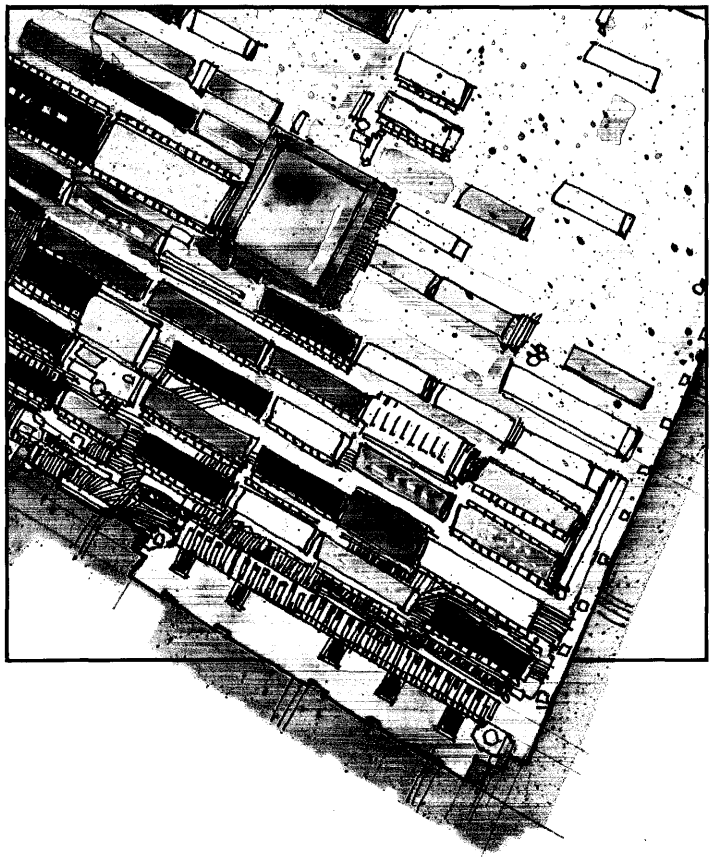
V/SMD 4400 Phoenix

High-Performance SMD
Disk Controller for
Sun Microsystems' Workstations

Installation Guide



INTERPHASE
CORPORATION

V/SMD 4400 Phoenix

High-Performance SMD
Disk Controller for
Sun Microsystems' Workstations

Installation Guide

3200 and 4200 differances

- 3200-3 and below will not handle SMD ECL interface
- 3200-6 and -7 will handle 24mhz SMD drives
- 3200 does not have bus packet (with out this the data transfers accross the VME bus will be slow)
- 4200 has two drive support and the 4400 has four drive support
- 4200 is a 6U board and the 4400 is a 9U board
- 4200 and 4400 will handle 24mhz drives
- 4200 and 4400 has bus packet interface (fast bus transfers)
- 4200 and 4400 have speed enhancements

Document Number UG-0820-000-000 Release Date March 31, 1989

Information in this user document supersedes any preliminary specifications and/or data sheets that may have been made available. Every effort has been made to supply accurate and complete information. However, Interphase Corporation assumes no responsibility for it use. In addition, Interphase reserves the right to make product improvements without prior notice. Such improvements may include, but are not limited to, command codes and error codes.

FOR ASSISTANCE IN USING THE V/SMD 4400 OR ANY OTHER INTERPHASE PRODUCT CALL:

INTERPHASE APPLICATIONS ENGINEERING DEPARTMENT
(214) 350-9000
IN THE U.K., CALL: (0296) 435661

THIS PAGE IS LEFT BLANK INTENTIONALLY

TABLE OF CONTENTS

6 INSTALLATION FOR FULL EMULATION MODE

TABLES

FIGURES

1

V/SMD 4400 PHOENIX INSTALLATION OVERVIEW

Information for all sun user's is included in this V/SMD 4400 Installation Guide.

If you are interested in writing software drivers the V/SMD 4200 User's Guide may be used in conjunction with this installation guide as a technical reference. Contact Interphase Corporation and a user's guide will be shipped to you immediately.

Hardware Reference

Section one deals with the actual hardware installation. Refer to this section when installing the 4400 into the system as it contains information on configuration of the board and system. It also contains specifations, pinouts, and cabling information.

Driver Installation

Section two will show how to build a kernel which will allow you to use the 4400 as an add-on controller in a Sun system. Use this document if your system currently has a Sun supported boot device.

Ipdiag

Section three contains instructions on using the Interphase utility ipdiag to format, partition, and label the drive(s) attached to the 4400. In order to use this utility, you must boot a kernel that includes the Interphase driver.

Boot Installation

Section four explains how to set up the hardware and software to allow booting on the "ip" device. This section assumes that the Interphase controller has firmware Rev. 04C or greater, and is currently installed as an add-on device.

Full Emulation

Section five instructs the user on installing the 4400 as a boot device into a Sun system that has no other boot device available. This is accomplished using full xy emulation firmware rev 04F or greater. This firmware allows the user to format and partition a disk using the SunOS4.0 format utility, and load the SunOS from tape. Using the Sun format should be avoided whenever possible as it is more difficult and time-consuming than using ipdiag. If you are running more than one drive on the 4400, or have access to another boot device, use the Interphase utility to format.

2

HARDWARE REFERENCE

OVERVIEW

Before attempting installation, read through this chapter thoroughly to insure the V/SMD 4400 is installed into your system safely.

The V/SMD 4400 is designed to insure easy installation into the VMEbus system, and can be accomplished in five simple steps:

- Visual Inspection
- On-Board Jumpers
- Set Base Address (Switch Setting Options)
- Power-Off System
- Cabling Procedures

Though the installation of the V/SMD 4400 may be easy, the following **WARNING** must be observed.

WARNING

1. The V/SMD 4400 is extremely sensitive to electrostatic discharge (ESD), and the board could be damaged if it is handled improperly. Interphase ships the board enclosed in a special anti-static bag. Upon receipt of the board, take the proper measures to eliminate board damage due to ESD (i.e., wear a wrist ground strap or other grounding device).

2. Do NOT install or apply power to a damaged board. Failure to observe this warning could result in extensive damage to the board and/or system.

V/SMD 4400-3 is the only variation available for Phoenix at this time. It is an intelligent SMD controller designed specifically for Sun workstations and will support any combination of up to four drives.

For information regarding the board layout, refer to Figure 1, for the position of the cable connections, the jumpers, and the switch block (S1). Please refer to this diagram for information as indicated in the remainder of this section of the user's guide.

If you have any questions regarding installation which are not answered in this chapter, please contact Interphase Customer Service at (214) 350-9000.

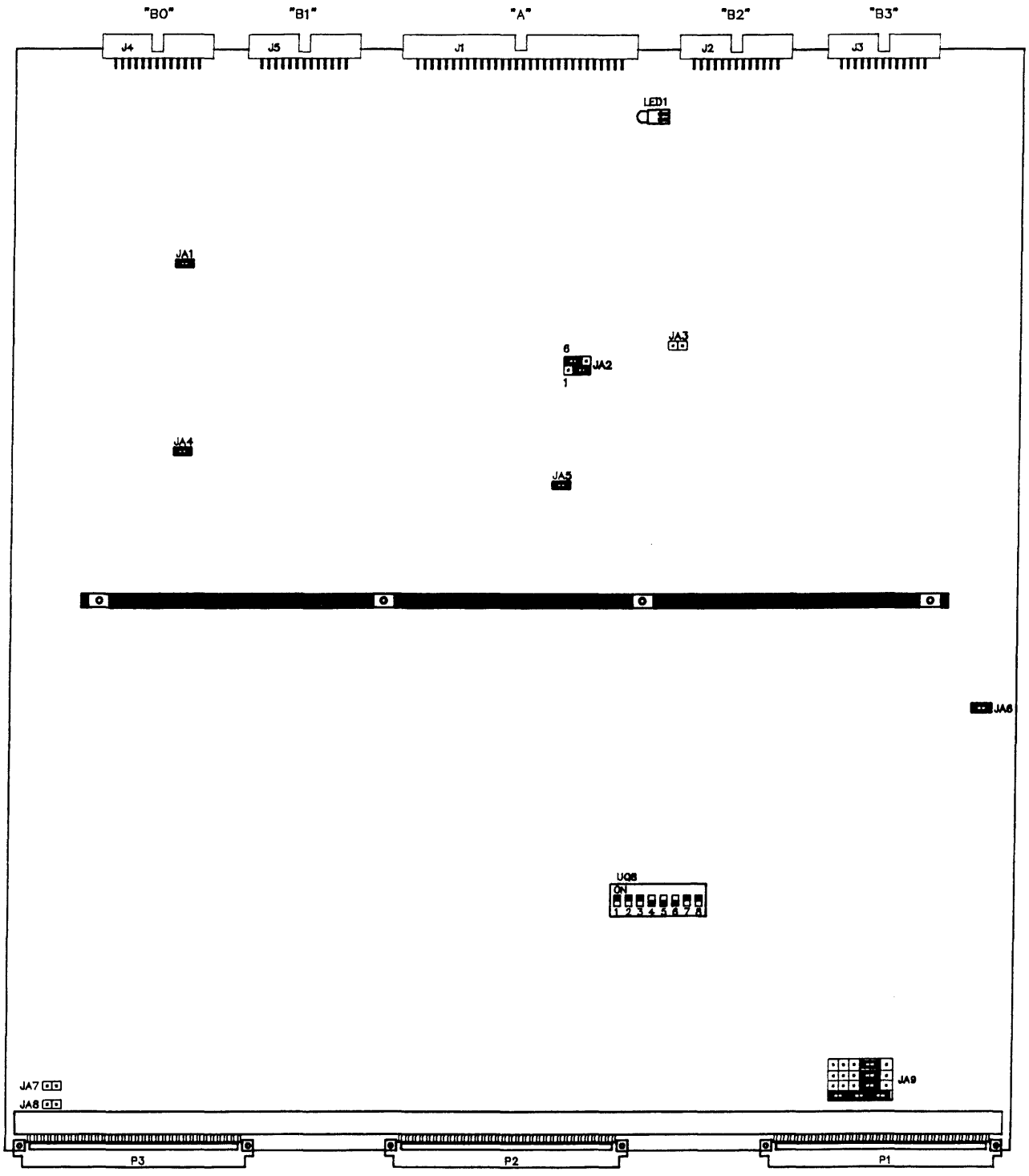


Figure 1. Board Layout V/SMD 4400-3

(The printed circuit board number, PB-0823-A06-000, will be found on the solder side of the board).

For proper installation it is imperative the steps listed below are followed:

STEP 1. Visual Inspection

Before attempting the installation of this board, make sure you are wearing an anti-static or grounding device. Remove the V/SMD 4400 board from the anti-static bag, visually inspect the board to ensure no damage has occurred during shipment. A visual inspection usually is sufficient since each board is thoroughly checked at Interphase just prior to shipment.

If the board is undamaged and all parts are accounted for, proceed with the installation.

STEP 2. On-Board Jumpers

All on-board jumpers on the V/SMD 4400 are default settings which is set at the factory and must not be changed. Refer to the board layout for the physical location of the jumpers in Figure 1.

STEP 3. Set Base Address (Switch Setting Options)

Set the base address of the controller using the switch block (S1, shown in the board layout of Figure 1) so the V/SMD 4400 is properly configured for operation within your system. Table 2 represents all possible base address and switch settings for the controller.

The switch block (S1) contains eight switches numbered 1-8. Switches one through seven correspond to VMEbus address lines A9-A15 respectively. An OFF switch has a value of '1', and an ON switch has a value of '0'.

Switch 8 is used to select the address modifiers for the V/SMD 4400 Short I/O space. If switch 8 is on, only supervisor accesses are permitted (address modifier 2D only). If switch 8 is off, then both 2D and 29 address modifiers are selected.

Table 1. Base Address Switch Settings

BASE ADDRESS SWITCH SETTINGS

O = ON / CLOSE
F = OFF / OPEN

ADDRESS	SWITCH SETTING 7 6 5 4 3 2 1	ADDRESS	SWITCH SETTING 7 6 5 4 3 2 1
0000	OOOOOOO	8000	FOOOOOO
0200	OOOOOOF	8200	FOOOOOF
0400	OOOOOFO	8400	FOOOOFO
0600	OOOOOFF	8600	FOOOOFF
0800	OOOOFOO	8800	FOOOFOO
0A00	OOOOFOF	8A00	FOOOFOF
0C00	OOOOFFO	8C00	FOOOFFO
0E00	OOOOFFF	8E00	FOOOFFF
1000	OOOFOOO	9000	FOOFOOO
1200	OOOFOOF	9200	FOOFOOF
1400	OOOFOFO	9400	FOOFOFO
1600	OOOFOFF	9600	FOOFOFF
1800	OOOFFOO	9800	FOOFFOO
1A00	OOOFFOF	9A00	FOOFFOF
1C00	OOOFFFO	9C00	FOOFFFO
1E00	OOOFFFF	9E00	FOOFFFF
2000	OOFOOOO	A000	FOFOOOO
2200	OOFOOOF	A200	FOFOOOF
2400	OOFOOFO	A400	FOFOOFO
2600	OOFOOFF	A600	FOFOOFF
2800	OOFOFOO	A800	FOFOFOO
2A00	OOFOFOF	AA00	FOFOFOF
2C00	OOFOFFO	AC00	FOFOFFO
2E00	OOFOFFF	AE00	FOFOFFF
3000	OOFFOOO	B000	FOFFOOO
3200	OOFFOOF	B200	FOFFOOF
3400	OOFFOFO	B400	FOFFOFO
3600	OOFFOFF	B600	FOFFOFF
3800	OOFFFOO	B800	FOFFFOO
3A00	OOFFFOF	BA00	FOFFFOF
3C00	OOFFFFO	BC00	FOFFFFO
3E00	OOFFFFF	BE00	FOFFFFF
4000	OFOOOOO	C000	FFOOOOO
4200	OFOOOOF	C200	FFOOOOF
4400	OFOOOFO	C400	FFOOOFO
4600	OFOOOFF	C600	FFOOOFF
4800	OFOOFOO	C800	FFOOFOO
4A00	OFOOFOF	CA00	FFOOFOF
4C00	OFOOFFO	CC00	FFOOFFO
4E00	OFOOFFF	CE00	FFOOFFF
5000	OFOFOOO	D000	FFOFOOO
5200	OFOFOOF	D200	FFOFOOF
5400	OFOFOFO	D400	FFOFOFO
5600	OFOFOFF	D600	FFOFOFF
5800	OFOFFOO	D800	FFOFFOO
5A00	OFOFFOF	DA00	FFOFFOF
5C00	OFOFFFO	DC00	FFOFFFO
5E00	OFOFFFF	DE00	FFOFFFF
6000	OFFOOOO	E000	FFFOOOO
6200	OFFOOOF	E200	FFFOOOF
6400	OFFOOFO	E400	FFFOOFO
6600	OFFOOFF	E600	FFFOOFF
6A00	OFFOFOF	EA00	FFFOFOF
6C00	OFFOFFO	EC00	FFFOFFO
6E00	OFFOFFF	EE00	FFFOFFF
7000	OFFFOOO	F000	FFFFOOO
7200	OFFFOOF	F200	FFFFOOF
7400	OFFFOFO	F400	FFFFOFO
7600	OFFFOFF	F600	FFFFOFF
7800	OFFFFOO	F800	FFFFFOO
7A00	OFFFFOF	FA00	FFFFFOF
7C00	OFFFFFO	FC00	FFFFFFO
7E00	OFFFFFF	FE00	FFFFFFF

STEP 4. Power-Off System

Once the board is configured, ensure that both the system power and the disk drive power are OFF.

WARNING

System power and disk drive power must be OFF before the V/SMD 4400 can be installed. Failure to do so may result in severe damage to the board and/or system.

STEP 5. Cabling Procedures

When the power is off, connect the "A" cable (see Tables 2 through 7 for pin-out details) to the disk drive, making sure that the pins are properly oriented. If only one disk drive is used, it must be attached to the last connector on the cable.

Then, install terminators on the last drive on the cable. (If only one drive is connected, the terminators must be connected to that drive.)

Route the "A" and "B" cables to the proper connector, and insert the V/SMD 4400 about one-third of the way into the slot.

The connection of the cables are as follows:

Connector J1 - "A" cable must be daisy-chained to each drive, 60 pin connector.
Connector J2 - "B" cable for drive 2, 26 pin connector.
Connector J3 - "B" cable for drive 3, 26 pin connector.
Connector J4 - "B" cable for drive 0, 26 pin connector.
Connector J5 - "B" cable for drive 1, 26 pin connector.

Carefully slide the board the rest of the way into the slot. It should slide all the way in without any difficulty. If it doesn't, pull it out and check to make sure that the cables are not in the way.

Once the board is properly seated in the slot 8 through 12. tighten the captive mounting screws on each end of the board.

Table 2. J1 Connector

PIN #	DESCRIPTION	PIN #	DESCRIPTION
1	TAG1-	31	TAG1+
2	TAG2-	32	TAG2+
3	TAG3-	33	TAG3+
4	B0-	34	B0+
5	B1-	35	B1+
6	B2-	36	B2+
7	B3-	37	B3+
8	B4-	38	B4+
9	B5-	39	B5+
10	B6-	40	B6+
11	B7-	41	B7+
12	B8-	42	B8+
13	B9-	43	B9+
14	OCD-	44	OCD+
15	FAULT-	45	FAULT+
16	SKERR-	46	SKERR+
17	ONCYL-	47	ONCYL+
18	INDEX-	48	INDEX+
19	READY-	49	READY+
20	NC	50	NC
21	BUSY-	51	BUSY+
22	SELTAG-	52	SELTAG+
23	U0-	53	U0+
24	U1-	54	U1+
25	SECTOR-	55	SECTOR+
26	U2-	56	U2+
27	U3-	57	U3+
28	WRTPRT-	58	WRTPRT+
29	PICK	59	HOLD
30	B10-	60	B10+

Table 3. J2 and J3

PIN #	J2 DESCRIPTION
1	NC
2	OSCLK-
3	ORXD-
4	GND
5	ORCLK-
6	OTXC-
7	GND
8	OTXD-
9	OSELD+
10	NC
11	GND
12	NC
13	NC
14	OSCLK+
15	GND
16	ORXD+
17	ORCLK+
18	GND
19	OTXC+
20	OXTD+
21	GND
22	OSELD-
23	NC
24	NC
25	GND
26	NC

PIN #	J3 DESCRIPTION
1	NC
2	1SCLK-
3	1RXD-
4	GND
5	1RCLK-
6	1TXC-
7	GND
8	1TXD-
9	1SELD+
10	NC
11	GND
12	NC
13	NC
14	1SCLK+
15	GND
16	1RXD+
17	1RCLK+
18	GND
19	1TXC+
20	1XTD+
21	GND
22	1SELD-
23	NC
24	NC
25	GND
26	NC

Table 4. J4 and J5

PIN #	J4 DESCRIPTION
1	NC
2	2SCLK-
3	2RXD-
4	GND
5	2RCLK-
6	2TXC-
7	GND
8	2TXD-
9	2SELD+
10	NC
11	GND
12	NC
13	NC
14	2SCLK+
15	GND
16	2RXD+
17	2RCLK+
18	GND
19	2TXC+
20	2XTD+
21	GND
22	2SELD-
23	NC
24	NC
25	GND
26	NC

PIN #	J5 DESCRIPTION
1	NC
2	3SCLK-
3	3RXD-
4	GND
5	3RCLK-
6	3TXC-
7	GND
8	3TXD-
9	3SELD+
10	NC
11	GND
12	NC
13	NC
14	3SCLK+
15	GND
16	3RXD+
17	3RCLK+
18	GND
19	3TXC+
20	3XTD+
21	GND
22	3SELD-
23	NC
24	NC
25	GND
26	NC

Table 5. P1 Connector Scheme

Pin	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	VD00	BBUSY*	VD08
2	VD01	BUSCLR*	VD09
3	VD02	ACFAIL*	VD10
4	VD03	BG0IN*	VD11
5	VD04	BG0OUT*	VD12
6	VD05	BG1IN*	VD13
7	VD06	BG1OUT*	VD14
8	VD07	BG2IN*	VD15
9	GND	BG2OUT*	GND
10	SYSCLK	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BUSERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WAT*	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	A17
22	IACKOUT*	SERDAT*	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	+5V STDBY	+12V
32	+5V	+5V	+5V

Table 6. P2 Connector Scheme

Pin	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	NC	+5V	NC
2	NC	GND	NC
3	NC	RESERVED	NC
4	NC	A24	NC
5	NC	A25	NC
6	NC	A26	NC
7	NC	A27	NC
8	NC	A28	NC
9	NC	A29	NC
10	NC	A30	NC
11	NC	A31	NC
12	NC	GND	NC
13	NC	+5V	NC
14	NC	VD16	NC
15	NC	VD17	NC
16	NC	VD18	NC
17	NC	VD19	NC
18	NC	VD20	NC
19	NC	VD21	NC
20	NC	VD22	NC
21	NC	VD23	NC
22	NC	GND	NC
23	NC	VD24	NC
24	NC	VD25	NC
25	NC	VD26	NC
26	NC	VD27	NC
27	NC	VD28	NC
28	NC	VD29	NC
29	NC	VD30	NC
30	NC	VD31	NC
31	NC	GND	NC
32	NC	+5V	NC

Table 7. P3 Connector Scheme

Pin	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	+5V	NC	GND
2	+5V	NC	GND
3	+5V	NC	GND
4	+5V	NC	GND
5	+5V	NC	GND
6	+5V	NC	GND
7	+5V	NC	GND
8	+5V	NC	GND
9	+5V	NC	GND
10	+5V	NC	GND
11	+5V	NC	GND
12	+5V	NC	GND
13	+5V	NC	GND
14	+5V	NC	GND
15	+5V	NC	GND
16	+5V	NC	GND
17	+5V	NC	GND
18	+5V	NC	GND
19	+5V	NC	GND
20	+5V	NC	GND
21	+5V	NC	GND
22	+5V	NC	GND
23	+5V	NC	GND
24	+5V	NC	GND
25	+5V	NC	GND
26	NC	NC	NC
27	NC	NC	NC
28	-12V	NC	-12V
29	-12V	NC	-12V
30	-5V	NC	-5V
31	-5V	NC	-5V
32	-5V	NC	-5V

SPECIFICATIONS OF THE V/SMD 4400

VMEbus Specifications:

DTB Master	A32, D16, D32 DMA transfers
DTB Slave	A16, D8, D16 (Commands & Status)
Requester	Any of R (0-3) (static)
Interrupter	Any of I (1-7) (Dynamic)
Data Rate	Up to 30 Megabytes per second

Environmental Specifications:

Operating Temperature	0-55 degrees C. (32-130 F)
Relative Humidity	10-90% noncondensing

Electrical Specifications:

Power	6.5 amps (max) @ 5V DC (±5%) 4.6 amps nominal 1 amp (max) @ -12V DC (±5%) 0.5 amps nominal

Physical Specifications:

Dimensions (9U)	367 x 400 mm (14.59 x 15.8 in.)
Thickness (1 slot)	19.6 mm (0.77 in.)
Weight	3 lbs. (1.4 kg)

3

V/SMD 4400 DRIVER INSTALLATION PROCEDURE

The following driver installation procedure also applies to the V/SMD 4200.

1. Remove the files from the distribution media and place in the appropriate directory.

   cd /usr/sys
   tar -xvf /dev/rst8 ;1/4 inch cartridge tape
   or
   tar -xvf /dev/rmt0 ;1/2 inch reel tape

   Copy driver and include files (ip.c and ipreg.h) to /usr/sys/sundev

   cp /usr/sys/iphase/ip/drvr/* /usr/sys/sundev/

2. Copy the current system configuration file and add an entry for the Interphase controller.

   cd /usr/sys/conf ;SunOS 3.x
   cd /usr/sys/sun3/conf ;Sun3 OS4
   cd /usr/sys/sun4/conf ;Sun4 OS4
   cp GENERIC IP

   a. Edit the new system configuration file (IP) and add the following entry for the first Interphase controller ipc0, at the end of the new configuration file.

      controller ipc0 at vme16d16 ? csr 0x7000 priority 2 vector ipnint 200 ipeint 201 ipscint 202
      disk ip0 at ipc0 drive 0 flags 0
      disk ip1 at ipc0 drive 1 flags 0
      disk ip2 at ipc0 drive 2 flags 0
      disk ip3 at ipc0 drive 3 flags 0

      NOTE: The first two lines in the entry above are entered as one. The 4400 supports from 1 to 4 drives.

b. If installing in a Sun 3 with Sun OS 4.0, add a line to allow the driver to use the correct "include" files.

```
cpu        "OS4"        #Suport for OS4 in Sun3
```

3. Add the proper reference for the Interphase controller to the "**files**" (**files.sun3** for OS3.x) file. This file is in the same directory as the configuration file.

Add or modify the following line:

```
sundev/ip.c                    optional ip device-driver
```

NOTE: May have "ip not supported".

4. Edit the file /usr/sys/sun/conf.c to include the entry points of the Interphase driver.

a. Add or update the following in the conf.c file :

```
#include "ip.h"
#if NIP > 0
extern int ipopen(), ipstrategy(), ipread(), ipwrite();
extern int ipdump(), ipioctl(), ipsize(), ipclose();
#else
#define     ipopen      nodev
#define     ipclose     nodev
#define     ipstrategy  nodev
#define     ipread      nodev
#define     ipwrite     nodev
#define     ipdump      nodev
#define     ipioctl     nodev
#define     ipsize      0
#endif
```

b. Add to the end of bdevsw structure in the conf.c file:

```
{ ipopen,     ipclose,     ipstrategy,  ipdump,     /*blk#*/
  ipsize,     0 },
```

NOTE: The "blk#" above represents the block device major number.

EXAMPLE:
Change from:

.
.

```
};
int             nblkdev = sizeof (bdevsw) / sizeof (bdevsw[0]);
.
.
```

To:

```
.
.
{ ipopen,       ipclose,    ipstrategy, ipdump,     /*22*/
  ipsize,       0 },
};
int             nblkdev = sizeof (bdevsw) / sizeof (bdevsw[0]);
.
.
```

NOTE: "blk#" has been replaced with the actual block device major number determined by the sequential position of the entry within the structure (in the example above, 22 is used - yours may or may not be the same). This number will be used later in the installation.

For **SunOS3.x**, there is an entry in position 0 of the bdevsw table that must be changed as follows:

Change from:

```
{ ipopen,       nulldev,    ipstrategy, ipdump,     /*0*/
  ipsize,       0 },
```

To:

```
{ nodev,        nodev,      nodev,      nodev,      /*0*/
  nodev,        0 },
```

c. Change the "ip" entry in the cdev structure of the conf.c file.

SunOS4.0 EXAMPLE:

Change from:

```
{
    nodev,    nodev,    nodev,    nodev,    /*4*/
    nodev,    nodev,    nodev,    0,        /*was ip*/
    0,
},
```

To:

```
{
    ipopen,   ipclose,   ipread,   ipwrite,   /*4*/
    ipioctl,  nulldev,   seltrue,  0,
    0,
},
```

SunOS3.x EXAMPLE:

Change from:

```
{
    ipopen,   nulldev,   ipread,   ipwrite,   /*4*/
    ipioctl,  nodev,     nulldev,  0,
    seltrue,  0,         0,
},
```

To:

```
{
    ipopen,   ipclose,   ipread,   ipwrite,   /*4*/
    ipioctl,  nodev,     nulldev,  0,
    seltrue,  0,         0,
},
```

NOTE: The 4 in "/*4*/" above represents the character device major number.

d. If using a Sun4 with SunOS3.2 rev2, change the following:

#if NIPC > 0

to

#if NIP > 0

5. Edit the file "**devices**" or "**devices.sun3**" to include the Interphase designation.

Add or change the following line:

ip blk#

NOTE: "blk#" in the line above must be replaced with the block device major number assigned earlier in step 4.

6. Edit the /dev/MAKEDEV file, changing or adding the Interphase designation, "ip" in the appropriate places.

   EXAMPLE:

   Change from:

```
xd*|xy*|sd*)
    unit='expr $i : '..\(.*\)''
    case $i in
    xd*) name=xd; blk=10; chr=42;;
```

   To:

```
ip*|xd*|xy*|sd*)
    unit='expr $i : '..\(.*\)''
    case $i in
    ip*) name=ip; blk=blk#; chr=4;;
    xd*) name=xd; blk=10; chr=42;;
```

   NOTE: The "blk#" in blk=blk# should be replaced with the block device major number assigned in step 4.

7. Create the appropriate nodes in /dev for communicating with the driver.

   EXAMPLE:

   MAKEDEV ip0 ip1 ip2 ip3

   MAKEDEV must be executed for each drive attached to the controller.

8. Use the config command to place the devices in the configuration process, and generate the new kernel.

   EXAMPLE:

```
cd /usr/sys/conf          ;SunOS3.x
cd /usr/sys/sun3/conf     ;Sun3 OS4.0
cd /usr/sys/sun4/conf     ;Sun4 OS4.0
config IP
```

   After the config is completed:

```
cd ../IP
make
```

Any errors in the installation will normally show up at this time. If no errors occur, save the current working kernel and copy the new kernel to the root partition.

cp /vmunix /vmunix.bak
cp vmunix /vmunix.IP

9. Boot the new kernel and test the driver functions.

   Shut down the system and try the new kernel.

   % halt

   >b xx(0,0,0)vmunix.IP

   (Where xx is the boot device)

   Monitor the boot sequence for the presence of the ipc0 controller.

   EXAMPLE:

   Phoenix 4400 Firmware Rev: 6.040 (9/30/88)
   IP Driver modes < MACSI,Overlapped seeks >.
   ipc0 at vme16d16 7000 vec 0x50 vec 0x51 vec 0x52
   ip0 at ipc0 slave 0

10. Format, partition, and label the drive(s).

    To compile the disk utility, enter:

    cd /usr/sys/iphase/ip/util
    make ipdiag
    ipdiag ip0

    NOTE: Refer to the ipdiag users guide for further information.

11. Create and tune a filesystem on the specific partitions to be used.

    Create a filesystem on the partitions to be used.

    newfs /dev/rip0X
    (where X is one of the partitions a, d, e, f, g, or h)
    Tune the filesystem for faster data I/O.

    tunefs -a 40 -d 0 /dev/rip0X
    (where X is one of the partitions a, d, e, f, g, or h)

4

DISK FORMAT UTILITY

LIST OF COMMANDS

The following commands are supported through the ipdiag program.

attrs	reset attributes and reinit
burst	set dma burst length
dump	dump sector in hex
flawlist	display manufacturers flaw map
flawload	load manufacturers flaw map from file
flawscan	read manufactures flaw map
format	(re)format entire disk
gaps	reset gaps and reinit
labelit	write label onto disk
maplist	print mapped flaw table
mapload	load map table from file
mapout	manually map out a flaw
parts	edit partitions
pconfig	print current configuration
quit	exit program
readuib	display UIB (Unit Initalization Block)
reinit	reinitialize configuration
restore	clear fault and recalibrate
rhdrs	read sector headers
verify	display pack label
scan	scan disk for bad spots
help	display command list

COMMAND DESCRIPTIONS

RESET ATTRIBUTES and REINITIALIZE CONFIGURATION - "**attrs**"

Modify the UIB attribute bits. The utility displays the default attributes determined from the disk label or from the drive type list taken from the disktab file.

All the options available are explained in the disk controller user's guide under the Initialization section.

RESET DMA BURST LENGTH - "**burst**"

Sets the number of VME tranfers before releasing the bus and re-requesting it.

Refer to the controller user's guide for detailed description.

DUMP SECTOR IN HEX - "**dump**"

View the data contents of a given sector in hexadecimal. If no arguments are supplied, the user is prompted for the cylinder, head, and sector.

DISPLAY FLAW MAP - "**flawlist**"

Displays the manufacturers defect list after the list has been read from the drive (with flawscan). Also writes list to a text file (flawlist 'filename'). It is recommended that the flawlist be written to a text file before formatting.

LOAD FLAW MAP from FILE- "**flawload**"

Load the manufacturers defective media list from a text file (written with flawlist) into the utility for processing.

READ MANUFACTURES FLAW MAP - "**flawscan**"

Scans the drive for the manufacturers defect list. This command will determine the type of device being used (SMD or ESDI) and check the proper areas according to the interface specification.

FORMAT ENTIRE DISK - "**format**"

Formats disk drive. This command does not allow the user to perform a partial format.

RESET GAPS and REINITIALIZE - "**gaps**"

Changes intersector gaps. This command defaults to the gaps from the current UIB. Gaps are entered in number of words (decimal).

WRITE LABEL ONTO DISK - "**labelit**"

Writes the label information to the disk after formatting.

PRINT MAPPED FLAW TABLE - "**maplist**"

Displays the tracks that have been mapped to the alternate forwarding area, and the sectors that have been slipped. This command also writes the maplist to a text file (maplist 'filename').

LOAD MAP TABLE FROM FILE - "**mapload**"

Reads in an ASCII text file (created with maplist) that contains the defective sector and track information.

MANUALLY MAP OUT A DEFECT - "**mapout**"

Allows the user to selectively slip a sector or map a track.

EDIT PARTITION TABLE - "**parts**"

Modify the partition table retrieved from the disktab file or to manually enter a new one. The ipdiag utility does not modify the disktab file.

PRINT CURRENT CONFIGURATION - "**pconfig**"

Displays the controller information, the firmware rev., and all other current configuration information from internal tables within the utility.

EXIT PROGRAM - "**quit**"

Terminates the ipdiag program.

READ AND DISPLAY DRIVE UIB - "**readuib**"

Reads the device UIB (Unit Initialization Block) from the controller and displays it.

REINITIALIZE CONFIGURATION - "**reinit**"

This command allows the user to select the desired drive type from the disktab file or manually enter a UIB and reinitializes the controller.

CLEAR FAULT and RECALIBRATE - "**restore**"

Clears drive faults and recalibrates the drive.

READ SECTOR HEADERS - "**rhdrs**"

Displays the sector headers for the selected track, and will step one track at a time.

DISPLAY PACK LABEL - "**verify**"

This command reads the disk label.

SCAN DISK FOR BAD SPOTS - "**scan**"

This command allows the user to scan selective areas of the disk for defective media. If bad areas are encountered, ipdiag will automatically map the bad track or slip the sector if slip sectoring is enabled.

DISPLAY COMMAND LIST - "**help**"

This command displays the available commands for the ipdiag utility.

DISK GEOMETRY, PARTITION, and UIB LAYOUT TABLE - (The 'disktab' file)

The first field is a drive type - it is currently not used. The second field is a drive type name, and is what the utility will display. The utility will use the first field if the second field is absent.

Numeric fields should be entered as follows:

key#n	key is the keyname and n is in decimal
key#0n	key is the keyname and n is in octal
key#0xn	key is the keyname and n is in hex

Key:

ty	type of disk 'esdi' ¦ 'smd' ¦ 'scsi'
nh	#heads/cylinder
ns	#sectors/track
nc	#cylinders/disk
na	#alternate cylinders for track mapping
skew	#Spiral SKEW
g1	#gap 1
g2	#gap 2
RSE	Runt Sector Enable
SSE	Spare Sector Enable
CE	Cacheing Enable

DLP	Dual Port Enable (SMD only)
INH	Increment by head
MBD	Move Bad Data
RSK	Reseek Enable
s[a-h]	partition starting cylinder
l[a-h]	partition length in cylinders (not blocks)

Example ESDI entry

```
1355|Micropolis 1355:\
    :ty=esdi:nh#8:ns#35:nc#901:na#4:\
    :skew#30:g1#0x08:g2#0x08:\
    :CE:INH:RSK:\
    :sa#0:la#75:\
    :sb#75:lb#100:\
    :sc#0:lc#905:\
    :sg#175:lg#725:
```

Example SMD entry

```
C9720-1230|CDC 9720-1230 83spt+SSE:\
    :ty=smd:nh#15:ns#83:nc#1633:na#2:\
    :skew#71:g1#0x0d:g2#0x08:\
    :SSE:CE:INH:RSK:\
    :sa#0:la#24:\
    :sb#24:lb#100:\
    :sc#0:lc#1633:\
    :sd#124:ld#476:\
    :se#600:le#600:\
    :sg#1200:lg#433:
```

DEFECT LIST FILE FORMAT

The manufacturer's defect list is written on the drive at the factory. It is recommended that this list be written to a text file (flawlist) because the information is destroyed the first time the drive is formatted.

The manufacturers defect list is read by both flawscan and format.

Following are examples entries that may appear in a flawlist file:

```
  20     3 17190    27
  21     3 17190    25
 920     7 49920     8
 926     4 13170     3
* 956    11 10367    6
```

The first line indicates an error in cylinder 20 head 3 at byte 17190 and is 27 bits in length.

The asterisk denotes CDC recommendation for forwarding the entire track.

Following are examples entries that may appear in a "maplist" file:

```
156     3    23
157     3    23
1246   14    30
1313    4    -1
1316    4    -1
```

The first line indicates an error in cylinder 156 head 3 at sector 23 and the sector has been slipped.

The -1 in the last field indicates that the entire track has been mapped to the forwarding area.

5

BOOT DEVICE INSTALLATION

PURPOSE

With the release of the Phoenix/Cheetah/Panther firmware rev 04C, it is now possible to boot your Sun system with an Interphase controller. This document explains the procedure that must be followed to set up the Interphase disk to boot.

ASSUMPTIONS

In order to install the bootable Interphase device in the system, the user must meet the following conditions:

- Know basic UNIX commands.
- Know how to use an editor like vi.
- Have a general understanding of a driver installation.
- The system must have a Sun supported boot device installed.
- Have adequate hardware knowledge to install the board in the system.
- The Interphase controller must be installed and working as an "add-on" device and the block device major number assigned to the 'ip' devices must be known.

OVERVIEW

During the boot process the Interphase disk controller emulates the basic Xy 451 commands needed to read and execute the kernel. Once the kernel is up and running, and the first Interphase command is executed, Xy emulation ceases. The following describes the steps to create a bootable "ip" device:

1. Edit swapgeneric.c to include support for "ip".
2. Create a new kernel, with the "ip" controller addressed at 0xee00.
3. Copy the files from the current boot device to the "ip" device.
4. Edit the fstab file so the system will mount the correct partitions at boot time.
5. Execute installboot to put the bootstrap on the "ip" device.
6. Physically address the "ip" controller at 0xee00.

LIMITATIONS

The first Xy 451 controller in the system is addressed from 0xee40 to 0xee47 (8 bytes), while the Interphase controller is addressed from 0xee00 to 0xefff (512 bytes). In order for the system to boot properly, there cannot be any other devices addressed between 0xee00 and 0xefff. The Xylogics disk and tape controllers fall into this memory space and must be either removed or readdressed to a memory location other than 0xee00 thru 0xefff. If Xylogics 451 controller is used as an add on, do not configure it as controller 0.

CREATING THE INTERPHASE BOOT KERNEL

A configuration file should already exist that can be modified to create the Interphase boot kernel. Copy current configuration file to a new file and edit it in the following manner:

1. Change directory to the "conf" directory and copy the current configuration file to a new file.

```
cd /usr/sys/conf                ; Sun3 and Sun4 SunOS3.x
cd /usr/sys/sun3/conf           ; Sun3 SunOS4.0
cd /usr/sys/sun4/conf           ; Sun4 SunOS4.0

cp IP BIP       ; copy configuration file with ip support
vi BIP          ; edit new configuration file
```

2. Change the ident line.

```
from    ident   GENERIC
to      ident   "IPHASE_BOOT"
```

3. Change the config line.

```
from    config  vmunix  swap generic
to      config  vmunix  root ip0
```

4. Change the address for the Interphase controller from 0x7000 to 0xee00 as shown below:

```
controller  ipc0 at vme16d16 ? csr 0xee00 priority 2 vector
ipnint 200 ipeint 201 ipscint 202
disk            ip0 at ipc0 drive 0 flags 0
disk            ip1 at ipc0 drive 1 flags 0
disk            ip2 at ipc0 drive 2 flags 0
disk            ip3 at ipc0 drive 3 flags 0
```

NOTE: The 4400 supports up to four drives.

5. The following lines must be commented out if they exist (insert a '#' at the beginning of each line) or addressed to a location other than 0xee00 thru 0xefff.

```
#controller xyc0 at vme16d16 1 csr 0xee40 priority 2
vector yintr 0x48
#controller xyc1 at vme16d16 1 csr 0xee48 priority 2
vector xyintr 0x49
#disk   xy0 at xyc0 drive 0
#disk   xy1 at xyc0 drive 1
#disk   xy2 at xyc1 drive 0
#disk   xy3 at xyc1 drive 1
```

```
#controller xdc0 at vme16d32 ? csr 0xee80 priority 2
vector xdintr 0x44
#controller xdc1 at vme16d32 ? csr 0xee90 priority 2
vector xdintr 0x45
#controller xdc2 at vme16d32 ? csr 0xeea0 priority 2
vector xdintr 0x46
#controller xdc3 at vme16d32 ? csr 0xeeb0 priority 2
vector xdintr 0x47
#disk   xd0 at xdc0 drive 0
#disk   xd1 at xdc0 drive 1
#disk   xd2 at xdc0 drive 2
#disk   xd3 at xdc0 drive 3
#disk   xd4 at xdc1 drive 0
#disk   xd5 at xdc1 drive 1
#disk   xd6 at xdc1 drive 2
#disk   xd7 at xdc1 drive 3
#disk   xd8 at xdc2 drive 0
#disk   xd9 at xdc2 drive 1
#disk   xd10 at xdc2 drive 2
#disk   xd11 at xdc2 drive 3
#disk   xd12 at xdc3 drive 0
#disk   xd13 at xdc3 drive 1
#disk   xd14 at xdc3 drive 2
#disk   xd15 at xdc3 drive 3

#controller xtc0 at vme16d16 ? csr 0xee60 priority 3
vector xtintr 0x64
#controller xtc1 at vme16d16 ? csr 0xee68 priority 3
vector xtintr 0x65
#tape   xt0 at xtc0 drive 0 flags 1
#tape   xt1 at xtc1 drive 0 flags 1
```

6. Find the block device major number for "ip" devices if not known.

```
ls -l /dev/ip*
brw-r-----  1 root     22,  0 Dec 19 11:20 /dev/ip0a
brw-r-----  1 root     22,  1 Dec 19 11:20 /dev/ip0b
brw-r-----  1 root     22,  2 Dec 19 11:20 /dev/ip0c
brw-r-----  1 root     22,  3 Dec 19 11:20 /dev/ip0d
brw-r-----  1 root     22,  4 Dec 19 11:20 /dev/ip0e
brw-r-----  1 root     22,  5 Dec 19 11:20 /dev/ip0f
brw-r-----  1 root     22,  6 Dec 19 11:20 /dev/ip0g
brw-r-----  1 root     22,  7 Dec 19 11:20 /dev/ip0h
```

NOTE: In this example, the block device major number is 22. If this number is 0 and you are running OS4.0, 'conf.c', 'files', and 'MAKEDEV' files must be changed. New /dev entries must also be created if the major number is changed. Refer to the Device Driver Installation Guide for further information.

7. Modify swapgeneric.c file.

```
cd /usr/sys/sun        ; change to correct directory

vi swapgeneric.c       ; edit file
```

Add or modify the following lines:

```
.
.
#include "ip.h"        ; add this entry above xy entry
#if NIP > 0
extern  struct mb_driver ipcdriver;
#endif

#include "xy.h"        ; this will already exist
#if NXY > 0
extern  struct mb_driver xycdriver;
#endif
.
.
.
#if  NIP > 0           ; add this entry above xy entry
   {"ip", &ipcdriver, makedev(blk#, 0)},      ; see note
#endif

#if  NXY > 0           ; this will already exist
   {"xy", &xycdriver, makedev(3, 0)},
#endif
.
.
.
#if  NIP > 0           ; add this entry above xy entry
   {"ip", &ipcdriver, makedev(4, 0), makedev(blk#,0)},
#endif                 ;see note
```

```
#if NXY > 0              ; this will already exist
   {"xy", &xycdriver, makedev(9, 0), makedev(3, 0)},
#endif
.
.
```

NOTE: The "blk#" in "makedev(blk#, 0)}" must be replaced with the block device major number for the "ip" devices.

8. Compile the new boot kernel.

```
cd ../conf  or  cd conf      ; change directory

config BIP                   ; configure new kernel
cd ../BIP                    ; config will create the directory ../BIP
make                         ; compile new kernel
cp /vmunix /vmunix.bak       ; back up old kernel
cp vmunix /vmunix.BIP        ; copy kernel to root directory
cd /                ; goto root directory
```

CREATE NEW FILE SYSTEMS ON THE INTERPHASE BOOT DISK.

A new file system should be created on the Interphase boot drive, and all partitions of the current boot device must be copied to the corresponding partitions of the Interphase boot drive.

1. Create and tune new file systems.

```
newfs /dev/rip0a      ; run newfs on each partition to be
                        mounted
newfs /dev/rip0d
newfs /dev/rip0e
newfs /dev/rip0f
newfs /dev/rip0g
newfs /dev/rip0h

tunefs -d 0 -a 40 /dev/rip0a      ; run tunefs on each new
                                    filesystem
tunefs -d 0 -a 40 /dev/rip0d
tunefs -d 0 -a 40 /dev/rip0e
tunefs -d 0 -a 40 /dev/rip0f
tunefs -d 0 -a 40 /dev/rip0g
tunefs -d 0 -a 40 /dev/rip0h
```

mount /usr /dev/xx0p ; mount all partitions of current boot device - where xx is current boot device and p is partition.

2. Copy all files in each partition of the current boot device to the corresponding "ip" disk partition.

   Create directories only for the partitions that are used.

```
cd /            ; move to root directory
mkdir ip0a      ; make directory for each partition
mkdir ip0d
mkdir ip0e
mkdir ip0f
mkdir ip0g
mkdir ip0h
```

   Mount only the Interphase partitions that are used.

```
mount /dev/ip0a /ip0a
mount /dev/ip0d /ip0d
mount /dev/ip0e /ip0e
mount /dev/ip0f /ip0f
mount /dev/ip0g /ip0g
mount /dev/ip0h /ip0h
```

   Copy the files from each partition of the current boot disk to the corresponding "ip" partition.

```
dump 0f - /dev/rxx0a  | (cd /ip0a;restore xf - )
dump 0f - /dev/rxx0d  | (cd /ip0d;restore xf - )
dump 0f - /dev/rxx0e  | (cd /ip0e;restore xf - )
dump 0f - /dev/rxx0f  | (cd /ip0f;restore xf - )
dump 0f - /dev/rxx0g  | (cd /ip0g;restore xf - )
dump 0f - /dev/rxx0h  | (cd /ip0h;restore xf - )
```

   ; xx = Sun boot device

   NOTE: Only execute the dump and restore commands on the partitions that the Sun boot disk uses.

MOVE THE BOOT FILES TO THE DISK

The Interphase disk must contain the bootstrap code and the boot file (bootxy) before it can be booted. This is accomplished with installboot.

cd /usr/mdec

Sun3 OS3.x :
installboot bootxy /dev/rip0a

Sun3 OS 4.0, Sun4 :
installboot /ip0a/boot bootxy /dev/rip0a

UPDATE FSTAB FILE

The fstab file on the "a" partition of the Interphase boot device must be updated.

cd /ip0a/etc ; change directory
vi fstab ; edit fstab file

Change the file as follows:

Example:

from **/dev/xx0a / 4.2 rw 1 1** ; where xx = boot device
to **/dev/ip0a / 4.2 rw 1 1**

NOTE: Change xx to ip for all partitions. Make sure this file is edited correctly, or system will not mount root.

Halt the system.

sync;halt

Power System Down

HARDWARE INSTALLATION

Remove Interphase controller and change switch settings to the proper address as shown below: (change from 7000 to ee00)

Switch Number	Position 0x70000	Position 0xee00
1	on	**off**
2	on	**off**
3	on	**off**
4	off	**on**
5	off	**off**
6	off	**off**
7	on	**off**
8	off	**off**

Place the Interphase controller in the Sun card cage. Slots 8 through 12 may be used. If a 6u board is used (Cheetah or Panther) it should be placed in the upper two VME connectors of the chosen slot or placed in a 9u adapter board.

Power System Up.

MODIFY THE SUN EEPROM

The EEPROM must be set up to think it is booting off of the Xy disk controller. This is accomplished by doing the following:

```
L1 A                          ; abort autoboot
>q 18                         ; open EEPROM
>EEPROM 018: 00 ? 12
>EEPROM 019: 00 ? 78          ; set for xy boot device
>EEPROM 01A: 00 ? 79
>EEPROM 01B: 00 ? 00
>EEPROM 01C: 00 ? 00
>EEPROM 01D: 00 ? 00
>EEPROM 01E: 00 ?            ; press space bar and return
>
```

NOTE: The system will not work in boot device polling mode (EEPROM address 018 = 00). This is because the xd device is polled first and the Interphase controller sends back a dtack. The system believes that the xd boot device exists and will attempt to communicate with it instead of the xy device.

BOOTING WITH THE NEW BOOT DISK

The system is now ready to boot off of the Interphase disk.

Test new kernel:

```
L1 A                          ; abort autoboot
>b xy()vmunix.BIP -s          ; test Interphase boot kernel
```

If the system boot with no problems, rename kernel.

```
mv vmunix.BIP vmunix
sync;halt                     ; bring system down
```

Power the system down/up and the autoboot process should boot the system. If the message "waiting for disk to spin up..." appears you must boot manually.

```
>b                            ; boot system
```

6

INSTALLATION FOR FULL EMULATION MODE

PURPOSE

With the release of the latest 4400 Phoenix/4200 Cheetah/4201 Panther (05A) firmware or above, it is now possible to format, load tapes and boot your Sun system with an Interphase controller. This document explains the procedure that must be followed to set up the Interphase disk to allow booting.

ASSUMPTIONS

In order to install the bootable Interphase device in the system, the customer must have the following skills:

- Be able to build a virgin system.
- Know basic UNIX commands.
- Know how to use an editor like vi.
- Have a general understanding of a driver installation.
- Have adequate hardware knowledge to install the board in the system.
- Use the Sun supplied documentation for building a virgin system with SunOS4.0 (OS3.x not fully tested). This document shows deviations from the normal install.

OVERVIEW

Where possible, use the Interphase utility "ipdiag" to format and partition the drive (refer to Ipdiag Users Guide). This utility is a "runtime" program that requires the "ip" driver to be loaded in the kernel. If this is not possible - ie. no other boot device -, use the Sun "format" utility to format and partition the disk.

This document was written to support the user building the system from scratch using full emulation. Use this document and the Sun supplied documentation to complete this process. During the boot process, the Interphase disk controller emulates Xy 451 commands. Once the kernel with the ip driver is up and running and the first Interphase command is executed, Xy emulation ceases. The following describes the steps to create an Interphase bootable system:

1. Place Interphase board in system at address 0xee00.
2. Format the Interphase disk as if it were a xy451, using the Sun supplied utility.
3. Load the Sun distribution tapes to the Interphase disk as if it were a xy disk.
4. Boot the system in xy mode.
5. Build a new kernel that includes the Interphase driver.
6. Halt the system and reboot new kernel.

LIMITATIONS

The first xy451 disk controller used is addressed from 0xee40 to 0xee47 (8 bytes), while the Interphase controller is addressed from 0xee00 to 0xefff (512 bytes).In order for the system to execute properly, there cannot be any other device mapped between 0xee00 and 0xefff. The Xylogics disk and tape controllers fall into this memory space and must be moved to a different part of memory or removed. If Xylogics 451 controller is used as an add on, **do not** configure it as controller 0.

Formatting the Interphase disk with the Sun utility is more difficult and time-consumming than with Ipdiag. For this reason it is recommended that ipdiag be used whenever possible.

The controller should run in emmulation mode only long enough to format, load the OS and build a new kernel. During normal operation, always use a kernel with the ip driver.

HARDWARE INSTALLATION

Set the Interphase controller switch settings to the proper address as shown below:

Switch Number	Position 0xee00
1	**off**
2	**off**
3	**off**
4	**on**
5	**off**
6	**off**
7	**off**
8	**off**

Power the Sun system down. Place the Interphase controller in the Sun card cage. Slots 8 through 12 may be used. If a 6u board is used (Cheetah or Panther) it should be placed in the upper two VME connectors of the slot chosen or placed in a 9u adapter board.

On the back of the card cage there are jumpers that must be removed. Find the jumper block for the slot that the controller was placed in and remove the bus grant 3 and IACK jumpers. The BG3 jumper is the fourth jumper down in a group of four. The IACK jumper is the fifth jumper down and is by itself.

Attach the cables from the controller to the drive.

FORMAT AND PARTITION INTERPHASE DISK

The Interphase disk needs to be formatted, partitioned, and labeled before the OS can be loaded (if using ipdiag to format, skip this section and refer to the Ipdiag Users Guide). Load MUNIX per Sun's installation procedure to system memory. Boot MUNIX and execute the formatter.

NOTE: User input is shown in **bold** print.

1. Power up system and boot mini unix:

```
>b st()
Boot: st(0,0,0)
Boot: st(0,0,4) -asw
...
Ethernet address = x:x:x:x:x:x
xdc0: controller reset failed
xdc1: controller reset failed
xdc2: controller reset failed
xdc3: controller reset failed
xyc0 at vme16d16 0xee40 vec 0x48
xy0 at xyc0 slave 0
xy0: unable to read label
or xy0: <drive name cyl xxxx alt x hd xx sec xx>
   xyc1: controller reset failed
   si0 at vme 24d16 0x200000 vec 0x40
   ...
   root filesystem type (spec 4.2 nfs) : 4.2
   root device (rd%d[a-h]) : rd0a
   init ram disk from [st1, st0] : st1
   tape file number? 5
   swap filesystem type (spec 4.2 nfs) : spec
   swap device (ns%d[a-h]) : ns0a
   #
```

NOTE: The controller reset failed errors are normal.

2. Execute format program:

Before the disk can be formatted, the Interphase UIB must be set up. This is accomplished by issuing the type command twice. The first type command sets the gaps, number of alternate cylinders and the drive attributes. The second type command sets the drive geometry (number of cylinders, heads and sectors).

NOTE: All values are entered in decimal.

```
# format
Specify disk (enter its number): 0
```

If the drive has a valid label the following will appear.

```
Need info -- Enter total bytes/sector [600]: xx
   ; enter total number of bytes per sector
Need info -- Enter drive type [0]: x
   ; ALWAYS use a spare sector
   ; where x equals one of the following:
   ; 0 - no spare and no runt (do not use)
   ; 1 -    spare and no runt
   ; 2 - no spare and    runt (do not use)
   ; 3 -    spare and    runt
```

3. First type command:

This first type command does not use drive parameters as would be expected. The firmware on the Interphase controller will look at the cylinder count, and if it is less than 12, will go into a special mode to set the gaps, alternate cylinders and the attribute byte.

```
format> type
Specify disk type (enter its number) [7]: 8
   ; ALWAYS choose other
   ; this may be 9 if a valid label exists on the disk
Enter number of data cylinders: 1
Enter number of alternate cylinders [2]: xx
   ; this number must be between 2 and 10
Enter number of physical cylinders [xx]: <cr>
Enter number of heads: xx
   ; number of words in gap2
Enter number of data sectors/track: xx
   ; number of words in gap1
Enter rpm of drive [3600]: <cr>
Enter total bytes/sector: xx
   ; enter total number of bytes per sector
Enter drive type: x
   ; ALWAYS run with a spare sector
   ; where x equals one of the following:
   ; 0 - no spare and no runt (do not use)
   ; 1 -    spare and no runt
   ; 2 - no spare and    runt (do not use)
   ; 3 -    spare and    runt
Enter disk type name (remember quotes): "drive name uib"
```

4. Second type command:

   This second type command will function normally. The cylinder count must be greater than 12.

```
format> type
Specify disk type (enter its number) [8]: 9
    ; ALWAYS choose other
    ; this may be 10 if a valid label exists on the disk
Enter number of data cylinders: xx
Enter number of alternate cylinders [2]: xx
    ; this number must equal the number entered in the
    ; first type command
Enter number of physical cylinders [xx]: <cr>
Enter number of heads: xx
    ; number of heads per cylinder
Enter number of data sectors/track: xx
    ; number of sectors per track not including runt or
    ; spare (ex. total sec = 48 but running runt and spare,
    ; then xx will equal 46)
Enter rpm of drive [3600]: <cr>
Enter total bytes/sector: xx
    ; enter total number of bytes per sector
Enter drive type: x
    ; this number must equal the number entered in the
    ; first type command
Enter disk type name (remember quotes): "drive name"
```

5. Defect command:

   If the manufactures defect list still exists on the disk it will be extracted with this command.

```
format> defect
defect> original
Ready to update working list. This cannot be interrupted and
may take a long while. Continue? y
```

The following may appear, if the disk has already been formatted and the manufactures defect list does not exist.

```
Disk is not fully encoded with defect info.
Do you wish to stop extracting? y
defect> q
Warning: working defect list modified; but not committed
Do you wish to commit changes to current defect list? y
```

NOTE: Follow the Installing the SunOS[tm] documentation on how to save manufactures defect list to tape.

6. Format and verify alternate cylinders:

The alternate cylinders must be formatted first. This is because the Sun software will map a sector if there are no more spare sectors available on the track. The Interphase controller will not map sectors, instead it will map tracks. This is why you should always run with a spare sector. The Interphase controller only has one spare sector per track. If there is more then one bad sector on a track, the controller will map the whole track. If all the cylinders are formatted and verified at the same time the Sun software will not work properly if a track is mapped.

a. Analyze command for alternate cylinders:

```
format> analyze
analyze> setup
Analyze entire disk [yes]? no
Enter starting block number [0, 0/0/0]: xx/xx/xx
; where xx/xx/xx equals the start of the alternate
; cylinders. ex. # of data cyl = 1200 and the
; # of alt cyl = 4 then the starting block will
; be 1200/0/0.
```

```
Enter ending block number [yy, yy/yy/yy]: xx/xx/xx
    ; where xx/xx/xx equals the end of the alternate
    ; cylinders. in the example above it would be
    ; 1203/9/47 if the number of heads = 10 and the
    ; number of sectors = 48.
Loop continuously [no]? <cr>
Enter number of passes [2]: 10
Repair defective blocks [yes]? <cr>
Stop after first error [no]? <cr>
Use random bit patterns [no]? <cr>
Enter number of blocks per transfer [126, xx/xx/xx]: 0/1/0
Verify media after formatting [yes]? <cr>
Enable extended messages [no]? <cr>
analyze> q
```

b. Format command for alternate cylinders

The format program is now set up and the alternate cylinders can be formatted. Once this process begins do not ^C out of it until it is finished or has run a minimum of ten passes.

```
format> format
Enter starting block number [0, 0/0/0]: xx/xx/xx
    ; enter the start of the alternate cylinders.
    ; in the example above this would be equal to
    ; 1200/0/0.
Enter ending block number [yy, yy/yy/yy]: xx/xx/xx
    ; enter the end of the alternate cylinders.
    ; in the example above this would be equal to
    ; 1203/9/47.
Ready to format. Formatting cannot be interrupted and takes x
minutes (estimated). Continue? y
```

NOTE: If any of the tracks on the alternate cylinders have more than 1 bad sector, that track cannot be used. If this is the case, the number of data cylinders must be lowered so the bad track in the alternate area is not used. To do this exit the format program and begin again.

7. Format and verify data cylinders:

   a. Analyze command for data cylinders:

```
format> analyze
analyze> setup
Analyze entire disk [yes]? no
Enter starting block number [0, 0/0/0]: xx/xx/xx
    ; where xx/xx/xx equals the start of the data
    ; cylinders.  ex. # of data cyl = 1200 and the
    ; # of alt cyl = 4 then the starting block will
    ; be 0/0/0.
Enter ending block number [yy, yy/yy/yy]: xx/xx/xx
    ; where xx/xx/xx equals the end of the data
    ; cylinders.  in the example above it would be
    ; 1199/9/47 if the number of heads = 10 and the
    ; number of sectors = 48.
Loop continuously [no]? <cr>
    ; it is best to say yes and let run over night
Enter number of passes [2]: 10
Repair defective blocks [yes]? <cr>
Stop after first error [no]? <cr>
Use random bit patterns [no]? <cr>
Enter number of blocks per transfer [126, xx/xx/xx]: 0/1/0
Verify media after formatting [yes]? <cr>
Enable extended messages [no]? <cr>
analyze> q
```

   b. Format command for data cylinders:

      The format program is now set up and the data cylinders can be formatted. Once this process begins do not ^C until the format phase is completed.

```
format> format
Enter starting block number [0, 0/0/0]: xx/xx/xx
    ; enter the start of the data cylinders.
    ; in the example above this would be equal to
    ; 0/0/0.
Enter ending block number [yy, yy/yy/yy]: xx/xx/xx
    ; enter the end of the data cylinders.
    ; in the example above this would be equal to
    ; 1199/9/47.
Ready to format. Formatting cannot be interrupted and takes x
minutes (estimated). Continue? y
```

NOTE: The number of minutes may be different than estimated.

8. Partition command:

   Partition the disk as desired.

```
format> partition
partition> a
    ; do this command for each partition used
Enter new starting cyl [0]: xx
    ; enter starting cyl
Enter new # blocks [yy, zz/zz/zz]: xx
    ; enter number of blocks or cylinders.
partition> print
    ; verify that the partition table is correct
partition> label
Ready to label disk, continue? y
partition> q
```

9. Exit format program:

```
format> q
```

LOADING THE SUN DISTRIBUTION TAPES

The file system can be built at this time and the files loaded from tape. Follow the Sun installation guide provided with the OS 4.0 operating system (Installing the SunOS[tm]).

```
>b st()
Boot: st(0,0,0)
Boot: st(0,0,2)
Size ...
Standalone Copy
From: st(0,0,3)
To: xy(0,0,1)
Copy completed ... bytes
Boot: xy(0,0,1)vmunix -asw
...
Ethernet address = x:x:x:x:x:x
xdc0: controller reset failed
xdc1: controller reset failed
xdc2: controller reset failed
xdc3: controller reset failed
xyc0 at vme16d16 0xee40 vec 0x48
xy0 at xyc0 slave 0
xy0: <drive name cyl xxxx alt x hd xx sec xx>
xyc1: controller reset failed
si0 at vme 24d16 0x200000 vec 0x40
...
root filesystem type (spec 4.2 nfs) : 4.2
root device (xy%d[a-h]) : xy0b
swap filesystem type (spec 4.2 nfs) : spec
swap device (xy%d[a-h]) : xy0b
Swapping on root device? y
...
# suninstall
```

Set up system as if using xy451, per suninstall documentation. After the SunOS is loaded, reboot.

```
#halt
L1 A

>b xy()
Boot: xy(0,0,0)
Size ...
...
```

```
Ethernet address = x:x:x:x:x:x
xdc0: controller reset failed
xdc1: controller reset failed
xdc2: controller reset failed
xdc3: controller reset failed
xyc0 at vme16d16 0xee40 vec 0x48
xy0 at xyc0 slave 0
xy0: <drive name cyl xxxx alt x hd xx sec xx>
xyc1: controller reset failed
si0 at vme 24d16 0x200000 vec 0x40
...
systemname login: root
systemname#
```

The system will be running in xy mode. The Interphase kernel must be made as shown below.

CREATING THE INTERPHASE BOOT KERNEL

1. Remove the files from the distribution media and place in the appropriate directory.

   ```
   cd /usr/sys
   tar -xvf /dev/rst8     ;1/4 inch cartridge tape
   or
   tar -xvf /dev/rmt0     ;1/2 inch reel tape
   ```

   Copy driver and include files (ip.c and ipreg.h) to /usr/sys/sundev

   ```
   cp /usr/sys/iphase/ip/drvr/* /usr/sys/sundev/
   ```

2. Copy the current system configuration file to a new file and add an entry for the Interphase controller.

   ```
   cd /usr/sys/conf          ;SunOS 3.x
   cd /usr/sys/sun3/conf     ;Sun3 OS4
   cd /usr/sys/sun4/conf     ;Sun4 OS4
   cp GENERIC BIP
   ```

a. Edit this new system configuration file and add the following entry for the first Interphase controller ipc0, at the end of the new configuration file.

```
controller  ipc0 at vme16d16 ? csr 0xee00 priority 2 vector
ipnint 200 ipeint 201 ipscint 202
disk      ip0 at ipc0 drive 0 flags 0
disk      ip1 at ipc0 drive 1 flags 0
disk      ip2 at ipc0 drive 2 flags 0
disk      ip3 at ipc0 drive 3 flags 0
```

NOTE: The first two lines in the entry above are entered as one. The 4400 supports from 1 to 4 drives.

b. Change the "ident" line to something other than generic.

```
from  ident    GENERIC
to    ident    "BOOT_IP"
```

c. Change the "config" line as follows:

```
from  config  vmunix      swap generic
to    config  vmunix      root ip0
```

d. The following lines must be commented out (if they exist) or their memory address must be changed so they do not fall between 0xee00 and 0xefff

```
#controller xyc0 at vme16d16 1 csr 0xee40 priority 2 vector
xyintr 0x48
#controller xyc1 at vme16d16 1 csr 0xee48 priority 2 vector
xyintr 0x49
#disk xy0 at xyc0 drive 0
#disk xy1 at xyc0 drive 1
#disk xy2 at xyc1 drive 0
#disk xy3 at xyc1 drive 1

#controller xdc0 at vme16d32 ? csr 0xee80 priority 2 vector
xdintr 0x44
#controller xdc1 at vme16d32 ? csr 0xee90 priority 2 vector
xdintr 0x45
#controller xdc2 at vme16d32 ? csr 0xeea0 priority 2 vector
xdintr 0x46
#controller xdc3 at vme16d32 ? csr 0xeeb0 priority 2
vector xdintr 0x47
```

```
#disk xd0 at xdc0 drive 0
#disk xd1 at xdc0 drive 1
#disk xd2 at xdc0 drive 2
#disk xd3 at xdc0 drive 3
#disk xd4 at xdc1 drive 0
#disk xd5 at xdc1 drive 1
#disk xd6 at xdc1 drive 2
#disk xd7 at xdc1 drive 3
#disk xd8 at xdc2 drive 0
#disk xd9 at xdc2 drive 1
#disk xd10 at xdc2 drive 2
#disk xd11 at xdc2 drive 3
#disk xd12 at xdc3 drive 0
#disk xd13 at xdc3 drive 1
#disk xd14 at xdc3 drive 2
#disk xd15 at xdc3 drive 3

#controller xtc0 at vme16d16 ? csr 0xee60 priority 3 vector
xtintr 0x64
#controller xtc1 at vme16d16 ? csr 0xee68 priority 3 vector
xtintr 0x65
#tape xt0 at xtc0 drive 0 flags 1
#tape xt1 at xtc1 drive 0 flags 1
```

e. If using a Sun3 with SunOS4.0, add a line to allow the driver to use the correct "include" files.

```
cpu   "OS4"          # Support for SunOS4 in Sun3
```

3. Add the proper reference for the Interphase controller to the "**files**" (**files.sun3** for Sun3 w/OS3.x) file. This file is located in the same directory as the configuration file.

   Add or modify the following line:

   sundev/ip.c optional ip device-driver

   NOTE: May have "ip not supported".

4. Edit the file /usr/sys/sun/conf.c to include the driver entry points of the Interphase device driver.

   a. Add or update the following in the conf.c file :

```
#include "ip.h"
#if NIP > 0
```

```
extern int ipopen(), ipstrategy(), ipread(), ipwrite();
extern int ipdump(), ipioctl(), ipsize(), ipclose();
#else
#define ipopen      nodev
#define ipclose     nodev
#define ipstrategy  nodev
#define ipread      nodev
#define ipwrite     nodev
#define ipdump      nodev
#define ipioctl     nodev
#define ipsize      0
#endif
```

b. Add to the end of bdevsw structure in the conf.c file:

```
{ ipopen,     ipclose,     ipstrategy,  ipdump, /*blk#*/
ipsize,       0 },
```

NOTE: "blk#" above represents the block device major number.

EXAMPLE:

Change from:

```
.
.
};
int       nblkdev = sizeof (bdevsw) / sizeof (bdevsw[0]);
.
.
```

To:

```
.
.
{ ipopen,     ipclose,     ipstrategy,  ipdump,/*22*/
  ipsize,     0 },
};
int       nblkdev = sizeof (bdevsw) / sizeof (bdevsw[0]);
.
.
```

NOTE: "blk#" above should be replaced with the block device major number determined by the sequential position of the entry within the structure (in the example above, 22 is used - yours may not be the same). This number will be used later in the installation.

For **SunOS3.x**, there is an entry in position 0 of the bdevsw table that must be changed as follows:

Change from:

```
{ ipopen,       nulldev,     ipstrategy,  ipdump,/*0*/
  ipsize,       0 },
```

To:

```
{ nodev,        nodev,       nodev,       nodev,/*0*/
  nodev,        0 },
```

c. Change the "ip" entry at position 4 in the cdev structure of the conf.c file.

SunOS4.0 EXAMPLE:

Change from:

```
{
  nodev,        nodev,       nodev,       nodev,/*4*/
  nodev,        nodev,       nodev,       0,/*was ip*/
  0,
},
```

To:

```
{
  ipopen,       ipclose,     ipread,      ipwrite,/*4*/
  ipioctl,      nulldev,     seltrue,     0,
  0,
},
```

SunOS3.x EXAMPLE:

Change from:

```
{
 ipopen,      nulldev,     ipread,      ipwrite,/*4*/
 ipioctl,     nodev,       nulldev,     0,
 seltrue,     0,           0,
},
```

To:

```
{
 ipopen,      ipclose,     ipread,      ipwrite,/*4*/
 ipioctl,     nodev,       nulldev,     0,
 seltrue,     0,           0,
},
```

NOTE: The "/*4*/" above represents the character device major number.

d. If using a Sun4 with SunOS3.2 rev2, change the following:

#if NIPC > 0

to

#if NPC > 0

5. Edit the swapgeneric.c file (same directory as conf.c) to include the following:

...

```
#include "ip.h"              ; add this entry above xy entry
#if NIP > 0
extern  struct mb_driver ipcdriver;
#endif

#include "xy.h"              ; this will already exist
#if NXY > 0
extern  struct mb_driver xycdriver;
#endif
```

...

```
#if NIP > 0                    ; add this entry above xy entry
   {"ip", &ipcdriver, makedev(blk#, 0)},     ; see note
#endif

#if NXY > 0                    ; this will already exist
   {"xy", &xycdriver, makedev(3, 0)},
#endif
...

#if NIP > 0                    ; add this entry above xy entry
   {"ip", &ipcdriver, makedev(4, 0), makedev(blk#, 0)},
#endif                                ; see note below

#if NXY > 0                    ; this will already exist
   {"xy", &xycdriver, makedev(9, 0), makedev(3, 0)},
#endif

...
```

NOTE: makedev(blk#, 0) - blk# must be replaced by the Interphase block device major number assigned in step 4.

6. Edit the file "**devices**" or "**devices.sun3**" (in the same directory as the configuration file) to include the correct block device major number for "ip".

   Add or change the following line:

   ip **blk#**

   NOTE: "blk#" in the line above must be replaced with the block device major number assigned earlier in step 4.

7. Edit the /dev/MAKEDEV file, changing or adding the Interphase designation, "ip" in the appropriate places.

   EXAMPLE:
   Change from:

```
xd*|xy*|sd*)
  unit=`expr $i : '..\(.*\)'`
  case $i in
  xd*) name=xd; blk=10; chr=42;;
```

To:

```
ip*|xd*|xy*|sd*)
  unit='expr $i : '..\(.*\)''
  case $i in
  ip*) name=ip; blk=blk#; chr=4;;
  xd*) name=xd; blk=10; chr=42;;
```

NOTE: The "blk#" in blk=blk# should be replaced with the block device major number.

8. Create the appropriate nodes in /dev for communicating with the driver.

   EXAMPLE:

   MAKEDEV ip0 ip1 ip2 ip3

   MAKEDEV must be executed for each drive being used.

9. Use the config command to place the devices in the configuration process, and generate the new kernel.

   EXAMPLE:

   cd /usr/sys/conf ;SunOS3.x
   cd /usr/sys/sun3/conf ;Sun3 OS4.0
   cd /usr/sys/sun4/conf ;Sun4 OS4.0
   config BIP

   After the config is completed:

   cd ../BIP
   make

   Any errors in the installation will normally show up at this time. If no errors occur, save the current working kernel and copy the new kernel to the root partition.

   cp /vmunix /vmunix.bak
   cp vmunix /vmunix.BIP

UPDATE FSTAB FILE

The fstab file on the root partition of the Interphase boot device must be updated.

```
cd /etc          ; change directory
vi fstab         ; edit fstab file
```

Change all of the partitions in the following manner:

```
from    /dev/xy0a / 4.2 rw 1 1
to      /dev/ip0a / 4.2 rw 1 1
```

NOTE: Change xy to ip for all partitions.

The Interphase disk is now set up and ready to boot.

MODIFY THE SUN EEPROM

The EEPROM must be set up to think it is booting off of the Xy disk controller. Halt the system and do the following:

```
>q 18                          ; open EEPROM
>EEPROM 018: 00 ? 12
>EEPROM 019: 00 ? 78           ; set for xy boot device
>EEPROM 01A: 00 ? 79
>EEPROM 01B: 00 ? 00
>EEPROM 01C: 00 ? 00
>EEPROM 01D: 00 ? 00
>EEPROM 01E: 00 ?              ; press space bar and return
>
```

Power system down.

NOTE: The system will not work in boot device polling mode (EEPROM address 018 = 00). This is because the xd device is polled first and the Interphase controller sends back a dtack. The system believes that the xd boot device exists and will attempt to communicate with it instead of the xy device.

BOOTING WITH THE NEW BOOT DISK

Test new boot kernel

Power system up

```
L1 A                          ; abort autoboot
b xy()vmunix.BIP -s           ; test Interphase boot kernel
```

If the system boots with no problems, rename new kernel to vmunix.

```
mv vmunix.BIP vmunix
```

Tune file system for better system performance. Execute tunefs on each partition of the "ip" disk that will be mounted.

```
tunefs -d 0 -a 40 /dev/ip0x
```

```
sync;halt                     ; bring system down
```

Cycle power to test the auto boot process.

If the message "waiting for disk to spin up..." appears you must boot manually.

LET US KNOW WHAT YOU THINK!

At Interphase, we are constantly working to improve our products to better meet the needs of our customers. One area we are always concerned with is user documentation. Make a copy of this form, fill it out and mail it to us, but save the original. That way, we can get a better understanding of what would be most helpful to you, and in the future if you have any further comments, you will still have a handy way of communicating them to us.

	Excellent	Good	Average	Poor
Contents	☐	☐	☐	☐
Readability	☐	☐	☐	☐
Organization	☐	☐	☐	☐
Appearance	☐	☐	☐	☐
Accuracy	☐	☐	☐	☐
Usefulness as a Reference	☐	☐	☐	☐

Have you used User Manuals provided with other manufacturer's disk/tape controllers? yes_____ no_____

If yes, how does this one compare?_____

_____

Although every effort is made to insure the integrity of the information provided to our customers, sometimes oversights do occur. Inform us of any errors you may have noticed in the text and give the page number on which it occurs if possible.

_____

_____

Comments:_____

_____

_____

_____

_____

_____

Name:_____

Title:_____

Company:_____

Mailing Address:_____

City:_____ State:_____

Telephone:_(____)_____ Zip:_____

We at Interphase want to hear from you.

Interphase Corporation
Technical Writing Dept.
2925 Merrell Road
Dallas, Tx 75229-9990

2925 Merrell Road • Dallas, Texas 75229
(214) 350-9000 • Telex: 9109976245