BoardSite™

In-Circuit Programmer

User Manual

981-0055-005

DATA I/O

July 1997 981-0055-005

Data I/O Corporation
10525 Willows Road N.E., P.O. Box 97046
Redmond, Washington 98073-9746 USA
(425) 881-6444

Acknowledgments:

Data I/O is a registered trademark of Data I/O Corporation.
BoardSite is a trademark of Data I/O Corporation.

Table of Contents

5. Hardware Design Guide

6. Advanced Operation

7. Sequence Editor Reference

B. Translation Formats

Index

Figures

Tables

Safety Summary

General safety information for operating personnel is contained in this summary. In addition, specific **WARNINGS** and **CAUTIONS** appear throughout this manual where they apply and are not included in this summary.

Antistatic Wrist Strap

To avoid electric shock, the antistatic wrist strap must contain a 1 MΩ (minimum) to 10 MΩ (maximum) isolating resistor.

Definitions

WARNING statements identify conditions or practices that could result in personal injury or loss of life. **CAUTION** statements identify conditions or practices that could result in damage to equipment or other property.

Fuse Replacement

For continued protection against the possibility of fire, replace the fuse only with a fuse of the specified voltage, current and type ratings.

Grounding the Product

The product is grounded through the grounding conductor of the power cord. To avoid electric shock, plug the power cord into a properly wired and grounded receptacle only. Grounding this equipment is essential for its safe operation.

Power Cord

Use only the power cord specified for your equipment.

Power Source

To avoid damage, operate the equipment only within specified line (ac) voltage.

Servicing

To reduce the risk of electric shock, perform only the servicing described in this manual.

Symbols

⚠ This symbol indicates that the user should consult the manual for further detail.

V∿ This symbol stands for V ac, for example, 120 V∿ = 120 V ac.

This symbol denotes a fuse rating for a user-replaceable fuse.

This symbol denotes the protective ground connection.

⏚ This symbol denotes a ground connection for a signal or for an antistatic wrist strap with impedance of 1 MΩ (minimum) to 10 MΩ (maximum).

Preface

The Preface describes how to contact Data I/O for technical assistance, repair and warranty services, and Keep Current™ subscription service using various methods, including the World Wide Web and Data I/O's Bulletin Board Service.

Data I/O Offices

United States

For technical assistance, contact

Data I/O Customer Resource Center
Telephone: 800-247-5700
Fax: 425-869-2821

For repair or warranty service, contact

Data I/O Central Dispatch
Telephone: 800-735-6070
Fax: 425-881-0561

For Keep Current subscription service, contact

Data I/O Sales
Telephone: 800-332-8246
Fax: 425-869-7423

Canada

For technical assistance, contact:

Data I/O Customer Resource Center
Telephone: 800-247-5700
Fax: 425-869-2821

For repair, warranty service, or Keep Current subscription service, contact

Data I/O Canada
6725 Airport Road, Suite 302
Mississauga, Ontario, L4V 1V2
Telephone: 905-678-0761
Fax: 905-678-7306

Japan

For technical assistance, repair, warranty service, or Keep Current subscription service, contact

Data I/O Japan
Osaki CN Building 2F
5-10-10 Osaki
Shinagawa-ku
Tokyo 141
Telephone: 3-3779-2151
Fax: 3-3779-2203

Germany

For technical assistance, repair, warranty service, or Keep Current subscription service, contact

Data I/O GmbH
Lochhamer Schlag 5
82166 Gräfelfing
Telephone: 089-858-580
Fax: 089-858-5810

Other Countries

For technical assistance, repair, warranty service, or Keep Current subscription service, contact your local Data I/O representative.

Contacting Data I/O

Technical Assistance

You can contact Data I/O for technical assistance by calling, sending a fax or electronic mail (e-mail), or using the Bulletin Board Service (BBS). To help us give you quick and accurate assistance, please provide the following information:

- Product version number
- Product serial number (if available)
- Detailed description of the problem you are experiencing
- Error messages (if any)
- Device manufacturer and part number (if device-related)

Telephone

Call the appropriate Data I/O Customer Support number listed at the front of the Preface. When you call, please be at your programmer or computer, have the product User Manual nearby, and be ready to provide the information listed above.

Fax

Fax the information listed above with your name, telephone number, and address to the appropriate Data I/O Customer Support fax number listed at the front of the Preface.

E-mail

To reach Data I/O via e-mail, send a message including your name, telephone number, e-mail address, and the information listed above to the following address:

techhelp@data-io.com

World Wide Web (www.data-io.com)

The Data I/O Home Page on the World Wide Web (WWW) has general company information, sales offices, and online technical and product information, including application notes and device lists. To access the WWW, you need an Internet account with Web access, and a Web browser, such as Netscape or Mosaic.

The address of Data I/O's Home Page is **http://www.data-io.com**.

Bulletin Board Service

The Data I/O Bulletin Board System (BBS) enables you to:

- Obtain a wide range of information on Data I/O products, including product descriptions, new revision information, technical support information, application notes, and other miscellaneous information.
- Access device support information
- Request support for a particular device
- Leave messages for the BBS system operator, Customer Support personnel, or other customers
- Download many DOS and Windows utilities

Online help files provide more information about the BBS. BBS numbers are as follows:

Germany	49-89-858-5833 49-89-858-5880
Japan	81-3-3779-2233
United States	425-882-3211

Warranty Information

Data I/O Corporation warrants this product against defects in materials and workmanship at the time of delivery and thereafter for a period of one (1) year. The foregoing warranty and the manufacturers' warranties, if any, are in lieu of all other warranties, expressed, implied or arising under law, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Data I/O maintains customer service offices throughout the world, each staffed with factory-trained technicians to provide prompt, quality service. For warranty service, contact Data I/O Customer Support at the numbers listed at the front of the Preface.

Repair Service

After the warranty period expires, repair services are available at Data I/O Service Centers worldwide. Single instance repairs and fixed price annual agreements that cover all parts and labor needed to correct normal malfunctions are also available. The annual agreements include semiannual performance certification.

For more information, or to order a Repair Service Agreement, contact Data I/O Customer Support at the numbers listed at the front of the Preface.

End User Registration and Address Change

If the end user for this product or your address has changed since the Registration Card was mailed, please notify Data I/O Customer Support at the numbers listed at the front of the Preface. This ensures that you receive information about product enhancements. Be sure to include the product serial number, if available.

Typographic Conventions

Throughout this manual different typographic conventions represent different cases of input and output.

Keys

Keys appear in boxes (for example, [Q]) or as bolded text. The **Enter** key (or on some keyboards, the **Return** key) is represented by this symbol: [↵].

Key Combinations

An instruction for pressing two keys at once, such as ^Z (Control and Z), is represented by two key boxes separated by a plus, such as [Ctrl] + [Z]. A key combination like [Esc][Ctrl] + [T] means press and release [Esc], then press [Ctrl] and [T] at the same time.

Variable Input

Variable input is italicized and should be replaced with the requested information. For example, "enter **copy** ***filename.hex***" means type **copy** just as you see it and replace ***filename.hex*** with the name of your file.

Displayed Text

Text displayed on an LCD or screen appears in a typewriter-like typeface.

```
You will see this text displayed on the screen.
```

1 Introduction

What Is BoardSite?

With BoardSite, you can program memory devices that are already installed on a circuit board. BoardSite programs and tests NMOS and CMOS EPROMs, EEPROMs, and single-chip microcomputers.

BoardSite 4000 programmers come in five standard models:

BoardSite Model	Capability
4100 (benchtop)	Program and test one isolated or 8 non-isolated circuit boards
4100 (portable)	Program and test one isolated or 8 non-isolated circuit boards
4400 (benchtop)	Program and test four isolated or 32 non-isolated circuit boards
4420X (benchtop)	Program and test four isolated or 32 non-isolated circuit boards. Accepts an expansion power supply to quadruple power output.

Another BoardSite model is the **BoardSite 5100**, which is a self-contained circuit programming station. This manual is used for both 4XXX and 5100 versions. 5100-specific information is contained in a separate manual.

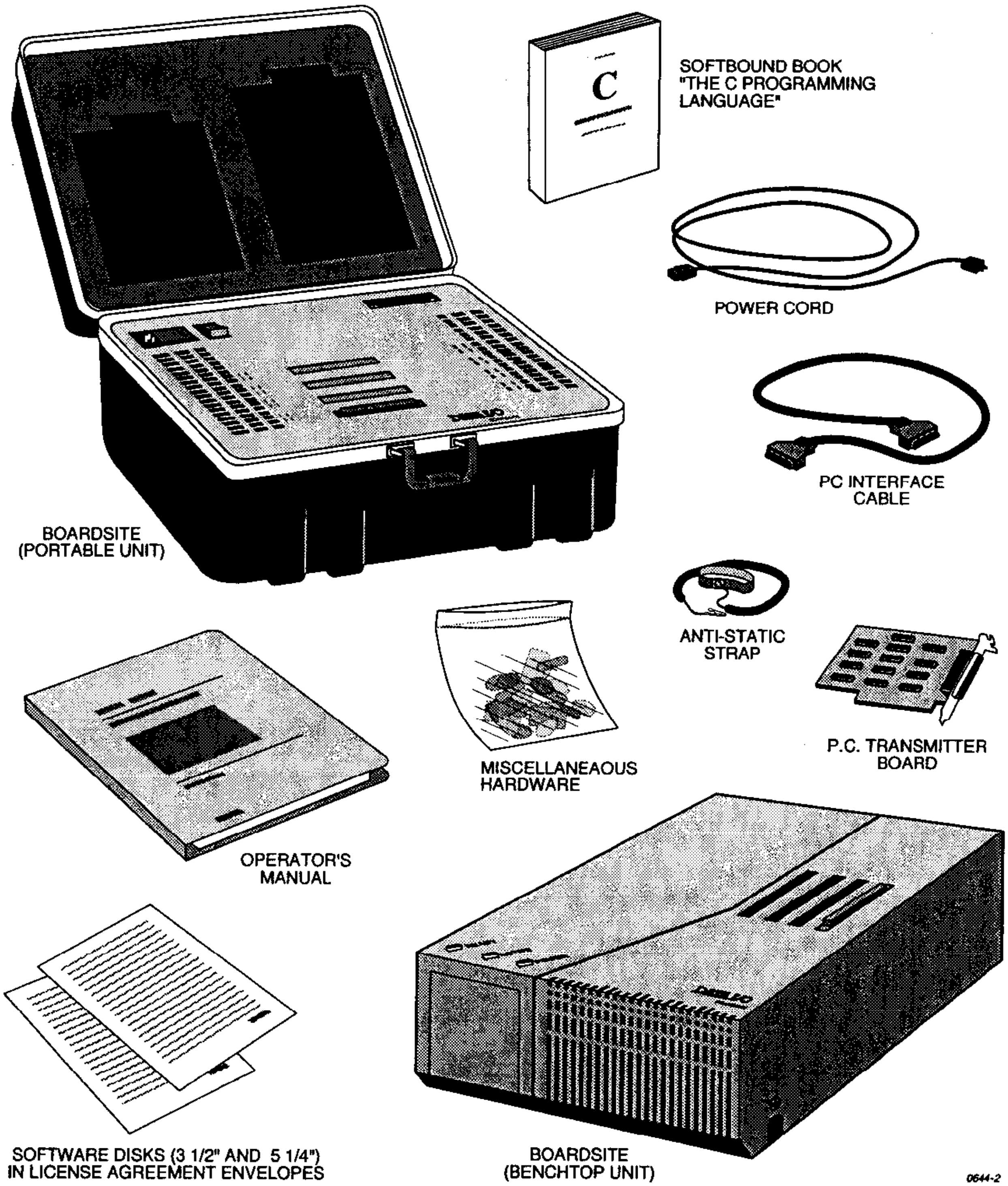
BoardSite also includes a complete software development system, which you use to customize BoardSite for your circuit board designs.

Contents of Package

Figure 1-1 illustrates the contents of your BoardSite package. You should check the contents of your package against Figure 1-1.

IMPORTANT: Before you open the envelope containing the BoardSite software, make sure that you have read and understand the licensing agreement that is printed on the software envelope. Opening the package indicates that you have accepted the terms of the agreement.

Figure 1-1
Contents of BoardSite Package



BoardSite External Features

Figure 1-2 (for benchtop) and Figure 1-3 (for portable) show BoardSite's external features.

Figure 1-2
External Features of the Benchtop Unit

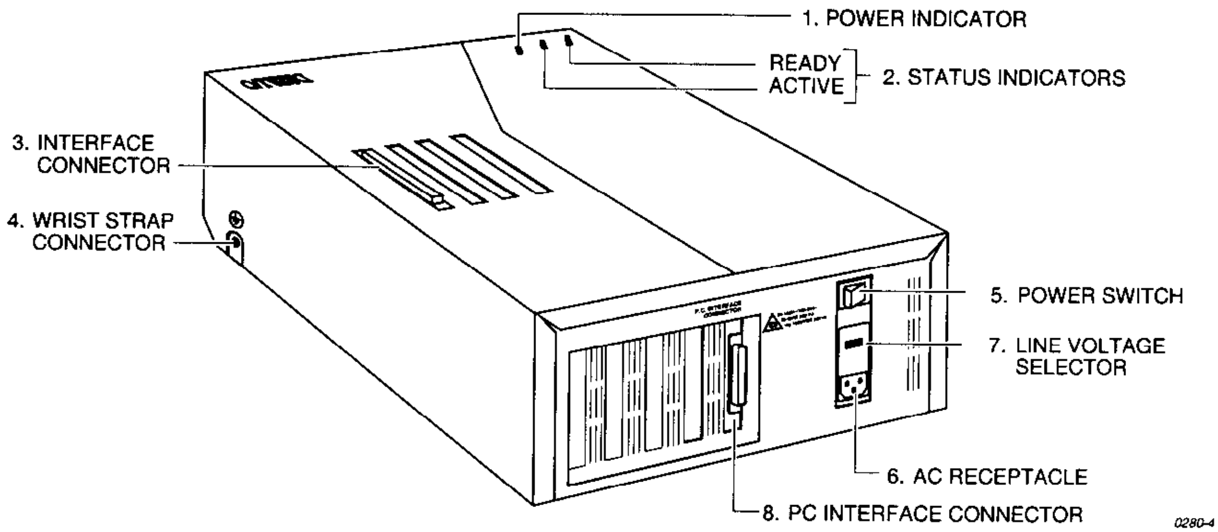
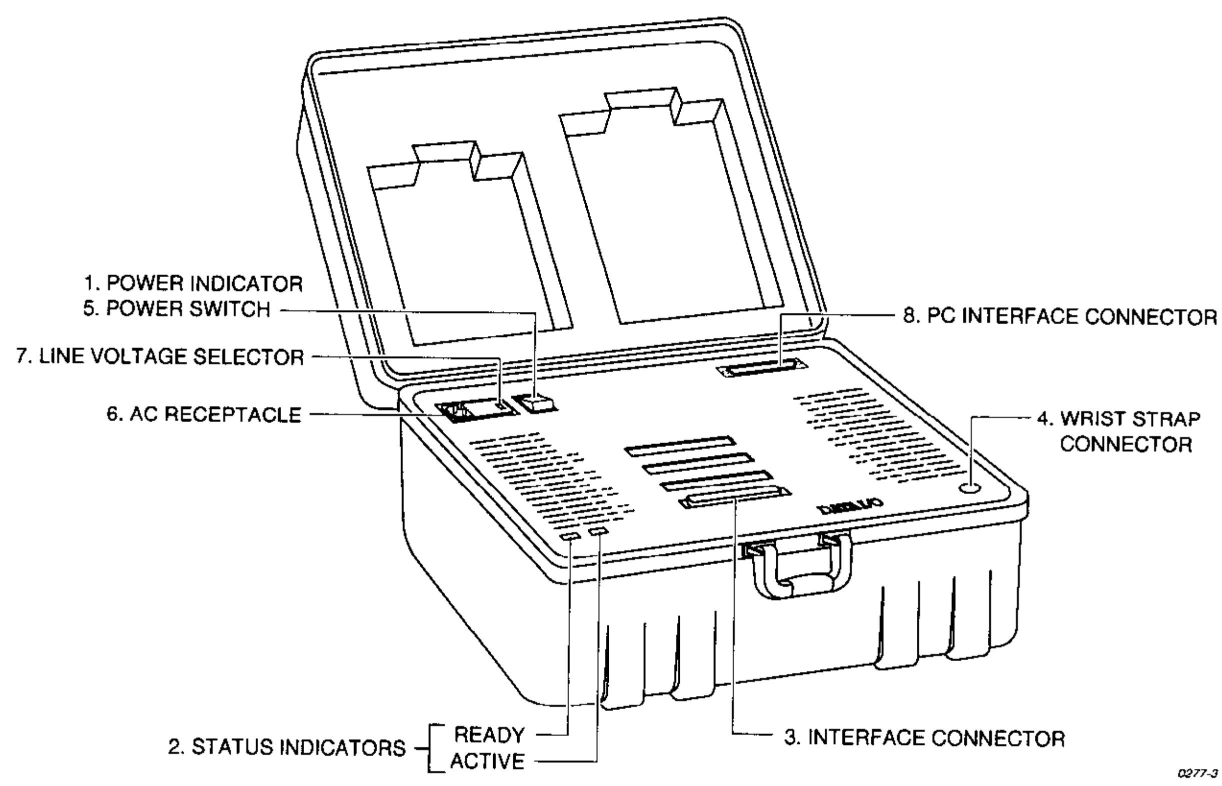


Figure 1-3
External Features of the Portable Unit



Each feature is described in the following list.

1. **Power indicator** - This lamp is lit when power is on.
2. **Status indicators** - These lamps provide information about BoardSite's operational status:

   Active indicator - When a device-related operation is in progress, the ACTIVE lamp is lit. When this lamp is lit, DO NOT remove the interface adapter or insert or extract a circuit board from the interface adapter.

   Ready indicator - When BoardSite has established communications with your PC, this lamp is lit.
3. **Interface adapter connector** - The interface adapter plugs in here.
4. **Wrist strap connector** - Connect the antistatic wrist strap here.
5. **Power switch** - Applies AC power to BoardSite.
6. **AC receptacle** - Connects BoardSite to AC power.
7. **Line voltage selector** - BoardSite operates on the nominal AC line voltage shown by this selector.
8. **PC interface connector** - Connects BoardSite to the transmitter board in your PC, via a digital interface cable.

Available Options

Although not part of the standard system, the following items are designed to complement BoardSite. For more information or to order an item listed below, contact your nearest Data I/O representative.

Diagnostic Test Adapter

The Diagnostic Test Adapter (model number BDS-4144DTA) is a passive test fixture to help you verify that BoardSite is operating within specification.

Spares Kit

To minimize downtime, you can order a Spares Kit, which contains one of each active-component subassembly (such as the controller board).

4000 Upgrade Kit

BoardSite service-installed upgrade kit converts any 41XX to a 4420X.

4000 Expansion Power Supply

The expansion power supply connects to the BoardSite 4420X to quadruple power outputs.

Prototype and Connector Kits

The prototype kit contains items you need to build a BoardSite interface adapter. The mating connectors are available separately.

Software Update Kit

This update kit brings your BoardSite up to the latest version.

Specifications

Compatible PC

Hewlett Packard Vectra Pentium PCs up to 166 MHz have been tested with BoardSite and are approved for use with BoardSite.

Minimum System Requirements

Following are the minimum system requirements for BoardSite:

- 640K RAM
- DOS (PC-DOS or MS-DOS) 2.11 or higher
- Monochrome or color monitor and display adapter card
- One 3-1/2" floppy disk drive and a hard drive
- Standard 84-key keyboard
- One half-width interface card slot available

Functional: Hardware Interface Signals

Power Supply Outputs

Vcc1: 0 to 7VDC* (for current, see below)
Vpp1: 0 to 25VDC* (for current, see below)
Vcc2: 0 to 7VDC* (for current, see below)
Vpp2: 0 to 25VDC* (for current, see below)
Vneg: 0 to -8V* (for current, see below)
+12VDC @.25A
-12VDC @.25A

** These supplies provide overvoltage, undervoltage, overcurrent detection, and remote sensing.*

Power Supply Current Capability

For the **4100** and **4400** (benchtop or portable), the maximum current per interface board and per system is:

Combined VCC1 and VCC2 current, 6A
Combined VPP1 and VPP2 current, 2A
VNEG current, .25A

For the **4420X**, the maximum current per interface board is:

Combined VCC1 and VCC2 current, 6A
Combined VPP1 and VPP2 current, 2A
VNEG current, .5A

Also for the **4420X**, the maximum current per system is:

Combined VCC1 and VCC2 current, 24A
Combined VPP1 and VPP2 current, 8A
VNEG current, 2A

Digital Interface

Signal	Description
A0-A15	16 low-order address lines
A16-A31	16 high-order address lines, or 16 individually programmable chip enable lines (PCE0-PCE15)
D0-D31	32 bidirectional data lines
BE0-BE7	8 board enable lines
BD0-BD7	8 board detect lines
ID0-ID7	8 adapter identification lines
PGM	1 program strobe line
XTAL0, XTAL1	2 clock lines with programmable frequency
ADAP0, ADAP1	2 adapter detect lines
C0-C23	24 digital control and status lines
LED0-LED7	8 status indicator control lines
GROUND	17 ground connections

Functional: Software

Commands and Functions

- Program/verify from disk file to circuit board (data files on disk)
- Program/verify from master board to circuit board (using a "golden master" circuit board)
- Test board (blank check and illegal bit check)
- Display error log, statistics, system configuration, data file, text file, batch file, memory board
- Diagnostics (self test/power supply calibration/system verification)
- Edit data file/text file/board profile/batch file
- File utilities (delete, copy, rename, directory, DOS, import, move, create)
- Simulate memory board operations
- Communications (upload, download, translate, parameters, transparent mode)
- Setup (set system defaults)
- Batch Command (automate BoardSite operations)
- Help (on-line help system)

Host Computer Data Translation Formats	Binary DEC Binary Hewlett-Packard 64000 Abs. Obj. HP-UX Intel 286 Intel 386 Intel Intellec 8/MDS Intel MCS-86 Hex Obj. Intel HEX-32 JEDEC Full Format JEDEC Kernel Mode Motorola 32-bit (S3) Motorola EXORciser Motorola EXORmax Tektronix Hex. Extended Tektronix Hexadecimal ASCII-Hex Space

Power Requirements

Operating Voltage	100VAC to 120VAC or 220VAC to 240VAC
Frequency Range	50Hz to 60Hz
Power Consumption	600VA maximum 350W maximum
Fuse Ratings	For either 115VAC (nominal) or 230VAC (nominal) operation, 6A/250V fast blow

Physical and Environmental

Dimensions	Benchtop:	16h x 40w x 56d cm 6.3h x 15.7w x 22.0d inches
	Portable:	25h x 52w x 40d cm 9.8h x 20.3w x 15.5d inches
Weight	Benchtop:	22.7 kg (50 lb)
	Portable:	13.6 kg (30 lb)
Temperature	Operating:	+5°C to +45°C (+40°F to +110°F)
	Storage:	-40°C to +70°C (-40°F to +158°F)
Relative Humidity	Operating:	20% to 80% RH non-condensing
	Storage:	10% to 90% RH non-condensing
Altitude	Operating:	To 5,000 meters (16,400 ft.)
	Storage:	To 8,500 meters (28,000 ft.)

Safety

BoardSite is designed to comply with the following safety standards:

Underwriters Laboratories—UL 1950

Canadian Standards Association—
CSA C22.2 No. 231

Technischer Überwachungsverein —TÜV
GS-Mark Certification EN60950

Class 1 LED product according to EN60825-1 A11 Oct. 1996.

Electrostatic Discharge (ESD)

BoardSite withstands up to an 8kV discharge to any point that is exposed to the operator's fingertip (exposed circuitry on interface adapter and customer circuit boards excluded). The unit is equipped with an antistatic wrist strap.

BoardSite is designed and tested to comply with IEC 801-2.

Certificate of RFI/EMI Conformance

Data I/O certifies that BoardSite (not including the BoardSite 5100) complies with the Radio Frequency Interference (RFI) and Electromagnetic Interference (EMI) requirements of EN55022 Class A and EN50082-1 as called out in 89/336/EEC, the EMC Directive for the European Community.

CE - *EC conformity mark*

CAUTION: ***This equipment is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.***

2 *Setting Up BoardSite*

This chapter describes how to set up your BoardSite system hardware, install the system software, and troubleshoot the system.

Installing the PC Transmitter Board

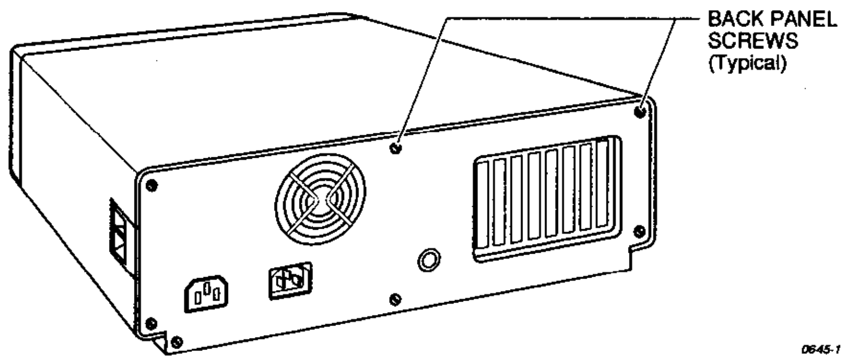
BoardSite uses a Data I/O-supplied transmitter board to communicate with your controller PC. This section describes how to install the transmitter board.

Note: *This procedure shows you how to install the transmitter board in a benchtop PC. If you are installing the transmitter board in a laptop PC, please refer to your laptop PC instruction manual to learn how to disassemble and reassemble the PC.*

WARNING: **Never perform the following procedure when power is applied to the PC. To avoid electrical shock or damage to the PC, disconnect the power cord before removing the top cover.**

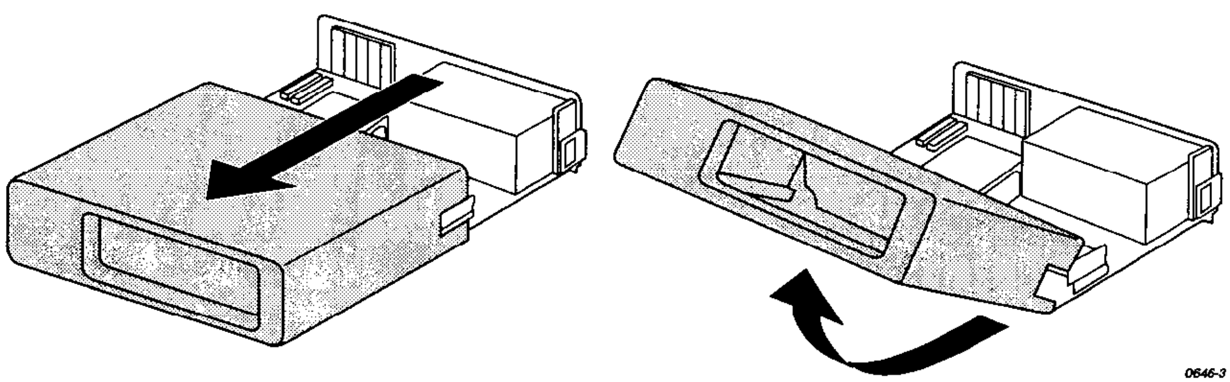
1. Turn off the PC power switch, and then unplug the PC power cord from both the PC and the wall outlet.
2. Remove the PC cover mounting screws, located in the corners and center of the PC back panel. See Figure 2-1. Save the screws.

Figure 2-1
Removing the Cover Screws



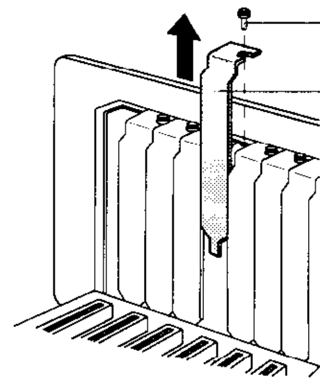
3. Slide the PC cover forward, toward the front of the PC. Lift the front of the cover up, remove the cover, and set it aside. See Figure 2-2.

Figure 2-2
Removing the Cover



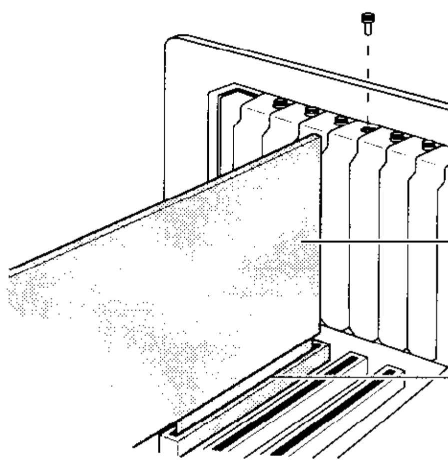
4. Select any vacant expansion slot on the PC bus, and then remove the screw that secures the expansion slot cover plate. Remove the cover plate. Save the retaining screw for installation of the PC transmitter board. See Figure 2-3.

Figure 2-3
Removing the Expansion Slot Cover Plate



5. Carefully insert the PC transmitter board into the selected expansion slot. See Figure 2-4.

Figure 2-4
Inserting the Board



6. Secure the PC transmitter board to the PC, using the screw you removed in step 4.
7. Replace the cover on the PC. Secure the cover with the mounting screws you removed in step 2.
8. Connect the power cord to the PC and to the wall outlet. Do not turn on the PC power switch yet.

Connecting BoardSite to the PC

You connect BoardSite to the PC transmitter board using a Data I/O-supplied interface cable.

1. Connect one end of the interface cable to the BoardSite PC Interface Connector. This connector is located on the rear panel of the benchtop unit, and the front panel of the portable unit. Secure the cable to the connector by tightening the thumb screws on the cable.
2. Connect the other end of the interface cable to the PC transmitter board that you installed in your PC. Secure the cable to the connector by tightening the thumb screws on the cable.

Note: *Always secure the cable to both BoardSite and your PC, using the thumb screws. These screws ensure a low-impedance ground connection between the two instruments. If you don't use the thumb screws, you may encounter erratic operation of the system.*

Installing the BoardSite Software

BoardSite is designed to work in a DOS PC environment. If your system has Windows installed, exit Windows 3.X or reboot in DOS mode in Windows 95.

If you choose to run in a DOS window in a Windows environment, refer to "Using a Windows 95 DOS Window" on page 2-7. Be aware, however, that operation in this mode is not guaranteed. Known problems include:

- Windows interrupts may affect programming times and cause errors.
- The DOS command cannot be used because information on the screen is not displayed correctly.

Using DOS

Always install the BoardSite software using the install utility included with the software. Simply copying the files from the original disks to your PC will not work.

IMPORTANT: Before you open the envelope containing the BoardSite software, make sure that you have read and understand the licensing agreement that is printed on the software envelope. Opening the package indicates that you have accepted the terms of the agreement.

Note: *See the Specifications section, at the front of this manual, for the minimum system requirements of your PC.*

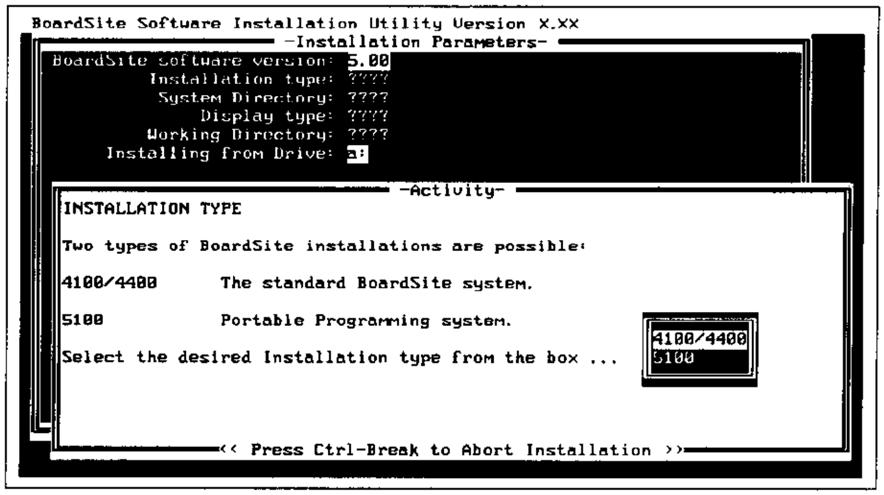
1. Turn on the power switch on the PC. Do not turn on the BoardSite power switch yet.

   If the PC doesn't boot successfully, or if it doesn't operate properly, see the following section, "Troubleshooting your System."
2. After the PC boots successfully, insert the BoardSite software disk labeled "Disk 1 of ..." into the floppy disk drive. If your PC has two drives, use the A: drive.

3. Type **A:** ↵ to make drive A: the current drive.
4. Type **install** ↵ to start the BoardSite install utility.
5. Follow the instructions that appear in the Activity pop-up. As you choose options, you'll see those options appear in the Installation Parameters pop-up.

   You can stop installation at any time by pressing Ctrl + Break .
6. When the utility asks for the Installation type, use the direction keys to select 4100/4400 or 5100 installation. If you pick 5100, both the 5100 and 4100/4400 software will be installed (see Figure 2-5).

Figure 2-5
Choosing the Installation Type



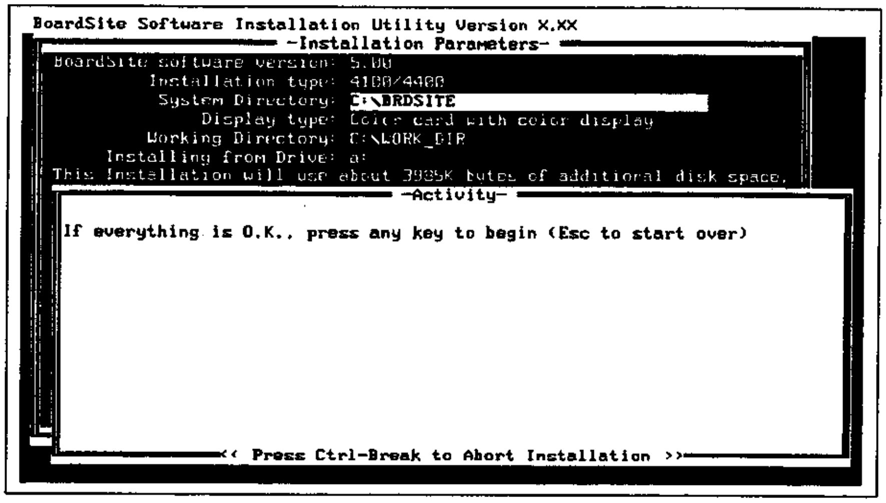
7. When the utility asks for the disk drive, type **c** to install the software onto your C: drive.
8. When the utility asks for the installation path, type **brdsite** to install the software into the \BRDSITE directory.

Note: *This procedure installs BoardSite software in a directory named \BRDSITE and installs your working files in a directory named \BRDSITE\WORK_DIR. You can use other directory names.*

9. When the utility asks for the disk drive, type **c** to store your working files on the C: drive. You typically store your working files on the same drive as the system software. If you want to store the files on another drive, type another drive letter.
10. When the utility asks for the path, type **brdsite\work_dir** to create a subdirectory called \WORK_DIR under the \BRDSITE directory. The path to your working files is then C:\BRDSITE\WORK_DIR.

11. At this point, you should see the screen in Figure 2-6. The cursor is moving from parameter to parameter.

Figure 2-6
Reviewing Installation Options



If anything is incorrect, press Esc to reset all options, then reselect the options. If everything is correct, press ↵ to start copying files.

12. After all files are copied, the install utility displays an information screen that describes changes that must be made to the autoexec.bat and config.sys files. Note these changes before proceeding.
13. The utility displays the READ.ME file. READ.ME is a standard ASCII file, so you can also use any standard text editor to print it.

Note: *Several application note files (with extension ".APP") are installed in the BoardSite program directory. You may want to print or list these files to read the technical information they contain.*

14. Use any standard text editor to make the required changes to your autoexec.bat and config.sys files.

    The **autoexec.bat** file must contain `C:\BRDSITE` in the path and the `SET BOARDSITE=C:\BRDSITE` statement.

    The **config.sys** file must contain `FILES=20` (20 files is the minimum number needed).

15. Press Ctrl + Alt + Del to reboot your PC.

This completes the BoardSite software installation.

Using a Windows 95 DOS Window

Windows 95 manages the system resources needed to run BoardSite. This section describes how to determine what resources are currently in use and how to reserve resources for BoardSite.

Note: Operation in this mode is not guaranteed. See page 2-4 for known problems.

Selecting BoardSite Resources

1. From the **Control Panel**, double click on the **System** icon.
2. Select **Device Manager** to view a list of all the system devices.
3. Select **Computer**, then click the **Properties** button.
4. Select the **View Resources** tab to view the hardware configurations.
5. BoardSite requires three resources: the Interrupt Request (IRQ), Input/Output (I/O) and Direct Memory Access (DMA). Select these resources as described below.

   Note: If resources needed by BoardSite are not available, note the hardware using them, follow the directions in the "Moving Resources" section to make these selections available if possible, then return to this section.

   IRQ Settings: Select the **Interrupt Request** radio button. If IRQ 2,3,4,5,6, or 7 is available (not displayed), select the **Reserve Resources** tab, click on the **Add** button, enter an available IRQ value, and click **OK**.

   I/O Address Allocation: Select the **Input/out** radio button. If addresses 220 to 2E0 are available (not displayed), select the **Reserve Resources** tab, click on the **Add** button, enter an available I/O address range (see Table 2-1 for BoardSite settings), and click OK.

   DMA Allocation: Select the **Direct Memory Access** radio button. If DMA 1,2 or 3 are available (not displayed), select the **Reserve Resources** tab, click on the **Add** button, enter an available DMA value, and click **OK**.

   Win95 will reserve these settings for BoardSite after they are saved and the PC is rebooted. Set up BoardSite to use these settings as described in the "Controller Board Setup" section on page 2-11.
6. Click on the **Start** button and select **Run**.
7. In the Run command line type **a:\install**.
8. Follow the BoardSite installation instructions in the "Using DOS" section on page 2-4.

Moving Resources

If a resource needed by BoardSite is being used by some other hardware, you may be able to move the existing resource to another location as described below.

1. From the **Control Panel**, double click **System**.
2. Select **the Device Manager** tab.
3. Select the device you want to move.
4. Select the **Properties** button.
5. Select the **Resource** tab.
6. Change the settings based on `Basic Configuration XXX` and check to see if the resource is now available.

Powering Up BoardSite

Check the Line Fuse

Before you power up BoardSite, check the fuse, which is located behind the door covering the voltage selector.

WARNING: If the line fuse blows, always replace it with a new fuse having the same size and rating. If you don't, you may create a fire hazard.

1. Using a slot-head screwdriver, gently pry open the door that covers the voltage selector wheel and fuse holders.

Note: The line cord module accepts two fuse sizes. The right-hand fuse holder accepts U.S.-size fuses (1/4" x 11/4") and the left-hand fuse holder accepts metric-size (5mm x 20mm) fuses. Only the right-hand fuse holder is active in BoardSite.

2. Pull the right-hand fuse holder out of its slot.
3. Check the fuse to determine if it's intact. If it is, proceed to step 4. If it is blown, install a new fuse.
4. Insert the fuse holder back into its slot so the arrow on the fuse holder points in the same direction as the arrows on the inside of the door.
5. Snap the door closed.

Power Up the System

After checking the line fuse (and replacing it if necessary), you are ready to power up BoardSite.

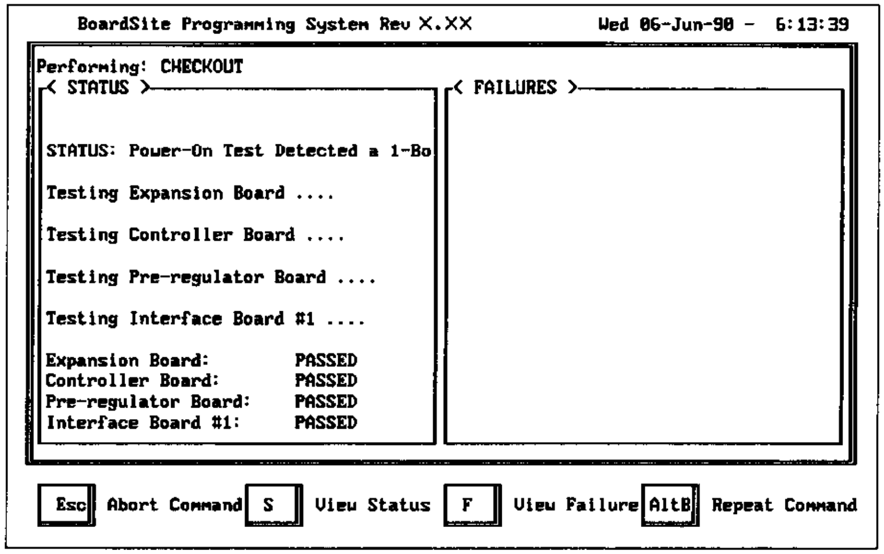
Note: Although you cannot copy, program, or verify boards without an interface adapter, you can always power up BoardSite without an interface adapter to perform a self-test.

1. Connect the power cord to the BoardSite AC receptacle, and then plug the other end into a properly grounded AC outlet.

WARNING: To ensure proper grounding for electrostatic discharge (ESD) protection and to avoid electrical shock hazard, connect BoardSite ONLY to a properly grounded AC outlet.

2. If you have not already done so, turn on your PC's power switch. Wait until the PC boots with the DOS prompt displayed.
3. Turn on the BoardSite power switch, located near the AC receptacle.
4. Type **brdsite** ↵ to start the BoardSite software.
5. After a few seconds, the software tells BoardSite to perform a full self-test. The ACTIVE light flashes as the test is performed. When the test is finished, you should see the screen in Figure 2-7.

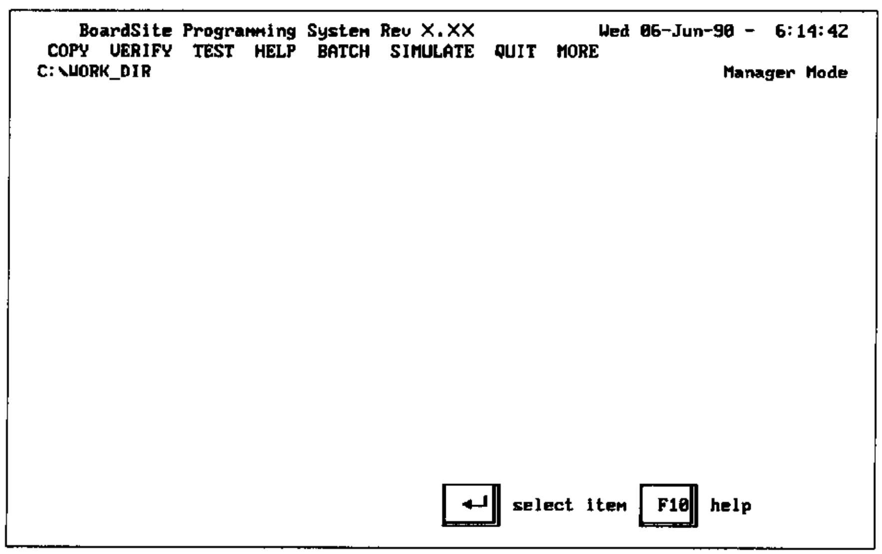
Figure 2-7
Screen After Successful Self-Test



If you don't see this screen, if there are error messages displayed, or if you have any other problems, see the following section, "Troubleshooting your System," for assistance.

6. If you see the screen in Figure 2-8, and the self-test gives all "PASSED" results, press [Esc] to display BoardSite top-level menus. You should see the following screen.

Figure 2-8
BoardSite Top-level Menus



If everything appears to be correct, go to Chapter 3, "Basic Operation," to learn how to test, program, and verify your circuit boards.

Troubleshooting Your System

The most common problem with BoardSite installation is a conflict between BoardSite and another device in your PC. For example, your PC could have a network card assigned to the same addresses as BoardSite, or it could have a second serial port assigned to the same interrupt as the transmitter board.

Because there are so many different ways you can configure a PC, these conflicts can cause many different problems. Identifying these problems is well beyond the scope of this manual. However, if your PC behaves differently after you install the BoardSite hardware and software (particularly when BoardSite power is on), then you should suspect an address or interrupt conflict. In this case, perform the controller board setup in the following section.

Disassembling BoardSite

To access the controller board, disassemble your BoardSite unit using the appropriate procedure below.

CAUTION: ***The controller board and expansion board components are static-sensitive. Perform all work on these boards at an antistatic workstation, and ground yourself by using a conductive wrist strap.***

Benchtop Unit Disassembly

Use the following procedure to disassemble your benchtop BoardSite unit.

1. Turn off the power to your PC and BoardSite.
2. Disconnect the AC power cord and the PC interface cable from BoardSite.
3. Carefully rotate BoardSite into a vertical position by lifting the right side. Remove two screws from the bottom panel, located at the left and right edges near the front of the unit. Lower the right side until BoardSite is horizontal again.
4. Remove all five screws from the rear panel.
5. Lift off the top cover and set it aside. To clear the LED mounting bracket, rotate the rear of the cover as it nears the top of the unit.
6. Carefully disconnect the cable from J7, in the upper-left corner of the controller board.
7. The expansion board and 62-pin connector are attached to the controller board. Remove the screw that holds the expansion board to the chassis and then extract the controller board assembly from the card edge connector.

Note: *You may need to use considerable force to remove the controller board. Always use the extractor lever mounted on the front of the controller board while you apply upward force to the opposite end of the board.*

8. After removing the controller board, proceed to the "Controller Board Setup" section. After you set up the controller board, perform the above steps in reverse order to reassemble BoardSite.

Portable Unit Disassembly

Use the following procedure to disassemble your portable BoardSite unit.

1. Turn off the power to your PC and BoardSite.
2. Disconnect the AC power cord and the PC interface cable from BoardSite.
3. Remove the 12 machine screws from BoardSite's front panel. Remove the panel and set it aside.

Note: *You may have to pry gently around the perimeter of the panel to free it from the base. Use a slot-head screwdriver to pry the panel.*

4. Carefully disconnect the cable from J7, in the upper-left corner of the controller board.
5. Carefully extract the controller board assembly from its card edge connector.

Note: *You may need to use considerable force to remove the controller board. Always use the extractor lever mounted on the front of the controller board while you apply upward force to the opposite end of the board.*

6. Lift the controller board assembly just far enough to disconnect the cable from the PC interface connector, which is the 62-pin "D" connector mounted on the expansion board. Then completely remove the board assembly.
7. After removing the controller board, proceed to the "Controller Board Setup" section. After setting up the controller board, perform the above steps in reverse order to reassemble BoardSite.

Controller Board Setup

This section describes the jumper options and switch settings for the controller board and expansion board.

Note: *Make changes to the controller board ONLY if your PC works incorrectly after installing BoardSite. The factory controller board settings will work with most popular PC configurations.*

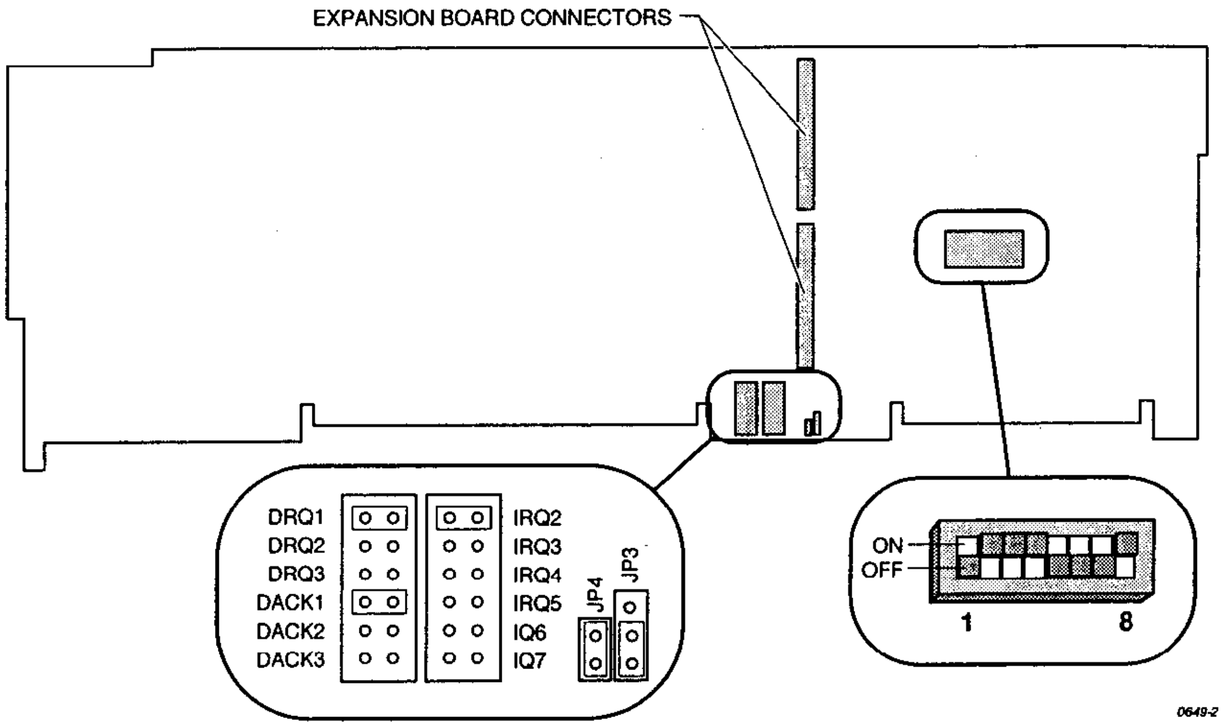
If you're not sure of your PC configuration, contact your System Administrator, or contact the person who set up your PC. If you're not sure what to do, call Data I/O Customer Support as listed in the Preface.

The controller board has two jumper blocks and a DIP switch (see Figure 2-9). Jumper blocks JP1 and JP2 are located at the bottom of the board, to the right of the edge connector. DIP switch SW1, hidden by the expansion board, must be removed to change the switch setting.

Jumper block JP1 allows you to reassign the Direct Memory Access Request channels (DRQ1 through DRQ3), and the Direct Memory Access Acknowledge channels (DACK1 through DACK3).

Jumper block JP2 allows you to reassign the Interrupt Request channels (IRQ2 through IRQ7).

Figure 2-9
Controller Board



1. Lay the controller board down on the antistatic work surface, component side up, with the card edge connectors toward you.
2. Perform this step ONLY if you must reassign the starting I/O address. Remove the expansion board from the controller board by removing three screws. Lift the expansion board to separate the connectors, and then set the expansion board aside.

   When replacing the expansion board, be sure to insert the pins on the back side of the expansion board into J3 and J4 on the controller board. See Figure 2-9. Be very careful to align all the pins before you apply pressure to the boards.
3. Bits 1 through 5 of switch SW1 set the starting I/O address. The factory default setting is address 220_{HEX}. Two address ranges are available for the PC: 220_{HEX}-$25F_{HEX}$ and 280-$2DF_{HEX}$. The switch settings are listed in Table 2-1.

Table 2-1
Switch SW1 Settings, Bits 1 - 5

Starting I/O Address	SW1 Bit Switch Settings Bit 1 (IOA9)	Bit 2 (IOA8)	Bit 3 (IOA7)	Bit 4 (IOA6)	Bit 5 (IOA5)
220_{HEX} (Factory default)	OFF	ON	ON	ON	OFF
240_{HEX}	OFF	ON	ON	OFF	ON
280_{HEX}	OFF	ON	OFF	ON	ON
$2A0_{HEX}$	OFF	ON	OFF	ON	OFF
$2C0_{HEX}$	OFF	ON	OFF	OFF	ON

4. Bits 6, 7, and 8 of switch SW1 set the length of an I/O cycle. This is done by inserting wait states. Table 2-2 shows SW1 settings for several popular PCs.

Note: *If your system locks up as soon as the powerup test begins, it is probably because of a hardware compatibility problem. You can modify your autoexec.bat file to try only one address instead of having the software scan all the I/O addresses looking for the BoardSite circuitry. Make sure you know which I/O address you have set the SW1 switch to on the BoardSite controller board. For example, if you set the SW1 switches for I/O address 220, add the following line to your autoexec.bat file: SET IOADDRE=220.*

Table 2-2
Switch SW1 Settings, Bits 6, 7, and 8

PC Manufacturer	SW1 Bit Switch Settings Bit 6	Bit 7	Bit 8
10MHz AT Compatible or 20MHz 386 (Factory default)	OFF	OFF	ON
IBM PC, XT	ON	OFF	OFF
IBM AT 6MHz	OFF	ON	ON
IBM AT 8MHz or Compatible	OFF	ON	OFF

5. See Figure 2-9 for details on setting jumper block JP2.

Note: *If you change the IRQ number by changing JP2, you must run the Setup command and change the Set BoardSite IRQ Number parameter to match. See the section, "Setup," in Chapter 6 for more information.*

6. Jumpers JP1, JP3, and JP4 are for factory tests and should NOT be changed.

3 *Basic Operation*

Introduction

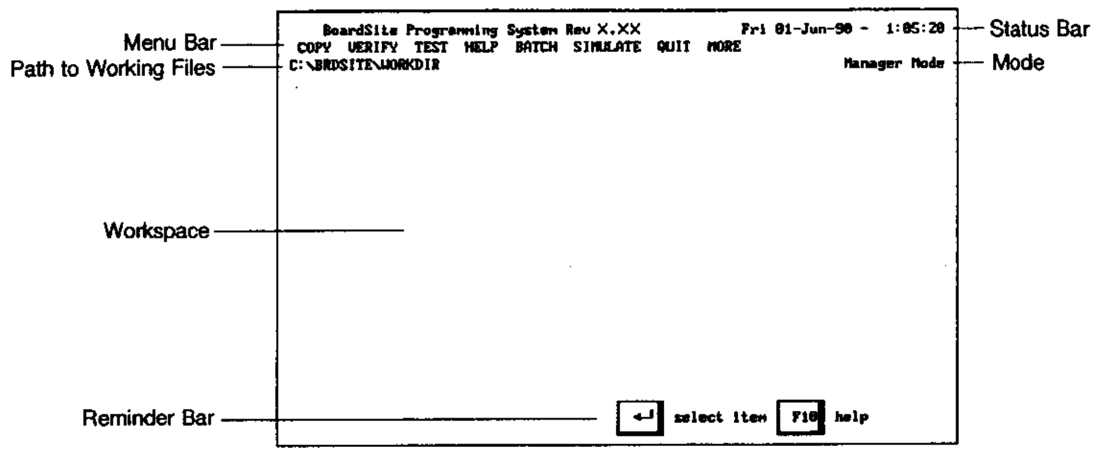
This chapter teaches you how to use the BoardSite system software. Included in this chapter are the following major topics:

- Using the BoardSite Screen
- Using Menus and Selecting Commands
- Getting Help
- Operator Mode, Technician Mode, and Manager Mode
- Operator Mode Command Reference

Using the BoardSite Screen

Before you continue, make sure you have set up BoardSite as described in Chapter 2. If you followed the instructions in Chapter 2, you should see the main BoardSite screen on your PC monitor, as shown in Figure 3-1. If you don't see this screen, turn back to Chapter 2, and follow the procedure in the section "Powering Up the System."

Figure 3-1
The BoardSite Screen



The BoardSite screen has the following components:

Status Bar The status bar is at the top of the screen. It shows you the BoardSite software version number, date and time.

Menu Bar The menu bar contains all the BoardSite menu commands. Later in this chapter, you'll learn how to select commands from the menus.

Working Files Path The path to your working files includes the drive letter and directory names.

Mode BoardSite has three operating modes: Operator Mode, Technician Mode, and Manager Mode. You'll learn more about using these modes later in this chapter.

Note: If BoardSite was set up as described in Chapter 2, it is in Manager Mode. Press Alt+O to put your system in Operator Mode.

Workspace Messages, pop-ups, help information, status and error windows, and other items appear in the workspace. The workspace should be empty because you haven't selected any commands yet.

Reminder Bar The reminder bar helps you remember what to do next. It shows you several keystrokes that take you to the next step in your current operation. For example, the reminder bar in Figure 3-1 tells you to press ↵ to select the command, or to press F10 to get help.

Using Menus and Selecting Commands

BoardSite commands appear in the menu bar. When you select a menu command (Copy, for example), you usually see another box with additional options. This box is called a pop-up. This section teaches you how to use menus and pop-ups.

Some commands (Edit, for example) actually load other programs that change the look of the BoardSite screen. In these programs (the Sequence Editor, for example), the menu bar contains drop-down menus with several commands for each menu name. To learn more about these advanced commands, see Chapter 6, "Advanced Operation," and Chapter 7, "Sequence Editor Reference."

Selecting Commands from the Menu Bar

There are two ways to select a command from the menu bar:

1. Press the [←] or [→] keys (on your PC keyboard) to highlight the command you want, and then press [↵].

Note *If your PC beeps when you press an arrow key, make sure NumLock is turned off.*

2. Or, type the first letter of the command you want (you don't have to press [↵]).

   If two commands have the same first letter (Checkout and Communications, for example), and you select the wrong command, just press [Esc]. Then, press the first letter again to select the other command.

Using the More Command

When you select the More command, BoardSite displays a new menu bar. For example, in Operator Mode, the menu bar changes as shown in Figure 3-2.

Figure 3-2
Operator Mode Menu Bars

```
   BoardSite Programming System Rev X.XX            Fri 01-Jun-90 -  1:05:43
 COPY  VERIFY  TEST  HELP  BATCH  QUIT  MORE
C:\BRDSITE\WORKDIR                                            Operator Mode
```

```
   BoardSite Programming System Rev X.XX            Fri 01-Jun-90 -  1:05:50
 CHECKOUT  DISPLAY  HELP  MORE
C:\BRDSITE\WORKDIR                                            Operator Mode
```

The second menu bar contains commands that are used less frequently, such as Checkout and Display. Notice that the More command also appears on the second menu bar. To move back to the first menu bar, just select the More command again.

Using Pop-ups

When you select a command from the menu bar, you usually see a pop-up that contains additional options. Most commands have a series of pop-ups that help you select a series of options. Figure 3-3 shows the first pop-up you see when you select the Copy command.

Figure 3-3
Board Profile Name Pop-up

```
C:\BRDSITE\WORKDIR
 Board Profile Name
   Press first letter then ↵

   demo_2816a
   007-9801-001
   007-9801-002
   test
```

With this pop-up, you select the board profile name for the Copy command. There are two ways you can select an option in this pop-up:

1. Press the ↑ or ↓ keys to highlight the option you want, and then press ↵ .
2. Or, type the first letter of the option you want, and then press enter. If two options have the same first letter, continue pressing the letter until you highlight the correct option, and then press ↵ .

Some pop-ups allow you to select more than one option before you press ↵ . Figure 3-4 shows an example of this kind of pop-up.

Figure 3-4
Device Names Pop-up with Several Options Selected

```
C:\BRDSITE\WORKDIR
 Board Profile Name
  Press first letter then ↵
   Device Names
     Press Ins to select, Del to de-select, then press ↵

     ALL devices on the board
    ►U1,U2                  ◄
    ►U3,U4                  ◄
     U5,U6
```

In this pop-up, you can select one option or several options. For example, you could select the two options U1,U2 and U3,U4. Or, you could select just U1,U2. Or, you could select the All Devices on the Board option.

To select several options in this pop-up:

1. Highlight the first option you want to select.
2. Press [Ins]. Arrowheads appear on the right and left sides of the options to tell you the option is selected.

   If you accidentally select an option you don't want, highlight the option and then press [Del] to deselect it.

3. Highlight the next option you want to select.
4. Press [Ins].
5. Repeat steps 3 and 4 until you have selected all options you want.
6. Press [↵].

Note: *In the Device Names pop-up, if you select the All Devices on the Board option, BoardSite automatically removes the arrowheads from any device group options you already selected.*

Getting Help

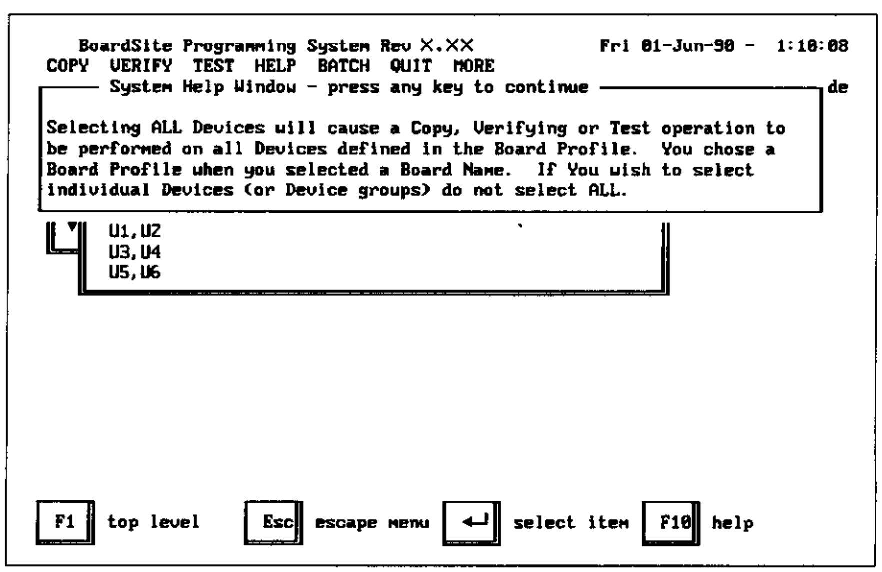
BoardSite has an online help system you can use at any time. Because BoardSite always knows where you are in an operation, it can display a help screen that is customized for your current situation. This is called context-sensitive help.

Context-Sensitive Help

You can always press [F10] to get context-sensitive help. BoardSite determines exactly what you need help on, and displays the appropriate help topic.

For example, assume that you select the All Devices on the Board option in the Device Names pop-up of Figure 3-4, and then press [F10]. BoardSite displays the help screen in Figure 3-5.

Figure 3-5
Context-Sensitive Help Screen



If you select the U1,U2 option and press [F10], you'll see the screen shown in Figure 3-6.

Figure 3-6
Context-Sensitive Help Screen

```
   BoardSite Programming System Rev X.XX          Fri 01-Jun-90 -  1:10:28
COPY  VERIFY  TEST  HELP  BATCH  QUIT  MORE
 ------ System Help Window - press any key to continue ----------------- de
|The Device names are those specified in the Board Profile, which was
|selected by picking a Board Name.
|
|Select ALL Devices by putting the cursor on 'ALL' and then pressing
|Enter.
|
|Individual Devices (or Device groups) may be SELECTED by moving the
|highlighted cursor to the desired name(s) by use of the UP and DOWN ARROW
|keys or by entering the FIRST LETTER of the Device Name, and then
|pressing the INSERT key to select it.  When selected the name will be
|bracketed by two marks which indicate this Device Group has been added to
|the list of Device Groups to perform the operation on.
|
|Press the DELETE key to DE-SELECT a previously selected Device group.
|
|You may continue to select and DE-select names until you press the ENTER
|key.  If you press the escape key the names selected will have no effect.
|
|
|--- More --- Press any key to continue ---
```

Notice how the help information changes to suit your current situation. Remember, you can always press [F10] to get context-sensitive help.

Help Command

You can also select the Help command from the menu bar to get general help on using the menus. If you select the Help command, you'll see the screen shown in Figure 3-7.

Figure 3-7
Help Screen

```
   BoardSite Programming System Rev X.XX          Fri 01-Jun-90 -  1:10:45
COPY  VERIFY  TEST  HELP  BATCH  QUIT  MORE
 ------ System Help Window - press any key to continue ----------------- de
|
|The Keys used by BoardSite and their functions are:
|
|F1 - return to top level menu.
|
|F10 - context sensitive help.
|
|Escape - aborts a menu without any change, and backs up to the previous
|level.
|
|Enter - selects the highlighted command or option in the current menu.
|
|Pressing the first letter of a command or option moves the highlight to
|it.
|
|Alt-M - to enter password to change to Manager Mode of operation.
|
|Alt-T - to enter password to change to Technician Mode of operation.
|
|
|--- More --- Press any key to continue ---
```

Operator Mode, Technician Mode, and Manager Mode

With the three BoardSite modes, you can customize the menus for different environments. For example, if you are using BoardSite to design and develop a new programmable board, you would put the system into Manager Mode. This mode gives you access to all BoardSite commands, and lets you create and change board profiles and other files.

If you install BoardSite on your manufacturing floor, you would probably put the system into Operator Mode. In this mode, the operator can copy (program), verify, and test boards, display certain system information, and run a batch file. However, the operator cannot change any board profiles, data files, or any other files, nor can the operator make any changes to BoardSite's setup. Operator mode protects your files and setup from being accidentally changed.

Technician mode is similar to Operator Mode, but it also allows access to the full Checkout and Display commands (as in Manager Mode), and allows access to the Simulate and Communications commands. For more information on these commands, see Chapter 6, "Advanced Operation."

To enter Technician Mode or Manager Mode, you need to know the appropriate password. The passwords are defined by choosing the Setup command. For more information, see Chapter 6, "Advanced Operation."

To Enter Operator Mode

If you are in any other mode, you can switch to Operator Mode by pressing [Alt] + [O]. Remember, you need to know the appropriate password to switch back to either Technician Mode or Manager Mode.

To Enter Technician Mode

From Manager Mode, press [Alt] + [T]. Remember, you need to know the Manager Mode password to switch back to Manager Mode.

From Operator Mode:

1. Press [Alt] + [T].
2. In the pop-up, type the Technician Mode password.
3. Press [↵].

To Enter Manager Mode

From any other mode:

1. Press [Alt] + [M].
2. In the pop-up, type the Manager Mode password.
3. Press [↵].

Note: The default password for both Technician Mode and Manager Mode is ***password****. See the section "Setup," in Chapter 6, "Advanced Operation," to learn how to change the passwords.*

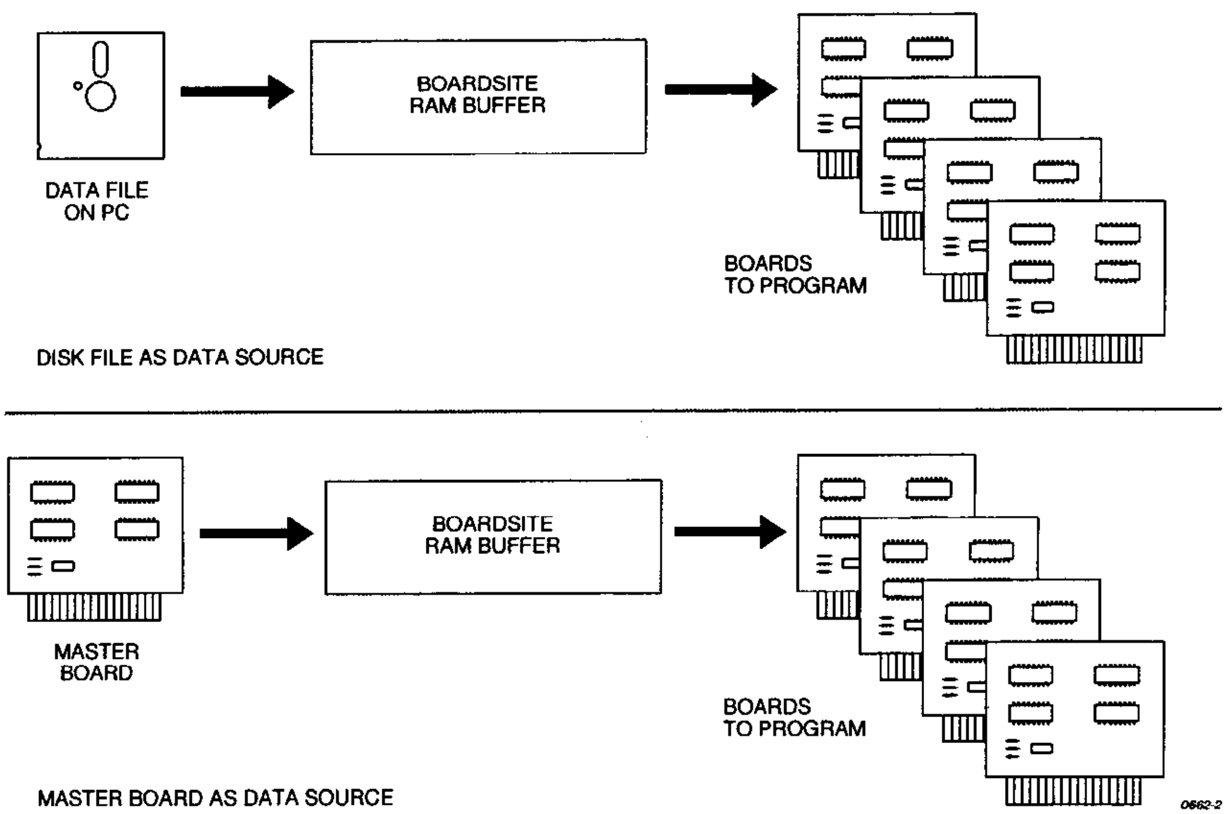
Operator Mode Command Reference

This section describes all the commands that are available in Operator Mode. For commands that are available only in Manager Mode, see Chapter 6, "Advanced Operation."

Copy

With the Copy command, you can either program boards with data from a disk file on your PC, or program boards with data from a master board. See Figure 3-8.

Figure 3-8
Copy Command



To use the Copy command:

1. Select the Copy command from the menu bar.
2. In the Board Profile Name pop-up, select the board profile you want to use. Press ↵.
3. In the Device Names pop-up, select the options corresponding to the devices you want to copy. You can either copy all devices by selecting the All Devices on the Board option, or you can copy just one (or several) device groups by selecting the appropriate device group options. Press ↵.

Note: If you don't see the Device Names pop-up, it means that the options have been automatically selected by the board profile. Skip this step, and go to the next step.

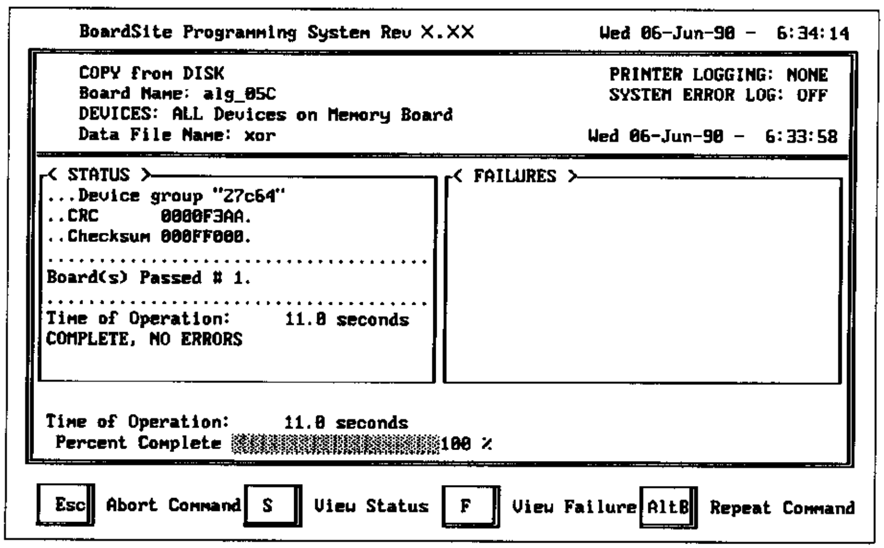
4. In the Source Options pop-up, select the data source, either a disk file or a master board. Press ↵.

Note: If you don't see the Source Options pop-up, it means that the options have been automatically selected by the board profile. Go directly to step 6.

5. If you selected the disk file option, select the data file name in the Data File Name pop-up, and then press ↵.
6. Make sure the interface adapter is correctly attached to BoardSite, and that the correct boards are inserted in the interface adapter.
7. Press Alt + B to begin the Copy command.

BoardSite copies data to your circuit boards and displays the results on the status screen shown in Figure 3-9.

Figure 3-9
Status Screen After Copy Command



This screen lists the information you entered in the procedure steps (board profile name, data file name, and so on), and some additional status information listed below.

Status—shows the current status of the operation in progress, or shows the status of the completed operation.

Failures—lists the board number, device names, and error information for any failures that occurred during the operation.

Percent Complete—uses an action symbol to graphically show the progress of the current operation.

Time of Operation—shows elapsed time of the completed operation.

You can zoom the Status window to full size by pressing [S], or you can zoom the Failures window to full size by pressing [F]. Zooming in makes the windows easier to read.

After the Copy command is finished, you may either press [Alt] + [B] to repeat the command or press [Esc] to return to the last pop-up.

Note: *To remove all pop-ups from the screen and return to the menu bar, press F1.*

Verify

BoardSite compares the data in the programmed circuit boards with the disk data file or master board. BoardSite performs the Verify command automatically after the Copy command, but you can also select the Verify command separately.

To use the Verify command:

1. Select the Verify command from the menu bar.
2. In the Board Profile Name pop-up, select the board profile you want to use. Press [↵].
3. In the Device Names pop-up, select the options corresponding to the devices you want to verify. You can either verify all devices by selecting the All Devices on the Board option, or you can verify just one (or several) device groups by selecting the appropriate device group options. Press [↵].

Note: *If you don't see the Device Names pop-up, it means that the options have been automatically selected by the board profile. Skip this step, and go to the next step.*

4. In the Source Options pop-up, select the data source, either a disk file or a master board. Press [↵].

Note: *If you don't see the Source Options pop-up, the options were automatically selected by the board profile. Go directly to step 6.*

5. If you selected the disk file option, select the data file name in the Data File Name pop-up, and then press [↵].
6. Make sure the interface adapter is correctly attached to BoardSite, and that the correct boards are inserted in the interface adapter.
7. Press [Alt] + [B] to begin the Verify command.

BoardSite verifies your circuit boards and displays the results on a status screen similar to the Copy command status screen shown in Figure 3-9.

After the Verify command is finished, you may either repeat the command by pressing [Alt] + [B], or return to the last pop-up by pressing [Esc].

Test

With the Test command, you can perform a blank check and/or illegal bit check on blank circuit boards. BoardSite performs the Test command automatically when you select the Copy command, but you can also select the Test command separately.

To use the Test command:

1. Select the Test command from the menu bar.
2. In the Board Profile Name pop-up, select the board profile you want to use. Press [↵].
3. In the Test Options pop-up, select Blank Check, Illegal Bit Test, or Both. Press [↵].
4. In the Device Names pop-up, select the options corresponding to the devices you want to test. You can either test all devices by selecting the All Devices on the Board option, or you can test just one (or several) device groups by selecting the appropriate device group options. Press [↵].

Note: *If you don't see the Device Names pop-up, it means that the options have been automatically selected by the board profile. Skip this step, and go to the next step.*

5. In the Source Options pop-up, select the data source, either a disk file or a master board. Press [↵].

Note: *If you don't see the Source Options pop-up, the options were automatically selected by the board profile. Go directly to step 7.*

6. If you selected the disk file option, select the data file name in the Data File Name pop-up, and then press [↵].
7. Make sure the interface adapter is correctly attached to BoardSite, and that the correct boards are inserted in the interface adapter.
8. Press [Alt] + [B] to begin the Test command.

BoardSite tests your circuit boards and displays the results on a status screen similar to the Copy command status screen shown in Figure 3-9.

After the Test command is finished, you may either repeat the command by pressing [Alt] + [B], or return to the last pop-up by pressing [Esc].

Help

The Help command gives you help on using the BoardSite main menus. For more information on the Help command, see the section, "Getting Help" in this chapter.

Batch

You can automate long command sequences using the BoardSite batch command. In Operator Mode or Technician Mode, you can only run an existing batch file, but in Manager Mode, you can create and modify batch files or record BoardSite operations in a batch file. For more information on creating, modifying, and running batch files, see Chapter 6, "Advanced Operation."

To use the Batch command:

1. Select the Batch command from the menu bar.
2. In the Execute Batch pop-up, select an option as follows:
   - To run a batch file from a host computer connected to your PC's COM1 port, select the COM1 option.
   - To run a batch file from a host computer connected to your PC's COM2 port, select the COM2 option.
   - To run a batch file that resides in your BoardSite working directory, select the batch file name (all batch files are listed in the pop-up).
3. Press ↵.
4. Press Alt + B to run the batch file.

Note: The COM1 or COM2 parameters must match the host computer's communications parameters. If the parameters don't match, you may either see an error message, or BoardSite may not respond to the host. For more information on running batch commands from a host computer, see Chapter 6, "Advanced Operation."

You can stop the batch job while it's running, and return to the menu bar, by pressing Esc.

After the batch job is finished, press any key to return to the menu bar.

Checkout

With the Checkout command, you can tell BoardSite to perform a brief self-test to determine if the hardware is working properly. These tests are described below.

Expansion—Tests the link between the PC (controller) and BoardSite.

Controller—Tests all hardware functions on the controller board, such as sequence port, timers, analog multiplexers and DACs, and status indicators.

Pre-regulator—Tests all hardware functions on the pre-regulator board.

Interface #*N*—Tests all hardware functions on interface board #*N*, where *N* is the number of each interface board detected in the system. Tests include read/write all digital ports, power sequence, and analog voltage controls.

Note: If you don't see the Checkout command on the menu bar, select the More command to display the command.

To use the Checkout command:

1. Select the Checkout command from the menu bar.
2. Press Alt + B to begin the Checkout command.

In Technician Mode, BoardSite runs the full Checkout command as in Manager Mode.

Diagnostic Test Adapter

When you install the optional Diagnostic Test Adapter, you can use the Checkout command (in Manager Mode or Technician Mode) to perform various tests of digital and analog signals, under resistive loading.

To run these tests with resistive loading provided by the Diagnostic Test Adapter, install the 20-pin header on J5 of the Diagnostic Test Adapter.

Display

With the Display command, you can view the system error log and the statistics information that BoardSite accumulates during board operations. You can also use this command to view the BoardSite configuration information. You can print the output of the Display command to create a permanent record.

To use the Display command:

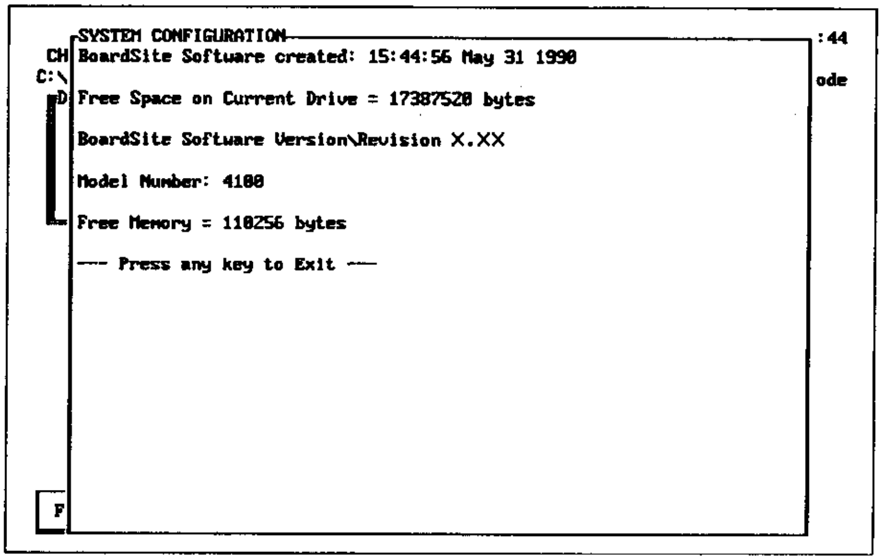
1. Select the Display command from the menu bar.
2. In the Display Command Options pop-up, select an option as follows:
   - To display the system error log, which contains all error messages that occurred during board operations, select the Error Log option.
   - To display the programming statistics associated with a particular board profile, select the Statistics option.
   - To display BoardSite system information, select the Configuration option.
3. Press ↵.
4. In the Destination Options pop-up, select an option to tell BoardSite where to send the information. Press ↵.

Note: If you select the printer option, BoardSite sends plain ASCII text to the Print Output Device as specified in Setup. The data contains no printer formatting characters or other special characters, other than Carriage Return, Line Feed, and Form Feed.

5. If you selected the Statistics option in the Display Command Options pop-up, select a board profile name in the Board Profile Name pop-up. Press ↵.

Figure 3-10 shows the Display command screen output if you select the Configuration option.

Figure 3-10
Display Command Output



In Technician Mode, BoardSite runs the full Display command as in Manager Mode. See Chapter 6, "Advanced Operation," for more information.

4 BoardSite Design Process

Introduction

This chapter shows you how to use Chapters 5, 6, and 7 to help you design your programmable circuit board, interface adapter, and BoardSite software. This chapter also lists the tasks in a typical BoardSite design project, and gives you a flow chart to help you plan your design project.

How to Use the Following Chapters

Here is the information you will find in the following three chapters:

Chapter 5, "Hardware Design Guide"

This chapter contains information to help you design your programmable circuit board and interface adapter. It gives you general design information on programmable devices, information on designing your programmable boards, general guidelines for designing the hardware interface to BoardSite, and specific information on the BoardSite interface signal lines.

If your boards are already designed, read this chapter to learn how to design the interface adapter.

If your boards are not yet designed, read this chapter to learn how to design boards that are in-circuit-programmable. Also, read this chapter to learn how to design the interface adapter.

Chapter 6, "Advanced Operation"

This chapter describes the advanced BoardSite software features available in Manager Mode. The commands described are Simulate, Communications, Display, Edit, Setup, File, and Batch.

Read this chapter to learn how to create and edit board profiles, download data files to BoardSite, create and edit batch files, and do many other design-related tasks.

Chapter 7, "Sequence Editor Reference"

This chapter describes the Sequence Editor. With the sequence editor, you can modify and compile the sequence file, which contains programming algorithms for all BoardSite operations.

Read this chapter to learn about algorithms, sequences, and primitives. Also, read this chapter to learn how to compile the default sequence file generated by the Board Profile Editor, or to edit and customize the sequence file.

BoardSite Design Task List and Flow Chart

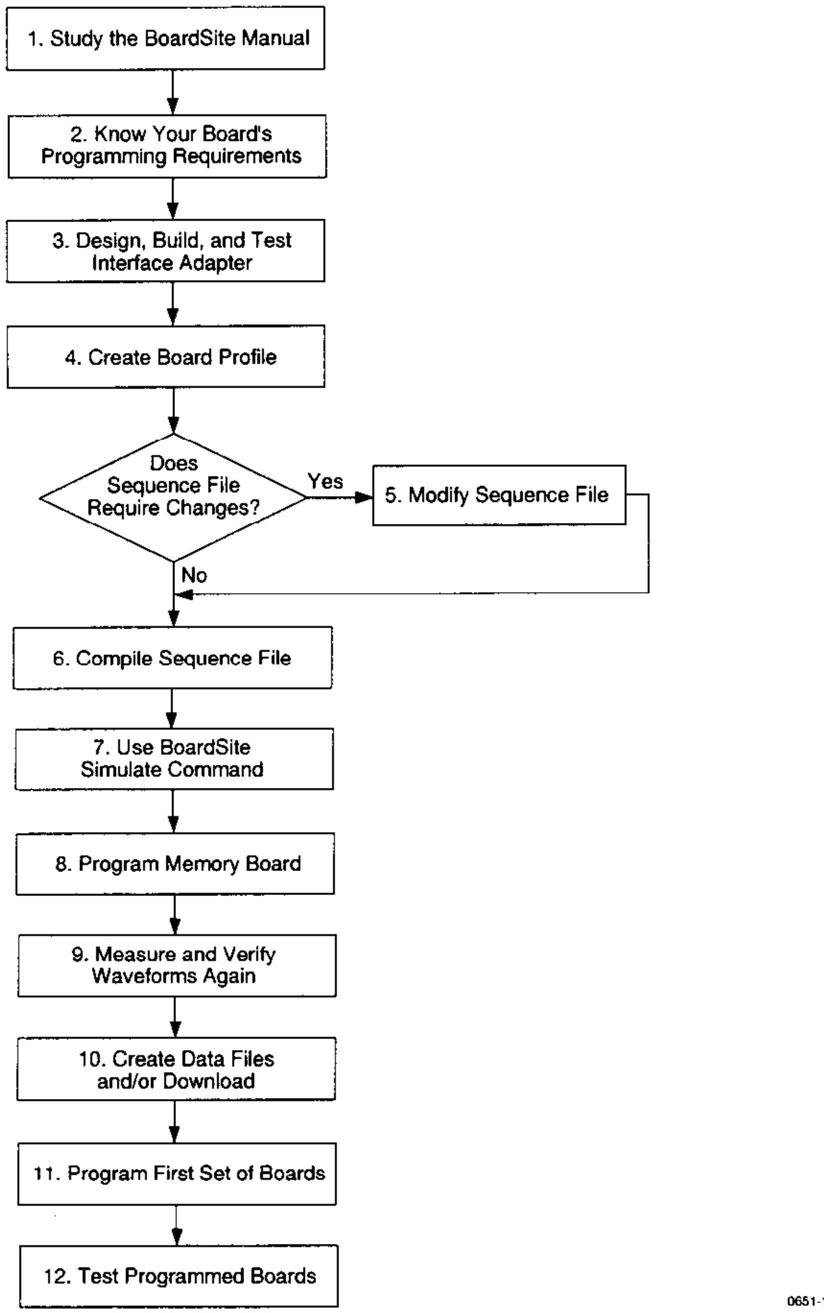
This section describes the design process you follow to successfully program your circuit boards. This section is divided into two topics:

- The first topic is a design task checklist and flow chart. It is an overview of the design tasks you need to accomplish. If you want, you can check off the tasks as you accomplish them.
- The second topic is a brief discussion of each of the tasks, with references to the appropriate sections of the manual.

Design Task Checklist and Flow Chart

1. Study the *BoardSite User Manual*, especially Chapters 5, 6, and 7.
2. Know your board's programming requirements. Verify that your board is in-circuit programmable, or design it to be.
3. Design, build, and test interface adapter.
4. Create board profile.
5. Modify sequence file (optional).
6. Compile sequence file.
7. Use BoardSite Simulate command, with an oscilloscope, to measure and verify waveforms and to debug interface adapter, board profile, and sequence file.
8. Program memory board, using BoardSite Copy command.
9. Measure and verify waveforms again, with board installed in the interface adapter. Use Simulate command if desired.
10. Create data files and/or download actual data files to BoardSite.
11. Program first set of boards.
12. Test programmed boards in target system.

Figure 4-1
Design Task Flow Chart



Design Task Cross-Reference

This section shows you how to use this manual to help you accomplish each design task listed in the previous section.

1. Study *BoardSite User Manual*

You should read (or at least scan) this manual before you begin your design. If you have installed your BoardSite software, you can practice some of the important design tasks in the manual. For example, you can use the sample design file **demo_2816a** to practice editing the board profile and compiling the sequence file.

2. Know your Board's Programming Requirements

If your boards are already designed, read the following sections in Chapter 5, "Hardware Design Guide," to learn more about how the devices on your boards interface to BoardSite:

- Design Rules for Programmable Devices
- Designing Circuit Boards to be In-Circuit-Programmable

If you haven't designed the boards yet, read and study these two sections before you start your design. You may be able to use the techniques in these sections to speed your design process and to take advantage of some of BoardSite's advanced features.

3. Design, Build, and Test Interface Adapter

Read the last three sections of Chapter 5 to learn how to design the interface adapter. These sections are:

- General Information on Hardware Interfacing
- Design Rules for BoardSite Interface Signals
- Designing the Interface Adapter

Remember that your board design affects the interface adapter design. Also remember that both board design and interface adapter design affect the board profile and sequence file.

4. Create Board Profile

Each board you program has a board profile that describes the board to the system. Read the following topics in Chapter 6, "Advanced Operation," to learn how to create the board profile:

- Using the Board Profile Editor
- Board Profile Reference

You may also want to refer to the following sections in Chapter 5 for information about how the board profile is affected by different interface designs:

- General Information on Hardware Interfacing
- Design Rules for BoardSite Interface Signals

5. Modify Sequence File (Optional)

The Board Profile Editor automatically creates a default sequence file for you. Usually, you don't have to modify the default sequence file unless your board has some specific requirements such as additional control signals.

If you do have to modify the sequence file, read (or at least scan) Chapter 7, "Sequence Editor Reference," to learn about the sequence file and Sequence Editor.

When you are ready to modify the file, read the following section in Chapter 7 for a detailed example:

- How to Modify a Sequence

6. Compile Sequence File

Even if you didn't modify the sequence file, you still must use the Sequence Editor to compile and link the file. If you didn't modify the sequence file, the procedure is simple. Read the following sections in Chapter 7:

- Compile the Sequence File
- Compile Menu

7. Use BoardSite Simulate Command

Before you try to program a board, you should always run the Simulate command to test your interface adapter and verify programming waveforms. Some of the tests you should make are:

- Are DC voltages correct?
- Are DC voltages at the correct pins on the board connector?
- Are the following operating: Adapter Detect, Board Detect, Adapter ID, and so on?
- Is power sequencing correct?
- Are programming signals (CS, OE, VPP, PGM, and so on) timed correctly?
- Are programming voltages correct (VCC, VPP, and so on)?
- Are programming voltages stable?

Read the following section in chapter 6 to learn how to use the Simulate command:

- Simulate

8. Program Memory Board

After you have used the Simulate command to verify that your system is performing correctly, you can try to program a board. You can use the Edit command to create a "dummy" data file if the actual data file is not available. If one board programs and verifies successfully, then add more boards to the interface adapter (up to your maximum design specification) and then try programming and verifying again. Read the following sections to create a data file and program the board:

- Creating and Editing Data Files, in Chapter 6
- Copy, in Chapter 3

9. Measure and Verify Waveforms Again

You should never assume that your system is ready to program boards after successfully programming only one board. You should program several more boards (repeat step 8) and measure waveforms again (like you did in step 7). Remember to test the system with the interface adapter fully loaded with boards.

10. Create Data Files and/or Download

After you have completed step 9, you can download the actual data files and program your first production run of boards. Read the following section in Chapter 6 to download data files from your development environment:

- Download

11. Program First Set of Boards

When you are confident that the entire system is ready for use, you can program the first set of boards with actual data. Read the following section to program the boards:

- Copy, in Chapter 3

12. Test Programmed Boards

You can now test the first set of boards in your target system.

5 Hardware Design Guide

Introduction

This chapter contains information to help you design your programmable circuit board and interface adapter. The chapter is divided into the following sections:

- **Design Rules for Programmable Devices**—gives you general design information on different programmable device types
- **Designing Circuit Boards to be In-Circuit-Programmable**—gives you general information on designing your programmable boards
- **General Information on Hardware Interfacing**—gives you some general guidelines to follow to design the hardware interface to BoardSite
- **Design Rules for BoardSite Interface Signals**—gives you specific information on how to use the BoardSite interface signal lines
- **Designing the Interface Adapter**—gives you specific information on the BoardSite interface connectors, including connector pin assignment, signal descriptions, and physical dimensions

Design Rules for Programmable Devices

The first topic in this section describes the four programmable device types, and gives general information on how to use these four types on programmable circuit boards. You must understand the differences between these four types to successfully design programmable circuit boards.

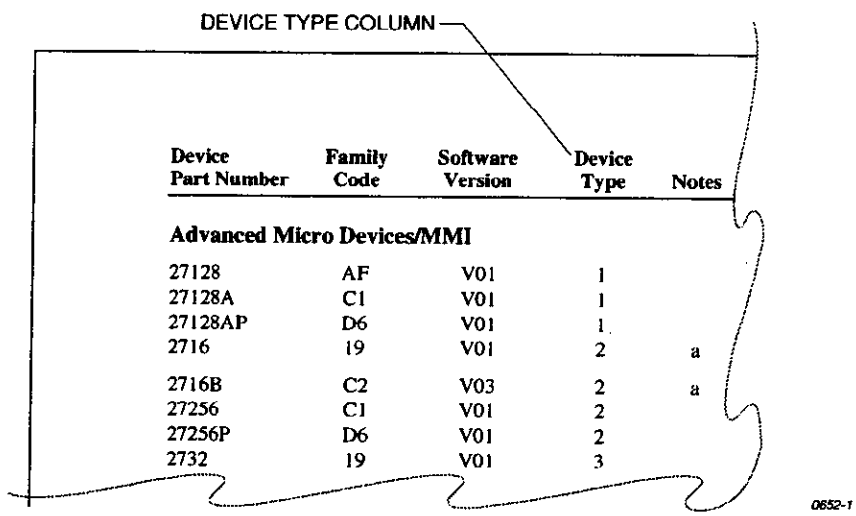
The second topic in this section contains general information on programmable device architectures (EPROMs, EEPROMs, and so on).

The Four Programmable Device Types

There are four basic device types, (called Type 1, Type 2, Type 3, and Type 4) that you can use on a programmable circuit board. Each type is assigned to a set of devices with similar programming requirements. Because device types are based only on programming requirements, functionally different devices may have the same device type. For example, the 27128 EPROM, 2864 EEPROM, and 8751 microcontroller are all Type 1 devices, even though they have very different post-programming functions.

The BoardSite Device List (located behind the "Device List" tab of this manual) has a column for device type number. Figure 5-1 shows a portion of the BoardSite Device List and shows where you find the device type number. Always refer to the device list to select the correct device type.

Figure 5-1
Device Types on the Device List



These four device types have different programming and interfacing requirements, as described in the following topics.

Type 1 Devices

Type 1 devices have independent CE, OE, PGM, VCC, and VPP inputs. Examples of Type 1 devices are 2764, 27128, 27010, and 271024. For Type 1 devices, all address, data, VCC, VPP, OE, and PGM lines can be connected in parallel to all devices. The devices should have their individual CE lines controllable by BoardSite, either directly or through an address decoder. The decoder can be continuously enabled, or enabled externally by BoardSite.

Type 2 Devices

Type 2 devices have a multiplexed CE/PGM pin, and have independent OE, VCC, and VPP inputs. Examples of Type 2 devices are 2716 and 27256. Type 2 devices have unique programming characteristics that make them more difficult to program in-circuit. During programming, the CE pin is used for the programming pulse while OE and VPP are both high. When OE is brought low, the device always presents its data outputs on the data bus, regardless of the state of CE. If you don't follow some special design precautions, this can cause problems for boards containing multiple devices on a data bus.

All address, data, and VCC lines can be connected in parallel to all devices. The devices should have their individual CE lines controllable by the programmer, either directly or through an address decoder. BoardSite must have control of the address decoder enable lines, because the CE pin is the programming pin for these devices, and it requires a precise programming pulse from BoardSite.

Each OE line for each device group must also be controllable by BoardSite, either directly or through an address decoder. BoardSite must have control of the address decoder enable lines. When the programmer reads the data in the device during a programming operation, it needs to selectively enable each OE line, depending on which group it needs to access.

Alternately, each VPP line for each device group must be directly controllable by BoardSite. This allows BoardSite (or the interface adapter) to apply a high-level VPP voltage to the devices being programmed, while VCC is at normal level on the other devices. When the programmer reads the data in the device during a programming operation, only the devices with a high-level VPP voltage would respond when OE is brought low; the other devices would be disabled by driving their CE pins high.

Type 3 Devices

Type 3 devices have a multiplexed CE/PGM input, and a multiplexed VPP/OE input. Examples of Type 3 devices are 2732 and 27512. For Type 3 devices, all address, data, VCC and VPP/OE lines can be connected in parallel to all devices. The devices should have their individual CE lines controllable by the programmer, either directly or through an address decoder. BoardSite must have control of the address decoder enable lines, because the CE pin is the programming pin for these devices, and it requires a precise programming pulse from BoardSite.

Type 4 Devices

Type 4 devices have a multiplexed CS/VPP input, and also require a pulsed VPP voltage. Examples of Type 4 devices are 57C49, 36C16, and 36C32. For Type 4 devices, all address, data and VCC lines can be connected in parallel to all devices. If the devices have a CS or CE pin that disables programming, then all VPP pins can be connected in parallel to the devices, with the individual CS or CE pins available directly to BoardSite. If the devices have no CS or CE pin available, then the individual VPP lines must be directly controllable by the programmer.

Note: For several design examples using each of these device types, see "General Information on Hardware Interfacing" later in this chapter.

Programmable Device Architectures

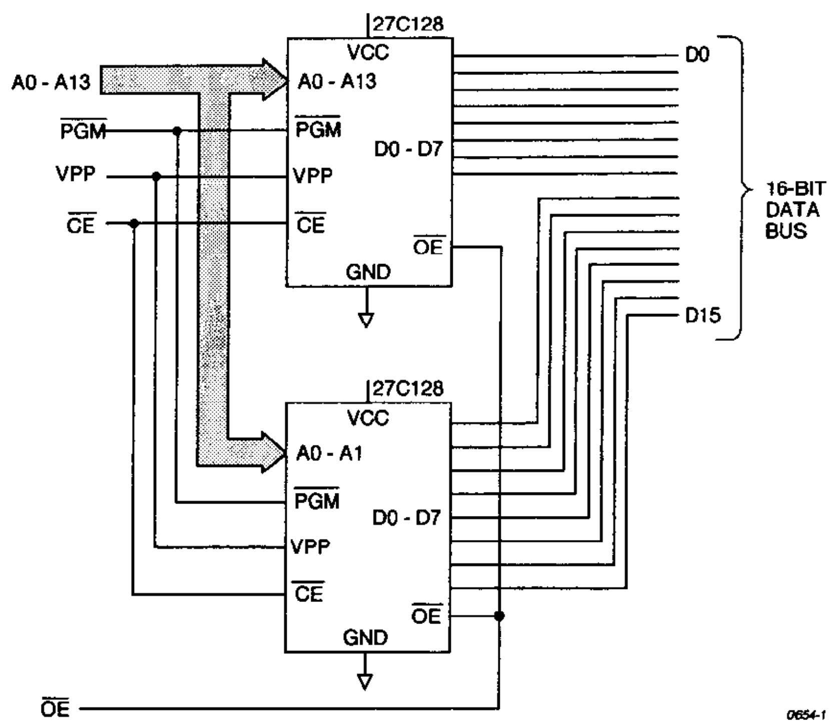
This section describes design rules for different memory device architectures, including EPROMs, EEPROMs, FLASH, and microcontrollers.

EPROMs

EPROMs are the most common type of memory device programmed in-circuit. In general, all EPROM address and data lines can be bused together to the interface adapter. All memory VCCs can be connected, and all memory VPPs usually can be connected, and then run off-board to the interface adapter.

When you design a 16-bit memory data bus, you usually use two 8-bit EPROMs to create a 16-bit data word. All EPROMs responding to the same address can have their CE, OE, and PGM pins driven in parallel. The programmer treats the EPROM pair as a single word-wide device. The same technique can be used to create 32-bit (or wider) buses. See Figure 5-2.

Figure 5-2
Connecting 8-bit EPROMs to Create 16-bit Data Bus



EEPROMs

EEPROMs are also popular in-circuit programmable devices, and come in two basic configurations: those requiring a high programming or erase voltage, and those not requiring it.

In general, all EEPROM address and data lines can be bused together to the interface adapter. All memory VCCs can be connected also.

For devices requiring high programming voltage (VPP), this voltage is often a pulsed VPP, unlike EPROMs. If you can disable programming with a CE or CS pin, then you can connect all VPP pins together to the interface adapter. If the pulse is applied to the CE pin instead of a VPP pin, then individual CE pins should be run off-board.

For devices not requiring high programming voltage, such as 2816A, 2864 and 28256, the OE and WE pins for each device group at the same address can be connected. Each device CE pin can be driven by an on-board address decoder, or can be run off-board for direct control from the programmer. If an address decoder is used and requires an enable line, connect it off-board.

Some devices provide a RDY/BUSY pin to decrease programming time. Some manufacturers allow for ganging of the RDY/BUSY pins for memory arrays. These devices can then be bused to a single point off-board. Use of the RDY/BUSY pin is optional, and is not actually required for programming the devices.

Another useful feature of EEPROMs is the ability to provide software data protection (SDP), or locking of the data so it cannot be erased or overwritten without being unlocked. This feature is usually accomplished by writing a special sequence to the EEPROM in a manner similar to normal programming, so that no special design considerations are required for implementing this feature in-circuit.

FLASH

FLASH devices are similar to EEPROMs, but require special programming considerations. Many EEPROMs provide their own transparent erase-before-write operation, so they don't need a separate erase cycle before programming. Most FLASH devices must be erased before programming. Also, many EEPROMs do not require high voltages for programming, whereas most FLASH devices require a high-voltage VPP. Some FLASH devices are sensitive to being over-erased or over-programmed, which is not a concern for EPROMs or EEPROMs.

In general, all address and data lines to the FLASH devices can be bused together and run off-board. All FLASH VCCs can be bused to a single point off-board, and all FLASH VPPs can be bused to a single point off-board.

Typically, the OE and WE pins can be bused to single points off-board. Each device CE pin can be driven by an on-board address decoder, or can be run off-board for direct control from the programmer. If an address decoder is used and requires an enable line, connect it off-board.

Some FLASH devices require additional circuitry on the interface adapter for programming and erasing. See the Device List, later in this manual, and read the notes that accompany the Device List for information on the FLASH devices you are using. Also, you can call the Data I/O Customer Resource Center and ask for the FLASH Application Note.

Microcontrollers

Microcontrollers can contain EPROM, or EEPROM, and will eventually contain FLASH memory so that any of the above discussions could apply to programming them.

In general, all address and data lines from the microcontrollers can be bused together and run off-board. Multiple microcontrollers can coexist on the same buses. All VCCs can be bused off-board to a single point.

For devices that require a digital pulse, such as 8751, all VPPs can be bused to a single point off-board. The programming pin for each device should be connected independently off-board. All other control lines required for programming can probably be bused to single points off-board.

For devices that require a high voltage pulse, such as 8748 and 8749, all high voltage pins should be independently connected off-board for each micro. Any other digital control lines required for programming should also be connected off-board.

Remember that the programmer must have complete control of the board and its address and data buses, so any other circuitry interfacing to the microcontroller must be capable of being disabled by the programmer.

Designing Circuit Boards to be In-Circuit-Programmable

This section contains general information on programmable circuit board design.

Programmable Circuit Board Design Rules

There are several basic design rules you follow to design a memory board that is in-circuit programmable. When you design a memory board, you usually check the manufacturer's data book to ensure that the memory device's operating specifications are met. With in-circuit programming, you must also ensure that the device's programming specifications are met. You must create a "programming environment" on the circuit board. Here are the design rules you follow to create that environment.

1. Use MOS or CMOS

Use either MOS or CMOS memory devices. The MOS/CMOS programming environment is very similar to the MOS/CMOS operating environment. On the other hand, TTL or bipolar memory devices require unique timing and voltage requirements, which makes them difficult to in-circuit program. You can use externally programmed (not in-circuit programmed) TTL and bipolar memory devices on the circuit board, however. They should be programmed and installed before the BoardSite in-circuit programming operations.

2. Understand Programming Specifications

Thoroughly understand your device's programming requirements. Check the device manufacturer's data book for timing and voltages required for programming your specific memory type. This will help ensure that your board design does not inadvertently prevent full compliance to the manufacturer's programming specification.

3. Don't Mix Devices

Do not mix functionally equivalent memory devices from different manufacturers. For example, the 27C256 EPROM is manufactured by many different companies. After programming, all 27C256s are functionally identical (assuming they have identical access times, and so on). However, the AMD 27C256 programs differently than the ATMEL 27C256, which programs differently than the Fujitsu 27C256, which programs differently than the Intel 27C256.

If you plan to specify several alternate sources for memory devices, refer to the BoardSite device list to determine whether the alternate sources are programming-compatible with the primary source. To be programming-compatible, all devices must have the same family code. If two devices have different family codes, then they are not programming-compatible.

If you mix devices in a BoardSite device group, you must ensure that they all use the same programming algorithm (which means they must have the same family code). Similarly, if you program several boards in parallel, devices programmed simultaneously on each board must have the same family code.

To ensure programming compatibility, you should assign different part numbers to programming-incompatible devices. For example, assign different part numbers to the AMD 27C256, the ATMEL 27C256, the Fujitsu 27C256, and the Intel 27C256. Because the Fujitsu 27C256 and the NEC 27C256 both have family code 45 (refer to the BoardSite device list), they are programming-compatible, and you can assign them the same part number.

Also, be very careful not to mix devices with the same base part number and different suffixes. For example, the Fujitsu 27C256 has a different family code than the Fujitsu 27C256A, so they are not programming-compatible.

4. Use Separate VPP and VCC

Design your board with separate VCC and VPP lines. VPP is usually raised to a high voltage during programming (typically from 12.5V to 21V), and must be connected to the devices separately from VCC, to prevent damage to other devices. Be sure that BoardSite can access both VCC and VPP at your board's interface connector.

5. Use Separate VCC for Memory and Logic

Design your board with separate TTL (logic) VCC and memory VCC. Memory device VCC is typically raised to a higher voltage (usually 6.5V) during programming, and must be connected to the devices separately from logic VCC, to prevent damage to logic devices. Boards designed with CMOS digital logic may not have this constraint; be aware of the requirements of your logic circuits. Be sure that BoardSite can access both VCCs at your board's interface connector.

6. Provide Access to Control Signals

Provide the programmer full access to all control signals. The programmer should have access to PGM, OE, ALE, R/W, RESET, CE, CS, DECODER ENABLE, BUFFER ENABLE, and so on.

The example in Figure 5-3 shows you how to provide direct access, through interface connector P1, to each EPROM's PGM line and OE line (P1-15 and P1-16), and to the data buffer's OE line and DIR line (P1-28 and P1-14). The address decoder selects the EPROM CE lines, based on the three high-order address lines from the interface connector (P1-17, P1-18, and P1-19). Notice the separate VPP and VCC lines (P1-30 and P1-31), and the separate memory VCC and logic VCC (P1-31 and P1-32).

7. Provide Access to Buses

Provide the programmer full access to all buses. The programmer must be able to write addresses and to write and read data. Figure 5-3 shows a board that uses two 74LS244s as the address bus drivers, and a 74LS245 as a bidirectional data bus transceiver.

8. Use Decoupling Capacitors

Use decoupling capacitors on VCC and VPP. Device manufacturers often require decoupling capacitors on each device's VCC and VPP lines, to prevent transient spikes from damaging the device or causing spurious operation. These capacitors should be as close as possible to the device's power and ground pins. Also, strategically place extra capacitors around the board if there are many memory devices. See Figure 5-4.

Figure 5-3
Example Design Showing PGM, OE, and CE

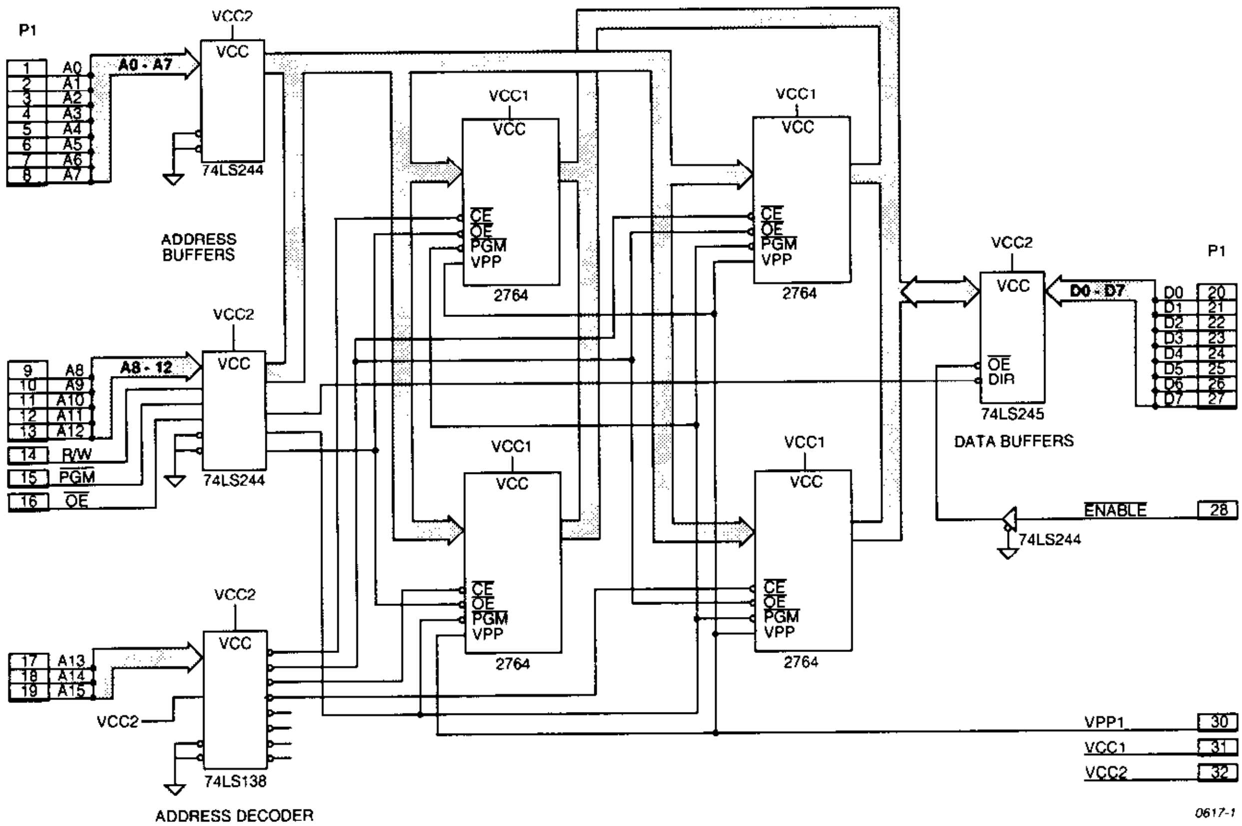
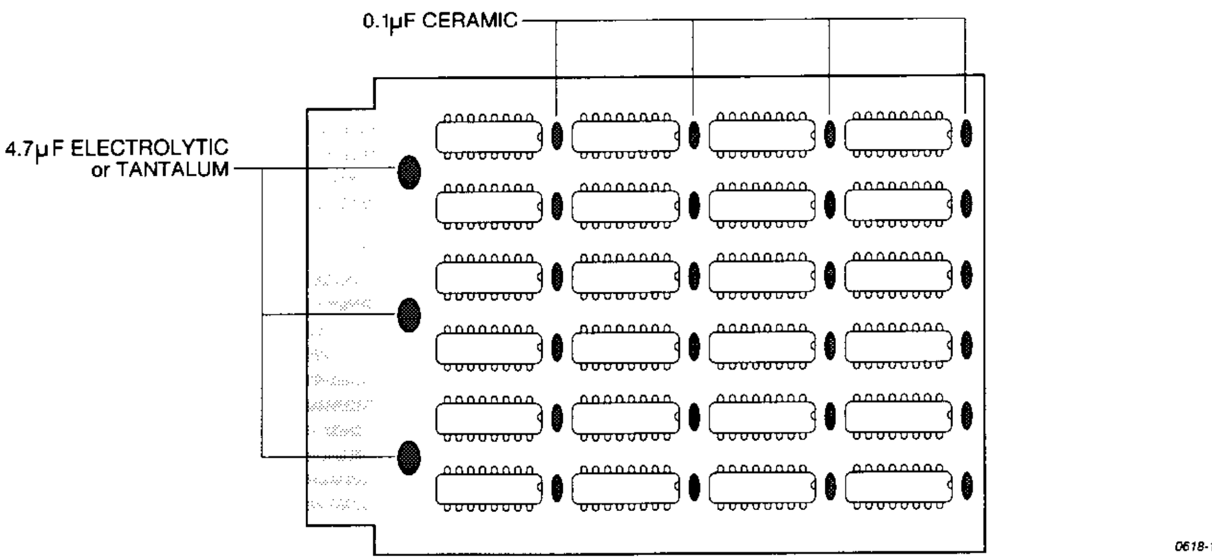


Figure 5-4
Decoupling Capacitors

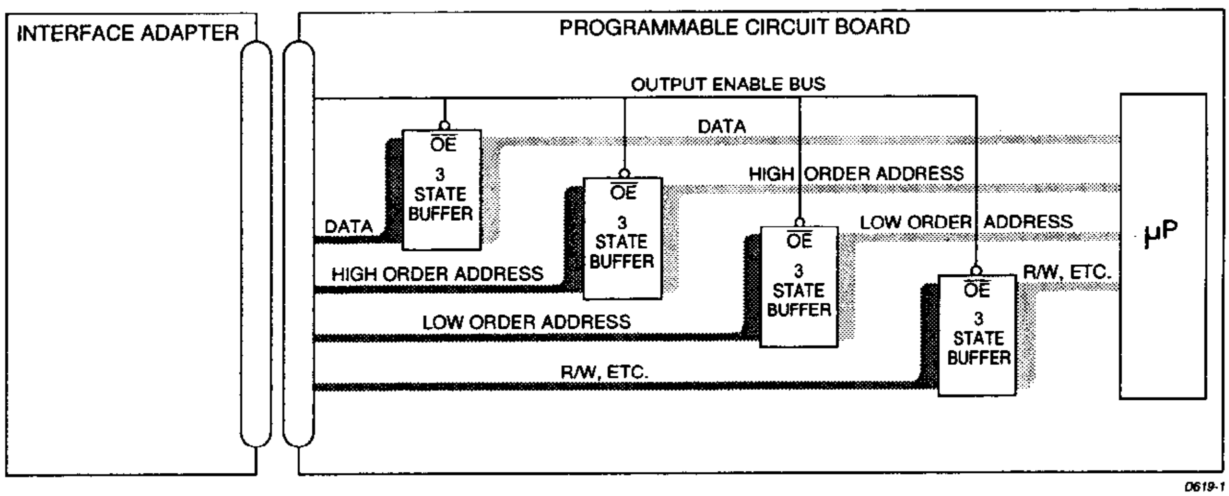


9. Disable Microprocessors During Programming

If your board contains a microprocessor or microcontroller, allow the programmer to disable it. If you don't disable it, then the microprocessor and the programmer may contend for the buses and control signals.

If the micro has a RESET or HALT pin that tri-states all its outputs, then allow the programmer to control that pin. If the micro's outputs cannot be disabled, use tri-state bus transceivers and buffers to disable the microprocessor, as shown in Figure 5-5. Connect the transceiver/buffer OE lines to the interface connector. Use a BoardSite control line (C0, for example) to disable the microprocessor by driving the OE line high.

Figure 5-5
Disabling a Microprocessor



Miscellaneous Design Rules

Here are a few miscellaneous design rules for programmable circuit boards.

Memory Board IDs and Electronic Identifiers

You should always consider designing a board identification code (board ID) into your programmable circuit boards. You can change the board ID based on the device types being used, the hardware revision of the board, and so on.

For example, you may program the same data into every board, but you manufacture some boards with AMD 27C256s and others with Intel 27C256s. Because these devices have different family codes, BoardSite must use different programming algorithms. You should provide a different board ID for each board, so that BoardSite can determine if the operator has selected the correct board profile.

Another way to ensure correct algorithms is to program BoardSite to read the electronic identifier (for devices so equipped), and then compare the identifier with a predefined constant in the sequence file.

To read the electronic identifier, the programmer enables a high voltage (typically 12V) on address pin A9 of the interface adapter. For circuit boards that contain only EPROMs, applying a high voltage to this pin should not be a problem. For boards containing additional logic connected to pin A9, application of 12V may damage the logic. In this case, you can design additional isolation circuitry to protect the logic. To implement this capability, additional circuitry must be added to the interface adapter to multiplex the 12V identifier-read voltage to A9.

Serial EPROMs, and Serial EEPROMs

Some EPROMs and EEPROMs have serial data inputs, and are known as "serial" devices. Serial devices typically have small arrays and small packages. The small packages are due to the reduced number of address lines, and the serial data input.

To program these devices, you must provide access to all the lines on the device. BoardSite has a parallel data bus that won't interface directly to the serial data input of the device. You must design a serial-to-parallel/parallel-to-serial converter on the interface adapter, to make the serial device appear like a parallel device to the programmer. Or, if you can tolerate slightly slower programming, you can modify the sequence file to shift the data (on a single data line) in software.

Multiplexed Address and Data Buses

Memory boards often have multiplexed address and data buses. If your board uses a multiplexed address/data bus, the address latch signal (typically designated ALE) must be available to BoardSite. You can use one of BoardSite's control lines (C0-C23) to drive the ALE line. For more information, see the section, "Multiplexed Address and Data," later in this chapter, and the section, "How to Modify a Sequence," in Chapter 7.

General Information on Hardware Interfacing

This section describes several different hardware interface design techniques and design rules. The section is organized by programmable device type (Type 1, Type 2, and so on).

Type 1 Devices 2764, 27128, 27010, 271024

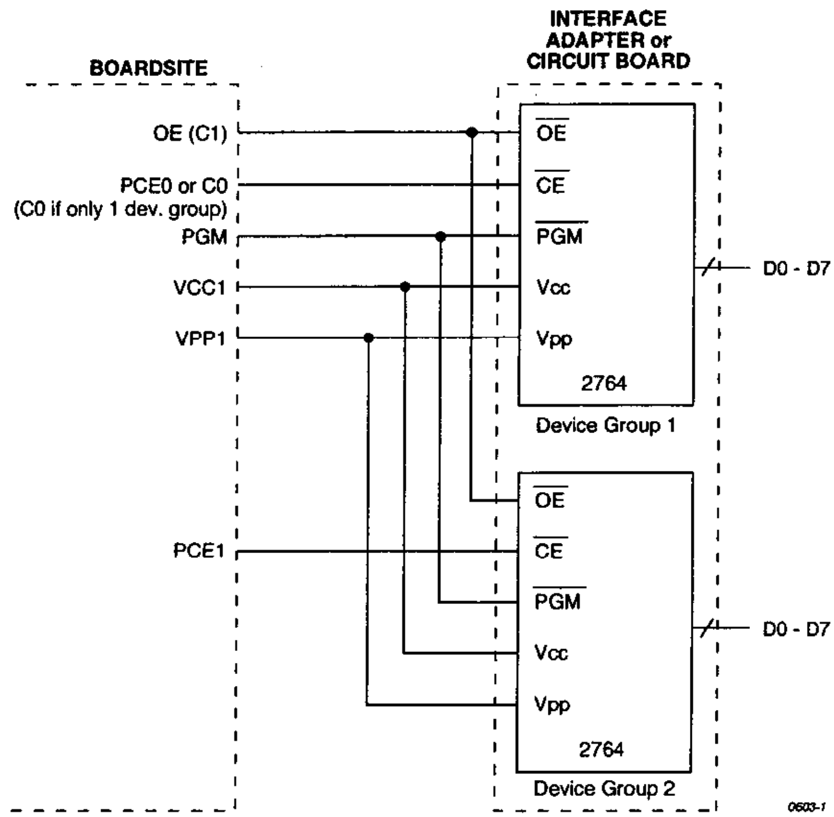
Type 1 devices have independent inputs for CE, OE, PGM, VCC, VPP, address, and data. Two Type 1 designs are shown here: a board without an address decoder (circuit board 1-A), and a board with an address decoder (circuit board 1-B).

Note: EPROMs that support page mode programming, such as the Fujitsu 27C1024, use the CE pin as a page/byte mode control. Even though the Fujitsu 27C1024 is functionally equivalent (after programming) to Type-1-classified 27C1024s from other manufacturers, it is not a Type 1 device. For programming purposes, page mode EPROMs are Type 2 devices.

Circuit Board 1-A

Circuit board 1-A contains two 8-bit device groups. Circuit board 1-A has all of the CEs for all of the device groups available to BoardSite. You can connect all OEs together on the circuit board or on the interface adapter. You can also connect all PGMs together on the circuit board or on the interface adapter. See Figure 5-6.

Figure 5-6
Circuit Board with Type 1 Devices, No Address Decoder



Design rules for circuit board 1-A are:

1. Connect the BoardSite PGM line to all of the board PGM pins.
2. Connect C1 to the board OE. If the board has only one device group, connect C0 to the device group CE. If the board has several device groups, connect any programmer PCE line to each device group CE. If you require more PCE lines than BoardSite has, design an address decoder on the interface adapter (see circuit board 1-B for an example).
3. Connect the programmer VCC1 to all the device VCCs and the programmer VPP1 to all the device VPPs. Connect the programmer address and data lines to all devices.

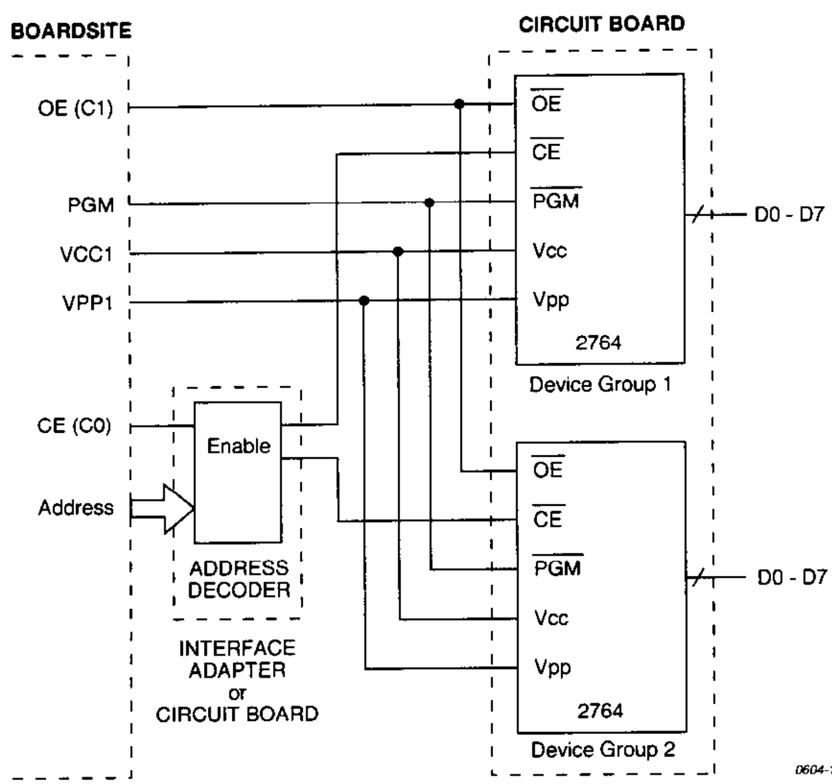
The PCE lines select the proper device group based on the PCE assignments in the Board Profile. To enable the PCE lines, you may have to add thc primitives pce_enable and pce_set to the SEQ_power_up sequence. See Chapter 7, "Sequence Editor Reference," for more information on these primitives and on using the BoardSite Sequence Editor.

If you only use a single device group, you don't have to modify the sequence file except to add additional control lines the board requires. If you use multiple device groups, you may have to make minor changes to the sequence file to enable the PCE lines.

Circuit Board 1-B

Circuit board 1-B contains two 8-bit device groups. Circuit board 1-B has an on-board address decoder (or you design an address decoder on the interface adapter) to select the device CEs. The address decoder's enable line is available to the programmer. You can connect all OEs together on-board or on the interface adapter. You can also connect all PGMs together on-board or on the interface adapter. See Figure 5-7.

Figure 5-7
Circuit Board with Type 1 Devices and Address Decoder



Design rules for circuit board 1-B are:

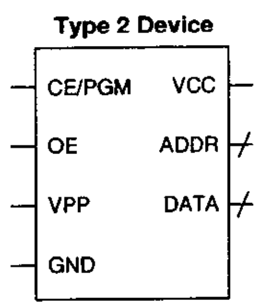
1. Connect the BoardSite PGM line to all of the board PGM pins.
2. Connect C1 to the board OE. Either connect C0 to the address decoder enable line, or hard-wire the enable line to ground to continuously enable the decoder.
3. Connect the programmer VCC1 to all the device VCCs and the programmer VPP1 to all the device VPPs.
4. Connect the programmer address and data lines to all devices or decoders.

The address decoder selects the required device group based on the address output from the programmer. For this design, you don't have to change the sequence file, except to add any additional control lines the board requires.

Type 2 Devices 2716, 27256

Type 2 devices have a multiplexed CE/PGM pin, and have independent inputs for OE, VPP, VCC, address, and data. Figure 5-8 shows a Type 2 device.

Figure 5-8
Type 2 Device Characteristics



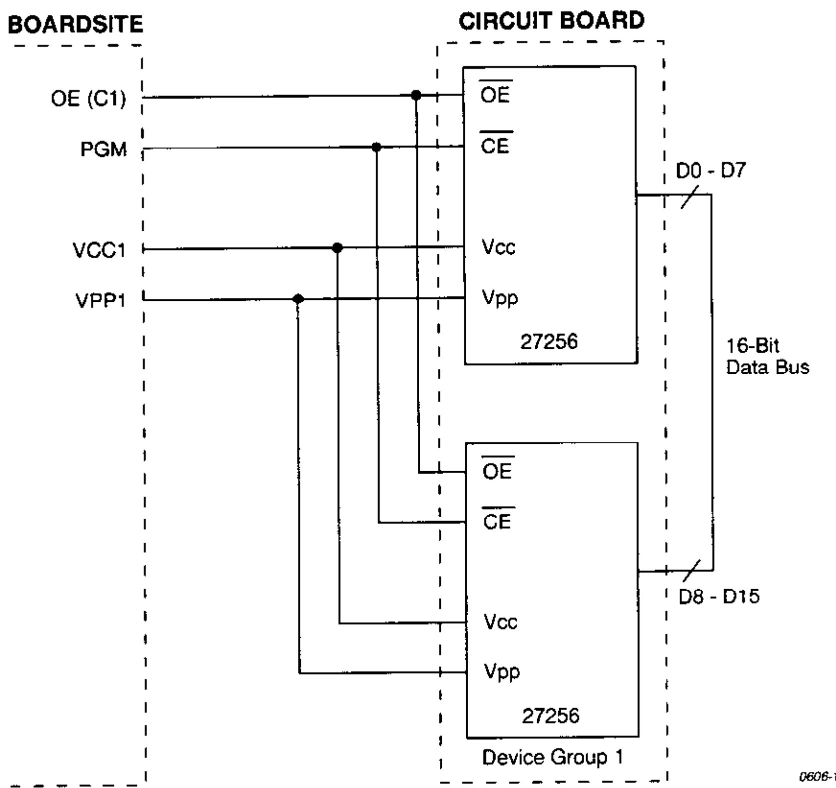
This section describes five Type 2 designs:

1. Circuit board 2-A: board has only 1 device group.
2. Circuit board 2-B: board contains multiple device groups without an address decoder. Circuit board 2-B has individual OEs available.
3. Circuit board 2-C: board contains multiple device groups with an address decoder. Circuit board 2-C has individual OEs available.
4. Circuit board 2-D: board contains multiple device groups, does not have individual OEs available, and allows access to individual CEs and VPPs.
5. Circuit board 2-E: board contains multiple device groups and does not allow individual access to OEs or VPPs.

Circuit Board 2-A

Circuit board 2-A contains only one 16-bit device group and provides access to the device group CEs and OEs. You can connect all CEs together on the circuit board or on the interface adapter. You can also connect all OEs together on the circuit board or on the interface adapter. See Figure 5-9.

Figure 5-9
Circuit Board with a Single Type 2 Device Group



Design rules for circuit board 2-A are:

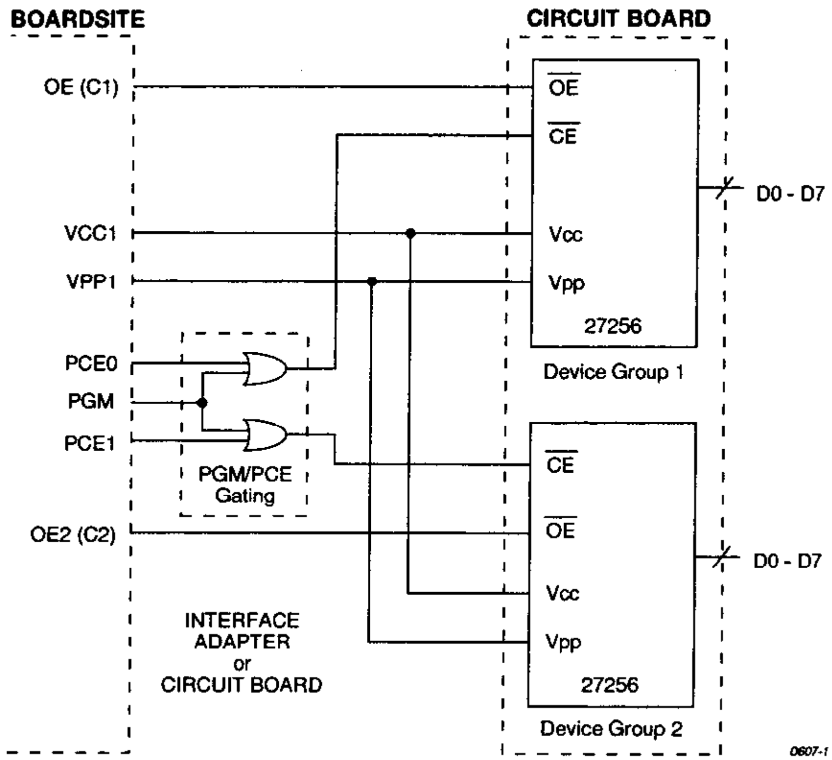
1. Connect the BoardSite PGM line to all device CEs.
2. Connect C1 to the board OE.
3. Connect the programmer VCC1 to all the device VCCs and the programmer VPP1 to all the device VPPs.
4. Connect the programmer address and data lines to all devices.

The programmer PGM line provides the programming pulse to the CE pin, and also drives CE active to read the devices. For this design, you don't have to change the sequence file, except to add any additional control lines or power-up sequencing the board requires.

Circuit Board 2-B

Circuit board 2-B contains two 8-bit device groups. Circuit board 2-B provides access to all device CEs and OEs and does not have an on-board address decoder. This board requires additional logic gates on the interface adapter. See Figure 5-10.

Figure 5-10
Circuit Board with Type 2 Devices, Multiple Device Groups, and No Address Decoder



Design rules for circuit board 2-B are:

1. Connect a BoardSite control line (C0, C1, etc.) to each device OE.
2. Connect the programmer VCC1 to all the device VCCs and the programmer VPP1 to all the device VPPs.
3. Connect the programmer address and data lines to all devices.

To deliver an accurately timed CE programming pulse, gate the BoardSite PGM line with individual PCE lines on the interface adapter. In the board profile, program the PGM and PCE lines active low for OR gates (as shown in Figure 5-10), or active high for NAND gates. BoardSite automatically drives the gated PCE/PGM signals active to read the devices.

Note: If there are more device groups on the board than available control lines on BoardSite, design an OE gating circuit (similar to the PGM gating circuit of Figure 5-10) on the interface adapter. Use C1 as the gated control line.

For this design, change the sequence file to enable the PCE lines (for more information, see the section "Programmable Chip Enable Lines," later in this chapter). If you need more PCE lines than BoardSite has available, design an address decoder on the interface adapter to drive the CE lines. For more information, see the next section, "Circuit Board 2-C."

Also, change the sequence file to enable the proper control line to drive the desired OE pin. For example, to enable OE2:

1. Start the Sequence Editor.
2. Select the Define Control Pin Alias command from the Define menu. Press ↵.
3. In the H/W-pin Alias pop-up, select **C2**. Press ↵.
4. In the Define Control Pin Alias pop-up, type **OE2.** Press ↵.
5. Notice that the OE2 alias has been assigned to control pin C2, as shown in the H/W-pin Alias pop-up.
6. Press Esc to remove the pop-up.

Next, use the Find String command from the Miscellaneous menu to search the SEQ_enable_device sequence for the variable OE. Then, edit the sequence to add the appropriate control statements and primitives. Table 5-1 shows the code changes you make.

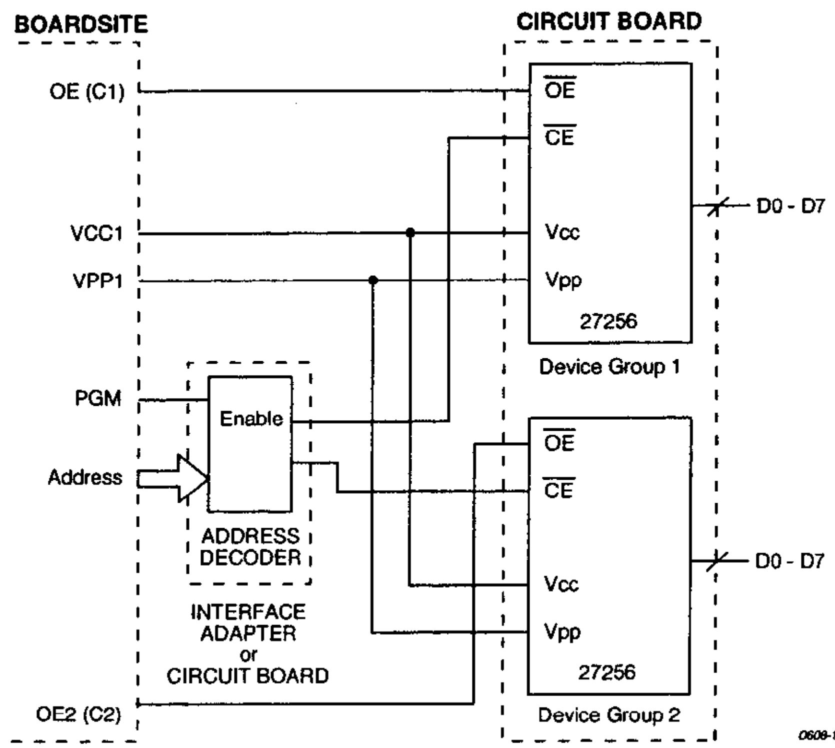
Table 5-1
Example Code Changes to Sequence File

Existing Sequence	**Change Sequence To**
SEQ_enable_device	SEQ_enable_device
.	.
.	.
.	.
{	{
control_bit_set (OE,LO);	if (DEVICE_GROUP_NUMBER = = 1)
}	{
	control bit_set (OE,LO);
	}
	else if (DEVICE_GROUP_NUMBER = = 2)
	{
	control_bit_set (OE2, LO):
	}
	}

Circuit Board 2-C

Circuit board 2-C contains two 8-bit device groups. The board either has an on-board address decoder for device group CEs, or you design a decoder on the interface adapter. The programmer has access to the individual device group OEs. See Figure 5-11.

Figure 5-11
Circuit Board with Type 2 Devices, Multiple Device Groups, and Address Decoder



Design rules for circuit board 2-C are:

1. Connect a programmer control line to each device group OE.
2. Connect the programmer PGM line to the address decoder's enable line.
3. Connect the programmer VCC1 to all the device VCCs and the programmer VPP1 to all the device VPPs.
4. Connect the programmer address and data lines to all devices and decoders.

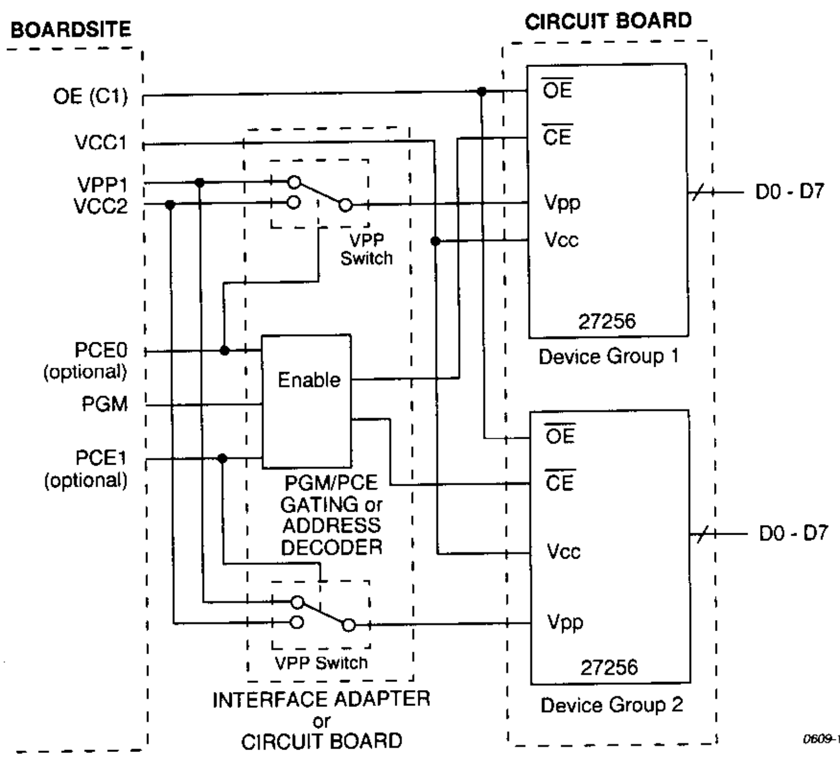
The programmer PGM line provides the programming pulse to the CE pin through the address decoder. The PGM line also drives CE active to read the devices. For this design, you don't have to change the sequence file, except to enable the control lines to drive the device group OEs. For more information on changing the sequence file to enable OE lines, see the preceding section, "Circuit Board 2-B."

Note: If there are more device groups on the board than available control lines on BoardSite, design an OE gating circuit (similar to the PGM gating circuit of Figure 5-10) on the interface adapter. Use C1 as the gated control line.

Circuit Board 2-D

Circuit board 2-D contains two 8-bit device groups. Because all the OEs are connected on the board, individual OE lines for each device group are not available. The programmer must have individual access to each device VPP. If VPPs are not individually available, see the section, "Circuit Board 2-E." The board may contain an on-board address decoder or the device CEs may be individually available to the programmer. See Figure 5-12.

Figure 5-12
Circuit Board with Type 2 Devices, Multiple Device Groups, and Access to CEs and VPPs



Design rules for circuit board 2-D are:

1. Connect C1 to the board OE.
2. Connect the programmer PGM line to the address decoder enable, or gate PGM with PCE lines as shown in Figure 5-10 (circuit board 2-B).
3. Connect the programmer VCC1 to all the device VCCs.
4. Connect the programmer address and data lines to all devices.

On the interface adapter, you must design a VPP switch circuit for each device group. The VPP switch drives VPP to programming voltage (12 volts) to program the device, and drives VPP to 5V (VCC2) to verify the device. Only one device group VPP is at programming voltage, while all the other device group VPPs are at 5V. You can control the VPP switches with the PCE lines corresponding to each device group.

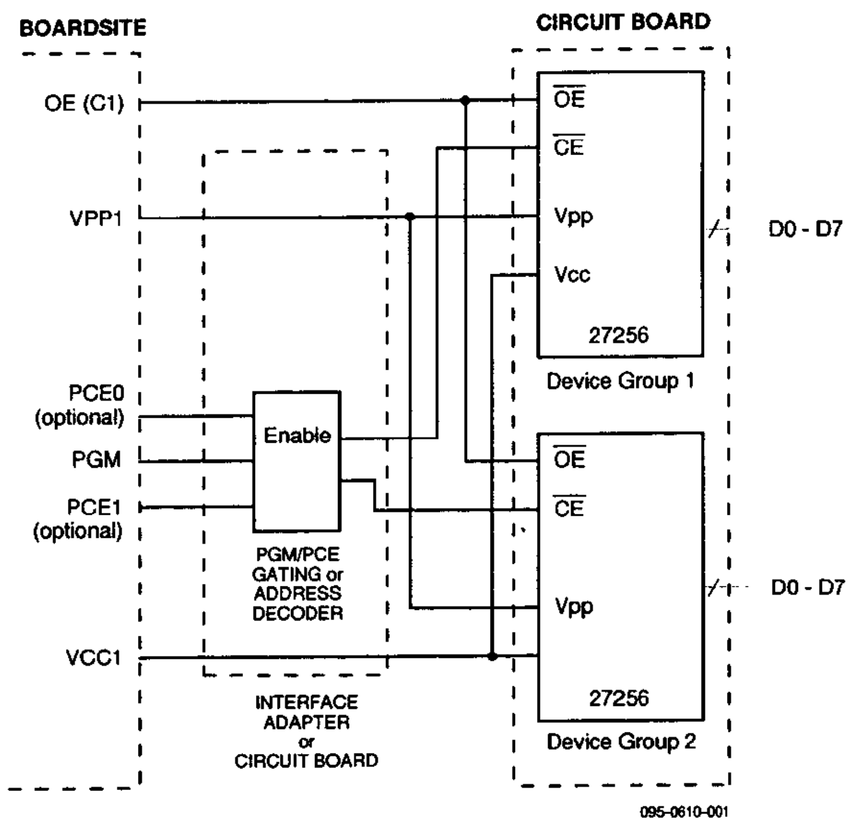
The programming algorithms for Type 2 devices specify a verify operation immediately after a programming pulse, with VPP at programming voltage. To do this, the programmer drives OE low, independent of the level of CE. This means that all devices that have programming voltage on VPP, and also have OE low, output data to the data bus. This may cause a data bus contention. If the devices not being programmed have 5V on VPP, then they are disabled by the CE level, and will not output data.

For this design, you only have to change the sequence file to enable the PCE lines, as described in the section "Circuit Board 2-B."

Circuit Board 2-E

Circuit board 2-E contains two device groups. Because all the OEs are connected on the board, individual OE lines for each device group are not available. The VPPs for all devices are also connected on the board. The board may contain an address decoder or the device CEs may be individually available to the programmer. See Figure 5-13.

Figure 5-13
Circuit Board with Type 2 Devices, Multiple Device Groups, and OEs Connected



Design rules for circuit board 2-E are:

1. Connect any BoardSite control line to the board OE.
2. Connect the programmer PGM line to the address decoder enable input, or gate PGM with the programmer PCE lines as shown in Figure 5-10 (circuit board 2-B) and Figure 18-11 (circuit board 2-C).
3. Connect the programmer VCC1 to all the device VCCs.
4. Connect the programmer address and data lines to all devices.
5. Connect the programmer VPP1 line to the board VPP. Between programming pulses, the programmer drives VPP from the programming voltage to 5V to verify the device, as required by the programming algorithm. Only those devices with both CE and OE low output data to the data bus.

For this design, you must change the sequence file to drive VPP to 5V after the programming pulse to verify the device, and then drive VPP back to programming voltage for the next pulse. If you use the PCE lines, you must change the sequence file to enable them.

In this design, BoardSite verifies the devices after the programming pulse by driving VPP to 5V, CE low, and OE low. Most manufacturers define the "normal" program verify cycle for Type 2 devices as VPP at programming voltage, with CE high and OE low. Although circuit board 2-E doesn't strictly comply with these programming specifications, it typically doesn't jeopardize device programming yield or data retention.

We recommend that you contact your memory device manufacturer's Application Engineering department if you plan to use this design. Ask them if using read mode (VPP=5V, CE=OE=0V) is acceptable as the program verify mode. Read mode is the "normal" post-programming memory mode—the mode the device is in when operating in your system.

Type 3 Devices 2732, 27512

Type 3 devices have a multiplexed CE/PGM input, a multiplexed OE/VPP input, address, data, and VCC. See Figure 5-14.

Figure 5-14
Type 3 Device Characteristics

This section describes three Type 3 designs:

1. Circuit board 3-A: board contains only 1 device group.
2. Circuit board 3-B: board contains multiple device groups and no address decoder.
3. Circuit board 3-C: board contains multiple device groups and an address decoder.

Circuit Board 3-A

This circuit board contains one 16-bit device group. All CEs are connected either on the board or on the interface adapter. See Figure 5-15.

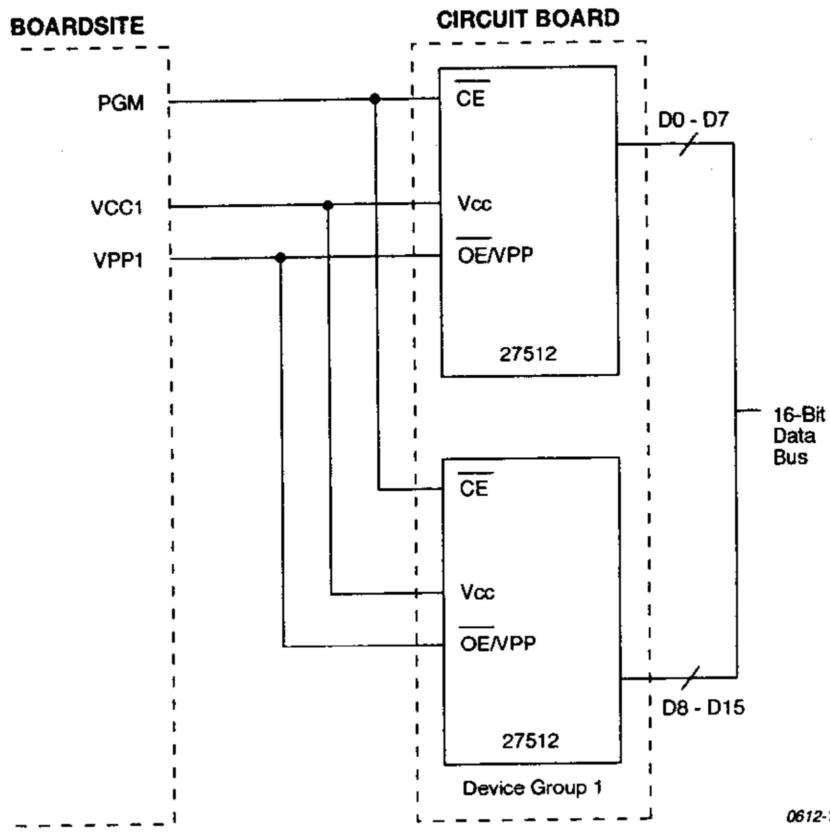


Figure 5-15
Circuit Board with Type 3 Devices, One Device Group

Design rules for circuit board 3-A are:

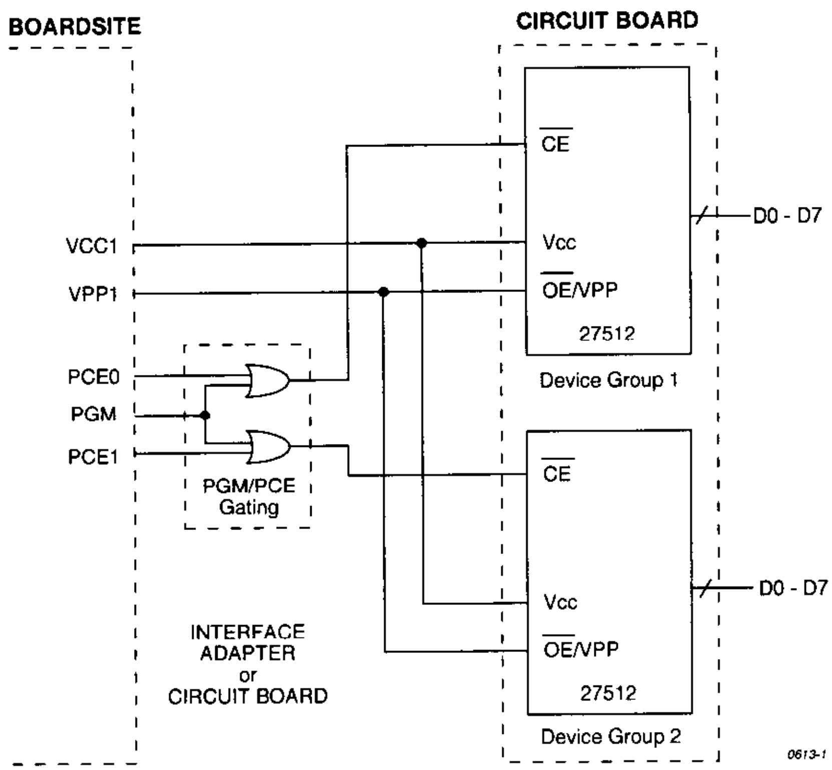
1. Connect the programmer PGM line to all the device CEs.
2. Connect the programmer VPP1 line to all device OE/VPPs.
3. Connect the programmer VCC1 to all the device VCCs.
4. Connect the programmer address and data lines to all devices.

The programmer PGM line provides the programming pulse to the CE pin. The programmer automatically drives CE active and VPP to 5V to read the devices. For this design, you don't have to change the sequence file, except to add any additional control lines or board power-up sequencing.

Circuit Board 3-B

This design contains two 8-bit device groups. The board provides access to all device CEs. All the OE/VPPs are connected either on the board or on the interface adapter. See Figure 5-16.

Figure 5-16
Circuit Board with Type 3 Devices, Multiple Device Groups, and with No Address Decoder



Design rules for circuit board 3-B are:

1. Connect the programmer VPP1 line to all device OE/VPPs.
2. Connect the programmer VCC1 line to all device VCCs.
3. Connect the programmer address and data lines to all devices.

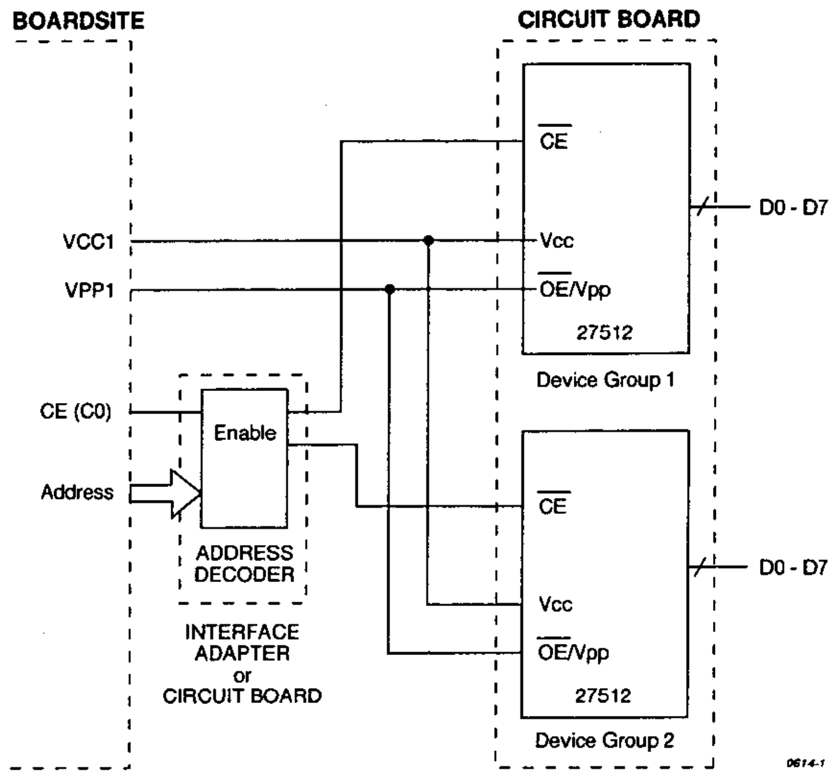
To deliver an accurately timed CE programming pulse, gate the BoardSite PGM line with individual PCE lines on the interface adapter. In the Board Profile, program the PGM and PCE lines active low for OR gates (as shown in Figure 5-10), or active high for NAND gates. BoardSite automatically drives the gated PCE/PGM signals active to read the devices.

For this design, change the sequence file to enable the PCE lines. For more information, see the section "Programmable Chip Enable Lines," later in this chapter. If you need more PCE lines than BoardSite has available, design an address decoder on the interface adapter to drive the CE lines. For more information, see the next section, "Circuit Board 3-C."

Circuit Board 3-C

This design contains several 8-bit device groups. The design has an address decoder either on the board or on the interface adapter. You can connect all OE/VPPs together either on-board or on the interface adapter. See Figure 5-17.

Figure 5-17
Circuit Board with Type 3 Devices, Multiple Device Groups, and an Address Decoder



Design rules for circuit board 3-C are:

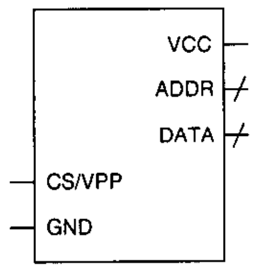
1. Connect the programmer VPP1 to all device OE/VPPs.
2. Connect the programmer VCC1 to all device VCCs.
3. Connect the programmer address and data lines to all devices.

The programmer PGM line provides the programming pulse to the CE pin through the address decoder. The PGM line also drives CE active to read the devices. For this design, you don't have to change the sequence file, except to add any control lines or board power-up sequencing.

Type 4 Devices 57C49, 36C16, 36C32

Type 4 devices require an accurately timed VPP pulse for programming. These devices usually have a multiplexed CS/VPP pin. Figure 5-19 shows a diagram of the 57C49 8K x 8 CMOS PROM.

Figure 5-18
Type 4 Device (57C49)



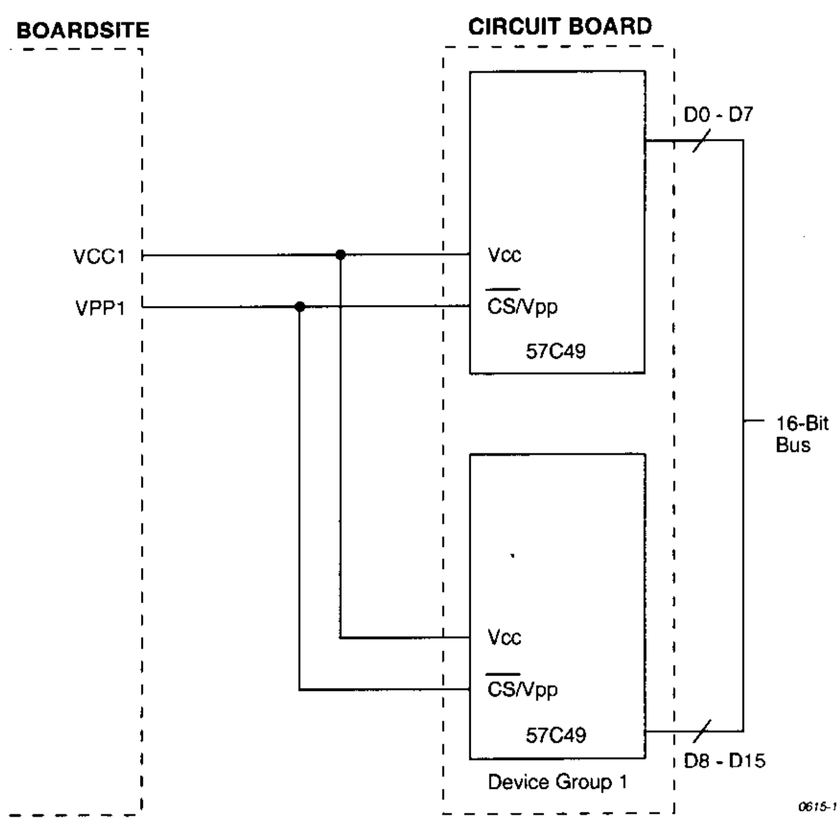
This section describes two Type 4 designs:

1. Circuit board 4-A: board contains one device group.
2. Circuit board 4-B: board contains several device groups.

Circuit Board 4-A

This design contains only one device group. See Figure 5-18.

Figure 5-19
Circuit Board with Type 4 Devices and One Device Group



Design rules for circuit board 4-A are:

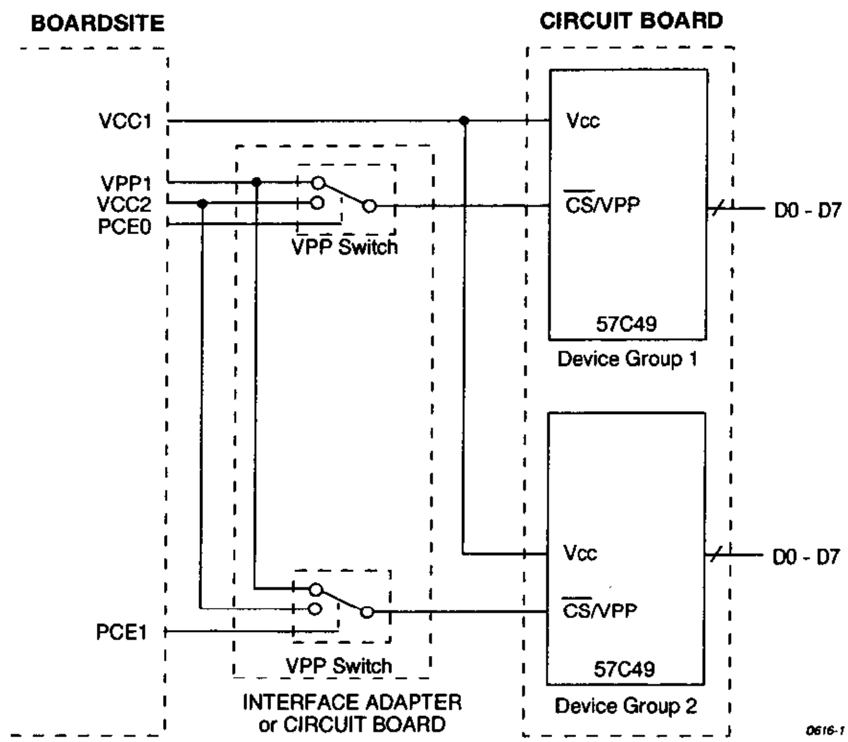
1. Connect the programmer VPP1 to the programming pin of the devices (in this case, the CS/VPP pin).
2. Connect the programmer VCC1 to all device VCCs.
3. Connect the programmer address and data lines to all devices. If other control lines are required, connect any programmer control line to these pins.

The programmer VPP line provides the programming pulse to the CS/VPP pin, and also drives CS/VPP active to read the devices. For this design, you don't have to change the sequence file, except to add any additional control lines or power-up sequencing the board requires.

Circuit Board 4-B

This design contains two 8-bit device groups. See Figure 5-20.

Figure 5-20
Circuit Board with Type 4 Devices, Multiple Device Groups



Design rules for circuit board 4-B are:

1. Connect the programmer VCC1 to all device VCCs.
2. Connect the programmer address and data lines to all devices.

If the device doesn't have an additional read/write control pin to disable programming (as in the 57C49 of Figure 5-19), then design VPP switches for each device group. To control the switches, use either the PCE lines or an address decoder on the interface adapter.

If the device doesn't have an additional read/write control pin to disable programming (as in the 57C49 of Figure 5-19), then design VPP switches for each device group. To control the switches, use either the PCE lines or an address decoder on the interface adapter.

If the device has additional read/write control pins (such as the CS2 and CS3 pins on the 36C16 EEPROM), connect VPP1 to all device programming pins (CS1 for the 36C16). Then connect the control pins (CS2 and CS3) to either the PCE lines or to an address decoder on the board or interface adapter.

Design Rules for BoardSite Interface Signals

With BoardSite, you can test, program, and verify a wide variety of programmable circuit boards. BoardSite gives you this flexibility by providing a large number of hardware interface signals that are controlled by the BoardSite system software.

This section explains isolated and non-isolated programming, describes the BoardSite hardware interface signals, and gives you specific design rules that you follow when you design your circuit board and BoardSite interface adapter.

The design rules for the BoardSite interface signals are divided into the following sections:

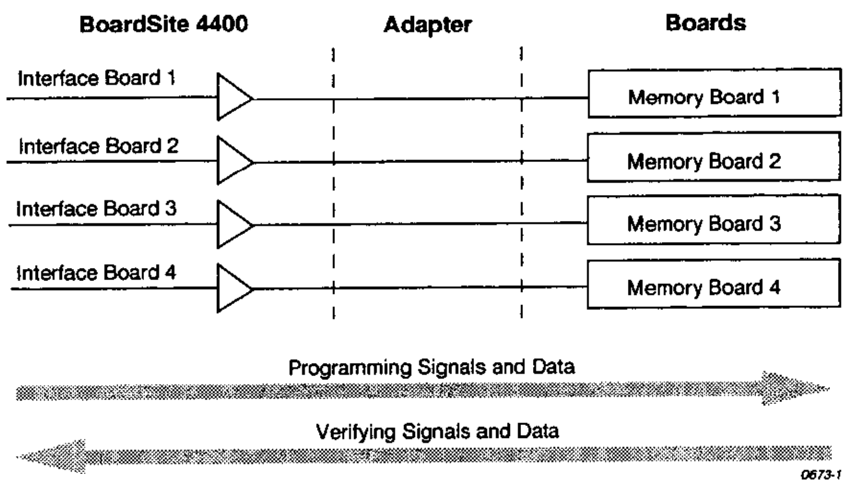
- Isolated Programming and Non-isolated Programming
- Adapter Detect Lines (ADAP0-ADAP1)
- Adapter ID Lines (ID0-ID7)
- Board Detect Lines (BD0-BD7)
- Board Enable Lines (BE0-BE7)
- Programmable Chip Enable Lines (PCE0-PCE15)
- LED Lines (LED0-LED7)
- Multiplexed Address and Data
- Programmable Power Supply Outputs
- Digital Outputs

Isolated Programming and Non-isolated Programming

Isolated programming is defined as programming a single circuit board for each BoardSite interface board. Therefore, using isolated programming, the single-interface-board BoardSite 4100 can program one board, and the four-interface-board BoardSite 4400 can program four boards.

In isolated programming, where each BoardSite interface board is dedicated to one programmable circuit board, failures on one circuit board do not affect any other board. For isolated programming, the interface adapter design is relatively simple, because you don't have to multiplex BoardSite signals to several boards. Figure 5-21 shows an example of isolated programming.

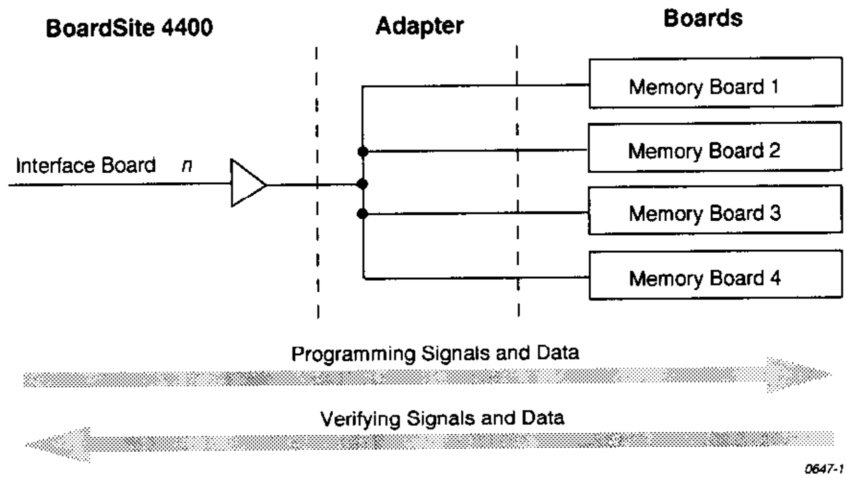
Figure 5-21
Isolated Programming



Non-isolated programming is defined as programming more than one board for each BoardSite interface board. Using non-isolated programming, BoardSite can program up to eight boards per interface board. Therefore, the 4100 can program up to eight boards and the 4400 can program up to 32 boards. Non-isolated programming can increase programming throughput.

In non-isolated programming, all the boards on a single interface board are programmed in parallel, but they are tested and verified one-at-a-time, using the Board Enable signals (BE0-BE7) to select the active board. For non-isolated programming, the interface adapter design may be more complicated, depending on the memory device types on your boards. Figure 5-22 shows an example of non-isolated programming.

Figure 5-22
Non-isolated Programming



For high-volume manufacturing with good programming yield, use non-isolated programming to increase BoardSite throughput. For low-volume manufacturing or one-of-a-kind applications, use isolated programming to simplify your interface designs. Table 5-2 summarizes the tradeoffs involved in this decision.

Table 5-2
Design Tradeoffs, Isolated versus Non-isolated Programming

	Isolated	**Non-isolated**
Throughput	Low	High
Fault Isolation	High	Moderate
Interface Complexity	Low	Moderate

Adapter Detect Lines (ADAP0-ADAP1)

Purpose

BoardSite uses the Adapter Detect lines (ADAP0 and ADAP1) to determine if there is an interface adapter installed on the programmer. If BoardSite doesn't detect an interface adapter, the programmer will not apply power to the interface connector (except when you run the Simulate command).

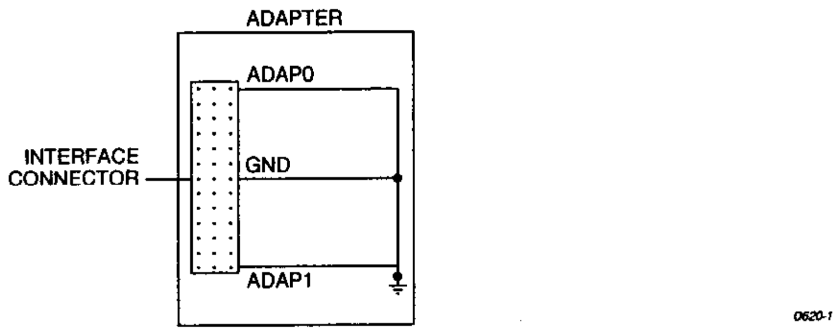
Operation

The two Adapter Detect lines (ADAP0 and ADAP1) are pulled up to a switched +5V in BoardSite through 3.3kΩ resistors. There are no voltages present on the interface connector until a circuit board operation is invoked. When a circuit board operation is invoked, BoardSite activates the switched +5V to perform the adapter detect operation. After BoardSite detects the adapter, the +5V pull-up is switched off again.

Implementation of Adapter Detect

To implement adapter detect, tie both ADAP0 and ADAP1 to any ground pin on the adapter (see Figure 5-23). Then, when the adapter is attached to the connector, ADAP0 and ADAP1 will be pulled to a TTL low level.

Figure 5-23
Adapter Detect



Note: Both ADAP0 and ADAP1 on each interface board connector must be connected to a TTL low (ground) on the adapter. ADAP0 and ADAP1 are arranged at opposite ends of the interface connector to ensure proper continuity of the connector. This guarantees that the interface connector pins all make proper contact before BoardSite applies power.

Also, power is not available to the circuit boards when BoardSite is reading ADAP0 and ADAP1. Therefore, you cannot use active circuits to drive ADAP0 and ADAP1.

Software Implementation

Add the adapter detect information in the Board Profile Editor.

1. Select the Edit command from the menu bar.
2. In the Edit Options pop-up, select the Board Profile option. Press [↵].
3. In the Edit File Name pop-up, select the board profile name you want to edit. Press [↵].
4. In the Edit Board Profile or Sequence File pop-up, select Board Editor. Press [↵].
5. In the Board Profile Editor, press [Tab] to activate the form window.
6. In the Interface Cards Used parameter, type Y for each interface card used. BoardSite detects adapters only for those interface cards you activate by typing Y. See Figure 5-24.

Figure 5-24
Adding Interface Information to the Board Information Form

```
BoardSite Board Profile Editor                         Fri 01-Jun-90 -  1:20:04
Editing:"demo_2816a"
Board Information                             ↑ ↓       Previous, Next Field
  Algorithm: XICOR 2816A                      Shift-Tab  Previous Field
    Devices: u16 u17 u18 u19                  ↵         Accept this Field
                                              PgUp PgDn  Previous,Next Page
                                              Ins,Del    Edit numbers, strings
                                              Alt-b      Save this Form
                                              Esc        Exit Form, Discard Changes
                                              F10        Context Sensitive Help
 Board Information Form
                            Required Parameters
                            Address Bus Width: 16
         Number of Programmable Chip Enables: No Programmable Chip Enables
                            Address Increment:    1
                          Interface Card Used: Y N N N
                          Master Board Number:  1
                              Isolated Boards: Y
                            Adapter I.D. Used: N
                                 Adapter I.D.: 00
                              Clock Frequency: 1
                     Board Enables Active High: N
                          PGM Line Active High: N
            < Enter 'Y' to ENABLE, 'N' TO DISABLE >
```

7. When you finish editing the form, press [Alt] - [B].

Adapter ID Lines (ID0-ID7)

Purpose

With BoardSite, you can use the Adapter ID Lines to determine if the interface adapter installed on the system corresponds to the board profile selected by the operator. The Adapter ID lines are hard-wired to create a hexadecimal number that must match the Adapter I.D. number in the board information form. If the numbers do not match, BoardSite will not apply power to the interface connector. Therefore, if you build different adapters for different memory boards, each adapter should have a unique Adapter ID.

Another possible use of the Adapter ID lines is to create unique IDs for different types of boards installed in a single adapter. For example, an interface adapter could be created with different connectors for different types of boards, with each board wired to create a different ID.

Operation

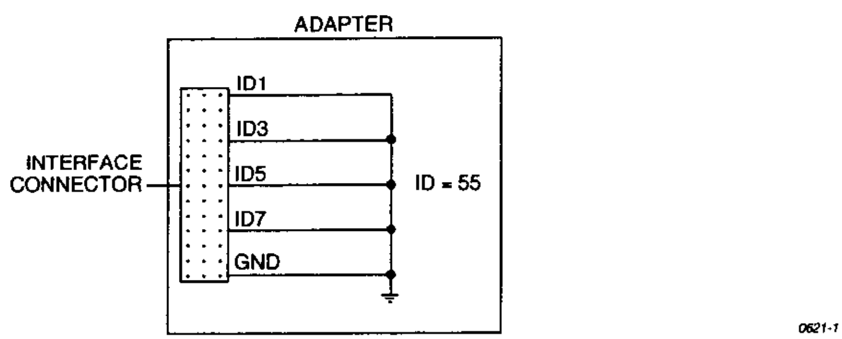
ID0-ID7 are pulled up to a switched +5V in the system through 270kΩ resistors. There are no voltages present on the interface connector until a circuit board operation is invoked. When a circuit board operation is invoked, BoardSite activates the switched +5V to perform the adapter ID operation. After BoardSite reads the ID, the +5V pull-up is switched off again.

Note: *Design your interface adapter so that the appropriate Adapter ID lines are pulled to a TTL low when the adapter is attached to the system. The Adapter ID lines must be pulled to a TTL low level to create an ID pattern. Because there are eight ID lines, there are 256 possible IDs. However, don't use FF_{HEX}, because this is the ID generated by an empty interface connector.*

Hardware Implementation of the Adapter ID

Create the Adapter ID by connecting the appropriate ID lines to ground on the interface adapter. BoardSite automatically pulls unconnected lines to a TTL high. For example, to create Adapter ID FE_{HEX}, connect ID0 to ground, and leave ID1-ID7 unconnected. To create Adapter ID 55_{HEX} connect ID1, ID3, ID5, and ID7 to ground, and leave ID0, ID2, ID4, and ID6 unconnected. See Figure 5-25.

Figure 5-25
Creating Adapter ID 55_{HEX}



To create an ID that is determined by both an adapter and a board, route one of the Adapter ID lines to a spare ground pin on the circuit board connector. This is similar to the Board Detect implementation (see the next section in this chapter, "Board Detect Lines"). When you insert the board into the connector, the board grounds one of the adapter ID lines, creating a new ID. Different connectors on the interface adapter could have different Adapter ID lines routed to them, thus creating different Adapter IDs for each board.

Note: *If your BoardSite contains several interface connectors (for example, the BoardSite 4400), you only have to connect ID0-ID7 on one interface connector. The system will look for a valid ID on any connector.*

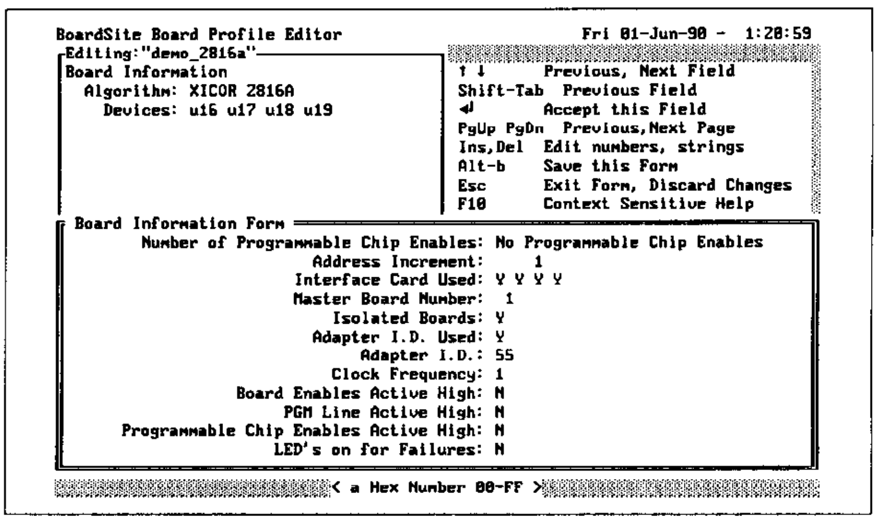
Also, power is not available to the circuit boards when BoardSite is reading ID0-ID7. Therefore, you cannot use active circuits to drive ID0-ID7.

Software Implementation

Add the Adapter ID information in the Board Profile Editor.

1. Select the Edit command from the menu bar.
2. In the Edit Options pop-up, select the Board Profile option. Press [↵].
3. In the Edit File Name pop-up, select the board profile name you want to edit. Press [↵].
4. In the Edit Board Profile or Sequence File pop-up, select Board Editor. Press [↵].
5. In the Board Profile Editor, press [Tab] to activate the form window.
6. In the Adapter ID Used parameter, type **Y.**
7. In the Adapter ID parameter, type the hex number for the ID. In this example, type 55. See Figure 5-26.

Figure 5-26
Adding Adapter ID Information to the Board Information Form



8. When you finish editing the form, press [Alt] - [B].

Board Detect Lines (BD0-BD7)

Purpose

BoardSite uses the Board Detect lines (BD0-BD7) to determine if circuit boards are installed in the interface adapter. If BoardSite doesn't detect any boards, it will not apply power to the interface connector (except when you run the Simulate command).

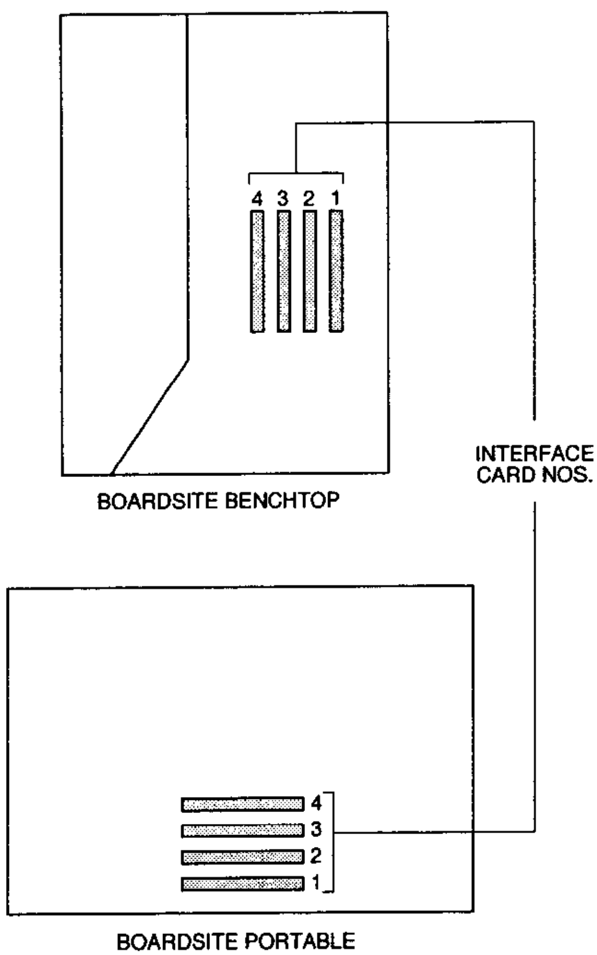
BoardSite also uses BD0-BD7 to determine the current limits of the programmable power supplies for non-isolated programming (see the preceding section, "Isolated Programming and Non-isolated Programming"). If BoardSite detects only one board, it sets the current limit to a lower limit than if it detects eight boards. BoardSite calculates the actual current limit from the Board Current Limits and Interface Adapter Current Offsets parameters in the board information form, in conjunction with the BD0-BD7 count.

Operation

BD0-BD7 are pulled up to a switched +5V in the system through 270kΩ resistors. There are no voltages present on the interface connector until a circuit board operation is invoked. When a circuit board operation is invoked, BoardSite activates the switched +5V pull-up to perform the board detect operation. After boards are detected, the +5V pull-up is switched off.

To signify the presence of an installed circuit board, design your interface adapter to connect the appropriate board detect line to ground when you insert a circuit board. There are three different ways to implement this. See Figure 5-27 and the following section.

Figure 5-27
Interface Boards and Board Detect Lines



Interface Card No.	Board Detect Number	Board No. Isolated	Board No. Non-isolated
1	BD0	1	1
	BD1		2
	BD2		3
	BD3		4
	BD4		5
	BD5		6
	BD6		7
	BD7		8
2	BD0	2	9
	BD1		10
	BD2		11
	BD3		12
	BD4		13
	BD5		14
	BD6		15
	BD7		16
3	BD0	3	17
	BD1		18
	BD2		19
	BD3		20
	BD4		21
	BD5		22
	BD6		23
	BD7		24
4	BD0	4	25
	BD1		26
	BD2		27
	BD3		28
	BD4		29
	BD5		30
	BD6		31
	BD7		32

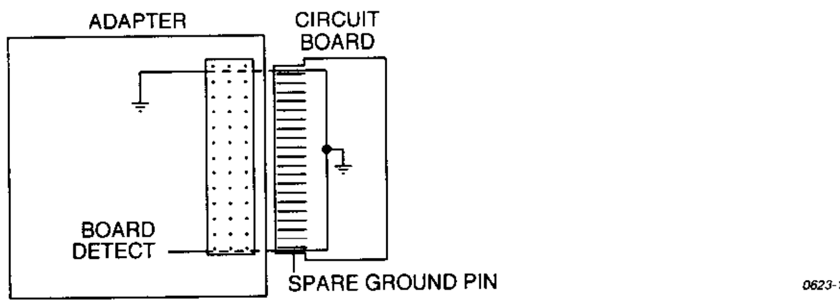
0622-1

Implementation of Board Detect

Method 1: Spare Ground Pin on Board Connector

This method requires that your circuit board have at least two ground pins on its connector. Connect every ground except the one for Board Detect to the BoardSite system ground. Connect the Board Detect ground pin to the appropriate Board Detect line. When the board is inserted into the interface adapter, the Board Detect ground pin is grounded through the electrical path of the board, and BoardSite detects the board. See Figure 5-28.

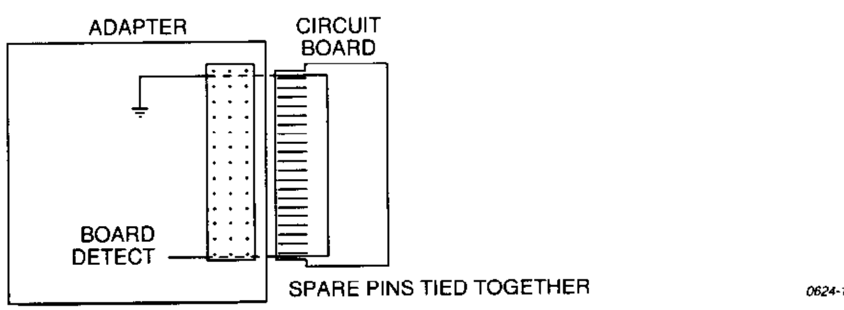
Figure 5-28
Board Detect, Method 1



Method 2: Unused Pins on Board Connector

This method requires that your circuit board has at least two unused pins on its connector, and that these pins are electrically connected. The pins do not have to be connected to circuit board ground. Connect one of the unused pins to the system ground and the other to a Board Detect line. See Figure 5-29.

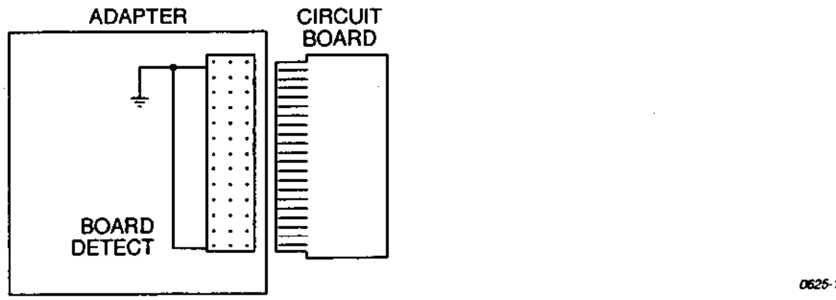
Figure 5-29
Board Detect, Method 2



Method 3: No Pins Available on Board Connector

If your board connector has no spare pins available, there are a few alternatives. One method is to hard-wire the Board Detect lines to ground on the interface adapter. This means that BoardSite always detects the boards, whether they are installed or not. This method is recommended only as a last resort. See Figure 30-30.

Figure 5-30
Board Detect, Method 3



If you plan to program only one board at a time, the Board Detect line for that board can be hard-wired to ground on the interface adapter. This means that BoardSite always detects the board, whether it is installed or not. This shouldn't be a problem for single-board programming.

If you plan to program more than one board at a time, Method 3 is not recommended, because:

1. If less than a full set of boards is installed, BoardSite won't know it. The programmer will fail the empty board positions whenever it does a verify operation.
2. If you use non-isolated programming, and you install only one circuit board in a multi-board adapter, the current limit will be set too high. If there is a catastrophic failure on the board, the excess current may cause severe damage.

If your board has no pins available for Board Detect, use a manual or mechanical detection method. One option requires the operator to set a switch on the interface adapter for each board. This switch connects the appropriate Board Detect to ground. Another option is a microswitch on the interface adapter that connects the appropriate Board Detect to ground when a board is inserted into the adapter.

Note: Power is not available to the circuit boards when BoardSite is reading BD0-BD7. Therefore, you cannot use active circuits to drive BD0-BD7.

Board Enable Lines (BE0-BE7)

Purpose

As described in the section, "Isolated Programming and Non-isolated Programming," you use the Board Enable lines (BE0-BE7) to implement non-isolated programming. Each BoardSite interface board has eight Board Enable lines (BE0-BE7) that control up to eight circuit boards per interface board. You can selectively enable or disable individual circuit boards by gating BE0-BE7 with control signals. All eight boards receive address, data, and power lines in parallel, and BE0-BE7 select which board is active.

Note: In either isolated mode or non-isolated mode, BoardSite programs all boards in parallel. However, in non-isolated mode, boards are verified individually.

Operation

For operations such as copy or verify, BoardSite first determines if an interface adapter is installed, using ADAP0-ADAP1. Next, the system reads the Adapter ID lines ID0-ID7 to determine if the adapter is correct. Finally, BoardSite uses BD0-BD7 to determine how many boards are inserted and where each board is inserted.

For example, assume that the BoardSite 4100 (single interface card) has an eight-slot interface adapter for programming eight non-isolated boards, and that boards are inserted in slots 3, 6, and 8. BoardSite reads the Board Detect lines to determine that only three boards are inserted. When BoardSite starts programming, it drives BE2, BE5, and BE7 (primitive enable_all_boards) active so that all boards program in parallel. To verify board 3, BoardSite drives BE2 active, and BE5 and BE7 inactive (primitive enable_first_board). To verify the next board, BoardSite drives BE5 active, and BE2 and BE7 inactive (primitive enable_next_board). To verify the last board, BoardSite drives BE7 active, and BE2 and BE5 inactive (primitive enable_next_board).

You select the programming mode (isolated or non-isolated), and define the active state of the board enable lines in the board information form. For more information, see the section, "Edit," in Chapter 6, "Advanced Operation."

Implementation of Board Enables

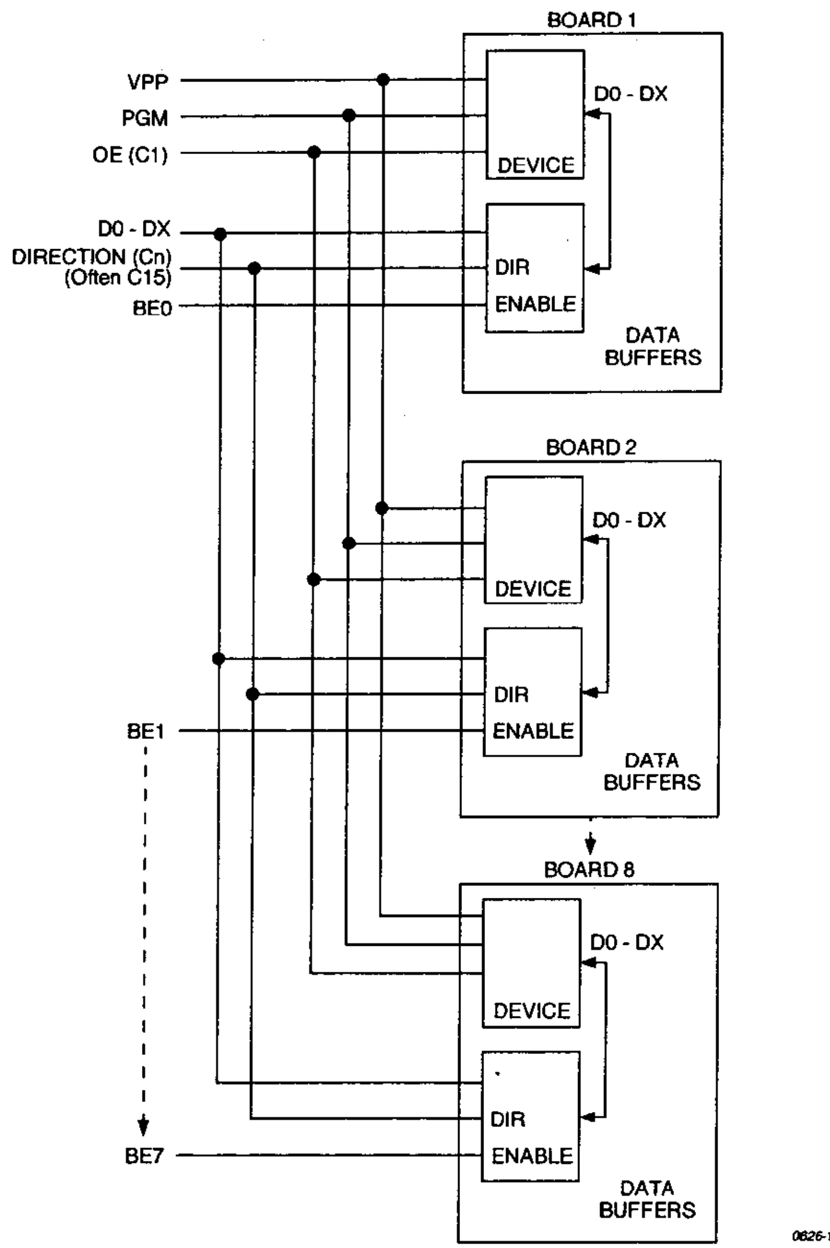
You can use several methods to implement Board Enable. This section contains several different Board Enable methods, and discusses the application of each method.

In general, BE0-BE7 enable individual boards by controlling logic on each board, or by gating other lines going to the boards. These other lines could be control lines, PCE lines, PGM lines, or VPP multiplexer circuits.

Method 1: Externally Controlled Bidirectional Buffers

This method requires circuit boards with externally controlled bidirectional buffers, such as 74LS245s, on the data lines. This is the most straightforward method for non-isolated programming. This method is also usually independent of the memory device types (Type 1, Type 2, etc.) used on the board. Simply connect BE0-BE7 to the enable inputs of the buffers. Disabling the 74LS245s prevents data from passing into or out of the board. See Figure 5-31.

Figure 5-31
Externally Controlled Buffers



If the buffers are not already on the circuit boards, you can place them on the interface adapter, with each board's buffer group controlled by one of the board enable lines.

Method 2: Address Decoder

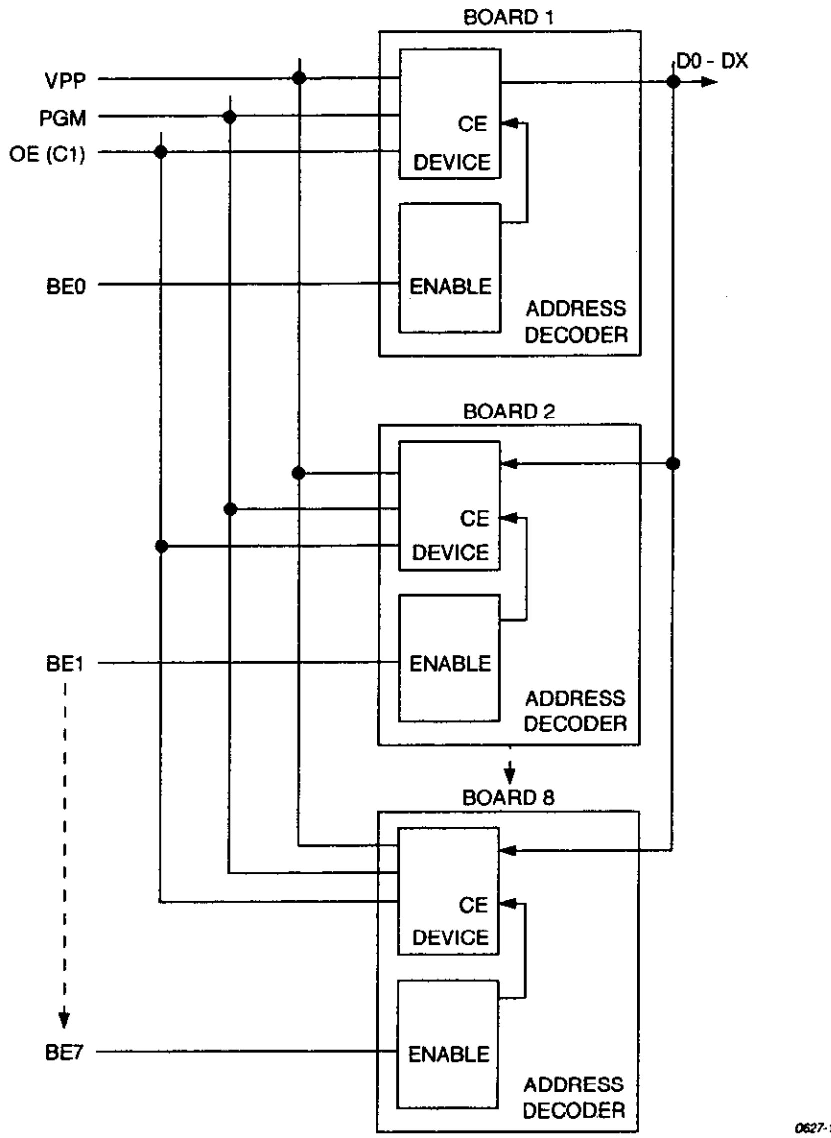
Use this method for boards that have an externally-controlled address decoder, such as a 74LS138, to decode the chip enable (CE) lines for the memory devices. The address decoder has an enable input that either drives all CE outputs inactive, or drives the decoded CE output active. The exact implementation of the address decoder method depends on the memory devices types being programmed. See the preceding section, "Design Rules for Programmable Devices," for more information on device types.

There are four basic device types:

1. Type 1 devices have dedicated PGM, CE, OE, and VPP pins. Examples are 2764, 27128, and 27010.
2. Type 2 devices have a multiplexed PGM/CE pin and separate OE and VPP pins. Examples are 2716 and 27256.
3. Type 3 devices have a multiplexed PGM/CE pin and a multiplexed VPP/OE pin. Examples are 2732 and 27512.
4. Type 4 devices use a pulsed VPP pin for programming. Examples are 57C49 and 8748.

For non-isolated programming, Type 1 devices are the easiest to interface to BoardSite. Connect the Board Enable lines to the enable inputs on the CE address decoders. When the address decoders are disabled, all outputs are disabled. When they are enabled, only the decoded output is active and the others are inactive. Thus, BoardSite can enable and disable each board by enabling or disabling the address decoders on each board. See Figure 5-32.

Figure 5-32
Address Decoder for CE, Type 1 Devices

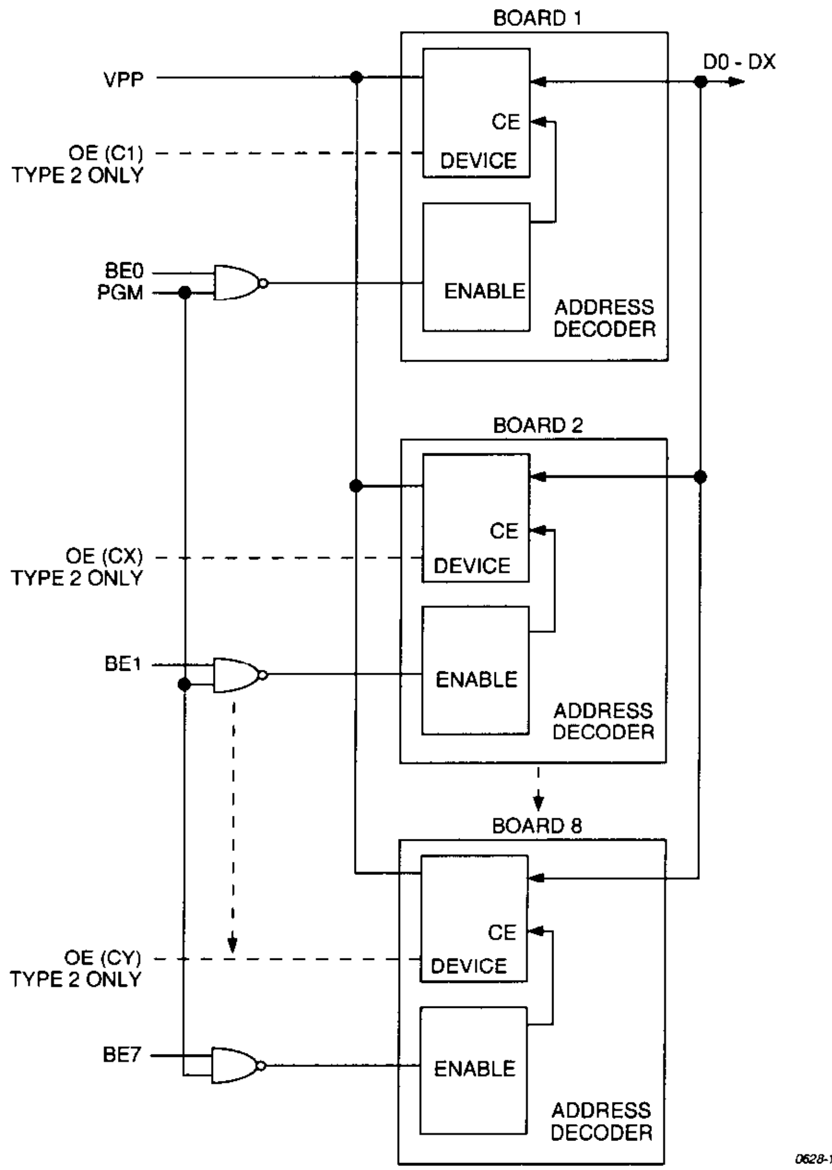


The address decoder method with non-isolated programming is slightly more complicated with Type 2 and 3 devices, because they have a multiplexed PGM/CE input. The address decoder's enable line must be gated by BoardSite's PGM output to drive the PGM/CE with an accurately timed programming pulse. If the address decoder has two identical enable inputs, the PGM line can be connected directly to one of these inputs. The address decoder is enabled only when both the PGM line and Board Enable are active.

If the address decoder has a gated enable line (as the 74LS138 does), gate PGM with Board Enable in the decoder. If the address decoder doesn't have a gated enable, gate the signals on the interface adapter. See Figure 5-33.

Note: Type 2 devices have special requirements for the OE and VPP pins. See the preceding section, "Design Rules for Programmable Devices."

Figure 5-33
Address Decoder for CE, Type 2 and Type 3 Devices



Method 3: No Address Decoder or Bidirectional Buffers

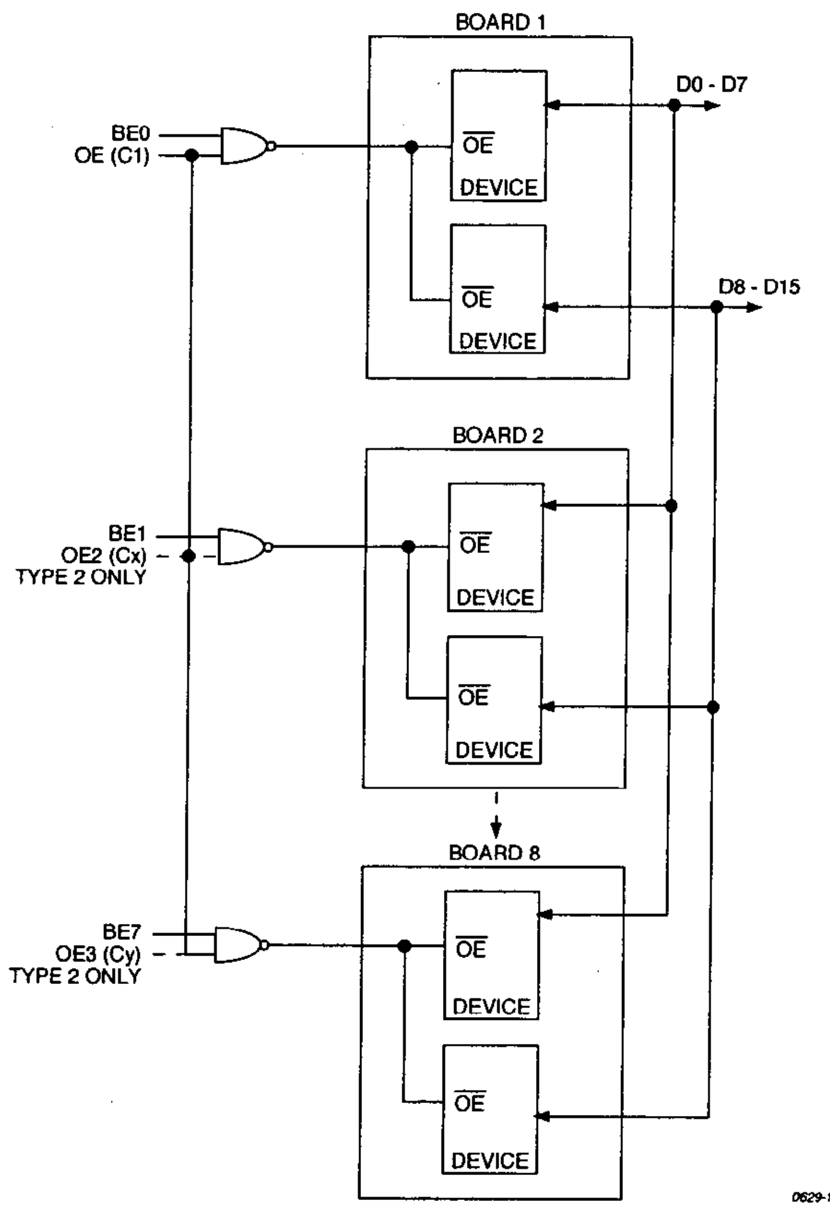
Use this method when the board doesn't have either bidirectional buffers or an address decoder. Also, use this method with Type 4 devices, and in some cases, Type 2 devices.

For non-isolated programming with Type 1, 2 or 3 devices, gate the Board Enable signal with the OE or the CE signal. With BoardSite, you can specify the active level of all control and Board Enable lines, which may reduce the number of external gates.

Type 1 devices can use either OE or CE to disable the board, but the OE line is recommended because all memory devices in a group are usually connected in parallel to OE. This allows BoardSite to read the devices only if OE and Board Enable are both active.

Drive the CE inputs from the PCE lines or by an address decoder on the interface adapter. Connect PGM to all the PGM inputs on all boards, and connect VPP1 to all VPP inputs on all boards. See Figure 5-34.

Figure 5-34
No Address Decoders or Bidirectional Buffers, Type 1 and Type 2 Devices

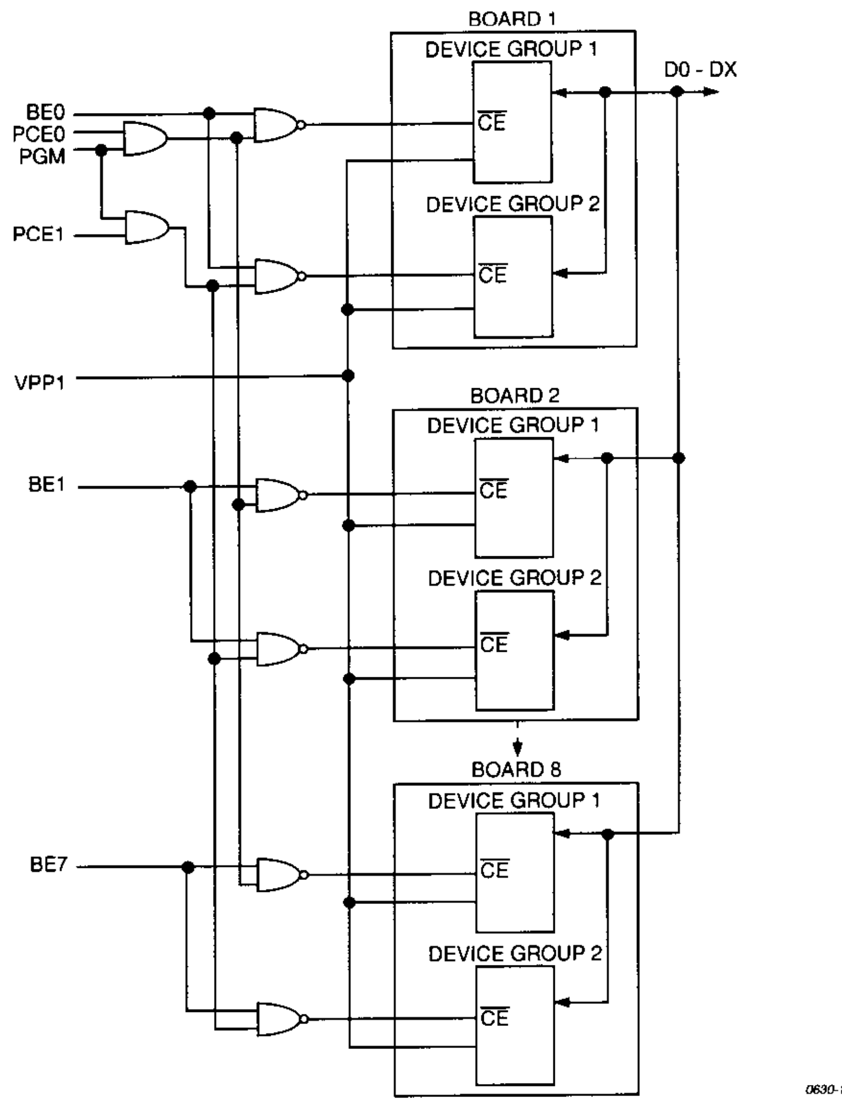


For Type 2 devices, you must gate Board Enable with OE for each device group, because this is the only way to control the devices when the high programming voltage is present on VPP. This allows BoardSite to read the devices only when OE and Board Enable are both active. Drive the CE inputs from the PCE lines gated with PGM, or from a PGM-gated address decoder on the interface adapter.

For Type 3 devices, you must disable the boards with the device CE signal gated with Board Enable. This is because OE is the high-voltage VPP input during programming.

Figure 5-35 shows an example of a board containing two Type 3 device groups. The PCE signals enable the correct device group, and the Board Enable signals enable the correct board. PGM is gated with PCE to deliver the programming pulse to the correct device group.

Figure 5-35
Board with Two Type 3 Device Groups



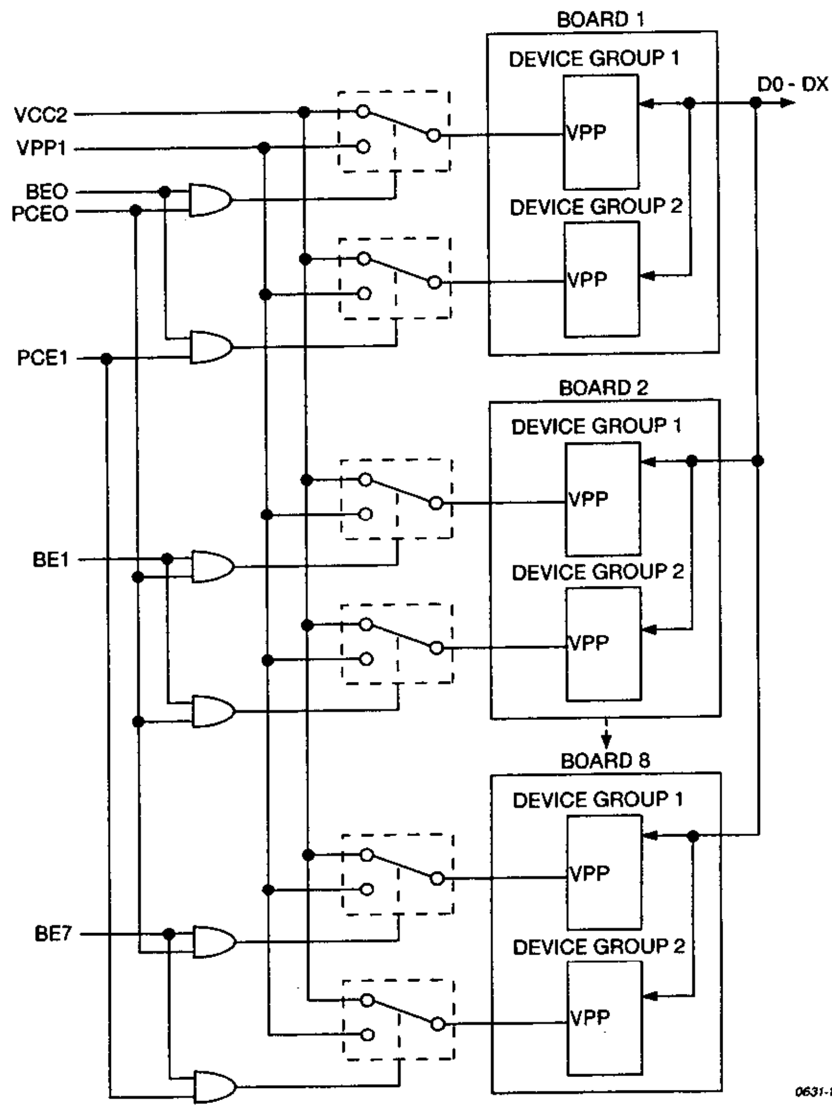
Type 4 devices require special considerations for non-isolated programming. Because Type 4 have a multiplexed CS/VPP pin, you don't have a CS line available to disable devices on other boards during a verify operation.

For boards containing these devices, you must design a VPP multiplexer circuit on the interface adapter. The multiplexer applies the VPP pulse only to those devices currently being programmed, and allows BoardSite to select individual boards for the verify operation.

The Board Enable lines control analog switches, analog multiplexers, or transistor switches to deliver the VPP pulse to the correct devices and to select the correct devices for verification. For each board, the corresponding Board Enable line controls the analog switch.

You can also gate Board Enable lines with PCE lines to control multiple device groups on several circuit boards. See Figure 5-36.

Figure 5-36
Board with Analog Multiplexers



Programmable Chip Enable Lines (PCE0-PCE15)

Purpose

BoardSite provides up to 16 Programmable Chip Enable lines (PCE0-PCE15), which are physically shared with the high-order Address lines (A16-A31). PCE0-PCE15 simplify the interface for boards that don't have an address decoder, or for boards on which the address decoder is bypassed by connecting the individual CE lines directly to the board connector.

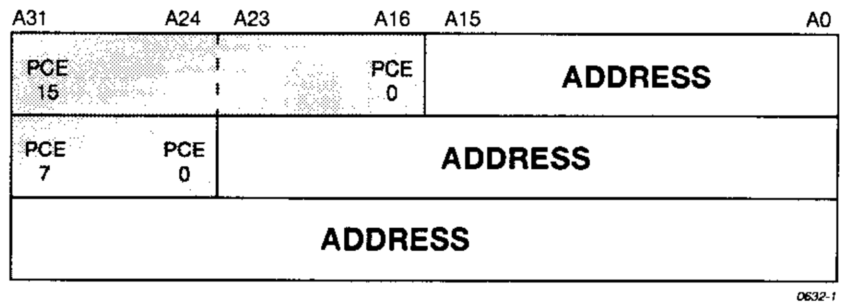
Without PCE0-PCE15, you would have to design an address decoder on the interface adapter. With PCE0-PCE15, you can assign a separate PCE to each device group, and then drive the PCE line active to access the device group.

Operation

Figure 5-37 shows the three available Address/PCE combinations. Select one of these combinations in the board information form. The combinations are:

1. 16 Address Lines with 16 PCE lines
2. 24 Address Lines with 8 PCE lines
3. 32 Address Lines with no PCE lines.

Figure 5-37
Address/PCE Combinations



In the device information form (in the board profile), you can specify a Programmable Chip Enable Mask for each device. For example, assume you use 8 PCE lines and 24 address lines, and the mask you specify for device U1 (in the device information form) is 04HEX. Whenever you attempt an operation on U1, such as copy or verify, the mask pattern for U1 automatically appears on the PCE lines. Because the mask is 04HEX, only PCE2 is driven active and all other PCE lines are driven inactive. See Figure 5-38.

Figure 5-38
PCE Mask Pattern

ADDRESS	31	30	29	28	27	26	25	24
PCE	7	6	5	4	3	2	1	0
BINARY MASK	0	0	0	0	0	1	0	0
HEX MASK	0				4			

PCE MASK: 04

0633-1

For device groups with multiple devices, you can use a single PCE line for the entire group, or use individual PCE lines for each device. If you use individual PCE lines for each device, BoardSite automatically ORs the PCE masks in software to generate the correct PCE signals.

Software Implementation

First, use the Board Profile Editor to select the Number of Programmable Chip Enables and Programmable Chip Enables Active High parameters in the board information form. Next, tell BoardSite to use PCEs by typing **Y** for the Programmable Chip Enables Used By This Group parameter in the device group information form.

Finally, define the masks for each device in the device information forms. For more information, see the section, "Edit," in Chapter 6, "Advanced Operation." The board information form should look like the screen in Figure 5-39.

Figure 5-39
Selecting PCE Parameters in the Board Profile

```
BoardSite Board Profile Editor                       Fri 01-Jun-90 -  1:24:00
Editing:"007-9801-001"
Board Information                           ↑ ↓        Scroll Up, Down
  Algorithm: INTEL 27C256                   Shift-Tab  Previous Field
    Devices: U1 U2                          ↵          Next Field
    Devices: U3 U4                          PgUp PgDn  Previous,Next Page
    Devices: U5 U6                          Alt-b      Save this Form
    Devices: U7 U8                          Esc        Exit Form, Discard Changes
                                            F10        Context Sensitive Help
                                            Press any other key to Edit Field
 Board Information Form
                      Address Bus Width: 16
     Number of Programmable Chip Enables: 8 Programmable Chip Enables
                      Address Increment:    1
                    Interface Card Used: Y Y Y Y
                    Master Board Number:  1
                        Isolated Boards: Y
                      Adapter I.D. Used: Y
                           Adapter I.D.: 55
                        Clock Frequency: 1
              Board Enables Active High: N
                   PGM Line Active High: N
  Programmable Chip Enables Active High: N
< Press ↑ and ↓ to View Form, any other key to Edit Field >
```

Use the Sequence Editor to enable the PCEs (using the pce_enable primitive in the seq_powerup sequence) and then set PCEs active or inactive at the correct times by using the pce_set primitive. For complete information on the Sequence Editor, see Chapter 7, "Sequence Editor Reference."

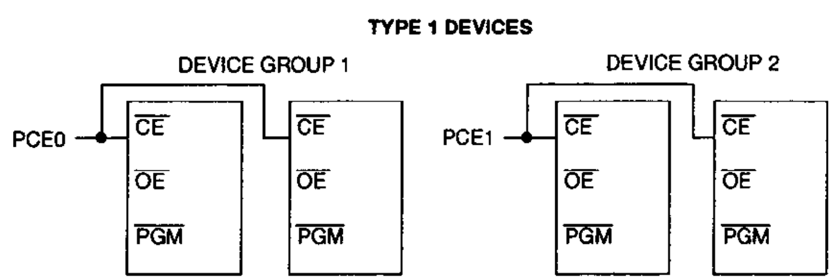
Implementation of Programmable Chip Enables

There are three methods to implement PCE. The method you use depends on the device types on your board.

Method 1: Devices with Independent CE and PGM

For devices with independent CE and PGM pins (Type 1 devices), connect the PCE lines directly to the CE inputs of the devices. For devices within the same device group, connect all the CE pins to the same PCE line. Connect the device OE pins to a common control line, and connect the PGM pins to the BoardSite PGM line. See Figure 5-40.

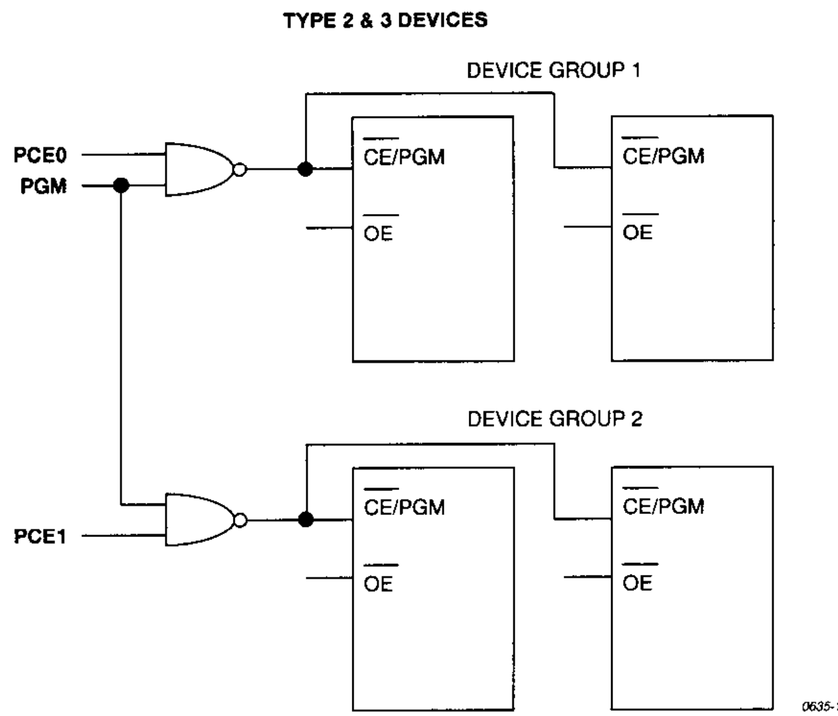
Figure 5-40
Programmable Chip Enables, Method 1



Method 2: Devices with multiplexed CE/PGM

For devices with multiplexed CE and PGM pins (Type 2 and Type 3 devices), do not connect the PCE lines directly to the CE inputs of the devices. Instead, gate the PCE lines with PGM on the interface adapter. As in method 1, device groups with multiple devices can share the same PCE line (after gating that PCE with PGM), because all devices in the group will be active at the same time. See Figure 5-41.

Figure 5-41
Programmable Chip Enables, Method 2



Newer device programming algorithms require very short and precise programming pulses on the CE inputs, and these precision pulses can be guaranteed only by using the PGM line from BoardSite. The PCE lines alone cannot guarantee precision programming pulses, so gating the PCE lines with PGM is mandatory.

One exception to the gating rule is when the board contains only a single device group. For this design, the PCE lines are not required. Simply connect PGM to all the CEs for the device group.

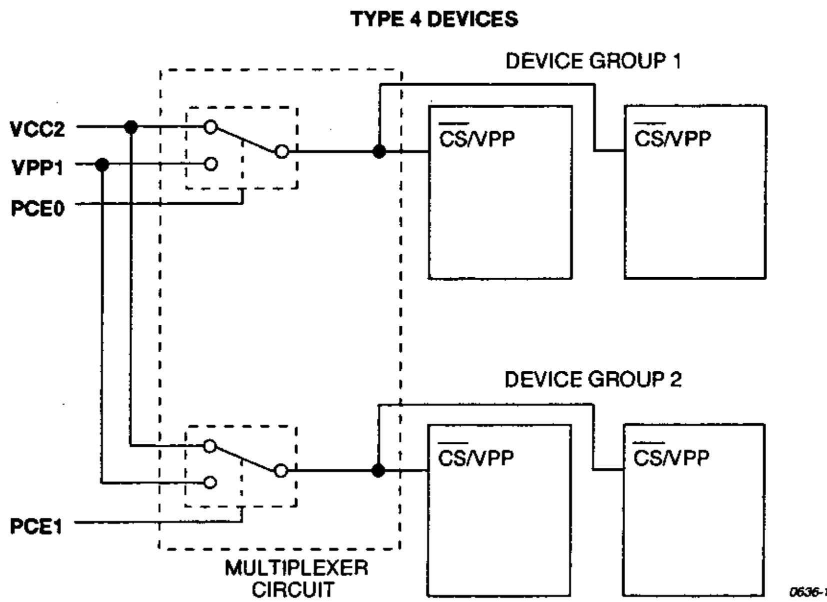
Method 3: Devices with Pulsed VPPs

For non-isolated programming of Type 4 devices with multiplexed CS/VPP lines, see Figure 5-20 in the previous section to learn how to design the interface.

For isolated programming of Type 4 devices with multiple device groups, design a VPP multiplexer on the interface adapter. This method applies the VPP pulse only to the device group currently being programmed, and allows BoardSite to select the correct board for the verify operation.

In this method, the PCE lines control analog switches, analog multiplexers, or transistor switches to deliver the VPP pulse to the correct devices and to select the correct devices for verification. For each device group, BoardSite applies a specific PCE mask pattern to the multiplexer circuit, enabling the correct switches. See Figure 5-42.

Figure 5-42
Programmable Chip Enables, Method 3



LED Lines (LED0-LED7)

Purpose

BoardSite provides 8 LED lines (LED0-LED7) on each interface board. You can use these lines to drive pass/fail LEDs on the interface adapter. Instead of looking at the PC screen, the BoardSite operator can use these pass/fail indicators to quickly determine the status of the boards. You can select the polarity of LED0-LED7 and select whether the lines are active on a pass condition or active on a fail condition.

Operation

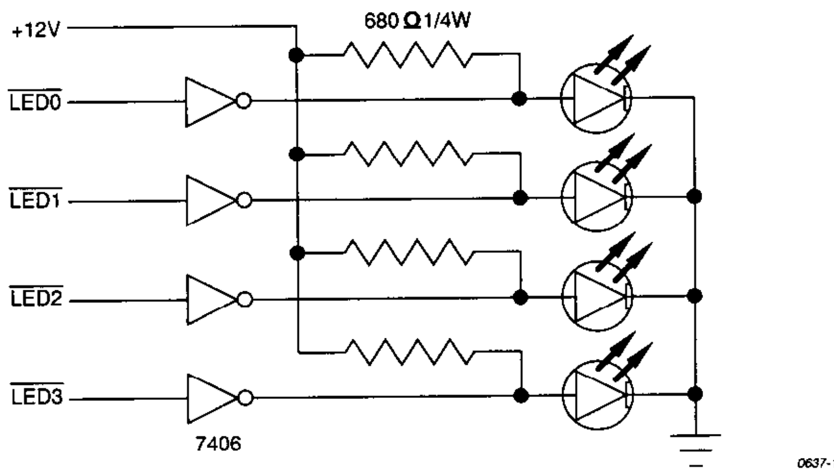
After BoardSite completes a copy, verify, or test operation, it runs an LED routine to drive the appropriate LED lines. If you use the Board Profile Editor to set the LEDs on for Failures parameter to **Y**, BoardSite drives the LED lines active for boards that failed the operation. If you set the LEDs on for Failures parameter to **N**, BoardSite drives the LED lines active for boards that passed the operation.

The LEDs remain on while you scroll through the BoardSite status or failure screens. If you start another operation (or press **Esc** to stop the current operation), LED0-LED7 will be driven inactive until the completion of the next operation.

Implementation

LED0-LED7 are TTL-compatible control lines for LED drivers. Do not use LED0-LED7 to directly drive LEDs. Always use an LED driver or a transistor to drive the LEDs. Design your interface adapter to use an LED power source that is not used by the boards being programmed. For example, use BoardSite's +12V power supply output (pin D41 on the interface connector) if it is not used by the board. See Figure 5-43.

Figure 5-43
LED Lines



You can also use VPP2 or VCC2 if they are not used by the board. If you cannot isolate a supply exclusively for the LEDs, remember that the supply voltage you use will be present on your memory board whenever the LEDs are active. In this case, removing or inserting boards while LEDs are on may be a problem.

Software Implementation

In the Board Profile Editor, move to the board information form. Set the LEDs on for Failures and LEDs Active High parameters to your specifications. For more information, see the section, "Edit," in Chapter 6, "Advanced Operation."

In the sequence file, use the LED primitives led_enable and led_disable to enable and disable the entire LED bus (LED0-LED7). For more information, see Chapter 7, "Sequence Editor Reference."

Multiplexed Address and Data

Operation

BoardSite provides separate ports for address and data. The sequence file controls these ports. This control includes enabling and disabling the ports, writing address and data values, and reading the data lines.

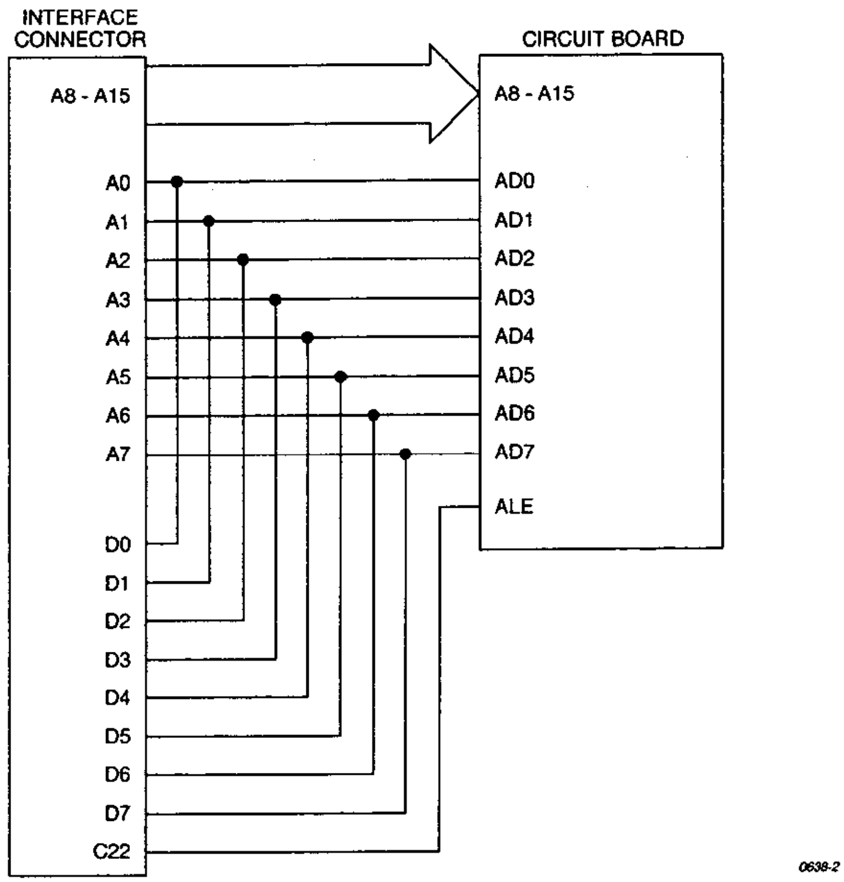
With BoardSite's control capability, you can easily implement a multiplexed address and data bus on the interface adapter.

Implementation of Multiplexed Address and Data

To implement a multiplexed address and data bus, simply connect the appropriate address and data lines in parallel on the interface adapter, and then route the lines to the circuit board connector as a single bus. Connect only those lines actually multiplexed.

For example, to create an 8-bit multiplexed bus, connect D0-D7 to A0-A7, and route the lines directly to the circuit board connector. See Figure 5-44.

Figure 5-44
Multiplexed Address and Data



Software Implementation

Modify the sequence file for the device type(s) you are using. The default sequence file assumes a non-multiplexed address and data bus, and you can easily change this using the Sequence Editor. A typical change involves modifying the SEQ_output_address and SEQ_output_data sequences.

For a detailed description of how to make these sequence file changes, see the section, "How to Modify a Sequence File," in Chapter 7, "Sequence Editor Reference."

Programmable Power Supply Outputs

Purpose

BoardSite provides five programmable power supply outputs on each interface board. These supplies power both the programmable circuit board and the interface adapter.

Supply	Supply Capability (Interface Board only)
VCC1	0 to 7V @ 6A
VCC2	0 to 7V @ 6A
VPP1	0 to 25V @ 2A
VPP2	0 to 25V @ 2A
VNEG	0 to -8V @ .25A (0 to -19.5V @ 0.5A with Expansion Power Supply)

Supply	Standard Configuration	Expansion Power Supply Configuration
VCC (Combined VCC1 and VCC2 current)	6A	24A
VPP (Combined VPP1 and VPP2 current)	2A	8A
VNEG	.25A	2A

Each supply has built-in overvoltage, undervoltage, and programmable overcurrent detection. Each supply also has remote-sensing capability. This section describes power supply operation and design rules that you need to follow. For more information on connector pin assignments for the power supply outputs, see the following section, "Designing the Interface Adapter."

Operation

Each power supply is programmable over its full voltage range using Sequence Editor primitives. Each power supply is also programmable over its full overcurrent detection range using the board information form. Table 5-3 lists the resolution and accuracy (both voltage and overcurrent detect) for each supply.

Table 5-3
Resolution and Accuracy

	Voltage		Overcurrent detect	
Supply	Resolution	Accuracy	Resolution	Accuracy
VCC1	0.01V	0.1V	0.01A	0.2A
VCC2	0.01V	0.1V	0.01A	0.2A
VPP1	0.01V	0.2V	0.01A	0.1A
VPP2	0.01V	0.2V	0.01A	0.1A
VNEG	0.01V	0.3V	0.01A	0.05A

The VPP power supply slew rate is also adjustable. There are two options, Fast and Slow, in the board information form. Unless the memory devices require the slower slew rate, you usually select the Fast option.

Each BoardSite power supply uses a two-stage regulator design. The first stage is called the pre-regulator. It is a programmable switching regulator that provides the basic output voltage. The second stage is the actual output amplifier that delivers the power to the interface connector. These two stages can be controlled independently.

BoardSite usually sets all pre-regulators to some predefined levels, and then enables the output amplifiers. This allows BoardSite to apply power from several supplies simultaneously, or to apply power in a timed sequence, depending on the memory board requirements.

If BoardSite detects an overcurrent, overvoltage, or undervoltage on any supply, it will automatically shut down the supplies in the order defined in the board profile, and then remove all other signals to the memory board.

Implementation

You must follow some very specific design rules when you interface the programmable power supply outputs to your memory board.

Assume the Worst

Always assume that the supplies are working incorrectly until you actually observe the voltage levels and timing on an oscilloscope. Often, one memory board will function correctly, but other boards will malfunction. This is usually caused by power supply oscillations or incorrect power supply sequencing that can easily be corrected by observing the supplies with an oscilloscope. A high-quality digital storage oscilloscope is ideal for this application.

You can use BoardSite's Simulate command to make power supply waveform observations. The optional BoardSite Diagnostic Test Adapter also provides a convenient method for direct waveform observation without an interface adapter. The Diagnostic Test Adapter provides test points for all outputs, and you can observe full programming operations. See your Diagnostic Test Adapter manual for more information.

Telltale signs of power supply oscillations or incorrect power supply sequencing are overcurrent, overvoltage, or undervoltage errors. Other symptoms could be random program or verify errors, especially when using CMOS memory devices.

VCC/VPP Capacitance Requirements

BoardSite power supply outputs are designed to work correctly with large values of capacitance, as you would encounter on a large memory board with many decoupling and filter capacitors in parallel. Because of this design, the supply outputs must have a minimum value of capacitance or the supplies will oscillate.

If your board (or parallel combination of boards) does not present to each interface board the minimum capacitance shown in Table 5-4, you need to add capacitance to the interface adapter. The capacitors should be high-quality and should have a low impedance at high frequency. Ceramic or tantalum capacitors are preferable. Connect the capacitors between each supply and ground as close to the memory board connectors as possible. This compensates for the inductance of the traces on the interface adapter. If you use a tantalum capacitor, connect a .01µF to .1µF ceramic capacitor in parallel with the tantalum capacitor to reduce the impedance at high frequencies.

Table 5-4
Minimum and Maximum Capacitance

	——Per BoardSite Interface Board——	
Supply	**Minimum C.**	**Maximum C.**
VCC1, VCC2	47µF	100µF
VPP1, VPP2	2.2µF	10µF
VNEG	1µF	10µF

After you have provided the correct capacitance, load the interface adapter with programmable circuit boards. Use the Copy command to start a programming operation, and then look at the power up and power down waveforms for each supply. There should be little or no overshoot or ringing on the power-up waveform, and no undershoot or ringing on the power-down waveform.

Note: You may have to adjust the Board Current Limits parameters in the board information form to compensate for the power-up current required to charge the capacitors (I = C x dv/dt). For more information on using the Board Profile Editor, see the section, "Edit," in Chapter 6, "Advanced Operation."

Use an oscilloscope (preferably a digital storage oscilloscope) to observe VCC and VPP power supply waveforms. Connect the oscilloscope probe tip to VCC and VPP at the memory device pins. Use a short ground clip to prevent waveform distortion. Also, check for power supply oscillation during a programming cycle. Finally, measure the actual voltages to verify compliance with the device data sheet specifications.

VCC/VPP Spikes

Another common cause of memory board failures are voltage transients (spikes), especially on VPP. These spikes typically occur during a programming cycle, and are usually caused by excessive inductance in the VPP line. The spikes are typically very fast (less than 100ns pulse width) and can be up to a few volts in amplitude. The spikes may actually exceed the data sheet absolute maximum specifications for a very short period of time.

Observe these spikes by connecting the oscilloscope probe tip to the memory device VCC or VPP pins, and then triggering the scope on the CE or PGM pin. The spikes on the supply will correspond to the leading and trailing edges of the programming pulse. If the spikes are large, you can usually reduce them by placing larger decoupling capacitors on the interface adapter (see the preceding section), and by adding ceramic capacitors (.01µF to .1µF) as close to the device supply and ground pins as possible.

Power Supply Sequencing

Power supply sequencing problems are more difficult to isolate. Power supply sequencing problems happen when the power supplies and/or digital lines are not powered up or powered down in the correct order. Power supply sequencing problems are especially acute for CMOS memory devices.

Almost all devices require VCC to power up before VPP. You must strictly follow the device manufacturer's VCC/VPP sequence, because some memory devices can be destroyed by applying VPP power before VCC.

If the device is not destroyed, power supply sequencing problems usually cause overcurrent errors on VPP. Verify the correct VCC/VPP sequence with no memory boards installed, using an oscilloscope and the Simulate command.

Another power supply sequencing problem is SCR latch-up on CMOS memory devices. SCR latch-up happens when digital inputs to a CMOS memory device are driven to a TTL high state before VCC is applied. This problem is usually caused by powering up VCC2 (for logic on the board) before powering up VCC1 for the memory devices. SCR latch-up can result in random programming or verify errors, power supply oscillations or overcurrents, or destruction of the CMOS memory devices.

The best solution for SCR latch-up is to use the simultaneous power-up primitive power_up_supplies to ensure that VCC2 and VCC1 turn on at the same time. For more information, see Chapter 7, "Sequence Editor Reference."

If the memory devices are CMOS and the logic devices are TTL, power up the CMOS devices first (VCC1), and then power up the TTL logic (VCC2), or power them up simultaneously. If there are several memory devices on the same data bus, you may encounter intermittent power supply oscillation on VCC1, because the memory devices are all attempting to drive the data bus until the logic signals driving their OE lines have stabilized. When the TTL logic outputs have stabilized, the data bus contention will be resolved, and the VCC1 oscillation should stop.

If the memory devices are not CMOS, but the logic circuits are, then the logic can be powered up first or at the same time as the memory. NMOS memory devices do not exhibit SCR latch-up, so you don't normally need special power up sequences.

If the memory devices and the logic circuits are both CMOS, they must be powered up simultaneously. Use the power primitive power_up_supplies to guarantee simultaneous power up of all devices.

If simultaneous power up is not possible, then you must place current limiting resistors in all memory device digital lines to limit the current to less than 20mA. This is a last resort, because these resistors may seriously degrade your board's noise margin. The series resistors also use extra board space on the interface adapter.

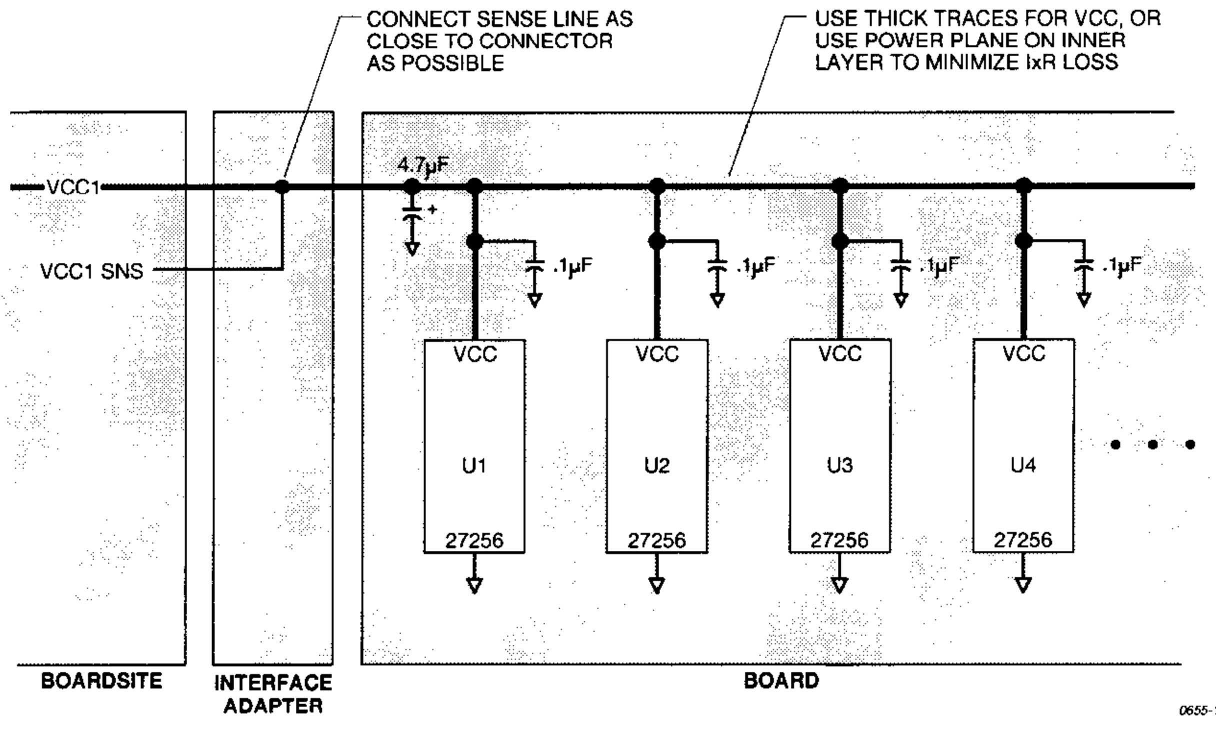
Using Remote Sense Lines

BoardSite's programmable power supplies (VCC1, VCC2, VPP1, VPP2, and VNEG) provide remote sense lines to compensate for resistive losses in circuit board traces. If your boards have heavy current requirements on any of the programmable supplies, you may have to use remote sensing to ensure reliable programming and verification.

The remote sense lines have the letters SNS after the power supply name, For example, VCC1 SNS is the remote sense line for VCC1. If you don't need remote sensing in your design, then don't connect anything to the VCC1 SNS, VCC2 SNS, VPP1 SNS, VPP2 SNS, and VNEG SNS pins on the interface connectors.

If you do require remote sensing, connect the sense line to its corresponding power supply voltage as close as possible to the circuit board connector. The sense lines carry almost no current, so the conductor size is not critical. However, the conductor should not be smaller than a 0.010" circuit board trace or 30 AWG wire. See Figure 5-45.

Figure 5-45
Using a Remote Sense Line for VCC1



Because the sense lines are connected to a high-gain differential amplifier, they must be isolated from circuit board noise. On the interface adapter, use a shielded cable or a twisted pair, and ground the shield or return only at the BoardSite end.

On your circuit board, isolate the sense line trace to prevent inductive or capacitive coupling to adjacent traces. Connect a ceramic capacitor from the sense line to ground, at the circuit board connector. You may have to determine the value of the capacitor empirically. Typical values are from .01µF to .47µF.

If the power supply oscillates with the sense line connected, but doesn't oscillate with the sense line disconnected, the problem is usually improper sense line shielding or capacitive coupling to the sense line.

The sense voltage compliance (the amount of resistive drop that can be compensated for) is .5VDC for VCC supplies, and 1V for VPP supplies. The compliance is reduced if the output voltage is near the supply's maximum output voltage.

Digital Outputs

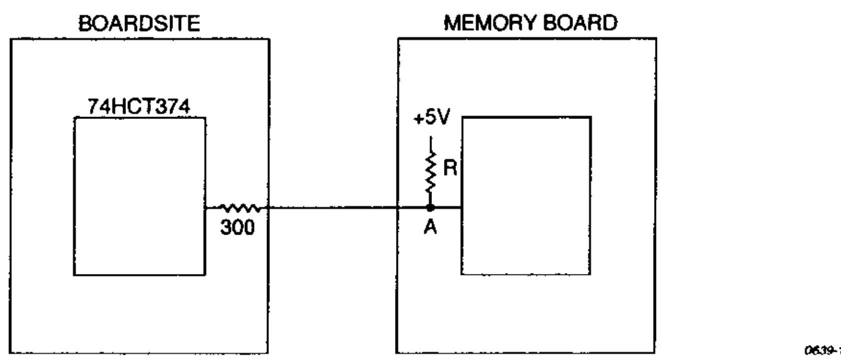
Operation

BoardSite drives all address, data, and control lines with 74HCT374s in series with 300Ω resistors. The resistors protect the 74HCT374s from electrostatic discharge (ESD) damage. Each output also has a clamp diode to ground and a clamp diode to an overvoltage detect circuit to protect the outputs if they are accidentally shorted to an analog voltage on the interface adapter.

Implementation

A common BoardSite interface problem is the inability of digital signals to go to a valid TTL low (V). This is usually caused by a pull-up resistor to +5V somewhere on the interface adapter or on the memory board. A pull-up will cause VIL to increase due to the voltage divider created by the pull-up resistor and the 300Ω protection resistor. See Figure 5-46.

Figure 5-46
Equivalent Circuit of BoardSite and Memory Board

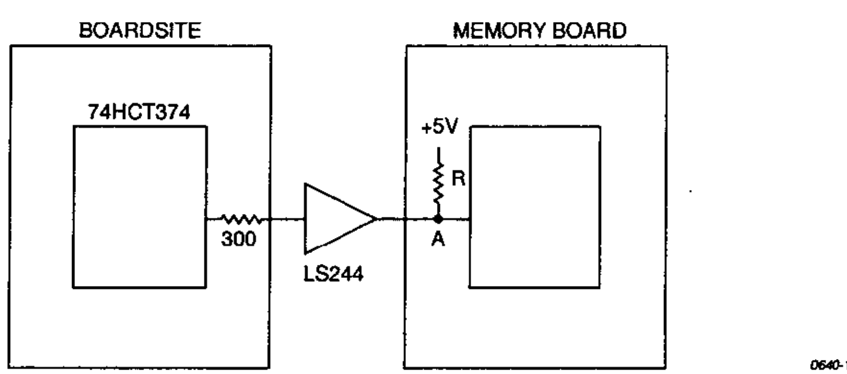


With a pull-up to +5V, and a TTL low at the output of the 74HCT374 driver, VIL is calculated as follows: **VIL = 5V x 300/(R + 300).**

For pull-up resistor values less than approximately 1.9kΩ, a valid TTL low of 0.7V cannot be achieved. For TTL high, the 300Ω resistor has no effect.

If the pull-up resistor value cannot be increased above 1.9kΩ, then you must provide a buffer (a 74LS04 or 74LS244, for example) between the two resistors. See Figure 5-47.

Figure 5-47
Adding a Buffer to Solve VIL Problem



Because you can control the active state of the lines in the board information form, you can use either inverting or non-inverting buffers.

Designing the Interface Adapter

Interface adapter design is usually straightforward. Often, the interface adapter is as simple as point-to-point wiring between the BoardSite interface connector and the memory board connector. You may need additional circuitry on the interface adapter, as described in the preceding sections.

Read the following sections to learn about the BoardSite interface connector and the mechanical layout of the interface adapter.

Interface Connector

The BoardSite interface connectors are 168-pin Bendix low insertion force connectors. The BoardSite interface connector is Bendix part number PC4-168P. The mate to this connector (the one you have to build into your interface adapter) can be either a DB4-168P, which is a right-angle PC-board-mounted connector, or an IO4-168P, which is a straight PC-board-mounted connector.

You can purchase BoardSite connector kits from Data I/O. Use model number BDS-4100CNK to order one mate connector and mounting hardware, or model number BDS-4400CNK to order four mate connectors and mounting hardware.

Figure 5-48 shows a top view of the connector and Table 5-5 shows the pin assignments for the interface connector

WARNING: Limit the current through any individual pin to 2A. Higher current may damage the connector.

Figure 5-48
Interface Connector, Top View

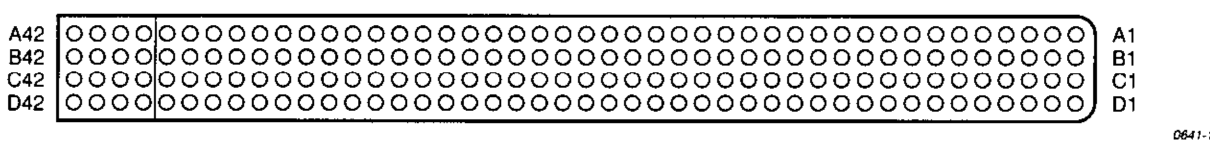


Table 5-5
Pin Assignments for the Interface Connector

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	ADAP0	B1	SPARE	C1	LED6	D1	LED7
A2	LED2	B2	LED3	C2	LED4	D2	LED5
A3	ID6	B3	ID7	C3	LED0	D3	LED1
A4	ID2	B4	ID3	C4	ID4	D4	ID5
A5	BD6	B5	BD7	C5	ID0	D5	ID1
A6	BD2	B6	BD3	C6	BD4	D6	BD5
A7	BE6	B7	BE7	C7	BD0	D7	BD1
A8	BE2	B8	BE3	C8	BE4	D8	BE5
A9	C22	B9	C23	C9	BE0	D9	BE1
A10	C18	B10	C19	C10	C20	D10	C21
A11	C14	B11	C15	C11	C16	D11	C17
A12	C10	B12	C11	C12	C12	D12	C13
A13	C6	B13	C7	C13	C8	D13	C9
A14	C2	B14	C3	C14	C4	D14	C5
A15	D30	B15	D31	C15	C0	D15	C1
A16	D26	B16	D27	C16	D28	D16	D29
A17	D22	B17	D23	C17	D24	D17	D25
A18	D18	B18	D19	C18	D20	D18	D21
A19	D14	B19	D15	C19	D16	D19	D17
A20	D10	B20	D11	C20	D12	D20	D13
A21	D6	B21	D7	C21	D8	D21	D9
A22	D2	B22	D3	C22	D4	D22	D5
A23	A30	B23	A31	C23	D0	D23	D1
A24	A26	B24	A27	C24	A28	D24	A29
A25	A22	B25	A23	C25	A24	D25	A25
A26	A18	B26	A19	C26	A20	D26	A21
A27	A14	B27	A15	C27	A16	D27	A17
A28	A10	B28	A11	C28	A12	D28	A13
A29	A6	B29	A7	C29	A8	D29	A9
A30	A2	B30	A3	C30	A4	D30	A5
A31	A0	B31	A1	C31	SPARE	D31	VPP2 REF
A32	GND	B32	GND	C32	VNEG SNS	D32	VPP1 REF
A33	PGM	B33	GND	C33	GND	D33	VNEG
A34	GND	B34	GND	C34	SPARE	D34	VCC2
A35	XTAL1	B35	GND	C35	GND	D35	VCC2
A36	XTAL0	B36	GND	C36	GND	D36	VCC2
A37	GND	B37	GND	C37	GND	D37	VCC1
A38	CHAS GND	B38	VCC2 SNS	C38	GND	D38	VCC1
A39	VCC1 SNS	B39	VPP1 SNS	C39	SPARE	D39	VCC1
A40	VPP1	B40	GND	C40	SPARE	D40	-12V
A41	GND	B41	VPP2 SNS	C41	SPARE	D41	+12V
A42	VPP2	B42	GND	C42	SPARE	D42	ADAP1

SPARE lines are not connected.

Digital Signal Definitions

All digital signals listed below are TTL-compatible. They are driven with high-speed CMOS devices (74HCT374s) and have 300Ω series resistors and clamp diodes. For more information, see the preceding section, "Digital Outputs." Table 5-6 lists guaranteed specifications for the signals at the interface connector.

Table 5-6
Guaranteed Specifications

Signal	Definition	Guaranteed spec.
VIH	Minimum high level input voltage	2.0V
VIL	Maximum low level input voltage	0.8V
IIN	Maximum input current	1mA
CIN	Maximum input capacitance	15pF
VOH	Minimum high level output voltage	2.4V@IOH=6mA; 4.6V@IOH=20μA
VOL	Maximum low level output voltage	0.8V @ IOL=2mA; 0.4V @ IOL=1mA; 0.1V @ IOL=20μA
IOL	Maximum low level output current	-2mA
IOH	Maximum high level output current	6mA

The following BoardSite digital signal lines are available at the interface connection:

A0-A15	Sixteen low-order address lines, tri-statable in groups A0-A7, and A8-A15.
A16-A31	Sixteen high-order address lines, tri-statable in groups A16-A23, and A24-A31. Can also be used as individual programmable chip enable lines (PCE0-PCE15). Address/PCE combinations are A0-A15 with PCE0-PCE15, A0-A23 with PCE8-PCE15, and A0-A31 with no PCE. The active state of PCE is selectable in the board information form.
D0-D31	Thirty-two bidirectional data lines, tri-statable in groups D0-D7, D8-D15, D16-D23, and D24-D31.
C0-C23	Twenty-four bidirectional control/status lines that are under direct control of the software. Tri-statable in groups C0-C7, C8-C15, and C16-C23.
ID0-ID7	Eight adapter ID lines which are used to create specific IDs for different adapters. Lines are pulled up to a switched +5 volts in the system through 270kΩ resistors. For systems containing four interface boards, only the ID lines on one connector need to be wired.
ADAP0, ADAP1	Two adapter detect lines which must be connected to ground on the interface adapter. Lines are pulled up to a switched +5 volts in the system through 3.3kΩ resistors. For systems containing four interface boards, all the detect lines on all the connectors must be wired.
XTAL0, XTAL1	Two programmable clock lines selectable for frequencies of 1, 2, 4, and 8MHz. XTAL0 is inverted from XTAL1. These lines are tri-statable.
PGM	One programming pulse line originating from a hardware timer. The active state is selectable in the board profile.
BD0-BD7	Eight board detect lines which must be connected to ground on the memory boards. These lines are pulled up to a switched +5V in the system through 270kΩ resistors. For isolated interfaces use only BD0.
BE0-BE7	Eight board enable lines used on non-isolated systems to gate the control lines to individual boards.
LED0-LED7	Eight LED lines used for a pass or fail indication on the interface adapter.

Power Signal Definitions

All power signals listed below specify maximum capability of the supply. The current drawn through each pin of the interface connector must be limited to 2A or less. For higher current, connect several pins in parallel.

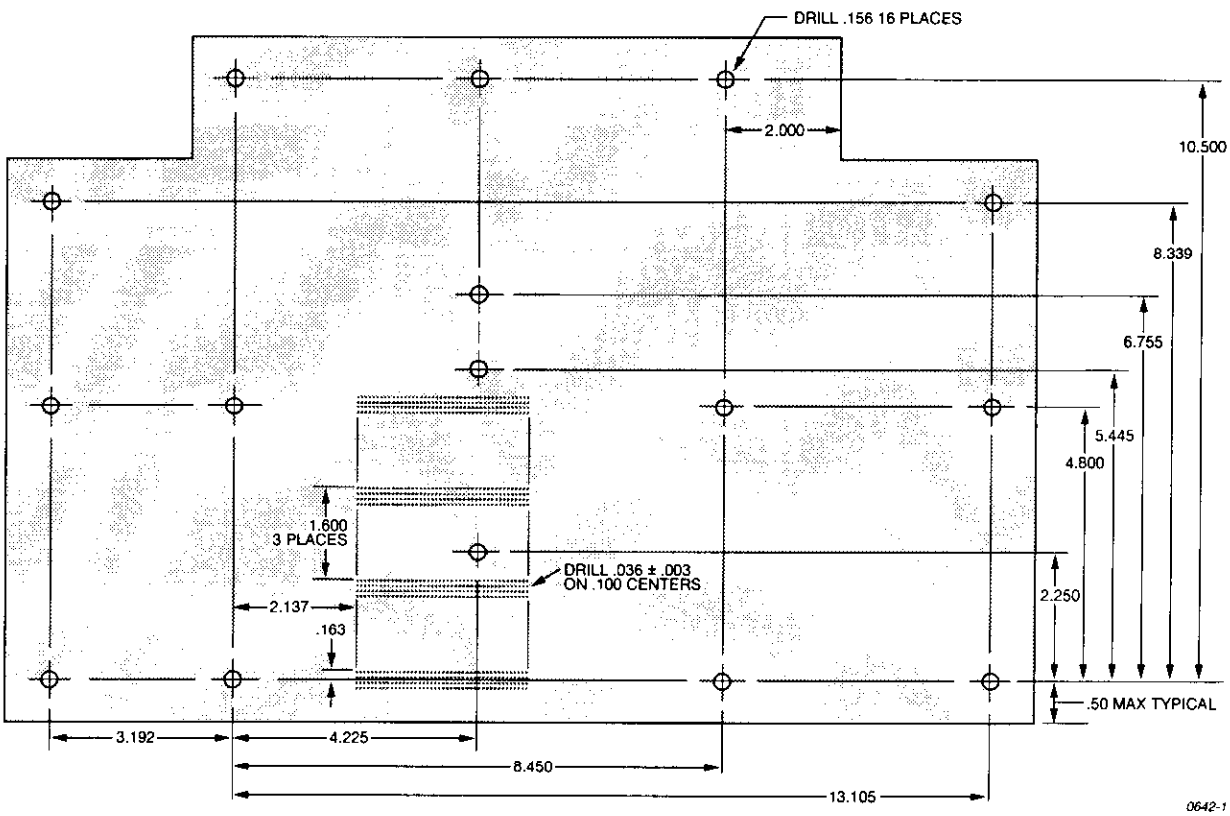
VCC1	Primary VCC supply, used for VCC on the memory devices. Programmable 0-7V at 6A. Combined VCC1 and VCC2 current must be less than 6A.
VCC2	Secondary VCC supply, used for VCC on logic devices. Programmable 0-7V at 6A. Combined VCC1 and VCC2 current must be less than 6A.
VPP1	Primary VPP supply, used for VPP on the memory devices. Programmable 0-25V at 2A. Combined VPP1 and VPP2 current must be less than 2A.
VPP2	Secondary VPP supply. Programmable 0-25V at 2A. Combined VPP1 and VPP2 current must be less than 2A.
VNEG	Negative supply. Programmable 0-8V at .25A.
+12V	General-purpose +12V @ 0.25A maximum.
-12V	General-purpose -12V @ 0.25A maximum.
GND	Ground. Primary ground for all analog and digital circuits on the interface adapter and circuit boards.
VCC1 SNS **VCC2 SNS** **VPP1 SNS** **VPP2 SNS** **VNEG SNS**	Power supply sense line for each programmable supply. Compensates for resistive losses in the circuit board traces from the power supply output amplifier to the circuit board load.
VPP1 REF **PP2 REF**	Precision reference voltages equal to the output of the programmable VPP supply.
CHAS GND	Chassis ground. Use this line to tie the metalwork of an interface adapter to the chassis ground of BoardSite.

Interface Adapter Mechanical Design

Figure 5-49 shows a mechanical drawing of the circuit board for a typical interface adapter. This board fits on either the portable unit or benchtop unit.

The notches in the top two corners of the board are not required if the board is used only on a benchtop unit.

Figure 5-49
Typical Interface Adapter Circuit Board



BoardSite Prototyping Kit

You can order a BoardSite Prototyping Kit from Data I/O. This kit includes a pre-drilled interface adapter circuit board, interface adapter connectors, wire wrap headers, screws, and standoffs.

Use model number BDS-4100PTK to order BoardSite Prototyping Kit for BoardSite 4100.

Use model number BDS-4400PTK to order BoardSite Prototyping Kit for BoardSite 4400.

6 *Advanced Operation*

Introduction

This chapter describes the Simulate, Communications, Display, Edit, Setup, File, and Batch commands. These commands are available in Manager Mode. Commands that are available in all modes (Copy, Verify, Test, Help, and Checkout) are documented in Chapter 3, "Basic Operation."

Here is a brief description of the commands described in this chapter:

Simulate—Use the Simulate command to debug board profiles and sequence files, and to test your interface adapter.

Communications—Use the Communications command to move data files between BoardSite and the host computer.

Display—Use the Display command to display your board data, the system error log, programming statistics, text files, and the system configuration.

Edit—Use the Edit command to create and edit board profiles, sequence files, data files, and text files.

Setup—Use the Setup command to change your system setup.

File—Use the File command to manage your BoardSite files.

Batch—Use the Batch command to create and run BoardSite batch files, which automatically perform a series of tasks.

This chapter is a command reference, organized by command name.

Simulate

Use the Simulate command to debug board profiles, sequence files, and interface adapters. With this command, you can check all critical waveforms and voltages by exercising the BoardSite hardware and software. You don't need to risk damage to a programmable board, because you can run the Simulate command with no boards installed.

To use the Simulate command:

1. Insert your interface adapter into a BoardSite interface connector.
2. Select the Simulate command from the menu bar.
3. In the Board Profile Name pop-up, select the desired board profile. Press [↵].
4. In the Device Names pop-up, select the device group you want to simulate, or select the All Devices on the Board option. Press [↵].
5. In the Simulation Source Options pop-up, select either the Master option or the Disk option. Press [↵].

Note: *You must have a data file to run the Simulate command. If you don't have any data files, and you don't have a master board to copy, you can use the Data Editor to create a new data file, and then fill it with a data pattern. See the topic, "Creating and Editing Data Files," in the section, "Edit," later in this chapter.*

6. If you selected the Disk source option in step 5, select the data file from the Data File Name pop-up. Press [↵].
7. In the Simulation Options pop-up, select the options you want. Follow the instructions that appear in the upper window. When all options are correct, press [Alt] + [B].
8. Press [Alt] + [B] again to begin the simulation.

Use your test equipment (oscilloscope, logic analyzers, and so on) to measure and verify all waveforms.

Note: *Looping occurs only in the programming segment (copy overhead definition code P) of the Copy operation.*

Also, board positions that fail when you run the Simulate command are not disabled, because you usually run this command with no boards installed.

Communications

Use the Communications command to transfer data files between BoardSite and the host computer, to translate files, and to emulate a terminal with your PC. This command contains five different options: Upload, Download, Translate, Parameters, and Transparent. Each option is described below.

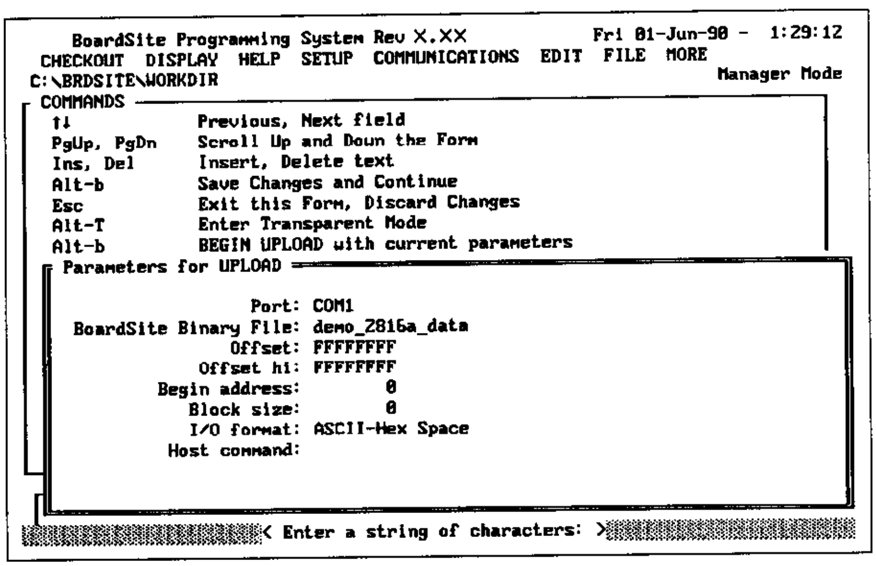
Upload

Use Upload to transmit a BoardSite data file to a host computer. Upload uses BoardSite's data translation formats to format the data for the host computer. For more information on these translation formats, see Appendix B, "Translation Formats."

To use the Upload command:

1. Make sure you set the communications parameters to match the host computer (see the topic "Parameters" in this section), and that you have the correct RS-232 cable connected between your PC and the host.
2. Select the Communications command from the menu bar.
3. In the Communications Options pop-up, select the Upload option. Press ↵ .
4. In the Parameters for Upload pop-up (see Figure 6-1), select options as described below.

Figure 6-1
Parameters for Upload Pop-up



Port—Accept the default port, or change to another port.

BoardSite Binary File —Select the name of the data file to be transmitted.

Offset—Offset is the beginning address of the host's RAM. Type the beginning address of the data to be sent. The address is added to the user memory addresses before the data is sent. Either FFFFFFFF or 00000000 will set the data beginning address to zero.

Offset hi—This parameter sets the high-order address offset for the Extended Tektronix Hexadecimal data translation format.

Begin address—This is the first memory address in the file from which data is read. The default is the beginning address of the file, which is 00000000.

Block size—This is the number of bytes of data read from the file.

I/O format—Select the data translation format to match your host computer.

Host command—Type a host command here if your host requires one to begin the data transfer. BoardSite sends the command as soon as you begin the command. If your host computer does not require a command, press ⏎.

5. When all parameters are correct, press Alt + B to begin transmitting data to the host.

Download

Use Download to receive a data file from a host computer. Download uses BoardSite's data translation formats to format the received data. For more information on these translation formats, see Appendix B.

To use the Download command

1. Make sure you set the communications parameters to match the host computer (see the topic "Parameters" in this section), and that you have the correct RS-232 cable connected between your PC and the host.
2. Select the Communications command from the menu bar.
3. In the Communications Options pop-up, select the Download option. Press ⏎.
4. In the Parameters for Download pop-up, select options as described below. This pop-up is similar to the one in Figure 6-1.

   Port—Accept the default port, or change to another port.

   BoardSite Binary file—Type the name of the file to store the received data, or select an existing data filename.

Note: If you select an existing data filename, BoardSite replaces the data in the file with the data received from the host.

Offset—Offset is the beginning address of the incoming data. The address is subtracted from received addresses. Selecting **FFFFFFFF** causes the first address received to be used as the default. Some of the received data may be lost, depending on the contents of the data.

Offset hi—This parameter sets the high-order address offset for the Extended Tektronix Hexadecimal data translation format.

Begin address—This is the first memory address in the file to which data is written. The default is the beginning address of the file, which is 00000000.

Block size—This is the number of bytes of data to be received.

I/O format—Select the data translation format to match your host computer.

Host command—Type a host command here if your host requires one to begin the data transfer. BoardSite sends the command as soon as you begin the command. If your host computer does not require a command, press [↵].

Fill byte—Fill the data file (before transfer) with a data pattern. The data pattern is usually 00 or FF.

5. When all parameters are correct, press [Alt] + [B] to begin receiving data.

Both the Download and Translate (ASCII to BoardSite Binary file) commands allow you to merge the new data with an existing file. Previous versions of BoardSite always deleted the existing file and then recreated a new file.

Once you press [Alt] + [B], and if you are downloading or translating to an existing file, you will be prompted with the following, as shown in Figure 6-2.

Figure 6-2
Merge Data with Existing File

```
   MODEL 4100 Rev X.XX                                       Wed 16-Oct-91 -  8:17:30
 CHECKOUT  DISPLAY  HELP  SETUP  COMMUNICATIONS  EDIT  FILE  MORE
E:\REV5                                                                Manager Mode
 COMMANDS
  ↑↓             Previous, Next field
  PgUp, PgDn     Scroll Up and Down the Form
  Ins, Del       Insert, Delete text
  Alt-b          Save Changes and Continue
  Esc            Exit this Form, Discard Changes
  Alt-b          BEGIN TRANSLATION with current parameters
   MERGE DATA WITH EXISTING FILE?
   Do you wish to overwrite the file "Missile Nav Data"
   with new data, or do you wish to merge this new data with the file?

   Select Yes if you wish to merge with the existing file,
   or No if you wish to overwrite the file.



    Press "Y" to merge data with existing file:  N

                    [Esc] escape menu  [↵] select item  [F10] help
```

This option allows you to specify to overwrite the existing file with new data or to merge the new data with the existing data file.

If you choose to overwrite the file (default), the existing file is deleted and then recreated with the new data.

If you choose to merge the new data with the existing file, the merge operation writes the data directly to the existing file.

If you plan on building a data file by downloading or translating several files and merging the data, you should perform the first operation with a block size large enough to account for the final desired data file. This will cause the BoardSite software to fill the file with the fill byte value (typically 00 or FF). Then perform the second and subsequent operations by merging with the existing data.

You will probably need to specify the beginning RAM address for the second and subsequent operations unless your input data already has the correct addresses specified within the format so that the data is merged properly.

Translate

Use Translate to convert files. You can convert a BoardSite binary file to an ASCII file, or vice versa.

To use the Translate command:

1. Select the Communications command from the menu bar.
2. In the Communications Options pop-up, select the Translate option. Press ⏎.
3. In the File Conversion Type pop-up, select either Binary to convert a BoardSite binary file to an ASCII file, or select ASCII to convert an ASCII file to a BoardSite binary file. Press ⏎.
4. In the Parameters to File Transfer pop-up, select options as described below.

   BoardSite Binary file (for either BoardSite-to-ASCII or ASCII-to-BoardSite conversions)—Select the BoardSite binary filename, or type a filename.

   DOS ASCII Source file (for ASCII-to-BoardSite conversion only)—Type the name of the ASCII source file.

   Destination DOS file (for BoardSite-to-ASCII conversion only)—Type the name of the ASCII destination file.

   Offset—For ASCII-to-BoardSite conversion, offset is the beginning address of the ASCII data. For BoardSite-to-ASCII conversion, offset is the beginning address of the BoardSite binary file.

   Offset hi—This parameter sets the high-order address offset for the Extended Tektronix Hexadecimal data translation format.

   Begin address—This is the first memory address in the file to which data is sent after conversion. The default is the beginning address of the file, which is 00000000.

   Block size—This is the number of bytes of data to be converted.

   I/O format—Translate uses a data translation format to convert the file. Select the translation format you want to use.

   Fill byte (for ASCII-to-BoardSite conversion only)—Fill the BoardSite data file (before conversion) with a data pattern.
5. When all parameters are correct, press **Alt** + **B** to convert the file. (See step 5 in the previous section, "Download," for a description of the overwrite and merge data options.)

Parameters

Use Parameters to set BoardSite's I/O port parameters to match your host computer's parameters.

To use the Parameters command:

1. Determine the communications parameters of the host computer.
2. Select the Communications command from the menu bar.
3. In the Communications Options pop-up, select the Parameters option. Press ⏎.
4. In the Communications Port pop-up, select the port you want to set parameters for. Press ⏎.

5. In the Port Parameters for COMx pop-up, select options as described below.

   Baud rate—Select the baud rate that matches your host computer system. Select either 300, 1200, 2400, 4800, 9600, or 19200.

   Parity—Select the parity option that matches your host computer system. Select either none, even, odd, mark, or space.

   Data bits—Select the option that matches your host computer system. Select either 5, 6, 7, or 8.

   Stop bits—Select the option that matches your host computer system. Select either 1 or 2.

   XON/XOFF handshaking—Turn software handshaking on or off. Select ON or OFF.

   XON character—Select the option that matches your host computer system. The XON character is usually ASCII character number 11.

   XOFF character—Select the option that matches your host computer system. The XOFF character is usually ASCII character number 13.

   RTS/CTS handshaking—Turn hardware handshaking on or off. Select ON or OFF.

   I/O time-out —Set the period of time that BoardSite waits to receive data. If no data are received within that time, BoardSite stops the operation. Type a time between 0 and 99 seconds. You can disable the I/O time-out by typing 0.

6. When all parameters are correct, press [Alt] + [B] to make the changes.

Transparent

Use Transparent to set your PC to emulate a terminal and communicate with the host. To use the Transparent command:

1. Make sure you set the communications parameters to match the host computer (see the topic "Parameters" in this section) and have the correct RS-232 cable connected between your PC and the host.
2. Select the Communications command from the menu bar.
3. In the Communications Options pop-up, select the Transparent option. Press [↵].
4. In the Communications Port pop-up, select the port that is connected to your host computer. Press [↵].
5. You can now transmit characters to the host by typing on your PC keyboard. Any characters the host sends are displayed on your PC screen. If the host does not echo characters, you will not see the characters you type.
6. Press [Alt] + [T] to end terminal emulation.

Display

Use the Display command to list board data, to list the system error log, to list programming statistics, to list a text file or batch file, and to show the current system configuration.

Display Board Data

1. Select the Display command from the menu bar.
2. In the Display Command Options pop-up, select the Board option. Press ↵.
3. In the Destinations Options pop-up, select Screen to send the list to your PC screen, or select Printer to send the list to the LPT1 port. Press ↵.
4. In the Board Profile Name pop-up, select the board profile that corresponds to the board you want to read and list. Press ↵.
5. In the Device Names pop-up, select an option. Press ↵.
6. In the Board Number pop-up, type the number of the board you want to read and list. Press ↵.
7. Press Alt + B to begin the command.

BoardSite applies power to the board, reads the data, and then sends the data to the screen or printer.

Display Error Log

1. Select the Display command from the menu bar.
2. In the Display Command Options pop-up, select the Error Log option. Press ↵.
3. In the Destinations Options pop-up, select Screen to send the log to your PC screen, or select Printer to send the log to the LPT1 port.
4. Press ↵.

Display Programming Statistics

1. Select the Display command from the menu bar.
2. In the Display Command Options pop-up, select the Statistics option. Press ↵.
3. In the Destinations Options pop-up, select Screen to send the statistics to your PC screen, or select Printer to send the statistics to the LPT1 port.
4. In the Board Profile Name pop-up, select the board profile that you want statistics for.
5. Press ↵.

Display Text File

1. Select the Display command from the menu bar.
2. In the Display Command Options pop-up, select the Text File option. Press ↵.

3. In the Destinations Options pop-up, select Screen to send the list to your PC screen, or select Printer to send the list to the LPT1 port. Press ↵.
4. In the Display Text File pop-up, select the text file you want to list.
5. Press ↵.

Display Batch File

1. Select the Display command from the menu bar.
2. In the Display Command Options pop-up, select the Batch File option. Press ↵.
3. In the Destinations Options pop-up, select Screen to send the list to your PC screen, or select Printer to send the list to the LPT1 port. Press ↵.
4. In the Display Batch File pop-up, select the batch file you want to list.
5. Press ↵.

Display System Configuration

1. Select the Display command from the menu bar.
2. In the Display Command Options pop-up, select the Configuration option. Press ↵.
3. In the Destinations Options pop-up, select Screen to send the configuration to your PC screen, or select Printer to send the configuration to the LPT1 port.
4. Press ↵.

Edit

Use the Edit command to create and edit text files, data files, board profiles, and sequence files. For full documentation on the Sequence Editor, see Chapter 7, "Sequence Editor Reference."

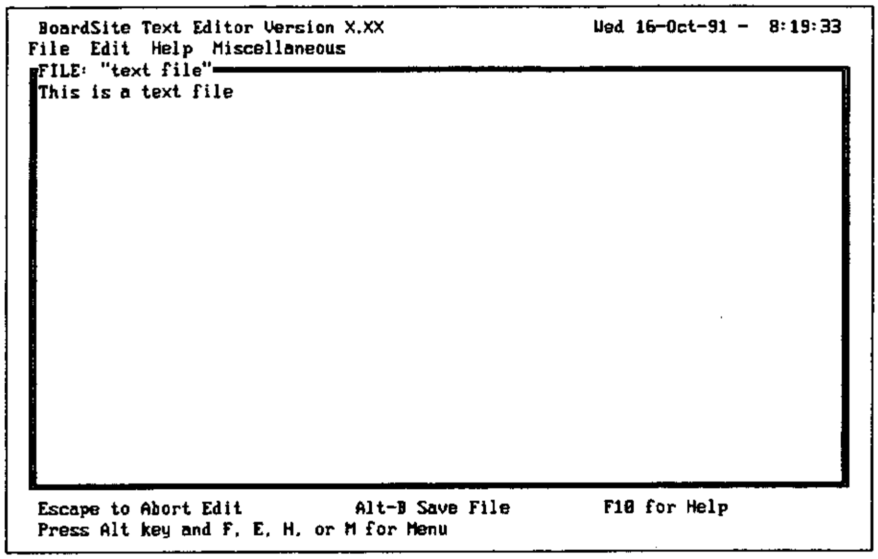
Creating and Editing Text Files

This option allows you to create or edit text files, which can be used for prompts, messages, and warnings for the BoardSite operator.

To use the Text Editor:

1. Select the Edit command from the menu bar.
2. In the Edit Options pop-up, select the Text File option. Press ↵.
3. In the Edit File Name pop-up, select the text file you want to edit, or select the Create New User Text File option to create a new file. Press ↵.
4. If you selected the Create New User Text File option in step 3, type a filename for the new file. Press ↵.
5. In a few seconds, you should the Text Editor screen. See Figure 6-3, which shows how the screen looks when you create a new text file.

Figure 6-3
Text Editor Screen



6. You can use the following editing keys in the Text Editor:

Key	Action
↑, ↓, →, ←	Move cursor up, down, right, or left
Pg Up, Pg Dn	Scroll one page up or down
Home	Move to the beginning of the line
End	Move to the end of line
Ctrl + Home	Move to beginning of file
Ctrl + End	Move to end of file
Enter	Break or insert line (depends on cursor position)
Esc	Exit, do not save
Alt + B	Exit, save file

7. Press Alt + B to exit the Text Editor and save the text file.

Creating and Editing Data Files

You can use the Data Editor to edit a BoardSite binary data file that you previously downloaded from a host computer. You can also use the Data Editor to create a new BoardSite binary data file.

To use the Data Editor:

1. Select the Edit command from the menu bar.
2. In the Edit Options pop-up, select the Data File option. Press ↵.
3. In the Edit File Name pop-up, select the data file you want to edit, or select the Create New Data File option to create a new file.
4. If you selected the Create New Data File option in step 3, type a filename for the new file. Press ↵.

5. You should see the Data Editor screen. See Figure 6-4, which shows the screen looks when a sample data file **demo_2816a_data** is edited.

Figure 6-4
Data Editor Screen

```
File: demo_2816a_data                                          Cursor: 0

 ADDRESS                     HEXADECIMAL                            ASCII
00000000   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ................
00000010   10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F    ................
00000020   20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F     !"#$%&'()*+,-./
00000030   30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F    0123456789:;<=>?
00000040   40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F    @ABCDEFGHIJKLMNO
00000050   50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F    PQRSTUVWXYZ[\]^_
00000060   60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F    `abcdefghijklmno
00000070   70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F    pqrstuvwxyz{|}~.
00000080   80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F    ................
00000090   90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F    ................
000000A0   A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF    ................
000000B0   B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF    ................
000000C0   C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF    ................
000000D0   D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF    ................
000000E0   E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF    ................
000000F0   F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF    ................

↑←↓→   move cursor        PgUp   previous page       Alt-J  jump to address
Tab    switch hex/ASCII   PgDn   next page           Alt-P  print block
Home   first page         Alt-W  change word size    Alt-F  fill block with data
End    last page          Alt-O  change byte order   Alt-R  restore page/file
Esc    quit
```

Note: *If you create a new data file, you must fill the file with data before you edit it. Press **Alt+F**, select a fill option, type a block size (in hexadecimal), and then press **Alt+B**.*

6. You can use the following editing keys in the Data Editor:

Key	Action
↑, ↓, ←, →	Move cursor up, down, left, or right
Tab	Move cursor between the hex and ASCII data fields
Home	Move cursor to address 0
End	Show last page of data in file
Esc	Exit editor
PgUp	Display previous page of data
PgDn	Display next page of data
Alt + L	Search for match
Alt + N	Search for next match (used after **Alt-L**)
Alt + O	Reverse each data byte
Alt + J	Move cursor to a specified address
Alt + W	Change word size
Alt + P	Send block of file to LPT1 port
Alt + F	Fill a block of file with specified data
Alt + R	Revert file or page to previously saved version

7. Press [Esc] to exit the Data Editor.

8. If you made changes to the data file, the Leaving the Editor pop-up appears. Select an option, and then press [↵].

Using the Board Profile Editor

Use the Board Profile Editor to create and edit board profiles and to generate default sequence files. Each board you program has a board profile that describes the board to the system.

The board profile contains four different forms:

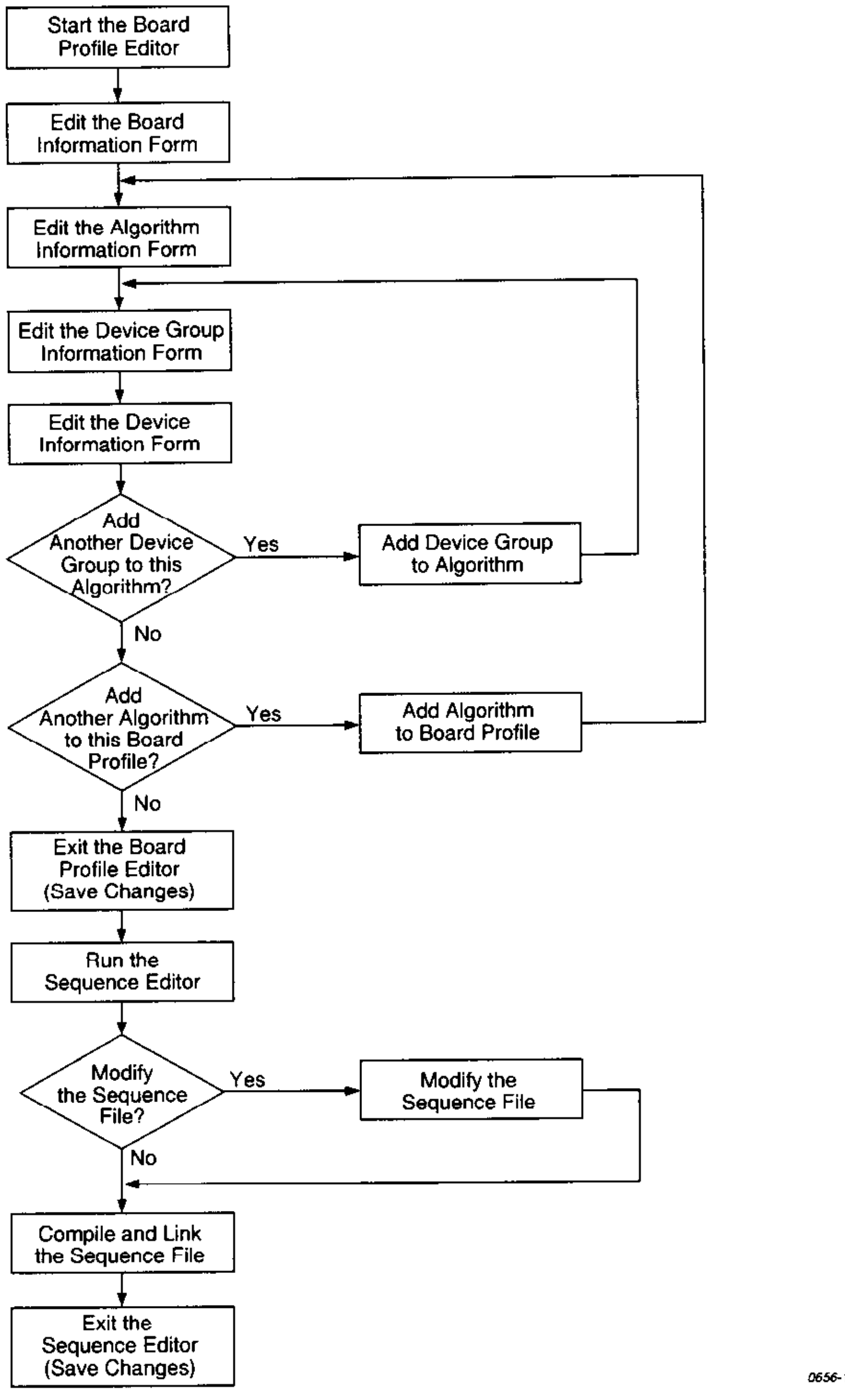
- Board information form
- Algorithm information form
- Device group information form
- Device information form

When you create a new board profile, all the forms contain default information. You can change the default information to customize the board profile.

The sequence file contains information that controls the flow of events required to test, program, and verify a circuit board. When you create a new board profile, BoardSite automatically creates a default sequence file for you. Usually, you only compile and link the default sequence file. You can, however, modify the sequence file to suit your board's special requirements. For complete information about sequence files and the Sequence Editor, see Chapter 7, "Sequence Editor Reference."

This section explains how to use the Board Profile Editor. Figure 6-5 shows a flow chart that describes how to create a new board profile. The topics following Figure 6-5 are procedures that explain each flow chart step. These procedures work for either creating a new board profile or for editing an existing board profile.

Figure 6-5
Creating a New Board Profile



To Start the Board Profile Editor

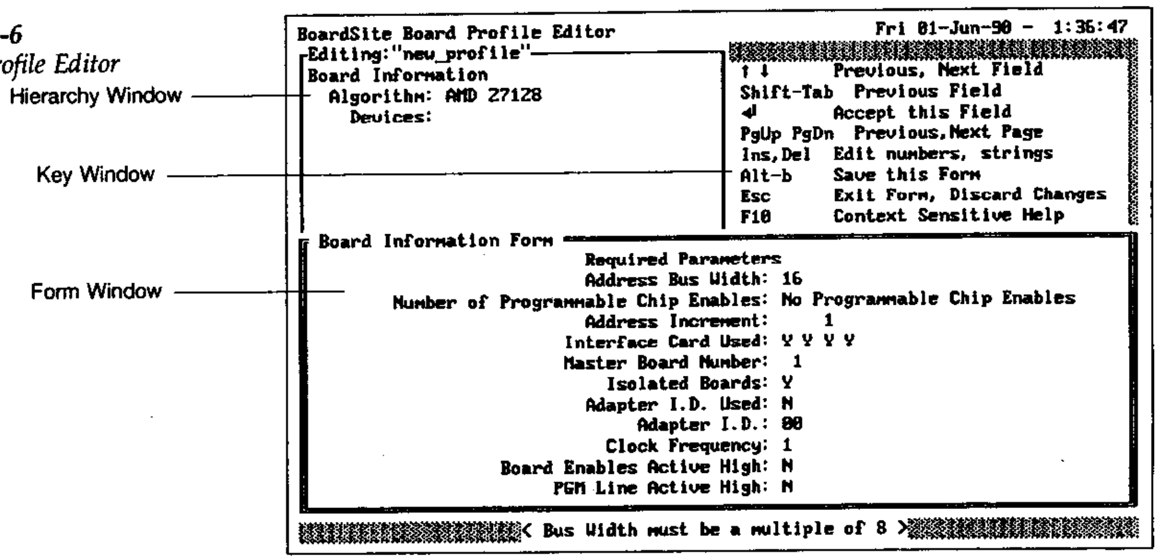
1. Select the Edit command from the menu bar.
2. In the Edit Options pop-up, select the Board Profile option. Press [↵] .
3. In the Edit File Name pop-up, select the Create New Board Profile option. Press [↵] .

Note: If you are editing an existing board profile, the Edit Board Profile or Sequence File pop-up appears. Select the Board Editor option, and then press Enter.

4. Type a name for the new board profile. Press [↵] .

After a few seconds, the Board Profile Editor screen appears. The screen should look like Figure 6-6.

Figure 6-6
Board Profile Editor



The editor screen is divided into three windows. The hierarchy window shows the board profile name and displays the profile hierarchy. The key window shows the editor keys and their actions. The form window contains the current form being edited.

The keys listed in the key window change, depending on what you are currently editing. Always check this window to find the keys that are appropriate for your current editing operation. Remember, you can always get context-sensitive help by pressing [F10] .

To Edit the Board Information Form

Note: If you are creating a new board profile, skip to step 2.

1. In the hierarchy window, select Board Information. Press [Tab] to activate the form window.

2. The board information form is in the form window, the window is active, and you can edit the form now. For complete descriptions of all board information form options, see the following section, "Board Information Form Reference."
3. When you finish editing the form, press Alt + B.

To Edit the Algorithm Information Form

Note: If you are creating a new board profile, skip to step 2.

1. In the hierarchy window, select the algorithm you want to edit. Press Tab to activate the form window.
2. The algorithm information form is in the form window, the window is active, and you can edit the form now.

   For complete descriptions of all algorithm information form options, see the following section, "Algorithm Information Form Reference."
3. When you finish editing the form, press Alt + B.

To Edit the Device Group Information Form

Note: If you are creating a new board profile, skip to step 2.

1. In the hierarchy window, select the device group you want to edit. Press Tab to activate the form window.
2. The device group information form is in the form window, the window is active, and you can edit the form now.

   For complete descriptions of all device group information form options, see the following section, "Device Group Information Form Reference."
3. When you finish editing the form, press Alt + B.

To Edit the Device Information Form

1. When you press Alt + B in the device group information form, the device information form automatically appears in the form window.
2. Edit the device information form now.

   For complete descriptions of all device information form options, see the following section, "Device Information Form Reference."
3. When you finish editing the form, press Alt + B.

To Add a Device Group to an Algorithm

If your board contains more than one device group, you can add more device groups to the board profile. When you add device groups, they always appear under the algorithm for the devices in the original group. In other words, you cannot mix programming-incompatible devices in device groups under the same algorithm.

You can add algorithms, however, to accommodate programming-incompatible devices. To learn how to add algorithms, see the next topic, "To Add an Algorithm to the Board Profile."

1. Make sure the cursor is in the hierarchy window, not the form window.
2. Press the [↑] and [↓] keys to position the cursor on the original device group. If you already have several device groups, position the cursor on the last device group in the series.
3. Press [Ins] to add the new device group.
4. When you press [Ins], the form window becomes active, and you can edit the form.

   For complete descriptions of all device group information form options, see the following section, "Device Group Information Form Reference."
5. When you finish editing the form, press [Alt] + [B].
6. When you press [Alt] + [B], the device information form automatically appears in the form window. Edit this form as described in the preceding topic, "To Edit the Device Information Form."
7. When you finish editing the device information form, press [Alt] + [B].

Notice that the new device group has been added to the hierarchy window.

To Add an Algorithm to the Board Profile

If your board has several programming-incompatible device types (devices having different family codes), you can add the algorithms for the other devices to the board profile. Each algorithm has its own set of device groups, so when you add an algorithm, you must also edit new device group information forms and device information forms.

1. Make sure the cursor is in the hierarchy window, not the form window.
2. Press the [↑] and [↓] keys to position the cursor on the original algorithm. If you already have several algorithms, position the cursor on the last algorithm in the series.
3. Press [Ins] to add the new algorithm.
4. When you press [Ins], the form window becomes active, and you can edit the form.

   For complete descriptions of all algorithm information form options, see the section, "Algorithm Information Form Reference," later in this chapter.
5. When you finish editing the form, press [Alt] + [B].

6. When you press [Alt] + [B], the device group information form automatically appears in the form window. Edit this form as described in the preceding topic, "To Edit the Device Group Information Form."
7. When you finish editing the device group information form, press [Alt] + [B].
8. When you press [Alt] + [B], the device information form automatically appears in the form window. Edit this form as described in the preceding topic, "To Edit the Device Information Form."
9. When you finish editing the device information form, press [Alt] + [B].

Notice that the new algorithm has been added to the hierarchy window, and its device group is under it.

To Exit the Board Profile Editor

1. Make sure the cursor is in the hierarchy window, not the form window.
2. Press [Alt] + [B] to save all the changes you made to the board profile, and to exit the editor.

To Run the Sequence Editor

If you create a new board profile, BoardSite automatically creates a default sequence file for you. However, you must run the Sequence Editor to compile and link the file.

1. When you press [Alt] + [B] to exit the Board Profile Editor, BoardSite displays the Run the Sequence Editor pop-up.
2. To run the Sequence Editor, type Y and then press [↵].

BoardSite automatically runs the Sequence Editor. For more information, see Chapter 7, "Sequence Editor Reference."

To Modify a Sequence File

If your board requires modifications to the sequence file, see the section, "Modifying a Sequence File," in Chapter 7, "Sequence Editor Reference." Usually, you only have to compile and link the sequence file.

To Compile and Link the Sequence File

You must always compile and link the default sequence file generated by the Board Profile Editor, even if you make no changes to the file.

1. From within the Sequence Editor, select Compile and Link from the Compile menu, or press [F8].
2. A message pop-up appears that tells you the status of the compile and link operations. Depending on the sequence file, the operations may take several minutes.
3. When the compile and link operations are finished, select Save File from the File menu.

To Exit the Sequence Editor

1. From the File menu, select Exit Editor.
2. BoardSite displays a message. Press any key (other than [Esc] or [N]) to exit the Sequence Editor.

Board Profile Reference

This section is the reference for the board profile forms. The section is divided into four topics. Each topic corresponds to a board profile form.

Board Information Form Reference

The board information form contains both required parameters and optional parameters. The following list describes all parameters in this form.

Required Parameters

Address Bus Width—Type the number of address lines you need for your circuit board. The number you type must not exceed 32 and must be a multiple of 8. The sum of the number of address lines and the number of programmable chip enable lines (PCE) cannot exceed 32.

Number of Programmable Chip Enables—Select the number of programmable chip enable lines that your circuit board requires. BoardSite uses the 16 upper address lines as chip enable lines, so the sum of the number of address lines and the number of programmable chip enable lines cannot exceed 32.

Address Increment—Type a number from 1 to 256. This specifies the number of address changes per data word.

Interface Card Used—There may be up to four interface cards installed in BoardSite. If you're not using all interface cards, type N in the appropriate place(s) in this field. This increases programming speed.

Master Board Number—Type a number to specify which memory board is the master board.

Isolated Boards—Type Y to use isolated programming, or N to use non-isolated programming.

Adapter I.D. Used—Type Y to use the adapter ID lines (ID0-ID7).

Adapter I.D.—If you typed Y for the Adapter I.D. Used field, type a hexadecimal number between 00 and FF for the adapter ID code.

Clock Frequency—Type a number to specify the clock frequency that you want applied to your circuit board. Type either 1 (for 1MHz), 2 (for 2MHz), 4 (for 4MHz), or 8 (for 8MHz).

Board Enables Active High—Type Y to configure the board enable lines (BE0-BE7) for active high or type N for active low.

PGM Line Active High—Type Y to configure the program pulse line (PGM) for active high or type N for active low.

Programmable Chip Enables Active High—Type Y to configure the programmable chip enable lines (PCE0-PCE15) for active high or type N for active low.

LEDs On for Failures—Type Y to turn on the LED drivers when a failure occurs.

LEDs Active High—Type **Y** to configure the LED lines (LED0-LED7) for active high or type **N** for active low.

Number of Verify Passes—Selects the number of verify passes for the board. If you type 1, BoardSite verifies your circuit board at the VCC voltage for pass 1 only (see next field). If you type 2, BoardSite verifies your circuit board at the VCC voltages for pass 1 and pass 2. If you type 3, BoardSite verifies your circuit board at the VCC voltages for pass 1, pass 2, and pass 3.

Verify VCC Voltages for Pass x—You typically choose these voltages so that pass 1 is the nominal VCC value, and pass 2 and 3 are the upper and lower VCC limits specified by the device manufacturer. Type the appropriate voltages in each of these fields.

Board Current Limits—Type the one-circuit-board current limit for each power supply. The current limits are programmed simultaneously for all interface cards. See the following parameter for more information.

Adapter Current Offsets—Compensates the current limits for the additional current drawn by the interface adapter. BoardSite calculates the current limit for each interface board using the following formula:

Current Limit = (number of boards x Board Current Limits)+ Adapter Current Offsets

Optional Parameters

Board Password—Use this field to require an operator to type a password before a board operation can start. Type any combination of letters or numbers. The password may be up to 72 characters long.

Instructions Text File Name—Press the [→] key to display a list of BoardSite text files. Select the text file that contains the operator instructions you want. You can create text files with the Text Editor (see the section "Edit," in this chapter).

Prompt Operator to Insert Boards—Type **Y** to tell BoardSite to display a prompt message to the operator.

Allow Operator to Select Device Names—Type **Y** to allow the operator to select device groups for board operations. If you type **N**, the Device Names pop-up won't appear when the operator selects the copy, verify, or test commands.

Blank Check—Press the [→] key to display the options, and then select an option.

Illegal Bit—Press the [→] key to display the options, and then select an option.

Stop on EEPROM Erase Blocklimit Error—Type **Y** to stop the operation and display an error message.

EEPROM Erase Error—Press the [→] key to display the options, and then select an option.

Lock Error—Press [→] to display the options, and then select an option.

Program Error—Press [→] to display the options, and then select one.

Verify Error—Press [→] to display the options, and then select one.

CRC/Checksum Error—Press [→] to display the options, and then select an option.

Log Errors to Error Log File—Type Y to log errors to the log file.

Log Errors to Printer—Type Y to send board operation errors to the LPT1 port.

Log All Status & Errors to Printer—Type Y to send all status messages, error messages, operation types, board profile names, data sources, and so on, to the LPT1 port.

Print Failures Summary—Type Y to send an error summary to the LPT1 port when the operation is complete.

Print Passed Boards Summary—Type Y send a summary of circuit boards that have passed the board operation to the LPT1 port.

Algorithm Information Form Reference

The algorithm information form contains information for each device listed in the hierarchy window.

Device Type— When you enter the Algorithm Information Form, the device name pop-up appears. If you're editing an existing board profile, the list automatically scrolls to the previously-selected device type; only devices with the same algorithm are displayed. If you're creating a new board profile, all available devices are displayed, and you can select any one of them.

Data Bus Width—Type the number of data lines the circuit board requires. The number must not exceed 32 and must be a multiple of 8.

Data MSB—If the data bus width is greater than 8 bits, you can change the alignment of the file data with the hardware data buses. Press the [→] key to display the options, and then select an option.

Board Data Mode—You can select the following data source options:

- **Master Board Data Only**—causes all board programming operations to use the master board for the data source. In Operator Mode, the operator cannot select data source options, and the Source Options pop-up does not appear.
- **Fixed Board Data File**—causes all board programming operations to use a BoardSite data file for the data source. You select the data filename in the board profile. In Operator Mode, the operator cannot select data source options, and the Source Options pop-up does not appear.
- **One Data File per Device**—uses an individual data file for each device. You select the files in the board profile. In Operator Mode, the operator cannot select data source options, and the Source Options pop-up does not appear.

- **One Data File for Entire Board**—allows the operator to change the data source. The Source Options pop-up appears, and the operator may select either the master board option or the data file option. If the operator selects the data file option, BoardSite uses the data filename defined in the board profile, and the operator cannot change this name.
- **Prompt for a Board Data File**—allows the operator to change the data source. The Source Options pop-up appears, and the operator may select either the master board option or the data file option. If the operator selects the data file option, BoardSite prompts the operator for a filename.

Copy Overhead Definition—You can use this field to customize the Copy command. Type one or more letters from the following list of options:

T Add the test operation to the overhead definition to perform blank check and/or illegal bit check.

P Add the program operation to the overhead definition to program boards.

V Add the verify operation to the overhead definition to verify boards.

C Add the CRC and checksum operations to the overhead definition.

L Add the lock operation to the overhead definition to perform a device lock operation, if applicable.

E Add the EEPROM erase operation to the overhead definition.

U Add the user algorithm operation to the overhead definition.

For example, to test, program, and then verify a board, type **TPV** in this field. When you select the Copy command, BoardSite performs these three operations in sequence.

Note: *You can define operations in any order; operations run in the order you type the letters. You can repeat an operation by typing its letter twice.*

Power Down Sequence—Define the sequence in which BoardSite shuts down the power supplies. This defines the power down sequence if BoardSite detects a major power problem with a board. Use the NULL option for delays and to fill unused positions at the end of the list. The normal power down sequence is defined in the sequence file.

Power Down Delay—Specifies the delay time between the shutdown of each power supply. Press [→] to display the options, and then select an option.

Vpp1 Slew Rate—Select the Fast option or the Slow option. Most algorithms require the Fast option.

Vpp2 Slew Rate—Select the Fast option or the Slow option. Most algorithms require the Fast option.

CRC and Checksum Option—Press [→] to display the options, and then select an option.

CRC and Checksum Operation—Either display or verify CRC and checksum results. If you verify CRC and checksum results, BoardSite generates an error message if the results don't match the reference values. Press [→] to display the options, then select an option.

CRC and Checksum Data Mode—Either select the Entire Board option or the Individual Devices option.

Board File—Type the filename for the board data file, or press [→] to display the data file list, and then select a filename from the list. You cannot change this field if the Board Data Mode is Master Board Only, One Data File per Device, or Prompt for a Board Data File.

Board CRC Value—Type the reference CRC value.

Board Checksum Value—Type a hexadecimal number for the reference checksum.

CRC Polynomial—Select one of the following polynomial options: CRC16, CRC16R, CCITT, CCITTR, CRC32, USER16, or USER32. The equations for these options are:

CRC16:	X16 + X15 + X2 + 1(mask = 8005)
CRC16R:	X16 + X14 + X1 + 1(mask = 4003)
CCITT:	X16 + X12 + X5 + 1(mask = 1021)
CCITTR:	X16 + X11 + X4 + 1(mask = 0811)
CRC32:	X32 + X31 + X2 + 1(mask = 80000005)

User CRC Polynomial—If you selected the USER16 or USER32 options in the previous field, you specify your own polynomial mask for the CRC calculation.

The following example shows how to derive the mask for the CRC16 polynomial. You derive masks by setting each bit within the mask high for each term present in the polynomial. You begin at X15 for a 16-bit polynomial, with X16 understood to be high.

CRC16 = X16 + X15 + X2 + 1

Most significant nibble =	$1 \times X^{15} = 8_{HEX}$
	$0 \times X^{14}$
	$0 \times X^{13}$
	$0 \times X^{12}$
2nd significant nibble =	$0 \times X^{11} = 0_{HEX}$
	$0 \times X^{10}$
	$0 \times X^{9}$
	$0 \times X^{8}$
3rd significant nibble =	$0 \times X^{7} = 0_{HEX}$
	$0 \times X^{6}$
	$0 \times X^{5}$
	$0 \times X^{4}$
Least significant nibble =	$0 \times X^{3} = 5_{HEX}$
	$1 \times X^{2}$
	$0 \times X^{1}$
	$1 \times X^{0}$

Mask = 8005

For 32-bit polynomials, you use the same procedure, only starting with X31 and assuming X32 is high.

Checksum Type —Press [→] to display the options, and then select an option.

Checksum Data Width—Type a number for the width of the checksum. The number can be no greater than 32 bits, and must be a multiple of 8.

Device Group Information Form Reference

The Device Group Information Form contains information for each device group under each algorithm in the hierarchy window.

Number of Address Ranges this Device Group—You can specify up to four non-contiguous address ranges. Type a number from 1 to 4.

Address Range 1
Address Range 2
Address Range 3
Address Range 4—Type hexadecimal numbers for the starting and ending addresses for each non-contiguous address block.

Programmable Chip Enables Used By This Group—If you use programmable chip enables, type **Y**. Define the PCE mask in the Device Information Form.

Number of Devices in Device Group—Type the number of devices in this device group.

Device Information Form Reference

After you type the information for a device group, you press [Alt] + [B] and then BoardSite displays the device information form. Use this form to set parameters for each individual device in the device group.

Device Name —Give the device a name by typing up to 8 alphanumeric characters in this field. Typical device names are U101, U102, and so on.

Device Data File —If you selected the One Data File per Device option in the algorithm information form, you select the filename for this device. The file should be the same size as the device.

Device Data File Offset—If you selected the One Data File per Device option in the algorithm information form, type the address offset (hexadecimal number) for this device.

Programmable Chip Enable Mask—If you are using programmable chip enables, type the hexadecimal number for the PCE mask for this device.

BoardSite ORs the chip enable lines for all devices in this device group to form a chip enable mask. In other words, if you want to activate PCE0, the PCE mask would be 01_{HEX}. If you want to activate PCE1, the PCE mask would be 02_{HEX}. If you want to activate both PCE0 and PCE1, the PCE mask would be 03_{HEX}.

Device CRC Value —If you selected the One Data File per Device option, and you selected CRC and checksum on individual devices (both in the algorithm information form), type the reference CRC for this device.

Device Checksum Value—If you selected the One Data File per Device option, and you selected CRC and checksum on individual devices (both in the algorithm information form), type the reference checksum for this device.

Setup

Use the Setup command to change your system's configuration.

To use the Setup command:

1. Select the Setup command from the menu.
2. In the Default System Parameters pop-up, select the desired setup options. See Figure 6-7 and the accompanying text. Follow the instructions in the upper window of the pop-up.

Figure 6-7
Default System Parameters Pop-up

```
  MODEL 4100 Rev X.XX                                  Wed 16-Oct-91 -  8:20:15
COMMANDS
 Key            Action
 ↑↓             Previous, Next field
 PgUp, PgDn     Scroll Up and Down the Form
 Ins, Del       Insert, Delete text
 Alt-b          Save Changes and Continue
 Esc            Exit this Form, Discard Changes
 F10            Context Sensitive help
 Alt-P          Print Form
Default System Parameters
                        Define Manager Mode Password: password
                     Define Technician Mode Password: password
    Specify Default Initial Drive and Directory Path
                                              Drive: E:
       Path: \REVS
                        Set Default Initial Mode: MANAGER
                   Enable Warning Tone on Errors: Y
                        Set BoardSite IRQ Number: 2
                      Disable 32-Character Names: N
    Specify Print Output Device Name or Path
       Name: AUX
                 Expansion Power Supplies Enabled: N
          < Enter up to 12 characters, embedded spaces allowed >
```

3. Press [Alt] + [B] to save the new configuration, or press [Esc] to discard all changes.

Define Manager Mode Password—Type up to 12 alphanumeric characters to define the password required to enter Manager Mode (when you press [Alt] + [M]). Embedded spaces are allowed. The default password is "password." The password is case-sensitive.

Define Technician Mode Password—Type up to 12 alphanumeric characters to define the password required to enter Technician Mode (when you press [Alt] + [T]). Embedded spaces are allowed. The default password is "password". The password is case-sensitive.

Specify Default Initial Drive and Directory Path—Type a drive letter (do not type the colon) and/or directory name.

Set Default Initial Mode—Select Manager, Technician, or Operator.

Enable Warning Tone on Errors—Type **Y** to enable the warning tone, or type **N** to disable it.

Set BoardSite IRQ Number—If you changed the IRQ jumper (JP2) setting when you set up your controller board (see the section, "Controller Board Setup," in Chapter 2) , you must change this option to match the IRQ number.

Disable 32-Character Names—If you prefer, you can use standard 8-character DOS filenames (with 3-character extensions) instead of the 32-character BoardSite filenames. The BoardSite software automatically appends 3-character filename extensions to the filenames. The advantage of using DOS 8-character filenames is that the filename displayed on the screen is the same as the first 8 characters of the filename you would see if you did the DOS DIR command. Using DOS filenames is desirable if you require strict traceability between the BoardSite filenames and DOS filenames.

The following extensions are appended to the BoardSite file types when you use the DOS Filenames option:

File Type	Extension
Board profiles	.PRO
Sequence files	.SEQ, .AOP, .EXE
Data files	.DAT
Text files	.TXT
BoardSite batch files	.JOB

Specify Print Output Device Name or Path—This parameter lets you specify the name of the device you wish to have printer output sent to when logging programming status and errors to the printer during programming operations, or when you select to print something from one of the editors or from a command menu.

The initial default value is LPT1. You could specify LPT2, LPT3, COM1, COM2, or some other valid DOS device name depending on your system configuration. You can also specify a file name by entering the whole path including the drive letter, such as: C:\REPORT.TXT or A:\OUT

Expansion Power Supplies Enabled—Type **Y** to enable the expansion power supplies in the BoardSite model 4420X.

Validate CRC/Checksum Calculations—Type **Y** to enable CRC/checksum calculations. If you type **Y**, BoardSite compares the calculated CRC and checksum against the expected CRC/checksum defined in the board profile. If the values are not equal, BoardSite displays an error message.

If you type **N**, BoardSite only displays the CRC/checksum, and does not generate an error message if they are not equal.

Default Cyclic Redundancy Check (CRC) Polynomial—Select the default CRC polynomial for all CRC calculations. Select CRC16, CRC16R, CCITT, CCITTR, CRC32, USER16, or USER32.

User Defined CRC Polynomial (optional)—If you select either USER16 or USER32 in the above option, then type a CRC polynomial mask. For more information, see the section "Board Profile," in this chapter.

Send final word through CRC calculations—Type Y to send an optional final CRC word through the CRC calculation. This final CRC word is sent after all other CRC data have been calculated.

Final CRC word size (optional)—Type the number of 8-bit bytes in the final CRC word. You usually type **2** if you use a 16-bit CRC polynomial, or **4** if you use a 32-bit CRC polynomial.

Final CRC word value (optional)—Type a hexadecimal number for the final CRC word.

Default Checksum Type—Select an option for the checksum calculation.

Default Checksum Data Width—Select the data width for the checksum calculation.

Change Video Attributes—To change your system's video attributes, type **Y** and then press ↵. Follow the instructions on the pop-up that appears.

File

Use the File command to manage your BoardSite working files. This command contains ten options: Create, Copy, Rename, Delete, List, Move, DOS, Import, Clear, and Statistics. Each option is described below.

IMPORTANT: BoardSite provides enhanced filenames for all BoardSite files. Filenames for data files, text files, board profiles, and so on can be up to 32 characters long, including embedded spaces. This allows you to have more meaningful names for files than the DOS 8-character name and 3-character extension allow.

To implement this extended name feature, BoardSite maintains its own catalog file, which has the DOS filename CATALOG (no extension). The catalog file maps the BoardSite filenames to DOS filenames. If you delete the file CATALOG, BoardSite will not be able to find your working files. Also, if you use DOS commands to manage files instead of using the File command, the catalog file will be incorrect and you may lose files.

Always use the File command to manage your BoardSite files. Do not use the DOS commands such as Delete, Erase, Rename, and so on.

Create

Use this command to create a new board data file or device data file from a master board.

To use the Create command:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the Create option. Press ↵ .
3. In the Board Profile Name pop-up, select the board profile that corresponds to the board from which you will create the file. Press ↵ .
4. If the Device Names pop-up appears, select either All Devices on the Board (to create a board data file) or an individual device (to create a device data file).

   If the Device Names pop-up does not appear, the options were automatically selected by the board profile. Go to step 5.
5. If enabled in the board profile, a Create Data File Name pop-up will allow you to select the BoardSite-generated filename, or to select the Create New Data File option. Press ↵ .
6. If you selected the Create New Data File option in step 6, type a filename for the new data file. Press ↵ .
7. In the Board Number pop-up, type a number for the position of the master board. Press ↵ .
8. Press Alt + B to begin the command.

Copy

Use this command to copy BoardSite files to a different filename. You can copy board profiles, data files, and text files.

To use the Copy command:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the Copy option. Press ↵ .
3. In the List Which Files pop-up, select the type of files you want to list, or select List All Files. Press ↵ .
4. In the Copy from BoardSite File pop-up, select the file you want to copy. Press ↵ .
5. Type a name for the new file. Press ↵ .
6. If the original file is a board profile, you can share its sequence file with the new file. This saves you the additional work of creating a new sequence file. To share the sequence file, type Y in the Copy Board Profile Option pop-up.

Note: If you type N in this pop-up, which means you don't want to share the sequence file, you must use the Board Profile Editor and Sequence Editor to create and compile a new file. See the section, "Edit," for information on these editors.

7. Press [↵] to copy the file.

Rename

Use this command to rename a BoardSite file to a different filename.

To use the Rename command:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the Rename option. Press [↵].
3. In the List Which Files pop-up, select the type of files you want to list, or select List All Files. Press [↵].
4. In the Rename BoardSite File pop-up, select the file you want to rename. Press [↵].
5. Type a new name for the file.
6. Press [↵] to rename the file.

Delete

Use this command to delete a BoardSite file.

To use the Delete command:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the Delete option. Press [↵].
3. In the List Which Files pop-up, select the type of files you want to list, or select List All Files. Press [↵].
4. In the Delete BoardSite File pop-up, select the file you want to delete. Press [↵].
5. Press [Alt] + [B] to delete the file.

List

Use this command to list the BoardSite catalog file. The list contains the current drive and directory, the free space on the current drive, the BoardSite filename, the corresponding DOS filename, the file size in bytes (also in hexadecimal for data files), and the date that the file was last modified.

To use the List command:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the List option. Press [↵].
3. In the List Catalog Options pop-up, select the Screen option to send the list to your PC screen, or select the Printer option to send the list to the LPT1 port.

4. In the List Which Files pop-up, select the type of files you want to list, or select List All Files. Press ↵ .
5. To match a filename fragment with the files in the catalog, type the fragment in the pop-up. If you don't want to match a filename fragment, do nothing.
6. Press ↵ to list the catalog.

Move

Use this command to change the current drive and directory for your BoardSite working files.

To use the Move command:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the Move option. Press ↵ .
3. Type a new drive letter (do not type the colon) and then press ↵ .
4. Type a new directory name and then press ↵ .
5. Press Esc to remove the pop-up.

DOS

Use this command to run the DOS shell. When you run the DOS shell, you temporarily return to DOS, from which you can run DOS commands or other programs. To return to BoardSite, type **exit** and then press ↵ .

To use the DOS command:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the DOS option. Press ↵ . You're now in the DOS shell.
3. To return to BoardSite, type **exit** and then press ↵ .

Import

Use this command to assign BoardSite filenames to DOS files, or to import BoardSite files from another drive and/or directory.

To assign BoardSite filenames to DOS files:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the Import option. Press ↵ .
3. In the Import from DOS or BoardSite File pop-up, select the DOS option. Press ↵ .
4. In the DOS File Name pop-up, select the DOS file. Press ↵ .

Note: *The file list is from the current drive and directory. To change the current drive and directory, see the topic, "Move," in this section.*

5. In the BoardSite File Type pop-up, select the type of BoardSite file you want the DOS file to be. Press ↵.
6. Type the new filename (you can use the full 32-character BoardSite filename).
7. Press ↵.

To import BoardSite files from another drive and/or directory:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the Import option. Press ↵.
3. In the Import from DOS or BoardSite pop-up, select the BoardSite option. Press ↵.
4. Type a drive letter (do not type the colon) and/or a directory name for the BoardSite file you want to import. Press ↵.

Note: Remember to type the leading backslash character (\) before the directory name. For example, type **\brdsite\work_dir**, *not* **brdsite\work_dir**.

5. In the Import Source File Name pop-up, select the BoardSite file you want to import. Press ↵.
6. Type the filename for the destination file. BoardSite writes the file in the current drive and directory.
7. Press ↵ to import the file.

Clear

Use this command to clear the system error log.

To use the Clear command:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the Clear option. Press ↵.
3. Press Alt + B to clear the error log.

Statistics

Use the statistics command to clear the programming statistics for a board profile.

To use the Statistics command:

1. Select the File command from the menu bar.
2. In the File Maintenance Options pop-up, select the Statistics option. Press ↵.
3. In the Board Profile Name pop-up, select the board profile for which you want to clear statistics.
4. Press ↵ to clear the statistics.

Batch

With the BoardSite system software, you can create and run batch files. When you run a batch file, BoardSite performs a series of predefined tasks, such as selecting the Copy command, selecting a board profile name, selecting device names, selecting a data filename, and so on. Batch files can be stored on disk, or the batch commands can be sent by a host computer via an RS-232 port.

In Manager Mode, you can create, edit, and run batch files. In Operator Mode and Technician Mode, however, you may only run batch files. The Batch command allows Manager Mode commands to be "recorded." These commands can then be run in Operator Mode. For example, in Manager Mode, a batch file may be created which, when run by an operator (in Operator Mode), changes the current drive and directory. These tasks are normally unavailable in Operator Mode.

Running a Batch File

There are two ways to run a BoardSite batch file. You can select the Batch command to run the file from the menu bar. Or, you can automatically run a batch file when you start the BoardSite system software from the DOS prompt.

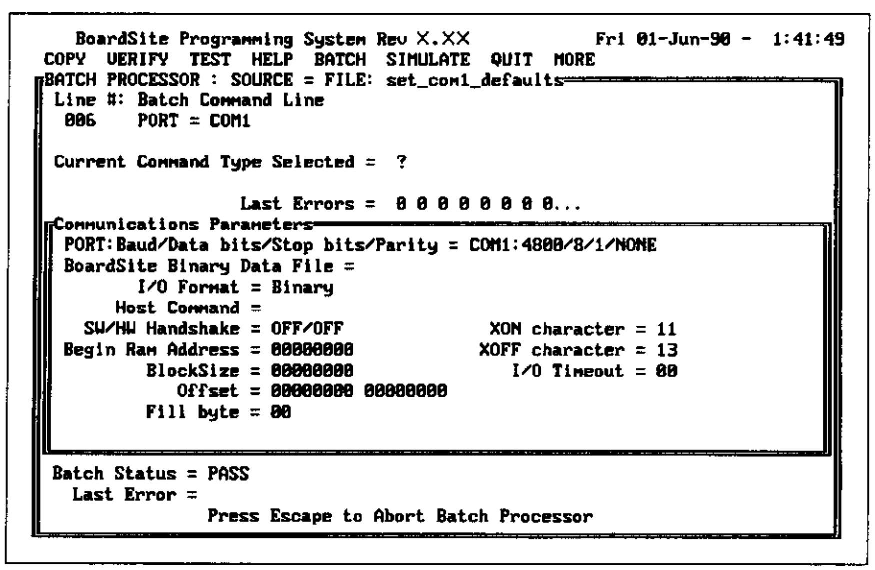
To Run a Batch File from the Menu Bar

You can run a batch file from the menu bar when the BoardSite software is running.

1. Select the Batch command from the menu bar.
2. In the Batch Options pop-up, select the Execute Batch option. Press [↵].
3. In the Execute Batch pop-up, select the batch file you want to run. Press [↵].
4. Press [Alt] + [B] to run the batch file.

Figure 6-8 shows how the screen looks when the batch file runs.

Figure 6-8
Batch File Screen



```
   BoardSite Programming System Rev X.XX          Fri 01-Jun-90 -  1:41:49
COPY  VERIFY  TEST  HELP  BATCH  SIMULATE  QUIT  MORE
BATCH PROCESSOR : SOURCE = FILE: set_com1_defaults
 Line #: Batch Command Line
  006    PORT = COM1

 Current Command Type Selected =  ?

                  Last Errors =  0 0 0 0 0 0 0 0...
 Communications Parameters
  PORT:Baud/Data bits/Stop bits/Parity = COM1:4800/8/1/NONE
  BoardSite Binary Data File =
         I/O Format = Binary
       Host Command =
    SW/HW Handshake = OFF/OFF               XON character = 11
  Begin Ram Address = 00000000             XOFF character = 13
          BlockSize = 00000000                I/O Timeout = 00
             Offset = 00000000 00000000
          Fill byte = 00

 Batch Status = PASS
   Last Error =
              Press Escape to Abort Batch Processor
```

To Run a Batch File when BoardSite Starts

You can automatically run a batch file when you start the BoardSite system software from the DOS prompt. For example, you could set the BoardSite software to run an initial batch file that moves to a specific drive and directory, selects the Copy command, selects the board profile name, selects the devices to be programmed, select the data file to use, and then wait for the operator to begin the operation.

1. Type the following at the DOS prompt:

   brdsite -b"*filename*"

   where *filename* is the BoardSite filename (not the DOS filename) for the batch file you want to run. For example, if you have a BoardSite batch file named **set_com1_defaults**, you could run it automatically by typing the following line at the DOS prompt:

   brdsite -b"set_com1_defaults"

2. Press ↵ to start BoardSite and to run the batch file.

Sending Batch Commands Via the Serial Port

You can send batch commands to BoardSite via your PC's serial port. Using this method, you can create batch files on your host computer that remotely control BoardSite.

You can either start the remote batch operation from the menu bar, or you can start the remote batch operation automatically when you start the BoardSite software.

To Start the Remote Batch Operation from the Menu Bar

1. Select the Batch command from the menu bar.
2. In the Batch Options pop-up, select the Execute Batch option. Press ↵.
3. In the Execute Batch pop-up, select either COM1 or COM2, depending upon which port you want to use. Press ↵.
4. Press Alt + B to begin remote batch operation.
5. You can now start sending batch commands from the host.
6. To stop the remote batch operation at any time, press Esc.

To Start the Remote Batch Operation Automatically

1. Type the following at the DOS prompt: **brdsite -bcom1**. If the port connected to the host is COM2, type: **brdsite -bcom2**
2. Press ↵ to start the remote batch operation.

Note: *If you send batch commands via a serial port, you may want to run another batch file before you start the remote operation. This batch file selects the correct serial port parameters (using the Communications command) and then selects COM1 or COM2 as the batch command port. This will ensure that the serial port parameters are always correct no matter what the previous port settings were. To create this batch file, try using the learn batch mode described in the next section.*

3. You can now start sending batch commands from the host.
4. To stop the remote batch operation at any time, press Esc.

Creating a BoardSite Batch File

There are two ways you can create a BoardSite batch file. You can place the system in learn batch mode, and record every command you select. Or, you can use the Text Editor to write a batch file using the batch commands.

To Learn a Batch File

1. Select the Batch command from the menu bar.
2. In the Batch Options pop-up, select the Learn Batch option. Press ↵.
3. In the Learn Batch File pop-up, select the Create New Batch File option. Press ↵.

Note: If you select an existing file in this pop-up, all batch commands in the file will be replaced by the commands you record.

4. Type a name for the new batch file. Press ↵. The system is now in learn batch mode.
5. Type all the commands, keystrokes, and so on, that you want to record.
6. When you're done recording, select the Batch command from the menu bar.
7. In the Batch Options pop-up, select the Learn option to stop recording. Press ↵.

To see how the Learn Batch command works, try the following procedure to record a batch file that automatically sets your COM1 parameters.

1. Select the Batch command from the menu bar.
2. In the Batch Options pop-up, select the Learn Batch option. Press ↵.
3. In the Learn Batch File pop-up, select the Create New Batch File option. Press ↵.
4. Type the following name for the new batch file:
   set_com1_defaults
5. Press ↵.
6. Select the Communications command from the menu bar.
7. In the Communications Options pop-up, select the Parameters option. Press ↵.
8. In the Communications Port pop-up, select the port you want to set parameters for (usually COM1). Press ↵.
9. Assume the parameters you see on the screen are your default parameters. Press Alt + B to save the default parameters.
10. Press F1 to return to the menu bar.
11. Select the Batch command from the menu bar.

12. In the Batch Options pop-up, select the Learn option to stop recording. Press ↵ .

    Your new batch file is now saved on disk. To see what the batch file looks like, perform the following steps.

13. Select the Batch command from the menu bar.
14. In the Batch Options pop-up, select the Edit option. Press ↵ .
15. In the Edit Batch File pop-up, select the filename set_com1_defaults. Press ↵ . After a few seconds, the batch file appears in the text editor window of the screen. Notice how the batch commands appear in the file.
16. When you're finished examining the batch file, press Alt + B to exit the editor.

    You can also run this batch file if you want.

17. To run this batch file, see the preceding section, "To Run a Batch File from the Menu Bar."

To Create and Edit a Batch File

You can create a batch file by writing batch commands in a file, or you can edit any existing batch file. The editor you use is the BoardSite Text Editor. See the preceding section, "Creating and Editing Text Files," for more information on the Text Editor.

1. Select the Batch command from the menu bar.
2. In the Batch Options pop-up, select the Edit option. Press ↵ .
3. To create a new batch file, select the Create New Batch File option. To edit an existing batch file, select the filename. Press ↵ .
4. If you selected the Create New Batch File option in step 3, type a filename for the new batch file. Press ↵ .
5. BoardSite runs the Text Editor. You should see the text editor screen shown in Figure 6-3.
6. Type the batch commands. See the following section, "Batch Language Reference," for a complete description of the batch commands.
7. When you finish typing the batch commands, press Alt + B to exit the editor and save the batch file.

Batch Language Description

This section describes the batch language protocol. The batch language is designed to mimic the way you select commands and options from the menu bar. The batch language consists of commands that specify which BoardSite command to run, commands that set parameters or options, and commands that actually cause an operation to run.

In this section, names in quotes (for example "data file name") indicate that this parameter is a BoardSite filename. This filename may contain embedded spaces, is case sensitive, and must be enclosed in quotes. In general, the batch commands and parameters are not case sensitive, except for filenames and translation format names.

Host Responses

When you receive batch commands via the serial port, BoardSite sends responses back to the host. In general, all commands send PASS if no error occurred and FAIL if an error occurred. Some commands send additional information to the host and then send PASS or FAIL. If a command sends additional information (a response), the response is described in the documentation for the command. For command documentation, see the following section, "Batch Language Reference."

Errors

If a batch syntax error occurs, the batch processor stops.

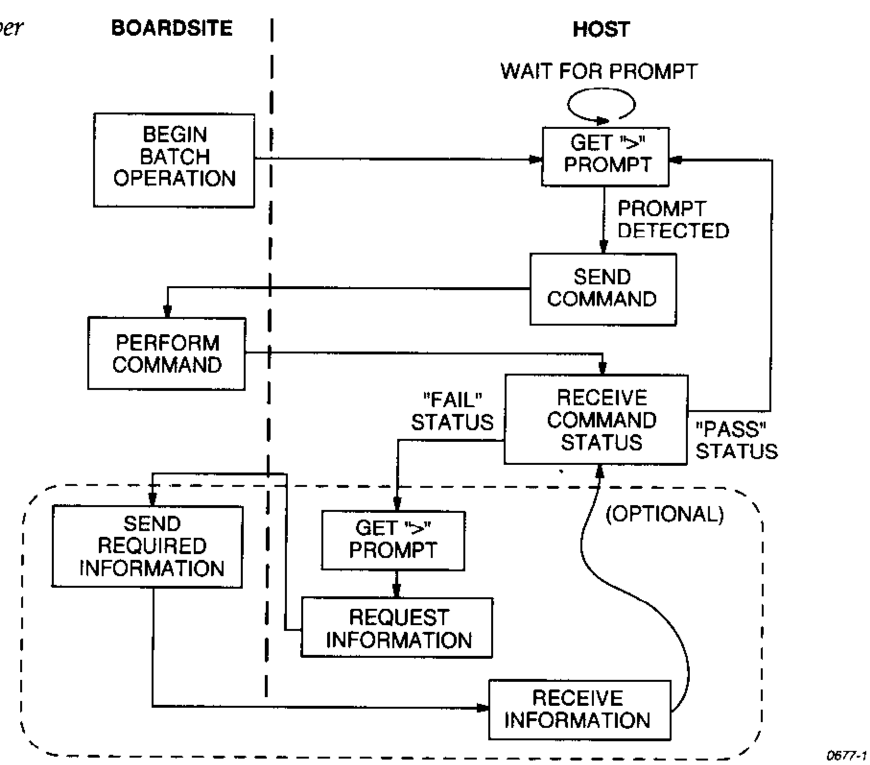
If an error (not syntactical) occurs during the batch COPY command, you can use the batch commands IF_ERROR GOTO label, IF_ERROR EQ number GOTO label, and IF_ERROR NE number GOTO label, to control your batch program flow during error conditions. If you don't use these commands, the batch processor only writes the error number in the error buffer. It does not stop processing.

When BoardSite processes batch commands and runs programming commands, the software stores the last 16 errors in a circular error buffer. You can clear this buffer or obtain its status output when you send batch commands via the serial port.

Protocol for a Remote Control Driver

When you send batch commands via the serial port, BoardSite sends responses to the host. You can use these responses to write a remote control driver for your host computer. Figure 6-9 shows a typical protocol for a remote control driver.

Figure 6-9
Protocol for Remote Control Driver



Syntax for Remote Batch Commands

1. When BoardSite sends the prompt character > and the ASCII line feed <LF> and carriage return <CR> characters, it is ready to receive the next command from the host.
2. Commands sent by the host must be terminated by a <CR>.
3. All commands (except request-information commands) send PASS or FAIL when they are complete, followed by <LF><CR>, and then the prompt > <LF><CR>.
4. Request-information commands (SEND_ERROR_FILE, for example) send the requested information, followed by the status (PASS or FAIL) and prompt characters as shown above.
5. The host may query BoardSite during long operations to determine if BoardSite is still responding. To do this, the host sends a Ctrl-R character (ASCII DC2, hexadecimal character number 12). BoardSite responds by sending the prompt character > to the host.
6. The host may immediately stop the COPY, VERIFY, TEST, or CHECKOUT commands by sending an escape character (ASCII ESC, hexadecimal character number 1B) to BoardSite. If the host sends this character, BoardSite immediately stops the command in process, and writes 215 to the error buffer.

Remote Batch Session Example

The following example shows the actual characters sent back and forth during a remote batch session. Characters in normal type are the characters sent by BoardSite to the host. Characters in bold and italic type are sent by the host to BoardSite.

Characters Sent/Received	Action
> <LF><CR>	BoardSite prompt
clear_errors<CR>	Host sends command to clear errors
PASS <LF><CR>	Response back from BoardSite
> <LF><CR>	
display_message="Press any Key to Continue:"<CR>	
	Host sends command to display message
PASS <LF><CR>	Response back from BoardSite
> <LF><CR>	
wait_for_key<CR>	Host sends command, waits for key press
PASS <LF><CR>	
> <LF><CR>	
copy<CR>	Set the command to copy
PASS <LF><CR>	
> <LF><CR>	
board_profile_name=demo_2816a<CR>	Set the Board Profile name
PASS <LF><CR>	
> <LF><CR>	
device_name=u16<CR>	Set the device names
PASS <LF><CR>	
> <LF><CR>	
source_option=disk <CR>	Set the source option
PASS <LF><CR>	
> <LF><CR>	

data_file_name=demo_2816a_data<CR> Select the data file
PASS <LF><CR>
> <LF><CR>

copy_overhead=TPV<CR> Specify copy overhead definition
PASS <LF><CR>
<LF><CR>

begin_command<CR> Start the copy operation

FAIL <LF><CR>
> <LF><CR>
Response from BoardSite about failed Copy operation

send_errors<CR> Request errors from BoardSite

201 421 0 0 0 0 0 0 0 0 0 0 0 0 0 0 <LF><CR>
PASS <LF><CR>
> <LF><CR>

send_board_stats <CR> Request Board Statistics

```
00000000000000 00000000000000 00000000000000 00000000000000 <LF><CR>
00000000000000 00000000000000 00000000000000 00000000000000 <LF><CR>
00000000000000 00000000000000 00000000000000 00000000000000 <LF><CR>
00000000000000 00000000000000 00000000000000 00000000000000 <LF><CR>
00000000000000 00000000000000 00000000000000 00000000000000 <LF><CR>
00000000000000 00000000000000 00000000000000 00000000000000 <LF><CR>
00000000000000 00000000000000 00000000000000 00000000000000 <LF><CR>
00000000000000 00000000000000 00000000000000 00000000000000 <LF><CR>
```

PASS <LF><CR>
> <LF><CR>

Batch Language Reference

This section describes each batch command. You can also get online help for the batch commands by pressing F10 when you're editing a batch file.

The Batch Language Reference is divided into four sections: Batch Environment Commands, Serial Port Commands, Programming Operation Commands and Miscellaneous Commands.

Batch Environment Commands

These commands modify the batch environment:
BATCH_INPUT = COM1
BATCH_INPUT = COM2
BATCH_INPUT = *"filename"*
This command changes the batch command input source to either COM1, COM2, or another batch file. This allows a batch file to chain to another batch file, specified by filename, or to switch the batch input to COM1 or COM2. The command can also switch from COM1 or COM2 to a batch file. This command also resets the batch line number.

Response: If the command cannot switch the input source, the system sends FAIL. If the input source is switched successfully from COMn to a batch file, no response is sent, because the batch processor is no longer communicating with COM1.

CHANGE_PATH = *drive:\path*
This command changes the current drive and directory to the drive and directory specified. A typical drive and path specification might be:
c:\brdsite\work_dir

CLEAR_ERRORS
This command clears the circular error buffer.

CLEAR_BOARD_STATS
This command clears the current board profile statistics. Before you use this command, use the BOARD_PROFILE_NAME command to select a board profile.

CLEAR_ERROR_LOG
This command clears the system error log.

DISPLAY_MESSAGE = "*string*"
This command sends a message to the PC screen. The *string* may be up to 50 characters long. The message is displayed in a pop-up that appears in the upper right-hand corner of the screen. The *string* may contain special characters. The characters "\n" cause a line feed, and the characters "\b" cause the bell to beep. A typical *string* might be: **"Please put memory boards in adapter\n\b"**

END_BATCH
This command causes BoardSite to leave batch mode. BoardSite displays a message on the batch status display, and then waits for the operator to press a key before returning to the menu bar.

QUIT_BOARDSITE
This command causes the PC to leave the BoardSite software and return to DOS.

HELP
This command sends a list of batch commands to the host.

PRINTER_ECHO = ON
PRINTER_ECHO = OFF
This command sets the printer echo flag to either ON or OFF. If the flag is ON, status and error messages are sent to the printer.

SEND_ERRORS
This command sends the last 16 errors to the host. The errors are in decimal format. For example, the command may send the following:

409 410 403 0 0 0 0 0 0 0 0 0 0 0 0 0

Each number corresponds to a BoardSite message. For example, 409 corresponds with Error 409. For a complete list of messages, see Chapter 8, "Messages."

SEND_CONFIGURATION
This command sends to the host the software version, free space on the current working drive, model number, and free memory.

SEND_HOST_COMMAND = "*string*"
Sends a text string to the host. The *string* may contain any alphanumeric characters, and the following special control characters:

Character	Action	ASCII Character Sent
\n	Line feed	Ctrl + J
\r	Carriage return	Ctrl + M
\b	Backspace	Ctrl + H

\f	Form feed	Ctrl + L
\a	Alert (bell)	Ctrl + G
\t	Horizontal tab	Ctrl + I
\\	Backslash	\

If you send a backslash character \ followed by any character other than the ones shown in this table, both the backslash character and the following character are sent without conversion.

BEGIN_COMMAND
This command simulates pressing Alt + B from the keyboard. It causes the previously selected batch command (COPY, VERIFY, CHECKOUT, UPLOAD, for example) to begin, assuming enough parameters have been set. For example, a COPY command requires that the board profile has been selected, and, depending on the options specified in the board profile, you may have to specify device names, source options, and data filename.

Response: Sends PASS if the command runs without any errors, or FAIL if errors occur. If FAIL is sent, use the SEND_ERRORS command to read the error number.

WAIT_FOR_ESCAPE
This command is similar to BEGIN_COMMAND in that it begins the previously selected command. WAIT_FOR_ESCAPE, however, requires the operator to press Alt + B to begin the command. After the command is finished, WAIT_FOR_ESCAPE causes BoardSite to display the standard prompt along the bottom of the screen. The operator can either press Alt + B to repeat the command, or press Esc to run the next batch command.

Response: Sends PASS if the command runs without any errors, or FAIL if errors occur. If FAIL is sent, use the SEND_ERRORS command to read the error number.

WAIT_FOR_KEY
This command waits for the operator to press a key on the keyboard, and then continues to run the batch file. A typical use of this command would be to send a prompt to the operator using the DISPLAY_MESSAGE command, and then pause until the operator reads the prompt and presses a key.

Response: Sends PASS if the operator presses a key. If the operator does not press a key, and the operator time-out occurs (see OPERATOR_TIMEOUT below), then FAIL is sent and ERROR 421 is written into the error buffer.

BATCH_DELAY = *nn*
This command sets the number of seconds the batch processor waits after each command. This is useful when you debug a batch file, so that you can see the status and error messages on the screen. The value *nn* is the delay in seconds, and can be from 0 to 99.

BATCH_TIMEOUT = *nn*
This command sets the batch time-out value. This is the number of seconds the batch processor waits for a command from the host before timing out. The value *nn* is a decimal number from 0 to 99. If *nn*=0 (default), the batch time-out is disabled. If a batch time-out occurs, the batch processor sends another prompt character to the host, and then writes 705 in the error buffer.

OPERATOR_TIMEOUT = *nn*
This command sets the operator time-out value, which is the number of seconds the batch processor will wait for the operator to press a key when the WAIT_FOR_KEY command is used. The operator time-out also applies to the CHECKOUT command which waits for the operator to press a key, and the COPY, VERIFY, and TEST commands, when waiting for the operator to press a key after a STOP ON ERROR message. If the operator time-out occurs, the batch processor behaves as if a key were pressed, but writes 421 into the error buffer.

If *nn*=0, the operator time-out is disabled and the batch processor waits indefinitely for a key.

If *nn*= -1, the batch processor waits only about 1/18 second for a key. When the time-out occurs, the batch processor behaves as if a key were pressed, but does not write an error into the error buffer.

WAIT = *nn*
This command causes the system to pause for *nn* seconds, where *nn* is a decimal number from 0 to 99. After the pause, BoardSite write 421 to the error buffer.

GOTO *label*
This command causes an immediate jump to the batch command preceded by *label*, where *label* is an alphanumeric name containing no spaces, and preceded by a colon in column one. Any characters in the line after the *label* are ignored. In the following example, the GOTO command causes an immediate jump to the COPY_OVERHEAD command.

```
...
...
GOTO :overhead
...
...
...
:overhead
COPY_OVERHEAD = TPVC
...
...
```

IF_ERROR GOTO *label*
This command causes a conditional jump to the batch command preceded by *label*, where *label* is an alphanumeric name containing no spaces, and preceded by a colon in column one. Any characters in the line after the *label* are ignored. The jump occurs only if there is an error in the error buffer.

IF_ERROR EQ *number* GOTO *label*
This command causes a conditional jump to the batch command preceded by *label*, where *label* is an alphanumeric name containing no spaces, and preceded by a colon in column one. Any characters in the line after the *label* are ignored. The jump occurs only if the error *number* is in the error buffer.

IF_ERROR NE *number* GOTO *label*
This command causes a conditional jump to the batch command preceded by *label*, where *label* is an alphanumeric name containing no spaces, and preceded by a colon in column one. Any characters in the line after the *label* are ignored. The jump occurs only if the error *number* is not in the error buffer.

VERBOSE = ON
VERBOSE = OFF
This command controls the amount of information displayed on the batch screen (see Figure 6-7) when the batch processor runs. If VERBOSE=OFF, BoardSite displays less information on the screen, but the batch processor runs faster. These settings are remembered between BoardSite sessions.

Serial Port Commands

The following commands all apply to the COM1 and COM2 serial ports. When you run a command to change a serial port parameter (baud rate, parity, and so on), the parameter is automatically saved internally. When you run the BATCH_INPUT = COM1 or BATCH_INPUT = COM2 commands, the corresponding serial port is opened with the current parameters.

Note: *You set the default serial port parameters using the Communications command. Any batch commands that change the serial port parameters only change the parameters while the batch processor is running. When the batch processor stops running, the parameters return to the default settings.*

PORT = COM1
PORT = COM2
To select the correct port, issue this command before you change parameters or run the UPLOAD or DOWNLOAD commands. The default port is the last port used in a communications operation.

BAUD_RATE = 300
BAUD_RATE = 1200
BAUD_RATE = 2400
BAUD_RATE = 4800
BAUD_RATE = 9600
BAUD_RATE = 19200
This command sets the baud rate for the previously selected port.

BEGIN_ADDRESS = *XXXXXXXX*
This command sets the beginning address for the UPLOAD and DOWNLOAD commands. *XXXXXXXX* is a hexadecimal number up to 8 digits long.

BLOCK_SIZE = *XXXXXXXX*
This command sets the block size for the UPLOAD and DOWNLOAD commands. *XXXXXXXX* is a hexadecimal number up to 8 digits long.

OFFSET = *XXXXXXXX*
This command sets the I/O offset for the UPLOAD and DOWNLOAD commands. *XXXXXXXX* is a hexadecimal number up to 8 digits long.

OFFSET_HI = *XXXXXXXX*
This command sets the high I/O offset for the UPLOAD and DOWNLOAD commands. *XXXXXXXX* is a hexadecimal number up to 8 digits long. This command applies only to the Tektronix Hex Extended format.

DATA_BITS = 7
DATA_BITS = 8
This command sets the data bits for the previously selected port. Always use DATA_BITS = 8 for the Binary and DEC Binary translation formats.

PARITY = NONE
PARITY = ODD
PARITY = EVEN
PARITY = MARK
PARITY = SPACE
This command sets the parity of the previously selected port.

STOP_BITS = 1
STOP_BITS = 2
This command sets the stop bits to either 1 or 2.

TIMEOUT = *nn*
This command sets the I/O time-out to *nn* seconds, where *nn* is a decimal number from 0 to 99. This affects DOWNLOAD, UPLOAD, and the Batch command itself.

HARDWARE_HANDSHAKE = ON
HARDWARE_HANDSHAKE = OFF
This command enables or disables hardware handshaking (RTS/CTS) during UPLOAD and DOWNLOAD commands. It should only be used if full hardware handshaking is implemented on the RS-232 cable.

SOFTWARE_HANDSHAKE = ON
SOFTWARE_HANDSHAKE = OFF
This command enables or disables software handshaking (XON/XOFF) during UPLOAD and DOWNLOAD commands. It should be used when a 3-wire RS-232 cable is being used.

XOFF = *XX*
XON = *XX*
These commands define either the XOFF or XON characters. *XX* is a hexadecimal number from 00 to FF. This command affects UPLOAD and DOWNLOAD commands, and the Batch command itself, but only if SOFTWARE_HANDSHAKE=ON. The defaults are

XOFF = 13
XON = 11

DOWNLOAD_OUTPUTFILE = *"filename"*
This command defines the BoardSite filename for DOWNLOAD operations. If the file already exists, it will be overwritten. If the file does not exist it will be created.

UPLOAD_INPUTFILE = "*filename*"
This command sets the BoardSite filename for UPLOAD operations. The file must already exist, because it is opened in the read mode.

FILL_BYTE = *XX*
This command sets the fill-byte value. *XX* is a hexadecimal number from 00 to FF. This value is used when creating new files during DOWNLOAD operations.

FORMAT = Binary
FORMAT = DEC Binary
FORMAT = Hewlett-Packard 64000 Abs. Obj.
(and so on)
This command selects the translation format for the UPLOAD and DOWNLOAD commands. The translation format name is case-sensitive, and must exactly match the translation format name in the I/O Format field of the Communications command. See Appendix B, "Translation Formats," for more information.

HOST_COMMAND = "*string*"
The *string* may be up to 50 characters long. This command defines the host command, which is sent to the host at the start of every UPLOAD and DOWNLOAD operation. The special characters described in the SEND_HOST_COMMAND batch command can be used this command also.

DOWNLOAD
This command sets the command type to DOWNLOAD. Running BEGIN_COMMAND then initiates the DOWNLOAD operation.

UPLOAD
This command sets the command type to UPLOAD. Running BEGIN_COMMAND then initiates the UPLOAD operation.

TRANSLATE
Translates files from BoardSite binary to DOS ASCII or vice-versa. See the following commands, TRANSLATE_OPTION, DOS_FILE, and TRANSLATE_FILE. Following is an example of how to use these commands to translate a BoardSite binary file to an ASCII file.

```
OFFSET = FFFFFFFF
OFFSET_HI = FFFFFFFF
BEGIN_ADDRESS = 00000000
BLOCK_SIZE = 00000100
FORMAT = "Motorola 32-bit (S3)"
TRANSLATE
TRANSLATE_OPTION = BINARY
DOS_FILE = "test.dat"
FILL_BYTE = 00
TRANSLATE_FILE = "data"
BEGIN_COMMAND
```

TRANSLATE_OPTION = BINARY
TRANSLATE_OPTION = ASCII
This command sets the type of translation for the TRANSLATE command. For BoardSite binary-to-ASCII, use TRANSLATE_OPTION = BINARY. For ASCII-to-BoardSite binary, use TRANSLATE_OPTION = ASCII.

DOS_FILE = "*filename*"
This command sets the filename for the ASCII file for the TRANSLATE command. *Filename* must be a DOS filename.

TRANSLATE_FILE = "*filename*"
This command sets the filename for the BoardSite binary file for the TRANSLATE command. *Filename* must be a BoardSite filename.

Programming Operation Commands

The following commands are related to programming operations:

COPY
VERIFY
TEST

These three commands set the command type to Copy, Verify or Test. To actually run the operations, set the board profile name and other parameters as required, and then issue BEGIN_COMMAND or WAIT_FOR_ESCAPE.

BOARD_PROFILE_NAME = "*board profile name*"
This command selects the board profile name. The board profile must already exist in the current working directory. When you select the board profile, the copy overhead definition (see COPY_OVERHEAD command) is cleared, and the list of device names (see DEVICE_NAME command) is cleared.

COPY_OVERHEAD = ETPVLCU
This command specifies a copy overhead definition. For more information, see the section, "Board Profile Reference," earlier in this chapter. This command temporarily overrides, but does not modify, the copy overhead definition contained in the board profile. This command affects only the COPY command.

Use the following letter codes in any combination:

E Perform an erase operation if device is an EEPROM.

T Perform a test (blank-check and illegal-bit) operation.

P Perform a program operation.

V Perform a verify operation.

L Perform a lock operation if the device allows security programming.

C Perform a CRC/checksum display or CRC/checksum verify operation.

U Run a user algorithm.

Example: The default for most devices is TPVC. First select a board profile using the BOARD_PROFILE_NAME command and then issue the COPY_OVERHEAD = E command. The E would cause only an EEPROM erase to occur when you issue BEGIN_COMMAND.

DATA_FILE_NAME = "*data file name*"
This command specifies the binary data file. The file must exist in the current working directory.

Note: If the previously selected board profile has the Board Data Mode set to either Master Board Data Only, Fixed Board Data File, or One Data File per Device, you do not need to select a data file. For more information, see the topic, "Algorithm Information Form Reference," in the section, "Board Profile Reference," earlier in this chapter.

DEVICE_NAME = *name*
This command selects device name(s) and device group(s) for the COPY command. Use DEVICE_NAME = ALL to select all the devices on the board.

To program all of the devices in any device group, you need only select one of the device names in the group. For example, if you have two devices called U1 and U2, in the same device group, then the command DEVICE_NAME = U1 will program both devices.

Note: If the Allow Operator to Select Device Names option is set to N in the board profile, you do not need to use the DEVICE_NAME command, because all devices on the board will be selected automatically.

BOARD_CRC = *XXXXXXXX*
This command sets the board CRC value to *XXXXXXXX*, where *XXXXXXXX* is a hexadecimal number. This command temporarily overrides, but does not modify, the Board CRC Value option in the board profile. It also tells BoardSite to perform a CRC verify, ignoring whatever options are in the board profile. See the next command, BOARD_CHECKSUM, for more information.

BOARD_CHECKSUM = *XXXXXXXX*
This command sets the board checksum value to *XXXXXXXX*, where *XXXXXXXX* is a hexadecimal number. This command temporarily overrides, but does not modify, the Board Checksum Value option in the board profile. It also tells BoardSite to perform a checksum verify, ignoring whatever options are in the board profile.

The BOARD_CHECKSUM and BOARD_CRC commands are very useful when you use the same board profile but different data files. These commands allow you to override the CRC and checksum values in the board profile, so you don't have to create different board profiles just to change the CRC and checksum values. Here is an example of how to use the commands:

```
COPY
BOARD_PROFILE_NAME = "controller"
DEVICE_NAME = ALL
DATA_FILE_NAME = "Rev. 2.0"
BOARD_CHECKSUM = 87FB
BEGIN_COMMAND
```

DEVICE_DATA_FILE = *"filename1","filename2","filename3","filename4"*
This command specifies up to four device data filenames. To use this command, you must have the Board Data Mode option set to One Data File per Device (in the board profile). You must also have one *filename* for each device in the previously selected device group. Finally, you must program one device group at a time, because you can only define one set of filenames with this command.

This command temporarily overrides, but does not modify, the Device Data File option in the board profile.

DEVICE_CRC = *XXXXXXXX* , *XXXXXXXX* , *XXXXXXXX* , *XXXXXXXX*
This command sets the device CRC value to *XXXXXXXX*, where *XXXXXXXX* is a hexadecimal number. You must have one CRC value for each device in the previously selected device group. This command temporarily overrides, but does not modify, the Device CRC Value option in the board profile. It also tells BoardSite to perform a CRC verify, ignoring whatever options are in the board profile. You must program one device group at a time, because you can only define one set of CRC values with this command.

DEVICE_CHECKSUM = *XXXXXXXX* , *XXXXXXXX* , *XXXXXXXX* , *XXXXXXXX*
This command sets the device checksum value to *XXXXXXXX*, where *XXXXXXXX* is a hexadecimal number. You must have one checksum value for each device in the previously selected device group. This command temporarily overrides, but does not modify, the Device Checksum Value option in the board profile. It also tells BoardSite to perform a checksum verify, ignoring whatever options are in the board profile. You must program one device group at a time, because you can only define one set of checksum values with this command.

The three preceding commands, DEVICE_DATA_FILE, DEVICE_CRC, and DEVICE_CHECKSUM, are typically used together. For example, assume that a board profile has two device groups, U1,U2 and U3,U4. To program this board using these commands you would write the following batch file:

```
COPY
BOARD_PROFILE_NAME = "profile1"

# Program first device group

DEVICE_NAME = "U1"
DEVICE_FILE_NAME = "U1 data","U2 data"
DEVICE_CRC = 3BF9015A,0028C3F1
DEVICE_CHECKSUM = 5A31DC4F,3A01C3A8
BEGIN_COMMAND

# Program second device group

DEVICE_NAME = "U3"
DEVICE_FILE_NAME = "U3 data","U4 data"
DEVICE_CRC = 59E3BA15,10D3193C
DEVICE_CHECKSUM = 4A415526,01411A3F
BEGIN_COMMAND
```

FILE_CREATE
This command creates a binary data file from a master board. It is identical to the Create option in the File command (see the preceding section, "File," for more information). Here is an example of the FILE_CREATE command:

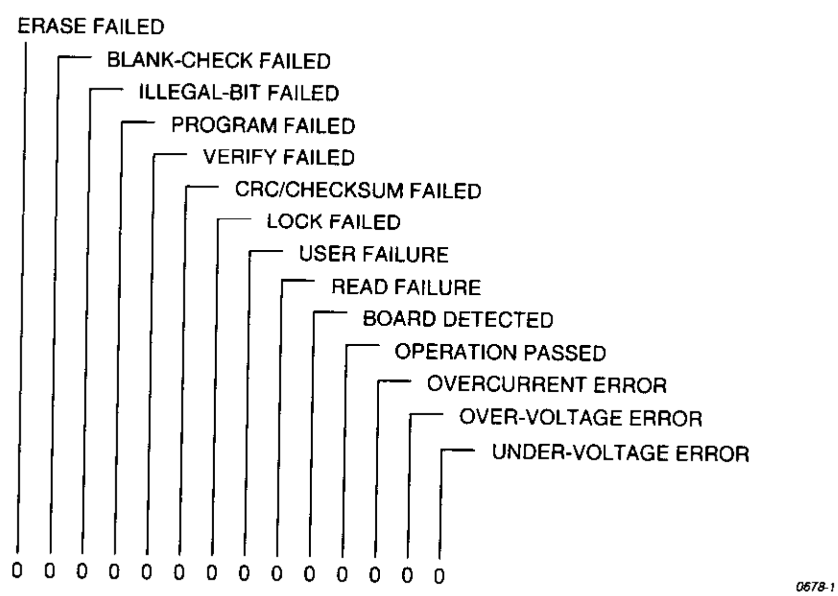
```
FILE_CREATE
BOARD_PROFILE_NAME = "profile1"
DEVICE_NAME = ALL
DATA_FILE_NAME = "Rev. 2.0"
BOARD_NUMBER = 1
BEGIN_COMMAND
```

If the file specified by the DATA_FILE_NAME exists, it will be overwritten.

TEST_OPTION = ***option***
This command selects the test options, where *option* can be BLANK_CHECK, ILLEGAL_BIT_CHECK, or BOTH.

SEND_BOARD_STATS
This command sends the board status words from the last programming operation. Each time you run a programming operation, BoardSite saves the status information in a status word. The information tells if a board was detected and if other errors occurred during the operation. BoardSite keeps an array of 32 status words, one for each of the 32 possible boards in a non-isolated adapter. The system sends the status words as 32 strings of ASCII decimal numbers, 14 digits in length, separated by a space. The batch processor sends 8 lines of 4 status words each. The status words correspond, in order, to the 32 possible memory boards. Each digit is either a 1 or a 0, corresponding to TRUE or FALSE. Figure 6-10 shows how to interpret each status word. Also, an example of this command is shown in the section, "Remote Batch Session Example," earlier in this chapter.

Figure 6-10
Status Word



SEND_ERROR_FILE
SEND_STATUS_FILE
These commands send the error message file or status message file to the host. These files contain the information displayed in the status and error windows during copy, verify, test, and checkout operations.

Miscellaneous Commands

CHECKOUT

This command performs a self-test of the BoardSite programming hardware, equivalent to selecting the Checkout command from the menu bar. To use the command, write the following batch file:

```
CHECKOUT
BEGIN_COMMAND
```

Response: Sends PASS if the command runs without any errors, or FAIL if errors occur. If FAIL is sent, use the SEND_ERRORS command to read the error number.

DOS_COMMAND = "*command*"

This command runs a DOS command line in a window, similar to the DOS option in the File command from the menu bar. *Command* may be up to 50 characters long.

REBOOT

This command causes your PC to warm boot, just as if you had pressed Ctrl + Alt + Del.

7 Sequence Editor Reference

Introduction

With the Sequence Editor, you can modify and compile the sequence file, which contains algorithms for all BoardSite operations. The sequence file, combined with the parameters in the Board Profile, provide the complete programming description of your memory board.

BoardSite automatically creates a default sequence file whenever you create a new Board Profile or change an existing Board Profile. If your design doesn't require changes to the sequence file (most designs don't), you use the Sequence Editor only to compile the sequence file. Even if you make no changes to the sequence file, you still must compile the default sequence file. You cannot perform any board-related operations (Copy, Verify, Test, and so on) until you compile it.

Note: We must emphasize that, even if you make no changes to the sequence file, you still must compile and save the default sequence file that BoardSite creates automatically.

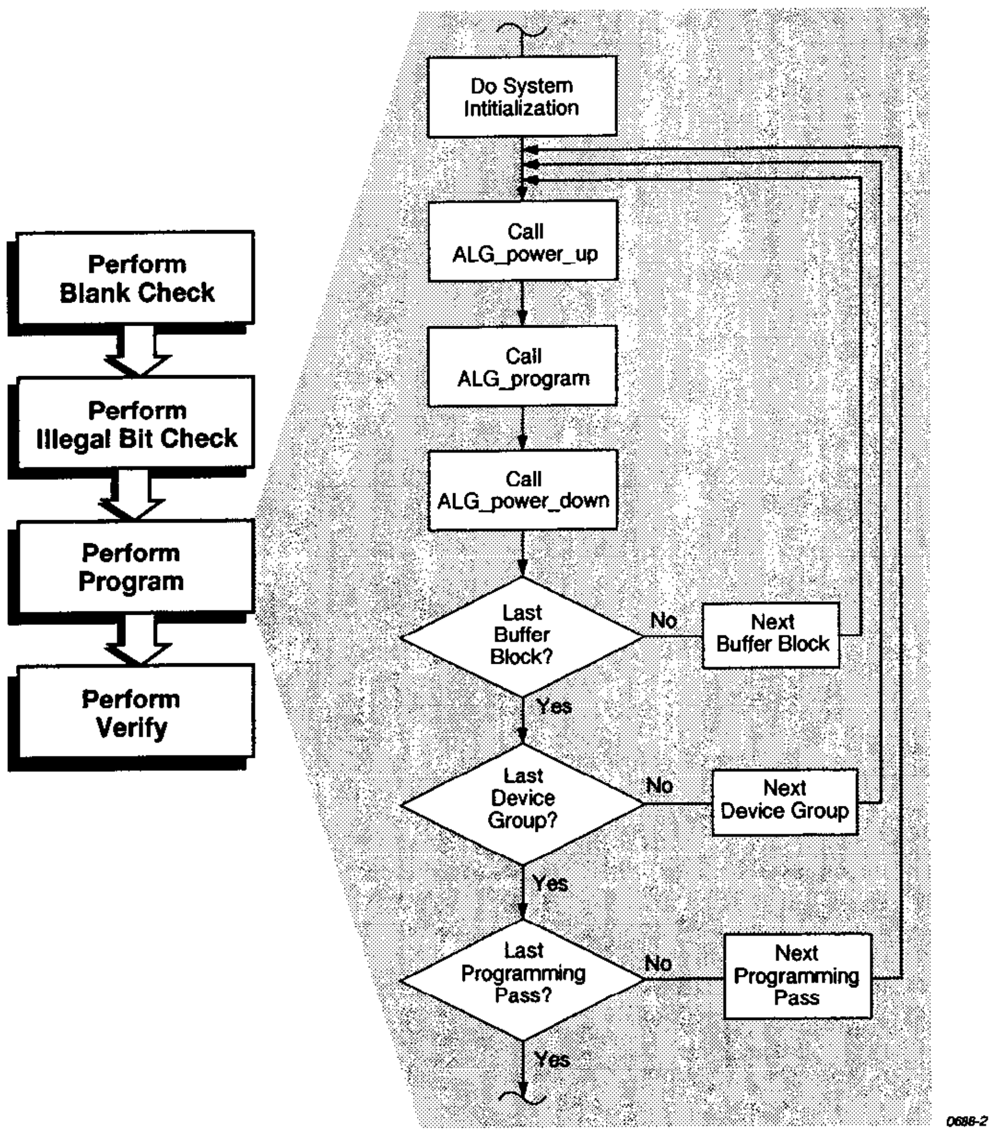
The Sequence Editor contains a specialized editor, a C compiler and linker, and an online help system. The C compiler and linker are commercial products, but they are called from the Sequence Editor Compile menu, and are transparent to you.

Understanding Algorithms, Sequences, and Primitives

Before you use the Sequence Editor, you should understand how BoardSite uses algorithms, sequences, and primitives.

Whenever you select a command that performs a board-related operation (for example, Copy or Verify), the BoardSite software calls a series of algorithms and control statements. Each algorithm has a specific function. For example, the algorithm ALG_power_up causes the BoardSite hardware to apply power to the board. Figure 7-1 shows a flow chart of the algorithms that are called when you run the Copy command to program a device.

Figure 7-1
Copy Command



When you compile the sequence file, BoardSite generates a different set of algorithms for each different device family code. For example, if your board contains an AMD 27C256 (family code 11D) and an Intel 27C010 (family code 5C), BoardSite generates two sets of algorithms. When BoardSite programs the AMD 27C256, it calls the algorithms for family code 11D, and when BoardSite programs the Intel 27C010, it calls the algorithms for family code 5C.

Although you cannot list the complete algorithm sequence shown in Figure 7-1, you can use the Sequence Editor to list the individual algorithms. There are several other algorithms that perform other board-related operations. As new devices are added to BoardSite, new algorithms may be added also, to support special device features. Table 7-1 shows a list of typical BoardSite algorithms.

Table 7-1
Typical BoardSite Algorithms

Algorithms
ALG_program
ALG_verify
ALG_read
ALG_power_up
ALG_power_down
ALG_blank_check
ALG_illegal_bit
ALG_lock
ALG_erase
ALG_leds
ALG_user

The algorithms are written in the C programming language. They consist of both C language statements and calls to BoardSite sequences. You cannot modify algorithms, but you can modify the sequences called by the algorithms. You can also create your own sequences that can be called by other sequences.

Your BoardSite software contains a set of predefined sequences. These sequences perform lower-level tasks such as presenting the current address to the interface connector, and applying a programming pulse. Table 7-2 lists typical sequences supplied with your BoardSite software.

Table 7-2
Typical BoardSite Sequences

Sequence Name
SEQ_output_address
SEQ_output_data
SEQ_program_pulse
SEQ_increment_address
SEQ_read_board
SEQ_power_up
SEQ_power_down
SEQ_enable_device
SEQ_disable_device
SEQ_clear_digital
SEQ_simultaneous_power_up
SEQ_enable_first_board
SEQ_enable_next_board
SEQ_enable_master_board
SEQ_leds_on
SEQ_user_instructions

Sequences are also written in C. Sequences contain C statements, calls to other sequences, and calls to primitives. Primitives are low-level instructions that control BoardSite's hardware. For example, the primitive data_enable enables the address buffers on the interface card. For a list of common primitives, see the section, "Summary of Primitives," later in this chapter.

Algorithm Example

Here is a simple example of a BoardSite algorithm. Assume you have chosen the 2764 EPROM in the Board Profile, and that your board has two control lines, ALE (address latch enable) and DLE (data latch enable), that need to be toggled to latch the address and data. Also, assume that the device has already been powered up by another algorithm.

```
ALG_program
{                                   /*start of algorithm*/
 while ( ADDRESS <= ADDRESS_END )/*repeat for all addresses*/
  {
   SEQ_output_address ();      /*call a Sequence*/
   SEQ_output_data ();         /*Sequence*/
   SEQ_apply_program_pulse (); /*Sequence*/
   SEQ_increment_address ();
  }                             /*end of current address*/
}                               /*end of algorithm*/
```

Notice that the algorithm contains C statements that control the algorithm flow, and calls to sequences. The algorithm also contains a reference to a BoardSite global variable, ADDRESS. For more information on BoardSite global variables, see the section, "BoardSite Global Variables," later in this chapter.

Here is the sequence SEQ_output_address that the algorithm calls.

```
SEQ_output_address ()          /*sequence name*/
{                              /*start of sequence*/
 address_set ( ADDRESS );      /*output the address-primitive*/
 control_bit_set( ALE,HI );    /*set control line high-primitive*/
 control_bit_set( ALE,LO );    /*set control line low-primitive*/
}                              /*end of sequence*/
```

Notice that this sequence calls the address_set primitive once and the control_bit_set primitive twice. The address_set primitive causes the BoardSite hardware to present the current value of the global variable ADDRESS to the interface connector. The control_bit_set primitive causes the BoardSite hardware to change the state of the ALE line. The first call to this primitive causes the ALE line to go high, and the second call causes the ALE line to go low.

The other sequences in ALG_program are similar to SEQ_output_address. In the section, "How to Modify a Sequence," you'll learn how to use the Sequence Editor to modify an existing sequence.

Default Algorithms

BoardSite provides default programming algorithms for all supported devices. The default algorithms usually support several different devices that use the same programming flow. These default algorithms are developed and tested on an evaluation circuit board, which contains a very basic hardware interface. Depending on your board design, you may have to modify the default algorithms.

The default algorithms usually have control pin aliases assigned to some of the BoardSite control lines (C0-C23). These control pin aliases correspond to signals on the evaluation circuit board. Some of the control pin aliases are described in Table 7-3.

Table 7-3
Control Pin Aliases for Default Algorithms

Control Line	Alias	Function
C0	CE	Chip Enable (where applicable)
C1	OE	Output Enable (where applicable)
C15	CTRL_READ_WRITE	Bidirectional control

Control pin aliases may be added or removed in future versions of the algorithms.

Using the Sequence Editor

This section contains information on the Sequence Editor screen, including how to select commands, how to use the sequence editing window, and how to get online help. For information on the commands, see the following section, "Sequence Editor Command Reference."

Sequence Editor Screen

When you start the Sequence Editor from BoardSite, the first screen you see looks like Figure 7-2. The screen contains the menu bar, file documentation window, and message line. The file documentation window, which occupies the entire window space (the window is "zoomed"), allows you to add documentation to your sequence file.

Figure 7-2
Sequence Editor Screen

```
File  Edit  Define  List  Compile  Window  Help  Miscellaneous
                         [File Documentation]
-------------------- Sequence File Documentation ----------------------
     Algorithm 0B7
     Families Supported - 0B7

     Created from Software Version - 2.00
     Copyright 1988, 1989, 1990  Data I/O Corporation

     Last modified:  2/14/90

     This algorithm supports self-timed EEPROMs needing no special
     pin routing. These type of devices require only WE, OE, and CE
     signals, as defined below.

        DEVICE        INTERFACE CONNECTOR

          WE           PGM control line.
          OE           C1 control line.
          CE           C0 control line.

     The algorithm in its original form contains specific calls to
     Primitives to program a Data I/O evaluation board.  These specifics
Help: F10  Menu: Alt-F10  File: demo_2816a  Sequences: in-line
```

You can also select an algorithm (see the section, "Sequence Editor Command Reference" for more information). When you do this, BoardSite removes the sequence file documentation window, and then splits the window space into two windows. The upper window contains the selected algorithm. This window is read-only, because you cannot modify algorithms.

The lower window is the sequence editing window. To display a sequence for editing, you can move the cursor in the algorithm window until the cursor is in a sequence call, or you can use the Select Sequence command from the Window menu. Figure 7-3 shows the Sequence Editor screen, with the ALG_program algorithm in the upper window, and the SEQ_output_address sequence in the lower window.

Figure 7-3
Sequence Editor Screen

```
File  Edit  Define  List  Compile  Window  Help  Miscellaneous
                          [Algorithm: ALG_program]
   *          Apply fixed program pulse.
   *          Get next address and data source word.
   *       END (address loop)
   *   END (ALG_PROGRAM)
   */

  /* start of programming algorithm */
  while (ADDRESS <= ADDRESS_END)                /* loop on all addresses */
  {
     SEQ_output_address();                      /* output the address    */
                                                                [READ ONLY]
                    [Sequence: SEQ_output_address]
SEQ_output_address ()
{
  /*  This Sequence outputs the current address.  It assumes a
   *  non-multiplexed address/data bus, and that the address bus is
   *  enabled.  For multiplexed busses, the user will need to add
   *  disabling the data bus, enabling the address bus, plus some
   *  control signal for latching the address, and then disabling the
   *  part of the address bus to be multiplexed with the data.
   */

Help: F10  Menu: Alt-F10  File: demo_2816a  Sequences: in-line
```

BoardSite gives you several commands to manipulate these windows. For complete information, see the section, "Sequence Editor Command Reference."

Pulling Down Menus and Choosing Commands

There are two ways to pull down menus from the menu bar:

1. Press [Alt] + [F10] to pull down the File menu. Then, use the [→] and [←] keys to pull down the other menus.
2. Or, press [Alt] and the first letter of the menu you want to pull down. For example, [Alt] + [F] pulls down the File menu, [Alt] + [E] pulls down the Edit menu, and so on.

When a menu is pulled down, there are two ways to select a command:

1. Use the [↑] and [↓] keys to highlight the command you want. Then press [↵] to select the command.
2. Or, press the first letter of the command to highlight it. Then press [↵] to select the command.

Most of the commands also have hot keys that can save you keystrokes. When you press the hot key, you select the command immediately, without using the menus. Table 7-4 shows the menus, commands, and hot keys for the Sequence Editor.

Table 7-4
Sequence Editor Commands

Menu/Command	Hot Key
File Menu	
Save File Print File	[Alt] + [P]
Save C Source	[Alt] + [V]
Exit Editor	[Alt] + [B]
Edit Menu	
Insert Sequence	[F1]
Insert Variable	[F3]
Insert Constant	[F5]
Insert Primitive	[F7]
Delete Line	[Alt] + [Y]
Delete Constant	
Delete Variable	
Delete Control Pin Alias	
Delete Sequence	
Edit Constant	
Define Menu	
Define Sequence	[F2]
Define Variable	[F4]
Define Constant	[F6]
Define Control Pin Alias	[Alt] + [J]
List Menu	
List Defined Constants	[Alt] + [F5]
List Defined Variables	[Alt] + [F3]
List Control Pin Aliases	[Alt] + [K]
List BoardSite Global Variables	

Table 7-4 (continued)
Sequence Editor Commands

Menu/Command	Hot Key
Compile Menu	
Compile & Link	F8
Next Error	F9
Sequence Expansion	Alt + U
External Objects	
External Libraries	
Window Menu	
Switch Window	Alt + N
Delete Window	Esc
Select Algorithm	Alt + A
Select Sequence	Alt + S
Zoom All Windows	
Edit Documentation	
Edit Globals	
Zoom/Unzoom Window	Ctrl + Z
Help Menu	
About This Editor	Alt + F9
Display Key Bindings	
About this Sequence File	
Show Hierarchy	
Miscellaneous Menu	
Find String	Alt + F1
Find Next String	Alt + F2
Copy to File	Ctrl + C
Copy from File	Ctrl + P
Cut to File	Ctrl + K

Using the Editing Windows

In addition to the hot keys and other Alt + **key** combinations, the keys shown in Table 7-5 can be used in the sequence editing and file documentation windows.

Table 7-5
Editing Window Keys

Press	To do this
↑ ↓	Move the cursor up or down one line
→ ←	Move the cursor right or left one character
Home	Move the cursor to beginning of line
End	Move the cursor to end of line
PgUp	Move the cursor one page up
PgDn	Move the cursor one page down
Tab	Insert four spaces in file
Del	Delete character at cursor
F10	Get context-sensitive help

Getting Online Help

The Sequence Editor includes a context-sensitive online help system. You can get several kinds of online help, depending on your current operation.

Window Help

If no menus are pulled down, or no pop-ups are displayed, press F10 to get information on the window containing the cursor.

Command Help

To get help on a command, highlight the command and then press F10 . If you don't remember which menu contains the command you want, refer to Table 7-4, or pull down each menu until you find the command.

Primitive Type Help

To get a description of what each primitive type does (address primitives, data primitives, and so on), select Insert Primitive from the Edit menu (or press F7). In the Primitive Type pop-up, highlight the primitive type on which you want help. Then press Alt + F10 .

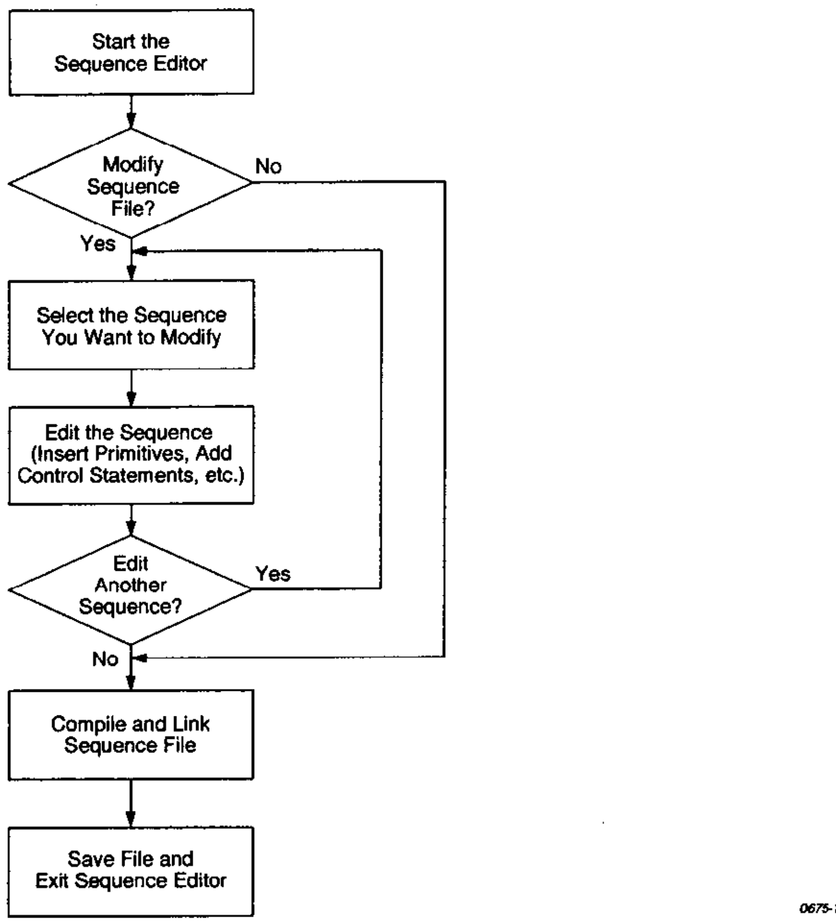
Specific Primitive Help

To get a detailed description of a primitive, including syntax and list of parameters, select Insert Primitive from the Edit menu (or press F7). In the Primitive Type pop-up, select the primitive type and press ↵ . In the Insert Primitive pop-up, highlight the primitive on which you want help. Then press Alt + F10 .

How to Modify a Sequence

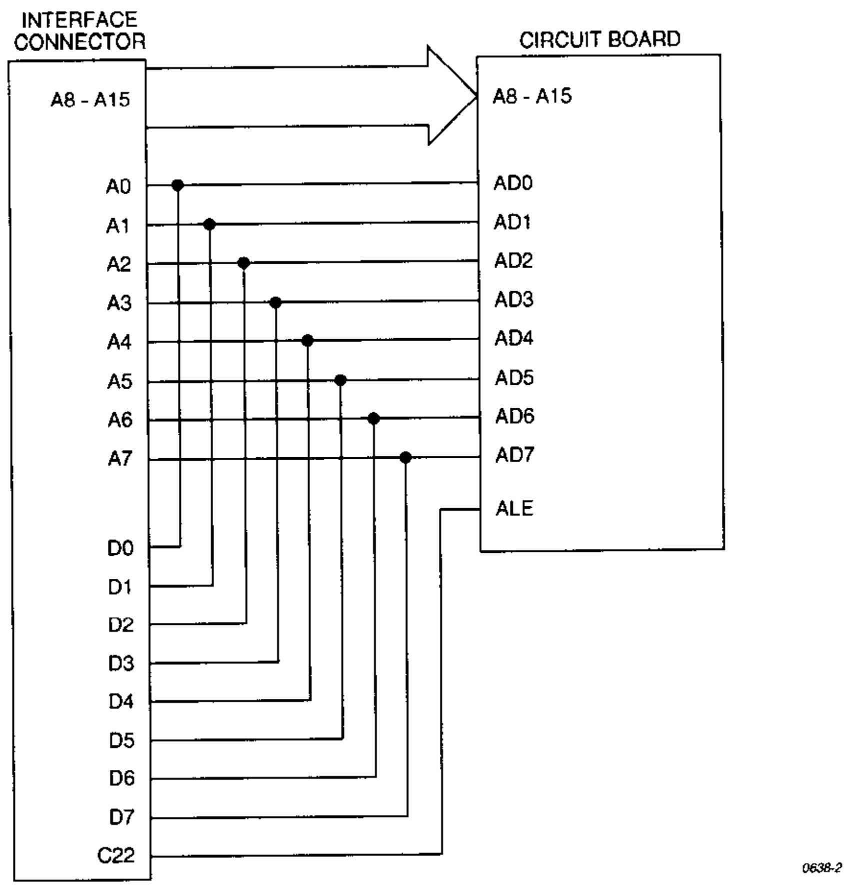
Occasionally, you may need to modify an existing BoardSite sequence. The following example shows you how to modify the sequences SEQ_output_address and SEQ_output_data to interface BoardSite to a memory board with a multiplexed address/data bus. Figure 7-4 shows the task flow for using the Sequence Editor.

Figure 7-4
Task Flow for Using the Sequence Editor



The multiplexed address/data bus board design is shown in Figure 7-5, and is also described in the section, "Design Rules for BoardSite Interface Signals," in Chapter 5.

Figure 7-5
SEQ_output_address in Editing Window



For more information on the Sequence Editor commands in the following example, see the section, "Sequence Editor Command Reference," in this chapter.

Note: *This example uses the sequence file called demo_2816a, which is supplied with your BoardSite software. If you used the software installation procedure described in Chapter 2, the demo_2816a file will be available to you, and you can follow the steps in this example.*

The example is divided into the following sections:

Start the Sequence Editor—open the sequence file called **demo_2816a.**

Select SEQ_output_address—display the first sequence and zoom the editing window.

Insert Primitives—insert additional primitives into the first sequence.

Select SEQ_output_data—display the second sequence and zoom the editing window.

Insert Primitive—insert an additional primitive into the second sequence.

Compile the Sequence File—compile the modified sequence file.

Save and Exit—save the new files and then exit the Sequence Editor.

Start the Sequence Editor

In the following steps, you start the Sequence Editor from the BoardSite menus. The BoardSite software must be running, and you must be at the top-level menu bar. You must also be in Manager Mode to use the Sequence Editor (see Chapter 3 for more information on entering Manager Mode).

1. Select the Edit command from the menu bar.
2. In the Edit Options pop-up, select Board Profile. Press [↵].
3. In the Edit File Name pop-up, select **demo_2816a**. Press [↵].
4. In the Edit Board Profile or Sequence File pop-up, select Sequence Editor. Press [↵].

The Sequence Editor takes a few seconds to load. When the Sequence Editor is finished loading, you should see the screen shown in Figure 7-6.

Figure 7-6
Sequence Editor Screen

```
File  Edit  Define  List  Compile  Window  Help  Miscellaneous
                          [File Documentation]
-------------------- Sequence File Documentation ----------------------
      Algorithm 0B7
      Families Supported - 0B7

      Created from Software Version - 2.00
      Copyright 1988, 1989, 1990  Data I/O Corporation

      Last modified:  2/14/90

      This algorithm supports self-timed EEPROMs needing no special
      pin routing. These type of devices require only WE, OE, and CE
      signals, as defined below.

         DEVICE        INTERFACE CONNECTOR

           WE           PGM control line.
           OE           C1 control line.
           CE           C0 control line.

      The algorithm in its original form contains specific calls to
      Primitives to program a Data I/O evaluation board.  These specifics
Help: F10  Menu: Alt-F10  File: demo_2816a  Sequences: in-line
```

Select SEQ_output_address

In the following steps, you select the first sequence you want to modify, and then zoom the editing window.

1. From the Window menu, choose Select Sequence ([Alt] + [S]).
2. In the Select Sequence pop-up, select SEQ_output_address. Press [↵].
3. From the Window menu, select Window Zoom/Unzoom ([Ctrl] + [Z]).

You should see the first sequence in the editing window, as shown in Figure 7-7.

Figure 7-7
SEQ_output_address in Editing Window

```
File  Edit  Define  List  Compile  Window  Help  Miscellaneous
                  [Sequence: SEQ_output_address]
SEQ_output_address ()
{
   /*  This Sequence outputs the current address.  It assumes a
    *  non-multiplexed address/data bus, and that the address bus is
    *  enabled.  For multiplexed busses, the user will need to add
    *  disabling the data bus, enabling the address bus, plus some
    *  control signal for latching the address, and then disabling the
    *  part of the address bus to be multiplexed with the data.
    */

   address_set(ADDRESS);        /* output the current address */
}

Help: F10  Menu: Alt-F10  File: demo_2816a  Sequences: in-line
```

Insert Primitives

In the following steps, you add several primitives to SEQ_output_address. These additional primitives tell the BoardSite hardware how to control the multiplexed address/data bus.

1. Move the cursor to the blank line above the line that contains the primitive call address_set(ADDRESS). Make sure the cursor is at the beginning of the line.
2. From the Edit menu, select Insert Primitive ([F7]).
3. In the Primitive Type pop-up, select Data Primitives. Press [↵] .
4. In the Insert Primitive pop-up, select data_disable. Press [↵] .

   The Sequence Editor inserts the data_disable primitive into the file.
5. In the Parameter List pop-up, select DATA_EN_D0_7. Press [↵] .

   The Sequence Editor automatically inserts the bus segment parameter DATA_EN_D0_7 into the primitive, completes the primitive, and moves the cursor to the end of the primitive.
6. Press [↵] to insert a blank line.
7. From the Edit menu, select Insert Primitive ([F7]).
8. In the Primitive Type pop-up, select Address Primitives. Press [↵] .
9. In the Insert Primitive pop-up, select address_enable. Press [↵] .

10. In the Parameter List pop-up, select ADDR_EN_A0_7. Press [↵] . The Sequence Editor automatically inserts the bus segment parameter ADDR_EN_A0_7 into the primitive, completes the primitive, and moves the cursor to the end of the primitive.
11. Move the cursor to the beginning of the line after the line containing the primitive call address_set(ADDRESS). Press [Home] to make sure the underscore is under the right-brace character.
12. Press [↵] to insert a blank line. Move the cursor up one line (into the blank line).
13. From the Edit menu, select Insert Primitive ([F7]).
14. In the Primitive Type pop-up, select Control Primitives. Press [↵] .
15. In the Insert Primitive pop-up, select control_bit_set. Press [↵] .

Note: You may have to scroll the list in the pop-up (press ***PgDn*** *) so that the control_bit_set primitive appears.*

16. In the Parameter List pop-up, select ADDR_DATA_MUX. Press [↵] .
17. In the Parameter List pop-up, select HI. Press [↵] .

    Notice that BoardSite automatically prompts you for every parameter required by the primitive.
18. Press [↵] to insert another blank line.
19. Using the procedures you learned above, add the remaining two primitives, control_bit_set(ADDR_DATA_MUX,LO) and address_disable(ADDR_EN_A0_7).
20. Use the editor to make the file more readable by inserting tabs and adding comments. When you're finished, the file should look like Figure 7-8.

Figure 7-8
Modified Sequence

```
File  Edit  Define  List  Compile  Window  Help  Miscellaneous
                    [Sequence: SEQ_output_address]
{
   /*  This Sequence outputs the current address.  It assumes a
    *  non-multiplexed address/data bus, and that the address bus is
    *  enabled.  For multiplexed busses, the user will need to add
    *  disabling the data bus, enabling the address bus, plus some
    *  control signal for latching the address, and then disabling the
    *  part of the address bus to be multiplexed with the data.
    */
    /* disable data bus and enable address bus */
    data_disable(DATA_EN_D0_7);
    address_enable(ADDR_EN_A0_7);

    address_set(ADDRESS);         /* output the current address */

    /* toggle the ALE line (control pin alias is ADDR_DATA_MUX) */
    control_bit_set(ADDR_DATA_MUX,HI);
    control_bit_set(ADDR_DATA_MUX,LO);

    /* disable the address bus */
    address_enable(ADDR_EN_A0_7);
}
Help: F10  Menu: Alt-F10  File: demo_2816a  Sequences: in-line
```

Select SEQ_output_data

In the following steps, you select the next sequence you want to modify, and then zoom the editing window.

1. From the Window menu, choose Select Sequence ([Alt] + [S]).
2. In the Select Sequence pop-up, select SEQ_output_data. Press [↵] .
3. From the Window menu, select Window Zoom/Unzoom ([Ctrl] + [Z]).

You should see SEQ_output_data in the editing window.

Insert Primitive

In the following steps, you add a primitive to SEQ_output_data.

1. Move the cursor to the blank line above the line that contains the primitive call data_set_from_source(). Make sure the underscore is at the beginning of the line.
2. From the Edit menu, select Insert Primitive ([F7]).
3. In the Primitive Type pop-up, select Data Primitives. Press [↵] .
4. In the Insert Primitive pop-up, select data_enable. Press [↵] .
5. In the Parameter List pop-up, select DATA_EN_D0_7. Press [↵] .
6. Use the editor to make the file more readable by inserting tabs and adding comments.

This completes the modification of the sequence file.

Compile the Sequence File

Before you leave the editor, you must compile the modified sequence file.

1. From the Compile menu, select Compile and Link ([F8]). Press [↵] .

   The compiler and linker are automatically invoked. The compile and link operations take a few minutes to complete; when they're done, you should see the screen shown in Figure 7-9.

Figure 7-9
Screen After Successful Compile and Link

```
File  Edit  Define  List  Compile  Window  Help  Miscellaneous
                          [Sequence: SEQ_output_data]
SEQ_output_data ()
{
   /* This Sequence outputs the current buffer data.  It assumes a
    * non-multiplexed address/data bus, and that the data bus is
    * enabled.  For a multiplexed bus, the user will need to enable
    * the bus first, and supply any control lines, if needed.
    */

   /* enable data bus *
   data_enable(DATA_EN_  Message
                         No Compile Error Found
   data_set_from_source  Press Any Key to Continue  buffer data */
}

COMPILING: Press Escape to Abort after Current Pass
```

Save and Exit

Finally, you save the new files and exit the Sequence Editor.

1. From the File menu, select Save File.

   BoardSite saves the sequence file under the original filename, **demo_2816a**.

2. From the File menu, select Exit Editor (Alt + B).

3. Because you made changes to the sequence file, BoardSite displays a message. Press any key (other than Esc or N) to save files and exit the editor.

   After a few seconds, the BoardSite main menus appear again.

Sequence Editor Command Reference

The following reference contains all the commands listed in Table 7-4. The reference is arranged by menu and then by command, in the order they appear in the Sequence Editor. If a command has a hot key, it precedes the command description.

File Menu

Save File — Alt + S — Save the current sequence file.

Print File — Alt + P — Send the current sequence file to the LPT1 port. The output is straight ASCII, with no printer formatting codes.

Save C Source — Alt + V — The file created during the compile and link operation is normally deleted by the system. To save the C source file created during compilation, use this command.

Exit Editor — Alt + B — Exit the editor and return to the BoardSite main menu. If you made changes to the file, BoardSite displays a message asking you whether you want to exit and lose changes, exit and save changes, or return to the editor.

Edit Menu

Insert Sequence — F1 — Insert a previously-defined sequence at the current cursor position. To define a new sequence, use the Define Sequence command from the Define menu.

Note: If you want to insert a sequence, you **must** *use the Insert Sequence command. Simply typing the sequence will not work, and will cause an error that prevents the sequence file from being compiled.*

Insert Variable — F3 — Insert a previously-defined variable at the current cursor position. To define a new variable, use the Define Variable command from the Define menu.

Insert Constant	F5	Insert a previously-defined constant at the current cursor position. To define a new constant, use the Define Constant command from the Define menu.
Insert Primitive	F7	Insert a primitive at the current cursor position.
Delete Line	Alt + Y	Delete the line that contains the cursor.
Delete Constant		Delete the constant at the current cursor position.
Delete Variable		Delete the variable at the current cursor position.
Delete Control Pin Alias		Delete the control pin alias at the current cursor position.
Delete Sequence		Delete the sequence at the current cursor position.
Edit Constant		Edit the value of a defined constant.

Define Menu

Define Sequence F2

Define a new sequence. Sequences can contain C statements, primitives, and calls to other sequences. A sequence cannot call itself.

When you select this command, BoardSite displays a pop-up for the sequence name. BoardSite automatically appends the prefix SEQ_ to the sequence name.

After you type the name, BoardSite displays another pop-up. In this pop-up, you can tell BoardSite to make this sequence a function. When a sequence is defined as a function, BoardSite does not expand the sequence as separate instructions in the sequence file. The sequence name also becomes FNC_xxxxx instead of SEQ_xxxxx.

Functions may run more slowly, but they save a significant amount of memory. If you create a sequence with many lines of code, or if you experience memory problems during compilation, you should define the sequence as a function.

Define Variable F4

Define a new variable. Variables can be created to store values needed for special applications. A typical application is to read a status bit or byte, save it in a defined variable, and then perform some test on the value to control program flow.

When you select this command, BoardSite displays a pop-up for the sequence name and the variable type. BoardSite automatically appends the prefix USV_ to the variable name, to prevent name conflicts with existing variables. The name may be up to 32 characters long.

The variable type may be any one of the following:

char	8 bits, signed
int	16 bits, signed
long	32 bits, signed
unsigned char	8 bits, unsigned
unsigned int	16 bits, unsigned
unsigned long	32 bits, unsigned

Define Constant [F6] Define a new constant. Constants are names used in place of numeric values.

When you select this command, BoardSite displays a pop-up for the sequence name and the constant type. BoardSite automatically appends the prefix USC_ to the constant name, to prevent name conflicts with existing constants. The name may be up to 32 characters long. The constant type may be any one of the types listed in the Define Variable command above.

Define Control Pin Alias [Alt] + [J] Define a new control pin name, and attach it to one of the control pin signals (C0-C23).

When you select this command, BoardSite prompts for the control pin alias name, which can be up to 32 characters long. After you type the name, a list of the default names and their bit position will appear in a pop-up. In this pop-up, select the control pin to which you want the name attached.

List Menu

List Defined Constants [Alt] + [F5] List defined constants and their values.

List Defined Variables [Alt] + [F3] List defined variables and their types.

List Control Pin Aliases [Alt] + [K] List control pins and their alias names.

List BoardSite Global Variables List all BoardSite global variables. The list contains the name, type, and a brief explanation of each global variable. For more information, see the section, "BoardSite Global Variables," later in this chapter.

Compile Menu

Compile & Link [F8] Compile and link the sequence file.

Note: Temporary files generated by the compiler and linker may be redirected to another drive and/or directory. If your PC has a RAM disk, you can speed up compiling and linking by redirecting the temporary files to the RAM disk.

Specify your temporary files directory by defining a DOS environment variable TMP in your AUTOEXEC.BAT file. For example, if your D: drive is your RAM disk, place the following command in your AUTOEXEC.BAT file.

set TMP=d:

If you do not specify a TMP variable, the BoardSite software will default to the current working directory.

If the BoardSite cannot access the drive and/or directory in the TMP variable, you will receive Error Message 67, and the BoardSite software will terminate.

Next Error	F9	Single-step through each error found by the compiler or linker.
Sequence Expansion	Alt + U	Output sequences as functions instead of expanding sequence as in-line code. The default mode is to create sequences as in-line code. If you are experiencing any "Out of Memory" errors when you compile the sequence file, you should select this command. You can toggle between in-line code and functions by pressing the hot keys again or by choosing this command again. Depending on the type of algorithm, board operations will run approximately 2-10% slower when you output sequences as functions.
External Objects		This command allows you to specify an external object file that you wish to have linked with your Algorithm Sequence file. For example, you may wish to link in a proprietary encryption algorithm, or a bar-code reader routine.
External Libraries		This command allows you to specify an external library file that you wish to have linked with your Algorithm Sequence file. For example, you may wish to link in a library containing routines for a bar-code reader.

Window Menu

Switch Window	Alt + N	Switch the active window between upper and lower windows. The active window has a double-line border.
Delete Window	Esc	If windows are overlapped, delete the current window to reveal the one below it.
Select Algorithm	Alt + A	Select the algorithm to display in the upper window. When you select this command, a list of available algorithms appears in a pop-up. There is a special type of algorithm, the user algorithm (ALG_user), which you can use to further customize a Board Profile. You can customize the user algorithm by editing the SEQ_user_instruction sequence.

An example of a user algorithm application is to define a sequence of primitives to halt a microprocessor on a memory board. The instructions would be inserted into the sequence (SEQ_user_instruction) called by the user algorithm.

Select Sequence	[Alt] + [S]	Select a sequence for editing.
Zoom All Windows		Automatically zoom the active window to full size.
Edit Documentation		Edit the file documentation. The file documentation appears in its own window. The first and last lines within the documentation block are write-protected. You may add, delete, or modify any of the lines in-between.
Edit Globals		Edit the global variable block. The global variable block provides support for more advanced "C" constructs (i.e., pointer variables, data structures, and include files).
Zoom/Unzoom Window	[Ctrl] + [Z]	Enlarge the active window to full size, or reduce it to half size. The Zoom/Unzoom Window command is available only if you don't select the Zoom All Windows command.

Help Menu

About This Editor	[Alt] + [F9]	Display information about the Sequence Editor.
Display Key Bindings		Display a complete list of the hot keys. Use hot keys to eliminate keystrokes and speed up your work.
About this Sequence File		Display the static values contained in the system variables in your sequence file.
Show Hierarchy		Show which sequences are called by each algorithm.

Miscellaneous Menu

Find String	[Alt] + [F1]	Find a string of characters. You type the string in a pop-up.
Find Next String	[Alt] + [F2]	Find the next occurrence of the string defined in the find string command.
Copy to File	[Ctrl] + [C]	Copy marked lines to a file.
Copy from File	[Ctrl] + [P]	Paste lines into the sequence file from another file.
Cut to File	[Ctrl] + [K]	Copy marked lines to a file, then delete them from the sequence file.

C Language Summary

Here is a summary of common C language expressions used in the Sequence Editor. For a more detailed description of the C language, refer to the paperback book included with your system, *The C Programming Language*, by Kernighan and Ritchie.

Statement Formats

```
/* comment */
```

/* comments are enclosed within slash-star and star-slash */

```
x = 1;
```

Simple statement terminated with semicolon

```
{
   tmp = a;
   a = b;
   b = tmp;
}
```

Multiple simple statements are enclosed within braces. Indentations used for clarity.

```
if (a < 0) a = b;
else a = c;
```

Perform statement if condition is true. Optional else after if statement

```
while (a < MAX) a = b;
```

Perform statement while condition is true. Test done at top of loop.

```
do a = b;
while (a < MAX);
```

Perform statement until condition is false. Test done at bottom of loop.

```
for (i=0; i<MAX; i++)
a[i] = 0;
```

Perform initialization once, then do statement and increment while condition is true. Test done at top of loop. Equivalent to:

```
i = 0;
while (i < MAX)
{
  a[i] = 0;
  i++;
}
```

```
switch (a)
{
  case 5: a = b;
  case 10: a = c;
  break;
  case 15: b = c;
  break;
  default: break;
}
```

Evaluate expression (value of a) and go to appropriate case statement.

default: Default if no case matches

```
break;
```

Terminate smallest enclosing WHILE, DO, FOR, or SWITCH

```
continue;
```

Go to bottom of loop in WHILE, DO, and FOR

Special Characters (for strings)

<table>
<tr><td>'\n'</td><td>New line</td></tr>
<tr><td>'\r'</td><td>Carriage return</td></tr>
<tr><td>'\t'</td><td>Tab</td></tr>
<tr><td>'\f'</td><td>Form feed</td></tr>
</table>

Constants

<table>
<tr><td>1234</td><td>Decimal number</td></tr>
<tr><td>1234L</td><td>Long decimal number</td></tr>
<tr><td>0xAA55</td><td>Hex number</td></tr>
<tr><td>0xAA55L</td><td>Long hex number</td></tr>
<tr><td>0177</td><td>Octal number</td></tr>
<tr><td>0177L</td><td>Long octal number</td></tr>
<tr><td>32.5</td><td>Float number</td></tr>
<tr><td>1.2e-5</td><td>Scientific notation</td></tr>
<tr><td>'a'</td><td>Character</td></tr>
<tr><td>"abcd"</td><td>Null terminated string</td></tr>
</table>

Variable Declarations

<table>
<tr><td>char a;</td><td>Signed 8-bit</td></tr>
<tr><td>int i, j;</td><td>Signed integers, 16-bit</td></tr>
<tr><td>long c;</td><td>Signed long, 32-bit</td></tr>
<tr><td>unsigned char b;</td><td>Unsigned, 8-bit</td></tr>
<tr><td>unsigned int i;</td><td>Unsigned integer, 16-bit</td></tr>
<tr><td>unsigned long d;</td><td>Unsigned long, 32-bit</td></tr>
</table>

Operators

<table>
<tr><td>-a</td><td>Negate</td></tr>
<tr><td>!a</td><td>2's complement</td></tr>
<tr><td>~a</td><td>1's complement</td></tr>
<tr><td>a++</td><td>Increment</td></tr>
<tr><td>a--</td><td>Decrement</td></tr>
<tr><td>a*b</td><td>Multiply</td></tr>
<tr><td>a/b</td><td>Divide</td></tr>
<tr><td>a%b</td><td>Modulus</td></tr>
<tr><td>a+b</td><td>Add</td></tr>
<tr><td>a-b</td><td>Subtract</td></tr>
<tr><td>a>>n</td><td>Right shift by n bits</td></tr>
<tr><td>a<<n</td><td>Left shift by n bits</td></tr>
<tr><td>a&b</td><td>Bitwise and</td></tr>
<tr><td>a|b</td><td>Bitwise or</td></tr>
<tr><td>a^b</td><td>Bitwise xor</td></tr>
</table>

Logical Operators

<table>
<tr><td>a<b</td><td>Less than</td></tr>
<tr><td>a>b</td><td>Greater than</td></tr>
<tr><td>a<=b</td><td>Less than or equal</td></tr>
<tr><td>a>=b</td><td>Greater than or equal</td></tr>
<tr><td>a==b</td><td>Equal to</td></tr>
<tr><td>a!=b</td><td>Not equal to</td></tr>
<tr><td>a&&b</td><td>Logical and</td></tr>
<tr><td>a||b</td><td>Logical or</td></tr>
</table>

Functions

name (value);	Call to a function passing optional parameter

name (value)
int value;
{
a = b;
}
	Function defined with optional parameter. Braces enclose body of function

Summary of Primitives

Sequences include primitives, which control the BoardSite hardware, and C statements, which control the program (sequence) flow. This section summarizes the primitives available in the Sequence Editor. To get complete information on primitives, use the online help system in the Sequence Editor, as described in the next section.

You can insert any primitive into a sequence by pressing F7, or by choosing Insert Primitive from the Edit menu.

To Get Online Help for any Primitive

You can get online help for any BoardSite primitive. The help information contains the primitive syntax, purpose of the primitive, and a brief explanation of its use.

To get online help for any primitive:

1. Select Insert Primitive from the Edit menu (or press F7).
2. In the Primitive Type pop-up, select the primitive type. Press ↵.
3. In the Insert Primitive pop-up, highlight the primitive on which you want help.
4. Press Alt + F10.

Address Primitives

Use the address primitives to set, enable, or disable the address bus drivers (A0-A31).

```
address_set(address_value)
address_enable(bus_position)
address_disable(bus_position)
quick_addr_set(address_value)
short_addr_set(address_value)
```

Programmable Chip Enable Primitives

Use the programmable chip enable primitives to set, enable, or disable the programmable chip enable drivers (PCE0-PCE15).

```
pce_set(on_off)
pce_enable()
pce_disable()
```

Board Enable Primitives

Use the board enable primitives to set, enable, or disable the board enable drivers (BE0-BE7).

be_enable()
be_disable()
enable_all_boards()
disable_all_boards()
enable_master_board()
disable_master_board()
enable_first_board()
enable_next_board()

Data Primitives

Use the data primitives to set, enable, or disable the data bus drivers (D0-D31), or to read data from D0-D31.

data_set(data_value)
data_set_from_source()
data_get_from_source(variable)
data_read(variable)
data_read_to_buffer()
compare(value,compare_value)
compare_to_source(value)
compare_for_illegal_bit(value)
compare_for_blank_check(value)
compare_device_blank(value)
compare_device_source(value)
compare_to_blank_state(value,array)
compare_to_illegal_bit_state(value,array)
data_enable(bus_position)
data_disable(bus_position)
next_source_word()
last_source_word()
set_source_word(src_value)

Control Primitives

Use the control primitives to set, enable, or disable the control bus drivers (C0-C23), or to read the control bus.

control_bit_set(signal,level)
control_bitset_fnc(signal,level)
control_port_set(bus_position,control_word)
control_enable(bus_position)
control_disable(bus_position)
control_read(bus_position,variable)
control_bit_compare(bit_position,level)
user_clock_frequency_set(frequency)
pgm_clocks_enable()
pgm_clocks_disable()
pgm_set(state)
pgm_delay()
pulse_pgm(pulse_delay)
pulse_pce(pulse_delay)
pulse_control_bit(control_bit,pulse_delay)
toggle_control_bit(control_bit)
delay(delay_duration)
timer_set_pgm(pulse_duration)
timer_set_pgm_scalar(pulse_duration)

Programmable Logic Primitives

These primitives are only used in sequences that involve programmable logic devices.

get_logic_word(logval,dev_num,word_num)
store_logic_word(srcval,dev_num,word_num)
next_logic_word(dev_num,num_words)
last_logic_word(dev_num,num_words)
compare_to_logic_source(logic_value,dev_num,word_num)
brd_logic_error(error_type)

Status/Debug Primitives

Use the status/debug primitives to enhance debugging and report status.

break_point(breakpoint_str)
debug_get_key()
debug_display_message(debug_str)
status_display_message(status_str)
error_display_message(error_str)
brd_error(error_type)
fatal_error(error_code)
move_action_symbol()
led enable()
led_disable()
get_variable_value(variable_name,store_value)

Power Primitives

Use the power primitives to control BoardSite's programmable power supplies.

vpp1_pre_charge(value)
vpp2_pre_charge(value)
vcc1_on(value)
vcc2_on(value)
vpp1_on (value)
vpp2_on(value)
vpp1lo_on(value)
vpp2lo_on(value)
vneg_on(value)
vcc1_off()
vcc2_off()
vpp1_off()
vpp2_off()
vneg_off()
vcc1_preset(value)
vcc2_preset(value)
vpp1_preset(value)
vpp2_preset(value)
vpp1lo_preset(value)
vpp2lo_preset(value)
vneg_preset(value)
vp12_preset()
vm12_preset()
power_up_supplies()
power_down_supplies()
load_preset_supplies
plus_12V(switch)
minus_12V(switch)
vpp_pgm_mux(vpp_supply,new_state)
vpp_hi_lo(vpp_supply,level)
vpp1lo_set(value)
vpp2lo_set(value)
vcc1_ramp_up(start_value,end_value,step_value,ramp_duration)
vcc2_ramp_up(start_value,end_value,step_value,ramp_duration)
vpp1_ramp_up(start_value,end_value,step_value,ramp_duration)
vpp2_ramp_up(start_value,end_value,step_value,ramp_duration)

simultaneous_vcc_ramp_up(start_value,end_value,step_value,ramp_duration)
simultaneous_vpp_ramp_up(start_value,end_value,step_value,ramp_duration)
simultaneous_supply_ramp_up(start_value,end_value,step_value,ramp_duration)
vcc1_ramp_down(start_value,end_value,step_value,ramp_duration)
vcc2_ramp_down(start_value,end_value,step_value,ramp_duration)
vpp1_ramp_down(start_value,end_value,step_value,ramp_duration)
vpp2_ramp_down(start_value,end_value,step_value,ramp_duration)
simultaneous_vcc_ramp_down(start_value,end_value,step_value,ramp_duration)
simultaneous_vpp_ramp_down(start_value,end_value,step_value,ramp_duration)
simultaneous_supply_ramp_down(start_value,end_value,step_value,ramp_duration)

BoardSite Global Variables

BoardSite global variables are used to pass data between algorithms. For example, the global variable ADDRESS is a variable that represents the current address. It could be used as the test value in a loop, as shown in the third line of the following algorithm:

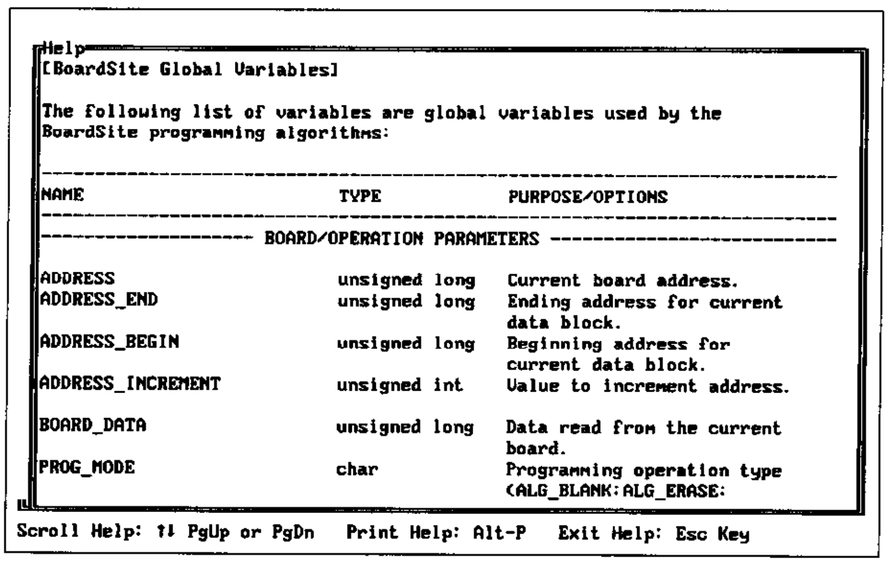
```
ALG_program
{                                               /*start of algorithm*/
  while ( ADDRESS <= ADDRESS_END )              /*repeat for all addresses*/
  {
    SEQ_output_address ();                      /*call a Sequence*/
    SEQ_output_data ();                         /*Sequence*/
    SEQ_apply_program_pulse ();                 /*Sequence*/
    SEQ_increment_address ();
  }                                             /*end of current address*/
}                                               /*end of algorithm*/
```

When you use the Sequence Editor, you can get online help about the global variables. The help screen includes the variable name, the variable type, and a description.

To Get Help on Global Variables

1. In the Sequence Editor, select List BoardSite Global Variables from the List menu.
2. Press [↵].
3. Press [Ctrl] + [Z] to zoom the help window. You should see the help window shown in Figure 7-10.
4. Press [Esc] to remove the help window.

Figure 7-10
List Global Variable Help Window



```
Help
[BoardSite Global Variables]

The following list of variables are global variables used by the
BoardSite programming algorithms:

---------------------------------------------------------------------------
NAME                           TYPE             PURPOSE/OPTIONS
---------------------------------------------------------------------------
------------------- BOARD/OPERATION PARAMETERS -----------------------------

ADDRESS                        unsigned long    Current board address.
ADDRESS_END                    unsigned long    Ending address for current
                                                data block.
ADDRESS_BEGIN                  unsigned long    Beginning address for
                                                current data block.
ADDRESS_INCREMENT              unsigned int     Value to increment address.

BOARD_DATA                     unsigned long    Data read from the current
                                                board.
PROG_MODE                      char             Programming operation type
                                                (ALG_BLANK:ALG_ERASE:
Scroll Help: ↑↓ PgUp or PgDn   Print Help: Alt-P   Exit Help: Esc Key
```

Summary of Constants

Table 7-6 lists constants commonly used in the BoardSite algorithms. For example, the constant DATA_EN_D0_7 can be used in the data primitive data_enable(DATA_EN_D0_7) to enable the data drivers for D0 through D7.

Table 7-6
Constants

Constant	Description
TRUE FALSE	Logical-valued constants
DATA_EN_D0_7 DATA_EN_D0_15 DATA_EN_D0_23 DATA_EN_D0_31 DATA_EN_D8_15 DATA_EN_D16_23 DATA_EN_D24_D31	Bus segment constants used in the data primitives. For example, data_enable(DATA_EN_D0_7).
ADDR_EN_A0_7 ADDR_EN_A0_15 ADDR_EN_A0_23 ADDR_EN_A0_31 ADDR_EN_A8_15 ADDR_EN_A16_23 ADDR_EN_A24_31	Bus segment constants used in the address primitives. For example, address_enable(ADDR_EN_A0_7).
C0_C7 C8_C15 C16_C23 C0_C23	Bus segment constants used in the control primitives. For example, control_enable(C0_C7).
HI LO	Control bit output constants used to drive the control lines C0-C23. For example, control_bit_set(C0,HI).
ACTIVE INACTIVE	Output constants for signals with selectable active states, such as the PGM line. For example, pgm_set(ACTIVE).
NORMAL_PULSE SCALAR_PULSE	Programming pulse scaling constants. For example, pulse_pgm(NORMAL_PULSE).
U1MHZ U2MHZ U4MHZ U8MHZ	User clock frequency constants. For example, user_clock_frequency_set(U8MHZ).
ON OFF	Power on/off constants. For example, plus_12v(ON).
VPP1 VPP2 TO_LO_LEVEL TO_HI_LEVEL	VPP switching constants. For example, vpp_hi_lo(VPP1,TO_HI_LEVEL).
FAIL_BLANK FAIL_CRC_CHK FAIL_ERASE FAIL_ILLEGAL_BIT FAIL_LOCK FAIL_PROGRAM FAIL_READ FAIL_USER FAIL_VERIFY	Error code constants. For example, fatal_error(FAIL_BLANK).

8 Messages

This chapter lists all of BoardSite's warning and error messages. Each message has a number, and these numbers correspond to the categories in the following table.

Table 8-1
Message Categories

Message Number	Category
1 – 99	General Operator Interface Messages
100 – 199	(Reserved for future expansion)
200 – 399	Programming Messages
400 – 499	Batch Command Messages
500 – 599	Binary Data Editor Messages
600 – 699	Board Profile Messages
700 – 799	Communications Messages
800 – 999	Diagnostic Messages

General Operator Interface Messages

`ERROR 1: Unable to find Board Data File requested`

The data file was deleted by a program other than BoardSite, or the disk or catalog file has been corrupted. To correct this, use the Delete command from the File menu to remove the name from the catalog.

`ERROR 2: Error occurred while trying to read Board Data File`

The disk or data file has been corrupted.

`ERROR 3: Error occurred while trying to write Board Data File`

A disk error occurred while trying to write a data file to disk. The disk is full or corrupted.

ERROR 4: `Unable to create Board Profile`

BoardSite was unable to create a Board Profile. The disk or catalog file has been corrupted.

ERROR 5: `Unable to open Board Profile`

The Profile was deleted by a program other than BoardSite, or the disk or catalog file has been corrupted. To correct this, use the Delete command from the File menu to remove the name from the catalog.

ERROR 6: `Error occurred while trying to read Board Profile`

BoardSite could not read the Board Profile. The disk or catalog file has been corrupted.

ERROR 7: `Unable to write Board Profile`

The BoardSite system was unable to write a Board Profile to disk. The disk has been corrupted or is full.

ERROR 8: `This directory does not contain a BoardSite Catalog file.`

You must change the current directory, and/or the current drive to a drive and directory that does contain a BoardSite catalog file.

ERROR 9: `Unable to read Catalog file`

The catalog file has been deleted or corrupted.

ERROR 10: `Unable to update Catalog file`

The disk is full or write protected, or it has been corrupted.

ERROR 11: `Unable to find Batch File requested`

The Batch file was deleted by a program other than BoardSite, or the disk or catalog file has been corrupted. To correct this, use the Delete command from the File menu to remove the name from the catalog.

ERROR 12: `Unable to read Batch File`

This file has been corrupted.

ERROR 13: `Unable to access System Error Message File`

This is a fatal error. The BoardSite system could not locate the System Error Message File (SYSTEM.SEF). The BoardSite system was installed incorrectly or the System Error Message File has been deleted. To correct thi,s re-install the BoardSite system from the installation disks.

ERROR 14: `Unable to open Library Sequence Source File`

This is a fatal error. The Board Profile Editor could not locate the required Sequence File template from the library of sequence files. The BoardSite system has either been installed incorrectly, or the Sequence file has been deleted. To correct this the BoardSite software must be re-installed from the installation disks.

ERROR 15: Unable to open Sequence Source File

You need to edit the Board Profile and select the Device Type, which copies the correct device Sequence Source File for that Device Type from the System Library. Once you have saved the Board Profile you can edit the sequence file.

ERROR 16: Unable to write to Sequence Source File

This is a fatal error. The Board Profile Editor was unable to write out the Sequence Source file for the current Board Profile being saved. This could be caused by a disk error, a write-protected disk, or some type of DOS error.

ERROR 17: Unable to access System Help Message File

This is a fatal error. The BoardSite system could not locate the System Help Message file. The BoardSite system has either been installed incorrectly, or the System Help Message file has been deleted. To correct this the BoardSite software must be re-installed from the installation disks.

ERROR 18: Unable to create System Setup File

The BoardSite system was unable to create a System Setup File. The disk is full, write protected or has been corrupted.

ERROR 19: Unable to read System Setup File

The BoardSite system was unable to read a System Setup File. A DOS error has occurred or the file is corrupted. To correct this you should run the DOS utility CHKDSK to make sure your disk is not corrupted. You can delete the System Setup File (SYSTEM.SSF) and re-install the BoardSite software from the installation disks.

ERROR 20: Unable to update to System Setup File

The BoardSite system was unable to update the System setup File. The disk is full, write protected or has been corrupted, or the file may be read-only. Try running the DOS utility CHKDSK and make sure the file is not read-only.

ERROR 21: Unable to open User Text File

The user text file has been deleted by a program other than BoardSite, or the disk or catalog file has been corrupted. To correct this use the Delete command from the File menu to remove the name from the catalog.

ERROR 22: Unable to read User Text File

An error has occurred while trying to read a user text file. You may try running the DOS utility CHKDSK to make sure your disk is not corrupted.

Note: *Errors 23 through 27 indicate that the BoardSite system files have been corrupted in some manner. You should re-install the BoardSite software from the installation disks.*

ERROR 23: COMMAND name truncated while parsing System Command Definition File

ERROR 24: Syntax error encountered while parsing System Command Definition File

ERROR 25: DESCRIPTION STRING truncated while parsing System Command Definition file

ERROR 26: HELP KEYWORD truncated while parsing System Command Definition File

ERROR 27: Invalid HELP KEYWORD encountered while parsing System Command Definition File

ERROR 28: Unable to allocate enough memory to initialize BoardSite menu system

The system does not have enough free RAM to create its various buffers and storage areas. You may not have a full 640K system, or you may have installed too many RAM-resident utilities (TSRs). Check the CONFIG.SYS and AUTOEXEC.BAT files to see if RAM resident utilities (such as Sidekick, Superkey, etc.) are being run when the system boots up. BoardSite needs most of the 640K of RAM to run properly.

ERROR 29: Unable to read the System Command Definition File

BoardSite was unable to open the System Command Definition File (SYSTEM.CDF). The file may have been deleted or the BOARDSITE environment variable (set BOARDSITE=c:\brdsite) may not be in your AUTOEXEC.BAT file.

ERROR 30: Unable to open System Command Definition File

This means that either the BoardSite software was not installed correctly, or that the System Command Definition file has been deleted, or the disk corrupted.

ERROR 31: Unable to change the current Drive and Directory

You tried to change the drive and directory to one that does not exist, or you typed a path that is not valid. DOS only accepts a limited set of characters for path names. DOS drive names are A through Z only.

ERROR 32: No Board Names (Board Profiles) are in this Directory

You must change to a drive and/or directory that does contain Board Profiles, or you must Create a new file. New files may be created using the Edit command.

ERROR 33: No Board Data Files are in this Directory

You must change to a drive and/or directory that does contain Board Data Files, or you must create a new file. New files may be created using the Edit command.

ERROR 34: No Job Files are in this Directory

You must change to a drive and/or directory that does contain batch file, or you must create a new file. New files may be created using the Edit command.

ERROR 35: No Algorithm/Sequence Source Files are in this Directory

You must change to a drive and/or directory that does contain Algorithm/Sequence Source Files.

ERROR 36: Initialization of Device Support Information Failed

This is a fatal error. The BoardSite system could not locate the System Device Tables file. The BoardSite system has either been installed incorrectly, the BOARDSITE environment variable is setup incorrectly, or the System Device Tables file has been deleted.

ERROR 37: The Printer has timed out

Please check your printer, printer cable, or other items that may have caused the printer to time out.

ERROR 38: The Printer has returned an I/O Error

Please check your printer, printer cable or other items that may have caused the printer to fail.

ERROR 39: The Printer has returned status that it is Out of Paper

Please correct the situation and then press any key to continue operation.
You may press Esc to abort the operation.

ERROR 40: The Printer has returned status that it is Busy

Please correct the situation and then press any key to continue. You may press Esc to abort the operation.

ERROR 41: No BoardSite Text Files are in this Directory

You must change to a drive and/or directory that does contain text files.

ERROR 42: Unable to create a Catalog File in this Directory

The disk is full, write protected, or corrupted. Run the DOS utility CHKDSK to verify that your disk has not been corrupted.

ERROR 43: There are no files in this Catalog file.

You must change to a drive and/or directory that does contain BoardSite files.

ERROR 44: The file or path name of the program was not found

You probably didn't modify your AUTOEXEC.BAT file to add the path for the BoardSite software. Use a text editor to change the path statement in the AUTOEXEC.BAT file, and then reboot your PC and try to run BoardSite again.

ERROR 46: There is not enough memory available to execute the sub-program

There is not enough RAM to run the Board Profile Editor, the Sequence Editor, or a programming algorithm. This usually means that another program has used all the free RAM and DOS will not load one of the editors. If you consistently run out of memory, this indicates that something (such as a device driver or network shell) is not releasing RAM to DOS. Check your AUTOEXEC.BAT file to see if RAM-resident utilities (such as Sidekick, Superkey, etc.) are being run when the system boots up.

ERROR 48: The Password you have entered does not match the Password defined in the System Setup Parameters

The Password must match exactly, including upper and lower case letters.

ERROR 49: The Requested file was not found

Your catalog needs to be updated to reflect changes made to this directory. For example, if you have deleted files created by BoardSite without using the BoardSite program, the catalog file will not reflect the deletion of the file. To correct this use the Delete command from the File menu to remove the name from the catalog.

ERROR 50: There already is a BoardSite file with the name you have entered

Enter a different name if you really intend to create a new file, or perhaps you mean to edit an existing file. You may see all the existing BoardSite file names by going back to the menu bar, selecting the File command, and then choosing the List option to list the contents of the catalog file.

ERROR 51: The floppy Disk Drive door is Open

The BoardSite System is unable to select the default drive and directory because the floppy disk drive is open. Check to see if your system setup specifies the floppy disk as the default drive and directory, or check to see if you have a DOS environment variable set to the floppy disk drive. For example: BOARDSITE=A:\x.x

ERROR 52: The Floppy Disk is Write Protected

BoardSite attempted to write to a file on the floppy disk drive and it failed because the disk is write protected.

ERROR 53: A bad address mark was encountered when trying to access the floppy disk drive

Perhaps the disk is of the wrong format for DOS. Use the DOS utility CHKDSK to test the disk to make sure it is a DOS-formatted disk.

ERROR 54: Sector not found when trying to access the floppy disk drive

A sector may be bad. Discard the floppy diskette after recovering any data. Consult your DOS manual.

ERROR 55: Data or Seek Error when trying to access the floppy disk drive

This indicates that a hardware occurred when trying to access the floppy disk. Your floppy disk drive may need to be aligned.

ERROR 56: Unable to read DOS directory

Use the DOS utility CHKDSK to make sure your disk is not corrupted.

ERROR 57: This Board Profile needs to have it's Sequence Source File(s) Compiled and Linked before it can be used to perform a COPY, VERIFY, TEST, DISPLAY BOARD or CREATE DISKFILE Operation.

This means that the Board Profile you have selected references an executable algorithm file name that has not been compiled and linked using the Sequence Editor. You should run the Sequence Editor to compile and link the sequence file.

ERROR 58: Unable to Copy BoardSite File

Disk full or corrupted.

ERROR 59: Enter a valid board number (1-32)

You typed a board number of 0 or greater than 32. Type the correct board number and then try the command again.

ERROR 60: There are no DOS files to IMPORT from this drive/directory.

You tried to import a DOS file, but the only files in the current directory are BoardSite files. Move to a directory that contains files other than BoardSite files.

ERROR 61: There is insufficient memory to execute the BoardSite Text Editor.

This means that the system does not have enough free RAM to run the Text Editor program. You may not have a full 640K system, or you may have installed too many RAM-resident utilities (TSRs). Check the CONFIG.SYS and AUTOEXEC.BAT files to see if RAM resident utilities (such as Sidekick, Superkey, etc.) are being run when the system boots up. BoardSite needs most of the 640K of RAM to run properly.

ERROR 62: Unable to open the Text File specified.

A DOS error occurred while trying to open a text file. Check to make sure your disk is not full or corrupted by using the DOS utility CHKDSK.

ERROR 65: The Text Editor has failed internally and the edit session has been aborted.

This means that the system does not have enough free RAM to run the Text Editor program. You may not have a full 640K system, or you may have installed too many RAM-resident utilities (TSRs). Check the config.sys and autoexec.bat file to see if RAM-resident utilities (such as Sidekick and Superkey) run when the system boots up. BoardSite needs most of the 640K of RAM to run properly.

ERROR 66: A BoardSite DOS environment variable is not defined. This may cause BoardSite software to behave oddly or to crash, depending on the circumstances. For example, your AUTOEXEC.BAT file may need SET BOARDSITE=C:\BRDSITE

To check your DOS environment using the DOS command set, type **set** and then press ↵. If you do not see a display such as `set boardsite=c:\brdsite`, change your **autoexec.bat** file to include **set boardsite=*drive\directory*,** where *drive\directory* points to your **brdsite.exe** path.

WARNING 67: TMP environment variable does not point to a valid drive and/or directory, or the drive/directory is write protected

Check your DOS environment TMP variable to make sure it points to a drive/directory that is not full and can be written. The BoardSite software uses this path to create temporary files during compilation of sequence files.

ERROR 68: Unable to load Algorithm. Initialization failed!

Your algorithm has been corrupted. Try running the Sequence Editor and recompile the algorithm. If that doesn't work, run the DOS utility CHKDSK.

Programming Messages

ERROR 200: Unidentified programming error, please contact Data I/O Customer Resource Center

An error was encountered during the programming operation that was probably caused by a hardware failure.

ERROR 201: Adapter was not found - the adapter is not installed, the connector pins are not fully engaged, or the Board Profile did not specify the proper interface board locations

Adapter is not installed or connector pins not fully engaged.

ERROR 202: Adapter ID was not found or does not match the ID specified in the board profile

The adapter ID read from the adapter ID lines does not match the ID specified in the Board Profile.

ERROR 203: Master Board was not found. The Master Board is in the wrong position, missing or not fully engaged

The master board is in wrong position in the adapter, it is missing, or its connector pins are not fully engaged in the adapter connector.

ERROR 204: No boards were found, the board(s) were not fully engaged or the board detect lines were not setup

No boards are installed, their connector pins are not fully engaged, or the board detect lines (BD0-BD7) are not wired correctly.

WARNING 205: Non-isolated boards were detected on an adapter which is specified as an isolated adapter (one board per interface card). Check your board detect lines to ensure that only one board detect line is set low.

Board Profile specified isolated adapter, (one board per interface card) but found more than one board per interface card.

ERROR 206: Invalid current-limit values detected. The Board Profile current-limits are not within system limits.

Board Profile current limits are not within the system limits.

ERROR 207: AC power supply failure

The hardware detected a fault on the AC supply.

ERROR 208: Unable to access device group record

Edit the Board Profile to determine if the file is intact.

ERROR 209: Unable to read source

If you're copying from a master board, the error occurred while attempting to read the master board. If you're copying from a disk, the error occurred while attempting to read the data file.

ERROR 210: Unable to open data file. Check if the file exists, if the directory path is setup, or if using a fixed data file mode, the data file name should be specified in the Board Profile.

Check path setup, or insert the disk containing the source data file.

ERROR 211: An error occurred while attempting to write to the data file

This could be caused by a disk error, a write-protected disk, or some type of DOS error.

ERROR 212: An error occurred while attempting to read from the data file. Check the data file size to determine if an end of file was found prematurely.

Error occurred when attempting to read from data file.

ERROR 213: Data Ram could not be allocated. The system was unable to access the minimum memory needed. This may be corrected by restarting the BoardSite program.

RAM is already being fully utilized.

ERROR 215: Operation Aborted

You aborted the operation by pressing Esc, or, if running in Batch mode via a serial port, the host sent an ASCII escape character.

WARNING 216: Unable to perform simultaneous power-up due to too many parameters

Please refer to the "vxxxx_preset" and "power_up_supplies" primitives online help, in the Sequence Editor.

WARNING 217: Invalid board number received by the board-error routine

There is a problem with your sequence file. Call Data I/O Customer Resource Center.

WARNING 218: Logic RAM corrupted, JEDEC data and word-wide data do not correspond

Call Data I/O Customer Resource Center.

Programming Power Supply Messages

ERROR 240: Over-Current error Interface board xx

Current exceeded the over-current value specified in the Board Profile. Check for improperly installed or defective devices.

ERROR 241: Over-Voltage error Interface board xx

The interface board has been shut down due to an over-voltage detection.

ERROR 242: Under-voltage error Interface board xx

The interface board has been shut down due to detection of under-voltage.

ERROR 243: Unknown programming supply error

Call Data I/O Customer Support. A power supply failure occurred but the failure mechanism was not detected.

ERROR 244: Hardware Sequence Port Fault

This error is usually caused by another error and will disappear when the other error is corrected.

ERROR 245: Unknown Interrupt Detected

An interrupt was received that could not be serviced. You probably have a device in your PC that conflicts with BoardSite hardware.

Board Programming Messages

Note: *For the following messages, vv=device name, xx=board number (1-32), yy=board address of failure, zz=source data, and ww=board data. Also, the messages are self-explanatory.*

WARNING 250: Non-blank device board xx "vv" addr yyyyyyyy dev wwwwwwww

ERROR 251: Erase error board xx "vv" addr yyyyyyyy dev wwwwwwww

ERROR 252: Illegal-bit error board xx "vv" addr yyyyyyyy src zzzzzzzz dev wwwwwwww

ERROR 253: Lock error board xx "vv" addr yyyyyyyy src zzzzzzzz dev wwwwwwww

ERROR 254: Program error board xx "vv" addr yyyyyyyy src zzzzzzzz dev wwwwwwww

```
ERROR 255: Read error board xx addr yyyyyyyy

ERROR 256: Verify error board xx "vv" addr yyyyyyyy src zzzzzzzz
dev wwwwwwww

ERROR 257: CRC fail board xx actual yyyyyyyy expected zzzzzzzz

ERROR 258: Checksum fail board xx actual yyyyyyyy expected zzzzzzzz

ERROR 259: CRC fail board xx prom "vv" actual yyyyyyyy expected
zzzzzzzz

ERROR 260: Checksum fail board xx prom "vv" actual yyyyyyyy
expected zzzzzzzz

ERROR 261: User failure board xx addr yyyyyyyy
```

Note: Messages 264, 265, 267, and 268 are for blank check of differential cell devices, which have separate ones and zeroes arrays.

```
WARNING 264: Blank 0's, board xx "uu" address yyyyyyyy dev wwwwwwww

WARNING 265: Blank 1's, board xx "uu" address yyyyyyyy dev wwwwwwww

ERROR 266: Non-blank, board xx "uu" address yyyyyyyy dev wwwwwwww

ERROR 267: Blank 0's, board xx "uu" address yyyyyyyy dev wwwwwwww

ERROR 268: Blank 1's, board xx "uu" address yyyyyyyy dev wwwwwwww

WARNING 280: Block limits set for bulk erase device
```

No block limits are valid for erasing bulk erase device

```
WARNING 281: System current-limit exceeded: wwww programming supply
xxxxxmA

WARNING 282: Interface card current-limit exceeded: wwww programming
supply xxxxxmA

WARNING 283: Read error occurred on file "file name"

WARNING 284: Write error occurred on file "file name"

WARNING 285: Device does not support blank check
```

You can suppress this warning by disabling the test operation in the Copy Overhead Definition field in the board profile.

```
WARNING 286: Device does not support illegal-bit check
```

You can suppress this warning by disabling the test operation in the Copy Overhead Definition field in the board profile.

```
WARNING 292: CRC did not match the value specified in the board
profile. Expected XXXXXXXX, Calculated YYYYYYYY

WARNING 293: Checksum did not match the value specified in the
Board Profile. Expected XXXXXXXX, Calculated YYYYYYYY
```

Batch Command Messages (400 – 499)

ERROR 400: End-Of-File encountered in Batch Source File

The batch processor encountered end-of-file while reading the batch file and terminated the batch job. Normally you would end the batch file with the END_BATCH command.

ERROR 401: Error reading Batch Source File

A DOS error occurred while reading from the batch file. Use the DOS utility CHKDSK to make sure your disk is not corrupted.

ERROR 402: Syntax Error. No Batch Command specified

If you have a BEGIN_COMMAND or WAIT_FOR_ESCAPE command without setting the batch command type (COPY, VERIFY, TEST, and so on) this error will occur. Before trying to start the command you need to specify the command type.

ERROR 403: Syntax Error. Too many parameters

The command has too many parameters for the parser to handle. Check your batch file to make sure it has valid commands.

ERROR 404: Syntax Error. No parameters specified

A command was specified without parameters that requires parameters. For example, you typed PORT =

ERROR 405: Unable to open Batch Serial Port

An error occurred while trying to open the serial port to run batch commands via the serial port.

ERROR 406: Unable to open Batch Source File

The batch file specified with a BATCH_INPUT command does not exist. Use the File command to list the batch file names.

ERROR 408: Unknown Batch Command

An invalid batch command was encountered.

ERROR 409: Syntax Error in Batch parameter

An invalid parameter was encountered for a batch command. For example, a value that is not valid or is out of range was specified.

ERROR 410: Invalid Batch Command or not enough parameters

The batch parser had trouble with the current command. Check to make sure it is a valid command and that the parameters are of the correct type.

ERROR 411: Board Profile has not been selected

A COPY, VERIFY, TEST, FILE_CREATE, or CLEAR_BOARD_STATS command was specified, then a BEGIN_COMMAND was executed but a Board Profile name was never specified correctly with the BOARD_PROFILE_NAME command.

ERROR 412: Device Names have not been selected

A COPY, VERIFY, TEST or FILE_CREATE command was specified, then a BEGIN_COMMAND was executed but the device names were never selected correctly with the DEVICE_NAME command.

ERROR 413: A Command has not been selected

A BEGIN_COMMAND was executed but no command was selected (COPY, VERIFY, TEST, FILE_CREATE, UPLOAD, DOWNLOAD, TRANSLATE, and so on).

ERROR 414: A Port has not been selected

A BEGIN_COMMAND was executed for UPLOAD, DOWNLOAD or SEND_HOST_COMMAND but a PORT=COM1 or PORT=COM2 command was not executed. You must execute a PORT=COMn command before one of these commands even if you are running batch via a serial port.

A typical sequence of commands for an UPLOAD would be:

PORT = COM1
BAUD_RATE = 9600
PARITY = EVEN
UPLOAD_INPUTFILE = "Acme Controller Data"
FORMAT = Binary UPLOAD
BEGIN_COMMAND

ERROR 415: Data Source (DISK or MASTER) has not been selected

A COPY, VERIFY or TEST command was attempted but the data source (DISK or MASTER) was not specified. For example, you may need to specify SOURCE_OPTION=DISK.

ERROR 416: BoardSite Binary Data File has not been selected

A COPY, VERIFY, TEST or FILE_CREATE command was attempted but a data file name was not specified correctly. For example, you may need to specify DATA_FILE_NAME="Acme Controller Data".

ERROR 417: Invalid Board Profile Name

An invalid name was used when trying to select the Board Profile with the BOARD_PROFILE_NAME= command.

ERROR 418: Invalid Device Name

An invalid device name was specified when trying to select devices with the DEVICE_NAME= command.

ERROR 419: Invalid BoardSite Binary Data File Name

An invalid data file name was specified when trying to select the data file with the DATA_FILE_NAME= command.

ERROR 420: Invalid Batch Input Name

An invalid name was specified when trying to select a new batch file name with the BATCH_INPUT= command.

ERROR 421: Operator Time-Out

Operator time-out is really more of a warning, because it indicates that (1) the Operator Time-Out has been set to a non-zero value, and that (2) an operation was waiting for the operator to press a key and the Operator Time-Out occurred before the operator pressed a key. If you want operations to proceed without operator intervention and don't want errors generated, use OPERATOR_TIMEOUT=-1 which ignores the time-out and generates no errors.

ERROR 423: Missing Batch Command parameter

A batch command that requires at least one parameter has no parameters specified. Check the command syntax.

ERROR 424 Translate Option (BINARY or ASCII) not selected

You tried to use the TRANSLATE command without first specifying the translate option with the TRANSLATE_OPTION command.

ERROR 425 Translate DOS File Name not selected

You tried to use the TRANSLATE command without first specifying the DOS filename with the DOS_FILE command.

ERROR 426 Translate BoardSite Binary File Name not selected

You tried to use the TRANSLATE command without first specifying the BoardSite binary filename with the TRANSLATE_FILE command.

ERROR 427 GOTO Label does not exist

The label you specified in the GOTO, IF_ERROR GOTO, IF_ERROR EQ number GOTO, or IF_ERROR NE number GOTO command does not exist. Make sure you have the colon before the label.

ERROR 428 Batch Command not valid in Host Mode

You cannot use this command while running batch commands from the host.

Binary Data Editor Messages (500 – 599)

Note: Errors 502 through 509 indicate that your BoardSite software has been corrupted. Install the software again, using the original disks shipped with your equipment.

ERROR 502: Unrecognized parameters

ERROR 503: Bad RAM address

ERROR 504: Bad board address

ERROR 505: Bad block size

ERROR 506: Ram block over 64K

ERROR 507: Bad word size

ERROR 508: Address range too big

ERROR 509: Unable to open file

ERROR 510: Not enough memory

The Binary Data Editor could not find enough free memory to edit the data file. Check to make sure you are not running resident programs such as TSRs or network shells.

ERROR 511: Disk read error

An error occurred while the Binary Data Error was trying to read the data file to be edited. Use the DOS utility CHKDSK to make sure your disk is not corrupted.

ERROR 512: Disk write error

An error occurred while the Binary Data Editor was trying to write the data file being edited. Use the DOS utility CHKDSK to make sure your disk is not full or corrupted.

Board Profile Messages (600 – 699)

ERROR 600: When you create a new Board Profile, you must at least successfully define the Board Profile parameters, at least one Algorithm, and at least one Device Group. This is a 'minimum' Board Profile

If you pressed [Esc] in any of these editing forms, the entire Board Profile is scrapped and the editor will abort.

ERROR 601: You cannot specify an Address Bus Width such that the sum of the Address Bus Width and the number of Chip Enable Lines exceed 32

Reduce the Address Bus Width to correct this parameter.

ERROR 602: This is the only Algorithm Group defined for this Board Profile so you cannot delete it

A Board Profile requires a minimum of one algorithm group with at least one device group. If you want to delete this Board Profile, press [Esc] to exit the Board Profile Editor, and then select the File command to delete the Board Profile.

If you really want to delete the algorithm but you want to define a different algorithm (device type) in this Board Profile, first define the new algorithm and at least one device group, and then delete the first algorithm.

ERROR 603: This is the only Device Group defined for this Algorithm Group

A Board Profile requires a minimum of one algorithm group with at least one device group. This device group cannot be deleted without first adding another device group. If this Board Profile has more than one algorithm, you may delete the algorithm for this device group (which deletes this device group as well).

ERROR 604: Device Names must be unique for all Device Groups, for all Algorithm Groupings

Device names can contain up to 8 characters, upper and lower case, numeric and alphabetic characters.

ERROR 605: Empty Device Names!

This Device Information cannot be saved by pressing [Alt] + [B]) until all device names have a unique name. If you want to discard the information in this Device Information Form, then press [Esc] . If you want to save this information, then enter names for all the devices and then press [Alt] + [B] again.

ERROR 606: Test sub-operation not supported for this device

This device does not support blank check or illegal bit test operations during a Copy operation.

ERROR 607: Erase sub-operation not supported for this device

This device does not support an erase operation during a Copy operation.

ERROR 608: Lock sub-operation not supported for this device

This device does not support a lock (security programming) operation during a Copy operation.

ERROR 609: There is no Program sub-operation specified in your Copy Overhead Definition

You must have a program operation in your Copy overhead definition.

Communications Messages (700 – 799)

ERROR 700: Bad RAM address

ERROR 701: Not enough Storage space for this operation

The combination of Beginning Ram Address and Blocksize specified for this operation would require more storage space (disk or RAM) than is available in your system. Check the parameters again and make sure you understand how much storage would be required. The total storage required is Blocksize + Beginning Ram Address.

ERROR 702: Framing error

This error indicates that a framing error occurred during an upload or download operation. This can occur if the BoardSite communications parameters (baud rate, parity, data bits, or stop bits) do not match those of the host.

ERROR 703: I/O overrun error

This error indicates that the host system has sent data faster than BoardSite can process it. This usually can be corrected by using XON/XOFF handshaking, CTS/RTS handshaking, or by reducing the baud rate.

ERROR 704: Parity error

Make sure the parity setting matches the host. If it does, check the other communications parameters. If all parameters appear to be correct, you may have a defective RS-232 cable.

ERROR 705: I/O time-out

BoardSite didn't receive data within the I/O time out period, and cancelled the operation.

ERROR 706: Receive buffer overflow error

Check the BoardSite handshaking settings and make sure they match the host. If your are not using handshaking, try enabling XON/XOFF handshake or CTS/RTS handshake.

ERROR 707: Transmission checksum error

The transmitted checksum did not match the calculated checksum.

ERROR 708: Invalid data error (wrong translation format chosen)

This error can occur under several different circumstances. If the I/O translator expects hex ASCII characters and the characters are not valid then this error will occur. Also if the I/O Format is one that has specific header blocks (such as HP-UX), and the header information appears to be incorrect, this error may occur.

ERROR 709: I/O format error

This error can occur if the I/O Format selected requires specific characters, and they are not received or invalid characters are received.

ERROR 710: Disk read error

During Upload or Translate commands, an error occurred when trying to read the BoardSite binary data file. Use the DOS utility CHKDSK to make sure your disk is not corrupted, or use the Data Editor to see if the file can be read.

ERROR 711: Disk write error

This error can occur during a Download or Translate command, if the disk is full or has been corrupted. Use the DOS utility CHKDSK to see if there is free space on your disk or if it is corrupted.

ERROR 712: Not enough disk space to create file

During a Download or Translate operation, the disk was full when BoardSite attempted to fill the file with the fill-byte value. Make sure you have enough free space on your disk for this operation. The size of the file created will be the sum of Beginning Ram Address and Blocksize.

ERROR 713: Unable to open source file

A DOS error has occurred when trying to read the source file during a Translate operation. Use the DOS utility CHKDSK to make sure your disk is not corrupted and make sure the file exists.

ERROR 714: Unable to open destination file

The BoardSite software was unable to create the destination file during a Download or a Translate operation. Check to make sure your disk is not full.

ERROR 715: Operation Aborted by user

The Upload, Download or Translate operation was aborted.

ERROR 716: Unable to open serial port

An error occurred when trying to initialize the serial port before beginning an Upload or Download operation. Make sure that your system has a serial port configured as COM1 (or COM2) and that hardware or other software is not interfering with operation of the port. For example, some serial mouse boards interfere with the serial ports.

ERROR 717: Out of Memory in Communications Module

This error indicates that the BoardSite software was not able to allocate its required buffers and storage space in order to operate properly. Check to make sure that you do not have RAM-resident software or network shells.

ERROR 724: JEDEC File Fuse Checksum Error

The checksum calculated over the fuse map area does not match the fuse checksum value received (JEDEC translation format only).

Diagnostic Messages (800 – 999)

ERROR 800: Expansion Bus Failed Address Readback

The PC was not able to successfully perform a read/write test of the address bus between BoardSite and the PC. Suspect the Transmitter Board in the PC, the BoardSite-to-PC cable, the Expansion Board, or the Controller Board.

ERROR 801: Expansion Bus Failed Data Readback

The PC was not able to successfully perform a read/write test of the data bus between BoardSite and the PC. Suspect the Transmitter Board in the PC, the BoardSite-to-PC cable, the Expansion Board, or the Controller Board.

ERROR 802: Sequence Port Failed to Reset

Suspect a problem with the Controller Board.

ERROR 803: Sequence Port Failed Read/Write

BoardSite was unable to successfully perform a read/write test of its internal bus system. Suspect either a faulty Controller Board or Interface Board.

ERROR 804: Timer Status Busy When Not Expected

Suspect a problem with the Controller Board.

ERROR 805: Timer Status Not Busy When Expected
Suspect a problem with the Controller Board.

ERROR 806: Timer 0 Failed Read/Write
Suspect a problem with the Controller Board or an Interface Board.

ERROR 807: Timer 1 Failed Read/Write
Suspect a problem with the Controller Board or an Interface Board.

ERROR 808: Timer 2 Failed Read/Write
Suspect a problem with the Controller Board or an Interface Board.

ERROR 809: User Clock Failed Rate Setup
Suspect a problem with the Controller Board.

ERROR 810: Timer Clock Failed Rate Setup
Suspect a problem with the Controller Board.

ERROR 811: Program Pulse Width Incorrect
Suspect a problem with the Controller Board.

ERROR 812: Overprogram Pulse Width Incorrect
Suspect a problem with the Controller Board.

ERROR 813: PGM Control Circuit Failed
Suspect a problem with the Controller Board.

ERROR 814: PGM Generation Circuit Failed
Suspect a problem with the Controller Board.

ERROR 815: VPP1 Hi/Lo Control Switch Failed
Suspect a problem with the Controller Board.

ERROR 816: VPP2 Hi/Lo Control Switch Failed
Suspect a problem with the Controller Board.

ERROR 817: MUX Control Switch Failed
Suspect a problem with the Controller Board.

ERROR 818: +32V Supply Over-Voltage
Main Power Supply out of calibration or suspect a problem with the Controller Board.

ERROR 819: +32V Supply Under-Voltage
Main Power Supply out of calibration or suspect a problem with the Controller Board.

ERROR 820: +12V Supply Over-Voltage

Main Power Supply out of calibration or suspect a problem with the Controller Board.

ERROR 821: +12V Supply Under-Voltage

Main Power Supply out of calibration or suspect a problem with the Controller Board.

ERROR 822: +10V Reference Over-Voltage

Suspect a problem with the Controller Board.

ERROR 823: +10V Reference Under-Voltage

Suspect a problem with the Controller Board.

ERROR 824: +5V Supply Over-Voltage

Main Power Supply out of calibration or suspect a problem with the Controller Board.

ERROR 825: +5V Supply Under-Voltage

Main Power Supply out of calibration or suspect a problem with the Controller Board.

ERROR 826: Analog Ground Over-Voltage

Suspect a problem with the Controller Board, the Mother Board, or the Main Power Supply.

ERROR 827: Analog Ground Under-Voltage

Suspect a problem with the Controller Board, the Mother Board, or the Main Power Supply.

ERROR 828: -3.5V Supply Over-Voltage

Suspect a problem with the Controller Board, an Interface Board, or the Pre-regulator Board.

ERROR 829: -3.5V Supply Under-Voltage

Suspect a problem with the Controller Board, an Interface Board, or the Pre-regulator Board.

ERROR 830: -5V Supply Over-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 831: -5V Supply Under-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 832: -12V Supply Over-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 833: -12V Supply Under-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 834: +12V PC Supply Over-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 835: +12V PC Supply Under-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 836: +5V PC Supply Over-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 837: +5V PC Supply Under-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 838: -5V PC Supply Over-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 839: -5V PC Supply Under-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 840: -12V PC Supply Over-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 841: -12V PC Supply Under-Voltage

Suspect a problem with the Mother Board or the Main Power Supply.

ERROR 842: VCC1 DAC Cannot be Set to 0V

Suspect a problem with the Controller Board.

ERROR 843: VPP1HI DAC Cannot be Set to 0V

Suspect a problem with the Controller Board.

ERROR 844: VPP1LO DAC Cannot be Set to 0V

Suspect a problem with the Controller Board.

ERROR 845: VCC2 DAC Cannot be Set to 0V

Suspect a problem with the Controller Board.

ERROR 846: VPP2HI DAC Cannot be Set to 0V

Suspect a problem with the Controller Board.

ERROR 847: VPP2LO DAC Cannot be Set to 0V

Suspect a problem with the Controller Board.

ERROR 848: VNEG DAC Cannot be Set to 0V

Suspect a problem with the Controller Board.

ERROR 849: ICC1 DAC Cannot be Set to 0V
Suspect a problem with the Controller Board.

ERROR 850: IPP1 DAC Cannot be Set to 0V
Suspect a problem with the Controller Board.

ERROR 851: ICC2 DAC Cannot be Set to 0V
Suspect a problem with the Controller Board.

ERROR 852: IPP2 DAC Cannot be Set to 0V
Suspect a problem with the Controller Board.

ERROR 853: INEG DAC Cannot be Set to 0V
Suspect a problem with the Controller Board.

ERROR 854: VCC1 Regulator Out of Range at 0V
Suspect a problem with the Pre-regulator Board.

ERROR 855: VPP1 Regulator Out of Range at 0V
Suspect a problem with the Pre-regulator Board.

ERROR 856: VCC2 Regulator Out of Range at 0V
Suspect a problem with the Pre-regulator Board.

ERROR 857: VPP2 Regulator Out of Range at 0V
Suspect a problem with the Pre-regulator Board.

ERROR 858: Test REF 1 DAC Cannot be Set to 0V
Suspect a problem with the Controller Board.

ERROR 859: Test REF 2 DAC Cannot be Set to 0V
Suspect a problem with the Controller Board.

ERROR 860: VCC1 DAC Cannot be Set to 5V
Suspect a problem with the Controller Board.

ERROR 861: VPP1HI DAC Cannot be Set to 5V
Suspect a problem with the Controller Board.

ERROR 862: VPP1LO DAC Cannot be Set to 5V
Suspect a problem with the Controller Board.

ERROR 863: VCC2 DAC Cannot be Set to 5V
Suspect a problem with the Controller Board.

ERROR 864: VPP2HI DAC Cannot be Set to 5V
Suspect a problem with the Controller Board.

ERROR 865: VPP2LO DAC Cannot be Set to 5V
Suspect a problem with the Controller Board.

ERROR 866: VNEG DAC Cannot be Set to 5V
Suspect a problem with the Controller Board.

ERROR 867: ICC1 DAC Cannot be Set to 6A
Suspect a problem with the Controller Board.

ERROR 868: IPP1 DAC Cannot be Set to 2A
Suspect a problem with the Controller Board.

ERROR 869: ICC2 DAC Cannot be Set to 6A
Suspect a problem with the Controller Board.

ERROR 870: IPP2 DAC Cannot be Set to 2A
Suspect a problem with the Controller Board.

ERROR 871: INEG DAC Cannot be Set to 250mA
Suspect a problem with the Controller Board.

ERROR 872: VCC1 Regulator Out of Range at 5V
Suspect a problem with the Pre-regulator Board.

ERROR 873: VPP1 Regulator Out of Range at 5V
Suspect a problem with the Pre-regulator Board.

ERROR 874: VCC2 Regulator Out of Range at 5V
Suspect a problem with the Pre-regulator Board.

ERROR 875: VPP2 Regulator Out of Range at 5V
Suspect a problem with the Pre-regulator Board.

ERROR 876: VCC1 Regulator Out of Range at 7V
Suspect a problem with the Pre-regulator Board.

ERROR 877: VPP1 Regulator Out of Range at 25V
Suspect a problem with the Pre-regulator Board.

ERROR 878: VCC2 Regulator Out of Range at 7V
Suspect a problem with the Pre-regulator Board.

ERROR 879: VPP2 Regulator Out of Range at 25V

Suspect a problem with the Pre-regulator Board.

ERROR 880: Test REF 1 DAC Cannot be Set to 5V

Suspect a problem with the Controller Board.

ERROR 881: Test REF 2 DAC Cannot be Set to 5V

Suspect a problem with the Controller Board.

ERROR 882: BE Port Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 883: Control Port 3 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 884: BD Driver Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 885: ID Driver Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 886: LED Driver Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 887: Control Port 1 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 888: Control Port 2 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 889: Address Port 1 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 890: Address Port 2 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 891: Address Port 3 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 892: Address Port 4 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 893: Data Port 1 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 894: Data Port 2 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 895: Data Port 3 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 896: Data Port 4 Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 897: 8-Bit CE Port Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 898: 16-Bit CE Port Failed Read/Write

Suspect a problem with an Interface Board. Could also indicate a problem with the Controller Board.

ERROR 899: Delay Timer Busy When Not Expected

Suspect a problem with the Controller Board.

ERROR 900: Delay Timer Not Busy When Expected

Suspect a problem with the Controller Board.

Note: *Errors 901 through 990 indicate a problem with an Interface Board. Several errors on multiple Interface Boards could indicate a problem with the Controller Board. When the Expansion Power Supply is used, errors associated with programming power supplies may be due to a faulty Expansion Power Supply.*

ERROR 901: Power Sequencer RAM Failed Read/Write

ERROR 902: Power Sequencer Manual Control Failed

ERROR 903: Power Reset Circuit Failed to Reset

ERROR 904: Power Sequencer Automatic Control Failed

ERROR 905: VCC1 Supply Cannot be turned OFF

ERROR 906: VPP1HI Supply Cannot be turned OFF

ERROR 907: VPP1LO Supply Cannot be turned OFF

ERROR 908: VCC2 Supply Cannot be turned OFF

ERROR 909: VPP2HI Supply Cannot be turned OFF

ERROR 910: VPP2LO Supply Cannot be turned OFF

ERROR 911: VNEG Supply Cannot be turned OFF

ERROR 912: +12V Switch Cannot be turned OFF

ERROR 913: -12V Switch Cannot be turned OFF

ERROR 914: BE Port Failed Loopback

ERROR 915: Control Port 3 Failed Loopback

ERROR 916: BD Driver Failed Loopback

ERROR 917: ID Driver Failed Loopback

ERROR 918: LED Driver Failed Loopback

ERROR 919: Control Port 1 Failed Loopback

ERROR 920: Control Port 2 Failed Loopback

ERROR 921: Address Port 1 Failed Loopback

ERROR 922: Address Port 2 Failed Loopback

ERROR 923: Address Port 3 Failed Loopback

ERROR 924: Address Port 4 Failed Loopback

ERROR 925: Data Port 1 Failed Loopback

ERROR 926: Data Port 2 Failed Loopback

ERROR 927: Data Port 3 Failed Loopback

ERROR 928: Data Port 4 Failed Loopback

ERROR 929: Analog MUX Failed Data Readback

ERROR 930: Power Error Latch Failed to Set Error

ERROR 931: VCC1 Supply Over-Voltage at 5V

ERROR 932: VCC1 Supply Under-Voltage at 5V

ERROR 933: VCC1 Supply Over-Voltage at 7V

ERROR 934: VCC1 Supply Under-Voltage at 7V

ERROR 935: VCC1 Supply Over-Voltage Circuit Failed

ERROR 936: VCC1 Supply Under-Voltage Circuit Failed

```
ERROR 937: VCC1 Supply Over-Current Circuit Failed
ERROR 938: VCC2 Supply Over-Voltage at 5V
ERROR 939: VCC2 Supply Under-Voltage at 5V
ERROR 940: VCC2 Supply Over-Voltage at 7V
ERROR 941: VCC2 Supply Under-Voltage at 7V
ERROR 942: VCC2 Supply Over-Voltage Circuit Failed
ERROR 943: VCC2 Supply Under-Voltage Circuit Failed
ERROR 944: VCC2 Supply Over-Current Circuit Failed
ERROR 945: VPP1 Supply Over-Voltage at 5V
ERROR 946: VPP1 Supply Under-Voltage at 5V
ERROR 947: VPP1 Supply Over-Voltage at 25V
ERROR 948: VPP1 Supply Under-Voltage at 25V
ERROR 949: VPP1 Supply Over-Voltage Circuit Failed
ERROR 950: VPP1 Supply Under-Voltage Circuit Failed
ERROR 951: VPP1 Supply Over-Current Circuit Failed
ERROR 952: VPP1 Supply Slew Select Circuit Failed
ERROR 953: VPP1 Supply Hi/Lo Switch Circuit Failed
ERROR 954: VPP2 Supply Over-Voltage at 5V
ERROR 955: VPP2 Supply Under-Voltage at 5V
ERROR 956: VPP2 Supply Over-Voltage at 25V
ERROR 957: VPP2 Supply Under-Voltage at 25V
ERROR 958: VPP2 Supply Over-Voltage Circuit Failed
ERROR 959: VPP2 Supply Under-Voltage Circuit Failed
ERROR 960: VPP2 Supply Over-Current Circuit Failed
ERROR 961: VPP2 Supply Slew Select Circuit Failed
ERROR 962: VPP2 Supply Hi/Lo Switch Circuit Failed
ERROR 963: VNEG Supply Over-Voltage at -5V
ERROR 964: VNEG Supply Under-Voltage at -5V
ERROR 965: VNEG Supply Over-Voltage at -8V
ERROR 966: VNEG Supply Under-Voltage at -8V
ERROR 967: VNEG Supply Over-Voltage Circuit Failed
ERROR 968: VNEG Supply Under-Voltage Circuit Failed
ERROR 969: VNEG Supply Over-Current Circuit Failed
```

ERROR 970: +12V Switch Over-Voltage

ERROR 971: +12V Switch Under-Voltage

ERROR 972: -12V Switch Over-Voltage

ERROR 973: -12V Switch Under-Voltage

ERROR 974: VPP1 Reference Output Out of Range

ERROR 975: VPP2 Reference Output Out of Range

ERROR 976: Power Regulator Error Latch Failed

ERROR 977: Power Error Generator Failed to Set Error

ERROR 978: Power Error Generator Failed to Reset

ERROR 979: Sequence Port Failed to Set Fault

ERROR 980: Digital Drivers Failed to Tri-state

ERROR 981: ID Pull-up Switch Failed

ERROR 982: BD Pull-up Switch Failed

ERROR 983: Power Sequencer Address Inverter failed

ERROR 984: Power Sequencer Stop Indicator Failed

ERROR 985: Common Mode Decode Control Failed

ERROR 986: Interrupt Test Failed, Check Setup

This Error could indicate a bad Controller Board, but first use the Setup command to check the system configuration. Make sure the BoardSite IRQ number is set properly for the jumpers on the Controller Board and make sure the IRQ is not being used for something other than BoardSite. See Chapter 2 for BoardSite setup.

ERROR 987: Board ON/OFF control circuit failed

ERROR 988: INEG DAC cannot be set to 500mA

ERROR 989: VNEG Supply over voltage at 19.5V

ERROR 990: VNEG Supply under voltage at -19.5V

ERROR 991: Expansion Board not detected

This error could indicate a bad PC Transmitter Board, a bad BoardSite-to-PC cable, a bad Expansion Board, or a bad Controller Board. Try replacing each board until the problem board is found.

ERROR 992: Controller Board not detected

If error 991, "Expansion Board not detected" does not also appear with this error, suspect a problem with the Controller Board.

ERROR 993: Pre-regulator Board not detected

If all boards in the system are detected except the Pre-regulator Board, suspect a bad Interface Board in slot #4. This error is almost never caused by the actual Pre-regulator board (unless the board is missing or not inserted properly.)

ERROR 994: Interface Board not detected

If all Interface Boards are not detected, suspect a bad Controller board. Otherwise, suspect a problem with the Interface Board not detected.

ERROR 995: Expansion Power Supply not detected

Make sure power is on for both BoardSite and the Expansion Power Supply. If the error persists, suspect a bad Expansion Mother Board or a bad Interface Board in BoardSite.

ERROR 996: Power Connector not detected on Diagnostic Test Adapter

One or more of the jumpers on the J5 connector in the Diagnostic Test Adapter were not detected when performing tests on the interface board. Check to see that all 10 of the jumpers are seated properly on the J5 connector pins.

ERROR 997: Expansion Power Supply may be defective

This error is displayed if the Expansion Power Supply was detected, but programming power supply errors were encountered which could be due to a faulty Expansion Power Supply or Expansion Power Supply cables.

ERROR 998: Diagnostic Test Adapter not detected on Interface Board

The Diagnostic Adapter is either not on the proper interface board connector, or is not seated properly. If the error persists, suspect a bad Interface Board.

A Power Supply Calibration

This Appendix contains two sections. The first section describes how to adjust the BoardSite power supply. The second section describes how to measure the Diagnostic Test Adapter load resistors to verify that they are within specification.

Adjusting the Power Supply

WARNING: The following procedure should be done by qualified service personnel only. Lethal voltages are present in this equipment. Use extreme caution when making adjustments with the cover removed and power applied.

Equipment Required

You will need the following equipment:

- Digital multimeter (at least 4-1/2 digit resolution)
- Insulated potentiometer adjustment tool

Although the system software performs various voltage tests, there is still a possibility that these critical voltage levels may be out of specification. This section describes how to verify that voltage levels are within acceptable tolerances and how to adjust them if necessary.

Procedures describing calibration with and without the Expansion Power Supply are included.

Calibration without the Expansion Power Supply

The +5V, +12V, -12V (only units with PSI power supply), and +32V power supply outputs are adjustable. The DMM test points are located on the top edge of the 701-2168 circuit board, which is the circuit board next to the open-frame power supply in the main BoardSite assembly. These test points are labeled for your reference.

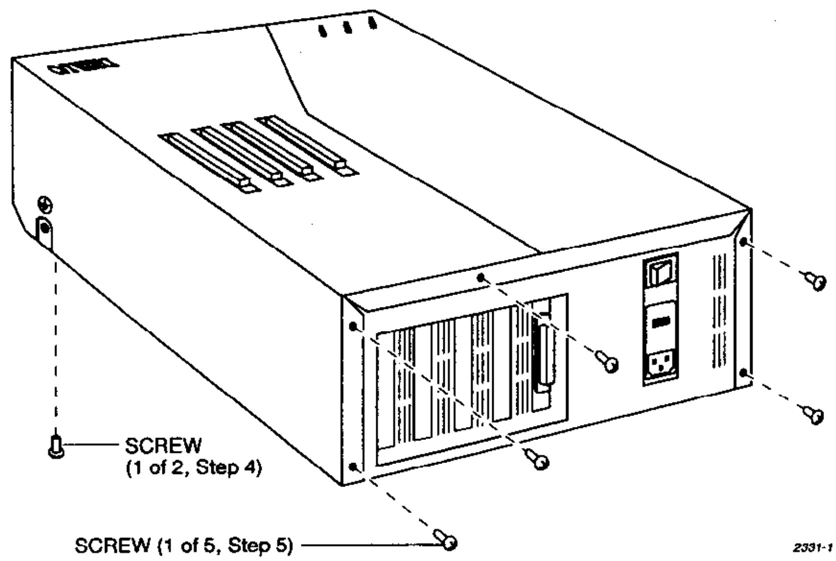
1. Turn off power to BoardSite.
2. Remove the ac power cord from BoardSite's ac power entry module .

3. Disconnect the 62-pin connector that connects BoardSite to the PC.
4. Remove the cover from BoardSite as described below.

   Benchtop Unit

   a. Place BoardSite on its side and remove two screws from the bottom panel. Then set the unit back on its feet. (See Figure A-1.)

Figure A-1
BoardSite Screw Locations



   b. Remove the five back panel screws.
   c. Raise the back of the cover and move it towards the front of BoardSite. Lift the cover clear and set it aside.

   Portable Unit

   a. Remove the 12 machine screws from BoardSite's front panel.
   b. Remove the panel and set it aside. You may have to gently pry around the perimeter of the panel with a slot-head screwdriver to free it from the base.

5. Locate the test points on the 701-2168 circuit board (see Figure A-2).
6. Locate the adjustment potentiometers on the open-frame power supply (see Figure A-3).
7. Connect the ac power cord to BoardSite and turn on the BoardSite power switch.
8. Carefully measure the three voltages by connecting the DMM between the appropriate test point on the 701-2168 board and the GND test point.
9. If any voltage falls outside the limits listed in Table 1, adjust the appropriate potentiometer on the power supply to bring the voltage back to within the limits shown in the table.
10. When you are finished, turn off the power, remove the ac power cord, and then reassemble BoardSite.

Table 1
Test Point Limits without Expansion Power Supply

Test Point	Min.	Nominal	Max.
+5V	+4.98V	+5.10V	+5.22V
+12V	+11.95V	+12.25V	+12.55V
+32V	+30.74V	+31.50V	+32.26V
PSI Supply Only			
-12V	-11.90*	-12.00V*	-12.10V*

Figure A-2
BoardSite 701-2168 Circuit Board

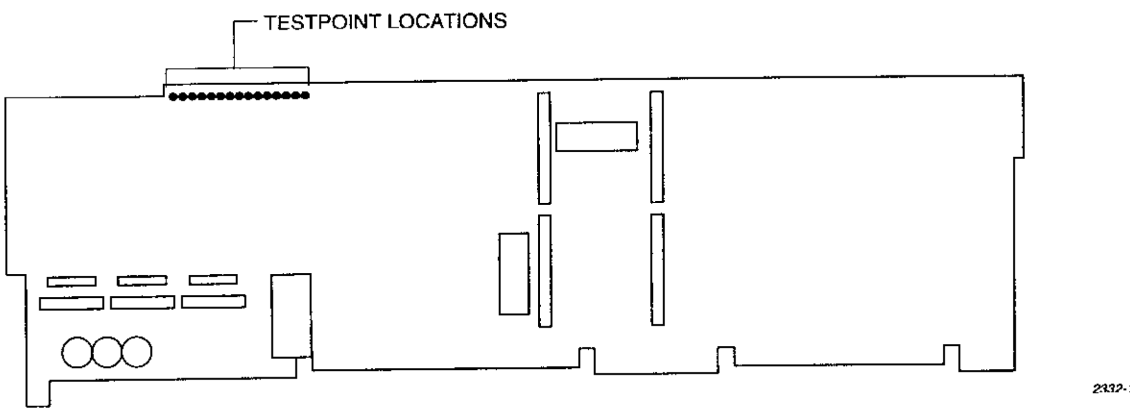
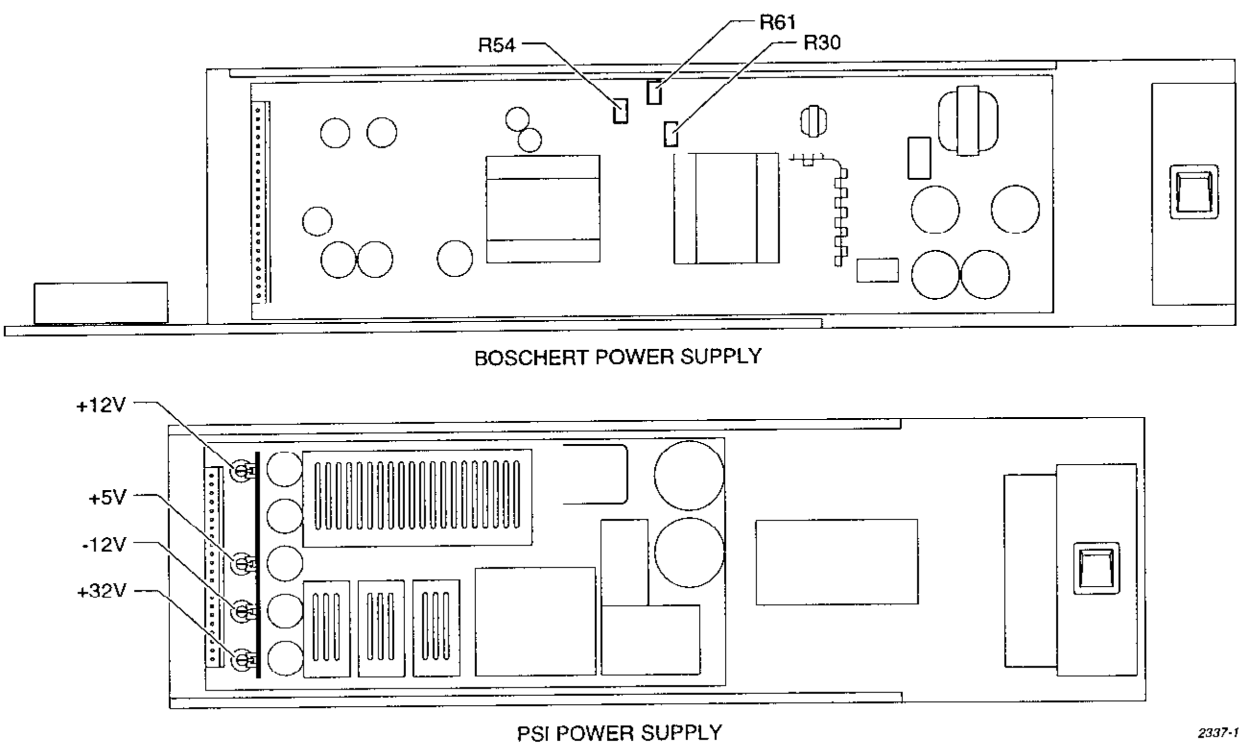


Figure A-3
BoardSite Open-Frame Power Supply



Note: Use this value instead of any other that may be displayed.

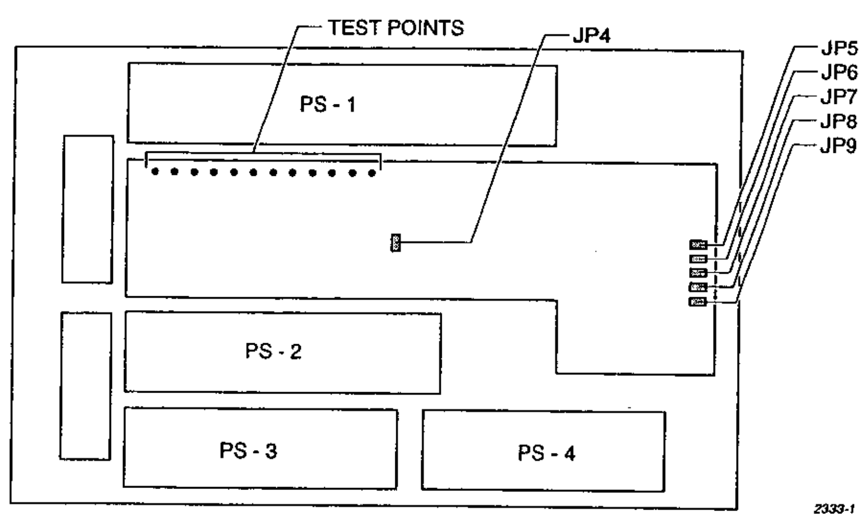
Calibration with the Expansion Power Supply

The +5V, +15V, +30V, and -24V power supply outputs are adjustable. No adjustment is provided for the +12V, -12V, -3.5V, +10V, and -24VSW power supply outputs.

The DMM test points are located on the top edge of the 701-2242 circuit board (the main circuit board in the Expansion Power Supply assembly) and are labeled for your reference.

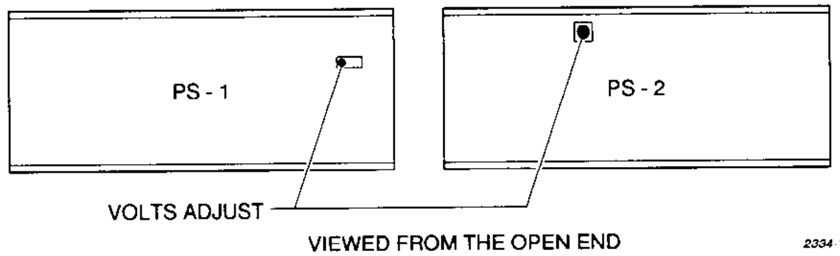
1. Turn off power to BoardSite.
2. Remove the ac power cord from the ac power entry module of the Expansion Power Supply and disconnect both the interface cables connecting BoardSite to the Expansion Power Supply.
3. Remove the cover from the Expansion Power Supply.
   a. Place the Expansion Power Supply on its side and remove two screws from the bottom panel, and then set the unit back on its feet.
   b. Remove the five back panel screws.
   c. Raise the back of the cover and move it towards the front of the unit. Lift the cover clear and set it aside.
4. Connect the negative voltmeter lead to the test point labeled GND.
5. Install jumpers on JP4 through JP9 (Figure A-4) on the Expansion Power Supply Motherboard (701-2242). JP4 through JP9 have jumpers installed on only one pin. You must remove the jumper and reinstall it on two pins.

Figure A-4
Expansion Power Supply Motherboard



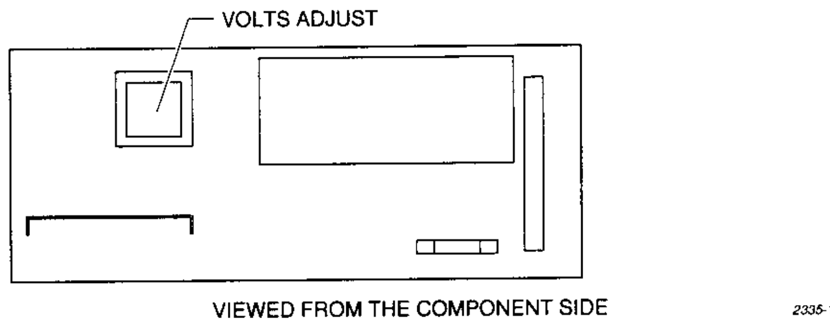
6. Reconnect the Expansion Power Supply and turn on the power.
7. Measure the voltages on the test points labeled +15V and +30V. If these voltages are not within the ranges specified in Table 2, adjust the voltage control on PS-1 or PS-2 as shown in Figure A-5.

Figure A-5
PS-1 or PS-2 Voltage Control



8. Measure the voltage on the test point labeled +5V. If the voltage is outside the tolerance range as specified in Table 2, adjust the voltage control on PS-3 (see Figure A-6) until the voltage is within its tolerance range.

Figure A-6
PS-3 Voltage Control



Note: *Before you adjust the +15V or the+30V supplies, determine if your BoardSite has alternate power supplies by reading the manufacturer and model numbers of the power circuits PS-1.*

If PS-1 is a Power One model SPL250-1024, both PS-1 (+30V supply) and PS-2 (+15V supply) are alternate supplies, and you should adjust for the voltages listed under "Alternate Supplies" in Table 2.

Table 2
Test Point Limits with Expansion Power Supply

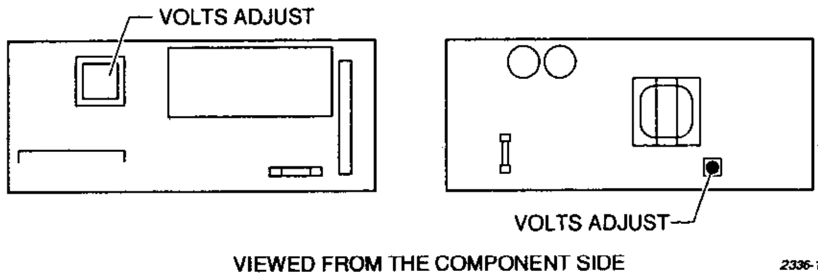
Test Pt's	**Min.**	**Nom.**	**Max.**	**Location**
+15V	+14.3V	+15.0V	+15.8V	PS-1
+30V	+29.8V	+30.8V	+31.7V	PS-2
+5V	+4.84V	+5.10V	+5.36V	PS-3
+12V	+11.3V	No Adj.	+13.0V	PS-3
-12V	-11.4V	No Adj.	-13.1V	PS-3
-24V	-23.8V	-24.0V	-24.2V	PS-4
-3.5V	-3.26V	No Adj.	-3.74V	
+10V	+9.80V	No Adj.	+10.2V	
-24V SW	-22.8V	No Adj.	-25.2V	

Alternate Supplies

Test Pt's	Min.	Nom.	Max.	Location
+15V	+14.3V	+15.0V	+15.8V	PS-2
+30V	+28.6V	+29.4V	+30.3V	PS-1

9. Measure the voltage on the test point labeled -24V. If the voltage is outside the tolerance range as specified in Table 2, then adjust the voltage control on PS-4 (see Figure A-7) until the voltage is within its tolerance range.

Figure A-7
PS-4 Voltage Control



10. Measure the voltage on the test points labeled +12V, -12V, -3.5V, +10V, and -24VSW. If these voltages are outside their tolerance range as specified in the voltage table, then the corresponding power supply will need to be replaced.
11. Remove the voltmeter leads from the test points.
12. Turn the Expansion Power Supply OFF, remove the jumpers from JP4 through JP9, and reinstall them onto one pin.
13. Reassemble the Expansion Power Supply.
14. Reconnect the Expansion Power Supply to BoardSite and resume operation.

Diagnostic Test Adapter Resistance Measurements

This section describes how to measure the Diagnostic Test Adapter load resistors to verify that they are within tolerance.

Equipment Required

Digital multimeter (at least 3-1/2 digit resolution).

Procedure

1. Remove the Diagnostic Test Adapter from BoardSite.

Note: Do not perform this procedure with the Diagnostic Test Adapter installed in BoardSite. You will get false resistance readings.

2. Slide open the top of the Diagnostic Test Adapter's metal housing. All test points are accessible through this opening. The test points are labeled for your reference.
3. Verify that the jumper header is installed on J5.
4. Place the DMM in resistance range. Connect the DMM leads to the test points shown in Table A-2. Verify the measured resistances against the ranges noted in the table.
5. Repeat step 4 for all test points in Table A-2.

From	To	Expect	Acceptable Range
A0	BD0	10kW	9.7kW to 10.3kW
A1	BD1	10kΩ	9.7kΩ to 10.3kΩ
A2	BD2	10kΩ	9.7kΩ to 10.3kΩ
A3	BD3	10kΩ	9.7kΩ to 10.3kΩ
A4	BD4	10kΩ	9.7kΩ to 10.3kΩ
A5	BD5	10kΩ	9.7kΩ to 10.3kΩ
A6	BD6	10kΩ	9.7kΩ to 10.3kΩ
A7	BD7	10kΩ	9.7kΩ to 10.3kΩ
A8	BD0	10kΩ	9.7kΩ to 10.3kΩ
A9	BD1	10kΩ	9.7kΩ to 10.3kΩ
A10	BD2	10kΩ	9.7kΩ to 10.3kΩ
A11	BD3	10kΩ	9.7kΩ to 10.3kΩ
A12	BD4	10kΩ	9.7kΩ to 10.3kΩ
A13	BD5	10kΩ	9.7kΩ to 10.3kΩ
A14	BD6	10kΩ	9.7kΩ to 10.3kΩ
A15	BD7	10kΩ	9.7kΩ to 10.3kΩ
A16	BD0	10kΩ	9.7kΩ to 10.3kΩ
A17	BD1	10kΩ	9.7kΩ to 10.3kΩ
A18	BD2	10kΩ	9.7kΩ to 10.3kΩ
A19	BD3	10kΩ	9.7kΩ to 10.3kΩ
A20	BD4	10kΩ	9.7kΩ to 10.3kΩ
A21	BD5	10kΩ	9.7kΩ to 10.3kΩ
A22	BD6	10kΩ	9.7kΩ to 10.3kΩ
A23	BD7	10kΩ	9.7kΩ to 10.3kΩ
A24	BD0	10kΩ	9.7kΩ to 10.3kΩ
A25	BD1	10kΩ	9.7kΩ to 10.3kΩ
A26	BD2	10kΩ	9.7kΩ to 10.3kΩ
A27	BD3	10kΩ	9.7kΩ to 10.3kΩ
A28	BD4	10kΩ	9.7kΩ to 10.3kΩ
A29	BD5	10kΩ	9.7kΩ to 10.3kΩ
A30	BD6	10kΩ	9.7kΩ to 10.3kΩ
A31	BD7	10kΩ	9.7kΩ to 10.3kΩ
D0	BD0	10kΩ	9.7kΩ to 10.3kΩ
D1	BD1	10kΩ	9.7kΩ to 10.3kΩ
D2	BD2	10kΩ	9.7kΩ to 10.3kΩ
D3	BD3	10kΩ	9.7kΩ to 10.3kΩ
D4	BD4	10kΩ	9.7kΩ to 10.3kΩ
D5	BD5	10kΩ	9.7kΩ to 10.3kΩ
D6	BD6	10kΩ	9.7kΩ to 10.3kΩ
D7	BD7	10kΩ	9.7kΩ to 10.3kΩ
D8	BD0	10kΩ	9.7kΩ to 10.3kΩ
D9	BD1	10kΩ	9.7kΩ to 10.3kΩ
D10	BD2	10kΩ	9.7kΩ to 10.3kΩ
D11	BD3	10kΩ	9.7kΩ to 10.3kΩ
D12	BD4	10kΩ	9.7kΩ to 10.3kΩ
D13	BD5	10kΩ	9.7kΩ to 10.3kΩ
D14	BD6	10kΩ	9.7kΩ to 10.3kΩ
D15	BD7	10kΩ	9.7kΩ to 10.3kΩ

From	To	Expect	Acceptable Range
D16	BD0	10kΩ	9.7kΩ to 10.3kΩ
D17	BD1	10kΩ	9.7kΩ to 10.3kΩ
D18	BD2	10kΩ	9.7kΩ to 10.3kΩ
D19	BD3	10kΩ	9.7kΩ to 10.3kΩ
D20	BD4	10kΩ	9.7kΩ to 10.3kΩ
D21	BD5	10kΩ	9.7kΩ to 10.3kΩ
D22	BD6	10kΩ	9.7kΩ to 10.3kΩ
D23	BD7	10kΩ	9.7kΩ to 10.3kΩ
D24	BD0	10kΩ	9.7kΩ to 10.3kΩ
D25	BD1	10kΩ	9.7kΩ to 10.3kΩ
D26	BD2	10kΩ	9.7kΩ to 10.3kΩ
D27	BD3	10kΩ	9.7kΩ to 10.3kΩ
D28	BD4	10kΩ	9.7kΩ to 10.3kΩ
D29	BD5	10kΩ	9.7kΩ to 10.3kΩ
D30	BD6	10kΩ	9.7kΩ to 10.3kΩ
D31	BD7	10kΩ	9.7kΩ to 10.3kΩ
C0	BD0	10kΩ	9.7kΩ to 10.3kΩ
C1	BD1	10kΩ	9.7kΩ to 10.3kΩ
C2	BD2	10kΩ	9.7kΩ to 10.3kΩ
C3	BD3	10kΩ	9.7kΩ to 10.3kΩ
C4	BD4	10kΩ	9.7kΩ to 10.3kΩ
C5	BD5	10kΩ	9.7kΩ to 10.3kΩ
C6	BD6	10kΩ	9.7kΩ to 10.3kΩ
C7	BD7	10kΩ	9.7kΩ to 10.3kΩ
C8	BD0	10kΩ	9.7kΩ to 10.3kΩ
C9	BD1	10kΩ	9.7kΩ to 10.3kΩ
C10	BD2	10kΩ	9.7kΩ to 10.3kΩ
C11	BD3	10kΩ	9.7kΩ to 10.3kΩ
C12	BD4	10kΩ	9.7kΩ to 10.3kΩ
C13	BD5	10kΩ	9.7kΩ to 10.3kΩ
C14	BD6	10kΩ	9.7kΩ to 10.3kΩ
C15	BD7	10kΩ	9.7kΩ to 10.3kΩ
C16	BD0	10kΩ	9.7kΩ to 10.3kΩ
C17	BD1	10kΩ	9.7kΩ to 10.3kΩ
C18	BD2	10kΩ	9.7kΩ to 10.3kΩ
C19	BD3	10kΩ	9.7kΩ to 10.3kΩ
C20	BD4	10kΩ	9.7kΩ to 10.3kΩ
C21	BD5	10kΩ	9.7kΩ to 10.3kΩ
C22	BD6	10kΩ	9.7kΩ to 10.3kΩ
C23	BD7	10kΩ	9.7kΩ to 10.3kΩ
BE0	BD0	10kΩ	9.7kΩ to 10.3kΩ
BE1	BD1	10kΩ	9.7kΩ to 10.3kΩ
BE2	BD2	10kΩ	9.7kΩ to 10.3kΩ
BE3	BD3	10kΩ	9.7kΩ to 10.3kΩ
BE4	BD4	10kΩ	9.7kΩ to 10.3kΩ

From	To	Expect	Acceptable Range
BE5	BD5	10kΩ	9.7kΩ to 10.3kΩ
BE6	BD6	10kΩ	9.7kΩ to 10.3kΩ
BE7	BD7	10kΩ	9.7kΩ to 10.3kΩ
ID0	BD0	10kΩ	9.7kΩ to 10.3kΩ
ID1	BD1	10kΩ	9.7kΩ to 10.3kΩ
ID2	BD2	10kΩ	9.7kΩ to 10.3kΩ
ID3	BD3	10kΩ	9.7kΩ to 10.3kΩ
ID4	BD4	10kΩ	9.7kΩ to 10.3kΩ
ID5	BD5	10kΩ	9.7kΩ to 10.3kΩ
ID6	BD6	10kΩ	9.7kΩ to 10.3kΩ
ID7	BD7	10kΩ	9.7kΩ to 10.3kΩ
LED0	BD0	10kΩ	9.7kΩ to 10.3kΩ
LED1	BD1	10kΩ	9.7kΩ to 10.3kΩ
LED2	BD2	10kΩ	9.7kΩ to 10.3kΩ
LED3	BD3	10kΩ	9.7kΩ to 10.3kΩ
LED4	BD4	10kΩ	9.7kΩ to 10.3kΩ
LED5	BD5	10kΩ	9.7kΩ to 10.3kΩ
LED6	BD6	10kΩ	9.7kΩ to 10.3kΩ
LED7	BD7	10kΩ	9.7kΩ to 10.3kΩ
BUFPG	BD0	10kΩ	9.4kΩ to 10.6kΩ
XTALO	BD1	10kΩ	9.4kΩ to 10.6kΩ
XTAL1	BD2	10kΩ	9.4kΩ to 10.6kΩ
1REF	BD0	100kΩ	99.5kΩ to 100.5kΩ
2REF	BD1	100kΩ	99.5kΩ to 100.5kΩ
1REF	GND	4.99kΩ	4.9kΩ to 5.1kΩ
2REF	GND	4.99kΩ	4.9kΩ to 5.1kΩ
VCC1	GND	1.2Ω	1.1Ω to 1.6Ω
VCC2	GND	1.2Ω	1.1Ω to 1.6Ω
VPP1	GND	15Ω	14.6Ω to 15.6Ω
VPP2	GND	15Ω	14.6Ω to 15.6Ω
VNEG	GND	150Ω	141.0Ω to 159.3Ω
+12VS	-12VS	150Ω	141.0Ω to 159.3Ω

B Translation Formats

Introduction

Data translation formats are different ways of encoding the data in a data file. A data file contains the information to be programmed into a board or a device. For example, a data file could contain the data for a memory board or the fuse pattern for a logic device on a board.

Usually, the data file is created on a development platform and is then stored using a particular data translation format. When you want to download the data file to BoardSite, you will need to select the translation format that matches your development platform. During download, BoardSite translates the formatted data and then stores it in the data RAM buffer.

About This Chapter

This chapter describes all the translation formats that BoardSite supports. This chapter does not cover how to select an individual translation format. For information on how to select a translation format, see the section, "Communications," in Chapter 6, "Advanced Operation."

Table B-1 lists all the translation formats supported by BoardSite. The format numbers are included only for reference. BoardSite does not recognize these numbers, but if you have used other Data I/O programmers, you may find the numbers helpful. Following the table is a description and, in most cases, an example of each translation format.

Table B-1
Translation Formats Supported by BoardSite

BoardSite Format Name*	Format Number
Binary	10
DEC Binary	11
Hewlett-Packard 64000 Abs. Obj.	89
Honeywell Standard File Format	n/a
HP-UX	96
Intel 286	97
Intel 386	98
Intel Intellec 8/MDS	83
Intel MCS-86 Hex Obj.	88
Intel HEX-32	99
JEDEC Full Format	91
JEDEC Kernel Mode	92
Motorola 32-bit (S3)	95
Motorola EXORciser	82
Motorola EXORmax	87
Tektronix Hex. Extended	94
Tektronix Hexadecimal	86
ASCII-Hex Space	50

* *When you use the BoardSite Batch command* FORMAT="format_name" *you must type the format name exactly as shown, including punctuation marks and upper/lower case letters. For more information, see the section "Batch," in Chapter 6, "Advanced Operation."*

General Notes

Here is some general information about data translation.

Aborting a Data Transfer

To abort a data transfer at any time, press **Esc** on your PC keyboard.

Compatibility

When translating data, you may use any remote source that produces formats compatible with the descriptions listed in this section.

Formats with Limited Address Fields

Formats 82 (Motorola EXORciser), 83 (Intel Intellec 8/MDS), and 86 (Tektronix Hexadecimal) cannot use address fields greater than 64K bytes. Therefore, if you transfer a block greater than 64K bytes, the data at addresses above the 64K limit may overwrite data already received.

Hardware Handshaking

Hardware handshaking (RTS/CTS) may be used if compatible with the host interface.

Leader/Trailer

During output of all formats except Hewlett-Packard 64000 Abs. Obj., a 50-character leader precedes the formatted data and a 50-character trailer follows. This leader/trailer consists of null characters.

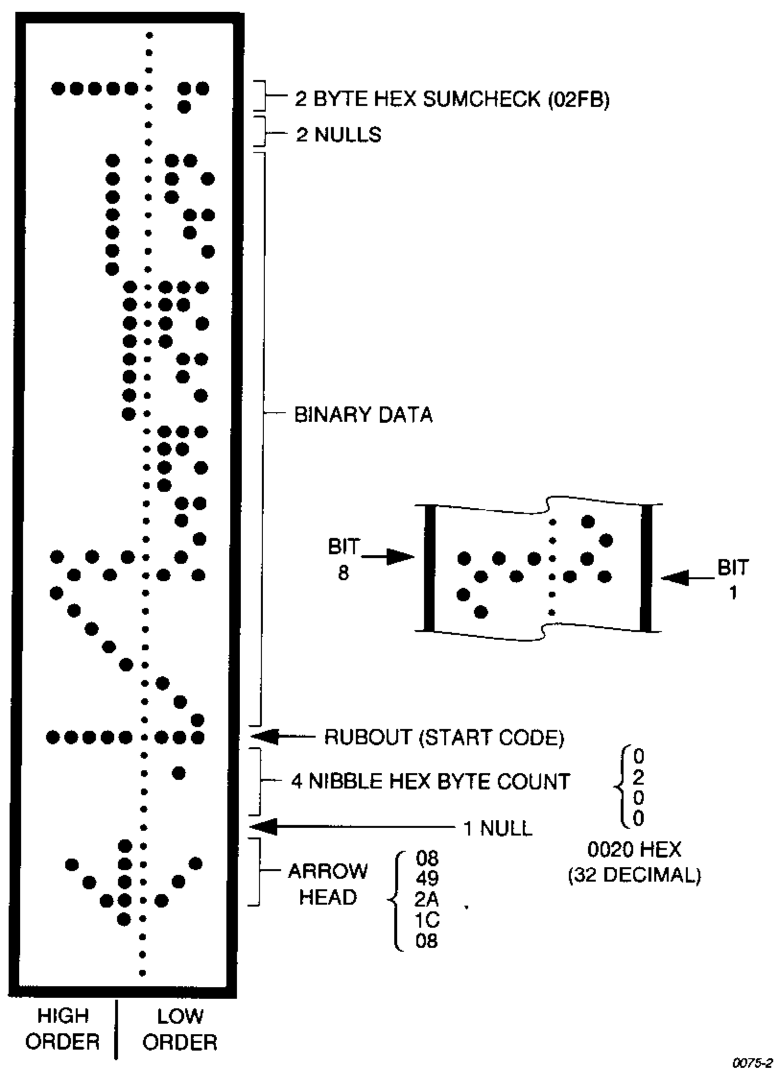
Binary, DEC Binary, and Hewlett-Packard 64000 Abs. Obj. Formats

These formats do not function properly unless you select NO parity, and 8 data bits in the Port Parameters pop-up. For more information on selecting parameters, see the section, "Communications," in Chapter 6, "Advanced Operation."

Binary

Data transfer in the Binary format consists of a stream of 8-bit data words preceded by a byte count and followed by a sumcheck. The Binary format does not have addresses. See Figure B-1.

Figure B-1
An Example of Binary Format

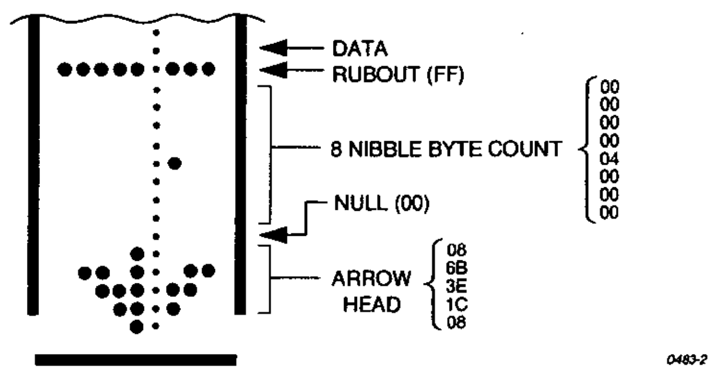


BoardSite stores incoming binary data upon receipt of the start character. Data is stored in RAM starting at the first RAM address specified by the Begin Address parameter (Parameters for Download pop-up) and ending at the last incoming data byte.

A paper tape generated by a programmer will contain a 5-byte, arrow-shaped header followed by a null and a 4-nibble byte count. The start code, an 8-bit rubout, follows the byte count. The end of data is signaled by two nulls and a 2-byte sumcheck of the data field.

If the data output has a byte count GREATER than or equal to 64K, an alternate arrow-shaped header is used. This alternate header is followed by an 8-nibble byte count, surrounded by a null and a rubout. The byte count shown here is 40000H (256K decimal). If the byte count is LESS than 64K, the regular arrowhead is used instead. Data that is input using Binary format will accept either version of this format. See Figure B-2.

Figure B-2
Alternate Binary Format



In addition, a third variation of this format is accepted on download. This variation does not have an arrowhead and is accepted only on download. The rubout begins the format and is immediately followed by the data. There is no byte count or sumcheck.

DEC Binary

Data transmission in the DEC Binary format is a stream of 8-bit data words with no control characters except the start code. The start code is one null preceded by at least one rubout. A file output from BoardSite will contain 32 rubouts in the leader. The DEC Binary format does not have addresses.

Hewlett-Packard 64000 Abs. Obj.

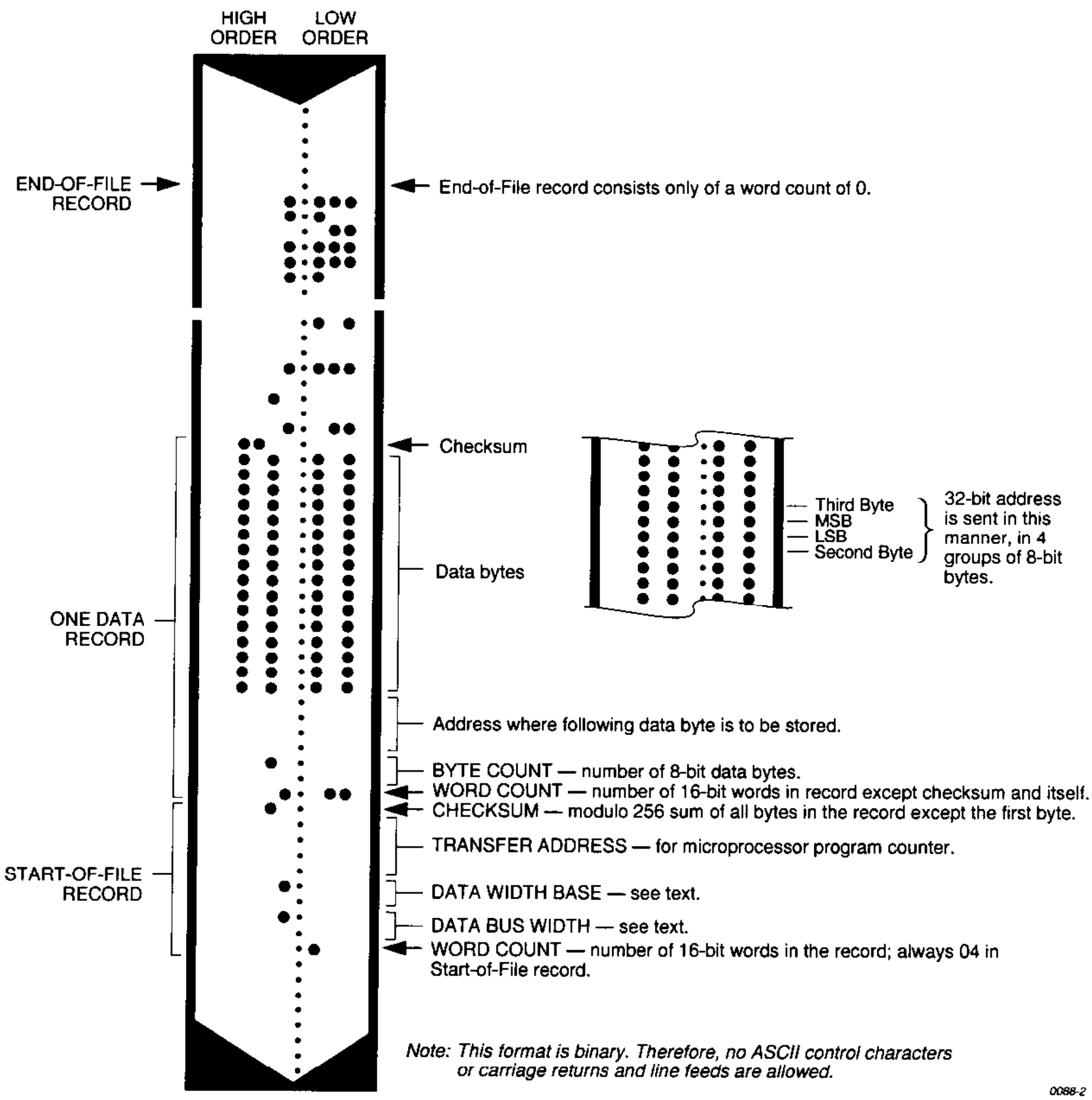
Hewlett-Packard 64000 Abs. Obj. is a binary format with control and data-checking characters. See Figure B-3.

Data files begin with a start-of-file record including the data bus width, data width base, transfer address, and checksum of bytes in the record.

Data records follow the start-of-file record. Each begins with 2 byte counts: the first expresses the number of 16-bit words in the record not including the checksum and itself; the second expresses the number of 8-bit data bytes in the record. Next comes a 32-bit address, which specifies the storage location of the following data byte. Data bytes follow; after the last data byte is a checksum of every byte in the record except the first byte, which is the word count.

The end-of-file record consists only of a 1-byte word count, which is always zero. Leader and trailer nulls, normally 50 each, are suppressed.

Figure B-3
An Example of HP 64000 Abs. Obj. Format



Honeywell Standard File Format

This data translation format is considered, by Honeywell, to be proptietary information. Contact your local Honeywell representative.

HP-UX

HP-UX (Hewlett-Packard UNIX) format divides the data file into data records. Each data record has a maximum size of 250 bytes not including header information. An I.D. header is added to the beginning of the first record. Each subsequent record has its own header section. The section at the beginning of the file contains the following elements: the header, filename, byte count for the processor information record, and the processor information record.

The header identifies the type of file being transferred. The first byte of this header can be 80, 81, 82, or 83. The first byte indicates that this file is binary and the second byte, 04, indicates the type of file (absolute).

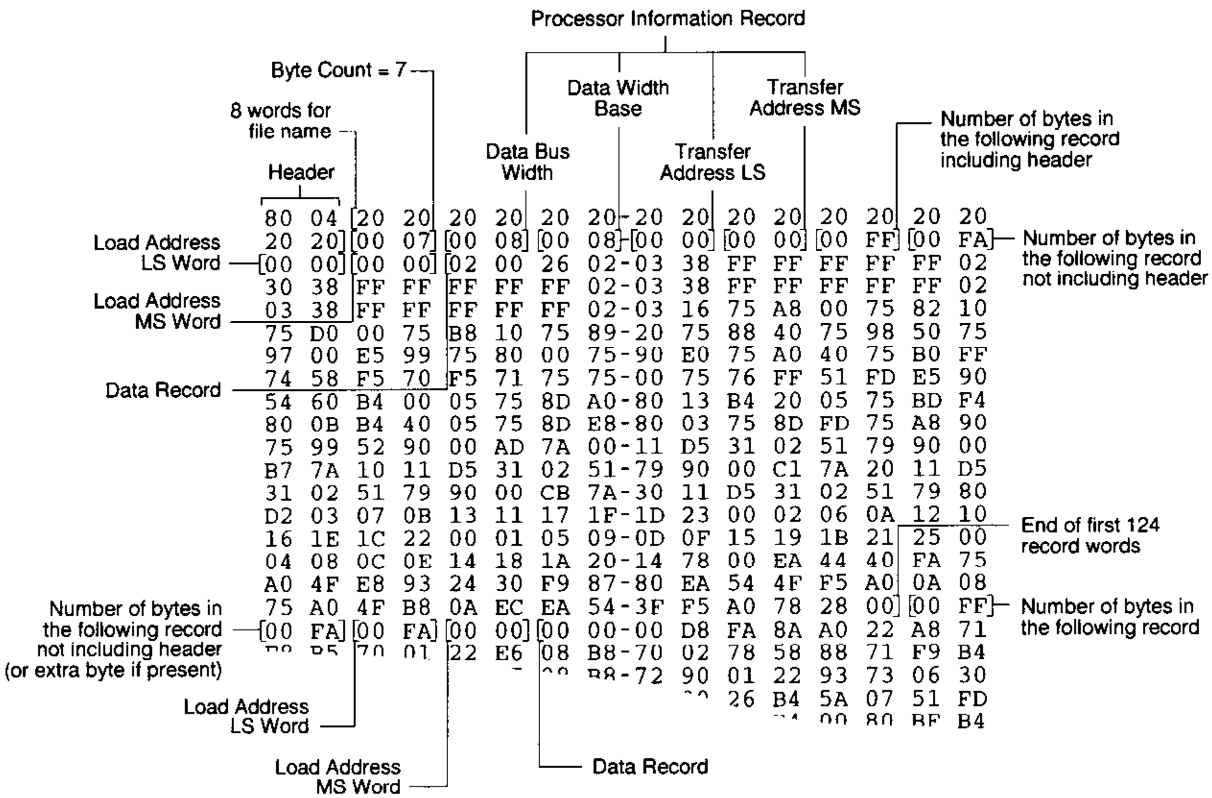
The I.D. header is followed by a sixteen-byte filename (not used by BoardSite).

The byte count which appears next indicates the size (minus one) of the processor information record that follows. The processor information record is divided into the following data words: Data Bus Width, Data Width Base, Transfer Address LS (least significant), and Transfer Address MS (most significant). These four words specify the destination of the entire data file.

The data records follow the processor information record. The data records consist of a header (eight bytes) and the data bytes. The first two bytes of the header indicate the size of the data record including the header (minus one). If the number of data bytes in the data record (not including the header) is odd, one extra byte will be added to the data record to ensure that an even number of data bytes exist in the data record. The maximum value for this field is 00FF hex. The next two bytes indicate the number of actual data bytes in the record, not including the header bytes and the extra byte (if present). The maximum value for this field is 00FA hex. The four bytes that follow represent the destination address for the data in this record. The rest of the bytes in the record are the data bytes. This format has no end of file identifier.

The record length during upload is not affected by the Block Size parameter in the Parameters for Upload pop-up. It is automatically set to transfer records using the maximum size (250 bytes) except for the last record. The size of the last record will be set according to the remaining number of data bytes. See Figure B-4.

Figure B-4
HP-UX Format



This data translation format was generated by a "dump utility" for illustrative purposes. Actual data files are in binary code and are typically generated by the appropriate development software.

0474-2

Intel 286

The Intel 286 format is a dynamically allocatable file format.

This format has three basic parts; the file header, data file module, and a 1-byte checksum. The file header is hexadecimal number (A2) that identifies this file as an Intel 286 format file.

The first 75 bytes of the data file module is the data file header. The header information is generated and used by the development system and is not used by BoardSite, although some characters must fill those bytes. The rest of the data file module consists of one partition. See Figure B-5.

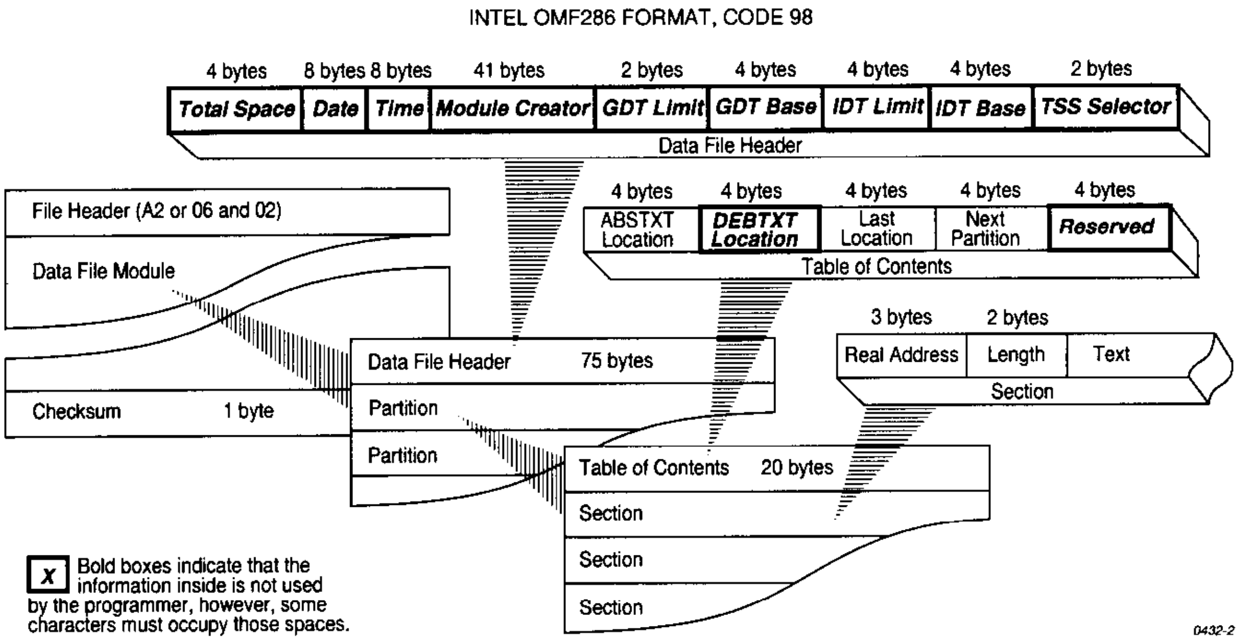
Figure B-5
Intel 286 Format

```
File Header ----[A2] F3 FF FF 00 30 38 2F-30 34 2F 38 37 30 38 3A ]
                 34 33 3A 30 31 1C 69 41-50 58 32 38 36 20 53 59 ]- Data File Header
                 53 54 45 4D 20 42 55 49-4C 44 45 52 2C 20 56 33 ]
                 2E 32 20 20 20 20 20 20-20 20 20 20 20 20 3F 01 ]
Last Location -- 00 80 FF 00 FF 00 40 81-FF 00 18 00              -- ASBTXT Location
                                                      [50 00 00 00] ]- Table of Contents
DEBTXT Location -[5B 66 00 00][6B EF 00 00][00 00 00 00][00 00 00 00] ]
                                                                    -- Reserved
ASBTXT Location-[40 01 00][2C 00] 00 00 00-04 28 00 00 00 00 00 00 ]  -- Next Partition
                 00 00 00 6E 4F 00 02 00-00 00 00 00 00 00 00 00 ]- Section
                 04 00 04 00 00 00 00 28-00 20 00 28 00 28 00 00 ]
                 00
                   [70 01 00][2C 00] 00 00-00 04 28 00 00 00 00 00 ]- Section
                 00 00 00 00 F5 38 00 02-00 00 00 00 00 00 00 00 ]
                 00 00
                      [A0 01 00][2C 00] 00-00 00 04 28 00 00 00 00 ]
                 00 00 00 00 00 33 39 00-02 00 00 00 00 00 00 00 ]- Section
                 00 00 04 00 04 00 00 00-00 28 00 20 00 28 00 28 ]

Length of ASBTXT                        00-00 00 00 04 28 00 00 00
                                           00 00 00 00 00 00 00
```

0431-2

The partition begins with a 20-byte table of contents. The table of contents specifies the locations of ABSTXT (absolute text), DEBTXT (debug text), the last location of this partition, and the location of the next partition. The Intel 286 format consists of only one partition so this field will be zeros. The rest of the partition consists of sections. The actual data is located in the sections. The first 3 bytes in each section specify the real address of the text. The next 2 bytes specify the length of the text and the remainder of the section is the text (or data). Following the final section of the final partition is a 1-byte checksum. BoardSite ignores this checksum. See Figure B-6.

Figure B-6
A Close-up of Intel 286 Format



Intel 386

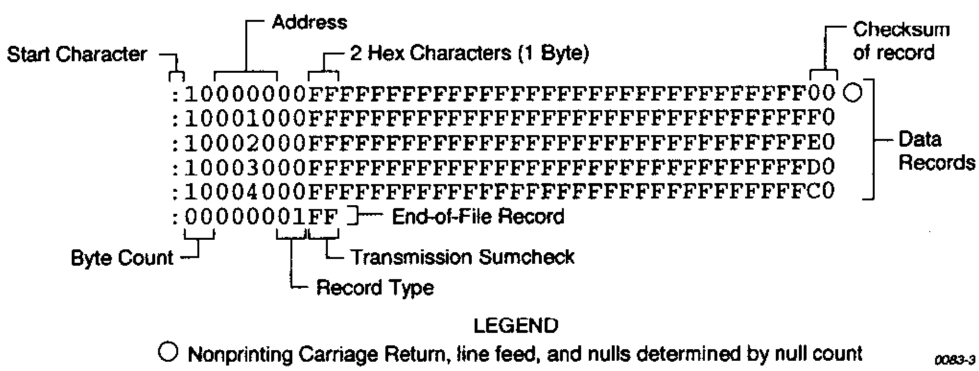
This data translation format is considered, by Intel, to be proprietary information. Contact your local Intel representative or call (408) 987-8080 for information about the structure of this format.

Intel Intellec 8/MDS

Intel Intellec 8/MDS data records begin with a 9-character prefix and end with a 2-character suffix. The byte count must equal the number of data bytes in the record.

Figure B-7 shows a series of valid data records. Each record begins with a colon, which is followed by a 2-character byte count. The 4 digits following the byte count give the address of the first data byte. Each data byte is represented by 2 hexadecimal digits; the number of data bytes in each record must equal the byte count. Following the data bytes of each record is the checksum, the two's complement (in binary) of the preceding bytes (including the byte count, address, record type and data bytes), expressed in hex.

Figure B-7
An Example of Intel Intellec 8/MDS Format

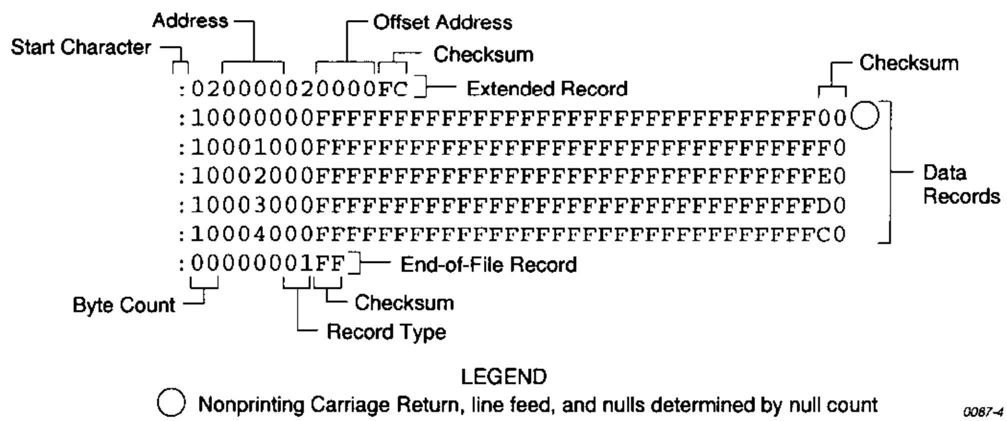


The end-of-file record consists of the "colon" start character, the byte count (equal to "00"), the address, the record type (equal to "01") and the checksum of the record.

Intel MCS-86 Hex Obj.

The Intel MCS-86 Hex Obj. file record format has a 9-character (4-field) prefix that defines the start of record, byte count, load address, and record type, and a 2-character checksum suffix. Figure B-8 shows a sample record of this format.

Figure B-8
An Example of Intel MCS-86 Hex Object



The four record types are described below.

00 – Data Record

This begins with the colon start character, which is followed by the byte count (in hex notation), the address of the first data byte, and the record type (equal to "00"). Following these are the data bytes. The checksum follows the data bytes and is the two's complement (in binary) of the preceding bytes in the record, including the byte count, address, record type and data bytes.

01 – End Record

This end-of-file record also begins with the colon start character. This is followed by the byte count (equal to "00"), the address (equal to "0000"), the record type (equal to "01") and the checksum, "FF".

02 – Extended Segment Address Record

This is added to the offset to determine the absolute destination address. The address field for this record must contain ASCII zeros (Hex 30). This record type defines bits 4 to 19 of the segment base address; it can appear randomly anywhere within the object file and affects the absolute memory address of subsequent data records in the file. The following example illustrates how the extended segment address is used to determine a byte address.

Problem:

Find the address for the first data byte for the following file.

```
: 02 0000 02 1230 BA
: 10 0045 00 55AA FF .....BC
```

Solution:

Step 1. Find the record address for the byte. The first data byte is 55. Its record address is 0045 from above.

Step 2. Find the offset address. The offset address is 1230 from above.

Step 3. Shift the offset address one place left, then add it to the record address, like this:

1230	Offset address (upper 16 bits)
+ 0045	Record address (lower 16 bits)
12345	20-bit address

The address for the first data byte is **12345**.

Note: Always specify the Offset parameter when using this format, even when the offset is zero.

During upload, BoardSite forces the record size to 16 (decimal) if the record size is specified greater than 16. There is no such limitation for record sizes specified less than 16.

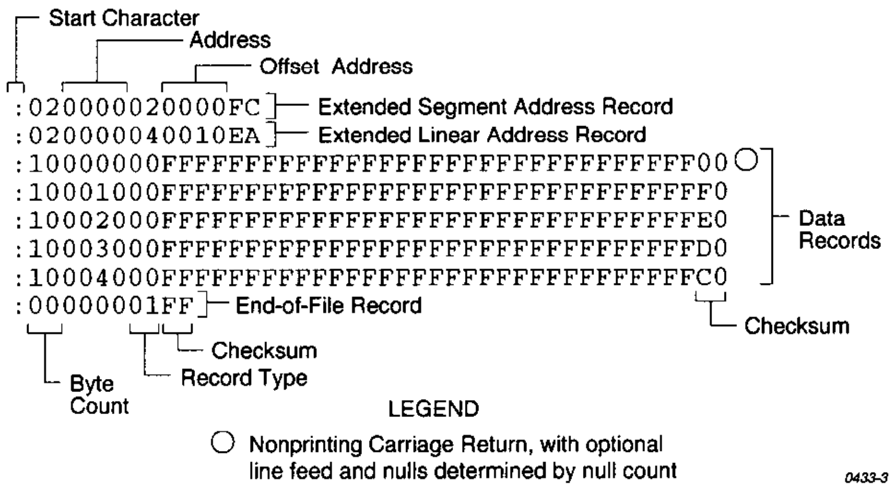
03 – Start Record

This record type is not sent by BoardSite during upload.

Intel HEX-32

The Intel HEX-32 format has a 9-character (4-field) prefix that defines the start of record, byte count, load address, and record type and a 2-character checksum suffix. Figure B-9 illustrates the sample records of this format.

Figure B-9
An Example of the Intel HEX-32 Format



The six record types are described below.

00 – Data Record

This record begins with the colon start character, which is followed by the byte count (in hex notation), the address of the first data byte, and the record type (equal to "00"). Following these are the data bytes. The checksum follows the data bytes and is the two's complement (in binary) of the preceding bytes in the record, including the byte count, address, record type and data bytes.

01 – End Record

This end-of-file record also begins with the colon start character and is followed by the byte count (equal to "00"), the address (equal to "0000"), the record type (equal to "01") and the checksum, "FF".

02 – Extended Segment Address Record

This is added to the offset to determine the absolute destination address. The address field for this record must contain ASCII zeros (Hex 30). This record type defines bits 4 to 19 of the segment base address; it can appear randomly anywhere within the object file and affects the absolute memory address of subsequent data records in the file. The following example illustrates how the extended segment address is used to determine a byte address.

Problem

Find the address for the first data byte for the following file.

```
:02 0000 04 0010 EA
:02 0000 02 1230 BA
:10 0045 00 55AA FF ..... BC
```

Solution:

Step 1. Find the extended linear address offset for the data record (**0010** in the example).

Step 2. Find the extended segment address offset for the data record (**1230** in the example).

Step 3. Find the address offset for the data from the data record (**0045** in the example).

Step 4. Calculate the absolute address for the first byte of the data record as follows:

`00100000`	Linear address offset, shifted left 16 bits
`+ 12300`	Segment address offset, shifted left 4 bits
`+  0045`	Address offset from data record
`00112345`	32-bit address for first data byte

The address for the first data byte is **112345**.

Note: Always specify the address offset when using this format, even when the offset is zero.

During upload, BoardSite forces the record size to 16 (decimal) if the record size is specified greater than 16. There is no such limitation for record sizes specified less than 16.

03 – Start Segment Address Record

This record, which specifies bits 4-19 of the execution start address for the object file, is not used by BoardSite.

04 – Extended Linear Address Record

This record specifies bits 16-31 of the destination address for the data records that follow. It is added to the offset to determine the absolute destination address, and can appear randomly anywhere within the object file. The address field for this record must contain ASCII zeros (Hex 30).

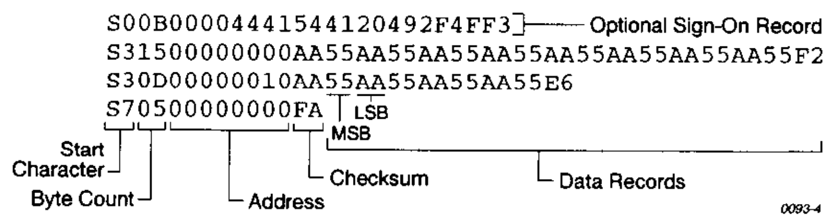
05 – Start Linear Address Record

This record, which specifies bits 16-31 of the execution start address for the object file, is not used by BoardSite.

Motorola 32-bit (S3)

The Motorola 32-bit (S3) form(Code 95) closely resembles the Motorola EXORmax format, the main difference being the addition of the "S3" and "S7" start characters. The "S3" character is used to begin a record containing a 4-byte address. The "S7" character is a termination record for a block of "S3" records. The address field for an "S7" record may optionally contain the 4-byte instruction address that identifies where control is to be passed and is ignored by BoardSite. A sample transmission is shown in Figure B-12.

Figure B-10
Motorola 32-bit (S3) Format



Motorola 32-bit (S3) data files may begin with an optional sign-on record, initiated by the start characters "S0." Data records start with an 8- or 10-character prefix and end with a 2-character suffix.

Each data record begins with the start characters "S1", "S2" or "S3"; "S1" if the following address field has 4 characters, S2 if it has 6 characters, S3 if it has 8 characters. The third and fourth characters represent the byte count, which expresses the number of data, address and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix (6 characters for addresses above hexadecimal FFFF and 8 characters for addresses above hexadecimal FFFFFF). Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be 3, 4 or 5 less than the byte count. The suffix is a 2-character checksum, the one's complement (in binary) of the preceding bytes in the record, including the byte count, address and data bytes.

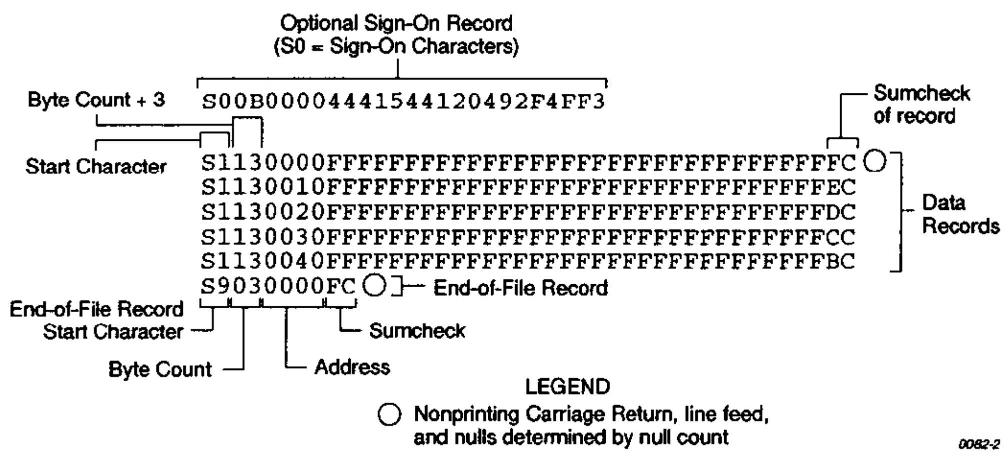
The end-of-file record begins with an "S9" start character. Following the start characters are the byte count, the address and a checksum. The maximum record length is 250 data bytes.

Note: *Motorola and Intel formats swap the data byte order. Motorola formats have MSB first, lower address, and left side when displayed in editor.*

Motorola EXORciser

Motorola EXORciser data files may begin with an optional sign-on record, which is initiated by the start characters "S0". Valid data records start with an 8-character prefix and end with a 2-character suffix. Figure B-13 shows a series of valid Motorola EXORciser data records.

Figure B-11
An Example of Motorola EXORciser Format



Each data record begins with the start characters "S1". The third and fourth characters represent the byte count, which expresses the number of data, address and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix. Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be three less than the byte count. The suffix is a 2-character checksum, which equals the one's complement of the binary summation of the byte count, address and data bytes.

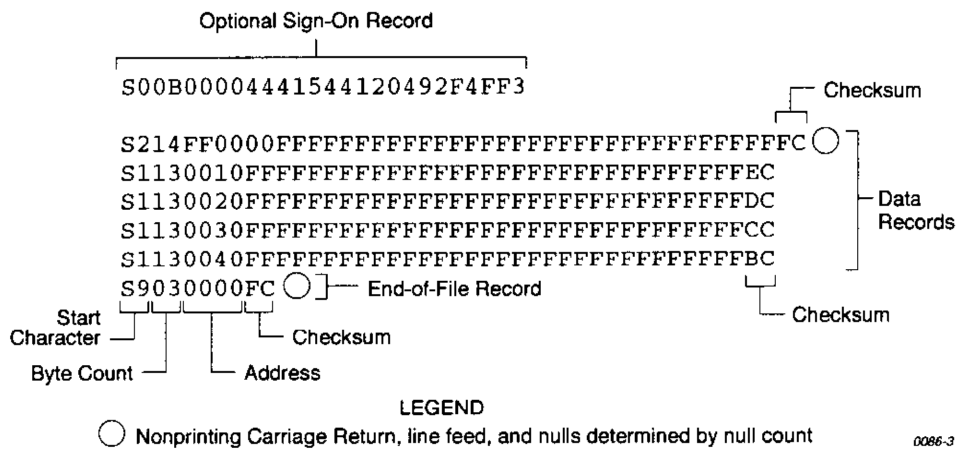
The end-of-file record consists of the start characters "S9", the byte count, the address (in hex) and a checksum. The maximum record length is 250 data bytes.

Note: Motorola and Intel formats swap the data byte order. Motorola formats have MSB first, lower address, and left side when displayed in editor.

Motorola EXORmax

Motorola EXORmax data files may begin with an optional sign-on record, initiated by the start characters "S0". Data records start with an 8- or 10-character prefix and end with a 2-character suffix. Figure B-14 shows a series of Motorola EXORmax data records.

Figure B-12
An Example of Motorola EXORmax Format



Each data record begins with the start characters "S1" or "S2"; "S1" if the following address field has 4 characters, S2 if it has 6 characters. The third and fourth characters represent the byte count, which expresses the number of data, address and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix (6 characters for addresses above hexadecimal FFFF). Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be 3 or 4 less than the byte count. The suffix is a 2-character checksum, the one's complement (in binary) of the preceding bytes in the record, including the byte count, address and data bytes.

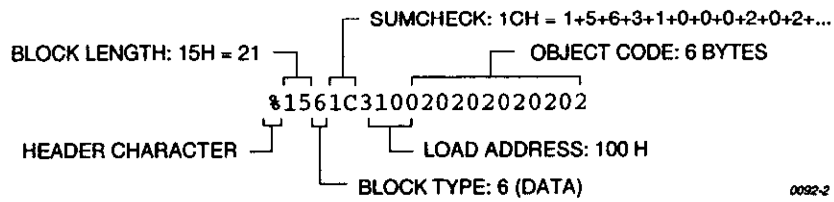
The end-of-file record begins with an "S9" start character. Following the start characters are the byte count, the address and a checksum. The maximum record length is 250 data bytes.

Note: Motorola and Intel formats swap the data byte order. Motorola formats have MSB first, lower address, and left side when displayed in editor.

Tektronix Hex. Extended

The Tektronix Hex. Extended format has three types of records: data, symbol and termination. The data record contains the object code. Information about a program section is contained in the symbol record (BoardSite ignores symbol records) and the termination record signifies the end of a module. The data record (see sample below) contains a header field, a load address and the object code. The header field contains the information shown in Figure B-15.

Figure B-13
Tektronix Hex. Extended Format



Item	No. of ASCII Characters	Description
%	1	Signifies that the record is the Extended Tek Hex format.
Block length	2	Number of characters in the record, minus the %.
Block type	1	6 = data record 3 = symbol record (ignored by BoardSite) 8 = termination record
Checksum	2	A 2-digit hex sum, modulo 256, of all the values in the record except the % and the checksum.

Character Values for Checksum Computation

Characters	Values (decimal)
0 . . 9	0 . . 9
A . . Z	10 . . 35
$	36
%	37
. *(period)*	38
_ *(underline)*	39
a . . z	40 . . 65

The number of fields in the file will vary, depending on whether a data or a termination block is sent. Both data and termination blocks have a 6-character header and a 2-to-17 character address.

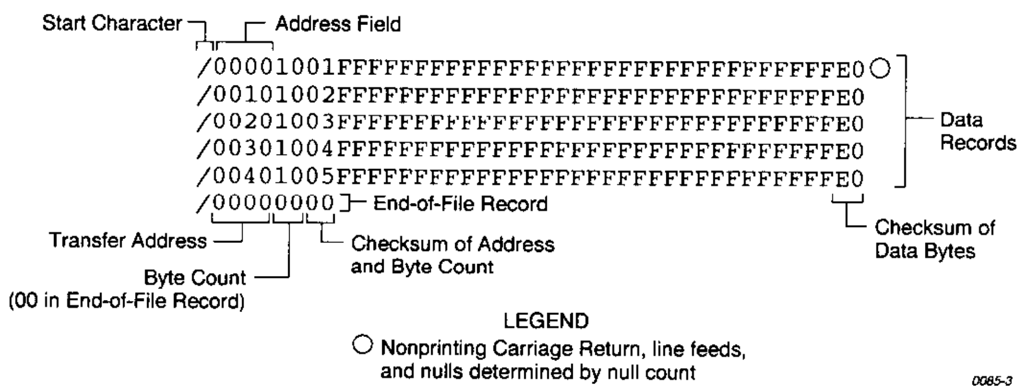
The load address determines where the object code will be located. This is a variable length number that may contain up to 17 characters. The first number determines the address length, with a zero signifying a length of 16. The remaining characters of the data record contain the object code, 2 characters per byte.

When you are uploading or downloading, make sure to set the Offset Hi parameter if the low-order address is not at the default value.

Tektronix Hexadecimal

Figure B-16 shows a valid Tektronix Hexadecimal data file. The data in each record is surrounded by the start character (a slash) and a 2-character checksum. Following the start character, the next 4 characters of the prefix express the address of the first data byte. The address is followed by a byte count, which represents the number of data bytes in the record, and by a checksum of the address and byte count. Data bytes follow, represented by pairs of hexadecimal characters. Succeeding the data bytes is their checksum, an 8-bit sum, modulo 256, of the 4-bit hexadecimal values of the digits making up the data bytes. All records are followed by a carriage return.

Figure B-14
An Example of Tektronix Hex Format



Data is output from BoardSite starting at the first RAM address and continuing until the number of bytes in the specified block have been transmitted. BoardSite divides output data into records prefaced by a start character and an address field for the first byte in the record.

The end-of-file record consists of a start character (slash), followed by the transfer address, the byte count (equal to "00"), and the checksum of the transfer address and byte count.

An optional abort record contains 2 start characters (slashes), followed by an arbitrary string of ASCII characters. Any characters between a carriage return and a "/" are ignored.

ASCII-Hex Space

This format has a start and end code, and address and checksum specifications. Figure B-17 shows 4 data bytes coded in the ASCII-Hex Space format. Data in this format is organized into sequential bytes separated by the space character. Characters immediately preceding the space character are interpreted as data. The format expresses 8-bit data using 2 hexadecimal characters. Line feeds, carriage returns and other characters may be included in the data stream as long as a data byte directly precedes each space character.

Figure B-15
An Example of ASCII – Hex Space Format

```
                      Optional Hex Address Field
FORMAT 50 ①  $A0000,
             FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ②
             $S0FF0,
                      Optional Hex Sumcheck Field
```

LEGEND

① Start Code is nonprintable STX - CTRL B (optionally SOH - CTRL A)

② End Code is nonprintable ETX - CTRL - C

0657-1

Although each data byte has an address, most are implied. Data bytes are addressed sequentially unless an explicit address is included in the data stream. This address is preceded by a "$" and an "A", must contain 2 to 8 hex characters, and must be followed by a comma. BoardSite skips to the new address to store the next data byte; succeeding bytes are again stored sequentially.

The format has an end code, which terminates input operations. However, if a new start code follows within 16 characters of an end code, input will continue uninterrupted. If no characters come within 2 seconds, input operation is terminated.

After receiving the final end code following an input operation, BoardSite calculates a sumcheck of all incoming data. Optionally, a sumcheck can also be entered in the input data stream. BoardSite compares this sumcheck with its own calculated sumcheck. If they match, BoardSite will display the sumcheck; if not, a sumcheck error will be displayed.

Note: *The sumcheck field consists of 2 to 4 hex characters, surrounded by "$" and "," characters. The sumcheck immediately follows an end code. The sumcheck is optional in download, but is always included in upload.*

BoardSite divides the output data into 8-line blocks. Data transmission is begun with the start code, a non-printable STX character. Data blocks follow, each one prefaced by an address for the first data byte in the block. The end of transmission is signaled by the end code, a non-printable ETX character. Directly following the end code is a sumcheck of the transferred data.

Index

C

D

I

J

K

L

R

S

T

U

V

W

X

BoardSite™

In-Circuit Programmer

V5.2 Device List

DATA I/O

Device List Version 5.2

The Device List describes all the devices that can be programmed by BoardSite as of Version 5.2. The devices are listed alphabetically by manufacturer and in order of device size. Insert this Device List behind the Device List divider tab in your *BoardSite User Manual* and discard other BoardSite device lists.

Data I/O Device Support Policy/Liability

1. Data I/O strives to achieve more device support approvals from semiconductor manufacturers than any other programmer manufacturer or software developer.
2. Every effort is made to program an adequate number of samples according to the manufacturer-supplied specification, and verify waveforms as per that specification prior to release of support. Manufacturers' approvals are sought in parallel with this process.
3. Data I/O's objective is to seek and obtain approvals on all devices.
4. Data I/O has made every attempt to ensure that the device information (as provided by the device manufacturer) contained in our programmers, software and documentation is accurate and complete. However, Data I/O assumes no liability for errors, or for any damages, whether direct, indirect, consequential or incidental, that result from use of documents provided with equipment or from equipment or software which it accompanies, regardless of whether or not Data I/O has been advised of the possibility of such loss or damage.

Key to Device List Headings and Footnotes

Manufacturer The name of the manufacturer of the device.

Device Part Number The number assigned by the device manufacturer.

Family Code A 3-digit hexadecimal number that designates the programming algorithm.

Software Version A number specifying the earliest version of BoardSite software that will program the device to the manufacturer's latest specifications. The version number of a previously supported device changes if the programming specifications change.

Device Category A number in this column assigns the device to one of four device type categories. Each category is described in the *BoardSite User Manual*.

Notes A letter in this column corresponds to a footnote, which contains special information about the device. The footnotes are described below.

a These devices have an active high PGM input; you may have to change the PGM polarity in the Board Profile.

b These devices use high-speed outputs to achieve fast access times, so it may be necessary to add a single address transition just after powerup in the device algorithm. Refer to the device specification sheet for more information.

c Programming these FLASH devices requires special BoardSite Interface Adapter hardware. Please contact Data I/O as listed in the Preface of the User Manual for an Application Note on FLASH device support.

d Programming these microprocessors requires some additional BoardSite Interface hardware. Please read the Application Note contained on the BoardSite disks for special hardware considerations. The Application Note is titled "Programming the 8748H and 8749H" and the file name is **8748h.app**.

e The Lattice 16Z8 requires unique programming considerations. Please read Application Note "Lattice 16Z8 Device Support" contained on the BoardSite disks for special instructions regarding programming this device. The filename is **logic.app**.

f These devices support page-mode programming and use the CE pin to determine if the programming operation will be done in page or byte mode. Because the CE pin is not functioning as an "enable" in this case, we will treat these as Category 2 rather than Category 1 devices.

g Intel has released new versions of the 87C51, 87C51FA, and 87C51FB microprocessors. These new versions are called 87C51(FX), 87C51FA(FX), and 87C51FB(FX) respectively. The new devices use the same programming algorithm as the 87C51FC and should not be confused with their older verisons. The newer devices have the same name and can be distinguised by examining the "po" number below the device name. On newer FX-coredevices the last character of this number is "A."

h AMD released new versions of the 28F010. These new devices use the same algorithm as the 28F020 and should not be confused with the older versions. The newer devices have the same name and can be distinguised by examining the "po" number below the device name. The older devices have a C3 in this number. These older devices must be supported according the footnote c. For customers supporting the 28F010, the older devices can still be supported by selecting the non-preferred family code, designated by an * following the family code under Device Type in the Algorithm Information Form of the Board Profile Editor.

Device Part Number	Family Code	Software Version	Device Category	Notes

Advanced Micro Devices / MMI

27128	AF	V1	1	
27128A	C1	V1	1	
27128AP	D6	V1	1	
2716	19	V1	2	a
2716B	C2	V3	2	a
27256	C1	V1	2	
27256P	D6	V1	2	
2732	19	V1	3	
2732A	27	V1	3	
2732B	C2	V3	3	
27512	DD	V1	3	
27512P	DA	V1	3	
2764	AF	V1	1	
2764A	C1	V1	1	
2764AP	D6	V1	1	
27C010	D6	V4.3	1	
27C020	D6	V3	1	
27C040	D6	V4.3	2	
27C100	D6	V3	1	
27C1024	5F	V1.2	1	
27C128	D6	V4.3	1	
27C191	EA	V4.2	4	
27C2048	5F	V4.1	1	
27C256	D6	V4.3	2	
27C256P	D6	V1	2	
27C400	5F	V5.1	2	
27C4096	5F	V5.1	2	
27C43	EA	V4.2	4	
27C49	EA	V4.2	4	
27C512	D6	V4.3	3	
27C512P	DA	V1	3	
27C64	D6	V4.3	1	
27H010	D6	V4.3	1	
27HB010	D5C	V5.1	1	
2817A	C3	V1.4	1	
2864A	B7	V1.4	1	
2864AE	B7	V1.4	1	
2864B	B7	V1.4	1	
2864BE	B7	V1.4	1	
28F010	26F	V5.1	1	h
28F010(R)	26F	V5.1	1	h
28F020	273	V5.1	1	
28F020A	15B	V5.2	1	
28F256	135	V51	1	c
28F512	135	V5.0	1	c
29F010	15B	V5.2	1	
29F016	15B	V5.2	1	
29F040	15B	V5.2	1	
29F080	15B	V5.2	1	
29F100B/T8	400	V5.2	1	

Device Part Number	Family Code	Software Version	Device Category	Notes
29F200B/T8	400	V5.2	1	
29F400B/T8	400	V5.2	1	
29F100B/T16	402	V5.2	1	
29F200B/T16	402	V5.2	1	
29F400B/T16	402	V5.2	1	
8751H	54	V1.4	1	
8753H	54	V1.4	1	
87C51	5A	V4.1	1	
87C521	5A	V4.1	1	
87C541	5A	V4.1	1	
9864	C3	V1.4	1	

Atmel Corporation

Device Part Number	Family Code	Software Version	Device Category	Notes
27256	93	V1	2	
27C010	115	V3	1	
27C040	115	V5.0	2	
27C1024	141	V3	1	
27C128	93	V1.1	1	
27C256	93	V1	2	
27C256R	115	V4.3	2	
27C512	4B	V1	3	
27C512R	116	V2	3	
27C64	93	V1	1	
27HC1024	141	V5.0	1	
27HC256	93	V1	2	
27HC64	93	V1	1	
27HC641	90	V4.2	4	
27HC642	90	V4.2	4	
28C010	BA	V5.1	1	
28C04	C4	V1.4	1	
28C16	C4	V1.4	1	
28C17	C4	V1.4	1	
28C256	B7	V1.4	1	
28C64	C4	V1.4	1	
28HC16	C4	V4.1	1	
28HC256	B7	V1.4	1	
28HC64	C4	V1.4	1	
28MC010	B7	V4.1	1	
28PC64	C4	V1.4	1	
29C010	3ED	V5.0	1	
29C020	3ED	V5.2	1	
29C040	3ED	V5.2	1	
29C040A	3ED	V5.2	1	
29C1024	29C	V5.2	1	
29C256	BA	V4.2	1	
29C512	3ED	V5.1	1	
89C52	2B7	V5.2	1	

Catalyst Semiconductor

Device Part Number	Family Code	Software Version	Device Category	Notes
27010	5C	V3	1	
27128A	5C	V1.1	1	
27256	5C	V1.1	2	
27512	5E	V1.1	3	
2764A	5C	V1.1	1	

Device Part Number	Family Code	Software Version	Device Category	Notes
27C210	5F	V3	1	
27HC256	5C	V3	2	
27HC010	5C	V5.2	1	
28C16AK	B7	V4.1	1	
28C16AN	B7	V4.1	1	
28C16AP	B7	V4.1	1	
28C17A	B7	V1.4	1	
28C64A	B7	V4.1	1	

Cypress Semiconductor

Device Part Number	Family Code	Software Version	Device Category	Notes
27HC010	4E	V5.2	1	
7C225	F0	V3	3	
7C235	F0	V3	3	
7C245	F4	V3	3	
7C245A	10B	V2	3	
7C251	EB	V3	2	
7C254	EB	V3	2	
7C261	EF	V5.1	2	
7C263	EF	V5.1	2	
7C264	EF	V5.1	2	
7C266	EF	V5.2	1	
7C271	142	V5.0	1	
7C274	142	V5.0	2	
7C291	F2	V3	3	
7C291A	10C	V3	3	
7C292	F2	V3	3	
7C292A	10C	V3	3	
7C293A	10C	V4.1	3	

Exel Microelectronics, Inc.

Device Part Number	Family Code	Software Version	Device Category	Notes
2804	B7	V1.1	1	
2816A	B7	V1.1	1	
2864A	B7	V1.4	1	
2865A	B7	V1.4	1	

Fujitsu Microelectronics, Inc.

Device Part Number	Family Code	Software Version	Device Category	Notes
27128	45	V1.4	1	
27256	93	V1	2	
2732	19	V1	3	
2732A	27	V1	3	
2764	45	V1.4	1	
27C1000	6C	V1	2	f
27C1000A	1FE	V5.1	2	f
27c1001	6C	V1	2	f
27C1024	6D	V1	2	f
27C1024A	200	V5.1	2	f
27C128	45	V1.4	1	
27C2000	1FE	V5.1	2	f
27C2001	1FE	V5.1	2	f
27C2048	200	V5.1	2	f
27C256	45	V1.4	2	

Device Part Number	Family Code	Software Version	Device Category	Notes
27C256A	93	V1	2	
27C256H	93	V1	2	
27C32A	27	V1	3	
27C4000	1A8	V5.1	2	
27C4001	1A8	V5.1	2	
27C4096	1A9	V5.1	2	
27C512	4B	V1.1	3	
27C64	45	V1.4	1	
28C64	B7	V1.4	1	
28C65	B7	V1.4	1	
28F010	18C	V5.1	1	
8516	19	V1	2	a

Hitachi, Ltd.

Device Part Number	Family Code	Software Version	Device Category	Notes
27128A	93	V1	1	
27256	93	V1	2	
27512	4B	V1	3	
27C101	8F	V1.1	2	f
27C101-PG	F8F	V5.1	2	f
27C1024	8E	V1.1	2	f
27C256	93	V1	2	
27C301	8F	V1.1	2	f
27C301A	8F	V4.1	2	f
27C4096	163	V5.1	2	
27C64	79	V1	1	
28F101	21D	V5.1	1	
462532	19	V1	3	
462716	19	V1	2	a
462732	19	V1	3	
4827128	79	V1	1	
482732A	27	V1	3	
482764	79	V1	1	
58C1001	246	V5.1	1	
58C256	B7	V4.1	1	
58C65	C3	V1.4	1	

Integrated Device Technology, Inc.

Device Part Number	Family Code	Software Version	Device Category	Notes
78C16A	B7	V1.4	1	

Intel Corporation

Device Part Number	Family Code	Software Version	Device Category	Notes
27010	5C	V1	1	
27011	5C	V5.2	1	
27128	79	V1	1	
27128A	93	V1	1	
27128B	93	V1.1	1	
2716	19	V1	2	a
27210	5F	V1	1	
27256	93	V1	2	
2732	19	V1	3	
2732A	27	V1	3	

Device Part Number	Family Code	Software Version	Device Category	Notes
27512	4B	V1	1	
2764	79	V1	1	
2764A	93	V1	1	
27C010	5C	V1	1	
27C010A	160	V4.2	1	
27C020	5C	V3	1	
27C040	5C	V4.2	2	
27C100	5C	V4.2	1	
27C128	5C	V1	1	
27C210	5F	V1.3	1	
27C220	5F	V1.3	1	
27C240	5F	V3	2	
27C256	5C	V1	2	
27C256A	5C	V1	2	
27C400	5F	V4.3	2	
27C512	5E	V1.1	3	
27C64	93	V1	1	
2816	37	V5.1	1	
2816A	A5	V1.4	1	
2817A	C3	V1.4	1	
2864A	B7	V1.4	1	
28F001BX-B	19B	V5.1	1	
28F001BX-T	19C	V5.1	1	
28F002BX-B	400	V5.2	1	
28F002BX-T	401	V5.2	1	
28F004BX-B	400	V5.2	1	
28F004BX-T	401	V5.2	1	
28F008SA	208	V5.1	1	
28F010	18C	V5.0	1	
28F010-P1	186	V5.0	1	c
28F010-R	18C	V5.0	1	
28F016SA	416	V5.2	1	
28F020	18C	V5.0	1	
28F020-R	18C	V5.0	1	
28F032SA	416	V5.2	1	
28F200BX-B8	400	V5.2	1	
28F200BX-T8	401	V5.2	1	
28F200BX-B16	402	V5.2	1	
28F200BX-T16	403	V5.2	1	
28F256-P1	186	V5.0	1	c
28F256-P2	A8	V5.0	1	c
28F256A	18C	V5.0	1	
28F400BX-B8	400	V5.2	1	
28F400BX-T8	401	V5.2	1	
28F400BX-B16	402	V5.2	1	
28F400BX-T16	403	V5.2	1	
28F512	18C	V5.1	1	
8744	53	V1.3	1	
8744H	D5	V1.3	1	
8748H	50	V4	1	d

Device Part Number	Family Code	Software Version	Device Category	Notes
8749H	50	V4	1	d
8751	53	V1.3	1	
8751BH	5A	V1.3	1	
8751H	D5	V1.3	1	
8752BH	5A	V1.3	1	
8797BH	D8	V1.4	1	
87C196KC	166	V5.1	1	
87C196KD	166	V5.1	1	
87C257	5C	V4.2	2	
87C51	5A	V1.3	1	
87C51(FX)	156	V5.0	1	g
87C51FA	5A	V1.3	1	
87C51FA(FX)	156	V5.0	1	g
87C51FB	5A	V1.3	1	
87C51FB(FX)	156	V5.0	1	g
87C51FC	156	V4.1	1	
E28F008SA	208	V5.1	1	
E28F008SA-L	208	V5.1	1	
F28F008SA	208	V5.1	1	
F28F008SA-L	208	V5.1	1	
P27128A	5C	V1	1	
P27256	5C	V1	2	
P2764A	5C	V1	1	
P27C256	5C	V1.1	2	
P27C64	5C	V1.1	1	
P8748H	50	V4.1	1	
PA28F008SA	208	V5.1	1	
PA28F008SA-L	208	V5.1	1	

International CMOS Technology, Inc.

27CX010	5C	V4.1	1	

Lattice Semiconductor

16Z8	36	V4	1	e

Macronix

27C256	5C	V5.1	2	
27C512	5C	V5.1	3	

Matsushita Electronics Corporation

27128	79	V1	1	
2764	79	V1	1	

Microchip Technology Inc.

24C02/A	120	V5.1	1	
24C02/A-SM	120	V5.1	1	
24C02/A-SN	120	V5.1	1	
24C04/A	120	V5.1	1	
24C04/A-SM	120	V5.1	1	

Device Part Number	Family Code	Software Version	Device Category	Notes
24C04/A-SN	120	V5.1	1	
24LCS52		V5.2		
27256	5C	V1	2	
27C128	187	V5.0	1	
27C256	187	V5.0	2	
27C512	187	V5.1	3	
27C64	187	V5.0	1	
27HC256	187	V5.0	2	
27HC64	187	V5.0	1	
28C04/A	C4	V1.4	1	
28C16/A	C4	V1.4	1	
28C17/A	C4	V1.4	1	
28C256	BA	V5.1	1	
28C64/A	C4	V1.4	1	
28CP256	B7	V2	1	
28CP64	B7	V2	1	
93C46	1F5	V5.0	1	

Mitsubishi Electronics of America

27128	79	V1	1	
2716	19	V1	2	a
27256	93	V1	2	
2732	19	V1	3	
27512	4B	V1	3	
2764	79	V1	1	
27C100	8F	V1.3	2	f
27C100-PG	F8F	V5.1	2	f
27C101	8F	V1.3	2	f
27C101-PG	F8F	V5.1	2	f
27C102	8E	V1.3	2	f
27C201	8F	V3	2	f
27C201-PG	F8F	V5.0	2	f
27C202	8E	V3	2	f
27C256	8C	V4.1	2	
27C256A	8C	V4.1	2	
27C401	5C	V4.3	2	
27C402	5F	V4.3	2	
27C512A	5E	V2	3	

Mostek

2716	19	V1	2	a

Motorola Inc.

2532	19	V1	3	
2716	19	V1	2	a

NEC Electronics Corporation

27128	79	V1	1	
27128B	79	V1.1	1	
2716	19	V1	2	a
27256	45	V1.4	2	
27256A	93	V1	2	

Device Part Number	Family Code	Software Version	Device Category	Notes
2732	19	V1	3	
2732A	27	V1	3	
27512	5E	V1	3	
2764	79	V1	1	
27C1000	71	V5.1	2	f
27C1000A	71	V5.1	2	f
27C1001	71	V5.1	2	f
27C1001A	71	V5.1	2	f
27C1024	6F	V5.1	2	f
27C1024A	6F	V5.1	2	f
27C128	45	V1.4	1	
27C2001	71	V5.1	2	f
27C256	45	V1.4	2	
27C256A	71	V5.1	2	
27C4000	6F	V5.1	2	
27C4001	71	V5.1	2	
27C4096	6F	V5.1	2	
27C512	4E	V5.1	3	
27C64	79	V1	1	
28C256	B7	V4.1	1	
28C64	B7	V1.4	1	
8748H	50	V4.1	1	
8749H	50	V4.1	1	

National Semiconductor Corporation

24C02L	120	V5.2		
2532	19	V1	3	b
2716	19	V1	2	a,b
2732	19	V1	3	b
2758A	19	V1	2	a,b
2758B	19	V1	2	a,b
27C010	5C	V1.1	1	b
27C040	5C	V5.0	2	b
27C1024	5F	V1.1	1	b
27C128	5D	V1.2	1	b
27C128B	5C	V1.1	1	b
27C16	19	V1	2	a,b
27C16B	5C	V1.1	2	a,b
27C16H	BD	V1	2	a,b
27C210	5F	V5.1	1	
27C256	5D	V1.2	2	b
27C256B	5C	V1.1	2	b
27C32	19	V1	3	b
27C32B	5C	V1.1	3	b
27C32H	BD	V1	3	b
27C512	4C	V1.2	3	b
27C512A	5E	V1.1	3	b
27C58A	19	V1	2	a,b
27C58B	19	V1	2	a,b
27C64	5D	V1.2	1	b

Device Part Number	Family Code	Software Version	Device Category	Notes
27C64B	5C	V1.1	1	b
27CP128	5D	V1.2	1	b
34C02	120	V5.2		
93C06	118	V5.0	1	
93C46	118	V5.0	1	
93C56	174	V5.0	1	
93C66	174	V5.0	1	
93CS06	173	V5.0	1	
93CS46	173	V5.0	1	
93CS66	175	V5.0	1	

Oki Electric Industry Co., Ltd.

271000	5C	V3	1	
27128	79	V1	1	
27128A	5C	V1	1	
2716	19	V1	2	a
27256	5C	V1	2	
27512	5E	V1	3	
2764	79	V1	1	
2764A	5C	V1	1	
27C2000	5C	V4.2	1	
27C256H	93	V4.1	2	
28C16A	B7	V1.1	1	
28C64A	B7	V4.1	1	

Omni-Wave Semiconductor

27C101	8F	V5.1	1	f

Panasonic Semiconductor

27C256	5C	V4.1	2	
27C512	5E	V4.1	3	
27C64A	5C	V4.1	1	

Philips Semiconductor

27C010	5C	V5.0	1	
27C210	5F	V3	1	
27C240	5F	V4.3	2	
27C256	5C	V1	2	
27C512	5E	V3	3	
27C64A	5C	V1	1	
48F010	18B	V5.0	1	
87C51	5A	V1.3	1	

Ricoh Corporation

27C256	93	V1	2	
27C32	27	V1	3	
27C64	79	V1	1	
5H32	27	V1	3	

Rockwell Internations

87C32	27	V1.1	3	
87C64	79	V1.1	1	

Device Part Number	Family Code	Software Version	Device Category	Notes
SEEQ Technology, Inc.				
27128	79	V1	1	
27256	93	V1.1	2	
2764	79	V1	1	
27C256	93	V1.1	2	
2804A	B7	V4.2	1	
2816A	B7	V1.1	1	
2816AH	C4	V1.4	1	
2817A	B7	V1.4	1	
2817AH	C4	V1.4	1	
2864	B7	V1.4	1	
2864H	C4	V1.4	1	
28C010	B7	V1.4	1	
28C256	B7	V1.4	1	
28C64	B7	V1.4	1	
28C65	B7	V2	1	
36C16	9C	V1.4	4	
36C32	9C	V1	4	
47F010	147	V5.0	1	
47F512	147	V5.0	1	
48F010	10F	V5.0	1	
48F512	10F	V5.0	1	
5133	79	V1	1	
5143	79	V1	1	
52B13	A5	V1.4	1	
52B13H	B9	V1.4	1	
52B33	A5	V1.4	1	
52B33H	B9	V1.4	1	
5516A	B7	V1.4	1	
5516AH	C4	V1.4	1	
5517A	B7	V4.2	1	
5517AH	C4	V1.4	1	
SGS-Thomson Microelectronics				
24C02	120	V5.1	1	
24C02A	120	V5.1	1	
2532	19	V1	3	
25C02A	120	V5.0	1	
25C04	120	V5.0	1	
25C08	23E	V5.1	1	
27128A	93	V1	1	
2716	19	V1	2	a
27256	93	V1	2	
2732	19	V1	3	
2732A	27	V1	3	
27512	7F	V4.1	3	
2764	79	V1	1	
2764A	93	V1	1	
27C64A	93	V5.2	1	

Device Part Number	Family Code	Software Version	Device Category	Notes
27C160	2A1	V5.2		
27C1000	5C	V3	1	
27C1001	5C	V3	1	
27C1024	5F	V1	1	
27C16	19	V1	2	a
27C2001	5C	V4.2	1	
27C256	93	V1	2	
27C256B	5C	V4.1	2	
27C32	19	V1	3	
27C4001	5C	V5.0	2	
27C4002	5F	V4.2	2	
27C512	144	V4.1	3	
27C516	5F	V5.1	1	
27C64	93	V1	1	
28F410/8	401	V5.2	1	
28F410/16	403	V5.2	1	
28F420/8	400	V5.2	1	
28F420/16	402	V5.2	1	
28F512	135	V5.1	1	c
87C257	5C	V4.2	2	

SMOS Systems, Inc.

27128H	79	V1	1	
27C256H	93	V1	2	
27C64H	79	V1	1	

Samsung Semiconductor, Inc.

2816A	B7	V1.1	1	
2817A	B7	V1.4	1	
2864A	B7	V1.4	1	
2864AH	C4	V1.4	1	
2865A	B7	V1.4	1	
2865AH	C4	V1.4	1	
28C16	B7	V4.1	1	
28C17	B7	V4.1	1	
28C256	B7	V4.1	1	
28C64	B7	V4.1	1	
28C65	B7	V4.1	1	

Sharp Coporation

28F016SA	416	V5.2	1	
28F016SUT	416	V5.2	1	
571000	158	V4.2	1	
571001	158	V4.2	1	
57126	93	V1	1	
57127	93	V1	1	
57128	5C	V1	1	
57254	93	V1.1	2	
57255	93	V1.1	2	
57256	5C	V3	2	

Device Part Number	Family Code	Software Version	Device Category	Notes
5749	7C	V4.3	4	
57512	157	V4.2	3	
5762	93	V1	1	
5763	93	V1	1	
5764	5C	V1	1	
Sony Corporation				
27C1000	16A	V4.1	1	
27C1001	16A	V4.1	1	
27C256	16A	V4.1	2	
27C4001	5C	V5.1	2	
27C4002	5F	V5.1	2	
27C512	144	V4.1	3	
Texas Instruments				
2516	BD	V1	2	a
2532	BD	V1	3	
2532A	63	V1	3	
2564	BD	V1.4	1	
27128	79	V1	1	
27128A	93	V1	1	
27256	93	V1	2	
2732	BD	V1	3	
2732A-HS	63	V1	3	
27512	4B	V1	3	
2758	19	V1	2	a
2764	79	V1	1	
27C010	12B	V1.2	2	f
27C010A	115	V5.0	1	
27C020	115	V5.0	1	
27C040	115	V5.0	2	
27C128	115	V1.2	1	
27C210	73	V1.2	2	f
27C240	141	V5.0	2	
27C256	115	V1.2	2	
2732	116	V1.2	3	
27C510	115	V5.1	2	
27C512	116	V1.2	3	
27C64	115	V1.2	1	
27P32A	63	V1	3	
27P64	79	V1	1	
27PC128	115	V1.2	1	
27PC256	115	V1.2	2	
27PC32	116	V1.2	3	
27PC512	116	V1.2	3	
27PC64	115	V1.2	1	
28F010	26F	V5.1	1	
28F010DU	26F	V5.1	1	
87C257	115	V4.2	2	
320E14	220	V5.2		

Device Part Number	Family Code	Software Version	Device Category	Notes
Toshiba America				
24128	45	V1.4	1	
24128A	5C	V1	1	
24256	45	V1.4	2	
24256A	5C	V1	2	
24256B	5C	V4.1	2	
24512	5E	V1	3	
24512A	5E	V4.1	2	
2464	45	V1.4	1	
2464A	5C	V1	1	
27128	79	V1	1	
27128A	5C	V1	1	
27256	45	V1.4	2	
27256A	5C	V1	2	
27256B	5C	V1	2	
2732	19	V1	3	
27512	5E	V1	3	
27512A	5E	V1	3	
2764	79	V1	1	
2764A	5C	V1	1	
323	19	V1	3	
541000	5C	V4.1	1	
541001	5C	V4.1	1	
54256A	5C	V1	2	
544000	12E	V5.0	2	
544096	169	V5.1	2	
544200	169	V5.1	2	
54512A	5E	V4.1	3	
571000	5C	V1	1	
571000A	5C	V4.3	1	
571001	5C	V1	1	
571001A	5C	V4.3	1	
571024	5F	V1	1	
57256	45	V1.4	2	
57256A	5C	V1	2	
574000	12E	V4.1	2	
574096	169	V5.0	2	
5742000	169	V4.3	2	
57512A	5E	V1.1	3	
57H1000A	5C	V5.0	1	
57H1001A	5C	V5.0	1	
57H1024	5F	V3	1	
57H1024A	5F	V5.0	1	
57H1025A	5F	V5.0	1	
57H256	5C	V3	2	
58257A-LV	150	V5.0	2	
VLSI Technology, Inc.				
27C128	5D	V1.2	1	b
27C256	5D	V1.2	2	b
27C512	4c	V1.2	3	b
27C64	5D	V1.2	1	b

Device Part Number	Family Code	Software Version	Device Category	Notes
Waferscale Integration, Inc.				
27C010L	11B	V5.1	1	
27C010R	11B	V5.1	1	
27C040L	164	V5.1	1	
27C040LS	164	V5.1	1	
27C128F	3C	V2	1	
27C128L	11B	V5.1	1	
27C210L	15C	V5.1	1	
27C256F	124	V2	2	
27C256L	11B	V5.1	2	
27C512F	125	V2	3	
27C512L	11C	V5.1	3	
27C64F	3C	V2	1	
27C64L	11B	V5.1	1	
57C128F	3C	V2	1	
57C191	7B	V5.1	4	
57C191B	7B	V5.1	4	
57C191C	12D	V5.1	4	
57C256F	124	V2	2	
57C257	1F	V4.3	1	
57C291	7B	V5.1	4	
57C291B	7B	V5.1	4	
57C291C	12D	V5.1	4	
57C43	7B	V5.1	4	
57C43B	7B	V5.1	4	
57C43BT	7B	V5.1	4	
57C43C	12D	V5.1	4	
57C45	122	V5.0	1	
57C45T	122	V5.0	1	
57C49	7B	V5.1	4	
57C49B	7B	V5.1	4	
57C49BT	7B	V5.1	4	
57C49C	12D	V5.1	4	
57C49CT	12D	V5.1	4	
57C51	F7B	V5.1	4	
57C51B	F7B	V5.1	4	
57C51BT	F7B	V5.1	4	
57C51C	22D	V5.1	4	
57C64F	3C	V2	1	
57C65	2C	V2	1	
57C71C	12D	V5.1	4	
57C128FB	12D	V5.2	1	
PSD4X1	410	V5.2	1	
PSD5X1	410	V5.2	1	
White				
WE128K8	BA	V5.2		
WE256K8	BA	V5.2		
WE512K8	BA	V5.2		

Device Part Number	Family Code	Software Version	Device Category	Notes
Xicor, Inc.				
2402	120	V5.1	1	
2404	120	V5.1	1	
24C04	120	V5.1	1	
24LC04	120	V5.1	1	
2804A	B7	V1.4	1	
2816A	B7	V1	1	
2816B	B7	V1.4	1	
28256	B7	V1.4	1	
2864A	B7	V1.4	1	
2864B	B7	V1.4	1	
2864H	B7	V4.1	1	
28C010	B7	V3	1	
28C256	B7	V1.1	1	
28C256B	BA	V5.1	1	
28C512	B7	V5.0	1	
28C64	B7	V3	1	

BoardSite Application Note:
PROGRAMMING INTEL DD28F032SA DEVICES

The Intel DD28F032SA device is effectively two 28F016SA devices in one IC. This device can operate in 8- or 16-bit modes. This application note is intended to help configure BoardSite to program this device in any mode. All configurations would have the profile setup with an Address bus width of 24.

OPTION 1: Configure BoardSite for two devices. CE1 and CE2 control the two halves of this device. In this case the Algorithm for 28F016SA is selected.

Number of programmable Chip Enables: 8
Address Increment: 1 (for 8 bit), 2 (for 16 bit)
Data Bus Width: 8 or 16

Device Group #1
Number of Address Ranges this Device Group: 1
Address Range 1: 00 -->1FFFFF
Programmable Chip Enables Used by this Device Group: Y
Number of devices in Device Group: 1
Programmable Chip Enable Mask: 01 (for Address line 24 as CE1)

Device Group #2
Number of Address Ranges this Device Group: 1
Address Range 1: 20000 -->3FFFFF
Programmable Chip Enables Used by this Device Group: Y
Number of devices in Device Group: 1
Programmable Chip Enable Mask: 02 (for Address line 25 as CE1)

OPTION 2: Configure BoardSite for one 28F032 device in 8 bit mode. Address line 21 could be used as CE1 and inverted to be CE2.

Number of programmable Chip Enables: 0
Address Increment: 1
Data Bus Width: 8
Device Group #1
Number of Address Ranges this Device Group: 1
Address Range 1: 00 -->3FFFFF
Programmable Chip Enables Used by this Device Group: N
Number of devices in Device Group: 1

OPTION 3: Configure BoardSite for one 28F032 device in 16 bit mode. Address line 20 could be used as CE1 and inverted to be CE2.

Number of programmable Chip Enables: 0
Address Increment: 2
Data Bus Width: 16

Device Group #1
Number of Address Ranges this Device Group:
Address Range 1: 00 -->1FFFFF
Programmable Chip Enables Used by this Device Group: N
Number of devices in Device Group: 1
Programmable Chip Enable Mask: 01 (for Address line 24 as CE1)

BoardSite Application Note:
PROGRAMMING 8748H AND 8749H MICROCONTROLLERS

The 8748H and 8749H microcontrollers require extended circuitry on the BoardSite Programmer. These devices require 3 high-voltage supplies (Vdd, Vea, and PROG) to program. Since BoardSite has only two high-voltage supplies (Vpp1 and Vpp2), external hardware is needed to create a third supply.

Examination of the Manufacturer's Device Specification shows a simple method of accomplishing this. The Program/Verify Waveforms in the specification show that VDD and PROG have similar characteristics. In fact, they can be identical except for their DC levels. The VDD supply switches to 21V, where the PROG signal switches to 18V.

A resistor divider circuit can be used to create the PROG signal from the Vpp1 supply (Vdd). This divider must meet the PROG signal voltage requirements as well as allowing for the maximum IPROG supply current.

We recommend the following circuit connections and resistor values for a single microprocessor on the circuit board. If circuitry for more than one microprocessor is needed, the user must modify these resistor values to compensate for the total IPROG supply current.

```
      8748H
      8749H
    +---------+
    |   Vea   |----------------------- Vpp2
    |         |
    |   Vdd   |----------------------- Vpp1
    |         |        |
    |         |       +-+  348 ohm 1%
    |         |       | |  1/4 W
    |         |       +-+
    |   Prog  |--------|
    |         |       +-+  2.21K ohm 1%
    |         |       | |  1/2 W
    +---------+       +-+
                       |
                      gnd
```

Simple circuit solutions for the resulting PROG supply, shows the following high and low DC values.

	Min	Nom	Max	Notes
VPH	17.6	18.0	18.4	VDD = 21.2V, IPROG = 0
				VDD = 20.8V, IPROG = 1mA
VPL	4.1	4.2	4.5	VDD = 21.2V
				VDD = 20.8V

BoardSite Application Note:
REMOVING NULL FILE NAMES FROM YOUR CATALOG FILE

If you list out your catalog file using the FILE LIST command and it displays file names as quote marks enclosing 32 spaces, this file probably is a "null" file name. These file names make it difficult to delete and rename files.

Additional code was added to the catalog file manager that will enable you to correct your catalog file and rename or delete these null file names.

The procedure is as follows:

First delete an existing file in your catalog, preferably a data file or user text file. If you do not have any files that you can delete, copy a file to a new name and then delete the original. When you delete a file from the catalog, the catalog manager will give the null file names valid names starting with "--- BAD FILE NAME 1 ---". Each subsequent name is given next number until all null files names are replaced with these specific names.

Now list out your catalog again and there should be no more null names. You may now delete or rename the file names.

BoardSite Application Note
MULTIPLE ALGORITHMS IN A SINGLE BOARD PROFILE

This application note is provided to clarify some issues that could arise if you use the multiple algorithm feature. Please refer to the User Manual for information on how to add multiple algorithms to a Board Profile.

A memory board that contains multiple devices (such as EEPROM, EPROM, and microcontroller) will require different algorithms to program each devices type. By using the multiple algorithm feature, you can program all the memory devices on your board in one operation. Another benefit is that all of the information (parameters and sequence files) is contained within one Board Profile. Read sections A through C to ensure proper setup for multiple algorithms.

A. Board Profile Editor: Board Data Mode Parameter

1. If you have multiple device types on your board, you must specify the same data mode for all algorithms within that that Board Profile.

2. If you use one data file for all devices on a Multiple Device Board, you must specify the Board Data File Offsets

B. Board Profile Editor: CRC and Checksum data mode options

The "Entire Board" option will produce a CRC and/or Checksum calculation over each algorithm (device type).

C. Designing your board and/or adapter for multiple algorithms

When designing your memory board or adapter for multiple algorithms you should be aware of the following items.

1. If the programming voltages differ from device type to device type, you must isolate the Vcc and/or Vpp voltages. You can do this using the auxiliary Vcc and Vpp supplies (VCC2, VPP2) or design an analog switch on your adapter that is controlled by an available control or programmable chip enable line.

2. If any of the device types on your Board Profile have a multiplexed program and chip enable pin (PGM/CE such as 27C256), you must determine how to disable that device when other devices are being programmed. Figure 1 shows an example a multiple algorithm board.

   a. If your supply voltages (Vcc and Vpp) are isolated from the other devices, you can guarantee that the devices will not be programmed inadvertently (while the other device types are being programmed) by setting the Vcc and Vpp voltages to their read level.

   b. If your supply voltages (Vcc and Vpp) are not isolated from other devices, you need to guarantee that the devices will not be programmed inadvertently, by disabling the PGM/CE signal on the device. You will need to provide separate PGM signals to the multiplexed and non-multiplexed PGM/CE devices.

   One solution is to gate the programmable chip enable lines (PCE's) with the PGM line. By specifying the PCE masks in the Board Profile Editor you can disable the pgm signal from the multiplexed PGM/CE device, while the other device types are programming. See Figure 2.

```
      ---------------------------------------------------------
     | Memory Board                                            |
     |                                         27010           |
     |                                       ---------         |
     |            27C256                    |         |        |
   - |VCC1       ---------       ---------- |VCC      |        |
  |  |---------|VCC      |     |            |         |        |
  |  |VPP1     |         |     |   -------- |VPP      |        |
  |  |---------|VPP      |     |  |         |         |        |
  |  |         |         |     |  |   ----- |OE       |        |
  |  |C3 (OE2) |         |     |  |  |      |         |        |
  |  |---------|OE       |     |  |  |  --- |PGM      |        |
  |  |PGM_B    |         |     |  |  | |    |         |        |
  |  |---------|PGM/CE   |     |  |  | |  - |CE       |        |
  |  |         |         |     |  |  | | |  |         |        |
  |  |          ---------      |  |  | | |   ---------         |
  |  |VCC2                     |  |  | | |                     |
  |  |-------------------------   |  | | |                     |
  |  |VPP2                        |  | | |                     |
  |  |----------------------------   | | |                     |
  |  |C1 (OE1)                       | | |                     |
  |  |-------------------------------  | |                     |
  |  |PGM                              | |                     |
  |  |---------------------------------  |                     |
  |  |C0 (CE)                            |                     |
  |  |-----------------------------------                      |
   - |                                                         |
     |                                                         |
      ---------------------------------------------------------
```

Connector

Figure 1.

3. If any of the device types in your Board Profile have a multiplexed Vpp and output enable line (Vpp/OE such as 27512), you may need to isolate or separate the Vpp supply from the non-multiplexed Vpp/OE devices.

   Use the auxiliary Vpp supply (VPP2) for the multiplexed device. You must modify the sequence file to use VPP2 instead of VPP1.

```
                                  ---------
PGM (from BoardSite)             |         |
---------------------------------|         |          PGM_B
PCEx                             |   AND   |--------------------
---------------------------------|         |
(set to 0 if programming         |         |
PGM/CE device, set to             ---------
1 otherwise)
```

Figure 2.

BoardSite Application Note:
BOARD PROFILES SHARING SEQUENCE FILES

This application note is provided to clarify some issues that could arise if you share sequence files between Board Profiles.

An example of a shared sequence file application would be a memory board that has multiple data configurations. One configuration may use a 64K data file and another may use a 32K data file. You will need two Profiles to cover the two data configurations due to the different address ranges. Since the memory devices on the boards are the same for either configuration, you can create two Board Profiles that will share the same sequence file.

To create a sequence-sharing Board Profile, you must first create a "golden" Board Profile - a Board Profile that is known to work on the targeted memory board. Use the FILE COPY command to copy the golden Board Profile to a new Board Profile. You will be prompted to specify whether the new Board Profile is to share the sequence file with the golden Board Profile.

Board Profiles that share the same sequence file should not change the following Board Profile Editor parameters:

Address Bus Width
Number of Programmable Chip Enables
Isolate Boards
PGM Line Active High
Chip Enables Active High
Device Type
Data Bus Width
Data MSB

If any of the above parameters must be changed, the sequence file must be recompiled and any Board Profiles that share the sequence file should also be changed so that they match the golden Board Profile.

Tape Backup/Restore Procedures

Note: It is our recommendation that you use a Colorado Trakker 700 Tape Backup System that connects to the parallel port of the Boardsite 5100 processor card. This system has been tested with the Pentium based Boardsite Programmer and works very well.

The following text describes the procedures to copy files from a 286/386 Boardsite system cartridge to the newer Pentium based removeable media cartridge:

I. Boot the Boardsite Programmer with the cartridge that came with the unit. The Pentium cartridge will only boot a Boardsite programmer with a Pentium single board computer installed. It will not boot the older Boardsites
with 286/386 single board computers installed in them.

II. On the "C" or "D" drive partition, install your tape backup software. This provides you with a permanent copy of the tape backup software for future use.

III. Go to the Ram Drive (Drive "E") and make a directory for your tape backup software with the same name as the directory for the tape software you installed on Drive "C" or "D". Copy all of the tape backup files from the "C" or "D" drive to the tape backup directory that you created on Ram Drive ("E"). This step essentially copies the tape backup software from your "C"
or "D" drive partition to the Ram Drive ("E").

Note: The tape backup software <u>must</u> be executed from the Ram Drive ("E"). When you remove the original cartridge from the Syquest Drive (with the tape backup software installed on it) and insert the old 286/386 cartridge that you want to backup, the tape backup software is no longer present. When the tape backup software is loaded onto the Ram Drive ("E"), you can remove and insert cartridges as necessary to perform the tape backup and restore functions.

IV. Once the tape backup/restore software is installed on the Ram Drive ("E") then go ahead and run the program.
V. Insert the cartridge that you want to back the files off of and transfer them to tape. Select the appropriate files and/or directories that you want to backup and perform the steps necessary to execute the tape backup procedures.

VI. Insert the cartridge that you want to transfer the backed up files to (the newer cartridge that came with your Pentium based Boardsite System) and perform the necessary steps to execute a restore operation from your tape backup software.

VII. Exit the tape backup software to conclude the tape backup procedures.

Note: If you encounter the following error message, remove the cartridge in your Syquest removeable media drive and insert the original cartridge that
you used to boot your Pentium based Boardsite System:

Invalid COMMAND.COM
Insert disk with DOS/Command.Com
in Drive C.
Press any key to continue

General Considerations:

** The key point to remember regarding the tape backup/restore process, is that you must execute the tape backup software (DOS Version) from the Ram Drive ("E").

** No attempt was made to explain the finer details of the tape backup/restore procedures because there are many different software packages one could use and they are all different as to how they are setup and executed.
Refer to the manuals that came with your tape backup software for specific details on the tape backup/restore operations.

** The procedures listed in this document may also vary somewhat, depending on which tape backup/restore software package you are using.

** If you have several smaller files that you want to back off of your old 286/386 cartridge (under 1.44 Megs), you can copy them to a 1.44 Meg Disk installed in Drive "A" on the Boardsite System. Then you can restore them from the "A" Drive back onto your new cartridge. If the files are small enough and there are not that many of them, then you do not have to be concerned about using the tape backup/restore process.

Pentium Processor Jumper & Cable Schedule:

Modified 6/19/96

The following jumper and cable schedule must be adhered to for the proper operation of the Pentium Processor Board when installed in the Boardsite 5100 System:

Jumpers:

JP1 - Installed
JP2 - Not Installed
JP3 - Not Installed
JP4 - Jumpers NOT installed on pins 1&2 and 3&4
JP5 - Top two positions jumpered (1&2)
JP6 - Top two positions jumpered (1&2)

Cables:

1) The Pentium Processor fan plugs into the J11 connector on the processor board.

2) The keyboard/speaker cable assembly plugs into J3 on the processor board.

3) The floppy drive plugs into the J6 connector on the processor board.

4) The serial port cable (COM1) plugs into the J9 connector on the processor card.

5) The Syquest Drive cable plugs into the J4 connector on the processor card. This requires the special SCSI Cable Adapter that is shipped to Data I/O with each processor board.

1.0 Boardsite System Boards

A basic Boardsite system consists of a Controller Board (with Expansion daughter board), one to four Interface Board(s), a Pre-regulator Board and a Motherboard.

1.1 Controller Board

This board provides the interface, through its Expansion daughter board, between the PC system controller and the Boardsite system. It re-drives the I/O channel signals, from the IBM Expansion Bus cable, for use by the programmer. Directional control logic is contained on the board to resolve any bus contention and directs data flows between the PC and the programmer.

This board communicates with the Interface Board(s) and the Pre-regulator Board, and provides all necessary control signals to run the Boardsite system.

This board also allows for an optional Programming Controller for speed enhancement during device related operations.

1.2 Interface Board

The Interface Board provides all required digital signals and programming voltages to the Boardsite user interface connector(s). This board is designed to work with the Controller Board to support all device related hardware functions in Boardsite.

This board also allows for future addition of high-speed hardware verify, access time testing and threshold testing options.

1.3 Pre-regulator Board

This board provide the required programming power to the Interface Board(s). The power switching capability of the Pre-regulator Board allows regulation of programming power, during all programming operations, and reduces excess heat dissipation of the analog circuitry.

1.4 Motherboard

This board is a passive system backplane to hold the Controller Board, the Interface Board(s) and the Pre-regulator Board for all Boardsite configurations.

This board also supports an internal PC option for the Benchtop configuration. There are power connectors on this board to connect to the External Power Supply option to increase the programming power of the system.

1.5 Diagnostic Test Adapter Board

This board is designed and packaged, in a Diagnostic Test Adapter, which adapts to the Boardsite user interface connector(s). It is a diagnostics tool for the user to check or diagnose the integrity of the Boardsite system.

1.6 EPROM Evaluation Board

This board is designed and used to program all memory devices in the Boardsite device support list. This board aids in checking out programming algorithms and interface signals provided by the programmer.

1.7 Prototyping Board

This board is designed and used to enable the user to easily interface existing circuit boards, or add any necessary circuity to route electrical signals provided from the interface connector(s), for in-circuit feasibility or to actually use in lieu of building a PCB interface.

2.0 Boardsite Circuit Board Functions

2.1 Controller Board Function Blocks

The following describes major function blocks of the Controller Board:

2.1.1 Expansion daughter board to interface Boardsite to the PC Controller. It provides on-board diagnostic capability on the Expansion Bus address and data lines. The interface is designed to operate with a variety of IBM compatible PC's and laptops. The following machines have been tested to operate with Boardsite:

IBM PC, XT, AT 6 or 8MHz
IBM AT compatible 8MHz
AST 10MHz/286 0 wait state
Toshiba Laptops T1000
T1100
T3100

There is, also, circuitry providing signals to the bus arbitrator to detect direction of the data bus.

The Power-on Reset circuitry allows Boardsite to be turned ON or OFF without affecting the PC's operation.

2.1.2 PC Bus Buffers to buffer or drive certain address and control signals, which enable the Controller Board to operate inside a PC or through the Expansion Bus.

2.1.3 Data Bus Arbitrator to direct data flows between external/ internal PC configurations during byte-by-byte or DMA operations. It looks at the Memory and I/O control signals, along with AEN (DMA active signal) and Data Detects to decide on data bus directions.

2.1.4 Sequencing Port to minimize direct I/O addressing space. Most of Boardsite's I/O operations are performed through the Sequence Port by way of a simple Instruction Set. Data transfers are made by first writing an instruction opcode to the Sequence Port, followed by an I/O address, and then followed by up to 4 bytes of data.

Data can also be sent to the Sequence Port in DMA. The Sequence Port also isolates necessary circuitry from the PC bus, that the future Programming Controller can access the control ports directly.

2.1.5 Wait State Generator to enable Boardsite to operate with PC Controllers running at different I/O bus speeds, i.e. 4.77, 6, 8, 10, 16 MHz.

2.1.6 Programmable Clock to provide a user programmable clock signal and a programmable timer reference for program pulse. The selectable clock rates are 1, 2, 4, 8 MHz $\pm$ 5%.

2.1.7 PGM Pulse Generator to provide program pulse polarity, manual override, VPP timer/manual pulse controls. The Heart of the PGM Pulse Generator circuitry is the Intel 82C54, a Programmable Interval Timer complete with 3 independent 16-bit counter/timers. In this design the following timer assignments are made:

Timer 0 - PGM Pulse timer.
Timer 1 - Overprogram scalar.
Timer 2 - Delay Timer.

Timer 0's output can be used as the PGM signal sent to each Interface Board, or used with the Vpphi reference signal to generate a timed VPP pulse. Timer 1 can be programmed as a Pre-scalar to Timer

0. These cascaded timers can deliver program pulse width from 1 us to 4280 sec (1MHz clock rate).

2.1.8 Programmable Analog References to generate fourteen precision programming supply voltage and current references.

2.1.9 Analog MUX/Readbacks to verify all system voltages, programmable references and Interface Board analog outputs. The system supports a total of 32 MUXed readback channels.

2.1.10 Test Circuitry to measure accuracy of programmable clocks, program pulse widths, programmable voltage rise/fall times and digital signal output levels.

2.1.11 Hardware Interrupt channel, DMA channel and I/O addresses are jumper/switch selectable to fit in the user system. Both the Sequence Port and the Optional Programming Controller can use DMA for data write transfer.

2.1.12 Optional Programming Controller to enhance multi-board programming speeds. A 50-pin daughter board connector has been provided to adapt a future Programming Controller Daughter Board to the Controller Board. This Controller will take control of the Sequence Bus directly in generating address, data, and control signals.

2.2 Interface Board Function Blocks

The following describes major function blocks of the Interface Board:

2.2.1 Decoding Logic to support both private and common addressing, i.e. single board read/write access and multi-board broadcast mode, respectively.

2.2.2 Digital Hybrids and protection circuits to provide bi-directional and tri-statable digital signals. There are fifteen digital hybrids that are grouped in subsets to form special purpose digital drivers/receivers, i.e. 32-bit address, 32-bit data, 8-bit board detect, etc. All lines have over/under voltage clamps or detects, and have in-line resistive ESD protection.

2.2.3 Programmable Analog Drivers with over/under voltage and over-current protections; they are VCC1, VCC2, VPP1, VPP2 and VNEG. VCC1 and VCC2 can each deliver 0 - 7V at 6A but not simultaneously. VPP1 and VPP2 can each deliver 0 - 25V at 2A but not simultaneously. VNEG can deliver 0 - 8V at 250mA.

2.2.4 VPP Hi/Lo Switch to enable fast switching of VPP output voltage. It is designed for use in programming devices with MUXed OE/VPP pin that switches between programming and verify cycles, i.e. INTEL 27512.

2.2.5 VPP Slew Rate Select to enable either fast or slow switching of the VPP output voltage. The two selectable rates are 20us and 50us respectively. The slew rates for VCC and VNEG supplies are fixed at 50us.

2.2.6 Power Sequencer to enable safe powerdown of all programming supplies during a fault condition, i.e. overcurrent or output voltage errors. The Sequencer controls all programming supplies, ±12V switches and the digital interface tri-state control. The Sequencer is a state machine that cycles through fifteen memory addresses containing data pointers to the controlled power circuits. It can also be used to power up/down supplies individually or simultaneously. The Sequencer rate select range is from 250ns to 2ms delay between supplies.

2.2.7 Analog MUXes to route back both analog and digital signals, to the Controller Board, for output level and rise/fall time testings.

2.2.8 Test Circuitry to enable on-board testing of over/under voltage protection circuits of all programmable supplies. The over/under voltage trip ranges for all programmable supplies are set at ±500mV.

2.2.9 Board On/Off Control to isolate faulty boards from the system bus, such that the device operation can continue on other good boards. All digital and analog outputs at the User Interface Connector are at zero volt or tri-stated.

2.3 Pre-regulator Board Function Blocks

The following describes major function blocks of the Pre-regulator Board:

2.3.1 Negative Supply Voltage Sense to ensure proper power sequencing of critical analog components, i.e. OP-AMPs.

2.3.2 Regulated supply voltages for Interface Board programming supplies, i.e. VCC and VPP supply voltages.

2.3.3 Summing Circuit to allow for optional overhead voltages; that is higher output voltages than 7V for VCC and 25V for VPP.

2.4 Diagnostic Test Adapter Board Function Blocks

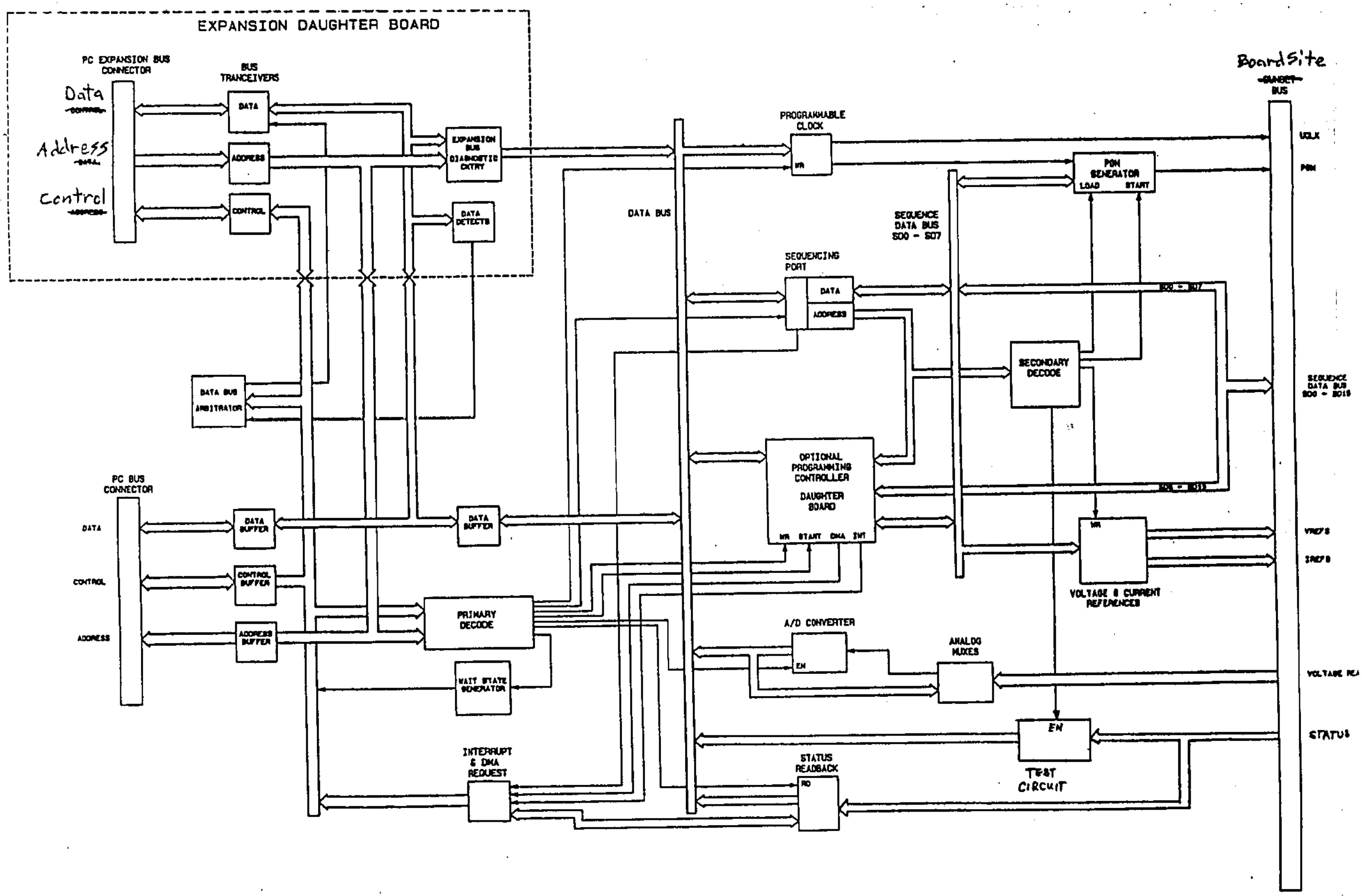
The following describes major function blocks of the Diagnostic Test Adapter Board:

2.4.1 Resistive test loads for all analog output voltages.

2.4.2 24VDC fan for cooling of the load resistors during load testing, also being part of the resistive loads.

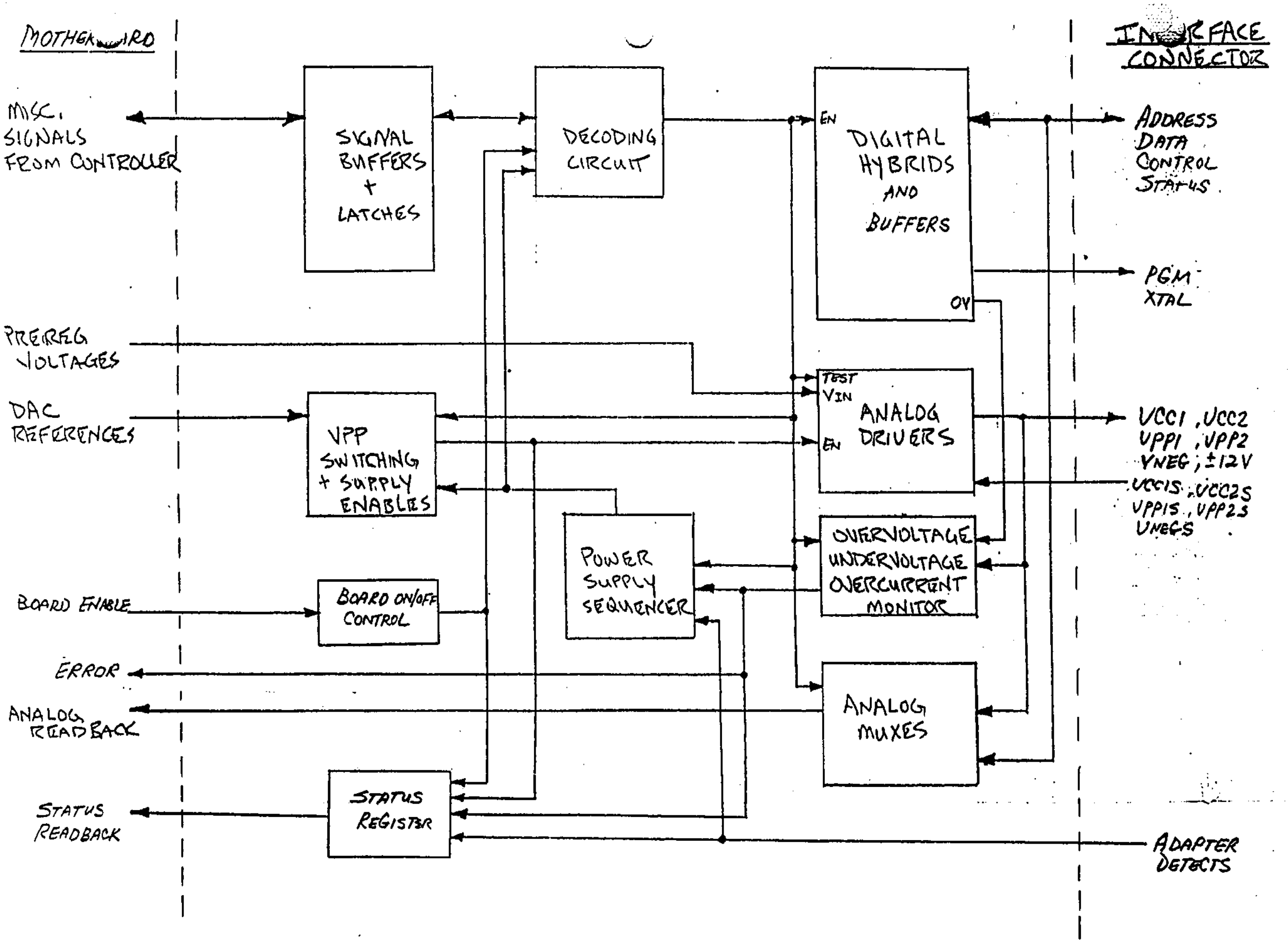
2.4.3 Signal loopbacks of all digital outputs. This enables logic level measurements and rise/fall time testings.

2.4.4 Test Points for all analog and digital signals. This facilitates verification of programming algorithms or signals.

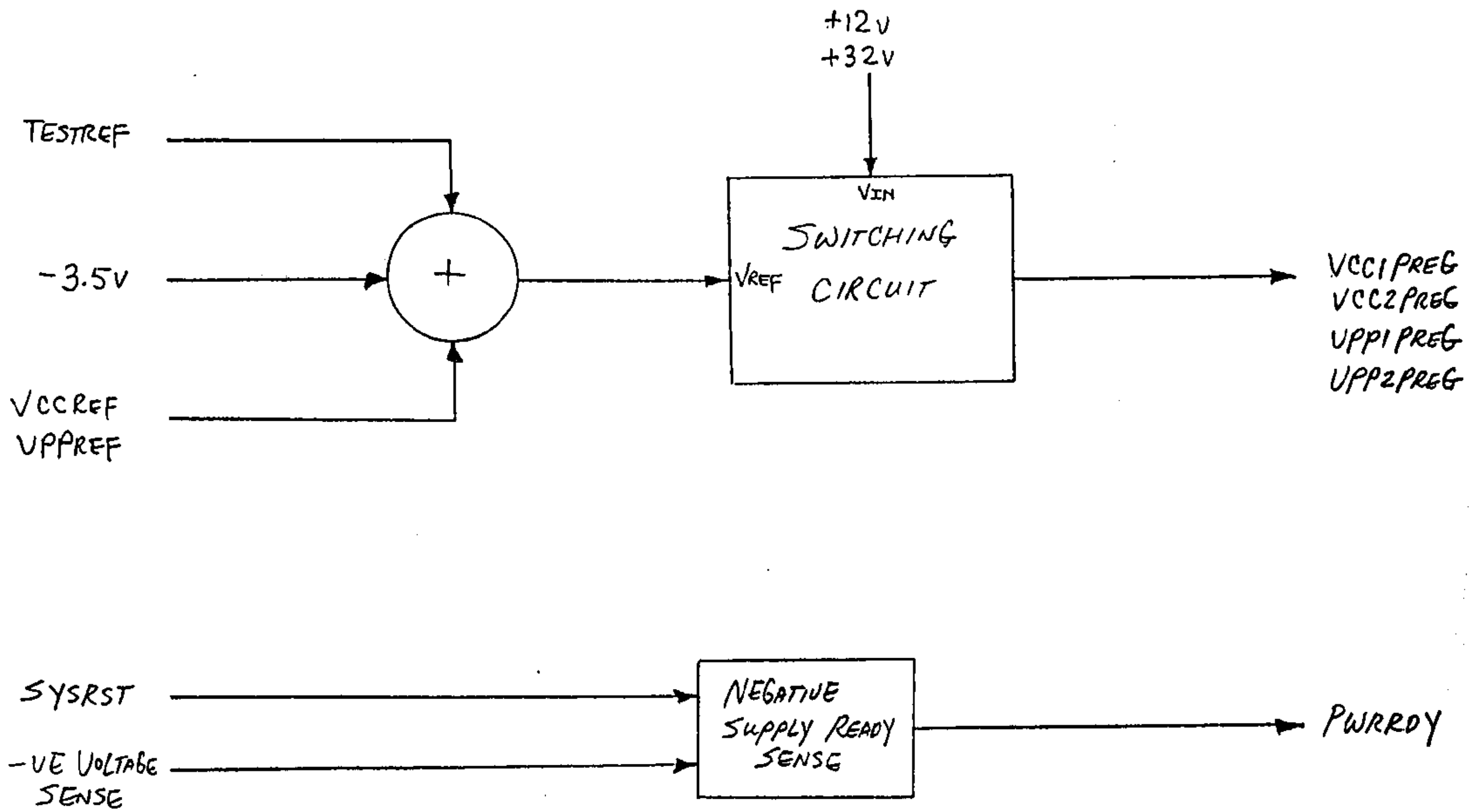


CONTROLLER BOARD BLOCK DIAGRAM

1/89 RC

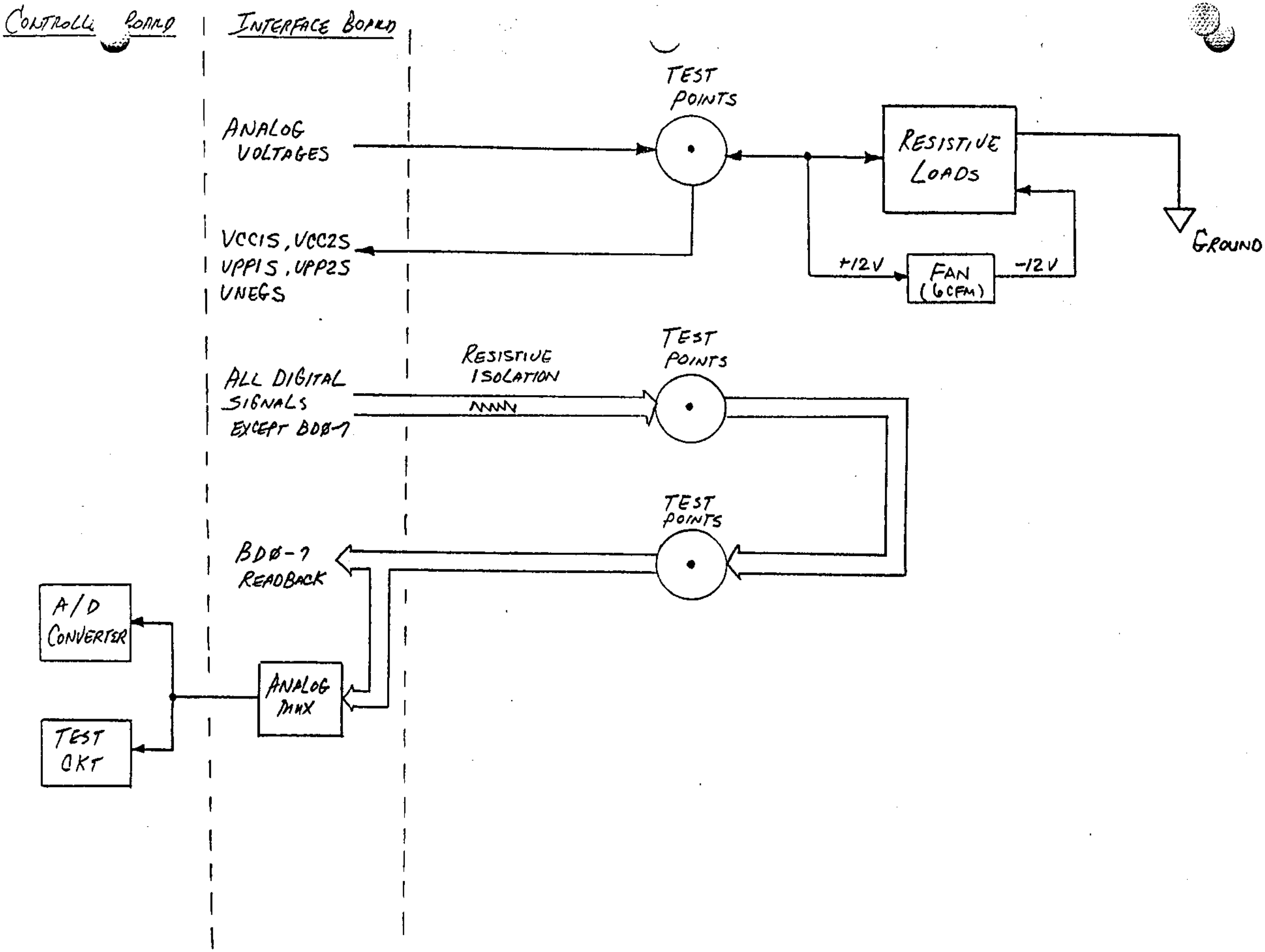


INTERFAC. BOARD BLOCK DIAGRAM



PRE-REGULATOR BOARD BLOCK DIAGRAM

1/00 dr



DIAGNOSTIC TEST ADAPTER BOARD BLOCK DIAGRAM

1/89 RC

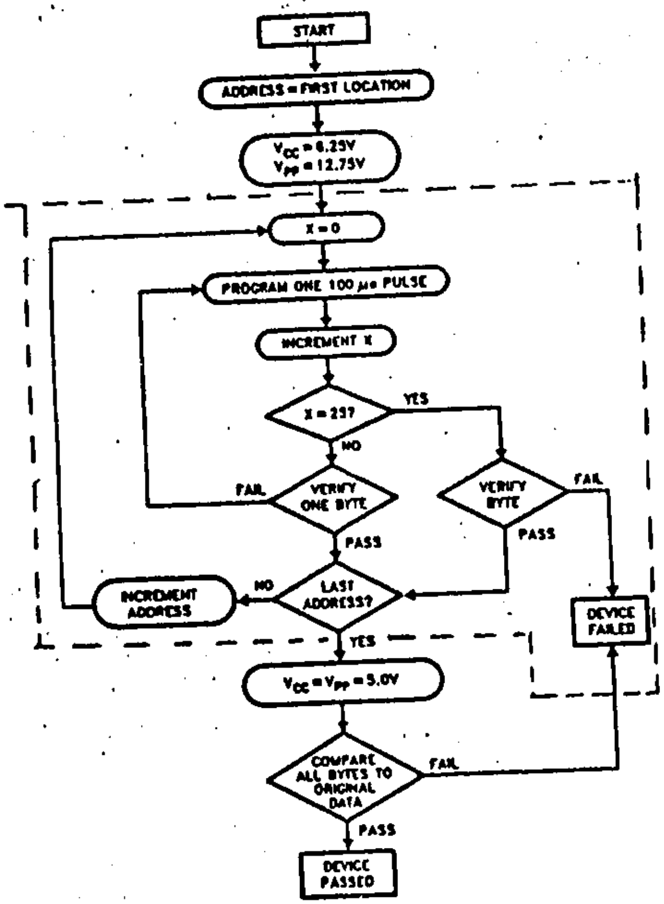
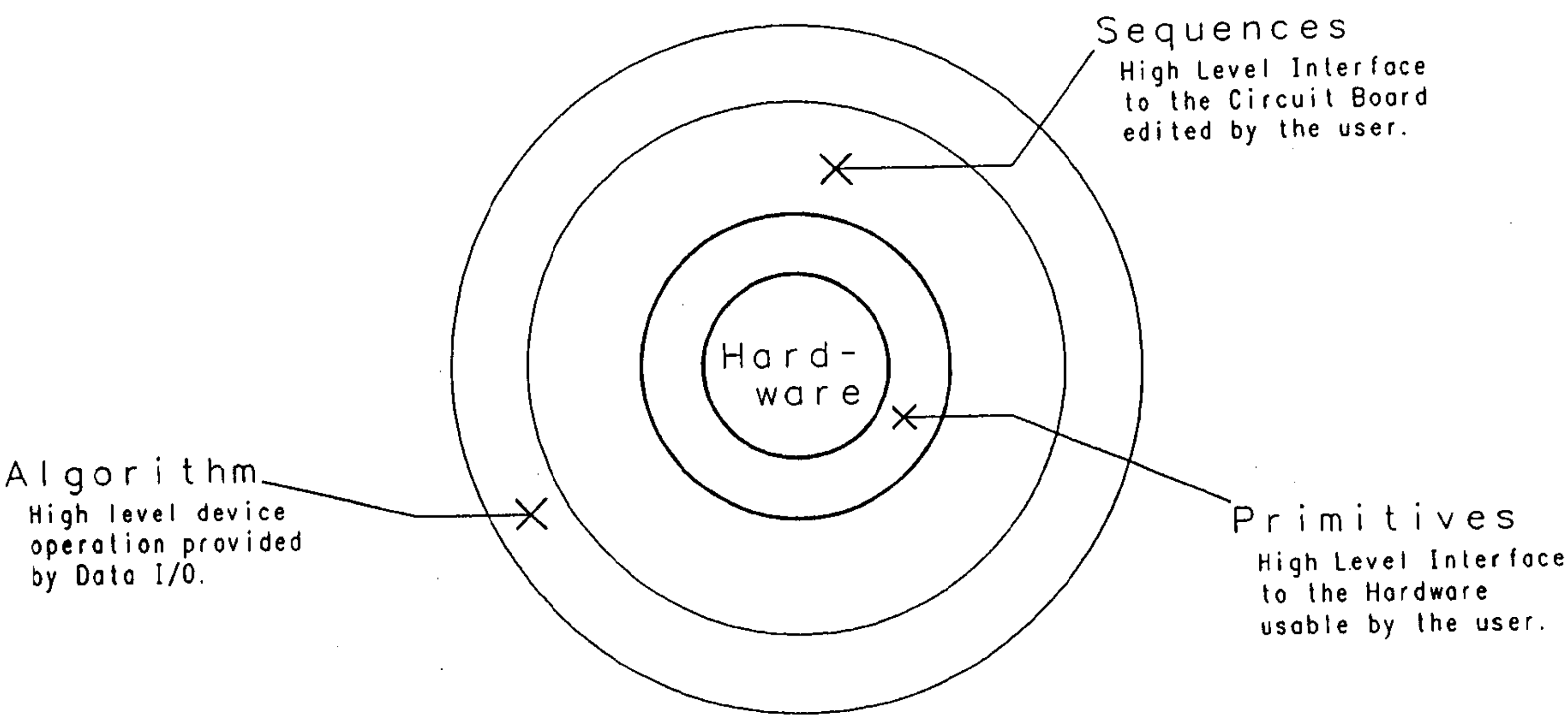


Figure 5. Quick-Pulse Programming™ Algorithm

```
/* programming algorithm for Intel Quick Pulse */
/* in C language for Boardsite Algorithm Example */

program ()        /* algorithm */
{
    ADDRESS = FIRST_ADDRESS; /* initialize address */

    /* loop on addresses from first to last */
    do
    {
        output_address ();        /* sequence */
        output_data ();           /* sequence */
        x = 0;              /* initialize reject count */

        /* now loop on reject count */
        /* if all boards pass, exit loop */
        /* if reject gets to 25, exit loop */
        do
        {
            program_pulse ();/* sequence */
            verify_boards ();/* sequence */
            if ( STATUS == ALL_PASS ) break;
            x = x + 1;        /* increment reject count */
        }
        while ( x < 25 );

        /* if no boards passed, exit loop */
        if ( STATUS == ALL_FAIL ) break;

        increment_address ();
    }
    while ( ADDRESS <= LAST_ADDRESS );
}
```

BOARDSite ALGORITHM
EXAMPLE

BoardSite Sequence Editor



Board Profile

Board Information
(Static Parameters)

Number of Address Lines
Number of Data Lines
Names of PROMs
Types of PROMs
Address of PROMs
Power Supply Levels
Current Limits
(etc.)

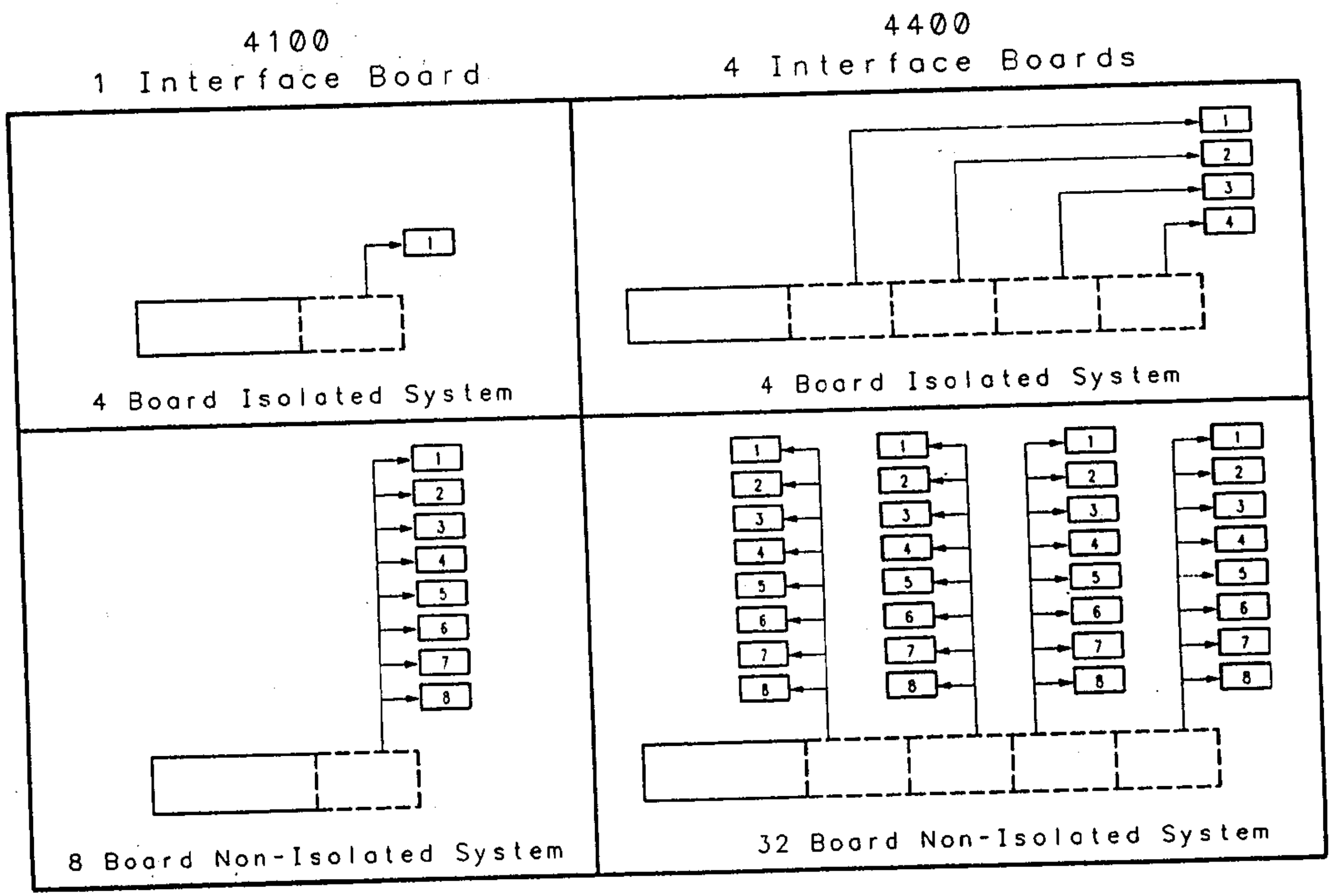
Sequence File
(Dynamic Flow)

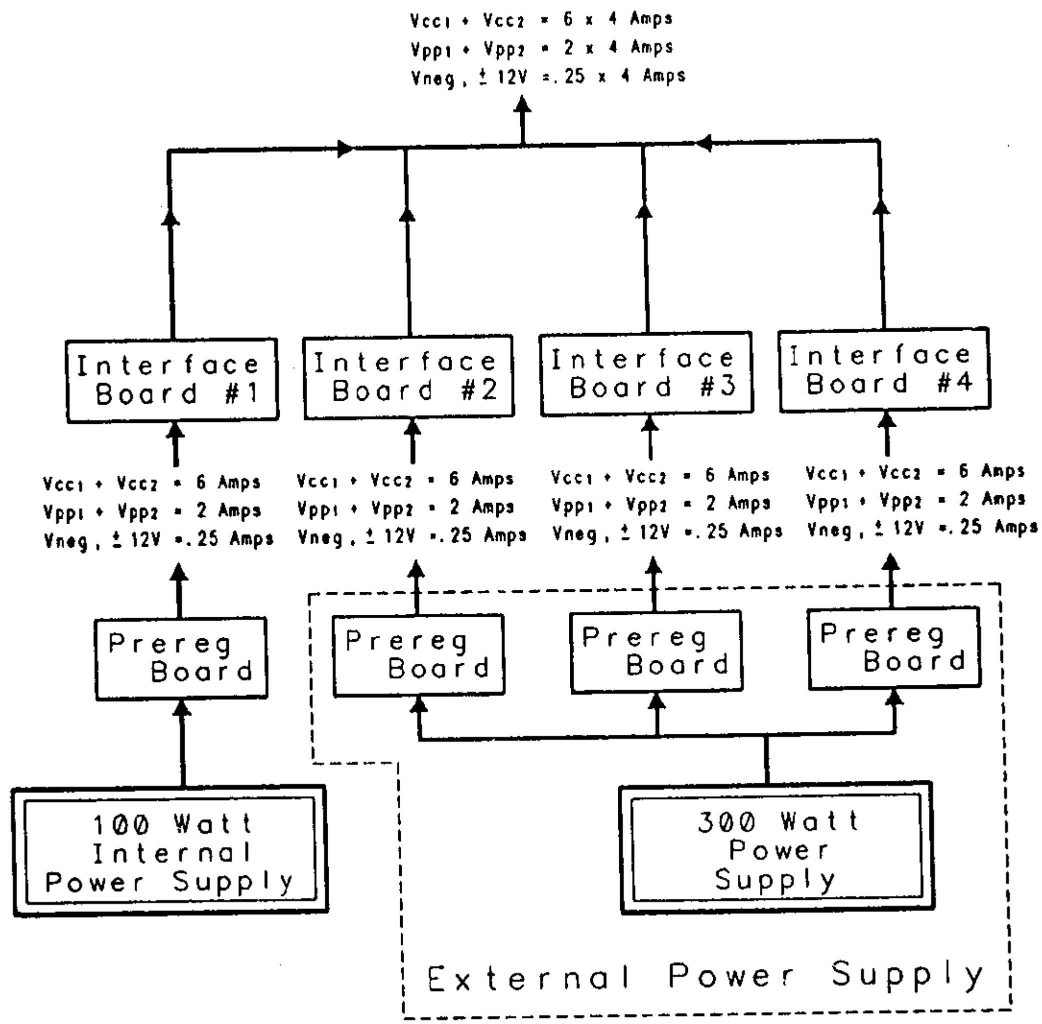
Timing Sequences for all Signals at the Interface Connector
Address
Data
Control
Power Signals
(etc.)

Non Isolated will be slower because after Flow is compiled the Bd. detect lines are Monitored and looped through

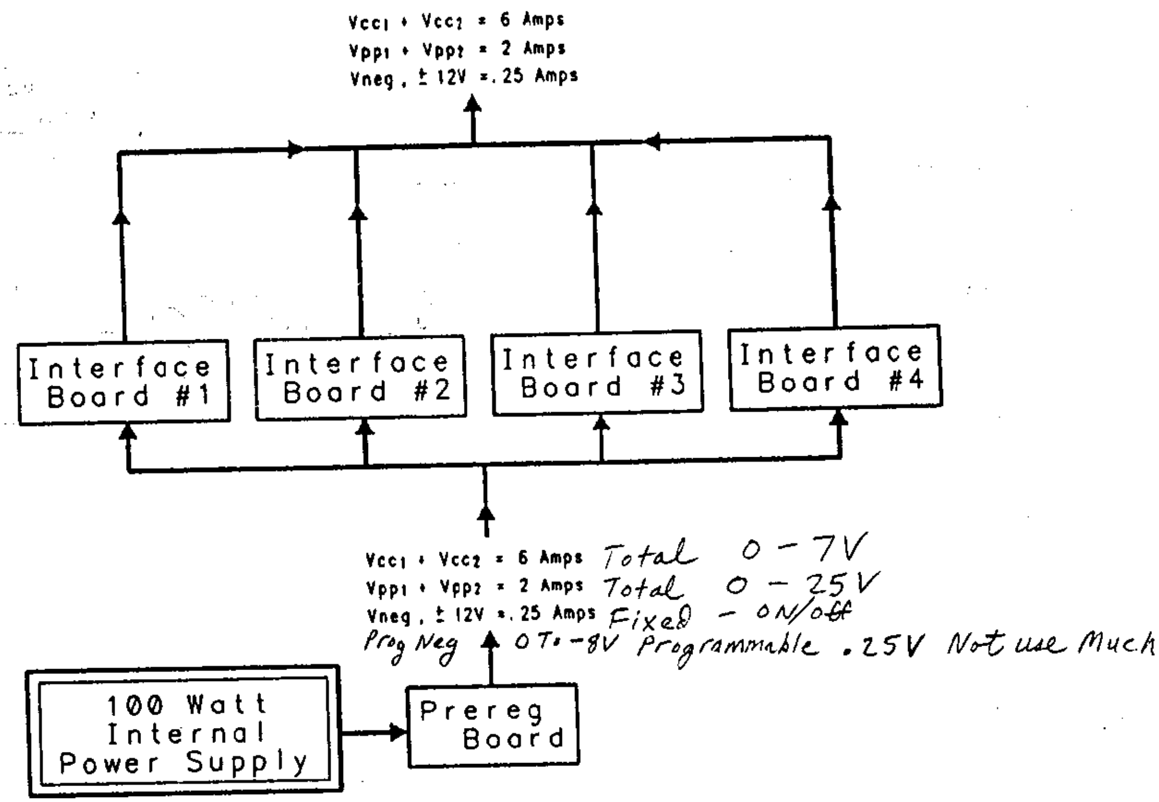
Isolated - Referes to Fault detection to 1 Bd level

BoardSite Configurations

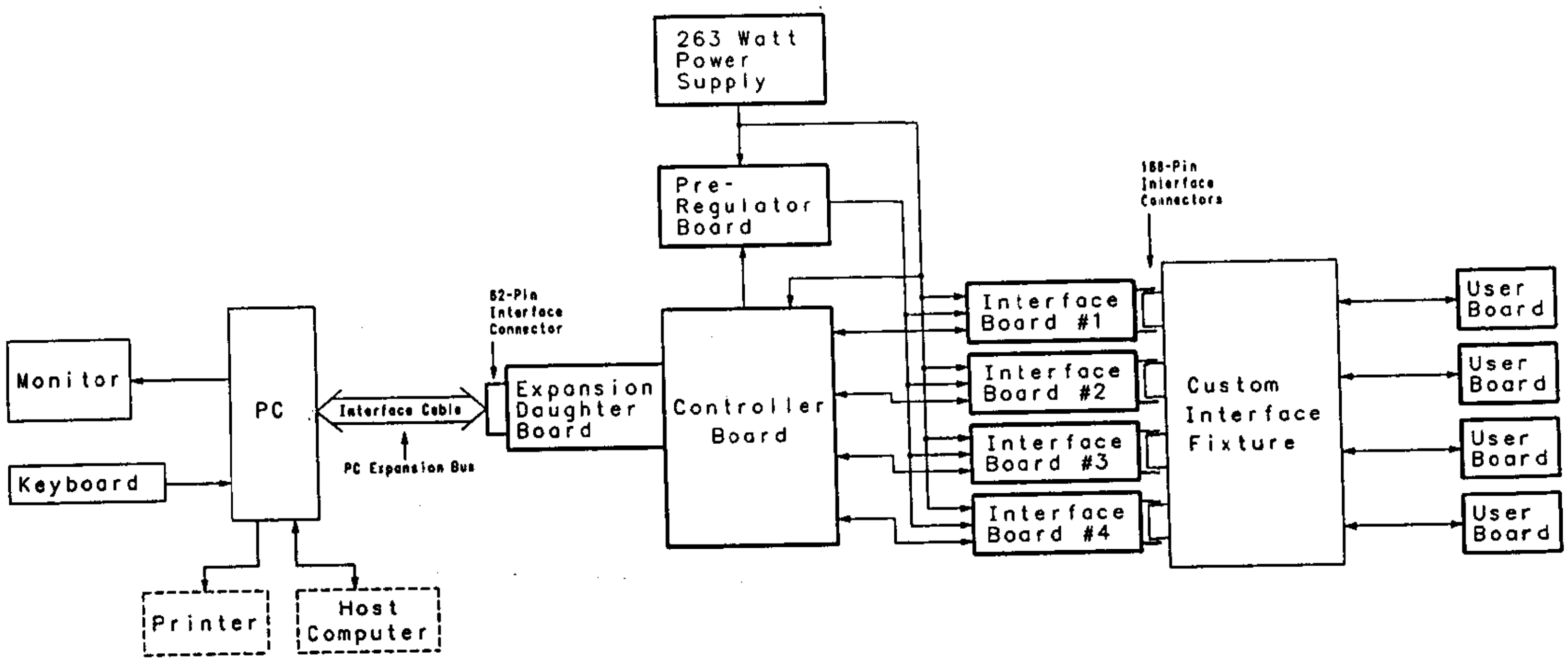




EXTERNAL POWER SUPPLY



INTERNAL POWER SUPPLY



BOARDSITE SYSTEM BLOCK DIAGRAM