examples 70
COMMON PROBLEMS 71

APPENDIX A
/AmsInt 73
/d0 74
/d0h 75
/g 76
/h0 77
/h0fmt 78
/mt0 79
/p 80
/pps 81
/s0 82
/s1 83
/t1 84
/t2 85

/t3 86
/t4 87
/t5 88
/t6 89
/term 90
/time 91

Section 1 SYSTEM OVERVIEW

2040 OS9 COMPUTER FEATURES:

- 5 slot VME motherboards, supporting A and B size VME cards
- 200 watt power supply
- Motorola MC68030 microprocessor, clock speed: 32 MHz
- 8 to 32 Mb of “on board” DRAM
- Floating point coprocessor
- SCSI bus for mass storage peripheral communication
    - 240 Mb hard drive, 12 ms access time
    - 600 Mb tape backup option
    - 2 Mb floppy disk drive
- 2040 Testhead controller providing optoisolated communication
- GPIB controller
- IEEE 802.3 (Ethernet) interface
- Parallel printer port
- Six RS232 serial ports standard (two more optional)
- Keypad interface
- OS9 Version 2.4 operating system

The 2040 OS9 computer system is located in the lower left corner of the front of the 2040 test system. It has all of the capabilities needed to program and control the resources of the 2040 tester. In the standard configuration, 3 of the 5 slots of the motherboards are used. The three cards that occupy these slots are the Motorola MVME147 computer card, the Digalog I/O controller/memory card, and the Digalog Testhead Controller/GPIB Controller card. The slots that the cards occupy are fixed, in that inputs and outputs of each card have to correspond to the connections used on the J2 VME motherboard. Slot one is used for the Motorola MVME147 computer card, which contains the DRAM, SCSI controller, 4 serial ports, Ethernet interface controller and the MC68030 microprocessor.

Slot two contains the I/O controller/Memory card, which adds 4 more serial ports, a parallel printer port, 2 Mb of flash programmable ROM, and 2 Mb of EPROM.

Slot 3 contains the Testhead/GPIB controller, which provides control capability for the 2040 Testhead, the Digalog Emulator and GPIB devices. Communications to the Testhead and the Emulator are opto isolated.

Slot 4 of the computer system is reserved for use with a second Testhead controller for those systems that use both an Emulator and the analog Testhead.

Slot 5 of the computer system is open for expansion.
The computer system uses Microware Systems OS9 operating system. This real time operating system is multiuser and multitasking, and supports both Microware BASIC and the Microware Ultra C compiler as test development tools. Software modules called “functional calls” have been developed by Digalog Systems to enable the use of these development tools to create test programs. Functional calls are described in the 2040 OS9 Programming Manual.

In addition to the functional calls, Digalog has developed or tested all drivers for the other devices listed above. These drivers and their associated descriptors relieve the users from having to concern themselves with configuring hardware or installing software. Descriptors are software modules that describe a device that driver software needs to talk to. They serve as labels or names of devices that a user needs to access that device. A list of descriptors is provided in the software section of this manual. The factory settings for these are listed in Appendix A. For more information on the operating system, see the Microware manual set provided with the tester.

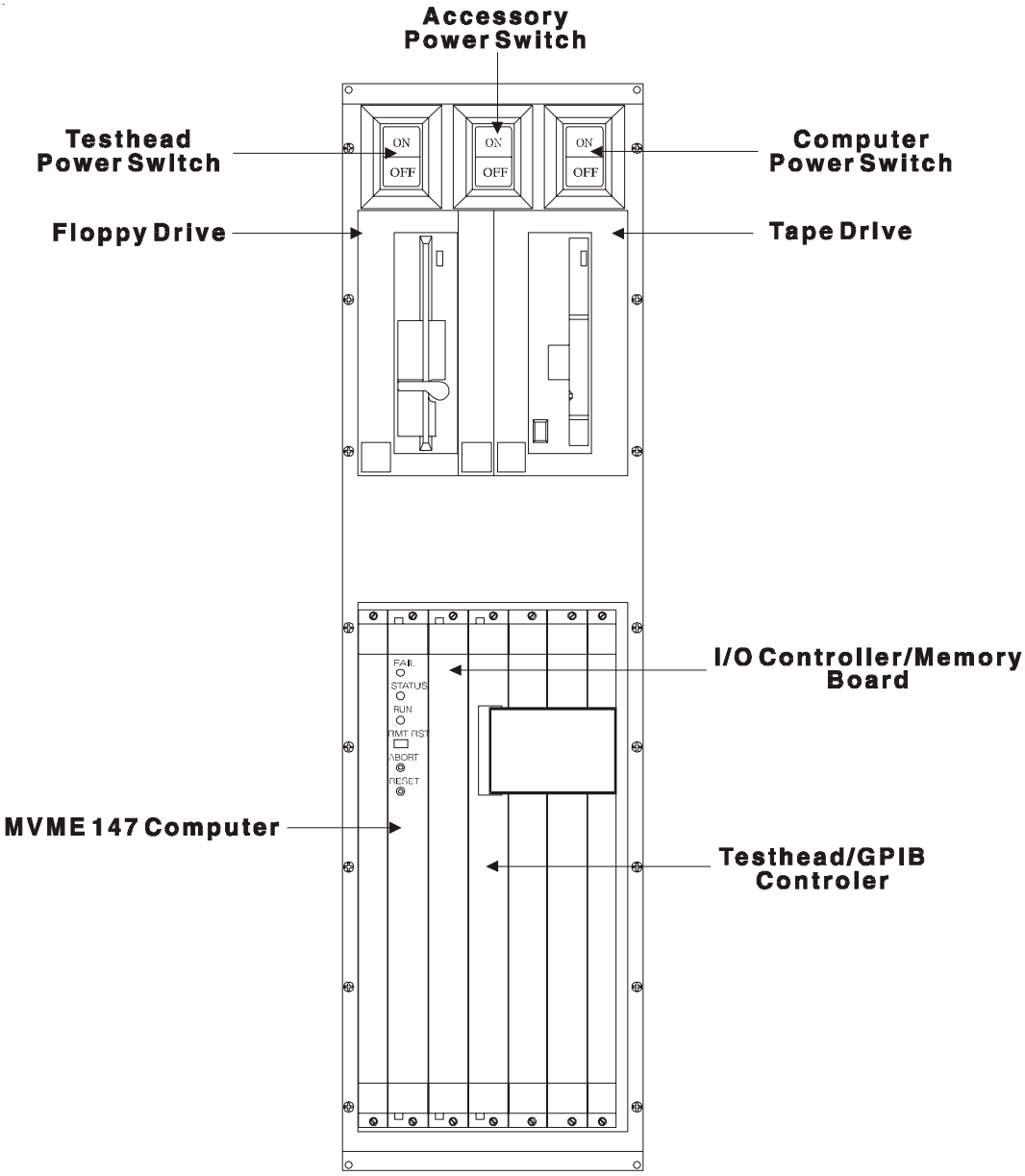

Accessory
Power Switch
Testhead
Power Switch
Computer
Power Switch
ON
OFF
ON
OFF
ON
OFF
Floppy Drive
Tape Drive
FAIL
STATUS
RUN
RMT RST
ABORT
RESET
I/O Controller/Memory
Board
MVME 147 Computer
Testhead/GPIB
Controler

Figure 1

SYSTEM MEMORY MAP

The following table lists the memory map of the 2040 OS9 computer. Except for the unused segments all other segments are fixed in software and should not be changed.

START ADDRESS	END ADDRESS	USED FOR
$00000000	$00FFFFFF	DRAM
$01000000	$E07FFFFF	Unused
$E0800000	$E0FFFFFF	Digalog I/O Controller/Memory
$E1000000	$F0FAFFFF	Unused
$F0FB0000	$F0FBFFFF	Digalog Analog Testhead (A24/D16)
$F0FC0000	$F0FCFFFF	Digalog Digital Testhead (A24/D16)
$F0FD0000	$FF7FFFFF	Unused
$FF800000	$FF81FFFF	ROM Bank 0: OS9
$FF820000	$FF9FFFFF	ROM Bank 1: Unused
$FFA00000	$FFBFFFFF	ROM Bank 2: Unused
$FFC00000	$FFFDFFFF	Reserved
$FFFE0000	$FFFE4FFF	Local I/O Devices
$FFFE5000	$FFFEFFFF	Reserved
$FFFF0000	$FFFFFFFF	Unused

Board Address/Data Type Memory Map

START ADDRESS	END ADDRESS	BOARD ADDRESS/DATA TYPE
$01000000	$EFFFFFFF	A32/D32
$F0000000	$F0FFFFFF	A24/D16
$F1000000	$FF7FFFFF	A32/D16
$FFFF0000	$FFFFFFFF	A16/[D80,D8E, D16]

SYSTEM INTERRUPT TABLE

DEVICE	INTERRUPT	VECTOR
Digalog Analog Testhead AMS	7	90
MPU Abort	7	
MPU AC Fail	7	
MPU Bus Error	7	
MPU DMA	07	
MPU LAN	07	
MPU Printer Port	07	
MPU SCSI Port	07	
MPU Serial Ports	07	
MPU Software 1, 2	07	
MPU Timer 1, 2	07	

MPU Utility WPERR	7	
MPU Utility SYSFAIL	6	
MPU Utility SIGHP	5	
MPU Utility LM1	4	
MPU Utility IACK	3	
MPU Utility LM0	2	
MPU Utility SIGLP	1	
Digalog Analog Testhead timer	6	92
Network I/F	5	68
Hard Drive	4	69
Floppy Drive	4	69
Tape Drive	4	69
Parallel Port	3	98
Serial Ports T1, T2	3	94
Serial Ports T3, T4	3	95
Serial Ports T5, T6	3	89
Terminal	3	98
GPIB	2	93

EXTERNAL CONNECTORS

Testhead Connection

The Testhead is connected to the OS9 computer via an external cable from the rear of the assembled computer unit. It is identified as the lower of two 64pin DIN connectors, halfway down the back of the upright unit. The pin listings are as follows:

Pin	Name	Description	Pin	Name	Description
A1	TGND	TH Reference	C1	TE*	Write Strobe
A2	TGND	TH Reference	C2	TR/W*	Read/Write
A3	TGND	TH Reference	C3	TINT*	Interrupt
A4	TGND	TH Reference	C4	TEN*	Enable
A5	TGND	TH Reference	C5	TRESET*	Reset
A6	TGND	TH Reference	C6	TD0*	Data Line
A7	TGND	TH Reference	C7	TD1*	Data Line
A8	TGND	TH Reference	C8	TD2*	Data Line
A9	TGND	TH Reference	C9	TD3*	Data Line
A10	TGND	TH Reference	C10	TD4*	Data Line
A11	+5T	TH Power	C11	TD5*	Data Line
A12	+5T	TH Power	C12	TD6*	Data Line
A13	+5T	TH Power	C13	TD7*	Data Line
A14	+5T	TH Power	C14	TDS1*	Data Strobe
A15	+5T	TH Power	C15	TA1	Address Line

Pin	Name	Description	Pin	Name	Description
A16	+5T	TH Power	C16	TA2	Address Line
A17	+5T	TH Power	C17	TA3	Address Line
A18	+5T	TH Power	C18	TA4	Address Line
A19	+5T	TH Power	C19	TA5	Address Line
A20	TA7	Address Line	C20	TA6	Address Line
A21	TA9	Address Line	C21	TA8	Address Line
A22	TA11	Address Line	C22	TA10	Address Line
A23	TGND	TH Ground	C23	TA12	Address Line
A24	NEWTEN*		C24	TADRS*	Address Enable
A25	TDS0*	Data Strobe	C25	TGND	TH Ground
A26	8/16*		C26	TDTACK*	Data Ack
A27	TD9*	Data Line	C27	TD8*	Data Line
A28	TD11*	Data Line	C28	TD10*	Data Line
A29	TD13*	Data Line	C29	TD12*	Data Line
A30	TD15*	Data Line	C30	TD14*	Data Line
A31	TGND		C31	TSPAREOUT	Spare output
A32	TGND		C32	TSPAREIN	Spare input

Emulator Connection

The Emulator is connected to the OS9 computer via an external cable from the rear of the assembled computer unit. It is identified as the upper of two 64pin DIN connectors, halfway down the back of the upright unit. The pin listings are as follows:

Pin	Name	Description	Pin	Name	Description
A1	TGND	Emulator Ground	C1	EWTSTB*	Write Strobe
A2	TGND	Emulator Ground	C2	ER/W*<T>	Read/Write
A3	TGND	Emulator Ground	C3	EINT*<T>	Interrupt
A4	TGND	Emulator Ground	C4	TEN*<T>	Enable
A5	TGND	Emulator Ground	C5	ERESET*<T>	Reset
A6	TGND	Emulator Ground	C6	ED0*<T>	Data Line
A7	TGND	Emulator Ground	C7	ED1*<T>	Data Line
A8	TGND	Emulator Ground	C8	ED2*<T>	Data Line
A9	TGND	Emulator Ground	C9	ED3*<T>	Data Line
A10	GND	Emulator Ground	C10	ED4*<T>	Data Line
A11	+5T	Emulator Power	C11	ED5*<T>	Data Line
A12	+5T	Emulator Power	C12	ED6*<T>	Data Line
A13	+5T	Emulator Power	C13	ED7*<T>	Data Line
A14	+5T	Emulator Power	C14	EUDS*<T>	Data Strobe
A15	+5T	Emulator Power	C15	EA0<T>	Address Line
A16	+5T	Emulator Power	C16	EA1<T>	Address Line

Pin	Name	Description	Pin	Name	Description
A17	+5T	Emulator Power	C17	EA2	Address Line
A18	+5T	Emulator Power	C18	EA3	Address Line
A19	+5T	Emulator Power	C19	EA4	Address Line
A20	EA6	Address Line	C20	EA5	Address Line
A21	EA8	Address Line	C21	EA7	Address Line
A22	EA10	Address Line	C22	EA9	Address Line
A23	TGND	Emulator Ground	C23	EA11	Address Line
A24	NEWTEN*		C24	EVEA*	Address Enable
A25	ELDS*	Data Strobe	C25	TGND	Emulator Gnd
A26	8/16*		C26	EDTACK*	Data Ack
A27	ED9	*Data Line	C27	ED8*	Data Line
A28	ED11	*Data Line	C28	ED10*	Data Line
A29	ED13	*Data Line	C29	ED12*	Data Line
A30	ED15	*Data Line	C30	ED14*	Data Line
A31	TGND		C31	TSPAREOUT	Spare output
A32	TGND		C32	TSPAREIN	Spare input

GPIB Connection

A Standard IEEE488 (GPIB) connection is available on the rear of the computer unit. It is located halfway down the back of the unit, below the Testhead and Emulator external connectors. The standard pin listing follows. Please see the IEEE488 specification for detailed descriptions.

Pin	Name	Pin	Name
1	DIO1	13	DIO5
2	DIO2	14	DIO6
3	DIO3	15	DIO7
4	DIO4	16	DIO8
5	IO	17	REN
6	DAV	18	GND (DAV)
7	NRFD	19	GND (NRFD)
8	NDAC	20	GND (NDAC)
9	IFC	21	GND (IFC)
10	SRQ	22	GND (SRQ)
11	ATN	23	GND (ATN)
12	GND	24	Signal Ground

Keypad Connection

A Digalog supplied keypad may be connected to the computer via a 15pin port on the lower half of the computer's rear panel. The pin listing for this port is listed below:

Pin	Name
1	CA2
2	CA1
3	PA7
4	PA6
5	PA5
6	PA4
7	PA3
8	PA2
9	PA1
10	PA0
11	Frame Ground
12	Ground
13	Ground
14	Ground
15	Ground

Printer Connection

A printer port is provided on the rear of the computer. The 36pin connector is found on the lower portion of the computer rear panel. The pin listing for this port is listed below:

Pin	Name	Pin	Name
1	STRB	19	GND
2	DAT1	20	GND
3	DAT2	21	GND
4	DAT3	22	GND
5	DAT4	23	GND
6	DAT5	24	GND
7	DAT6	25	GND
8	DAT7	26	GND
9	DAT8	27	GND
10	ACK	28	GND
11	BUSY	29	GND
12	NC	30	GND
13	NC	31	NC

Pin	Name	Pin	Name
14	NC	32	PE
15	NC	33	NC
16	NC	34	NC
17	Frame Ground	35	NC
18	NC	36	NC

EIA561 Connections (T1..T5, Terminal)

There are 6 serial ports available near the bottom of the computer's rear panel (see Figure 2 for their exact location). One of the ports (TERM) is used as the primary terminal connection. Their standard pin listing follows for each port. Please refer to EIA561 for pin descriptions. All ports used are DCE.

Pin	Name
1	CEn
2	DCDn
3	DTRn
4	GND
5	RXDn
6	TXDn
7	CTSn
8	RTSn

AUI Ethernet Connection

Ethernet connections may be made from a standard IEEE802.3 (AUI) Ethernet port found on the rear of the computer unit. This 15pin connector's pin listing follows. Please see the IEEE802.3 for a complete description of the listing.

Pin	Name	Pin	Name
1	NC		
2	C+	9	C
3	T+	10	T
4	NC	11	NC
5	R+	12	R
6	GND	13	+12VF
7	NC	14	NC
8	NC	15	NC

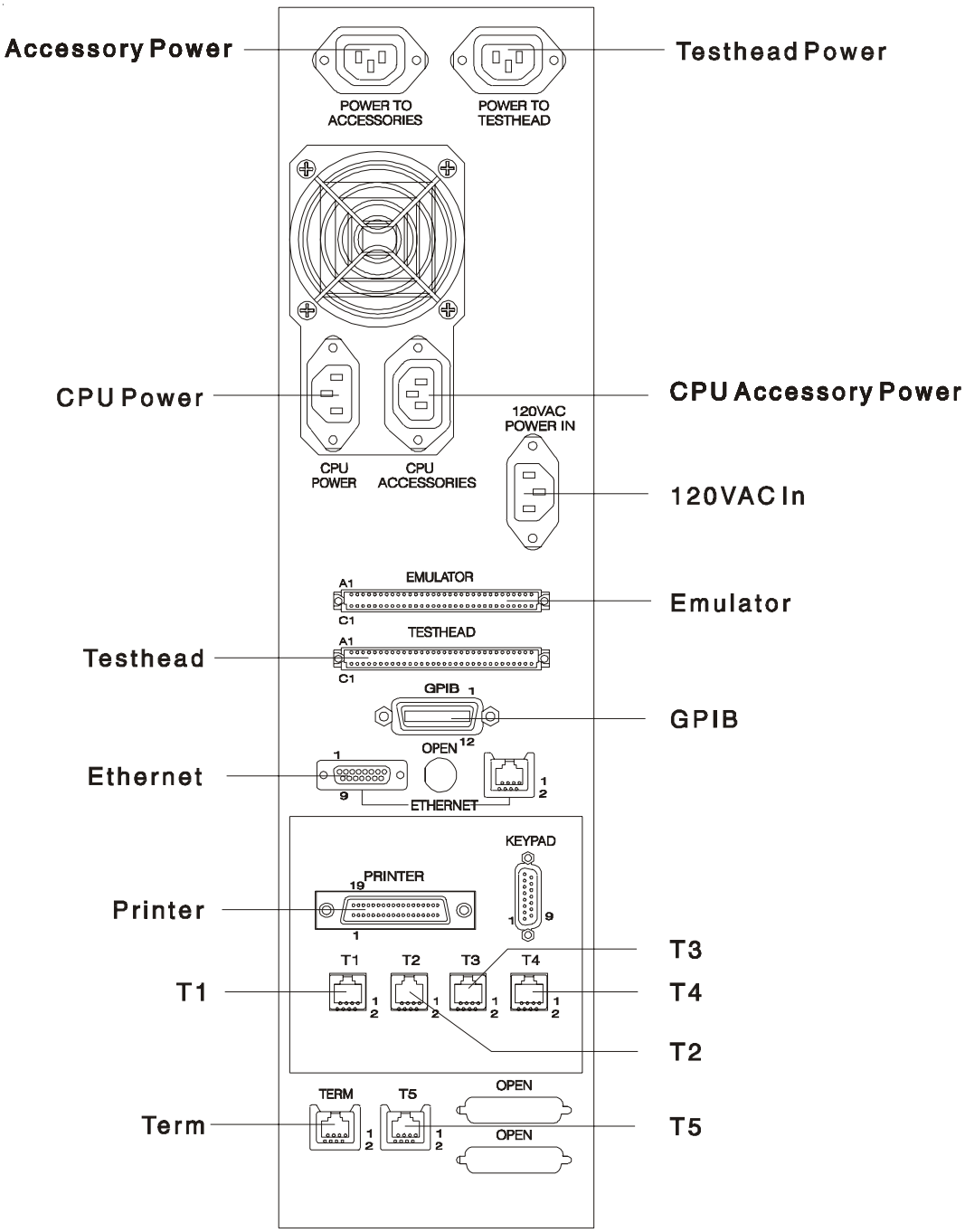

Accessory Power
Testhead Power
POWER TO ACCESSORIES
POWER TO TESTHEAD
CPU Power
CPU Accessory Power
120VAC POWER IN
CPU POWER
CPU ACCESSORIES
120VAC In
EMULATOR
Emulator
TESTHEAD
Testhead
GPIB
GPIB
OPEN
Ethernet
ETHERNET
KEYPAD
PRINTER
Printer
T1
T2
T3
T4
T3
T4
T2
TERM
T5
OPEN
Term
T5
OPEN

Figure 2

Twisted Pair 802.3 (Ethernet)

A twisted pair IEEE802.3 (AUI) connector may be found on Digalog computer units so equipped, which follows the industry standard governing such connections.

INTERNAL CABLES

GPIB

A GPIB cable routes the connection from the front of the Testhead controller card to the rear panel GPIB port.

Pin	Name	Pin	Name
1	DIO1	13	DIO5
2	DIO2	14	DIO6
3	DIO3	15	DIO7
4	DIO4	16	DIO8
5	EIO	17	REN
6	DAV	18	GND (DAV)
7	NRFD	19	GND (NRFD)
8	NDAC	20	GND (NDAC)
9	FC	21	GND (IFC)
10	SRQ	22	GND (SRQ)
11	ATN	23	GND (ATN)
12	GND	24	Signal Ground

TBus/EMBus Data Cable

There is one cable each for the Testhead and Emulator. The Testhead data cable runs from the third slot (from the left) on the J2 VME backplane to the back of the appropriate connector on the computer case's rear panel (the lower 64pin DIN connector.) The Emulator's data cable runs from the fourth connector on the J2 to the rear panel (the upper 64pin DIN connector.) The third and fourth slots (from the left) correspond directly to where a Testhead controller card would be installed and configured for a Testhead or an Emulator. The general pin listing for these slots follows is shown on the next page.

The "B" row of the motherboard connector is not used in the cable.

Pin	Name	Description	Pin	Name	Description
A1	TGND	TH Reference	C1	TE* (EWTSTB*)	Write Strobe
A2	TGND	TH Reference	C2	TR/W* (ER/W*)	Read/Write
A3	TGND	TH Reference	C3	TINT* (EINT*)	Interrupt
A4	TGND	TH Reference	C4	TEN*	Enable
A5	TGND	TH Reference	C5	TRESET* (ERESET*)	Reset
A6	TGND	TH Reference	C6	TD0* (ED0*)	Data Line
A7	TGND	TH Reference	C7	TD1* (ED1*)	Data Line
A8	TGND	TH Reference	C8	TD2* (ED2*)	Data Line
A9	TGND	TH Reference	C9	TD3* (ED3*)	Data Line
A10	TGND	TH Reference	C10	TD4* (ED4*)	Data Line
A11	+5T	TH Power	C11	TD5* (ED5*)	Data Line
A12	+5T	TH Power	C12	TD6* (ED6*)	Data Line
A13	+5T	TH Power	C13	TD7* (ED7*)	Data Line
A14	+5T	TH Power	C14	TDS1* (EUDS*)	Data Strobe
A15	+5T	TH Power	C15	TA1 (EA0)	Address Line
A16	+5T	Testhead Power	C16	TA2 (EA1)	Address Line
A17	+5T	Testhead Power	C17	TA3 (EA2)	Address Line
A18	+5T	Testhead Power	C18	TA4 (EA3)	Address Line
A19	+5T	Testhead Power	C19	TA5 (EA4)	Address Line
A20	TA7 (EA6)	Address Line	C20	TA6 (EA5)	Address Line
A21	TA9 (EA8)	Address Line	C21	TA8 (EA7)	Address Line
A22	TA11 (EA10)	Address Line	C22	TA10 (EA9)	Address Line
A23	TGND	TH Ground	C23	TA12 (EA11)	Address Line
A24	NEWTEN*		C24	TADRS* (EVEA*)	Address Enable
A25	TDS0* (ELDS*)	Data Strobe	C25	TGND	TH Ground
A26	8/16*		C26	TDTACK* (EDTACK*)	Data Ack
A27	TD9* (ED9*)	Data Line	C27	D8* (ED8*)	Data Line
A28	TD11* (ED11*)	Data Line	C28	TD10* (ED10*)	Data Line
A29	TD13* (ED13*)	Data Line	C29	TD12* (ED12*)	Data Line
A30	TD15* (ED15*)	Data Line	C30	TD14* (ED14*)	Data Line
A31	TGND		C31	TSPAREOUT	Spare output
A32	TGND		C32	TSPAREIN	Spare input

CPU I/O Cable

This 50 contact cable connects the CPU I/O card (in the second slot from the left) to the I/O Assembly attached to the inside of the case rear panel. This assembly provides communication from the CPU to the external printer port, the optional keypad, and the serial connections T1T4.

Pin	Name	Device	Pin	Name	Device
1	+12 v		26	STRB	Printer
2	5v		27	RDR4	T4
3	BUSY	Printer	28	TDR4	T4
4	VCC		29	CTS4	T4
5	PE	Printer	30	RTS4	T4
6	GND		31	RD4	T4
7	PA0	Keypad	32	TD4	T4
8	PA1	Keypad	33	RDR3	T3
9	PA2	Keypad	34	TDR3	T3
10	PA3	Keypad	35	CTS3	T3
11	PA4	Keypad	36	RTS3	T3
12	PA5	Keypad	37	RD3	T3
13	PA6	Keypad	38	TD3	T3
14	PA7	Keypad	39	RDR2	T2
15	CA1	Keypad	40	TDR2	T2
16	CA2	Keypad	41	CTS2	T2
17	ACK	Printer	42	RTS2	T2
18	DAT8	Printer	43	RD2	T2
19	DAT7	Printer	44	TD2	T2
20	DAT6	Printer	45	RDR1	T1
21	DAT5	Printer	46	TDR1	T1
22	DAT4	Printer	47	CTS1	T1
23	DAT3	Printer	48	RTS1	T1
24	DAT2	Printer	49	RD1	T1
25	DAT1	Printer	50	TD1	T1

SCSI Cable

A 50 lead cable connects a standard SCSI device to the J2 Breakout Assembly. Please refer to Section 6 (J2 Adapter Card) for more information concerning the pin listing for the connector, and to ANSI X3.1311986 for information on the SCSI specification.

Terminal Cable

A twisted pair cable connects the RJ45 connector (TERM) on the inside of the rear panel to the J2 Breakout Assembly (J5). Please see Section 6 for more information on the pin listings of the terminal connector.

AUI Ethernet Cable

Ethernet connections are made via the J2 Breakout Assembly (J7). The cable connects to the inside of the rear panel, providing easy external access. Please

see the Section 6 for pin listings, or IEEE802.3 for information on the AUI connection.

DC POWER DISTRIBUTION

DC Terminal Block

The DC Terminal Block provides the distribution of the power sources made available from the computer power supply within the CPU case. The distribution of the power supplies follows:

Pin	P/S Wire	Description
1	Blue	5V
2	Brown	-12V
3	Yellow	+12V
4	Black	GND
5	Black	GND
6	Black	GND
7	White	5V Fault Line
8	Red	+5V
9	Red	+5V

DC Harness A/B (9 inch), C (13 inch), D (17 inch)

Five volt DC power is provided to the J1 and J2 backplanes by the power supply through the DC terminal block. The four harnesses (Labeled A through D) are connected to various points on the terminal block to specific connectors on the backplanes. (See Figure 3). Harnesses A, B, and C connect to the terminal blocks as follows.

1,2 GND (Black)
3,4 +5V (Red)

The D harness connects to the terminal block as shown below:

1 GND (Black)
2,3 NC
4 +5V (Red)

Section 2 - VME MOTHERBOARDS

VMEbus J1 BACKPLANE

The VMEbus J1 backplane is a five slot, five layer board with plated through holes. It fully conforms to the VMEbus specification IEEE P1014R rev D. 96/96 way connectors are mounted on a slot pitch of 20.32 mm(0.8"). The five layers of the backplane (signal/0V/+5V/0V/signal), ensure constant characteristic impedance values to both outer signal tracks with respect to 0V, and prevents signal skew.

Short tail connectors reduce antenna effect of radiated crosstalk. Through board pins between each connector slot position on the backplane allow the use of jumpers to implement “bus grant” and “interrupt acknowledge” signal daisy chains where a connector position is not used. Jumpers for termination may be placed on either the front or the back side of the backplane.

A fiveway, low profile pin header is provided in the upper margin between the power connectors near the center of the backplane, to allow for connections to implement AC fail, System Fail, System Reset, 0V and +5V sense. A +5V sense provides DC level feedback to the power supply. AC fail, system fail, and system reset connections via solder holes are picked up directly from the backplane.

Backplane mounting holes in line with connector mounting holes are positioned across the backplane width. 5 screws on each side of the backplane secure it to the tiebars of the card frame. 4HP front filler panels are used.

AMP MateNLok power connectors bring +5V and ±12V power to the backplane. See the section titled Internal Cables for further information. Resistive termination circuitry is located at each end of the backplane. In accordance with the VMEbus specification, the specified 72 signal lines are terminated with 330 and 470 ohm resistors to give a termination of 2.94V(±10%). 0.1uF decoupling capacitors are placed between each series of resistor packs to minimize high speed signal degradation. The eight high current signal lines are clamped by Schottky diodes to eliminate negative undershoot and ringing during high speed transitions.

Placement of Jumpers

All jumpers adjacent to any empty slot (on either side) should be installed. All jumpers between adjacent installed boards should be removed.

J1 Connector Pin Assignments

PIN #	ROW A	ROW B	ROW C
1	D00	BBSY*	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	GND	BG2OUT*	GND
10	SYSCLK	BG3IN*SYS	FAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	GND	A18
21	IACKIN*	SERCLK(1)	A17
22	IACKOUT*	SERDAT*(1)	A16
23	AM4	GND	A15
24	A07	IRQ7*	A14
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	12V	+5VSTDBY	+12V
32	+5V	+5V	+5V

Notes: * Designates an active low signal

Refer to VMEbus documentation for further information.

VMEbus J2 BACK PLANE

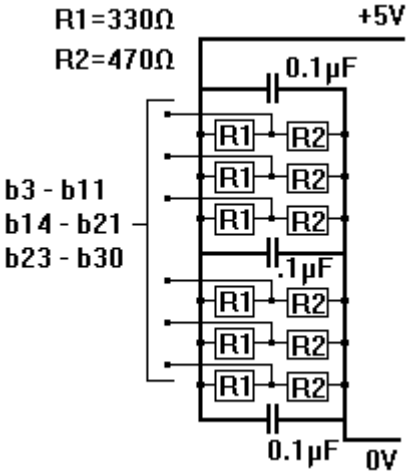
The VMEbus J2 back plane is a VMEbus (IEEE P1014R rev D) compliant motherboard with five board positions aligned with the upper (J1) back plane. In the Digalog 2040 OS9 Computer, slot assignments are determined by the cabling connected to the rear of the back plane in the following order (from left to right, as viewed from the front):

Computer
I/O Board
Testhead Controller
Emulator Controller
(Reserved for future expansion)

The J2 back plane fully conforms to the VMEbus specification, IEEE P1014R rev D. Each of the 96/96way connectors has long wire wrap tails for rows “a” and “c” extending through the back plane and forming 64/96way (reversed IEC) male connectors on the back side. The “b” rows are cropped short and fully bussed and terminated to allow for the 32 bit expansion.

There are two power connectors on the J2 back plane, top and bottom, on the back side of the board. Cabling and pinout of these connectors may be found in Section 1, Internal Cables.

Full “onboard” resistive termination circuitry is located at each end of the back plane outside the first and last connector positions together with high speed decoupling capacitors which have been placed between each series of resistor packs. This configuration is in accordance with the VMEbus specification and the specified 25 signal lines are resistor terminated with the required 330 Ohm and 470 Ohm resistors to give a termination voltage of 2.94V(+/10%).



J2 Connector Pin Assignments

PIN	SIGNAL	PIN	SIGNAL
B1	+5V	B17	D19
B2	GND	B18	D20
B3	Retry*	B19	D21
B4	A24	B20	D22
B5	A25	B21	D23
B6	A26	B22	GND
B7	A27	B23	D24
B8	A28	B24	D25
B9	A29	B25	D26
B10	A30	B26	D27
B11	A31	B27	D28
B12	GND	B28	D29
B13	+5V	B29	D30
B14	D16	B30	D31
B15	D17	B31	GND
B16	D18	B32	+5V

Rearside pinouts of the “a” and “c” rows are determined by the board plugged into the front of the back plane. Rows “a” and “c” are neither connected to nor bussed nor terminated on the VMEbus J2 back plane. They simply pass straight through.

Section 3 COMPUTER CARD

Model Motorola MVME147 Computer card
(Digalog P.N. 2025-1042)
The 2040 OS9 Computer makes use of the Motorola MVME147SB2 CPU card which runs at a clock speed of 32 MHz, has 16 MB of memory on board, and has an Ethernet transceiver interface. Other features of this board include use of the MC68030 microprocessor with the MC68882 FloatingPoint Coprocessor, SCSI bus interface with DMA (Direct Memory Access), 4 serial ports with RS232C buffers, a Centronics printer port, RESET and ABORT switches, status LEDs for SCON, DUAL, FAIL, and STATUS, tick timer, watchdog timer, timeofday clock/calendar with battery backup, 2 KB of static RAM with battery backup, 4 ROM/PROM/EPROM sockets, and A32/D32 VMEbus interface with system controller functions.

The controls and indicators on the front panel of the card include two switches and four LED indicators. The two switches available to the user are the RESET switch S2 and the ABORT switch S1. The RESET switch generates a VMEbus system reset while the ABORT switch is normally used to abort program execution and return to the debugger by causing a Level 7 interrupt when enabled. The four LED indicators are the red FAIL indicator, the yellow STATUS indicator, the green RUN indicator, and the green SCON indicator. The FAIL LED is lit when the BRDFAIL bit in the VME chip is set or when watchdog timeout occurs. The STATUS LED is lit whenever the MC68030 STATUS* pin is low. The green RUN LED is connected to the MC68030 address strobe (AS*) signal and indicates that the CPU is executing a bus cycle. The SCON LED is lit when this card is the VMEbus system controller. Additional details on the operation and functionality of the MVME147 card can be found in Sections 3 and 5 of the User Manual included by Motorola.

The MC68030 Main Memory Map is shown on p35 of the User Manual. For this specific board, the first section of Onboard DRAM should read from 0000000000FFFFFF for 16 MB of DRAM. This means that the next section of memory (00FFFFFFEFFFFFFF) is for VMEbus A32 only. The rest of the memory map follows the configuration of Table 33 in the manual.

EPROMs in sockets U22 and U30 are replaced with Microware EPROMs (Digalog P.N. 31322440 for U22 and 31322441 for U30). These EPROMs contain the boot handler for the OS9 operating system and the ROM debugger.

Jumper configuration for this card is as follows: Set jumper J1 24, 68, 35, 1315, 1416; jumper J2 24, 68, 35, 1315, 1416; jumper J3 12; jumper J5 12; jumper J6 12; jumper J8 23; and jumper J9 12. Jumpers J1 and J2 are used to configure the ROM/PROM/EPROM sockets. Configuration on this board follows the documentation in the MVME147 CPU VME module User's Manual p25 for the 128K x 8 ROM/PROM/EPROM setup. Individual pin outs for the EPROMs for the resulting configuration are given on p29 for configuration #5. Setting jumper J3 allows the card to be selected as the system controller. Jumpers J5 and J6 are for Motorola factory use only and should remain set. Setting jumper J8 23 and J9 12 configures Serial port 4 to use the clock signals provided by the TRXC4 and RTXC4 signal lines. J8 23 sets the port to receive RTXC4 signal and J9 12 sets the port to drive the TRXC4 signal line. More detailed information about these jumpers can be found in Section 2 of the MVME147 user manual.

Section 4 I/O CONTROLLER/MEMORY CARD

2040 Serial/Flash/EPROM

(Digalog P.N. 00002978)

The 2978 Serial/Flash/EPROM Assembly is a multifunction VME bus card for the Series 2040 Functional Tester. (See Figure 5.) The 2978 contains one 24bit hardware timer, two 8bit parallel ports, 4 RS232 serial ports, 64 Kbytes of EEPROM, 2 megabytes of EPROM, and up to 2 megabytes of either Flash or EPROM memory. The Flash memory chips can be replaced with EPROMs with only a change to jumper settings. All serial and parallel I/O is routed through the P2 connector on the VME expansion bus.

VME BUS INTERFACE

The 2978 Serial/Flash/EPROM Assembly is an A32:D32 slave and interrupter, conforming to VMEbus specifications Revision C.1 and IEEE Std 10141987. The card occupies 8 megabytes of the VMEbus address space. The base address is selected by switches (see switch settings).

Addressing

The Serial/Flash/EPROM Assembly will respond to the following Address Modifier Codes:

Address Modifier CODES (HEX)	FUNCTION
09	Extended Nonprivileged Data Access
0A	Extended Nonprivileged Program Access
0D	Extended Supervisory Data Access
0E	Extended Supervisory Program Access

LOCAL ADDRESS MAP

The address map shown is relative to the selected base address. To get the physical VMEbus address of a device, add the base address for the board to the local address of that device.

ADDRESS RANGE	DEVICE	NOTES
$00510000 - $007FFFFF	unused	board will not DTACK*
$00500000 - $0050FFFF	EEPROM Memory	16 X 32K (byte & word access only)
$00400100 - $004FFFFF	unused	board will not DTACK*
$004000C0 - $0040FFFF	Programming Voltage	odd byte access only
$00400080 - $004000BF	Parallel Ports and Timer	odd byte access only
$00400040 - $0040007F	Serial Port #3 and 4	odd byte access only
$00400000 - $0040003F	Serial Port #1 and 2	odd byte access only
$00200000 - $003FFFFF	EPROM Memory	32 X 512K (any type access)
$00100000 - $001FFFFF	Flash/EPROM Memory 2	32 X 256K (any type access)
$00000000 - $000FFFFF	Flash/EPROM Memory 1	32 X 256K (any type access)

EEPROM Memory

The 2978 contains 64 Kilobytes of EEPROM memory. This memory can be accessed as singlebyte or 2byte words (4byte long words are not allowed). The EEPROM memory is used to store calibration constants and similar types of data.

Flash Voltage Programming

This circuitry is provided to allow the use of 12 volt Flash memories. Writing a $01 (byte only) to an odd address in this block will turn the 12 volt programming voltage on. Writing a $00 to an odd address in this block will turn the 12 volt programming voltage off.

Hardware Timer

The 2978 uses a combination timer and parallel interface chip. The timer section of the chip is a 24 bit counter running on a 8 megahertz clock derived from the VMEbus system clock. The timer clock is routed through a 5 bit prescaler allowing a wide range of baseclock frequency (250 KHz to 8 MHz). The timer has it's own dedicated interrupt line to the CPU and is selected by jumper (see jumper and switch settings).

Parallel Port #1

Parallel port #1 of the combination timer and parallel interface chip is used to drive the printer. This port is unidirectional and 8 bits wide. The output and input handshake lines are buffered using opencollector drivers to a Centronics

type interface. The parallel port uses one output signal (STRB*) to indicate valid data on the 8 data bits, and 3 input signals (ACK* acknowledge, BUSY, and PE* error) for handshake. This parallel port shares an interrupt line with the second parallel port and all 4 serial ports. Interrupts generated by the parallel ports have the lowest priority.

Parallel Port #2 (keypad Operator Interface)
Parallel port #2 of the combination timer and parallel interface chip is used for the external Application Keypad or the Operator Interface (OPIF) units. This is an 8 bit bidirectional interface with 2 handshake lines (one bidirectional, one output only). This parallel port shares an interrupt line with the first parallel port and all 4 serial ports. Interrupts generated by the parallel ports have the lowest priority.

Serial Ports
The 2978 uses two DUART chips for controlling the 4 serial ports. The 4 serial ports can be individually configured as either RS232 or RS422 (see jumper settings). All 4 serial ports share an interrupt line with the 2 parallel ports. Interrupts generated by the serial ports have the highest priority and are ordered from 1 to 4 (highest to lowest).

EPROM Memory
The onboard EPROM memory is designed to contain the operating system, functional calls, and other programs necessary for a “diskless” system. Either 1 or 2 megabytes of total memory can be used depending on the type of EPROM chips used (2 megabit or 4 megabit). This section of the board can be accessed in single, double, triple, or quad byte transfers both aligned and unaligned (see VMEbus specifications for more information).

Flash / EPROM Memory
The onboard Flash/EPROM memory section is designed to contain the operating system, functional calls, and other programs necessary for a “diskless” system. If Flash memory is used, software updates to the system can be accomplished by simply overwriting this section of memory. Either 1 or 2 megabytes of total memory can be used depending on the type of Flash or EPROM chips used (2 megabit or 4 megabit). This section of the board can be accessed in single, double, triple, or quad byte transfers both aligned and unaligned (see VMEbus specifications for more information).

P1 CONNECTOR PIN ASSIGNMENTS

PIN	NAME	PIN	NAME	PIN	NAME
A1	D0	B1	BBSY*	C1	D8
A2	D1	B2	BCLR*	C2	D9
A3	D2	B3	ACFAIL*	C3	D10
A4	D3	B4	BG0IN*	C4	D11
A5	D4	B5	BG0OUT*	C5	D12
A6	D5	B6	BG1IN*	C6	D13
A7	D6	B7	BG1OUT*	C7	D14
A8	D7	B8	BG2IN*	C8	D15
A9	GND	B9	BG2OUT*	C9	GND
A10	SYSCLK	B10	BG3IN*	C10	SYSFAIL*
A11	GND	B11	BG3OUT*	C11	BERR*
A12	DS1*	B12	BR0*	C12	SYSRESET*
A13	DS0*	B13	BR1*	C13	LWORD*
A14	WRITE*	B14	BR2*	C14	AM5
A15	GND	B15	BR3*	C15	A23
A16	DTACK*	B16	AM0	C16	A22
A17	GND	B17	AM1	C17	A21
A18	AS*	B18	AM2	C18	A20
A19	GND	B19	AM3	C19	A19
A20	IACK*	B20	GND	C20	A18
A21	IACKIN*	B21	SERCLK	C21	A17
A22	IACKOUT*	B22	SERDAT*	C22	A16
A23	AM4	B23	GND	C23	A15
A24	A7	B24	IRQ7*	C24	A14
A25	A6	B25	IRQ6*	C25	A13
A26	A5	B26	IRQ5*	C26	A12
A27	A4	B27	IRQ4*	C27	A11
A28	A3	B28	IRQ3*	C28	A10
A29	A2	B29	IRQ2*	C29	A9
A30	A1	B30	IRQ1*	C30	A8
A31	-12V	B31	+5VSTDBY	C31	+12 V
A32	VCC	B32	VCC	C32	VCC

P2 CONNECTOR PIN ASSIGNMENTS

PIN	NAME	PIN	NAME	PIN	NAME
A1	5V	B1	VCC	C1	+12V
A2	VCC	B2	GND	C2	BUSY
A3	GND	B3	RESERVED	C3	PE*
A4	PA1	B4	A24	C4	PA0
A5	PA3	B5	A25	C5	PA2
A6	PA5	B6	A26	C6	PA4
A7	PA7	B7	A27	C7	PA6
A8	CA2	B8	A28	C8	CA1
A9	DAT8	B9	A29	C9	ACK*
A10	DAT6	B10	A30	C10	DAT7
A11	DAT4	B11	A31	C11	DAT5
A12	DAT2	B12	GND	C12	DAT3
A13	STRB*	B13	VCC	C13	DAT1
A14	TDR4	B14	D16	C14	RDR4
A15	RTS4	B15	D17	C15	CTS4
A16	TD4	B16	D18	C16	RD4
A17	TDR3	B17	D19	C17	RDR3
A18	RTS3	B18	D20	C18	CTS3
A19	TD3	B19	D21	C19	RD3
A20	TDR2	B20	D22	C20	RDR2
A21	RTS2	B21	D23	C21	CTS2
A22	TD2	B22	GND	C22	RD2
A23	TDR1	B23	D24	C23	RDR1
A24	RTS1	B24	D25	C24	CTS1
A25	RD1	B25	D26	C25	RD1
A26	N/C	B26	D27	C26	N/C
A27	N/C	B27	D28	C27	N/C
A28	N/C	B28	D29	C28	N/C
A29	N/C	B29	D30	C29	N/C
A30	N/C	B30	D31	C30	N/C
A31	N/C	B31	GND	C31	N/C
A32	N/C	B32	VCC	C32	N/C

INITIAL SWITCH SETTINGS

SW1

This switch is used to indicate the jumper setting for the parallel and serial port interrupts to the controller PAL chip (factory set to “3”).

SW2

This switch is used to indicate the jumper setting for the hardware timer interrupt to the controller PAL chip (factory set to “6”).

SW3 through SW5

These switches are used to set the base address of the Serial/Flash/EPROM card.

SW3: factory set to “E” (most significant byte)
SW4: factory set to “0”
SW5: factory set to “8”

JUMPER SETTING

JP1 through JP7

These jumpers are used to select the interrupt for the serial ports and parallel ports. JP1 through JP7 correspond to IRQ7 through IRQ1 respectively (factory set to JP5 IRQ3).

JP8 through JP14

These jumpers are used to select the interrupt for the hardware timer. JP8 through JP14 correspond to IRQ7 through IRQ1 respectively (factory set to JP9 IRQ6).

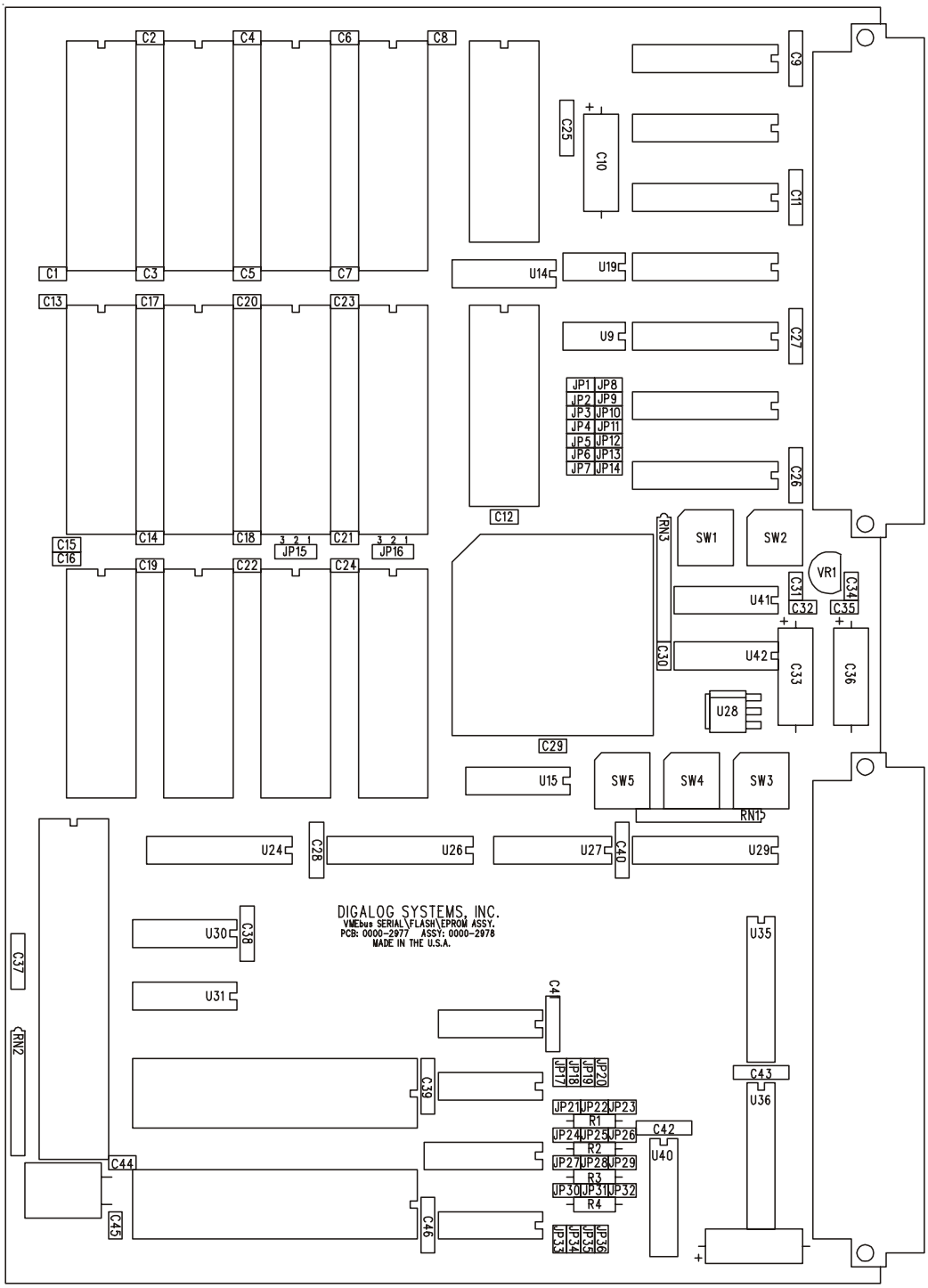
JP15 and JP16

These jumpers are used to select either FLASH or EPROM chips. NOTE: Both jumper settings MUST match or possible damage to the board or chips will result.

JUMPER	FLASH	EPROM
JP15	1 to 2	3 to 4
JP16	1 to 2	3 to 4

JUMPERS SETTINGS FOR SERIAL PORTS (JP17 through JP36)

PORT	RS232	RS422
t1	JP36	JP35, JP21, JP23
t2	JP34	JP33, JP24, JP26
t3	JP20	JP19, JP27, JP29
t4	JP18	JP17, JP30, JP32


C2
C4
C6
C8
C9
C25
C10
C11
C1
C3
C5
C7
U14
U19
C13
C17
C20
C23
U9
C27
JP1 JP8
JP2 JP9
JP3 JP10
JP4 JP11
JP5 JP12
JP6 JP13
JP7 JP14
C26
C12
C15
C16
C14
C18
C21
JP15
JP16
C19
C22
C24
RN3
SW1
SW2
VR1
C31
C34
C32
C35
U41
C30
U42
C33
C36
U28
C29
U15
SW5
SW4
SW3
RN1
U24
C28
U26
U27
C40
U29
DIGALOG SYSTEMS, INC.
VMEbus SERIAL\FLASH\EPROM ASSY.
PCB: 0000-2977 ASSY: 0000-2978
MADE IN THE U.S.A.
U30
C38
U35
C37
U31
C41
RN2
C39
JP17 JP18 JP19 JP20
C43
U36
JP21 JP22 JP23
R1
JP24 JP25 JP26
C42
R2
U40
JP27 JP28 JP29
R3
JP30 JP31 JP32
R4
C44
C45
C46
JP33 JP34 JP35 JP36

Section 5 TESTHEAD/GPIB CONTROLLER CARD

2040 Testhead Controller

(Digalog P.N. 00002974)
The 2974 Testhead Controller is a VME bus to TBus interface card for the 2040 Series Functional Tester. (See Figure 6.) The 2974 also contains a GPIB interface/controller (TMS9914) available via the J1 connector on the front faceplate. The 2974 will also interface to a Series 2030 Type III System using the first 40 pins of the P2 connector or to a Series 2030 Type II System (Emulator) using the new Emulator Data Cable interface board.
Although more than one Emulator may be driven by a Testhead Controller, an analog Testhead cannot be connected to the same controller and cable. A second controller card must be installed into the VME computer to make up a Type IV Analog/Digital system.

VME BUS INTERFACE

The 2974 Testhead Controller is an A24:D16:D08(EO) slave and interrupter, conforming to VMEbus specifications Revision C.1 and IEEE Std 10141987. The card occupies 64 Kbytes of the VMEbus address space. The base address is selected by jumper. The 2974 can also generate and service up to two separate interrupts (one for the Testhead, and one for the GPIB) which are both selected by jumper.

Addressing

The Testhead Controller will respond to the following Address Modifier Codes:

Address Modifier

CODES (HEX)	FUNCTION
39	Standard Nonprivileged Data Access
3A	Standard Nonprivileged Program Access
3D	Standard Supervisory Data Access
3E	Standard Supervisory Program Access

Local Address Map

The address map shown is relative to the selected base address. To get the physical VMEbus address of a device, add the base address for the board to the local address of that device.

ADDRESS RANGE	DEVICE	NOTES
$0000FFE0 - $0000FFFF	unused	odd byte access only
$0000FFC0 - $0000FDFF	GPIB vector register	odd byte access only
$0000FFA0 - $0000FFBF	EEPROM	odd byte access only
$0000E020 - $0000FF9F	unused	board will not DTACK*
$0000E000 - $0000E01F	GPIB controller chip	odd byte access only
$00009000 - $0000DFFF	Unused	board will not DTACK*
$00008000 - $00008FFF	Emulators	16 total at 256 addresses each
$00002000 - $00007FFF	Unused	board will not DTACK*
$00001F00 - $00001FFF	Testhead polling registers	byte/word access
$00001C00 - $00001EFF	Unused	board will not DTACK*
$00001000 - $00001BFF	Upper Testhead Map	12 slots at 256 addresses each
$00000000 - $0000FFFF	Lower Testhead Map	16 slots at 256 addresses each

Vector Registers

Two 8bit vector registers are provided for both the test head and the GPIB interrupts. Both registers are initialized to $1F (uninitialized interrupt vector) upon powerup or reset. PLDs on the Testhead Controller respond to interrupt acknowledges from the CPU with the contents of the appropriate vector register. Interrupts generated by a test head card or emulator have priority over the GPIB. These registers are accessed via reads and writes to odd addresses only.

EEPROM

The Testhead controller has a 1Kbit serial EEPROM used to store configuration, serial number, ship date, ECO, and RMA information. The EEPROM is accessed via reads and writes to odd addresses only.

GPIB Controller

The Testhead controller uses a TMS9914A general purpose interface bus (GPIB) controller chip. The GPIB port conforms to the IEEE488 1975/78 standards and the IEEE488A 1980 supplement. All I/O to external instruments is performed via the J1 connector on the front panel. This 8bit port is accessed via reads and writes to odd addresses only.

Testhead And Emulator Accesses

The Testhead and Emulator interface uses 64 pins of the VMEbus P2 connector. All data, address, control, and handshake lines to the Testhead and Emulators are optically isolated (up to 2500 VAC) from the computer via the Testhead controller. The timing for read/write accesses to the Testhead or Emulators is based on the VMEbus system clock (SYSCLK) running at 16 MegaHertz and is controlled by a timing chip (PAL3). Accesses to the Testhead or Emulators can be either byte (8 bit) or word (16bit).

P1 CONNECTOR PIN ASSIGNMENTS

PIN	NAME	PIN	NAME	PIN	NAME
A1	D0	B1	BBSY*	C1	D8
A2	D1	B2	BCLR*	C2	D9
A3	D2	B3	ACFAIL*	C3	D10
A4	D3	B4	BG0IN*	C4	D11
A5	D4	B5	BG0OUT*	C5	D12
A6	D5	B6	BG1IN*	C6	D13
A7	D6	B7	BG1OUT*	C7	D14
A8	D7	B8	BG2IN*	C8	D15
A9	GND	B9	BG2OUT*	C9	GND
A10	SYSCLK	B10	BG3IN*	C10	SYSFAIL*
A11	GND	B11	BG3OUT*	C11	BERR*
A12	DS1*	B12	BR0*	C12	SYSRESET*
A13	DS0*	B13	BR1*	C13	LWORD*
A14	WRITE*	B14	BR2*	C14	AM5
A15	GND	B15	BR3*	C15	A23
A16	DTACK*	B16	AM0	C16	A22
A17	GND	B17	AM1	C17	A21
A18	AS*	B18	AM2	C18	A20
A19	GND	B19	AM3	C19	A19
A20	IACK*	B20	GND	C20	A18
A21	IACKIN*	B21	SERCLK	C21	A17
A22	IACKOUT*	B22	SERDAT*	C22	A16
A23	AM4	B23	GND	C23	A15
A24	A7	B24	IRQ7*	C24	A14
A25	A6	B25	IRQ6*	C25	A13
A26	A5	B26	IRQ5*	C26	A12
A27	A4	B27	IRQ4*	C27	A11
A28	A3	B28	IRQ3*	C28	A10
A29	A2	B29	IRQ2*	C29	A9
A30	A1	B30	IRQ1*	C30	A8
A31	-12V	B31	+5VSTDBY	C31	+12 V
A32	VCC	B32	VCC	C32	VCC

P2 CONNECTOR PIN ASSIGNMENTS

PIN	NAME	PIN	NAME	PIN	NAME
A1	TGND	B1	VCC	C1	TE*
A2	TGND	B2	GND	C2	TR/W*
A3	TGND	B3	RESERVED	C3	TINT*
A4	TGND	B4	A24	C4	TEN*
A5	TGND	B5	A25	C5	TRESET*
A6	TGND	B6	A26	C6	TD0*
A7	TGND	B7	A27	C7	TD1*
A8	TGND	B8	A28	C8	TD2*
A9	TGND	B9	A29	C9	TD3*
A10	TGND	B10	A30	C10	TD4*
A11	+5T	B11	A31	C11	TD5*
A12	+5T	B12	GND	C12	TD6*
A13	+5T	B13	VCC	C13	TD7*
A14	+5T	B14	D16	C14	TDS1*
A15	+5T	B15	D17	C15	TA1
A16	+5T	B16	D18	C16	TA2
A17	+5T	B17	D19	C17	TA3
A18	+5T	B18	D20	C18	TA4
A19	+5T	B19	D21	C19	TA5
A20	TA7	B20	D22	C20	TA6
A21	TA9	B21	D23	C21	TA8
A22	TA11	B22	GND	C22	TA10
A23	TGND	B23	D24	C23	TA12
A24	NEWTEN*	B24	D25	C24	TADRS*
A25	TDS0*	B25	D26	C25	TGND
A26	8/16*	B26	D27	C26	TDATCK*
A27	TD9*	B27	D28	C27	TD8*
A28	TD11*	B28	D29	C28	TD10*
A29	TD13*	B29	D30	C29	TD12*
A30	TD15*	B30	D31	C30	TD14*
A31	TGND	B31	GND	C31	TSPAREOUT
A32	TGND	B32	VCC	C32	TSPAREIN

J1 Connector Pin Assignments (Front Panel GPIB)

Pin	Name	Pin	Name
1	DIO1	2	DIO2
3	DIO3	4	DIO4
5	EOI	6	DAV
7	NRFD	8	NDAC
9	IFC	10	SRQ
11	ATN	12	SHIELD
13	DIO5	14	DIO6
15	DIO7	16	DIO8
17	REN	18	GND
19	GND	20	GND
21	GND	22	GND
23	GND	24	GND

JUMPER SETTINGS

JP1 through JP8

This set of jumpers selects the base address of the Testhead controller card. JP1 is the most significant bit (A23) and JP8 is the least significant bit (A16). Factory default for an analog Testhead is JP6 installed, all others removed ($XXFBXXXX). Factory default for an emulator is JP7 and JP8 installed, all others removed ($XXFCXXXX).

JP9 through JP15

This set of jumpers selects the interrupt to be used by devices installed in the Testhead. JP9 through JP15 correspond to IRQ7 through IRQ1 respectively (factory default is JP9 installed IRQ7).

JP16 through JP18

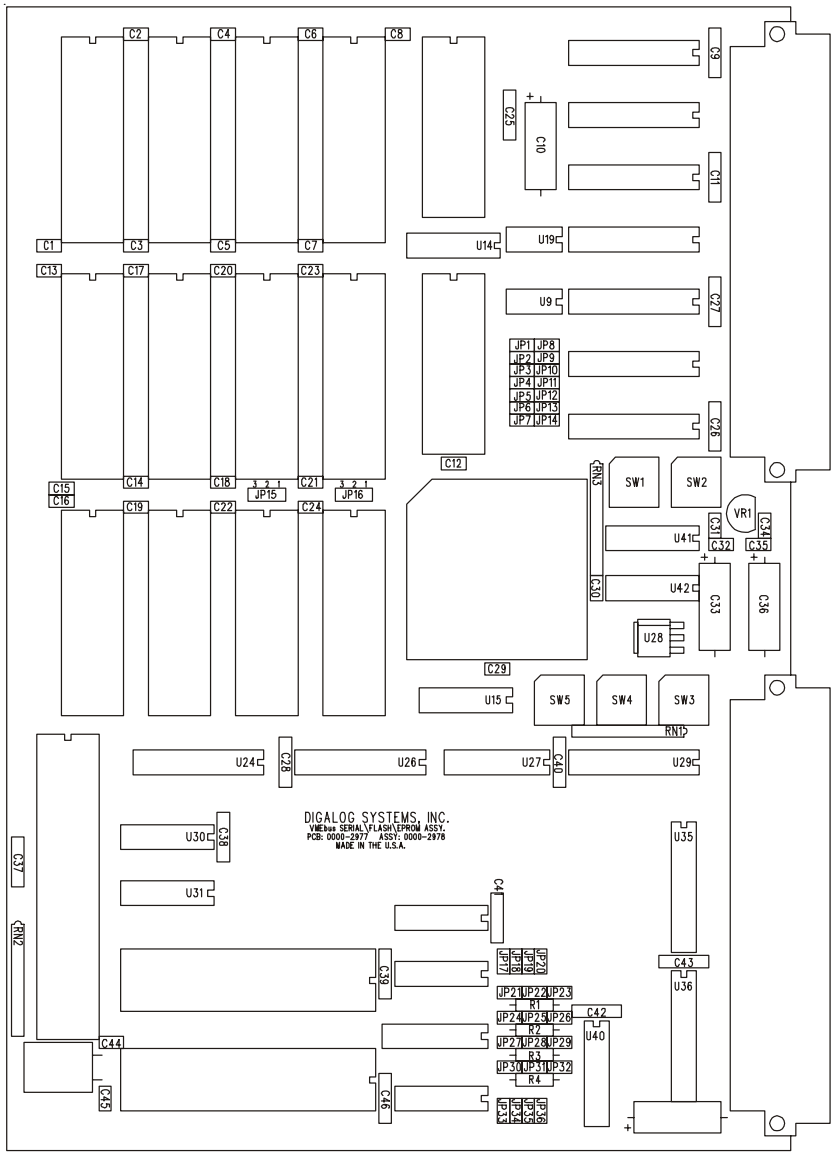
This set of jumpers is used to indicate the interrupt setting for the Testhead to the controller PAL chip. The binary combination of JP16 (most significant bit) to JP18 (least significant bit) indicate IRQ1 through IRQ7. All 3 jumpers installed is an illegal condition (factory default is all removed IRQ7).

JP19 through JP21

This set of jumpers is used to indicate the interrupt setting for the GPIB to the controller PAL chip. The binary combination of JP19 (most significant bit) to JP21 (least significant bit) indicate IRQ1 through IRQ7. All 3 jumpers installed is an illegal condition (factory default is JP19 and JP21 installed IRQ2).

JP22 through JP28

This set of jumpers selects the interrupt to be used by the GPIB controller chip. JP22 through JP28 correspond to IRQ7 through IRQ1 respectively (factory default is JP27 installed IRQ2).



Testhead/GPIB Controller Board

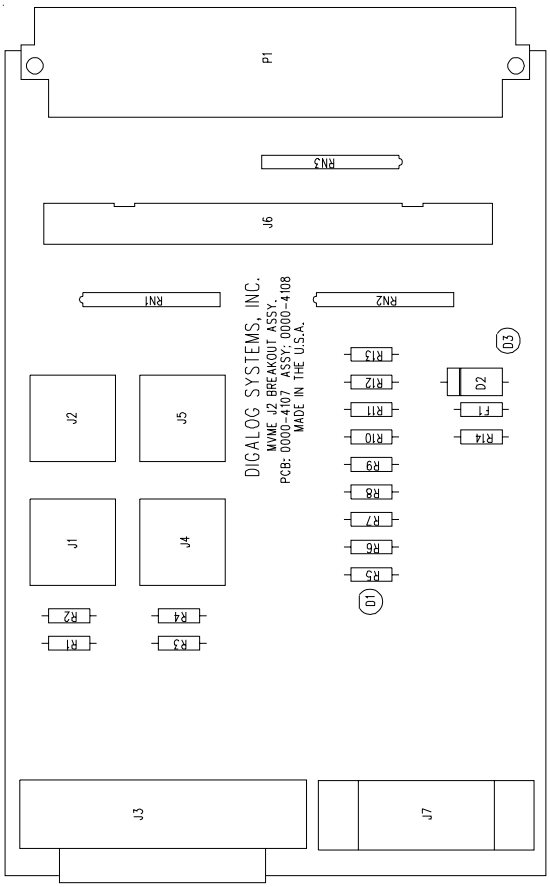
Section 6 - J2 ADAPTER CARD

2040 MVME147 J2 Breakout Assembly

(Digalog P.N. 00004110)

Digalog's 2040 MVME147 J2 Breakout Assembly is used for grouping VME back plane J2 signals for appropriate port connectors. (See Figure 7.) This assembly plugs onto the back of the VME back plane through a DIN connector and provides port connections for 4 Serial ports (/Term and /T5 /T7), Ethernet (AUI), SCSI, and a printer port. Appropriate internal cables are used to make these ports available on the back panel or to the internal devices.

This assembly also provides SCSI termination, LEDs to indicate SCSI terminator power and transceiver power, fuse for VCC, and necessary pullups and pulldowns for serial ports.



CONNECTOR PIN ASSIGNMENTS

VME J2 Back Plane Connector P1

(Please refer to Motorola MVME147 CPU VME module Support Information manual, Table 22, for pin descriptions. NOTE : Pin 1 here corresponds to pin 32 on the table and pin 32 here corresponds to pin 1 on the table.)

PIN	NAME	PIN	NAME
A1	NC	C1	DTR2
A2	DTR4	C2	DCD2
A3	DCD4	C3	RTS2
A4	RTS4	C4	CTS2
A5	NC	C5	TXD2
A6	CTS4	C6	RXD2
A7	TXD4	C7	RTS1
A8	RXD4	C8	CTS1
A9	DTR3	C9	TXD1
A10	DCD3	C10	RXD1
A11	RTS3	C11	PRFAULT*
A12	CTS3	C12	INPRIME*
A13	TXD3	C13	PRSEL
A14	RXD3	C14	PRPE
A15	O/I*	C15	PRBUSY
A16	REQ*	C16	PRACK*
A17	D/C*	C17	PRD07
A18	SEL*	C18	PRD06
A19	MSG*	C19	PRD05
A20	RST*	C20	PRD04
A21	ACK*	C21	PRD03
A22	BSY*	C22	PRD02
A23	ATN*	C23	PRD01
A24	DBP*	C24	PRD00
A25	DB07*	C25	PRSTB*
A26	DB06*	C26	+12VF
A27	DB05*	C27	R+
A28	DB04*	C28	R -
A29	DB03*	C29	T+
A30	DB02*	C30	T -
A31	DB01*	C31	C+
A32	DB00*	C32	C -

B1 through B32 are Standard J2 (power) connections.

SCSI Port J6
(Please refer to ANSI X3.1311986 for SCSI pin descriptions.)

PIN	NAME	PIN	NAME
2	DB00*	26	VCC
4	DB01*	28	GND
6	DB02*	30	GND
8	DB03*	32	ATN*
10	DB04*	34	GND
12	DB05*	36	BSY*
14	DB06*	38	ACK*
16	DB07*	40	RST*
18	DBP*	42	MSG*
20	GND	44	SEL*
22	GND	46	D/C*
24	GND	48	REQ*50O/I*

Note: Pin 25 is NC and all other odd numbered pins are grounded.

Ethernet Port J7
(Please refer to IEEE802.3 for Ethernet (AUI) pin descriptions.)

PIN	NAME
1	NC
2	C+
3	T+
4	NC
5	R+
6	GND
7	NC
8	NC
9	C -
10	T -
11	NC
12	R -
13	+12VF
14	NC
15	NC

Parallel (Printer) Port J3

(Please refer to Motorola MVME147S MPU VME Module Support Information manual, Table 22, for signal descriptions.)

PIN	NAME	PIN	NAME
1	PRSTB*	14	NC
2	PRD00	15	PRFAULT*
3	PRD01	16	INPRIME*
4	PRD02	17	NC
5	PRD03	18	GND
6	PRD04	19	GND
7	PRD05	20	GND
8	PRD06	21	GND
9	PRD07	22	GND
10	PRACK*	23	GND
11	PRBUSY	24	GND
12	PRPE	25	GND
13	PRSEL		

Serial Port J5 (Port #1, /Term)

(Please refer to EIA561 for pin descriptions)

Pin 1 CE (pulleddown w/3Kohm)
Pin 2 DCD (pulledup to +12v w/3Kohm)
Pin 3 NC
Pin 4 GND
Pin 5 RXD
Pin 6 TXD
Pin 7 CTS
Pin 8 RTS (pulledup to +12v w/3Kohm)

Serial Ports J4 (Port #3, /t5), J1 (Port #4, /t6), J2 (Port #2, /t7)

(Please refer to EIA561 for pin descriptions)

Pin 1<T>CE (pulleddown w/3Kohm)
Pin 2<T>DCD
Pin 3<T>DTR (pulledup to +12v w/3Kohm)
Pin 4<T>GND
Pin 5<T>RXD
Pin 6<T>TXD
Pin 7<T>CTS
Pin 8<T>RTS (pulledup to +12v w/3Kohm)
NC Not connected

Section 7 MASS STORAGE DEVICES

The 2040 OS9 Computer contains several mass storage devices. These devices are all connected to the computer via a SCSI interface. There are three devices: hard drive, floppy drive, and tape drive. The hard drive is used to store all of the programs and data used frequently. It has the fastest access time. The floppy drive is used to store programs and data on removable media. It has a lot less capacity. Finally, the tape drive is used to make backup copies of programs and data if loss of data occurs. It can store the entire contents of the hard drive on it.

The devices are all connected to the 2040 OS9 Computer by a single 50pin flat ribbon cable. There are three 50pin connectors on the cable for connecting to each device and they are ordered: hard drive as the first 50pin header furthest from the end of the cable, tape drive as the next 50pin header, and finally the floppy drive as the 50pin header at the end of the cable. This is done because the last device must contain termination resistors.

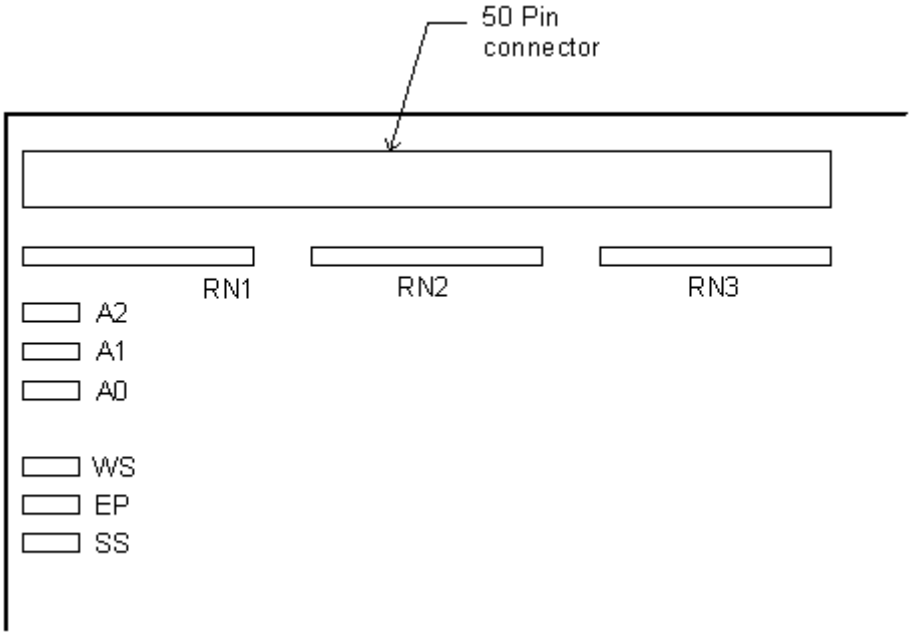
> CAUTION: WHEN REPLACING THE HARD DRIVE, FLOPPY DRIVE, OR TAPE DRIVE, THE RIBBON ASSEMBLY MUST BE CAREFULLY ROUTED IN THE COMPUTER CASE TO AVOID PINCHING OR TENSION.
>
> REMOVAL OF THE COMPUTER ASSEMBLY TOP PLATE (4 screws each side, 2 screws top rear) WILL PROVIDE ACCESS NECESSARY TO PROPERLY POSITION DISK OR TAPE DRIVE RIBBON CABLES WHEN MAKING A REPLACEMENT. FAILURE TO POSITION CABLES PROPERLY MAY RESULT IN DAMAGE AND SUBSEQUENT SYSTEM FAILURE.

HARD DRIVE

The hard drive has a capacity of at least 240MB and a 12 ms access time. It has a 1" height and is mounted behind the floppy drive within the computer. It communicates with the OS9 Computer via a SCSI interface and has a SCSI id #0.

Replacement and Configuration

The hard drive must be configured for the OS9 Computer as SCSI device id #0. Hard drive termination resistors must be removed and some jumpers that configure other options of the hard drive must also be removed. Figure 1 shows the location of the jumpers and resistor network terminators. The resistor network terminators RN1, RN2, and RN3 must all be removed. Next, the jumpers WS, SS, A0, A1, and A2 must be removed and EP should be installed. This configures the hard drive for proper operation.



Disassembly

To remove the hard drive, loosen the four screws located around the floppy drive (topleft of computer). Once these screws are loosened, the assembly can be slid out of the computer. The hard drive is mounted behind the floppy drive. Once the assembly is free from the computer, the 50pin ribbon cable and the power cable can be removed.

The hard drive needs to be removed from the assembly. There are four screws at the bottom of the assembly under the drive that need to be removed. This frees the hard drive mounting plate from the assembly. The last four screws under the mounting plate also need to be removed. This completes the disassembly process.

Assembly

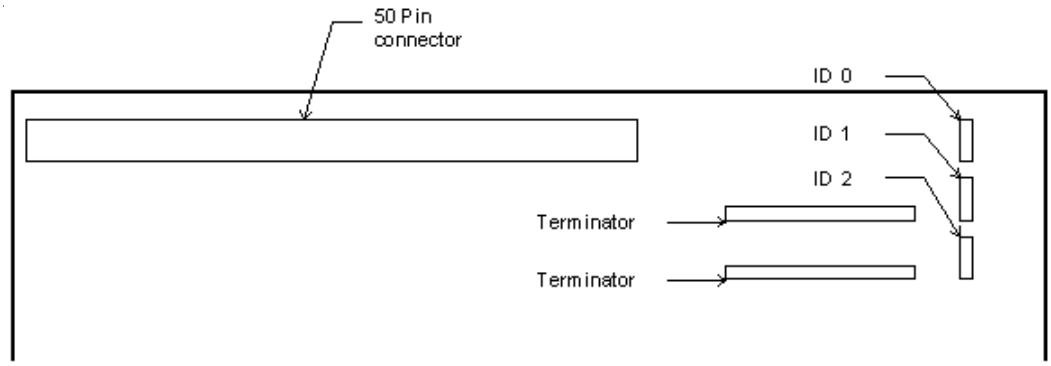
After the hard drive is configured, it is now ready to be placed into the 2040 OS9 Computer. The drive needs to be mounted on the mounting plate. This is done with the four screws removed from the disassembly. Then, the drive with the mounting plate can be mounted to the assembly. Again, this is done with four screws. Once the drive and the mounting plate is placed on the assembly, the 50pin ribbon cable can be connected to the drive via the connector that is furthest from the end of the cable. There is a red wire on the cable that designates pin 1. This must be placed towards the power connector. After this connector is seated properly, the power cable can be seated in it's connector on the drive. Now the assembly is ready to be placed into the OS9 Computer. When sliding the device into the computer housing check to see that the assembly is aligned on the tracks. MAKE SURE THE CABLES DO NOT GET PINCHED when sliding the assembly into the OS9 Computer chassis. When the assembly is seated completely then the four front screws may be tightened.

FLOPPY DRIVE

The floppy drive is a Teac 31/2" Micro Floppy Disk Drive. It has an unformatted capacity of 2 MB. It also supports 1 MB floppies . It communicates with the OS9 Computer via a SCSI interface and has a SCSI id #6.

Replacement and Configuration

The floppy hard drive must be configured for the OS9 Computer as SCSI device id #6. It must have the termination resistor networks installed and some jumpers that configure other options of the floppy drive must also be installed. Figure 2 shows the location of the jumpers and resistor network terminators. The resistor network terminators must be present and jumpers ID1 and ID2 must be removed and ID0 should have a jumper installed.



Disassembly

To remove the floppy drive from the OS9 computer, loosen the four screws located around the floppy drive (topleft of computer). Once these screws are loosened the assembly can be slid out of the computer. The floppy drive is toward the front of the assembly. Once the assembly is free from the computer, the 50pin ribbon cable and the power cable can be removed from the floppy.

The floppy drive needs to be removed from the assembly. There are four screws at the bottom of the assembly under the drive that need to be removed. This frees the floppy drive mounting plate from the assembly. The last four screws under the mounting plate also need to be removed. This completes the disassembly process.

Assembly

After the floppy drive is configured, it is now ready to be placed into the 2040 OS9 Computer. The drive needs to be mounted on the mounting plate. This is done with the four screws removed from the disassembly. Then, the drive with the mounting plate can be mounted to the assembly. Again, this is done with four screws. Once the drive and the mounting plate is placed on the assembly, the 50pin ribbon cable can be connected to the drive via the connector that is at the end of the cable. There is a red wire on the cable that designates pin 1. This must be placed towards the power connector. After this connector is seated properly, the power cable can be seated in it's connector on the drive. Now the assembly is ready to be placed into the OS9 Computer. When sliding the device into the computer housing check to see that the assembly is aligned on the tracks. MAKE SURE THE CABLES DO NOT GET PINCHED when sliding the assembly into the OS9 Chassis. When the assembly is seated completely then the four front screws can be tightened.

TAPE DRIVE

The tape drive is a Teac MT2ST series Cassette Streamer. It has a capacity of 600 MB. It communicates with the OS9 Computer via a SCSI interface and has a SCSI id #4.

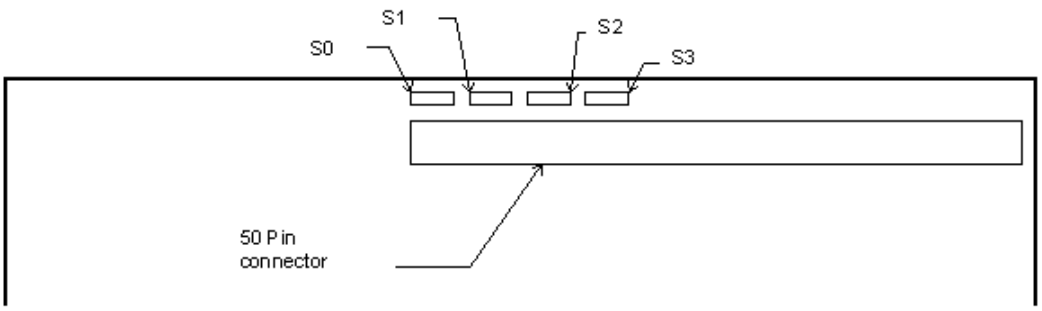
Replacement and Configuration

The tape hard drive must be configured for the OS9 Computer as SCSI device id #4. Jumper S2 must be removed and jumpers S0, S1 and S3 installed to set the device to SCSI id #4. Figure 3 shows the location of the jumpers with respect to the 50pin header. The termination resistor networks must be removed. They are located on the opposite side of the PCB.

Disassembly

To remove the tape drive from the OS9 computer, loosen the four screws located around the tape drive (to the right of the floppy drive). Once these screws are loosened, the assembly can be slid out of the computer. The tape drive is the only device located on the assembly. Once the assembly is free from the computer, the 50pin ribbon cable and the power cable can be removed from the tape drive.

The tape drive needs to be removed from the assembly. There are four screws at the bottom of the assembly under the drive that need to be removed. This frees the tape drive mounting plate from the assembly. The last four screws under the mounting plate also need to be removed. This completes the disassembly process.



Assembly

After the tape drive is configured, it is now ready to be placed into the 2040 OS9 Computer. The drive needs to be mounted on the mounting plate. This is done with the four screws removed from the disassembly. Then, the drive with the mounting plate can be mounted to the assembly. Again, this is done with four screws. Once the drive and the mounting plate is placed on the assembly, the 50pin ribbon cable can be connected to the drive via the connector that is at the end of the cable. There is a red wire on the cable that designates pin 1. This must be placed towards the power connector. After this connector is seated properly, the power cable can be seated in it's connector on the drive. Now the assembly is ready to be placed into the OS9 Computer. When sliding the device into the computer housing check to see that the assembly is aligned on the tracks. MAKE SURE THE CABLES DO NOT GET PINCHED when sliding the assembly into the OS9 Computer chassis. When the assembly is seated completely then the four front screws can be tightened.

Section 8 PERIPHERAL DEVICES

TERMINAL AND ASSOCIATED CABLING

The power cord for the terminal is plugged into one of the external accessory outlets located at the bottom rear of the 2040 cabinet.

The external keyboard cable (Digalog part number 00002867) is routed from the “hand set” modular jack located on the back of the terminal to the “hand set” modular jack located at the top right rear of the 2040 cabinet.

The internal keyboard cable (Digalog part number 00002866) is routed from the back panel feedthrough jack to the keyboard tray at the front of the cabinet.

If the cabinet keyboard tray is not used, the terminal may be set up to use the coiled keyboard cable provided with the keyboard. This method would bypass the internal and external keyboard cables.

The external communication converter (Digalog part number 00002858) plugs into Serial 1 Port, indicated on the back of the terminal. The external eight conductor serial communications cable (Digalog part number 00002863) is routed from the RJ45 jack of the converter to the RJ45 jack located at the upper right rear of the 2040 cabinet.

The internal communication cable (Digalog part number 00002862) is routed from the back panel feedthrough jack to the jack labeled TERM on the back of the OS9 computer.

The power, brightness and contrast controls are located on the right side of the terminal.

The terminal configuration can be set by restoring the default settings and making the following changes:

F2	personality = VT100
F2	wrap EOL = OFF
F3	Corner Key = COMPOSE
F7	Char Mode = NATIONAL
F7	DEL<T>= BS / DEL

Terminal feature and configuration explanation can be found in the terminal manual.

PRINTER AND ASSOCIATED CABLING

The power cord for the printer is plugged into one of the external accessory outlets located at the bottom rear of the 2040 cabinet. The power switch is located on the right side of the printer.

The communication cable for the printer (Digalog part number 60001607) is routed from the printer through a one inch grommeted hole located at the top right rear of the 2040 cabinet. From there it is routed to the port labeled PRINTER located on the back of the computer.

The printer is preconfigured and ready to operate once turned on, loaded with paper and selected.

Section 9 - SOFTWARE

OS9 DRIVERS AND DESCRIPTORS

The following are short explanations of the device descriptors and drivers provided with the OS9 operating system.

/term This descriptor uses serial driver sc8x30. The /term descriptor is for the main terminal device and is set to the following values 9600 Baud 1 Stop 1 Start. Additional settings can be found using the tmode command. The boot loader code that loads the operating system from the hard drive uses its own driver to communicate with the /term device. This driver communicates at 9600 baud.

/t5 /t6 /t7 These descriptors use serial driver sc8x30. They are installed with the baud rate set at 9600.

/t1 /t2 /t3 /t4 These descriptors use serial driver sc68681. They are installed with the baud rate set at 9600.

/pps This descriptor uses serial driver sc68681 and is used to communicate with the UUT controllers. The descriptor is a copy of the /t2 descriptor but the settings have been changed to force it to transmit and receive transparently. The settings can be viewed using the utility xmode.

/timer The timer descriptor uses driver idle68230. This driver and descriptor implements the hardware delay timing for the functional calls which eliminates software delay loops.

/h0 /s0 These are the descriptors for the hard drive and they use driver rbvccs. /s0 is a copy of /h0 and is provided for compatibility with previous Digalog testers.

/d0 /s1 /d0h These are the descriptors for the floppy drive and they use the driver rbteac. /s1 is a copy of /d0 and is provided for compatibility with previous Digalog testers. /d0h is for high density floppies of 2 MB unformatted.

/mt0 This is the descriptor for the tape drive. It uses the sbteac driver.

/**h0fmt** This descriptor allows the hard drive to be opened for formatting. The /h0 and /s0 descriptors do not allow formatting.

/**nv0** This descriptor is for the ROM drive which holds most of the OS9 utilities. It uses the driver called ram. You should include /nv0/cmds in your environment path statement.

/**amsint** This descriptor uses driver amsd. This descriptor and driver implement the interrupt mode of the AMS board.

/**nil** This descriptor is a bit bucket. It is used when you want to run a program in the background and do not want to see its output on the screen. You can redirect the output to /nil.

sc8x30 This driver is for the serial devices located on the computer board. The driver supports hardware and software handshaking. Valid baud rates for the driver are 50, 75, 110, 134.5,150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200.

sc68681 This driver is for the serial ports located on the I/O controller/ memory card. The driver supports hardware and software handshaking. Valid baud rates are 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19200.

HARD DISK DIRECTORY STRUCTURE
There are several main subdirectories.

CMDS holds utilities provided by Microware. Most of these have been moved to /nv0/CMDS.

CMDS/BOOTOBJS holds descriptors and drivers provided by Microware.

SYS holds the password file and errmsg file in addition to some other miscellaneous files.

ISP holds the Internet network files.

NFS holds the network file system files.

DIGALOG holds most of the files and utilities provided by Digalog.

DIGALOG/CMDS holds functional calls, calibration files, and utilities.

DIGALOG/CMDS/BOOTOBJS holds descriptors and drivers provided by Digalog.

DIGALOG/SELFTEST holds the menus for the self test executive.

DIGALOG/SELFTEST/SOURCES holds the self test sources.

DIGALOG/HELP holds help files.

In addition to directories, the hard drive has two important files located in the root of the drive, os9boot and startup. Os9boot is loaded into memory by the boot loader and contains the operating system, drivers, descriptors, and file managers. Startup is a script file that is run automatically when the system is booted. This script loads additional drivers and descriptors, starts the timeshare system, and calls other scripts for starting the network systems.

/nv0 DIRECTORY STRUCTURE

The ROM drive has one subdirectory called CMDS. This directory hold most of the commands and utilities provided by Microware. These include utilities provided with the ISP and NFS systems.

OS9 BOOT SEQUENCE

The 2040 boots in three phases. First the bootloader in ROM is executed. The bootloader loads the OS9 kernel and any drivers found in the boot file. Second the kernel executes and starts the initial shell process. Third, the initial shell process executes the file called startup located in the root of the system drive /h0.

The bootloader can be configured for different modes of execution. These are boot menu, debugger, boot menu and debugger, or no boot menu and no debugger. In addition the boot order of devices can be specified. If debugger is enabled the bootloader jumps into the debugger and the user must enter a g <<RET>> to continue. If the boot menu is enabled the user must choose the device to boot from. The boot menu also allows the user to reconfigure the bootloader. If the bootloader is configured for no boot menu, you can set

it to the default boot menu by resetting the system six times in a row, making sure that the red fail light goes on 6 times. The default boot menu lists many devices that are not available on the system. The only devices that can be booted from are the Teac floppy drive, Teac tape drive, and the SCSI hard drive. If the boot menu is not enabled, the bootloader will attempt to boot from each device in the order it appears in the menu. This order can be configured by the user. The fastest boot order is hard drive, floppy drive, tape drive.

The startup file contains commands that the initial shell executes. These commands load additional drivers and descriptors, functional call library, and calibration tables. The startup file also contains commands to execute the startup files for the Ethernet and start the timeshare system. Additions to this file may safely be inserted before the last line of the file which is the line that starts the timeshare monitor tsmon. Do not make changes to the file unless you are familiar with OS9.

The timeshare system requires users to login with a user name and password. The file /h0/sys/password contains the valid user names and passwords. The system is shipped with the following user names and passwords:

digalog mke	This is the superuser account
selftest	Logs in and runs the selftest executive

In addition, pressing the <<RET>> key logs in as a default nonsuperuser account.

GPIB SOFTWARE

The GPIB port is called /G. Its operation is similar to the other system I/O ports such as the printer and terminal ports. The main Basic statements used with an I/O port are OPEN, CLOSE, READ, PRINT, and INKEY. /G may also be used in Shell commands like Copy, Echo, etc. For more information on the Microware I/O system, refer to the OS9 technical manual. I/O redirection from Shell is covered in the OS9 users manual. I/O from Basic is treated in chapter 11 of the Basic manual.

The following sections discuss features peculiar to the GPIB. It must be kept in mind that all GPIB instruments behave somewhat differently, so only a general outline may be given here.

The primary difference between the GPIB and other character interfaces is that commands are necessary to select and deselect instruments. Some instruments also may be triggered and cleared, have the front panel controls locked out, etc. by GPIB commands. These commands are all single byte codes, but with the ATN signal line asserted.

These command bytes are sent over the GPIB, with ATN asserted, by prefixing them with the special command character ‘ (hex value $60). For example, the string AB‘CD would send A, B, and D as ASCII characters but would send C (hex value $43) as a command. The character ‘ is not sent. If for some reason the character ‘ must be sent as a literal ASCII character, it may be written double. Thus the string AB“CD would be sent literally as AB‘CD without any command bytes or ATN assertion.

Some of the GPIB command codes are impossible to generate on your ASCII keyboard or display on your screen , so an ASCII to hexadecimal conversion feature is included. The $ character will convert the next two ASCII characters, sent to the GPIB, to a single byte value. For example the string PQ$52S would be sent as PQRS ($52 == “R”). The string `$08 would be sent as the command Group Execute Trigger (GET). This works for either command or message characters. In case you need to send a literal $ character, it may be written double ($$).

IFC/REN

When the GPIB port is initialized by opening a path to /G, the Interface Clear (IFC) line is pulsed and the Remote Enable (REN) line is asserted. When the path is closed, REN is dropped.

Pulsing IFC resets all instruments on the bus to their poweron states. Asserting REN disables the front panel controls. When REN is dropped, instruments revert to manual control.

To avoid resetting the bus every time a path is opened, it is advisable to open a dummy path that never gets closed. Then, additional paths may be opened and closed at any time without affecting IFC and REN. A good way to do this is with the OS9 utility command Iniz. You may put the statement Iniz /G in your Startup file so that the GPIB port is automatically initialized once and for all. You may also use an OPEN statement in your main Basic program, but then be sure to close the path before quitting the program. Otherwise, each

time the program runs another path will be opened and the system will eventually run out of available path numbers.

When using /G in a subprogram, you may either pass an open path number from the main program, or let the subroutine open and close its own path. The latter method is preferable since it does not require parameter passing and it keeps all I/O operations local to the subprogram. A dummy path must still be open at all times to avoid resetting the bus.

Hexadecimal Conversion

Many GPIB commands require the use of nonASCII character codes. To simplify programming, the $ character automatically performs ASCIIto binary conversion on the following two characters. For example, the threecharacter sequence $1B would be sent as an escape character.

If the $ character itself is to be sent, it may be written twice as $$, similar to the way quotation marks must be written in print statements. It may also be written as $24.

GPIB Commands

GPIB commands are introduced by the ‘ character (this is NOT the single quote character. The ASCII code for the ‘ character is $60). Most of these commands are nonASCII codes and are written using hexadecimal conversion:

‘$01 GTL Go To Local	(enables manual controls)
‘$04 SDC Selected Device Clear	(resets listeners)
‘$05 PPC Parallel Poll Configure	(not used)
‘$08 GET Group Execute Trigger	(triggers listeners)
‘$09 TCT Take Control	(see slave operation, below)
‘$11 LLO Local Lockout	(disables manual controls)
‘$14 DCL Device Clear	(resets all devices)
‘$15 PPU Parallel Poll Unconfigure	(not used)
‘$18 SPE Serial Poll Enable	(see Serial Poll, below)
‘$19 SPD Serial Poll Disable	(see Serial Poll, below)
‘$20 through ‘$3E	(select listeners 0 through 31)
	LAG Listen Address Group
‘$3F UNL Unlisten	(deselects all listeners)
’$40 through ‘$5E	(select talkers 0 through 31)
	MTA My Talk Address
‘$5F UNT Untalk	(deselects all talkers)
‘$60 through ‘$7F	(not used)
	SCG Secondary Command Group

Once again, if the ‘ character itself is to be sent, it may be written as either “ or $60.

GPIB Address

The tester has a GPIB address, just like any other instrument. This address is normally 0. The address is stored in the device descriptor module G, and may be changed with the Shell command Xmode /G TABC = xx. Legal addresses are 00 through 1E (decimal 0 through 30).

Controller Protocol

Before the controller can send data (other than GPIB commands) it must select itself as talker by sending ‘$40 (talk address 0). Failure to do so will result in Error #000:203 Bad I/O Mode.

Before the controller can receive data, it must select itself as listener by sending ‘$20 (listen address 0). Failure to do so will result in Error #000:203 Bad I/O Mode.

Before the controller can send data, there must be a listener on the bus. The listener is selected with one of the Listen Address Group commands ‘$21 through ‘$3E (address 1 through 30). Failure to have a listener will result in Error #000:245 Write Error. This error will also occur on GPIB commands if there are no devices attached whatsoever (or all instruments are powered off). All instruments listen to GPIB commands whether they are selected or not.

When sending GPIB commands, a PRINT statement should end with a semicolon to suppress the carriage return. Here is an example of a typical exchange with a voltmeter:

```
DIM g:integer                REM path number must be byte or integer
OPEN #g,”/g”                 REM initialize port, clear GPIB
PRINT #g,”‘$21‘$40DCV”       REM meter listen, tester talk, DC volts mode
PRINT #g,”‘$20‘$41";         REM tester listen, meter talk, no terminator
READ #g, voltage             REM get voltage reading
CLOSE #g                     REM return meter to manual
```

Time Out

A time out feature may be enabled with the Shell command “Xmode /G TABS=nnn” (nnn ranges from 0 to 255 in tenths of a second). If the time out is set to zero, or if the system clock is not running, then the time out feature is disabled. The TAB character in the device descriptor is used as OS9 does not have a way to expand the device descriptor fields.

Time out on input or output will result in Error 000:246 Device not ready. The Basic function INKEY(#path) may be used to determine if data is ready to be received, so as to avoid getting hung up waiting for I/O. Since the GLIB is unbuffered, INKEY will only indicate 1 or 0 for ready or not ready, respectively.

Terminator/EOI

PRINT statements automatically end with carriage return. If the Xmode LF parameter is on, this becomes carriage return + linefeed. If a semicolon is used at the end of the print statement, there is no automatic terminator, but you may explicitly code a terminator. For example, PRINT “hello$0A”; would send a linefeed only.

The bus signal EOI (End or Identify) is asserted whenever a carriage return character is sent. The standard descriptor module G is set for Xmode NOLF, so the terminator consists of carriage return with EOI.
When receiving messages, a carriage return is required and the EOI signal is ignored. Warning: if an instrument sends carriage return + linefeed, the linefeed will not be read at the end of the message, but will appear at the beginning of the following message. If the instrument cannot be programmed to omit the linefeed, it may be necessary to strip it off in your program.

XMode

The Shell command Xmode is used to set the GPIB address, time out, and terminator parameters. Xmode works by modifying the device descriptor module G. This must be done before initializing the GPIB port. If the parameters are to be changed on the fly, all paths to the GPIB port must be closed. The new parameters take effect when the next path is opened. To make the changes permanent, you may save the modified descriptor. However, the module CRC will be wrong and must be corrected before the module can be loaded. The following example command sequence will modify, save, and correct the descriptor: The tab character is used for the time out value.

```
Xmode /G tabs=10 <<CR>>
chd /h0/digalog/cmds/bootobjs <<CR>>
Save r G <<CR>>
Fixmod u G <<CR>>
```

Note that Xmode refers to device name /G while Save and Fixmod use the module name G.

Slave Operation

When the GPIB port is first initialized, it assumes that it is the ControllerInCharge. However, up to 15 testers may be networked on a GPIB, and only one of them can be controller. In this case, each slave unit must have a unique address (Xmode /G TABC = xx), and the slaves must relinquish control. This can be done two ways.

First, the tester automatically relinquishes control when it sees bus activity indicating the presence of another controller. When a unit sees its own talk or listen address command occur on the bus, it assumes another controller is present and relinquishes control.

Second, a unit may be programmed to relinquish control by giving the Take Control command (‘$09). This passes control to any unit currently addressed as talker. If each slave unit starts by giving the command sequence ‘$40‘$09 (Talk Address 0, Take Control), then unit #0 will be the controller.

Slave units cannot send any GPIB commands. Attempts to print the ‘ character will produce Error #000:203, Bad I/O Mode. The same error will occur on any PRINT or READ statement, unless the slave unit has been selected to talk or listen by the controller.

When a slave unit wishes to send or receive data, it must wait its turn. This means using the ON ERROR GOTO statement to handle I/O Error #000:203. The program should continue trying to send or receive until it succeeds, that is, until error 203 does not occur.

USING THE IEEE 488 FROM BASIC

Opening A "path"

Opening a "path" to a serial port is handled the same way as opening a disk file (the GPIB is considered a byteserial port). In BASIC:

DIM G: INTEGER \ OPEN #G, "/G"

The string /G is the name of the GPIB port. Any string variable or expression may be used in an OPEN statement. For example, the console terminal is called /term, the printer is /P, etc. Names beginning with a slash refer to hardware devices; without a slash refer to disk files.

The operating system will assign a small integral value, called the path number, to the variable G. This number is used in all subsequent I/O statements.

Sending Gpib Commands: TALK, LISTEN

Unlike many GPIB controllers, the 2040 does not use subroutines or calls to control the GPIB. Instead, commands are embedded within the ASCII data strings passed over the bus. A special control character is used to distinguish command bytes from normal ASCII data strings. This control character is the accent mark ‘, hexadecimal code $60. (One of the most frequent problems is confusion of this character with the apostrophe ‘, hex code $27).

GPIB listen commands for addresses 030 are hexadecimal codes $20$3E. Talk commands are hex codes $40$5E. To send, for example, listen #5, talk #6, you would write:

PRINT #G, "`$25`$46";

If, in the course of communicating with some device, it is necessary to send the accent mark character without indicating a command byte, it may be written double (“). This will not cause a command to be sent but will be sent as a single literal ‘ character. Similarly, if the dollar sign character is written double ($$) it will not cause hex conversion of the next two characters, but will be sent as a single literal $ character. (The hex conversion feature may be used at any time, not just for commands.)

The GPIB address of the 2040 is normally set to 0, but may be reconfigured in software as described on the next page.

Sending, Receiving Data

The print and input statements are normally used for ASCII strings. PUT and GET statements are used for sending/receiving single bytes, as during a serial poll. Don't use an input statement during a serial poll, as the talker will not issue a carriage return and the input statement will never return.
To configure a device at GPIB address 1, for example,

PRINT #G,"'$20'$41", voltage

The semicolon at the end of the print statement prevents the transmission of a carriage return at the end of the string. This may or may not be desirable, depending on the individual requirements of the attached instrument. However, it is always necessary to suppress the carriage return if the 2040 is not selected as a talker (otherwise an error will occur when the 2040 attempts to send the carriage return). The 2040 may select itself as talker, as in the above example, or it may be selected by another controller on the GPIB.

Closing The Path

Before exiting your program, you must close the path:

CLOSE #G

Failure to close the path may not be detected for some time (if at all) because the system has the capacity to keep many paths open simultaneously; however, the system path tables or main memory might become full. If the program is aborted by an error keyboard abort, all paths will be closed automatically. Warning: Do not accidentally CLOSE #0, #1, OR #2. This will knock out the console keyboard.

Handling Errors

By default, an error will abort your program. If you are running a source program, a debug mode will be entered during which you may examine program variables, list the program, etc. If you are running a "packed" or Icode program, it simply halts and returns to keyboard control.

You may direct error control to any numbered line with the statement:

ON ERROR GOTO nnn

and return to default mode with the statement:

ON ERROR

There are many error numbers that may occur during I/O operation; only those with special meaning on the GPIB will be discussed here. The others are described in the operation system handbooks and, briefly, in the SYS/ERRMSG disk file.

Error #000:203 Bad I/O mode

This normally occurs when attempting to read (write) a file that has been opened for write (read). For the GPIB, it may also mean that an attempt has been made to issue a command when the 2040 is not the system controller. The 2040 will normally assume that it is the system controller; however, if you issue the take control command, '$09, the 2040 will relinquish controller. Control may only be regained in this case if another controller gives the take control command back to the 2040 (or by closing and reopening the path).

Error #000:245 Write error

This may occur during transmission of either a command or data byte, if there is no other instrument attached to the bus to receive the byte. More frequently, however, it occurs because the 2040 attempts to send a byte when it has not been selected as a talker. Especially insidious is the tendency for many I/O statements to follow data with a carriage return. Careful programming will prevent this.

Error #000:246 Not ready

For the GPIB, this represents a time out error. The time limit may be set from 0.1s to 25.5 seconds in tenths. Or, if set to zero, the time out feature is disabled and the 2040 will wait forever for I/O to finish. Also, if the system realtime clock is not running, the time out feature is disabled.

A special consideration is serial poll mode. This is entered by sending the command '$18. In this mode, the time out is always set to the minimum value, 0.1s, in case a nonexistent address is polled. Serial poll mode is exited by sending '$19.

System Configuration

Several necessary steps are typically performed automatically at system initialization by the macro file STARTUP. This includes setting the system clock (from the batterybackedup clock) and loading the I/O modules into memory. For example:

SETIME s
LOAD G GPIB

The utility program Xmode is used to change the GPIB address of the 2040 and to set the time out limit:

XMODE /G TABS = nnn TABC = hh

Where nnn is the time limit in tenths of a second from 0255, and hh is the hexadecimal GPIB address from $00 $1E. These values must be configured prior to opening a path to /G.

You should also be able to work interactively over the GPIB using utility keyboard commands like list, copy, echo, etc. In this case, each command line will automatically open a path, do the I/O, then close the path. This will prevent you from giving talk and listen commands on one line and doing I/O on the next line because of the intervening path closure.

The keyboard command "INIZ /G" will open a path and keep it open forever, preventing this problem. The path may be closed with "DEINIZ /G". For example, suppose you have a voltmeter at address 1. You should be able to type:

INIZ /G
ECHOR/G '$20'$41
LIST /G

to read the voltmeter directly from the keyboard.

Examples

This example will perform a serial poll on all addresses 030, printing out the response bytes for responding devices and skipping devices that produce a time out error:

```
PROCEDURE serpoll
ON ERROR GOTO 1
REM OPEN PATH TO GPIB
DIM G:INTEGER; x;BYTE
OPEN #G,"/G"

REM UNTALK, SERIAL POLL, LISTEN#0
PRINT #G,"'$5f'$18'$20";
FOR n=1 to 30 \ REM CAREFUL, DON'T POLL #0, THAT'S
                        YOURSELF
PRINT #G,"'"+CHR$(n=$40); \ REM TALK ADDRESS
GET #G,x \ REM IF NO TALKER, ERROR 246 OCCURS HERE
PRINT USING "i5,`(`,h2,s2",n,x,")"; \ REM PRINT n(xx) ON
                                          TERMINAL 1
NEXT n \ PRINT \ PRINT #G,"'$19"; \ CLOSE #G \ REM
                                          DISABLE  POLL
```

This example will send a configuration string to a hypothetical meter at address 1, and return the meter reading as a "parameter".

```
PROCEDURE meter
DIM G:INTEGER \ OPEN #G,"/G
PARAM voltage \ REM declares a value passed back to caller
PRINT #G,"'$21'$40...hypothetical configuration string...";
READ #G,"'$20'$41",voltage
```

The above procedure is called from another program:

```
PROCEDURE whatever
DIM reading:REAL \ REM redundant, default type is real
READ nominal, tolerance \ DATA 5.0,0. \ REM pretend test
                                          data
RUN meter (reading)
IF ABS (readingnominal)  tolerance THEN
```

```
PRINT"FAIL READING =;reading
ELSE PRINT "PASS!" \ ENDIF
```

COMMON PROBLEMS

The most common GPIB programming problems relate to the data terminator and the trigger mode. Usually, reading data from a device does not trigger a measurement, the reading obtained refers to some previous measurement. These final notes cover most of the problems reported by GPIB users.

Avoid auto triggered readings, use GPIB triggered mode. Otherwise the data you receive may represent old readings.

Command listener first, then talker. Otherwise bytes may be missed.

The normal input terminator is a carriage return (CR = $0D). If a device sends a linefeed (LF = $0A) or CR/LF, you should set XMODE /g EOR=0A. Other wise bytes may be missed.

Microware BASIC INPUT statements send a question mark to the talker, with strange results. Use the READ statement.

GPIB commands should not be followed by CR or other terminator. Device messages may require CR, LF or CR/LF. Print normally sends CR/LF. The line feed can be suppressed by using XMODE /G NOLF.

Both CR and LF are suppressed by a semicolon at the end of a PRINT statement.

Appendix A

/A-msInt

descriptor name ..... amsint
file manager name ..... Scf
device driver name ..... AMSd
port address $f0fb0000
irq vector 90
irq level 7
irq priority 1
device mode capabilities $23
device class $00
upper case lock 0
backspace option 0
delete line option 0
echo flag 0
automatic line feed flag 0
end of line null count 0
end of page pause flag 0
page length 0
backspace input character ..... ^@
delete line character ..... ^@
end of record character $00
end of file character $00
reprint line character ..... ^@
duplicate line character ..... ^@
pause character ..... ^@
keyboard interrupt character ..... ^@
keyboard quit character ..... ^@
backspace output ..... ^@
line overflow character ..... ^@
parity code $00
adjustable baud rate $00
output device name ..... amsint
xon character ..... ^@
xoff character ..... ^@
tab character ..... ^@
tab column width 0

/d0

descriptor name d0
file manager name ..... RBF
device driver name ..... rbteac
port address $fffe4000
irq vector 69
irq level 4
irq priority 5
device mode capabilities $a7
device class $01
drive number 0
step rate 3
device type $26
density $03
reserved $00
number of cylinders 79
number of heads/sides 2
disk write verification 1
default sectors/track 16
default sectors/track 0 16
segment allocation size 8
sector interleave factor 1
dma transfer mode 0
track base offset 1
sector base offset 1
sector size 256
control word $0002
number of tries (1=no retry) 7
SCSI unit number of drive 0
write precompensation cylinder 0
reduced write current cylinder 0
cylinder to park disk head 0
logical sector offset<T>0
total cylinders on device 80
SCSI controller id $06
datatransfer/rotation rate $10
SCSI options flags $00000001
maximum transfer count $003ddfd2

/d0h

descriptor name d0h
file manager name ..... RBF
device driver name ..... rbteac
port address $fffe4000
irq vector 69
irq level 4
irq priority 5
device mode capabilities $a7
device class $01
drive number 0
step rate 3
device type $26
density $03
reserved $00
number of cylinders 80
number of heads/sides 2
disk write verification 1
default sectors/track 32
default sectors/track 0 32
segment allocation size 8
sector interleave factor 1
dma transfer mode 0
track base offset 0
sector base offset 1
sector size 256
control word $0002
number of tries (1=no retry) 7
SCSI unit number of drive 0
write precompensation cylinder 0
reduced write current cylinder 0
cylinder to park disk head 0
logical sector offset 0
total cylinders on device 80
SCSI controller id $06
datatransfer/rotation rate $30
SCSI options flags $00000001
maximum transfer count $00ffffff

/g

descriptor name ..... g
file manager name ..... Scf
device driver name ..... gpib
port address $f0fbe001
irq vector 93
irq level 2
irq priority 4
device mode capabilities $23
device class $00
upper case lock 0
backspace option 0
delete line option 0
echo flag 0
automatic line feed flag 0
end of line null count 0
end of page pause flag 0
page length 0
backspace input character ..... ^@
delete line character ..... ^@
end of record character $0d
end of file character $00
reprint line character ..... ^@
duplicate line character ..... ^@
pause character ..... ^@
keyboard interrupt character ..... ^@
keyboard quit character ..... ^@
backspace output ..... ^@
line overflow character ..... ^@
parity code $00
adjustable baud rate $00
output device name ..... g
xon character ..... ^@
xoff character ..... ^@
tab character ..... ^@
tab column width 0

//h0

descriptor name h0
file manager name ..... RBF
device driver name ..... rbvccs
port address $fffe4000
irq vector 69
irq level 4
irq priority 5
device mode capabilities $a7
device class $01
drive number 0
step rate 0
device type $80
density $00
reserved $00
number of cylinders 0
number of heads/sides 0
disk write verification 1
default sectors/track 0
default sectors/track 0 0
segment allocation size 32
sector interleave factor 1
dma transfer mode 0
track base offset 0
sector base offset 0
sector size 0
control word $000b
number of tries (1=no retry 7
SCSI unit number of drive 0
write precompensation cylinder 0
reduced write current cylinder 0
cylinder to park disk head 0
logical sector offset 0
total cylinders on device 0
SCSI controller id $00
datatransfer/rotation rat $00
SCSI options flags $00000001
maximum transfer count $00ffffff

/h0fmt

descriptor name h0fmt
file manager name ..... RBF
device driver name ..... rbvccs
port address $fffe4000
irq vector 69
irq level 4
irq priority 5
device mode capabilities $a7
device class $01
drive number 0
step rate 0
device type $80
density $00
reserved $00
number of cylinders 0
number of heads/sides 0
disk write verification 1
default sectors/track 0
default sectors/track 0 0
segment allocation size 32
sector interleave factor 1
dma transfer mode 0
track base offset 0
sector base offset 0
sector size 0
control word $000a
number of tries (1=no retry) 1
SCSI unit number of drive 0
write precompensation cylinder 0
reduced write current cylinder 0
cylinder to park disk head 0
logical sector offset 0
total cylinders on device 0
SCSI controller id $00
datatransfer/rotation rate $00
SCSI options flags $00000001
maximum transfer count $003e27a2

/mt0

descriptor name mt0
file manager name ..... SBF
device driver name ..... sbteac
port address $fffe4004
irq vector 69
irq level 4
irq priority 5
device mode capabilities $67
device class< $03
tape drive number 0
reserved $00
number of blocks/buffers 8
block size 24576
driver process priority 1000
SBF capability flags $00
drive capability flags $00
dma mode $0000
SCSI controller id 4
SCSI controller lun 0
SCSI options flags $00000001

/p

descriptor name ..... p
file manager name ..... Scf
device driver name ..... >cnx68230
port address $e0c00081
irq vector 98
irq level 3
irq priority 1
device mode capabilities $3b
device class $00
upper case lock 0
backspace option 0
delete line option 0
echo flag 1
automatic line feed flag 1
end of line null count 0
end of page pause flag 0
page length 66
backspace input character ..... ^H
delete line character ..... ^X
end of record character $0d
end of file character $1a
reprint line character ..... ^D
duplicate line character ..... ^A
pause character ..... ^W
keyboard interrupt character ..... ^C
keyboard quit character ..... ^E
backspace output<T> ^H
line overflow character ..... ^G
parity code $00
adjustable baud rate $00
output device name ..... p
xon character ..... ^Q
xoff character ..... ^S
tab character ..... ^I
tab column width 4

/pps

descriptor name ..... pps
file manager name ..... Scf
device driver name sc68681
port address $e0c00011
irq vector 94
irq level 3
irq priority 3
device mode capabilities $1b
device class $00
upper case lock 0
backspace option 0
delete line option 0
echo flag 0
automatic line feed flag 0
end of line null count 0
end of page pause flag 0
page length 0
backspace input character ..... ^@
delete line character ..... ^@
end of record character $00
end of file character $00
reprint line character ..... ^@
duplicate line character ..... ^@
pause character ..... ^@
keyboard interrupt character ..... ^@
keyboard quit character ..... ^@
backspace output ..... ^@
line overflow character ..... ^@
parity code $00
adjustable baud rate $0e
output device name ..... pps
xon character ..... ^@
xoff character ..... ^@
tab character ..... ^@
tab column width 0
sc68681: global image offset 90

/s0

descriptor name s0
file manager name ..... RBF
device driver name ..... rbvccs
port address $fffe4000
irq vector 69
irq level 4
irq priority 5
device mode capabilities $a7
device class $01
drive number 0
step rate 0
device type $80
density $00
reserved $00
number of cylinders 0
number of heads/sides 0
disk write verification 1
default sectors/track 0
default sectors/track 0 0
segment allocation size 32
sector interleave factor 1
dma transfer mode 0
track base offset 0
sector base offset 0
sector size 0
control word $000b
number of tries (1=no retry) 7
SCSI unit number of drive 0
write precompensation cylinder 0
reduced write current cylinder 0
cylinder to park disk head 0
logical sector offset 0
total cylinders on device 0
SCSI controller id $00
datatransfer/rotation rate $00
SCSI options flags $00000001
maximum transfer count $00ffffff

/s1

descriptor name s1
file manager name ..... RBF
device driver name ..... rbteac
port address $fffe4000
irq vector 69
irq level 4
irq priority 5
device mode capabilities $a7
device class $01
drive number 0
step rate 3
device type $26
density $03
reserved $00
number of cylinders 79
number of heads/sides 2
disk write verification 1
default sectors/track 16
default sectors/track 0 16
segment allocation size 8
sector interleave factor 1
dma transfer mode 0
track base offset 1
sector base offset 1
sector size 256
control word $0002
number of tries (1=no retry) 7
SCSI unit number of drive 0
write precompensation cylinder 0
reduced write current cylinder 0
cylinder to park disk head 0
logical sector offset 0
total cylinders on device 80
SCSI controller id $06
datatransfer/rotation rate $10
SCSI options flags $00000001
maximum transfer count $003ddfd2

/t1

descriptor name t1
file manager name ..... Scf
device driver name sc68681
port address $e0c00001
irq vector 94
irq level 3
irq priority 2
device mode capabilities $23
device class $00
upper case lock 0
backspace option 1
delete line option 0
echo flag 1
automatic line feed flag 1
end of line null count 0
end of page pause flag 1
page length 24
backspace input character ..... ^H
delete line character ..... ^X
end of record character $0d
end of file character $1a
reprint line character ..... ^D
duplicate line character ..... ^A
pause character ..... ^W
keyboard interrupt character ..... ^C
keyboard quit character ..... ^E
backspace output ..... ^H
line overflow character ..... ^G
parity code $00
adjustable baud rate $0e
output device name t1
xon character ..... ^Q
xoff character ..... ^S
tab character ..... ^I
tab column width 4
sc68681: global image offset 90

/t2

descriptor name t2
file manager name ..... Scf
device driver name sc68681
port address $e0c00011
irq vector 94
irq level 3
irq priority 3
device mode capabilities $23
device class $00
upper case lock 0
backspace option 1
delete line option 0
echo flag 1
automatic line feed flag 1
end of line null count 0
end of page pause flag 1
page length 24
backspace input character ..... ^H
delete line character ..... ^X
end of record character $0d
end of file character $1a
reprint line character ..... ^D
duplicate line character ..... ^A
pause character ..... ^W
keyboard interrupt character ..... ^C
keyboard quit character ..... ^E
backspace output ..... ^H
line overflow character ..... ^G
parity code $00
adjustable baud rate $0e
output device name t2
xon character ..... ^Q
xoff character ..... ^S
tab character ..... ^I
tab column width 4
sc68681: global image offset 90

/t3

descriptor name t3
file manager name ..... Scf
device driver name sc68681
port address $e0c00041
irq vector 95
irq level 3
irq priority 2
device mode capabilities $23
device class $00
upper case lock 0
backspace option 1
delete line option 0
echo flag 1
automatic line feed flag 1
end of line null count 0
end of page pause flag 1
page length 24
backspace input character ..... ^H
delete line character ..... ^X
end of record character $0d
end of file character $1a
reprint line character ..... ^D
duplicate line character ..... ^A
pause character ..... ^W
keyboard interrupt character ..... ^C
keyboard quit character ..... ^E
backspace output ..... ^H
line overflow character ..... ^G
parity code $00
adjustable baud rate $0e
output device name t3
xon character ..... ^Q
xoff character ..... ^S
tab character ..... ^I
tab column width 4
sc68681: global image offset 92

/t4

descriptor name t4
file manager name ..... Scf
device driver name sc68681
port address $e0c00051
irq vector 95
irq level 3
irq priority 3
device mode capabilities $23
device class $00
upper case lock 0
backspace option 1
delete line option 0
echo flag 1
automatic line feed flag 1
end of line null count 0
end of page pause flag 1
page length 24
backspace input character ..... ^H
delete line character ..... ^X
end of record character $0d
end of file character $1a
reprint line character ..... ^D
duplicate line character ..... ^A
pause character ..... ^W
keyboard interrupt character ..... ^C
keyboard quit character ..... ^E
backspace output ..... ^H
line overflow character ..... ^G
parity code $00
adjustable baud rate $0e
output device name t4
xon character ..... ^Q
xoff character ..... ^S
tab character ..... ^I
tab column width 4
sc68681: global image offset 92

/t5

descriptor name t5
file manager name ..... Scf
device driver name sc8x30
port address $fffe3802
irq vector 89
irq level 3
irq priority 5
device mode capabilities $23
device class $00
upper case lock 0
backspace option 1
delete line option 0
echo flag 1
automatic line feed flag 1
end of line null count 0
end of page pause flag 1
page length 24
backspace input character ..... ^H
delete line character ..... ^X
end of record character $0d
end of file character $1b
reprint line character ..... ^D
duplicate line character ..... ^A
pause character ..... ^W
keyboard interrupt character ..... ^C
keyboard quit character ..... ^E
backspace output ..... ^H
line overflow character ..... ^G
parity code $00
adjustable baud rate $0e
output device name t5
xon character ..... ^Q
xoff character ..... ^S
tab character ..... ^I
tab column width 4

/t6

descriptor name t6
file manager name ..... Scf
device driver name sc8x30
port address $fffe3800
irq vector 89
irq level 3
irq priority 5
device mode capabilities $23
device class $00
upper case lock 0
backspace option 1
delete line option 0
echo flag 1
automatic line feed flag 1
end of line null count 0
end of page pause flag 1
page length 24
backspace input character ..... ^H
delete line character ..... ^X
end of record character $0d
end of file character $1b
reprint line character ..... ^D
duplicate line character ..... ^A
pause character ..... ^W
keyboard interrupt character ..... ^C
keyboard quit character ..... ^E
backspace output ..... ^H
line overflow character ..... ^G
parity code $00
adjustable baud rate $0e
output device name t6
xon character ..... ^Q
xoff character ..... ^S
tab character ..... ^I
tab column width 4

/term

descriptor name ..... term
file manager name ..... Scf
device driver name sc8x30
port address $fffe3002
irq vector 88
irq level 3
irq priority 5
device mode capabilities $23
device class $00
upper case lock 0
backspace option 1
delete line option 0
echo flag 1
automatic line feed flag 1
end of line null count 0
end of page pause flag 1
page length 24
backspace input character ..... ^H
delete line character ..... ^X
end of record character $0d
end of file character $1b
reprint line character ..... ^D
duplicate line character ..... ^A
pause character ..... ^W
keyboard interrupt character ..... ^C
keyboard quit character ..... ^E
backspace output ..... ^H
line overflow character ..... ^G
parity code $00
adjustable baud rate $0e
output device name ..... term
xon character ..... ^Q
xoff character ..... ^S
tab character ..... ^I
tab column width 4

/timer

descriptor name ..... timer
file manager name ..... Scf
device driver name idle68230
port address $e0c00081
irq vector 92
irq level 6
irq priority 2
device mode capabilities $23
device class $00
upper case lock 0
backspace option 0
delete line option 0
echo flag 0
automatic line feed flag 0
end of line null count 0
end of page pause flag 0
page length 0
backspace input character ..... ^@
delete line character ..... ^@
end of record character $00
end of file character $00
reprint line character ..... ^@
duplicate line character ..... ^@
pause character ..... ^@
keyboard interrupt character ..... ^@
keyboard quit character ..... ^@
backspace output ..... ^@
line overflow character ..... ^@
parity code $00
adjustable baud rate $00
output device name ..... timer
xon character ..... ^@
xoff character ..... ^@
tab character ..... ^@
tab column width 0