



ZLP323ICE01ZEM

# Crimzon™ In-Circuit Emulator

QS003906-1204

**Startup Guide**

---

## Introduction

Thank you for purchasing the ZiLOG Crimzon™ In-Circuit Emulator (ICE). The Crimzon ICE provides ZLP32300 family chip emulation with a trace and event system for program debugging using ZDS II development tools. Once your code is complete, use the included OTP programming module to burn your design to OTP devices.

This startup guide tells you how to:

1. Install ZDS II software.
2. Configure the Crimzon ICE for connection to your PC.
3. Use a supplied target POD to connect the Crimzon ICE to a target board with a 20-, 28-, or 40-PDIP socket. (Converters for 20- and 28-SOIC and 20-, 28, and 48 SSOP sockets are included.)
4. Connect the Crimzon ICE to a supplied OTP programming module.
5. Use the supplied OTP programming module with ZDS II software to program a 20-, 28-, or 40-PDIP ZLP32300 family device. Adapters for 20- and 28-SOIC and 20-, 28-, and 48 SSOP packages are included.)
6. Run a demonstration program to verify proper operation, illustrate basic operation of the trace and event system, and burn a ZLP32300 OTP device using the OTP programming module.

## Software Requirements

Table 1 lists the PC requirements for running ZDS II.

**Table 1. ZDS II System Requirements**

<b>Recommended Configuration</b>	<b>Minimum Configuration</b>
<ul style="list-style-type: none"><li>• PC running MS Windows XP, SP1</li><li>• Pentium III/500 MHz processor</li><li>• 128 MB RAM</li><li>• 110 MB hard disk space</li><li>• Super VGA video adapter</li><li>• CD-ROM drive</li><li>• Ethernet port</li><li>• One or more RS-232 communications ports</li><li>• Internet browser (Internet Explorer or Netscape)</li></ul>	<ul style="list-style-type: none"><li>• PC running MS Windows 98SE/WinNT 4.0–SP6/Win2000–SP3/WinXP–SP1</li><li>• Pentium II/233 MHz processor</li><li>• 96 MB RAM</li><li>• 25 MB hard disk space (documentation not included)</li><li>• Super VGA video adapter</li><li>• CD-ROM drive</li><li>• Ethernet port</li><li>• One or more RS-232 communications ports</li><li>• Internet browser (Internet Explorer or Netscape)</li></ul>

## Install the Software

Follow these steps to install ZDS II with the ANSI C-Compiler.

1. Insert the ZDS II CD into your computer's CD-ROM drive. *DemoShield* launches automatically. If it does not automatically launch, go to the root of the CD-ROM and double-click the file launch.exe.
2. *DemoShield* provides several installation choices. Select "Install ZDS II" to install now. You can install other software and accompanying documentation later.
3. Follow the instructions on the screen to complete the installation.
4. To receive free technical support, please register your software at <http://www.zilog.com>. Access the registration page by opening the Support menu at the top of the web page and clicking "Product Registration."

## Install the Hardware

The Crimzon™ In-Circuit Emulator features an Ethernet interface and an RS-232 serial port. Hardware installation consists of:

- Installing a target POD into a target development board; into the 20-, 28-, or 40-PDIP socket on the Crimzon™ RC Development Platform board;
- Connecting the Crimzon ICE to the target POD;
- Connecting the Crimzon ICE to the OTP programming module; and
- Connecting the Crimzon ICE to a PC.

You may have to reconfigure network settings on the PC or on the Crimzon ICE before using the emulator.

### Installing a Target POD onto a Target Development Board

The Crimzon ICE kit comes with a Crimzon™ RC Development Platform (Figure 1) for use as a target development board. The 20-, 28, and 40-PDIP target PODs plug into the associated PDIP sockets on the development platform.

If you are using a different target development board, use an appropriate target POD and pin converter to connect the Crimzon ICE to the board. For example, if your target board has a 20-SOIC socket, mate the 20-PDIP target POD onto the 20-PDIP to 20-SOIC converter. Then install the target POD and converter assembly into the board's 20-SOIC socket.

► **Note:** If you are not using a target development board, insert the 34-pin null target connector into Crimzon ICE target interface connector P16.

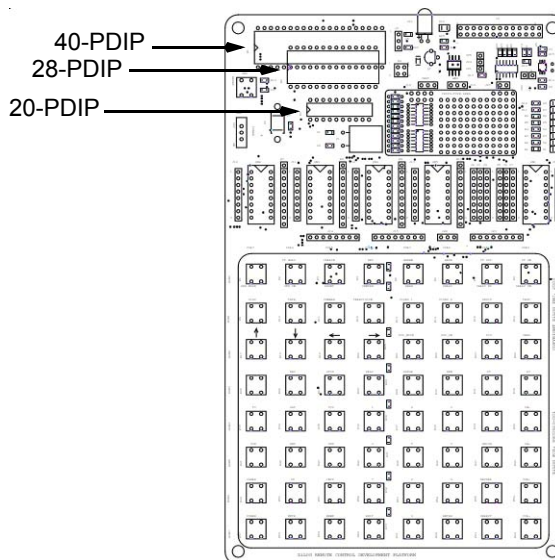


Figure 1. Crimzon™ RC Development Platform Target Board

## Connecting the Crimzon ICE to the Target POD

Once you have installed the appropriate target POD (and converter, if required) onto the target development board, connect the Crimzon ICE to the target POD as follows:

- 40-PDIP target POD:
  - Connect the 16-circuit cable from P17 on the emulator to P17 on the 40-PDIP target POD.
  - Connect the 34-circuit cable from P16 on the emulator to P16 on the 40-PDIP target POD.
- 20-PDIP and 28-PDIP target PODs: Connect the 34-circuit cable from P16 on the emulator to P16 on the target POD. (Emulator connector P17 is not used.)

## Connecting the Crimzon ICE to the OTP Programming Module (Optional)

After developing and debugging your software, use the following instructions to connect the Crimzon ICE to the OTP programming module so you can burn your code onto chips.

1. Connect the 40-circuit ribbon cable from the Crimzon ICE OTP Programming connector to connector P1 on the OTP programming module.
2. The 40-PDIP ZIF socket on the OTP programming module is designed to accept 40-PDIP OTP chips. The OTP programming adapters supplied with the Crimzon ICE allow you to adapt the ZIF socket to accept 20-SOIC, 20-SSOP, 20-PDIP, 28-SOIC, 28-SSOP, 28-PDIP, and 48-SSOP chip packages.

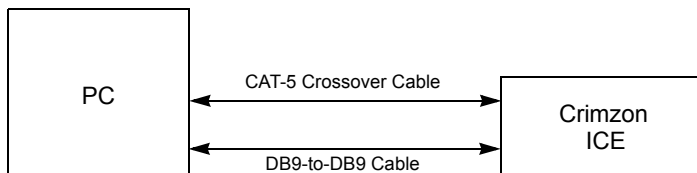
Once you have installed the OTP chip into the ZIF socket (or programming adapter), you are ready to program the chip using the instructions in “OTP Programming” on page 19.

## Connecting the Crimzon ICE to a PC

1. Connect a CAT-5 crossover cable from the PC to the Ethernet port on the Crimzon ICE. See Figure 2.

► **Note:** If you prefer, you can connect the emulator to an Ethernet hub using a standard CAT-5 patch cable.

2. Connect the serial COM port on the PC to the SETUP serial port on the Crimzon ICE using the DB9-to-DB9 serial cable. See Figure 3.

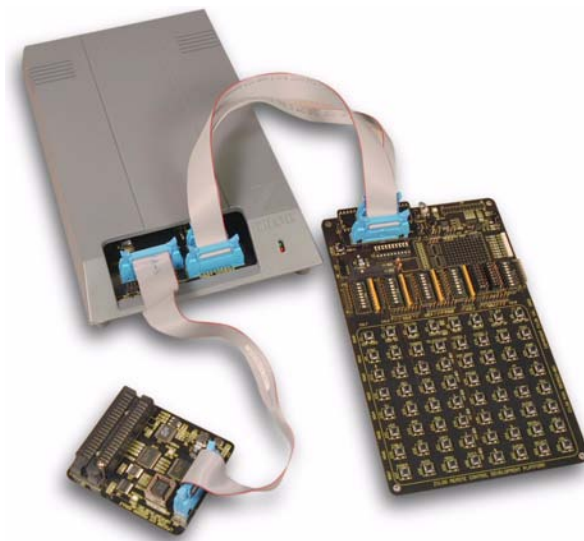


**Figure 2. Connecting a PC to the Crimzon ICE**

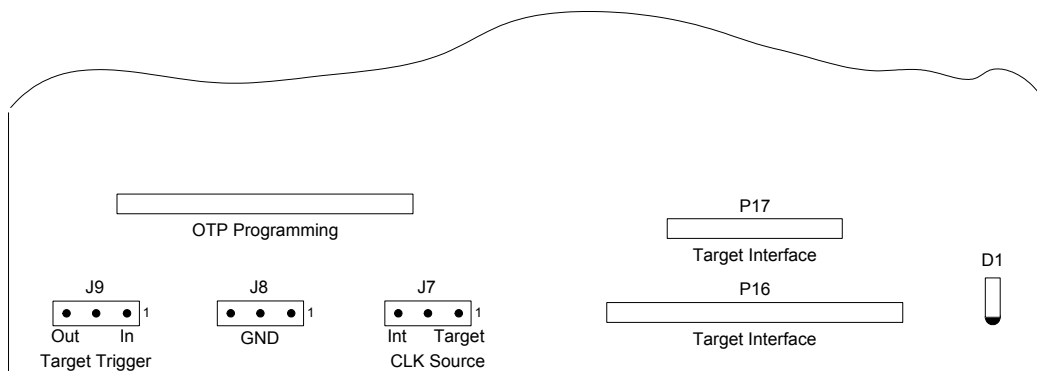


**Figure 3. Crimzon ICE Rear Panel**

3. Connect a 5VDC power supply to the Crimzon ICE. The 3.3VDC should illuminate (see Figure 3). If the 3.3VDC LED failed to illuminate or the ICE Fail LED illuminates (see Figure 6), there is a problem with the unit. Contact ZiLOG support at <http://www.zilog.com> for a replacement unit.
4. Insert two AAA batteries into the battery holder on the bottom of the Crimzon™ RC Development Platform. Check to make sure there is good battery contact.



**Figure 4. Connecting the Crimzon ICE to the Target  
(Typical Connection Shown)**



**Figure 5. Crimzon ICE Top View**

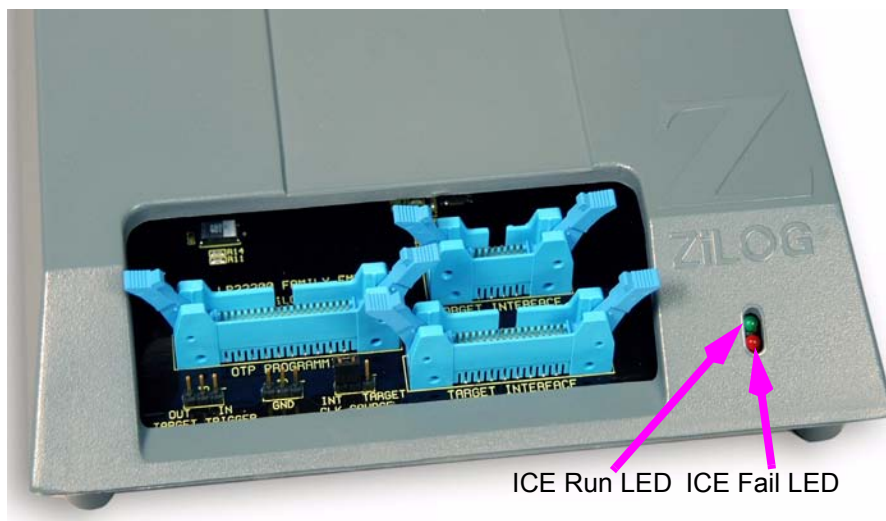


Figure 6. Crimzon ICE Front-Panel

## Configure the Hardware

Configuring the Crimzon ICE consists of selecting emulator jumper options and setting up Ethernet communications between the emulator and your PC.

### Setting Jumpers on the Crimzon ICE

There is one jumper on the Crimzon ICE. Jumper J7 allows you to select whether the emulator uses the target board clock or is programmable using the settings in ZDS II.

Table 2. Jumper J7 Settings on the Crimzon ICE

Emulator...	Jumper Position
uses the target clock.	1 - 2 (default)
uses the internal ZDS II programmable clock.	2 - 3



## **Setting Up Ethernet Communications**

The default IP address and subnet mask of the Crimzon ICE are 192.168.1.50 and 255.255.255.0, respectively. To enable communication between the PC running ZDSII and the Crimzon ICE, you must either change the PC's Ethernet settings to match those of the Crimzon ICE or vice versa.

If using the PC in a stand-alone configuration, set the PC's IP address to 192.168.1.21 and its subnet mask to 255.255.255.0. See "Changing the PC's Settings to Match the Crimzon ICE" on page 9.

In a networked environment, set the Crimzon ICE IP address and subnet mask to match the network setup. See "Changing Crimzon ICE Settings to Match the PC" on page 13.



## Changing the PC's Settings to Match the Crimzon ICE

After completing the following steps to change the PC's Ethernet settings, proceed to [Running a Sample Project](#) on page 15.

► **Note:** The following instructions are for MS Windows XP. If your Windows operating system is different, refer to your MS Windows OS online help for details.

1. Open the Windows Control Panel and double-click the Network and Internet Connections icon. The Network Connections dialog box appears (see Figure 7).

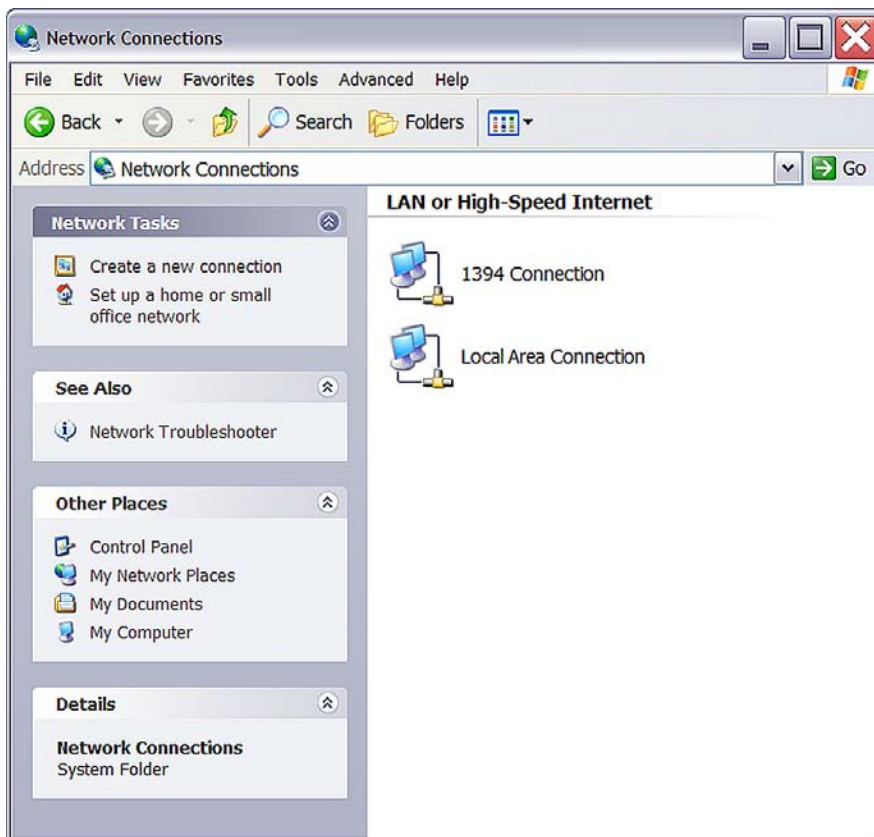


Figure 7. The Network Dialog

2. In the panel labeled LAN or High-Speed Internet, double-click the Local Area Connection icon. The Local Area Connection Status dialog box appears (Figure 8).

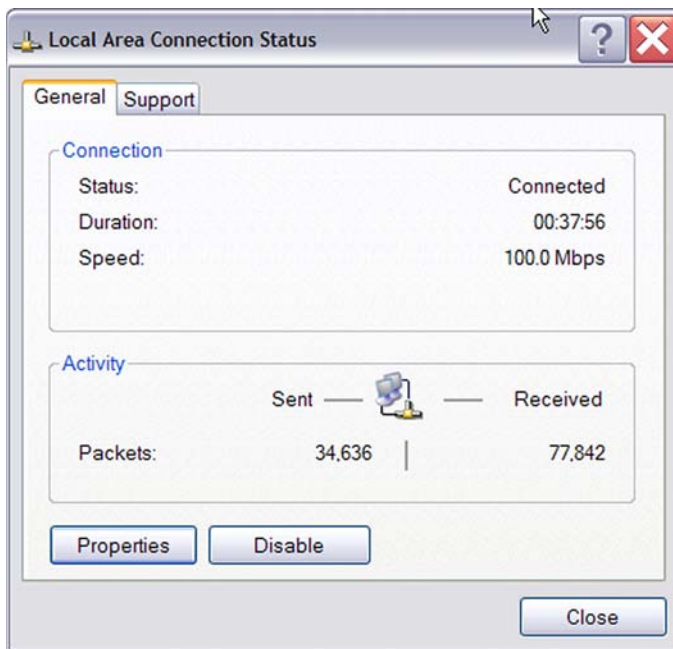


Figure 8. The Local Area Connection Status Dialog

3. In the Local Area Connection Status dialog box, click the Properties button. The Local Area Connection Properties dialog box appears (Figure 9).

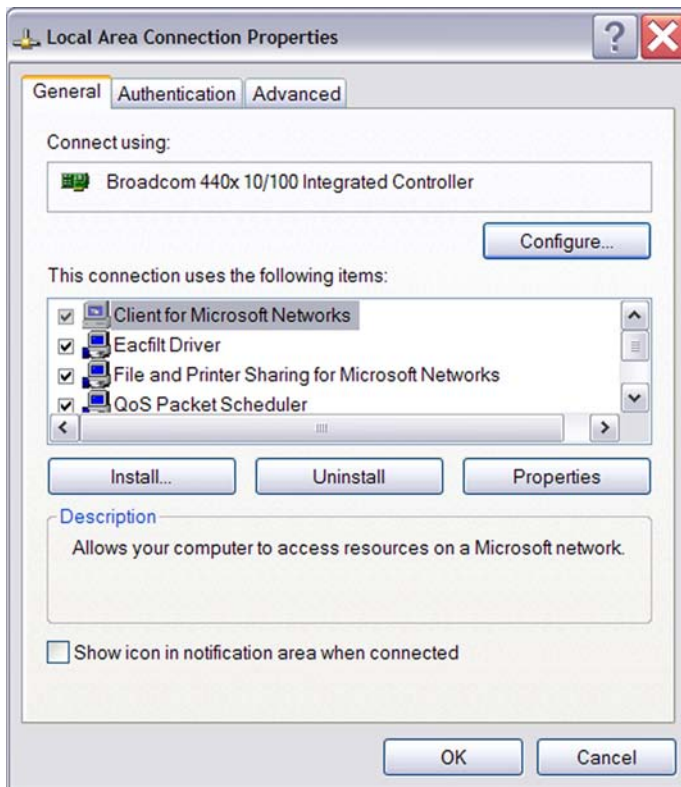
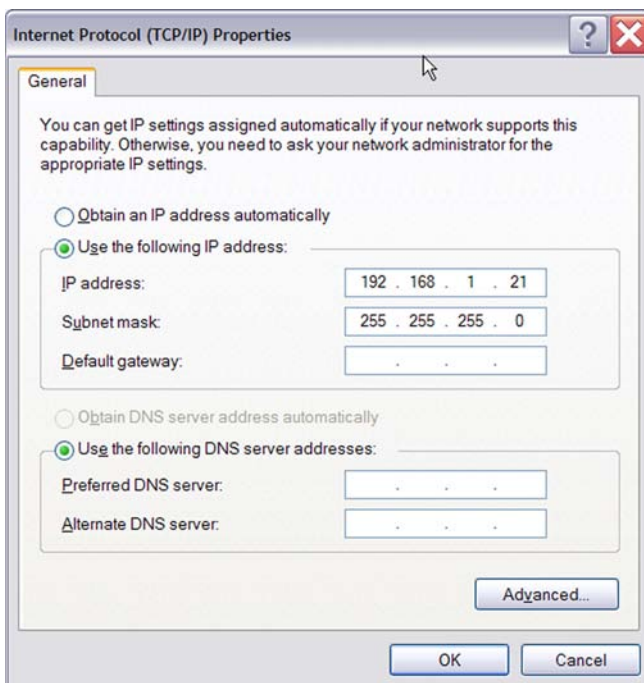


Figure 9. The Local Area Connection Properties Dialog

4. In the panel labeled This connection uses the following items:, select the Internet Protocol (TCP/IP) item to highlight it, and click the Properties button. The Internet Protocol (TCP/IP) Properties dialog box appears (Figure 10).



**Figure 10. The Internet Protocol Properties Dialog**

5. Enter values for the IP address and subnet mask to match those shown in Figure 4. Leave any remaining fields blank. In this example, an IP address of 192.168.1.21 and a subnet mask of 255.255.255.0 are being assigned to the PC. These values place the PC on the same network as the Crimzon ICE unit.
6. Click OK and restart the PC.
7. Proceed to “Running a Sample Project” on page 15.

## Changing Crimzon ICE Settings to Match the PC

1. Connect the PC serial port to the Crimzon ICE serial port using the DB9-to-DB9 serial cable.
2. Launch HyperTerminal on the PC by selecting Start --> Programs --> Accessories --> Communications --> HyperTerminal.
3. In the Connect To dialog, set the Connect Using: drop-down menu to match the COM port to which the Crimzon ICE is connected. Click OK.
4. A COM Properties dialog appears. Enter the following port settings and click OK.
  - Bits per second:57600
  - Data bits:8
  - Parity:None
  - Stop bits:2
  - Flow control:None
5. HyperTerminal should automatically attempt a connection. If not, select Call --> Connect.
6. When the emulator is turned on or reset, HyperTerminal displays a Crimzon ICE console boot-up message in HyperTerminal. A typical boot-up message is shown below.

```
ZiLOG TCP/IP Software Suite v1.1  
  
Copyright (C) 2003 ZiLOG Inc.  
All Rights Reserved  
  
clock enabled  
IP Address: 192.168.1.50  
IP Subnet: 192.168.1.0/255.255.255.0  
IP Gateway: 192.168.1.254  
  
Z8 ICE Firmware x.x (Dec 5 2003 13:45:05)  
Copyright (C) 2001-2003 ZiLOG, Inc.  
All Rights Reserved.
```

```
Press 'Ctl-Z' to enter configuration mode
```

7. Press ctrl-z. The emulator command prompt displays:

```
eth1 %
```

► **Note:** The emulator console prompt is not case-sensitive.
8. Type help or ? at the emulator command prompt to see a list of available commands. The following list displays:

Commands are:

?	debugport	devs	exit
help	ifconfig	kbuf	kill
mem	password	ps	reboot
restore			
eth1 %			

Command usage is described in Table 3.

**Table 3. Crimzon ICE Commands**

<b>Command</b>	<b>Description and Options</b>
?	Displays available emulator command shell options.
debugport	Configures the TCP port. Usage: debugport – displays current setting debugport tcp_port – sets debugport to specified TCP port.  Example: debugport 4040  sets debugport to TCP port 4040.
devs	not used
exit	Exits the command shell.
help	Displays available emulator command options.
ifconfig	Configures the emulator network interface. Entering ipconfig with no options lists current configuration.  The following command options are available: <ul style="list-style-type: none"><li>• i – specifies an IP address</li><li>• s – specifies a subnet mask</li><li>• g – specifies a network gateway address</li><li>• dhcp – configures the emulator network interface to look for a dhcp host to obtain network settings</li></ul> Example:  ifconfig i 192.168.1.1 s 255.255.255.0 g 192.168.1.254  configures the emulator to use IP address 192.168.1.1 on subnet 255.255.255.0 with gateway address 192.168.1.254.
kbuf	unused
kill	unused

**Table 3. Crimzon ICE Commands**

Command	Description and Options
mem	unused
password	unused
ps	unused
reboot	Reboots the emulator.
restore	Restores factory default network interface settings.

9. When you have finished configuring the emulator, type `exit` to exit the command shell.
10. Exit HyperTerminal.
11. Cycle the power on the Crimzon ICE for the new settings to take effect.
12. The hardware is now configured and ready for application development.

## Running a Sample Project

After installing the ZDS II software and setting up the hardware, you can run a sample software project to verify proper emulator operation and experiment with the trace and event system. The sample project `irremote.pro` is included in the ZDS II sample directory, located in:



```
c:\Program Files\ZiLOG\ZDSII_<product>_<version>  
  \samples\<processor type>_<demo name>
```

Start ZDS II for the Crimzon ICE Emulator and follow the instructions below to run `irremote.pro`.

1. File --> Open Project--> `c:\Program Files\ZDSII_Crimzon_Emulator_<version>\samples\ZLX32300_IRRemote\src\irremote.pro`.
2. Double-click `irmain.s` in the Project Files window.
3. Open Project --> Settings.
4. In the General tab, verify the CPU Family field is set to ZLP32300.
5. In the General tab, select a CPU type.
6. In the Debugger tab, select Emulator Driver and then click the Configure Driver button.

7. The Emulator Driver dialog box appears. The IP Address field displays a default IP address, 192.168.1.50. Enter the Crimzon ICE IP address if it has been modified. Leave the other settings as they are.
8. Select the voltage for the target. If the emulator is connected to the Crimzon™ RC Development Platform board, select 2.7V. If running the emulator in standalone mode, select Standalone and insure that the null target connector is installed in emulator connector P16. If the emulator is connected to a custom target, select the voltage appropriate for that target.
  - ▶ **Note:** The emulator's voltage comparator is designed as a target power sensor, not as a precision voltage measurement device. If you set the Target VCC to match your target and the target's voltage drifts downward, the power sensor may no longer detect it. The emulator may therefore not connect to the target. In such cases, set the Target VCC voltage progressively lower until you get a good connection.
9. Select the Target Clock Frequency. For the Crimzon™ RC Development Platform, select External.
10. Click OK to close the Emulator Driver dialog box.
11. An IDE dialog box appears acknowledging that project settings have changed since the last build and asking whether it should rebuild the files. Click NO.
12. Click OK to close the Project Settings dialog box.
13. Build the project now by selecting Build --> Build, or by pressing F7.
  - ▶ **Note:** The following steps describe two ways to use the trace and event system. For details on running the trace and event system, refer to the ZDS II online help and the *ZDS II—Crimzon User Manual* (UM0146), located in the docs directory of the ZDS II CD-ROM.

### Collecting a Simple Trace

14. Now we'll collect a simple trace by starting the program, then stopping it and viewing the trace buffer. Click the Go button () in the toolbar, wait a moment, and then press the Stop button (). The trace buffer acts as a ring buffer that continuously fills and then overwrites itself until you stop execution. Click Get Frames to display the trace information.



## Using an Event to Stop Execution

Events allow you to stop execution based on more complex conditions than a simple instruction address.

The following events are available:

- Program counter position, with mask.
- Data on Port0 (state of its pins), with mask.
- Data on Port2 (state of its pins), with mask.
- Data on Port3 (state of its three input pins), with mask.
- External Trigger In (0 or 1).

Let's define a simple event and see how it works.

1. Select Tools --> Trace and Event System. The Trace and Event System window appears (Figure 11).

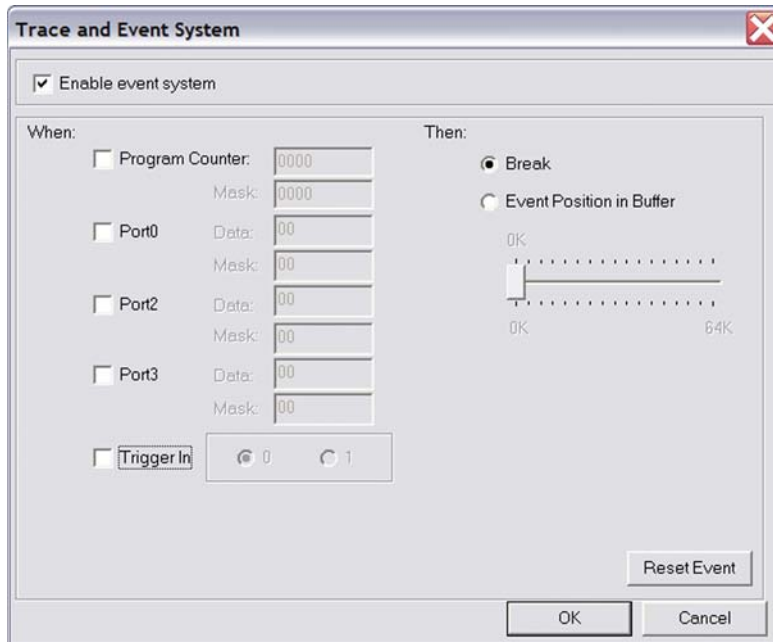


Figure 11. Trace and Event System Window

2. Click the Enable Event System checkbox, and click the Break checkbox.
3. In the When: section, click the Program counter checkbox and set the Program Counter to 004D and the Mask to FFFF.
4. In the Then: section, click the Break radio button.
5. Click OK.
6. Open the Trace window by selecting View --> Debug Windows --> Trace.
7. In the Trace window, click the Clear Trace button.
8. Reset the Debugger by clicking the Reset button in the toolbar, or by selecting Build --> Debug --> Reset.
9. Run the Debugger by clicking the Go button or by selecting Build --> Debug --> Go.
10. When the program counter reaches 004D, execution stops on event match.
11. Click Get Frames to display the trace information. Study the contents of the Trace window to see how the trace and event system reports program execution for the segment we set using the Event tools.

### **Collecting Trace Before and After an Event**

You can also use the Trace and Event System to capture trace data before and after an event. Set up your events as described in “Using an Event to Stop Execution” on page 17. In the Then: section, click the Event Position in Buffer radio button instead of Break. Use the slider to select the number of cycles from the 64K buffer to be captured before and after the event.


In this case, all cycles are traced until the event is detected, then the selected number of cycles after the event are collected. Execution stops after the cycles are collected. What remains in the trace buffer are the selected number of cycles after the event. The remainder of the 64K frames contain cycles before the event occurred.

If you move the slider all the way to the left, only cycles before the event are captured. If you move the slider all the way to the right, only cycles after the event are captured.

### **Single-Stepping Through a Program**

ZDS II provides a simple mechanism for single-stepping through a program. To single-step through a program:

1. Reset the program to Main() by either the Reset icon or with Build --> Debug --> Reset. Set the Reset to Main option by selecting Tools --> Options. In the Options window, select the Debugger tab and select the Reset to symbol ‘main’ checkbox.

- To step through the program one instruction at a time, use F11 or click the  button in the Debug toolbar (also accessible by selecting Build --> Debug --> Step Into).

### Peek/Poke Registers

- ZDS II makes it easy for you to set and read emulator register contents. With the irremote.pro project open and ZDS II connected to the emulator (target), select View --> Debug Windows --> Registers.
- In the Registers window, double-click the value of any register and type in a new value.
- Press Enter. The new value displays in red.

Refer to the *ZDS II – Crimzon User Manual* (UM0164) on the ZDS II CD-ROM and the ZDS II online help for further information on setting and reading register values.

### Peek/Poke Memory

- ZDS II also allow to set and read memory contents. With the irremote.pro project open and ZDS II connected to the emulator (target), select View --> Debug Windows --> Memory.
- In the Memory window, double-click the value you want to change and type in a new value. (Values begin in the second column after the Address column.)
- Press Enter. The new value displays in red.

Refer to the *ZDS II User Manual* (UM0164) on the ZDS II CD-ROM and the ZDS II online help for further information on setting, filling, and reading memory.

### OTP Programming

Once your program is running properly, use the Crimzon ICE OTP Programming Module to burn your program onto a ZLP32300 family chip. There are two ways to burn an OTP chip:

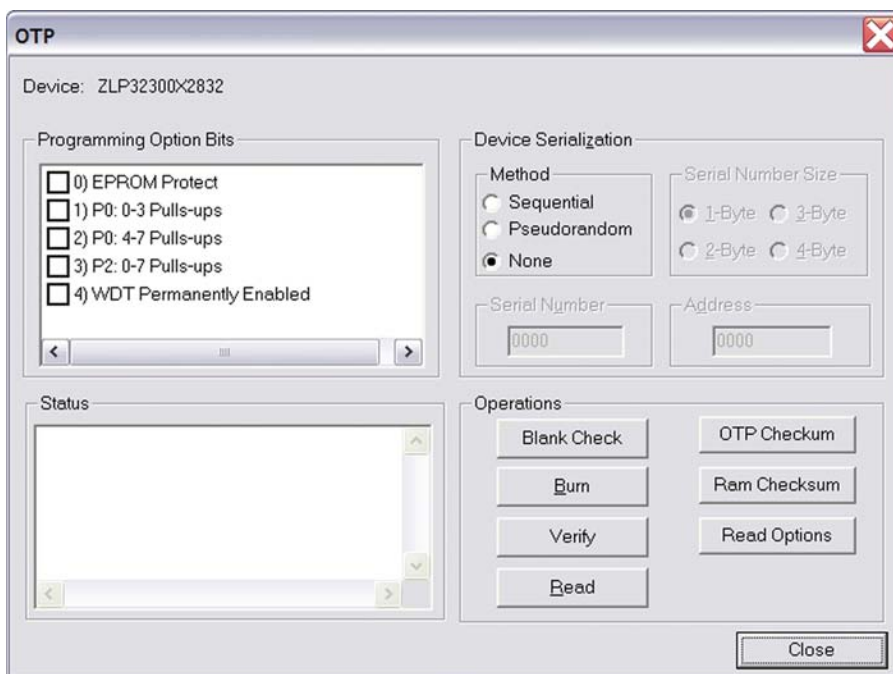
- Burn the code in the current ZDS II project from emulator RAM onto the OTP chip.
- Load an existing hex file into emulator RAM and burn it onto the OTP chip.

### Burning Code from the Current Project

To burn code from the project currently built in ZDS II (loaded in emulator RAM):

- Connect the OTP programming module to the emulator as described in “Connecting the Crimzon ICE to the OTP Programming Module (Optional)” on page 4.

2. Select the OTP chip you wish to burn and the appropriate package converter.
3. Install the package converter, if used, into the ZIF socket on the OTP programming module.
4. Install the OTP chip you wish to burn into the ZIF socket on the OTP programming adapter. Match pin 1 of the chip with pin one of the ZIF socket.
5. In ZDS II, open the project for the code you wish to burn onto the chip.
6. Select Tools --> OTP Programming to open the OTP window (Figure 12).



**Figure 12. OTP Programming Window (Current ZDS II Project Example)**

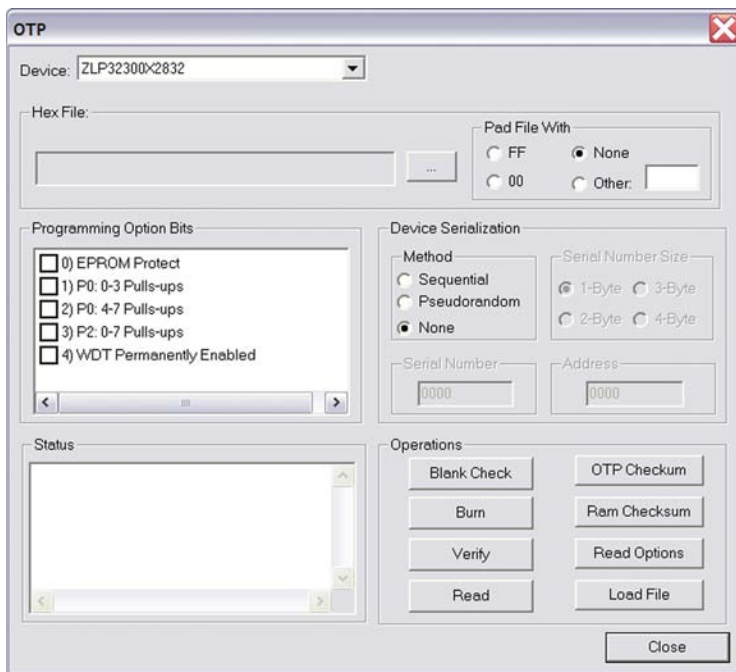
7. If the appropriate target device is not selected, set it now in the Device drop-down menu.
8. If you do not want to pad the hex file, select the None button in the Pad File With area. Otherwise, select the FF, 00, or Other button. If you select the Other button, type the hex value to pad the hex file with in the Other field.

9. Click the Ram Checksum button to calculate the checksum of the data in emulator RAM. Use this to compare with the OTP checksum after burning.
10. Select which option bits to program in the Programming Option Bits area.
11. If you do not want a serial number loaded into the part, select the None button in the Method area. To load a serial number:
  - a. Select the Sequential or Pseudorandom button to determine how the serial number is incremented on subsequent burns.
  - b. Select the size of the serial number (1, 2, 3, or 4 bytes) in the Serial Number Size area.
  - c. Enter the starting serial number in the Serial Number field.
  - d. Enter the address into which you want the serial number loaded in the Address field.
12. Click Blank Check to verify that the OTP chip is actually blank.
13. Click the Burn button to program the OTP chip with the contents of emulator RAM. As part of the Burn function, the OTP chip is also verified.
14. When the burn is complete, click OTP checksum to calculate the checksum of data on the OTP chip and compare it to the RAM checksum calculated earlier.
15. Click Close to close the OTP Programming window.


### **Burning Code from an Existing Hex File**

To load an existing hex file into emulator RAM and burn an OTP:

1. Connect the OTP programming module to the emulator as described in “Connecting the Crimzon ICE to the OTP Programming Module (Optional)” on page 4.
2. Select the OTP chip you wish to burn and the appropriate package converter.
3. Install the package converter, if used, into the ZIF socket on the OTP programming module.
4. Install the OTP chip you wish to burn into the ZIF socket on the OTP programming adapter. Match pin 1 of the chip with pin one of the ZIF socket.
5. Stop any current debugging process by selecting Build --> Debug --> Stop Debugging.
6. In ZDS II, open the project for the code you wish to burn onto the chip.
7. Select Tools --> OTP Programming to open the OTP window (Figure 13).



**Figure 13. OTP Programming Window (Hex File Example)**

8. If the appropriate target device is not selected, set it now in the Device drop-down menu.
9. In the Hex File: section, click the  button and select the hex file you wish to burn into the OTP chip.
10. If you do not want to pad the hex file, select the None button in the Pad File With area. Otherwise, select the FF, 00, or Other button. If you select the Other button, type the hex value to pad the hex file with in the Other field.
11. Click the Load File button to load the hex file into emulator RAM.
12. Click the Ram Checksum button to calculate the checksum of the data in emulator RAM. Use this to compare with the OTP checksum after burning.
13. Select which option bits to program in the Programming Option Bits area.
14. If you do not want a serial number loaded into the part, select the None button in the Method area. To load a serial number:

- a. Select the Sequential or Pseudorandom button to determine how the serial number is incremented on subsequent burns.
  - b. Select the size of the serial number (1, 2, 3, or 4 bytes) in the Serial Number Size area.
  - c. Enter the starting serial number in the Serial Number field.
  - d. Enter the address into which you want the serial number loaded in the Address field.
15. Click Blank Check to verify that the OTP chip is actually blank.
  16. Click the Burn button to program the OTP chip with the contents of emulator RAM. As part of the Burn function, the OTP chip is also verified.
  17. When the burn is complete, click OTP checksum to calculate the checksum of data on the OTP chip and compare it to the RAM checksum calculated earlier.
  18. Click Close to close the OTP Programming window.

## **LED Indicators**

There are three sets of dual LED indicators on the Crimzon ICE (see Figures 3 and 6):

- The dual ICE RUN LED on the front panel indicates emulator status. If the top LED is lit, the emulator is executing your system code. When the top LED is off, emulation has stopped. If the bottom LED is lit, the emulator is not functioning properly. Contact technical support at <http://www.zilog.com> for assistance.
- The dual 3.3VDC/1.8VDC LED on the rear panel indicates the status of internal voltages. Both LEDs are normally illuminated when power is connected.
- The dual LAN/LINK LED on the rear panel indicates Ethernet status. The LINK LED indicates that the Ethernet connection is live. The LAN LED indicates that data is being transferred across the connected network.

## **External Interface Connectors**

There are five external interface connectors on the Crimzon ICE. Connectors P16 and P17 are used to connect the emulator to the target POD and adapter board assembly. (See Figure 5.)

The OTP Programming connector P15 is used to connect the emulator to the OTP programming module.

Connector J8 on the emulator front panel (see Figures 5 and 6) provides a ground connection on all three pins.

Connector J9 on the emulator front panel (see Figures 5 and 6) provides access to the following functions:

- Pin 1 provides a 3.3VDC external trigger out for use in triggering a device such as a logic analyzer or oscilloscope. Pin 1 is under software control, and can be set to activate through the ZDS II trace and event system. The trigger can be set to toggle or pulse.
- Pin 3 provides an input for an external 3.3VDC trigger in, allowing use of an external trigger as an event for the ZDS II trace and event system.

### Using J9 Pin 1, External Trigger Out

The Crimzon ICE external trigger out feature is always enabled. Set your trace and event system parameters, then run your code. When the event you set up occurs, pin 1 of connector J9 goes active and stays active as long as the event is active. The bigger the event window, the longer trigger out stays active.

### How to Set Connector J9 Pin 3, External Trigger In (see Figure 6)

To use the Crimzon ICE external trigger in feature:

1. With the irremote.pro project open in ZDS II as described in “Running a Sample Project” on page 15, select Tools --> Trace and Event System.
2. In the Trace and Event System window, select an Event entry. In the When section, check the Trigger In box.
3. Select either 0 or 1 to trigger on trigger low or high, respectively.
4. Click the OK button to set the trace and event system parameters. When the external device you are using sets pin 3 of connector J9 to 3.3VDC, an event is generated in the ZDS II trace and event system.





This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact:

**ZiLOG Worldwide Headquarters**

532 Race Street  
San Jose, CA 95126  
Telephone: 408.558.8500  
Fax: 408.558.8300  
[www.ZiLOG.com](http://www.ZiLOG.com)

**Document Disclaimer**

ZiLOG is a registered trademark of ZiLOG Inc. in the United States and in other countries. All other products and/or service names mentioned herein may be trademarks of the companies with which they are associated.

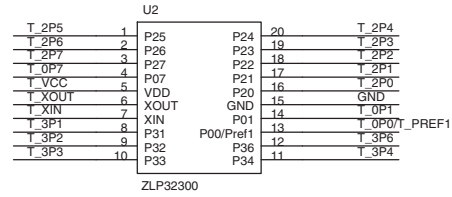
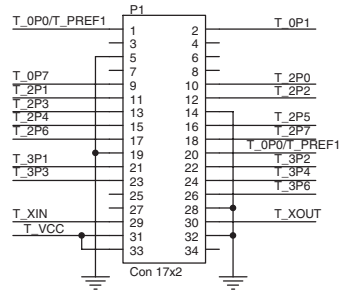
©2004 by ZiLOG, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZiLOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZiLOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Devices sold by ZiLOG, Inc. are covered by warranty and limitation of liability provisions appearing in the ZiLOG, Inc. Terms and Conditions of Sale. ZiLOG, Inc. makes no warranty of merchantability or fitness for any purpose Except with the express written approval of ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses are conveyed, implicitly or otherwise, by this document under any intellectual property rights.



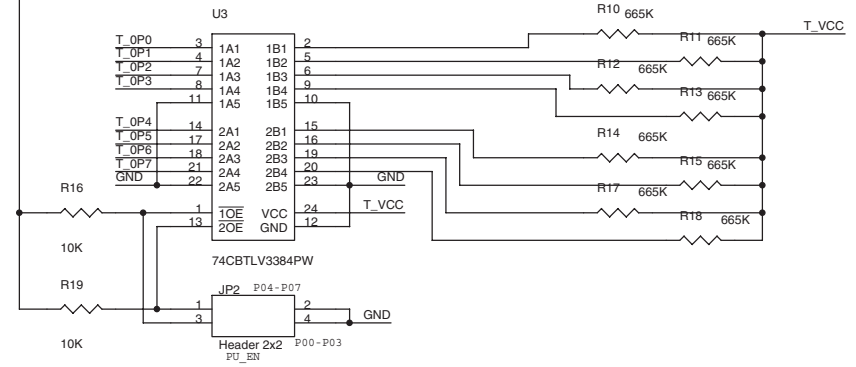
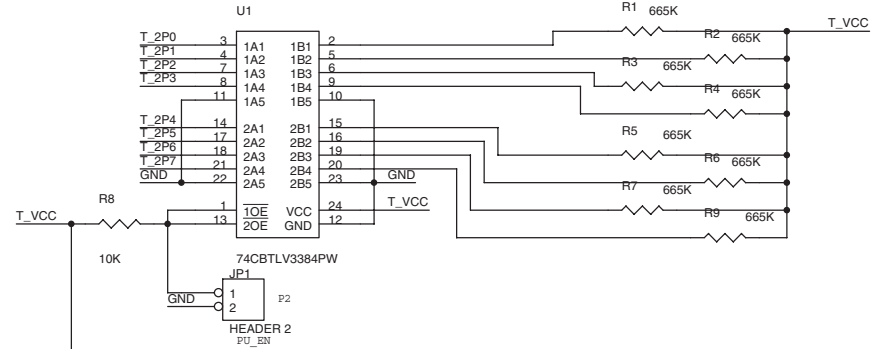
## Crimzon In-Circuit Emulator Schematic Diagrams

The following pages are schematic diagrams for the Crimzon In-Circuit Emulator containing:

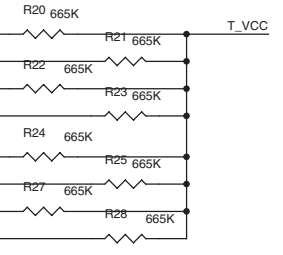
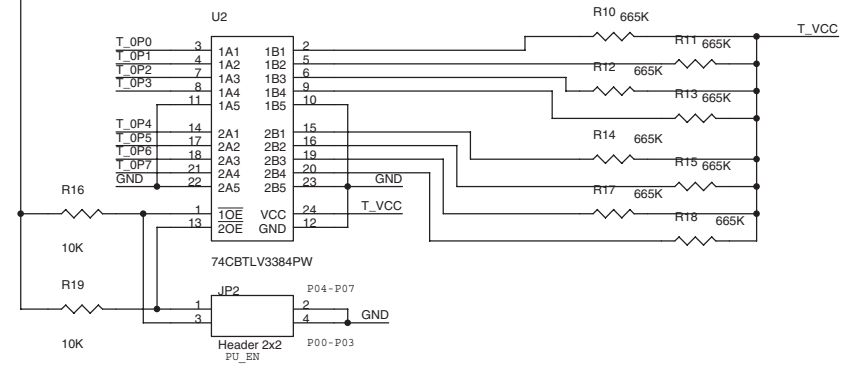
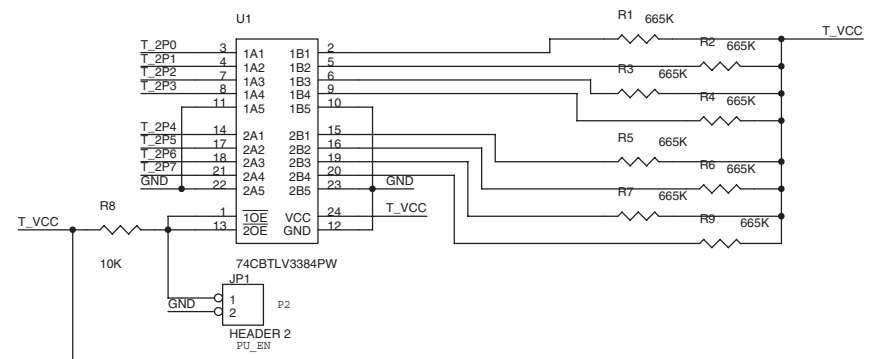
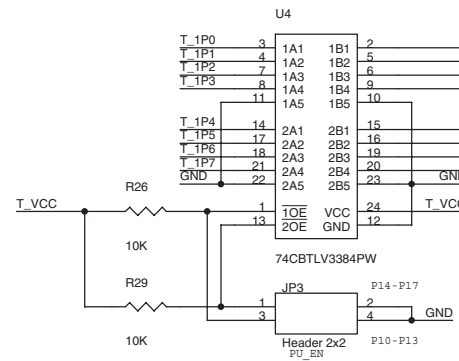
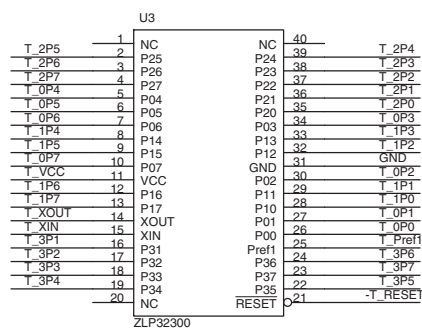
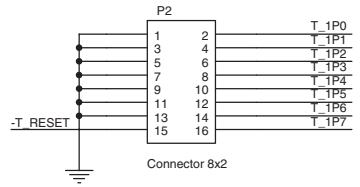
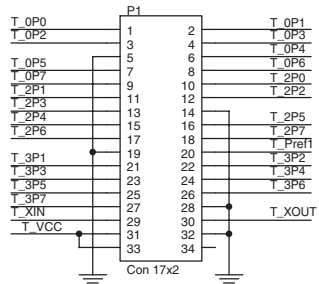
- Target POD, 20-pin with pull-ups, 1 page
- Target POD, 28-pin with pull-ups, 1 page
- Target POD, 40-pin with pull-ups, 1 page
- FPGA Board, 6 pages
- Main Board, 13 pages

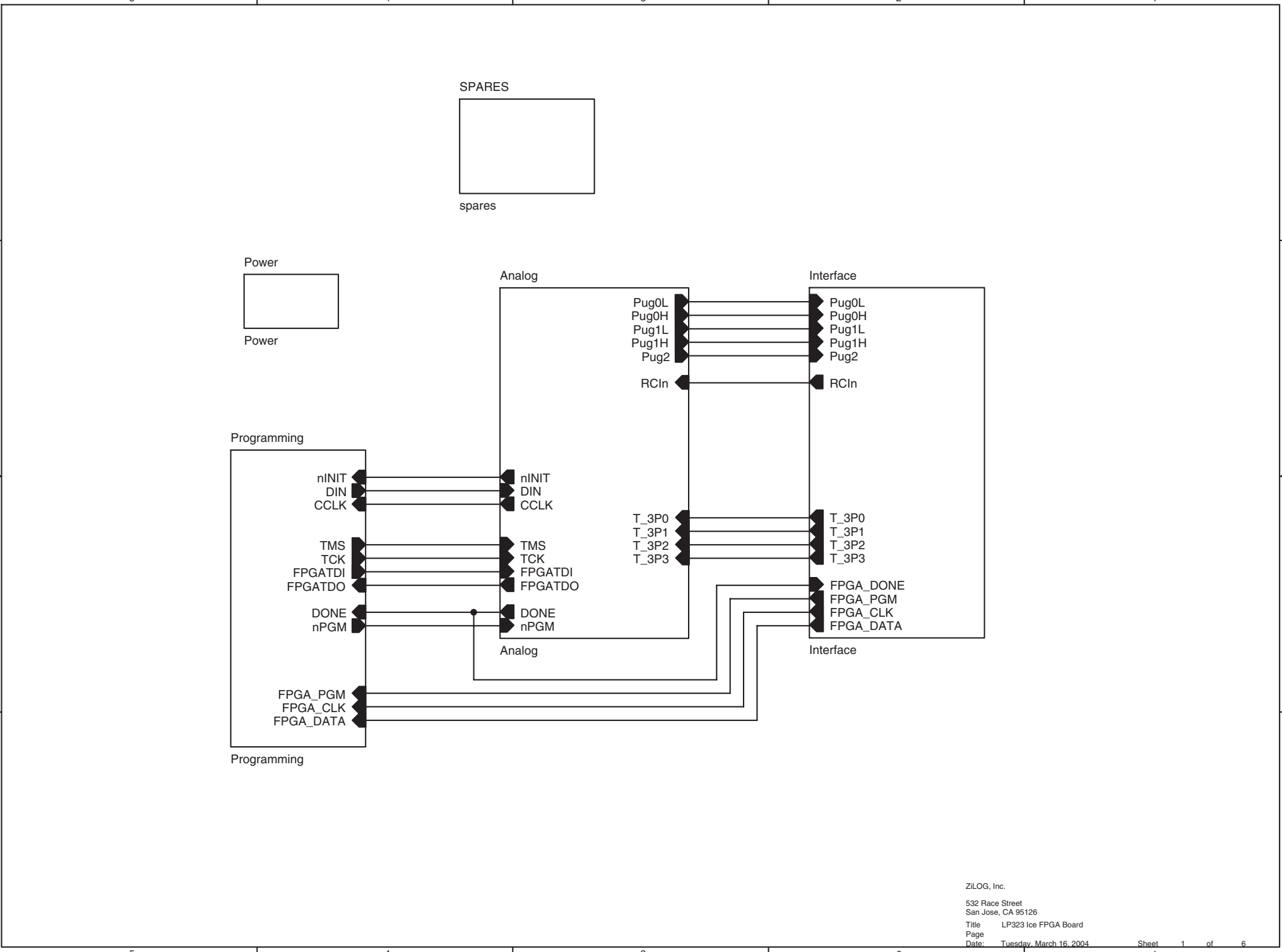


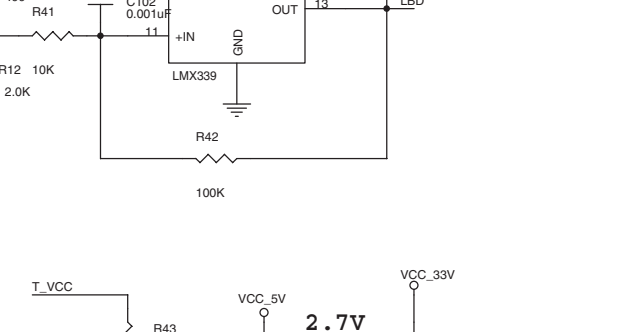
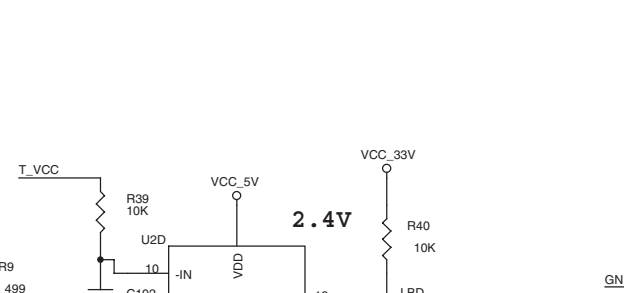
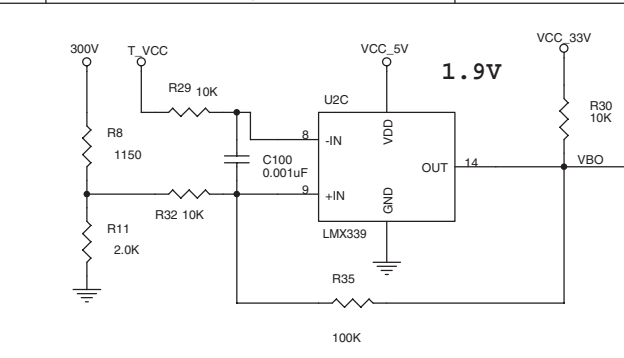
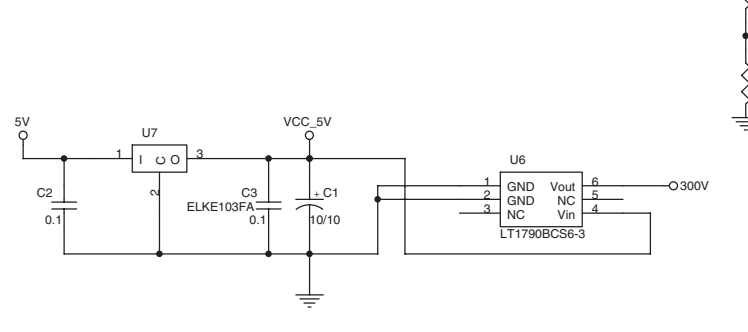
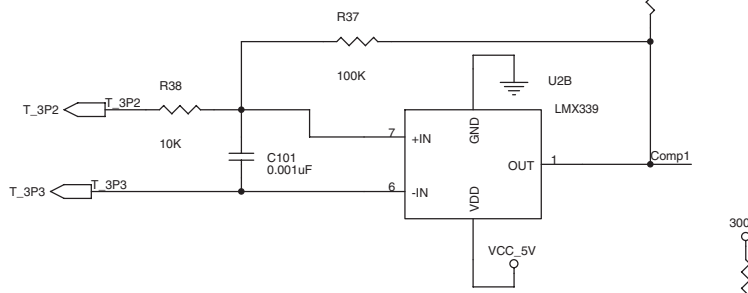
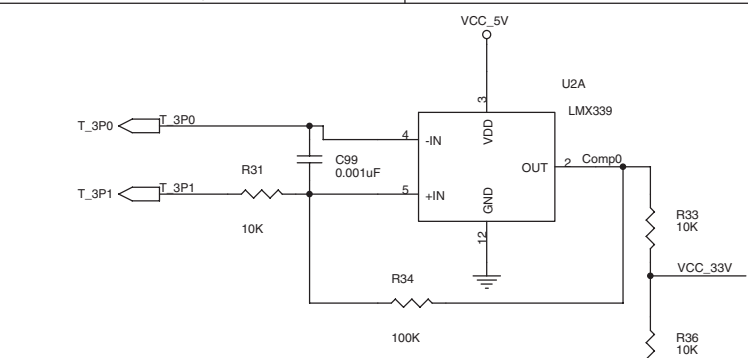
ZLP32300







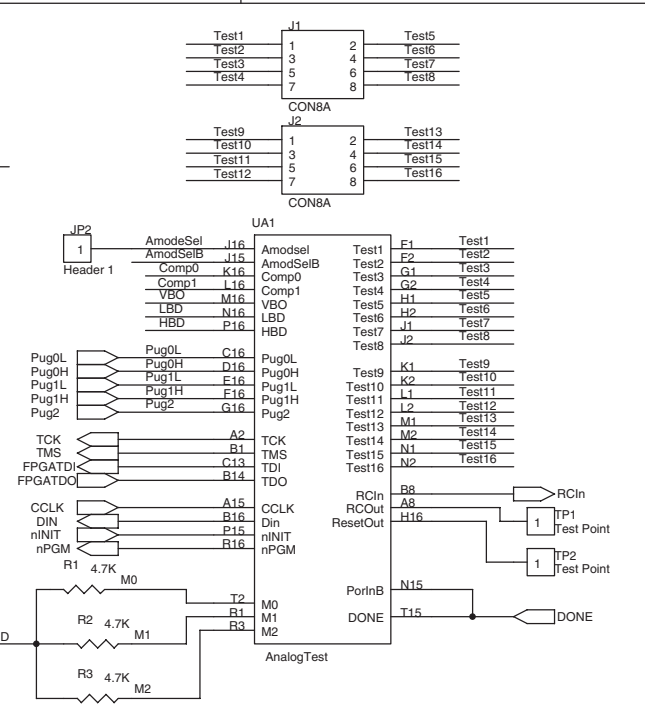


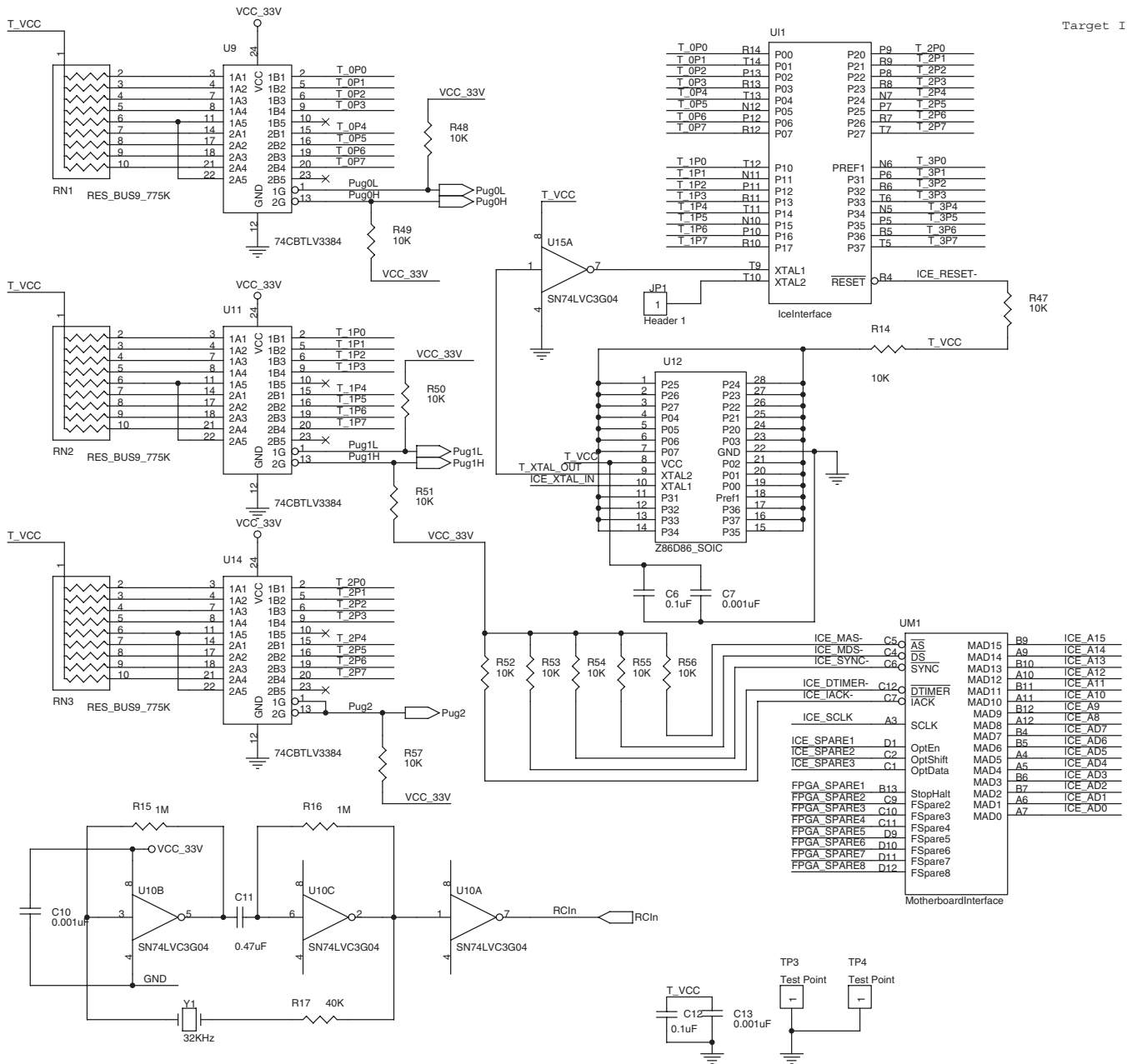


1.9V

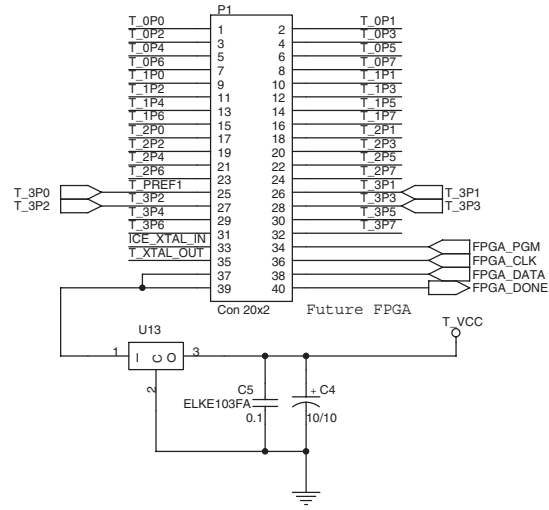
2.4V

2.7V

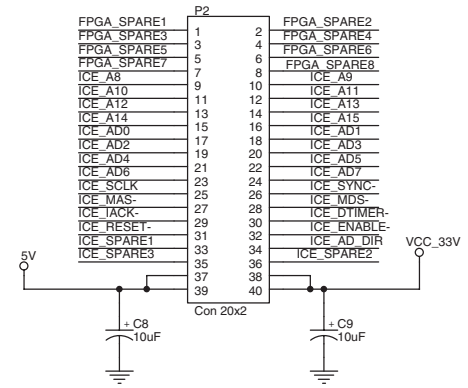




These connectors must match the connectors on the motherboard



These connectors have four central grounds that must be connected manually. They are not in the netlist.

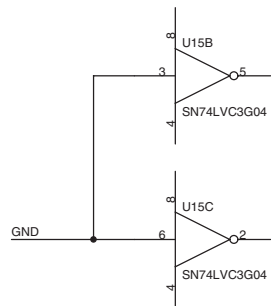
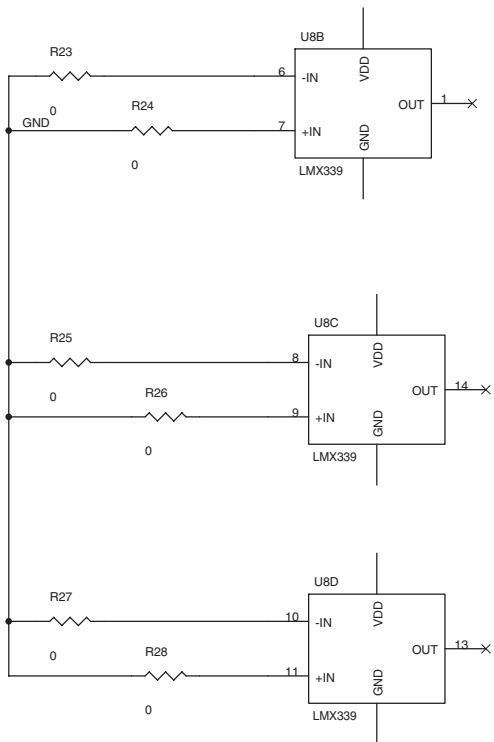


ZiLOG, Inc.  
 532 Race Street  
 San Jose, CA 95126  
 Title  
 Page  
 Date: Monday, March 15, 2004









ZiLOG, Inc.

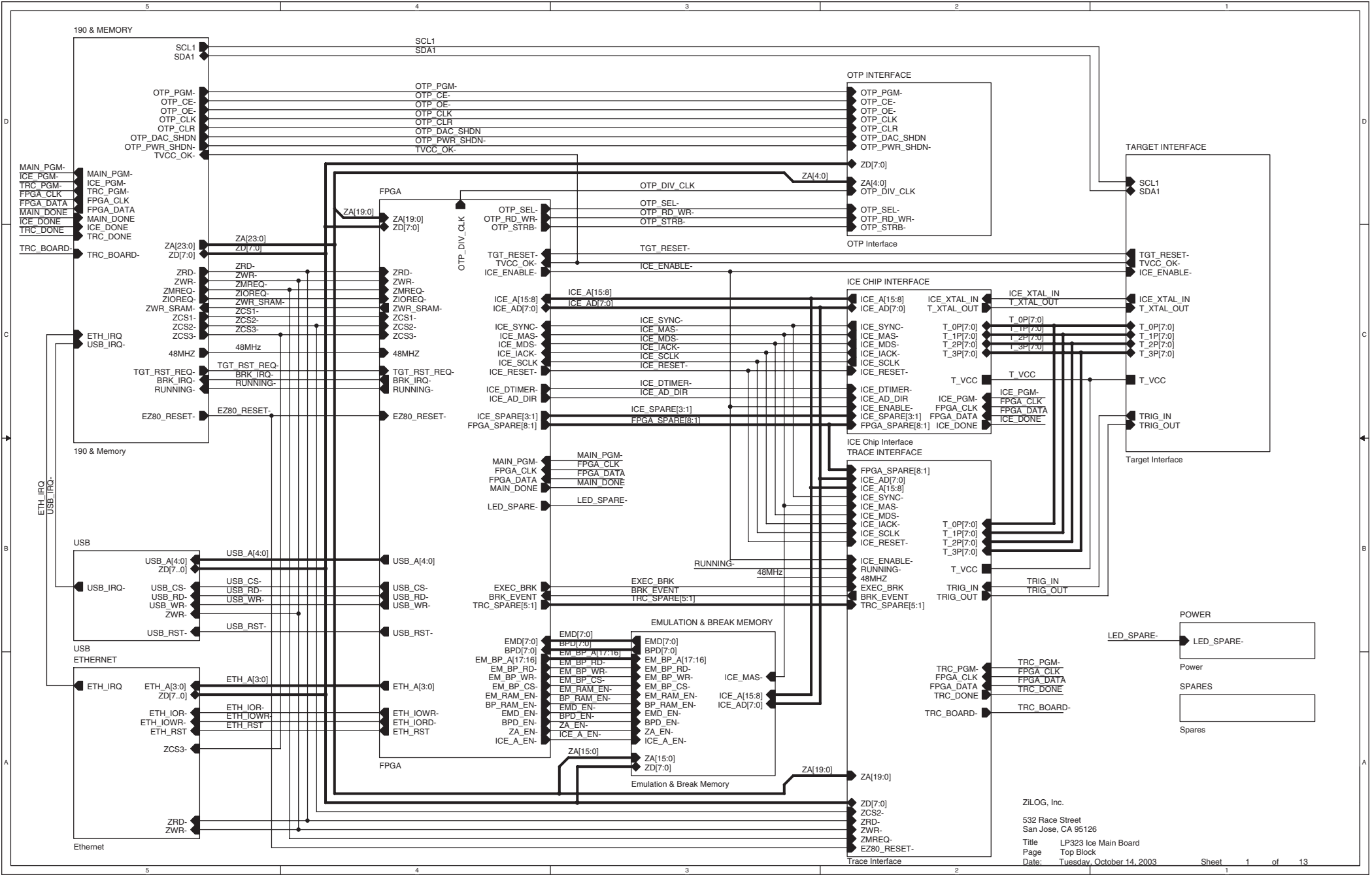
532 Race Street  
San Jose, CA 95126

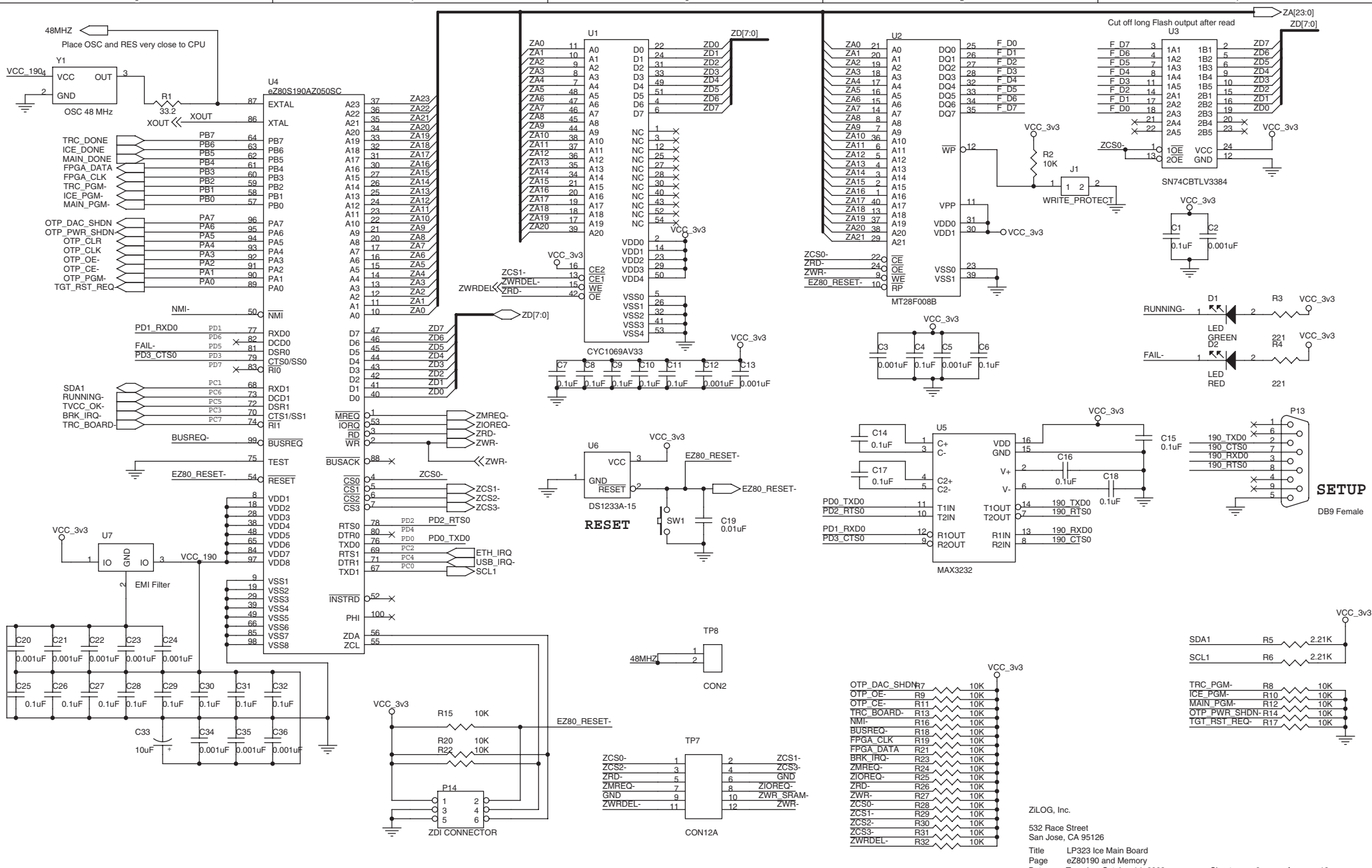
Title Z86D74 CPU Module

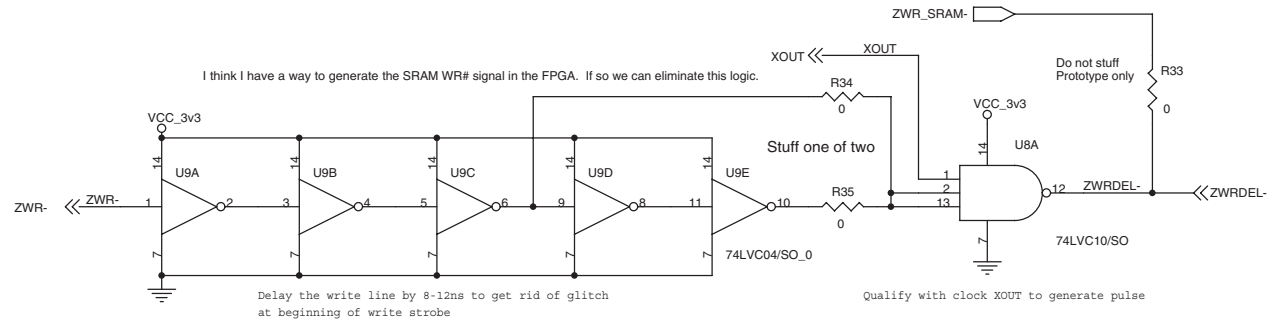
Page

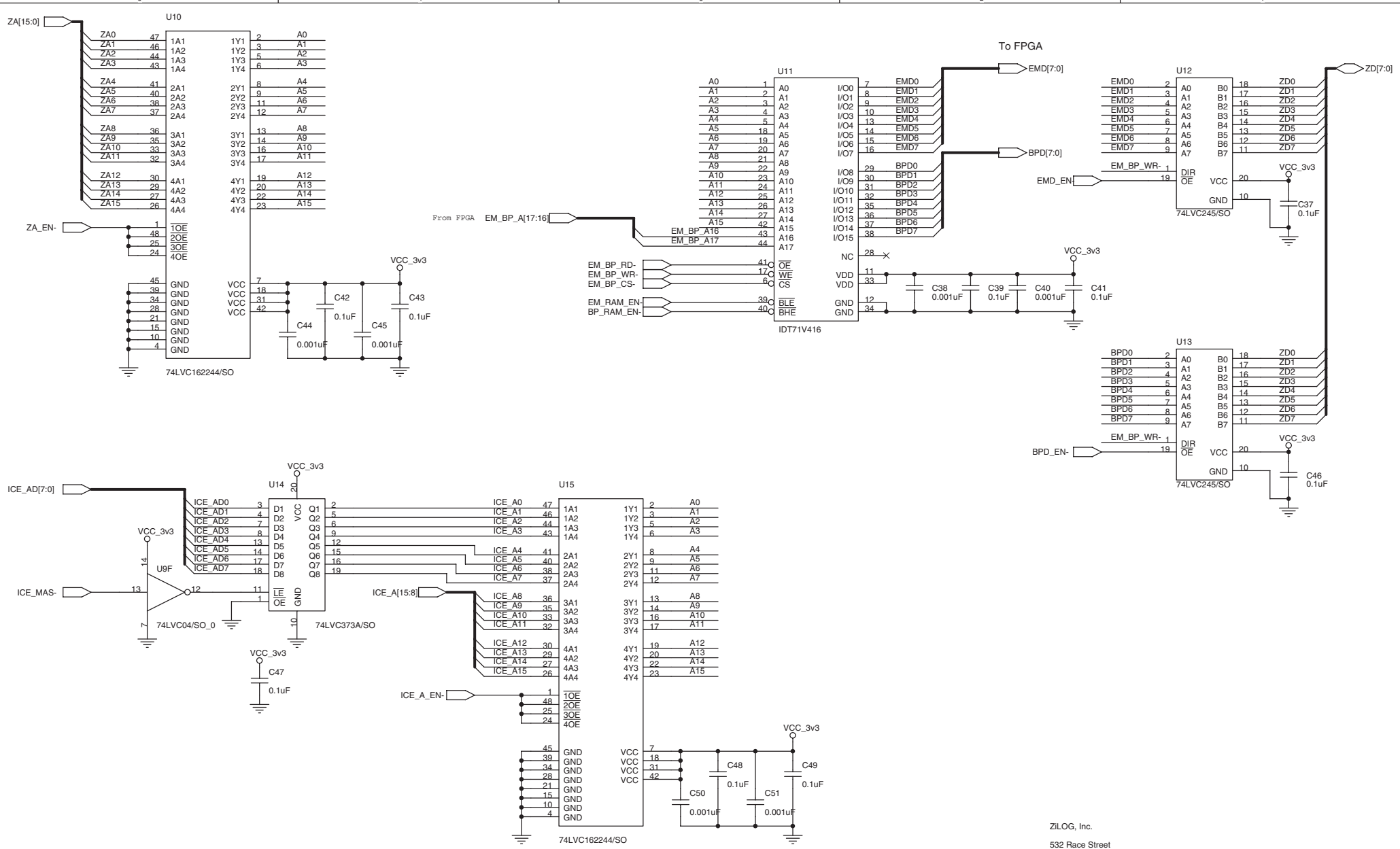
Date: Friday, March 12, 2004

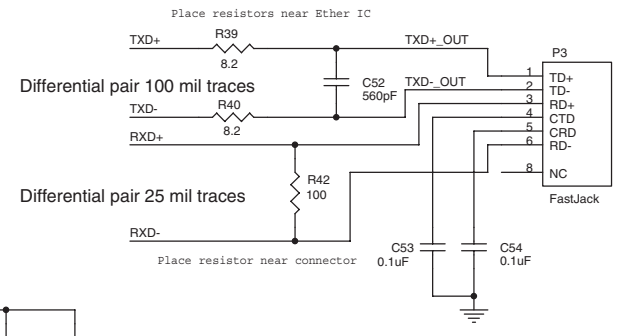
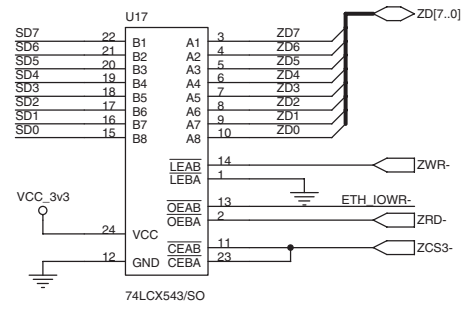
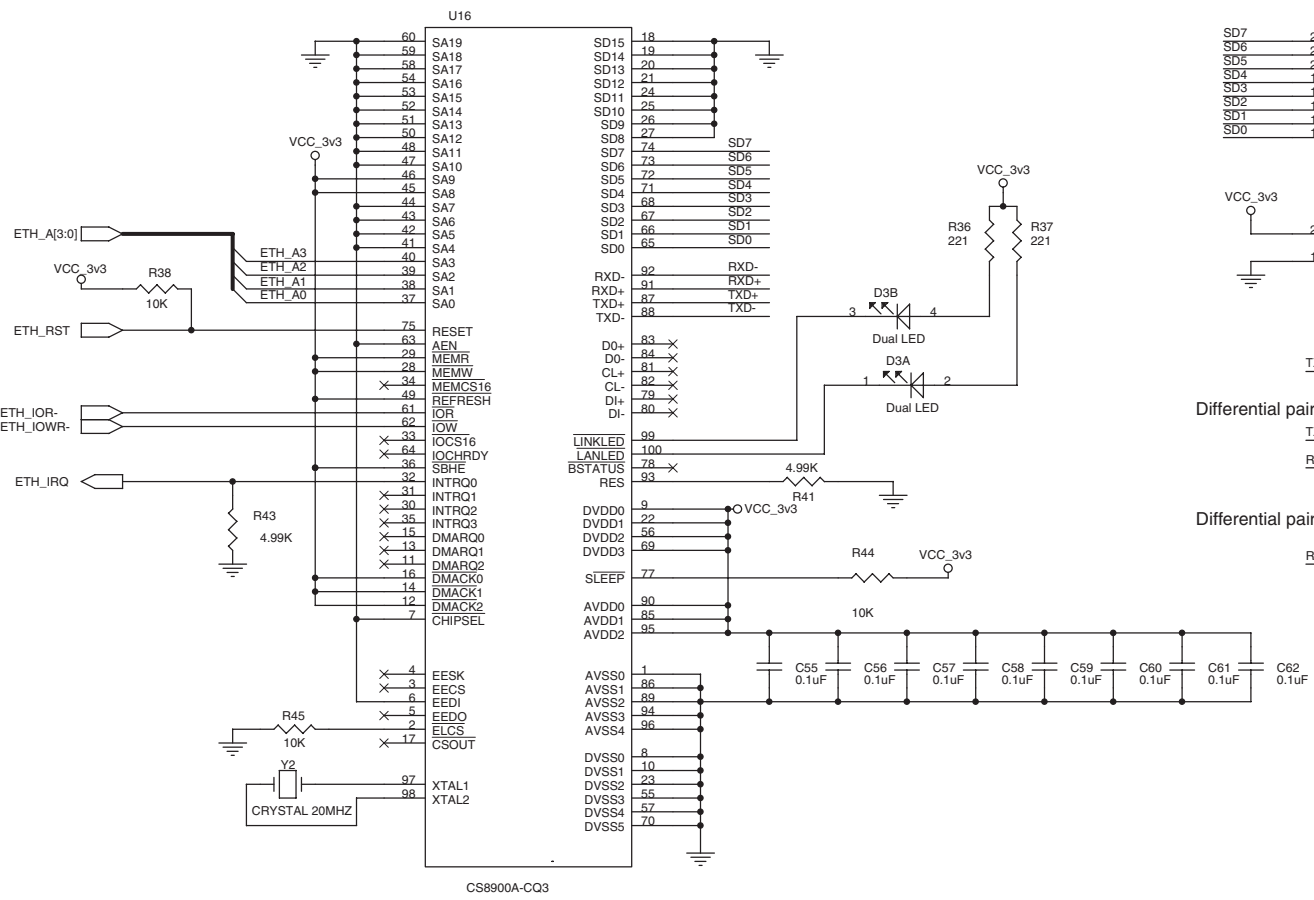
Sheet 6 of 6



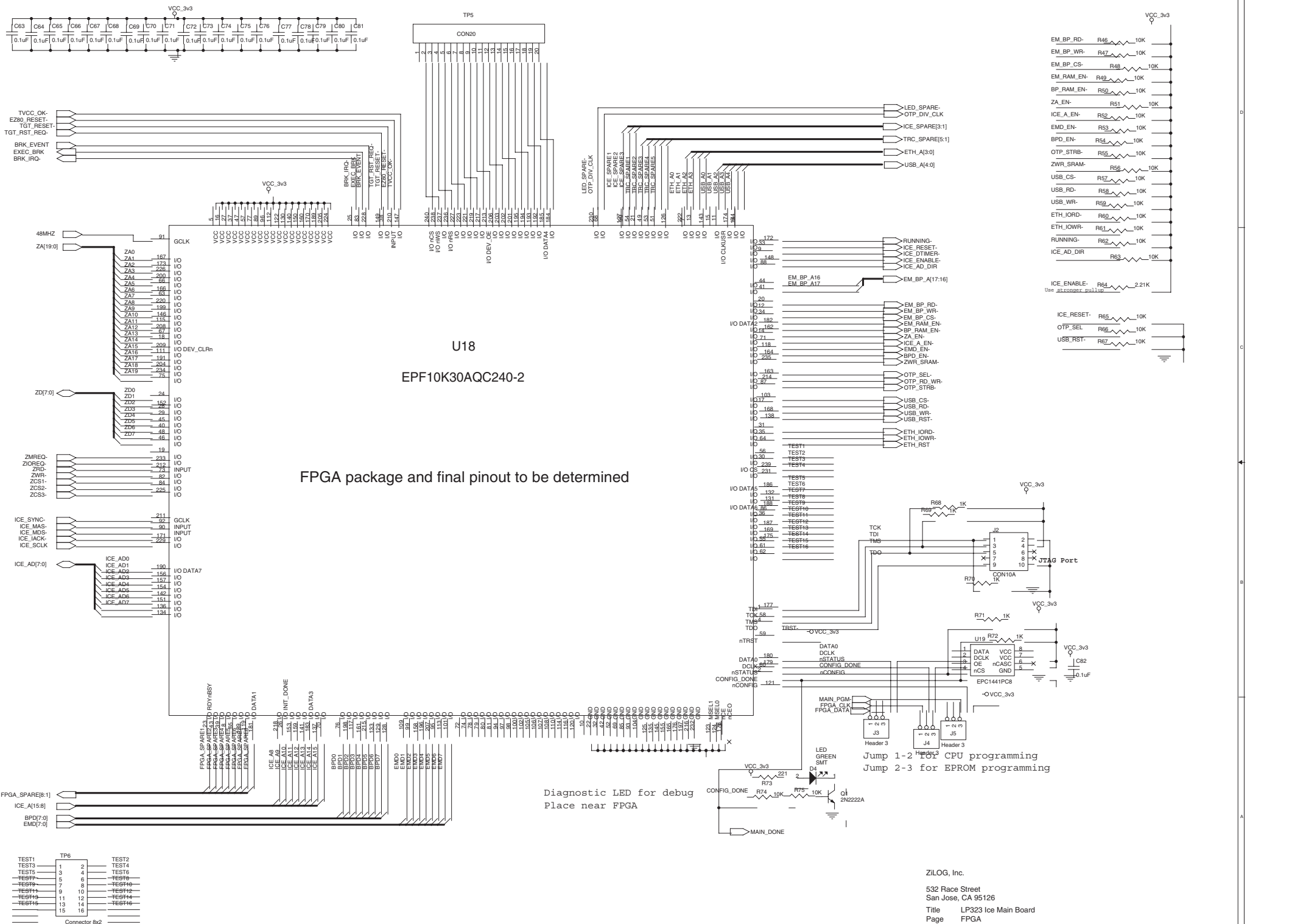












FPGA package and final pinout to be determined

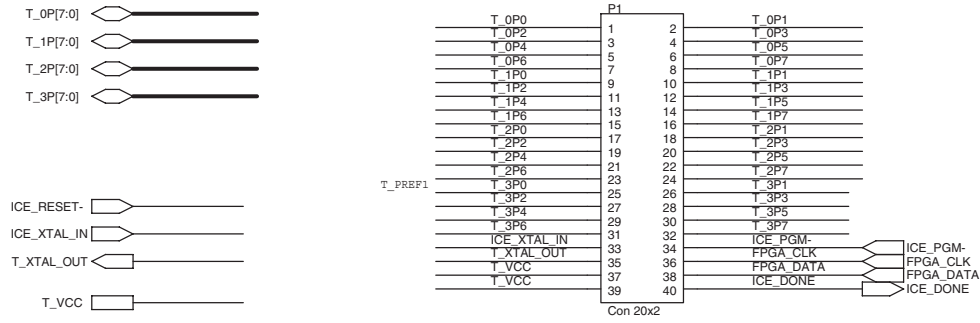
Diagnostic LED for debug  
Place near FPGA

Jump 1-2 for CPU programming  
Jump 2-3 for EPROM programming

TEST1	TP6	TEST2
TEST3	1 2	TEST4
TEST5	3 4	TEST6
TEST7	5 6	TEST8
TEST9	7 8	TEST10
TEST11	9 10	TEST12
TEST13	11 12	TEST14
TEST15	13 14	TEST16
	15 16	

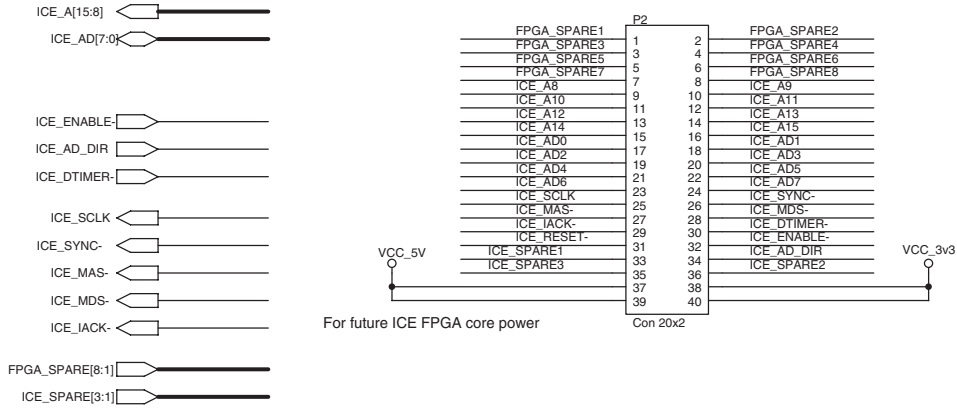
Connector 8x2

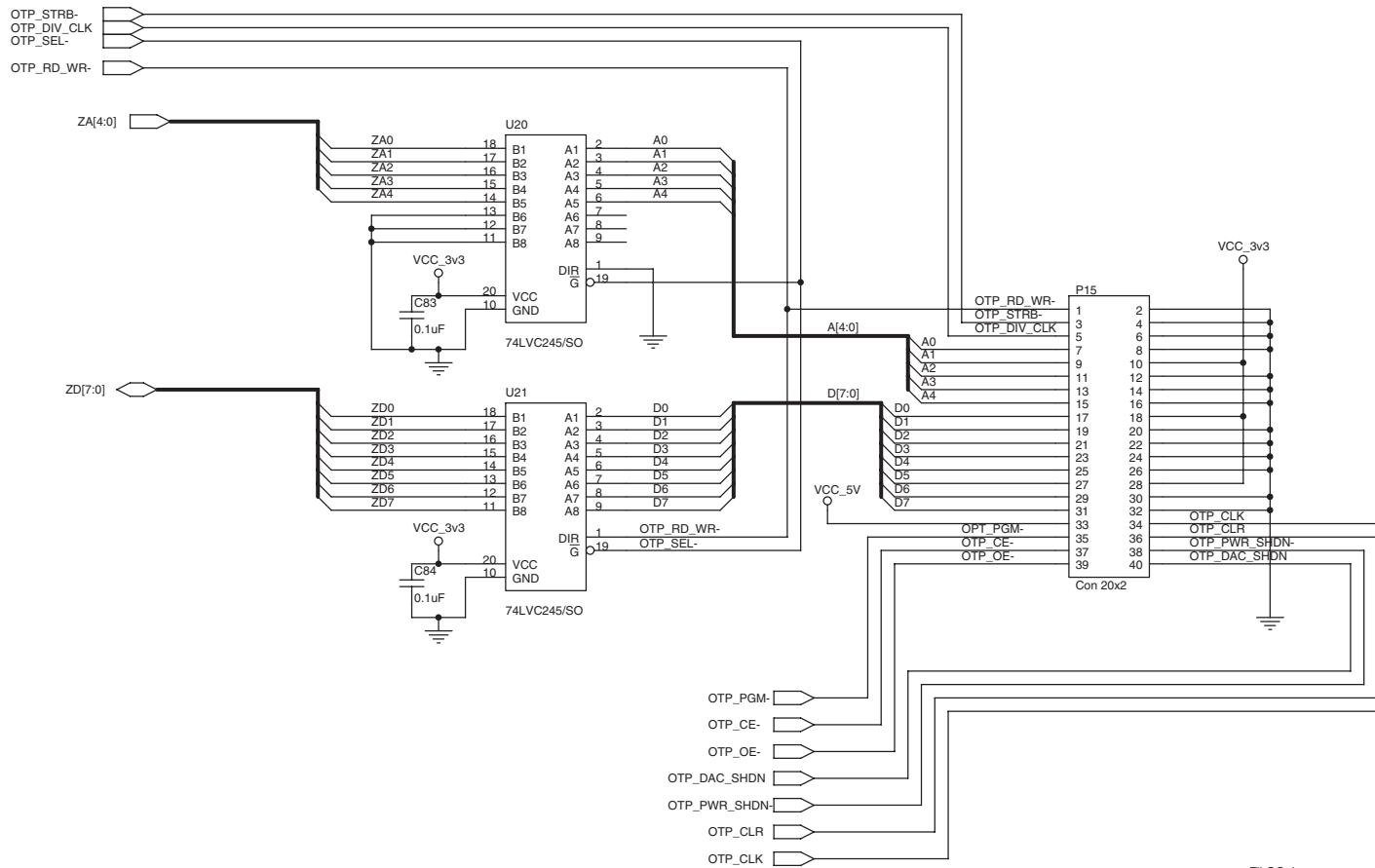
These connectors have four central grounds



Place connectors on top side of board  
 These connectors are mirror images of trace interface connectors on bottom of board  
 Place with pads aligned on both sides of board and use single via to connect pads with common signals

These connectors have four central grounds





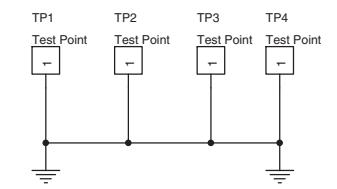
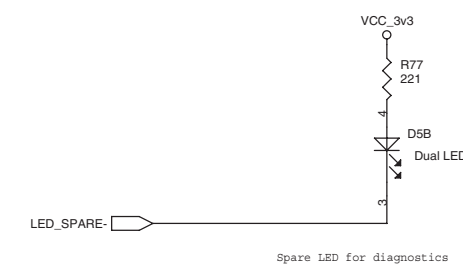
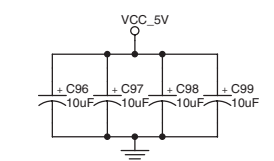
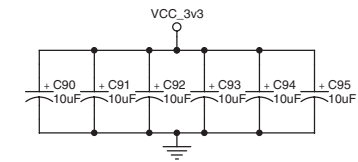
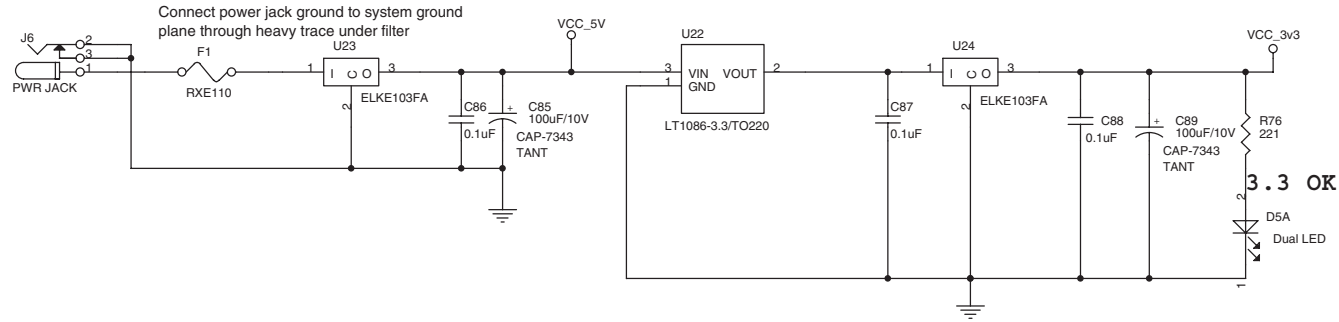
ZILOG, Inc.

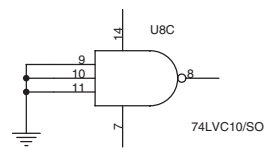
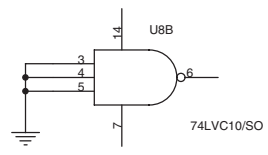
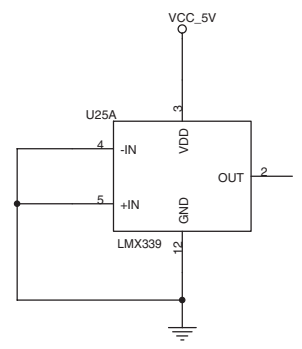
532 Race Street  
San Jose, CA 95126

Title LP323 Ice Main Board

Page OTP Interface

Date: Wednesday, September 17, 2003 Sheet 8 of 13



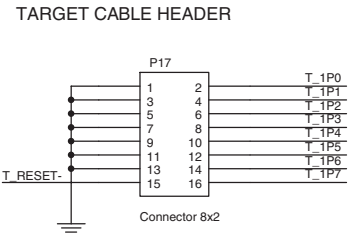
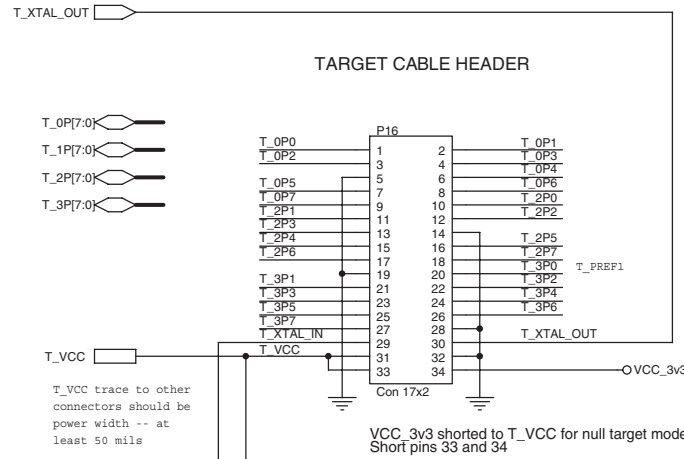


ZILOG, Inc.

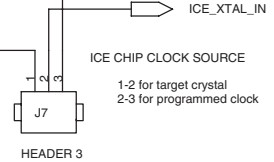
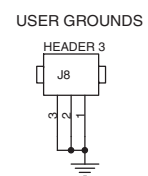
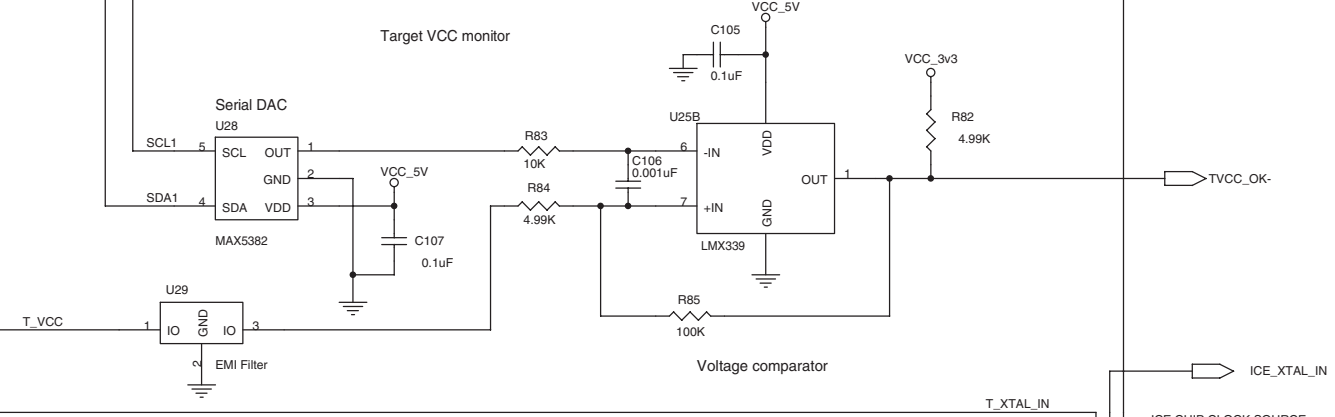
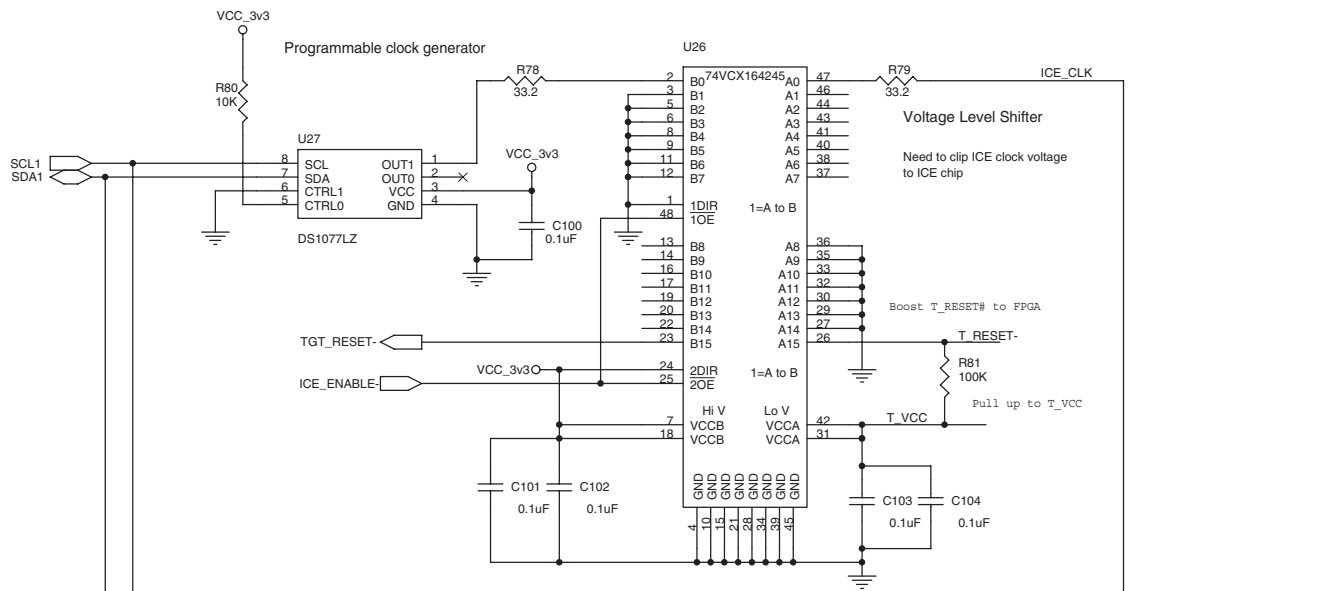
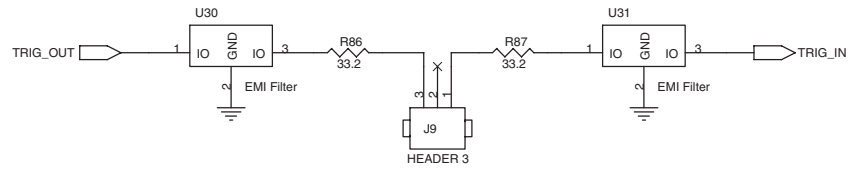
532 Race Street  
San Jose, CA 95126

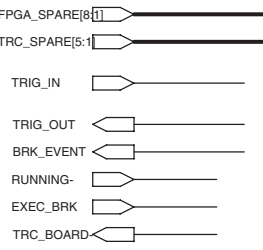
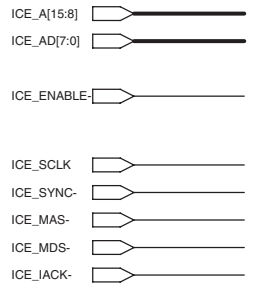
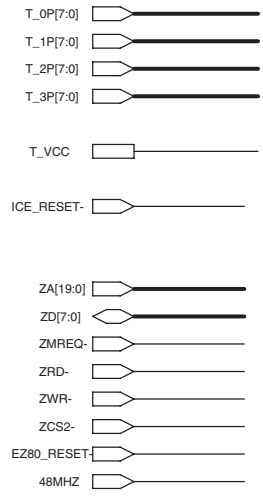
Title LP323 Ice Main Board  
Page Spare Components

Date: Tuesday, October 14, 2003 Sheet 10 of 13

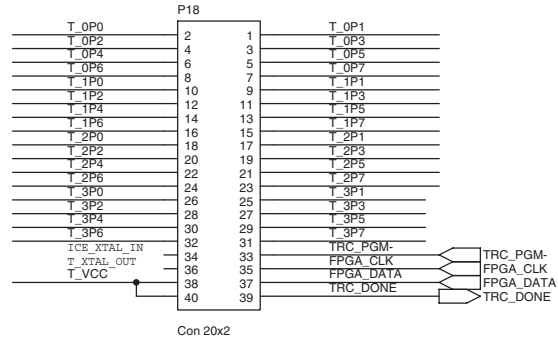


This part of T\_VCC trace can be standard trace width

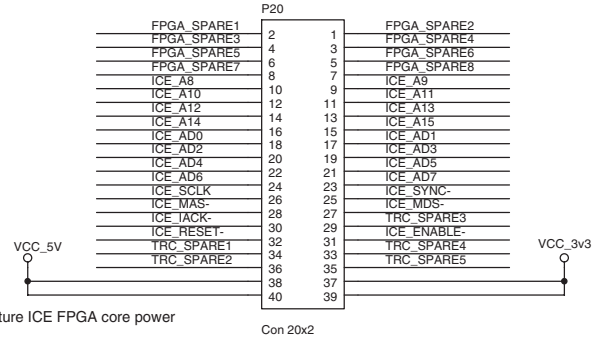




These connectors have four central grounds not shown in netlist  
They must be connected manually



Place these connectors on bottom side of board  
This connector is mirror image of ICE interface connectors on top of board  
Place with pads aligned on both sides of board and use single via to connect common signals



For future ICE FPGA core power

These connectors have four central grounds

This connector does not have a corresponding connector on the opposite side of the board. It can be placed for optimum routing.

