PCI 9050RDK

Development Kit Manual

Version 1.2

January 28, 1998

Product Sales: 1-800-759-3735
Fax: 1-408-774-2169
Email: info@plxtech.com
Web and FTP Site: //www.plxtech.com

PLX TECHNOLOGY®

Table of Contents

Preface vii

Document Organization vii

Style Conventions viii

Contacting PLX viii

Chapter 1 1

PCI 9050RDK User's Guide 1

Who Should Use the 9050RDK? 3

Late Breaking News 3

Hardware Installation 4

PLX 9050RDK Windows 95 Bus Driver 7

PCI Exploration Programs 15

Configuration Issues 19

Windows 95 Tips 22

Troubleshooting 23

Chapter 2 25

PCI 9050RDK Hardware Manual 25

PCI Interface 26

PCI 9050 Target Interface Chip 26

SRAM Subsystem 30
ISA Subsystem 32
Daughter-Card Connector 39
Test Headers and Prototype Area 40
Additional Development Resources Provided by PLX 42

Chapter 3 43

PCI 9050RDK Schematics 43

Preface

This manual provides all the information you need to use the PCI 9050RDK Development Kit.

Document Organization

This manual is organized into the following chapters for easy reference:

- Chapter 1, “PCI 9050RDK User’s Guide,” discusses how to use the PCI 9050RDK. In particular, it provides installation information and instructions for piggybacking ISA comm cards and other ISA cards.
- Chapter 2, “PCI 9050RDK Hardware Manual,” describes each subsystem found on the RDK and provides suggestions for using the tools and circuitry included as a starting point for a new design.
- Chapter 3, “PCI 9050RDK Schematics,” provides schematics of the PCI 9050RDK for easy reference.

The PLXMon software used with this product is documented separately, in the *PLXMon User’s Guide*.

Style Conventions

The following styles are used in this manual:

- Commands, file names, keyboard keys, text you manually enter, and URL paths are in `Courier type` (for example, `DEBUG.EXE` or `http://www.plxtech.com`).
- Variables and parameters are in ***`Courier italic type`*** (for example, ***`hbuf`*** or ***`x`***).
- Emphasized text, names of diskettes, and names of publications are in *italic type* (for example, *PLX 9050RDK Distribution Disk*).
- Screen references are in **Arial Bold type** (for example, "Choose **OK**").

Contacting PLX

We want to hear from you! If you have comments, corrections or suggestions please let us know.

PLX Technology, Inc.
390 Potrero Ave
Sunnyvale, CA 94086

Phone	(408) 774-9060
FAX	(408) 774-2169
Email	apps@plxtech.com
Website	http://www.plxtech.com

Chapter 1

PCI 9050RDK User's Guide

Welcome to the PLX 9050RDK (Resource Development Kit). This chapter is the primary companion to the kit—it will help you get the most out of your kit in the shortest amount of time. Use the information in this chapter to

- Install the 9050RDK card hardware in your computer
- Piggyback an ISA comm card to the 9050RDK card (the kit includes either a modem card or a serial card) and install the PLX 9050RDK Windows 95 bus driver.
- Piggyback any ISA card on the 9050RDK
- Use PLX PCI software to explore and configure the PCI 9050 and piggybacked ISA cards
- Troubleshoot problems you may have with the PCI 9050RDK

The 9050RDK offers PCI computer engineers hardware and software to learn about the PLX PCI 9050 interface chip, to build PCI-based projects, and test functional feasibility of migrating an existing ISA product to PCI—all quickly and easily.

The key features of the 9050RDK card include the PLX PCI 9050 interface chip and the ISA 8/16 connector. You can plug most ISA cards into the 9050RDK ISA connector; and, with the 9050RDK-exploration software, configure the PCI 9050 and test the functionality of the ISA card. Once hardware functionality is satisfactory, you can concentrate on updating your driver software.

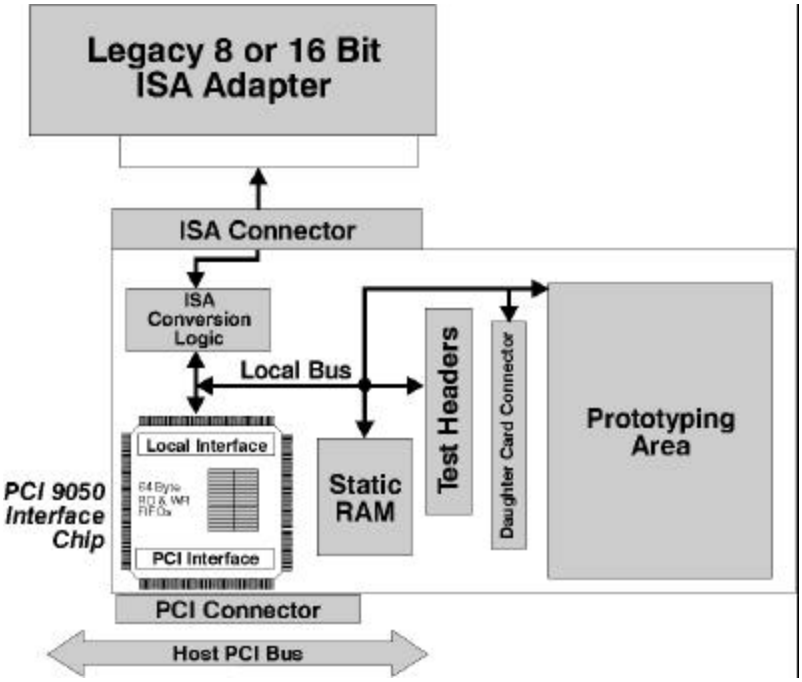


Figure 1-1. Major Functions of PCI 9050RDK

Most piggybacked ISA cards are fully functional without hardware changes. However, if your ISA board has special interface requirements, you can add custom hardware to the prototyping area of the 9050RDK. In particular, the 9050RDK is not designed to support the following:

- ISA boards that require refresh cycles
- ISA bus masters or DMA adapters (the PCI 9050 does not have PCI bus master capabilities)

The kit includes a standard ISA comm card (either a modem card or a serial card) and a PLX RDK bus driver. These components demonstrate a successful migration from ISA to PCI.

Aside from ISA to PCI migration, you can use the 9050RDK to learn more about the PCI 9050 interface chip or build a PCI-based prototype. The 9050RDK card provides a large daughter-card connector, along with a generous prototyping area and on-board RAM. On-board headers provide wiring access to all local-side pins of the PCI 9050 chip.

For all your uses of the 9050RDK, you will find the PCI exploration software included with the kit to be especially useful. This software enables you to read and write all physical memory and I/O in your system. It also provides easy access to all PCI registers on all PCI devices, as well as local registers on the PLX family PCI devices, including the PCI 9050. The software comes in DOS and Windows 95 versions.

Who Should Use the 9050RDK?

The 9050RDK is a PCI engineering toolkit. It is targeted for computer engineers investigating the PCI 9050 interface chip as part of a commercial product. You are a good candidate if you are interested in converting an existing ISA product to PCI or in designing a new high performance PCI (slave-only) product.

To use this product, you or your team should be able to

- Use hardware tools (such as logic analyzers and oscilloscopes)
- Design and build interface logic, if necessary
- Write or modify driver code
- Use command-line style programs (such as the DOS program, `DEBUG.EXE`)

Late Breaking News

Be sure to read any addenda and errata to the 9050RDK. These documents discuss important changes, corrections, and improvements to the kit. The PLX website (`http://www.plxtech.com`) may contain important late breaking news as well.

Hardware Installation

This section will help you set up the 9050RDK.

Standard Installation

Standard installation is the best starting point for all uses of the 9050RDK. This section describes how to install the 9050RDK card along with the communications card that is furnished in the kit.

Once you have followed the standard installation procedure, it is recommended that you go to the PCI Exploration Programs (PLXMon or PLXMon95), on page 15. Follow this procedure to install and use the PLX PCI Exploration Software.

Your 9050RDK card should not require any special jumper settings for general access to the PCI 9050. However, if you have problems, refer to Chapter 2, "PCI 9050RDK Hardware Manual," for jumper settings.

- **Install the 9050RDK card with the ISA Communications card in your system:**

  1. Carefully piggyback the ISA comm card onto the ISA connector of the 9050RDK card. Note the following:
     - The component side of ISA cards is opposite from PCI cards
     - You will not be able to replace the cover of your computer while an ISA card is piggybacked on the 9050RDK card
     - If your computer case interferes with external cabling to the ISA card, you may want to remove the motherboard from the case
  2. Connect appropriate external cabling (by way of phone line or serial cable) to a remote computer.
  3. Use safe and reasonable procedures to open your computer and install the 9050RDK card with the ISA communications card in an available PCI slot.
  4. Power on your computer and boot either DOS or Windows 95.
  5. **Systems running Windows 95**—If this is the first time a PCI 9050 is being installed on your system, your system's PCI bus enumerator will discover that a new PCI card has been installed. You will be prompted to install a driver for the new card—choose ***driver from disk provided by hardware manufacturer***.
  6. Next, you will be asked to ***install from disk***. Choose copy manufacturers files from A:_____. This will install all the files from the disk. (Do not specify, just a single file. The Windows O.S. will find a communication port and install the software for the communications port).

You should now be able to

- Use the PLX PCI programs to explore the PCI 9050 and the 9050RDK memory (refer to the section, "PCI Exploration Programs," on page 15 and the *PLXMon User's Guide*)
- Build a PCI-based prototype circuit
- Test the ISA comm card on the 9050RDK
- Piggyback any other ISA card on the 9050RDK (refer to the section, "Piggybacking Any ISA Card," on page 7)

Once the ISA comm card is piggybacked to the 9050RDK card, you can install software to test your setup. Table 1-1 lists the installation choices.

Table 1-1. Installation Choices

Install	Comments
PLXMon	PCI exploration software (DOS or Windows 95 versions). Refer to the section "PCI Exploration Programs," on page 15 and the PLXMon User's Guide
PLX RDK Bus Driver	This Windows 95 driver allows you to use popular applications with the piggybacked comm card as if the card were plugged into a standard ISA slot. Refer to the section "PLX 9050RDK Windows 95 Bus Driver," on page 7.

Piggybacking Any ISA Card

The 9050RDK is designed to provide full functionality for most ISA cards. However, if extra interface hardware is required, you can add it in the 9050RDK prototyping area. In particular, the 9050RDK is *not* designed to support

- ISA boards that require refresh cycles
- ISA bus masters or DMA adapters (the PCI 9050 does not have PCI bus master capabilities)

♦ **To piggyback any ISA card:**

1. Follow the steps outlined in the section, "Standard Installation," on page 4.
2. Power off your computer.
3. Carefully piggyback your ISA card onto the ISA connector of the 9050RDK card. Note the following:
   - The component side of ISA cards is opposite from PCI cards
   - You will not be able to replace the cover of your computer while an ISA card is piggybacked on the 9050RDK card
   - If your computer case interferes with external cabling to the ISA card, you may want to remove the motherboard from the case
4. Power on your computer and boot DOS or Windows 95.

Once your ISA card is piggybacked to the 9050RDK card, you can install PCI exploration software for DOS or Windows 95. Refer to the section, "PCI Exploration Programs," on page 15 and the *PLXMon User's Guide*.

PLX 9050RDK Windows 95 Bus Driver

This section discusses information related to the Windows 95 bus driver for the PLX 9050RDK.

Driver Overview

The PLX RDK bus driver is a Windows 95 VxD (virtual device driver) that logically connects a piggybacked comm card to the rest of the system. Programmers call this type of driver a "bus enumerator"—it programs the PCI 9050, and enumerates the one-and-only device on the 9050 local bus. Using the enumerator technique allows standard device drivers to be used with the comm card. (An alternative is to modify the device driver to directly program the PCI 9050.)

Only two files are needed for the complete driver package—`PLXRDK.VXD` and `PLXRDK.INF`. The first file is the actual PLX RDK bus driver, and the second guides the Windows 95 installer program.

You do not have to set the port address of the comm card as long as it is one of the four standard COM addresses. The comm card can be set to use any interrupt—all interrupts are `OR`'d on the RDK ISA connector into one line of the PCI 9050.

Note: Windows 95 handles copying of the `PLXRDK.INF` file differently, depending on which version of Windows 95 you are running (refer to the note under Step 3 in "Install the 9050RDK card with the ISA Communications card in your system" on page 5 to determine which version is running on your system):

- **Windows 95**—Renames the installer file, `PLXRDK.INF`, when it copies the file from the *PLX 9050 Distribution Disk* to the `\WINDOWS\INF` folder. The file is renamed as `OEM`*n*`.INF`, where *n* is the *n*th OEM driver installation.
- **Windows 95 OSR 2**—Copies the installer file, `PLXRDK.INF`, from the *PLX 9050 Distribution Disk* to the `\WINDOWS\INF\OTHER` folder.

Note: PLX has seen instances where the enclosed PLX PCI 9050 Windows95 driver (PLXSDK.VXD) does not operate with Windows95 OSR2. The following is suggested.

1. Windows95 OSR2 is not supported. Customers are advised to use a third-party software solution from KRFTech. You have received their software and manuals in the kit. The KRFTech solution is a device driver development tool kit. The same driver will work on both Windows95 and WindowsNT. KRFTech can be contacted at http://www.krftech.com.
2. The source code for the Windows 95 driver (PLXSDK.VXD) is enclosed in the PLX PCI 9050RDK package and may be modified to support Windows95 OSR2.

♦ PLX RDK bus driver tasks (simplified):

1. Negotiate with the Windows Configuration Manager for an eight-byte range of standard ISA COM addresses (maps above the ISA space if all standard addresses are taken)
2. Program the PCI 9050 for local I/O access, mapping the range of local I/O to the standard COM address
3. Probe standard COM ranges of the PCI 9050 local I/O space for an 8250 Universal Asynchronous Receiver Transmitter (UART)

Contact PLX if you need a copy of the PLX RDK bus driver source code for use or reference.

Install, Check, Change or Remove the Driver

This section describes how to install, check, change, or remove the Windows 95 driver for the ISA comm card of the 9050RDK. Refer to Table 1-2 below to perform the desired task.

Table 1-2. What to Do when You Want to Install, Check, Change or Remove the Driver

Task	Comments
Install	If you boot Windows 95 with the 9050RDK card installed, and a dialog box appears stating the system found new hardware, install the driver using the method described in the section, "Installing when Windows 95 Finds New PCI Hardware," on page 10. If Windows 95 boots normally with the 9050RDK card installed, and you have never installed the 9050RDK bus driver, chances are you were prompted about the new PCI card hardware in a previous session, and you chose not to install the driver. In this case, use the method described in the section, "Installing after Previously Choosing Not to Install a Driver," on page11 to install the driver.
Check	If you are not sure the driver was properly installed, you can verify the installation as described in the section, "Verifying the Driver Installation," on page 11.
Change	If necessary, you can change the driver or examine the driver properties (file name, provider and version) and the resource allocations (port, memory and interrupt) of the device as described in the section, "Driver Properties, Device Resources, and Changing the Driver," on page 12.
Remove or Reinstall	You can remove or reinstall the driver, as described in the section, "Removing or Reinstalling the Driver," on page 13.
Installation problems?	If you are still having problems installing the driver, try the steps outlined in the section, "If You Have Problems Installing the Driver," on page 14.

Installing when Windows 95 Finds New PCI Hardware

If, during the boot process, Windows 95 displays a box saying it found a PCI card, followed by a "New Hardware Found" dialog box ("Update Device Driver Wizard" dialog box in Windows 95 OSR 2), it indicates that the Windows 95 internal configuration database does not know of a driver to associate with the Vendor ID and Device ID of the PCI 9050. A dialog box pops up, prompting you to install a driver for Windows 95 to associate with the card.

♦ To install the PLX RDK bus driver—Windows 95:

1. Insert the *PLX 9050RDK Distribution Disk* into the floppy drive of your computer.
2. Choose **Driver from disk provided by hardware manufacturer**.
3. Choose **OK**.
4. Choose **OK** in the **Install From Disk** dialog box (or browse, as necessary, to find `PLXRDK.INF`).

♦ To install the PLX RDK bus driver—Windows 95 OSR 2:

1. Insert the *PLX 9050RDK Distribution Disk* into the floppy drive of your computer.
2. Choose **Next** to search for the `PLXRDK.INF` driver. Windows 95 displays a confirmation window when it finds what it believes to be the correct driver.
3. Choose **Finish** to use this driver.
4. Choose **Other Locations** if this is not the correct driver, to open a browse window to search for it manually.

Your system should show that it is adding a communications port to the system, then continue to boot normally.

You can now use any application for the piggybacked ISA comm card that you would use when the card is plugged into an ISA slot.

Some notes about driver installation:

- If you are prompted to restart your computer after installing the driver, the PLX RDK bus driver failed to find a comm card piggybacked to the 9050RDK.
- If you are prompted with a **Select Device** dialog box after installing the driver, the Vendor ID or Device ID of the 9050RDK do not match the IDs found in `PLXRDK.INF`.
- If you removed the 9050RDK bus driver using only the Device Manager, Windows 95 can still reload the driver automatically—you will not be prompted.

- If a PLX 9050RDK driver is installed, and you no longer want to use it, follow the instructions in the section "Removing or Reinstalling the Driver," on page 13 to "permanently" disassociate all PLX drivers from the 9050RDK card.
- You can use the Device Manager, as described in the section, "Verifying the Driver Installation," on page 11 to verify the driver installation.

Installing after Previously Choosing Not to Install a Driver

If your 9050RDK card is installed, your computer boots Windows 95 normally, and you have either removed or never installed the 9050RDK bus driver, then chances are you chose not to install the driver when prompted in an earlier session.

This section describes how to remove the Windows 95 dummy driver and install the 9050RDK bus driver in its place.

♦ **To remove the Windows 95 dummy driver:**

1. Open **Start | Settings | Control Panel | System | Device Manager**.
2. Double click **Other Devices**.
3. Select **PCI Card.**
4. Choose **Remove.**
5. Choose **OK** from the **Confirm Device Removal** dialog box.
6. Choose **Close.**

Now, use normal procedures to reboot the system and follow the instructions in the section "Installing when Windows 95 Finds New PCI Hardware," on page 10.

Verifying the Driver Installation

After installing the driver, you can verify whether it was installed properly.

♦ **To verify the driver installation:**

1. Open **Start | Settings | Control Panel | System | Device Manager**.
2. Double click **System devices**.

If you see **PLX 9050RDK Bus** in the list, then the correct driver is installed.

For more details on the driver, refer to the section, "Driver Properties, Device Resources, and Changing the Driver," on page 12.

Driver Properties, Device Resources, and Changing the Driver

This section describes how to examine the properties of the serial port driver (such as provider and version) and resources (such as ports, memory, and interrupts). This section also provides information for changing the driver.

The following procedures are standard. You use the Device Manager to examine the properties and resources of the serial port driver, and optionally change the device driver.

♦ To examine the driver properties and resources:

1. Follow instructions in the section, "Verifying the Driver Installation," on page 11 to open the Device Manager, and show the list of system devices.
2. Select **PLX 9050RDK Bus**.
3. Choose **Properties** to open the properties box for the selected device.
   a. You can examine the driver properties by selecting the **Driver** tab.
   b. You can examine the resources (such as ports, memory, and interrupts) allocated to the 9050RDK by selecting the **Resources** tab.

♦ To change the driver—Windows 95:

1. Select the **Driver** tab in the properties box of the selected device.
2. Select the **PLXRDK.VXD** entry.
3. Choose **Change Driver**.
4. You can either choose **Have Disk** -or- from the list of models, choose **PLX 9050RDK Bus**.
5. Close all the dialog boxes.

♦ To change the driver—Windows 95 OSR 2:

1. Select the **Driver** tab in the properties box of the selected device.
2. Choose **Update Driver**.
3. Allow Windows 95 to search for the driver or select it from the list provided.
4. Choose **Finish** to complete the installation.
5. Close all the dialog boxes.

Removing or Reinstalling the Driver

In Windows 95, the terms *removing* or *reinstalling* a driver really mean removing or reinstalling *references* to devices that *use* the driver.

♦ To remove the PLX 9050RDK bus driver:

1. Follow instructions in the section, "Verifying the Driver Installation," on page 11 to open the Device Manager, and show the list of **System devices**.
2. Select **PLX 9050RDK Bus**. (If you do not see **PLX 9050RDK Bus** listed, then the driver is *not* installed.)
3. If **PLX 9050RDK Bus** is selected, choose **Remove**.

To reinstall the driver, use normal procedures to reboot the system and follow the instructions in the section "Installing when Windows 95 Finds New PCI Hardware," on page 10.

PCI devices are "Plug and Play" (PnP)—and Windows 95 makes it difficult to get rid of drivers for PnP devices. Follow these steps to permanently disassociate the PLX 9050RDK bus driver from the PLX 9050RDK card.

♦ To "permanently" remove the PLX 9050RDK bus driver—Windows 95:

1. Follow the instructions above to remove the PLX 9050RDK bus driver.
2. In the `\WINDOWS\INF` folder, find the `PLXRDK.INF` files named `OEM`*n*`.INF` (where *n* as the *n*th OEM driver installed on your system). Open each of these files, using any text editor (such as Notepad or WordPad).
3. Within each file, look for text identifying the file as a PLX 9050RDK bus driver installation file.
4. Delete *all* `INF` files that are PLX 9050RDK bus driver files. (There should only be one, but it's a good idea to check for more. There will be multiple files if you installed the file more than once.)
5. Use normal procedures to reboot your computer.
6. When Windows 95 prompts you for a driver, choose **Do not install a driver** as described in the section "Standard Installation," on page 4.

♦ **To "permanently" remove the PLX 9050RDK bus driver—Windows 95 OSR 2:**

1. Follow the instructions above to remove the PLX 9050RDK bus driver.
2. Delete the `PLXRDK.INF` file from the `\WINDOWS\INF\OTHER` folder.
3. Use normal procedures to reboot your computer.
4. When Windows 95 prompts you for a driver, cancel out of the Update Device Driver Wizard, as described in the section "Standard Installation," on page 4.

If You Have Problems Installing the Driver

If you have problems installing the driver, it may be because of an installation error that occurred in a previous Windows 95 session. You might want to use the Windows 95 registry to resolve the problem.

♦ **To use the Windows 95 registry to resolve installation problems:**

1. Follow driver removal instructions in the section, "Removing or Reinstalling the Driver," on page 13 to remove references to the driver.
2. Open the registry (`REGEDIT.EXE`).
3. Expand the registry tree to `HKEY_LOCAL_MACHINE\Enum`.
4. Look for, and delete the `PlxEnum` key.
5. Expand the registry tree to `HKEY_LOCAL_MACHINE\Enum\PCI`.
6. Look for, and delete the `VEN_10B5&DEV_9050` key.
7. Close the registry.
8. Reboot the system to ensure the changes take effect.

If you did not find these registry keys, your problem may be somewhere else.

Note: Ensure the PCI 9050 configuration registers have the correct Vendor ID and Device ID. The Device Manager associates the 9050RDK bus driver to the PCI device with a Vendor ID of *0x10B5* and a Device ID of *0x9050*. You can use PLXMon95 to check these registers (refer to the section, "PCI Exploration Programs," on page 15 for additional information).

PCI Exploration Programs

PLX provides two applications to aid PCI engineering development and verification—PLXMon and PLXMon95. As the names imply, these programs are basically the same—one runs under DOS, the other under Windows 95. Common aspects of these programs are described using the name, "PLXMon". Differences between the programs are clearly labeled.

PLXMon is a command-line style program that provides open access to all physical memory and I/O on your computer. The PLXMon PCI commands enable you to

- Select any PCI device on the PCI bus
- Read and write the PCI configuration registers of the selected device
- Read and write PLX local registers (if the selected device is in the PLX family of PCI devices)
- Read and write the EEPROM of a PLX device

and more.

PLXMon features include user-defined variables, macros and several unique commands that are useful during product development.

PLXMon is described further in the *PLXMon User's Guide*. If you are not familiar with PLXMon, it is recommended that you install it and become familiar with it.

Installing and Starting the Programs

Refer to the *PLXMon User's Guide* for proper installation and startup of the PCI exploration programs.

Selecting the PCI 9050 in the 9050RDK

When PLXMon starts, it registers all the PCI devices it can find, then selects the first PLX device it registers. Use the `dev` command to see the PCI devices that PLXMon found, then verify the 9050 of the 9050RDK is selected. The PLX Vendor ID is "0x10B5", and the Device ID of the 9050 is "0x9050". If your computer has other PLX devices in it, or if the Vendor ID of the 9050 is being changed by the 9050RDK EEPROM, you may need to use the parameterized variation of the `dev` command to select the PCI 9050 (refer to the *PLXMon User's Guide* for information about this command).

Displaying the PCI 9050 PCI Configuration Registers

If necessary, select the 9050 of the 9050RDK (refer to the section, "Selecting the PCI 9050 in the 9050RDK," on page 15). Use the `pcr` command to display (and modify) the PCI configuration registers (refer to the *PLXMon User's Guide* for information about this command).

Displaying the PCI 9050 Local Configuration Registers

If necessary, select the 9050 of the 9050RDK (refer to the section, "Selecting the PCI 9050 in the 9050RDK," on page 15 for additional information). Use the `lcr` command to display (and modify) the local configuration registers (refer to the *PLXMon User's Guide* for information about this command).

Accessing ISA Memory, I/O and 9050RDK Memory

First, display the PCI 9050 PCI Configuration Registers as described in the section, "Displaying the PCI 9050 PCI Configuration Registers," on page 15.

Table 1-3 lists the four PCI Base Address for Local Address Space registers (PCR 0x18 through 0x24) and the Interrupt Line register (PCR 0x3C).

Table 1-3. Notable PCI Configuration Registers

PCR	PCI 9050 Name	9050RDK Chip Select
0x18	PCI Base Address for Local Address Space 0	ISA MEMCS
0x1C	PCI Base Address for Local Address Space 1	ISA IOCS
0x20	PCI Base Address for Local Address Space 2	CSRAM
0x24	PCI Base Address for Local Address Space 3	CSROM
0x3C	Interrupt Line (8 bit)	—

When your computer starts, the PCI BIOS dynamically sets up these registers, based on values in various PCI 9050 configuration registers and the system resource allocations of your computer. The local configuration register values are usually loaded from the 9050RDK EEPROM; otherwise, they are loaded with PCI 9050 reset values.

If everything is set up properly, you should be able to use the values in the PCI Base Address for Local Address Space registers (after stripping off any control bits) to activate the various chip selects.

Use memory access commands, such as `d`, `e`, and `m` (display, enter and move) for memory-mapped addresses; and I/O commands such as `i` and `o` (input and output) for I/O-mapped addresses (refer to the *PLXMon User's Guide* for information about these commands).

For example:

Perhaps you are interested in accessing I/O registers from an ISA card piggybacked on the 9050RDK. You look at Table 1-3 and see that to activate the ISA IOCS line, you must read the PCI Base Address for Local Address Space 1 Register (PCIBAR3; PCR 0x1C). You use the `pcr` command, and find that the PCI BIOS has set PCIBAR3 to 0x0000E001. Bit zero is set, indicating the register refers to a PCI I/O address, and bits 1 and 0 should be treated as zeros. Therefore, the base address for ISA IOCS is a PCI I/O port at 0xE000. The range for this address space can be determined from Local Address Space 1 Range Register (LAS1RR; LCR 0x04). Use the input and output commands to read and write the ISA registers, using 0xE000 as the base address.

If you have problems with local-side addresses or chip selects, your local configuration registers may be set wrong. Refer to the section, "Accessing EEPROM," on page 18 for information about setting up local configuration registers, and to the sections, "SRAM Subsystem Register Settings" and "ISA Subsystem Register Settings," on pages 30 and 34, respectively. Use the `lcr` command (refer to the *PLXMon User's Guide* for information about this command) to examine and change these registers.

Any changes to the PCI configuration registers or the local configuration registers will be lost at reset or power-down unless you change the appropriate EEPROM registers. To modify the EEPROM, refer to the section, "Accessing EEPROM," on page 18.

Note: The PCI Base Address for Local Address Space registers can be programmed to be referenced by either I/O or memory PCI cycles. This means that you can program the ISA IOCS (or any other chip select) to be memory or I/O mapped. Read the PCI 9050 data sheet regarding Local Address Space Range Registers for additional information.

Accessing EEPROM

Use the `eep` command to read and write EEPROM. The `eep` command works the same as `pcr` and `lcr`. You can also read and write EEPROM with the `re` and `we` commands; however, they are less intuitive.

Writing to EEPROM is successful under most conditions; however, National NM93CS46 (and compatible) EEPROMs may have special write-protection features enabled. It is recommended that you verify the EEPROM values after writing them.

If you want to reload all EEPROM locations used by the PCI 9050, you can run a PLXMon command file that writes all EEPROM locations with 9050RDK factory defaults. The file, named `RDK_EEP.MON`, is on the *PLX 9050RDK Distribution Disk*. Use the `read` command to read and execute the file, as in

```
read RDK_eep.mon
```

This command file contains several lines of `eep` commands. You may find it useful to make a copy of the file and modify it for your own needs, using any text editor (such as Notepad or WordPad).

It is possible to program the EEPROM with values that prevent your computer from booting. If this happens, and your EEPROM is in a socket, you can remove the EEPROM and restart your computer. If you are careful, you can reinstall the EEPROM "hot," then reprogram the EEPROM with less-offensive values.

Note: PLXMon and the EEPROM programming commands are discussed in the *PLXMon User's Guide*.

Configuration Issues

This section describes issues associated with configuring the PCI 9050.

Configuring PCI 9050 Registers to Access an ISA Card

This section describes how to configure the PCI 9050 registers to access an ISA card, by way of an example. The example applies to the comm card included in the kit, so it can easily be tested. You can use PLXMon to quickly test the example and set up your own ISA card.

Of the four chip select lines (ISA MEMCS, ISA IOCS, RAMCS and ROMCS) only ISA IOCS is used in the example. For reference, however, the PCI 9050 registers associated with all chip selects are listed in Table 1-4 through Table 1-7.

Table 1-4. PCI 9050 Registers Used for ISA MEMCS

PCI 9050 Registers Used for ISA MEMCS		
PCI 9050 Register Name	**Mnemonic**	**Register Offset**
PCI Base Address for Local Address Space 0	PCIBAR2	PCR 0x18
Local Address Space 0 Range	LAS0RR	LCR 0x00
Local Address Space 0 Local Base Address (Remap)	LAS0BA	LCR 0x14
Local Address Space 0 Bus Region Descriptors	LAS0BRD	LCR 0x28
Chip Select 0 Base Address	CS0BASE	LCR 0x3C

Table 1-5. PCI 9050 Registers Used for ISA IOCS (Referenced in the Example)

PCI 9050 Registers Used for ISA IOCS (Referenced in the Example)		
PCI 9050 Register Name	**Mnemonic**	**Register Offset**
PCI Base Address for Local Address Space 1	PCIBAR3	PCR 0x1C
Local Address Space 1 Range	LAS1RR	LCR 0x04
Local Address Space 1 Local Base Address (Remap)	LAS1BA	LCR 0x18
Local Address Space 1 Bus Region Descriptors	LAS1BRD	LCR 0x2C
Chip Select 1 Base Address	CS1BASE	LCR 0x40

Table 1-6. PCI 9050 Registers Used for CSRAM

PCI 9050 Registers Used for CSRAM		
PCI 9050 Register Name	**Mnemonic**	**Register Offset**
PCI Base Address for Local Address Space 2	PCIBAR4	PCR 0x20
Local Address Space 2 Range	LAS2RR	LCR 0x08
Local Address Space 2 Local Base Address (Remap)	LAS2BA	LCR 0x1C
Local Address Space 2 Bus Region Descriptors	LAS2BRD	LCR 0x30
Chip Select 2 Base Address	CS2BASE	LCR 0x44

Table 1-7. PCI 9050 Registers Used for CSROM

PCI 9050 Registers Used for CSROM		
PCI 9050 Register Name	**Mnemonic**	**Register Offset**
PCI Base Address for Local Address Space 5	PCIBAR5	PCR 0x24
Local Address Space 3 Range	LAS3RR	LCR 0x0C
Local Address Space 3 Local Base Address (Remap)	LAS3BA	LCR 0x20
Local Address Space 3 Bus Region Descriptors	LAS3BRD	LCR 0x34
Chip Select 3 Base Address	CS3BASE	LCR 0x48

The 9050RDK EEPROM, as configured at the factory, sets up the PCI 9050 to access the full range of ISA I/O ports (0 to 0x3FF). PLX configured it this way so you can plug in any ISA board and start accessing its I/O ports, quickly and easily. Any individual board, however, requires only a fraction of the 0x400 port addresses. The following example configures the PCI 9050 for the small range of I/O ports required by a single channel of an ISA comm card.

- **Assume the following:**
  - The ISA comm card is setup as COM1 only (eight consecutive I/O ports—0x3F8 through 0x3FF)
  - The ISA card will be accessed using PCI I/O cycles
  - The chip select base address for ISA IOCS is arbitrarily chosen to be 0x03000000
  - The PCI 9050 data sheet and this manual are available for easy reference (you will need to reference Chapter 2, "PCI 9050RDK Hardware Manual") and the *PLXMon User's Guide*)

- **Apply the following settings:**

  1. Set LAS1RR to 0xFFFFFFF9, because only three bits are required to select a unique register on the comm card and the card will be referenced using PCI I/O cycles.
  2. Set LAS1BA to 0x030003F9, because the COM1 base address is 0x3F8 and the chip select base address is 0x03000000.
  3. Set CS1BASE to 0x030003FD, following the outline for local chip selects in the bus operation section of the PCI 9050 data sheet.
  4. Set LAS1BRD. This setting must also be set properly; however, its values are not discussed here. Your best sources for configuring this register can be found in Chapter 2, "PCI 9050RDK Hardware Manual," and the PCI 9050 data sheet.

Note: Any changes to the local configuration registers will be lost when you power off your computer unless you change the EEPROM. To modify the EEPROM, refer to the section, "Accessing EEPROM," on page 18.

You can use this example as a basis for configuring the PCI 9050 registers that control the ISA MEMCS line.

Configuring the PCI 9050 Local Interrupt Register

Refer to the PCI 9050 data sheet for a complete description of the PCI 9050 local interrupt control/status register (INTCSR). The 9050RDK ORs all ISA interrupts together, and the logic level is inverted before entering the PCI 9050 interrupt 1 input (LINTI1). Therefore, the local interrupt 1 polarity (INTCSR bit 1) should be "0". To enable interrupts, both the local interrupt 1 enable (INTCSR bit 0) and PCI interrupt enable should be "1".

Ensure the system BIOS of your computer is configuring interrupts correctly. The BIOS of some systems enable you to configure specific interrupts for PnP, ISA or PCI.

PCI Base-Class, Sub-Class, and Programming Interface

PCI Configuration Registers 0x0B, 0x0A, and 0x09 specify the PCI base-class, sub-class and programming interface, respectively. Refer to the latest PCI specifications for values that are appropriate to your project. You must change the proper EEPROM entries for any new values to be loaded into PCI 9050 registers at reset or power-up. To modify the EEPROM, refer to the section, "Accessing EEPROM," on page 18.

Windows 95 Tips

This section includes other useful tips that do not fit anywhere else.

Often, you will not want to waste time booting Windows 95—you can use DOS programs, such as the DOS version of PLXMon.

♦ To make your Windows 95 machine boot DOS first:

1. Find the `MSDOS.SYS` file on your system (usually located under the root, `C:\`).
2. Create a backup copy of the `MSDOS.SYS` file (for example, `MSDOS.BAK`) *before* you edit it.
3. Remove the *hidden*, *read-only*, and *system* attributes from the file.
4. Open the file using a standard text editor (such as Notepad or WordPad).
5. Modify the line "`BootGUI=1`" to "`BootGUI=0`".
6. Save and close the file.
7. **Optional:** Replace the attributes of the file.
8. Reboot your computer.

Your computer will now boot DOS first. You can type `win` at the DOS prompt to boot Windows 95 at any time.

♦ To make your Windows 95 machine boot Windows 95 first:

1. Rename the `MSDOS.SYS` file you created in the previous process (for example, rename the file to `MSDOS.DOS`).
2. Rename the `MSDOS.BAK` file you created in the previous process to `MSDOS.SYS`.
3. Reboot your computer.

Troubleshooting

The most common problems encountered in an otherwise working configuration are

- 9050RDK card jumpers set incorrectly
- PCI 9050 local configuration registers set incorrectly

There are only a few jumper settings on the 9050RDK card, so it is probably easiest to check the jumper settings of the 9050RDK documented in Chapter 2, "PCI 9050RDK Hardware Manual."

The EEPROM values loaded into the PCI 9050 at reset time are "factory set" for running a piggybacked ISA card. These values may have changed or are not appropriate for your project. The first step is to install the PLX PCI exploration programs (refer to the section, "PCI Exploration Programs," on page 15). These programs are useful in tracking down and correcting PCI 9050 local configuration registers and EEPROM values. Chapter 2, "PCI 9050RDK Hardware Manual," provides guidelines for correct local configuration register settings. The section, "Accessing EEPROM," on page 18 also includes register configuration tips.

Other troubleshooting information can be found in the section, "If You Have Problems Installing the Driver," on page 14.

Notes:

Chapter 2

PCI 9050RDK Hardware Manual

This chapter describes each subsystem found on the RDK and provides suggestions for using the tools and circuitry included as a starting point for a new design.

The RDK is designed as both a demonstration board and a launch pad for new designs. The hardware consists of the following:

- PCI slot interface
- PCI 9050 bus target interface chip
- SRAM system to demonstrate PCI memory accesses
- ISA slot and controller to facilitate migration of existing designs to a PCI platform
- Daughter-card connector, to which developers can attach new circuit board designs
- Prototyping space and headers, for testing and benchtop experimentation

PCI Interface

The PCI 9050 is fully compliant with version 2.1 of the PCI specification. The RDK is designed to plug directly into a PCI expansion slot. As the 9050 is intended as a target only, all expansion slot signals relating to bus mastering are appropriately terminated.

The two halves of the PCI slot connector are labeled J1 and J2. Please refer to the schematics provided in Chapter 3, "PCI 9050RDK Schematics," for specifics regarding connections to the PCI slot.

PCI 9050 Target Interface Chip

The PCI 9050 IC represents the core of the RDK. The 9050 is responsible for translating the PCI bus cycles to the appropriate local bus cycles to properly interface to the RDK subsystems. Some of the relevant features of this part are

- PCI Specification 2.1 compliance
- Low cost
- Five remappable local address spaces and four internally decoded chip selects
- Bidirectional FIFO for zero wait state bursts
- Performs to the PCI maximum 132 MB/sec transfer rate
- Supports a 32-, 16-, or 8-bit wide, multiplexed or nonmultiplexed local bus
- Allows the local bus to operate asynchronously to the PCI clock
- Performs Big Endian/Little Endian conversion (if required by the local bus)
- Supports local bus clocks up to 40 MHz
- Serial EEPROM interface for storage of configuration information

Discussion here is limited to board-level configuration concerns and certain aspects of the 9050 that are directly applicable to the RDK. For a full discussion of chip operation, refer to the PCI 9050 data sheet, available from PLX and the PLX website (`http://www.plxtech.com`).

Chip Operating Modes

Test mode—The JP3 jumper places the 9050 chip into Factory Test mode (refer to Table 2-1). This jumper should be removed for normal operation.

Table 2-1. Test Mode Jumper Settings

JP3 in Place	Effect on Operation
Yes	Test mode enabled
*No	Test mode disabled

* Default loading

Local Bus Multiplex mode—The JP1 and JP2 jumpers determine whether the local bus is operating in Multiplexed or Nonmultiplexed mode (refer to Table 2-2). When operating in Multiplexed mode, the address and data paths share the same IC pins, whereas when operating in Nonmultiplexed mode, the address and data paths operate independently. Any subsystems accessed must share the same multiplex setting. The SRAM and ISA subsystems are defined as nonmultiplexed; however, developers are free to use Multiplex mode when accessing custom circuitry through the daughter-card connector or prototype areas.

Table 2-2. Mux Mode Jumper Settings

JP1 in Place	JP2 in Place	Effect on Operation
No	No	Invalid
No	Yes	Multiplexed operation
*Yes	*No	Nonmultiplexed operation
Yes	Yes	Invalid

* Default loading

Local Bus Clock

The local bus clock may be set to any frequency up to 40 MHz. Several clock signals are made easily available—8 MHz for ISA operation, 33 MHz for fast access to the static RAM, and a buffered version of the PCI clock. An additional user-selectable clock may be connected to post 1 of header J10. Note that there are two methods available for routing these signals to the local clock input of the 9050—two pin header jumpers (on header J10), for quick selection, and pads for 0 Ω resistors for a more dependable connection (refer to Table 2-3).

See text for defaults.

Caution: Ensure that only one method is used, as damage may result if clock signals are shorted together.

Table 2-3. Clock Selection

Local Clock Signal	J10 Jumper Position	Resistor Location
BCLK or PCI Clock	Jump Pins 1 - 2	R28
33 MHz	Jump Pins 3 - 4	R27
16 MHz	Jump Pins 5 - 6	R25
8 MHz	Jump Pins 7 - 8	R26

Note: R28, R27, R25, & R26 are not installed on the 9050RDK board

Local Bus Interrupts

The PCI 9050 provides two inputs for two local bus interrupts. When using the ISA circuitry on the RDK, both interrupts are used for handling ISA events. LINTi1 is used for the ISA IRQ interrupts, and LINTi2 is used to service ISA error conditions. Jumpers are provided on the J10 header (refer to Table 2-4) to allow routing of interrupts from custom circuitry located on the daughter-card connector, or tied directly to the posts of J10.

Table 2-4. Interrupt Routing

J10 Jumper Position	PCI 9050 Input	System Source
*Jump Pins 9 - 10	LINTi1	ISA IRQ Interrupt
Jump Pins 11 - 12	LINTi1	Custom Interrupt 1
*Jump Pins 13 - 14	LINTi2	ISA Bus Error Signal
Jump Pins 15 - 16	LINTi2	Custom Interrupt 2

* Default loading

Local Bus Chip Selects

The PCI 9050 will internally decode up to four chip selects, set at user-defined ranges. Jumpers on the J14 header are provided to route these chip selects to the circuitry on the RDK as described in Table 2-5. The chip selects are also routed to the daughter-card connector for use in user designs, or may be tapped from the header posts of J14 to route to the prototyping area.

Note: When used for purposes other than the standard circuitry, remove the appropriate jumper from the J14 header to prevent conflict. Chip selects CS2 and CS3 may also be configured as user definable I/O pins; in this case also, remove the appropriate jumper.

All loaded by default.

Note: Pins 3 –4, 7 –8, 11 –12, 15 – 16 are not used.

Table 2-5. Local Bus Chip Selects

J14 Jumper Position	PCI 9050 Signal	RDK System Usage
Jump Pins 1 - 2	CS0#	(ISA) MEMCS#
Jump Pins 5 - 6	CS1#	(ISA) IOCS#
Jump Pins 9 - 10	CS2#/USER2	CSRAM#
Jump Pins 13 -14	CS3#/USER3	CSROM#

SRAM Subsystem

The following sections discuss the SRAM subsystem architecture and register settings.

Subsystem Architecture

There is a small amount of static RAM supplied on the RDK to demonstrate memory accesses from the PCI bus. The memory is organized to use the full 32 bit local bus, and is 32k long words deep. The memory is accessed using CSRAM, and the proper jumper must be in place, as indicated in Table 2-5. The RAM system operates at 33 MHz and is capable of supporting zero wait state read and write accesses.

There are five gates added to use the Burst mode capabilities of the 9050. Burst mode provides a single address cycle, followed by many data cycles to what are understood to be successive addresses, thereby deleting the additional address cycles and increasing throughput. In new designs where only single accesses are to be supported, the gates can be eliminated and the 9050 interface to the static RAM would be entirely glueless.

SRAM Subsystem Register Settings

To access the static RAM, choose one of the four mappable address spaces. Table 2-6 through Table 2-9 provide examples as to how to set up the PCI 9050 local registers.

Note: All four registers should reference a consistent address space, and the base address of the local memory space, and the base address of the chip select, although configurable, should match.

Table 2-6. Local Address Space Range Register (LAS2RR)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Memory Space Indicator	0h	SRAM is memory mapped
2:1	Mapping location	0h	Locate anywhere in PCI space.
3	Prefetch support	0h	Prefetching from SRAM is acceptable
27:4	Decode range	FFE0000h	Address lines 27:17 are used to decode
31:28	Unused	0h	Must be set to zero

Table 2-7. Local Address Space Base Address Register (LAS2BA)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Space Enable	1h	Enable decoding of this address space
1	Unused	0h	Must be set to zero
3:2	Unused	0h	Must be set to zero
27:4	Remap address	1000000h	Arbitrary, but must match chip select base
31:28	Unused	0h	Must be set to zero

Table 2-8. Local Address Space Region Descriptor Register (LAS2BRD)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Bursting Enable	1h	Enable SRAM bursting
1	Ready Input Enable	0h	Use internal wait state generator
2	BTERM Enable	1h	Disable BTERM input
4:3	Prefetch Count	0h	Prefetch 16 long words
5	Prefetch Count Enable	0h	Enable prefetching
10:6	NRAD Wait States	0h	No address to data wait states
12:11	NRDD Wait States	0h	Zero wait state on burst accesses
14:13	NXDA Wait States	0h	Zero wait state from data to address
19:15	NWAD Wait States	0h	No address to data wait states
21:20	NWDD Wait States	0h	Zero wait state for burst accesses
23:22	Bus Width	2h	32 bit local bus to SRAM
24	Endian Order	0h	Little Endian (adjust for system)
25	Big Endian Byte Lane	0h	32 bit bus
27:26	Read Strobe Delay	0h	Assert RD immediately
29:28	Write Strobe Delay	0h	Assert WR immediately
31:30	Write Cycle Hold	0h	No need for additional wait states

Table 2-9. Chip Select Base Register (CS2BASE)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Chip Select Enable	1h	Enable this chip select
1	Unused	0h	Must be set to zero
27:2	Local Base of Chip Select	1010000h	Must match the address space
31:28	Unused	0h	Must be set to zero

ISA Subsystem

One of the principle uses of the PCI 9050 will be to upgrade existing ISA bus designs to PCI. A designer can get a head start in this process by simply plugging their ISA card into the slot provided on the RDK. This allows the engineer to get a feel for the operation of the 9050 and the PCI bus, and provides a development tool to begin software driver development. The PCI 9050, as well as additional ISA conversion logic on the RDK, convert the ISA bus cycles to PCI and vice versa.

The ISA bus interface is a compilation of several different card styles and backward-compatibility issues. The RDK is designed to support a wide variety of ISA cards operating as slaves to the bus. The heart of the ISA interface is the PLD, which houses a state machine to provide the requisite signals to the card and responds to signals returned for bus sizing and wait state control.

As only one slot is needed for development, the designer's driver software can indicate to the state machine whether the card is an 8-bit or 16-bit device. This is done by using one of the 9050 user controlled I/O pins. The PLD can then be expected to behave as if the card is plugged into an ISA motherboard.

The PCI 9050 provides four independent chip selects to assist in address decoding. Two of these chip selects are used by the state machine to distinguish between memory and I/O accesses to the ISA card so that the proper timings can be observed.

Bus Cycle Conversion

The ISA Controller is embedded into an AMD MACH210 PLD. The goal of the state machine is to converse between the PCI 9050 IC and an ISA expansion card by mimicking the behavior of an ISA motherboard as closely as possible. The controller is designed to handle ISA slave accesses only, and does *not* provide for ISA masters, DMA, or refresh cycles.

The controller is driven at twice the ISA bus frequency, and consists of eight, two-phase states. Each state represents one full cycle of the ISA clock, with Phase 0 corresponding to a logic high voltage of the clock, and Phase 1 corresponding to a logic low. Alternatively, the controller could be viewed as a 16-state machine, but the naming convention is chosen to match the state of the system as viewed from the ISA perspective.

Three distinct bus cycles are supported:

- 16 bit memory accesses
- 16 bit I/O accesses
- 8-bit accesses (of either type)

These cycles may be shortened or lengthened by the expansion card within the limits maintained by the standard ISA motherboard. Additional control is provided by the PCI 9050 in the form of two chip selects and one user I/O pin. One chip select is intended to be used to indicate an I/O access (referred to as “IOCS~”) and the other indicates memory accesses (referred to as “MEMCS~”). The user I/O pin indicates whether the card is an 8-bit or 16-bit device (referred to as “8BIT”).

Two additional states have been added to the standard ISA cycle to support the timing requirements of that bus. There is a leading wait state that supports the split addressing, or address pipelining of the ISA bus signals LA[23:17], and there is a trailing state that maintains the proper data hold times of the bus. For back-to-back accesses, these two states are merged, resulting in only one additional state to minimize overhead.

Following the leading bus cycle, the standard ISA addressing cycle is performed, and BALE and ~SBHE are asserted as expected. The chip selects, address, and ~M16/~IO16 are decoded to generate the proper ~IORD, ~IOWR, ~MEMRD, ~MEMWR, ~SMRD, and ~SMWR commands to the slot.

Use of ~NOWS and CHRDY are supported to shorten or extend the bus cycles to the ISA card, and wait states are added or removed, as required. A data latch is present to provide the data hold times expected by an ISA device after the final data cycle.

Upon completion of the ISA access, ~LRDYI is asserted to the 9050, and the local bus cycle is concluded.

Several signals are driven locally, regardless of the selection state:

- RESET—The ISA reset signal is opposite in sense to the 9050 local bus reset.
- ISAA0 and ISAA1—These signals are derived from the 9050 local bus byte enables, depending on whether the addressed device is 8 or 16 bits.
- SBHE—This signal indicates to the ISA device whether the upper 8 bits of a 16 bit data bus are being accessed.
- DATARW—This signal determines the direction of the ISA bus data transceiver.
- G1MB—This signal is used internally to indicate whether the current access is above the 1 MB boundary. This signal is not driven externally.

ISA Subsystem Register Settings

Table 2-10 through Table 2-17 list the ISA subsystem register settings:

Table 2-10. Local Address Space Range Register (Memory) (LAS0RR)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Memory Space Indicator	0h	I/O and Memory are Memory Mapped
2:1	Mapping Location	0h	Locate anywhere in PCI space
3	Prefetch Support	0h	Do not prefetch from ISA
27:4	Decode Range	F000000h	Address lines 27:24 are used to decode
31:28	Unused	0h	Must be set to zero

Table 2-11. Local Address Space Range Register (I/O) (LAS1RR)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Memory Space Indicator	0h	I/O and Memory are Memory Mapped
2:1	Mapping Location	0h	Locate anywhere in PCI space
3	Prefetch Support	0h	Do not prefetch from ISA
27:4	Decode Range	F000000h	Address lines 27:24 are used to decode
31:28	Unused	0h	Must be set to zero

Table 2-12. Local Address Space Base Register (Memory) (LAS0BA)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Space Enable	1h	Enable decoding of this address space
1	Unused	0h	Must be set to zero
3:2	Unused	0h	Must be set to zero
27:4	Remap Address	2000000h	Arbitrary, but must match chip select base
31:28	Unused	0h	Must be set to zero

Table 2-13. Local Address Space Base Register (I/O) (LAS1BA)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Space Enable	1h	Enable decoding of this address space
1	Unused	0h	Must be set to zero
3:2	Unused	0h	Must be set to zero
27:4	Remap Address	3000000h	Arbitrary, but must match chip select base
31:28	Unused	0h	Must be set to zero

Table 2-14. Local Address Space Region Descriptor (Memory) (LAS0BRD)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Bursting Enable	0h	No ISA bursting
1	Ready Input Enable	1h	Use external LRDY indicator
2	BTERM Enable	0h	Disable BTERM input
4:3	Prefetch Count	0h	No prefetching
5	Prefetch Count Enable	0h	No prefetching
10:6	NRAD Wait States	1h	One address to data wait state
12:11	NRDD Wait States	0h	No bursting
14:13	NXDA Wait States	0h	Data to address controlled by LRDY
19:15	NWAD Wait States	1h	One address to data wait state
21:20	NWDD Wait States	0h	No bursting
23:22	Bus Width	1h	16 bit (according to ISA card)
24	Endian Order	0h	Little Endian
25	Big Endian Byte Lane	0h	Use lower data path
27:26	Read Strobe Delay	0h	Assert RD immediately
29:28	Write Strobe Delay	0h	Assert WR immediately
31:30	Write Cycle Hold	1h	One wait state on writes

Table 2-15. Local Address Space Region Descriptor (I/O) (LAS1BRD)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Bursting Enable	0h	No ISA bursting
1	Ready Input Enable	1h	Use external LRDY indicator
2	BTERM Enable	0h	Disable BTERM input
4:3	Prefetch Count	0h	No prefetching
5	Prefetch Count Enable	0h	No prefetching
10:6	NRAD Wait States	1h	One address to data wait state
12:11	NRDD Wait States	0h	No bursting
14:13	NXDA Wait States	0h	Data to address controlled by LRDY
19:15	NWAD Wait States	1h	One address to data wait state
21:20	NWDD Wait States	0h	No bursting
23:22	Bus Width	0h	8bit (according to ISA card)
24	Endian Order	0h	Little Endian
25	Big Endian Byte Lane	0h	Use lower data path
27:26	Read Strobe Delay	0h	Assert RD immediately
29:28	Write Strobe Delay	0h	Assert WR immediately
31:30	Write Cycle Hold	1h	One wait state on writes

Table 2-16. Memory Chip Select Base Register (CS0BASE)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Chip Select Enable	1h	Enable this chip select
1	Unused	0h	Must be set to zero
27:2	Local Base of Chip Select	2800000h	Must match the address space
31:28	Unused	0h	Must be set to zero

Table 2-17. I/O Chip Select Base Register (CS1BASE)

Bit Field	Field Description	Set to Value	Setting Indicates
0	Chip Select Enable	1h	Enable this chip select
1	Unused	0h	Must be set to zero
27:2	Local Base of Chip Select	3000400h	Must match the address space
31:28	Unused	0h	Must be set to zero

Handling of ISA Interrupts

ISA interrupts are supported by a cascade of OR gates that allow the ISA interrupts to generate a local bus interrupt to the 9050. If the ISA slave uses multiple interrupts, care must be taken in the design of the PCI driver software to account for this, especially in cases where the interrupts may occur simultaneously. The ISA ~CHCHK error signal is routed to the second 9050 local bus interrupt to allow handling of these conditions.

ISA Features Not Supported on the RDK

The RDK supports a wide variety of ISA expansion cards; however, it does not support ISA bus masters or boards requiring DMA support. For ISA designs that make use of these features, PLX offers a range of PCI bridge chips that allow migration to the PCI bus.

The RDK does not provide refresh cycles to the ISA slot. If this is necessary for support of a specific card, the designer may make use of the prototyping area of the RDK to create the needed timing logic and to integrate into the existing ISA support circuitry.

Sixteen-bit accesses to an 8-bit peripheral are not supported and must be managed by the driver firmware.

Moving to a New Design

The ISA implementation of the 9050RDK is designed to be as general as possible to support a wide range of legacy cards. Depending on the design, most or all of this ISA conversion logic can be eliminated when doing the actual PCI development. This section calls out several points of analysis for reducing the required hardware.

Determine what interrupt sources are required and route them directly to the 9050 local interrupt lines. The OR/NAND gates used for interrupt routing (U15, U17 and U18) can then be removed. If there is no need to support a CHCHK-type error signal, this interrupt source can be removed. If the design can benefit from multiple interrupt sources, both 9050 input lines may be used.

Note: The polarity of the interrupt lines is programmable.

Note: The address buffers (U11 and U12) are not required for the ISA design, but are present to reduce the loading on these buses, and to allow the SRAM to function as fast a possible. Check the fanout of the new design, as the drive capability of the 9050 is probably sufficient in most cases.

Check the address setup and data hold requirements of the resident hardware. If the design uses an ASIC, or standard ICs with an integrated ISA interface, review the specifications. Many parts can tolerate, but do not require the setup and hold times provided. The leading and trailing bus wait state cycles are more likely a limitation of the ISA bus than a limitation of the hardware on the card, and can probably be eliminated as the design is migrated to the PCI bus.

The data latch (U16) can be removed if the data hold times are satisfied by the standard local bus cycle. If the design requires longer data hold times, check the PCI 9050 documentation to determine whether the programmable wait state generator can save the extra component.

Check the clocking requirements of the design. If there is no need for the 14.31818 MHz clock, you can eliminate its crystal (Y2). If the various strobes and timing signals of the 9050 are sufficient to synchronize the bus cycles, the 8 MHz and 16 MHz clocks (Y1 and U14) may also be removed. Check the 9050 documentation, as many of the bus control signals have programmable timings that may ease this interface considerably. If the full PLD state machine is to be used, a 16 MHz standard is still required. The 9050 provides a buffered version of the PCI bus clock, which may be used by the local bus. Keep in mind that the PCI specification allows this clock to vary in frequency dynamically, from 0 to 33 MHz. If the performance of the local bus hardware is capable of being scaled, or if it can be safely assumed that the PCI clock is fixed, this buffered clock can save the complications of multiple clock sources.

For many designs, the intermediate conversion to ISA can be eliminated entirely, and the PLD (U15) removed with it. This is the ideal solution, and is very likely, given the flexibility in the 9050 local bus cycles. For designs where this is not possible (such as ASICs or standard ICs that expect an ISA interface), the state machine equations are available to make this integration straightforward. In cases where the ISA interface is required, analysis may reveal that only a subset of the functionality present in the state machine is required, which would allow a reduction of the equation set, and the use of a more compact PLD. Obvious opportunities for economy include the knowledge of the bus width at design time, knowing whether the design is to operate in Memory or I/O mode, and in many cases, knowing how many wait states are required for each access. Selecting these parameters allows much of the flexibility to be removed from the PLD, and much of the logic can then be removed along with it.

Daughter-Card Connector

The RDK provides several facilities for building up new designs. One of these is a daughter-card connector (J3), located midway down the board. This connection point breaks out most of the 9050 local bus connections for use on a plug in PC board module. Table 2-18 describes the pinout of J3.

Note: Use of a daughter card affects the jumper settings. Other functions of the RDK may require disabling to accommodate the added PC card.

Table 2-18. Daughter-Card Connector (J3)

Pin #	Function	Pin #	Function	Pin #	Function	Pin #	Function
1	ADS#	26	BD29	51	n.c.	76	LA8
2	GND	27	BD28	52	n.c.	77	LA7
3	LCLK	28	BD27	53	n.c.	78	LA6
4	GND	29	BD26	54	n.c.	79	LA5
5	BLAST#	30	BD25	55	LA27	80	LA4
6	USER1	31	BD24	56	LA26	81	LA3
7	LW/R	32	GND	57	LA25	82	LA2
8	GND	33	BD23	58	LA24	83	5V
9	LRDY#	34	BD22	59	5V	84	BD7
10	LRESET#	35	BD21	60	LA23	85	BD6
11	LBE0#	36	BD20	61	LA22	86	BD5
12	LBE1#	37	BD19	62	LA21	87	BD4
13	LBE2#	38	BD18	63	LA20	88	BD3
14	LBE3#	39	BD17	64	LA19	89	BD2
15	USER2	40	BD16	65	LA18	90	BD1
16	GND	41	GND	66	LA17	91	BD0
17	XINT1	42	BD15	67	LA16	92	GND
18	XINT2	43	BD14	68	5V	93	5V
19	n.c.	44	BD13	69	LA15	94	LHOLDA
20	n.c.	45	BD12	70	LA14	95	LHOLD
21	USER0	46	BD11	71	LA13	96	GND
22	GND	47	BD10	72	LA12	97	n.c.
23	5v	48	BD9	73	LA11	98	n.c.
24	BD31	49	BD8	74	LA10	99	12V
25	BD30	50	GND	75	LA9	100	-12V

Test Headers and Prototype Area

In addition to the daughter-card connector, the RDK provides a series of open header posts (J8 through J11) and a large prototyping area. The combination of these two features are ideal for signal monitoring, fast benchtop prototyping, and small simple circuits where formal PC card development is not practical. Local bus signals may be accessed by wiring directly to the posts of these and other headers on the board, and routing directly to the prototype area. Again, it may be necessary to disable some of the native circuitry of the RDK to prevent resource contention.

Table 2-19 through Table 2-24 describe the pinout of the test headers.

Note: The odd numbered pins on each of these connectors is connected to the internal ground plane.

Table 2-19. Pinout for J5

Pin #	Function
4	LBE0#
6	LBE1#
8	BA2
10	BA3
12	BA4
14	BA5
16	BA6
18	BA7
20	BA8
22	BA9
24	BA10
26	BA11
28	BA12
30	BA13
32	BA14
34	BA15
36	GND
38	LBE0#

Table 2-20. Pinout for J8

Pin #	Function
4	BA16
6	BA17
8	BA18
10	BA19
12	BA20
14	BA21
16	BA22
18	BA23
20	BA24
22	BA25
24	BA26
26	BA27
28	LBE0#
30	LBE1#
32	LBE2#
34	LBE3#
36	GND
38	BA16

Table 2-21. Pinout for J6

Pin #	Function
4	BD0
6	BD1
8	BD2
10	BD3
12	BD4
14	BD5
16	BD6
18	BD7
20	BD8
22	BD9
24	BD10
26	BD11
28	BD12
30	BD13
32	BD14
34	BD15
36	GND
38	BD0

Table 2-22. Pinout for J9

Pin #	Function
4	BD16
6	BD17
8	BD18
10	BD19
12	BD20
14	BD21
16	BD22
18	BD23
20	BD24
22	BD25
24	BD26
26	BD27
28	BD28
30	BD29
32	BD30
34	BD31
36	GND
38	BD16

Table 2-23. Pinout for J4

Pin #	Function
4	LCLK
6	LRESET#
8	ADS#
10	ALE
12	LW/R
14	BLAST
16	LRDYI#
18	BTERM#
20	CS0#
22	CS1#
24	RD#
26	WR#
28	n.c.
30	n.c.
32	n.c.
34	n.c.
36	GND
38	LCLK

Table 2-24. Pinout for J7

Pin #	Function
4	LINTi1
6	LINTi2
8	LHOLD
10	LHOLDA
12	USER0
14	USER1
16	USER2
18	USER3
20	n.c.
22	n.c.
24	n.c.
26	n.c.
28	n.c.
30	n.c.
32	n.c.
34	n.c.
36	GND
38	LINTi1

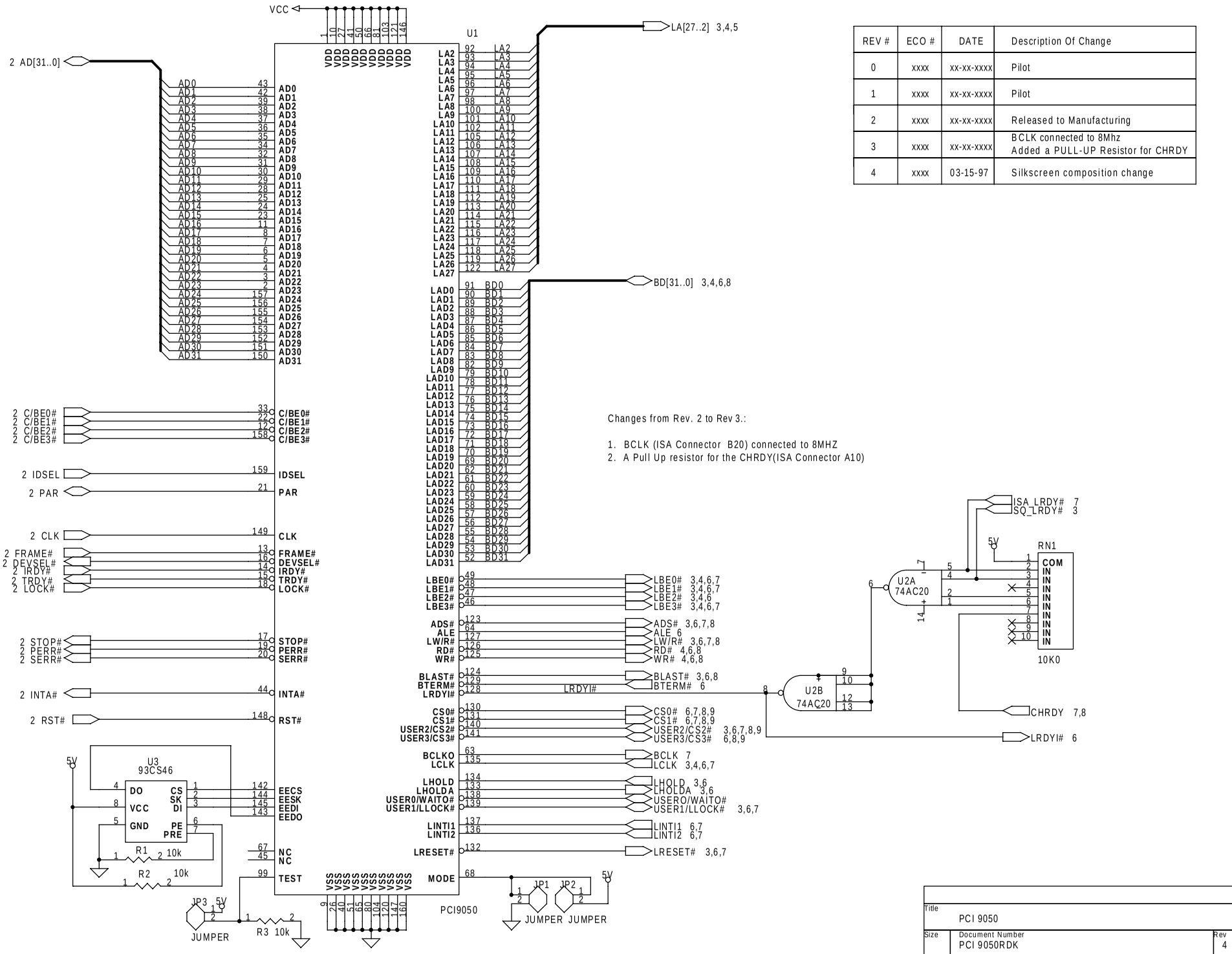
Additional Development Resources Provided by PLX

In addition to a broad family of PCI bridge chip products, PLX Technology provides a variety of resources, including evaluation boards and software, to simplify the development of PCI boards and systems. For an index of resources, contact PLX directly or through our website at `http://www.plxtech.com`.

Chapter 3

PCI 9050RDK Schematics

This chapter provides schematics of the PCI 9050RDK for easy reference.

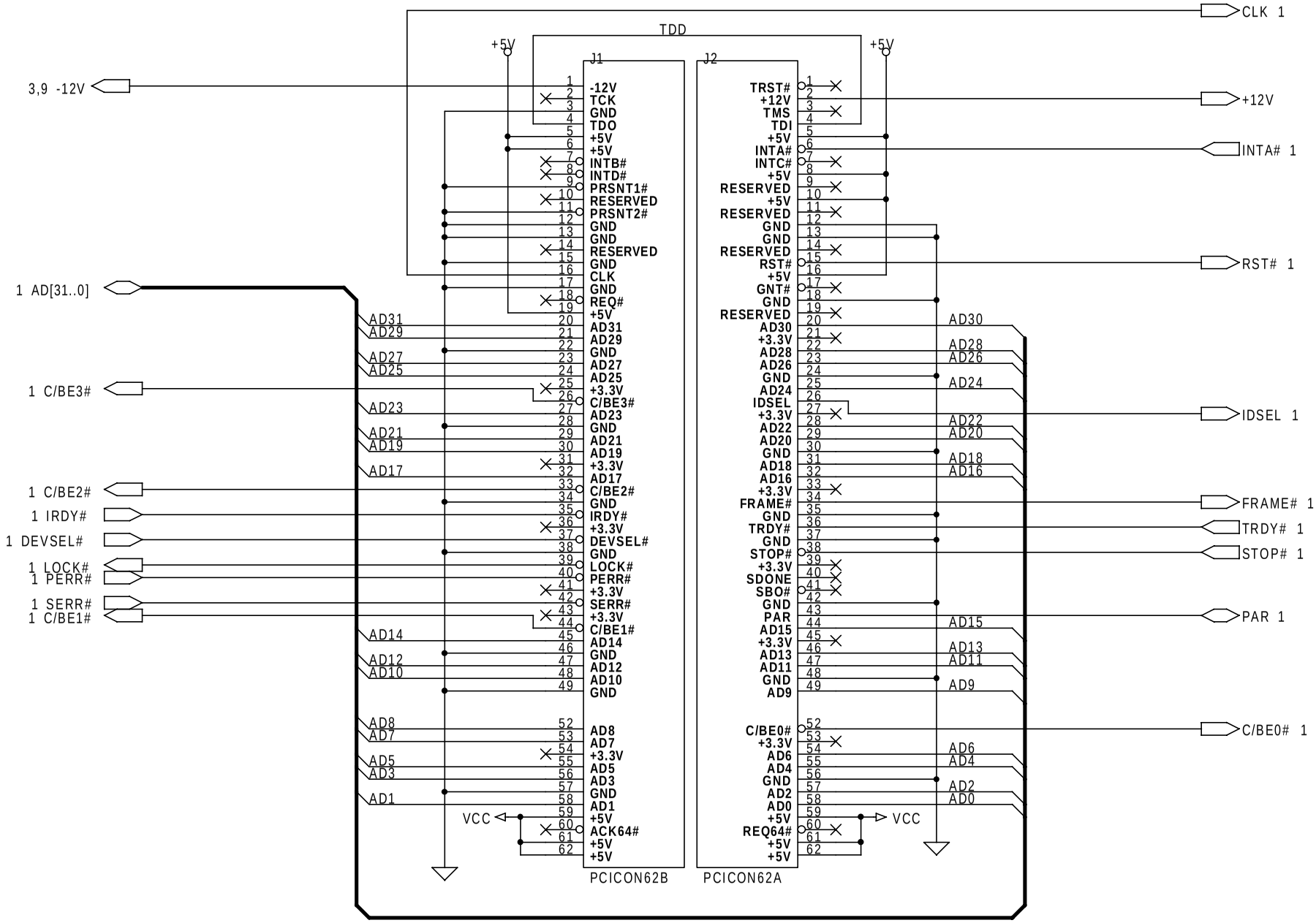


REV #	ECO #	DATE	Description Of Change
0	xxxx	xx-xx-xxxx	Pilot
1	xxxx	xx-xx-xxxx	Pilot
2	xxxx	xx-xx-xxxx	Released to Manufacturing
3	xxxx	xx-xx-xxxx	BCLK connected to 8Mhz Added a PULL-UP Resistor for CHRDY
4	xxxx	03-15-97	Silkscreen composition change

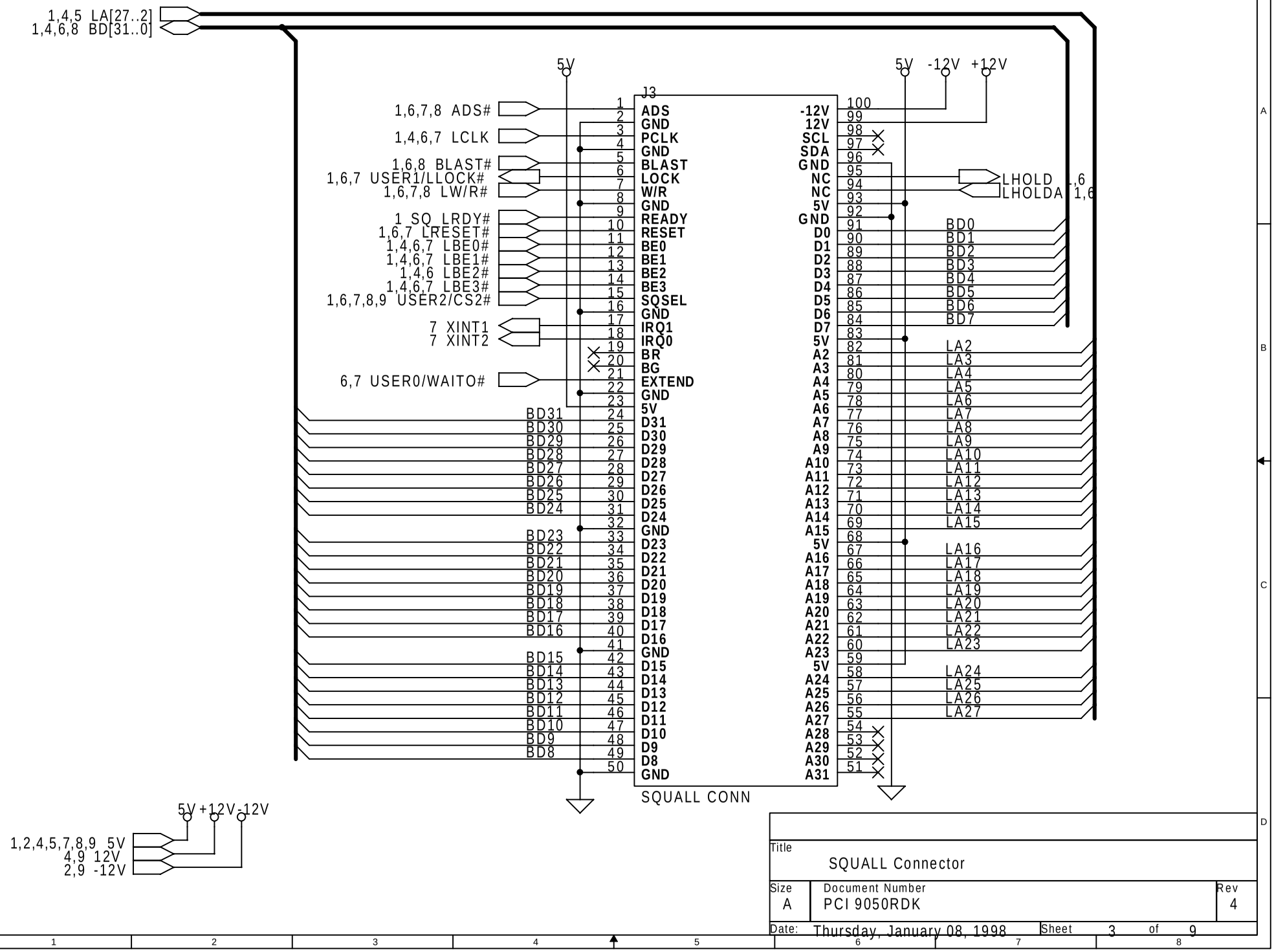
Changes from Rev. 2 to Rev 3.:

1. BCLK (ISA Connector B20) connected to 8MHZ
2. A Pull Up resistor for the CHRDY(ISA Connector A10)

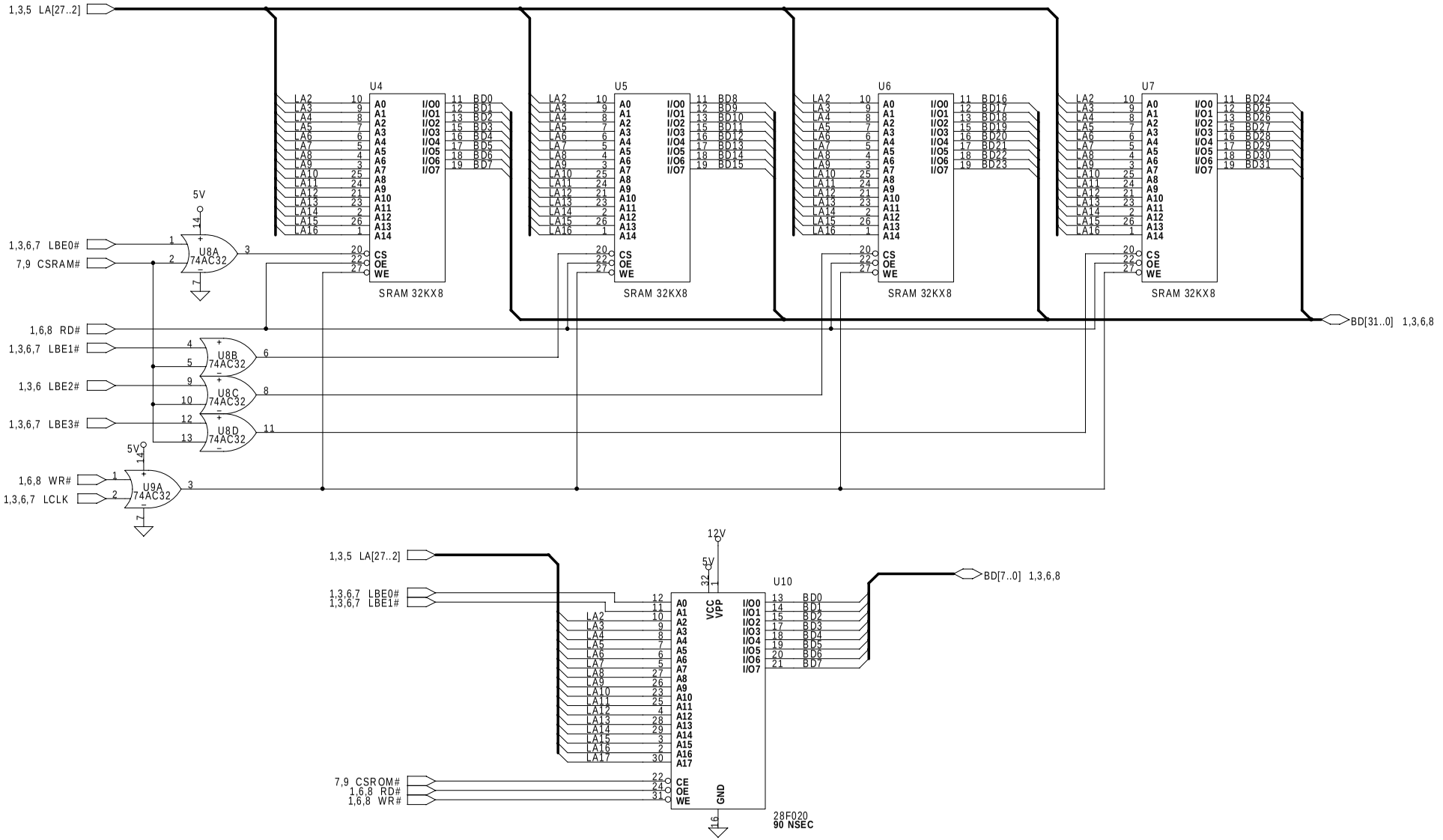
Title	PCI 9050		
Size	Document Number PCI 9050RDK		Rev 4
Date:	Thursday, January 08, 1998	Sheet 1 of 9	



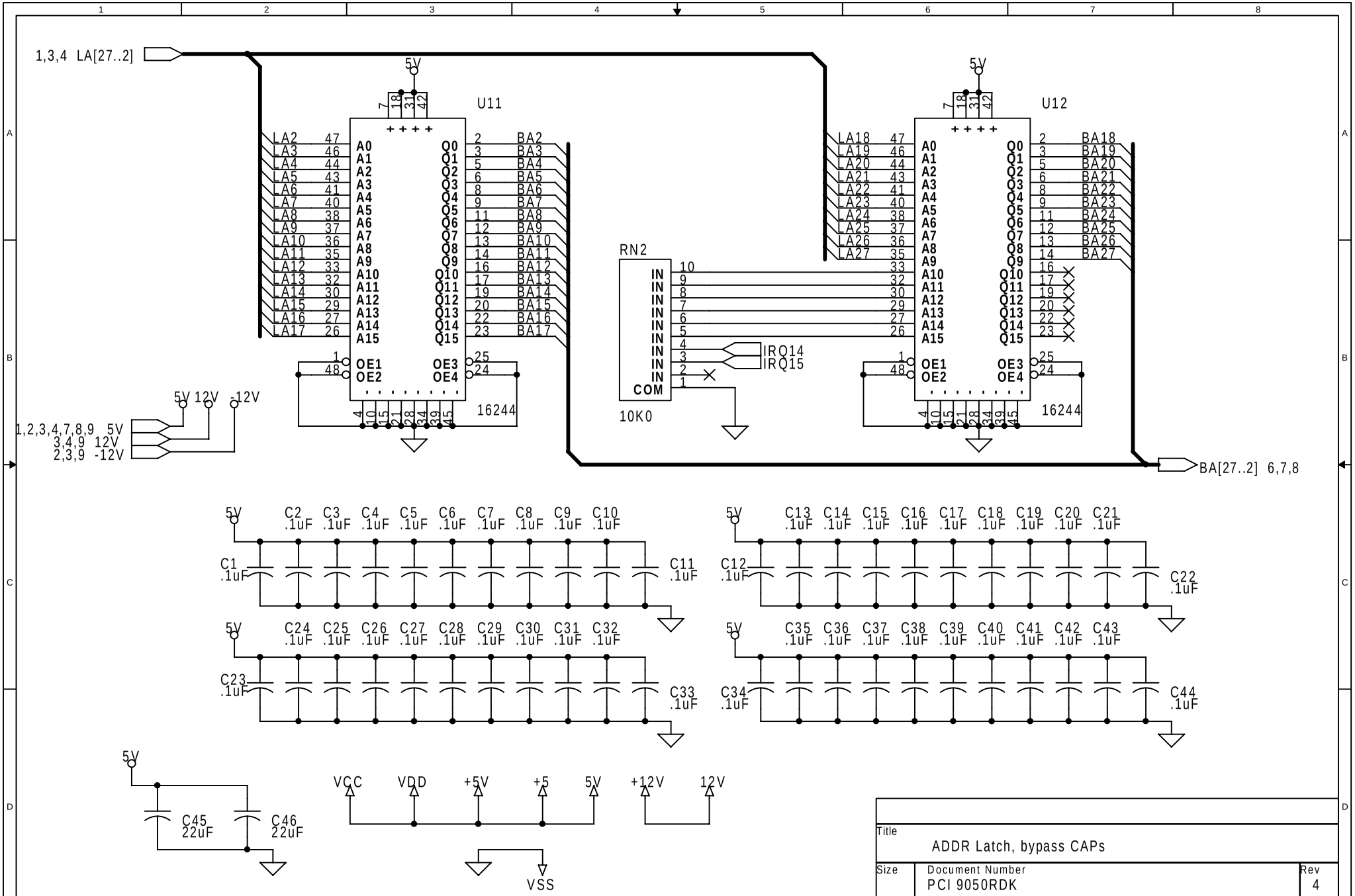
Title	PCI BUS CONNECTOR	
Size	Document Number: PCI 9050RDK	Rev: 4
Date: Thursday, January 08, 1998	Sheet 2 of 9	



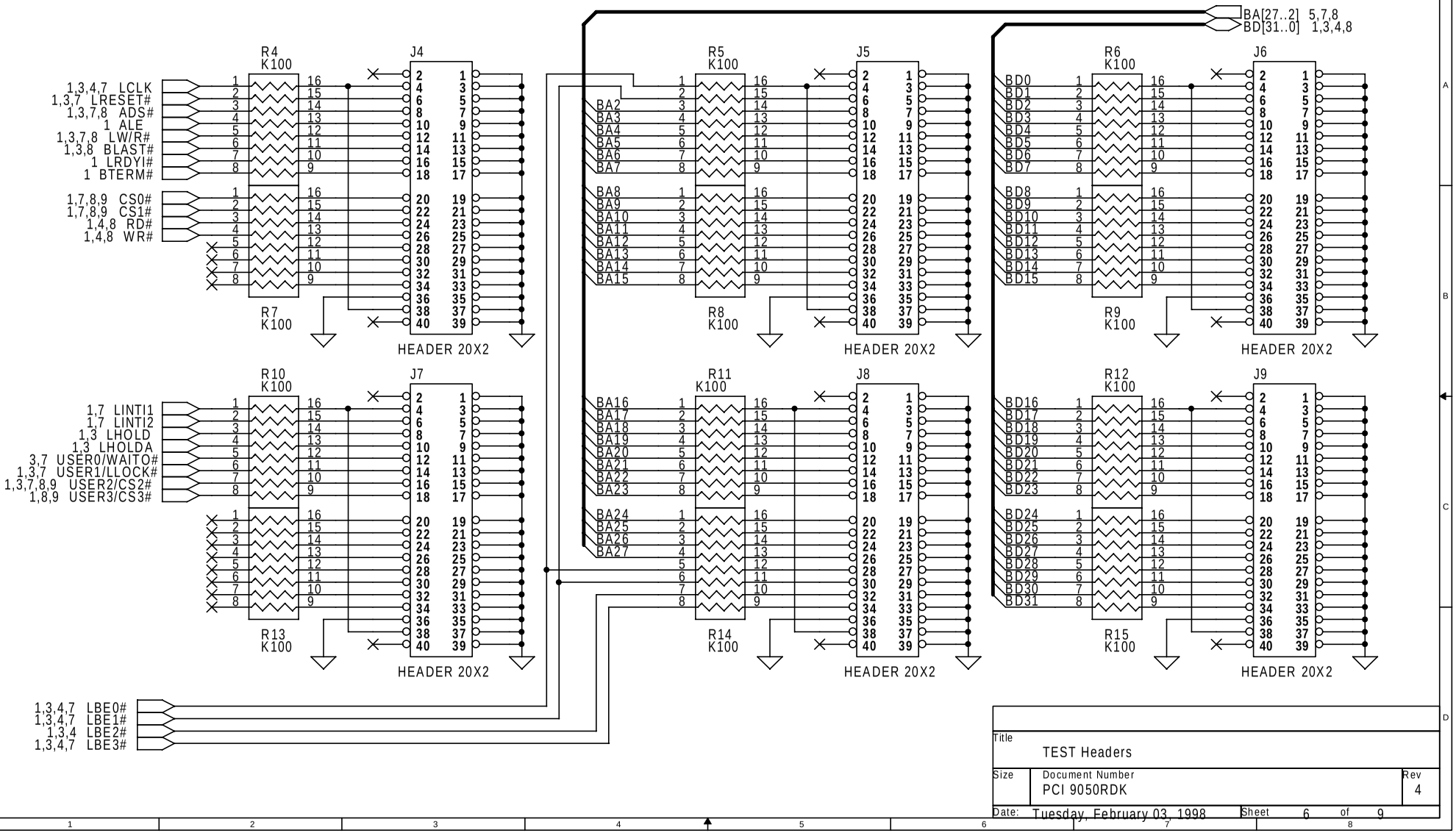
Title	SQUALL Connector	
Size: A	Document Number: PCI 9050RDK	Rev: 4
Date: Thursday, January 08, 1998	Sheet 3 of 9	

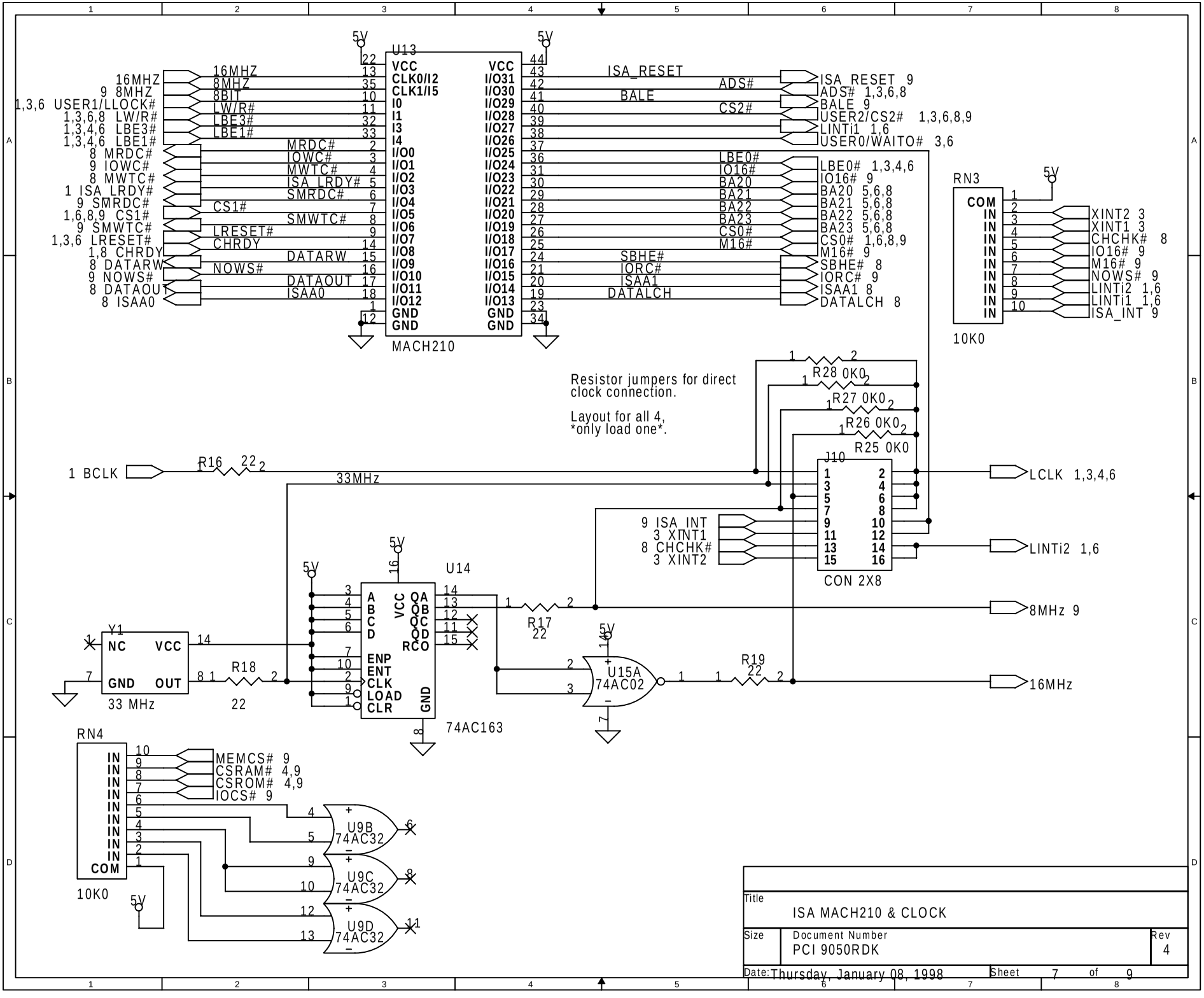


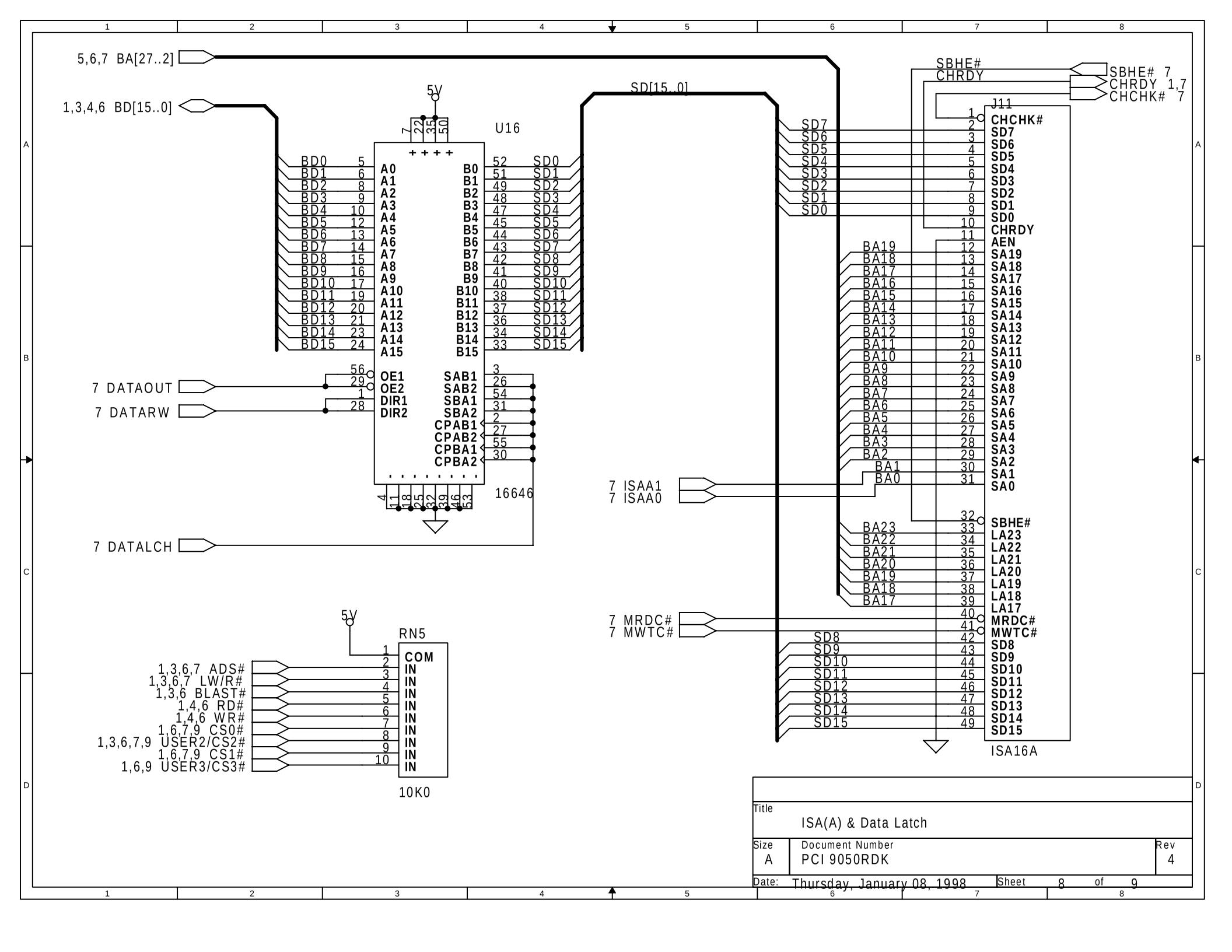
Title	SRAM & EEPROM	
Size	Document Number: PCI 9050RDK	Rev: 4
Date: Thursday, January 08, 1998	Sheet 4 of 9	



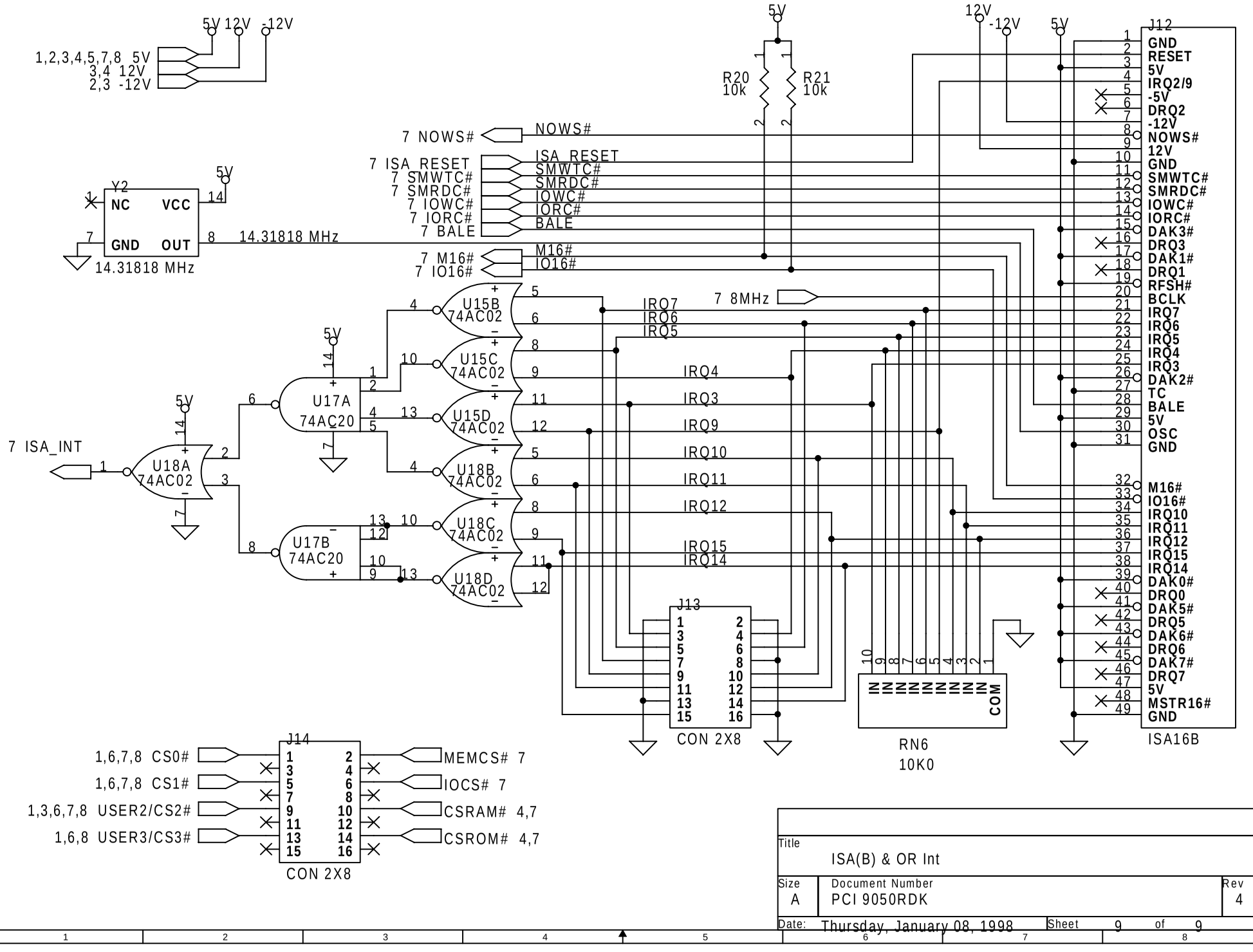
Title	ADDR Latch, bypass CAPs	
Size	Document Number: PCI 9050RDK	Rev: 4
Date: Thursday, January 08, 1998	Sheet 5 of 9	


BA[27..2] 5,7,8
BD[31..0] 1,3,4,8
R4
K100
J4
1,3,4,7 LCLK
1,3,7 LRESET#
1,3,7,8 ADS#
1 ALE
1,3,7,8 LW/R#
1,3,8 BLAST#
1 LRDYI#
1 BTERM#
1,7,8,9 CS0#
1,7,8,9 CS1#
1,4,8 RD#
1,4,8 WR#
R7
K100
HEADER 20X2
R5
K100
J5
BA2
BA3
BA4
BA5
BA6
BA7
BA8
BA9
BA10
BA11
BA12
BA13
BA14
BA15
R8
K100
HEADER 20X2
R6
K100
J6
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
BD8
BD9
BD10
BD11
BD12
BD13
BD14
BD15
R9
K100
HEADER 20X2
R10
K100
J7
1,7 LINTI1
1,7 LINTI2
1,3 LHOLD
1,3 LHOLDA
3,7 USER0/WAITO#
1,3,7 USER1/LLOCK#
1,3,7,8,9 USER2/CS2#
1,8,9 USER3/CS3#
R13
K100
HEADER 20X2
R11
K100
J8
BA16
BA17
BA18
BA19
BA20
BA21
BA22
BA23
BA24
BA25
BA26
BA27
R14
K100
HEADER 20X2
R12
K100
J9
BD16
BD17
BD18
BD19
BD20
BD21
BD22
BD23
BD24
BD25
BD26
BD27
BD28
BD29
BD30
BD31
R15
K100
HEADER 20X2
1,3,4,7 LBE0#
1,3,4,7 LBE1#
1,3,4 LBE2#
1,3,4,7 LBE3#
Title
TEST Headers
Size
Document Number
PCI 9050RDK
Rev
4
Date: Tuesday, February 03, 1998
Sheet 6 of 9


U13
MACH210
16MHZ
8MHZ
8BIT
LW/R#
LBE3#
LBE1#
MRDC#
IOWC#
MWTC#
ISA_LRDY#
SMRDC#
CS1#
SMWTC#
LRESET#
CHRDY
DATARW
NOWS#
DATAOUT
ISAA0
ISA_RESET
BALE
ADS#
CS2#
LBE0#
IO16#
BA20
BA21
BA22
BA23
CS0#
M16#
SBHE#
IORC#
ISAA1
DATALCH
RN3
10K0
XINT2 3
XINT1 3
CHCHK# 8
IO16# 9
M16# 9
NOWS# 9
LINTi2 1,6
LINTi1 1,6
ISA_INT 9
Resistor jumpers for direct clock connection.
Layout for all 4, *only load one*.
R28 0K0
R27 0K0
R26 0K0
R25 0K0
J10
CON 2X8
LCLK 1,3,4,6
LINTi2 1,6
8MHz 9
16MHz
9 ISA_INT
3 XINT1
8 CHCHK#
3 XINT2
1 BCLK
R16 22
33MHz
Y1
33 MHz
R18 22
U14
74AC163
R17 22
U15A
74AC02
R19 22
RN4
10K0
MEMCS# 9
CSRAM# 4,9
CSROM# 4,9
IOCS# 9
U9B
U9C
U9D
74AC32
Title
ISA MACH210 & CLOCK
Size
Document Number
PCI 9050RDK
Rev
4
Date: Thursday, January 08, 1998
Sheet 7 of 9



Title	ISA(A) & Data Latch	
Size: A	Document Number: PCI 9050RDK	Rev: 4
Date: Thursday, January 08, 1998	Sheet 8 of 9	



Title: ISA(B) & OR Int		
Size: A	Document Number: PCI 9050RDK	Rev: 4
Date: Thursday, January 08, 1998	Sheet 9 of 9	

PLX Technology, Inc.
390 Potrero Ave.
Sunnyvale, CA. 94086 USA
Tel: 1-800-759-3735
Fax: 1-408-774-2169
Email: info@plxtech.com
Web and FTP Site:
www.plxtech.com

Order Number: 9050RDK-M002