Quick Installation

Installing the Board

Step 1: Insert the *Installation Disk* into drive A.

Step 2: Change to drive A and enter **SURVEY**.

Step 3: If **SURVEY** detects a conflict, see the user's guide for instructions. If it doesn't, remove the diskette, turn your PC off, and install the SDB20 in your computer. Be sure to discharge any static electricity before touching the SDB20—ground yourself by touching the PC's power supply (usually a large silver or black box in the rear of the computer case).

*Remove the PC cover*

*Install the SDB20 into a vacant slot*

Step 4: Connect the VGA passthrough cable to your VGA board. Connect your monitor cable to your SDB20 and turn on your PC.

*Connect the VGA passthrough cable*

*Connect the monitor cable*

Step 5: Insert the *Installation Disk* into drive A.

Step 6: Change to drive A and enter **TIGASET**.

Step 7: Select the *Installation TIGA* option and follow the instructions on the screen.
Quick Installation

Installing the Windows Driver

If a TIGA Windows driver is supplied with your Microsoft Windows package, install it using the standard Windows set up. If the TIGA Windows driver is not supplied, install the driver that came with the board by following the steps below.

**Step 1:** Make sure the TIGA communications are running by rebooting your PC (if you had tigaset, edit your autoexec.bat file) or by running TIGAAUTO.

**Step 2:** Change directories to your Windows directory and enter SETUP.

**Step 3:** At the System Information screen, choose Display.

---

### Windows Setup

**If your computer or network appears on the Hardware Compatibility List with an asterisk next to it, press F1 before continuing.**

**System Information**

- **Computer:** MS-DOS or PC-DOS System
- **Display:** VGA
- **Mouse:** Microsoft, or IBM PS/2
- **Keyboard:** Enhanced 101 or 102 key US and Non US
- **Keyboard Layout:** US
- **Language:** English (American)
- **Network:** LAN Manager 1.x (or 100% compatible)

**Complete Changes:** Accept the configuration shown above.

To change a system setting, press the UP or DOWN ARROW key to move the highlight to the setting you want to change. Then press ENTER to see alternatives for that item. When you have finished changing your settings, select the “Complete Changes” option to exit Setup.

ENTER = Continue  F1 = Help  F3 = Exit

---

**Step 4:** You are presented with a list of drivers. Go to the bottom of the list and choose Other.

---

### Windows Setup

**You have asked to change the type of Display to be installed. Select a Display from the following list:**

- 8514/a
- CGA
- Compaq Portable Plasma
- EGA
- Hercules Monochrome
- Olivetti/AT&T Monochrome or PVC Display

(To see more of the list, press the ↓ arrow key)

Press the UP or DOWN ARROW key to move the highlight to the item you want to select. Then press ENTER.

If you want to return to the System Information screen without changing your Display type, press ESC.

ENTER = Continue  F1 = Help  F3 = Exit  ESC = Cancel

---

**Step 5:** Enter your TIGA directory path followed by \win30 (i.e., `<TIGA DIRECTORY>`\win30). The TIGA Windows driver is presented in 3 variations. Choose the appropriate driver for your system.
TMS34020
Software Development Board
User’s Guide

2558670-9721 revision B
July 1991
IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to or to discontinue any semiconductor product or service identified in this publication without notice. TI advises its customers to obtain the latest version of the relevant information to verify, before placing orders, that the information being relied upon is current.

TI warrants performance of its semiconductor products to current specifications in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Unless mandated by government requirements, specific testing of all parameters of each device is not necessarily performed.

TI assumes no liability for TI applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Texas Instruments products are not intended for use in life-support appliances, devices, or systems. Use of a TI product in such applications without the written consent of the appropriate TI officer is prohibited.
This BYTE Award of Excellence recognizes this product as one that bridges to the future and truly advances the state of the art.

Frederic S. Longo, Editor in Chief

J. Burt Tolara, Publisher/Group Vice President

Byte editors chose TIGA as one of 1989's ten most technologically advanced products. The magazine concluded that TIGA is well on its way to becoming the standard for high-resolution graphics applications development.
How to Use This Manual

This document contains the following chapters:

Chapter 1  Introduction
Provides an overview of the key features for the TMS34020 Software Development Board (SDB20).

Chapter 2  Installing SDB20 Hardware and Software
Identifies hardware and software system requirements; provides step-by-step installation procedures.

Chapter 3  Theory of Operation

Appendix A  Physical Specifications
Provides detailed technical information about the SDB20.

Appendix B  Troubleshooting
Outlines troubleshooting procedures; gives customer support information.

Appendix C  Schematics
Contains the schematics for the SDB20.

Appendix D  Glossary
Related Documentation

The following TMS34020 documents are available from Texas Instruments. To obtain a copy of any of these TI documents, please call the Texas Instruments Customer Response Center (CRC) at (800) 336-5236, or if you are outside of the U.S. and Canada, call (214) 995-6611. When ordering, please identify the book by its title and its literature number.

- The **TMS340 Family Code-Generation Tools User’s Guide**, literature number SPVU020, describes the C compiler, assembler, linker, archiver, and auxiliary tools that are available for developing TMS34010 or TMS34020 code.

- The **TMS340 Family C Source Debugger User’s Guide**, literature number SPVU021, tells how to use the TMS340 Family C Source Debugger with the TMS34010 TIGA Development Board (TDB10), the TMS34020 Software Development Board (SDB20), and the TMS34020 Emulator.

- The **TMS340 Family Graphics Library User’s Guide**, literature number SPVU027, describes the graphics operations library that is available for a TMS340-based graphics system.

- The **TIGA Interface User’s Guide**, literature number SPVU015, describes the Texas Instruments Graphics Architecture (TIGA), a software interface that standardizes communication between application software and TMS340-based hardware for IBM-compatible PCs.

- The **TMS34020 User’s Guide**, literature number SPVU019, describes the TMS34020, which is the second-generation graphics processor in the TMS340 family of computer video products.

- The **TMS34082 SRAM Module Technical Reference**, literature number SCGU002, describes the TMS34082 SRAM module, which extends the capabilities of the SDB20 by providing a platform for developing custom, floating-point routines that use external instructions.

- The **TMS34082 Designer’s Handbook**, literature number SCGU004, discusses the hardware aspects of the TMS34082, such as pin functions, architecture, stack operation, and interface. It also contains the TMS34082 internal and external instructions sets.

- **Pixel Perspectives**, literature number SPVN05x, is a newsletter published by the Computer Video Products Group of Texas Instruments Incorporated. It describes new products, discusses support for existing products, and identifies new documentation releases.
Style and Symbol Conventions

This document uses the following conventions.

- Program listings, program examples, interactive displays, filenames, and symbol names are shown in a special font. Some examples use a bold version to identify code, commands, or portions of an example that you enter.

Here is an example of a system prompt and a command that you might enter:

```
A: tigaset
```

- `CTRL/V, \^V, and [CTRL][V]` are synonymous and mean to press the keyboard CTRL (CNTL on some keyboards) and the V keys together.

- `<CR>, <RETURN>, <ENTER>, ENTER, and \[RETURN\]` are synonymous and mean to press the keyboard ENTER or RETURN keys.

- `<SP>` and `SPACE` are synonymous and mean to press the keyboard space bar.

- `<ESC>` and `ESC` are synonymous and mean to press the keyboard ESC key.

Information About Cautions and Warnings

This book contains cautions and warnings. The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

This is what a caution looks like.

A caution describes a situation that could potentially damage your software or equipment.

This is what a warning looks like.

A warning describes a situation that could potentially cause harm to you.
Trademarks

Anvil 1000 and Anvil 5000 are trademarks of Manufacturing Consulting Services, Inc. AutoCAD Release 10, AutoShade, AutoSketch are registered trademarks of Autodesk, Inc. Bt478 is a trademark of Brooktree Corp. CADKEY 3 V3.5, CADKEY Render, DataCAD, and DataCAD Velocity are trademarks of CADKEY, INC. DesignCAD 2D and DesignCAD 3D are trademarks of American Small Business Computing. DGIS and GSS*CGI are trademarks of Graphic Software Systems, Inc. Drawbase is a trademark of CADworks, Incorporated. GEM Artline, GEM Presentation Team, and Digital Research Draw Plus are trademarks of Digital Research, Inc. Generic CADD is a trademark of Generic Software. GSPOT I, II, and III are trademarks of Pixelab, Inc. HALO is a trademark of Media Cybernetics, Inc. HOOPS is a trademark of Ithaca Software. IBM PC, IBM PC/AT, and IBM PC/XT are trademarks of International Business Machines. Intel, i286, and i386 are trademarks of Intel Corp. Lotus, Lotus 1-2-3 Release 2.01 and Release 2.2, and Symphony are trademarks of Lotus. MasterCAM is a trademark of CNC Software. MS-DOS, Presentation Manager, and Windows 3.0 are trademarks of Microsoft Corp. Microstation is a trademark of Intergraph Corporation. MultiSync and MultiSync Graphics are trademarks of NEC Home Electronics (U.S.A) Inc. OrCAD/SDT III, OrCAD PCB II, and OrCAD VST are trademarks of OrCAD L.P. PAL is a registered trademark of Advanced Micro Devices, Inc. Panacea is a trademark of Panacea, Inc. RoboCAD 2.2 is a trademark of Robo Systems International, Inc. SDB20 and TIGA are trademarks of Texas Instruments Incorporated. Ventura Publisher/GEM is a registered trademark of Ventura Software, Inc. X Window System is a trademark of the Massachusetts Institute of Technology. XSoftware is a trademark of AGE.
Contents

1 Introduction ................................................................. 1-1
Provides an overview of the key features for the TMS34020 software development board (SDB20).

1.1 An Overview of the SDB20 ........................................... 1-2
1.2 The TIGA Software Interface .......................... 1-4
1.3 The TMS34020 Graphics Processor ............... 1-5
1.4 The TMS34082 Graphics Floating-Point Processor 1-6

2 Installing SDB20 Hardware and Software ......................... 2-1
Identifies hardware and software system requirements; provides step-by-step installation procedures.

2.1 System Requirements ............................................... 2-2
2.1.1 Hardware and Software Requirements .............. 2-2
2.1.2 Cabling ...................................................... 2-2
2.2 Installing the SDB20 .............................................. 2-3
2.2.1 Determining Your System Configuration ...... 2-3
2.2.2 Preparing the SDB20 for Installation ...... 2-4
2.2.3 Procedures for Installing the SDB20 Into Your PC 2-6
2.3 Installing and Running TIGA Setup Utilities .......... 2-12
2.3.1 Before Running tigaset ............................... 2-12
2.3.2 Installing TIGA Software ............................. 2-12
2.4 Setting Up a Dual Monitor System ................. 2-23
2.5 Customizing Monitor Timing ....................... 2-24
2.6 TIGA Software Drivers ...................................... 2-28
2.6.1 TIGA Video Drivers .................................. 2-28
2.6.2 Installing the TIGA Windows Driver ........ 2-29

3 Theory of Operation .................................................... 3-1

3.1 Host Interface ...................................................... 3-2
3.1.1 Host I/O Decoding ....................................... 3-2
3.1.2 SDB20 Host Communication Registers .......... 3-4
3.1.3 Address Conversion Principles .................. 3-12
3.1.4 Host Interface Control PALs .................... 3-18
3.1.5 Host 8-Bit I/O Write Cycle ...................... 3-28
3.1.6 Host 8-Bit Memory Read Cycle .............. 3-30
3.1.7 Host 16-Bit Memory Write Cycle .............. 3-32
3.1.8 VGA Passthrough .................................................. 3-34
3.1.9 VGA Passthrough Cycle ........................................... 3-40
3.1.10 Resetting the SDB20 ............................................... 3-42
3.1.11 SDB20 Interrupt Control ......................................... 3-43
3.2 The Frame Buffer/Program Memory Interface ....................... 3-45
3.2.1 TMS34020 Memory Mapping ....................................... 3-45
3.2.2 The TMS34020-to-Memory Interface ................................ 3-46
3.2.3 The RASDCD PAL, U1 ............................................. 3-48
3.2.4 TMS34020-to-VRAM Interface .................................... 3-51
3.2.5 TMS34020-to-DRAM Memory Interface ........................ 3-52
3.3 The TMS34082 Floating-Point Coprocessor .......................... 3-53
3.3.1 How the TMS34082 Communicates With the TMS34020 ........... 3-53
3.3.2 The HAND PAL, U14 ............................................. 3-54
3.3.3 Installing the TMS34082 ........................................... 3-54
3.3.4 The TMS34082 SRAM Module ..................................... 3-55
3.3.5 The TMS34082 Bus Architecture .................................. 3-56
3.3.6 The TMS34082 Bus Architecture With the SRAM Module Installed . 3-56
3.3.7 For Additional Information ........................................ 3-57
3.4 Backend Section ....................................................... 3-58
3.4.1 The Frame Buffer-to-Color Palette Interface .................... 3-58
3.4.2 The MISC PAL, U8 ................................................. 3-59
3.4.3 The Bt478 Color Palette .......................................... 3-60
3.4.4 The SDB20 Video Output Connector, P4 .......................... 3-61
3.5 The TMS34020-to-XDS500 Interface .................................. 3-62

A SDB20 Specifications .................................................. A-1
Provides detailed technical information about the SDB20.
A.1 SDB20 Hardware Specifications .................................. A-2
A.2 Video Output Connector Pinouts .................................... A-3
A.3 Video Timing ........................................................ A-4
A.4 SDB20 Input/Output Registers ...................................... A-5
A.5 SDB20 Hardware Interrupt Settings ................................ A-6

B Troubleshooting Guide .................................................. B-1
Outlines troubleshooting procedures; gives customer support information.
B.1 TIGA Error Messages ............................................... B-2
B.2 Customer Support ..................................................... B-6

C Schematics ............................................................... C-1
Contains the schematics for the SDB20.

D Glossary ................................................................. D-1
## Figures

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Software Development Board Block Diagram</td>
<td>1-3</td>
</tr>
<tr>
<td>1-2</td>
<td>Graphics Processing Shared Between TMS340 and Host Processors</td>
<td>1-4</td>
</tr>
<tr>
<td>2-1</td>
<td>Removing the PC Cover</td>
<td>2-7</td>
</tr>
<tr>
<td>2-2</td>
<td>Removing the PC Mounting Bracket</td>
<td>2-8</td>
</tr>
<tr>
<td>2-3</td>
<td>Installing the SDB20</td>
<td>2-9</td>
</tr>
<tr>
<td>2-4</td>
<td>Connecting the VGA Passthrough Cable</td>
<td>2-10</td>
</tr>
<tr>
<td>2-5</td>
<td>Connecting the 9-Pin Monitor Cable</td>
<td>2-10</td>
</tr>
<tr>
<td>2-6</td>
<td>The TIGA Setup Utility Screen — Main Menu</td>
<td>2-13</td>
</tr>
<tr>
<td>2-7</td>
<td>TIGA Installation Screen — Host I/O Base Address</td>
<td>2-14</td>
</tr>
<tr>
<td>2-8</td>
<td>TIGA Installation Screen — Host Memory Segment Base Address</td>
<td>2-15</td>
</tr>
<tr>
<td>2-9</td>
<td>TIGA Installation Screen — Data Transfer Size</td>
<td>2-16</td>
</tr>
<tr>
<td>2-10</td>
<td>TIGA Installation Screen — Default Directory</td>
<td>2-17</td>
</tr>
<tr>
<td>2-11</td>
<td>TIGA Installation Screen — TIGA Environment</td>
<td>2-17</td>
</tr>
<tr>
<td>2-12</td>
<td>TIGA Installation Screen — Monitor Configuration</td>
<td>2-18</td>
</tr>
<tr>
<td>2-13</td>
<td>TIGA Installation Screen — Defined Monitors</td>
<td>2-19</td>
</tr>
<tr>
<td>2-14</td>
<td>TIGA Installation Screen — Determining the Monitor Color</td>
<td>2-19</td>
</tr>
<tr>
<td>2-15</td>
<td>TIGA Installation Screen — Correcting the Monochrome Display Problem</td>
<td>2-20</td>
</tr>
<tr>
<td>2-16</td>
<td>Typical Change TIGA Display Mode Screen</td>
<td>2-21</td>
</tr>
<tr>
<td>2-17</td>
<td>TIGA Information Screen for the SDB20</td>
<td>2-22</td>
</tr>
<tr>
<td>2-18</td>
<td>Typical Video Timing Diagram</td>
<td>2-24</td>
</tr>
<tr>
<td>2-19</td>
<td>Typical Change Monitor Option Window</td>
<td>2-26</td>
</tr>
<tr>
<td>2-20</td>
<td>Windows Setup Screen</td>
<td>2-30</td>
</tr>
<tr>
<td>2-21</td>
<td>Windows Setup Screen — Display</td>
<td>2-31</td>
</tr>
<tr>
<td>2-22</td>
<td>Windows Setup Screen—Display Driver</td>
<td>2-32</td>
</tr>
<tr>
<td>3-1</td>
<td>The SDB20 Software Development Board</td>
<td>3-3</td>
</tr>
<tr>
<td>3-2</td>
<td>The HINTF Register</td>
<td>3-6</td>
</tr>
<tr>
<td>3-3</td>
<td>The SDB20 Status Register</td>
<td>3-9</td>
</tr>
<tr>
<td>3-4</td>
<td>Hardware Space Control Register (HSREG)</td>
<td>3-10</td>
</tr>
<tr>
<td>3-5</td>
<td>ABEL Source File for the HARD PAL, U13</td>
<td>3-12</td>
</tr>
<tr>
<td>3-6</td>
<td>Converting a PC Host Address to a TMS34020 Local Address</td>
<td>3-14</td>
</tr>
</tbody>
</table>
3-7. Converting a TMS34020 Local Address to a Host PC Address ..................... 3-16
3-8. ABEL Source File for the GSPHST PAL, U26 ...................................... 3-19
3-9. ABEL Source File for the HBS PAL, U28 ............................................. 3-22
3-10. ABEL Source File for the SELECT PAL, U40 ..................................... 3-25
3-11. ABEL Source File for the MAPEN PAL, U27 ..................................... 3-26
3-12. Host-to-SDB20 I/O Write Cycle ..................................................... 3-28
3-13. Host-to-SDB20 8-Bit Memory Read Cycle ......................................... 3-30
3-14. Host-to-SDB20 16-Bit Memory Write Cycle ....................................... 3-32
3-15. VGA Passthrough Cable Connection ................................................ 3-35
3-16. ABEL Source File for the SHADOW Device, U25 ................................. 3-37
3-17. ABEL Source File for the MISC PAL, U8 ........................................... 3-39
3-18. VGA Passthrough Cycle ............................................................... 3-40
3-19. The ABEL Source File for the RASDCD PAL, U1 ................................. 3-50
3-20. ABEL Source File for the HAND PAL, U14 ....................................... 3-54
3-21. TMS34082 SRAM Module Installation ............................................. 3-55
3-22. The TMS34082 Bus Architecture in Standard SDB20 ............................. 3-56
3-23. The TMS34082 Bus Architecture Using the SRAM Module ...................... 3-57
B-1. SDB20 Installation Option Window .................................................. B-2
<table>
<thead>
<tr>
<th>Table number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Hardware Requirements</td>
<td>2-2</td>
</tr>
<tr>
<td>2-2</td>
<td>SW1 I/O Space Settings</td>
<td>2-5</td>
</tr>
<tr>
<td>2-3</td>
<td>SW2 Interrupt Settings</td>
<td>2-6</td>
</tr>
<tr>
<td>2-4</td>
<td>TIGA Video Drivers</td>
<td>2-28</td>
</tr>
<tr>
<td>3-1</td>
<td>The SDB20 Communication Registers</td>
<td>3-4</td>
</tr>
<tr>
<td>3-2</td>
<td>SDB20 Communication Register Memory Map</td>
<td>3-5</td>
</tr>
<tr>
<td>3-3</td>
<td>The MAPA and MAPB Mapping Registers</td>
<td>3-8</td>
</tr>
<tr>
<td>3-4</td>
<td>SDBST Bit Field Descriptions</td>
<td>3-9</td>
</tr>
<tr>
<td>3-5</td>
<td>Host and Data Latch Byte Select</td>
<td>3-21</td>
</tr>
<tr>
<td>3-6</td>
<td>The TMS34020 Local Memory Map</td>
<td>3-45</td>
</tr>
<tr>
<td>3-7</td>
<td>Local Memory Addresses-to-RCA Bus Connections</td>
<td>3-47</td>
</tr>
<tr>
<td>3-8</td>
<td>DRAM and VRAM Bank Selection</td>
<td>3-48</td>
</tr>
<tr>
<td>3-9</td>
<td>RASDCD PAL Signal Selection</td>
<td>3-48</td>
</tr>
<tr>
<td>3-10</td>
<td>TMS44C251 Functional Truth Table</td>
<td>3-52</td>
</tr>
<tr>
<td>A-1</td>
<td>SDB20 Hardware Specifications</td>
<td>A-2</td>
</tr>
<tr>
<td>A-2</td>
<td>Video Output Connector Pinouts</td>
<td>A-3</td>
</tr>
<tr>
<td>A-3</td>
<td>1024 × 768 Noninterlaced Video Timing</td>
<td>A-4</td>
</tr>
<tr>
<td>A-4</td>
<td>640 × 480 Noninterlaced Video Timing</td>
<td>A-4</td>
</tr>
<tr>
<td>A-5</td>
<td>ISA I/O Address Locations</td>
<td>A-5</td>
</tr>
<tr>
<td>A-6</td>
<td>SDB20 Hardware Interrupt Settings</td>
<td>A-6</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The TMS34020 Software Development Board (SDB20) is a high-performance ISA bus-compatible graphics card that features the TMS34020 graphics processor.

This software development tool is designed for developing application software that is compatible with the TMS34020 graphics processor. This module demonstrates the simplicity of designing hardware by using the TMS34020 to develop a high-performance bit-mapped graphics display.

The SDB20 package includes the board, this manual, a high-density floppy disk containing demonstration and utility software, and the TMS340 Family C Source Debugger.

Instructions for installation of the board and the demonstration software are contained in Section 2.2, *Installing the SDB20*.

---

**Note: TMS34020 Assembly Language Model**

It is assumed that persons using the SDB20 have a knowledge of the TMS34020 graphics processor assembly language. The instruction set is explained in the *TMS340 Family Code Generation Tools* package and in the *TMS34020 User's Guide*.

---

Topics covered in this introductory chapter include:

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 An Overview of the SDB20</td>
<td>1-2</td>
</tr>
<tr>
<td>1.2 TIGA Software Interface</td>
<td>1-4</td>
</tr>
<tr>
<td>1.3 The TMS34020 Graphics Processor</td>
<td>1-5</td>
</tr>
<tr>
<td>1.4 The TMS34082 Floating-Point Coprocessor</td>
<td>1-6</td>
</tr>
</tbody>
</table>
1.1 An Overview of the SDB20

The TMS34020 SDB consists of a 32-bit TMS34020 graphics processor with the following features:

- A powerful 32-MHz TMS34020 graphics processor
- TIGA 2.05 compatibility
- An optional TMS34082 floating-point coprocessor
- An optional TMS34082 SRAM module
- A 32-bit local data bus
- Configurable PC bus transfer width (8- or 16-bit)
- Variable display resolutions:
  - 1024 H \times 768 V noninterlaced
  - 640 H \times 480 V noninterlaced
- Selectable screen resolution as follows:
  - 1024 \times 768 pixels, 256 colors/gray levels
  - 1024 \times 768 pixels, 16 colors/gray levels
  - 640 \times 480 pixels, 256 colors/gray levels
  - 640 \times 480 pixels, 16 colors/gray levels
- VGA passthrough
- 1 MByte zero wait-state video RAM (VRAM)
- 1 MByte zero wait-state dynamic RAM (DRAM)
- Supports TMS34020 emulation
- Supports selectable PC bus interrupts
- Software-configurable base address, extended 16 MByte range

Figure 1–1 is a simplified block diagram that outlines the principle blocks and data paths within the SDB20. These are discussed in detail in Chapter 3, *Theory of Operation*. 
Figure 1–1. Software Development Board Block Diagram
1.2 The TIGA Software Interface

The Texas Instruments Graphics Architecture (TIGA) is a software interface standard for the TMS340 family of graphics processors. Figure 1–2 shows the relationship between the TMS340 and host processors.

Figure 1–2. Graphics Processing Shared Between TMS340 and Host Processors

The TIGA software interface provides:

- A standard communication protocol between the host processor and the TMS340 processor.
- The ability to divide tasks between the TMS340 processor and the 80x86 host so that the graphics-intensive functions can run in parallel, thus improving system efficiency.
- The ability to customize TIGA to take advantage of any added feature available on the target TMS340-based board.
- The ability to develop portable applications and application drivers for the diverse range of TMS340-based systems.
1.3 The TMS34020 Graphics Processor

The TMS34020 is the second generation of the TMS340 family of graphics processors. It combines the best features of a general-purpose processor and a graphics controller.

The instruction set for the TMS34020 provides a full complement of general-purpose instructions (e.g., jumps and calls) and supports pixel drawing, window, and Boolean operations. The TMS34020 architecture supports a variety of pixel sizes, frame buffer sizes, and screen sizes.

On-chip functions have been carefully selected so that no function ties the TMS34020 to a particular display resolution. This enhances the portability of graphics standards, such as MIT's X, CGI/CGM, PHIGS, and Microsoft's Windows and Presentation Manager.
1.4 The TMS34082 Graphics Floating-Point Processor

The TMS34082 Graphics Floating-Point Processor is designed to interface directly with the TMS34020. This allows the TMS34020 to perform computation-intensive functions many times faster than a software implementation.

The TMS34082 performs single- and double-precision floating-point operations, conforming to the IEEE 754 standard. It also performs complex 2- and 3-dimensional operations such as 3×3 convolution, 4×4 matrix, and cubic spline operations.
This chapter illustrates the installation requirements and procedure for using the SDB20. The SDB20 is a full-size printed circuit board that can be placed in either an 8- or 16-bit slot of a standard ISA bus. The board is software selectable for either 8- or 16-bit memory transfers; it is limited to 8-bit I/O transfers.

The SDB20 can be used effectively in a single monitor system because of the onboard VGA passthrough feature, which allows interactive switching between the high-resolution graphics generated by the TMS34020 and those generated by VGA.

The topics discussed in this chapter include:

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 System Requirements</td>
<td>2-2</td>
</tr>
<tr>
<td>2.2 Installing the SDB20</td>
<td>2-3</td>
</tr>
<tr>
<td>2.3 Installing and Running TIGA Utilities</td>
<td>2-12</td>
</tr>
<tr>
<td>2.4 Setting Up a Dual Monitor System</td>
<td>2-23</td>
</tr>
<tr>
<td>2.5 Customize Monitor Timing</td>
<td>2-24</td>
</tr>
<tr>
<td>2.6 TIGA Software Drivers</td>
<td>2-28</td>
</tr>
</tbody>
</table>
2.1 System Requirements

2.1.1 Hardware and Software Requirements

In addition to the items that were shipped with the SDB20, you will need the items listed in Table 2–1.

Table 2–1. Hardware Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDB20 requirements</td>
<td>Approximately 2 amps @ +5 V (10 watts)</td>
</tr>
<tr>
<td>Host PC</td>
<td>An IBM PC/AT, IBM PC/XT, or 100% compatible PC EISA system with a hard-disk system and a floppy-disk drive.</td>
</tr>
<tr>
<td>Display</td>
<td>Color (RGB)</td>
</tr>
<tr>
<td>Slot</td>
<td>One 8-bit or 16-bit slot</td>
</tr>
<tr>
<td>Graphics card</td>
<td>A VGA-compatible graphics display card. This card is needed only if you are using the VGA passthrough feature.</td>
</tr>
<tr>
<td>Miscellaneous materials</td>
<td>A small Phillips or flat-blade screwdriver, depending upon the type of screws used in your PC.</td>
</tr>
<tr>
<td>Operating system</td>
<td>MS-DOS or PC-DOS (version 3.0 or higher).</td>
</tr>
</tbody>
</table>

2.1.2 Cabling

The TMS34020 SDB20 provides a DB9 video output connector (P4). Its pinout is shown in Section A.2, Video Output Connector Pinouts. Some monitors may require a 9-pin to 15-pin connector adapter. If so, see your PC operator’s manual for video connector pinouts and see Section A.2, Video Output Connector Pinouts of this manual.
2.2 Installing the SDB20

This section contains all the information for installing the SDB20. The only tool required for installation is a Phillips or small flat-blade screwdriver for removing the PC cover, removing the slot mounting bracket, and installing the SDB20.

The installation process consists of the following steps:

Step 1: Determining your system configuration (subsection 2.2.1).

Step 2: Preparing the SDB20 for installation (subsection 2.2.2).

Step 3: SDB20 installation (subsection 2.2.3).

2.2.1 Determining Your System Configuration

The SDB20 can be installed in an 8-bit or 16-bit slot. Your choice should be based on the following considerations:

1) If you have a 16-bit VGA or EGA board installed, install the SDB20 into 16-bit slot.

2) If you have an 8-bit VGA or EGA board installed, install the SDB20 into either an 8-bit or 16-bit slot.

3) If you have a CGA or MDA board installed, install the SDB20 into either an 8-bit or 16-bit slot.

Accompanying the SDB20 is a diskette labeled \textit{SDB20 Installation Diskette}. On this diskette are two utility programs that you need for installing the SDB20.

\begin{itemize}
\item The first utility is named \textit{survey}; it checks for conflicts between the SDB20 I/O address and interrupt settings and the other boards that are installed in your PC.
\item The second utility is named \textit{tigaset}. This utility has many uses; for the purposes of this installation, it installs the TIGA software and sets up the TIGA environment.
\end{itemize}

Some peripheral hardware installed in your PC may conflict with the SDB20 bus assignments. To avoid potential problems, do the following to survey your PC environment:

Step 1: Insert the \textit{Installation Diskette} into your floppy drive A.

Step 2: Execute the \textit{survey} program by entering \texttt{A:survey} \texttt{ENTER}.
After a short pause, the utility displays a message similar to this:

```
Ready for SDB20 at I/O Address 230 using Interrupt IRQ15. This is the default setting for the I/O address. This is the default setting for the interrupt.
```

The default location for the I/O address is 0x230; the default setting for the interrupt is IRQ15.

If there is a potential conflict, the message will indicate the steps that you must take to resolve the problem. See subsection 2.2.2.1, How to Reconfigure Your SDB20, for details.

### 2.2.2 Preparing the SDB20 for Installation

Once the PC environment is determined, be sure that the SDB20's SW1 I/O register and SW2 interrupt switches are set correctly to identify the following information:

- Switch **SW1** identifies the **I/O address space** that the SDB20 will use for communicating with the PC.
- Switch **SW2** identifies the **interrupt** that the SDB20 will use for host interrupt requests.

#### 2.2.2.1 How to Reconfigure Your SDB20

Your SDB20 communicates with the host computer by means of 16 consecutive, 8-bit wide I/O address locations and a hardware interrupt. Your SDB20 is factory preconfigured to respond at address 230h and hardware interrupt 15.

If these locations are occupied by another add-in peripheral, such as a local area network (LAN) card, an address conflict occurs because two hardware components cannot occupy the same address location. The *survey* utility detects potential conflicts and warns you that relocation is necessary. To resolve the conflict, relocate either your SDB20 or the offending hardware to a new location.

**Note:**

If you are reconfiguring an existing TIGA environment, go to subsection 2.2.3, Procedures for Installing the SDB20 Into Your PC, on page 2-6, for instructions on installing the SDB20. After installing the board, go to Section 2.3, Installing and Running TIGA Setup Utilities, for instructions on running tigaset. Choose the **Modify Existing TIGA Settings** option, then choose the **Update TIGA** option.
The I/O address and host interrupt can be changed from the factory preset by setting SW1 and SW2 to an alternative location. The following steps show how this is done:

Do not touch the SDB20 before discharging any static electricity from your body. You could possibly damage SDB20 circuitry. Ground yourself by touching the PC's power supply (usually a large silver or black box in the rear of the computer case).

With the SDB20 board removed from your PC, locate the switch that you wish to change. The right side of each switch is labeled with an arrow: the arrow indicates the ON position. A switch in the ON position programs a 0 for that address bit.

To reconfigure the address range from 230h to another address, set SW1 to an alternate address listed in Table 2–2.

Table 2–2. SW1 I/O Space Settings

<table>
<thead>
<tr>
<th>Address</th>
<th>SW1-1</th>
<th>SW1-2</th>
<th>SW1-3</th>
<th>SW1-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0200</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>0210</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>0220</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>0230</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>0240</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>0250</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>0260</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>0270</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>0280</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>0290</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>02A0</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>02B0</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>02C0</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>02D0</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>02E0</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>02F0</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

To reconfigure the host interrupt from IRQ15 to another interrupt location, set SW2 to an alternate location listed in Table 2–3.
Table 2–3. SW2 Interrupt Settings

<table>
<thead>
<tr>
<th>Switches</th>
<th>Interrupt Line</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2–1</td>
<td>IRQ3</td>
<td>available</td>
</tr>
<tr>
<td>SW2–2</td>
<td>IRQ5</td>
<td>available</td>
</tr>
<tr>
<td>SW2–3</td>
<td>IRQ9</td>
<td>available</td>
</tr>
<tr>
<td>SW2–4</td>
<td>IRQ10</td>
<td>available</td>
</tr>
<tr>
<td>SW2–5</td>
<td>IRQ11</td>
<td>available</td>
</tr>
<tr>
<td>SW2–6</td>
<td>IRQ12</td>
<td>available</td>
</tr>
<tr>
<td>SW2–7</td>
<td>IRQ14</td>
<td>reserved</td>
</tr>
<tr>
<td>SW2–8</td>
<td>IRQ15</td>
<td>available</td>
</tr>
</tbody>
</table>

2.2.3 Procedures for Installing the SDB20 Into Your PC

To install the SDB20, do the following:

**WARNING**

Turn off power to your PC, monitor(s), peripherals, and the target system (if applicable) before removing or replacing any circuit cards or cables.

**Step 1:** Remove the cover from your PC by following the instructions contained in your PC’s operator manual. Typically, this entails removing the screws from the back of your PC and sliding the cover off as shown in Figure 2–1.
Step 2: Remove the mounting bracket from an unused 8-bit or 16-bit slot (see Figure 2–2) and save the screw for reinstallation.

- If you are using only one display monitor with your system, find an open slot. You may want to relocate the VGA board to a different bus slot to accommodate this configuration.

- If you are using two monitors on your system, one for the SDB20 and the other for a primary display, refer to Section 2.4, Setting Up a Dual Monitor System.
Step 3: Touch the PC's power supply case (usually a large silver or black box in the rear of the computer case) to discharge any static electricity on your body.

Do not touch the SDB20 before discharging any static electricity from your body. You could possibly damage SDB20 circuitry. Ground yourself by touching the PC's power supply (usually a large silver or black box in the rear of the computer case).

Step 4: Remove the SDB20 from its protective bag.

Step 5: Install the SDB20 in the selected slot (see Figure 2–3); press firmly and evenly on the top edge of the card so that it seats all the way into the slot connectors.
Figure 2–3. Installing the SDB20

Do not force the SDB20 into the slot. Avoid touching any board component while installing the board.

Step 6: Tighten down the mounting bracket with the screw saved in Step 2.

Step 7: Connect a VGA passthrough cable to your feature connector as shown in Figure 2–4.
Figure 2–4. Connecting the VGA Passthrough Cable

Note:
Some monitors may require a 9-pin to 15-pin connector adapter. See your monitor specifications for video connector pinouts and see Section A.2, Video Connector Pinouts.

Step 8: Connect the monitor cable to the 9-pin connector on the side of the SDB20 as shown in Figure 2–5.

Figure 2–5. Connecting the 9-Pin Monitor Cable
Step 9: Replace the PC cover.

Step 10: Write down your monitor's manufacturer and model number in the box below. The SDB20 setup utility requires this information.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Model Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 11: Turn on power to the PC.
2.3  Installing and Running TIGA Setup Utilities

2.3.1 Before Running *tigaset*

The *tigaset* utility requires 450 kilobytes of free memory and 1 MByte of fixed disk space to run. Make sure that you have this amount of memory available, or

1) TIGA will be configured incorrectly.

2) The system will lock up.

The *tigaset* utility writes data back to the installation diskette during the installation process. To maintain the integrity of the original installation diskette, copy the installation diskette onto a formatted diskette that is not write-protected, and use the copy for installation.

The screen diagrams contained in this user’s guide are representative of the screens that you will see when using the *tigaset* utility. The exact information on the screen may vary, depending on the board model that you are using and its memory population.

The installation diskette contains a file called *read.me*. It provides the latest information about TIGA installation and operation.

2.3.2 Installing TIGA Software

The SDB20 setup utility, *tigaset*, configures the SDB20 hardware and sets up the TIGA environment. Follow the simple instructions displayed on your screen and go through the following steps:

**Step 1:** Insert the *Installation Diskette* into your floppy drive A.

**Step 2:** Enter A: and press [ENTER].

**Step 3:** Enter *tigaset* and press [ENTER].

This invokes the *tigaset* utility. It displays a menu similar to that shown in Figure 2–6.

---

**Note:**

If this is a first-time installation, you will see a subset of the options shown in Figure 2–6.
Figure 2-6. The TIGA Setup Utility Screen — Main Menu

Position the cursor using the arrow keys, then press <ENTER> to select the highlighted option.

Note:
Pressing [ESC] aborts the screen that you are working in and returns the main menu.

An explanation of the main menu options follows:

- **(Re)Install TIGA** presents a series of instructions and options for installing TIGA. For a first-time installation, the screen shows Install TIGA.
  - Selects the host I/O base address setting.
  - Selects the host memory segment base address.
  - Selects the data transfer size: 8 bits or 16 bits.
  - Copy TIGA files into the TIGA directory.
  - Select TIGA system settings.

- **TIGA Board Diagnostics** provides you with a set of diagnostic tools to help you locate a suspected hardware failure.

- **Modify Existing TIGA Settings** allows you to:
  - Change TIGA system settings.
  - Update TIGA.

- **Exit to DOS** exits tigaset and returns to DOS.
Step 4: Use the arrow keys to select (Re)Install TIGA.

The screen shown in Figure 2–7 prompts for the host I/O base address set by the SDB20’s SW1. The highlight and check mark identify the current I/O base address: the factory preconfigured default is 0x230. See subsection 2.2.2.1, How to Reconfigure Your SDB20, on page 2-4, for details about reconfiguring SW1.

Figure 2–7. TIGA Installation Screen — Host I/O Base Address

Step 5: Do one of the following:

1) If you agree with the current selection, press 

2) If you want to select another option, use the or keys to make your selection and press 

Note:

The TIGA Board NOT AT THIS ADDRESS message will appear if you attempt to select an I/O base address different from that where the SDB20 is installed. Choose another address or use the address selected by tigaset.

The screen shown in Figure 2–8 prompts for an SDB20 host memory segment base address where all memory-access cycles will be directed. The highlight and check mark identify the current host memory base address: the default is D000 to D7FF.
Figure 2–8. TIGA Installation Screen — Host Memory Segment Base Address

Here are the HOST SEGMENT BASE ADDRESSes for the TIGA Board. The one highlighted was either used on a previous installation or it is a HOST memory area the program feels can be used.

Select a Host Segment Base Address

<table>
<thead>
<tr>
<th>Address Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>C800 to CFFF</td>
</tr>
<tr>
<td>D000 to D7FF</td>
</tr>
<tr>
<td>D800 to DFFF</td>
</tr>
<tr>
<td>E000 to E7FF</td>
</tr>
<tr>
<td>E800 to EFFF</td>
</tr>
</tbody>
</table>

<ESC> = Previous Screen    <ENTER> = Select

Step 6: Do one of the following:

1) If you agree with the current selection, press \[ \leftarrow \].

2) If you want to select another option, use the \[ \uparrow \] or \[ \downarrow \] key to make your selection and press \[ \leftarrow \].

VGA BIOS is typically located at C000–C7FF, and some PCs load shadow RAM at E000–EFFF. To avoid address conflict, determine your system's available addresses prior to selecting a different host memory segment address.

The screen shown in Figure 2–9 prompts for a data transfer width: 8-bit or 16-bit. The highlight and check mark identify the current data transfer size.
Step 7: Do one of the following:

- If the SDB20 is installed in an 8-bit slot, press $\rightarrow$. This will allow 8-bit transfers only.

- If the SDB20 is installed in a 16-bit slot, select either 8 or 16 and press $\rightarrow$. This allows either an 8-bit or a 16-bit transfer, respectively.

Note:

For proper operation, the transfer size selected for the SDB20 must be the same size as that of the system's EGA/VGA card, as well as any other memory-mapped device sharing the same 128K host memory page.

The screen shown in Figure 2–10 prompts for a directory where the TIGA software should be installed.
Installing and Running TIGA Setup Utilities

Figure 2–10. TIGA Installation Screen — Default Directory

**TIGA INSTALLATION**

TIGASET is ready to create the following directory in which to install TIGA

C:\TIGA

If you wish to install TIGA in a different location, use the <BACKSPACE> key to erase the name shown above and type in the complete path name of the drive and directory where you want to store TIGA. When you have completed entering the information, press <ENTER>.

<ESC> = Previous Screen  <ENTER> = Select

If you wish to install TIGA in a different location, follow the instructions on the screen to change the destination.

**Step 8:** Press → if you agree with the default location C: \ TIGA.

The software now asks for permission to append your AUTOEXEC.BAT file. NO TAG shows your choices.

Figure 2–11. TIGA Installation Screen—TIGA Environment

**TIGA INSTALLATION**

The following commands must be run to set up the TIGA environment. They are normally installed into the AUTOEXEC.BAT file.

\PATH C: \ TIGA; %PATH%
SET TIGA=-mC:\TIGA -lC:\TIGA -i0x60
TIGACD

Where should this information be placed?

<table>
<thead>
<tr>
<th>C:\TIGA\TIGAAUTO.BAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\AUTOEXEC.BAT</td>
</tr>
</tbody>
</table>

<ESC> = Previous Screen  <ENTER> = Select

These instructions execute a series of commands to your PC when you power up or reboot your system.
Installing and Running TIGA Setup Utilities

If you decide not to modify your AUTOEXEC.BAT file, tigaset will place these commands into a file called tigaauto.bat, which is located in your TIGA directory. You must manually input these lines into your AUTOEXEC.BAT file after the installation is completed.

Next, tigaset asks whether you are using a single monitor or dual monitor setup (Figure 2–12).

Figure 2–12. TIGA Installation Screen — Monitor Configuration

If you have separate monitors attached to the VGA and the TIGA board and you do not want VGA passthrough images to show on your secondary monitor, select the dual monitor option. Otherwise, choose the single monitor option.

Note:
The VGA passthrough cable must be connected between your VGA and the TIGA board if you are using a single monitor.

Step 9: Use the arrow keys and press <Enter> to select one of the following options:

- **Install as Single Monitor system** – if you do want VGA passthrough images to show on your TIGA board’s monitor.
- **Install as Dual Monitor system** – if you are using separate monitors attached to your VGA and the TIGA board, and you do not want VGA passthrough images to show on your TIGA board’s monitor.

You will be presented with a list of monitors (Figure 2–13).
Figure 2-13. TIGA Installation Screen — Defined Monitors

<table>
<thead>
<tr>
<th>Defined Monitors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mitsubishi Diamond Scan HL6605</td>
</tr>
<tr>
<td>SONY Multiscan CPD-1304</td>
</tr>
<tr>
<td>NEC Multisync JC-1401</td>
</tr>
<tr>
<td>NEC Multisync Plus JC-1501</td>
</tr>
<tr>
<td>NEC Multisync 3D JC-1404</td>
</tr>
<tr>
<td>NEC Multisync 4D JC-1601</td>
</tr>
<tr>
<td>NEC Multisync 5D JC-2002</td>
</tr>
</tbody>
</table>

<ESC> = Previous Screen       <ENTER> = Select

Step 10: Select the model number of the monitor that you are using. If the monitor you are using is not on this list, press → to accept the default selection.

Note:
See Section 2.5, Customizing Monitor Timing, for instructions on how to customize your monitor timing.

Figure 2-14. TIGA Installation Screen — Determining the Monitor Color

The installation program is designed to display several colors when run on a COLOR MONITOR/VGA system.

If you are running on a COLOR monitor and can only see TWO colors (for example, BLACK and WHITE) select the response YES.

<table>
<thead>
<tr>
<th>Is this screen monochrome when you expect color?</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
</tr>
<tr>
<td>NO</td>
</tr>
</tbody>
</table>

<ESC> = Previous Screen       <ENTER> = Select
Some VGA boards require that the monitor be connected to the VGA board at boot-up. If the monitor is not connected, the system may boot the VGA in the monochrome mode. This problem may occur in single monitor systems. Step 11 and Step 12 help you to determine whether this condition exists and provide a suggested remedy.

**Step 11:** Use the arrow keys and press ☐ to select one of the following options:

- **YES** — if the screen in Figure 2–15 is monochrome and you are expecting the screen to be in color.
- **NO** — if you are working on a monochrome system or your screen is already in color.

![TIGA Installation Screen — Correcting the Monochrome Display Problem](image)

**Step 12:** Select one of the following to remedy the monochrome display problem:

- **YES** — adds the `VGARESET` line to your `AUTOEXEC.BAT` file.
- **NO** — means that you will have to execute `VGARESET` to return your VGA to the color mode each time that you reboot your PC.

The `tigaset` utility presents you with a review menu.

- If you need to change any setting, return to the main menu and make your change.

- If not, select *Use Settings* and proceed to the screen shown in Figure 2–16.
This screen displays the number of colors, display pages, and resolutions supported by the TIGA board. A check mark identifies the current display mode. You can also access this screen from the DOS command line by entering `ti-gamode` at the prompt, and pressing ENTER.

**Figure 2-16. Typical Change TIGA Display Mode Screen**

TIGA MODE UTILITY Version x.x, Copyright (©) 1990 Texas Instruments, Inc

Available video modes: 15, Default Mode: 1024x768, 256 colors, 1 Page

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Colors</th>
<th>Pages</th>
<th>Pixel Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024x768</td>
<td>256</td>
<td>1</td>
<td>8</td>
<td>Palette</td>
</tr>
<tr>
<td>1024x768</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>Palette</td>
</tr>
<tr>
<td>640x480</td>
<td>256</td>
<td>1</td>
<td>8</td>
<td>Palette</td>
</tr>
<tr>
<td>640x480</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>Palette</td>
</tr>
<tr>
<td>1024x768</td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>Palette</td>
</tr>
<tr>
<td>640x480</td>
<td>256</td>
<td>2</td>
<td>8</td>
<td>Palette</td>
</tr>
<tr>
<td>640x480</td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>Palette</td>
</tr>
</tbody>
</table>

<↑>, <↓> = Move Highlight  <ESC> = Quit Display  <ENTER> = Select Mode

**Step 13:** Do one of the following:

- Use the arrow keys to select the desired display mode; press ENTER.

- If you are uncertain of the display mode, press ENTER to accept the current selection identified by the check mark.

The `tigaset` utility presents you with the SDB20 information screen (Figure 2-17). It lists configuration parameters: processor type, memory population, I/O address, and hardware interrupt.
Figure 2-17. TIGA Information Screen for the SDB20

![TIGA Installation Screen]

**Step 14:** Press →, which returns the main menu.

This completes TIGA installation for the SDB20.

**Step 15:** Do one the following before exiting the system:

1) If you directed `tigaset` to modify your `AUTOEXEC.BAT` file, remove the SDB20 installation diskette and press `CTRL` + `ALT` + `DELETE` simultaneously to reboot your system.

2) If you chose to manually append your `AUTOEXEC.BAT` file, select `Exit to DOS`. You will have to manually append the contents of `TIGAAUTO.BAT` to your `AUTOEXEC.BAT`, then press `CTRL` + `ALT` + `DELETE` simultaneously to reboot your system.

This concludes TIGA installation for the SDB20.
2.4 Setting Up a Dual Monitor System

You can use separate EGA or VGA monitors with the SDB20; a VGA monitor is preferred. In dual monitor systems, applications interact with DOS on the primary monitor; high-resolution graphics applications run on the secondary monitor.

Do the following to set up your system for dual monitor usage:

**Step 1:** Remove the cover from your PC by following the instructions contained in your PC’s operator manual.

**Step 2:** Locate an unused 8-bit or 16-bit bus slot.

**Step 3:** Install the SDB20 by executing Step 3 and Step 4 of subsection 2.2.3, Procedures for Installing the SDB20 into Your PC.

If you want VGA passthrough on the secondary monitor, install the VGA passthrough cable between the SDB20 and your VGA card.

**Step 4:** Connect the 15-pin monitor cables as follows:

- Connect the **primary monitor cable** to your primary EGA or VGA graphics display card connector.
- Connect the **secondary monitor cable** to the SDB20.

**Step 5:** Replace the PC cover.

**Step 6:** Write down your monitor’s manufacturer and model number in the box below. The SDB20 setup utility may require this information.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Model Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 7:** Turn on power to the PC.

**Step 8:** Run the *tigaset* utility by following the instructions in Section 2.3, *Installing and Running TIGA Setup Utilities*.

---

**Note:**

**Remember** that when the setup utility asks for monitor information, it is referring to the monitor connected to the SDB20.

The *tigaset* software will ask if you are using a single or dual monitor system.

- Selecting **DUAL MONITOR** leaves the previous TIGA image on the secondary monitor when you exit from TIGA applications.
- Selecting **SINGLE MONITOR** returns the VGA display to both your primary and secondary monitor.
2.5 Customizing Monitor Timing

After initial installation, you may decide to use a different monitor than the one configured when you installed your SDB20, or you may decide to use a monitor that is not listed. Whatever the reason, if it should be necessary to reconfigure the monitor timing, use the tigaset utilities to customize the timing for your monitor.

Typically, a monitor supports a set of horizontal and vertical timings for each standard resolution. The timing values are usually given as a frequency range rather than as a single frequency value.

A video image consists of successive frames. Each frame contains a large number of lines.

- The rate at which each line is presented to the screen is called **horizontal frequencies**.

- The rate at which the frames are presented is called the **vertical frequency**.

- **Sync widths**, and **back porch** timing allow the monitor to synchronize with the SDB20’s video output.

Figure 2–18 shows a typical video timing diagram.

Figure 2–18. Typical Video Timing Diagram
Complete these steps to customize the timing for your monitor.

**Step 1:** Check the timing specifications for your monitor in its operator manual, and record them in the blanks provided below.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal frequency (in kHz)</td>
<td></td>
</tr>
<tr>
<td>Horizontal sync width (in µs)</td>
<td></td>
</tr>
<tr>
<td>Horizontal back porch (in µs)</td>
<td></td>
</tr>
<tr>
<td>Vertical frequency (in Hz)</td>
<td></td>
</tr>
<tr>
<td>Vertical sync width (in ms)</td>
<td></td>
</tr>
<tr>
<td>Vertical back porch (in ms)</td>
<td></td>
</tr>
</tbody>
</table>

**Step 2:** Insert the *Installation Diskette* into your floppy drive A.

**Step 3:** Enter A: and press **ENTER**.

**Step 4:** Enter `tigaset` and press **ENTER**. This invokes the `tigaset` utility and displays the menu.

**Step 5:** Select *Modify Existing TIGA Settings*.

**Step 6:** Select *Change Monitor*.

The Change Monitor screen (see Figure 2–12 on page 2-18) displays and asks whether you are using a single monitor or dual monitor setup: choose either the single monitor or dual monitor system installation.

You are now presented with a list of monitors (see Figure 2–13 on page 2-19).

**Step 7:** Using the arrow keys, move the highlight and select *User Defined*, which is at the end of the list.

The screen shown in Figure 2–19 appears, supplying the default values for each entry that `tigaset` expects. The current resolution appears at the top of the screen.
A description of each key function follows:

- **F4** allows you to check monitor alignment by toggling between your TIGA display and the VGA display. The TIGA display is a full-page border with a crosshair at the center. By using **F4**, you can check your entries and see their effect on the alignment.

- **F2** toggles between the 640 x 480 and 1024 x 768 resolutions. Use **F2** to move to the next resolution after adjusting the timings for the current resolution.

- **F9** resets the present resolution to default values.

- **F10** saves the customized monitor timings and proceeds to the Change TIGA Display Mode screen.

- **ESC** aborts monitor customization and returns to the main menu.

- **+** increments the value highlighted.

- **-** decrements the value highlighted.

- **↑** moves the highlight up.

- **↓** moves the highlight down.
You can modify these values and customize them to meet your monitor's characteristics by doing the following:

**Step 8:** Use the arrow keys to position the highlight at the entry that you want to modify; enter the new value.

**Note:**

If your monitor operator's manual does not list all the required information, use the default values given on the screen. However, the more information that you provide, the better the result that *tigaset* will produce.

*tigaset* assumes that all 1024 × 768 displays with a horizontal frequency above 40 kHz are noninterlaced monitors.

- Some monitors support both interlaced and noninterlaced timings — in these cases, enter the values for noninterlaced timing.
- Some monitors can support only one set of video timing. For example, if your monitor does not support 640 × 480 displays, change all values under 640 × 480 to zero.

As you move from one display resolution to another, a different set of timing values appear.

**Step 9:** Repeat Step 8 for all resolutions.

**Step 10:** When you are satisfied with your selection, press [F10] to save your customized values.

**Step 11:** Perform Step 11 through Step 15 beginning on page 2-20.

**Note:**

To modify an existing set of monitor timings, the executable function *MTU.EXE* can be run from the TIGA directory as `<TIGA directory>\MTU [monitor filename]`. The screen shown in Figure 2–19 on page 2-26 appears. Run *MTU.EXE*, by performing Step 7 through Step 10, beginning on page 2-25.

**Step 12:** Rerun *tigaset* and select the altered monitor file.

**Step 13:** Allow *tigaset* to update itself; then reload TIGACD.
### 2.6 TIGA Software Drivers

The SDB20 uses TIGA software to provide compatibility with many applications. A TIGA Windows driver is supplied with the SDB20. This driver allows Windows users to realize the benefits of the TMS34020 graphics processor.

#### 2.6.1 TIGA Video Drivers

Table 2-4 lists the applications that work through TIGA and subsequently run on the SDB20. It also provides instructions for obtaining the drivers.

**Table 2-4. TIGA Video Drivers**

<table>
<thead>
<tr>
<th>Application</th>
<th>Where to Obtain Your TIGA Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anvil 1000/5000</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>AutoCAD</td>
<td>Call Panacea at (603) 437-5022</td>
</tr>
<tr>
<td>AutoShade/AutoSketch</td>
<td>Call Panacea at (603) 437-5022</td>
</tr>
<tr>
<td>CADKEY V3.5</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>CADKEY Render</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>DataCAD</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>DataCAD Velocity</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>DesignCAD 2D &amp; 3D</td>
<td>Call (918) 825-4844</td>
</tr>
<tr>
<td>Drawbase</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>GEM Artline</td>
<td>Call (800) 443-4200</td>
</tr>
<tr>
<td>GEM Presentation Team</td>
<td>Call (800) 443-4200</td>
</tr>
<tr>
<td>Digital Research Draw Plus</td>
<td>Call (800) 443-4200</td>
</tr>
<tr>
<td>Generic CADD</td>
<td>Contact your Generic CADD dealer</td>
</tr>
<tr>
<td>GSPOT I, II, III (TIGA debugger)</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>HALO V3.0</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>HOOPS</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>MasterCAM</td>
<td>Contact your MasterCAM dealer</td>
</tr>
<tr>
<td>Microstation V3.3</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>Ships with the SDB20</td>
</tr>
<tr>
<td>OrCAD/SDT III, PCB II, VST</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>Personal Designer 4.1</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>Point Line CADD</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>RoboCAD 2.0</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>Ventura Publisher/GEM</td>
<td>Works with the GEM TIGA driver</td>
</tr>
<tr>
<td>X-Windows Server</td>
<td>Ships with the application</td>
</tr>
<tr>
<td>XSoftware TIGA</td>
<td>Ships with the application</td>
</tr>
</tbody>
</table>
2.6.2 Installing the TIGA Windows Driver

Whether you are installing Windows for the first time or upgrading your VGA to the SDB20, install the TIGA Windows driver as outlined in the following paragraphs. However, if a TIGA Windows driver is provided with your Microsoft Windows software, install the driver by using the typical Windows installation procedure. In this case, **do not install** the TIGA Windows driver provided with your TIGA board.

Before proceeding further, make sure that TIGA communications have begun. When the `tigaset` installation utility was running, you chose between having `tigaset` alter your `AUTOEXEC.BAT` file or having it create a file called `TIGAAUTO.BAT`.

- If you chose to have your `AUTOEXEC.BAT` file edited and you rebooted your PC after installation was complete, the communications are working.
- If you chose to create the `TIGAAUTO.BAT` file, execute the file from your TIGA directory by entering `TIGAAUTO` and pressing `Enter`.

**Step 1:** At the prompt, change directories to your `\windows` directory.

You cannot install the TIGA Windows driver from within Windows. Exit Windows and use the DOS command line to execute the Windows **setup** utility.

**Step 2:** Enter `SETUP` and press `Enter` to execute the Windows setup. You are eventually presented with the Windows **Setup** screen shown in Figure 2–20.
Step 3: Position the highlight on Display: and press ✪. You are presented with a list of drivers as shown in Figure 2–21.
Figure 2–21. Windows Setup Screen — Display

You have asked to change the type of Display to be installed. Select a Display from the following list:

- 8514/a
- CGA
- Compaq Portable Plasma
- EGA
- Hercules Monochrome
- Olivetti/AT&T Monochrome or PVC Display

(To see more of the list, press the <↓> arrow key)

Press the UP or DOWN ARROW key to move the highlight to the item you want to select. Then press ENTER.

If you want to return to the System Information screen without changing your Display type, press ESC.

ENTER = Continue    F1 = Help    F3 = Exit    ESC = Cancel

Step 4:  At the end of the screen, position the highlight on the OTHER option and press [→].

Windows will prompt you for a directory where the new TIGA Windows driver is located (see Figure 2–22).
Figure 2–22. Windows Setup Screen—Display Driver

Step 5: Use BACK SPACE to erase the A:\ prompt.

Step 6: Enter <TIGA directory>\WIN30 and press 9.

    <tiga directory> is the pathname where TIGA was installed in Step 8 on page 2-17.

The TIGA Windows driver presents three variations.

Step 7: Choose the appropriate driver for your current TIGA display mode and press 9. Follow the instructions until Windows returns to the SYSTEM INFORMATION screen.

Note:

You may see only one option of the TIGA Windows driver, depending upon future revisions of the TIGA Windows driver.

Step 8: Press 9 to exit SETUP.

This completes the TIGA Window driver installation.
Chapter 3

Theory of Operation

This chapter provides a detailed description of the SDB20's hardware design. It should be used as a technical reference by hardware developers who design TMS34020-based video adapters. Also, the programmable logic array ABEL source file equations shown in this chapter are revision level B.

Topics discussed include:

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Host Interface</td>
<td>3-2</td>
</tr>
<tr>
<td>3.2 The Frame Buffer/Program Memory Interface</td>
<td>3-45</td>
</tr>
<tr>
<td>3.3 The TMS34082 Floating-Point Coprocessor</td>
<td>3-53</td>
</tr>
<tr>
<td>3.4 Backend Section</td>
<td>3-58</td>
</tr>
<tr>
<td>3.5 The TMS34020-to-XDS500 Interface</td>
<td>3-62</td>
</tr>
</tbody>
</table>
3.1 Host Interface

This section describes the hardware interface between the host ISA standard PC bus and the SDB20. Figure 3-1 shows the layout of the SDB20.

The SDB20 installs in either an 8-bit or 16-bit slot. It is software selectable for either 8-bit or 16-bit memory transfers; it is limited to 8-bit I/O memory transfers.

3.1.1 Host I/O Decoding

Host I/O mapping of the SDB20 is done through the SDB20's switchbank SIP (SW1). The available range for the SDB20 is 200h–2F0h. For more information, see subsection 2.2.2.1, How to Reconfigure Your SDB20.

Once an address has been selected, the SN74F85 4-bit I/O address comparator (U12) compares the host-supplied I/O local address LOCADD[4:7] to the SW1 switch settings. Upon receipt of a valid I/O address, its IOSEL output is brought high.

Note:

IOSEL is actually an active-high signal.

The SELECT PAL (U40) uses IOSEL and I/O local address bits LOCADD[8:9], along with a host-initiated I/O read (IOR) or I/O write (IOW), to decode a valid SDB20 I/O address. The LOCADD[8:9] bits must be 0 and 1, respectively, forcing the SDB20 into the 2xxh I/O address range.

The SELECT PAL (U40) outputs the IOENB signal when a valid SDB20 I/O address has been asserted. This signal is used to enable several other I/O-related SDB20 functions, such as the MAPEN (U27), GSPHST (U26), and HBS (U28) PALs.
Figure 3-1. The SDB20 Software Development Board
3.1.2 SDB20 Host Communication Registers

The host PC communicates with the TMS34020 device by means of seven 8-bit I/O registers that reside within the host PC I/O space and one 8-bit register that resides within the TMS34020 memory space. The registers are:

- The BASEP base pointer register (see subsection 3.1.2.1),
- The HINTF host interface control register (see subsection 3.1.2.2),
- The MAPAL mapping register A (LSByte) (see subsection 3.1.2.3),
- The MAPAM mapping register A (MSByte) (see subsection 3.1.2.3),
- The MAPBL mapping register B (LSByte) (see subsection 3.1.2.3),
- The MAPBM mapping register B (MSByte) (see subsection 3.1.2.3),
- The SDBST SDB status register (see subsection 3.1.2.4), and
- The HSREG hardware space control register resides in the local-memory space of the TMS34020 (see subsection 3.1.2.5).

Table 3-1 shows the communication registers of the SDB20 and their functionality. Table 3-2 shows the memory mapping for each.

Table 3-1. The SDB20 Communication Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Function</th>
<th>Register</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BASEP</strong></td>
<td>Base pointer</td>
<td><strong>MAPBL</strong></td>
<td>Mapping register B (LSB)</td>
</tr>
<tr>
<td><strong>HINTF</strong></td>
<td>Host interface control</td>
<td><strong>MAPBM</strong></td>
<td>Mapping register B (MSB)</td>
</tr>
<tr>
<td><strong>MAPAL</strong></td>
<td>Mapping register A (LSB)</td>
<td><strong>SDBST</strong></td>
<td>SDB status</td>
</tr>
<tr>
<td><strong>MAPAM</strong></td>
<td>Mapping register A (MSB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HSREG</strong></td>
<td>Hardware space control</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TMS34020 Memory Space Register**

<table>
<thead>
<tr>
<th>Register</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HSREG</strong></td>
<td>Hardware space control</td>
</tr>
</tbody>
</table>
Table 3-2. SDB20 Communication Register Memory Map

<table>
<thead>
<tr>
<th>PC Bus I/O Space</th>
<th>Address</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>02P0</td>
<td>Mapping register A, LSB (MAPAL)</td>
<td>R/W : 8</td>
<td></td>
</tr>
<tr>
<td>02P1</td>
<td>Mapping register A, MSB (MAPAM)</td>
<td>R/W : 8</td>
<td></td>
</tr>
<tr>
<td>02P2</td>
<td>Mapping register B, LSB (MAPBL)</td>
<td>R/W : 8</td>
<td></td>
</tr>
<tr>
<td>02P3</td>
<td>Mapping register B, MSB (MAPBM)</td>
<td>R/W : 8</td>
<td></td>
</tr>
<tr>
<td>02P4</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02P5</td>
<td>SDB status (SDBST)</td>
<td>R : 8</td>
<td></td>
</tr>
<tr>
<td>02P6</td>
<td>Base page pointer (BASEP)</td>
<td>R/W : 8</td>
<td></td>
</tr>
<tr>
<td>02P7</td>
<td>Host interface control register (HINTF)</td>
<td>R/W : 8</td>
<td></td>
</tr>
<tr>
<td>02P8</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02P9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02PA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02PB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02PC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02PD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02PE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02PF</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TMS34020 Local Memory Space

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0000000</td>
<td>Hardware space control register (HSREG)</td>
<td>W : 8</td>
</tr>
</tbody>
</table>

3.1.2.1 The Base Pointer Register (BASEP), U6

The base pointer register is an SN74ALS996 device that maps the SDB20 into a software selectable 64K-byte memory segment of the PC as follows:

- When placed into a 16-bit slot, BASEP selects a 64K-byte memory segment of the PC's 16-MByte memory map.
- When placed into an 8-bit slot, BASEP selects a 64K-byte memory segment of the PC's 1-MByte memory map.

When the SDB20 is installed in an 8-bit slot, the upper four bits of the PC's address bus (PCADD[20:23]) are not present. In their absence, the SDB20 pulls these bits high.

Note:
The PC address bus PCADD[0:23] is buffered by the host address latch (U54, U64, and U59), which generates the local address bus LOCADD[0:23].
This allows the SN74ALS521 base address comparator (U39) to correctly select the SDB20 when bits 0–3 of the programmed BASEP register compare to the received values of the PC’s host address, PCADD[16:19]. This means that the MSBs of the BASEP register must be programmed with logical 1s.

### 3.1.2.2 The Host interface Control Register (HINTF), U7

The HINTF register is an SN74ALS996 device that describes how the SDB20 responds to PC bus cycles. Bits 1–3 are reserved for future use and should be written with zeros to maintain compatibility with future versions of the SDB20. Figure 3–2 shows bit assignments.

#### Figure 3–2. The HINTF Register

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDBENB</td>
<td>16BITEN</td>
<td>SHADIS</td>
<td>HRST</td>
<td>RESERVED</td>
<td>BASE32K</td>
<td></td>
</tr>
</tbody>
</table>

An explanation of the HINTF register’s bit assignments follow:

- **BASE32K** (bit 0) is considered to be an extension of the BASEP register.
  - If **BASE32K = 0** (the default on power up and reset), it selects the lower 32K-byte block of the selected 64K-byte segment.
  - If **BASE32K = 1**, it selects the upper 32K-byte block of the selected 64K-byte segment.

  The SELECT PAL (U40) uses this signal to compare the value of BASE32K to the PC host address bit (PCADD[15]).

- **Reserved** (bits 1–3).

- **HRST** (bit 4) is the host reset bit.
  - When **HRST = 0** (the default on power-up and reset), the host reset is disabled.
  - When **HRST = 1**, the TMS34020 and the optional TMS34082 (when installed) are reset. Also, the HSREG and HINTF registers are reset to their default values.

  The HRST signal is also known as the HRS signal. The RESET PAL (U15) uses the HRST signal to generate the LRESET signal, which is the SDB20’s local reset signal. For more information, see subsection 3.1.10, Resetting the SDB20.

- **SHADIS** (bit 5) is the shadow palette access disable bit.
  - When **SHADIS = 0** (the default on power-up and reset), the SDB20 duplicates writes to the VGA palette and copies these to the SDB20 palette.
  - When **SHADIS = 1**, the SDB20’s host control logic disables SHADOW palette accesses.
The SHADIS bit relates to the SHADENB signal on the SDB20 schematic set (see Appendix C). This signal is used by the MISC PAL (U8), along with the SHADOW signal generated by the SHADOW PAL (U25), to generate the shadow active flag SHADOK signal. The SELECT device (U40) uses SHADOK to generate the PALDIR palette direct access enable signal.

1. **16BITEN** (bit 6) is the 16-bit transfer enable for host's transfers to the TMS34020's local memory.
   - When **16BITEN=0** (the default on power-up and reset), the SDB20 permits only 8-bit host accesses to the TMS34020's local memory.
   - When **16BITEN=1**, the SDB20 permits either 8-bit or 16-bit host accesses to TMS34020 local memory.

The GSPHST PAL (U26) uses the 16BITEN signal and the 16BSEL signal from the 16-bit transfer comparator (U30) to generate the active-low 16BTTR signal. This signal is buffered as M16 and is sent to the PC's 16-bit memory chip-select (MEMCS16) to indicate to the host that the SDB20 is capable of supporting 16-bit transfers.

The HBS PAL (U28) also uses 16BTTR as part of the decode for the host byte strobes HBS[0:3] and the byte selects (BYTSEL[0:3]) for the SN74ALS652 external data transceivers (U41, U42, U69, U70) host write and read cycles, respectively.

2. **SDBENB** (bit 7) is the SDB20 enable bit.
   - When **SDBENB=0** (the default on power-up and reset), the SDB20 cannot respond to any host memory accesses.
   - When **SDBENB=1**, the SDB20 can respond to host memory accesses.

The SELECT PAL (U40) uses the SDBENB bit, as well as a valid SHADOW VGA cycle or memory cycle, to generate the HCS host chip select signal. The HCS signal is used:

- To select the TMS34020.
- By the GSPHST PAL (U26) to generate the host read (HREAD) and host write (HWRITE) signals to initiate host read/write cycles.
- By the HBS PAL (U28) to enable the byte selects (BYTSEL[0:3]).

The SDBENB signal is inverted by the HAND PAL (U14) to generate the 16XF signal. This signal enables the SN74ALS521 16-bit transfer comparator (U30).
3.1.2.3 The MAPAL, MAPAM, MAPBL, and MAPBM Mapping Registers, U2–U4

The SDB20 has two sets of 15-bit I/O address mapping registers:

- An 8-bit MAPAM (U4) or MAPBM (U2) mapping register, and
- A 7-bit MAPAL (U5) MAPBL (U3) mapping register.

The MAP A and MAP B register sets are SN74ALS996 read-back latches that are used as pointers to 16K-byte sections of TMS34020 memory space. The memory locations pointed to by these registers can be contiguous, noncontiguous, or overlapping.

The GSPHST PAL (U26) decodes PCADD[14] of the PC’s address bus and determines which set of mapping registers (MAP A or MAP B) will be selected. This implies that

- When PCADD[14] is cleared, the lower 16K-byte of the 32K-byte block of host memory is selected, which corresponds to the MAP A register set.
- When PCADD[14] is set, the upper 16K-bytes of the 32K-byte block of host memory is selected, which corresponds to the MAP B register set.

For more information, refer to subsection 3.1.3, Address Conversion Principles.

Table 3–3 shows how the mapping registers are constructed.

### Table 3–3. The MAPA and MAPB Mapping Registers

<table>
<thead>
<tr>
<th>Bits</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPAL</td>
<td>add6</td>
<td>add5</td>
<td>add4</td>
<td>add3</td>
<td>add2</td>
<td>add1</td>
<td>add0</td>
<td>X</td>
</tr>
<tr>
<td>MAPAM</td>
<td>add14</td>
<td>add13</td>
<td>add12</td>
<td>add11</td>
<td>add10</td>
<td>add9</td>
<td>add8</td>
<td>add7</td>
</tr>
<tr>
<td>MAPBL</td>
<td>add6</td>
<td>add5</td>
<td>add4</td>
<td>add3</td>
<td>add2</td>
<td>add1</td>
<td>add0</td>
<td>X</td>
</tr>
<tr>
<td>MAPBM</td>
<td>add14</td>
<td>add13</td>
<td>add12</td>
<td>add11</td>
<td>add10</td>
<td>add9</td>
<td>add8</td>
<td>add7</td>
</tr>
</tbody>
</table>

Notes: These registers are undefined at power-up.

The GSPHST PAL (U26) generates the OEMAPA and the OEMAPB mapping register output enable signals. The OEMAPA signal is asserted when PCADD[14] is cleared. Similarly, the OEMAPB signal is asserted when PCADD[14] is set. These signals are used by the MAP A and MAP B mapping register sets.

During a host access cycle, the value programmed in the selected mapping register set (MAP A or MAP B) is placed onto the host address bus HA[17:31] of the TMS34020. The MSB of the mapping register set is connected to HA[31] and the LSB to HA[17].
3.1.2.4 The Status Register (SDBST), U29

The SDBST register is enabled by the HSRBK signal, which is generated by the MAPEN PAL (U27). The MAPEN PAL decodes the four LSBs of the PC's I/O address.

The SDB20 status register is a SN74ALS240 device that provides a read-back of:

- The contents of the HSREG register's four LSBs, and
- The current state of the TMS34020's HINT host interrupt pin. The status of the host interrupt pin is used to improve the performance of polled-interrupt systems. When HINT=1, the TMS34020 has asserted its host interrupt pin.

The SDBST register is a read-only register; do not write to it.

Figure 3–3 shows the SDBST register's bit assignments, and Table 3–4 describes its bit fields.

Figure 3–3. The SDB20 Status Register

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>HINT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>VGA</td>
<td>RATE</td>
<td>4BIT</td>
<td>8BIT</td>
</tr>
</tbody>
</table>

Table 3–4. SDBST Bit Field Descriptions

<table>
<thead>
<tr>
<th>Bit Field</th>
<th>Description</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>HINT</td>
<td>A logic 1 denotes that the TMS34020 has asserted its host interrupt pin.</td>
<td>0</td>
</tr>
<tr>
<td>VGA</td>
<td>A logic 1 denotes that the VGA mode is active.</td>
<td>1</td>
</tr>
<tr>
<td>RATE</td>
<td>A logic 1 denotes the 640 x 480 resolution mode. A logic 0 denotes the 1024 x 768 resolution mode.</td>
<td>1</td>
</tr>
<tr>
<td>4BIT</td>
<td>A logic 1 denotes the 4-bit/pixel mode.</td>
<td>0</td>
</tr>
<tr>
<td>8BIT</td>
<td>A logic 1 denotes the 8-bit/pixel mode.</td>
<td>0</td>
</tr>
</tbody>
</table>
3.1.2.5 **The Hardware Space Control Register (HSREG)**

The HSREG is actually the HARD PAL (U13), which is a TIBPAL16L8 device that acts as a 32-bit addressable 8-bit register. It is located at 0E000000h in the local-memory space of the TMS34020. The HSREG sets the video mode and resolution of the SDB20. Upon power-up or reset, the SDB20 defaults to the VGA mode with a 640 x 480 screen resolution.

Figure 3–4 shows the construction of the HSREG.

![Figure 3–4. Hardware Space Control Register (HSREG)](image)

An explanation of the HSREG bit fields follows:

- **8BIT** (bit 0) is the 8-bits/pixel screen resolution mode select. The related signal is PIX8BIT. The default is 0.
  - When 8BIT=0, the 8-bit/pixel output buffer (U57) is disabled.
  - When 8BIT=1, it sets the 8-bit mode by enabling the 8-bit/pixel output buffer to pass 8-bit/pixel data from the 8-bit/pixel shift registers (U44,U50,U52, U53), to the 8-bit output buffer (U57) pixel data PIXEL[0:7], palette dot clock (PALDOT) and palette blanking (PBLANK) signals.

- **4BIT** (bit 1) is the 4-bits/pixel screen resolution mode select. The related signal is PIX4BIT. The default is 0.
  - When 4BIT=0, the 4-bit/pixel output buffer (U60) is disabled.
  - When 4BIT=1, it sets the 4-bit/pixel mode by enabling the 4-bit/pixel output buffer to pass 4-bit/pixel data from the 4-bit/pixel shift registers (U46–U49). The 4-bit output buffer (U60) buffers the pixel data PIXEL[0:7], dot clock PALDOT, and blanking PBLANK signals.

- **RATE** (bit 2) is the dot clock rate selector that is used by the clock generation control logic. The default is 1.
  - When RATE=1, the SDB20 provides the backend logic a 25-MHz dot clock frequency to support a 640 × 480 screen resolution.
  - When RATE=0, the SDB20 provides the backend logic a 60-MHz dot clock frequency to support a 1024 × 768 screen resolution.

The related signal is PIXRATE. The SN74AS153 4-to-1 dot clock multiplexer (U66) uses PIXRATE to set the dot clock frequency that will be used by the Bt478 color palette.
**VGA** (bit 3) is the VGA passthrough enable signal. The related signal is called VGAENB. The default is 1.

- When VGA=1, the SDB20 enables the VGA passthrough mode and clears the 4BIT and 8BIT LSBs of HSREG.
- When VGA=0, the SDB20 disables the VGA passthrough capability.

The VGA passthrough enable signal enables the VGA SN74F861 passthrough buffers (U55), which passes the VGA-generated sync pulses, dot clock, blanking signal, and pixel data to the Bt478 color palette.

The HARD PAL (U13) prevents improper register programming by clearing the HSREG. An example of improper programming would be setting simultaneously the 4BIT or 8BIT fields to a logical 1.

The HSREG should be accessed only in a byte-aligned fashion and a single-byte-only transfer from the host.

**Do not perform a word transfer to HSREG if you are in the 16-bit transfer mode. A word transfer using two-byte transfers will result in an undefined value in the HSREG.**

When changing the SDB20's resolution, clear the HSREG register and write a new value to it. The HSREG is organized as an 8-bit write-only register that responds to 32-bit accesses.

The host memory cycle generates the local address bus DAT[28:31] of the TMS34020. The RASDCD PAL (U1) decodes them and generates the HWSPACE hardware space enable signal. This signal is used to enable writes to the HSREG.

When a host memory write cycle occurs, the TMS34020-generated WE signal is logically ANDed with the HWSPACE signal to assert the PAL internal WRITE_CYCLE signal, which enables the HSREG.

**Note:**

The local reset (LRESET) signal generated by the RESET PAL (U15) resets the HARD PAL (U13) to its default state.

Figure 3–5 shows the ABEL source file for programming the HARD PAL.
3.1.3 Address Conversion Principles

The TMS34020 has a 32-bit local address bus LAD[0:31], which relates to DAT[0:31], that can access four gigabits (512 MBytes) of memory. The TMS34020’s host address bus (HA[5:31]) consists of 27 bits.

The address conversion from the host address bus to the TMS34020’s address bus is not a one-to-one conversion because

- The PC’s host processor is byte-addressable, which means that the smallest addressable unit of memory is a byte.

- The TMS34020 is bit-addressable, which means that the smallest addressable unit of memory is a bit.
The address used by the host processor to access a given TMS34020 local memory location depends on several factors:

- The contents of the BASEP base pointer register,
- The state of the BASE32K bit (bit 0) of the HINTF register,
- The contents of the mapping register sets (MAP A or MAP B),
- The mapping register set selected by the host PC address PCADD[14], and
- The address provided on the PCADD[2:13] host PC bus.

### 3.1.3.1 Host-to-TMS34020 Address Conversion

The value programmed into the BASEP register depends upon the PC slot in which the SDB20 is installed: an 8-bit or 16-bit PC slot.

- When the SDB20 is installed in an 8-bit slot, the upper four address bits (PCADD[20:23]) of the PC bus are not present. Consequently, the floating-address bits are pulled up to all 1s on the SDB20. Thus, bits 4–7 (the MSBs) of the BASEP register must be programmed to all 1s. If you use a host segment base address of 0D000h, the BASEP would be programmed with 0FDh.

- When the SDB20 is installed in a 16-bit slot, the BASEP would be programmed with 0Dh.

For the purposes of this discussion, assume that the SDB20 is installed in a 16-bit slot.

Figure 3–6 shows how the PC-generated host address 0D8034h is converted to access the TMS34020 local address 0C0001AOh, which corresponds to one of the TMS34020 I/O registers. Host address 0D8034h denotes that the upper 16K bytes of the upper 32K-byte block of the 64K-byte segment 0D0000h is selected.

#### Note:

The LSB of the MAPAL register is not used. Consequently, it is considered to be 00h rather than 01h.

During a PC bus cycle, the SDB20 logic compares PCADD[16:23] of the PC bus address to the contents of the BASEP register. The SDB20 then compares PCADD[15] to the BASE32K bit (bit 0) in the HINTF register (U7). If both comparisons are true, the SDB20 is selected.

Host addresses PCADD[0] and PCADD[1] determine the host byte selects, which determine byte/word alignment. Because the SDB20 is installed in a 16-bit slot, and PCADD[0,1] are set to logic level 0 in this example, the host will access the least significant word (LSW)—in this case, the CONFIG register of the TMS34020.
Once selected, the SDB20 uses PCADD[14] to enable one of the two sets of mapping registers—MAPA or MAPB. Each mapping register is 15 bits wide.

- If PCADD[14] = 0, the MAP A register set is selected.
- If PCADD[14] = 1, the MAP B register set is selected.

For purposes of this discussion, the MAPA mapping register has been selected because PCADD[14] is a logical 0. It is composed of the MAPAM (MAP A, MSByte) and MAPAL (MAP A, LSByte less one bit) registers. The output of the mapping register is tied to the PC host address HA[17:31] pins.

Figure 3–6. Converting a PC Host Address to a TMS34020 Local Address
The address presented to the TMS34020 is a combination of the outputs from:

- The MAPAM and MAPAL mapping register set, which are connected directly to the HA[17:31] pins.
- The PCADD[2:13] address pins, which are connected directly to the host address pins HA[5:16].
- The TMS34020's host address bus HA[5:31] maps directly to its local address bus LAD[5:31].

The SDB20 select logic has now mapped a 16K-byte memory segment of PC bus memory space:

- The BASEP register selecting a 64K-byte segment,
- The BASE32K (bit 6) of the HINTF register selecting the upper 32K-byte block of the 64K-byte segment, and
- PCADD[14] selecting the upper half of the 32K-byte block.

### 3.1.3.2 TMS34020-to-Host Address Conversion

For the purposes of this discussion, assume that the SDB20 is installed in an 8-bit slot. When the SDB20 is installed in an 8-bit slot, the upper 4 bits (bits 7–4) of the BASEP register are pulled up to all 1s.

Figure 3–7 shows how TMS34020 local memory address 0E000000h, which corresponds to the SDB20's HSREG register, is derived from the following:

- The PC host address 0D4000h, which corresponds to BASEP set to 0FDh,
- The BASE32K (bit 6) of the HINTF register is set to 0,
- PCADD[14] set to select the lower half of the 32K-byte block.
- The MAPBM is set to 0E0h, and
- The MAPBL is set to 00h.
During a PC bus cycle, the SDB20 logic compares the upper eight bits of the PC address bus (PCADD[16:31]) to the contents of the BASEP register.

Because the SDB20 is installed in an 8-bit slot, the BASEP register is programmed with the value 0FDh, the MSNibble of the BASEP register must be set to logical 1s in order to be compatible with the pull-ups on the SDB20's upper PC address nibble. The SDB20 then compares PCADD[15] to the BASE32K bit (bit 0) of the HINTF register. The BASE32K bit is be set to 0 in order to select the lower 32K-byte block of the 64K-byte page memory segment. The SDB20 is selected when both of these comparisons are true.

Once selected, the SDB20 uses PCADD14 = 1 to enable the MAPBM and MAPBL mapping register set. They correspond to the upper 16K-bytes of the lower 32K-byte block, 0D4000h, of the 64K-byte page memory segment, 0D0000h.
The programmed value of the MAPBL and MAPBM mapping registers corresponds to TMS34020 local address bits LAD[17:31]. This means that MAPBM must be set to 0E0h and MAPBL must be set to 00h or 01h because the LSB of MAPBL is not used.

Host addresses PCADD[0,1] determine the host byte selects, which determine byte/word alignment. Because the SDB20 is installed in a 8-bit slot, the host accesses the LS Byte of the HSREG by setting PCADD[0,1] to logical 0s.

Note:
Always access the LSByte of the HSREG, which is a 32-bit addressable 8-bit-wide memory location.

The address presented to the TMS34020 is a combination of the outputs from:

- The MAPAM and MAPAL mapping register set, which are connected directly to the PC host HA[17:31] address pins.
- The PCADD[2:13] address pins, which are connected directly to the PC host HA[5:16] address pins.
- The TMS34020's host address bus HA[5:31] maps directly to its local address bus LAD[5:31].

The SDB20 select logic has now mapped a 16K-byte memory segment of PC bus memory space:

- The BASEP register selecting a 64K-byte block,
- The BASE32K of the HINTF register selecting the lower 32K-byte block of the 64K-byte segment, and
- The PCADD[14] selecting the lower half of the 32K-byte block.

This results in a PC host address of 0FD4000h. The host segment base address is 0D000h.
3.1.4 Host Interface Control PALs

The SDB20 contains five PALs that are used to control the interface between the SDB20 and the host PC. They are:

- The GSPHST TMS34020 host interface logic (U26) (see page 3-18),
- The HBS host byte-select decode logic (U28) (see page 3-20),
- The MAPEN map register control logic (U27) (see page 3-25), and
- The SELECT board select logic (U40) (see page 3-23).

3.1.4.1 The GSPHST PAL, U26

The TMS34020-to-host interface PAL (U26) is a TIBPAL22V10 device that provides synchronous communication between the PC host and the TMS34020 host.

On memory access cycles, the GSPHST controls the ISA input signal BUSRDY, which is defined as NOTREADY in the GSPHST PAL equation in Figure 3-8. Driving this signal low inserts wait states into the ISA memory cycle. Note that the SDB20 holds the BUSRDY line high for all ISA I/O Cycles.

The GSPHST PAL also generates the host read HREAD and host write HWRITE strobes for the TMS34020's host interface. When the HCS host chip select is asserted, one of these signals is asserted.

The READ signal that is generated by the SELECT PAL (U40) determines which strobe is selected.
- If READ is low, a host read cycle is initiated.
- If READ is high, a host write cycle is initiated.

The TMS34020 responds to the initiated cycle by asserting the HRDY signal. The GSPHST PAL asserts the LSTATE signal, which deasserts the NOTREADY signal and ends the ISA wait-state insertion. LSTATE stays asserted until HCS is deasserted.

During an ISA memory write cycle, the GSPHST device generates the HSTWCLK signal to clock data into the four SN74ALS652 external data transceivers (U69, U41, U70, U42).

The GSPHST PAL also generates a 16-bit transfer flag (16BTTR) when:
- The ISA bus asserts the SBHE signal,
- The 16BITENB bit of the HINTF register is set to a logic 1, and
- The SN74ALS521 (U30) 16-bit transfer comparator's output 16BSEL signal goes active low.

The 16SEL signal is asserted when the programmed base address of the BASEP register (U6) compares to the upper address lines EXPADD[0:2] and PCADD[20:23] of the ISA bus. These host address lines are buffered by a SN74ALS244 octal buffer (U43) as 16BADD[0:6]. The SDBENB signal of the HINTF register is buffered by the HAND PAL (U14) as 16XF, which is tied to the 16-bit transfer comparator's (U30) output enable.
On a host I/O cycle, the GSPHST generates the output enables for each of the SDB20's mapping register sets. There are two SN74ALS996 read-back latches for each set. The state of the buffered host address bit (LOCADD[14]) is used to assert either O̅EMA PA or O̅E M A PB when the host I/O cycle flag (IOENB), generated by the SELECT PAL (U40), is asserted.

- When LOCADD[14] = 0, the output enable (O̅E MAPA) for the MAP A mapping register set is asserted.
- When LOCADD[14] = 1, the output enable (O̅E MAPB) for the MAP B mapping register set is asserted.

Figure 3–8 shows the ABEL source file for programming the GSPHST PAL.

**Figure 3–8. ABEL Source File for the GSPHST PAL, U26**

```
GSPHST device 'P22V10';

"Inputs"

bclk  pin 1 ;
locadd14 pin 13 ;
!hcs   pin 2 ;
!read  pin 3 ;
hrdy  pin 4 ;
!hoe   pin 5 ;
!ioenb pin 6 ;
!sbhe  pin 7 ;
_16bitenb pin 8 ;
!_16bsel pin 9 ;
!memsel pin 10 ;
hdst  pin 11 ;
paldir pin 14 ;

"Outputs"

!hwrite pin 23 ;
!hread  pin 22 ;
!oemapa pin 21 ;
!oemapb pin 20 ;
!hstwclk pin 19 ;
!_16btr pin 17 ;
!notready pin 16 ;
lstate pin 15 ;

"Declarations and Intermediate Variable Definitions"

hwrite ISTYPE 'neg,reg_d,feed_reg';
hread ISTYPE 'neg,reg_d,feed_reg';
notready ISTYPE 'neg,reg_d,feed_reg';
hstwclk ISTYPE 'neg,com,feed_pin';

EQUATIONS

hwrite := hcs & !read & !lstate;
hread := hcs & read & !lstate;
oemapa = !locadd14 & !ioenb;
oemapb = locadd14 & !ioenb;
lstate = (hcs & (hwrite # hread) & hrdy) # (lstate & hcs);
hstwclk = (!lstate & !read & !hstwclk & hwrite) # (hstwclk & !lstate & !hoe));
notready = (hcs & !lstate) # (notready & paldir & !hoe);
_16btr  = sbhe & _16bitenb & _16bsel;

END _GSPHST
```
3.1.4.2 The HBS PAL, U28

The HBS host byte-select decode PAL (U28) is a TIBPAL22V10 that generates the host-byte strobe decoding for the TMS34020 on all ISA memory write cycles and the host data-latch byte-select decoding for all ISA memory read cycles.

The host-byte strobes HBS[0:3] are generated by the following:

- The two LSBs of the latched host address bus (LOCADD[0,1]),
- The 16BTTR signal from the GSPHST PAL (U26), and
- The host PC ISA system bus high enable SBHE signal.

These signals allow the TMS34020 to determine where to write a byte or word to its local memory. These strobes are latched along with the converted host address when HCS is asserted.

When the SELECT PAL (U40) asserts the HCS and READ signals, the TMS34020 LOCADD[0,1] address bits decode the byte-select signals (BYTSEL[0:3]). They control which byte or word of the TMS34020 32-bit memory, latched in the external transceiver, is placed onto the host's data bus.

- SD[0:7] compose the host data path for byte transfers.
- SD[0:15] compose the host data path for word transfers.

The ALIGN signal is generated when the LOCADD[0] address line is high (corresponding to an odd address) and the ISA bus SBHE signal is deasserted to enable odd-byte alignment when doing 8-bit transfers. The ALIGN signal enables the SN74F245 8-bit transfer odd-byte alignment buffer (U51).

The DBUFF signal is generated when the LOCADD[0] address line is low (corresponding to an even address) or the SBHE signal of the ISA bus is asserted to enable even-byte alignment when doing 8-bit transfers in a 16-bit slot or 16-bit transfers. When DBUFF is asserted, it enables the SN74F245 high-byte blocking buffer (SP2).

Table 3–5 shows the LOCADD[0:1] byte selects for both 8-bit and 16-bit transfers.
Host Interface

Table 3–5. Host and Data Latch Byte Select

<table>
<thead>
<tr>
<th>For 8-Bit Transfer (16BTTR Is High)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCADD0</td>
</tr>
<tr>
<td>LOCADD1</td>
</tr>
<tr>
<td>HBS0</td>
</tr>
<tr>
<td>HBS1</td>
</tr>
<tr>
<td>HBS2</td>
</tr>
<tr>
<td>HBS3</td>
</tr>
<tr>
<td>BYTSEL0</td>
</tr>
<tr>
<td>BYTSEL1</td>
</tr>
<tr>
<td>BYTSEL2</td>
</tr>
<tr>
<td>BYTSEL3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For 16-Bit Transfer (16BTTR Is Low)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCADD0</td>
</tr>
<tr>
<td>LOCADD1</td>
</tr>
<tr>
<td>SBHE</td>
</tr>
<tr>
<td>HBS0</td>
</tr>
<tr>
<td>HBS1</td>
</tr>
<tr>
<td>HBS2</td>
</tr>
<tr>
<td>HBS3</td>
</tr>
<tr>
<td>BYTSEL0</td>
</tr>
<tr>
<td>BYTSEL1</td>
</tr>
<tr>
<td>BYTSEL2</td>
</tr>
<tr>
<td>BYTSEL3</td>
</tr>
</tbody>
</table>

Host byte strobes HBS[0:3] are loaded with 1,0,0,0, respectively, when the SHADOK signal is asserted to signify a VGA direct palette access. For more information about direct palette access, see subsection 3.1.8, VGA Passthrough.

Figure 3–9 shows the ABEL source file for programming the HBS PAL.
Figure 3–9. ABEL Source File for the HBS PAL, U28

HBS device 'F22V10';

*Inputs

: hcs      pin 1 ;
locadd0   pin 2 ;
locadd1   pin 3 ;
=!_16bttr  pin 4 ;
!read      pin 5 ;
shadok    pin 6 ;
!loenb    pin 7 ;
!sbhe      pin 8 ;

*Outputs

hbs0      pin 23 ;
hbs1      pin 22 ;
hbs2      pin 21 ;
hbs3      pin 20 ;
bytse10   pin 19 ;
bytse11   pin 18 ;
bytse12   pin 17 ;
bytse13   pin 16 ;
!align    pin 15 ;
!dbuff    pin 14 ;

*Declarations and Intermediate Variable Definitions

X = .X .;
Z = .Z .;
C = .C .;
L = .L .;
H = .H .;

hbs0 ISTYPE 'pos,com,feed_pin';
hbs1 ISTYPE 'pos,com,feed_pin';
hbs2 ISTYPE 'pos,com,feed_pin';
hbs3 ISTYPE 'pos,com,feed_pin';
bytse10 ISTYPE 'pos,com,feed_pin';
bytse11 ISTYPE 'pos,com,feed_pin';
bytse12 ISTYPE 'pos,com,feed_pin';
bytse13 ISTYPE 'pos,com,feed_pin';
align ISTYPE 'neg,com,feed_pin';
dbuff ISTYPE 'neg,com,feed_pin';

EQUATIONS

hbs0 = (!locadd0 & !locadd1) # shadok;

hbs1 = (shadok & (!locadd0 & !locadd1) #
   (!locadd0 & !locadd1 & sbhe & _16bttr));

hbs2 = !locadd0 & locadd1 & !shadok;

hbs3 = !shadok & (!locadd0 & locadd1) #
   (!locadd0 & locadd1 & sbhe & _16bttr));

bytse10 = (hcs & read & !bytse10 & !locadd0 & !locadd1)
   # (bytse10 & hcs);

bytse11 = (hcs & read & !bytse11 & (!locadd0 & !locadd1)
   # (!locadd0 & !locadd1 & sbhe & _16bttr));

bytse12 = (hcs & read & !bytse12 & !locadd0 & locadd1)
   # (bytse12 & hcs);

bytse13 = (hcs & read & !bytse13 & (!locadd0 & locadd1)
   # (!locadd0 & locadd1 & sbhe & _16bttr));

align = !loenb & hcs & locadd0 & !sbhe;

dbuff = !loenb & hcs & (!locadd0 & !sbhe);

END _HBS*
3.1.4.3 The SELECT PAL, U40

The SELECT board select PAL (U40) is a TIBPAL22V10 device that provides access to the SDB20 from the decoded ISA I/O and memory cycles.

The SELECT PAL generates the following signals:
- The TMS34020's host chip select (HCS),
- The SDB20 I/O register enable flag (IOENB),
- The host access flag (READ), and
- The direct palette access flag (PALDIR).

The TMS34020 is selected when its HCS host chip-select input is asserted during a ISA memory access cycle. The HCS signal is asserted when:
1) The base address comparator's (U39) MEMSEL output is asserted,
2) The programmed BASE32K bit (bit0) of the HINTF register compares to LOCADD[15] of the ISA's memory address,
3) The SDBENB bit (bit 7) of the HINTF register is set high, and
4) An SMEMR, SMEMW, MEMR, or MEMW signal is asserted, which initiates an ISA memory cycle.

An SDB20 I/O register is selected when its I/O register enable (IOENB) signal is asserted. The IOENB signal is asserted when the following conditions are present:
- The I/O address comparator's (U12) IOSEL output is asserted,
- An ISA bus IOR or IOW signal is asserted,
- The latched host address bits LOCADD[8:9] are set to 0 and 1, respectively, which result in the decode of host I/O address x2xxh,
- A host-initiated refresh cycle (REFRESH) is not present, and
- The SHADOW flag (SHADOK) is deasserted, which means that the cycle is not a VGA write cycle.

The IOENB signal is asserted when the I/O address comparator (U12) asserts the IOSEL output.

The IOSEL signal is asserted when the latched host address bits LOCADD[4:7] compare to the I/O address space SW1 settings.

The READ signal designates whether the ISA bus cycle is a read cycle or a write cycle.
- When a SMEMR, MEMR, or IOR ISA signal is detected without a current ISA-initiated refresh cycle, READ is asserted, denoting a read cycle.
- Else, READ is held high.
**Note:**

Decoding the MEMx and SMEMx signals simultaneously in the host interface logic allows the SDB20 to function in an IBM PC/AT or IBM PC/XT.

The PALDIR signal is the direct palette access flag used for VGA passthrough. It is asserted when the following occurs:

- An ISA I/O write cycle is asserted (e.g., IOW) to address 03C6h, 03C7h, 03C8h, or 03C9h. These addresses correspond to VGA read mask, read-mode and write-mode registers, and the Bl478 color palette RAM.
- The VGAENB bit (bit 3) of the HSREG is cleared.

Figure 3–10 shows the ABEL source file for programming the SELECT PAL.
SELECT device 'P22V10';

*Inputs

!bclk    pin 1;
!smemr   pin 2;
!memr    pin 3;
!smemw   pin 4;
!memw    pin 5;
!iow     pin 6;
!ior     pin 7;
!memsel  pin 8;
iosel    pin 9;
shadok   pin 10;
!refresh pin 11;
!lreset  pin 13;
sdbenb   pin 14;
aen      pin 21;
base32k  pin 22;
locadd8  pin 20;
locadd9  pin 19;
locadd15 pin 23;

*Outputs

!read    pin 15;
paldir   pin 16;
!hcs     pin 18;
!ioenb   pin 17;

*Declarations and Intermediate Variable Definitions

x    = .x.;
hcs   ISTYPE 'neg,com,feed_pin';
_32kbit = base32k & locadd15 # !base32k & !locadd15;
memcycle = smemr # smemw # memr # memw;
shadow_write = shadok & iow;

EQUATIONS

hcs    = ((shadow_write # memsel & sdbenb & _32kbit & memcycle) &
          # hcs & (shadow_write # memcycle))
          & !refresh & !lreset & !aen;
ioenb  = iosel & !refresh & !aen & ((iow # ior) & !locadd8 & locadd9) & !(shadok &
           iow);
read   = (smemr # memr # ior) & !refresh;
paldir = shadok & iow & !aen;
hcs.ar = 0;
hcs.pr = 1;

END _SELECT

3.1.4.4 The MAPEN PAL, U27

The MAPEN map register control PAL (U27) is a TIBPAL22V10 device that provides the select logic to assert the corresponding SDB20 I/O registers readback latch enable. This is accomplished by decoding the LOCADD[0:3] latched host address bits to determine which register is being accessed. The corresponding enable is asserted when the SELECT PAL (U40) asserts its IOENB signal.
The IOCLK signal, which is related to the MAPCLK signal, is generated when any one of the seven I/O register enables is asserted. The IOCLK signal is used to clock data into the selected MAPA or MAPB mapping registers. When MAPCLK is deasserted, the selected register enable is deasserted.

The STAIT signal latches IOCLK until IOENB deasserts. When an I/O write (IOW) occurs, the MAPEN PAL generates the MAPREAD signal to enable the PC host for reading the selected mapping register.

Figure 3–11 shows the ABEL source file for programming the MAPEN PAL.

**Figure 3–11. ABEL Source File for the MAPEN PAL, U27**

```
MAPEN device 'P22V10';
" MAPEN decodes LOCADD[3:0] to select either:
" LOCADD[3:0] = 0000 Map register A LSB MAPAL
" 0001 Map register A MSB MAPAM
" 0010 Map register B LSB MAPBL
" 0011 Map register B MSB MAPBM
" 0100 Not Used, reserved ~
" 0101 SDB Status Register SDBST
" 0110 Base address register BASEP
" 0111 Host interface control register HINTF
" The unspecified register locations are reserved.

*Inputs
ibclk   pin 1 ;   " PC bus 8 MHz clock
ioenb   pin 2 ;   " IO enable from select.pld
iread   pin 3 ;   "
locadd0 pin 4 ;   " latched PC bus addresses
locadd1 pin 5 ;   "
locadd2 pin 6 ;   "
locadd3 pin 7 ;   "

*Outputs
imapread pin 23 ;  " map register read enable
iomap    pin 22 ;  " map register write clock
imapen0  pin 21 ;  " register selects MAP A LSB
imapen1  pin 20 ;  "   MAP A MSB
imapen2  pin 19 ;  "   MAP B LSB
imapen3  pin 18 ;  "   MAP B MSB
ibaseenb pin 17 ;  "   Base
icntlenb pin 16 ;  "   Control
ishr0    pin 15 ;  "   SDB Status
stait    pin 14 ;  "
```
Figure 3-11. ABEL Source File for the MAPEN PAL, U27 (Concluded)

"Declarations and Intermediate Variable Definitions

X = .x.;
L = 0;
H = 1;
P = .p.;
C = .c.;
ioclk ISTYPE 'neg,reg_d,feed_reg';
mapen0 ISTYPE 'neg,com,feed_pin';
mapen1 ISTYPE 'com,feed_pin';
mapen2 ISTYPE 'com,feed_pin';
mapen3 ISTYPE 'com,feed_pin';
baseenb ISTYPE 'com,feed_pin';
contlenb ISTYPE 'com,feed_pin';
hsrbk ISTYPE 'com,feed_pin';
stait ISTYPE 'com,feed_pin';  "unusual spelling used because of
"ABEL reserved word STATE

EQUATIONS

ioenb & !read & (mapen0 # mapen1 # mapen2 # mapen3 # baseenb # contlenb ) &
!stait;
mapread = ioenb & read;
mapen0 = !locadd0 & !locadd1 & !locadd2 & !locadd3 & ioenb & mapen0 & ioclk;
mapen1 = locadd0 & !locadd1 & !locadd2 & !locadd3 & ioenb & mapen1 & ioclk;
mapen2 = !locadd0 & locadd1 & !locadd2 & !locadd3 & ioenb & mapen2 & ioclk;
mapen3 = locadd0 & locadd1 & !locadd2 & !locadd3 & ioenb & mapen3 & ioclk;
baseenb = !locadd0 & locadd1 & locadd2 & !locadd3 & ioenb & baseenb & ioclk;
contlenb = locadd0 & locadd1 & locadd2 & !locadd3 & ioenb & contlenb & ioclk;
hsrbk = locadd0 & !locadd1 & locadd2 & !locadd3 & ioenb;
stait = ioclk & (stait & ioenb);

END _MAPEN"
3.1.5 Host 8-Bit I/O Write Cycle

The host-generated I/O write cycle shown in Figure 3-12 loads the SDB20's BASEP I/O register with the 8-bit data placed on the ISA data bus SD[0:7].

Figure 3-12. Host-to-SDB20 I/O Write Cycle

Note: The NOWS, REFRESH, and BUSRDY signals are not shown in this figure. However, they are all held high during the write cycle. For all I/O cycles, BUSRDY is held high to prevent wait-states from being inserted into the cycle.

The SDB20's I/O address comparator (U12) uses the latched host-address bits LOCADD[0:3] to generate the IOSEL signal. It is generated when LOCADD[0:3] compares to the SDB20's SW1 I/O address switch settings.

The SELECT PAL (U40) generates the I/O enable (IOENB) signal when the SDB20 detects a valid host-generated I/O write (IOW) or read cycle (IOR). The IOENB and IOSEL signals are asserted simultaneously with the latched host address bits LOCADD[8:9] set to logic levels 1 and 0, respectively. These represent the I/O address space x2xxh.
The IOENB signal enables the SDB20's bidirectional SN74ALS245 I/O data transceiver (U45). The READ signal from the SELECT PAL (U40) determines the direction of the data path.

- It is set for all host-initiated write cycles.
- It is cleared for all host-initiated read cycles.

The MAPEN PAL (U27) decodes the latched host address bits HA[0:3] and designates the I/O register to be accessed. For this example cycle, LOCADD[0:3] are set to 0,1,1,0, respectively. These are decoded by the MAPEN PAL, which generates the BASEENB signal to enable the BASEP register read-back latch.

The MAPEN PAL also generates the MAPCLK signal to clock data into the BASEP register read-back latch. This signal is asserted when the base enable BASEENB signal is asserted during the host I/O write cycle. The rising edge of MAPCLK clocks the data into the latch. The STAIT signal's assertion, resulting from MAPCLK going low, deasserts MAPCLK, providing the rising edge.

The STAIT signal is deasserted when the IOENB signal of the SELECT PAL (U40) is deasserted. This prevents MAPCLK from being reasserted during the host I/O write cycle. The host I/O write cycle ends when IOW deasserts.
3.1.6 Host 8-Bit Memory Read Cycle

Figure 3–13 shows the timing relationships for a host-generated 8-bit memory read cycle.

Figure 3–13. Host-to-SDB20 8-Bit Memory Read Cycle

Note: The NOWS, SDBENB, and BASE32K signals are not shown in this figure. However, they are held high during the read cycle.
For the purposes of this discussion, assume the following:

- The SDB20 is located in an 8-bit slot and is reading 8-bit data with odd-byte alignment (LOCADD[0] is set).
- The SDBENB (bit 7) of the HINTF register is set.

The SN74ALS521 base address comparator (U39) compares the latched host-address bits LOCADD[16:23] to the contents of the latched BASEP register (U6). When they are equal, the MEMSEL signal is asserted.

The latched host-address bit LOCADD[15] is compared to the programmed BASE32K bit (bit 0) of the HINTF register (U7). When they are equal, the 32KBIT signal is asserted.

The SELECT PAL (U40) generates the host chip-select HCS signal when the MEMSEL, 32KBIT, SDBENB, and MEMR signals are asserted. The HCS signal allows external access to the local memory or I/O space of the TMS34020.

The SELECT PAL (U40) also generates the READ signal. It is asserted when the MEMR read enable signal goes low. When the READ and HCS signals are asserted and the LSTATE signal of the GSPHST PAL (U26) is inactive (logic level 0), the GSPHST PAL asserts the HREAD signal. The HREAD signal notifies the TMS34020 that the host PC is requesting read access to its local memory or I/O space.

When the HCS signal is asserted and the LSTATE signal is low, the GSPHST PAL (U26) asserts the NOTREADY signal, which is also known as BUSRDY. This inserts a wait state into the ISA memory cycle.

During the last machine state of the TMS34020's memory access, the TMS34020 HRDY output signal is asserted, indicating that the TMS34020 is ready to complete the host-initiated read cycle. Once HRDY is asserted, the signal LSTATE becomes active and deasserts the HREAD and NOTREADY signals.

The byte-select signals BYTSEL[0:3] are generated from the decoded latched host-address bits LOCADD[0:1] when the HCS signal is asserted. These byte selects determine which byte of the 32-bit word (accessed from the TMS34020's memory and loaded into the SDB20's external data transceivers (U41, U42, U69, and U70) will be placed on the host PC's data bus.

Because LOCADD[0] is set, BYTSEL[1] or BYTSEL[3] will be asserted, depending upon the the state of LOCADD[1].

After the TMS34020 has loaded the external transceivers with data, it asserts the HDST signal to transfer the byte-selected transceivers data to the host PC data bus (SD[0:7]).
3.1.7 Host 16-Bit Memory Write Cycle

Figure 3–14 shows the timing relationships for a host-generated 16-bit memory write cycle.

Figure 3–14. Host-to-SDB20 16-Bit Memory Write Cycle

Note: Although they are not shown in this figure, the NOWS, SDBENB, READ, 16BITEN, and 16XF signals are held high; the BASE32K signal is held low.
For the purposes of this discussion, assume the following:

- The SDB20 is located in a 16-bit slot and is writing 16-bit data.
- The SDBENB (bit 7) of the HINTF register is set.

The SN74ALS521 base address comparator (U39) compares the latched host-address bits LOCADD[16:23] to the contents of the latched BASEP base address register (U6). When they are equal, the MEMSEL signal is asserted.

The latched host-address bit LOCADD[15] is compared to the programmed BASE32K bit of the HINTF register (U7). When they are equal, the 32KBIT signal is asserted.

The GSPHST PAL (U26) generates the HCS host chip-select signal. It is asserted when MEMSEL, 32KBIT, SDENB, and the ISA MEMW write enable signals are asserted. The HCS signal allows external access to the local memory or I/O space of the TMS34020.

The SELECT PAL (U40) generates the READ signal, which is deasserted when the MEMW read enable signal goes low. When READ is high and HCS is asserted, and the LSTATE signal of the GSPHST PAL (U26) is inactive (logic level 0), the GSPHST PAL (U26) asserts the HWRITE signal. The HWRITE signal notifies the TMS34020 that the host PC is requesting write access to its local memory or I/O space. The rising edge of the HWRITE indicates that data in the external tranceivers can be accessed.

When the HCS signal is asserted and the LSTATE signal is low, the GSPHST PAL (U26) asserts the NOTREADY signal, which is also known as BUSRDY. This inserts a wait-state into the ISA memory cycle.

During the last machine state of the TMS34020's memory access, the TMS34020 HRDY output signal is asserted, indicating that the TMS34020 is ready to complete the host-initiated write cycle. Once HRDY is asserted, the signal LSTATE becomes active and deasserts the HWRITE and NOTREADY signals.
The HBS[0:3] host byte selects are generated from the decoded latched host-address bits LOCADD[0:1], the SBHE ISA bus signal, and the 16BITR signal from the GSPHST PAL (U26).

For purposes of this discussion, it is understood that the latched host address LOCADD[0] line is cleared and the HBS[0,1] or HBS[2,3] host byte selects are asserted according to the state of LOCADD[1] signal.

When HCS is asserted, the HBS[0:3] host byte selects are latched to the TMS34020's host interface, the latched host address LOCADD[2:13] lines, and the value of the selected mapping register set (MAP A or MAP B).

The HBS[0:3] host byte selects determine which word of the accessed 32-bit-long word will be written to by the host PC's data bus, SD[0:15]. For more information, see Section 7.4, Basic Communication: How a Host Processor Reads from and Writes to TMS34020 Local Memory, of the TMS34020 User's Guide.

Because SBHE is asserted, the GSPHST PAL (U26) asserts the DBUFF signal, which enables the high-byte blocking buffer (SP2) of the SDB20 to allow 16-bit transfers.

Host PC data is clocked into the external tranceivers when the HSTWCLK signal of the GSPHST PAL (U26) is asserted as a result of the HWRITE signal going active. When the TMS34020 is ready to receive data, the TMS34020 asserts its host output enable HOE signal, which enables the external data tranceiver's output enables (U41, U42, U69, and U70). This signal transfers the host data to the TMS34020 data bus (DAT[0:31]).

### 3.1.8 VGA Passthrough

The VGA passthrough feature of the SDB20 provides a single monitor system the capability of switching between high-resolution graphics and standard VGA video display.

The SDB20 works in conjunction with an 8-bit or 16-bit VGA adapter card by simply connecting the VGA board's feature connector to the SDB20's 26-pin edge connector by means of the VGA passthrough cable (see Figure 3–15).
The SDB20 supports VGA passthrough for greater than or equivalent bus addressing. This means that the SDB20 must be installed in a 16-bit slot when the VGA adapter resides in a 16-bit slot, and in an 8-bit or 16-bit slot when used with an 8-bit VGA adapter.

The VGA passthrough mode is enabled by setting the VGA bit (bit 3) of the HSREG register (U13) and clearing the SHADIS (bit 5) of the HINTF register (U7). When the SHADIS bit is set to logic 0 and the SHADENB signal is asserted, the SDB20 allows shadow palette access. Shadow palette access duplicates VGA I/O write cycles, which write directly to the Bt478 color palette's internal read mask register, read mode address register, write mode address register, or pixel port. When the SDB20 is in the VGA mode, the pixel data, sync pulses, dot clock and blanking signals are passed through the VGA passthrough buffers (U55,U56) to the Bt478 color palette (U65) for display.

The VGA passthrough feature-connectors' output signals are:

- **VGASOURCE** (bits 0–7) is the 8-bit VGA pixel data path. When the SDB20 is in the VGA mode, these bits are buffered and sent to the Bt478 color palette (U65). Only 6 bits of the available eight bits of the pixel data supplied to the Bt478 are used; the two MSBs are not used.

- **VGACLK** is the VGA dot clock. When the SDB20 is in the VGA mode, it is buffered and sent to the Bt478 color palette (U65).

- **VGABL** is the video blank signal. When the SDB20 is in the VGA mode, it is buffered and sent to the Bt478 color palette (U65).
- **VGAHS** is the video horizontal sync signal. When the SDB20 is in the VGA mode, it is buffered and sent directly to the DB9 video connector (P4).

- **VGAVS** is the video vertical sync signal. When the SDB20 is in the VGA mode, it is buffered and sent directly to the DB9 video connector (P4).

### 3.1.8.1 VGA Passthrough Control PALs

Several SDB20 PAL devices allow the VGA feature to be executed. They are:

- The GSPHST PAL (U26) (see page 3-18).
- The HBS PAL (U28) (see page 3-20).
- The MISC PAL (U8) (see page 3-37).
- The SELECT PAL (U40) (see page 3-23).
- The SHADOW PAL (U25).

#### The SHADOW PAL, U25

The SHADOW PAL (U25) is a TIBPAL16L8 device that is part of the board-select logic. It decodes ISA I/O write cycles to addresses 0x03C6h–0x03C9h, which correspond to VGA I/O writes to the VGA color palette’s internal read mask register, read mode address register, write mode address register, and pixel port. When any of these addresses are detected during an ISA I/O write cycle (IOW is brought low), the SHADOW signal is asserted.

When the SHADIS bit (bit 5) of the HINTF register (U7) is cleared and the SHADOW signal is asserted, the MISC PAL (U8) asserts the shadow flag (SHADOK). The SHADOK signal enables the SDB20 to shadow or duplicate the VGA write cycle.

Figure 3–16 shows the ABEL source file for programming the SHADOW PAL.
**Figure 3-16. ABEL Source File for the SHADOW Device, U25**

```ABEL
module _shadow;

"Inputs
!iow     pin  1;
locadd0  pin  2;
locadd1  pin  3;
lccadd2  pin  4;
lccadd3  pin  5;
lccadd4  pin  6;
lccadd5  pin  7;
lccadd6  pin  8;
lccadd7  pin  9;
lccadd8  pin  11;
lccadd9  pin  14;

"Outputs
!sh      pin 19;  "active low direct VGA write enable

"Declarations and Intermediate Variable Definitions
x        = .X.;

EQUATIONS
sh      = iow & locadd9 & locadd8 & locadd7 & locadd6 & !locadd5
& !locadd4 & ((locadd3 & !locadd2 & !locadd1)
# (!locadd3 & locadd2 & locadd1));

END _shadow
```

**The MISC Miscellaneous PAL, U8**

The MISC PAL is a TIBPAL22V10. It generates the Bt478 color palette control signals. These include:

- The palette register address lines REGADD[0:2],
- The palette read (PALETRD) and write (PALETWR) lines.

The MISC PAL also checks to see if SHADIS (bit 5) of the HINTF register is set to enable or disable the shadow feature of the SDB20. When SHADIS is enabled and an ISA I/O VGA write cycle is decoded by the SHADOW PAL (U25), the SHADOK signal is asserted. The SHADOK signal asserts the HCS signal, which starts a TMS34020 host-memory write-cycle.

The SELECT PAL (U40) uses the SHADOK signal to assert the palette direct-access flag (PALDIR) signal. The PALDIR flag, the TMS34020 host-generated UWE signal, and the TMS34020 address-latch (ALTCH) signal generate the color palette register address lines and the PALETWR signal for shadow writes.
When the palette is accessed by a non-VGA ISA memory cycle, the palette-enable PEN signal from the RASDCD device (U1) and the cleared PALDIR flag generate the REGADD[0:2] palette register address lines and the PALETWR or PALETRD signals. The PALETWR and PALETRD signals depend on whether an ISA memory write or read cycle has been initiated.

When the SDB20 is in the non-VGA mode, the CSYNC composite sync signal of the TMS34020 is output on the CHSYNC line to control the system monitor's video timing.

Figure 3–17 shows the ABEL source file for programming the MISC PAL.
Figure 3-17. ABEL Source File for the MISC PAL, U8

MISC device 'P22V10';

"Inputs

dat5  pin 1;  " LSB's of LAD bus
dat6  pin 2;  "
dat7  pin 3;  "
!altch pin 4;  " address latch
!paleten pin 5;  " enable palette
!we   pin 6;  " write enable
locadd1 pin 8;  " support for read mask palette register
!shadow pin 9;
!shadenb pin 10;
!csync pin 11;  " composite sync for non VGA modes
!vgaenb pin 13;
!trqe  pin 14;
paldir pin 16;  " palette direct access
locadd0 pin 23;  " PC bus address LSB

"Outputs

regadd0 pin 22;  " Hardware space register add's
regadd1 pin 21;  "
regadd2 pin 20;  "
!paletrd pin 19;  " palette read enable
!paletwr pin 18;  " palette write enable
shadok pin 17;
!chsync pin 15;  " palette composite sync

*Declarations and Intermediate Variable Definitions

c = .c.;
x = .x.;
z = .z.;
p = .p.;
h = 1;
l = 0;

EQUATIONS

regadd0 = (((dat5 & !paldir) # vgaenb & paldir & locadd0)
 & !altch) # regadd0 & altch;
regadd1 = (((dat6 & !paldir) # vgaenb & paldir & locadd1)
 & !altch) # regadd1 & altch;
regadd2 = (dat7 & !altch & !paldir) # regadd2 & altch;
shadok = shadenc & shadow;
paletrw = we & (paleten # paldir) & altch;
paletrd = iwe & paleten & !paldir & altch & trqe;
chsync = csync & !vgaenb;
chsync.oe = !vgaenb;

END _MISC
3.1.9 VGA Passthrough Cycle

Figure 3–18 shows the VGA passthrough cycle for an ISA VGA I/O write cycle to the Bt478’s internal palette read mask register (x03C6h).

Figure 3–18. VGA Passthrough Cycle

Note: Although they are not shown in this figure, the NOWS and READ signals are held high; the SHADENB and VGAENB signals are held low.
For the purposes of this discussion, assume the following:

- The VGA mode bit (bit 3) of the HSREG register (U13) is set to a logic 1.
- The SHADIS shadow disable bit (bit 5) (SHADENB) of the HINTF register is cleared.

When the latched host address LOCADD[0:9] lines are decoded as 03C6h and the ISA I/O write enable IOW signal is asserted, the SHADOW signal of the SHADOW PAL (U25) is asserted. When the SHADOW signal goes low and the SHADIS bit (bit 5) of the HINTF register is cleared, the SHADOK shadow flag signal of the MISC PAL (U8) is asserted.

Asserting the shadow flag SHADOK causes several things to happen:

1) The HBS[0:3] host byte selects are set to 1,0,0,0, which place the ISA bus DAT[0:7] data into the least significant byte of the TMS34020’s data bus.

2) The PALDIR palette direct access flag is set, which is used by the MISC PAL (U8) to generate the palette control lines. It also drives the ISA NOTREADY signal low, which inserts a wait state.

3) The HCS signal low is driven low, which initiates a TMS34020 host memory access— the duplicated VGA cycle.

When the HCS host chip-select signal of the TMS34020 is asserted and the initial state of the GSPHST PAL (U26) internal LSTATE signal is low, the host write strobe (HWRITE) is asserted. This notifies the TMS34020 that a host write is pending.

PC host data is clocked into the external transceivers when the HSTWCLK signal of the GSPHST PAL (U26) is asserted. This occurs when HWRITE goes active.

The rising edge of the HWRITE signal is used to indicate that the data provided by the SDB20's external transceivers (U41, U42, U69, and U70) can be accessed. During the last machine state of the TMS34020’s memory access, the TMS34020’s HRDY output signal is asserted to indicate that the TMS34020 is ready to complete the host-initiated write cycle. When HRDY is asserted, the LSTATE signal activates and deasserts the HWRITE and NOTREADY signals.

When HCS is asserted, the DBUFF high-byte blocking buffer enable signal of the HBS PAL (U28) is asserted. This buffer allows only even byte-alignment data-transfers between the PC host and the TMS34020.

The TMS34020 asserts the HOE signal when the TMS34020 is ready to receive data from the external transceivers (U41, U42, U69, and U70). The HOE signal is the output enable from the external transceivers that enables the transfer of DAT[0:31] data to the TMS34020’s data bus.

The MISC PAL (U8) generates the REGADD[0:2] and PALETWR palette control signals. The palette register address lines are driven active when the PALDIR and VGAENB signals are asserted. Their state is determined by their corresponding latched host address lines.
The REGADD[0,2] palette register address lines are cleared because the latched host address LOCADD[0] line is cleared. Similarly, REGADD[1] is set because the latched host address LOCADD[1] line is set.

Palette register address 02h is the read mask register of the Bt478 color palette. The data to be written to the palette mask register is available on the TMS34020's data bus (DAT[0:7]) and is strobed into the Bt478 when the PALETWR signal is asserted. The PALETWR signal occurs when the following are asserted:

- The PALDIR flag.
- The write enable strobe (UWE) of the TMS34020.
- The address latch enable signal (ALTCH) signal from the TMS34020.

### 3.1.10 Resetting the SDB20

When you turn on your PC, the system reset RESDRV signal is asserted on the ISA bus. The time that it is active depends upon your PC's reset service routine (RSR) configuration. See your PC operator's manual for more information about RSRs.

The RESET PAL (U15) uses the RESDRV signal to generate the SDB20 local reset LOCRS signal, which resets the on-board TMS34020 and TMS34082 devices and the HSREG and HINTF registers to their default conditions. See subsections 3.1.2.2, *The Host Interface Control Register (HINTF), U7*, and 3.1.2.5, *The Hardware Space Control Register (HSREG)*, for more information about their default conditions.

The RESET PAL (U15) is used to filter noise that is typically present on the ISA bus' RESDRV line. This ensures that the reset to the TMS34020 device is asserted for its required forty local clock LCLK periods at start up. For more information about resetting the TMS34020, see Section 6.12, *Reset*, on page 6-22 of the *TMS34020 User's Guide*.

The RESET PAL also resets the SDB20 when the host reset HRST signal is received from the HINTF (bit 4) register.

You can reset the TMS34020 by setting the TMS34020's HSTCTLH I/O register's RST bit (bit 7). This signal resets the TMS34020 only; it does not affect the TMS34082, HSREG, or HINTF. When RST is set, the TMS34020’s RESET line is not pulled low.
3.1.11 SDB20 Interrupt Control

There are three types of interrupt controls:
1) Host-to-TMS34020 interrupt requests.
2) TMS34020-to-host interrupt requests.
3) TMS34082-to-TMS34020 interrupt requests.

3.1.11.1 How the Host Interrupts the TMS34020

The host processor interrupts the TMS34020 by writing a logic 1 to the INTIN bit (bit 3) of its HSTCTLL register. The host cannot clear INTIN; only the TMS34020 CPU can clear the INTIN bit to deactivate the interrupt request. Similarly, the TMS34020 CPU cannot set INTIN.

The message in MSGIN bits (bits 0–2) of the HSTCTLL register provide a mechanism for specifying the action to be taken by the TMS34020. The MSGIN bits provide up to eight different handling procedures.

Note:

Only the host can write to the MSGIN bits. They can be written to when the INTIN bit is set.

Before returning from the interrupt, the TMS34020 CPU must clear the INTIN bit by writing a logic 0. For more information about host interrupts, see Section 6.8, *Internal Interrupts*, on page 6-16 of the TMS34020 User’s Guide.

Table 2–3 on page 2-6 shows the available IRQ levels for SW2. Consult your PC operator’s manual for details on the applicability of various IRQ levels to your system.

3.1.11.2 How the TMS34020 Interrupts the Host

The TMS34020 CPU interrupts the host processor by asserting its host interrupt HINT pin, which is tied to the selected host IRQ line (IRQ3, IRQ5, or IRQ9–15). This occurs when the TMS34020 CPU writes a logic 1 to its INTOUT bit (bit 7) of the HSTCTLL register. The SDB20’s SDBST register uses the level of the HINT signal for polled-interrupt operation.

For more information interrupting a host, see Section 6.10, *Interrupting a Host Processor*, on page 6-21 of the TMS34020 User’s Guide.

3.1.11.3 How the TMS34020 Interrupts the TMS34082

The TMS34082 uses its hardware interrupt COINT pin to assert the TMS34020’s local hardware LINT1 pin. The COINT signal is asserted when the exception detect-enable ED flag of the TMS34082’s STATUS register is set and an exception has been detected. For more information about TMS34082 valid exception detections, see subsection 4.5.3, *Configuration Register (CONFIG)*, on page 4-13 of the TMS34082 Designer’s Handbook.
When TMS34082 asserts its COINT signal, the TMS34082 aborts the current instruction and goes into an idle state.

The TMS34020 responds to the TMS34082's interrupt by branching to its interrupt service routine (ISR) vector when the LINT1 enable flag is asserted. The COINT signal and the ED flag are reset when the TMS34020's ISR reads the TMS34082's STATUS register.

The TMS34082 SRAM module also uses the COINT signal to generate a software interrupt from the TMS34082's external SRAM program memory.

For more information about interrupt handling, see Section 5.7, Interrupt Handling, on page 5-15 of the TMS34082 Designers Handbook.
3.2 The Frame Buffer/Program Memory Interface

The SDB20 is composed of 1 MByte of DRAM program memory (U17–U24) and 1 MByte of VRAM frame buffer memory (U31–U38).

- The DRAM memories are TMS44C256SD (256K x 4-bit) 100-ns devices that are memory-mapped from 00800000h–00FFFFE0h.

- The VRAM memories are TMS44C251SD (256K x 4-bit) 100-ns devices that are memory-mapped from 00000000h–007FFFFE0h, which allows a maximum supported resolution of 1024 x 768 x 8-bits/pixel.

3.2.1 TMS34020 Memory Mapping

Table 3–6 shows the TMS34020 local memory allocations.

Table 3–6. The TMS34020 Local Memory Map

<table>
<thead>
<tr>
<th>Local Address (in Hex)</th>
<th>Resource</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000–007FFFFE0</td>
<td>VRAM space</td>
<td>R/W:32</td>
</tr>
<tr>
<td>00800000–00FFFFE0</td>
<td>DRAM space</td>
<td>R/W:32</td>
</tr>
<tr>
<td>01000000–BFFFFFE0</td>
<td>Aliased VRAM/DRAM addresses</td>
<td>R/W:32</td>
</tr>
<tr>
<td>C0000000–CFFFFFE0</td>
<td>TMS34020 I/O registers</td>
<td>RAS disabled</td>
</tr>
<tr>
<td>D0000000</td>
<td>Palette address register (RAM write mode)</td>
<td>R/W:32</td>
</tr>
<tr>
<td>D0000020</td>
<td>Color palette RAM</td>
<td>R/W:32</td>
</tr>
<tr>
<td>D0000040</td>
<td>Pixel read-mask register</td>
<td>R/W:32</td>
</tr>
<tr>
<td>D0000060</td>
<td>Palette address (read-only)</td>
<td>R/W:32</td>
</tr>
<tr>
<td>D0000080</td>
<td>Palette address (write-only)</td>
<td>The SDB20 does not use the overlay functions. Access to these registers has no effect on operation.</td>
</tr>
<tr>
<td>D00000A0</td>
<td>Overlay registers</td>
<td></td>
</tr>
<tr>
<td>D00000C0</td>
<td>Reserved for future use</td>
<td></td>
</tr>
<tr>
<td>D00000E0</td>
<td>Palette address for register overlay read</td>
<td>R/W:32</td>
</tr>
<tr>
<td>D0000100–DFFFFFE0</td>
<td>Aliased palette register addresses</td>
<td>R/W:32</td>
</tr>
<tr>
<td>E0000000</td>
<td>HSREG hardware space register</td>
<td>W:Byte:8</td>
</tr>
<tr>
<td>EFFFFFE0–EFFFFFE0</td>
<td>Aliased HSREG addresses</td>
<td>W:Byte:8</td>
</tr>
<tr>
<td>F0000000–FFFFFFFE0</td>
<td>Interrupt/trap vectors</td>
<td>R/W:32</td>
</tr>
</tbody>
</table>
3.2.2 The TMS34020-to-Memory Interface

The TMS34020 read/write local memory cycles are 32-bit memory accesses. The TMS34020 has an on-board instruction-cache and memory controller. The CPU runs independently of the memory controller, except when a TMS34082 coprocessor-inserted wait state is generated. For more information, see Section 3.3, The TMS34082 Floating-Point Coprocessor. The local-memory cycle consists of an address/status subcycle and a data cycle.

During the address/status subcycle, which is also known as row-address time, the local memory address DAT[5:31], status bits DAT[0:4], and local memory row/column address ROWCOL[0:8] are generated.

The RASDCD PAL (U1) decodes the local-memory address to select either BANK0 (VRAM) or BANK1 (DRAM). These signals assert the corresponding row-address strobe RAS as follows:

- When the VRAM row address strobe RAS0 is asserted, BANK0 is selected.
- When the DRAM row address strobe RAS1 is asserted, BANK1 is selected.
- When either row-address strobe is asserted, the row-address lines are latched by the selected strobe.

The status code contained in status bits DAT[0:4] are also decoded by the RASDCD PAL (U1) to determine if the local memory cycle is a refresh, a video-generated VRAM serial register transfer, or a CPU-generated VRAM serial register transfer cycle as discussed in Section 8.5, Local Memory Status Codes, on page 8-10 of the TMS34020 User’s Guide.

When the data subcycle is asserted active-low, which is also known as column-address time, it provides the local memory column address to the row/column address, and selects VRAM (BANK0). The ROWCOL[0:8] lines transfer data out of the TMS34020 by means of the multiplexed address/data bus, DAT[0:31]. When the HBS PAL (U28) asserts its host byte strobes (HBS[0:3]), the corresponding the column address strobes (CAS[0:3]) are asserted. The column address is latched by the local memory.

The data subcycle ends when the HAND PAL (U14) generates the LRDY signal to the TMS34020 and the RASDCD PAL’s BUSFLT signal is deasserted. The LRDY signal is derived from the TMS34082’s coprocessor-ready CORDY signal. In the absence of an on-board TMS34082, CORDY is pulled high. It is deasserted only when a coprocessor instruction is executing. This prevents any TMS34020 local memory cycles from being executed before the TMS34020-initiated coprocessor instruction has been completed. These signals provide zero wait-state local memory cycles on the SDB20.
When the data subcycle ends, the **PGMD** and **SIZE16** signals of the TMS34020 are sampled.

- The **PGMD** signal is tied low so that the SDB20 will use page mode whenever possible. The TMS34020’s page-mode addressing scheme is described in Section 8.7, on page 8-15 of the *TMS34020 User’s Guide*.

- The **SIZE16** signal is held high, which defines the local memory bus size to be 32 bits wide.

For more information on local memory cycles and general timing information, see Section 8.8, *Local-Memory Read and Write Cycles*, on page 8-18 of the *TMS34020 User’s Guide*.

The RASDCD PAL (U1) decodes a TMS34020-generated refresh cycle from its local address bus DAT[0:3], which comprises the TMS34020’s status bits on a local memory cycle. The SDB20 uses CAS before RAS refresh for its memory arrays. When a refresh cycle is asserted, the RASDCD PAL (U1) disables the palette and the HSREG from being accessed.

The TMS34020’s row/column address RCA[1:9] bus provides the memory addressing for VRAM and DRAM. The TMS34020’s local addresses LAD[5:22] are tied to the row/column address bus RCA[5:22] as shown in Table 3-7.

**Note:**

This multiplexing scheme applies to all local memory cycles.

### Table 3-7. Local Memory Addresses-to-RCA Bus Connections

<table>
<thead>
<tr>
<th>RCM</th>
<th>Array Size</th>
<th>Address Time</th>
<th>CAMD</th>
<th>Row/Column Address Bus Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>row</td>
<td>–</td>
<td>22 21 20 19 18 17 16 15 14 n/a</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>column</td>
<td>0</td>
<td>13 12 11 10 9 8 7 6 5 n/a</td>
</tr>
</tbody>
</table>

**Note:** n/a means not applicable.

The CAMD column address mode pin of the TMS34020 is tied low. The RCA bus configuration mode (RCM) bits of the CONFIG register control which local address lines are tied to the RCA bus. The SDB20 requires RCM[0,1] to be set to 1 and 0, respectively.

The RCM bits are loaded from bits 1 and 2 of the TMS34020’s reset vector, address FFFF FFE0h. The setting of the RCM is discussed in subsection 6.12.4, *System Configuration Following Reset*, on page 6–26 of the *TMS34020 User’s Guide*. 

3-47
3.2.3 The RASDCD PAL, U1

The RASDCD PAL (U1) is a TIBPAL16L8 that decodes the memory row-address strobes (RAS). It also provides the palette enable PEN and the hardware space register-enable HWSPACE signals.

The RASDCD PAL differentiates between frame-buffer VRAM (BANK0) and program-memory DRAM (BANK1) with the local-address bits, DAT[23,30,31]. Table 3–8 shows the bit levels for selecting the two memory banks.

Table 3–8. DRAM and VRAM Bank Selection

<table>
<thead>
<tr>
<th>State of the DAT bits</th>
<th>They select</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAT31</td>
<td>DAT30</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The RASDCD also decodes the local-address DAT[28:31] bits to assert the following internal signals:

- **IOREGS** when accessing the TMS34020’s I/O registers.
- **PALET** when accessing the Bt478 color-palette registers.
- **HREGS** when accessing the HSREG hardware space register of the SDB20.
- **INTVECT** when accessing the INT/TRAP vectors of the TMS34020.

Table 3–9 shows the bit levels for selecting the RASDCD PAL output signals.

Table 3–9. RASDCD PAL Signal Selection

<table>
<thead>
<tr>
<th>State of the DAT bits</th>
<th>Selected Signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAT31</td>
<td>DAT30</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The IOREGS signal asserts RAS0 to duplicate the I/O register locations of the TMS34020 into VRAM. This can used as a debugging feature.

The PALET signal enables the PEN signal, which enables the MISC PAL (U8) to access the Bt478 color palette. For more information, see the description of the MISC PAL on page 3-37.
The PEN signal is asserted only when the PALET signal is asserted and there are no current CPU- or VRAM-initiated serial-register transfers or TMS34020-initiated refresh cycles taking place.

The HREGS signal enables the HWSpace signal, which enables writes to the HSREG register.

The HWSpace signal is asserted as long as there are no current CPU- or VRAM-initiated serial register transfers or TMS34020-initiated refresh cycles taking place, and as long as the direct palette access flag PALDIR, which is generated by the SELECT device (U40), is deasserted.

The MSBs of the local-address DAT[28:31] bus are used to divide the memory range into 16 sections. All 16 sections of memory are reserved for compatibility of future revisions of the SDB20; do not overwrite them. The amount of aliasing is due to the lack of exhaustive decoding because the system was designed this way to reduce chip count and conserve board population.

Figure 3–19 shows the ABEL source file for programming the RASDCD PAL.
Figure 3-19. The ABEL Source File for the RASDCD PAL, U1

RASDCD device 'P16L8';

*Inputs

IRAS    pin 1 ;
DAT0    pin 3 ;
DAT1    pin 4 ;
DAT2    pin 5 ;
DAT3    pin 6 ;
DAT23   pin 7 ;    * bank select bit
DAT28   pin 8 ;    * MSB of LAD: 0000 RAM
DAT29   pin 9 ;    * 1100 IO Regs
DAT30   pin 11 ;   * 1101 Palette
DAT31   pin 2 ;    * 1110 HSREG
* 1111 interrupt vector fetch
PDPLR   pin 16 ;   * direct palette access flag from
* host interface.

*Outputs

BUSPLT  pin 19 ;  * not used
IRAS1   pin 18 ;  * DRAM RAS
IRAS0   pin 17 ;  * VRAM RAS
PNEN    pin 15 ;  * palette select
HSPLT   pin 13 ;  * HSREG select

*Declarations and Intermediate Variable Definitions

IOREGS  = DAT31 & DAT30 & !DAT29 & !DAT28;
PALET   = DAT31 & DAT30 & !DAT29 & DAT28;
RREGS   = DAT31 & DAT30 & DAT29 & !DAT28;
INTVCT  = DAT31 & DAT30 & DAT29 & DAT28;
REFRESH = !DAT3 & !DAT2 & DAT1 & DAT0 ;
BANK0   = !DAT3 & !DAT30 & !DAT23 ;
BANK1   = !DAT31 & !DAT30 & !DAT23 ;
VTRANS  = !DAT3 & DAT2 & !DAT1 & DAT0 ;
CTRANS  = !DAT3 & DAT2 & !DAT1 & DAT0 ;

EQUATIONS

RAS0    = (!RAS0 & !RAS1 & !PNEN & !HSPLT & !PDPLR & RAS &
            (BANK0 # IOREGS)) # (!RAS0 & RAS);
RAS1    = (!RAS0 & !RAS1 & !PNEN & !HSPLT & !PDPLR & RAS &
            (BANK1 # INTVCT)) # (!RAS1 & RAS);
PNEN    = (!RAS0 & !RAS1 & !PNEN & !HSPLT & !INTVCT &
            !PDPLR & PALET) # (!PNEN & RAS);
HSPLT   = (!RAS0 & !RAS1 & !PNEN & !HSPLT & !INTVCT &
            !PDPLR & IOREGS & !PDPLR & HREGS) # (!HSPLT & RAS);
BUSPLT  = 0 ;
END _RASDCD
3.2.4 TMS34020-to-VRAM Interface

The SDB20's frame buffer consists of eight TMS44C251SD devices (U31–U38) containing 1 Mbit of VRAM. These provide a maximum supported screen resolution of 1024 x 768 x 8 bits/pixel.

During update cycles, the VRAM transfers a row of memory to the TMS34020 on-chip 256-bit shift register or the contents of the shift register to the row of VRAM memory, depending on the state of the read/write line.

The RASDCD PAL (U1) decodes the status bits of the local-address bus LOCADD[0:3] on local-memory cycles to determine when one of the following is taking place:

- A video-generated VRAM serial-register transfer.
- A CPU-generated VRAM serial-register transfer.
- A TMS34020-initiated memory refresh cycle.

When any of these occur, the PALET and HWSPACE enables are deasserted.

The VRAM is controlled by the following TMS34020- and SDB20-generated signals:

- The MAD[0:8] series-terminated address bus.
- The RAS0 series-terminated row/column address strobe enable.
- The WE series-terminated write enable.
- The SF special function.
- The TR/OE transfer output enable.
- The SCLK serial clock.

The multiple functions of the VRAM are selected by the states of the TMS34020's control signals TR/OE, WE, SF and CAS[0:3] after the fall of RAS. Table 3–10 shows a functional truth table for the TMS44C251 VRAM.
### Table 3–10. TMS44C251 Functional Truth Table

<table>
<thead>
<tr>
<th>CAS3–CAS0</th>
<th>TRQE</th>
<th>WE</th>
<th>SF</th>
<th>CAS Fall</th>
<th>Address</th>
<th>DQ[0:3]</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>ROW</td>
<td>TAP POINT</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>ROW</td>
<td>TAP POINT</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>ROW</td>
<td>TAP POINT</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>ROW</td>
<td>TAP POINT</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ROW</td>
<td>COL</td>
<td>WRITE MASK</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>ROW</td>
<td>COL</td>
<td>WRITE MASK</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>ROW</td>
<td>COL</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>ROW</td>
<td>COL</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>ROW</td>
<td>COL</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>ROW</td>
<td>COL</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>REFRESH</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>REFRESH</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

**Notes:**
1) DQ0–3 are latched on the falling edge or later of W or CAS3–CAS0.
2) X = don’t care.
3) ADDR MASK = 1 means write to the address location that is enabled.
   WRITE MASK = 1 means write to the I/O enabled.

### 3.2.5 TMS34020-to-DRAM Memory Interface

The SDB20’s eight TMS44C256SD 256K x 4-bit 100-ns DRAM program memories are configured and accessed in the same manner as the VRAM framebuffer memory. The only difference between accessing DRAM and accessing VRAM is the address space in which the DRAM resides and the decoding of BANK1 by the RASDCD PAL. The RASDCD PAL’s signal RAS1 asserts BANK1. For specific timing information, see the TMS44C256SD data sheet.
3.3 The TMS34082 Floating-Point Coprocessor

The SDB20 is designed to use the optional TMS34082 graphics floating-point coprocessor; it installs in socket U10. The SDB20 is not shipped with this device. However, it can be purchased and installed separately.

The TMS34082 is a high-speed, IEEE Standard 754-compatible floating-point processor. It is composed of the following:

- A 16-bit sequencer for executing internal or programmed instructions.
- A 3-operand (source A, source B, and destination), 64-bit floating-point unit (FPU) with twenty-two 64-bit data registers. The data registers are organized into two files of 10 registers each with two registers used for internal feedback.
- An instruction register for controlling FPU execution.
- A status register for retaining the FPU’s most recent status.
- 8 control registers.

For more information, see the *TMS34082 Designer’s Guide*.

3.3.1 How the TMS34082 Communicates With the TMS34020

The TMS34020 communicates with the TMS34082 directly over the local address DAT[0:31] bus. The TMS34020 supplies the local-clock signals LCLK1 and LCLK2 to provide synchronous communication between it and the TMS34082. The address-latch ALTCH signal is used for latching the instruction and status from the local address DAT[0:31] bus. When the TMS34082 decodes all zeroes from the local address DAT[0:3] bus, the TMS34082 executes the TMS34020-initiated access cycle.

When the column address CAS[0:3] strobes fall, the TMS34020’s local address/data bus is accessed if the local bus data-ready LRDY signal is high. The level of the special function SF input signal determines the status of the DAT[0:31] input to the coprocessor.

- When SF is set, the DAT[0:31] is provided with an instruction or a data operand from the TMS34020’s registers.
- When SF is cleared, the DAT[0:31] is a data operand from the TMS34020’s memory.

The level of the write enable WE signal determines the TMS34020-initiated read or write cycle. The row address RAS strobe is held high for any TMS34082 accesses.

The TMS34082 handshakes with the TMS34020 by means of its coprocessor-ready CORDY and coprocessor-interrupt COINT signals. When the coprocessor executes a TMS34020-initiated instruction, CORDY is brought low until the instruction is completed. CORDY is then sent to the HAND PAL (U14). The HAND PAL generates the LRDY signal to the TMS34020. Wait states are inserted into the cycle when LRDY is asserted on a local-memory access.
For further information about communicating with the TMS34082 device, see Section 5.6, *Polling the Coprocessor*, on page 5-13 of the *TMS34082 Designers Handbook*.

### 3.3.2 The HAND PAL, U14

The HAND local bus handshake PAL is a TIBPAL20R4A that has very little functionality for the current revision level of the SDB20. However, it does supply the direct-connect logic of the TMS34082's CORDY signal to the TMS34020's LRDY signal.

The HAND PAL also inverts the SDBENB input signal to the 16XF output enable signal, which is used to enable the SN74ALS521 16-bit transfer address comparator (U30).

Figure 3–20 shows the ABEL source file for programming the HAND PAL.

*Figure 3-20. ABEL Source File for the HAND PAL, U14*

```ABEL
HAND device 'P20R4';

*Inputs
LCLK2    pin 1 ;
DAT0,DAT1,DAT2,DAT3 pin 2,3,4,5 ;
DAT23    pin 6 ;
DAT28,DAT29,DAT30,DAT31 pin 7,8,9,10 ;
!ALTCH   pin 11 ;
CORDY    pin 23 ;
!OUTENABLE    pin 13 ;
SDBENB    pin 14 ;

*Outputs
LRDY     pin 22 ;
!PGMD    pin 21 ;
!SIZE16   pin 16 ;
!_16XF    pin 15 ;
ST0,ST1,ST2 pin 18,19,20 ;

*Declarations and Intermediate Variable Definitions

EQUATIONS
PGMD    = 1 ;
SIZE16   = 0 ;
LRDY     = CORDY ;
_16XF    = SDBENB ;
END _HAND
```

### 3.3.3 Installing the TMS34082

You can install the TMS34082 by inserting a 145-pin PGA (pin-grid array) package into socket U10 on the SDB20. U10 is located below and to the left of the TMS34020 device that is located at U11 (see Figure 3–1 on page 3-3).
3.3.4 The TMS34082 SRAM Module

The TMS34082 SRAM module is a business-card-sized daughter board that is used with the TMS34020 software development board (SDB20). The SRAM module's powerful TMS34082 graphics floating-point coprocessor easily adds floating-point capability to your SDB20. Figure 3–21 shows the module’s layout and how it installs on the SDB20.

Figure 3–21. TMS34082 SRAM Module Installation

3.3.4.1 Key Features

The TMS34082 SRAM module has the following features:

- A TMS34082A-32 graphics floating-point coprocessor.
- 128K bytes of fast 20-ns static RAM (SRAM).
  This memory can be used for both external instructions and data (no program/data segment decoding).
- The ability to plug directly into the SDB20's TMS34082 socket.
- A low profile.
  This allows the SDB20 to remain a single-slot board.
3.3.5 The TMS34082 Bus Architecture

Figure 3–22 shows the standard TMS34082 bus architecture for the SDB20 with the optional TMS34082 device installed.

Figure 3–22. The TMS34082 Bus Architecture in Standard SDB20

This configuration allows the TMS34082 to execute internal instructions, one at a time, as directed by the TMS34020, using data that the TMS34020 has placed into the TMS34082's internal registers.

3.3.6 The TMS34082 Bus Architecture With the SRAM Module Installed

Adding the TMS34082 SRAM module extends the architecture, which allows the TMS34082 to execute custom, multi-instruction subroutines that are stored in external memory. Data for these external instructions can also be loaded or stored in the SRAM module memory.

Figure 3–23 shows the TMS34082 bus architecture with the SRAM module (that contains the TMS34082 and the SRAM) installed.
Figure 3–23. The TMS34082 Bus Architecture Using the SRAM Module

Data From LAD port; instructions From LAD and MSD Ports

Data and instructions From LAD and MSD Ports

For Additional Information...

For more information about the TMS34020 graphics processor and the TMS34082 graphics floating-point coprocessor, refer to the following:

- The *TMS34020 User’s Guide* for more information about the TMS34020 graphics processor.
- The *TMS34082 Designer’s Handbook* for more information about the TMS34082 floating-point processor.
3.4 Backend Section

This section describes the following:

- The SDB20's frame buffer-to-color palette interface,
- The SDB20's control logic and timing,
- The on-board Brooktree color palette’s (Bt478) functionality and associated logic, and
- The configurations of the video output connector (P4).

3.4.1 The Frame Buffer-to-Color Palette Interface

The frame buffer consists of eight TMS44C251 SD VRAM devices (U31–U38). The SN74FCT53 inverting synchronizer (U61) supplies the SCLK serial-clock, VCLK video-clock, BLANK, SRLOAD4, and SRLOAD8 signals. On its falling edge, the SCLK signal clocks data out of the VRAM 256-bit serial registers. On its rising edge, SCLK latches the data into the Bt478.

The SCLK signal is 1/4 of the system dot clock for 8 bits/pixel resolution and 1/8 of the system dot clock for 4 bits/pixel resolution. The HSREG register (U13) selects the clock rate (PIXRATE), which is controlled by the SDB20 clock-generation control logic (see the Video Clock Generation schematic sheet of Appendix C).

The VCLK signal synchronizes and clocks the TMS34020 video control registers, which control the VRAM memory-to-register transfer rate to the VRAMs internal shift registers. For more information about these controls, see Chapter 9, Video Timing and Screen Refresh, of the TMS34020 User’s Guide.

The RATE signal of the HSREG register is also known as PIXRATE. This signal selects the 25-MHz (U72) or 60-MHz (U71) crystal oscillators that drive the video-clock VCLK, serial-clock SCLK, and palette dot-clock PALDOT signals.

- When RATE=1, the SN74AS153 dual 4-to-1 dot-clock multiplexer (U66) selects the 60-MHz crystal oscillator (U71) for clocking both outputs of the system clock. This dot clock frequency provides the connected monitor with 1024 x 768 screen resolution.

- When RATE=0, the dot clock multiplexer selects the 25-MHz crystal oscillator (U72) for the system dot clock. This dot-clock frequency provides the connected monitor with 640 x 480 screen resolution.
The SN74F199 4-bits/pixel (U46–49) and 8-bits/pixel shift registers (U44, U50, U52, and U53) use the SRLOAD4 and SRLOAD8 signals, respectively, to shift 32-bit data words from the VRAM's serial port into the 4-bit/pixel and 8-bit/pixel shift registers.

The 4BIT parameter (bit 1) of the HSREG register enables the 4-bit shift registers with the PIX4BIT signal as shown in the SDB20 schematic set (see Appendix C). When enabled, 32 bits of data are parallel-loaded into the 4-bit shift registers. These registers can be loaded every eight cycles of DOTCLOCKB. Each cycle of DOTCLOCKB loads the Bt478 color palette with four bits of pixel data.

The 8BIT parameter (bit 0) of the HSREG register enables the 8-bit shift registers with the PIX8BIT signal as shown in the SDB20 schematic set (see Appendix C). When enabled, 32 bits of data are parallel-loaded into the 8-bit shift registers. These registers can be loaded every four cycles of DOTCLOCKA. Each cycle of DOTCLOCKA loads the Bt478 color palette with eight bits of pixel data.

The color palette translates each received pixel into an analog red, green, and blue signal. These signals are RS-343A compatible and are passed through the video output connector (P4) to the system’s monitor.

### 3.4.2 The MISC PAL, U8

As previously discussed in subsection 3.1.8.1, **VGA Passthrough Control PALs**, the MISC PAL (U8) provides read/write palette control for the Bt478 color palette’s pixel-address and data registers.

When the VGA bit (bit 3) of the HSREG is a logic 0, VGA passthrough is disabled. The TMS34020-generated composite video CSYNC signal is provided to the MISC PAL, which buffers the composite/horizontal sync CHSYNC signal. CHSYNC is provided to the video output connector (P4).

When the VGAENB signal is asserted, which means that the VGA passthrough is enabled, and the direct-palette access flag PALDIR is asserted by the SELECT PAL (U40), the register-address REGADD[0:2] signals address the Bt478 color palette registers. REGADD[0:2] signals are generated, depending on the state of the two LSBs of the latched host-address lines.

When the *shadow mode* is disabled by the SHADIS bit (bit 5) of the HINTF register being set, the Bt478 registers are addressed according to the value of the local address DAT[5:7] bus lines.
The color palette read-enable PALETRD signal is generated when the RASDCD PAL (U1) asserts the color palette enable PEN signal. The PEN signal occurs when the following conditions are present:

- The MSBs of the local address DAT[28:31] bus are decoded as palette register addresses,
- The direct-palette access flag PALDIR is deasserted,
- The TMS34020's address latch ALTCH and transfer output enable TR/QE are active, and
- The local cycle is not a write cycle, e.g. WE is deasserted.

The Bt478 color palette can be read only when the SDB20 is in the nonshadow mode. The color palette write-enable PALETWR signal is asserted when either PEN is asserted or the PALDIR is set. The TMS34020's ALTCH and WE signals must also be asserted.

The Bt478 color palette can be written to in shadow or nonshadow modes.

- The shadow mode is enabled, which means that VGA writes can update the palette, or
- The nonshadow mode is enabled, which means that writes from the SDB20 can update the palette.

The MISC PAL also verifies shadow-mode activity by generating the SHADOK flag. The SHADOK signal is asserted when a VGA I/O write cycle occurs and SHADOW is asserted.

When the SHADENB bit of the HINTF register is set, the shadow mode is enabled. This signal sets the PALDIR in the SELECT PAL (U40).

### 3.4.3 The Bt478 Color Palette

The Brooktree (RAMDAC) Bt478 color palette is a 66-MHz, 256-bit color palette, which provides RS-343A compatible analog IOR (red), IOG (green), and IOB (blue) video signals to the P4 video output connector of the SDB20. These signals are terminated with 75-Ω resistors. The palette's overlay capability is not available on the SDB20. The overlay OL[0:3] lines are grounded. For a functional description of the device, see the Bt478 data sheet contained in the Brooktree 1990 data book.
3.4.4 The SDB20 Video Output Connector, P4

The video output connector (P4) of the SDB20 is a DB9, 9-pin connector. When you are in the VGA passthrough mode, P4 provides a means for externally accessing the VGA-generated horizontal sync VGAHS and the vertical sync VGAVS signals. These signals are buffered by an SN74F244 8-bit octal transceiver (U56) as the CHSYNC and VSYNC signals.

When the VGA bit (bit 3) of the HSREG is a logic 0, the following occur:

1) The VGA passthrough buffers (U56) are disabled, leaving the VGA-generated sync VSYNC and CHSYNC signals in a high-impedance state.

2) The TMS34020-generated composite video CSYNC signal is provided to the MISC PAL, which provides the buffered composite/horizontal sync CHSYNC signal.

3) The CHSYNC signal is provided to the video output connector (P4).

For more information on the TMS34020's composite-sync signal, see Section 9.6, Composite Video Timing, on page 9-15 of the TMS34020 User's Guide.

Note:

Some monitors may require a 9-pin to 15-pin connector adapter. See your monitor specifications for video connector pinouts and Section A.2, Video Connector Pinouts.
3.5 The TMS34020-to-XDS500 Interface

You can interface the SDB20 TMS34020 device to an XDS500 in-circuit emulator by means of the SDB20's 12-pin emulator I/O connector (JP2). JP2 provides a path for connecting the TMS34020 emulator EMU[0:3] and local-clock LCLK signals to the XDS500. The EMU[0:3] signals are buffered by the SDB20's SN74F244 8-bit octal transceiver (U56).

The LCLK0 local-clock signal is not buffered. Consequently, it is not suited for simultaneous operation with a TMS34082 and XDS500/SDB20. For more information about the XDS500 emulator and its functionality, see the *XDS500 User’s Guide.*
This appendix contains technical information about the SDB20's physical characteristics, its operating environment, video output pinouts, hardware interrupt and I/O address settings.

These topics are discussed:

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 SDB20 System Specifications</td>
<td>A-2</td>
</tr>
<tr>
<td>A.2 Video Output Pinouts</td>
<td>A-3</td>
</tr>
<tr>
<td>A.3 Video Timing</td>
<td>A-4</td>
</tr>
<tr>
<td>A.4 SDB20 Input/Output Registers</td>
<td>A-5</td>
</tr>
<tr>
<td>A.5 Hardware Interrupt Settings</td>
<td>A-6</td>
</tr>
</tbody>
</table>

For additional information about the SDB20, its specifications, and TIGA applications, contact the Texas Instruments Graphics Hotline at (713) 274-2340.
## A.1 SDB20 Hardware Specifications

**Table A-1. SDB20 Hardware Specifications**

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graphics processor</strong></td>
<td>TMS34020 Speed: 32 MHz</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>Base: 1 MByte of DRAM 1 MByte of VRAM</td>
</tr>
<tr>
<td><strong>Bus interface</strong></td>
<td>8-bit or 16-bit ISA interface AT and EISA compatible</td>
</tr>
<tr>
<td><strong>Display modes</strong></td>
<td>640 H x 480 V, 16 colors 640 H x 480 V, 256 colors 1024 H x 768 V, 16 colors 1024 H x 768 V, 256 colors</td>
</tr>
<tr>
<td><strong>Color palette</strong></td>
<td>Brooktree 478 Speed: 66 MHz Size: 256 24-bit entries</td>
</tr>
<tr>
<td><strong>Monitor interface</strong></td>
<td>RS-343A compatible RGB levels into 75-Ω TTL-level horizontal and vertical sync</td>
</tr>
<tr>
<td><strong>Default display timings</strong></td>
<td>640H x 480V (noninterlaced) Pixel rate: 25 MHz Vertical frequency: 60 Hz 1024H x 768V (noninterlaced) Pixel rate: 60 MHz Horizontal frequency: 47.9 kHz Vertical frequency: 59.8 Hz</td>
</tr>
<tr>
<td><strong>Dimension</strong></td>
<td>Standard XT form factor with 16-bit slot capability</td>
</tr>
<tr>
<td><strong>Operating environment</strong></td>
<td>Temperature 0° C to 55° C Relative humidity 10% to 95% (noncondensing) Power supply 5 V ± 5%, up to 100mV peak-to-peak noise margin</td>
</tr>
<tr>
<td><strong>Power consumption</strong></td>
<td>10-W maximum Approximately 2 amps @ +5 VDC</td>
</tr>
</tbody>
</table>
The video output connector (P4) is a shielded, double-row, 9-position subminiature D female adapter. It is compatible with the IBM VGA standard. Table A-2 defines its pinouts.

Table A-2. Video Output Connector Pinouts

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Red video</td>
</tr>
<tr>
<td>2</td>
<td>Green video</td>
</tr>
<tr>
<td>3</td>
<td>Blue video</td>
</tr>
<tr>
<td>4</td>
<td>Composite/horizontal sync</td>
</tr>
<tr>
<td>5</td>
<td>Vertical sync</td>
</tr>
<tr>
<td>6–9</td>
<td>Ground</td>
</tr>
</tbody>
</table>
A.3 Video Timing

The SDB20 supports $1024 \times 768$ noninterlaced video and $640 \times 480$ noninterlaced video displays. Table A–3 and Table A–4 show the default video timings for all three display resolutions. By using the step-by-step procedures listed in Section 2.4, Setting Up a Dual Monitor System, you can customize these timings for your monitors.

Table A–3. $1024 \times 768$ Noninterlaced Video Timing

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel rate</td>
<td>63.960 MHz ±0.01%</td>
</tr>
<tr>
<td>Horizontal scan rate</td>
<td>48.19 kHz</td>
</tr>
<tr>
<td>Horizontal sync width</td>
<td>1.00 µs</td>
</tr>
<tr>
<td>Horizontal front porch</td>
<td>1.00 µs</td>
</tr>
<tr>
<td>Horizontal back porch</td>
<td>2.75 µs</td>
</tr>
<tr>
<td>Vertical frame rate</td>
<td>60.0 Hz</td>
</tr>
<tr>
<td>Vertical sync width</td>
<td>0.083 ms</td>
</tr>
<tr>
<td>Vertical front porch</td>
<td>0.02 ms</td>
</tr>
<tr>
<td>Vertical back porch</td>
<td>0.585 ms</td>
</tr>
</tbody>
</table>

Table A–4. $640 \times 480$ Noninterlaced Video Timing

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel rate</td>
<td>25.175 MHz ±0.01%</td>
</tr>
<tr>
<td>Horizontal scan rate</td>
<td>31.47 kHz</td>
</tr>
<tr>
<td>Horizontal sync width</td>
<td>3.81 µs</td>
</tr>
<tr>
<td>Horizontal front porch</td>
<td>1.11 µs</td>
</tr>
<tr>
<td>Horizontal back porch</td>
<td>1.91 µs</td>
</tr>
<tr>
<td>Vertical frame rate</td>
<td>59.9 Hz</td>
</tr>
<tr>
<td>Vertical sync width</td>
<td>0.06 ms</td>
</tr>
<tr>
<td>Vertical front porch</td>
<td>0.763 ms</td>
</tr>
<tr>
<td>Vertical back porch</td>
<td>1.08 ms</td>
</tr>
</tbody>
</table>
A.4 SDB20 Input/Output Registers

Table A-5 shows the host I/O address locations of the SDB20 that are configured by switch SW1. The SDB20 is factory preconfigured to respond at addresses 0x230. Subsection 2.2.2.1, How to Reconfigure Your SDB20, explains how to make optional address selections.

Table A-5. ISA I/O Address Locations

<table>
<thead>
<tr>
<th>Address</th>
<th>SW1-1</th>
<th>SW1-2</th>
<th>SW1-3</th>
<th>SW1-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0200</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>0210</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>0220</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>0230</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>0240</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>0250</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>0260</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>0270</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>0280</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>0290</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>02A0</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>02B0</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>02C0</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>02D0</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>02E0</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>02F0</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

default settings
A.5 SDB20 Hardware Interrupt Settings

Table A-6 shows the hardware interrupt locations that are configured by switch SW2. The SDB20 is factory preconfigured to respond at IRQ15. Subsection 2.2.2.1, How to Reconfigure Your SDB20, explains how to make optional interrupt selections.

Table A-6. SDB20 Hardware Interrupt Settings

<table>
<thead>
<tr>
<th>Switches</th>
<th>Interrupt Line</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2-1</td>
<td>IRQ3</td>
<td>available</td>
</tr>
<tr>
<td>SW2-2</td>
<td>IRQ5</td>
<td>available</td>
</tr>
<tr>
<td>SW2-3</td>
<td>IRQ9</td>
<td>available</td>
</tr>
<tr>
<td>SW2-4</td>
<td>IRQ10</td>
<td>available</td>
</tr>
<tr>
<td>SW2-5</td>
<td>IRQ11</td>
<td>available</td>
</tr>
<tr>
<td>SW2-6</td>
<td>IRQ12</td>
<td>available</td>
</tr>
<tr>
<td>SW2-7</td>
<td>IRQ14</td>
<td>reserved</td>
</tr>
<tr>
<td>SW2-8</td>
<td>IRQ15</td>
<td>available</td>
</tr>
</tbody>
</table>
Appendix B

Troubleshooting Guide

This appendix contains additional technical information about TIGA. It provides suggested solutions for problems that you may encounter in using TIGA with the SDB20.

These topics are discussed:

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1 TIGA Error Messages</td>
<td>B-2</td>
</tr>
<tr>
<td>B.2 Customer Support</td>
<td>B-6</td>
</tr>
</tbody>
</table>
B.1 TIGA Error Messages

If TIGA encounters a system problem, an error message appears. The exact error message depends upon the application that is running.

The three most common errors result from an improper or incomplete configuration of your TIGA environment. These errors occur because of the following:

1) The TIGA communications driver is not installed, or
2) The TIGA environment is incorrectly set, or
3) The TIGA path is incorrect.

The arrows left of Figure B-1 show these three lines.

Figure B-1. SDB20 Installation Option Window

The following is a list of TIGA messages, their causes, and possible remedies. If these problems occur when running a TIGA application, they are more likely to be caused by a bug in that program than by a hardware malfunction.

Error Message: Abort (no response from the display board)

Cause: The SDB20 failed to initialize. The following are possible causes:

4) The SDB20 is not installed or it is defective.
5) The SDB20 is set up incorrectly.
6) You are using the wrong TIGACD.
7) You are using the wrong TIGA configuration file (e.g., tiga.cfg)
Remedy: Do the following to correct the problem:

Turn off power to your PC, monitor(s), peripherals, and the target system (if applicable) before removing or replacing any cards or cables. Leave PC and peripheral power cords plugged into their respective outlets. This protects your PC from static electricity that may be generated during the installation process.

1) Make sure that the SDB20 is installed correctly by checking the following:
   a) Is the board installed in the correct slot? If not, remove it and install it in the slot selected during the software installation.
   b) Is the board fully seated into its PC slot? Remove and reinstall the board, making sure that it is fully seated into the PC slot.
   c) Make sure all socketed components (i.e., the oscillators) are fully seated into their respective sockets.

2) Rerun the tigaset utility to make sure that all I/O and memory addresses that were required for setup correspond to your board's configuration. Try different combinations to see if any of the setup options correct the problem.

3) Make sure that the TIGACD that you are using is intended for your display board. If in doubt, see the software installation manual for your board.

4) Make sure that the tiga.cfg file that you are using is intended for your display board. If in doubt, see the software installation manual for your board.

Error Message: Attempt to unload TIGA driver failed!

Cause: Using tigacd -u from the DOS line, an attempt was made to unload the TIGA CD from memory but failed because the interrupt vector that was present at the current interrupt level did not belong to TIGA.

Remedy: Do the following to correct the problem:

1) Do not modify the TIGA interrupt level with the -i option of the TIGA environment variable after loading the TIGA CD.

2) Make sure that no other driver is using the same interrupt as TIGA.
**Error Message:** Non TIGA Driver installed in software interrupt level 0x??

*Cause:* An attempt was made to load the TIGA CD at an interrupt level that is being used by some other driver.

*Remedy:* Do the following to correct the problem:

1) Load the TIGA CD at another interrupt level.

2) Change the software interrupt level by using the -i option of the TIGA environment variable. For more information, see Section 2.5, *The TIGA Environment Variable*, of the *TIGA Interface User's Guide*.

**Error Message:** Not enough memory to store parameters

*Cause:* An alternate _a entry point was called, but there is not enough TMS340 memory available to allocate a temporary command buffer.

*Remedy:* Reduce the amount of data being sent to the function.

**Error Message:** Required file tiga.cfg not found

*Cause:* The tiga.cfg file could not be found in the directory that was specified by the -m option of the TIGA environment variable.

*Remedy:* Do the following to correct the problem:

1) Make sure that the -m option is set to the TIGA main directory. For more information, see Section 2.5, *The TIGA Environment Variable*, of the *TIGA Interface User’s Guide*.

2) Make sure that the tiga.cfg file exists in the directory that was specified by the -m option.

3) Make sure that the tiga.cfg "hidden" or "read-only" file attributes are not set.

4) Make sure that the tiga.cfg file is not damaged. If in doubt, see the software installation manual for your board.

**Error Message:** TIGA Driver already installed in software interrupt level 0x??

*Cause:* An attempt was made to load the TIGACD at the current TIGA interrupt level when it is already loaded.

*Remedy:* Do not reload the TIGACD if it is already loaded—it remains memory-resident until:

1) The PC is rebooted or the power is turned off.

2) The TIGACD is uninstalled when using the tigacd -u command from the DOS line.
**Error Message:** TIGA Driver is not installed!

**Cause:** The TIGACD communications driver was not loaded. An attempt was made using `tigacd -u` from the DOS line to unload the TIGACD from memory. It failed because no interrupt vector was present at the current TIGA interrupt level.

**Remedy:** Do the following to correct the problem:

1) Reboot your PC to set up the TIGA environment or run TIGAAUTO.

2) **Do not** attempt to unload the TIGACD when it is not loaded.

3) After loading the TIGACD, **do not modify** the TIGA interrupt level by changing the `-i` option of the TIGA environment variable.

**Error Message:** Timeout waiting for current TIGA command

**Cause:** The TIGA graphics manager has been corrupted. A called TIGA extended function did not execute to completion.

**Remedy:** Do the following to correct the problem:

1) Check the TIGA-function source for problems.

2) Check the host entry-point invocation to ensure that the arguments are correctly passed.
B.2 Customer Support

If these suggested solutions do not solve your problem, do one the following:

1) Contact your local TIGA board dealer.

2) Contact the Texas Instruments Computer Video Products HOTLINE as follows:
   a) Call (713) 274-2340.
   b) Send a FAX to 713-274-2558.
   c) Use EMAIL and send a message to 4599744@mcimail.com.

3) Refer to the customer support guide that was shipped with this manual for information about product warranties and remedies.
Appendix C

SDB20 Schematics

This appendix contains the schematics (revision C) for the TMS34020 Software Development Board (SDB20), revision D.
NOTES, UNLESS OTHERWISE SPECIFIED:
1. RESISTANCE VALUES ARE IN OHMS, .25 WATT AND 1%
2. CAPACITOR VALUES ARE IN MICROPARDAS
3. REFERENCE DESIGNATION AND PIN NUMBER
   ASSIGNMENT ARE AS SHOWN BELOW:

7400 IS THE PART TYPE
1, 2, 3 ARE THE PIN NUMBERS
VIA IS THE REFERENCE DESIGNATOR AND DEVICE
   INDICATOR FOR MULTI-ELEMENT PACKAGES

REVISIONS

<table>
<thead>
<tr>
<th>REV</th>
<th>DESCRIPTION</th>
<th>DATE</th>
<th>APPROVED</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>ECN565444(S)</td>
<td>04-23-90</td>
<td>J.CLAKE</td>
</tr>
<tr>
<td>B</td>
<td>ECN565453(S)</td>
<td>04-24-90</td>
<td>J.CLAKE</td>
</tr>
<tr>
<td>C</td>
<td>ECN565502(S)</td>
<td>04-23-90</td>
<td>J.CLAKE</td>
</tr>
</tbody>
</table>

COMPUTER GENERATED DRAWING : DO NOT REVISE MANUALLY

PARTS LIST

<table>
<thead>
<tr>
<th>PART NO.</th>
<th>DESCRIPTION</th>
<th>NUMERATOR OR DESCRIPTION</th>
<th>NOTES</th>
</tr>
</thead>
</table>

TENNS INSTRUMENTS

FILE : PROJ01B.ROT

P.O.: 2236002-0001

SXBB:

APPLICATION
256k x 32 x 2 RAM ARRAY

Optional 256k x 32 DRAM

RAM ARRAY
- RAS[0..31]
- CAS[0..31]
- WE.L

DRAM, SCH

MAD [0 .. 8]
RAS[0 .. 1]
CAS[0 .. 3]
ME.L

PPX[0 .. 31]
SF
TRQ.L
BLK

VHANA, SCH

FILE ASSY ENG

TEXAS INSTRUMENTS/GRAPHIC PRODUCTS

DOCUMENT NUMBER

REV

FILE

DATE

2564602-0001

A

April 21, 1990 SHE
8 BIT/PIXEL
SHIFT REGISTERS

VGA PASS THROUGH
BUFFERS

TEXAS INSTRUMENTS/GRAPHICS PRODUCTS

FILE: 74F199 74F199
ASSY: 255602-0001
SCH: J.J.052
DATE: April 25, 1987 Sheet 8 of 8
Appendix D

Glossary

A

address: A location in an array of bits, bytes, or words of information.

address/status subcycle: First part of a local-memory cycle, sometimes referred to as row-address time.

ALTCH: Address latch signal. You can use the high-to-low transition of ALTCH to capture the address and status present on the LAD bus.

B

back porch: Portion of horizontal or vertical blanking that follows the trailing edge of the horizontal-sync or vertical-sync pulse.

bank: A group of memory devices that form a continuous selection of memory. On the SDB20, each memory bank consists of eight parts.

BASEP: The SDB20’s base pointer register (U6).

bit: A binary digit; usually 1 or 0.

BUSFLT: Bus fault signal. External logic asserts BUSFLT to indicate that a fault occurred on the current bus cycle.

BUSRDY: See NOTREADY.

byte: An 8-bit sequence of adjacent binary digits operated as a unit.

C

CAMD: The column address mode of the TMS34020 device. It shifts the column address on the RCA bus (RCA[0:12]) to allow mixing of DRAM and VRAM address matrices.

CAS: Column-address strobes (CAS0–CAS3). Drives the CAS inputs of the DRAMs and VRAMs.
**character:** A letter, digit, or symbol that is used as part of the organization, control, or representation of data.

**CHSYNC:** Composite/horizontal video synchronization.

**CSYNC:** Composite video synchronization. This signal is generated by the TMS34020.

**CMOS:** Complementary MOS technology.

**COINT:** Coprocessor hardware interrupt. The TMS34082 uses COINT to assert the TMS34020’s local hardware LINT1 pin.

**color depth:** The number of colors that are simultaneously displayable.

**CONFIG:** The TMS34020’s configuration register (16-bit I/O register, address C000 01A0h). Contains fields that selectively enable/disable various aspects of system configuration.

**configured memory:** Memory that is allocated.

**coprocessor:** An additional processor in the system; extends the functionality of the main processor. For example, the TMS34082 is a coprocessor for the TMS34020; in a TMS34020 system, the TMS34082 adds floating-point capabilities to the TMS34020’s functions.

**CORDY:** Coprocessor ready. In the coprocessor mode, if the TMS34020 sends an instruction before the TMS34082 has completed a previous instruction, CORDY goes low to indicate that the TMS34020 should wait.

**DAT[0:31]:** The TMS34020’s local address bus.

**data:** 1. General term for numbers, letters, symbols, and analog quantities that serve as input for computer processing. 2. Any representations of characters or analog quantities to which meaning, if not information, may be assigned.

**diagnostics:** Software that exercises hardware with the objective of uncovering faults.

**display:** A visual representation of data.

**DRAM:** Dynamic RAM. On the SDB20, DRAM stores program, fonts, and other data.

**driver:** A special piece of software that allows other software to exercise a specific piece of hardware.

**dual scan:** A monitor that supports two different display resolutions.
ED: Exception detect status signal. ED is bit 18 of the TMS34082's STATUS register. It is set when an exception has been detected.

EPROM: Erasable programmable read-only memory. A read-only memory in which stored data can be erased by ultraviolet light or other means and reprogrammed bit by bit with appropriate voltage pulses.


EMU[0:3]: Emulator data bus. These signals connect the TMS34020 emulator to the XDS500 in-circuit emulator.

fatal error: An error that prevents the SDB20 from performing its normal useful functions.

feature connector: A card-edge connector that allows VGA pixel and timing information to be available to the SDB20.

flag: A binary status indicator whose state indicates whether a particular condition has occurred or is in effect.

gray levels: Levels of single-color display intensity.

HA: Host address input bus (HA[5:31]). A host processor requests an address over these lines.

HBS: Host byte-select bus (HBS[0:3]). Identifies the bytes to be selected within a specific word.

HCS: Host chip-select signal. A host drives HCS low to latch the current address and byte-select requests.

HINT: Host interrupt signal.

HINTF: Host interface control register (U7). The HINTF is used to describe how the SDB20 responds to PC bus cycles.

HOE: Host output-enable signal.
HRDY: Host ready signal. Driven high when the TMS34020 is ready to complete a host-initiated access.

HREAD: Host read strobe. Driven low during a host’s read request.

HSREG: Hardware space control register. This is not a true register but it is actually the HARD PAL (U13) that acts as a 32-bit addressable 8-bit register. It is located at 0E000000h in the local-memory space of the TMS34020.

HSTCTLH: The TMS34020’s host control I/O register, high word (16-bits, address C000 0100h).

HSTCTLL: The TMS34020’s host control I/O register, low word (16-bits, address C000 00F0h).

HWSPACE: Hardware space enable signal. The RASDCD PAL (U1) decodes the TMS34020 local address DAT[28:31] bits and generates this signal, which enables writes to the HSREG.

HWRITE: Host write strobe. Driven low during a host’s write request.

instruction: A statement that specifies an operation and the values or locations of operands.

instruction set: A set of operation codes for a particular computer or family of processors.

interlaced video: A common means of image display that first draws all the even lines, then all the odd lines. Interlaced video often results in a slight flicker in the image. Television is an example of interlaced video.

interrupt: To stop a process in such a way that it can be resumed.

INTIN: Interrupt-in (bit 3) of the TMS34020’s HSTCTLL register.

INTOUT: Interrupt-out (bit 7) of the TMS34020’s HSTCTLL register.


ISR: Interrupt service routine.

LAD bus: 32-bit local address/data multiplexed bus (LAD[0:31]).

language: A set of representations, conventions, and rules used to convey information.

LCLK1, LCLK2: TMS34020 local output clocks. These clocks provide synchronous communication between the TMS34020 and the TMS34082.
**LINT1**: Local interrupt request. An external device generates an interrupt request to the TMS34020 by driving LINT1 to an active-low state.

**LOCRS**: The local reset signal generated by the RESET PAL (U15). LOCRS is used to reset the TMS34020, TMS34082, and the HINTF and HSREG registers.

**LRDY**: Local ready. External circuitry drives this signal low to stop the TMS34020 from completing a local-memory cycle that it has initiated.

**LRESET**: Local reset. The RESET PAL (U15) generates this signal to reset the HARD PAL (U13).

**LSB**: Least significant bit.

**LS Byte**: Least significant byte.

**LS Word**: Least significant word.

**MAD[0:8]**: Multiplexed series-terminated address bus.

**Mbyte**: Megabyte.

**MEMCS16**: 16-bit memory chip select.

**memory map**: A map of target system memory space that is partitioned into functional blocks.

**microcomputer**: An integrated circuit that consists of a microprocessor, controller, storage registers, some sort of ALU, and memory.

**microprocessor**: An integrated circuit that can be programmed with stored instructions to perform a wide variety of functions.

**mnemonic**: An instruction name that the assembler translates into machine code.

**monitor timings**: A set of parameters governing the synchronization signals required by monitors in order to display images.

**MSB**: Most significant bit.

**MS Byte**: Most significant byte.

**MSGIN**: Message-in (bits 0–2 of the HSTCTLL register).

**MSGOUT**: Message-out (bits 4–6 of the HSTCTLL register).

**MS Word**: Most significant word.

**multisync**: A monitor that supports more than one display resolution by adapting to the provided monitor timing signals.
noninterlaced video: A means of image display in which all lines are drawn sequentially on the screen (see interlaced video).

NOTREADY: A signal generated by the GSPHST PAL (U26). It is also known as BUSRDY. When NOTREADY is driven low, a wait-state is inserted into the ISA memory cycle. NOTREADY is deasserted by the LSTATE signal.

OL[0:3]: Overlay bits. These are the Bt478 color palette’s overlay lines. They are grounded and not used on the SDB20.

page buffer: An arrangement for storing the information of successive frames in different memory locations to reduce flickering during complex animation.

PALDIR: The palette direct access flag.

palette: A selection of simultaneously displayable colors that convert pixel information into color video images.

PALETRD: Color palette read enable. This signal is used to read data from the Bt478 color palette onto the TMS34020’s data bus DAT[0:7].

PALETWR: Palette write signal. It is used to write data from the TMS34020’s data bus DAT[0:7] into the Bt478 color palette’s mask register. PALETWR occurs when the PALDIR, UWE, and ALTCH signals are asserted.

Panning: Moving horizontally through text strings or graphics strings.

passthrough cable: A flat, multiconductor ribbon cable that connects the SDB20 to the VGA to allow VGA passthrough.

PEN: Palette enable signal. The RASDCD PAL (U1) generates PEN for enabling the Bt478 color palette.

personal computer (PC): An EISA- or ISA-compatible machine.

PGMD: Page-mode signal. This signal is tied low so that currently addressed memory supports page-mode accesses.

pixel: A picture element; a dot in a displayed image.

primary monitor: The monitor that is connected to the PC’s video adapter.

programmable read-only memory (PROM): A large-scale integrated circuit chip for storing digital data. It can be erased with ultraviolet light and reprogrammed, or it can be programmed only once, either at the factory or in the field.
RAS: Row-address strobe. When RAS0 is asserted active-low, VRAM (BANK0) is selected. When RAS1 is asserted active-low, DRAM (BANK1) is selected.

RCA: Multiplexed row/column address bus (RCA[0:12]) of the TMS34020.

RCM[0,1]: RCA[0:12] Row address configuration mode (bits 1 and 2 of the TMS34020’s CONFIG register). Determines which bits of the logical address are output on RCA[0:12] at row-address time.

real time: The actual time during which the physical process of a computation transpires in order that results of the computation interact with the physical process.

REGADD[0:2]: Palette register address lines.

refresh: Method of restoring the charge capacitance to a memory device (i.e., DRAM or VRAM) or of restoring memory contents.

register: Temporary storage area for digital data.

RESDRV: System reset signal. It is generated by your PC upon power-up. The RESET PAL (U15) uses RESDRV to generate the SDB20 local reset LOCRS signal, which resets the TMS34020.

resolution: The total number of pixels that an image displays simultaneously.

ROM: Read-only memory.

row-address time: See address/status subcycle.

RSR: Reset service routine.

RST: Reset signal. It is bit 7 of the TMS34020’s HSTCTLH I/O register. When set, it resets the TMS34020 only.

scrolling: Moving through text strings or graphic strings vertically.

SCLK: Serial data clock. This TMS34020 signal drives VRAM serial-data registers. This allows the TMS34020 to track VRAM serial-data-register count, providing serial register-transfer midline-reload cycles.

SDBENB: SDB20 enable bit. This is bit 7 of the SDB20’s HINTF register.

SDB20: The TMS34020 software development board.

secondary monitor: The monitor that displays SDB20 images in a dual-monitor system.
SF: Special function. This is the special function signal of the TMS34020 to 1-MByte VRAMs.

shadow palette access: Allows a VGA I/O write cycle to write directly to the BT478 color palette.

shadowing: Duplicating an ISA-generated write cycle.

SIZE16: Bus size signal. When memory decode logic pulls SIZE16 low, the currently addressed memory or port supports only 16-bit transfers. When held high, the local memory bus size is 32 bits wide.

software: A set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system: e.g., compilers, library routines, manuals, circuit diagrams, etc.

Software Development Board (SDB20): A high-performance, intelligent video adapter that uses a TMS340-family graphics processor and TIGA software interfaces.

strobe: Any control signal that begins or ends a memory access.

tap point: Column address provided to a VRAM during a memory-to-serial register cycle. The column address specifies the point at which the shift register is to be tapped: in other words, which cell of the serial register is to be connected to the VRAM's serial port.


TIGA display mode: A set of parameters, including display resolution, page buffering, and the number of simultaneously displayable colors, that describe the appearance of SDB20-displayed images.

TR/QE: Transfer/output enable. This signal drives the TR/QE input of VRAMs.

unconfigured memory: Memory that is not defined as part of the device’s memory map.

UWE: The TMS34020’s write-enable strobe.

VCLK: Video clock. VCLK is derivative of the video system’s dot clock. VCLK is internal to the TMS34020 and is used to drive the video timing logic.
Glossary

**VGA:** Video graphics adapter.

**VGAENB:** VGA passthrough enable. The signal is the VGA bit (bit 3) of the HSREG (U13).

**VGA passthrough:** Allowing VGA-generated displays to share the same monitor as the SDB20 when the SDB20 is not being used.

**video random-access memory (VRAM):** A special type of memory device designed to facilitate graphics processing and image display. On the SDB20, displayed images are contained in VRAM.

**VSYNC:** Vertical sync. This signal is generated by the TMS34020.

**wait state:** The clock period inserted into a memory cycle in order to permit accesses of slower memories and slower memory-mapped devices.

**WE:** Write-enable signal. Drives the WE inputs of the DRAMs and VRAMs.

**window:** A specified rectangular area of virtual space shown on the display screen.
ABEL source files
GSPHST PAL (U26), 3-19
HAND PAL (U14), 3-54
HARD PAL (U13), 3-12
HBS PAL (U28), 3-22
MAPEN PAL (U27), 3-26
MISC PAL (U8), 3-39
RASDCD PAL (U1), 3-50
SELECT PAL (U40), 3-25
SHADOW PAL (U25), 3-37
address conflict resolution, 2-4
address conversion principles, 3-12
  host-to-TMS34020, 3-13
  TMS34020-to-host, 3-15
address/status subcycle
  description, 3-46
  row-address time, 3-46
aliasing, 3-49
ALIGN signal, description, 3-20
ALTCH signal, 3-37, 3-42, 3-53, 3-60
AUTOEXEC.BAT modification, 2-18

BASE32K signal, 3-16, 3-31, 3-33
  bit 0 of the HINTF (U7), 3-13
  description, 3-6
BASEENB signal, enables the BASEP register, 3-29
BASEP register (U6), 3-13, 3-33
  description, 3-5
  reading 8-bit data into it, 3-30
  writing 16-bit data into it, 3-32
  writing 8-bit data into it, 3-28
BLANK signal, 3-58
BT478 color palette, 3-35
  description, 3-60
BUSFLT signal, 3-46
BUSRDY signal, 3-31
  see NOTREADY, 3-18, 3-33
BYTSEL[0:3] signals, 3-7, 3-20, 3-31

cabling, installation, 2-2, 2-9, 2-10, 2-23
cAMD signal, 3-47
CAS signals, 3-47
  CAS[0:3], 3-46, 3-53
CHSYNC signal, 3-38, 3-59, 3-61
COINT signal, 3-43, 3-53
column-address time, description, 3-46
comparators
  16-bit transfer (U30), 3-18
    generates the 16BTTR signal, 3-7
  4-bit I/O address (U12), 3-2, 3-23
  base address (U39), 3-6
CONFIG register, 3-47
connectors
  J2, emulator I/O, 3-62
  P4, DB9 video, 2-2, 3-36, 3-58, 3-61
CORDY signal, 3-46, 3-53, 3-54
CSYNC signal, 3-38, 3-59
customer support, B-6
customizing monitor timing, 2-24

DAT[0:3] signals, 3-47
DAT[0:31] signals, 3-34, 3-41, 3-46, 3-53
description, 3-12
relates to the TMS34020 LAD[0:31] bus, 3-12
DAT[0:4] signals, 3-46
DAT[0:7] signals, 3-41, 3-42
DAT[23,30,31], 3-48
DAT[28:31] signals, 3-11, 3-49, 3-60
DAT[5:31] signals, 3-46
DAT[5:7] signals, 3-59
data subcycle, description, 3-46
data transfer size screen, 2-16
DBUFF signal, 3-34, 3-41
description, 3-20
devices
25-MHz crystal oscillator (U72), generates 640 x 480 screen resolution, 3-58
60-MHz crystal oscillator (U71), generates 1024 x 768 screen resolution, 3-58
Bt478 color palette (U65), 3-35
SN74ALS153
4-to-1 dot clock multiplexer (U66), 3-58
U66, 3-10
SN74ALS240, SDBST register (U29), 3-9
SN74ALS244, U43, 3-18
SN74ALS245
8-bit transfer odd byte alignment buffer (U51), 3-20
I/O data transceiver (U45), 3-29
SN74ALS521
16-bit transfer comparator (U30), 3-7, 3-18, 3-54
base address comparator (U39), 3-6, 3-23, 3-31, 3-33
SN74ALS573
host address latch (U54), 3-5
host address latch (U59), 3-5
host address latch (U64), 3-5
SN74ALS652
U41, 3-18, 3-31, 3-34, 3-41
U42, 3-18, 3-31, 3-34, 3-41
U69, 3-18, 3-31, 3-34, 3-41
U70, 3-18, 3-31, 3-34, 3-41
SN74ALS996, 3-19
BASEP register (U6), 3-5
external data transceiver (U41), 3-7
external data transceiver (U42), 3-7
external data transceiver (U69), 3-7
external data transceiver (U70), 3-7
HINTF register (U7), 3-6
MAPAL (U5), 3-8
MAPAM (U4), 3-8
MAPBL (U3), 3-8
MAPBM (U2), 3-8
SN74F199
8-bit pixel output buffer (U57), 3-10
U44, 3-10
U44, U50, U52, U53, 3-59
U46–U49, 3-10, 3-59
U50, 3-10
U52, 3-10
U53, 3-10
SN74F244
U56, 3-61, 3-62
VGA passthrough buffer (U56), 3-35
SN74F245, high-byte blocking buffer (SP2), 3-20
SN74F51, U60, 3-10
SN74F85, U12, 3-2
SN74F861
U55, 3-11
VGA passthrough buffer (U55), 3-35
SN74FCT53, U61, 3-58
TIBPAL16L8
HARD PAL (U13), 3-10
RASDCCD PAL (U1), 3-48
SHADOW PAL (U25), 3-36
TIBPAL22V10
GSPHST PAL (U26), 3-18
HBS PAL (U28), 3-20
MAPEN PAL (U27), 3-25
MISC PAL (U8), 3-37, 3-59
SELECT PAL (U40), 3-23
TMS34082, description, 3-53
TMS44C251SD
(U31–U38), 3-51
VRAM (U31–U38), 3-45, 3-58
TMS44C256SD
DRAM (U17–U24), 3-52
DRAM (U17–U28), 3-45
dot clock frequencies
25 MHz (U72), 3-10
60 MHz (U71), 3-10
DOTCLKA signal, 3-59
DOTCLKB signal, 3-59
DRAM
   BANK1 selection, 3-46, 3-48
   TMS44C256SD, U17–U24, 3-45, 3-52
drivers, TIGA software, 2-29
dual monitor selection, 2-18

ED flag, description, 3-43
EMU[0:3] signals, 3-62
emulator I/O connector (J2), 3-62
error messages, B-2
   Abort, B-2
   Attempt to unload TIGA Driver failed!, B-3
   Non TIGA Driver installed in interrupt level 0x??, B-4
   Not enough memory to store parameters, B-4
   Required file tiga.cfg not found, B-4
   TIGA, B-2
   TIGA Driver already installed in interrupt level 0x??, B-4
   TIGA Driver is not installed!, B-5
   Timeout waiting for current TIGA command, B-5
EXPADD[0:2] signals, 3-18

frame buffer, description, 3-45

GSPHST PAL (U26), 3-7, 3-8, 3-31, 3-34
   ABEL source file, 3-19
description, 3-18
equation, 3-19
generated signals
   16BTTR, 3-7, 3-18
   HSTWCLK, 3-18
   LSTATE, 3-18, 3-31
   OEMAPA, 3-8
   OEMAPB, 3-8

HAND PAL (U14), 3-18
   ABEL source file, 3-54
description, 3-54
equation, 3-54
generated signals
   ALIGN, 3-20
   BYTSEL[0:3], 3-20
   DBUFF, 3-20
   HSB[0:3], 3-20
HBS[0:3] signals, description, 3-20
HCS signal, 3-18, 3-20, 3-23, 3-31, 3-33, 3-34, 3-41
description, 3-7
high-byte blocking buffer (SP2), 3-20
   enables 16-bit transfers, 3-34
HINT signal, description, 3-9
HINTF register (U7), 3-60
   16BITEN, bit 6, 3-7
   BASE32K, bit 0, 3-6
description, 3-6
   HRST (bit 4), 3-6
   reserved bits1–3, 3-6
   SDBENB, bit 7, 3-7
   SHADIS (bit 5), 3-6
horizontal frequencies, definition, 2-24
host address latches, U54, U64, U59, 3-5
host data latches, U41, U42, U69, U70, 3-18
host I/O base address screen, 2-14
host I/O registers, memory mapping, 3-2
host memory segment base address screen, 2-15
HRYD signal, 3-18, 3-31, 3-33, 3-41
HREAD signal, generated by the GSPHST PAL
   (U26), 3-7, 3-18
HREGS signal, 3-48
   enables HWSPACE, 3-49

HA[5:31] signals, description, 3-12

hardware requirements, 2-2
HBS PAL (U28), 3-41
   ABEL source file, 3-22
description, 3-22
equation, 3-22
generated signals
   ALIGN, 3-20
   BYTSEL[0:3], 3-20
   DBUFF, 3-20
   HBS[0:3], 3-20
HCS signal, 3-18, 3-20, 3-23, 3-31, 3-33, 3-34, 3-41
description, 3-7
high-byte blocking buffer (SP2), 3-20
   enables 16-bit transfers, 3-34
HINT signal, description, 3-9
HINTF register (U7), 3-60
   16BITEN, bit 6, 3-7
   BASE32K, bit 0, 3-6
description, 3-6
   HRST (bit 4), 3-6
   reserved bits1–3, 3-6
   SDBENB, bit 7, 3-7
   SHADIS (bit 5), 3-6
horizontal frequencies, definition, 2-24
host address latches, U54, U64, U59, 3-5
host data latches, U41, U42, U69, U70, 3-18
host I/O base address screen, 2-14
host I/O registers, memory mapping, 3-2
host memory segment base address screen, 2-15
HRYD signal, 3-18, 3-31, 3-33, 3-41
HREAD signal, generated by the GSPHST PAL
   (U26), 3-7, 3-18
HREGS signal, 3-48
   enables HWSPACE, 3-49
Index

HRST signal
   bit 4 of the HINTF (U7), 3-42
   description, 3-6

HSRBK signal, enables the SDBST register (U29), 3-9

HSREG register (U13), 3-42
   4BIT (bit 1), 3-10, 3-59
   8BIT (bit 0), 3-10, 3-59
   description, 3-10
   RATE (bit 2), description, 3-10
   see HARD PAL (U13), 3-58
   VGA (bit 3), 3-11, 3-59

HSTCTLH I/O register, internal to the TMS34020, 3-42

HSTCTLL I/O register
   internal to the TMS34020, 3-43
   INTIN (bit 3), 3-43
   MSGIN (bits 0–2), 3-43

HSTWCLK signal, 3-41
   generated by the GSPHST PAL (U26), 3-18, 3-34

HWRITE signal, generated by the GSPHST PAL
   (U26), 3-7, 3-18

HWSPACE signal, 3-11, 3-49, 3-51
   generated by the RASDCD PAL (U1), 3-48

I

I/O address comparator (U12), 3-28
I/O address locations, A-5
   W1, 2-4
I/O data transceiver (U45), 3-29
I/O memory map, 3-5
installation, 2-6
   cabling, 2-2
   jumpers
      W1, 2-4
      W4, 2-4
   monitor cables, 2-10
   PC cover, 2-11, 2-23
   PC mounting bracket, 2-9
   running tigaset, 2-12
   SDB20 preparation, 2-4
   surveying the environment, 2-3
   TMS34082, 3-54
   VGA passthrough cable, 2-9
   Windows 3.0 driver, 2-29
interfaces
   frame buffer/color palette, 3-58
   TIGA software, 1-4
   TMS34020-to-DRAM, 3-52
   TMS34020-to-memory, description, 3-46
   TMS34020-to-VRAM, 3-51
   video timing, A-4
   interrupt location, W4, 2-4, A-6
   interrupts
      how the host interrupts the TMS34020, 3-43
      how the TMS34020 interrupts a host, 3-43
      how the TMS34020 interrupts the TMS34082, 3-43
   INTIN signal, bit 3 of the HSTCTLL register, 3-43
   INTOUT signal, 3-43
      bit 7 of the HSTCTLL register, 3-43
   INTVECT signal, 3-48
   IOCLK signal, description, 3-26
   IOENB signal, 3-2, 3-19, 3-23, 3-25, 3-29
      description, 3-28
   IOR signal, 3-2, 3-23
      host I/O read strobe, 3-23
   IOREGS signal, description, 3-48
   IOSEL signal
      description, 3-28
      the I/O address comparator's output, 3-23
   IOW signal, 3-2, 3-36, 3-41
      host I/O read strobe, 3-23

J

jumpers
   W1, 2-4, A-5
   W4, 2-4, A-6

L

LAD[0:31] signals, description, 3-12
LAD[17:31] signals, 3-17
LAD[5:22] signals, 3-47
LCLK signal, 3-42, 3-62
LCLK1 signal, 3-53
LCLK2 signal, 3-53
LINT signal, 3-43
LOCADD[0:23] signals, description, 3-5
LOCADD[4:7] signals, 3-2
LOCRES signal, 3-42
LREADY signal, 3-46, 3-53, 3-54
LRESET signal, 3-11
  generated by the RESET PAL (U15), 3-6
LSTATE signal, 3-31, 3-33, 3-41
  generated by GSPHST PAL (U26), 3-18

M16 signal, 3-7
MAD[0:8] signals, 3-51
main menu, reinstall TIGA, 2-14
MAPAL register (U5)
  description, 3-8
  SN74ALS996, 3-8
MAPAM register (U4)
  description, 3-8
  SN74ALS996, 3-8
MAPBL register (U3)
  description, 3-8
  SN74ALS996, 3-8
MAPBM register (U2)
  description, 3-8
  SN74ALS996, 3-8
MAPCLK signal, 3-26
  generated by the MAPEN PAL (U27), 3-29
MAPEN PAL (U27), 3-29
  description, 3-25
  equation, 3-26
  generated signals
  BASEENB, 3-29
  HSRBK, 3-9
  IOCTLK, 3-26
  MAPREAD, 3-26
  STAIT, 3-26
  TIBPAL22V10, 3-25
mapping registers, 3-2
  MAPAL (U5), 3-8
  MAPAM (U4), 3-8
  MAPBL (U3), 3-8
  MAPBM (U2), 3-8
MAPREAD signal, generated by the MAPEN PAL (U27), 3-26
MEMCS16 signal, 3-7
memory mapping, host I/O registers, 3-2
MEMR signal, 3-23, 3-31
MEMSEL signal, 3-23, 3-31, 3-33
MEMW signal, 3-23, 3-33
Microsoft Windows, 2-29
MISC PAL (U8), 3-41, 3-60
  description, 3-59
  equation, 3-39
  generated signals
  CHSYNC, 3-38
  PALETRD, 3-37
  PALETWR, 3-37
  REGADD[0:2], 3-37
  SHADOK, 3-7, 3-36, 3-37
monitor cables, connecting, 2-10, 2-23
monitors
  customizing timing, 2-24
  dual monitor selection, 2-23
  dual monitor selection, 2-18
  single monitor selection, 2-23
  single monitor selection, 2-18
  video timing, A-4
mounting bracket
  installation, 2-9
  removal, 2-7
MSGIN signals, bits 0–2 of the HSTCTLL register, 3-43
multiplexed address/data bus, DAT[0:31], 3-46

N
noninterlaced video timing
  1024 x 768, A-4
  640 x 480, A-4
nonshadow mode, 3-60
NOTREADY signal, 3-18, 3-33, 3-41
  generated by the GSPHST PAL (U26), 3-31

O
odd-byte alignment buffer (U51), 3-20
OEMAP signal, 3-19
  enables MAP A mapping registers, 3-8
OEMAPB signal, 3-19
  enables MAP B mapping registers, 3-8
OL[0:3] signals, not used on SDB20, 3-60

P
P4, DB9 video connector, 3-58, 3-61
PALDIR signal, 3-23, 3-37, 3-41, 3-42, 3-59, 3-60
  description, 3-24
  generated by SELECT PAL (U40), 3-7
PALDOT signal, 3-10, 3-58
  description, 3-10
PALET signal, 3-48, 3-51
enables PEN, 3-48
PALETRD signal, 3-37, 3-38, 3-60
palette register addresses (REGADD[0:2]), 3-37
PALETWR signal, 3-37, 3-38, 3-60
description, 3-42
generated by the MISC PAL (U8), 3-41

PALs
  GSPHST (U26), 3-7, 3-8
equation, 3-19
  TIBPAL22V10, 3-18
HAND (U14), 3-53
buffers SDBENB, 3-18
description, 3-54
equation, 3-54
generates the 16XF signal, 3-7
HARD (U13), 3-11
PAL equation, 3-12
  TIBPAL16LB, 3-10
HBS (U28), 3-41, 3-46
enables host byte selects (HBS[0:3]), 3-7
equation, 3-22
  TIBPAL16LB, 3-20
MAPEN (U27), 3-29
equation, 3-26
generates HSBRK, 3-9
  TIBPAL22V10, 3-25
MISC (U8), 3-41
equation, 3-39
  TIBPAL22V10, 3-37
RASDCD (U1), 3-47
equation, 3-50
generated signals
  HWSPACE, 3-11
  PEN, 3-48, 3-49, 3-60
  RASO, 3-46, 3-48, 3-51
  RAS1, 3-46, 3-52
HREGS, 3-48
INTVECT, 3-48
IOREGS, 3-48
PALET, 3-48
RATE signal, 3-58
description, 3-9, 3-10
RCA bus configuration mode (RCM), 3-47
RCA[5:22] row/column address bus, 3-47
RCM bits, 3-47
read mask register (02h), 3-42
READ signal, 3-20, 3-23
REFRESH signal, 3-23
REGADD[0:2] signals, 3-37, 3-59
description, 3-38, 3-41
registers
  BASEP (U6), 3-13, 3-33
description, 3-5

PCADD[16:23] signals, 3-13
PCADD[16:31] signals, 3-16
PCADD[2:13] signals, 3-13, 3-15
connected to HA[5:16], 3-17
PCADD[20:23] signals, 3-5, 3-13, 3-18
PEN signal, 3-48, 3-60
enables MISC PAL (U8), 3-49
PGMD signal, 3-47
pinouts, video output connector, A-3
PIX4BIT signal, 3-59
description, 3-10
PIX8BIT signal, 3-59
description, 3-10
PIXEL[0:7] signals, 3-10
PIXRATE signal
  see RATE, 3-58
  see RATE (bit 2) of the HSREG, 3-10

R

RAS signals, 3-47, 3-48, 3-53
  RAS0, 3-46, 3-48, 3-51
  RAS1, 3-46, 3-52
RASDCD PAL (U1), 3-47, 3-51
description, 3-48
equation, 3-50
HREGS, 3-48
INTVECT, 3-48
IOREGS, 3-48
PALET, 3-48
 RATE signal, 3-58
description, 3-9, 3-10
RCA bus configuration mode (RCM), 3-47
RCA[5:22] row/column address bus, 3-47
RCM bits, 3-47
read mask register (02h), 3-42
READ signal, 3-20, 3-23
REFRESH signal, 3-23
REGADD[0:2] signals, 3-37, 3-59
description, 3-38, 3-41
registers
  BASEP (U6), 3-13, 3-33
description, 3-5

Index-6
HINTF (U7), 3-42, 3-60
16BITEN (bit 6), 3-7
BASE32K (bit 0), 3-6
description, 3-6
HST (bit 4), 3-6
reserved bits 1–3, 3-6
SDBENB (bit 7), 3-7
SHADIS (bit 5), 3-6
HSREG (U13), 3-42
4BIT (bit 1), 3-10
8BIT (bit 0), 3-10
description, 3-10
RATE (bit 2), 3-10
VGA (bit 3), 3-11, 3-61
HSTCTLH, internal to the TMS34020, 3-42
HSTCTLH, internal to the TMS34020, 3-43
MAPAL (U5), description, 3-8
MAPAM (U4), description, 3-8
MAPBL (U3), description, 3-8
MAPBM (U2), description, 3-8
SDBST (U29), 3-43
4BIT (bit 1), 3-9
8BIT (bit 0), 3-9
description, 3-9
HINT (bit 7), 3-9
RATE (bit 2), 3-9
VGA (bit 3), 3-9
reinstall TIGA option, 2-14
description, 2-13
removal
PC cover, 2-6
PC mounting bracket, 2-7
RESDRV signal, 3-42
RESET PAL (U15), 3-11, 3-42
generated signals
LOCRS, 3-42
LRESET, 3-6
resetting the SDB20, 3-42
ROWCOL[0:8] signals, 3-46
RST signal, bit 7 of the TMS34020's internal
HSTCTLH register, 3-42

S

SBHE signal, 3-18, 3-20, 3-34
SCLK signal, 3-51, 3-58
screens
change monitor option, 2-26
change TIGA display mode, 2-21
SDB20 installation, B-2
SDB20 information, 2-22
TIGA installation
correcting the monochrome display problem, 2-20
data transfer size, 2-16
default directory, 2-17
defined monitors, 2-19
determining the monitor system, 2-19
host I/O base address, 2-14
host memory segment base address, 2-15
monitor configuration, 2-18
TIGA environment, 2-17
TIGA setup utility screen, main menu, 2-13
user-defined monitor timing
down arrow key, 2-26
escape key, 2-26
F10, 2-26
F2, 2-26
F4, 2-26
F9, 2-26
minus key, 2-26
plus key, 2-26
up arrow, 2-26
Windows setup, 2-30
display driver, 2-32
display type, 2-31

SD[0:15] signals, 3-34
description, 3-20
SD[0:7] signals, 3-28
description, 3-20
SDB20
backend section logic, 3-58
block diagram, 1-3
board layout, 3-3
description, 1-1
frame buffer, (U31–U38), 3-51
I/O memory map, 3-5
installation preparation, 2-4
installation procedures, 2-6
jumpers
W1, 2-4
W4, 2-4
overview, 1-2
system specifications, A-2
TIGA software drivers, 2-28
list of TIGA video drivers, 2-28
video output connector (P4), 3-58
Index

video timing, A-4

SDBENB signal, 3-18, 3-31, 3-33, 3-54
description, 3-7

SDBST register (U29), 3-43
4BIT (bit 1), 3-9
8BIT (bit 0), 3-9
description, 3-9
HINT, bit 7, 3-9
RATE (bit 2), 3-9
VGA (bit 3), 3-9

SELECT PAL (U40), 3-6, 3-37, 3-59, 3-60
description, 3-23
equation, 3-25
generated signals
  HCS, 3-7, 3-31
  IOENB, 3-2, 3-28
  PALDIR, 3-7
  READ, 3-18, 3-31, 3-33

SF signal, 3-51, 3-53

SHADENB signal, 3-35, 3-41, 3-60
see SHADIS

SHADIS signal, 3-37
  bit 5 of the HINTF register, 3-35, 3-36, 3-41, 3-59
description, 3-6

SHADOK signal, 3-7, 3-21, 3-23, 3-36, 3-37, 3-41, 3-60
shadow mode, 3-59, 3-60

SHADOW PAL (U25), 3-36, 3-37, 3-41
equation, 3-37
generated signals, SHADOW, 3-7, 3-36, 3-41

shadow palette access, definition, 3-35

SHADOW signal, 3-7, 3-60
generated by the SHADOW PAL (U25), 3-36, 3-41

shift registers
  4-bit/pixel, 3-10
  8-bit/pixel, 3-10

signals
  16BADD[0:6], 3-18
  16BSEL, 3-7
    generated by 16-bit transfer comparator (U30), 3-18
  16BTTR, 3-7, 3-34
    generated by GSPHST PAL (U26), 3-18
  16SEL, 3-18
  16XF, 3-18, 3-54
    generated by HAND PAL (14), 3-7
  32KBIT, 3-31, 3-33
description, 3-31, 3-33
  4BIT, 3-59
  8BIT, 3-59
  ALIGN, 3-20
  ALTCH, 3-37, 3-42, 3-53, 3-60
  BASE32K, 3-6, 3-16, 3-31, 3-33
    bit 0 of the HINTF (U7), 3-13
  BASEENB, enable the BASEP register, 3-29
  BLANK, 3-58
  BUSFLT, 3-46
  BUSRDY, 3-31
    see NOTREADY, 3-18, 3-33
  BYTSEL[0:3], 3-7, 3-20, 3-31
    byte selects for U41, U42, U69, U70, 3-7
  CAS, 3-47
    CAS[0:3], 3-46, 3-53
  CHSYNC, 3-38, 3-59, 3-61
  COINT, 3-53
  CORDY, 3-46, 3-53, 3-54
  CSYNC, 3-38, 3-59
  DAT[0:3], 3-47
  DAT[0:31], 3-34, 3-41, 3-46, 3-53
  DAT[0:4], 3-46
  DAT[0:7], 3-41, 3-42
  DAT[23,30,31], 3-48
  DAT[28:31], 3-11, 3-49, 3-60
  DAT[5:31], 3-46
  DAT[5:7], 3-59
  DBUFF, 3-34, 3-41
description, 3-20
  DOTCLOCKA, 3-59
  DOTCLOCKB, 3-59
  ED, description, 3-43
  EMU[0:3], 3-62
  EXPADD[0:2], 3-18
  HA[5:31], description, 3-12
  HBS[0:3], description, 3-20
  HCS, 3-7, 3-18, 3-20, 3-23, 3-31, 3-33, 3-34, 3-41
  HDST assertion, 3-31
  HOE, 3-34, 3-41
  HRDY, 3-18, 3-31, 3-33, 3-41
  HREAD, generated by the GSPHST PAL (U26), 3-7, 3-18, 3-31
  HREGS, 3-48, 3-49
  HRSRBK, enables the SDBST register (U29), 3-9
  HSTWCLK, 3-41
generated by the GSPHST PAL (U26), 3-18, 3-34
HWRITE, 3-41
  generated by the GSPHST PAL (U26), 3-7, 3-18, 3-33, 3-34
HSPACE, 3-11, 3-48, 3-49, 3-51
INTIN, bit 3 of the HSTCTLL register, 3-43
INTOUT, bit 7 of the HSTCTLL register, 3-43
INTVECT, 3-48
IOCLK, description, 3-26
IOENB, 3-2, 3-19, 3-23, 3-25, 3-29
description, 3-28
IOR, 3-2, 3-23
  host I/O read strobe, 3-23
IOREGS, 3-48
IOSEL, 3-2
description, 3-28
  the I/O address comparator's output, 3-23
IOW, 3-2, 3-36, 3-41
  host I/O read strobe, 3-23
LAD[17:31], 3-17
LAD[5:22], 3-47
LCLK, 3-42
LCLK0, 3-62
LCLK1, 3-53
LCLK2, 3-53
LOCADD[0:23], description, 3-5
LOCADD[4:7], 3-2
LOCADD[8:9], 3-2
LOCRS, 3-42
LRDY, 3-46, 3-53, 3-54
LRESET, 3-6, 3-11
LSTATE, 3-31, 3-33, 3-41
generated by GSPHST PAL (U26), 3-18
M16, 3-7
MAD[0:8], 3-51
MAPCLK, 3-26, 3-29
MAPREAD, generated by the MAPEN PAL (U27), 3-26
MEMCS16, 3-7
MEMR, 3-23, 3-31
MEMSEL, 3-23, 3-31, 3-33
MEMW, 3-23, 3-33
MSGIN, bits 0–2 of the HSTCTLL register, 3-43
NOTREADY, 3-18, 3-41
generated by the GSPHST PAL (U26), 3-31, 3-33
OEMAPA, 3-19
  enables MAP A mapping registers, 3-8
OEMAPB, 3-19
  enables MAP B mapping registers, 3-8
OL[0:3], 3-60
PALDIR, 3-7, 3-23, 3-37, 3-41, 3-42, 3-59, 3-60
PALDOT, 3-10, 3-58
description, 3-10
PALET, 3-48, 3-51
  enables PEN, 3-48
PALETRD, 3-37, 3-38, 3-60
PALETWR, 3-37, 3-38, 3-60
description, 3-42
generated by the MISC PAL (U8), 3-41
PBLANK, 3-10
generated by U57, 3-10
PCADD[0,1], 3-13, 3-17
PCADD[0:23], description, 3-5
PCADD[14], 3-8, 3-13, 3-14, 3-15, 3-16
PCADD[15], 3-6, 3-13, 3-16
PCADD[16:19], 3-6
PCADD[16:23], 3-13
PCADD[16:31], 3-16
PCADD[2:13], 3-13, 3-15
c connected to HA[5:16] host address pins, 3-17
PCADD[20:23], 3-5, 3-13, 3-18
PEN, 3-48, 3-49, 3-60
generated by the RASCD PAL (U1), 3-38
PGMD, 3-47
PIX4BIT, 3-59
description, 3-10
PIX8BIT, 3-59
description, 3-10
PIXEL[0:7], 3-10
description, 3-10
PIXRATE, see RATE, 3-10, 3-58
RAS, 3-47, 3-53
description, 3-48
  RAS0, 3-46, 3-48, 3-51
  RAS1, 3-46, 3-52
RATE, 3-58
see PIXRATE, 3-10
RCA[1:9], description, 3-47
RCA[5:22], 3-47
READ, 3-20, 3-23, 3-29
generated by the SELECT PAL (U40), 3-18, 3-31, 3-33
REFRESH, 3-23
REGADD[0:2], 3-37, 3-59
description, 3-38, 3-41
RESDRV, description, 3-42
ROWCOL[0:8], 3-46
RST, bit 7 of the TMS34020's HSTCTLH register, 3-42
SBHE, 3-18, 3-20, 3-34
SCLK, 3-51, 3-58
Index

SD[0:15], 3-20, 3-34
SD[0:7], 3-20, 3-28
SDBENB, 3-18, 3-31, 3-33, 3-54
SF, 3-51, 3-53
SHADENB, 3-7, 3-35, 3-41, 3-60
SHADIS, 3-59
  bit 5 of the HINTF register (U7), 3-35, 3-36, 3-37
SHADOK, 3-7, 3-21, 3-23, 3-37, 3-41, 3-60
generated by the MISC PAL (U8), 3-36
SHADOW, 3-7, 3-36, 3-41, 3-60
SIZE16, 3-47
SMEMR, 3-23
SMEMW, 3-23
SRLOAD4, 3-58, 3-59
SRLOAD8, 3-58, 3-59
STAIT, 3-26
TMS34020, LINT1, 3-43
TMS34082, COINT, 3-43
TR/QE, 3-60
UWE, 3-37, 3-42
VCLK, 3-58
VGA, bit (3) of the HSREG (U13), 3-35, 3-59
VGABL, description, 3-35
VGACLK, description, 3-35
VGAENB, 3-24, 3-41, 3-59
  see VGA of the HSREG, 3-11
VGAHS, 3-61
description, 3-36
VGASOURCE, description, 3-35
VGAVS, 3-61
description, 3-36
VSYNC, 3-61
WE, 3-11, 3-51, 3-53, 3-60
single monitor selection, 2-18
SIZE16 signal, 3-47
SMEMR signal, 3-23
SMEMW signal, 3-23
software files
  AUTOEXEC.BAT, 2-18
tigaauto.bat, 2-18
tigaset.exe, 2-12
SP2, high-byte blocking buffer, 3-20, 3-34
SRAM module
description, 3-55
the TMS34082 bus architecture, 3-56
SRLOAD4 signal, 3-58, 3-59
SRLOAD8 signal, 3-58, 3-59
STAIT signal, 3-26

surveying the environment, 2-3
switch settings
  for interface registers, table of SW1 settings, 2-5
  for interrupt levels, table of SW2 settings, 2-6
sync widths, definition, 2-24
synchronization
  backporch, 2-24
  sync widths, 2-24
system requirements, hardware, 2-2
system specifications, A-2

T

TIGA board diagnostics option, description, 2-13
TIGA error messages
  Abort, B-2
  Attempt to unload TIGA Driver failed!, B-3
  Non TIGA Driver installed in interrupt level 0x??, B-4
  Not enough memory to store parameters, B-4
  Required file tiga.cfg not found, B-4
  TIGA Driver already installed in interrupt level 0x??, B-4
  TIGA Driver is not installed!, B-5
  Timeout waiting for current TIGA command, B-5
TIGA information screen for the SDB20, 2-22
TIGA installation screen
correcting the monochrome display problem, 2-20
data transfer size, 2-16
defined monitors, 2-19
determining the monitor color, 2-19
host I/O base address, 2-14
host memory segment base address, 2-15
main menu, 2-13
monitor configuration, 2-18
TIGA software drivers, 2-28
  list of TIGA video drivers, 2-28
tigaauto.bat, 2-18
tigamode, 2-21
tigaset.exe, 2-12
timing
  host-to-SDB20 16-bit memory write cycle, 3-32
  host-to-SDB20 8-bit memory read cycle, 3-30
  host-to-SDB20 I/O cycle, 3-28
  VGA passthrough cycle, 3-40
timing diagrams, typical video timing, 2-24
TMS34020

Index-10
CAMD, 3-47
CHSYNC, 3-59, 3-61
CONFIG register, RCM bits, 3-47
CSYNC, 3-59
description, 1-5
HA[5:31], description, 3-12
HINT pin, 3-9, 3-43
HSTCTLL I/O register, 3-42
HSTCTLL I/O register, 3-43
  INTIN bit (3), 3-43
  MSGIN (bits 0–2), 3-43
  MSGIN bits (0–2), 3-43
INTIN signal, bit 3 of the HSTCTLL register, 3-43
INTOUT, bit 7 of the HSTCTLL register, 3-43
LAD[0:31], description, 3-12
LAD[5:22], 3-47
LCLK1, 3-53
LCLK2, 3-53
local memory map, 3-45
LRDY, 3-54
PGMD signal, 3-47
RCA[1:9], description, 3-47
RCM, 3-47
SF, 3-51, 3-53
SIZE16 signal, 3-47
VSYNC, 3-61
WE, 3-53
TMS34082
  CINT signal, 3-43, 3-53
  CORDY signal, 3-46, 3-53, 3-54
description, 1-6, 3-53
ED flag, description, 3-43
installation, 3-54
SRAM bus architecture, 3-56
SRAM module, description, 3-55
TR/QE signal, 3-60

U

U10, TMS34082, 3-53
U12
  4-bit I/O address comparator, 3-2, 3-23
  SN74F85, 3-2, 3-23
U13
  HARD PAL, 3-10
  TIBPAL16L8, 3-10
U14, HAND PAL, TIBPAL20R4A, 3-54
U17–U24, TMS44C256SD, DRAM, 3-45, 3-52
U2
  MAPBM register, 3-8
  SN74ALS996, 3-8
U27
  MAPEN PAL, 3-25
  TIBPAL22V10, 3-25
U28
  HBS PAL, 3-7, 3-20
  TIBPAL22V10, 3-20
U29
  SDBST register, 3-9
  SN74ALS240, 3-9
U3
  MAPBL register, 3-8
  SN74ALS996, 3-8
U30
  16-bit transfer comparator, 3-7, 3-18, 3-54
  SN74ALS521, 3-18, 3-54
U31–U38
  TMS44C251SD, 3-58
  VRAM, TMS44C251SD, 3-58
U39
  base address comparator, 3-6, 3-23, 3-31, 3-33
  SN74ALS521, 3-6, 3-23, 3-31, 3-33
U4
  MAPAL register, SN74ALS996, 3-8
  MAPAM register, 3-8
  SN74ALS996, 3-8
U40
  SELECT PAL, 3-23
  TIBPAL22V10, 3-23
U41, SN74ALS652, 3-7, 3-18, 3-31, 3-34, 3-41
U42, SN74ALS652, 3-7, 3-18, 3-31, 3-34, 3-41
U43, SN74ALS244, 3-18
U44
  8-bits/pixel shift registers, 3-10, 3-59
  SN74F199, 3-10, 3-59
U45, SN74ALS245, I/O data transceiver, 3-29
U46–U49
  4-bit/pixel shift registers, 3-10
  SN74F199, 3-10, 3-59
U5
  MAPAL register, 3-8
U50
  8-bits/pixel shift registers, 3-10, 3-59
  SN74F199, 3-10
U51, SN74F245, 3-20
U52
  8-bits/pixel shift registers, 3-10, 3-59
  SN74F199, 3-10

Index-11
U53  8-bits/pixel shift registers, 3-10, 3-59
     SN74F199, 3-10, 3-59
U54  host address latch, 3-5
     SN74ALS573, 3-5
U55  SN74F861, 3-11, 3-35
     VGA passthrough buffer, 3-11, 3-35
U56  SN74F244, 3-35, 3-61, 3-62
     VGA passthrough buffer, 3-35, 3-61, 3-62
U57  8-bit output buffer, 3-10
     SN74F199, 3-10
U59  host address latch, 3-5
     SN74ALS573, 3-5
U6   BASEP register, 3-5
     SN74ALS996, 3-5
U60  4-bit/pixel output buffer, 3-10
     SN74F51, 3-10
U61  inverting synchronizer, 3-58
     SN74FCT53, 3-58
U64  host address latch, 3-5
     SN74ALS573, 3-5
U65  Bt478, 3-60
     color palette, 3-60
U66  4-to-1 dot clock multiplexer, 3-10, 3-58
     SN74ALS153, 3-10, 3-58
U69, SN74ALS652, 3-7, 3-18, 3-31, 3-34, 3-41
U7   HINTF register, 3-6
     SN74ALS996, 3-6
U70, SN74ALS652, 3-7, 3-18, 3-31, 3-34, 3-41
U71  60-MHz crystal oscillator, 3-58
     generates 1024 x 768 screen resolution, 3-58
U72  25-MHz crystal oscillator, 3-58
     generates 640 x 480 screen resolution, 3-58

V
VCLK signal, 3-58
vertical frequencies, definition, 2-24
VGA
     connecting the passthrough cable, 2-9, 3-35
     enabling passthrough, 3-35
     passthrough, description, 3-34
     passthrough buffers (U55,U56), 3-11, 3-35
VGA passthrough PALs
     description, 3-36
     GSPHST (U26), 3-36
     HBS (U28), 3-36
     MISC (U8), 3-36, 3-37
     SELECT (U40), 3-36
     SHADOW (U25), 3-36
VGA signal, 3-59
     bit 3 of the HSREG register (U13), 3-35, 3-61
     description, 3-9, 3-11
VGABL signal, description, 3-35
VGACLK signal, description, 3-35
VGAENB signal, 3-24, 3-41, 3-59
     see VGA of the HSREG, 3-11
VGAHS signal, 3-61
     description, 3-36
VGASOURCE signal, description, 3-35
VGAVS signal, 3-61
     description, 3-36
video connector (P4), 3-58, 3-61
     pinouts, A-3
video timing, A-4
     noninterlaced
     1024 x 768, A-4
     640 x 480, A-4
VRAM
     BANK0 selection, 3-46, 3-48
     TMS44C251SD, 3-58
VSYNC signal, 3-61

W
wait state insertion, 3-31
WE signal, 3-11, 3-51, 3-53, 3-60
Windows 3.0 driver installation, 2-29
Windows setup screen, 2-30
     display driver, 2-32
     display type, 2-31