Intel® IQ80303 Evaluation Platform

Board Manual

June 2000

Order Number: 273401-001

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel® products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel® products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The IQ80303 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Copyright © Intel Corporation, 2000

*Other brands and names are the property of their respective owners.

Contents

Figures

Tables

Introduction 1

This user's guide describes the Intel® IQ80303 Evaluation Platfrom for the Intel® 80303 I/O processor. The 80303 I/O processor combines an Intel® 80960JT processor core with two PCI bus interfaces, as well as a memory controller, DMA channels, an interrupt controller interface, I^2C Serial Bus and general purpose I/O. The 80303 I/O processor features 64-bit, 66 MHz, +3.3 V or +5 V primary and secondary PCI busses. The IQ80303 is a full-length PCI adapter board but is 8.9" in height to accommodate two standard PCI connectors on the secondary PCI bus. The board can be installed in any PCI host system that complies with the *PCI Local Bus Specification* Revision 2.1 or *PCI Local Bus Specification* Revision 2.2. PCI devices can be connected to the secondary bus to build powerful intelligent I/O subsystems.

Figure 1-1. Intel® IQ80303Functional Block Diagram

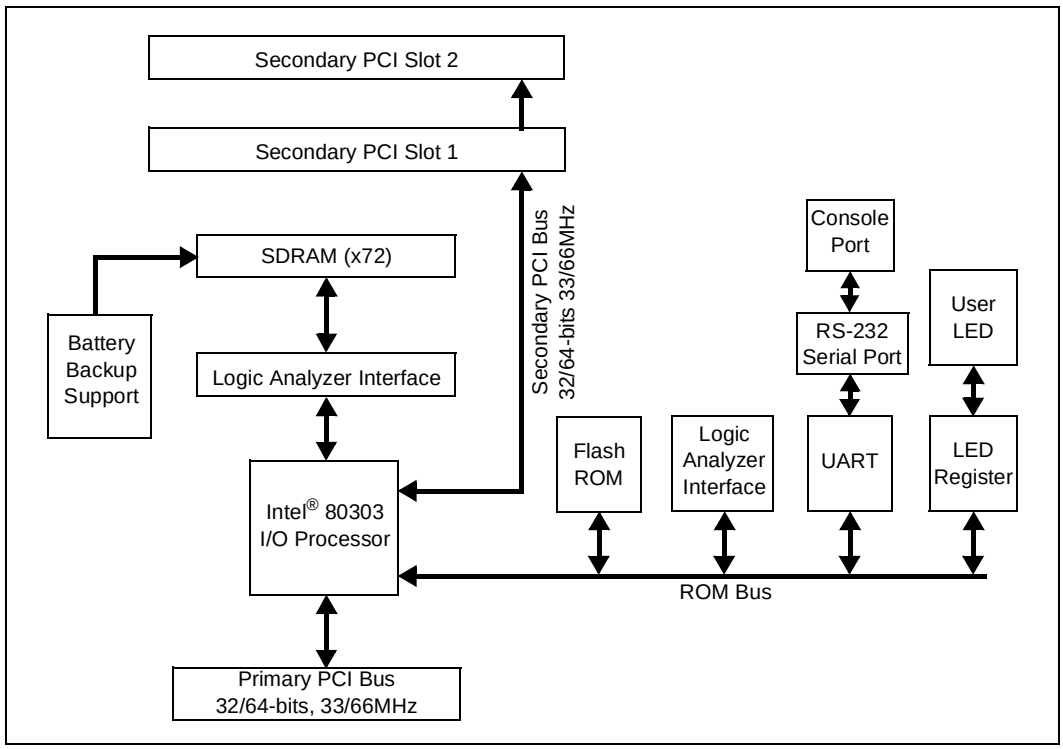
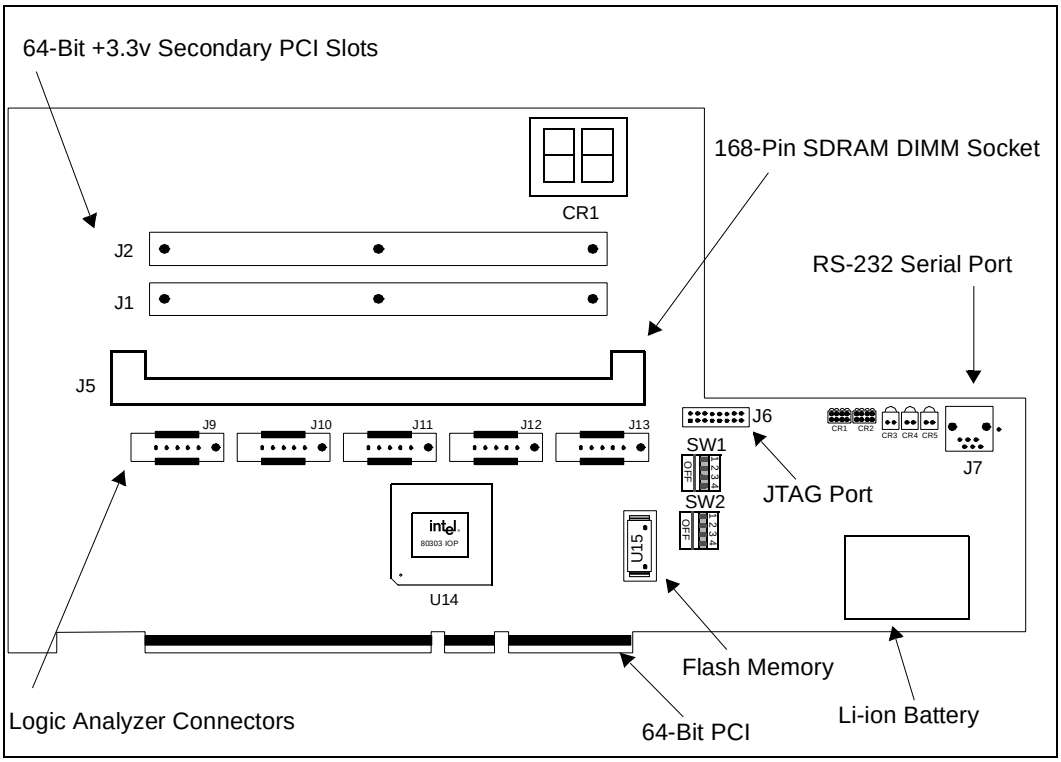


Figure 1-2. Intel® IQ80303 Evaluation Platfrom Physical Diagram



1.1 Intel® 80303 I/O Processor Advantages and Features

The 80303 I/O processor serves as the main component of a high performance, PCI-based intelligent I/O subsystem. The IQ80303 platform allows the developer to connect PCI devices to the 80303 I/O processor using the two secondary PCI expansion connectors. The features of the IQ80303 platform are enumerated below and shown in Figure 1-1 and Figure 1-2.

- 80303 I/O processor
- Modified PCI long-card form factor
- 64-bit or 32-bit, 66 MHz or 33 MHz, +3.3 V or +5 V primary PCI bus interface
- 64-bit or 32-bit, 66 MHz or 33 MHz, +3.3 V secondary PCI bus connected to the primary PCI interface with a PCI-to-PCI bridge
- DMA channels on both PCI buses
- I^2C Serial Bus
- 168-pin, 3.3 V DIMM socket supporting 64 to 512 Mbytes of Synchronous DRAM organized x72 to support Error Correction Code (ECC) and clocked at 100 MHz
- Serial console port based on 16C550 UART
- Two user-programmable 7-segment displays
- Four indicator LEDs: processor has passed self-test, 3.3 V is supplied to the platform, and 66 MHz is enable for the primary and the secondary PCI slots
- Flash ROM, 2 Mbytes
- Logic analyzer connectors for SDRAM bus, ROM bus and secondary PCI arbitration signals
- Battery backup for SDRAM
- JTAG header
- General purpose I/O header

1.2 Software Development Tools

A number of software development tools are available for the 80303 processor. This manual provides information on two software development toolsets: Wind River System IxWorks* and Intel® CTOOLS. If you are using other software development tools, read through the information in this section and in Chapter 2 to gain a general understanding of how to use your tools with this board.

1.3 IxWorks Software Development Toolset

IxWorks is a complete toolset featuring an integrated development environment including a compiler, assembler, linker, and debugger. It also features a real-time operating system.

1.3.1 IxWorks Real-time Operating System

The IQ80303 platform is equipped with Wind River Systems, Inc. IxWorks. IxWorks provides for the elements of the I_2O* standard: an event-driven driver framework, host message protocols, and executive modules for configuration and control. IxWorks also allows for the writing of basic device drivers and provides NOS-to-driver independence. Tornado* for I_2O provides a visual environment for building, testing and debugging of I_2O drivers.

1.3.2 Tornado Build Tools

Tornado for I_2O includes a collection of supporting tool that provide a complete development tool chain. These include the compiler, assembler, linker and binary utilities. Also provided is an I_2O module builder, which creates I_2O-loadable modules.

1.3.3 Tornado Test and Debug Tools

Tornado for I_2O test and debug tools include the dynamic loader, the CrossWind* debugger, the WINDSH* interactive shell, and a system browser.

The dynamic loader allows for interactive loading, testing, and replacement of individual object modules that comprise a driver.

CrossWind is an extended version of GBD960. Using it you can debug I_2O drivers by setting breakpoints on desired I_2O components. A variety of windows display source code, registers, locals, stack frame, memory and so on.

WindSh allows you to communicate to the IQ80303 platform via an RS-232 serial port. The IQ80303 platform supports port speed from 300 to 115,200 bps. The shell can be used to:

- control and monitor I_2O drivers
- format, send and receive driver messages
- examine hardware registers
- run automated I_2O test suites

The shell also provides essential debugging capabilities; including breakpoints, single stepping, stack checking, and disassembly.

1.4 Intel® CTOOLS Software Development Toolset

The 80303 I/O processor software development toolset, Intel® CTOOLS, features advanced C/C++ language compilers. CTOOLS development toolset is available for Windows* 95/NT-based systems and a variety of UNIX workstations hosts. These products provide execution profiling and instruction scheduling optimizations and include an assembler, a linker, and utilities designed for embedded processor software development.

1.4.1 Intel® CTOOLS and Intel® MON960 Debug Monitor

In place of IxWorks, the IQ80303 platform can be equipped with Intel® MON960, an on-board software monitor that allows you to execute and debug programs written for the 80303 I/O processor in a non-I_2O environment. The monitor provides program download, breakpoint, single step, memory display, and other useful functions for running and debugging a program.

The IQ80303 platform works with the source-level debuggers provided with CTOOLS, including Intel® GDB960 (command line version) and Intel® GDB960V (BUI version).

1.4.1.1 Intel® MON960 Host Communications

MON960 allows you to communicate and download programs developed for the IQ80303 platform across a host system's serial port or PCI interface. The IQ80303 platform supports two methods of communication: terminal emulation and Host Debugger Interface (HDI).

1.4.1.2 Terminal Emulation Method

Terminal emulation software on your host system can communicate to MON960 on the IQ80303 platform via na RS-232 serial port. The IQ80303 platform supports port speed from 300 to 115,200 bps. Serial downloads to MON960 require that the terminal emulation software support the XMODEM protocol.

Configure the serial port on the host system for 300-115,200 baud, eight bits, one stop bit, no parity with XON/XOFF flow control.

1.4.1.3 Host Debugger Interface (HDI) Method

You may use a source-level debugger, such as GDB960 and GDB960V to establish serial or PCI communications with IQ80303 platform. The MON960 Host Debugger Interface (HDI) provides a defined messaging layer between MON960 and the debugger. For more information on this interface, see *MON960 Debug Monitor User's Manual* (484290).

HDI connection requests cannot be detected by MON960 if the user has already initiated a connection using a terminal emulator. In this case, the IQ80303 platform must be reset before the debugger can connect to MON960.

1.5 SPI610 JTAG Emulation System

The SPI610 JTAG Emulation System from Spectrum Digital, Inc. is included in the IQ80303 development kit. It furnishes the default host development environment-to- evaluation board communication link based on the 80303 I/O processor JTAG interface. Refer to the SPI610 Reference Manual for JTAG emulation system installation and operation for both the Tornado and CTOOLS environment. Optionally, evaluation board serial port communications can be used for this communication link (see Section 1.3.3, “Tornado Test and Debug Tools” on page 1-4).

1.6 About This Manual

A brief description of the contents of this manual follows:

- Chapter 1, “Introduction”

  Introduces the IQ80303 Evaluation Board features. This chapter also describes Intel’s CTOOLS* and WindRiver Systems IxWorks* software development tools, and defines notational conventions and related documentation.

- Chapter 2, “Intel® IQ80303 Evaluation Platform Getting Started”

  Provides step-by-step instructions for installing the IQ80303 platform in a host system and downloading and executing an application program. This chapter also describes Intel’s software development tools, the MON960 Debug Monitor, Ixworks, software installation, and hardware configuration.

- Chapter 3, “Hardware Reference”

  Describes the locations of connectors, switches and LEDs on the IQ80303 platform. Header pinouts and register descriptions are also provided in this chapter.

- Chapter 4, “Intel® 80303 I/O Processor Overview”

  Presents an overview of the capabilities of the Intel® 80303 I/O processor and includes the CPU memory map.

- Chapter 5, “Intel® MON960 Support for Intel® 80303 I/O Processor”

  Describes a number of features added to MON960 to support application development on the 80303 I/O processor.

- Appendix A, “Bill of Materials”

  Shows complete parts list for IQ80303 Evaluation Platform.

- Appendix B, “Schematics”

  Complete set of schematics for the IQ80303 Evaluation Platform.

1.7 Notational Conventions

The following notation conventions are consistent with other 80303 I/O processor documentation and general industry standards.

- # or overbar

  In code examples the pound symbol (#) is appended to a signal name to indicate that the signal is active. Normally inverted clock signals are indicated with an overbar above the signal name (e.g., RAS).

- **Bold**

  Indicates user entry and/or commands. PLD signal names are in bold lowercase letters (e.g., **h_off**, **h_on**).

- *Italics*

  Indicates a reference to related documents; also used to show emphasis.

- `Courier` font

  Indicates `code` examples and file directories and names.

- Asterisks (*)

  On non-Intel company and product names, a trailing asterisk indicates the item is a trademark or registered trademark. Such brands and names are the property of their respective owners.

- UPPERCASE

  In text, signal names are shown in uppercase. When several signals share a common name, each signal is represented by the signal name followed by a number; the group is represented by the signal name followed by a variable *(n)*. In code examples, signal names are shown in the case required by the software development tool in use.

- Designations for hexadecimal and binary numbers

  In text, instead of using subscripted “base” designators (e.g., FF16) or leading “0x” (e.g., 0xFF) hexadecimal numbers are represented by a string of hex digits followed by the letter *H*. A zero prefix is added to numbers that begin with *A* through *F*. (e.g., *FF* is shown as *0FFH*.) In examples of actual code, “0x” is used. Decimal and binary numbers are represented by their customary notations. (e.g., 255 is a decimal number and 1111 1111 is a binary number. In some cases, the letter *B* is added to binary numbers for clarity.)

1.8 Technical Support, Schematics, and PLD Equations

For technical assistance with the 80303 I/O processor, contact the Intel Technical Support Hotline. For information about technical support in other geographical areas, contact the Intel North America Technical Support Hotline.

Up-to-date product and technical information is available from:

- Intel World-Wide Web: www.intel.com
- IQ80303 Product Information: developer.intel.com

1.8.1 Intel Customer Support Contacts

Contact Intel Corporation for technical assistance for the IQ80303 evaluation platform.

Table 1-1. Customer Support Contacts (U.S.A. and Canada)

Country	Literature	Technical Support
Australia National Sydney	Contact local distributor	008-257-307 61-2-975-3300 61-3-810-2141
Belgium, Netherlands, Luxembourg	010-4071-111	010-4071-111
Canada	800-468-8118	Contact local distributor
Finland	358-0-544-644	358-0-544-644
France	33-1-30-57-70-00	33-1-30-57-72-22
Germany	49-89-90992-257	Hardware: 49-89-903-8529 Software: 49-89-903-2025
Israel	972-3-498080	972-3-548-3232
Italy	39-02-89200950	39-02-89200950
Japan	Contact local distributor	0120-1-80387
Sweden	46-8-7340100	46-8-7340100
United States	800-548-4725	800-628-6249

1.8.2 Additional Information

To order manuals from Intel, contact your local sales representative or Intel Literature Sales (1-800-548-4725).

Table 1-2. Additional Information

Document Name	Company/Order #
Intel® Solutions960® catalog	Intel # 270791
Intel®80303 I/O Processor Developer's Manual	Intel # 273353
Intel®80303 I/O Processor Datasheet	Intel # 273358
MON960 Debug Monitor User's Guide	Intel #484290
PCI Local Bus Specification, Revision 2.2	PCI Special Interest Group 1-800-433-5177
Writing Writing I_2O® Device Drivers in IxWorks™	Wind River Systems, Inc. #DOC-1173-8D-02
IxWORKS™ Reference Manual	Wind River Systems, Inc. #DOC-1173-8D-03
IxWORKS™ Programmer's Guide	Wind River Systems, Inc. #DOC-11045-ZD-01
Tornado™ User's Guide	Wind River Systems, Inc. #DOC-11169-8D-01
Tornado™ for I_2O®	Wind River Systems, Inc. #DOC-12381-8D-00
Tornado™ for I_2O® Compact Disk, Revision 1.0	#TDK-12380-ZC-00

Intel® IQ80303 Evaluation Platform Getting Started 2

This chapter contains instructions for installing the IQ80303 Evaluation Platform in a host system and, how to download and execute an application program using the Wind River System IxWorks* or Intel® CTOOLS software development toolsets.

2.1 Pre-Installation Considerations

This section provides a general overview of the components required to develop and execute a program on the IQ80303 platform. IQ80303 evaluation boards support two software development toolsets, Wind River System IxWorks and Intel CTOOLS.

IxWorks is a complete toolset featuring an integrated development environment including a compiler, assembler, linker, and debugger. It also features a real-time operating system. If you are using the IxWorks operating system with the Tornado* development environment, refer to the Wind River Systems, Inc. documentation referenced in Section 1.8.2.

CTOOLS is a complete C/C++ language software-development toolset for developing embedded applications to run on the 80303 processor. It contains:

- C/C++ compiler
- Intel® GCC960 and Intel® IC960 compiler driver programs
- assembler
- runtime libraries
- collection of software-development tools and utilities
- printed and on-line documentation

The *MON960 Debug Monitor User's Guide* fully describes the components of the Intel® MON960, including MON960 commands, the Host Debugger Interface Library (HDIL), and the MONDB.EXE utility. If you are using MON960 and the CTOOLS toolset, refer to section Section 2.2.1, "Installing Software Development Tools" on page 2-2.

See Section 1 for more information on the IxWorks and CTOOLS features.

The IQ80303 evaluation boards are supplied with IxWorks intelligent real-time operating system pre-loaded into the on-board Flash. You also have the option of installing the MON960 debug monitor, which is required if you are using the CTOOLS debugging tools, Intel® GDB960, Intel® GDB960GDB960V, or MONDB. Section 3.3.1 describes the Flash ROM programming utility, which allows you to load MON960 onto the platform or re-load IxWorks.

2.2 Software Installation

2.2.1 Installing Software Development Tools

If you haven't done so already, install your development software as described in its manuals. All references in this manual to CTOOLS or CrossWind assume that the default directories were selected during installation. If this is not the case, substitute the appropriate path for the default path wherever file locations are referenced in this manual.

2.3 Hardware Installation

Follow these instructions to get your new IQ80303 platform running. Be sure all items on the checklist were provided with your IQ80303.

Warning: Static charges can severely damage the IQ80303 platforms. Be sure you are properly grounded before removing the IQ80303 platform from the anti-static bag.

2.3.1 Battery Backup

Battery backup is provided to save any information in SDRAM during a power failure. The IQ80303 platform contains a Li-ion battery, a charging circuit and a regulator circuit. The battery installed in the IQ80303 platform is rated at 1040 mA/Hr.

SDRAM technology provides enabling data preservation through the self-refresh command. When the processor receives an active Primary PCI reset, the self-refresh command issues, driving SCKE signals low. Upon seeing this condition, an IQ80303 platform logic circuit holds SCKE low before the processor loses power. Batteries maintain power to SDRAM and logic, to ensure self-refresh mode. When the circuit detects PRST# returning to inactive state, the circuit releases the hold on SCKE.

The battery circuit can be disabled by removing the battery. If the battery remains in the platform when it is depowered and/or removed from the chassis, the battery maintains SDRAM for about TBA hours. Once power is reapplied, the battery is fully charged in approximately TBA hours.

2.3.2 Installing Intel® IQ80303 Evaluation Platforms in Host System

For first time installation of the IQ80303 platform, visually inspect the board for any damage made during shipment. If there are visible defects, return the board for replacement. Follow the host system manufacturer instructions for installing a PCI adapter. The IQ80303 platform is a full-length PCI adapter and requires a PCI slot free from obstructions. The IQ80303 platform is taller than specified in *PCI Local Bus Specification*, Revision 2.1. The extended height of the board requires the cover of the PC to be kept off. Refer to Chapter 3 for physical board dimensions.

2.3.3 Verify Intel® IQ80303 Evaluation Platform is Functional

These instructions assume that you have already installed the IQ80303 platform in the host system as described in Section 2.3.2.

1. To connect the serial port for communicating with and downloading to the IQ80303 platform, connect the RS-232 cable (provided with the IQ80303) from a free serial port on the host system to the phone jack-style connector on the IQ80303 platform.
2. Upon power-up, the red FAIL LED turns off, indicating the processor has passed self-test.
3. With IxWorks installed in flash ROM, the two user 7-segments display the pattern TBD H. In the IxWorks development environment, raw serial input/output is not used. Instead, Wind DeBug (WDB) protocol is run over the serial port, communicationing with Tornado development tools. If the terminal emulation package is running at 115,200 baud, the letters “WDB_READY” display prior to launching in the WDB serial protocol.
4. With MON960 installed in flash ROM, press **<ENTER>** on a terminal connected to the IQ80303 platform, to bring up the MON960 prompt. MON960 automatically adjusts its baud rate to match the terminal at start-up. At baud rates other than 9600, it may be necessary to press **<ENTER>** several times.

2.4 Creating and Downloading Executable Files

To download code to the IQ80303 platform running IxWorks, consult Wind River documentation on the supplied Tornado for I_2O* CD-ROM. To download code to the IQ80303 platform, your compiler produces an ELF-format object file.

To download code to the IQ80303 platform running CTOOLS, consult the CTOOLS documentation for information regarding compiling, linking, and downloading applications. During a download, MON960 checks the link address stored in the ELF file, and stores the file at that location on the IQ80303 platform. If the executable file is linked to an invalid address on the IQ80303 platform, MON960 aborts the download.

2.4.1 Sample Download and Execution Using Intel® GDB960

This example shows you how to use GBD960 to download and execute a file named `myapp` via the serial port.

1. Invoke GDB960. From a Windows 95/NT command prompt, issue the command:
   gdb960 -r com2 myapp

This command establishes communication and downloads the file `myapp`.

2. To execute the program, enter the command from the GDB960 command prompt:
   (gdb960) run

More information on the GDB960 commands mentioned in this section can be found in the *GDB960 User's Manual*.

Hardware Reference 3

3.1 Power Requirements

The Intel® IQ80303 Evaluation Platform draws power from the PCI bus. The power requirements of the IQ80303 platform is shown in Table 3-1. The numbers do not include the power required by a PCI card(s) mounted on one or more of the IQ80303 platforms' two expansion slots.

Table 3-1. Intel® IQ80303 Evaluation Platform Power Requirements

Voltage	Typical Current	Maximum Current
+3.3 V	V	V
+5 V	A	A
+12 V	mA	mA
-12 V	mA	mA

NOTE: Does not include the power required by a PCI card(s) mounted on the IQ80303 platform.

3.2 SDRAM

The IQ80303 platform is equipped with a 168-pin DIMM socket formatted to accept +3.3 V synchronous DRAM with Error Correction Code (ECC). The socket accepts SDRAM from 32 Mbytes to 512 Mbytes. SDRAM modules must support ECC in the x72 configuration. SDRAM is accessible from either of the PCI buses, via the ATUs, and the local bus on the IQ80303 platform.

3.2.1 SDRAM Performance

The IQ80303 platform uses 72-bit SDRAM with ECC. SDRAM allows zero data-to-data wait state operation at 100 MHz. The 80303 I/O processor memory controller unit (MCU) supports SDRAM burst lengths of four. This enables seamless read/write bursting of long data streams, as long as the MCU does not cross the page boundary. Page boundaries are naturally aligned 2 Kbyte blocks. 72-bit SDRAM with ECC allows a maximum throughput of 800 Mbytes/sec.

64 Mbit, 128 Mbit and 256 Mbit SDRAM devices are supported. The MCU keeps four pages per bank open simultaneously for 64/128/256 Mbit devices. Simultaneously open pages allow for greater performance for sequential access, distributed across multiple internal bus transactions. Table 3-2 shows read and write examples of a single 8 byte access and for a multiple 40 byte access.

Table 3-2. SDRAM Performance

Cycle Type	Table Clocks	Performance Bandwidth
Read Page Hit (8 bytes)	7	114 Mbytes/sec
Read Page Miss (8 bytes)	12	67 Mbytes/sec
Read Page Hit (40 bytes)	11	360 Mbytes/sec
Read Page Miss (40 bytes)	16	248 Mbytes/sec
Write Page Hit (8 bytes)	4	198 Mbytes/sec
Write Page Miss (8 bytes)	8	100 Mbytes/sec
Write Page Hit (40 bytes)	8	495 Mbytes/sec
Write Page Miss (40 bytes)	12	330 Mbytes/sec

3.2.2 Upgrading SDRAM

The IQ80303 is equipped with 64 Mbytes of SDRAM with ECC inserted in the 168-pin DIMM socket. The memory may be expanded by inserting up to a 512 Mbyte module into the DIMM socket. The various memory combinations are shown in Table 3-3. Only 168-pin +3.3 V SDRAM modules with ECC rated at 8 ns should be used on the IQ80303 platform.

Table 3-3. SDRAM Configurations

SDRAM Technology	SDRAM Arrangement	# Banks	Row	Column	Total Memory SIze
64 Mbit	8M x 8	1	12	9	64 Mbytes
		2			128 Mbytes
	4M x 16	1	12	8	32 Mbytes
		2			64 Mbytes
128 Mbit	16M x 8	1	12	10	128 Mbytes
		2			256 Mbytes
	8M x 16	1	12	9	64 Mbytes
		2			128 Mbytes
256 Mbit	32M x 8	1	13	10	256 Mbytes
		2			512 Mbytes

3.3 Flash ROM

An E28F016S3 (2 Mbytes) Flash ROM is included on the IQ80303 platform. This Flash ROM contains IxWorks* and may be used to store user applications.

3.3.1 Flash ROM Programming

Two types of Flash ROM programming exist on the IQ80303 platform. The first is normal application development programming. This occurs using IxWorks to download new software and the 80960JT core to write the new code to the Flash ROM. During this time the boot sectors (containing IxWorks) are write protected.

The second type of Flash ROM programming is loading the boot sectors. You will not be required to load the boot sectors except:

- To load MON960
- To load a new release of IxWorks
- To change between the check build and the free build of IxWorks

The following steps are required to program the Flash ROM boot sectors:

1. Set switch S2 #’s 1 and 2 to the on position.
2. Reset the board by cycling power on the workstation.
3. Run the Intel DOS-based flash utility to program the Flash ROM boot sectors.
4. Set switch S2 #’s 1 and 2 to the off position.
5. Reset the board by cycling power on the workstation.

3.4 Console Serial Port

The console serial port on the IQ80303 platform, based on a 16C550 UART, is capable of operation from 300 to 115,200 bps. The port is connected to a phone jack-style plug on the IQ80303 platform. The DB25 to RJ-45 cable included with the IQ80303 can be used to connect the console port to any standard RS-232 port on the host system.

The UART on the IQ80303 platform is clocked with a 1.843 MHz clock, and may be programmed to use this clock with its internal baud rate counters. The UART register addresses are shown in Table 3-4; refer to the 16C550 device data book for a detailed description of the registers and device operation. Note that some UART addresses refer to different registers depending on whether a read or a write is being performed.

Table 3-4. UART Register Addresses

Address	Read Register	Write Register
E000 0000H	Receive Holding Register	Transmit Holding Register
E000 0001H	Unused	Interrupt Enable Register
E000 0002H	Interrupt Status Register	FIFO Control Register
E000 0003H	Unused	Line Control Register
E000 0014H	Unused	Modem Control Register
E000 0015H	Line Status Register	Unused
E000 0016H	Modem Status Register	Unused
E000 0017H	Scratchpad Register	Scratchpad Register

3.5 Secondary PCI Bus Expansion Connectors

Two PCI Expansion Slots are available on the IQ80303 platform. The IQ80303 supports 64-bit, +3.3 V, 33 MHz or 66 MHz PCI expansion. The Secondary PCI bus speed is controlled via a switch, 2 on S1.

3.5.1 Interrupt and IDSEL Routing

Table 3-5. Secondary PCI Bus Interrupt and IDSEL Routing

Connector	IDSEL	INTA#	INTB#	INTC#	INTD#
J11	SAD16	SINTA#	SINTB#	SINTC#	SINTD#
J12	SAD17	SINTB#	SINTC#	SINTD#	SINTA#
J13	SAD18	SINTC#	SINTD#	SINTA#	SINTB#
J14	SAD19	SINTD#	SINTA#	SINTB#	SINTC#

3.5.2 Battery Backup

Battery backup (also in Section 2.3.1, “Battery Backup) is provided to save any information in SDRAM during a power failure. The IQ80303 platform contains a Li-ion battery, a charging circuit and a regulator circuit. The battery installed in the IQ80303 platform is rated at 1040 mA/Hr.

SDRAM technology provides a simple way of enabling data preservation though the self-refresh command. When the processor receives an active Primary PCI reset it will issue the self-refresh command and drive the SCKE signals low. Upon seeing this condition a monitoring circuit on the IQ80303 platform will hold SCKE low before the processor loses power. The battery will maintain power to the SDRAM and the circuit to ensure self-refresh mode. When the circuit sees PRST# returning to inactive state it will release the hold on SCKE.

The battery circuit can be disabled by removing the battery. If the battery remains in the evaluation platform when it is depowered and/or removed from the chassis, the battery maintains the SDRAM for approximately TBD hours. Once power is again applied, the battery is fully charged in about TBD hours.

3.6 Logic Analyzer Headers

There are five logic analyzer connectors on the IQ80303 platform. The connectors are Mictor type, AMP part # 767054-1. Hewlett-Packard and Tektronix manufacture and sell interfaces to these connectors. The logic analyzer connectors allow for interfacing to the SDRAM and ROM buses along with secondary PCI arbitration signals. Table 3-6 shows the connectors and the pin assignments for each.

Table 3-6. Logic Analyzer Header Definitions

PIN	J12	J11	J10	J9	J13
3					
4	DQ15	SDQM7	DQ31	SHOLD#	RAD15
5	DQ14	SDQM6	DQ30	SHOLDA#	RAD14
6	DQ13	SDQM5	DQ29		RAD13
7	DQ12	SDQM4	DQ28		RAD12
8	DQ11	SDQM3	DQ27		RAD11
9	DQ10	SDQM2	DQ26		RAD10
10	DQ9	SDQM1	DQ25		RAD9
11	DQ8	SDQM0	DQ24		RAD8
12	DQ7	SCB7	DQ23		RAD7
13	DQ6	SCB6	DQ22		RAD6
14	DQ5	SCB5	DQ21		RAD5
15	DQ4	SCB4	DQ20		RAD4
16	DQ3	SCB3	DQ19	SCE0#	RAD3
17	DQ2	SCB2	DQ18	SCE1#	RAD2
18	DQ1	SCB1	DQ17	SBA1	RAD1
19	DQ0	SCB0	DQ16	SBA0	RAD0
20	DQ32	SA0	DQ48	SREQ0#	RAD16
21	DQ33	SA1	DQ49	SREQ1#	
22	DQ34	SA2	DQ50	SREQ2#	
23	DQ35	SA3	DQ51	SREQ3#	RALE
24	DQ36	SA4	DQ52	SREQ4#	RCE0#
25	DQ37	SA5	DQ53	SREQ5#	RCE1#
26	DQ38	SA6	DQ54	SGNT0#	ROE#
27	DQ39	SA7	DQ55	SGNT1#	RWE#
28	DQ40	SA8	DQ56	SGNT2#	
29	DQ41	SA9	DQ57	SGNT3#	I_RST#
30	DQ42	SA10	DQ58	SGNT4#	
31	DQ43	SA11	DQ59	SGNT5#	
32	DQ44	SA12	DQ60		
33	DQ45	SWE#	DQ61		
34	DQ46	SCAS#	DQ62		
35	DQ47	SRAS#	DQ63		
36			SDRAMCLK	P_PCICLK	RALE

3.7 JTAG Header

The JTAG header allows debugging hardware to be quickly and easily connected to some of the IQ80303 processor's logic signals.

The JTAG header is a 16-pin header. A 3M connector (part number 2516-6002UG) is required to connect to this header. The pinout for the JTAG header is shown in Table 3-7. The header and connector are keyed using a tab on the connector and a slot on the header, to ensure proper installation.

Each signal in the JTAG header is paired with its own ground connection to avoid the noise problems associated with long ribbon cables. Signal descriptions are found in the *Intel® 80303 Developer's Manual* (273353) and the *Intel® 80303 Datasheet* (273358).

Table 3-7. JTAG Header Pinout

Pin	Signal	Input/Output to 80303 IOP	Pin	Signal
1	TRST#	IN	2	SDA
3	TDI	IN	4	SCL
5	TDO	OUT	6	GND
7	TMS	IN	8	GND
9	TCK	IN	10	GND
11	LCDINIT#	IN	12	GND
13	I_RST#	OUT	14	GND
15	PWRVLD	OUT	16	GND

Table 3-8 and Table 3-9 describe S1 and S2 switch setting options and defaults. These switch settings are sampled at Primary PCI Reset. See Table 5-1 "Initialization Modes" on page 5-3 for processor initialization configurations.

Table 3-8. Switch S2 Settings

Position	Name	Description	Default
S2-1	RST_MODE#	Determines if the processor is to be held in reset. ON = hold in rest. OFF = allows processor initialization.	OFF
S2-2	RETRY	Determines if the Primary PCI interface will be disabled. ON = allows Primary PCI configuration cycles to occur. OFF = retries all Primary PCI configuration cycles.	OFF
S2-3	SPMEM#	Special Downstream Window enable. If enabled, a special dowstream memory window FEC0 0000h through FECF FFFFh is opened. ON = enabled. OFF = disabled.	OFF
S2-4	32BITPCI_EN#	Determines whether Secondary PCI bus is a 32- or 64-bit bus. ON = indicates Secondary PCI bus is a 32-bit bus. OFF = indicates Secondary PCI bus is a 64-bit bus	OFF

Table 3-9. Switch S1 Settings

Position	Name	Description	Default
S1-1	P_M66EN	User may force the primary PCI bus to start at 33 MHz mode. ON = forces the P-PCI to run at 33 MHz.	OFF
S1-2	S_M66EN	User may force the secondary PCI bus to start at 33 MHz mode. ON = forces the S-PCI to run at 33 MHz. *If S1-1 is ON, then S-PCI will run at 33 MHz	OFF

3.8 General Purpose I/O Header

The IQ80303 platform has eight programmable general purpose I/O pins. These pins are connected to a 16-pin 2.54 mm (0.100”) header. The definition of the header is shown in Table 3-10.

Table 3-10. GPIO Header Definition

PIN	SIGNAL	PIN	SIGNAL
1	GPIO0	2	GND
3	GPIO1	4	GND
5	GPIO2	6	GND
7	GPIO3	8	GND
9	GPIO4	10	GND
11	GPIO5	12	GND
13	GPIO6	14	GND
15	GPIO7	16	GND

3.9 User 7-Segment Displays

The IQ80303 platform has two user-programmable 7-segment displays located on the upper edge of the adapter board. The displays are controlled by two write-only registers and used as a debugging aid during development. Software can control the state of the user display by writing to the Display Registers, located at E004 0000H for the most significant digit and E005 0000h for the least significant digit. Each of the eight bits of these registers correspond to one of the display segments. Clearing a bit in the Display Register by writing a “0” to it turns the corresponding segment “on”, while setting a bit by writing a “1” to it turns the corresponding segment “off”. Resetting the IQ80303 platform results in clearing the register and turning all the segments “on”. The 7-segment Display Register bitmaps are shown in Figure 3-2 and Figure 3-3.

Figure 3-1. 7-Segment Display Bit Definition

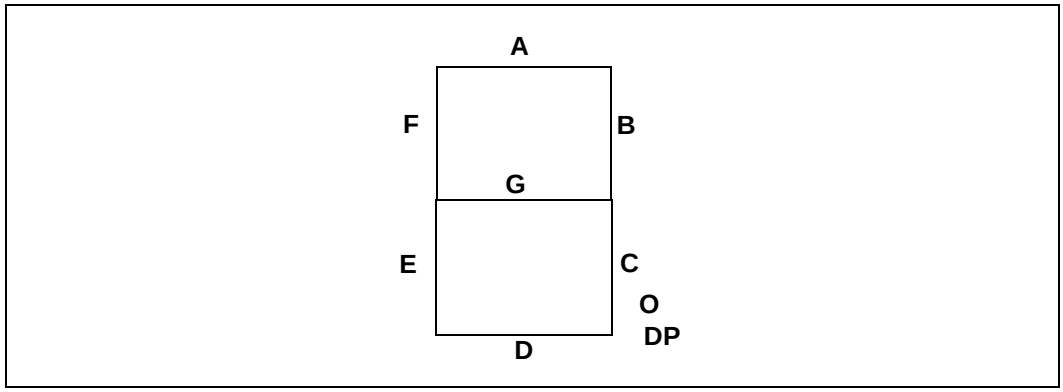


Figure 3-2. 7-Segment Display (MSB) Register Bitmap

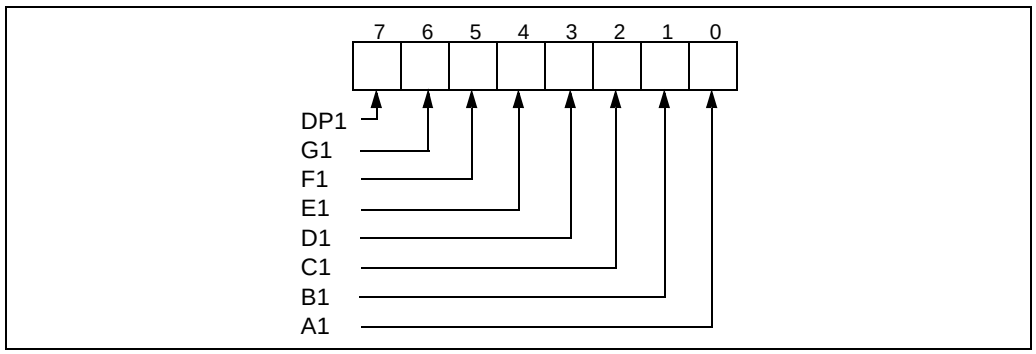
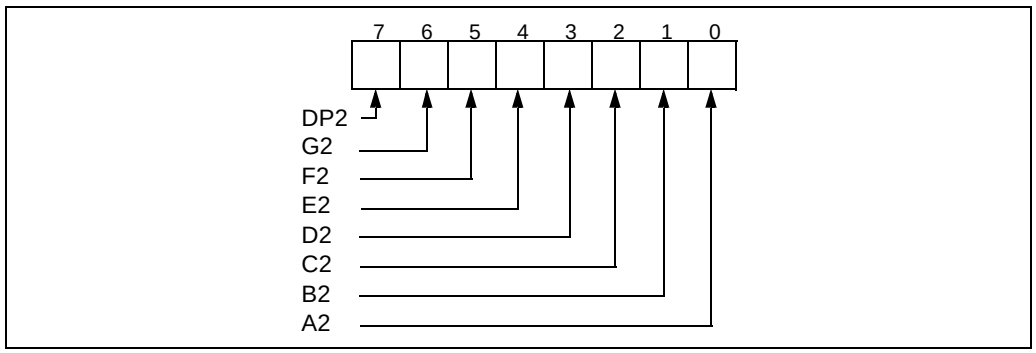


Figure 3-3. 7-Segment Display (LSB) Register Bitmap



3.9.1 User 7-Segment Displays During Initialization

MON960 indicates the progress of its hardware initialization on the user 7-segment displays. In the event that initialization should fail for some reason, the hex number on the displays can be used to determine the cause of the failure. Table 3-11 lists the tests and correspond hex value.

Table 3-11. Start-up 7-Segment Displays Intel® MON960

Display (HEX)	Tests
00	SDRAM serial EEPROM checksum validated
01	UART walking ones test passed
02	DRAM walking ones test passed
03	DRAM multiword test passed
04	Hardware initialization started
05	Flash ROM initialized
06	PCI-to-PCI Bridge initialized
07	UART internal loopback test passed

Table 3-12 lists the connectors and LEDs.

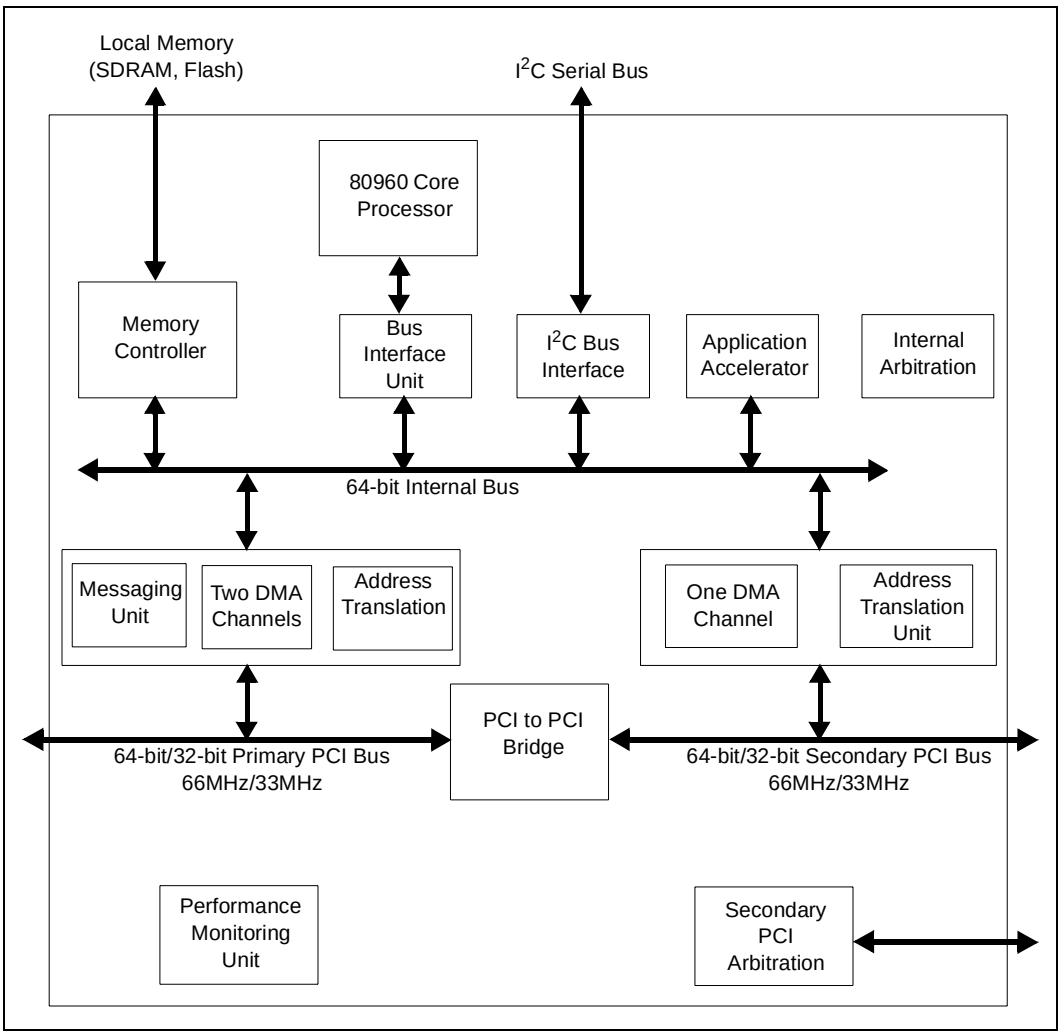
Table 3-12. Intel® IQ80303 Evaluation Platform Connectors and LEDs

Item	Description
J1-J2	Secondary PCI bus expansion connector
J5	168-pin SDRAM DIMM socket
J6	JTAG connector
J8	GPIO connector
J7	Serial port connector
J13	Logic analyzer connector for flash ROM bus
J9	Logic analyzer connector for Secondary PCI bus arbitration signals
J9,J10, J11, J12	Logic analyzer connector for access to SDRAM bus
CR1	Dual 7-segment display
CR3	Self-test fail LED
CR4	Primary PCI bus running @ 66 MHz
CR5	Secondary PCI bus running @ 66 MHz
CR2	Indicates host system providing 3.3 V
S1	DIP switch (Table TBD)
S2	DIP switch (Table 3-9)

Intel® 80303 I/O Processor Overview 4

This chapter describes the features and operation of the processor on the Intel® IQ80303 Evaluation Platform. For more detail, refer to the *Intel® 80303 Developer's Manual* (273353).

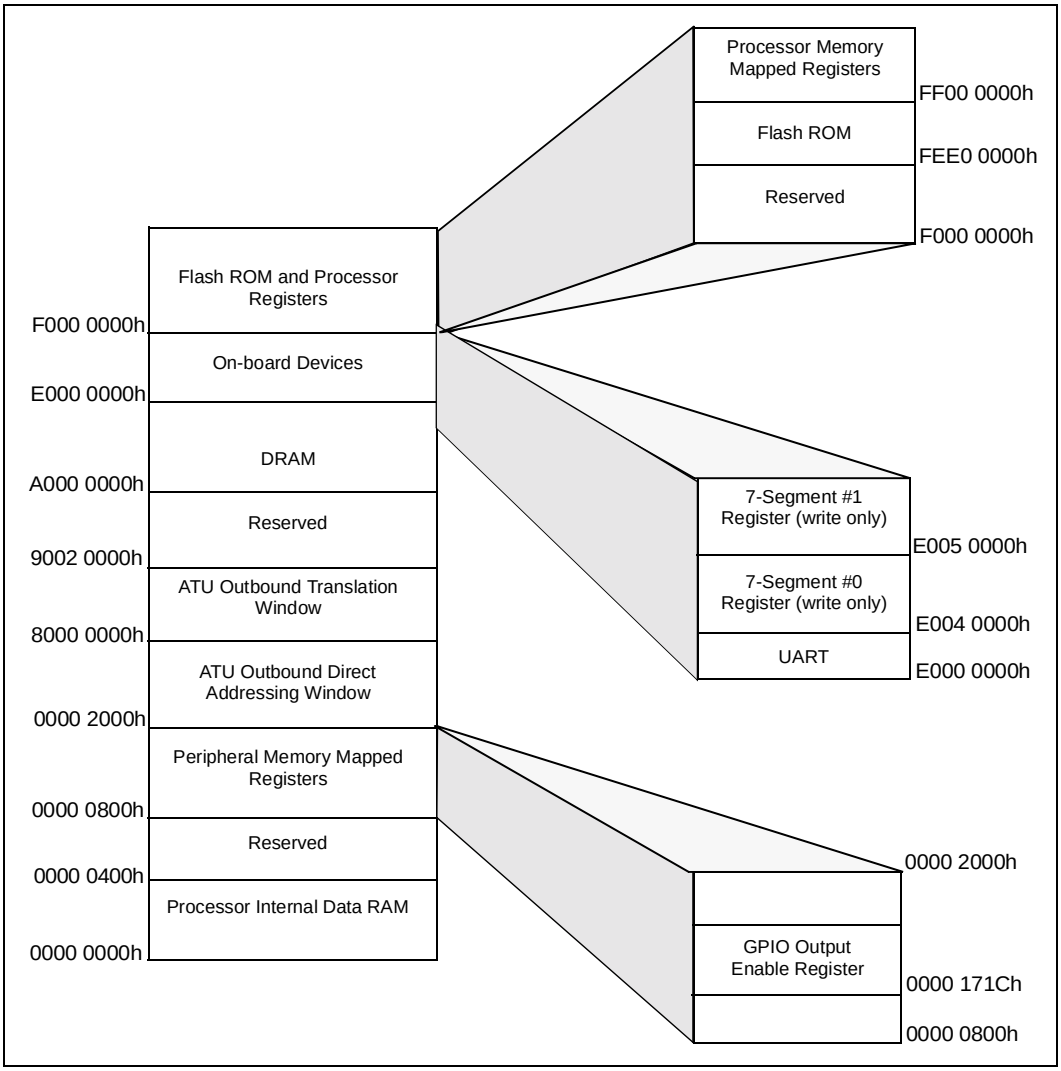
Figure 4-1. Intel® 80303 I/O Processor Block Diagram



4.1 CPU Memory Map

The memory map for the IQ80303 platform is shown in Figure 4-2. All addresses below 9002 0000H on the IQ80303 platform are reserved for various functions of the 80303® I/O processor, as shown on the memory map. Documentation for these areas, as well as the processor memory mapped registers at FF00 0000H and the IBR, can be found in the 80303® I/O Processor Developer' s Manual.

Figure 4-2. Intel® IQ80303 Evaluation Platform Memory Map



4.2 Local Interrupts

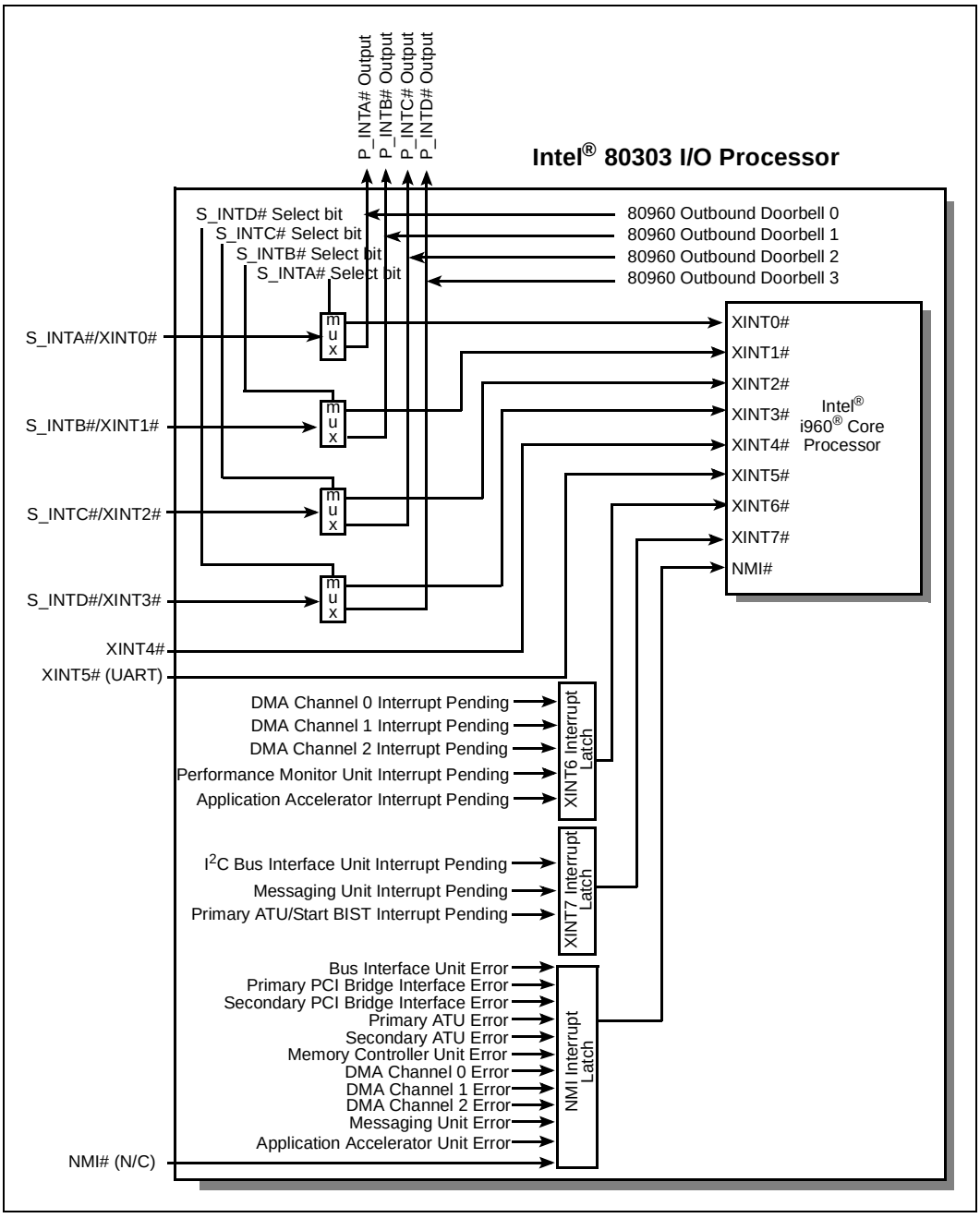
The 80303® I/O processor is built around an 80960JT core, which has seven external interrupt lines designated XINT0# through XINT5# and NMI#. In the 80303 I/O processor, these interrupt lines are not directly connected to external interrupts, but pass through a layer of internal interrupt routing logic. Figure 4-3 shows the interrupt connections on the 80303 I/O processor.

XINT0# through XINT3# on the 80960JT core can be used to receive PCI interrupts from the secondary PCI bus, or these interrupts can be passed through to the primary PCI interface, depending on the setting of the XINT Select bit of the PCI Interrupt Routing Select Register in the 80303 I/O processor. On the IQ80303 platform, XINT0# through XINT3# are configured to receive interrupts from the secondary PCI bus.

XINT4# and XINT5# on the 80303 I/O processor may be connected to interrupt sources external to the processor. On the IQ80303 platform, XINT4# is not used and XINT5# is connected to the 16C550 UART.

XINT6#, XINT7# receive interrupts from internal sources. NMI# receives interrupts from internal sources and from an external source. Since all of these interrupts accept signals from multiple sources, a status register is provided for each of them to allow service routines to identify the source of the interrupt. Each of the possible interrupt sources is assigned a bit position in the status register. The interrupt sources for these lines are shown in Figure 4-3. On the IQ80303 platform, the NMI# interrupt is not connected to any external interrupt source and receives interrupts only from the internal devices on the 80303 I/O processor. Note that all error conditions result in an NMI# interrupt.

Figure 4-3. Intel® 80303 I/O Processor Interrupt Controller Connections



4.3 CPU Counter/Timers

The 80303 I/O processor is equipped with two on-chip counter/timers, clocked with the 80303 I/O processor clock signal. The 80303 I/O processor receives its clock from the primary PCI interface clock, generated by the motherboard. Most motherboards generate a 33 MHz clock signal. *PCI Local Bus Specification*, Revision 2.1 requires a clock frequency between 0 and 33 MHz. The timers can be programmed for single-shot or continuous mode, and can generate interrupts to the processor when the countdown expires.

4.4 Primary PCI Interface

The primary PCI interface on the IQ80303 platform provides the 80303 I/O processor with a connection to the PCI bus on the host system. Only the PCI-to-PCI bridge unit on the 80303 I/O processor is directly connected to the primary PCI interface. Devices installed on the expansion slots are connected to the PCI bus via the bridge unit on the 80303 I/O processor. The PCI-to-PCI bridge accepts Type 1 configuration cycles destined for devices on the secondary bus, and forwards them as Type 0 or Type 1 configuration cycles, or as special cycles. The IQ80303 platform interfaces to a 64-bit PCI bus. The primary PCI bus can operated at either 33 MHz or 66 MHz. Refer to Table 4-1, “Switch S1 Settings” on page 4-6, to determine the primary PCI bus speed.

4.5 Secondary PCI Interface

The secondary PCI interface provided by the 80303 I/O processor is used to connect PCI cards via the expansion slots to the host system PCI bus. PCI cards are attached to the IQ80303 platform with a standard PCI connector and may contain up to two separate PCI devices. The 80303 I/O processor provides PCI-to-PCI bridge functionality to map installed PCI devices onto the host PCI bus, and supports transaction forwarding in both directions across the bridge. PCI devices connected via the expansion slots can therefore act as masters or slaves on the host system PCI bus. Additional PCI-to-PCI bridge devices are supported by the 80303 I/O processor on its secondary PCI interface and can be designed into add-on PCI cards. In addition, the 80303 I/O processor supports “private” PCI devices on its secondary bus. Private devices are hidden from initialization code on the host system, and are configured and accessed directly by the 80303 I/O processor. These devices are not part of the normal PCI address space, but can act as PCI bus masters and transfer data to and from other PCI devices in the system.

Unless designated as private devices, PCI devices installed on the secondary PCI interface of the IQ80303 platform are mapped into the system-wide PCI address space by configuration software running on the host system. No logical distinction is made at the system level between devices on the primary PCI bus and devices on secondary busses; all transaction forwarding is handled transparently by the PCI-to-PCI bridge. Configuration cycles and read and write accesses from the host are forwarded through the PCI-to-PCI bridge unit of the 80303 I/O processor. Master read and write cycles from devices on the secondary PCI bus are also forwarded to the host bus by the PCI-to-PCI bridge unit.

IxWORKS allows secondary PCI devices to be configured as Public or Private. Public devices are configured by the PCI host. Private devices are configured by the IxWORKS kernel and the device-specific HDM.

As with the primary PCI bus, the secondary PCI bus can operated at either 33 MHz or 66 MHz. Refer to Table 4-1, S1 Switch Settings, to determine the secondary PCI bus speed.

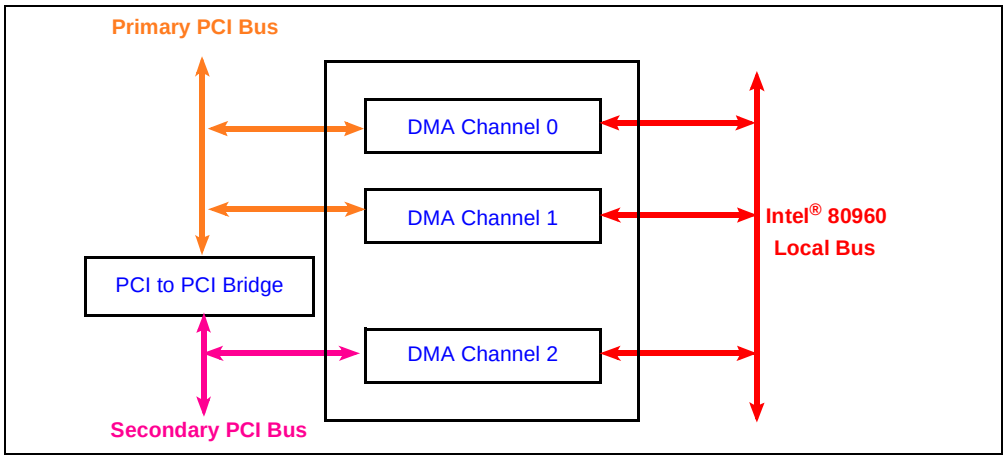
Table 4-1. Switch S1 Settings

Position	Name	Description	Default
S1-1	P_M66EN	Determines the clock rate of the Primary PCI bus. ON = 33 MHz OFF = 66 MHz	OFF
S1-2	S_M66EN	Determines the clock rate of the Secondary PCI bus. ON = 33 MHz OFF = 66 MHz	OFF

4.6 DMA Channels

The 80303 I/O processor features three independent DMA channels, two of which operate on the primary PCI interface, whereas the remaining one operates on the secondary PCI interface. All three of the DMA channels connect to the 80303 I/O processor's local bus and can be used to transfer data from PCI devices to memory on the IQ80303 platform. Support for chaining, and scatter/gather is built into all three channels. The DMA can address the entire 2^{64} bytes of address space on the PCI bus and 2^{32} bytes of address space on the internal bus.

Figure 4-4. Intel® IQ80303 Evaluation Platform DMA Controller



4.7 Application Accelerator Unit

The Application Accelerator provides low-latency, high-throughput data transfer capability between the AA unit and 80960 local memory. It executes data transfers to and from 80960 local memory and also provides the necessary programming interface. The Application Accelerator performs the following functions:

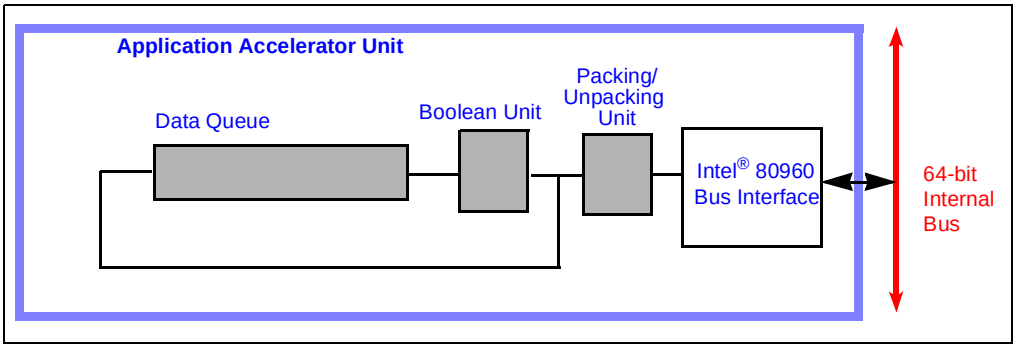
- Transfers data (read) from memory controller
- Performs an optional boolean operation (XOR) on read data
- Transfers data (write) to memory controller

The AA unit features:

- 1K-byte, arranged as 8-byte x 128-deep store queue
- Utilization of the 80303 processor memory controller interface
- 2^{32} addressing range on the 80960 local memory interface
- Hardware support for unaligned data transfers for the internal bus
- Full programmability from the i960 core processor
- Support for automatic data chaining for gathering and scattering of data blocks

Figure 4-5 shows a simplified connection of the Application Accelerator to the 80303 I/O Processor Internal Bus.

Figure 4-5. Application Accelerator Unit



4.8 Performance Monitor Unit

The Performance Monitoring features aid in measuring and monitoring various system parameters that contribute to the overall performance of the processor. The monitoring facility is generically referred to as PMON – Performance Monitoring. The facility is model specific, not architectural; its intended use is to gather performance measurements that can be used to retune/refine code for better system level performance.

The PMON facility provided on the 80303 I/O processor comprises:

- One dedicated global Time Stamp counter, and
- Fourteen (14) Programmable Event counters

The global time stamp counter is a dedicated, free running 32-bit counter.

The programmable event counters are 32-bits wide. Each counter can be programmed to observe an event from a defined set of events. An event consists of a set of parameters which define a start condition and a stop condition. The monitored events are selected by programming an event select register (ESR).

Intel® MON960 Support for Intel® 80303 I/O Processor 5

This chapter discusses a number of additions that have been made to Intel® MON960 to support the Intel® IQ80303 Evaluation Platfrom in an optional non-I_2O* capacity. For complete documentation on the operation of MON960, see the *MON960 Debug Monitor User's Guide*. The IQ80303 evaluation platform ships with IxWorks* from Wind River Systems installed in flash firmware. To use Intel® CTOOLS and MON960 instead of IxWorks, you need to download MON960 into the onboard Flash. See Chapter 2 for more information on updating the onboard Flash. See Chapter 1 for descriptions of both IxWorks and CTOOLS.

5.1 Secondary PCI Bus Expansion Connectors

The IQ80303 platform contains four secondary PCI bus expansion connectors for secondary PCI bus access to the 80303 I/O processor. Extensions to MON960 perform secondary PCI bus initialization, including the establishment of a secondary PCI bus address map. Routines compatible with *PCI Local Bus Specification,* Revision 2.1 allow software on the IQ80303 platform to search for devices on the secondary PCI bus and, read and write the configuration space of those devices.

5.2 Intel® MON960 Components

The remaining sections of this chapter assume that MON960 is installed in the onboard Flash, replacing IxWorks. The IQ80303 optional MON960 debug monitor consists of four main components:

- Initialization firmware
- MON960 kernel
- MON960 extensions
- Diagnostics/example code

These four components together are referred to as MON960.

5.2.1 Intel® MON960 Initialization

At initialization, MON960 puts the IQ80303 platform into a known functional state, allowing the host processor to perform PCI initialization. Once in this state, the MON960 kernel and extensions can load and execute correctly. Initialization is performed after a RESET. MON960 initialization encompasses all major portions of the 80303 I/O processor and IQ80303 platform, including 80960JT core initialization, Memory Controller initialization, SDRAM initialization, Primary PCI Address Translation Unit (ATU) initialization, and PCI-to-PCI Bridge Unit initialization.

The IQ80303 platform is designed to use the Configuration Mode of the 80303 I/O processor. Configuration Mode allows the 80960JT core to initialize and control the initialization process before the PCI host configures the 80303 I/O processor. By utilizing Configuration Mode, the user is given the ability to initialize the PCI configuration registers to values other than the default power-up values. Configuration Mode gives the user maximum flexibility to customize the way in which the 80303 I/O processor and IQ80303 platform appear to the PCI host configuration software.

5.2.2 Intel® 80960JT Core Initialization

The Intel® 80960JT core begins the initialization process by reading its Initial Memory Image (IMI) from a fixed address in the boot ROM (FEFF FF30H in the i960 address space). The IMI includes the Initialization Boot Record (IBR), the Process Control Block (PRCB), and several system data structures. The IBR provides initial configuration information for the core and integrated peripherals, pointers to the system data structures and the first instruction to be executed after processor initialization, and checksum words that the processor uses in its self-test routine. In addition to the IBR and PRCB, the required data structures are the:

- System Procedure Table
- Control Table
- Interrupt Table
- Fault Table
- User Stack (application dependent)
- Supervisor Stack
- Interrupt Stack

5.2.3 Memory Controller Initialization

Since the 80303 I/O processor Memory Controller is integral to the design and operation of the IQ80303 platform, the operational parameters for Bank 0 and Bank 1 are established immediately after processor core initialization. Memory Bank 0 is associated with the ROM on the IQ80303 platform. Memory Bank 1 is associated with the UART and the LED Control Register. Parameters such as Bank Base Address, Read Wait States, and Write Wait States must be established to ensure the proper operation of the IQ80303 platform. The Memory Controller is initialized so as to be consistent with the IQ80303 platform memory map shown in Figure 4-2.

5.2.4 SDRAM Initialization

SDRAM initialization includes setting operational parameters for the SDRAM controller, and sizing and clearing the installed SDRAM configuration. To configure the system properly, Presence Detect data is read from the EEPROM of the SDRAM module, using the IQ80303 I^2C Bus Interface Unit. Presence Detect data includes the number and size of SDRAM banks present on the installed module. On power-up, 64 bytes of Presence Detect data are read and validated. The SDRAM controller is then configured by setting the base address of SDRAM, the boundary limits for each SDRAM bank, the refresh cycle interval, and the output buffer drive strength. Once the SDRAM controller is configured, the SDRAM is cleared in preparation for the C language runtime environment. The actual SDRAM size is stored for later use (e.g., to establish the size of the IQ80303 platform PCI Slave image). The SDRAM controller is initialized to be consistent with the IQ80303 platform memory map shown in Figure 4-2.

5.2.5 Primary PCI Interface Initialization

The IQ80303 platform is a multi-function PCI device. On the primary PCI bus, two functions (from a PCI Configuration Space standpoint) are supported.

- Function 0 is the PCI-to-PCI Bridge of the 80303 I/O processor, which optionally provides access capability between the primary PCI bus and the secondary PCI bus.
- Function 1 is the Primary ATU providing access capability between the primary PCI bus and the local 80960 bus.

The platform can be initialized into one of four modes. Modes 0 and 3 are described below.

Table 5-1. Initialization Modes

RST_MODE#/SW1-1	RETRY/SW1-2	Initialization Mode	Primary PCI Interface	Intel® i960® Core Processor
0/ON	0/ON	Mode 0	Accepts Transactions	Held in Reset
0/ON	1/OFF	Mode 1	Retry All Configuration Transactions	Held in Reset
1/OFF	0/ON	Mode 2	Accepts Transactions	Initializes
1/OFF	1/OFF	Mode 3 (default)	Retry All Configuration Transactions	Initializes

When the IQ80303 is operating in Mode 0, the processor core is held in reset, allowing register defaults to be used on the Primary PCI interface. This mode is used to program the onboard Flash with either IxWORKS* or MON960.

When the IQ80303 platform is operating in Mode 3, the Configuration Cycle Disable bit in the Extended Bridge Control Register (EBCR) is set after IQ80303 processor reset. In this mode, the IQ80303 platform sends PCI Retries when the PCI host attempts to access the platform Configuration Space. This mode allows the IQ80303 processor time to initialize internal registers. The processor remains in this mode until the Configuration Cycle Disable bit in the EBCR is cleared. For this reason, and to prevent PCI host problems, Primary PCI initialization occurs at the earliest possible opportunity after Memory and SDRAM controller initialization.

5.2.6 Primary ATU Initialization

Primary ATU (Bridge) initialization includes initialization by the 80960JT core and initialization by the PCI host processor. Local initialization occurs first and consists mainly of establishing the operational parameters for local IQ80303 platform bus access. The Primary Inbound ATU Limit Register (PIALR) is initialized to establish the block size of memory required by the Primary ATU. The PIALR value is based on the installed SDRAM configuration. The Primary Inbound ATU Translate Value Register (PIATVR) is initialized to establish the translation value for PCI-to-Local accesses. The PIATVR value is set to reference the base of local SDRAM. The Primary Outbound Memory Window Value Register (POMWVR) is initialized to establish the translation value for Local-to-PCI accesses. The POMWVR value remains at its default value of “0” to allow the IQ80303 platform to access the start of the PCI Memory address map, typically occupied by PCI host memory. Likewise, the Primary Outbound I/O Window Value Register (POIOWVR) remains at its default value of “0” to allow the IQ80303 platform to access the start of the PCI I/O address map. PCI Doorbell-related parameters are also established to allow for communication between the IQ80303 platform and a PCI bus master using the doorbell mechanism.

By default, Primary Outbound Configuration Cycle parameters are not established. The ATU Configuration Register (ATUCR) is initialized to establish the operational parameters for the Doorbell Unit and ATU interrupts (both primary and secondary), and to enable the primary and secondary ATUs. The PCI host is responsible for allocating PCI address space (Memory, Memory Mapped I/O, and I/O), and assigning the PCI Base addresses for the IQ80303 platform.

5.2.7 PCI-to-PCI Bridge Initialization

PCI-to-PCI Bridge initialization includes initialization by the 80960JT core and initialization by the PCI host processor. Local initialization occurs first and consists mainly of establishing the operational parameters for the secondary PCI interface of the PCI-to-PCI bridge. On the IQ80303 platform, the secondary PCI bus is configured to consist of private devices (not visible to PCI host configuration cycles). To support a private secondary PCI bus, the Secondary IDSEL Select Register (SISR) is initialized to prevent the secondary PCI address bits [20:16] from being asserted during conversion of PCI Type 1 configuration cycles on the primary PCI bus to PCI Type 0 configuration cycles on the secondary PCI bus. Secondary PCI bus masters are prevented from initiating transactions that will be forwarded to the primary PCI interface. The PCI host is responsible for assigning and initializing the PCI bus numbers, allocating PCI address space (Memory, Memory Mapped I/O, and I/O), and assigning the IRQ numbers to valid interrupt routing values.

5.2.8 Secondary ATU Initialization

Secondary ATU (Bridge) initialization consists mainly of establishing the operational parameters for access between the local IQ80303 platform bus and the secondary PCI devices. The Secondary Inbound ATU Base Address Register (SIABAR) is initialized to establish the PCI base address of IQ80303 platform local memory from the secondary PCI bus. By convention, the secondary PCI base address for access to IQ80303 platform local memory is “0”. The Secondary Inbound ATU Limit Register (SIALR) is initialized to establish the block size of memory required by the secondary ATU. The SIALR value is based on the installed SDRAM configuration. The Secondary Inbound ATU Translate Value Register (SIATVR) is initialized to establish the translation value for Secondary PCI-to-Local accesses. The SIATVR value is set to reference the base of local SDRAM. The Secondary Outbound Memory Window Value Register (SOMWVR) is initialized to establish the translation value for Local-to-Secondary PCI accesses. The SOMWVR value is left at its default value of “0” to allow the IQ80303 platform to access the start of the PCI Memory address map. Likewise, the Secondary Outbound I/O Window Value Register (SOIOWVR) is left at its default value of “0” to allow the IQ80303 platform to access the start of the PCI I/O address map.

On the secondary PCI bus, the IQ80303 platform assumes the duties of PCI host and, as such, is required to configure the devices of the secondary PCI bus. Secondary Outbound Configuration Cycle parameters are established during secondary PCI bus configuration. Secondary PCI bus configuration is accomplished via MON960 Extension routines.

5.3 Intel® MON960 Kernel

The MON960 Kernel (monitor) provides the IQ80303 user with a software platform on which application software can be developed and run. The monitor provides several features available to the IQ80303 user to speed application development. Among the available features are:

- Communication with a terminal or terminal emulation package on a host computer through a serial cable with automatic baud rate detection
- Communication with a software debugger such as GDB960 (available from Intel) using the Host Debugger Interface (HDI) software interface
- Communication with the host computer via the primary PCI bus
- Downloads of ELF object files via the primary PCI bus or via the serial console port at rates up to 115,200 baud
- Downloads of ELF object files via the primary PCI bus
- On-board erasure and programming of Intel 28F016S5 Flash ROM
- Memory display and modification capability
- Breakpoint and single-step capability to support debugging of user code
- Disassembly of i960 processor instructions

5.4 Intel® MON960 Extensions

The monitor has been extended to include the secondary PCI bus initialization and also the BIOS routines which are contained in the *PCI BIOS Specification* Revision 2.1.

5.4.1 Secondary PCI Initialization

MON960 extensions are responsible for initializing the devices on the secondary PCI bus of the IQ80303 platform. Secondary PCI initialization involves allocating address spaces (Memory, Memory Mapped I/O, and I/O), assigning PCI base addresses, assigning IRQ values, and enabling PCI mastership. MON960 does not support devices containing PCI-to-PCI bridges and hierarchical buses.

5.4.2 PCI BIOS Routines

MON960 includes PCI BIOS routines to aid application software initialization of the secondary PCI bus. The supported BIOS functions are described in the subsections that follow.

- sysPCIBIOSPresent
- sysFindPCIDevice
- sysFINDPCIClassCode
- sysGenerateSpecialCycle
- sysReadConfigByte
- sysReadConfigWord
- sysReadConfigDword
- sysWriteConfigByte
- sysWriteConfigWord
- sysWriteConfigDword
- sysGetIrqRoutingOptions
- sysSetPCIIrq

These functions preserve, as closely as possible, the parameters and return values described in the *PCI Local Bus Specification*, Revision 2.1. Functions that return multiple values do so by filling in the fields of a structure passed by the calling routine.

You can access these functions via a `calls` instruction. The system call indices are defined in the MON960 source file `PCI_BIOS.H`. The function prototypes are defined in the `IQRP_ASM.H` file.

5.4.2.1 sysPCIBIOSPresent

This function allows the caller to determine whether the PCI BIOS interface function set is present, and the current interface version level. It also provides information about the hardware mechanism used for accessing configuration space and whether or not the hardware supports generation of PCI Special Cycles.

Calling convention:

```
int sysPCIBIOSPresent (
       PCI_BIOS_INFO *info
       );
```

Return values:

This function always returns SUCCESSFUL.

5.4.2.2 sysFindPCIDevice

This function returns the location of PCI devices that have a specific Device ID and Vendor ID. Given a Vendor ID, a Device ID, and an Index, the function returns the Bus Number, Device Number, and Function Number of the Nth Device/Function whose Vendor ID and Device ID match the input parameters.

Calling software can find all devices having the same Vendor ID and Device ID by making successive calls to this function starting with the index set to “0”, and incrementing the index until the function returns DEVICE_NOT_FOUND. A return value of BAD_VENDOR_ID indicates that the Vendor ID value passed had a value of all “1”s.

Calling convention:

```
int sysFindPCIDevice (
        int     device_id,
        int     vendor_id,
        int     index
        );
```

Return values:

This function returns SUCCESSFUL if the indicated device is located, DEVICE_NOT_FOUND if the indicated device cannot be located, or BAD_VENDOR_ID if the vendor_id value is illegal.

5.4.2.3 sysFindPCIClassCode

This function returns the location of PCI devices that have a specific Class Code. Given a Class Code and an Index, the function returns the Bus Number, Device Number, and Function Number of the Nth Device/Function whose Class Code matches the input parameters.

Calling software can find all devices having the same Class Code by making successive calls to this function starting with the index set to “0”, and incrementing the index until the function returns DEVICE_NOT_FOUND.

Calling convention:

```
int sysFindPCIClassCode (
    int class_code,
    int index
    );
```

Return values:

This function returns SUCCESSFUL when the indicated device is located, or DEVICE_NOT_FOUND when the indicated device cannot be located.

5.4.2.4 sysGenerateSpecialCycle

This function allows for generation of PCI Special Cycles. The generated special cycle is broadcast on a specific PCI Bus in the system.

PCI Special Cycles are not supported on the IQ80303 platform secondary PCI bus.

Calling convention:

```
int sysGenerateSpecialCycle (
    int bus_number,
    int special_cycle_data
    );
```

Return values:

Since PCI Special Cycles are not supported by the IQ80303 platform, this function always returns FUNC_NOT_SUPPORTED.

5.4.2.5 sysReadConfigByte

This function allows the caller to read individual bytes from the configuration space of a specific device.

Calling convention:

```
int sysReadConfigByte (
    int bus_number,
    int device_number,
    int function_number,
    int register_number,   /* 0,1,2,...,255 */
    UINT8   *data
    );
```

Return values:

This function returns SUCCESSFUL when the indicated byte was read correctly, or ERROR when there is a problem with the parameters.

5.4.2.6 sysReadConfigWord

This function allows the caller to read individual shorts (16 bits) from the configuration space of a specific device. The Register Number parameter must be a multiple of two (i.e., bit 0 must be set to “0”).

Calling convention:

```
int sysReadConfigWord (
    int bus_number,
    int device_number,
    int function_number,
    int register_number,   /* 0,2,4,...,254 */
    UINT16   *data
    );
```

Return values:

This function returns SUCCESSFUL when the indicated word was read correctly, or ERROR when there is a problem with the parameters.

5.4.2.7 sysReadConfigDword

This function allows the caller to read individual longs (32 bits) from the configuration space of a specific device. The Register Number parameter must be a multiple of four (i.e., bits 0 and 1 must be set to “0”).

Calling convention:

```
int sysReadConfigDword (
    int bus_number,
    int device_number,
    int function_number,
    int register_number,   /* 0,4,8,...,252 */
    UINT32 *data
    );
```

Return values:

This function returns SUCCESSFUL when the indicated long was read correctly, or ERROR when there is a problem with the parameters.

5.4.2.8 sysWriteConfigByte

This function allows the caller to write individual bytes to the configuration space of a specific device.

Calling convention:

```
int sysWriteConfigByte (
    int bus_number,
    int device_number,
    int function_number,
    int register_number,   /* 0,1,2,...,255 */
    UINT8   *data
    );
```

Return values:

This function returns SUCCESSFUL when the indicated byte was written correctly, or ERROR when there is a problem with the parameters.

5.4.2.9 sysWriteConfigWord

This function allows the caller to write individual shorts (16 bits) to the configuration space of a specific device. The Register Number parameter must be a multiple of two (i.e., bit 0 must be set to “0”).

Calling convention:

```
int sysWriteConfigWord (
    int bus_number,
    int device_number,
    int function_number,
    int register_number,   /* 0,2,4,...,254 */
    UINT16  *data
    );
```

Return values:

This function returns SUCCESSFUL when the indicated word was written correctly, or ERROR when there is a problem with the parameters.

5.4.2.10 sysWriteConfigDword

This function allows the caller to write individual longs (32 bits) to the configuration space of a specific device. The Register Number parameter must be a multiple of four (i.e., bits 0 and 1 must be set to “0”).

Calling convention:

```
int sysWriteConfigDword (
    int bus_number,
    int device_number,
    int function_number,
    int register_number,   /* 0,4,8,...,252 */
    UINT32  *data
    );
```

Return values:

This function returns SUCCESSFUL when the indicated long was written correctly, or ERROR when there is a problem with the parameters.

5.4.2.11 sysGetIrqRoutingOptions

The PCI Interrupt routing fabric on the IQ80303 platform is not reconfigurable (fixed mapping relationships); therefore, this function is not supported.

Calling convention:

```
int sysGetIrqRoutingOptions (
    PCI_IRQ_ROUTING_TABLE *table
    );
```

Return values:

This function always returns FUNC_NOT_SUPPORTED.

5.4.2.12 sysSetPCIIrq

The PCI Interrupt routing fabric on the IQ80303 platform is not reconfigurable (fixed mapping relationships); therefore, this function is not supported.

Calling convention:

```
int sysSetPCIIrq (
    int int_pin,
    int irq_num,
    int bus_dev
    );
```

Return values:

This function always returns FUNC_NOT_SUPPORTED.

5.4.3 Additional Intel® MON960 Commands

The following commands have been added to the UI interface of MON960 to support the IQ80303 platform.

5.4.3.1 print_pci Utility

A print_pci command to MON960 is accessed through the MON960 command prompt. This command displays the contents of the PCI configuration space on a selected adapter on the secondary PCI interface or on the 80303 I/O processor itself. For more information on the meaning of the fields in PCI configuration space, refer to the *PCI Local Bus Specification*, Revision 2.1. The syntax of this command is:

pp <bus number> <device number> <function number>

5.5 Diagnostics / Example Code

IQ80303 platform diagnostic routines serve a twofold purpose: to verify proper hardware operation and to provide example code for users who need similar functions in their applications. Diagnostic routines fall into two categories: board level diagnostics and PCI expansion module diagnostics.

5.5.1 Board Level Diagnostics

Board level diagnostics exercise all basic areas of the IQ80303 platform. Diagnostic routines include SDRAM tests, UART tests, LED tests, internal timer tests, I^2C bus tests, and primary PCI bus tests. Primary PCI bus tests exercise the primary ATU, the PCI Doorbell unit, and the PCI DMA controller. Interrupts from both local and PCI sources are generated and handled. The PCI bus tests require an external test suite running on a PC to verify complete functionality of the IQ80303 platform.

5.5.2 Secondary PCI Diagnostics

Secondary PCI diagnostics exercise the secondary PCI bus, thereby confirming hardware functionality, as well as illustrating the use of the PCI BIOS routines present in MON960.

Bill of Materials A

Table A-1. Intel® IQ80303 Evaluation Platfrom Bill of Materials (Sheet 1 of 4)

Location	Descripition	Qty	Manufacturer	Part Number
U4, U22	IC/SM 74LVC04D SOIC14 3.3V	2	Texas Instruments	SN74LVC04D
			Fairchild Semi.	74LVX04M
U2, U18	IC/SM 74LVC08 3.3V TSSOP-14	2	Texas Instruments	SN74LVC08APW
U3	IC/SM 74LVC14 3.3V TSSOP-14	1	Texas Instruments	SN74LVC14APW
			Philips	74LVC14APWDH
			Fairchild Semi.	74LCX14MTC
U25	IC/SM 74LVT125 (3.3V) SOIC-14	1	Texas Instruments	SN74LVT125D
			Philips	74LVT125D
U8, U9	IC/SM 74LVTH273 TSSOP-20 (3.3V)	2	Texas Instruments	SN74LVTH273PW
			Fairchild Semi.	74LVTH273MTC
U12, U19	IC/SM 74LVT573 SOIC-20 (3.3V)	2	Texas Instruments	SN74LVT573DW
			Fairchild Semi.	74LVT573WM
			Philips	74LVT573D
U1, U17, U21	IC/SM 74C32 TSSOP-14	3	Texas Instruments	SN74LVC32APW
			Fairchild	74LCX32MTC
Q5-Q7	IC/SM Si3441DV TSOP-6	3	Siliconix	Si3441DV
			Motorola	MGSF3441VT1
U11, U13	IC/SM LM393 Low Power SOIC-8	2	National Semi.	LM393M
			Texas Instruments	LM393D
			Motorola	LM393D
CR7	IC/SM LM385 SOIC-8	1	National Semi.	LM385M
			Texas Instruments	LM385D-1.2
			Linear Tech.	LM385S8-1.2
U23	IC/SM ADP380 Battery Charger	1	Analog Devices	ADP3801AR
U10	IC/SM MAX6328 SOT23-3	1	Maxim	MAX6328UR23-T
U5	IC/SM MAX921 SOIC-8	1	Maxim	MAX921CSA
U6	IC/SM MAX8863 SOT23-5	1	Maxim	MAX8863TEUK-T
U14	PROCESSOR 80303 (from Intel)	1	Intel	TBD

Table A-1. Intel® IQ80303 Evaluation Platfrom Bill of Materials (Sheet 2 of 4)

Location	Descripition	Qty	Manufacturer	Part Number
U16	VLSI I/O UART 16C550 PLCC 3.3V	1	Texas Instruments	TL15C550CFN
			Exar	ST16C550CJ44
U7	VLSI I/O RS232 Transvr 3.3V	1	Maxim	MAX3232ECUP
U15	MEM Flash E28F016S3-120 3.3V	1	Intel	E28F016S3-120
C114	CAP SM 20pf (chip 1206)	1		
C68	CAP SM 100pf (chip 1206)	1	AVX	1206A101JATMA
C77, C92	CAP SM 1000pf (chip 1206)	2	AVX	12061A102JATMA
C63	CAP TANT SM 0.22uf, 10V (1206)	1	AVX	TAJA244M035R
C70	CAP TANT SM 220uf, 10V (7343)	1	AVX	TPSE227K010R010
C95, C97, C99	CAP TANT SM 47uf, 16V (7343)	3	AVX	TPSD476K016R015
C48, C61, C85, C89	CAP TANT SM 4.7uf, 35V (7343)	4	Sprague	293D475X9035D2T
			AVX	TAJD475K035R
C41, C79	CAP TANT SM 22uf, 20V (7343)	2	Sprague	293D226X9020D2T
C102	CAP TANT SM 10uf 25/35V (7343)	1	Sprague	293D1060025D2T
C2, C5, C7, C12, C13, C16, C19, C21, C22, C24, C26, C69, C78, C84, C106	CAP CERM SM, 0.01uf 50V (0805)	15		
C1, C3, C4, C6, C8-C11, C14, C15, C17, C18, C20, C23, C25, C27-C40, C43-C47, C49-C55, C57-C60, C62, C64-C67, C71-C76, C80-C83, C86-C88, C90, C91, C93, C94, C96, C98, C100, C101, C103-C105, C107-C113, C115-C128	CAPCERM SM, 0.1uf(0805)50V,10%	95		
C56	CAP CERM SM, 470pf (0805)	1	Murata	GRM40COG471JAT2A
			AVX	08055A471JAT2A
			Phillips	0805CG471JOBB2
C42	CAP CERM SM, 22pf (0805)	1	Kemet	C0805C220J5GACTU
R41	R/SM 1/16W 1% 10K ohm (0603)	1	Dale	CRCW06031002FT
R64	R/SM 1/16W 5% 1.2M ohm (0603)	1		
R48	R/SM 1/16W 1% 14.7K ohm (0603)	1	Dale	CRCW06031472FT
R25, R40	R/SM 1/16W 1% 150K ohm (0603)	2	KOA	RK73H1J1503FT
R76	R/SM 1/16W 1% 16.5K ohm (0603)	1	KOA	RK73H1J1652FT
R36	R/SM 1/16W 1% 215K ohm (0603)	1	Dale	CRCW06032153FT

Table A-1. Intel® IQ80303 Evaluation Platfrom Bill of Materials (Sheet 3 of 4)

Location	Descripition	Qty	Manufacturer	Part Number
R56, R63	R/SM 1/16W 5% 39K ohm (0603)	2		
R55, R60	R/SM 1/16W 5% 4.3K ohm (0603)	2		
R44	R/SM 1/16W 5% 470K ohm (0603)	1		
R45	R/SM 1/16W 1% 4.99K ohm (0603)	1	KOA	RK73H1J4991FT
R77	R/SM 1/16W 5% 56 ohm (0603)	1		
R46	R/SM 1/16W 5% 62K ohm (0603)	1		
R67	R/SM 1/10W 5% 000 ohm (0805)	1		
R73	R/SM 1/10W 5% 100 ohm (0805)	1		
R34	R/SM 1/10W 5% 1K ohm (0805)	1		
R24, R33, R38, R39, R42, R49, R70, R72, R74, R75, R78	R/SM 1/10W 5% 10K ohm (0805)	11		
R30, R35, R47	R/SM 1/10W 1% 100K ohm (0805)	3	KOA	RK73H2A1003FT
R20	R/SM 1/10W 5% 100K ohm (0805)	1		
R69	R/SM 1/10W 5% 1.5K ohm (0805)	1		
R21	R/SM 1/10W 1% 165K ohm (0805)	1	Dale	CRCW08051653FT
R37, R81	R/SM 1/10W 5% 22 ohm (0805)	2		
R82	R/SM 1/10W 5% 24 ohm (0805)	1		
R52, R62	R/SM 1/10W 5% 2.7k ohm (0805)	2		
R26-R29	R/SM 1/10W 5% 330 ohm (0805)	4		
R59, R68	R/SM 1/10W 5% 4.7K ohm (0805)	2		
R3, R22, R23	R/SM 1/10W 5% 8.2K ohm (0805)	3		
R50, R66, R71	R/SM 1/8W 5% 10 Ohm chip 1206	3		
R58, R79, R80	Resistor/SM 1/2W 5% 100 ohm	3	Beckman	BCR 1/2 101 JT
			Dale	CRCW 2010 101 J
			SEI	RMC 1/2 100 ohm 5%
R53	Res/SM 1/2W 1% 0.5ohm (2010)	1	Dale	WSL-2010 0.5 ohm
			IRC	LR2010,1%,0.5 ohm
R54, R57, R61	Resistor Pk SM RNC4R8P 2.7kohm	3	CTS	742083272JTR
R31, R32, R43	Resistor Pk SM RNC4R8P 1.5Kohm	3	CTS	742083152JTR
R65	Resistor Pk SM RNC4R8P 30 ohm	1	CTS	742083300JTR
R1, R2, R4, R5	Resistor Pk SM RNC4R8P 330 ohm	4	KOA	CN1J4T3300J
R51	Resistor Pk SM RNC4R8P 24ohm	1	CTS	742083240JTR
R6-R19	Resistor Pk SM RNC4R8P 8.2Kohm	14	TBD	TBD
J9-J13	CONN SM/TH Mictor 38P Recptcl	5	AMP	767054-1

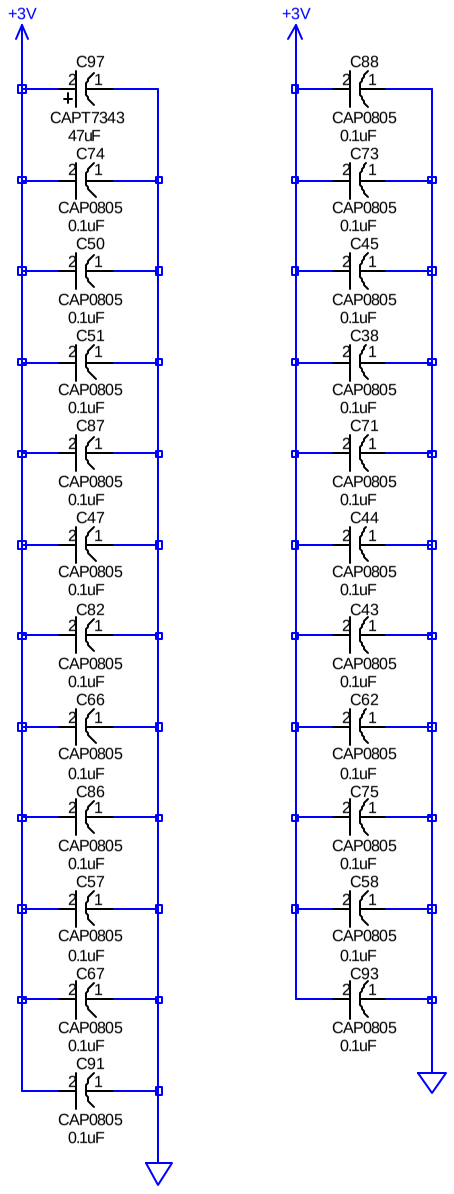
Table A-1. Intel® IQ80303 Evaluation Platfrom Bill of Materials (Sheet 4 of 4)

Location	Descripition	Qty	Manufacturer	Part Number
J1, J2	CONN PCI 64BIT 3.3V/PCB TH	2	AMP	145168-4
			Framatome	CEE2X92SC-V33Z14 W
J5	CONN DIMM 168P/RAng/Socket/TH	1	Berg	88638-61102
			AMP	1-390171-6
J7	CONN/SM TJ 6/6 (Non-Shielded)	1	Kycon	GM-SMT2-N-66
J14	CONN 3P SM Battery Header RAng	1	Molex	53261-0390
J6	CONN Hdr 16 pin/w shell, pcb	1	AMP	103308-3
Z1, Z3	Jumper JUMP2X1	2	Molex	22-28-4023
Z2	Jumper JUMP2X2	1	Molex	10-89-6044
J15	Jumper JUMP2X4 (Right Angle)	1	Samtec	BCS-104-LDHE
J8	Jumper JUMP2X8	1	Molex	10-89-6164
L1	Inductor/SM 68uh 20%	1	Coiltronics	UP1B-680
S1, S2	Switch/SM DIP4 DHS-4S	2	Morse	DHS-4S
U20	OSC/SM 1.8432 MHz (3.3V)	1	MMD	MI3100HH-1.8432MHz
			Ecliptek	EC2600TS-1.8432M
			Connor-Winfield	HSM933 1.8432MHz
CR2, CR4, CR5	LED Green (.125" height)	3	Hewlett Packard	HLMP-3507-D00B2
CR3	LED-Red	1	Hewlett Packard	HLMP3301$010
CR1	LED - Dual 7 Segment Display	1	Hewlett Packard	HDSP-G211
CR6	Diode/SM Schottky 3A/30V	1	Motorola	MBRD330
CR8, CR9	Diode/SM Schottky 1-2A/10V	2	Motorola	MBRS130LT3
Q8	Trnistr/SM General(NPN)(SOT23)	1	Motorola	MMBT3904LT1
			Fairchild	MMBT3904
Q1-Q4	Trnistr/SM General(PNP)(SOT23)	4	Motorola	MMBT 3906LT1
			Fairchild Semi.	MMBT 3906
			Central Semi.	CMPT 3906

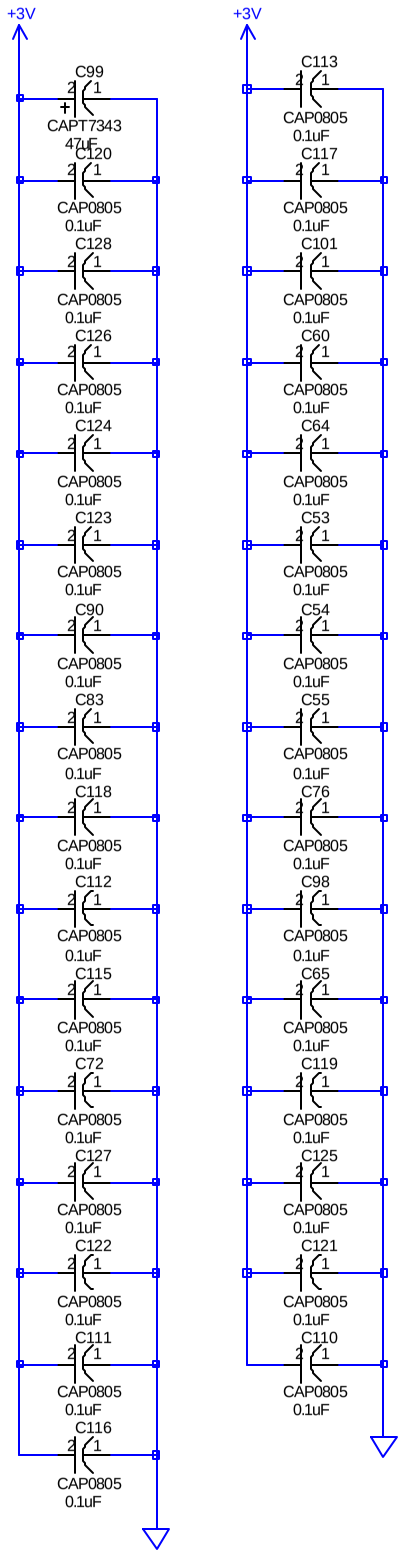
Schematics B

Figure B-1 through Figure B-13 are schematics for the Intel® IQ80303 Evaluation Platfrom.

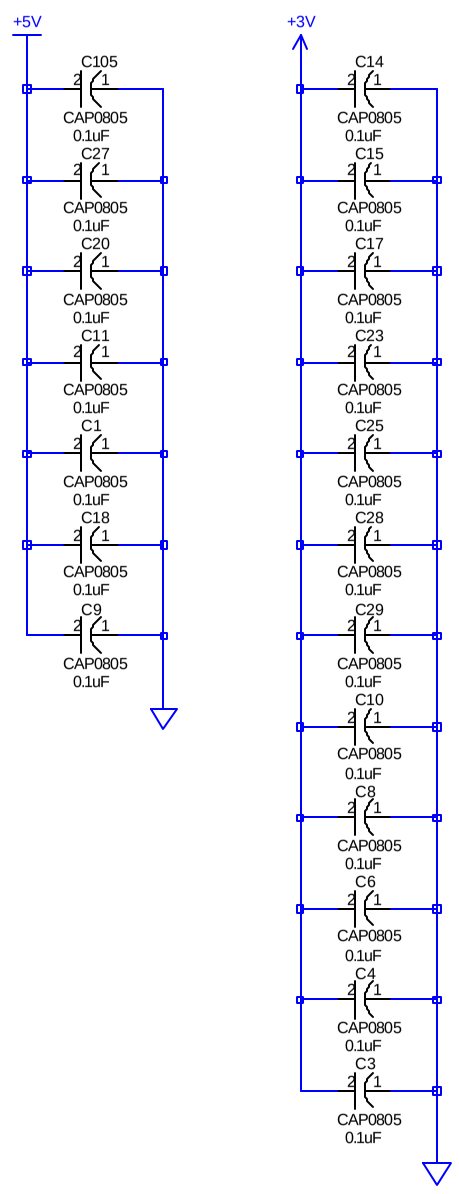
IC DECOUPLING



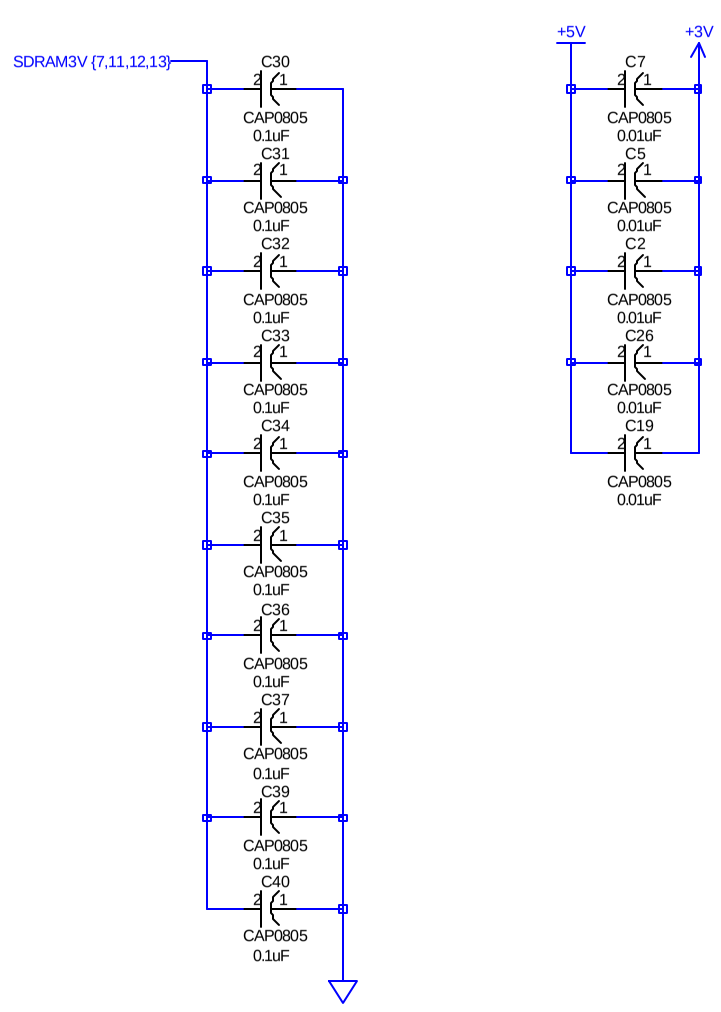
GC80303 DECOUPLING



SPCI DECOUPLING



SDRAM DECOUPLING



DECOUPLING

PRIMARY PCI INTERFACE

SPARES

U18 74LVC08

R78 10K 1/10W 5%

RST# (13)

JUMP1X2

Z3

PRST#

R68 4.7K 1/10W 5%

+3V

CONN_PCI_A

CONN_PCI_B

C95 47uF CAPT7343

C94 0.1uF CAP0805

+12V

C96 0.1uF CAP0805

N12V

-12V

TD

TDI TDO TMS TCK TRST#

+5V1 +5V2 +5V3 +3V1 +3V2 +3V3 +3V4

PCLK

PGNT# PREQ# PPAR PDSEL PSTOP# PTRDY# PIRDY# PDEVSEL# PFRAME# PC/BE7# PC/BE6# PC/BE5# PC/BE4# PC/BE3# PC/BE2# PC/BE1# PC/BE0#

PAD0 PAD1 PAD2 PAD3 PAD4 PAD5 PAD6 PAD7 PAD8 PAD9 PAD10 PAD11 PAD12 PAD13 PAD14 PAD15 PAD16 PAD17 PAD18 PAD19 PAD20 PAD21 PAD22 PAD23 PAD24 PAD25 PAD26 PAD27 PAD28 PAD29 PAD30 PAD31

PAD32 PAD33 PAD34 PAD35 PAD36 PAD37 PAD38 PAD39 PAD40 PAD41 PAD42 PAD43 PAD44 PAD45 PAD46 PAD47 PAD48 PAD49 PAD50 PAD51 PAD52 PAD53 PAD54 PAD55 PAD56 PAD57 PAD58 PAD59 PAD60 PAD61 PAD62 PAD63

PPCI

U14 GC80303 PRIMARY PCI SIGNALS

P_AD0 P_AD1 P_AD2 P_AD3 P_AD4 P_AD5 P_AD6 P_AD7 P_AD8 P_AD9 P_AD10 P_AD11 P_AD12 P_AD13 P_AD14 P_AD15 P_AD16 P_AD17 P_AD18 P_AD19 P_AD20 P_AD21 P_AD22 P_AD23 P_AD24 P_AD25 P_AD26 P_AD27 P_AD28 P_AD29 P_AD30 P_AD31

P_AD32 P_AD33 P_AD34 P_AD35 P_AD36 P_AD37 P_AD38 P_AD39 P_AD40 P_AD41 P_AD42 P_AD43 P_AD44 P_AD45 P_AD46 P_AD47 P_AD48 P_AD49 P_AD50 P_AD51 P_AD52 P_AD53 P_AD54 P_AD55 P_AD56 P_AD57 P_AD58 P_AD59 P_AD60 P_AD61 P_AD62 P_AD63

P_CLK P_M66EN P_REQ64# P_ACK64# P_PAR64 P_SERR# P_PERR# P_INTD# P_INTC# P_INTB# P_INTA#

P_PCLK P_PGNT# P_PREQ# P_PPAR P_PDSEL P_PSTOP# P_PTRDY# P_PIRDY# P_PDEVSEL# P_PFRAME# P_C/BE7 P_C/BE6 P_C/BE5 P_C/BE4 P_C/BE3 P_C/BE2 P_C/BE1 P_C/BE0

P_IDSEL P_LOCK# P_RST

PLOCK# ZRST# (13)

VREF_P

C106 CAP0805 0.01uF

SWDIP4 S1

S_M66EN (5,8,9) P_M66EN (5)

R79 100 1/2W 5%

V_IO_C

Note: U24 is provisional.

U24 P16C2502

CLK_IN FB_IN CLK_OUT FB_OUT

303_CLK PCLK

R67 0 1/10W 5%

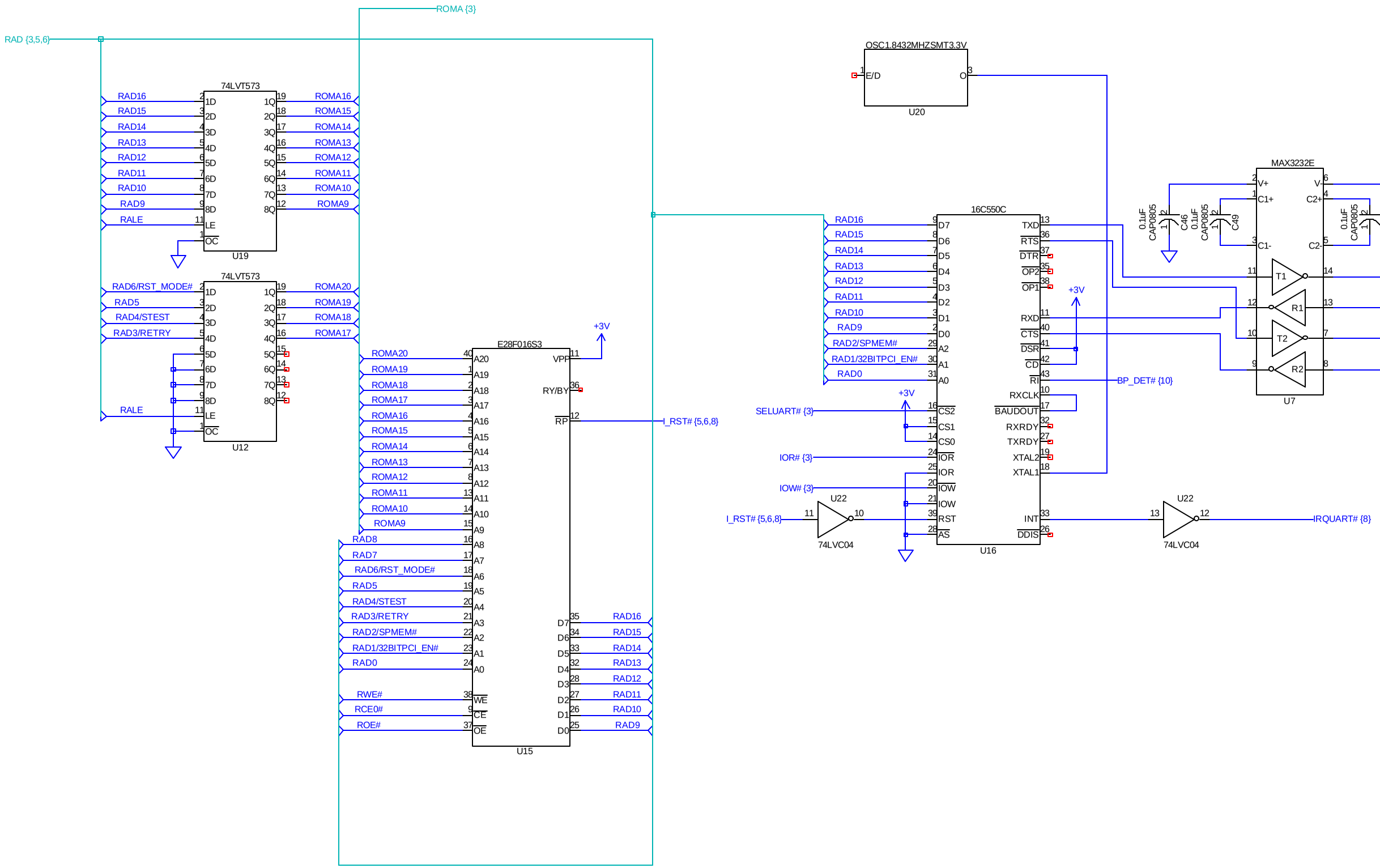
V_IO_F V_IO_E V_IO_D V_IO_C V_IO_B V_IO_A

C109 C108 C107 C100 C104 C103 0.1uF CAP0805

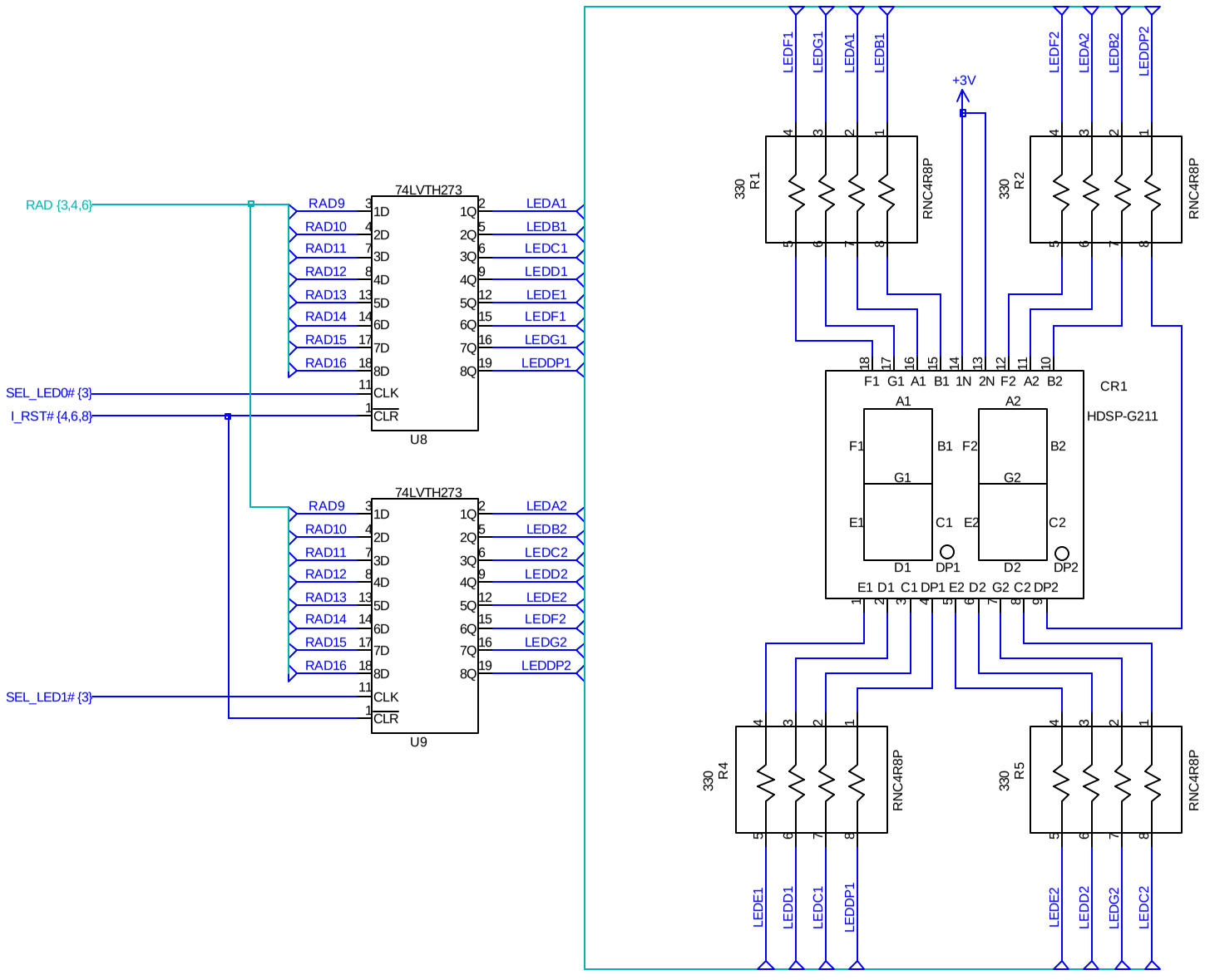
U14
GC80303
MEMORY CONTROLLER

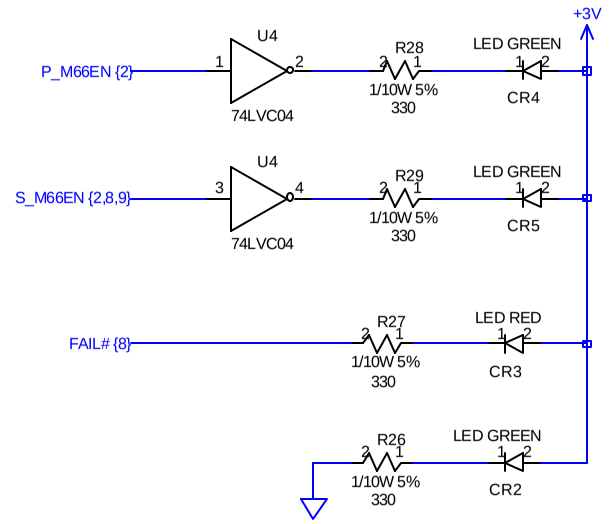
SDRAM {6,7}
RAD {4,5,6}
ROMA {4}
ROMA16
ROMA18
RCE1#
RWE#
SELUART# {4}
IOW# {4}
IOR# {4}
SEL_LED0# {5}
SEL_LED1# {5}
74LVC32
74LVC04
74LVT125
SCKE0 {7,13}
SCKE1 {7,13}
DCLKOUT
DCLKIN
DCLK0 {7}
DCLK1 {7}
DCLK2 {6,7}
DCLK3 {7}
RNC4R8P
SWDIP4
S2
R43 1.5K
RAD6/RST_MODE#
RAD3/RETRY
RAD2/SPMEM#
RAD1/32BITPCI_EN#
RAD4/STEST
RAD0
JUMP1X2
TP2
2.7K 1/10W 5%
BATT_PRES# {11}
BATT_CHRG {11}
BATT_DISCHRG {11}
+3V
SPARES

MEMORY CONTROLLER

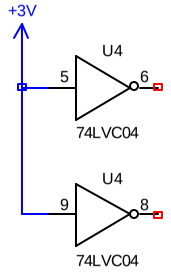


FLASH ROM and UART

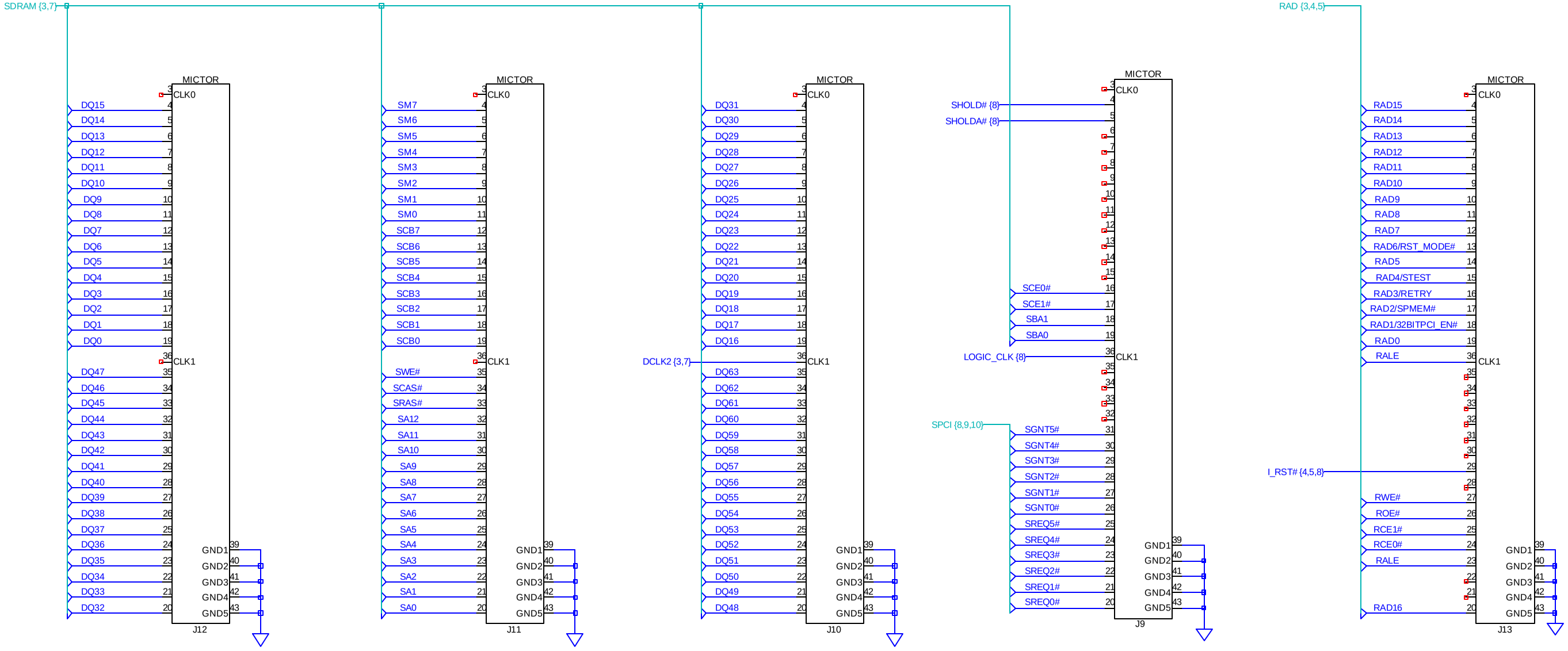

74LVTH273
U8
U9
RAD {3,4,6}
SEL_LED0# {3}
I_RST# {4,6,8}
SEL_LED1# {3}
RAD9
RAD10
RAD11
RAD12
RAD13
RAD14
RAD15
RAD16
1D
2D
3D
4D
5D
6D
7D
8D
CLK
CLR
1Q
2Q
3Q
4Q
5Q
6Q
7Q
8Q
LEDA1
LEDB1
LEDC1
LEDD1
LEDE1
LEDF1
LEDG1
LEDDP1
LEDA2
LEDB2
LEDC2
LEDD2
LEDE2
LEDF2
LEDG2
LEDDP2
330
R1
R2
R4
R5
RNC4R8P
+3V
CR1
HDSP-G211
A1
A2
B1
B2
C1
C2
D1
D2
DP1
DP2
E1
E2
F1
F2
G1
G2
1N
2N

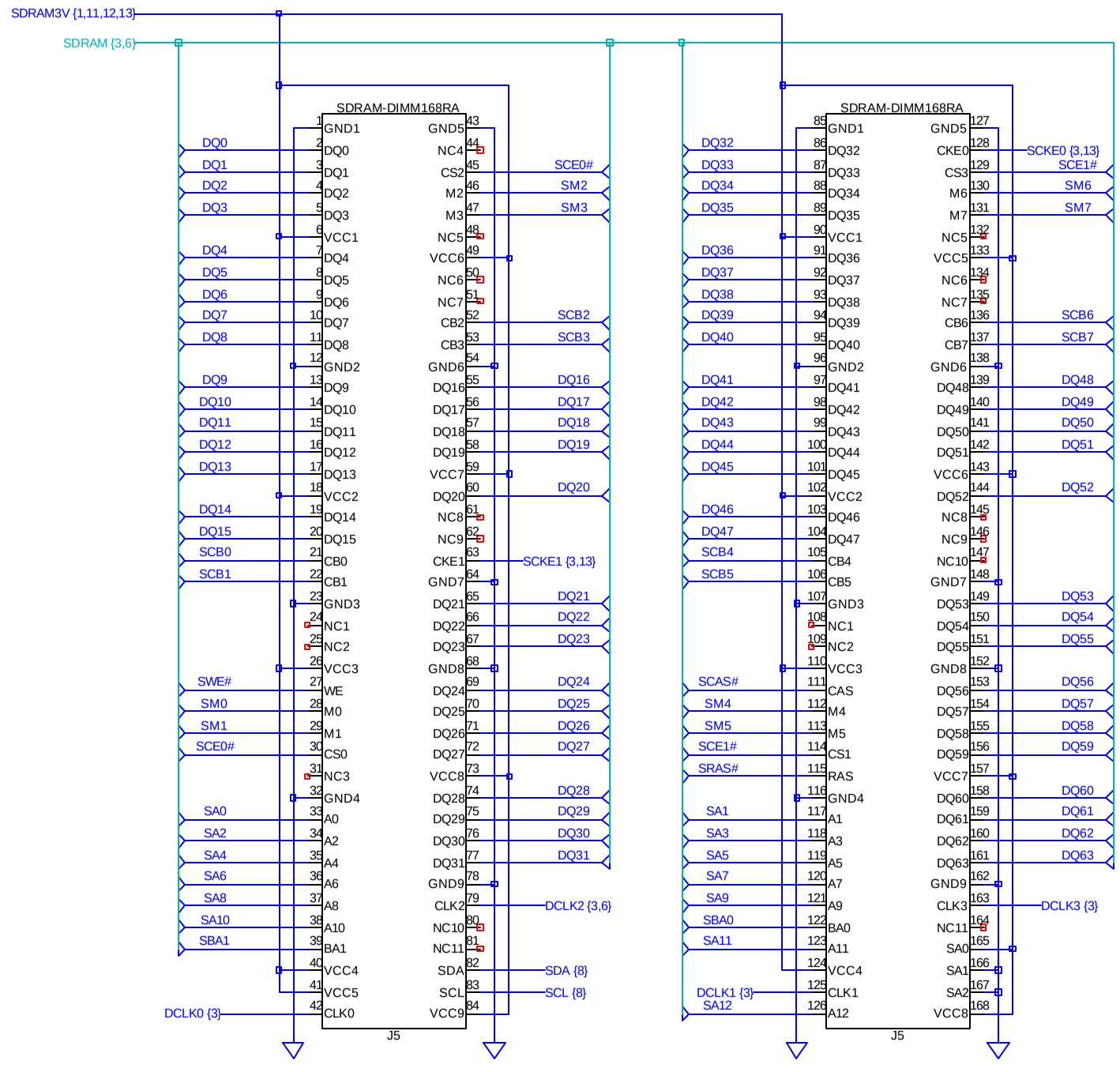

U4
74LVC04
P_M66EN {2}
S_M66EN {2,8,9}
FAIL# {8}
R28
R29
R27
R26
1/10W 5%
330
LED GREEN
LED RED
CR4
CR5
CR3
CR2
+3V

SPARES


+3V
U4
74LVC04
5
6
9
8

LEDs


SDRAM {3,7}
MICTOR
CLK0
DQ15 DQ14 DQ13 DQ12 DQ11 DQ10 DQ9 DQ8 DQ7 DQ6 DQ5 DQ4 DQ3 DQ2 DQ1 DQ0
CLK1
DQ47 DQ46 DQ45 DQ44 DQ43 DQ42 DQ41 DQ40 DQ39 DQ38 DQ37 DQ36 DQ35 DQ34 DQ33 DQ32
GND1 GND2 GND3 GND4 GND5
J12
MICTOR
CLK0
SM7 SM6 SM5 SM4 SM3 SM2 SM1 SM0 SCB7 SCB6 SCB5 SCB4 SCB3 SCB2 SCB1 SCB0
CLK1
SWE# SCAS# SRAS# SA12 SA11 SA10 SA9 SA8 SA7 SA6 SA5 SA4 SA3 SA2 SA1 SA0
GND1 GND2 GND3 GND4 GND5
J11
MICTOR
CLK0
DQ31 DQ30 DQ29 DQ28 DQ27 DQ26 DQ25 DQ24 DQ23 DQ22 DQ21 DQ20 DQ19 DQ18 DQ17 DQ16
DCLK2 {3,7}
CLK1
DQ63 DQ62 DQ61 DQ60 DQ59 DQ58 DQ57 DQ56 DQ55 DQ54 DQ53 DQ52 DQ51 DQ50 DQ49 DQ48
GND1 GND2 GND3 GND4 GND5
J10
SHOLD# {8}
SHOLDA# {8}
MICTOR
CLK0
SCE0# SCE1# SBA1 SBA0
LOGIC_CLK {8}
CLK1
SPCI {8,9,10}
SGNT5# SGNT4# SGNT3# SGNT2# SGNT1# SGNT0# SREQ5# SREQ4# SREQ3# SREQ2# SREQ1# SREQ0#
GND1 GND2 GND3 GND4 GND5
J9
RAD {3,4,5}
MICTOR
CLK0
RAD15 RAD14 RAD13 RAD12 RAD11 RAD10 RAD9 RAD8 RAD7 RAD6/RST_MODE# RAD5 RAD4/STEST RAD3/RETRY RAD2/SPMEM# RAD1/32BITPCI_EN# RAD0 RALE
CLK1
I_RST# {4,5,8}
RWE# ROE# RCE1# RCE0# RALE
RAD16
GND1 GND2 GND3 GND4 GND5
J13
LOGIC ANALYZER I/F



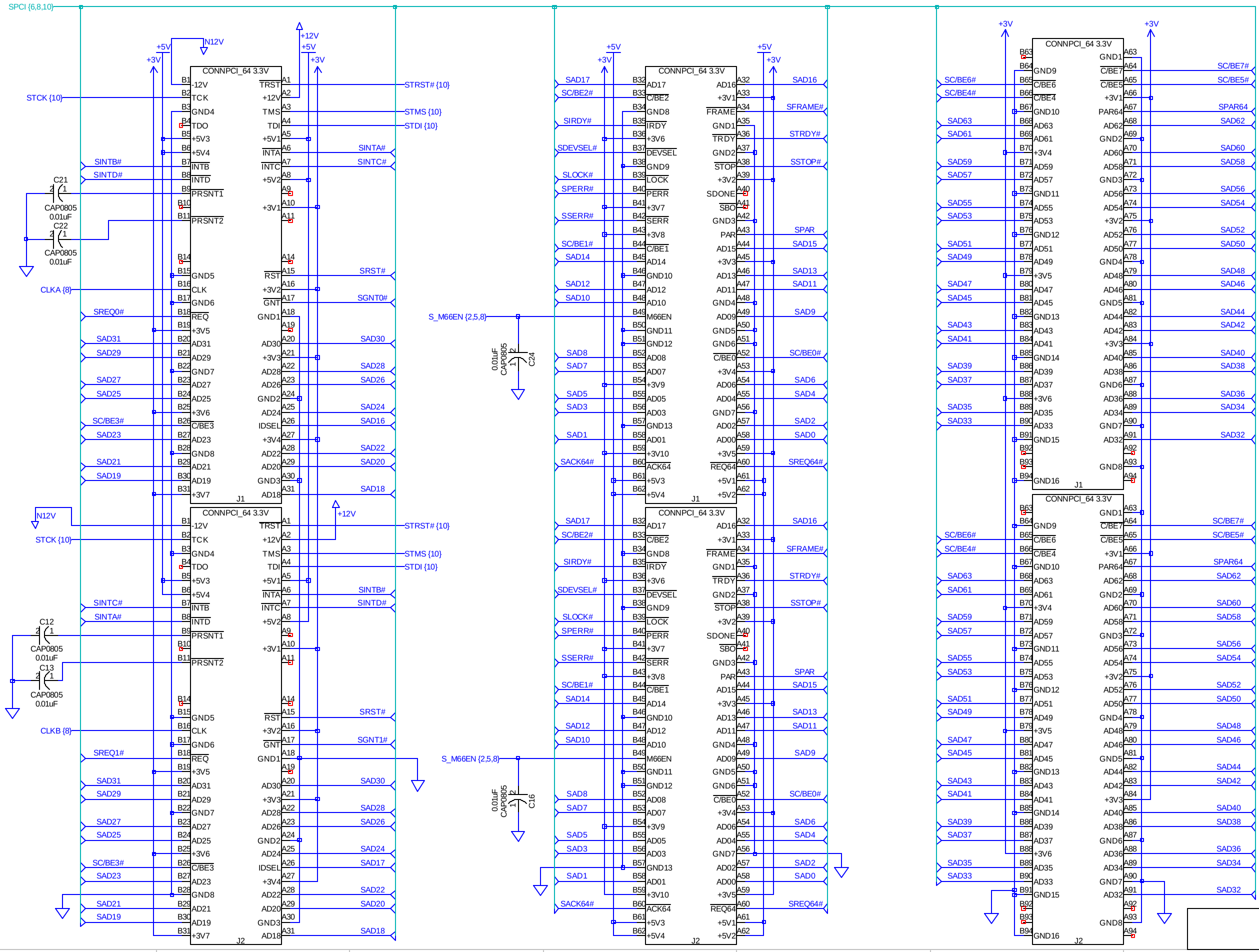
SDRAM 168-PIN DIMM

SPCI {6,9,10}

U14
GC80303
SECONDARY PCI SIGNALS

U14
GC80303
JX CORE/I2C/JTAG

SECONDARY PCI/960 CORE


SPCI {6,8,10}
CONNPCI_64 3.3V
J1
J2
STCK {10}
CLKA {8}
CLKB {8}
STRST# {10}
STMS {10}
STDI {10}
SINTA#
SINTB#
SINTC#
SINTD#
SRST#
SGNT0#
SGNT1#
SREQ0#
SREQ1#
SAD31
SAD29
SAD27
SAD25
SAD23
SAD21
SAD19
SAD30
SAD28
SAD26
SAD24
SAD22
SAD20
SAD18
SC/BE3#
S_M66EN {2,5,8}
SAD17
SC/BE2#
SIRDY#
SDEVSEL#
SLOCK#
SPERR#
SSERR#
SC/BE1#
SAD14
SAD12
SAD10
SAD8
SAD7
SAD5
SAD3
SAD1
SACK64#
SAD16
SFRAME#
STRDY#
SSTOP#
SPAR
SAD15
SAD13
SAD11
SAD9
SC/BE0#
SAD6
SAD4
SAD2
SAD0
SREQ64#
SC/BE6#
SC/BE4#
SAD63
SAD61
SAD59
SAD57
SAD55
SAD53
SAD51
SAD49
SAD47
SAD45
SAD43
SAD41
SAD39
SAD37
SAD35
SAD33
SC/BE7#
SC/BE5#
SPAR64
SAD62
SAD60
SAD58
SAD56
SAD54
SAD52
SAD50
SAD48
SAD46
SAD44
SAD42
SAD40
SAD38
SAD36
SAD34
SAD32
SPCI CONN 1
SPCI CONN 2
SECONDARY PCI BUS 1/2

SPCI {6,8,9}

RNC4R8P
STDI
STMS
STCK
STRST#
R6
8.2K
+3V

RNC4R8P
SDEVSEL#
STRDY#
SIRDY#
SFRAME#
R7
8.2K
+3V

RNC4R8P
SSERR#
SPERR#
SLOCK#
SSTOP#
R8
8.2K

RNC4R8P
SREQ0#
SREQ1#
SREQ2#
SREQ3#
R16
8.2K

RNC4R8P
SACK64#
SREQ64#
R9
8.2K

RNC4R8P
SAD32
SAD33
SAD34
SAD35
R15
8.2K
+3V

RNC4R8P
SAD36
SAD37
SAD38
SAD39
R14
8.2K

RNC4R8P
SAD40
SAD41
SAD42
SAD43
R13
8.2K

RNC4R8P
SAD44
SAD45
SAD46
SAD47
R12
8.2K

RNC4R8P
SAD48
SAD49
SAD50
SAD51
R11
8.2K

RNC4R8P
SAD52
SAD53
SAD54
SAD55
R10
8.2K
+3V

RNC4R8P
SAD56
SAD57
SAD58
SAD59
R19
8.2K

RNC4R8P
SAD60
SAD61
SAD62
SAD63
R18
8.2K

RNC4R8P
SC/BE4#
SC/BE5#
SC/BE6#
SC/BE7#
R17
8.2K

SPAR64
R3
1/10W 5%
8.2K

SREQ4#
R22
1/10W 5%
8.2K

SREQ5#
R23
1/10W 5%
8.2K

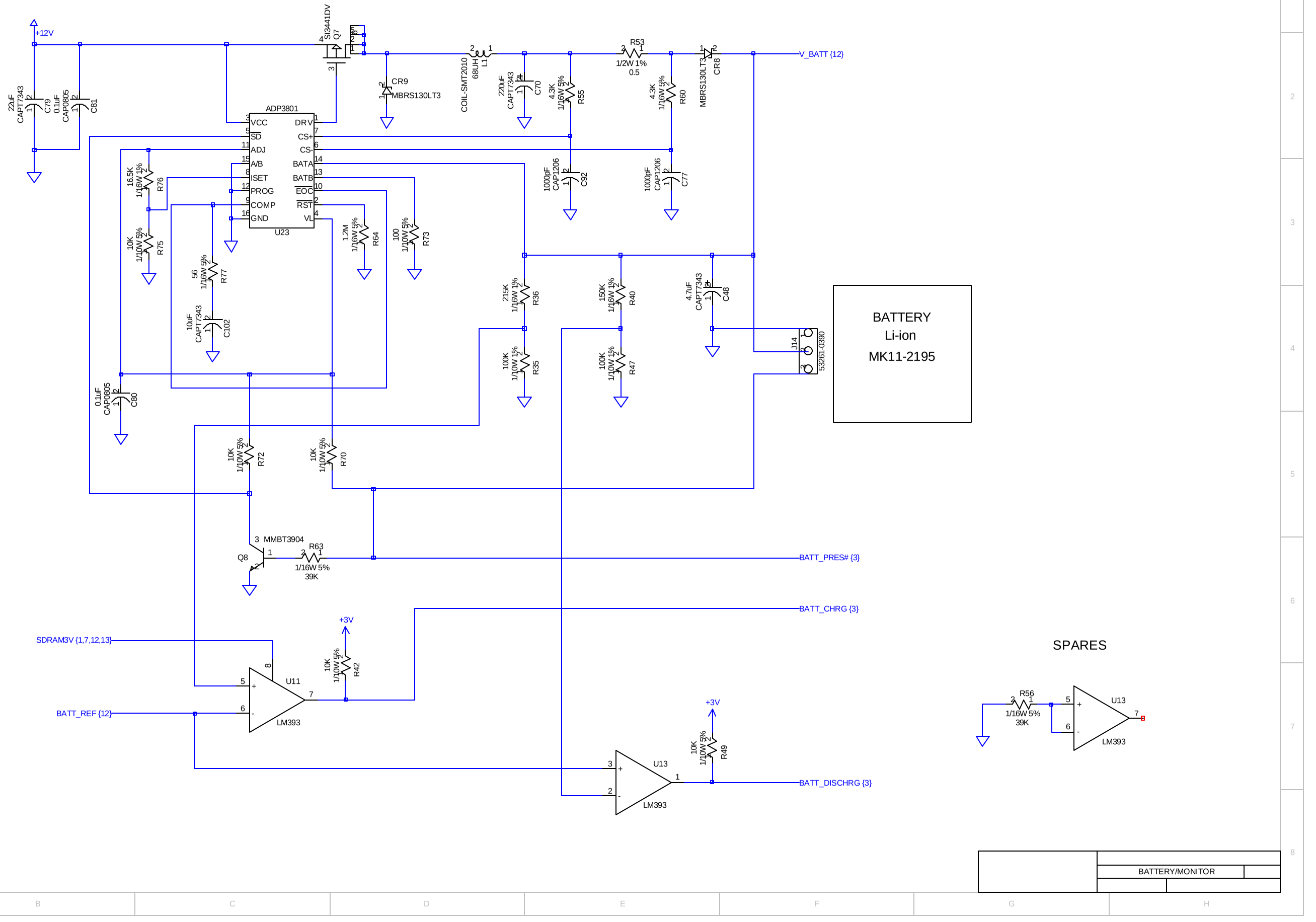
+3V
10K
1/10W 5%
R74

BP_DET# {4}

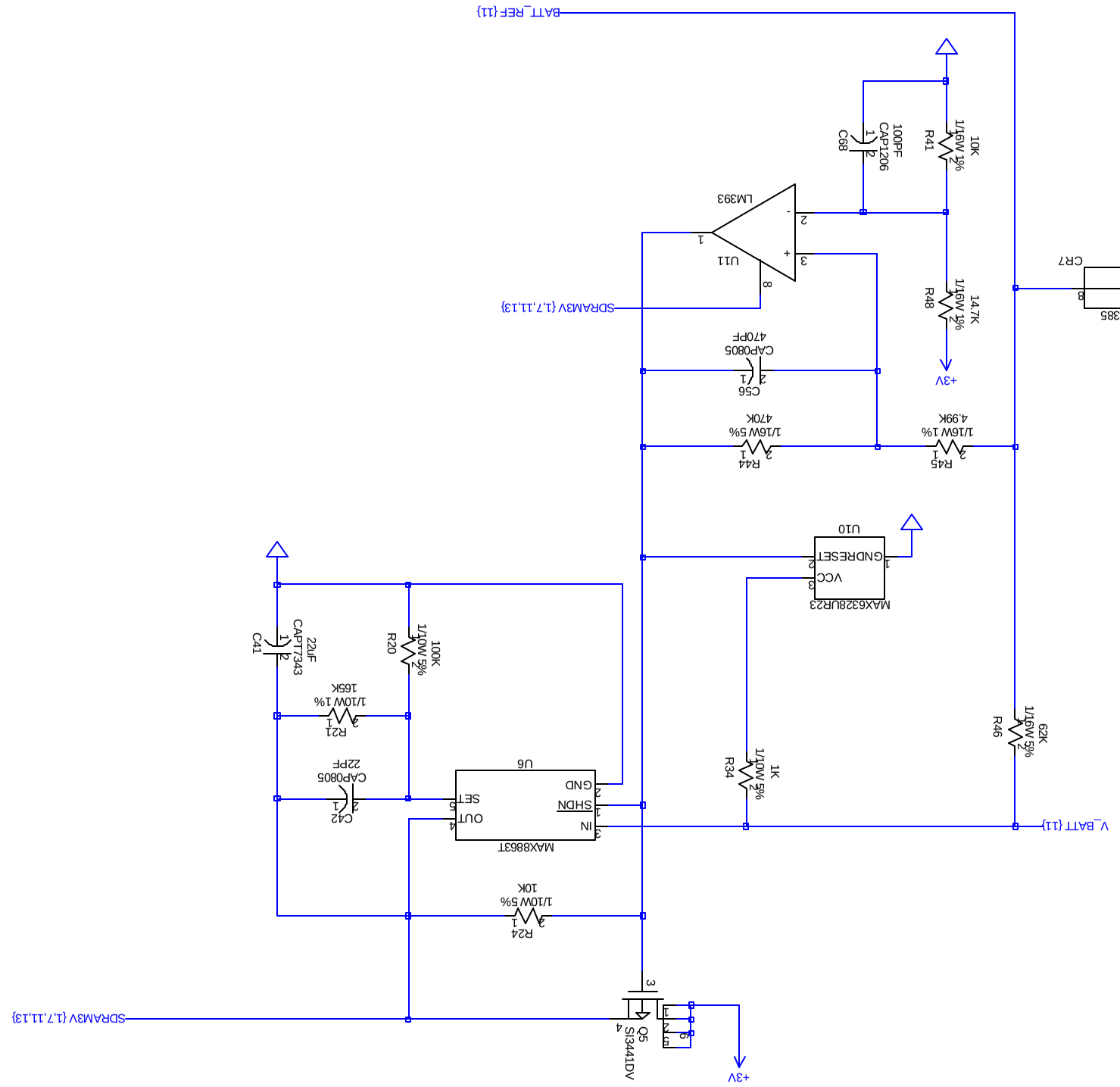
RIGHT2X4
J15

SPCI {6,8,9}
SINTA#
SINTB#
SINTC#
SINTD#

SPCI PULL-UPS


+12V
22uF CAPT7343 C79
0.1uF CAP0805 C81
Si3441DV Q7
CR9 MBRS130LT3
COIL-SMT2010 68UH L1
220uF CAPT7343 C70
4.3K 1/16W 5% R55
R53 1/2W 1% 0.5
4.3K 1/16W 5% R60
MBRS130LT3 CR8
V_BATT {12}
ADP3801
VCC
SD
ADJ
A/B
ISET
PROG
COMP
GND
DRV
CS+
CS-
BATA
BATB
EOC
RST
VL
U23
16.5K 1/16W 1% R76
10K 1/10W 5% R75
56 1/16W 5% R77
10uF CAPT7343 C102
1.2M 1/16W 5% R64
100 1/10W 5% R73
1000pF CAP1206 C92
1000pF CAP1206 C77
215K 1/16W 1% R36
150K 1/16W 1% R40
4.7uF CAPT7343 C48
100K 1/10W 1% R35
100K 1/10W 1% R47
J14 53261-0390
BATTERY
Li-ion
MK11-2195
0.1uF CAP0805 C80
10K 1/10W 5% R72
10K 1/10W 5% R70
Q8 MMBT3904
R63 1/16W 5% 39K
BATT_PRES# {3}
BATT_CHRG {3}
+3V
SDRAM3V {1,7,12,13}
10K 1/10W 5% R42
U11
LM393
BATT_REF {12}
+3V
10K 1/10W 5% R49
U13
LM393
BATT_DISCHRG {3}
SPARES
R56 1/16W 5% 39K
U13
LM393
BATTERY/MONITOR

SDRAM +3.3V REFERENCE



SCKE CONTROL

SPARES