Intel® IQ80315 I/O Processor Evaluation Platform

Board Manual

August 2004

Order Number: 253794-003US

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility 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 Intel® IQ80315 I/O Processor Evaluation Platform 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.

Contents

Figures

Tables

Revision History

Date	Revision	Description
August 2004	003	Minor corrections to Figure 1, “Intel® IQ80315 Customer Reference Board Functional Block Diagram” on page 12.
August 2004	002	Added Appendix B, “Getting Started and Using the Debugger.”
September 2003	001	Initial document.

THIS PAGE INTENTIONALLY LEFT BLANK

1.0 Introduction

This board manual describes in detail how the Intel® IQ80315 Customer Reference Board (CRB) is implemented, explains the CRB capabilities, and provides instructions on how to configure the CRB.

1.1 Intended Audience

This board manual is intended to provide detailed technical information about the Intel® IQ80315 Customer Reference Board (hereafter "IQ80315 CRB" or "IQ80315") to software developers, validation engineers, and other engineers and technicians who need this level of information.

Table 1. Terms and Definitions

Term	Definition
#	Used after a signal name to identify an active-low signal (such as USBP0#)
GB	Gigabyte (1,073,741,824 bytes)
GB/sec	Gigabytes per second
KB	Kilobyte (1024 bytes)
Kbit	Kilobit (1024 bits)
Kbits/sec	1000 bits per second
MB	Megabyte (1,048,576 bytes)
MB/sec	Megabytes per second
Mbit	Megabit (1,048,576 bits)
Mbit/sec	Megabits per second
0x	An address or data value beginning with "0x" indicates a hexadecimal value.
V	Volts. Voltages are DC unless otherwise specified.

2.0 Hardware

2.1 Overview

2.1.1 Feature Summary

Table 2 summarizes the major features of the IQ80315 CRB.

Table 2. Feature Summary (Sheet 1 of 2)

Feature	Details
Form Factor	Micro ATX—9.60 inches by 9.60 inches
Processor	Dual Intel® 80200 processors with Intel XScale® microarchitecture: • 333 MHz to 733 MHz CPUs supported • 100 MHz bus frequency
Memory	Double Data Rate (DDR) SDRAM: • Three registered 184-pin DDR200 SDRAM DIMM sockets • Single- or double-sided registered DIMM modules supported • Up to 12 GB (4 GB/module) supported
	SDRAM Battery Back-up: • Battery back-up charge/control circuitry • Four “AA” 1700 mAh NiMH batteries onboard — Approximately 15.4 hour backup time with 1 GB SDRAM — Approximately 2.0 hour backup time with 12 GB SDRAM
	Flash Memory: • 8 Mbytes Intel StrataFlash® • 8-bit interface
	Non-Volatile RAM (NVRAM): • 44 bytes of non-volatile RAM integrated into the RTC • 2 Kbytes in a serial I^{2}C EEPROM
Chipset	Intel® I/O processor 80315 chipset

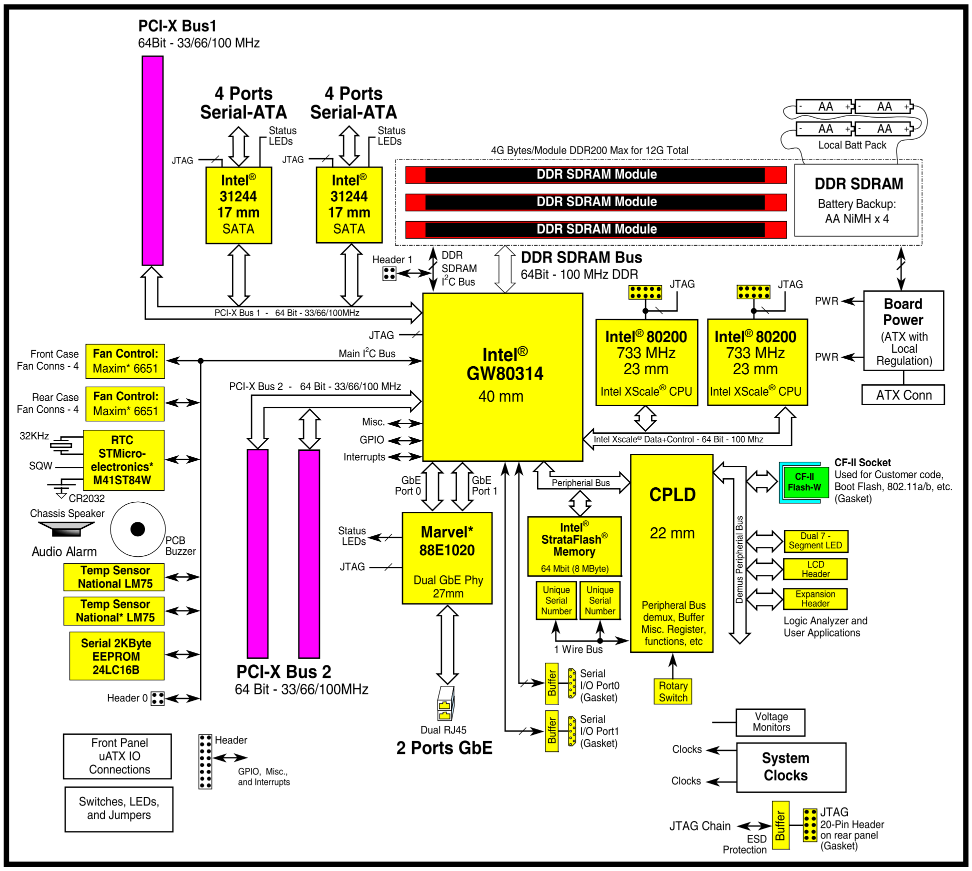
Table 2. Feature Summary (Sheet 2 of 2)

Feature	Details
I/O Ports	Dual Serial Ports: • Stacked DB9s mounted in the rear gasket area
	JTAG Ports: • Six JTAG headers (one main/CPU 0 JTAG header and a second dedicated header for CPU 1) • The Intel® 80200 processor JTAG can be chained together on the main JTAG header or use a jumper option for independent operation. • The Intel® 80314 I/O processor companion chip JTAG can be operated independently or chained with the remaining peripheral devices. • Main JTAG header is accessible from the rear panel gasket. • CPLD is programmed through the dedicated JTAG port.
	Dual I^2C Ports: • Four-pin header for access to main I^2C bus • Four-pin header for access to the DDR serial presence detect I^2C bus
	Serial ATA: • Eight serial ATA 1.0 standard copper connect ports
	CompactFlash* II: • One CompactFlash type-II socket mounted in the rear panel gasket
	Gigabit Ethernet: • Two RJ45 Ethernet ports with integrated LEDs and magnetics • Mounted in the rear panel gasket
	Fan-Control Headers: • Eight fan-control headers (up to four front fans and four rear fans)
	ATX standard front panel header
Expansion Connectors	PCI-X Connectors: • Bus A—Two 64-bit slots—100/66 MHz • Bus B—One 64-bit slot—100/66 MHz
	Peripheral Bus Header: • PC104-like 0.1” header for logic analyzer probing or peripheral bus prototyping
Misc. Functions	Real-time clock with replaceable CR2032 battery
	Dual temperature sensors
	Electronic serial numbers—two unique 48-bit serial numbers
	Supply voltage monitor for under-voltage detection
	CPLD: • Misc. logic collection • Misc. registers (Revision ID, battery status, rotary switch, and so on)
Optical	• Two 7-Segment LED displays for debug • Power-on LED (green) indicators for each power plane • Hard-drive activity LED indicators (blue) for each hard drive • Composite hard-drive activity LED (green) • Header for external hard-drive activity LEDs • Header for a two-line, 40-character LCD display.
Audible Alarm	PCB mounted buzzer and connector for chassis mounted speaker

2.1.2 Block Diagram

Figure 1 is a block diagram of the major functional areas of the IQ80315 CRB.

Figure 1. Intel® IQ80315 Customer Reference Board Functional Block Diagram



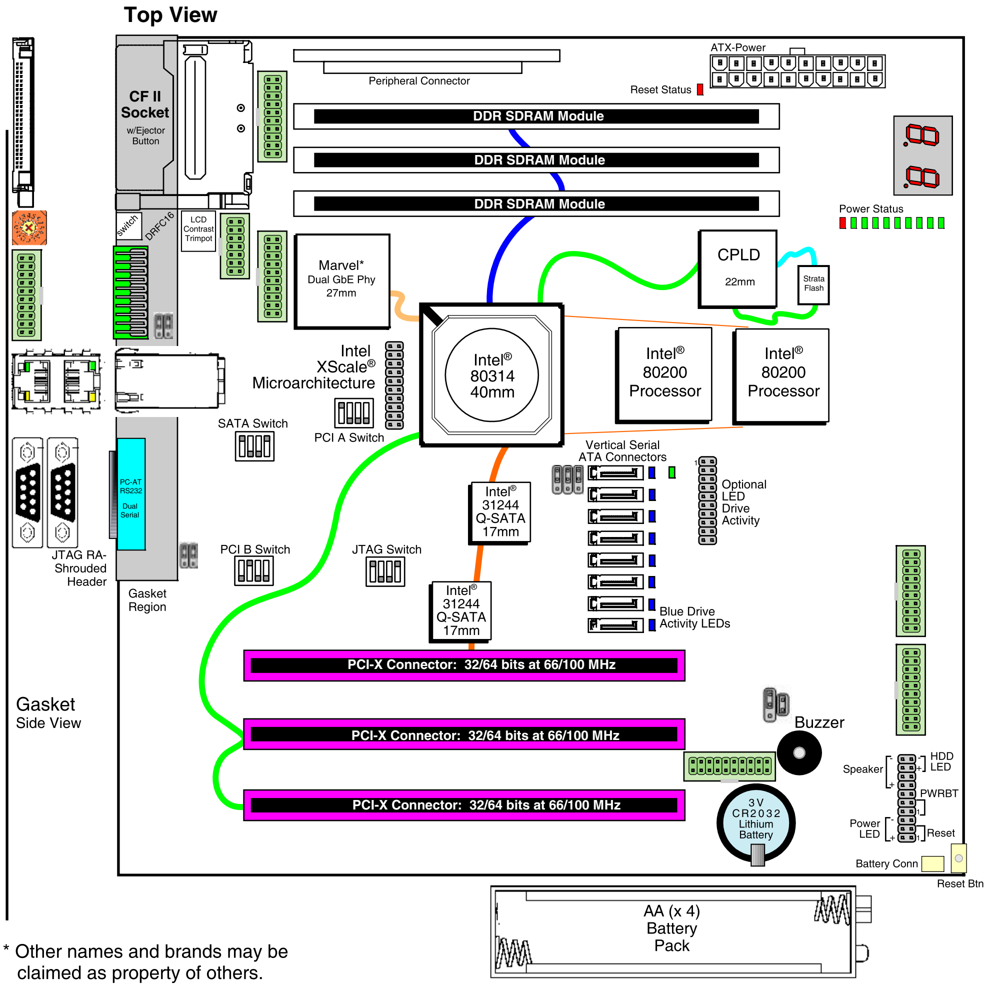
* Other names and brands may be claimed as property of others.

B3665-02

2.1.3 Layout Diagram

The CRB is implemented in a micro-ATX form factor with standard PC-type I/O. This implementation provides an easy development environment and facilitates rapid design development by allowing large portions of circuitry to be copied into customer designs.

Figure 2. Intel® IQ80315 Customer Reference Board Layout Diagram



* Other names and brands may be claimed as property of others.

B3446-01

2.2 Processor

The IQ80315 CRB has two Intel® 80200 processors with Intel XScale® microarchitecture. The key characteristics of the 80200 processor implementation on the IQ80315 CRB are as follows.

- 32-bit address, 64-bit data busses
- Core speed: 333–733 MHz
- Bus speed: 100 MHz
- Core voltage: 1.5 V ± 5%
- Bus voltage: 3.3 V ± 0.3 V
- Inter-CPU communication: The Intel® 80314 I/O processor companion chip doorbell registers are used for the CPUs to interrupt each other for messaging.

See Section 2.14.1 for a description of the CPU reset strapping options on the IQ80315 CRB.

2.3 System Memory

The IQ80315 CRB provides the following memory features:

- Three DIMM sockets supporting 2.5 V 184-pin 200 MHz registered Double Data Rate (DDR) SDRAM DIMMs
- ECC or non-ECC DIMMs supported
- Single-sided or double-sided DIMMs supported
- 12 GB maximum total system memory (with future 4 GB DIMM modules)
- Dedicated I^2C bus for SDRAM serial presence detection
- Battery backup for DIMM data retention

See Section 2.12.5 for a description of the DRAM battery back-up solution.

2.4 Intel® 80314 I/O Processor Companion Chip

The Intel® 80314 I/O processor companion chip (hereafter “80314”) consists of the following interfaces:

- 100 MHz 80200 bus interface
- Integrated DDR SDRAM controller
- Two integrated Gigabit Ethernet MAC controllers
- Two integrated PCI-X interfaces
- 33 MHz peripheral bus interface
- Two integrated 16550-compatible UARTs
- Two I^2C controllers
- Interrupt controller

See Section 2.14.1 for a description of the 80314 configuration strapping options.

2.5 PCI-X Subsystem

The IQ80315 CRB has two independent PCI-X busses. Both busses have 64-bit data paths, can run at 33, 66 or 100 MHz, and can support either PCI or PCI-X cards. PCI-X bus 1 has two Intel® 31244 Serial ATA controllers and one PCI-X slot. PCI-X bus 2 has two PCI-X slots.

Each PCI-X bus can be independently designated as a primary or secondary PCI-X bus through reset strapping options. See Section 2.14.1 for a description of the IQ80315 CRB reset strapping options.

The Intel® 80314 I/O processor companion chip is the PCI central function. Therefore, the 80314 PCI interfaces are run in embedded mode. See Section 3.3 and Section 3.4 for the PCI request/grant and IDSEL signal assignments.

The IQ80315 CRB supports PCI-X power management by connecting the PME# signal to an 80314 interrupt input. See Section 3.5 for the IQ80315 CRB interrupt mapping.

The IQ80315 CRB does not support PCI hot-swapping.

2.5.1 Serial ATA Subsystem

The IQ80315 CRB has two Intel® 31244 serial ATA controllers supporting up to eight serial ATA drives. DIP switches enable DPA mode and spread-spectrum clocking options on the 31244 SATA controllers.

The serial ATA drive connectors are Molex* 67491-003x vertical connectors.

The board has nine hard-drive activity LEDs. There are eight blue LEDs, one for each hard drive, and one aggregated green LED. One blue LED is connected to each 31244 LED output and is illuminated whenever the associated hard drive is being accessed. The green LED illuminates whenever any of the hard drives is being accessed.

The hard-drive activity LED signals are also routed to a 20-pin, 2 × 10, 0.1" center header. This allows cabling the hard-drive LED signals off-board to a chassis mount panel. These LED signals are 3.3 V and can each sink 20 mA. See Section 2.13.3 for the hard-drive LED header pinout.

2.6 Gigabit Ethernet

The Intel® 80314 I/O processor companion chip has two integrated 10/100/1000 Mbit/sec Ethernet MAC controllers. The IQ80315 CRB uses a Marvell* 88E1020 transceiver chip to implement the Ethernet physical interface. Shorting the 88E1020 Config[4:0] inputs to V_{CC}, GND, or one of the LED outputs configures the following options at reset:

- GMII interface to the 80314
- Ethernet controller 0 PHY address = 00000
- Ethernet controller 1 PHY address = 00001
- Internal 1.5 V regulators are enabled
- MDI crossover is enabled
- 125 MHz clock output is enabled
- Auto-negotiation is enabled
- A jumper option selects “Prefer Master” or “Prefer Slave” during auto negotiation (default jumper setting is “Prefer Master”)

A dual vertical RJ-45 LAN connector with integrated LEDs and magnetics is used for the LAN connectors.

One green and one amber/green LED is built into each RJ-45 LAN connector. Table 3 describes the LED states when the board is powered up and the Ethernet subsystem is operating.

Table 3. Ethernet Connector LED States

LED	LED Color	LED State	Condition
Speed (right)	–	Off	10 Mbit/sec data rate is selected.
	Green	On	100 Mbit/sec data rate is selected.
	Amber	On	1000 Mbit/sec data rate is selected.
Link / Active (left)	–	Off	LAN link is not established.
	Green	On (steady state)	LAN link is established.
	Green	Blinking	The system is transmitting or receiving data.

Note: To obtain the LED behavior described in Table 3, connect the LED_LINK1000 and LED_LINK100 signals to the Speed LEDs, and the LED_TX signal to the Link/Active LED. Program the Marvell* 88E1020 transceiver LED Link Control Register 24[3] = 0, and LED Control Register 24[2:0] = 1 (Mode 2).

2.7 Peripheral Bus

The Intel® 80314 I/O processor companion chip supports an asynchronous general-purpose peripheral bus for devices such as flash, ROM, or SRAM memory. The IQ80315 CRB runs the internal peripheral interface clock at the 80314 core clock frequency.

The CPLD buffers the peripheral bus data bus and demultiplexes the address bus. The CompactFlash* and expansion header have 16-bit data busses and are accessed on the PBI_AD[31:16] lines. The flash, seven-segment LEDs and LCD header have 8-bit data busses and are accessed on the PBI_AD[31:24] lines. The rotary switch is 4-bits and returns data on the PBI_AD[27:24] lines when read.

2.7.1 CPLD

A Xilinx* XC95144XL CPLD is used to implement glue logic needed on the IQ80315 CRB. The CPLD functions are listed below:

- Seven-segment LED decoder and latch
- Reset strapping outputs
- Battery status register
- Product code register
- Board stepping register
- CPLD firmware revision register
- Additional chip enable generation
- CompactFlash* I/O control signals
- PIO for reading 1-wire electronic serial number device

Section 3.2 shows how the CPLD registers, chip enables, and GPIO port map into system address space.

Table 4 defines the battery status register bits.

Table 4. BAT_STAT Register Definition

Bit	Name	Definitions
0	Battery present	0 = No DDR backup battery 1 = DDR backup battery is present
1	Reserved	Undefined
2	Battery enable	0 = Disable DDR backup battery 1 = Enable DDR backup battery
3	Reserved	Undefined
4–7	Reserved	Undefined

Table 5 defines the product code register bits.

Table 5. PROD_CODE Register Definition

Bit	Name	Definition
3–0	Product SKU	Identifies variants within a family of related products. This is a resistor strapping input to the CPLD. The IQ80315 product SKU definition is TBD.
7–4	Product code	Identifies the product. This value is hard-coded in the CPLD firmware. 0x1 = IQ80315 CRB

Table 6 defines the board stepping register bits.

Table 6. BOARD_STEPPING Register Definition

Bit	Name	Definition
3–0	Board rework level	Identifies rework completed on the board. This is a resistor strapping input to the CPLD.
7–4	Board stepping	Identifies the board stepping. This value is a resistor strapping input to the CPLD firmware.

Table 7 defines the CPLD firmware revision register bits.

Table 7. CPLD_FW Register Definition

Bit	Name	Definition
7–0	CPLD code revision	Identifies the CPLD code revision level. This value is coded into the CPLD code.

2.7.2 Flash

Eight MB of flash memory is implemented in the IQ80315 CRB using an Intel StrataFlash® device. The flash chip is packaged in a 64-ball easy BGA package. The IQ80315 CRB makes the following signal connections to the flash chip:

- The Intel® 80314 I/O processor companion chip PCE0# chip-enable output is connected to the flash chip enable input CE# to enable flash access after reset.
- The flash reset input is tied to the power supervisor reset output to prevent spurious flash writes during power-up.
- The program enable (VPEN) signal is tied high to enable flash programming under software control.
- The BYTE# signal is tied low to keep the flash in 8-bit interface mode.

2.7.3 CompactFlash*

The IQ80315 CRB supports a 3.3 V, 16-bit CompactFlash* type II socket on the peripheral bus. The CompactFlash interface is compliant with the CF+ v1.4 CompactFlash specification. The CompactFlash interface of the IQ80315 CRB supports memory and I/O accesses. CompactFlash ATA cards are not supported.

The CompactFlash card interface to the IQ80315 CRB board is through an AVX 50-pin socket mounted in the rear panel gasket area.

The CompactFlash socket is connected to PCE1# and responds to the address ranges identified in Section 3.2. The CompactFlash interface connects to the buffered peripheral bus interface generated by the CPLD to protect the 80314 from ESD. The CPLD also generates the I/O read and I/O write signals for accesses to the upper 512 KB of the CompactFlash address range.

When the system is powered-up and not in reset, inserting a CompactFlash card into the socket generates an interrupt. See Section 3.5 for interrupt mapping.

2.7.4 LCD Support

The IQ80315 CRB has a 14-pin LCD header and 2-pin backlight header to support a character-based LCD module. The IQ80315 CRB supports the following LCD modules:

- Lumex* Opto* LCM-S02404DSF
- Varitronix* MDLS-20265

The CPLD generates the LCD chip enable (LCD_CE#) and register-select (LCD_RS) signals. See Section 3.2 for the LCD address range.

A 10 KΩ trimpot mounted in the gasket area is used to adjust the LCD contrast voltage (Vo) between 0 V and 5 V.

Connecting the +5 V power plane through a diode to the backlight connector generates the 4.2 V backlight voltage.

2.7.5 LEDs

The IQ80315 CRB contains power, hard-drive, and seven-segment LEDs to communicate current system status.

There is a green LED on the board for each major power plane: +12 V, +5 V, and +3 V. When power is applied to these power planes, the associated LED is lit.

The board has nine hard-drive activity LEDs. There are eight blue LEDs (one for each hard drive), and one aggregated green LED. One blue LED is connected to each Intel® 31244 LED output and is illuminated whenever the associated hard drive is being accessed. The green LED is illuminated whenever any of the hard drives is being accessed.

The board has a pair of Agilent* HDSP-G211 seven-segment LEDs for displaying POST codes or other software generated debug codes. Both seven-segment LEDs are individually addressed. See Section 3.2 for the seven-segment LED addresses.

2.8 UART

The IQ80315 CRB has two 9-pin D-shell serial port connectors mounted on the back panel. These are driven by a Max3232E* 3.3 V dual serial port transceiver. The Intel® 80314 I/O processor companion chip TX, RX, CTS#, and RTS# signals for each UART are connected to the serial port transceiver. The DSR#, DTR#, DCD#, and RI# signals are muxed with 80314 GPIO signals and are used as GPIO signals on the IQ80315 CRB.

2.9 Electronic Serial Number

The IQ80315 CRB has two DS2401 electronic serial number components (ESN) connected to a GPIO pin on the CPLD. These components are accessed via the Maxim* Integrated Products 1-wire bus protocol. The bus protocol is implemented by “bit-banging” the CPLD GPIO pin with the appropriate timing and listening for the returned serial-number bits. The ESN GPIO pin is set high by writing 0x01 to any address within the ESN address space and is cleared by writing 0x00 to any ESN address. Reading any address within the ESN address space returns the current value of the ESN GPIO pin on data bit[0]. Data bits[7:1] are undefined during ESN read cycles. It requires about 8 mS to read both ESN devices. See Table 16 on page 40 for the ESN address range.

The software routine that reads the ESN devices can store the serial numbers in flash memory to simplify future serial-number accesses. Following this method, the ESN devices only need to be read after the flash memory has been erased.

2.10 I^{2}C

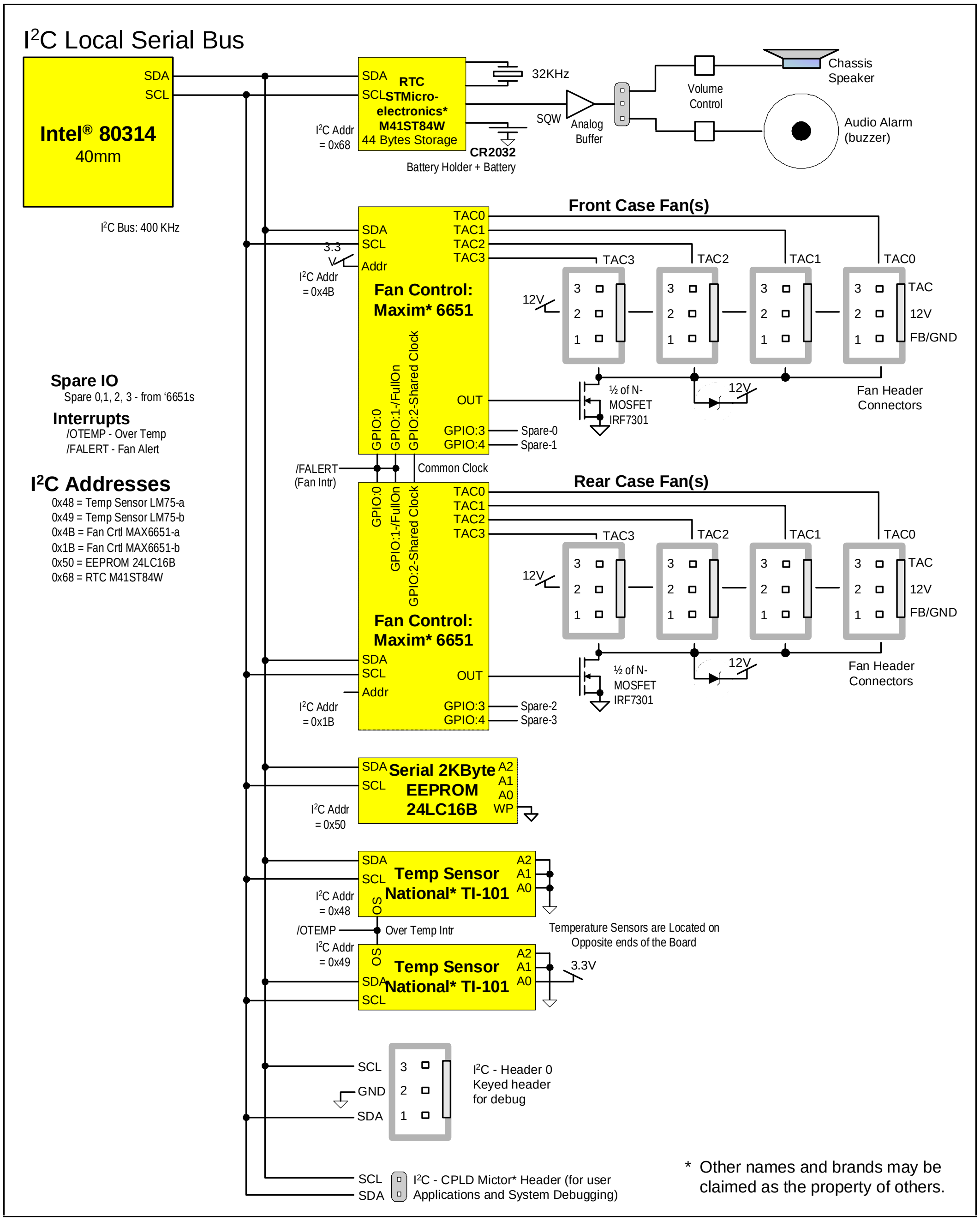
The IQ80315 CRB has two independent I^{2}C serial busses:

- Main I^{2}C bus
- I^{2}C bus dedicated to SDRAM presence detect and configuration

2.10.1 Main I²C Bus Subsystem

The main I²C bus is primarily used for miscellaneous storage and support functions. The IQ80315 CRB uses the main I²C bus for serial EEPROM, real-time clock, temperature sensors, and fan controllers, while operating at 400 KHz. Figure 3 illustrates the main I²C bus components.

Figure 3. I²C Local Serial Bus



2.10.1.1 Real Time Clock—ST Microelectronics* M41ST84W

An ST Microelectronics* M41ST84W Real-Time Clock chip is used to keep time on the IQ80315 CRB. This real time clock (RTC) chip uses an external CR2032 lithium cell battery, has 44 bytes of onboard storage, and provides a square-wave signal output for a buzzer or speaker. The battery life of the CR2032 cell exceeds 10 years.

The RTC square-wave output can be programmed to generate 15 discrete frequencies between 1 Hz and 32 KHz. The square-wave output is enabled or disabled by writing to a register in the RTC. The square-wave output is buffered and directed to either the buzzer or speaker output by a 3-pin jumper.

The IQ80315 CRB supports both a speaker and a buzzer for design flexibility. The buzzer is useful when the board is used without an ATX chassis and the speaker when used with a chassis.

2.10.1.2 Audio Buzzer

An RDI Electronics* DMT-1206-SMT buzzer is mounted on the CRB board. It generates a tone when the RTC square-wave signal is activated and connected to the buzzer. The default condition is off. A DIP switch selects low or high buzzer volume. The buzzer frequency range is approximately 500 Hz to 3 KHz.

2.10.1.3 Chassis Speaker

A standard ATX chassis has a speaker mounted on the front panel. The speaker is connected to the motherboard speaker drive electronics through a cable and a standard speaker connector. See Section 2.11 for a description of the speaker connector and pinout. A DIP switch selects low or high speaker volume.

2.10.1.4 Watchdog Timer

A watchdog timer is built into the M41ST84W real time clock. The watchdog timer is inactive on reset. Software can enable and disable the watchdog timer and configure the time-out period. The watchdog countdown timer is reset by writing to a register in the RTC.

2.10.1.5 Fan Control—Dual Maxim* MAX6651s

Two Maxim* MAX6651 Fan Control ICs (hereafter “6651”) provide control of up to eight chassis fans. Two fan controllers are provided to allow using different fans on the front and rear. Each 6651 fan controller monitors the tach outputs of up to four fans. The 6651 controls fan-speed by monitoring the tach0 fan signal and adjusting the fan control output to maintain the tach0 input at the programmed speed. For this reason, each 6651 must control similar fans.

The fan control connectors are standard PC type 3 pin Molex* connectors. A single MOSFET is used to gate motor power for four fans, and a diode is used to prevent over-voltage conditions.

2.10.1.6 Serial 2 Kbyte EEPROM—MicroChip* 24LC16B

The Serial EEPROM has 2 Kbytes of non-volatile storage and is used to store the 80314 configuration parameters and for general non-volatile storage needs. After reset, the 80314 device reads the serial EEPROM and updates the 80314 configuration registers prior to booting.

2.10.1.7 Temperature Sensors—Dual National* TI-101s

Temperature monitoring is achieved using two National* TI-101 temperature sensors. These are 3.3 V, I^2C sensors with addresses 0x48 and 0x49. Accuracy is nominally ± 2 °C in the IQ80315 CRBs operating range.

The TI-101s are placed on opposite ends of the board, just in case there is a strong temperature gradient present. This layout also provides for some redundancy in case one sensor is blocked off. The temperature alert signal from both TI-101s (O.S. out) are combined into a single interrupt into the 80314. See Section 3.5 for interrupt assignments.

2.10.1.8 I^2C Bus Test Headers

The I^2C bus test headers are generic expansion connectors for test probing and off–board peripherals development. The connectors are keyed 4-pin headers that are different from the fan connectors. Both of the I^2C busses on the IQ80315 CRB are brought out to dedicated 4-pin test headers.

2.10.1.9 I^2C Bus on the Peripheral Bus Expansion Header

The CPLD has a 0.1" PC104 style connector attached to it for peripheral bus test and user expansion. The main I^2C bus is also brought to one of these headers to simplify peripheral bus expansion module prototyping.

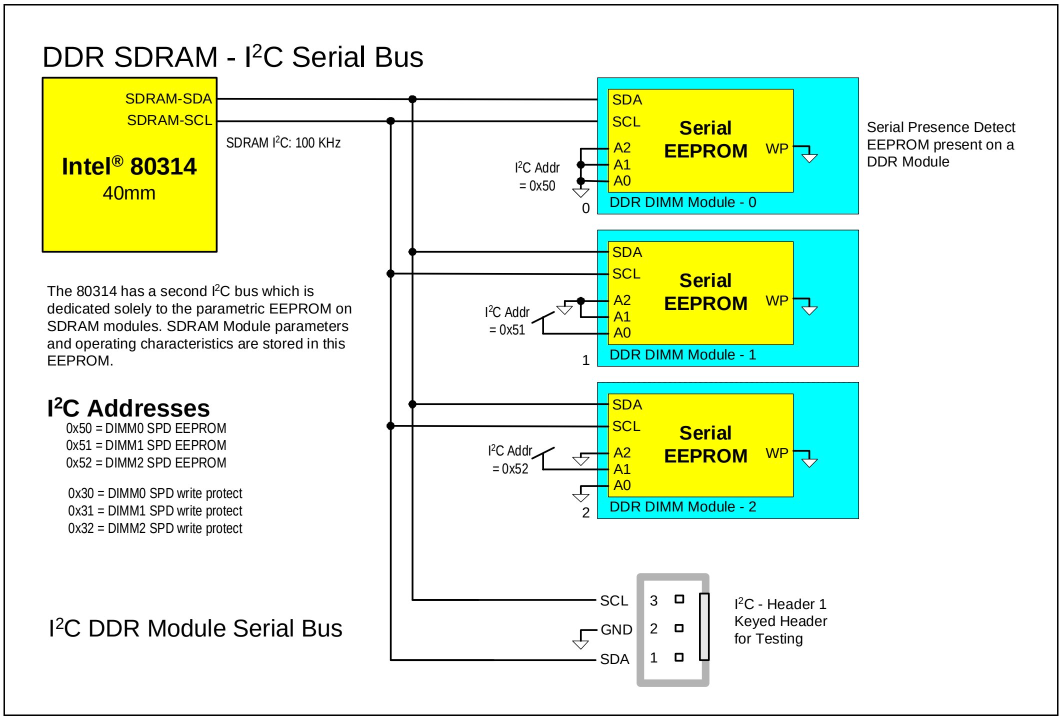
2.10.2 DDR I^{2}C Bus Subsystem

The Intel® 80314 I/O processor companion chip DDR SDRAM interface has a dedicated I^{2}C bus that communicates with Serial Presence Detect (SPD) EEPROMs on each DDR Module. These EEPROMs are read by the host system to identify whether a module is present and to store module configuration data. The SPD EEPROMs are addressed at 0x50, 0x51 and 0x52. After these EEPROMs have been configured, writing to 0x30, 0x31, or 0x32 write-protects the first 128 bytes of each EEPROM.

The IQ80315 CRB operates the DDR SDRAM I^{2}C bus at 100 KHz because some SDRAM modules cannot communicate on the I^{2}C bus at 400 KHz.

Figure 4 shows a block diagram of the DDR SDRAM I^{2}C bus.

Figure 4. DDR I^{2}C Serial Bus



The 80314 has a second I^{2}C bus which is dedicated solely to the parametric EEPROM on SDRAM modules. SDRAM Module parameters and operating characteristics are stored in this EEPROM.

I^{2}C Addresses

0x50 = DIMM0 SPD EEPROM
0x51 = DIMM1 SPD EEPROM
0x52 = DIMM2 SPD EEPROM

0x30 = DIMM0 SPD write protect
0x31 = DIMM1 SPD write protect
0x32 = DIMM2 SPD write protect

2.11 Connectors

Connectors, switches, and headers are mounted on the PCB for Rear Gasket I/O, internal use, and front panel I/O. This section describes the parts selected and the pinout for these components. See Figure 2, “Intel® IQ80315 Customer Reference Board Layout Diagram” on page 13 for the approximate layout of these components.

2.11.1 Rear Gasket

Most of the I/O connectors requiring user access are located in the rear gasket area. The IQ80315 CRB rear gasket area conforms to the standard ATX rear gasket dimensions and location. Figure 5 shows the standard ATX rear gasket dimensions and Figure 6 shows how the IQ80315 CRB rear gasket components are arranged.

Figure 5. ATX Rear Gasket Dimensions

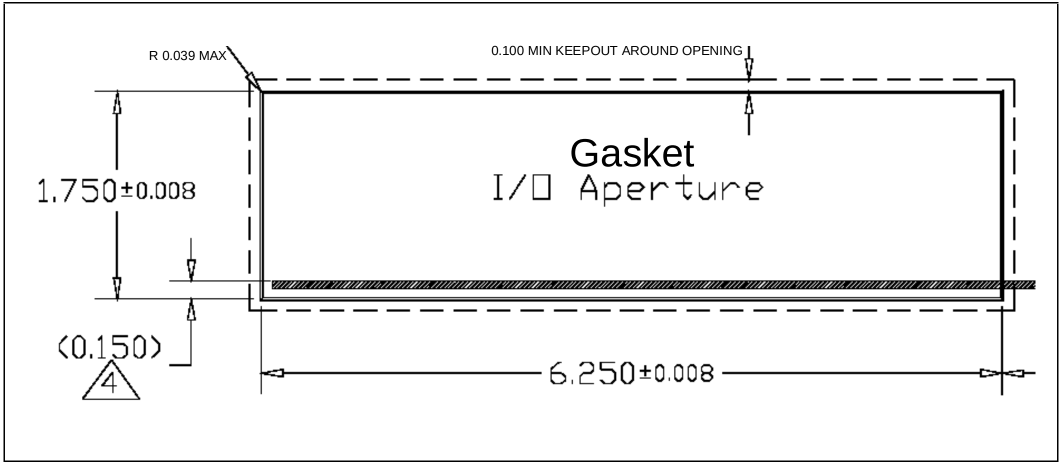
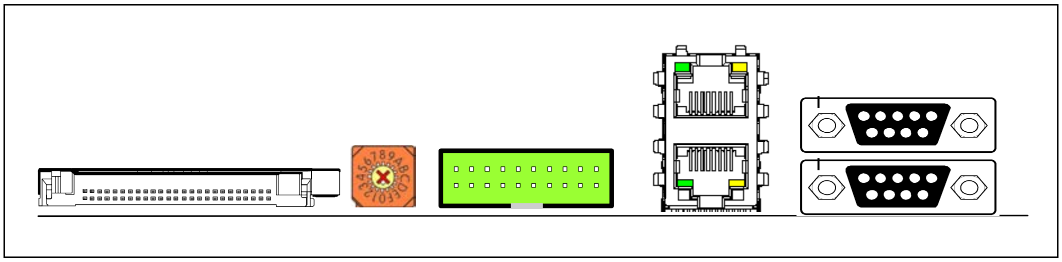


Figure 6. Rear Gasket Component Layout



2.11.2 Dual 10/100/1000 Base-T RJ45 Ethernet Connector

The RJ45 Ethernet connectors are dual 1GbE RJ45 jacks. These connectors have built-in magnetics and a green and amber/green LED for each port. The pinout follows standard Ethernet connector pinout.

2.11.3 Dual Vertical Stack Right Angle 9-Pin D Serial Port Connector

The dual vertical DB-9 serial port socket connector is TBD. Serial port 2 is on the bottom, and serial port 1 is on the top.

Table 8. Serial/GPIO–J6A1

Pin Top	Signal	Pin Bottom	Signal
1	COM0_DCD	1	COM1_DCD
2	COM0_RXD	2	COM1_RXD
3	COM0_TXD	3	COM1_TXD
4	COM0_DTR	4	COM1_DTR
5	NC	5	NC
6	COM0_DSR	6	COM1_DSR
7	COM0_RTS	7	COM1_RTS
8	COM0_CTS	8	COM1_CTS
9	COM0_RI	9	COM1_RI

2.11.4 JTAG Connector—20-Pin Right-Angle Header

The JTAG Header is a 20-pin, right-angle, shrouded connector. Section 2.11.4 shows the JTAG connector pinout.

Table 9. XS JTAG–J4A1

Pin	Signal	Pin	Signal
1	Vref	2	Vsupply
3	NTRST	4	GND
5	TDI	6	GND
7	TMS	8	GND
9	TCK	10	GND
11	NC (RTCK)	12	GND
13	TDO	14	GND
15	NSRST	16	GND
17	NC (DBGRQ)	18	GND
19	NC (DBGACK)	20	GND

2.11.5 CompactFlash* II Connector and Eject Mechanism

The CompactFlash* socket and eject mechanism is an AVX/Elco* 5620 CompactFlash* II Socket with right-hand straight ejector.

Table 10. CF–J2A1

Pin	Signal	Pin	Signal
1	GND	26	CD1_N
2	CFII_D03	27	CFII_D11
3	CFII_D04	28	CFII_D12
4	CFII_D05	29	CFII_D13
5	CFII_D06	30	CFII_D14
6	CFII_D07	31	CFII_D15
7	CFII_CE1_N	32	CFII_CE2_N
8	CFII_A10	33	VS1_N
9	CFII_OE_N	34	CFII_IORD_N
10	CFII_A09	35	CFII_IOWR_N
11	CFII_A08	36	CFII_WE_N
12	CFII_A07	37	CFII_RDYB_N
13	VCC	38	VCC
14	CFII_A06	39	CSEL_N
15	CFII_A05	40	VS2_N
16	CFII_A04	41	CFII_RST
17	CFII_A03	42	CFII_WAIT
18	CFII_A02	43	CFII_INPAK_N
19	CFII_A01	44	CFII_REG_N
20	CFII_A00	45	BVD2_N
21	CFII_D00	46	BVD1_N
22	CFII_D01	47	CFII_D08
23	CFII_D02	48	CFII_D09
24	CFII_WP_IO_N	49	CFII_D10
25	CD2_N	50	GND

2.11.6 Hex Right-Angle Rotary Switch

The rotary switch is an NKK DR FC 16 right-angle hex switch.

2.12 Internal Connectors

Internal connectors include the PCI, DDR SDRAM, power connector, configuration switches and headers, test headers, and battery connectors.

2.12.1 PCI-X Connectors

The PCI-X connectors are AMP* #145165-4 or #145168-4 connectors. The PCI-X bus 1 connector is a different color than the PCI-X bus 2 connectors.

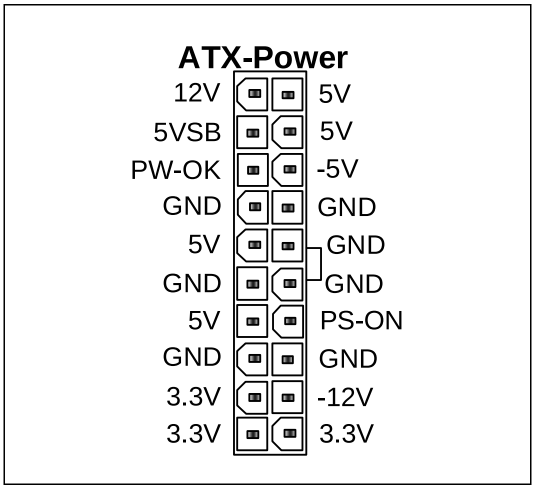
2.12.2 DDR SDRAM Connectors

The DDR SDRAM module connectors are 184-pin AMP* #390241-1 connectors. These are 2.5 V keyed connectors. The connector furthest from the Intel® 80314 I/O processor companion chip is a different color from the other two DDR SDRAM module connectors in order to help identify where to plug in a single module.

2.12.3 ATX Power Connector

The power connector is a standard ATX power connector. It does not use the supplementary ATX +12 V power connector. Figure 7 shows the ATX power connector pinout.

Figure 7. ATX Power Connector



2.12.4 Lithium CR2032 Battery Coin Cell Socket

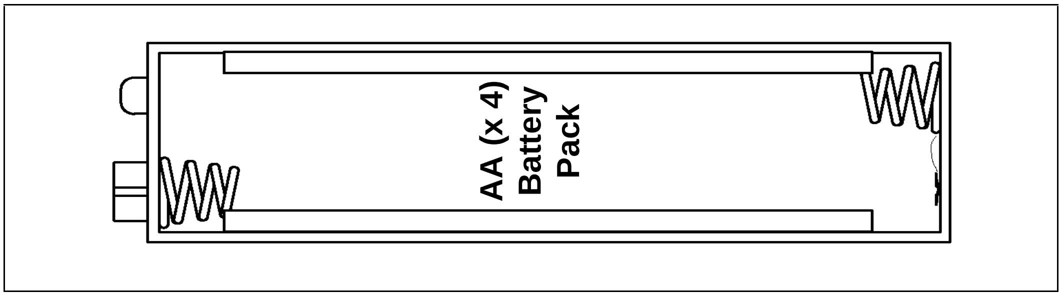
The lithium coin cell battery socket is a horizontal PCB mounted socket.

2.12.5 DDR SDRAM Backup Battery Pack Holder and Connector

The DDR SDRAM backup battery (four AA size NiMH batteries) is held in a Mouser Electronics* 12BH352 battery holder. The battery holder is attached to the board with adhesively bonded Velcro* strips. There are also four slots cut in the board, two on each side of the battery holder, for additional tie-downs if required.

The battery holder has a pair of solder pins. Two wires are soldered to these pins and then attached to a 2-pin keyed connector near the battery pack. Figure 8 shows the battery holder.

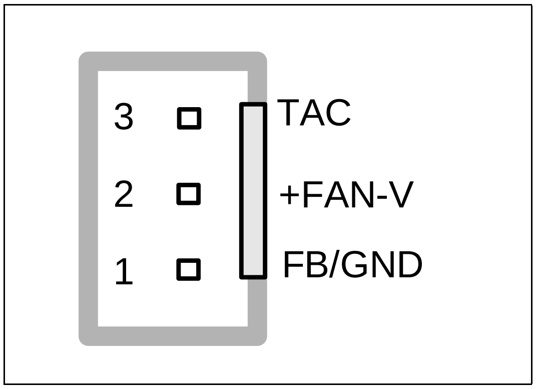
Figure 8. DDR SDRAM Backup Battery Holder



2.12.6 Fan Connectors for Front and Rear Case

The fan connectors are PC standard 3-pin keyed connectors. Figure 9 shows the fan controller pinout.

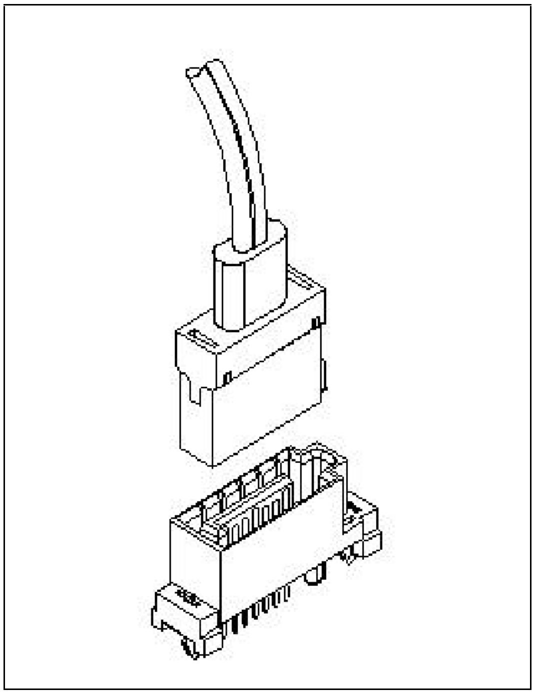
Figure 9. Fan Controller Pinout



2.12.7 Serial-ATA (SATA) Connectors

The serial ATA connectors are Molex* 67491-003x vertical mount S-ATA connectors. Figure 10 shows the serial ATA connector and mating cable.

Figure 10. Serial ATA Connector



2.12.8 Unused GPIO and Interrupt Header

The unused GPIO and interrupt header is a 0.1", 2 × 14 header. The header pinout is TBD.

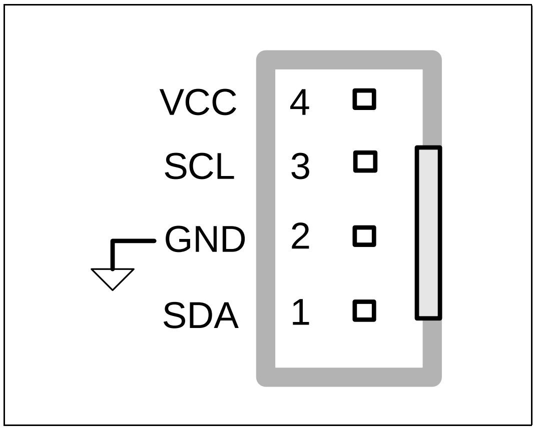
2.12.9 Peripheral Bus Test and User Expansion Connector

The peripheral bus test/expansion header is a PC104 like 0.1" header.

2.12.10 I^{2}C Headers

The I^{2}C tests headers are keyed 4-pin headers. Figure 11 shows the I^{2}C header pinout.

Figure 11. I^{2}C Header

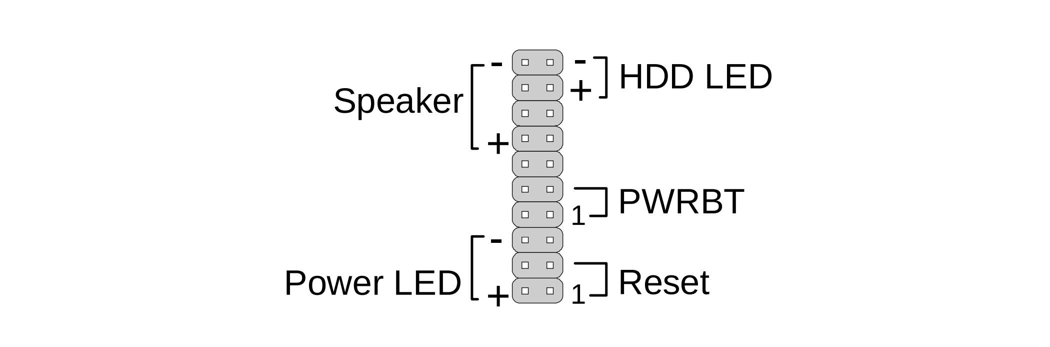


2.13 Front Panel Connectors

2.13.1 ATX Front Panel Connector

The IQ80315 CRB has a standard ATX 20-pin front panel connector for access to a speaker, reset switch, Power LED, HDD LED, and system power button. Figure 12 shows the ATX front panel header pinout.

Figure 12. ATX Front Panel Connector



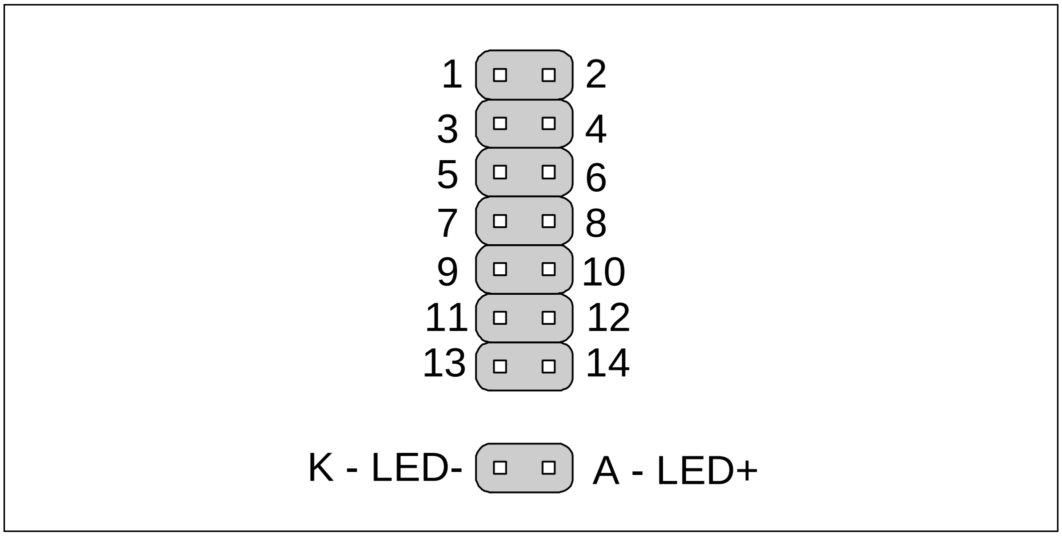
2.13.2 LCD and Backlight Headers

The LCD connector is a 0.1" pitch 7 × 2 header. The LCD backlight header is a 2-pin shrouded header. Backlight pin A is a resistively coupled connection to 5 V, and pin K is connected to GND. Table 11 shows the LCD and backlight header pinouts.

Table 11. LCD and Backlight Connector Pinout

Pin	Symbol	Level	Function
1	Vss	GND	Ground
2	Vdd	+5V	Module power
3	V0	Vss-Vdd	Liquid crystal drive
4	RS_	H/L	Register select, H = data, L = ins
5	R/W	H/L	Read/Write, H = read (LCD->CPU)
6	E	Edge	Edge-sensitive Enable
7	DB0	H/L	Data bit[0] (least significant)
8	DB1	H/L	Data bit[1]
9	DB2	H/L	Data bit[2]
10	DB3	H/L	Data bit[3]
11	DB4	H/L	Data bit[4]
12	DB5	H/L	Data bit[5]
13	DB6	H/L	Data bit[6]
14	DB7	H/L	Data bit[7] (most significant
	A	Vdd	LED (+)
	K	Vss	LED (-)

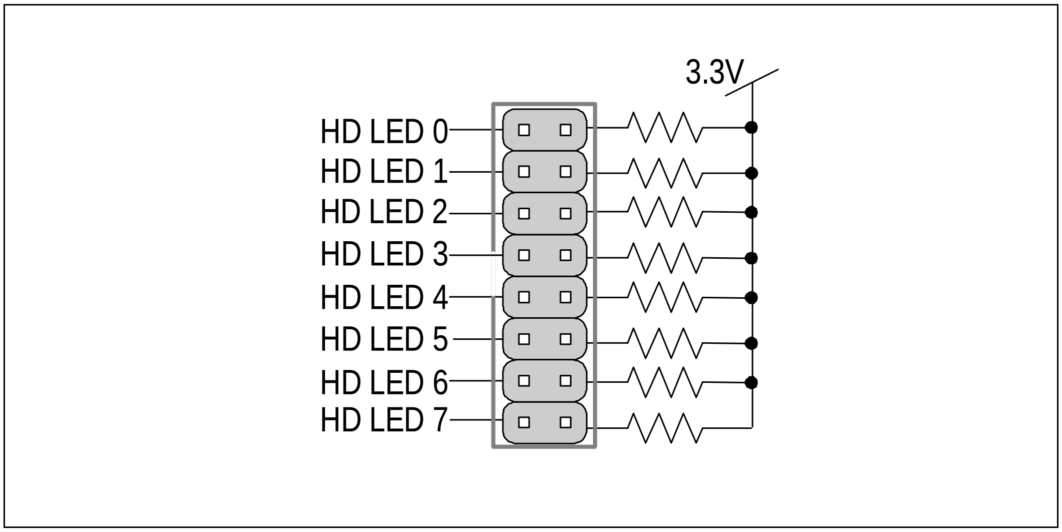
Figure 13. LCD and Backlight Connector Pinout



2.13.3 S-ATA Drive Activity LED Header

The serial ATA drive activity header is a shrouded 16-pin 0.1" 2 × 8 header. Figure 14 shows the serial ATA drive activity header pinout.

Figure 14. Serial ATA Drive Activity LED Header



2.14 Switches, Jumpers, and Reset Configuration Strapping

2.14.1 Reset Strapping Options

The IQ80315 CRB configures reset strapping options through a DIP switch, hard-wired pull-up or pull-down resistors, and tristate CPLD outputs. Table 12 shows the reset strapping options on the IQ80315 CRB. Configuration signals are pulled low when the switch is **on,** the jumper is **shorted** or in the **1–2** position, or a pull-down (**Pd**) resistor is installed (**bold indicates the default setting**). Configuration signals are pulled high when the switch is **off,** the jumper is **open** or in the **2–3** position, or a pull-up (**Pu**) resistor is installed (**bold indicates the default setting**). A silk-screen graphic on the board next to each switch identifies the default switch settings. SWxx, Jxx, and Rxx denote that a switch, jumper, or resistor sets the configuration.

Table 12. Reset Strapping Options (Sheet 1 of 2)

Signal	SW/J/R	Configuration Options
Intel® 80314 I/O processor companion chip (80314)		
PWRUP_P1_PRIM	R5P7	Pd PCI: 1 is not a primary PCI bus. Pu PCI: 1 is a primary PCI bus (default).
PWRUP_P2_PRIM	R5E5	Pd PCI: 2 is not a primary PCI bus. Pu PCI: 2 is a primary PCI bus (default).
PWRUP_XS_SWRST	J5F3	1–2: 80314 CPU interface is functional after reset (default) 2–3: 80314 CPU interface is disabled after reset
PWRUP_P1_SWRST	J5F1	1–2: 80314 PCI1 interface is functional after reset (default) 2–3: 80314 PCI1 interface is disabled after reset
PWRUP_P2_SWRST	J5F2	1–2: 80314 PCI2 interface is functional after reset (default) 2–3: 80314 PCI2 interface is disabled after reset
PWRUP_P1_ARB	R5N2	Pu Enable: PCI 1 arbiter (default) Pd Disable: PCI 1 arbiter
PWRUP_P2_ARB	R5N3	Pu Enable: PCI 2 arbiter (default) Pd Disable: PCI 2 arbiter
PWRUP_HLP_BSWP	R5P10	Pd Disable: Intel XScale® byte-swap mode Pu Enable: Intel XScale® byte-swap mode
Gigabit Ethernet		
P0 CONFIG3	J4A3	1–2: Prefer Master during auto negotiation (default) 2–3: Prefer Slave during auto negotiation
P0 CONFIG3	J4A2	1–2: Prefer Master during auto negotiation (default) 2–3: Prefer Slave during auto negotiation

Table 12. Reset Strapping Options (Sheet 2 of 2)

Signal	SW/J/R	Configuration Options
Serial ATA		
SATA 0 DPA_MODE#	S5C1 switch 1	On Master: Master mode for high performance (default) Off Master: Slave mode for software compatibility
SATA 1 DPA_MODE#	S5C1 switch 4	On Master: Master mode for high performance (default) Off Master: Slave mode for software compatibility
SATA 0 SSCEN	S5C1 switch 2	On: Spread-spectrum clock generation disabled on all channels Off: Spread-spectrum clock generation enabled on all SATA channels (default)
SATA 1 SSCEN	S5C1 switch 3	On: Spread-spectrum clock generation disabled on all channels Off: Spread-spectrum clock generation enabled on all SATA channels (default)
PCI/PCI-X A		
PCI-X Enable	S5C2 switch 1	On: PCI-X Enable (default)
Force 66 MHz PCI-X mode	S5C2 switch 2	On: PCI-X 66 MHz max. (default)
Force PCI mode	S5C2 switch 3	On: PCI mode only (default)
Force 33 MHz PCI mode	S5C2 switch 4	On: 33 MHz PCI mode only (default)
PCI/PCI-X B		
PCI-X Enable	S6B1 switch 1	On: PCI-X Enable (default)
Force 66MHz PCI-X mode	S2F1 switch 2	On: PCI-X 66 MHz max (default)
Force PCI mode	S6B1 switch 3	On: PCI mode only (default)
Force 33MHz PCI mode	S6B1 switch 4	On: 33 MHz PCI mode only (default)

Table 13 shows the 80314 reset configuration options that are fixed on the CRB and are not user-selectable.

Table 13. Fixed Reset Configuration Inputs

Signal	Pu / Pd	Option Selected
Intel® 80314 I/O Processor Companion Chip (80314)		
PWRUP_TRANS	Pd	80314 PCI interfaces are set to embedded mode
PWRUP_P1_BYP	Pd	Enable PCI 1 PLL
PWRUP_P2_BYP	Pd	Enable PCI 2 PLL
TEST_ON	Pd	Disable 80314 internal test mode
BIDIR_CTL	Pd	Test data direction. pull low for normal operation.
P1_RSTDIR	Pu	PCI 1 reset is an output from the 80314
P2_RSTDIR	Pu	PCI 2 reset is an output from the 80314
CPU		
PLLCFG	Pu	Selects 6× CPU core clock
LOWVCC	Pu	CPU Core voltage is greater than 1.0 V.
LOWVPP	Pu	CPU I/O voltage is greater than 2.5 V.
Serial ATA		
32BITPCI#	Pu	Sets the 31244 to 64-bit PCI/X bus

Table 14. CPLD Reset Configuration Outputs

Signal	Reset	Option Selected
XS_C_IRQ[0] / PWRUP_FADJ	0	100 MHz CPU bus clock
XS_C_IRQ[1] / PWRUP_XS_BYP	0	Don’t bypass Intel XScale® PLL
XS_C_FIQ[1] / PWRUP_SD_BYP	0	Don’t bypass SDRAM PLL

2.14.2 Miscellaneous Buttons, Switches, and Jumpers

In addition to the reset configuration options, the IQ80315 CRB contains the following miscellaneous switches and jumpers:

- **Reset Switch**—A momentary, normally open miniature push-button switch is mounted on the motherboard. When the button is pressed, the system reset is activated for a minimum of 100 ms. The real-time clock component debounces the reset switch input and ensures the minimum reset pulse width. The reset output from the RTC is wire-OR'ed with the power supervisor reset output.
- **Rear JTAG CPU 1 only**—S6C1 switch 1 enables CPU 1 in the main JTAG scan chain. When the switch is ON, CPU 1 is included in the scan chain; when OFF, CPU 1 is bypassed.
- **Rear JTAG CPU 1 and CPU 2**—S6C1 switch 2 enables CPU 2 in the main JTAG scan chain. When the switch is ON, CPU 2 is included in the scan chain; when OFF, CPU 2 is bypassed.
- **Rear JTAG Chain Enable**—S6C1 switch 3 enables the system components (80314, 31244, 88E1020, and CPLD) in the main JTAG scan chain. When the switch is ON, the system components are included in the scan chain; when OFF, the system components are bypassed.
- **JTAG Forced Reset**—S6C1 switch 4 forces the JTAG reset on the system components. When the switch is on, the system components are held in reset.
- **Reset Jumper CPU 1**—J6B3 enables the 80314 CPU reset to CPU 1 when installed. When removed, CPU 1 is held in reset.
- **Reset Jumper CPU 2**—J6B4 enables the 80314 CPU reset to CPU 2, when installed. When removed, CPU 2 is held in reset.
- **Square-Wave Out Jumper**—J8G2 routes the buffered RTC square-wave output to either the buzzer or the speaker. Shorting pins 1 and 2 sends the signal to the buzzer. Shorting pins 2 and 3 sends the signal to the speaker.
- **Buzzer Volume Switch**—JP8G1 controls the audio buzzer volume. When the switch is ON, the buzzer volume is high; when the switch is OFF, the volume is low.
- **GPIO/Intr Header**—All unused interrupts and GPIO signals are routed to a 28-pin (2 × 14), 0.1" dual row header.

2.15 Debug Features

The IQ80315 CRB provides a JTAG connector for debugging the CPUs and the 80314, a test point/expansion header on the peripheral bus interface, seven-segment LEDs, and several ground points around the board to assist with system debug. Most interfaces on the IQ80315 CRB can be tested and debugged using debug tools that plug into existing sockets. For example, a PCI bus analyzer, DDR logic analyzer card, Gigabit Ethernet tester, and USB tester can all be plugged into the respective sockets for interface debug.

2.15.1 JTAG

The IQ80315 CRB has a 20-pin, right-angle JTAG header that is compatible with the Wind River* HSI Vision-PROBE/CLICK*, ARM* Multi-ICE*, and MACRAIGOR* Raven/mpDemon* debuggers. This header is located in the rear gasket area for easy access. The components are isolated from the JTAG header through an IDT* QS3VH2861 QuickSwitch* for ESD protection.

2.15.2 Logic Analyzer Test Points

The demultiplexed peripheral bus and control signals from the CPLD are brought out to a PC104-like 0.1" header. This can be used for logic-analyzer probing or for prototyping peripheral bus circuits.

3.0 Technical Reference

3.1 Introduction

Section 3.2 through Section 3.7 contain several standalone tables that describe system resource allocation:

- Table 15 describes the system memory map.
- Table 16 describes the peripherals memory map.
- Table 17 defines the PCI request/grant signal mapping.
- Table 18 describes the PCI IDSEL mapping.
- Table 19 describes the interrupts.
- Table 20 describes the GPIO assignments.
- Table 21 describes the I^{2}C address map.

3.2 Memory Map

Table 15. System Memory Map

Address	Size (MB)	Description
0x00000000	1 GB	SDRAM
0x40000000	256	FLASH/Peripheral Bus (see Table 16)
0x50000000	1	SRAM
0x50100000	64 KB	Intel® 80314 I/O Processor Companion Chip Control Registers
0x50110000	960 KB	Reserved
0x50600000	762	No Access
0x80000000	495	PCI1 MEM32
0x9EFF0000	1	PCI1 I/O
0x9F000000	16	PCI1 CFG
0xA0000000	256	PCI1 PFM1
0xB0000000	256	PCI1 PFM2
0xC0000000	495	PCI2 MEM32
0xDEFF0000	1	PCI2 I/O
0xDF000000	16	PCI2 CFG
0xE0000000	256	PCI2 PFM1
0xF0000000	256	PCI2 PFM2

Table 16. Peripheral Bus Memory Map

Chip Enable	Address Range (hex)	Size	Data Bus Width	Description
PCE0#	4000 0000–40FF FFFF	16 MB	8-bit	Flash Memory (re-mapped)
PCE1#	4E60 0000–4E67 FFFF	512 KB	16-bit	CompactFlash* Memory
PCE1#	4E68 0000–4E6F FFFF	512 KB	16-bit	CompactFlash* I/O
PCE2#	4E70 0000–4E7F FFFF	1 MB	16-bit	USB
PCE3#	4E80 0000–4E80 FFFF	64 KB	8-bit	Product Code
PCE3#	4E81 0000–4E81 FFFF	64 KB	8-bit	Board Revision
PCE3#	4E82 0000–4E82 FFFF	64 KB	8-bit	CPLD Firmware Revision
PCE3#	4E83 0000–4E83 FFFF	64 KB	8-bit	LCD Display
PCE3#	4E84 0000–4E84 FFFF	64 KB	8-bit	Hex Display Left
PCE3#	4E85 0000–4E85 FFFF	64 KB	8-bit	Hex Display Right
PCE3#	4E86 0000–4E8C FFFF	448 KB	8-bit	Reserved
PCE3#	4E8D 0000–4E8D FFFF	64 KB	8-bit	Rotary Switch
PCE3#	4E8E 0000–4E8E FFFF	64 KB	8-bit	Electronic Serial Number
PCE3#	4E8F 0000–4E8F FFFF	64 KB	8-bit	Battery Status

3.3 PCI Request/Grant Mapping

Table 17. PCI Request/Grant Mapping

Request/Grant	System Resource
PCI-X Bus 1	
1	Intel® 31244 SATA Controller #0
2	Intel® 31244 SATA Controller #1
3	PCI Slot #0
4	Not used
5	Not used
6	Not used
7	Not used
PCI-X Bus 2	
1	PCI Slot #1
2	PCI Slot #2
3	Not used
4	Not used
5	Not used
6	Not used
7	Not used

3.4 PCI IDSEL Mapping

Table 18. PCI IDSEL Mapping

Bus	Device	IDSEL	System Resources
PCI-X Bus 1			
0	0	AD16	Intel® 80314 I/O processor companion chip
0	1	AD17	PCI Bus Connector 0
0	2	AD18	31244 SATA Controller 0
0	3	AD19	31244 SATA Controller 1
PCI-X Bus 2			
1	0	AD16	Intel® 80314 I/O processor companion chip
1	1	AD17	PCI Bus Connector 1
1	2	AD18	PCI Bus Connector 2

3.5 Interrupts

Table 19. Interrupts

Interrupt	System Resources
Intel® 80314 I/O Processor Companion Chip External Interrupts	
0	CompactFlash*
1	Debug header
2	Temperature sensor
3	Fan alert
4	PBI expansion header
5	Debug header
6	Debug header
7	Debug header
8	Debug header
9	Debug header
10	Debug header
11	Debug header
12	Debug header
13	Debug header
14	Debug header
15	Debug header
Intel® 80314 I/O Processor Companion Chip Internal Interrupts	
16	Gigabit Ethernet
17	DMA
18	Peripheral Bus
19	SDRAM
20	CPU
21	I^{2}C
22	PCI 2
23	PCI 1
PCI Interrupts	
P1_INTA#	PCI bus 1 connector INTA#, Debug header
P1_INTB#	PCI bus 1 connector INTB#, Debug header
P1_INTC#	PCI bus 1 connector INTC#, 31244 SATA Controller 0
P1_INTD#	PCI bus 1 connector INTD#, 31244 SATA Controller 1
P2_INTA#	PCI bus 2 connector 1 INTA#, connector 2 INTB#, Debug header
P2_INTB#	PCI bus 2 connector 1 INTB#, connector 2 INTC#, Debug header
P2_INTC#	PCI bus 2 connector 1 INTC#, connector 2 INTD#, Debug header
P2_INTD#	PCI bus 2 connector 1 INTD#, connector 2 INTA#, Debug header

3.6 GPIO Assignments

Table 20 shows the Intel® 80314 I/O processor companion chip GPIO signal assignments.

Table 20. GPIO Assignment

GPIO	Signal Name	Purpose
GPIO0	J8G1 pin 2	Debug header
GPIO1	J8G1 pin 4	Debug header
GPIO2	J8G1 pin 6	Debug header
GPIO3	J8G1 pin 8	Debug header
GPIO4	J8G1 pin 10	Debug header
GPIO5	J8G1 pin 12	Debug header
GPIO6	J8G1 pin 14	Debug header
GPIO7	J8G1 pin 16	Debug header

3.7 I^{2}C Address Map

Table 21. I^{2}C Address Map

Main I^{2}C Address	System Resource
0x48	Temperature sensor 0
0x49	Temperature sensor 1
0x4B	Fan controller 0
0x1B	Fan controller 1
0x50	2Kbyte Serial EEPROM
0x68	Real Time Clock

DDR I^{2}C Address	System Resource
0x30	DDR DIMM Serial Presence Detect EEPROM write protect
0x31	DDR DIMM Serial Presence Detect EEPROM write protect
0x32	DDR DIMM Serial Presence Detect EEPROM write protect
0x50	DDR DIMM Serial Presence Detect EEPROM
0x51	DDR DIMM Serial Presence Detect EEPROM
0x52	DDR DIMM Serial Presence Detect EEPROM

4.0 Software

4.1 Introduction

This section covers requirements for software and firmware running on the Intel® IQ80315 Customer Reference Board.

4.2 Boot-Loader/Target Monitor/Startup Firmware

The targeted boot-loader for the IQ80315 CRB is RedBoot*. RedBoot is a complete bootstrap environment based on the eCos Hardware Abstraction Layer. RedBoot supports downloading and debugging gcc-based and eCos applications, flash and network booting of the Linux* kernel, and downloading and updating of flash images remotely via serial or Ethernet connections. RedBoot can be used as for product development, debug, and deployment in final products. RedBoot provides a target-side hook known as a "stub" (or gdb-stub) that enables gdb, the GNU debugger, to communicate with applications built with gcc or g++ running on IQ80315 CRB. RedBoot source code is covered under the eCos license.

4.2.1 Initialization Flow

RedBoot* is in charge of booting the IQ80315 CRB to a known state where programs can be downloaded and debugged. The basic outline of how RedBoot boots the platform is described below.

The basic initialization sequence for RedBoot is as follows:

1. Disable interrupts. (Technically they are disabled at reset, but for soft reset this is included.)
2. Initialize the Peripheral Bus Interface (PBI) by initializing base address registers (BARs), Look-Up Tables (LUTs), and timings for board peripherals.
3. Enable I-cache.
4. Re-locate flash.
5. Set TTB and enable MMU.
6. Read DDR DIMM serial presence detect EEPROMs for memory parameters.
7. Set memory parameters.
8. Turn DDRAM on.
9. Enable data cache.
10. Enable BTB.
11. Flush all.
12. Clear CPU and SDRAM ECC error logs.
13. Test battery.
14. Enable CPU ECC.
15. Enable SDRAM ECC.
16. Scrub loop: Write zeroes to all SDRAM memory locations.
17. Scrub loop: Write zeroes to all SRAM memory locations.
18. Enable SRAM.

4.2.2 Memory Map Setup by RedBoot

Table 15 shows the system memory map after RedBoot* has setup the virtual–to–physical-address translation.

4.2.3 MMU Tables

The X, C, and B bit in the MMU page tables indicate the caching policy for a particular region. See Table 22 ("Caching Policy" on page 45) for a description of how the X, C, and B bits translate into the caching policy for a region. See Section 4.2.2 ("Memory Map Setup by RedBoot" on page 45) for the Virtual Memory Map and the caching policy setup by RedBoot.

Table 22. Caching Policy

X	C	B	Description
0	0	0	Un-cached/un-buffered
0	0	1	Un-cached/buffered
0	1	0	Cached/buffered: Write-through, read allocate
0	1	1	Cached/buffered: Write-back, read allocate
1	0	0	Invalid—not used
1	0	1	Un-cached/buffered: No write buffer coalescing
1	1	0	Mini data-cache: Policy set by Auxiliary Control Register
1	1	1	Cached/buffered: Write back, read/write allocate

Table 23. Memory Map with Caching Policy

Virtual Address	X	C	B	Description
0x00000000–0x40000000	1	1	1	SDRAM
0x40000000–0x50000000	0	1	0	FLASH/PBI
0x50000000–0x50100000	1	1	1	SRAM
0x50100000–0x50110000	0	0	0	Control Registers
0x50110000–0x60000000	0	0	0	No access
0x60000000–0x80000000	0	0	0	Un-cached SDRAM Alias
0x80000000–0x9EFF0000	0	0	0	PCI1 MEM32
0x9EFF0000–0x9F000000	0	0	0	PCI1 I/O
0x9F000000–0xA0000000	0	0	0	PCI1 CFG
0xA0000000–0xB0000000	0	0	0	PCI1 PFM1
0xB0000000–0xC0000000	0	0	0	PCI1 PFM2
0xC0000000–0xDEFF0000	0	0	0	PCI2 MEM32
0xDEFF0000–0xDF000000	0	0	0	PCI2 I/O
0xDF000000–0xE0000000	0	0	0	PCI2 CFG
0xE0000000–0xF0000000	0	0	0	PCI2 PFM1
0xF0000000–0xFFFFFFFF	0	0	0	PCI2 PFM2

4.2.4 Ethernet Communications

The IQ80315 CRB has two Gigabit Ethernet ports available. RedBoot initializes Port0 for its use and uses it in a polling mode. ***The MAC address for Port0 must be stored inside the Boot EEPROM, or RedBoot will not initialize the port.*** The IP address for Port0 can be set with three different methods:

1. Use the RedBoot "`fconfig`" command to program the IP address into the Flash.
2. The RedBoot "`ip_address`" command can be used to set an IP address, but this is not be stored in nonvolatile memory and is lost on a board reset or power cycle.
3. RedBoot can use BOOTP to obtain an IP address from a BOOTP server at boot time.

Note: BOOTP is selected as the default (it can be changed with `fconfig`), but if there is no BOOTP server available, it takes up to 30 seconds for the BOOTP search to time out

As soon as Ethernet communications have been established, a user can `telnet` to the IQ80315 CRB, instead of using serial communication. Application programs can be downloaded through the GNU debugger through the Ethernet connection as well.

RedBoot also supports downloading an application image across the Ethernet port with the TFTP protocol using the "load" command with the proper parameters.

4.2.5 UART Communications

RedBoot supports both UART ports on the IQ80315 CRB. The first UART port to receive a character from a host is designated as the debug port. If dual processor is started, the port which is not the debug port is used as communication for Processor 1. The connection setup for UART communications on either port:

- Baud Rate: 115200, 8-bit data, no parity bits, one stop bit, no flow control

Note: The baud rate can be changed using the "`fconfig`" or "`baudrate`" commands

4.2.6 Flash I/O

The RedBoot image, other executable images, as well as data, can be stored in flash in a simple file store. The `fis` command is used to manipulate and maintain flash images. To initialize the Flash file system, the command `fis init` must be entered. Entering the command, `help fis` lists the other RedBoot command options that can be used to manipulate the flash file system.

4.2.7 CompactFlash*

The IQ80315 CRB has an 8-bit CompactFlash* (CF) interface that can be used to store data and executable images. The command `cf info` can be used to check the status of a CF card plugged into the socket.

4.2.8 SATA Disk Communications

The RedBoot diagnostic menu has a selection to test the PCI-X busses. Under this sub-menu, a read/write to the SATA disks is available. The IQ80315 CRB RedBoot does not have support for storing images or data to the SATA disks.

4.2.9 General Peripherals

The location in the memory map for the following peripherals can be found in Table 15 ("System Memory Map" on page 40). The description of these peripherals can be found in Section 2.7 ("Peripheral Bus" on page 17).

4.2.9.1 Seven-Segment LED Devices

There are two seven-segment LED devices on the IQ80315 CRB. The address location for these devices is located in Table 15 ("System Memory Map" on page 40). RedBoot does not use the LEDs, except to indicate a failure condition. The devices default to "off" when the IQ80315 is powered up.

4.2.9.2 Product Code Register

The product code register is a location inside the CPLD that holds the product identification for the IQ80315 CRB. The register read back 1 for a IQ80315 CRB.

4.2.9.3 Board Revision Register

The board revision register is a location inside the CPLD that holds the stepping of the IQ80315 CRB.

4.2.9.4 CPLD Firmware Revision Register

The CPLD F/W Revision Register is a location inside the CPLD that indicates the version of firmware that is currently programmed into the CPLD.

4.2.9.5 Rotary Switch Register

The 16-position rotary switch can be read to indicate the current position of the rotary switch. The rotary switch is not used by RedBoot, but it can be tested under the RedBoot diagnostics menu. The rotary switch can be utilized to choose different boot flavors on power-up.

4.2.9.6 Electronic Serial Number(s)

The IQ80315 CRB has two DS2401 devices that each contain a unique serial number. These are not accessed by RedBoot. The diagnostic menu currently does not support reading the serial number devices.

4.2.9.7 Battery Status Register

The battery status register can be read from the RedBoot diagnostic menu.

4.2.10 Diagnostics

RedBoot provides the following utilities. Access to these utilities is provided through the IQ80315 CRB UART interface using a tool such as Windows Hyperterminal*. This screen can be obtained by typing `diag` at the RedBoot prompt.

The IQ80315 CRB hardware tests include:

- Memory Tests

  Description: This option is used to test SDRAM or Primary PCI memory. The selection runs five different tests over the memory ranges provided. If errors are encountered, the test is aborted. The test does not modify the seven-segment display.

- Repeating Memory Tests

  Description: This is the same as memory test 1, except that the test continues *ad infinitum*.

- Repeat-On-Fail Memory Test

  Description: This test operates the same as Memory Test 1, except if it encounters a failure, it continuously tests the memory location that failed.

- Rotary Switch Test

  Description: The position of the rotary switch is output on the 7-segment LED.

- IOC80314 Ethernet Configuration

  Description: This test allows the user to program the MAC address of Port0 and/or Port1 of the Ethernet controller. The MAC address is programmed into the Boot EEPROM.

- Battery Status Test

  Description: This test reads the battery status register and reports whether the battery is installed, charging, or fully charged.

- Timer Test

  Description: This test verifies that the IOC80314 timer is set correctly by printing out a "." for every 1-second tick of the timer.

- PCI Bus Test

  Description: Selecting this option brings up the PCI Test Menu which facilitates scanning and testing of the PCI busses. If Agilent* 2929 PCI-X exerciser cards are inserted into slots A1, B1, and/or B2, they can be tested using the PCI test menu. The 31244 SATA controllers can also be probed and tested through the PCI test menu.

  Note: The read/write test to the SATA disks are ***destructive*** and must not be used on any disk that needs its data preserved.

- CPU Cache Loop

  Description: This test puts the 80200 processor into a tight cache loop with no hope of escape.

- Read DDR0 and DDR1

  Description: This test probes the SPD devices on Bank0 and Bank1 of the IQ80315 CRB and reports how memory is inserted into the banks.

- Read Temperature Sensors on I^2C

  Description: The two temperature sensors located on the I^2C bus can be probed using this diagnostic. The output continuously reports the temperature of the two devices (in °C).

- Basic Sanity Tests

  Description: This test runs a selection of the available diagnostics in the following order:

  - Memory test on SDRAM
  - Memory test on SRAM
  - Segment LED test
  - PHY Register test
  - Read an internal Register from the IOC80314. This diagnostic is used for basic board checkout to verify that the board is healthy.

- Loop Version of the Basic Sanity Tests

  Description: This diagnostic is used to continuously loop through the basic sanity tests and keep track of any failures that it encounters. The test can be stopped between any of the repeat loops.

- EEPROM Menu

  Description: Selecting this diagnostic allows the user to read, write, and erase the Boot EEPROM on the IQ80315 CRB.

- RTC Menu

  Description: This diagnostic allows the programming and reading of the RTC on the I^2C interface. The RTC is not setup and calibrated by RedBoot, so calibration is left as an exercise for the user.

- Fan Menu

  Description: This test permits the enabling and disabling of the fans on the IQ80315 CRB. There is no support for throttling the fans; they are either turned completely on or completely off.

- Test Buzzer

  Description: The final test in the diagnostic suite outputs a square wave from the RTC to the buzzer or speaker. No musical tunes have been programmed in RedBoot.

4.3 Operating System Support

The IQ80315 CRB supports the following operating systems:

- NetBSD*
- Linux*
- VxWorks*

4.4 Debug Software

Several software debug solutions are available for the IQ80315 CRB. Among them are:

- code|lab* Debug software: Allows the downloading and debugging of applications through the JTAG interface. The FLASH memory of the IQ80315 CRB can also be programmed through code|lab.
- GDB–GNU* Debugger: Allows the download and debug of application through the serial link or through Ethernet Port0.
- OCD Commander*: A simple, free JTAG debugging tool that allows download and debug of applications through the JTAG interface.

When compiling an application for debug on the IQ80315 CRB, the RedHat* GNUPro* tools available on the Intel Developer website (http://developer.intel.com) can be used to compile, assemble, and link an application. The linker specifications file “redboot.specs” can be used with the GNUPro toolchain to link the application to the correct location in SDRAM. The application can be downloaded through serial, Ethernet, or JTAG into the IQ80315 SDRAM. The debug software listed above can assist in debugging the application.

Appendix A Design Specifications

Table 24 lists the specifications applicable to the IQ80315 CRB.

Table 24. Design Specifications

Reference Name	Specification Title	Version, Revision Date, and Ownership	The information is available from…
ATX	*ATX Specification*	Version 2.03, December 1998, Intel Corporation.	http://www.formfactors.org
ATX12V Power Supply	*ATX/ATX12V Power Supply Design Guide*	Version 1.2, August 2000, Intel Corporation.	http://www.formfactors.org
microATX	*microATX Motherboard Interface Specification*	Version 1.0, December 1997, Intel Corporation.	http://www.formfactors.org
DDR SDRAM	*Double Data Rate (DDR) SDRAM Specification*	Version 1.0, June 2000, JEDEC Solid State Technology Association.	http://www.jedec.org/
	Design Specification for a 184 Pin DDR Unbuffered DIMM	Revision 1.0, October 2001, JEDEC Solid State Technology Association.	http://www.jedec.org/
	Intel® JEDEC DDR 200/266 Unbuffered DIMM Specification Addendum	Revision 0.9, September 27, 2001, Intel Corporation.	http://developer.intel.com/technology/memory/index.htm
	DDR DRAM SPD		
	PC Serial Presence Detect (SPD) Specification	Revision 1.2B, November 1999, Intel Corporation.	http://www.intel.com/technology/memory
Front Panel	*Front Panel I/O Connectivity Design Guide*	Version 1.0, October 2000, Intel Corporation.	http://www.formfactors.org
PCI	*PCI Local Bus Specification*	Revision 2.2, December 18, 1998, PCI Special Interest Group.	http://www.pcisig.com/specifications
	PCI Bus Power Management Interface Specification	Revision 1.1, December 18, 1998, PCI Special Interest Group.	http://www.pcisig.com/specifications

Appendix B Getting Started and Using the Debugger

B.1 Introduction

This appendix pertains to Code|Lab* version 2.3 and later, which uses Microsoft* Visual Studio.NET*.

B.1.1 Purpose

The purpose of this appendix is to help the user to set up and become familiar with the Intel® IQ80315 Customer Reference Board (IQ80315) and other related hardware and software. This appendix steps the user through an example program using:

- Code|Lab EDE
- Code|Lab EDE debugger
- Macraigor* Raven* JTAG

This programming also includes:

- software setup
- compiling
- linking
- debugging example code

The user tours the major features of the debugger and explores some of the basics of debugging. By the end of this exercise, the user will have been given a general understanding of the ATI* development tools and can begin working on new applications.

B.1.2 Hardware and Software Requirements

This example uses the ATI* Code|Lab* plug-in for Microsoft* Visual Studio*, the GNU* Pro compiler, the Macraigor* MPDemon* JTAG, and the IQ80315.

B.1.3 Related Documents

Table 25. Related Documents

Document Title	Document Number
Intel® GW80314 I/O Companion Chip Developer's Manual	273756
Intel® 80200 Processor based on Intel XScale® Microarchitecture Developer's Manual	273411
Intel® IQ80315 I/O Processor Evaluation Platform Board Manual	253794
Hot-Debug for Intel XScale® Core Debug White Paper	273539
ARM Assemblers Guide (http://www.arm.com/documentation)	
ADS Debug Target Guide (http://www.arm.com/documentation)	
Code|Lab Debug for ARM[1]	

Note:

[1] This document installs to `C:\Ati\docs\codelab debug.pdf`.

Many of these documents load as part of the ATI Code|Lab installation (Start/Programs/Accelerated Technology/Documentation). This menu contains both the ARM* ADS and Code|Lab* documents.

B.1.4 Related Web Sites

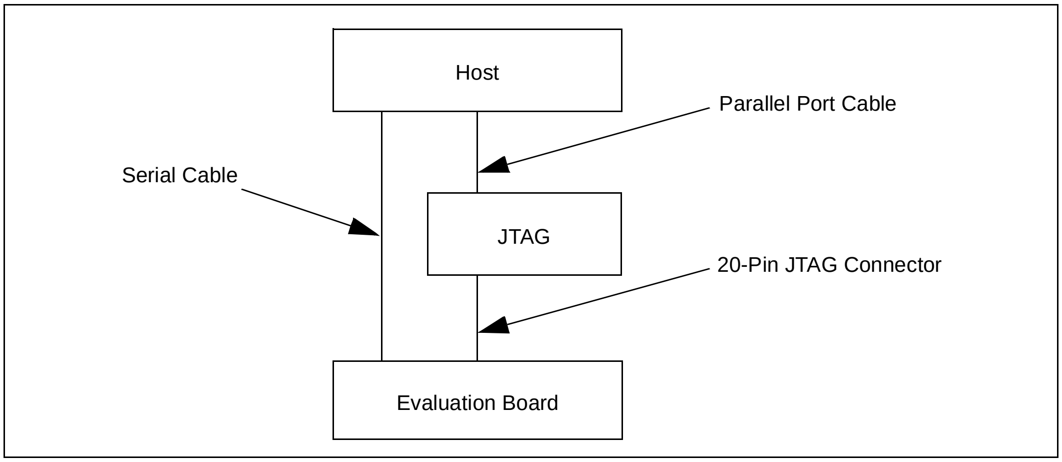
- Macraigor Systems*: http://www.ocdemon.net/
- Intel:
  — http://developer.intel.com/design/intelxscale/dev_tools/020523/index.htm
  — http://developer.intel.com/design/iio/iop315.htm
  — http://developer.intel.com/design/iio/docs/iop315.htm
  — http://developer.intel.com/design/iio/swsup/LED80315.htm

B.2 Hardware and Software Setup

B.2.1 Hardware Setup

Refer to Figure 15 and the rest of this appendix to set up the hardware.

Figure 15. Intel® IQ80315 Customer Reference Board Hardware Setup Flow Chart



- Connect the MPDemon to the host via the parallel port and to the evaluation board via the 20-pin JTAG connector.

Note: The parallel port must be configured to ECP mode for the Macraigor MPDemon to work properly. Also, Ethernet and serial connections are options. Refer to the MPDemon documentation to correctly configure the selected connection type. The parallel port setting can be changed in the BIOS setup program or in the Control Panel. More information on the MPDemon can be found at the Macraigor web site. Test software for the MPDemon is free and available for download at the following URL:

http://www.ocdemon.net/

- Connect a serial cable from the evaluation board to the host.

Note: The serial cable connects to the evaluation board with an RJ11-to-DB9F adaptor and connects to the host computer serial port via an RJ11-to-DB9F adaptor. The serial port configuration is discussed in Section B.4.2, “Using Flash Programmer” on page 59.

- The IQ80315 is a standalone platform. Either place the board on an insulating mat, or mount it in a proper enclosure to prevent shorting out the board. Refer to Figure 2, “Intel® IQ80315 Customer Reference Board Layout Diagram” on page 13 when setting up the board. Connect the ATX power connector, serial cable, and apply power. A power LED will light, but the board will not boot until pins 4 and 5 on the connector beside the reset button are shorted momentarily (for a fraction of a second). These pins are labeled PWRBT on Figure 2.

Note: There are many dip switches on the evaluation board. Refer to Section 2.14, “Switches, Jumpers, and Reset Configuration Strapping” on page 34 to configure these switches.

Check the system requirements of Microsoft* Visual Studio* and ATI* Code|Lab* to make sure that the host meets the requirements.

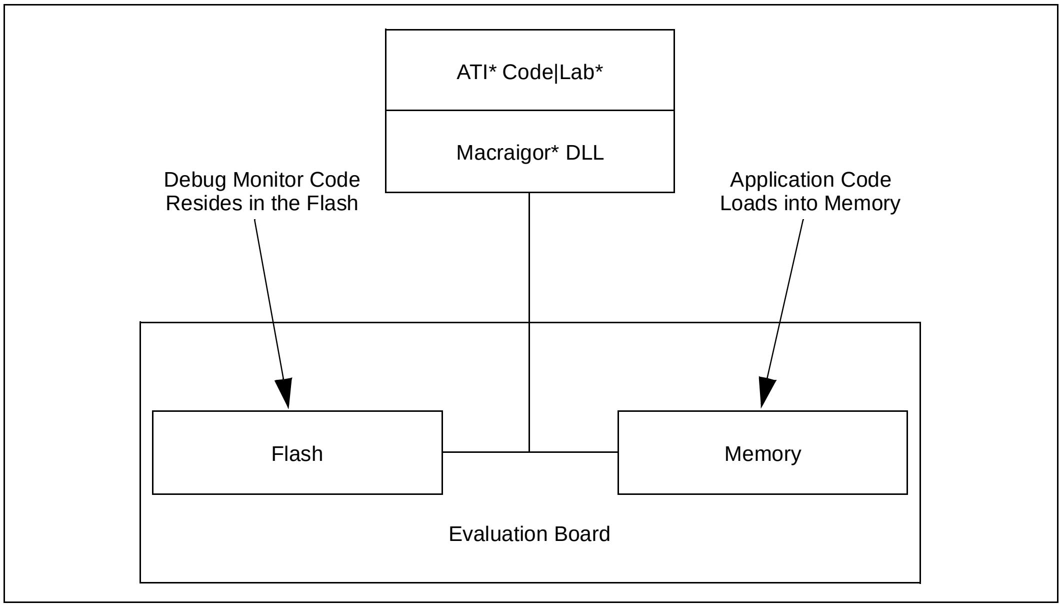
B.2.2 Software Setup

ATI* Code|Lab* is a plug-in for Microsoft* Visual Studio.NET*; therefore, Microsoft Visual Studio.NET must already be loaded on the system. To load ATI Code|Lab, run `setup.exe` under the program directory.

Note: Do not install over an older version of ATI Code|Lab. When necessary, uninstall the previous version of Code|Lab using “Add/Remove Programs” under “Control Panels” before reinstalling.

To view the Code|Lab documentation, Adobe* Reader* is required. The latest version can be downloaded at http://www.adobe.com/products/acrobat/readstep2.html.

Figure 16. Software Flow Diagram


ATI* Code|Lab*
Macraigor* DLL
Debug Monitor Code Resides in the Flash
Application Code Loads into Memory
Flash
Memory
Evaluation Board

B.3 New Project Setup

B.3.1 Creating a New Project

1. Launch Code|Lab EDE for .NET.
2. On the Start Page, select “New Project”.
   a. The “New Projects” window appears.
   b. Select “Code|Lab Projects” under Project Types, and name the project “Test80315” in the name field.

Note: The directory “Test80315” is created under the path specified in the Location box.

   c. Click OK.

3. In the Code|Lab EDE Project Wizard Window:
   a. Select “Redhat GNU Tools for XScale” under “Build Toolset”.
   b. Select “IQ80315” under “Project Template”.
   c. Select “Application” under “Project Type”.
   d. Click “Finish”.
4. Close the “Start Page” by clicking on the “X” in the top-right corner of the Start Page window.
5. The new project is now in the “Solution Explorer” window. If this window is not open, open it by “View, Solution Explorer”.
6. Right click on “Test80315” and select “Save Test80315”.
7. From http://developer.intel.com/design/iio/swsup/LED80315.htm, download the following zip file (…/LED80315) from the Software Support section to the newly created project folder. The zip file contains the example code files:

   LED80315.zip

   blink.c

   blink.h

   led.c

   led.h

   These files can be placed in any directory on the hard drive.
8. Add the newly downloaded files to the project:
   a. In the “Solution Explorer” window, right click on “Test80315” and select “Add”, “Add Existing Item”.
   b. In the “Add Existing Item” window, use the drop-down menu under “Look In” to find the four files listed in step 7 on the hard drive. Select all four files and click “open”. The “Solution Explorer” window now shows these files under “Test80315”.

B.3.2 Configuration

Examine the main menu of Code|Lab EDE for .NET.

- File
- Edit
- Project
- View
- code|lab EDE
- Build, Debug
- Tools
- Window
- Help
-

Since Code|Lab is a plug-in to Microsoft Visual Studio, some of these menu items are for Visual Studio and some are specific to Code|Lab. Click on any of these menu items, and the drop-down menu displays the subordinate menu items. Many of these items have defined tool-bar symbols, function keys, and keyboard patterns as alternatives.

Note: Projects can be built under the “code|lab EDE” menu or under the “build” menu. Always use the “code|lab EDE” menu to perform Code|Lab project builds. Builds under the “build” menu invoke the Microsoft Visual Studio C-compiler.

1. On the main menu, select “code|lab EDE, Configuration”.
2. When the “code|lab EDE Configuration” window appears, click on each of the words in the left box. Notice that the rest of the window changes when you click on different parts of the menu tree. This is a typical feature of Code|Lab EDE for .NET.
3. Click on Toolsets.
4. Click on the drop-down arrow and select “RedHat GNU Tools for XScale”. The build tool paths now appear in the box and must be modified as stated below in ***bold italic***. Note that the assembler and the linker are invoked by GCC.
    a. “Compiler path: $(ToolDir)\BIN\XSCALE-ELF-GCC.EXE”.

    b. “Assembler path: $(ToolDir)\BIN\XSCALE-ELF-***GCC***.EXE”.

    c. “Linker path: $(ToolDir)\BIN\XSCALE-ELF-***GCC***.EXE”.

    d. “Librarian path: $(ToolDir)\BIN\XSCALE-ELF-AR.EXE”.
5. Click “Apply”, and click “OK”.
6. On the main menu, click “code|lab EDE, Project Settings”.
7. When the “code|lab Project Settings” window appears, click on “C/C++/Assembler” in the left box. Use the drop-down arrow to select “C compiler” for “Build Tool”.
8. Edit the command line box at the bottom so that it contains the following:

    -v -specs=redboot.specs -gdwarf-2 -O0 -c -mcpu=xscale $(InputRelPath)
    -o $(OutDir)\$(InputName)$(OutputExt)
9. Use the drop-down arrow to select “Assembler” for “Build Tool”. Edit the command line box at the bottom so that it contains the following:

    -v -specs=redboot.specs -o $(OutDir)\$(InputName)$(OutputExt) $(InputRelPath)
10. In the left box, click on “Linker”. Edit the command line box at the bottom so that it contains the following:

    -v -specs=redboot.specs -o $(OutDir)\$(ProjectName).elf $(ObjectFiles) $(Libraries)
11. Click “Apply” and then click “OK”.
12. Under “Project Settings”, select Code|Lab Debugger ARM”. Set all four debug options to “false”.
13. In the “Solution Explorer” window, right click “Test80315” and select “Save Test80315”.

B.4 Flashing with JTAG

B.4.1 Overview

Code|Lab and MPDemon are capable of reading from, writing to, and erasing the contents of the Flash on the evaluation board. The board comes with RedBoot* loaded in the Flash. RedBoot is the RedHat* debug monitor that initializes the board; it also has some debug and diagnostic functions. It is capable of serial communication with the console of a debug program or with Microsoft* HyperTerminal*, and it prepares the board for accepting an application program.

Code|Lab invokes a Flash programmer written by Macraigor. More information on the Flash programmer is located at the following URL:

http://www.ocdemon.net

This Flash programmer supports only certain file formats:

- Intel® Hex
- Motorola* srec
- Standard elf (executable and linking format)

RedBoot.s19 and RedBoot.srec are both srec files. Worcester.i32 is an ARM* BootMonitor* Intel® Hex file. BootMonitor is an ARM version of a debug monitor, which is similar but not identical to RedBoot.

Macraigor offers conversion tools to convert existing file types to a supported file type. These conversion tools are located at the following directory path:

```
C:\MGC\codelab\codelab Debug\Macraigor\Flash Programmer
```

The `ReadMe.txt` file describes the conversions tools. The `BinToS19.exe` executable converts binary files to srec files, and `MakeIntelHex.exe` converts `a.out` files to Intel® Hex files. When using the `BinToS19.exe` conversion tool, use 0x0 for the starting address. For example, at the CMD prompt in the directory where `BinToS19.exe` is located, the command line looks like this:

```
C:\ATI\codelab\codelab Debug\Macraigor\Flash Programmer>bintos19

C:\temp\redboot_ROM.bin 0x0 c:\temp\redboot_ROM.s19
```

B.4.2 Using Flash Programmer

Note: The parallel port must be set to ECP mode or the Macraigor MPDemon will not work properly. Refer to the MPDemon manual to check the setting for the connection type selected. This example assumes a parallel connection with the MPDemon. ECP mode must be the selection for the parallel port.

Download the RedBoot executable files as follows:

1. Go to the following URL:
   http://developer.intel.com/design/intelxscale/dev_tools/021022/index.htm
2. Scroll down to find and download *Redboot Debug Monitor for the Intel® IQ80331 board.*

After downloading:

1. Double click on the “Code|Lab Debug” icon on the desktop. The Connection Window appears.
2. Select “Macraigor JTAG Connection”.
3. Click “Setup”.
4. Select “MPDemon Parallel” (note the other connection options).
5. Do not check “Script Options” or “Enable Hot Debug”.
6. Click “Configure”.
7. Select “Parallel Connection” and “LPT1”.
8. OCD Speeds must be “2” in both cases.

Note: The IQ80315 works only with speeds 2 through 7. Speed 1 and speed 8 do not work.

9. Click “OK”.
10. Select “XScale 80200 (80315)”.
11. Click “Additional Options”.
12. Do not check “Enable Option” under “Scan Chain Options”.
13. Do not check “Enable Option” under “Console Options”. Instead, open HyperTerminal* (or some equivalent serial terminal program). Select COM1 or COM2 as appropriate.
    Baud Rate: 115200
    Data Bits: 8
    Parity: None
    Stop Bits: 1
14. Then, back in Code|Lab, press OK, OK, OK (this returns to the Connect window).
15. Now press “Connect”.
    Assembly code is now visible.
16. Select “Memory/Flash…”
    The OCDemon Flash Memory Programmer window appears.
17. The Flash programmer needs a file which is architecture-specific, in this case. In the Flash programmer window, select “File/Open”, then choose the file “Xscale TBD.ocd”, located at `C:\MGC\Embedded\codelab\codelab Debug\Macraigor\`.

18. Click the “Program” button.
19. Click “Browse”, and under “Files of type:” choose “All Files”.
20. Choose the “redboot_ROM.srec” file (downloaded and uncompressed from http://developer.intel.com).
21. Check the box “Erase Target Flash Sector(s) Before Programming”.
22. Click “Program”.

    The Flash now programs and verifies; click “Close” when 100% complete.
23. Cycle power to the board to verify that the LEDs on the board sequence “SL”, then “A1”. This is the normal LED sequence of RedBoot. The board might need to be reset more than once.

Explore the other features of the Flash programming window. The contents of the Flash can be erased, copied to a file on the host, and verified against a file on the host.

B.5 Debugging Out of Flash

JTAG debuggers can be used on two levels: with or without the source code. When the Flash is programmed, the debugger can monitor the executable code, halt it, step through it, and monitor the memory and registers. The executable code is disassembled so that the assembly code can be examined.

Debugging with source code allows the user to examine the C code that is being executed. This requires that the source code is available and linked by the debugger to the executable code that is running on the evaluation board.

B.6 Building an Executable File from Example Code

1. Launch Code|Lab EDE and open “Test80315”.
2. Select “code|lab EDE, Rebuild Project”.

Note: A project can normally have more than one solution. However, in this example, there is only one solution for the project, so there is no difference between “Build Project” and “Build Solution” in this example.

Note: Rebuild cleans and builds. Clean deletes the old `.o` files in the project and build compiles, links, and produces the executable files.

3. When there are errors, carefully go back through Section B.3.2, “Configuration”.

B.7 Running the Code|Lab* Debugger

This section is provided to get the system up and running in the Code|Lab* Debug environment, but it is not intended as a fully-functional tutorial. Refer to the *ATI* Code|Lab Debug Reference Manual* for more detailed information.

B.7.1 Launching and Configuring the Debugger

1. In EDE, click on the icon that looks like a red bug. The “Connect” window appears.
2. When not configured from Section B.4.2, “Using Flash Programmer”, go to Section B.4.2 and perform steps 2–5.
3. Press Connect to enter debug mode.
   a. The Code|Lab Debug environment appears with the Assembly window open.

Note: Mouseovers are available for most of the toolbar icons. (Leave the mouse over the debug icons across the top on the toolbar to see a brief explanation of each.)

4. Click on the go icon and let RedBoot boot (takes a minute) until the RedBoot prompt “RedBoot>” appears in the Console window (click the Console tab at the bottom of the Debug window to view the Console window).
5. From the console window:
   a. Type “diag”.
   b. Press “Enter”.

   The RedBoot Diagnostic function is invoked.

   Try out a few of the tests as desired.
6. Close the Debugger and EDE environment.
7. Reset the board (cycle power).

B.7.2 Manually Loading and Executing an Application Program

1. Launch the Code|Lab Debug Environment from the desktop icon.
2. Ensure “File…/Program Load Options/Load Executable and Symbols” is checked.
3. Select menu File, Program Load Options, Load Executable and Symbols.
   a. Select “File, Open Program, Browse”.
   b. Browse to find `c:\…\Test80315\O\Test80315.elf`.
4. Click “Go” (80, 03, 31, and 15 cycle on the LEDs).
5. Cycle power on the board.

B.7.3 Displaying Source Code

1. Launch the Code|Lab EDE Debugger and open the “Test80315” ELF program.

Note: Use the File/Recent Programs menu for quick access.

2. Select the “Files” view in the lower tab of the WorkSpace window.
3. Bring up “blink.c” and “led.c” source code by double-clicking each filename.
4. Use the “Windows” Menu to arrange the windows, or maximize, minimize, and resize manually as desired.
5. Press the “Mixed” tab at the bottom of the “blink.c” window. Notice that the assembly code appears along with each C statement.
6. Press the “Source” tab to revert back to C code only.

B.7.4 Using Breakpoints

Note the small gray circles on the sidebar beside each line of source code. Single-click any of these gray circles, and a red dot appears. The red dot represents a break point. Single-click the red dot to remove it, or click the “Remove all breakpoints” icon to remove all.

Place a breakpoint on the following lines of code in “blink.c”:

```
displayLED(leds[8],leds[0]); /* LED display '80' */

displayLED(leds[0],leds[3]); /* LED display '03' */

displayLED(leds[3],leds[1]); /* LED Display '31' */

displayLED(leds[1],leds[5]); /* LED display '15' */

displayLED(leds[16],leds[16]); /* LED display ' ' */
```

1. Click the “Go” icon.

   The yellow arrow stops at the first break point and the HEX display does not change.
2. Click the “Go” icon again.

   The last instruction has now been executed and an “80” is displayed.
3. Continue in this fashion, watching the lines execute only as they are called, while the yellow arrow shows exactly what line is to be executed next.
4. Click the “Remove all breakpoints” icon.
5. Press “Go” again, and notice that the program loop is infinite.
6. Press the “Halt” icon to stop execution.
7. Close the debugger, and cycle power to the board.

B.7.5 Stepping through the Code

The led.c file contains a function that is called from code in blink.c. This exercise steps through the code and utilizes a few of the most common step tools.

1. Launch the debugger, open “Test80315”, and open the blink.c and led.c files.
2. Set a breakpoint on the following line in blink.c:

   displayLED(leds[8],leds[0]); /* LED display '80'*/
3. Press Go. The program execution stops on the breakpoint.
4. Press the “Step Over” icon and notice how execution jumps over the function call to the next line of execution.
5. Now try the “Step Into” icon, and note that the pointer has now jumped into the function “displayLED”, located in the led.c file.
6. Press the “Step Over” icon again, and watch the pointer advance within the function to the next executable line.
7. Now press the “Step Out of” icon, and notice how execution leaves the called function and waits on the next executable line in blink.c.
8. The “animate” icon can also be used to provide a “Step Into” effect that occurs at a specified time interval (default of 1 second). This can be modified in the “Settings” section of the “View/Options” menu. Experiment with this as desired.
9. Use Halt to stop the animate mode before the next breakpoint.
10. Also note that Go can be pressed at any time to continue execution from the current line to the next breakpoint or to program end.

B.7.6 Setting Code|Lab* Debug Options

In addition to the “Animate” debug time interval setting briefly mentioned in step 8 of the previous exercise, many useful options can be accessed from the “View/Options” menu.

1. Experiment here by bringing up the “Registers” window (click and change the view options between binary and decimal; for example).

   Hint: Settings tab, Interface, Radix
2. Also try bringing up the Memory window (click) and change the number of columns between 4 and 2, and notice the changes.

   Hint: Settings tab, Memory Window, Number of Columns

   Press window icons a second time to remove them from view.

Again, there are many features of the debug environment not discussed here. Please see the Code|Lab manuals for a full description of debug features.

B.8 Exploring the Code|Lab* Debug Windows

This section discusses some basics of the debug environment. Some of these windows and concepts have been addressed during previous exercises in this manual. However, many new windows are also discussed, and basic interaction exercises are given. Begin this section by launching the Code|Lab Debugger environment, and connect via the JTAG port.

B.8.1 Toolbar Icons

Placing the mouse arrow on any icon displays the text function of that icon. When the icon launches a special window (for example, Watch, Memory, Call Trace, and so on), the icon activates that window on the first click and removes the window when pressed again.

B.8.2 Workspace Window

Click on the “Workspace” icon. Click on the “Files” and “Browse” tabs, and examine the contents. Note that there are more files than the original source files. When you double-click on the source files, `blink.c` and `led.c`, the source window appears for that file. When you double-click on an included file, the debugger is not able to find the file.

B.8.3 Source Code

The source code windows are opened by double-clicking on the source files in the “Workspace” window under the files tab. Choosing the view (mixed assembly and C code, or C code only) is controlled by the tabs at the bottom of these windows.

B.8.4 Debug and Console Windows

The “Debug” window displays debugger activity messages when the “Debug” tab is displayed. Script commands can be entered manually at the top of the window. Serial output is displayed when the “Console” tab is active. Commands for the running application can be entered at the top of this window.

B.8.5 Memory Window

Click on the “Memory” window icon. Change the address at the top of the window to 0x50107430, and click on the green arrow to the right (or press Enter). This changes the viewable starting address of the Memory window. The registers used for setting up the timers are displayed. Change the address at the top of the window to 0x40000000, and click on the green arrow to the right (or press Enter). The vector table at the beginning of flash is displayed.

Note: The tabs at the bottom allow you to select two memory regions to observe.

B.8.6 Registers Window

Close all the active windows, then bring up the “Registers” window. Resize this window and its columns to get a good view of all the registers.

Notice that there is a “Flags” tab at the bottom of this window. This is useful for seeing the system flags defined by the CPSR. These are important especially during conditional code execution (see the *ARM Architecture Reference Manual* for more detail), but the flags are not changed during this exercise.

Click on the “Registers” tab of the Registers window, and click the Animate icon. Notice how the register values change during program execution (red values are those that were modified during the last execution cycle). Click the Halt icon at any time, then try right-clicking a register row and selecting “Go To Memory”. Notice how the Memory window is brought up and the address contained in that register is shown.

Click on the “Registers” tab. Red means that the register value changed since the last fetch, as opposed to black which represents no change. Register values can be changed manually in this window.

B.8.7 Watch Window

It is often useful during the debugging process to observe a few select program variables.

1. Open the “Test80315” program, and bring up “led.c”.
2. Click the “Watch” icon to bring up the Watch window.
3. Now add the “left” and “right” variables from “led.c” to the watch window.

Note: For each variable, double-click the variable name to highlight it, then drag it to the watch window.

4. Click the “Animate” icon and observe the changes.

Note: During this process, when the focus goes back to the Assembly window, try putting a breakpoint in `led.c`, then hit Go.

B.8.8 Variables Window

The “Variables” window behaves very similarly to the “Watch” window, except that it shows all active variables.

Bring up the Variables window, click Animate, and watch the changes.

B.9 Debugging Basics

B.9.1 Overview

Debuggers allow developers to interrogate application code by allowing program flow control, data observation, and data manipulation. The flow control functions include the ability to single-step through the code, step into functions, step over functions, and run to breakpoint (hardware or software). The data observation and manipulation functions include access to memory, registers, and variables. The combination of the flow control and data functions allows the developer to debug problems as they occur or to validate the application code. As the size of an application grows, the need to be able to isolate the cause of a problem to a few lines of code is imperative.

Debuggers have a finite set of capabilities and limitations. Debuggers can give insight that is difficult to obtain without them, but they can fail when they are not used within the limits of their functionality. They are intrusive by definition. They are software programs that interact with software monitors or hardware (JTAG) to control a target program. Ultimately, the debugger works best when the developer understands what it can and cannot do and uses it within those constraints.

B.9.2 Hardware and Software Breakpoints

The following section provides a brief overview of breakpoints. For more detailed information, see the *Intel® 80331 I/O Processor Developer's Manual*.

B.9.2.1 Software Breakpoints

Software breakpoints are set up and utilized via debugger utilities (such as Code|Lab*). The abilities of software breakpoints are discussed in Section B.7, "Running the Code|Lab* Debugger" on page 61. Program execution can be halted at a particular line of code, stepped through, and executed again to the next breakpoint via debuggers. During this process, register values, memory address contents, variable contents, and many other useful pieces of information can be monitored.

B.9.2.2 Hardware Breakpoints

Hardware breakpoints step and breakpoint in code in either ROM or RAM without altering the code, stacks, or other target information. Hardware breakpoints handle difficult issues, by providing the ability to set the processor conditions that cause the program to halt. Use hardware breakpoints to locate problems such as reentrance, obscure timing, and so on.

The 80331 contains two instruction breakpoint address registers (IBCR0 and IBCR1), one data breakpoint address register (DBR0), one configurable data mask/address register (DBR1), and one data breakpoint control register (DBCON). The 80331 also supports a 256-entry trace buffer that records program execution information. The registers for controlling the trace buffer are located in CP14.

B.9.3 Exception Trapping

A debug exception causes the processor to re-direct execution to a debug event-handling routine. The Intel® 80200 processor debug architecture defines the following debug exceptions:

- Instruction breakpoint
- Data breakpoint
- Software breakpoint
- External debug break
- Exception vector trap
- Trace-buffer full break

When a debug exception occurs, the processor actions depend on whether the debug unit is configured for Halt mode or Monitor mode.

§ §

THIS PAGE INTENTIONALLY LEFT BLANK