QL902M QuickMIPS™ Data Sheet

QuickLogic

QuickMIPS Embedded Standard Product (ESP) Family

Device Highlights

CPU Core

- 32-bit MIPS 4Kc processor runs up to 200 MHz (260 Dhrystone MIPS)
- 1.3 Dhrystone MIPS per MHz
- MDU supports MAC instructions for DSP functions
- 16 KB of instruction cache (4-way set associative)
- 16 KB of data cache (4-way set associative), lockable on a per line basis

SDRAM Memory Controller

- Support for PC-100 type SDRAMs, up to 256 MB total
- Two chip selects
- Operates at up to one-half CPU pipeline speed
- Support for x16 and x32 external memory bus configurations

I/O Peripheral Controller

- Direct support for SRAM, EPROM and Flash
- 8-bit, 16-bit and 32-bit device widths supported
- Eight independent chip selects

PCI Controller

- 32-bit v2.2 compatible
- Up to 66 MHz operation
- Supports host and satellite configurations
- Dedicated DMA channels for transmit and receive bus transactions
- Support for external bus master arbitration (through FPGA library provided by QuickLogic)

Two Ethernet Controllers

- Two 10/100 MACs
- Provides MII connection to external transceivers/devices

Two UARTs

- One with modem control signals
- Both with IRDA-compliant signals

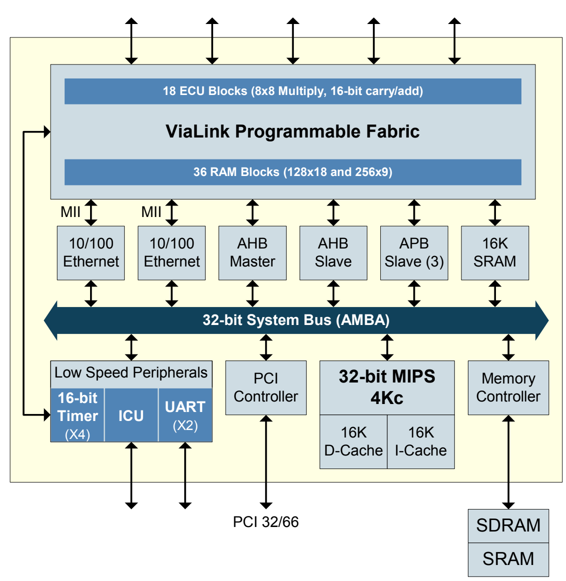
Four General Purpose 16-bit Timer/Counters

- 16-bit prescaler to increase timer/counter delay
- Four modes of operation: decrement, increment, interval, and Pulse Width Modulation (PWM)
- Operation from the System Bus clock or a clock source supplied from the Programmable Fabric

System SRAM

- 16 KB accessible by all System Bus masters or the Programmable Fabric

Figure 1: QL902M Block Diagram



High Performance 32-bit System Bus (AMBA Bus)

- Operates at one-half, one-third, or one-fourth of CPU pipeline speed
- One 32-bit AHB master port/one 32-bit AHB slave port to programmable Fabric
- Three 32-bit APB slave ports in the programmable Fabric

Flexible Programmable Fabric

- 2016 logic cells (533 K system gates)
- 152 I/O pins
- 1.95 V Vcc, 1.8/2.5/3.3 V drive capable I/O
- 4,284 dedicated flip-flops
- IEEE 1149.1 boundary scan testing compliant

Dual-Port SRAM Modules

- Thirty-six 2,304 bit Dual-Port High Performance SRAM Blocks
- 82,944 embedded RAM bits
- RAM/ROM/FIFO Wizard for automatic configuration
- Configurable and cascadable

Programmable I/O

- High performance I/O cell with fast clock-to-out time
- Programmable Slew Rate Control
- Programmable I/O Standards:
  - LVTTL, LVCMOS, LVCMOS18, PCI, GTL+, SSTL2, and SSTL3
  - Independent I/O Banks capable of supporting multiple standards in one device
  - I/O Register Configurations: Input, Output, Output Enable (OE)

Advanced Clock Network

- Multiple dedicated Low Skew Clock Networks
- High drive input-only networks
- Quadrant-based segmentable clock networks
- User-programmable Phase Locked Loop (PLL) circuit

Embedded Computational Units (ECUs)

Eighteen hardwired DSP building blocks with integrated Multiply, Add, and Accumulate functions.

Security Features

The QuickLogic products come with secure ViaLink® technology that protects intellectual property from design theft and reverse engineering. No external configuration memory is needed for the Fabric. The device is instant-on at power-up.

QuickWorks Design Software

The QuickWorks® package provides the most complete ESP and Field Programmable Gate Array (FPGA) software solution from design entry to logic synthesis, to place and route, and simulation. The package provides a solution for designers who use third party tools from Cadence, Mentor, Synopsys, and other third-party tools for design entry, synthesis, or simulation.

Process Data

The QL902M is fabricated on a 0.18 μ, six layer metal CMOS process. The core voltage is 1.95 V V_{CC} supply and the I/Os are up to 3.3 V compliant. The QL902M is available in commercial and industrial temperature grades.

QL902M Architectural Overview

The QL902M chip can be thought of as having two distinct *sides*, an Application Specific Standard Product (ASSP) side and a Programmable Fabric side. The ASSP side contains the standard cell circuitry of the device such as the MIPS 4Kc CPU and the Ethernet MACs, and the Fabric side contains all of the programmable logic elements (e.g., logic cells and dual-port RAMs) of the device.

ASSP Side

This section discusses the various circuits in the ASSP portion of the QL902M device.

CPU Core

The MIPS32 4Kc processor core is a high-performance, low-power, 32-bit MIPS RISC core capable of speeds up to 200 MHz. The 4Kc core contains a fully-associative translation lookaside buffer (TLB) based Memory Management Unit (MMU) and a pipelined MDU.

The core executes the MIPS32 instruction set architecture (ISA). It supports all application code in the MIPS I, II, III, and IV instruction sets. It also supports kernel code for the R4000 processor and above. The MIPS32 ISA contains special multiply-accumulate, conditional move, prefetch, wait, and zero/one detect instructions. The MMU contains a three-entry instruction TLB (ITLB), a three-entry data TLB (DTLB), and a 16 dual-entry joint TLB (JTLB) with variable page sizes.

The 4Kc multiply-divide unit (MDU) supports a maximum issue rate of one 32x16 multiply (MUL/MULT/MULTU), multiply-add (MADD/MADDU), or multiply-subtract (MSUB/MSUBU) operation per clock, or one 32x32 MUL, MADD, or MSUB every other clock.

Instruction and Data Caches

The instruction and data caches are both 16 Kbytes in size. Each cache is organized as four-way set associative. The data cache has lockout capability per cache line. On a cache miss, loads are blocked only until the first critical word becomes available. The pipeline resumes execution while the remaining words are being written to the cache. Both caches are virtually indexed and physically tagged. Virtual indexing allows the cache to be indexed in the same clock in which the address is generated rather than waiting for the virtual-to physical address translation in the MMU.

EJTAG Interface

The basic Enhanced JTAG (EJTAG) features provide CPU run control with stop, single stepping and re-start, and software breakpoints through the SDBBP instruction. In addition, instruction and data virtual address hardware breakpoints, and connection to an external EJTAG probe through the Test Access Port (TAP) is included.

ASSP PLL

On the ASSP side of the QL902M there is a single clock input that provides an input clock reference for the MIPS core, the System Bus, and all ASSP peripherals (other than the PCI Controller, which is independently driven by the PCI_CLK input). This clock input (CPU_PLL_CLKIN) is the input to a PLL that is fixed at an 8 times clock multiplication rate. For example, if the clock rate applied to CPU_PLL_CLKIN is 25 MHz, the resultant clock that drives the MIPS core is 200 MHz. **Table 1** shows the maximum input clock rates for CPU_PLL_CLKIN based upon the ASSP speed grade of the given QL902M device.

Table 1: Maximum Input Frequency for CPU_PLL_CLKIN and MIPS Core Frequency Based on QL902M ASSP Speed Grade

QuickMIPS Device Part Number Prefix	Maximum Input Frequency for CPU_PLL_CLKIN	Resultant Maximum MIPS Core Frequency
QL902M175	21.875 MHz	175 MHz
QL902M200	25.000 MHz	200 MHz

The System Bus clock can run at a maximum rate of one-half the MIPS core clock frequency. Other ratios (one-third and one-fourth) are also possible and controlled by the CPU_PLL_DIV(1) and CPU_PLL_DIV(0) inputs as shown in **Table 2**.

Table 2: MIPS Core Clock Rate to System Bus Clock Rate (hclk) Ratio Based on CPU_PLL_DIV(1) and CPU_PLL_DIV(0) Signals

CPU_PLL_DIV(1)	CPU_PLL_DIV(0)	Ratio MIPS Core Clock Rate : System Bus Clock Rate (hclk)	System Bus Clock (hclk) Duty Cycle
0	X	2:1	50%
1	1	3:1	33%[a]
1	0	4:1	50%

a. In 3:1 mode, the System bus clock duty cycle is not symmetric. This affects the internal System Bus clock as well as hclk and the SDRAM clock source (SD_CLKOUT). Therefore, care must be taken so that minimum clock pulse widths are not violated when these clock signals are used to drive externally connected devices.

SDRAM Memory Controller

The QL902M SDRAM Memory Controller (SDMC) provides all the necessary logic to connect to a wide variety of industry standard SDRAMs for use by the CPU, Ethernet Controllers, PCI Controller, and Programmable Fabric. The SDMC supports a minimum SDRAM size of 16 Mbytes and a maximum SDRAM size of 256 Mbytes.

The SDRAM Controller controls the SDRAM on the external bus. On receiving an access request, the SDRAM Controller decides on the appropriate commands to send to the SDRAM memory. The DRAM Bank Controller sequences all of the commands required to complete a read or write request to an SDRAM memory location with timing controlled by the CAS Delay and RAS Delay values.

The bus interface is a slave on the System Bus; it contains the control register block. The bus interface produces read, write, refresh and mode register write requests to the SDRAM control engine, and software supplied configuration information.

Data is transferred to and from the SDRAM as unbroken quad words. This data packet size is convenient for cache line fills and buffered writes. For accesses smaller than a quad word, extra read data is ignored by the SDRAM Controller; for writes, the SD_DQM(3:0) pins are used to force the SDRAMs to ignore invalid data. For access sizes larger than a quad word, multiple quad word accesses are issued to the SDRAM control engine.

I/O Peripheral Controller

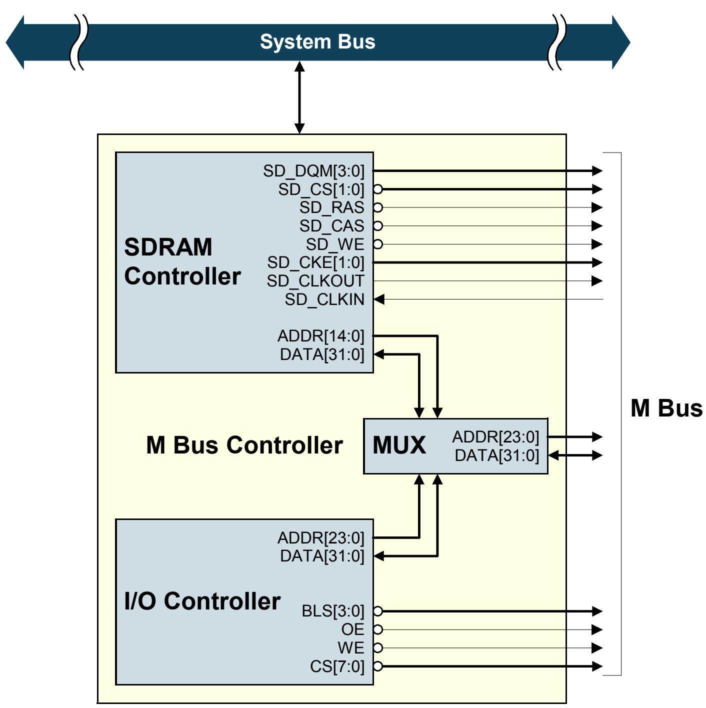
This section describes access to I/O and memory devices on the external M Bus (with the exception of SDRAM). The I/O Peripheral Controller (**Figure 2**) generates strobes and signals that can be used to interface the M Bus with common asynchronous peripheral devices.

The QL902M Peripheral Controller Unit (PCU) provides decoded strobe signals to control external peripherals such as SRAM, flash, real time clock (RTC) and memory mapped I/O devices. It supports 8-bit, 16-bit, and 32-bit widths with programmable wait states and bus turnaround time based on memory speed. The PCU provides the following functionality:

- Decoding of memory access in the local CPUs memory map to generate chip selects or strobes.
- Control of wait states for decoded regions. A total of eight chip select signals are available. Chip select seven is used as the boot ROM chip select.

The M Bus is a shared resource between the SDRAM Controller and I/O Controller. The M Bus is assigned to one of these two controllers by an internal arbiter. There is one turn-around cycle when switching from one controller to the other.

Figure 2: SDRAM and I/O Controllers



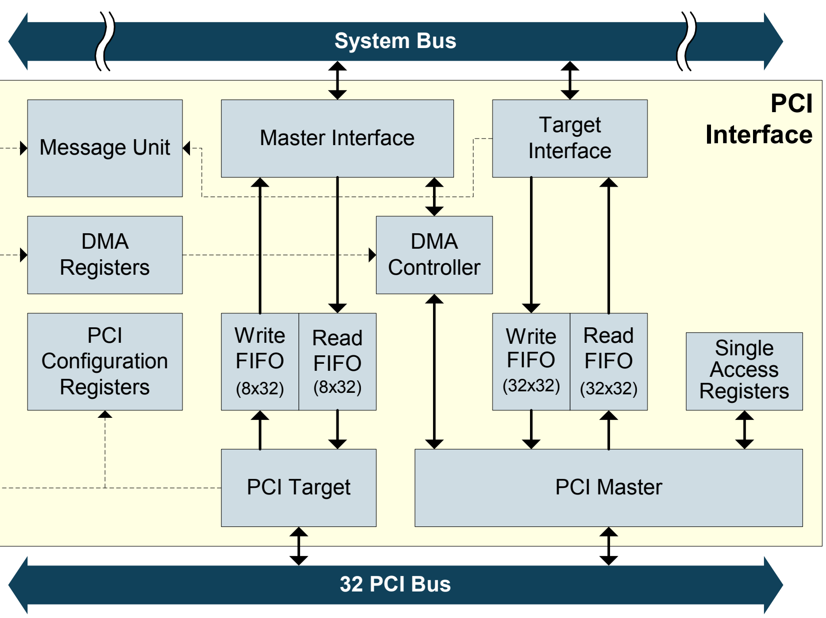
PCI Controller

The QuickMIPS PCI Interface is a 32-bit, 66 MHz, Revision 2.2 compliant interface as specified by the PCI Local Bus Specification. This section provides a detailed description of the PCI Controller functionality.

Figure 3 shows a simplified block diagram of the PCI Controller. The PCI Controller is a bridge between the on-chip System Bus and the external PCI bus. There are two main modes of operation of the PCI function, each utilizing several resources:

- PCI Target
  - PCI Target access to System Address Space
  - PCI Target to PCI configuration registers
  - PCI Target to extended registers (DMA and message/mailbox)
- PCI Master
  - System Address Space to PCI access
  - System-to-PCI and PCI-to-System DMA
  - PCI master read/write generated through the single access register

Figure 3: PCI Block Diagram



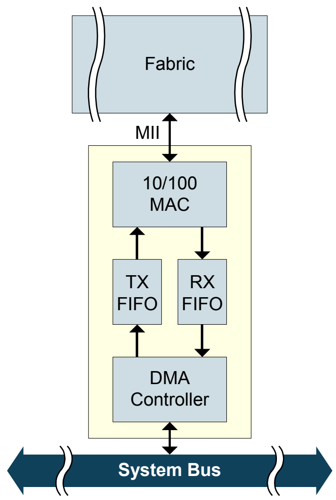
The PCI Controller can be used as a system host or as a secondary master:

- System Host: The QL902M PCI interface is connected to the root PCI bus segment (Bus 0) and the on-chip CPU performs bus enumeration to determine how all PCI Base Address Registers are configured. The SDRAM controlled by the QL902M is used as the main memory.
  - The Single Access Controller can be used for performing enumeration upon system boot, and then to perform I/O operations to I/O mapped PCI devices.
  - The System Bus to PCI Memory Aperture can be used by host drivers to control memory mapped PCI devices (PIO).
  - The DMA Controller provides the most efficient way to transfer large blocks of data between system memory and external PCI target devices (such as video controllers, etc.).
- Secondary Master: The QL902M is enumerated by an external PCI host CPU.
  - The PCI configuration registers of the QL902M are recognized by the host CPU and enumerated accordingly.
  - The DMA Controller can be used to efficiently move data between local QL902M-controlled SDRAM and the host memory.
  - The Single Access Controller can be used to transfer small amounts of data in and out of the host memory space.
  - The message unit can be used as an inter-CPU communication mechanism for a host driver to synchronize with a local driver running on the QL902M CPU.
  - The PCI to local System Bus aperture can be used to allow the host CPU or other PCI agent to access QL902M memory directly.

Ethernet Controllers

The QL902M has two Ethernet Media Access Controllers (MACs) embedded in the ASSP portion of the device. The Ethernet Controllers incorporate the essential protocol requirements for operation of Ethernet/IEEE 802.3 compliant nodes, and provide interfaces between the host subsystem and the Media Independent Interface (MII). Each 10/100 MAC can operate in 10 Mbps or 100 Mbps mode based on the transmit and receive clocks provided (2.5/25 MHz). The controllers contain transmit and receive FIFOs and embedded DMA control. **Figure 4** shows a block diagram of the QL902M Ethernet Controller.

Figure 4: Ethernet Controller Block Diagram



The DMA Controller is responsible for exchanging data between the FIFOs and the system memory. DMA operation is controllable through a set of control and status registers.

Each 10/100 MAC operates in half-duplex mode and full-duplex modes. When operating in the half-duplex mode, the 10/100 MAC core is fully compliant to Section 4 of ISO/IEC 8802-3 (ANSI/IEEE Standard) and ANSI/IEEE 802.3. When operating in the full-duplex mode, the 10/100 MAC core is compliant to the IEEE 802.3x standard for full-duplex operations. Each 10/100 MAC is also compatible with Home PNA 1.1.

The 10/100 MAC core provides programmable enhanced features designed to minimize host supervision, bus utilization, and pre- or post-message processing. These features include the ability to disable retires after a collision, dynamic FCS generation on a frame-by-frame basis, automatic pad field insertion and deletion to enforce minimum frame size attributes, automatic retransmission and detection of collision frames.

The 10/100 MAC core can sustain transmission or reception of minimal-sized back-to-back packets at full line speed with an inter-packet gap (IPG) of 9.6 µs for 10-Mb/s and 0.96 µs for 100-Mb/s.

Data to/from the MAC is buffered in transmit/receive FIFOs. In the case of data received by the Ethernet MAC, the data is drained from the receive FIFO by the DMA Controller and stored to the specified target (typically the data is stored in SDRAM). For Ethernet transmit, the DMA Controller reads data from memory (SDRAM typically) and pushes it into the transmit FIFO. DMA operation is controllable through a set of control and status registers.

With the exception of the M1_RXCLK, M2_RXCLK, M1_TXCLK and M2_TXCLK signals, the MII interface is located on the ASSP/Fabric boundary internal to the device. If the designer wants to use one or both of the Ethernet MACs, the signal ports on this interface must be brought out to Fabric IO pins in the top-level Fabric design. **Table 3** shows the recommended Fabric IO pin locations for the MII interface pins.

Table 3: MII Interface Signals and Recommended Fabric IO Pin Locations

MII Interface Signal	MII Interface Direction[a]	Fabric Pin Signal	Fabric Pin Type	Fabric Pin Location
M1_COL	I	M1_COL	I	F1
M1_CRS	I	M1_CRS	I	G2
M1_MDC	O	M1_MDC	O	F3
M1_MDI	I	M1_MDIO[b]	I/O/Z	K4
M1_MDO	O			
M1_MDO_EN_N	O			
M1_RXD(0)	I	M1_RXD(0)	I	F2
M1_RXD(1)	I	M1_RXD(1)	I	E1
M1_RXD(2)	I	M1_RXD(2)	I	E2
M1_RXD(3)	I	M1_RXD(3)	I	D1
M1_RXDV	I	M1_RXDV	I	L4
M1_RXER	I	M1_RXER	I	D2
M1_TXD(0)	O	M1_TXD(0)	O	L3
M1_TXD(1)	O	M1_TXD(1)	O	C1
M1_TXD(2)	O	M1_TXD(2)	O	L5
M1_TXD(3)	O	M1_TXD(3)	O	B1
M1_TXEN	O	M1_TXEN	O	E3
M2_COL	I	M2_COL	I	K22
M2_CRS	I	M2_CRS	I	L24
M2_MDC	O	M2_MDC	O	C25
M2_MDI	I	M2_MDIO[b]	I/O/Z	E25
M2_MDO	O			
M2_MDO_EN_N	O			
M2_RXD(0)	I	M2_RXD(0)	I	B26
M2_RXD(1)	I	M2_RXD(1)	I	L23
M2_RXD(2)	I	M2_RXD(2)	I	E24
M2_RXD(3)	I	M2_RXD(3)	I	C24
M2_RXDV	I	M2_RXDV	I	F24
M2_RXER	I	M2_RXER	I	L22
M2_TXD(0)	O	M2_TXD(0)	O	M24
M2_TXD(1)	O	M2_TXD(1)	O	C26
M2_TXD(2)	O	M2_TXD(2)	O	M23
M2_TXD(3)	O	M2_TXD(3)	O	D26
M2_TXEN	O	M2_TXEN	O	M22

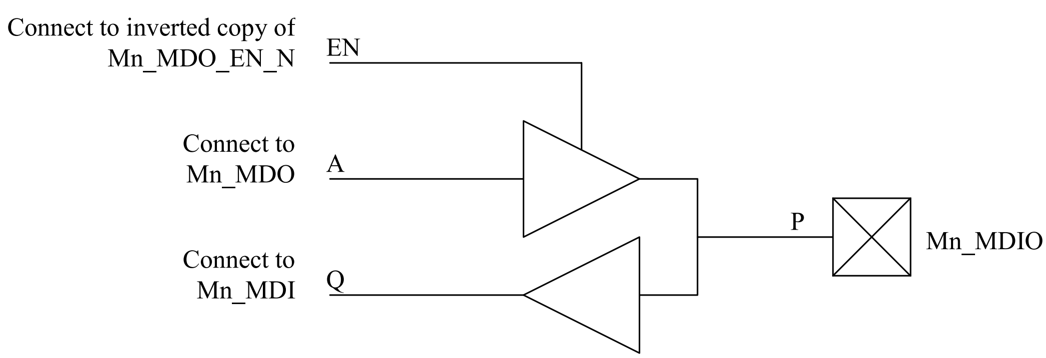
a. MII Interface direction is specified with respect to the ASSP portion of the device. I designates an input to the ASSP and O designates an output from the ASSP.

b. The M1_MDIO and M2_MDIO signals require the instantiation of a bipad_25um macro. See **Figure 5** for more details.

The pin assignments in **Table 3** represent connections to IO banks A and D of the Fabric. If the Ethernet MACs are connected in this fashion, these two banks must be configured for 3.3 V operation. See **Table 17** for more details.

As **Table 3** indicates, the Mn_MDIO pin is bi-directional. The signal on this pin is made up of three MII interface signals. Mn_MDI and Mn_MDO are the data input and output signals respectively, and Mn_MDO_EN_N is the tri-state buffer enable signal that turns on the pin output driver. Mn_MDO_EN_N is active low, and must be made active high by running it through a logic cell inverter before connecting it to the bipad_25um macro. **Figure 5** shows how these connections are made to the bipad_25um macro.

Figure 5: bipad_25um Macro and Connections to MII Interface Signals



The remainder of the MII signals are simple inputs or outputs and can be connected to the Fabric IO pins directly without the instantiation of any specific macros.

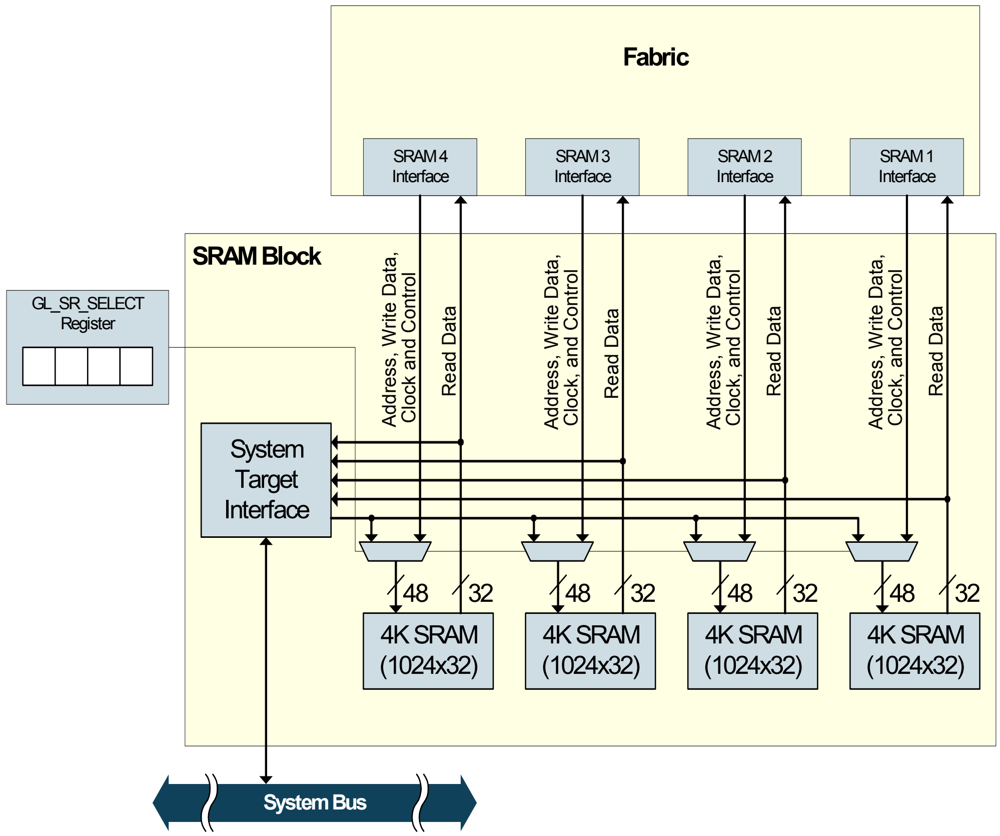
System SRAM

The QL902M contains 16 K bytes of SRAM internal to the ASSP portion of the device. This SRAM is divided into four equal sections of 4 KB each (each arranged as 32-bit x 1024 words), and each section can be configured to be connected to the System Bus (specifically, AHB) or connected to the Fabric directly.

When connected to the AHB, the SRAM (an AHB slave) can be accessed by any AHB master. Furthermore, the 4Kc core can use this internal SRAM for data or instruction storage. The SRAM supports 32-bit, 16-bit, or single byte accesses.

When connected to the Fabric, the SRAM can be accessed directly by any Fabric design without suffering the overhead of accessing it through the AHB interface. In addition, the Fabric has a master AHB interface, which it may also use to access the SRAM. However, in some high-speed applications, it may be necessary for the Fabric to have exclusive access to the SRAM memory. The Fabric interface of the SRAM block facilitates this function, and is described below. **Figure 6** shows the connection scheme of the four memory banks between the Fabric and the AHB.

Figure 6: SRAM Connection Scheme between the Fabric and AHB



Each SRAM block is single ported, and is connected to the Fabric or AHB through a selection MUX. For each SRAM block, the address bus, control signals, clock, and data write bus are all multiplexed by a register bit that is accessible through the System Bus. Therefore, any AHB master may control the switch of a given SRAM block between the AHB and the Fabric interface.

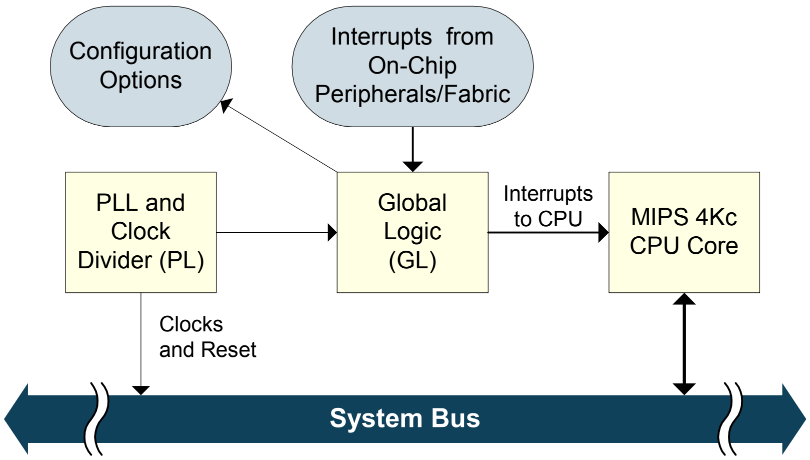
Because each SRAM clock is also switched by the selection MUX, it is possible for the Fabric to drive each SRAM block with a separate clock. However, when an SRAM block is connected to the AHB interface, the System Bus clock (hclk) is always used to drive the block.

Refer to **Table 51** for descriptions of all the System SRAM signals that interface to the Fabric.

Interrupt Controller

This section describes the function of the QL902M Interrupt Controller Unit (ICU). **Figure 7** shows a block diagram of the Interrupt Controller.

Figure 7: Simplified Interrupt Controller Block Diagram



The QL902M has 9 on-chip peripheral interrupts and 7 external interrupts (including on NMI), for a total of 16 interrupt sources. These 16 interrupt sources are combined into 7 interrupts by the interrupt controller and fed to the CPU core.

External interrupts must be asserted for at least two clock periods in order to be recognized as an interrupt. All interrupts are level triggered.

Each interrupt has an Emulation Enable Register bit and an Emulation Interrupt Value Register bit in the Interrupt Controller. The primary use for the Emulation registers is for testing purposes. The interrupt enable bits are stored in the GL_EMUL_EN register. The emulation interrupt value for each possible interrupt is stored in the GL_INT_EMUL register.

Each interrupt has an enable bit in the Global Individual Interrupt Enable (GL_IND_INT_EN) register in the Interrupt Controller. Each interrupt also has a status bit in the Global Individual Interrupt Status (GL_IND_INT_STATUS) register in the Interrupt Controller.

The Global CPU Interrupt Enable register (GL_CPU_INT_EN) enables masking of the interrupt groups after they have been grouped together.

The Interrupt Controller has no programmability for priority. That is, there is no hardware priority encoder. Priority is provided as a function of software.

High Performance 32-Bit System Bus (AMBA Bus)

The purpose of this section is to describe the AMBA[1] bus operation for the purposes of implementing user circuits in the Programmable Fabric. All circuits in the ASSP portion of the QL902M chip communicate with the Programmable Fabric primarily through the AMBA bus interfaces (Advanced Microcontroller Bus Architecture from ARM). Circuits implemented in the Programmable Fabric must be designed according to the AMBA Specification, Revision 2.0. The devices within the QL902M are interconnected through the Advanced High-performance Bus (AHB) or the Advanced Peripheral Bus (APB). Refer to the AMBA Specification, Revision 2.0, for more detailed information about the AHB and APB.

Advanced High-Performance Bus (AHB)

The AHB is the high-performance variant of the AMBA specification. It supports multiple bus masters and provides high bandwidth operation. The AHB implementation in the QL902M is 32 bits wide. All signals are synchronous to the rising clock edge of the bus clock (hclk).

The key features of the QL902M AHB include:

- Burst transfers
- Single-cycle bus master handover
- 32-bit bus runs at up to half the CPU clock frequency
- Multiple bus masters
- Arbitration through an AHB arbiter
- Address decoding through an AHB decoder

Table 4 lists the master and slave devices that connect to the AHB.

Table 4: Master and Slave Devices on the AHB

AHB Masters	AHB Slaves
MIPS 4Kc CPU	System SRAM
PCI	PCI
Ethernet Controller 1	Ethernet Controller 1
Ethernet Controller 2	Ethernet Controller 2
32-bit Master Interface to Programmable Fabric	32-bit Slave Interface to Programmable Fabric
	SDRAM and I/O Peripheral Controllers
	Interrupt Controller
	AHB to APB Bridge

The QL902M AHB supports multiple bus masters as well as bus slaves. Only one bus master can use the bus at a given time. The bus master provides address and control information when performing read and write operations. In response to the read or write operation from the bus master within a given address range, a bus slave provides information regarding the status of the data transfer (success, failure, or wait). The AHB arbiter ensures that only one bus master is initiating data transfers. The AHB decoder decodes the address of each transfer and provides a select signal for the slave that is involved in the transfer.

1. AMBA is a trademark of ARM Ltd.

AHB Arbitration

The internal arbiter of the QL902M provides either fixed arbitration priority or round robin rotating priority. Preemption occurs only after a burst of four. Consequently, when a low priority master is in control of the bus and a higher priority master requests access, the lower priority device will lose its grant after a burst of four before another master takes control. This minimizes the loss of performance that happens when bursts are preempted.

This preemption only happens when the designated AHB master is performing bursts of undefined lengths. If the AHB master is performing a burst of a defined (fixed) length, the burst will complete without interruption by the arbiter.

The ARB_FAIR_EN bit in the GL_INT_EMUL register determines the priority scheme.

When fixed priority is chosen, the priority of AHB masters is as follows (highest to lowest):

1. MIPS 4Kc CPU
2. Ethernet Controller 1
3. Ethernet Controller 2
4. PCI Controller
5. Programmable Fabric

Advanced Peripheral Bus (APB)

The APB is a simplified bus that is ideal for implementing device control registers and other non-burst transfers. The APB is a 32-bit wide bus that runs at the same frequency as AHB. The APB only accommodates slaves, does not support burst transfers, and does not support advanced slave response operations such as retries or wait state insertion. The APB on the QL902M is supported through an AHB-to-APB bridge. Three separately decoded APB regions are available for APB devices implemented in the Fabric. **Table 5** lists the slave devices on the APB.

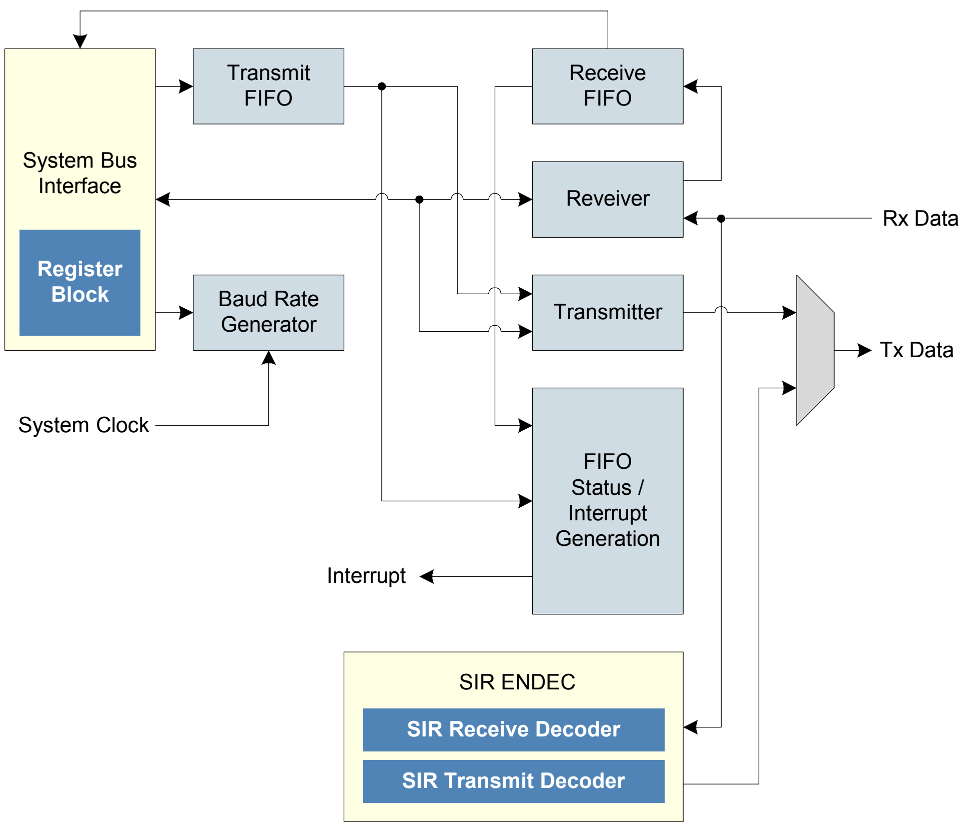
Table 5: Slave Devices on the APB

APB Slaves
Two UARTs
Four 16-bit Timers/Counters
Three 32-bit slave interfaces to Programmable Fabric

UARTs

The QL902M chip contains two UARTs. Each UART provides a full-duplex asynchronous receiver and transmitter and has programmable Baud rates. The UARTs contain an IrDA Serial Infrared (SIR) Encoder/ Decoder (ENDEC). One UART also has modem control signals. The serial output is software selectable between IrDA and generic serial modes. **Figure 8** shows a block diagram of the UART.

Figure 8: UART Block Diagram



The key features of the UARTs are as follows:

- Programmable Baud rate generation of up to 1/16 System Bus clock rate
- FIFO enable or disable
- 5, 6, 7, or 8 data bits
- 1 or 2 stop bits
- Odd and even, stick or no parity
- Parity, framing, and overrun error detection
- Line break generation and detection
- Loopback
- Interrupt generation

- IrDA SIR ENDEC block providing:
  - Programmable use of IrDA SIR or UART input/output
  - Support of IrDA SIR ENDEC functions for data rates up to 115.2 kilobits/second half-duplex
  - Support of normal 3/16 and low-power (1.41 to 2.23µs) bit durations
  - Programmable internal clock generator allowing division of reference clock by 1 to 512 for low-power mode bit duration

The System Bus (APB interface) generates read and write decodes for accesses to status/control registers and transmit/receive FIFO memories. The Register Block stores data written or to be read across the APB interface. The Baud Rate Generator contains free-running counters that generate a clock that is 16 times the transmit/receive bit rate.

The transmit FIFO is an 8-bit wide, 16-entry deep FIFO memory buffer. CPU data written across the APB interface is stored in the FIFO until read out by the transmit logic. The transmit FIFO can be disabled to act like a one-byte holding register. The receive FIFO is a 12-bit wide, 16-entry deep FIFO memory buffer. Received data and corresponding error bits, are stored in the receive FIFO by the receive logic until read out by the CPU across the APB interface. The receive FIFO can be disabled to act like a one-byte holding register.

The transmitter performs parallel-to-serial conversion on the data read from the transmit FIFO. Control logic outputs the serial bit stream beginning with a start bit, data bits, Least Significant Bit (LSB), a parity bit, and then stop bits according to the programmed configuration in control registers. The receiver performs serial-to-parallel conversion on the received bitstream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line break detection are also performed, and the data with associated overrun, parity, framing, and break error bits is written to the receive FIFO.

The Interrupt Generator outputs a single, combined interrupt to the QL902M Interrupt Controller.

The SIR Transmit Encoder modulates the Non-Return-to-Zero (NRZ) transmit bitstream output from the QL902M chip. The IrDA SIR physical layer specifies use of a Return To Zero, Inverted (RZI) modulation scheme that represents logic 0 as an infrared light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared Light Emitting Diode (LED).

The SIR Receive Decoder demodulates the return-to-zero bit stream from the infrared detector and outputs the received NRZ serial bitstream to the QL902M UART received data input. The decoder input is normally HIGH (marking state) in the idle state. The transmit encoder output has the opposite polarity to the decoder input.

General Purpose 16-bit Timer/Counters

The QL902M chip has four independent 16-bit timer/counter modules. The configuration registers for these modules are accessible through the System Bus (APB).

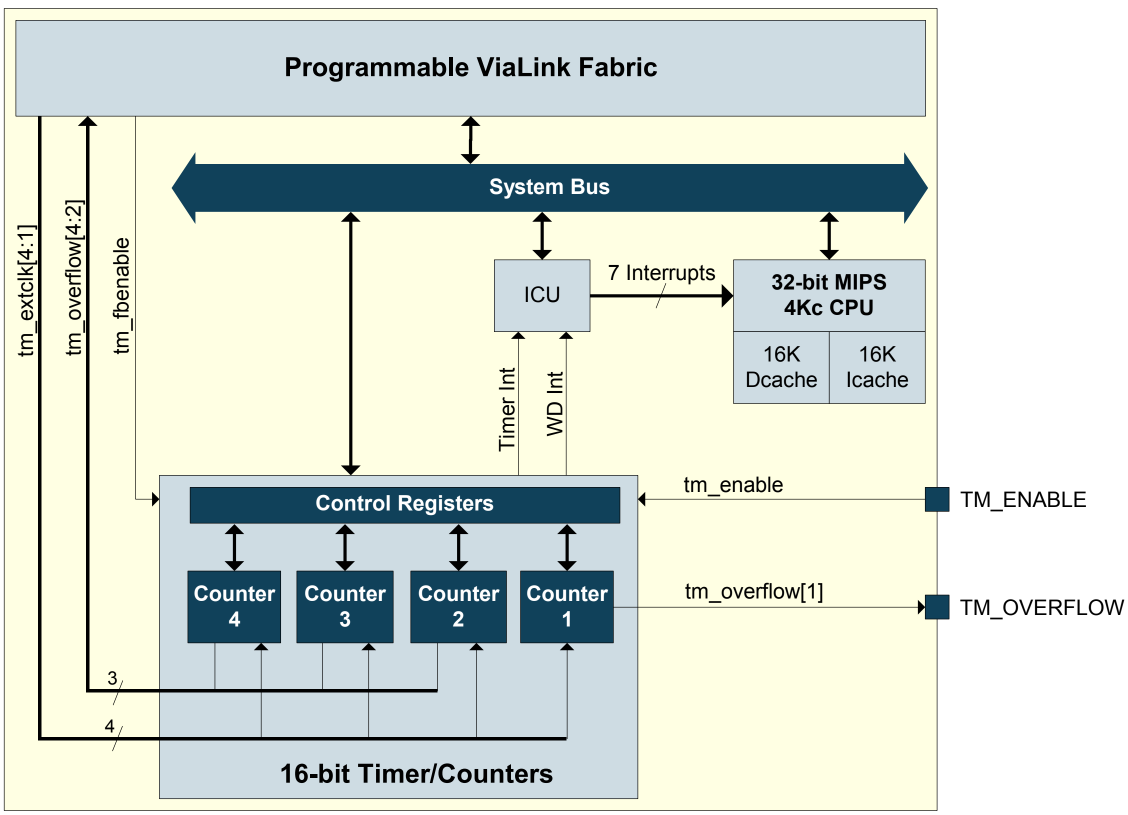
The System Bus clock (hclk), or an external clock supplied from the Fabric, drive the clock inputs on the timer/counter modules. These counters operate in one of four modes: decrement, increment, interval, or Pulse Width Modulation (PWM).

Each timer/counter module has the capability to generate system interrupts on various events.

One timer/counter is configured, by default, as a watchdog timer after a system reset. This watchdog timer has its own system interrupt output.

Figure 9 shows a functional block diagram of the timer module.

Figure 9: Timer Functional Block Diagram



The key features of each timer/counter module are as follows:

- Up to 100 MHz operation.
- 32-bit data path on the System Bus.
- 16-bit timer/counter.
- 16-bit pre-scaler to increase timer/counter delay.
- Four modes of operation: decrement, increment, interval and PWM.
- Operation from the System Bus clock (hclk) or an external clock from the Fabric.
- Two external hardware timer enable signals can be used to start/stop the timer/counter. One of these signals can be supplied from the Fabric and the other is a dedicated input pin on the chip.
- Three match interrupts, one interval interrupt and one overflow interrupt.
- Six control registers to control various counter functions, including enable/disable, load, and reset.

The timer/counters are controlled by a set of control registers. Each timer/counter module has six control registers. By contrast, one interrupt register is used to control and convey the status of the interrupts from all the modules.

One counter (Counter 4) is configured, by default, to be used as a watchdog timer after the system reset. This watchdog timer has its own system interrupt output, and it can be reconfigured by software for use as a standard timer/counter.

Fabric Side

This section discusses the various circuit elements in the Fabric portion of the QL902M device.

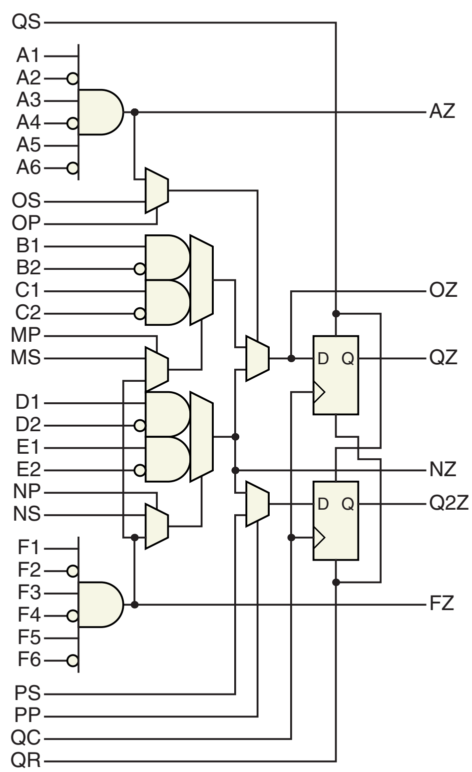
Logic Cells

The QL902M logic cell structure presented in **Figure 10** is a dual register, multiplexor-based logic cell. It is designed for wide fan-in and multiple, simultaneous output functions. Both registers share CLK, SET, and RESET inputs. The second register has a two-to-one multiplexer controlling its input. The register can be loaded from the NZ output or directly from a dedicated input.

NOTE: The input PP is not an "input" in the classical sense. It is a static input to the logic cell and selects which path (NZ or PS) is used as an input to the Q2Z register. All other inputs are dynamic and can be connected to multiple routing channels.

The complete logic cell consists of two 6-input AND gates, four two-input AND gates, seven two-to-one multiplexers, and two D flip-flops with asynchronous SET and RESET controls. The cell has a fan-in of 30 (including register control lines), fits a wide range of functions with up to 17 simultaneous inputs, and has six outputs (four combinatorial and two registered). The high logic capacity and fan-in of the logic cell accommodates many user functions with a single level of logic delay while other architectures require two or more levels of delay.

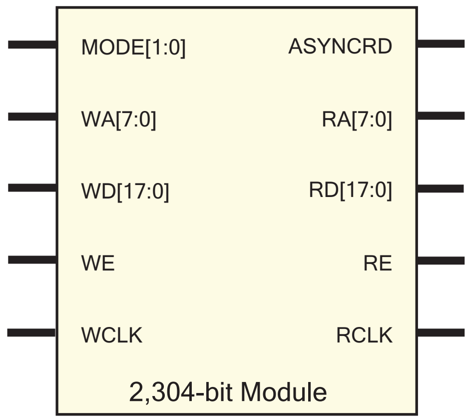
Figure 10: QL902M Logic Cell



Dual-Port SRAM Modules

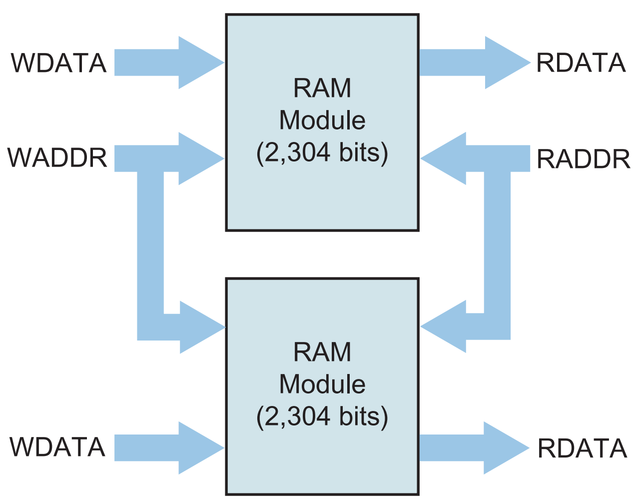
The QL902M includes up to 36 dual-port 2,304-bit RAM modules (shown in **Figure 11**) for implementing RAM, ROM, and FIFO functions. Each module is user-configurable into two different block organizations and can be cascaded vertically to increase their effective depth or horizontally to increase their effective width as shown in **Figure 12**.

Figure 11: 2,304-bit RAM Module



Using two *mode* pins, designers can configure each module into 128 x 18 (Mode 0) or 256 x 9 (Mode1).

Figure 12: Cascaded RAM Modules



The RAM modules are dual-port, with completely independent READ and WRITE ports and separate READ and WRITE clocks. The READ ports support asynchronous and synchronous operation, while the WRITE ports support synchronous operation. Each port has 18 data lines and 8 address lines, allowing word lengths of up to 18 bits and address spaces of up to 256 words. Depending on the mode selected, however, some higher order data lines or the highest order address line may not be used.

The Write Enable (WE) line acts as a clock enable for synchronous write operation. The Read Enable (RE) acts as a clock enable for synchronous READ operation (ASYNCRD input low). The RE and RCLK inputs are ignored when ASYNCRD is tied high.

Designers can cascade multiple RAM modules to increase the depth or width allowed in single modules by connecting corresponding address lines together and dividing the words between modules.

A similar technique can be used to create depths greater than 256 words. In this case address signals higher than the eighth bit are encoded onto the write enable (WE) input for WRITE operations. The READ data outputs are multiplexed together using encoded higher READ address bits for the multiplexer SELECT signals.

The RAM blocks can be loaded with data generated internally (typically for RAM or FIFO functions) or with data from an external PROM (typically for ROM functions).

Dual-Port SRAM Module Signals

The dual-port RAM module signal descriptions are shown in **Table 6**.

Table 6: Dual-Port RAM Module Signal Descriptions

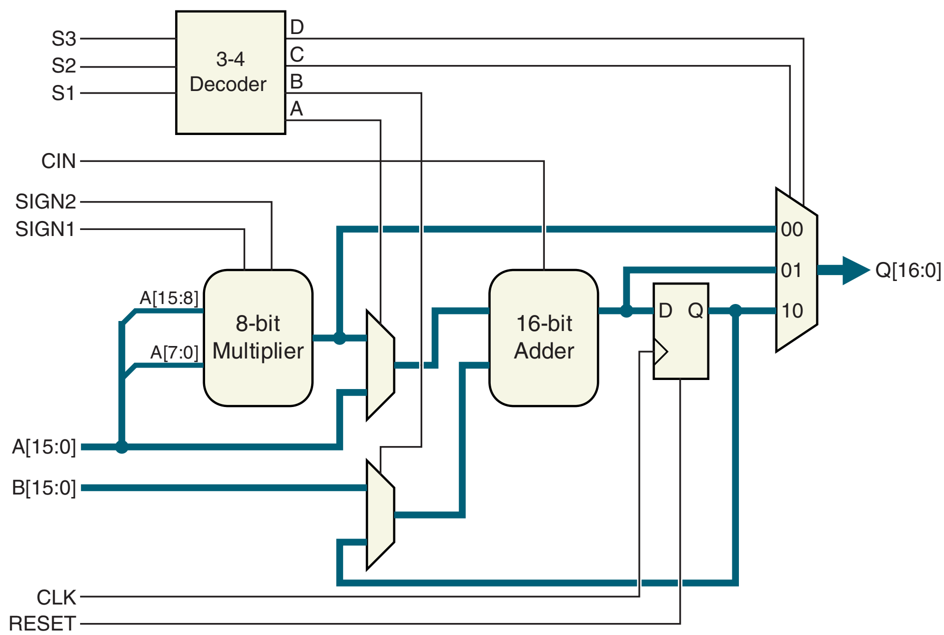
Signal Name	I/O	Description
WCLK	I	**Write Clock.** Clock input for the write port of the RAM module. All write port input signals are synchronous with this clock.
WE	I	**Write Enable.** Sampled on the rising edge of WCLK, when WE is high, data is written into the RAM module at the specified write address.
WA(7:0)	I	**Write Address.** Sampled on the rising edge of WCLK, this is the write address for the data to be written into the RAM module. WA(7:0) is ignored when WE is low. Note that some higher order bits of WA(7:0) may not be used depending on the selected mode for the RAM module (see MODE(1:0) signal description).
WD(17:0)	I	**Write Data.** Sampled on the rising edge of WCLK, this is the data to be written into the RAM module. WD(17:0) is ignored when WE is low. Note that some higher order bits of WD(17:0) may not be used depending on the selected mode for the RAM module (see MODE(1:0) signal description).
RCLK	I	**Read Clock.** This is the clock input for the read port of the RAM module. If ASYNCRD is low, all read port I/O signals are synchronous with this clock. If ASYNCRD is high, RCLK is ignored.
RE	I	**Read Enable.** Sampled on the rising edge of RCLK, when RE is high, data is read from the RAM module at the specified read address. If ASYNCRD is high, this RE is ignored.
RA(7:0)	I	**Read Address.** This is the read address for data to be read from the RAM module. If ASYNCRD is low, RA(7:0) is sampled only on the rising edge of RCLK while RE is high. If ASYNCRD is high, RA(7:0) is continuously sampled by the RAM module and RE has no effect. Note that some higher order bits of RA(7:0) may not be used depending on the selected mode for the RAM module (see MODE(1:0) signal description).
RD(17:0)	O	**Read Data.** This is the read output data from the RAM module. If the RAM module is in synchronous read mode (ASYNCRD low), valid read data is output immediately following the rising edge of RCLK which sampled RE as high. If the RAM module is in asynchronous read mode (ASYNCRD high), valid read data is output immediately after any change in the read address. Note that some higher order bits of RD(17:0) may not be used depending on the selected mode for the RAM module (see MODE(1:0) signal description).
ASYNCRD	I	**Asynchronous Read Input.** This signal, when high, indicates to the RAM block that the read port should operate asynchronously. When low, all read port I/O signals are synchronous with RCLK. This signal can only be tied to ‘1’ or ‘0’ inside the Fabric.
MODE(1:0)	I	**Mode for RAM Module.** These bits configure the width and depth of the RAM module (for both the read and write ports) and can only be tied to ‘1’ or ‘0’ inside the Fabric. The possible RAM module modes are: MODE(1:0) = “00” : 128 x 18 (locations x data bits) MODE(1:0) = “01” : 256 x 9 (locations x data bits) MODE(1:0) = “1X” : Reserved

Embedded Computational Units (ECUs)

Traditional Programmable Logic architectures do not implement arithmetic functions efficiently or effectively—these functions require high logic cell usage while garnering only moderate performance results.

The QL902M architecture allows for functionality above and beyond that achievable using programmable logic devices. By embedding a dynamically reconfigurable computational unit, the QL902M device can address various arithmetic functions efficiently. This approach offers greater performance than traditional programmable logic implementations. The embedded block is implemented at the transistor level as shown in **Figure 13**.

Figure 13: ECU Block Diagram



ECU Signals

Table 7 defines the ECU I/O signals. For more information on the operation of the ECU, see QuickLogic Application Note 52 at http://www.quicklogic.com/images/appnote52.pdf.

Table 7: ECU I/O Signals

Signal Name	I/O	Description
CLK	I	**Clock Input.** Input clock for the ECU output register.
RESET	I	**Reset Input.** Active high reset input for the ECU output register.
S1	I	**ECU Control S1.** One of three instruction signals that define the configuration mode of the ECU (see **Table 8**).
S2	I	**ECU Control S2.** One of three instruction signals that define the configuration mode of the ECU (see **Table 8**).
S3	I	**ECU Control S3.** One of three instruction signals that define the configuration mode of the ECU (see **Table 8**).
CIN	I	**Carry Input.** 1-bit Carry In for 16-bit adder operations.
SIGN1	I	**Sign Input for Multiplier A Input.** When SIGN1 = ‘1’, A(7:0) is treated as signed or two’s complement binary. When SIGN1 = ‘0’ A(7:0) is treated as unsigned binary.
SIGN2	I	**Sign Input for Multiplier B Input.** When SIGN2 = ‘1’, B(15:8) is treated as signed or two’s complement binary. When SIGN2 = ‘0’ B(15:8) is treated as unsigned binary.
A(15:0) A(15:8) A(7:0)	I	**Augend Input.** 16-bit augend input of the 16-bit adder when the ECU is in any of the adder configuration modes. **Multiplicand Input.** 8-bit multiplicand input when the ECU is in any of the multiplier configuration modes. **Multiplier Input.** 8-bit multiplier input when the ECU is in any of the multiplier configuration modes.
B(15:0)	I	**Addend Input.** 16-bit addend input when ECU is in any of the adder configuration modes.
Q(16:0)	O	**ECU Output.** This is the 17-bit output of the ECU. The interpretation of the value of Q(16:0) depends on the setting of S1, S2, S3, SIGN1, and SIGN2.

The QL902M ECU blocks are placed next to the SRAM circuitry for efficient memory/instruction fetch and addressing for DSP algorithmic implementations.

Up to eighteen 8-bit MAC functions can be implemented per cycle for a total of 1.8 billion MACs/s when clocked at 100 MHz. Additional multiply-accumulate functions can be implemented in the programmable logic.

The instruction modes for the ECU block are dynamically re-programmable through the programmable logic as shown in **Table 8**.

Table 8: ECU Mode Select Criteria

Instruction			Operation
S1	S2	S3	
0	0	0	Multiply
0	0	1	Multiply-Add
0	1	0	Accumulate
0	1	1	Add
1	0	0	Multiply (registered)[a]
1	0	1	Multiply- Add (registered)
1	1	0	Multiply - Accumulate
1	1	1	Add (registered)

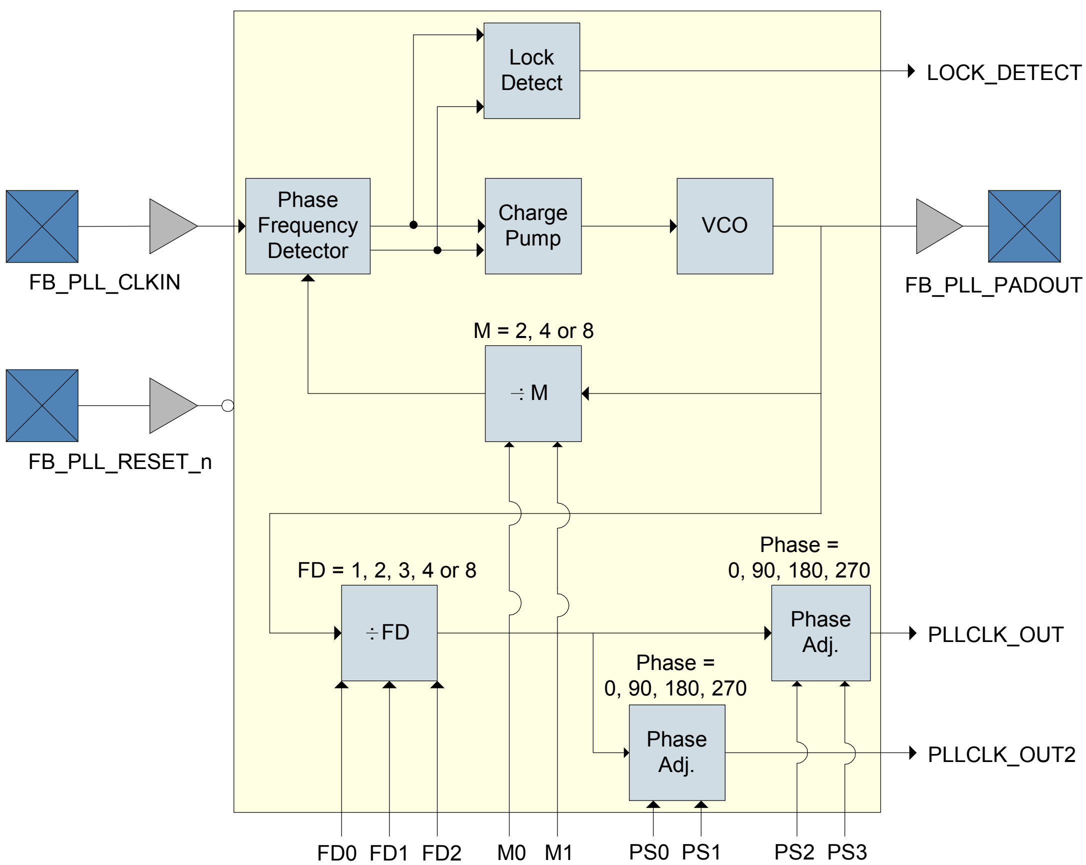
a. B (15:0) set to zero.

Fabric PLL

Instead of requiring extra components, designers simply need to instantiate the QL902M Fabric PLL model (described in this section). The Fabric PLL built into the QL902M supports a wider range of frequencies than many other PLLs. The PLL also has the ability to support different ranges of frequency multiplications or divisions, driving the Fabric at a faster or slower rate than the incoming clock frequency.

Figure 14 illustrates the QL902M Fabric PLL.

Figure 14: QL902M Fabric PLL Block Diagram



The QL902M Fabric PLL is driven by the FB_PLL_CLKIN input pin. This input is used by the PLL as a clock reference for the Voltage Controlled Oscillator (VCO) internal to the PLL circuit. Using the M1 and M0 inputs to the PLL block, the designer may choose a frequency multiplier value for the VCO. The PLL output pin, FB_PLL_PADOUT runs at the VCO operational frequency. **Table 9** shows the minimum and maximum input (F_{in}) and output (F_{pad}) frequency for each possible VCO multiplier mode.

Table 9: Frequency Ranges for Fabric PLL Input Pad (FB_PLL_CLKIN), Output Pad (FB_PLL_PADOUT) and VCO

M1	M0	VCO Mode	FB_PLL_CLKIN (F_{in}) Input Frequency Range	FB_PLL_PADOUT (F_{vco} and F_{pad}) Pad Output Frequency Range
0	0	2x	50.0 to 150.0 MHz	100 to 300 MHz
0	1	4x	25.0 to 75.0 MHz	100 to 300 MHz
1	0	8x	12.5 to 37.5 MHz	100 to 300 MHz
1	1	Reserved – should not be used		

In addition to the Fabric PLL output pin, the PLL can also drive two clock networks in the Fabric. These two additional outputs, PLLCLK_OUT and PLLCLK_OUT2 offer further control over frequency and phase by adding an output frequency divider and phase adjustment control. PLLCLK_OUT and PLLCLK_OUT2 can be configured for different output phases, but they both operate at the same frequency. **Table 10** shows the Fabric clock network PLLCLK_OUT resultant output frequency (F_{fab}), duty cycle and phase based on the FD2, FD1, FD0, PS3 and PS2 inputs.

Table 10: Frequency, Duty Cycle and Phase Values for Fabric PLL Output PLLCLK_OUT

Fabric Output PLLCLK_OUT							
FD2	FD1	FD0	(F_{fab}) Frequency	Duty Cycle	PS3	PS2	Phase
0	0	0	F_{VCO}	50%	X	X	0°
0	0	1	F_{VCO} / 2	50%	0	0	180°
					0	1	270°
					1	0	0°
					1	1	90°
0	1	0	F_{VCO} / 3	33%	X	X	0°
0	1	1	F_{VCO} / 4	50%	0	0	0°
					0	1	90°
					1	0	180°
					1	1	270°
1	X	X	F_{VCO} / 8	50%	0	0	270°
					0	1	0°
					1	0	90°
					1	1	180°

NOTE: PLLCLK_OUT phase shown in the **Table 10** is based on the period $1/F_{fab}$.

NOTE: An output phase of 0° indicates that the rising edge of PLLCLK_OUT will be aligned with the rising edge of FB_PLL_CLKIN.

NOTE: To maintain the phase values shown in **Table 10**, F_{VCO} should not exceed 200 MHz. If F_{VCO} is set above 200 MHz, the phase of PLLCLK_OUT will have no guaranteed relation to FB_PLL_CLKIN.

Table 11 shows the Fabric clock network PLLCLK_OUT2 resultant output frequency (F_{fab}), duty cycle and phase based on the FD2, FD1, FD0, PS1 and PS0 inputs.

Table 11: Frequency, Duty Cycle and Phase Values for Fabric PLL Output PLLCLK_OUT2

Fabric Output PLLCLK_OUT2							
FD2	**FD1**	**FD0**	**(F_{fab}) Frequency**	**Duty Cycle**	**PS1**	**PS0**	**Phase**
0	0	0	F_{VCO}	50%	X	X	0°
0	0	1	F_{VCO} / 2	50%	0	0	180°
					0	1	270°
					1	0	0°
					1	1	90°
0	1	0	F_{VCO} / 3	33%	X	X	0°
0	1	1	F_{VCO} / 4	50%	0	0	0°
					0	1	90°
					1	0	180°
					1	1	270°
1	X	X	F_{VCO} / 8	50%	0	0	270°
					0	1	0°
					1	0	90°
					1	1	180°

NOTE: PLLCLK_OUT2 phase shown in the **Table 11** is based on the period $1/F_{fab}$.

NOTE: An output phase of 0° indicates that the rising edge of PLLCLK_OUT2 will be aligned with the rising edge of FB_PLL_CLKIN.

NOTE: To maintain the phase values shown in **Table 11**, F_{VCO} should not exceed 200 MHz. If F_{VCO} is set above 200 MHz, the phase of PLLCLK_OUT2 will have no guaranteed relation to FB_PLL_CLKIN.

Fabric PLL Signals

Table 12 summarizes the key signals of the QL902M Fabric PLL.

Table 12: Fabric PLL I/O Signals

Signal Name	I/O	Description
FB_PLL_CLKIN	I	**Input clock signal.** See **Table 50** for more details.
FB_PLL_RESET_n	I	**Active low reset.** See **Table 50** for more details.
FB_PLL_PADOUT	O	**PLL output off chip.** See **Table 50** for more details.
PLLCLK_OUT	O	**PLL output to Fabric (1 of 2).** This is the PLL output clock driven to the Fabric clock network. This output runs at the F_{fab} frequency described in **Table 10**. Furthermore, this output is also phase adjustable. The output frequency and phase is determined by the M(1:0), FD(2:0) and PS(3:2) inputs.
PLLCLK_OUT2	O	**PLL output to Fabric (2 of 2).** This is the PLL output clock driven to the Fabric clock network. This output runs at the F_{fab} frequency described in **Table 11**. Furthermore, this output is also phase adjustable. The output frequency and phase is determined by the M(1:0), FD(2:0) and PS(1:0) inputs.

Table 12: Fabric PLL I/O Signals (Continued)

Signal Name	I/O	Description
LOCK_DETECT	O	**Active high lock detection signal.** This signal is output to the Fabric and goes high when the PLL VCO is locked to the input clock reference.
M(1:0)	I	**VCO frequency multiplier input.** These inputs are statically driven high or low by the Fabric. See **Table 9** for their effect on PLL operation.
FD(2:0)	I	**Fabric output frequency divide input.** These inputs are statically driven high or low by the Fabric. See **Table 10** for their effect on PLL operation.
PS(3:2)	I	**Phase select input for Fabric output.** These inputs are statically driven high or low by the Fabric. See **Table 10** for their effect on PLL operation.
PS(1:0)	I	**Phase select input for Fabric output2.** These inputs are statically driven high or low by the Fabric. See **Table 11** for their effect on PLL operation.

NOTE: Because FB_PLL_CLKIN, FB_PLL_RESET_n, and FB_PLL_PADOUT have appropriate INPAD and OUTPADs included, you do not have to add these pads to your design.

Advanced Clock Networks

The QL902M device has a large number of extremely advanced and highly flexible clock networks. These consist of three basic types of networks; a Global Network and a Dedicated Network (for low-skew applications), and an I/O Control/Hi-Drive Network (for high-fanout, multi-load applications).

Global and Dedicated Low-Skew Networks

The Global and Dedicated Networks are low-skew networks typically used to drive clock signals throughout the entire device. In addition, the Global Network can also be used to globally drive other high-fanout signals with low skew (e.g., flip-flop set or reset signals). Both networks are segmented, separated on a per-quadrant basis. (Unlike the QuickLogic Eclipse architecture, the QL902M has only two quadrants.) **Figure 15** shows a simplified view of the low-skew clock network architecture in the QL902M device.

Figure 15: Low Skew Clock Architecture

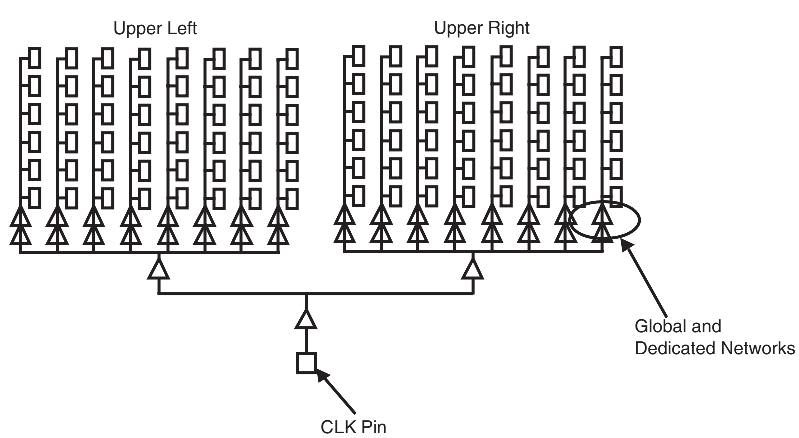


Table 13 shows the number of Global and Dedicated Networks available per quadrant in the QL902M.

Table 13: Number of QL902M Low-Skew Clock Networks

Clock Network Type	Quadrant		Total
	Upper Left	Upper Right	
Global	8 (5 are Quad-Nets)	8 (5 are Quad-Nets)	16
Dedicated	1	1	2
Total	9	9	18

As **Table 13** shows, there are a total of nine low-skew networks per quadrant in the QL902M, making a total of eighteen in the entire device. Each quadrant contains eight Global Networks and one Dedicated Network.

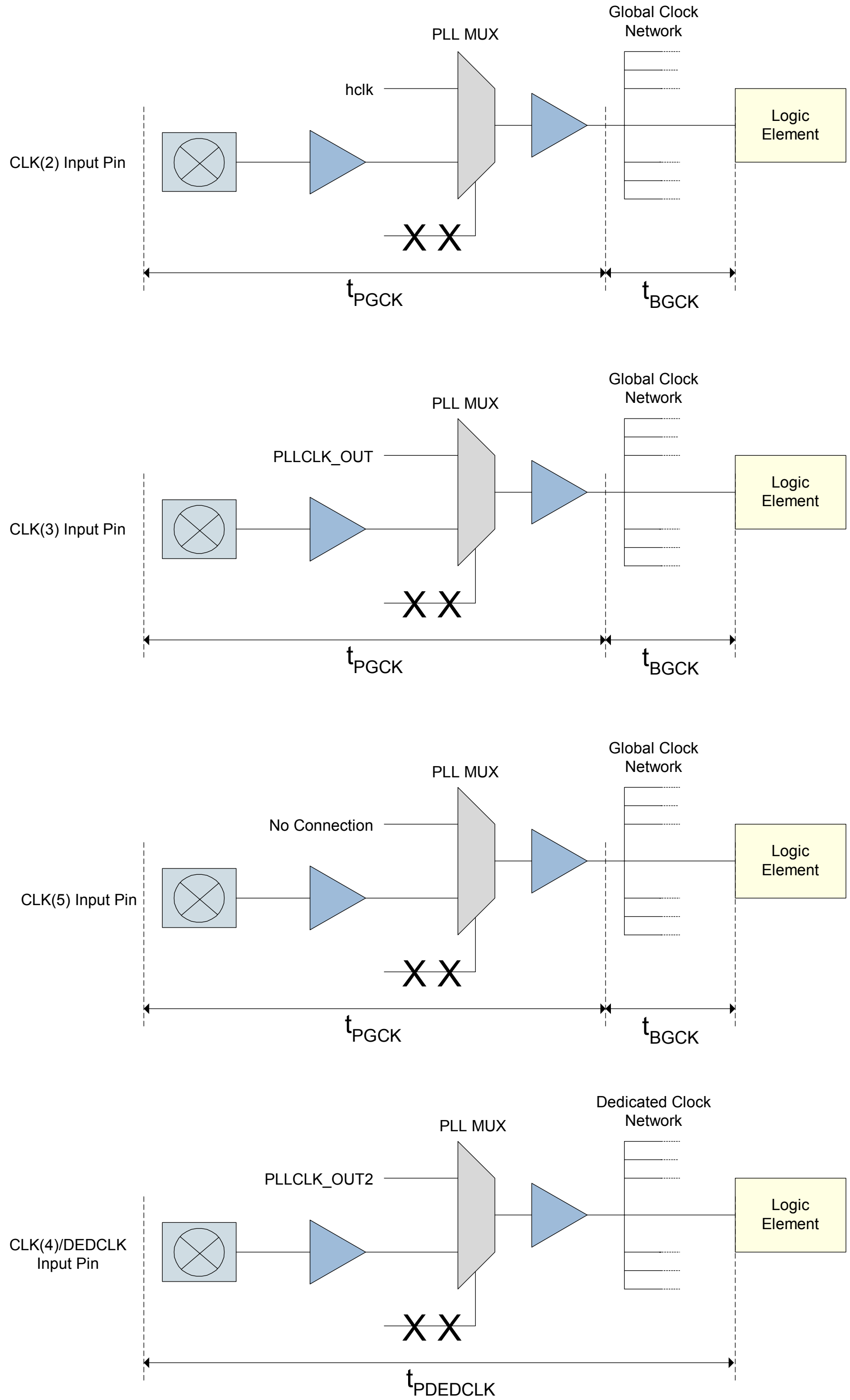
The Global Network and Dedicated Network differ slightly in performance and flexibility. The Dedicated Network offers superb low-skew and minimal pin to logic element delay performance, but can only drive the clock inputs of specific Fabric elements. The Global Network offers more flexibility to drive a variety of inputs in the Fabric as well as internal ASSP port inputs, but at a slight increase in skew and delay. **Table 14** outlines all allowable input destinations for each clock network type.

Table 14: Allowable Inputs Destinations for Global and Dedicated Networks

Element	Inputs That Can be Driven by the Global Clock Network	Inputs That Can be Driven by the Dedicated Clock Network
Logic Cells	QC	QC
	A2	
	F1	
	QS	
	QR	
RAM Modules	WCLK	WCLK
	RCLK	
	RE	RCLK
	WE	
ECUs	CLK	-
	RESET	
I/O Cells	IQC	IQC
	IQR	
	EQE	
	IQE	
	IE	
ASSP Interface	All Inputs	-

Each quadrant consists of an element called the PLLMUX that drives the Global Networks and the Dedicated Network in the quadrant. The PLLMUX selects between external CLK input pins and clock signals that are driven from other elements internal to the device such as the Fabric PLL and the ASSP System Bus clock (hclk). **Figure 16** shows a simplified schematic diagram of the CLK input pins, the PLLMUX elements and the associated clock networks.

Figure 16: Low Skew Clock Structure Schematic Based Upon PLLMUX Elements (1 of 2 Quadrants)



If either of the Fabric PLL outputs or hclk are utilized in a given quadrant of the Fabric design, the QuickWorks software automatically configures the corresponding PLLMUX to select the internal clock input. Each quadrant consists of four PLLMUX elements of which one input is tied to a specific CLK input pin as shown in **Table 15**.

Table 15: PLLMUX Input Signals and Output Type (Per Quadrant)

Input		Output Type
Pin	Internal Signals	
CLK(2)	hclk	Global clock network
CLK(3)	PLLCLK_OUT	Global clock network
CLK(4)/DEDCLK	PLLCLK_OUT2	Dedicated clock network
CLK(5)	-	Global clock network

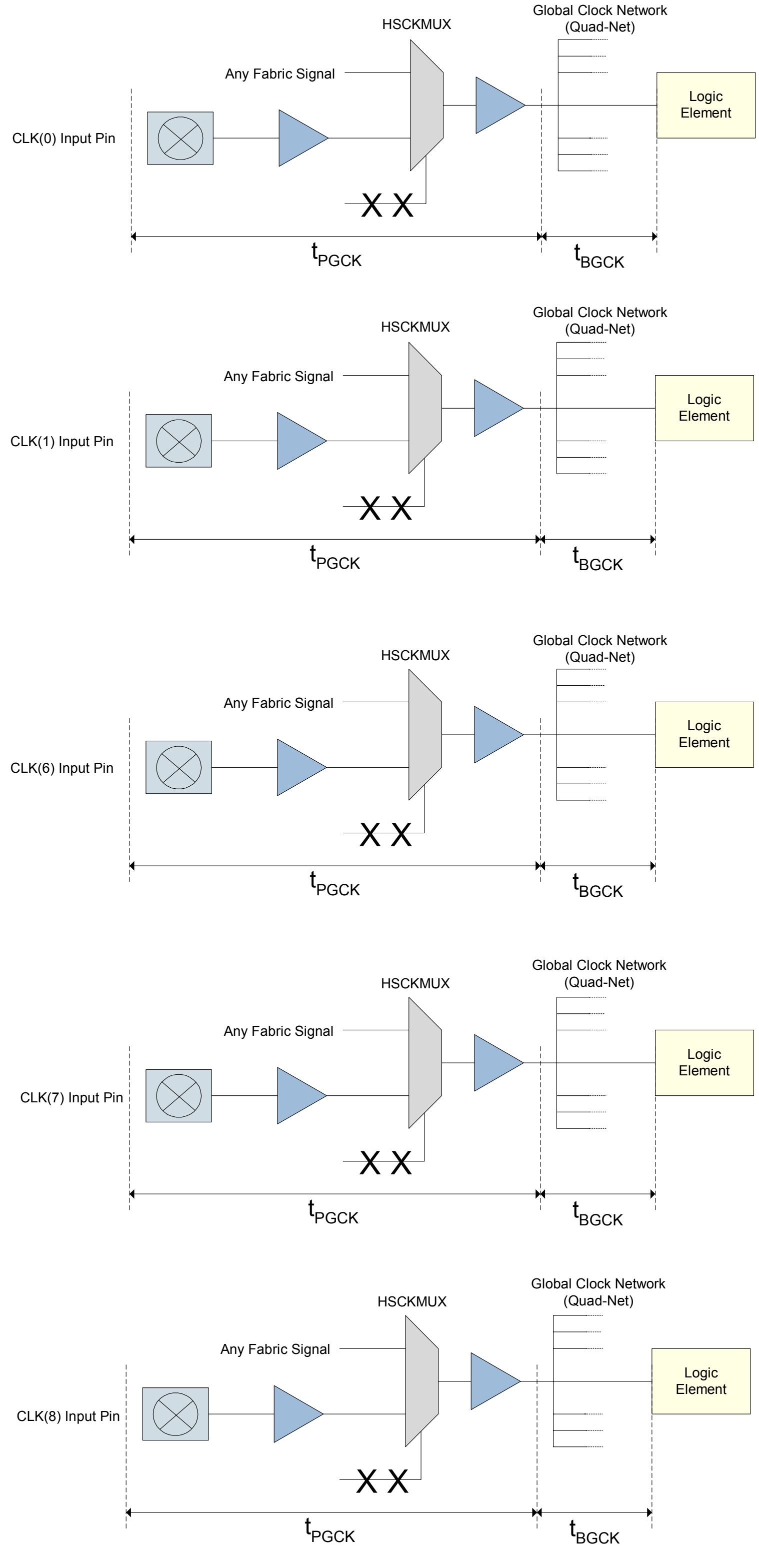
As **Table 15** indicates, once a PLLMUX is used to drive an internal signal onto the Global or Dedicated Networks, the corresponding CLK input pin is blocked from entering that quadrant.

NOTE: If either of the Fabric PLL outputs or hclk are utilized in the Fabric design, and external clock pins are also utilized, the designer should choose clock input pins on the device that do not conflict with these corresponding PLLMUX elements.

Quad-Net Network (Subset of the Global Network)

In each quadrant, the remaining five Global Networks are also referred to as Quad-Net Networks. Quad-Nets are networks that can be driven by input CLK pins or by signals that are generated internally to the Fabric. Quad-Nets are driven by an element in the Fabric called the HSCKMUX. **Figure 17** shows a simplified schematic diagram of the CLK input pins, the HSCKMUX elements and the associated Quad-Net networks.

Figure 17: Low Skew Clock Structure Schematic Based Upon HSCKMUX Elements (1 of 2 Quadrants)



By instantiating the gclkbuff_25um macro with a given Fabric signal as its input, the designer can program an HSCKMUX to drive this signal on a Quad-Net (the QuickWorks tool automatically chooses which HSCKMUX to use). Each quadrant consists of five HSCKMUX elements of which one input is tied to a specific CLK input pin as shown in **Table 16**.

Table 16: HSCKMUX Input Signals and Output Type (Per Quadrant)

Input		Output Type
Pin	Internal Signals	
CLK(0)	Any Signal	Global clock network (Quad-Net)
CLK(1)	Any Signal	Global clock network (Quad-Net)
CLK(6)	Any Signal	Global clock network (Quad-Net)
CLK(7)	Any Signal	Global clock network (Quad-Net)
CLK(8)	Any Signal	Global clock network (Quad-Net)

As **Table 16** indicates, once an HSCKMUX is used to drive an internal signal onto the Quad-Net Network, the corresponding CLK input pin is blocked from entering that quadrant.

NOTE: If the sum of utilized clock input pins from **Table 16** and the number of instantiated gclkbuff_25um macros is greater than five, the QuickWorks software may be unable to successfully resolve the conflicts between the CLK input pins and the internally generated clock input signals. This is dependant on several factors, but in most cases can be attributed to the size and complexity of the Fabric design.

I/O Control/Hi-Drive Network

The I/O Control/Hi-Drive Network is used primarily to drive high-fanout (typically other than clock or reset) signals throughout the device. Each bank of I/Os has two input-only pins entitled IOCTRL that can be programmed to drive the IQC (flip-flop clock), IQR (flip-flop reset), EQE & IQE (flip-flop enables), and IE (output enable) inputs of each I/O cell in that bank. These input-only pins also simultaneously serve as high drive inputs to any logic element input located in the adjacent quadrant. In addition, the I/O Control/Hi-Drive Network can be driven by the internal logic by instantiating the io_buff_25um macro. The QL902M has a total of eight IOCTRL input pins which are also shared with the io_buff_25um macros (i.e., if a io_buff_25um macro is utilized at a specific location, the corresponding IOCTRL input pin is ignored by the device. The performance of this network is presented in **Table 35**.

General Routing Network

QL902M devices are delivered with six types of routing resources as follows: short (sometimes called segmented) wires, dual wires, quad wires, express wires, distributed networks, and default wires. Short wires span the length of one logic cell, always in the vertical direction. Dual wires run horizontally and span the length of two logic cells. Short and dual wires are predominantly used for local connections. Default wires supply V_{CC} and GND (Logic ‘1’ and Logic ‘0’) to each column of logic cells.

Quad wires have passive link interconnect elements every fourth logic cell. As a result, these wires are typically used to implement intermediate length or medium fan-out nets.

Express lines run the length of the programmable logic uninterrupted. Each of these lines has a higher capacitance than a quad, dual, or short wire, but less capacitance than shorter wires connected to run the length of the device. The resistance will also be lower because the express wires don't require the use of “pass” links. Express wires provide higher performance for long routes or high fan-out nets.

Distributed networks are described in **Advanced Clock Networks** on page 29. These wires span the programmable logic and are driven by "column clock" buffers. All clock network pin buffers (both Dedicated and Global) are hard wired to individual sets of column clock buffers.

Programmable I/O

The QL902M features a variety of distinct I/O pins to maximize performance, functionality, and flexibility with bi-directional I/O pins and input-only pins. All input and I/O pins are 1.8 V, 2.5 V, and 3.3 V tolerant and comply with the specific I/O standard selected. For single-ended I/O standards, VCCIO(A:D) specifies the input tolerance and the output drive. For example, the VCCIO(A:D) pins must be tied to a 3.3 V supply to provide 3.3 V compliance. For voltage referenced I/O standards (e.g, SSTL), the voltage supplied to the INREF(A:D) pins in each bank specifies the input switch point. The QL902M can also support the LVDS and LVPECL I/O standards with the use of external resistors (see **Table 17**).

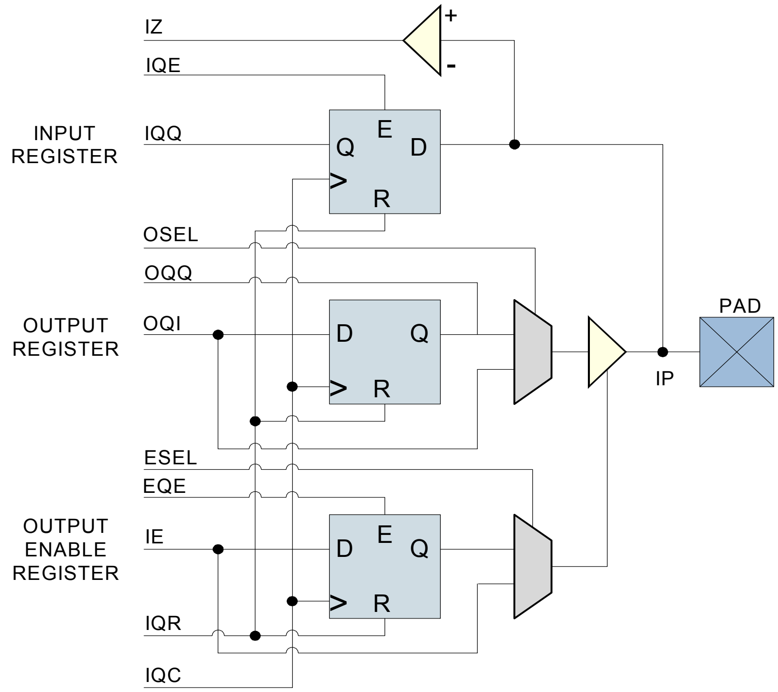
Table 17: I/O Standards and Applications

I/O Standard	Reference Voltage	Output Voltage	Application
LVTTL	n/a	3.3 V	General Purpose
LVCMOS25	n/a	2.5 V	General Purpose
LVCMOS18	n/a	1.8 V	General Purpose
PCI	n/a	3.3 V	PCI Bus Applications
GTL+	1	n/a	Backplane
SSTL3	1.5	3.3 V	SDRAM
SSTL2	1.25	2.5 V	SDRAM

As designs become more complex and requirements more stringent, several application-specific I/O standards have emerged for specific applications. I/O standards for processors, memories, and a variety of bus applications have become commonplace and a requirement for many systems. In addition, I/O timing has become a greater issue with specific requirements for setup, hold, clock to out, and switching times. The QL902M has addressed these new system requirements and includes a new I/O cell which consists of programmable I/Os as well as a new cell structure consisting of three registers—Input, Output, and OE.

The bi-directional I/O pin options can be programmed for input, output, or bi-directional operation. As shown in **Figure 18**, each bi-directional I/O pin is associated with an I/O cell which features an input register, an input buffer, an output register, a three-state output buffer, an output enable register, and 2 two-to-one output multiplexers. The select lines of the two-to-one multiplexers are static and must be connected to either V_{cc} or GND.

Figure 18: QL902M I/O Cell



For input functions, I/O pins can provide combinatorial, registered data, or both options simultaneously to the logic array. For combinatorial input operation, data is routed from I/O pins through the input buffer to the array logic. For registered input operation, I/O pins drive the D input of the input registers, allowing data to be captured with fast set-up times without consuming internal logic cell resources. The comparator and multiplexor in the input path allows for native support of I/O standards with reference points offset from traditional ground.

For output functions, I/O pins can receive combinatorial or registered data from the logic array. For combinatorial output operation, data is routed from the logic array through a multiplexer to the I/O pin. For registered output operation, the array logic drives the D input of the output register which in turn drives the I/O pin through a multiplexer. The multiplexer allows either a combinatorial or a registered signal to be driven to the I/O pin. Using the output register will also decrease the Tco. Since the output register does not need to drive the routing the length of the output path is also reduced.

The three-state output buffer controls the flow of data from the array logic to the I/O pin and allows the I/O pin to act as an input and/or output. The buffer's output enable can be individually controlled by the logic cell array or any pin (through the regular routing resources), or it can be bank-controlled through one of the global networks. The signal can also be either combinatorial or registered. This is identical to that of the flow for the output register. For combinatorial control operation data is routed from the logic array through a multiplexer to the three-state control. The IOCTRL pins can directly drive the OE and CLK signals for all I/O cells within the same bank.

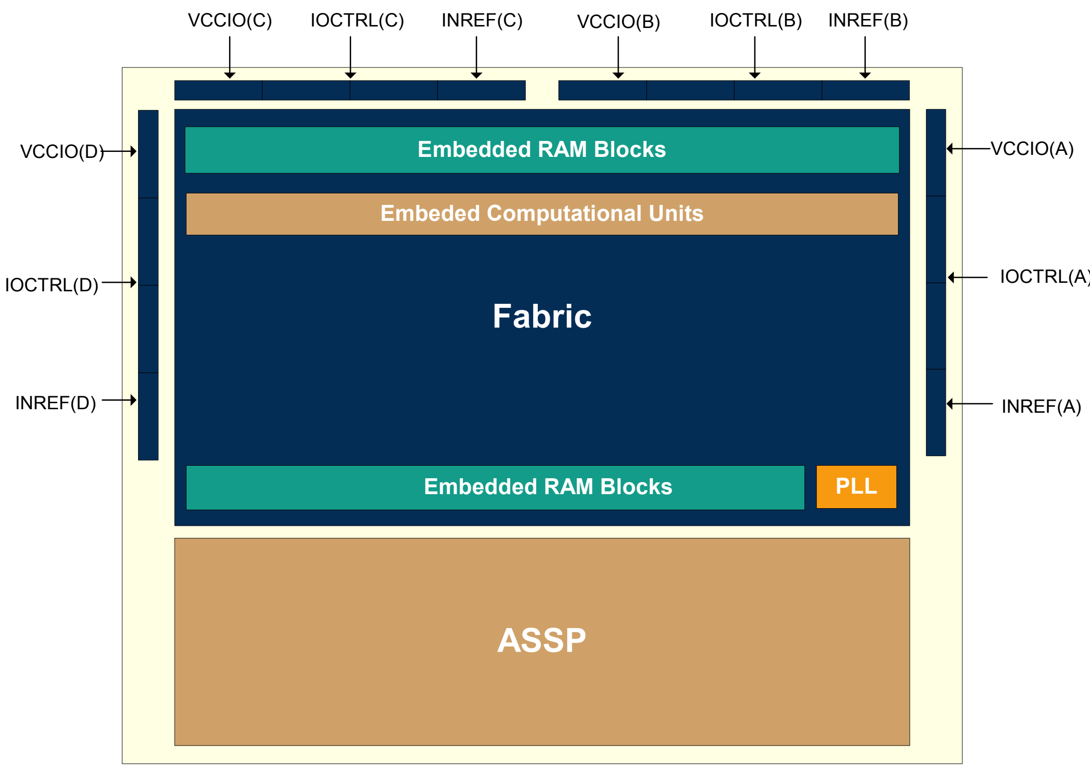
For registered control operation, the array logic drives the D input of the OE register which in turn drives the three-state control through a multiplexer. The multiplexer allows either a combinatorial or a registered signal to be driven to the three-state control.

When I/O pins are unused, the OE controls can be permanently disabled, allowing the output register to be used for registered feedback into the logic array.

I/O cell registers are controlled by clock, clock enable, and reset signals, which can come from the regular routing resources, from one of the global networks, or from two IOCTRL input pins per bank of I/Os. The CLK and RESET signals share common lines, while the clock enables for each register can be independently controlled. I/O interface support is programmable on a per bank basis. The QL902M contains four I/O banks. **Figure 19** illustrates the I/O bank configurations.

Each I/O bank is independent of other I/O banks and each I/O bank has its own VCCIO and INREF supply inputs. A mixture of different I/O standards can be used on the device; however, there is a limitation as to which I/O standards can be supported within a given bank. Only standards that share a common VCCIO and INREF can be shared within the same bank (e.g., PCI and LVTTL).

Figure 19: Multiple I/O Banks



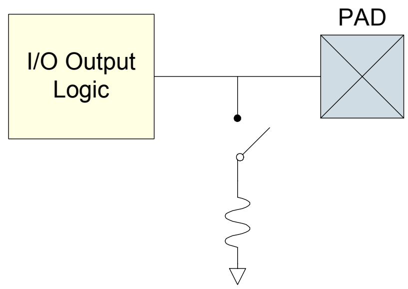
Programmable Slew Rate

Each I/O has programmable slew rate capability—the slew rate can be either fast or slow. The slower rate can be used to reduce the switching noise of each I/O. See **Table 41** through **Table 43** for specific information on the slew rates for the Fabric I/O pins. The option to change the slew rate is selectable through QuickWorks in the Tools/Configure Pins window in SpDE.

Programmable Weak Pull-Down

A programmable Weak Pull-Down resistor is available on each I/O. The I/O Weak Pull-Down eliminates the need for external pull down resistors for used I/Os. The spec for pull-down current is maximum of 150 µA under worst case conditions. The option to use the programmable weak pull-down resistor is selectable through QuickWorks in the Tools/Configure Pins window in SpDE.

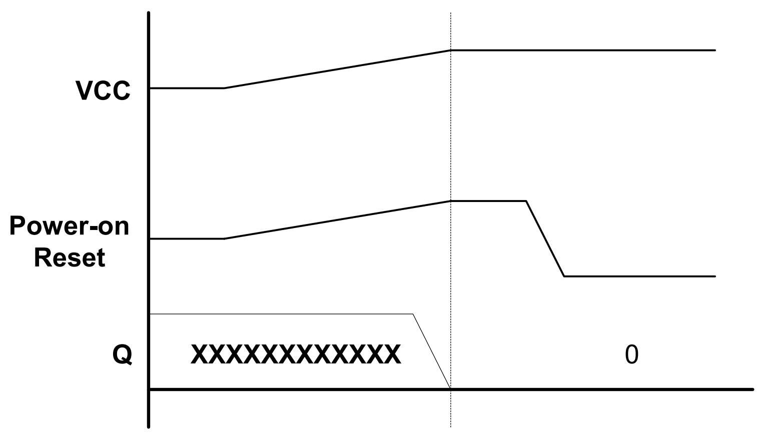
Figure 20: Programmable I/O Weak Pull-Down



Global Power-On Reset (POR)

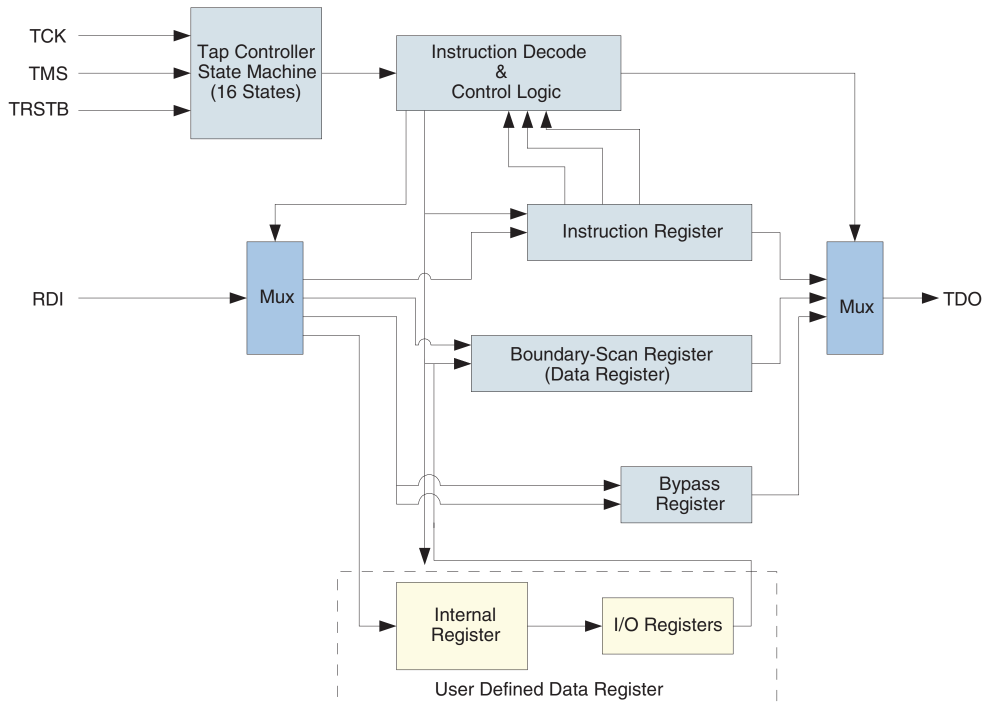
The QL902M family of devices features a global power-on reset. This reset is hardwired to all registers and resets them to Logic '0' upon power-up of the device. In QuickLogic devices, the asynchronous Reset input to flip-flops has priority over the Set input; therefore, the Global POR will reset all flip-flops during power-up. If you want to set the flip-flops to Logic '1', you must assert the "Set" signal after the Global POR signal has been deasserted. This is accomplished by holding the "Set" signal high for at least 1 ms after the V_{CC} supply has reached 1.95 V.

Figure 21: Power-On Reset



Joint Test Access Group (JTAG)

Figure 22: JTAG Block Diagram



Microprocessors and Application Specific Integrated Circuits (ASICs) pose many design challenges, one problem being the accessibility of test points. JTAG formed in response to this challenge, resulting in IEEE standard 1149.1, the Standard Test Access Port and Boundary Scan Architecture.

The JTAG boundary scan test methodology allows complete observation and control of the boundary pins of a JTAG-compatible device through JTAG software. A Test Access Port (TAP) Controller works in concert with the Instruction Register (IR), which allow users to run three required tests along with several user-defined tests.

JTAG tests allow users to reduce system debug time, reuse test platforms and tools, and reuse subsystem tests for fuller verification of higher level system elements.

The 1149.1 standard requires the following three tests:

- **Extest Instruction.** The Extest instruction performs a PCB interconnect test. This test places a device into an external boundary test mode, selecting the boundary scan register to be connected between the TAPs Test Data In (TDI) and Test Data Out (TDO) pins. Boundary scan cells are preloaded with test patterns (through the Sample/Preload Instruction), and input boundary cells capture the input data for analysis.
- **Sample/Preload Instruction.** This instruction allows a device to remain in its functional mode, while selecting the boundary scan register to be connected between the TDI and TDO pins. For this test, the boundary scan register can be accessed through a data scan operation, allowing users to sample the functional data entering and leaving the device.

- **Bypass Instruction.** The Bypass instruction allows data to skip a device's boundary scan entirely, so the data passes through the bypass register. The Bypass instruction allows users to test a device without passing through other devices. The bypass register is connected between the TDI and TDO pins, allowing serial data to be transferred through a device without affecting the operation of the device.

JTAG BSDL Support

- BSDL-Boundary Scan Description Language
- Machine-readable data for test equipment to generate testing vectors and software
- BSDL files available for all device/package combinations from QuickLogic
- Extensive industry support available and ATVG (Automatic Test Vector Generation)

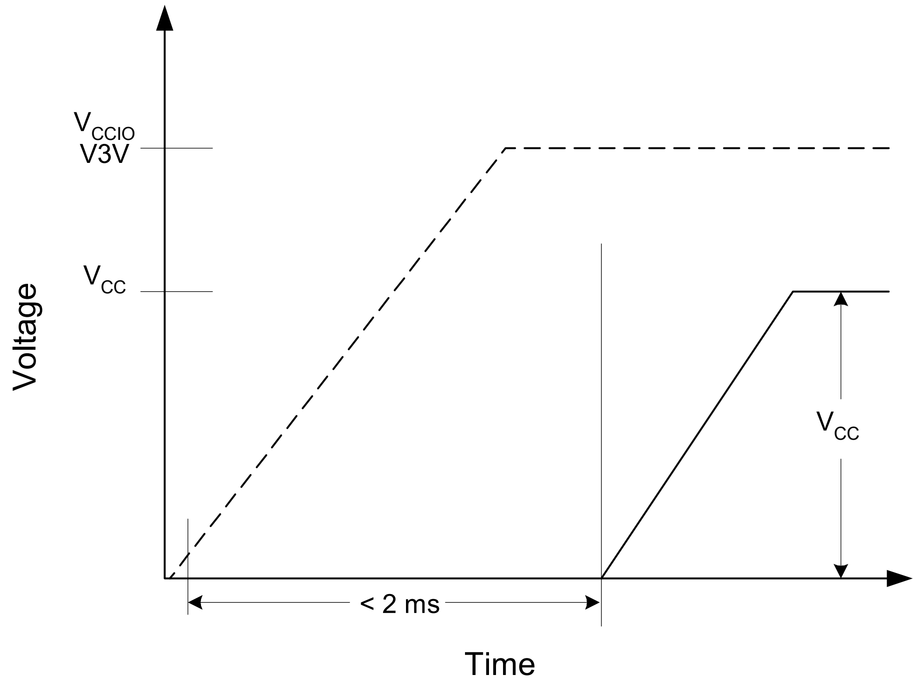
Security Links

There are several security links: to disable reading logic from the array, and to disable JTAG access to the device. Programming these optional links completely disables access to the device from the outside world and provides an extra level of design security not possible in SRAM-based FPGAs. The option to program these links is selectable through QuickWorks in the Tools/Options/Device Programming window in SpDE.

Power-Up Loading Link

The flexibility link enables Power-Up Loading of the Embedded RAM blocks. If the link is programmed, the Power Up Loading state machine is activated during power-up of the device. The state machine communicates with an external EPROM via the JTAG pins to download memory contents into the on-chip RAM. If the link is not programmed, Power-Up Loading is not enabled and the JTAG pins function as they normally would. The option to program this link is selectable through QuickWorks in the Tools/Options/Device Programming window in SpDE. For more information on Power-Up Loading, see QuickLogic Application Note 55 at http://www.quicklogic.com/images/appnote55.pdf. See the Power-Up Loading power-up sequencing requirement for proper functionality in **Figure 23**.

Figure 23: Required Power-Up Sequence when using Power-up Loading



Electrical Specifications

DC Characteristics

The DC Specifications are provided in **Table 18** through **Table 23**.

Table 18: Absolute Maximum Ratings

Parameter	Value	Parameter	Value
VCC Voltage	-0.5 V to 2.0 V	Latch-up Immunity	±100 mA
VCCIO(A:D) Voltage	-0.5 V to 4.0 V	DC Input Current	±20 mA
INREF(A:D) Voltage	0.5 V to VCCIO(A:D)	BGA Package Storage Temperature	-55° C to +125° C
Input Voltage	-0.5 V to VCCIO(A:D) + 0.5 V	V3V Voltage	-0.5 V to 4.0 V

Table 19: Operating Range

Symbol	Parameter		Industrial		Commercial		Unit
			Min.	Max.	Min.	Max.	
VCC	Supply Voltage		1.90	2.00	1.90	2.00	V
V3V	ASSP I/O Supply Voltage		3.00	3.60	3.00	3.60	V
VCCIO(A:D)	Fabric I/O Bank Supply Voltages		1.71	3.60	1.71	3.60	V
T_J	Junction Temperature		-40	100	0	85	°C
K	Delay Factor	-1 Speed Grade	0.48	1.32	0.51	1.29	n/a
		-2 Speed Grade	0.47	1.17	0.50	1.14	n/a

Table 20: DC Characteristics

Symbol	Parameter	Conditions	Min.	Max.	Units
I_{CC}	D.C. Supply Current	V_I, V_O = V3V, VCCIO(A:D) or GND	-	TBD	mA
$I_{CC(STATIC)}$	D.C. Supply Current, Static	V_I, V_O = V3V, VCCIO(A:D) or GND, All clock inputs are 0 MHz	-	600	µA
I_{V3V}	D.C. Supply Current on V3V	V3V = 3.3 V	-	TBD	mA

Table 21: Fabric I/O DC Characteristics

Symbol	Parameter	Conditions	Min.	Max.	Units
I_I	I or I/O Input Leakage Current	V_I = VCCIO(A:D) or GND	-1	1	µA
I_{OZ}	3-State Output Leakage Current	V_I = VCCIO(A:D) or GND	-	1	µA
C_I	I/O Input Capacitance[a]	-	-	8	pF
C_{CLOCK}	Clock Input Capacitance[a]	-	-	12	pF
I_{OS}	Output Short Circuit Current[b]	V_o = GND V_o = VCCIO(A:D)	-15 40	-180 210	mA mA
I_{REF}	D.C. Supply Current on INREF(A:D)	-	-10	10	µA
I_{PD}	Current on programmable Pull-down	VCCIO(A:D) = 3.6 V VCCIO(A:D) = 2.5 V VCCIO(A:D) = 1.8 V	-	150	µA
I_{CCIO}	D.C. Supply Current on VCCIO(A:D)	VCCIO(A:D) = 3.6 V VCCIO(A:D) = 2.5 V VCCIO(A:D) = 1.8 V	-	10 10 20	µA

a. Capacitance is sample tested only.
b. Only one output at a time. Duration should not exceed 30 seconds.

Table 22: ASSP I/O DC Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
I_I	I or I/O Input Leakage Current	V_I = V3V or GND	-1		1	µA
I_{OZ}	3-state Output Leakage Current	V_I = V3V or GND	-		TBD	µA
C_I	I/O Input Capacitance[a]	-		7		pF
C_{CLOCK}	Clock Input Capacitance[a]	-		7		pF
I_{OS}	Output Short Circuit Current[b]	V_O = GND V_O = V3V	TBD TBD		TBD TBD	mA mA
V_{RST_TL}	CPU_RESET_n, CPU_WARMRESET_n, EJTAG_TRST_n input low threshold	V3V = 3.3 V	0.90	1.24	1.46	V
V_{RST_TH}	CPU_RESET_n, CPU_WARMRESET_n, EJTAG_TRST_n input high threshold	V3V = 3.3 V	1.39	1.82	2.06	V
V_{RST_HYS}	CPU_RESET_n, CPU_WARMRESET_n, EJTAG_TRST_n input hysteresis	V3V = 3.3 V	0.49	0.51	0.54	V

a. Capacitance is sample tested only.
b. Only one output at a time. Duration should not exceed 30 seconds.

Table 23: Fabric DC Input and Output Levels[a]

	INREF		V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}
	V_{MIN}	V_{MAX}	V_{MIN}	V_{MAX}	V_{MIN}	V_{MAX}	V_{MAX}	V_{MIN}	mA	mA
LVTTL	n/a	n/a	-0.3	0.8	2.2	VCCIO(A:D) + 0.3	0.4	2.4	2.0	-2.0
LVCMOS2	n/a	n/a	-0.3	0.7	1.7	VCCIO(A:D) + 0.3	0.7	1.7	2.0	-2.0
LVCMOS18	n/a	n/a	-0.3	0.63	1.2	VCCIO(A:D) + 0.3	0.7	1.7	2.0	-2.0
GTL+	0.88	1.12	-0.3	INREF(A:D) - 0.2	INREF(A:D) + 0.2	VCCIO(A:D) + 0.3	0.6	n/a	40	n/a
PCI	n/a	n/a	-0.3	0.3 x VCCIO(A:D)	0.5 x VCCIO(A:D)	VCCIO(A:D) + 0.5	0.1 x VCCIO(A:D)	0.9 x VCCIO(A:D)	1.5	-0.5
SSTL2	1.15	1.35	-0.3	INREF(A:D) - 0.18	INREF(A:D) + 0.18	VCCIO(A:D) + 0.3	0.74	1.76	7.6	-7.6
SSTL3	1.3	1.7	-0.3	INREF(A:D) - 0.2	INREF(A:D) + 0.2	VCCIO(A:D) + 0.3	1.10	1.90	8	-8

a. The data provided in **Table 23** are JEDEC and PCI Specifications. QuickLogic devices either meet or exceed these requirements.

NOTE: All CLK and IOCTRL input pins are clamped to the V3V rail. Therefore, these pains can be driven up to V3V+0.3V.

Figure 24 through **Figure 24** show the VIL and VIH characteristics for Fabric I/O and clock pins.

Figure 24: VIL Maximum for Fabric I/O

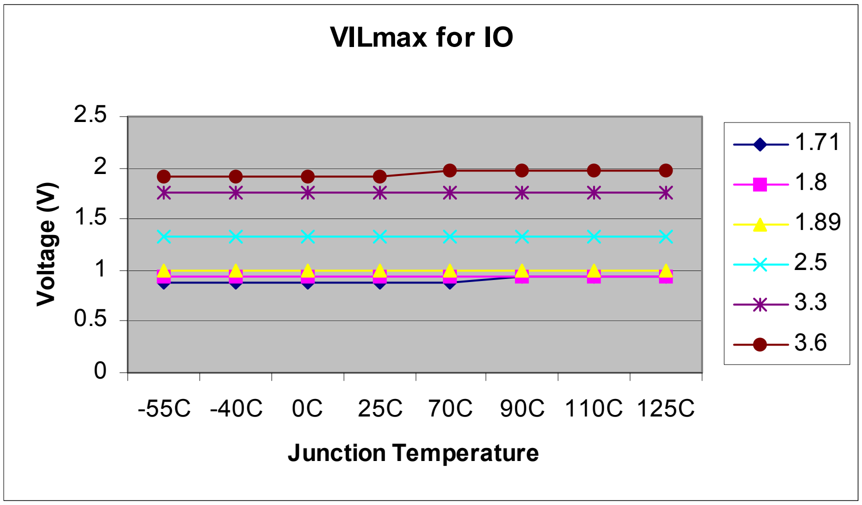


Figure 25: VIH Minimum for Fabric I/O

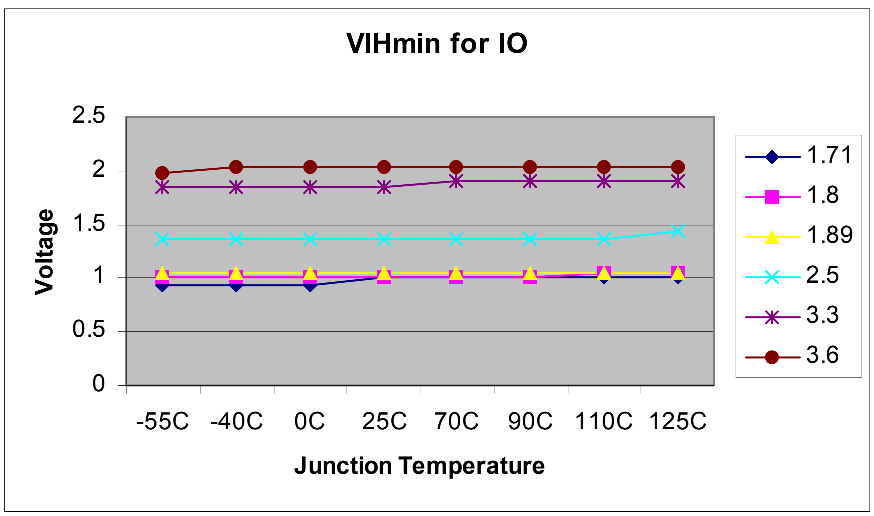


Figure 26: VIL Maximum for Fabric CLOCK Pins

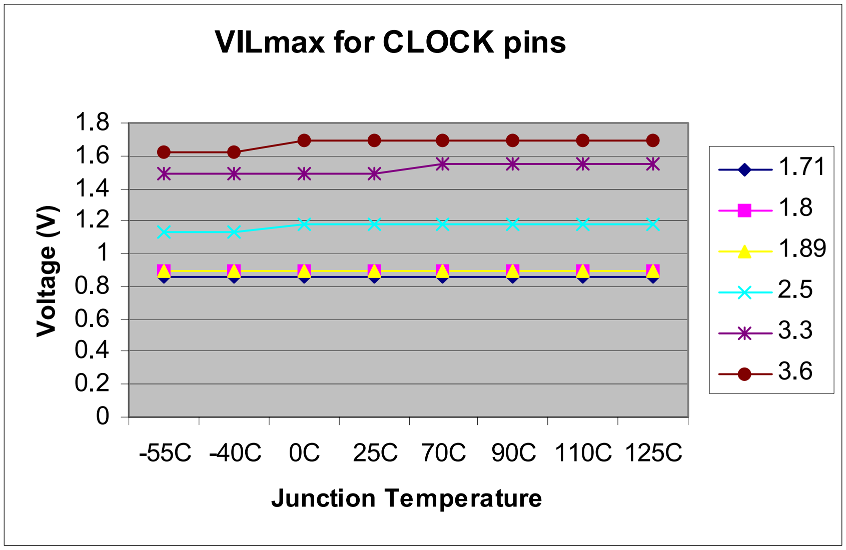


Figure 27: VIH Minimum for Fabric CLOCK Pins

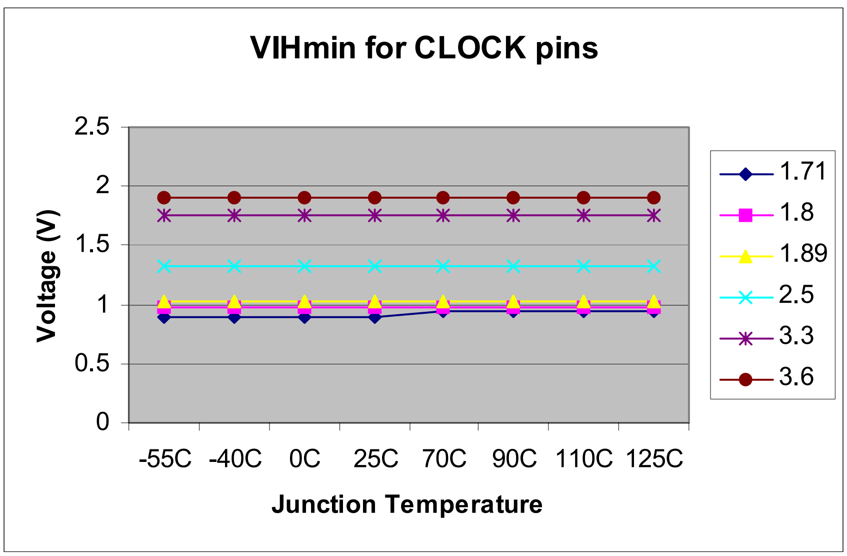


Figure 28 through **Figure 32** show the output drive characteristics for the Fabric I/Os across various voltages and temperatures.

Figure 28: Drive Current at VCCIO = 1.71 V

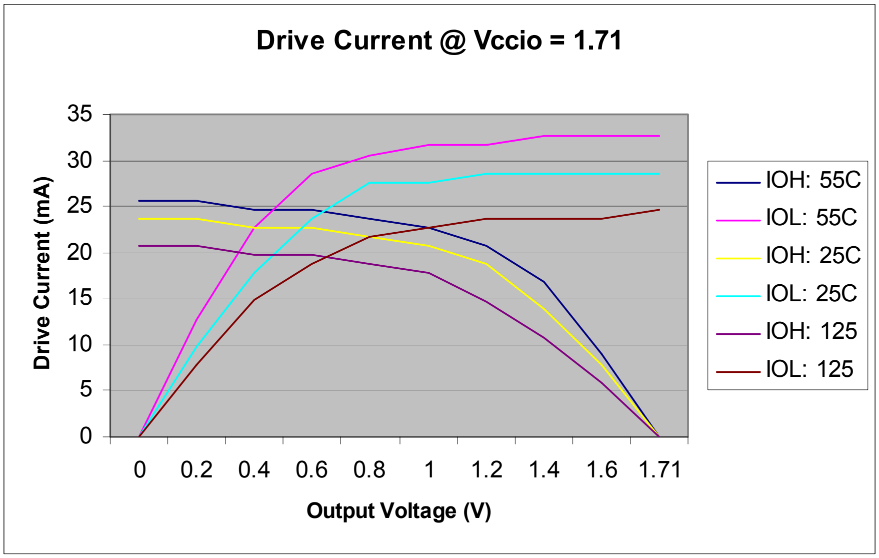


Figure 29: Drive Current at VCCIO = 1.8 V

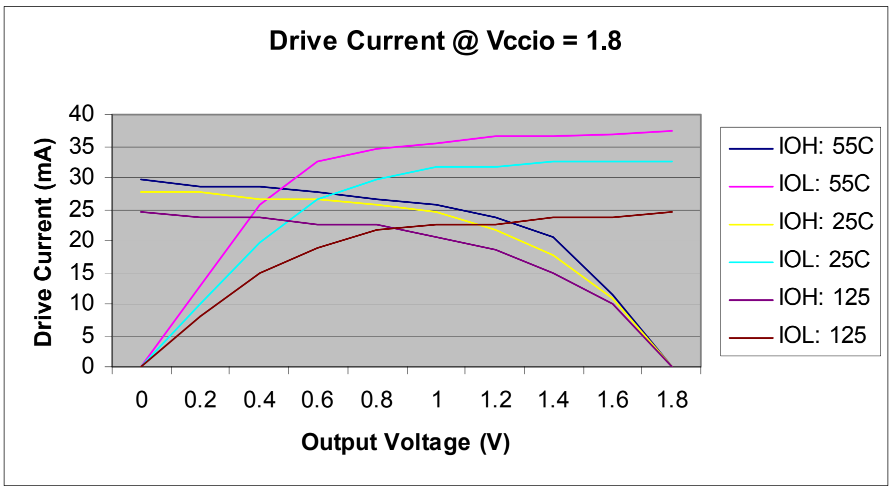


Figure 30: Drive Current at VCCIO = 2.5 V

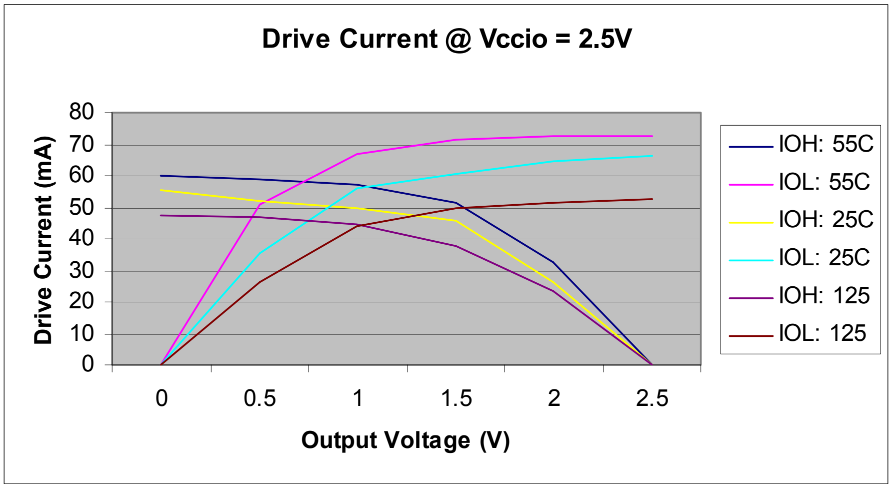


Figure 31: Drive Current at VCCIO = 3.3 V

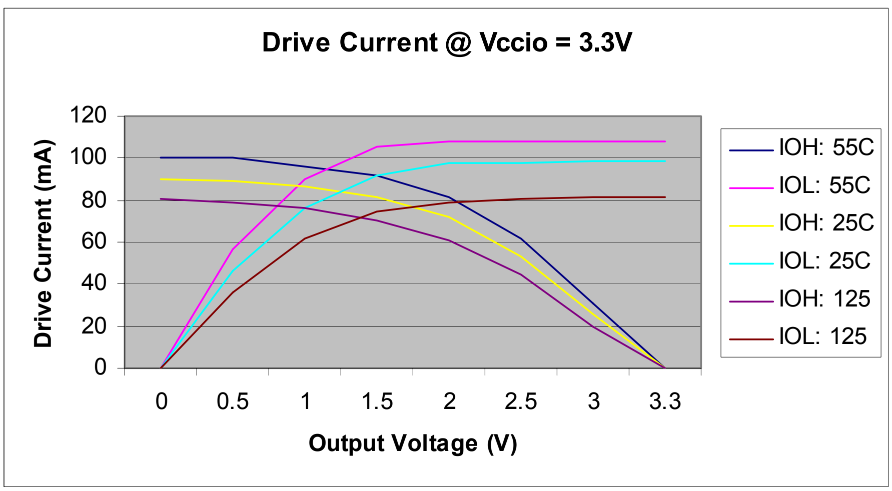
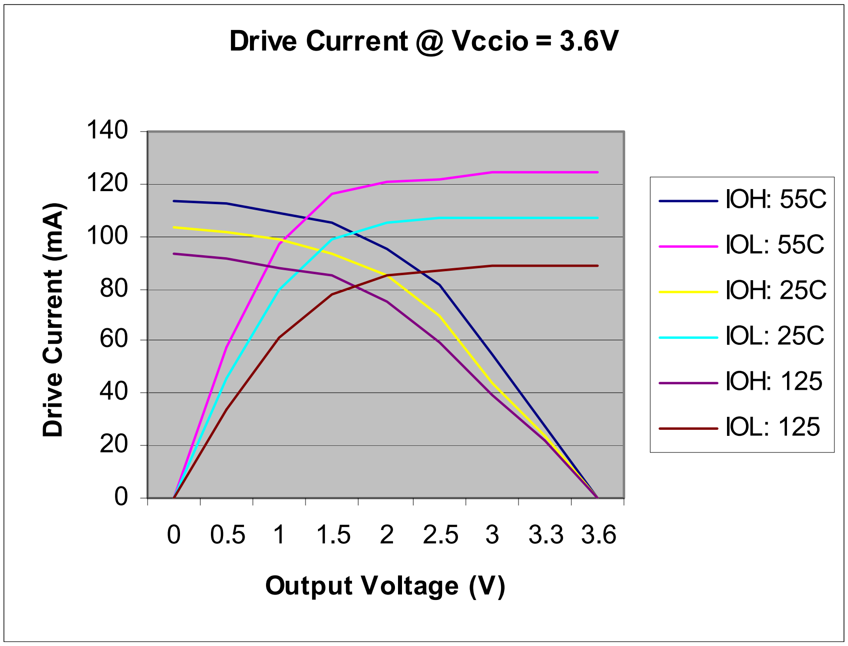


Figure 32: Drive Current at VCCIO = 3.6 V



AC Characteristics

The AC Specifications in this section are shown at V_{CC} = 1.95 V, TA = 25° C, Worst Case Corner, Fabric Speed Grade = -2 (K = 1.01) unless otherwise indicated.

ASSP PLL

Table 24: ASSP PLL Timing Parameters

Peak to Peak Jitter	VCO Frequency Range	Minimum Lock Frequency	Duty Cycle	Crystal Accuracy	Lock Time
200 ps	100 to 300 MHz	80 MHz	45% / 55%	±100 PPM	20 µs

SDRAM Controller

Figure 33: SDRAM Waveforms

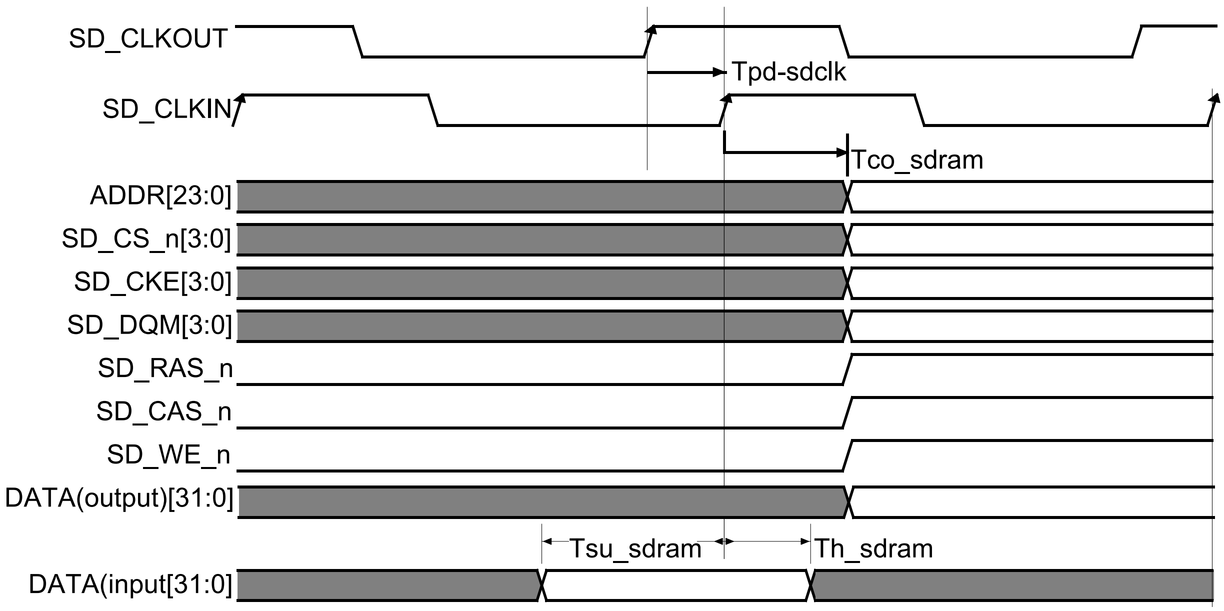


Table 25: SDRAM AC Timing

Symbol	Parameter[a]	QL902M175		QL902M200		Units
		Min.	Max.	Min.	Max.	
T_{CO}	Clock to out, all control and data signals	1.8	7.1	1.6	6.2	ns
T_{SU}	Setup time, read data	1.1	-	0.95	-	ns
T_{H}	Hold time, read data	0.22	-	0.19	-	ns
$T_{PD\text{-}SDCLK}$	Maximum allowed delay, SD_CLKOUT to SD_CLKIN	-	2.81	-	2.45	ns

a. All timing is measured with respect to the rising edge of SD_CLKIN. All measurements are based on I/Os with 35 pF load except for SD_CLKOUT, which has a load of 15 pF.

I/O Peripheral Controller

Figure 34: SRAM Read Waveforms

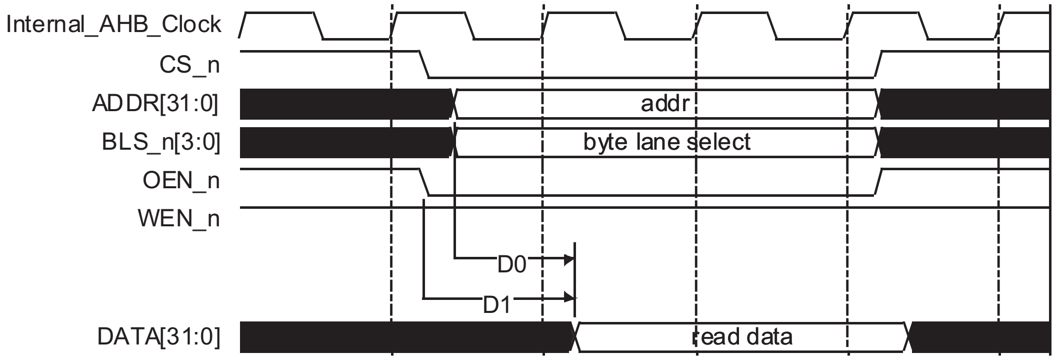


Table 26: SRAM AC Read Timing Requirements

Symbol	Parameter	QL902M175		QL902M200		Units
		Min.	Max.	Min.	Max.	
D0	Access time, address and byte lane output to read data valid[a]	-	8.3	-	7.3	ns
D1	Access time, output enable to read data valid	-	13.9	-	12.1	ns

a. Measurement is based on SD_CLKIN feedback with 0 ns delay and SD_CLKOUT load of 15 pF. Allowed access time will be decreased by SD_CLKIN to SD_CLKOUT delay.

Figure 35: SRAM Write Waveforms

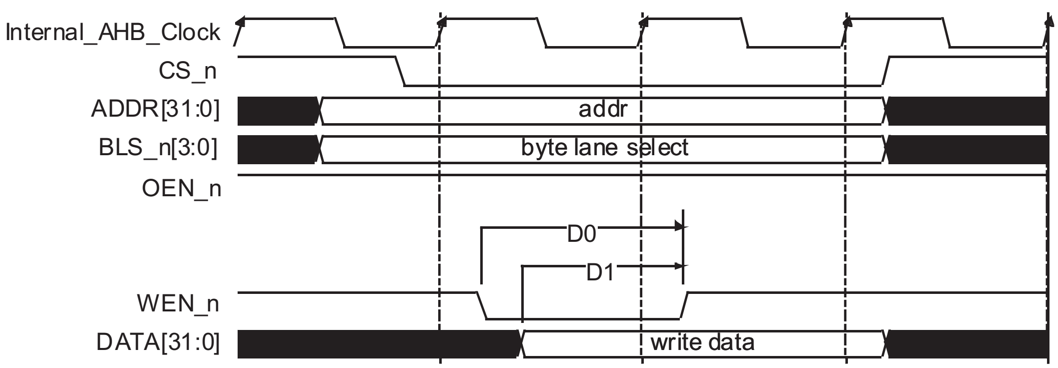


Table 27: SRAM AC Write Timing Characteristics

Symbol	Parameter	QL902M175		QL902M200		Units
		Min.	Max.	Min.	Max.	
D0	Write enable low pulse width	1	-	1	-	hclk period
D1	Write output data valid before rising edge of write enable[a]	4.7	-	4.1	-	ns

a. Measurement is based on SD_CLKIN feedback with 0 ns delay and SD_CLKOUT load of 15 pf. Setup time will be decreased by SD_CLKIN to SD_CLKOUT delay.

PCI Controller

Figure 36: PCI Waveforms

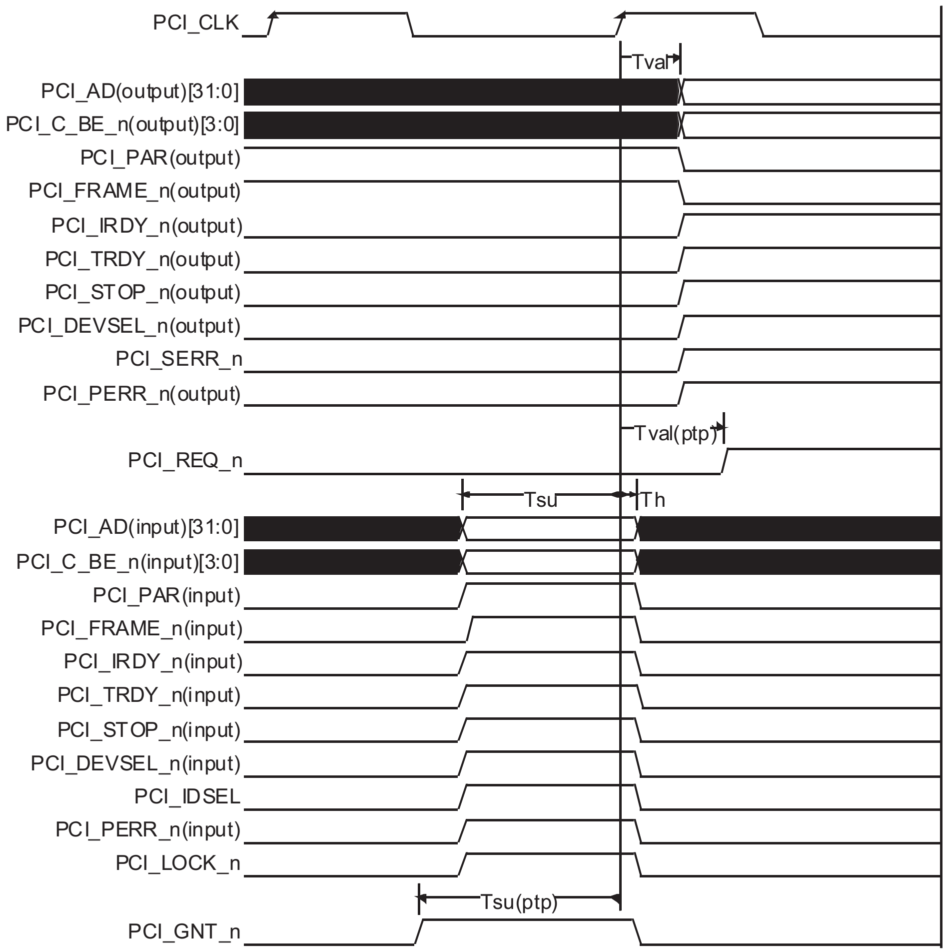


Table 28: PCI AC Timing

Parameter[a]		66 MHz		33 MHz		Units
		Min.	Max.	Min.	Max.	
t_{CYC}	PCI_CLK Cycle Time	15	-	30	-	ns
t_{HIGH}	PCI_CLK High Time	6	-	11	-	ns
t_{LOW}	PCI_CLK Low Time	6	-	11	-	ns
–	PCI_CLK Slew Rate	1.5	4	1	4	V/ns
t_{VAL}	PCI_CLK to Signal Valid Delay	2	6	2	11	ns
$t_{VAL\ (PTP)}$	PCI_CLK to Signal Valid Delay point-to-point signals[b]	2	6	2	12	ns
t_{ON}	Float to Active Delay	2	-	2	-	ns
t_{OFF}	Active to Float Delay	-	14	-	28	ns
t_{SU}	Input Setup Time to PCI_CLK bused signals	3	-	7	-	ns
$t_{SU\ (PTP)}$	Input Setup Time to PCI_CLK point-to-point	5	-	10, 12	-	ns
t_H	Input Hold Time from PCI_CLK	0	-	0	-	ns
t_{RST}	Reset Active Time after power stable	1	-	1	-	ms
$t_{RST\text{-}CLK}$	Reset Active Time after PCI_CLK stable	100	-	100	-	µs
$t_{RST\text{-}OFF}$[c]	Reset Active to output float delay		40		40	ns
t_{RHFA}	PCI_RST_n high to first configuration access	2	-	2	-	clocks
t_{RHFF}	PCI_RST_n high to first PCI_FRAME_n assertion	5	-	5	-	clocks

a. All PCI pins are synchronous to the PCI clock except for PCI_RST_n and PCI_INTA_n.
b. Point-to-point signals include PCI_REQ_n and PCI_GNT_n.
c. All output drivers must be 3-stated when PCI_RST_n is active.

System SRAM

Figure 37: System SRAM Block Diagram

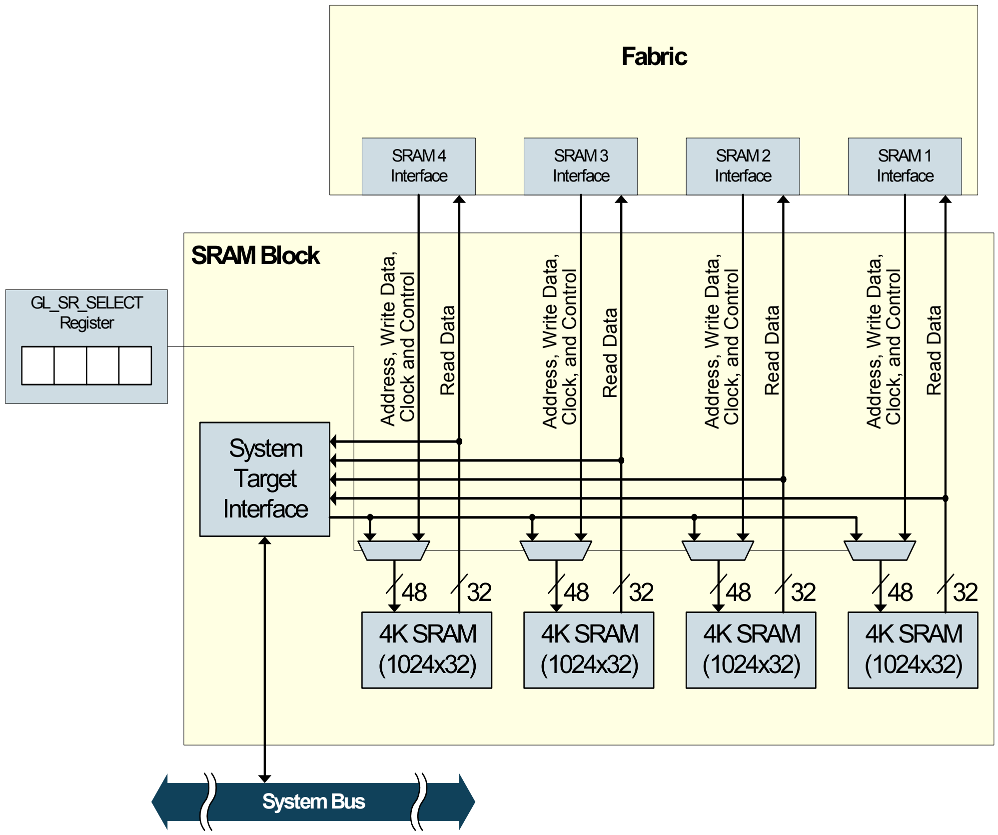


Figure 38: System SRAM Fabric Interface Read Timing (one SRAM interface shown)

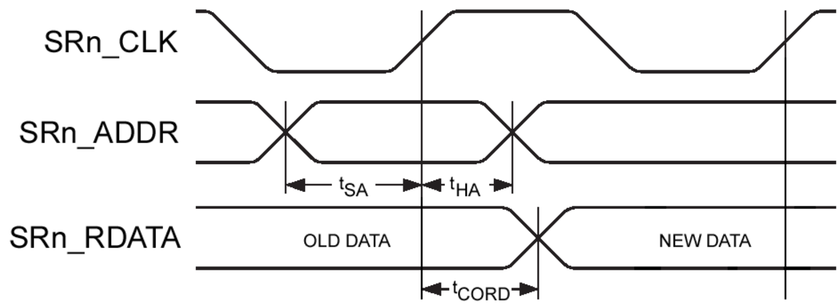


Figure 39: System SRAM Fabric Interface Write Timing (one SRAM interface shown)

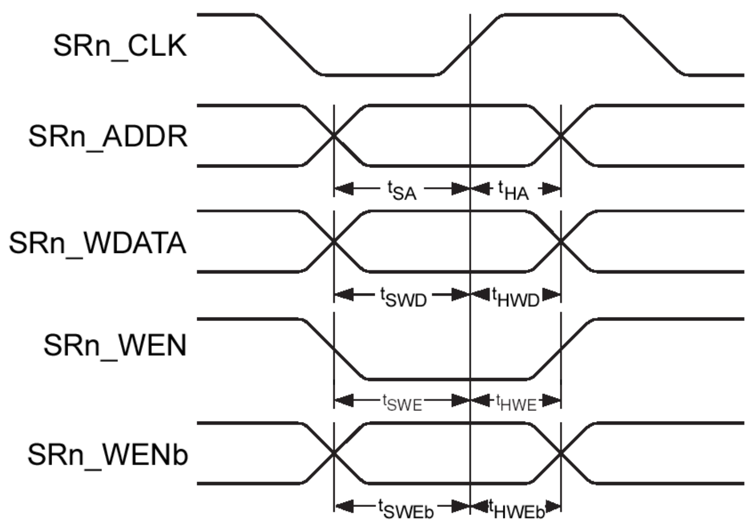


Table 29: System SRAM Fabric Interface Timing

Symbol	Parameter	QL902M175		QL902M200		Units
		Min.	Max.	Min.	Max.	
t_{SA}	Setup time, address input	1.40	-	1.23	-	ns
t_{HA}	Hold time, address input	0.48	-	0.42	-	ns
t_{SWD}	Setup time, write data input	1.81	-	1.59	-	ns
t_{HWD}	Hold time, write data input	0.56	-	0.49	-	ns
t_{SWE}	Setup time, write enable input	0.60	-	0.53	-	ns
t_{HWE}	Hold time, write enable input	0.54	-	0.48	-	ns
t_{SWEb}	Setup time, byte write enable input	1.13	-	0.99	-	ns
t_{HWEb}	Hold time, byte write enable input	0.94	-	0.82	-	ns
t_{CORD}	Clock-to-out time, read data output	-	5.17	-	4.52	ns

NOTE: For timing values related to each individual ASSP SRAM block, refer to **Table 46** and **Table 47**.

Logic Cells

Figure 40: QL902M Logic Cell

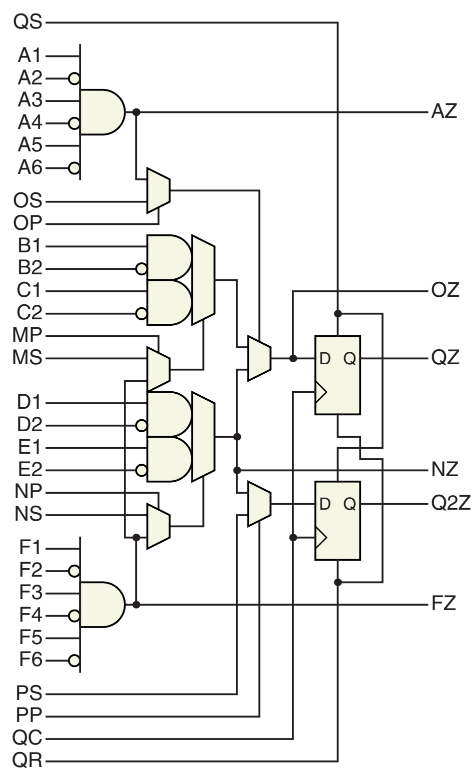


Figure 41: Logic Cell Flip-Flop

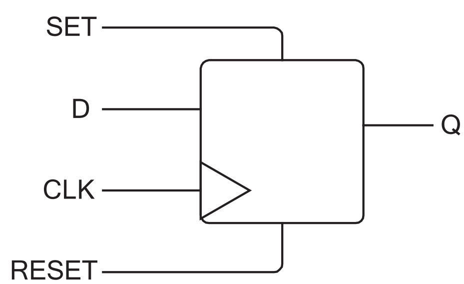


Figure 42: Logic Cell Flip-Flop Timings—First Waveform

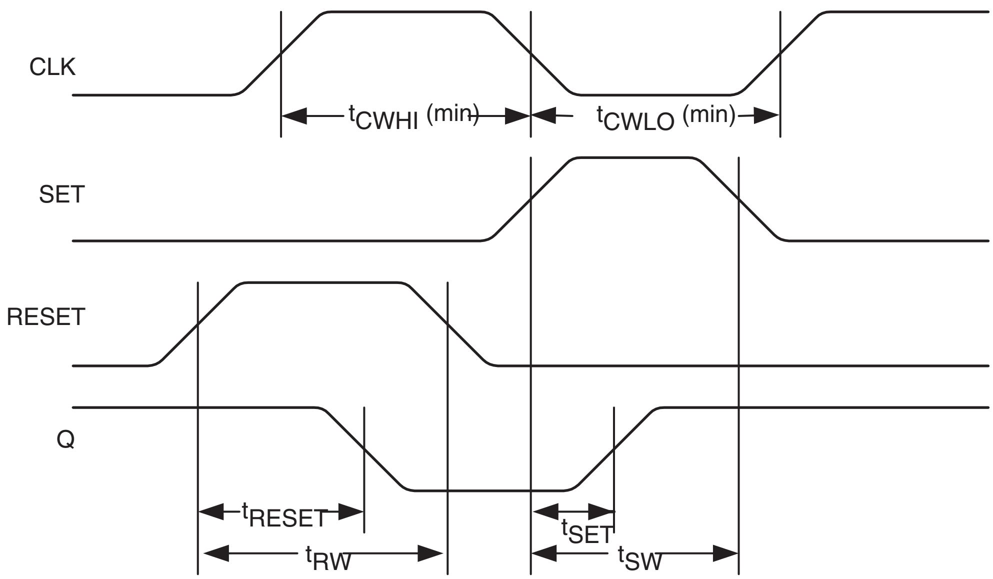


Figure 43: Logic Cell Flip-Flop Timings—Second Waveform

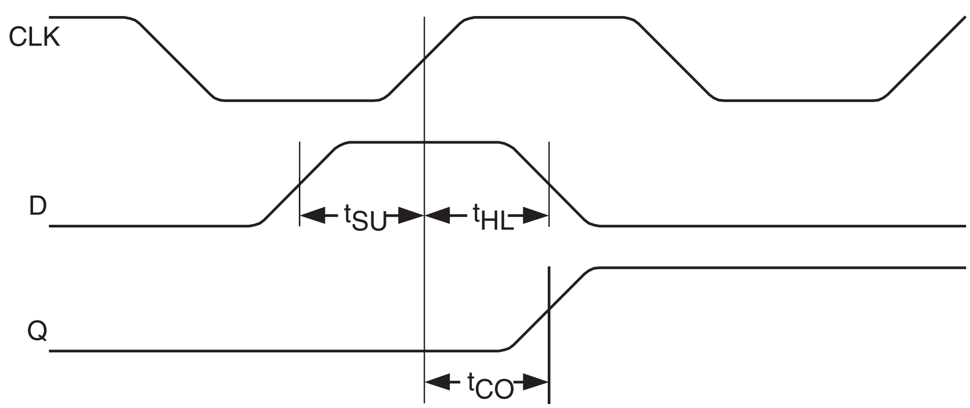


Table 30: Logic Cells

Symbol	Parameter	Value	
Logic Cells		Min.	Max.
t_{PD}	Combinatorial Delay of the longest path: time taken by the combinatorial circuit to output	0.28 ns	0.95 ns
t_{SU}	Setup time: time the synchronous input of the flip-flop must be stable before the active clock edge	0.25 ns	-
t_{HL}	Hold time: time the synchronous input of the flip-flop must be stable after the active clock edge	0 ns	-
t_{CO}	Clock-to-out delay: the amount of time taken by the flip-flop to output after the active clock edge.	0.22 ns	0.52 ns
t_{CWHI}	Clock High Time: required minimum time the clock stays high	0.46 ns	-
t_{CWLO}	Clock Low Time: required minimum time that the clock stays low	0.46 ns	-
t_{SET}	Set Delay: time between when the flip-flop is “set” (high and when the output is consequently “set” (high)	-	0.69 ns
t_{RESET}	Reset Delay: time between when the flip-flop is ”reset” (low) and when the output is consequently “reset” (low)	-	1.09 ns
t_{SW}	Set Width: time that the SET signal must remain high/low	0.3 ns	-
t_{RW}	Reset Width: time that the RESET signal must remain high/low	0.3 ns	-

Dual-Port SRAM Modules

Figure 44: RAM Module

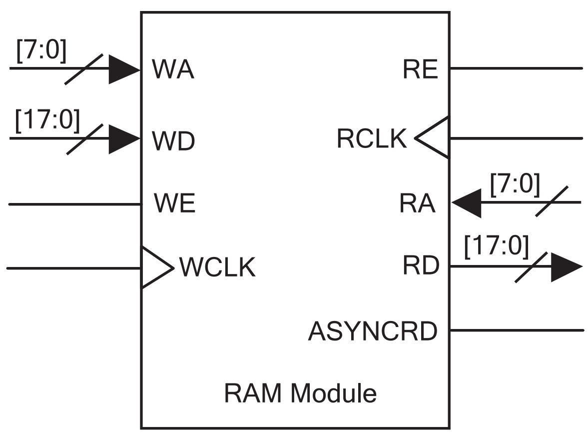


Figure 45: RAM Cell Synchronous Write Timing

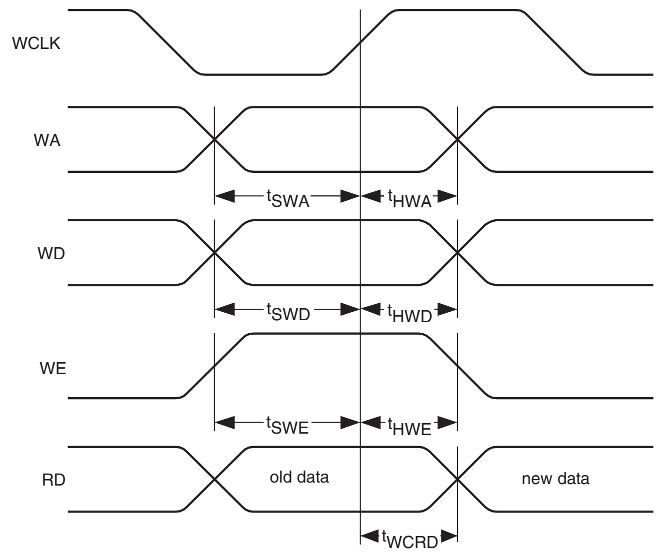


Table 31: RAM Cell Synchronous Write Timing

Symbol	Parameter	Value	
		Min.	Max.
RAM Cell Synchronous Write Timing			
t_{SWA}	WA setup time to WCLK: time the WRITE ADDRESS must be stable before the active edge of the WRITE CLOCK	0.675 ns	-
t_{HWA}	WA hold time to WCLK: time the WRITE ADDRESS must be stable after the active edge of the WRITE CLOCK	0 ns	-
t_{SWD}	WD setup time to WCLK: time the WRITE DATA must be stable before the active edge of the WRITE CLOCK	0.654 ns	-
t_{HWD}	WD hold time to WCLK: time the WRITE DATA must be stable after the active edge of the WRITE CLOCK	0 ns	-
t_{SWE}	WE setup time to WCLK: time the WRITE ENABLE must be stable before the active edge of the WRITE CLOCK	0.623 ns	-
t_{HWE}	WE hold time to WCLK: time the WRITE ENABLE must be stable after the active edge of the WRITE CLOCK	0 ns	-
t_{WCRD}	WCLK to RD (WA = RA): time between the active WRITE CLOCK edge and the time when the data is available at RD	-	4.38 ns

Figure 46: RAM Cell Synchronous and Asynchronous Read Timing

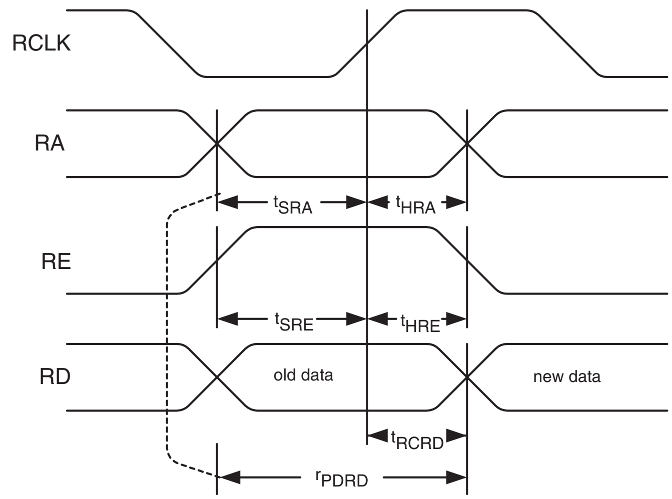


Table 32: RAM Cell Synchronous and Asynchronous Read Timing

Symbol	Parameter	Value	
		Min.	Max.
RAM Cell Synchronous Read Timing			
t_{SRA}	RA setup time to RCLK: time the READ ADDRESS must be stable before the active edge of the READ CLOCK	0.686 ns	-
t_{HRA}	RA hold time to RCLK: time the READ ADDRESS must be stable after the active edge of the READ CLOCK	0 ns	-
t_{SRE}	RE setup time to WCLK: time the READ ENABLE must be stable before the active edge of the READ CLOCK	0.243 ns	-
t_{HRE}	RE hold time to WCLK: time the READ ENABLE must be stable after the active edge of the READ CLOCK	0 ns	-
t_{RCRD}	RCLK to RD: time between the active READ CLOCK edge and the time when the data is available at RD	-	4.38 ns
RAM Cell Asynchronous Read Timing			
r_{PDRD}	RA to RD: time between when the READ ADDRESS is input and when the DATA is output	-	2.06 ns

ECUs

Figure 47: ECU Block Diagram

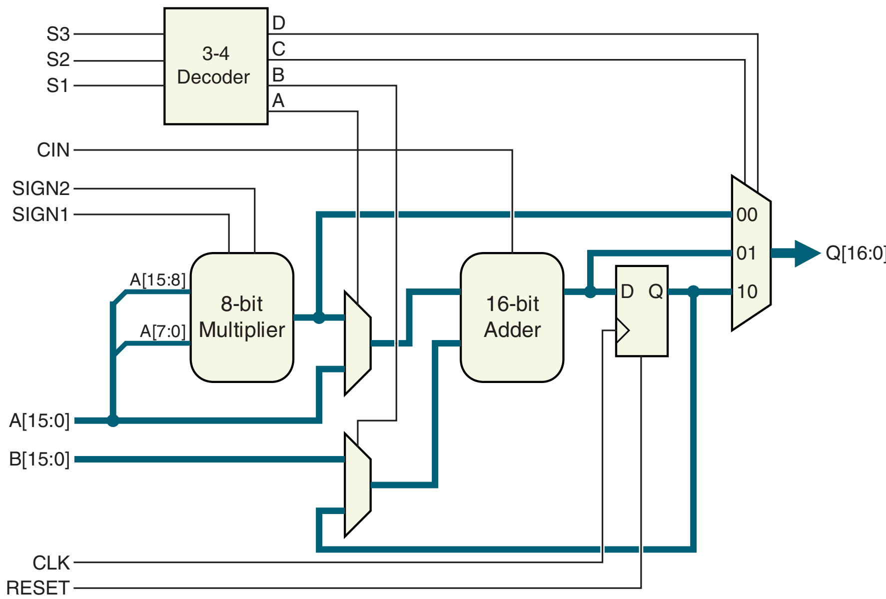


Table 33: ECU Mode Select Criteria

Instruction			Operation	ECU Performance[a], -2 WCC		
S1	A2	S3		t_{PD}	t_{SU}	t_{CO}
0	0	0	Multiply	6.6 ns max.	-	-
0	0	1	Multiply-Add	8.8 ns max.	-	-
0	1	0	Accumulate[b]	-	3.9 ns min.	1.2 ns max.
0	1	1	Add	3.1 ns max.	-	-
1	0	0	Multiply (registered)[c]	-	9.6 ns min.	1.2 ns max.
1	0	1	Multiply-Add (registered)	-	9.6 ns min.	1.2 ns max.
1	1	0	Multiply-Accumulate	-	9.6 ns min.	1.2 ns max.
1	1	1	Add (registered)	-	3.9 ns min.	1.2 ns max.

a. t_{PD}, t_{SU}, and t_{CO} do not include routing paths in/out of the ECU block.
b. Internal feedback path in ECU restricts max. clk frequency to 238 MHz.
c. B (15:0) set to zero.

NOTE: Timing numbers in **Table 33** represent -2 Worst Case Commercial conditions.

Fabric PLL

Figure 48: QL902M Fabric PLL Block Diagram

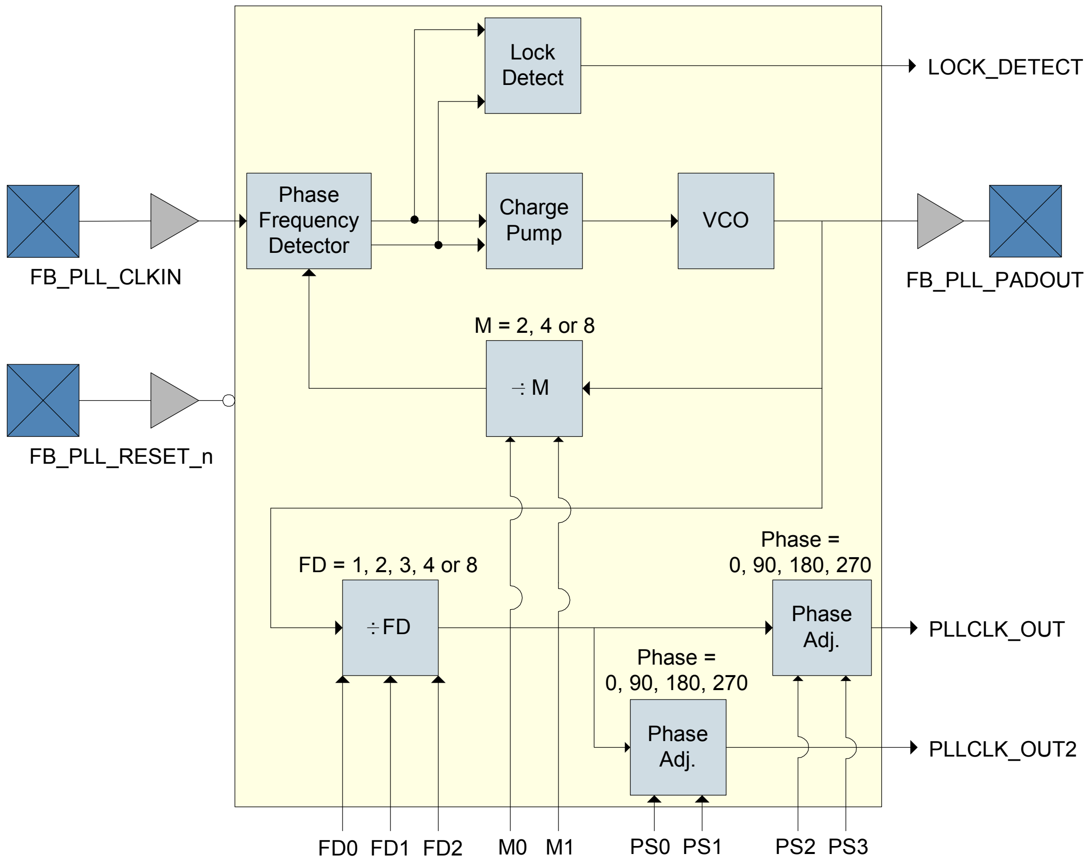


Table 34: Fabric PLL Timing Parameters

Peak to Peak Jitter	VCO Frequency Range	Minimum Lock Frequency	Duty Cycle	Crystal Accuracy	Lock Time	Propagation Delay FB_PLL_CLKIN to FB_PLL_PADOUT
200 ps	100 to 300 MHz	80 MHz	45% / 55%	±100 PPM	20 µs	TBD ns

Clock Network

Table 35: I/O Control Network/Local High-Drive

Destination	From Pad (max.)	From Array (max.)
I/O (far)	1.00 ns	1.14 ns
I/O (near)	0.63 ns	0.78 ns
Skew	0.37 ns	0.36 ns

Figure 49: Dedicated Clock Structure Schematic

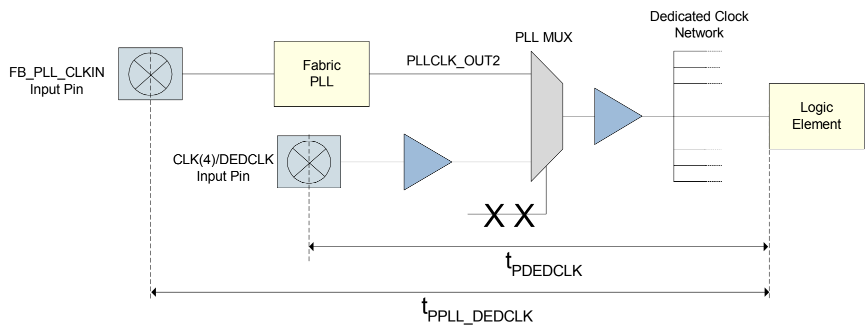


Table 36: Dedicated Clock Network Performance

Symbol	Parameters	Value	
		Min.	Max.
$t_{PDEDCLK}$	Delay from dedicated clock input pin to logic cell flip-flop	-	1.69 ns
$t_{SKEWDEDCLK}$	Skew on dedicated clock network	-	0.25 ns
t_{PPLL_DEDCLK}	Delay from Fabric PLL input pin to logic cell flip-flop	-	2.20 ns

Figure 50: Global Clock Structure Schematic

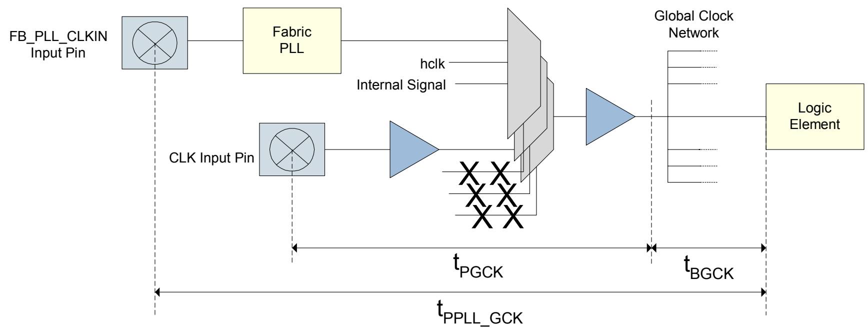


Table 37: Global Clock Network Performance

Symbol	Parameter	Value	
		Min.	Max.
t_{PGCK}	Global clock pin delay to quad net	-	1.95 ns
t_{BGCK}	Global clock tree delay (quad net to logic cell flip-flop)	-	0.28 ns
$t_{SKEWGCK}$	Skew on global clock network	-	0.25 ns
t_{PPLL_GCK}	Global clock tree delay, Fabric PLL input to logic cell flip-flop	-	TBD ns

I/O Cells

Figure 51: QL902M Input Register Cell

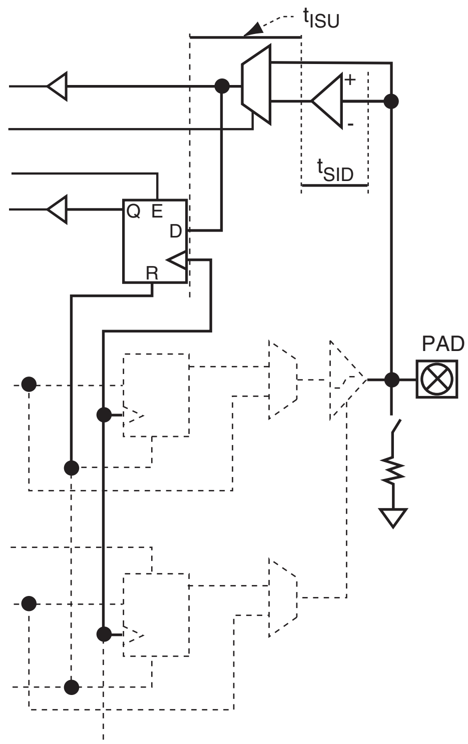


Table 38: Standard Input Delays

Symbol	Parameter	Value	
Standard Input Delays	To get the total input delay add this delay to t_{ISU}[a]	Min.	Max.
t_{SID} (LVTTL)	LVTTL input delay: Low Voltage TTL for 3.3 V applications	-	0.82 ns
t_{SID} (LVCMOS2)	LVCMOS2 input delay: Low Voltage CMOS for 2.5 V and lower applications	-	0.82 ns
t_{SID} (LVCMOS18)	LVCMOS18 input delay: Low Voltage CMOS for 1.8 V applications	-	TBD ns
t_{SID} (GTL+)	GTL+ input delay: Gunning Transceiver Logic	-	0.94 ns
t_{SID} (SSTL3)	SSTL3 input delay: Stub Series Terminated Logic for 3.3 V	-	0.94 ns
t_{SID} (SSTL2)	SSTL2 input delay: Stub Series Terminated Logic for 2.5 V	-	0.94 ns
t_{SID} (PCI)	PCI input delay: Peripheral Component Interconnect for 3.3 V	-	0.82 ns

a. See **Table 39** for t_{ISU} value.

Figure 52: QL902M Input Register Cell Timing

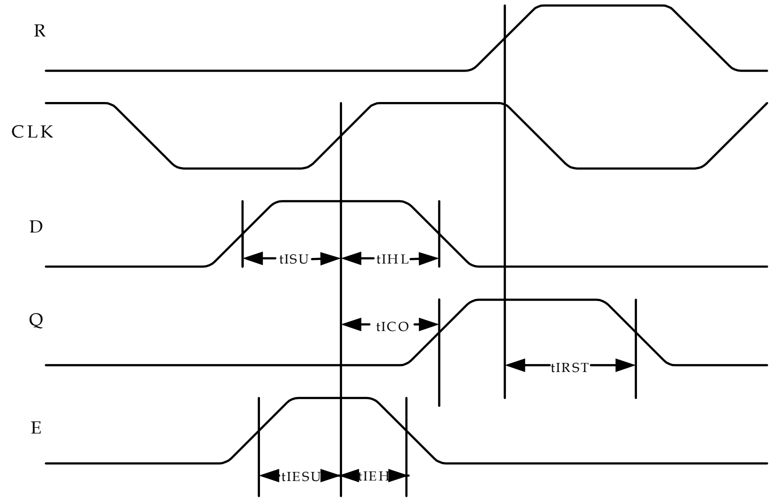


Table 39: Input Register Cell

Symbol	Parameter	Value	
		Min.	Max.
Input Register Cell Only			
t_{ISU}	**Input register setup time:** time the synchronous input of the flip-flop must be stable before the active clock edge.	2.15 ns	-
t_{IHL}	**Input register hold time:** time the synchronous input of the flip-flop must be stable after the active clock edge.	0 ns	-
t_{ICO}	**Input register clock-to-out:** time taken by the flip-flop to output after the active clock edge.	-	0.30 ns
t_{IRST}	**Input register reset delay:** time between when the flip-flop is “reset”(low) and when the output is consequently “reset” (low).	-	0.82 ns
t_{IESU}	**Input register clock enable setup time:** time “enable” must be stable before the active clock edge.	0.40 ns	-
t_{IEH}	**Input register clock enable hold time:** time “enable” must be stable after the active clock edge.	0 ns	-

Figure 53: QL902M Output Register Cell

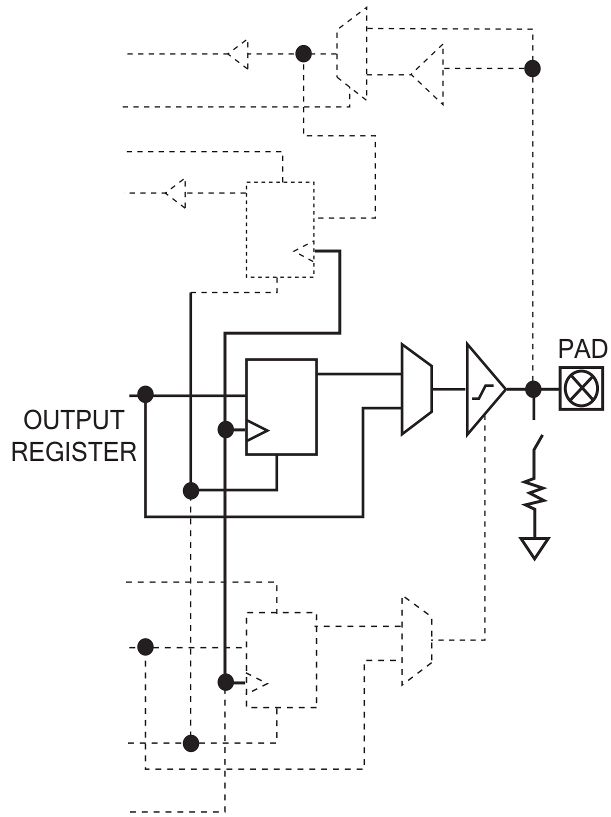


Figure 54: QL902M Output Register Cell Timing

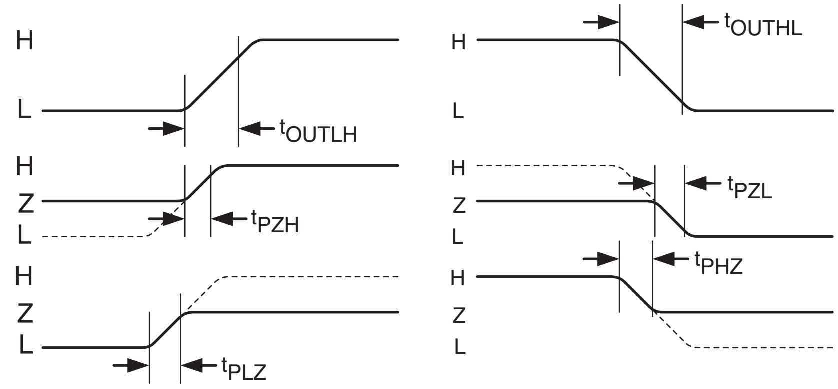


Table 40: Output Register Cell

Symbol	Parameter	Value	
		Min.	Max.
Output Register Cell Only			
t_{OUTLH}	**Output Delay low to high** (90% of H)	-	4.46 ns (fast slew) 6.12 ns (slow slew)
t_{OUTHL}	**Output Delay high to low** (10% of L)	-	3.31 ns (fast slew) 5.50 ns (slow slew)
t_{PZH}	**Output Delay tri-state to high** (90% of H)	-	4.19 ns (fast slew) 6.22 ns (slow slew)
t_{PZL}	**Output Delay tri-state to low** (10% of L)	-	4.19 ns (fast slew) 4.35 ns (slow slew)
t_{PHZ}	**Output Delay high to tri-state**	-	5.97 ns (fast slew) 3.79 ns (slow slew)
t_{PLZ}	**Output Delay low to tri-state**	-	3.58 ns (fast slew) 5.75 ns (slow slew)
t_{COP}	**Clock-to-out delay:** time taken by the flip-flop to output after the active clock edge.	-	6.50 ns (fast slew) TBD ns (slow slew)
t_{OUTSU}	**Output register setup time:** time the synchronous input of the flip-flop must be stable before the active clock edge.	TBD	-
t_{OUTH}	**Output register hold time:** time the synchronous input of the flip-flop must be stable after the active clock edge.	TBD	-
t_{OUTRST}	**Output register reset delay:** time between when the flip-flop is “reset”(low) and when the output is consequently “reset” (low).	-	TBD
t_{OESU}	**Output enable register setup time:** time the synchronous input of the flip-flop must be stable before the active clock edge.	TBD	-
t_{OEH}	**Output enable register hold time:** time the synchronous input of the flip-flop must be stable after the active clock edge.	TBD	-
t_{OERST}	**Output enable register reset delay:** time between when the flip-flop is “reset”(low) and when the output is consequently “reset” (low).	-	TBD

Table 40: Output Register Cell (Continued)

Symbol	Parameter	Value	
		Min.	Max.
t_{OEESU}	**Output enable register clock enable setup time:** time “enable” must be stable before the active clock edge.	TBD	-
t_{OEEH}	**Output enable register clock enable hold time:** time “enable” must be stable after the active clock edge.	TBD	-
t_{OECOP}	**Output enable register clock-to-out delay:** time taken by the flip-flop to output after the active clock edge.	-	TBD

Table 41: Fabric Output Slew Rates @ VCCIO(A:D) = 3.3 V, 25° C

	Fast Slew	Slow Slew
Rising Edge	2.8 V/ns	1.0 V/ns
Falling Edge	2.86 V/ns	1.0 V/ns

Table 42: Fabric Output Slew Rates @ VCCIO(A:D) = 2.5 V, 25° C

	Fast Slew	Slow Slew
Rising Edge	1.7 V/ns	0.6 V/ns
Falling Edge	1.9 V/ns	0.6 V/ns

Table 43: Fabric Output Slew Rates @ VCCIO(A:D) = 1.8 V, 25° C

	Fast Slew	Slow Slew
Rising Edge	TBD V/ns	TBD V/ns
Falling Edge	TBD V/ns	TBD V/ns

Table 44: ASSP Output Slew Rates @ V3V = 3.3 V, 25° C by Signal Groups

ASSP Output Group[a]	Slew (V/ns) Maximum		Slew (V/ns) Minimum	
	Rising	Falling	Rising	Falling
SDRAM Signals (except SD_CLKOUT)	3.00	3.87	0.96	1.12
SD_CLKOUT	4.20	5.30	1.50	1.95
SRAM Control Signals	2.30	2/97	0.55	0.58
PCI Signals	5.35	3.20	1.90	2.43
UART Signals	1.33	1.59	0.30	0.30

a. Loads are as follows: SDRAM signals - 35 pF, SD_CLKOUT = 15 pF, SRAM signals = 50 pF, PCI signals = 10 pF, and UART signals = 50 pF.

ASSP to Fabric Timing

Table 45 through **Table 47** list the synchronous and asynchronous timing for the QL902M ASSP to Fabric interface ports and ASSP I/O pins. Note the following regarding the Fabric timing:

- fb_int is asynchronous and is synchronized inside the core.
- fb_bigendian is a static signal and reflects the value on the CPU_BIGENDIAN pin.
- pm_* and si_* signals are synchronous to the internal MIPS clock which is at least twice the hclk frequency. This internal clock is not driven to the Fabric.
- All AF_PCI_* signals are static.

ASSP to Fabric Synchronous Output Timing

Table 45: ASSP to Fabric Interface Synchronous Output Timing

Clock Reference	Signal	QL902M175 t_{CO} (ns max.)	QL902M200 t_{CO} (ns max.)
hclk	ahb_hready_in	6.05	5.29
hclk	ahbm_hgrant	7.63	6.68
hclk	ahbm_hrdata(31:0)	7.94	6.95
hclk	ahbm_hresp(1:0)	6.05	5.30
hclk	ahbs_haddr(31:0)	6.03	5.27
hclk	ahbs_hburst(2:0)	5.61	4.90
hclk	ahbs_hprot(3:0)	4.75	4.15
hclk	ahbs_hsel	6.23	5.45
hclk	ahbs_hsize(2:0)	5.63	4.93
hclk	ahbs_htrans(1:0)	6.23	5.45
hclk	ahbs_hwdata(31:0)	5.92	5.18
hclk	ahbs_hwrite	5.55	4.85
hclk	apbs_paddr(15:2)	3.06	2.68
hclk	apbs_penable	2.98	2.61
hclk	apbs_psel0	3.32	2.91
hclk	apbs_psel1	3.36	2.94
hclk	apbs_psel2	3.31	2.90
hclk	apbs_pwdata(31:0)	3.32	2.91
hclk	apbs_pwrite	3.05	2.67
hclk	hresetn	4.07	3.56
hclk	M1_MDC	2.88	2.52
hclk	M1_MDO	2.84	2.48
hclk	M1_MDO_EN_N	2.87	2.51
hclk	M2_MDC	3.24	2.84
hclk	M2_MDO	3.26	2.85
hclk	M2_MDO_EN_N	3.30	2.89
CPU_PLL_CLKOUT	pm_dcachehit	2.99	2.61
CPU_PLL_CLKOUT	pm_dcachemiss	2.85	2.50
CPU_PLL_CLKOUT	pm_dtlbhit	2.76	2.42
CPU_PLL_CLKOUT	pm_dtlbmiss	2.84	2.49
CPU_PLL_CLKOUT	pm_icachehit	2.86	2.50
CPU_PLL_CLKOUT	pm_icachemiss	2.70	2.36
CPU_PLL_CLKOUT	pm_instncomplete	2.60	2.27
CPU_PLL_CLKOUT	pm_itlbhit	2.72	2.38

Table 45: ASSP to Fabric Interface Synchronous Output Timing (Continued)

Clock Reference	Signal	QL902M175 t_{CO} (ns max.)	QL902M200 t_{CO} (ns max.)
CPU_PLL_CLKOUT	pm_itlbmiss	2.70	2.36
CPU_PLL_CLKOUT	pm_jtlbhit	2.61	2.28
CPU_PLL_CLKOUT	pm_jtlbmiss	2.69	2.35
CPU_PLL_CLKOUT	pm_wtbmerge	4.49	3.93
CPU_PLL_CLKOUT	pm_wtbnomerge	2.88	2.52
CPU_PLL_CLKOUT	si_rp	3.24	2.83
CPU_PLL_CLKOUT	si_sleep	2.27	1.99
hclk	tm_overflow2	2.94	2.57
hclk	tm_overflow3	2.85	2.50
hclk	tm_overflow4	2.94	2.57
SR1_CLK	SR1_RDATA(31:0)	4.49	3.93
SR2_CLK	SR2_RDATA(31:0)	5.17	4.52
SR3_CLK	SR3_RDATA(31:0)	4.30	3.76
SR4_CLK	SR4_RDATA(31:0)	4.04	3.53

Fabric to ASSP Interface Timing Requirements

Table 46: Fabric to ASSP Interface Timing Requirements

Clock Reference	Signal	QL902M175		QL902M200	
		t_{SU} (ns min)	t_H (ns min)	t_{SU} (ns min)	t_H (ns min)
hclk	ahbm_haddr(31:0)	3.64	1.39	3.18	1.21
hclk	ahbm_hburst(2:0)	3.33	0.03	2.91	0.03
hclk	ahbm_hbusreq	3.47	0	3.04	0
hclk	ahbm_hprot(3:0)	4.42	0	3.87	0
hclk	ahbm_hsize(2:0)	3.07	0.96	2.69	0.84
hclk	ahbm_htrans(1:0)	3.56	0.30	3.12	0.26
hclk	ahbm_hwdata(31:0)	3.06	1.23	2.68	1.07
hclk	ahbm_hwrite	2.98	0.86	2.61	0.75
hclk	ahbs_hrdata(31:0)	3.25	0.89	2.84	0.78
hclk	ahbs_hready_out	4.15	0.46	3.63	0.40
hclk	ahbs_hresp(1:0)	4.03	0.77	3.52	0.67
hclk	apbs_prdata0(31:0)	3.44	0.53	3.01	0.46
hclk	apbs_prdata1(31:0)	3.42	0.51	3.00	0.45
hclk	apbs_prdata2(31:0)	3.34	0.65	2.92	0.57
-	M1_COL	Asynchronous input – no requirement			
-	M1_CRS	Asynchronous input – no requirement			
hclk	M1_MDI	0	1.38	0	1.20
M1_RXCLK	M1_RXD(3:0)	0.67	0	0.58	0
M1_RXCLK	M1_RXDV	1.06	0	0.92	0
M1_RXCLK	M1_RXER	0.57	0	0.50	0
-	M2_COL	Asynchronous input – no requirement			
-	M2_CRS	Asynchronous input – no requirement			
hclk	M2_MDI	0	1.62	0	1.42
M2_RXCLK	M2_RXD(3:0)	0.93	0.35	0.81	0.31
M2_RXCLK	M2_RXDV	1.65	0	1.45	0
M2_RXCLK	M2_RXER	0.71	0.09	0.62	0.08
SR1_CLK	SR1_ADDR(11:2)	1.18	0.45	1.03	0.39
SR1_CLK	SR1_WDATA(31:0)	1.36	0.15	1.19	0.13
SR1_CLK	SR1_WEN	0.54	0	0.47	0
SR1_CLK	SR1_WENb(3:0)	0.27	0.66	0.24	0.58
SR2_CLK	SR2_ADDR(11:2)	0.93	0.48	0.81	0.42
SR2_CLK	SR2_WDATA(31:0)	1.81	0.48	1.59	0.42
SR2_CLK	SR2_WEN	0.53	0.54	0.46	0.48
SR2_CLK	SR2_WENb(3:0)	0.43	0.94	0.38	0.82

Table 46: Fabric to ASSP Interface Timing Requirements (Continued)

Clock Reference	Signal	QL902M175		QL902M200	
		t_{SU} (ns min)	t_H (ns min)	t_{SU} (ns min)	t_H (ns min)
SR3_CLK	SR3_ADDR(11:2)	0.53	0.34	0.46	0.30
SR3_CLK	SR3_WDATA(31:0)	1.45	0.56	1.27	0.49
SR3_CLK	SR3_WEN	0.21	0.26	0.19	0.23
SR3_CLK	SR3_WENb(3:0)	0.26	0.59	0.23	0.52
SR4_CLK	SR4_ADDR(11:2)	1.40	0.23	1.23	0.20
SR4_CLK	SR4_WDATA(31:0)	1.78	0.15	1.56	0.13
SR4_CLK	SR4_WEN	0.60	0.15	0.53	0.13
SR4_CLK	SR4_WENb(3:0)	1.13	0.52	0.99	0.46
hclk	tm_extclk1	0	1.58	0	1.38
hclk	tm_extclk2	0	1.57	0	1.37
hclk	tm_extclk3	0	1.57	0	1.37
hclk	tm_extclk4	0	1.61	0	1.41
hclk	tm_fbenable	0	1.60	0	1.40

ASSP Propagation Delays

Table 47: ASSP Propagation Delays

Starting Signal	Ending Signal	QL902M175 t_{PD} (ns max)	QL902M200 t_{PD} (ns max)
Path: Fabric to Fabric through ASSP			
ahbm_haddr(31:0)	ahbs_haddr(31:0)	1.80	1.57
ahbm_haddr(31:0)	ahbs_hsel	2.12	1.86
ahbm_hburst(2:0)	ahbm_hgrant	1.78	1.55
ahbm_hburst(2:0)	ahbs_hburst(2:0)	2.33	2.04
ahbm_hbusreq	ahbm_hgrant	2.66	2.32
ahbm_hprot(3:0)	ahbs_hprot(3:0)	1.30	1.14
ahbm_hsize(2:0)	ahbs_hsize(2:0)	1.64	1.43
ahbm_htrans(1:0)	ahbs_htrans(1:0)	1.86	1.63
ahbm_htrans(1:0)	ahbm_hgrant	2.31	2.02
ahbm_hwdata(31:0)	ahbs_hwdata(31:0)	1.80	1.57
ahbm_hwrite	ahbs_hwrite	1.71	1.49
ahbs_hrdata(31:0)	ahbm_hrdata(31:0)	2.37	2.07
ahbs_hready_out	ahb_hready_in	1.79	1.57
ahbs_hready_out	ahbm_hgrant	3.31	2.90
ahbs_hresp(1:0)	ahbm_hgrant	2.07	1.81
ahbs_hresp(1:0)	ahbm_hresp(1:0)	3.17	2.77
apbs_prdata0(31:0)	ahbm_hrdata(31:0)	2.75	2.41
apbs_prdata1(31:0)	ahbm_hrdata(31:0)	2.66	2.33
apbs_prdata2(31:0)	ahbm_hrdata(31:0)	2.57	2.25
Path: ASSP Input Pin to Fabric			
CPU_BIGENDIAN	fb_bigendian	1.97	1.72
M1_TXCLK	M1_TXD(3:0)	2.32	2.03
M1_TXCLK	M1_TXEN	2.62	2.29
M2_TXCLK	M2_TXD(3:0)	3.19	2.79
M2_TXCLK	M2_TXEN	3.14	2.75
CPU_PLL_CLKIN	hclk	1.64	1.43
Path: ASSP Input Pin to ASSP Output Pin			
CPU_PLL_CLKIN	CPU_PLL_CLKOUT	2.38	2.09

Package Thermal Characteristics

Thermal Resistance Equations:

$\theta_{JC} = (T_J - T_C)/P$

$\theta_{JA} = (T_J - T_A)/P$

$P_{MAX} = (T_{JMAX} - T_{AMAX})/\theta_{JA}$

Parameter Description:

θ_{JC}: Junction-to-case thermal resistance

θ_{JA}: Junction-to-ambient thermal resistance

T_J: Junction temperature

T_C: Case temperature

T_A: Ambient temperature

P: Power dissipated by the device while operating

P_{MAX}: The maximum power dissipation for the device

T_{JMAX}: Maximum junction temperature

T_{AMAX}: Maximum ambient temperature

NOTE: Maximum junction temperature (T_{JMAX}) is 125°C. To calculate the maximum power dissipation for a device package look up θ_{JA} from **Table 48**, pick an appropriate T_{AMAX} and use:

$P_{MAX} = (125°C - T_{AMAX})/\theta_{JA}$

Table 48: Package Thermal Characteristics

Package Description		θ_{JA} (º C/W) @ Various Flow Rates (m/sec)				θ_{JC} (º C/W)
Pin Count	Package Type	0	1	2	2.5	
544	PBGA	20	16.4	15.5	14.4	8.0

Power vs. Operating Frequency

The basic power equation which best models power consumption is given below:

$$P_{TOTAL}\ (mW) = 0.350 + f_{Fabric}\ [0.0031\ \eta_{LC} + 0.0948\ \eta_{CKBF} + 0.01\ \eta_{CLBF} + 0.0263\ \eta_{CKLD} + 0.543\ \eta_{RAM} + 0.20\ \eta_{PLL} + 0.0035\ \eta_{INP} + 0.0257\ \eta_{OUTP}] + 28.1\ f_{asspio} + 5.55\ f_{mips}$$

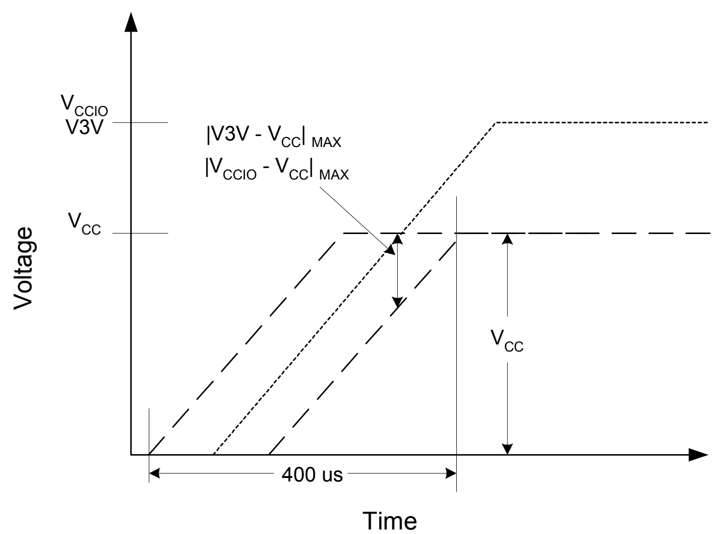
Where

η_{LC}	= number of logic cells in the design
η_{CKBF}	= number of clock buffers
η_{CLBF}	= number of column clock buffers
η_{CKLD}	= number of loads connected to the column clock buffers
η_{RAM}	= number of RAM blocks
η_{PLL}	= number of PLLs
η_{INP}	= number of input pins
η_{OUTP}	= number of output pins
f_{Fabric}	= average switching frequency of Fabric
f_{asspio}	= average switching frequency of ASSP I/O signals
f_{mips}	= CPU operational frequency

NOTE: To learn more about power consumption, please refer to Application Note 60 at http://www.quicklogic.com/images/appnote60.pdf.

Power-Up Sequencing

Figure 55: Power-Up Sequencing



When powering up a device, the VCC, VCCIO and V3V rails must take 400 µs or longer to reach the maximum value (refer to **Figure 55**).

NOTE: Ramping VCC, VCCIO and V3V to the maximum voltage faster than 400 µs can cause the device to behave improperly.

For users with a limited power budget, keep $(V3V - VCC)_{MAX}$ and $(VCCIO - VCC)_{MAX} \leq 500$ mV when ramping up the power supply.

Board Layout Recommendations

This section describes various recommendations for design and layout of the printed circuit board (PCB) that is used with the QL902M.

PLL Power Supply Filtering

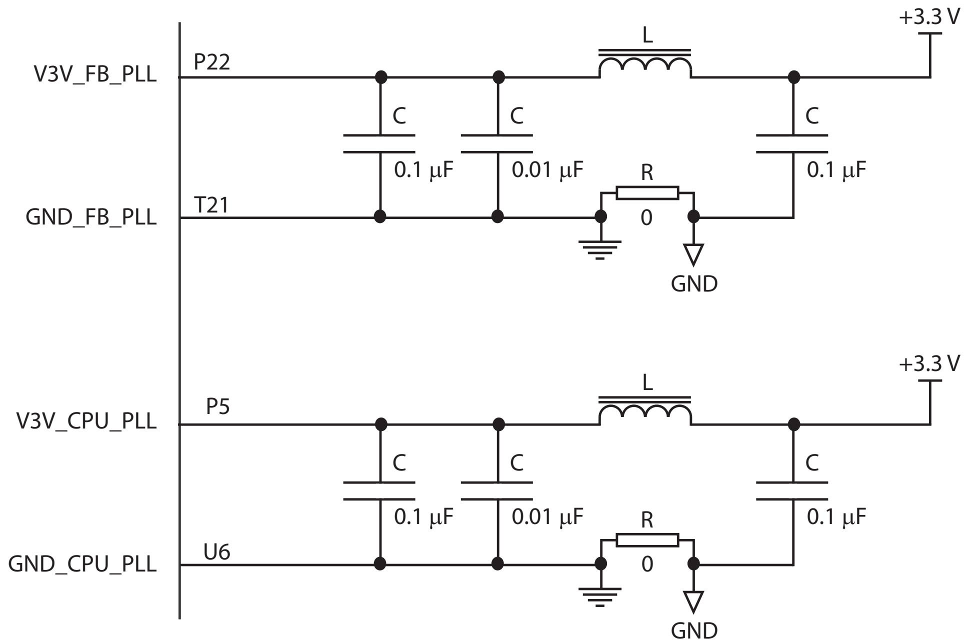
The PLL analog power supply inputs of the QL902M are:

Fabric PLL: GND_FB_PLL and V3V_FB_PLL

CPU PLL: GND_CPU_PLL and V3V_CPU_PLL

These signals must be filtered as shown in **Figure 56**:

Figure 56: PLL Power Supply Filtering Circuits



For best performance and noise immunity, separate or split power planes for digital ground (GND), analog ground (FB_PLL_GND and CPU_PLL_GND), digital V_{CC} (VCC), digital 3.3 V (V3V) and analog 3.3 V (FB_PLL_V3V and CPU_PLL_V3V) should be used.

Bypass Capacitor Guidelines

This section outlines the recommendations for capacitive bypassing for the various digital supplies of the QL902M. **Table 49** indicates the minimum recommended number of 0.1 µF and 0.01 µF capacitors for each digital supply input on the QL902M. A minimum total quantity of 48 bypass capacitors is recommended.

Table 49: Minimum Recommended Number of Bypass Capacitors for Each Digital Supply Input

Supply Input	Recommended Number of Capacitors 0.1 µF, X7R Type (or better)	Recommended Number of Capacitors 0.01 µF, X7R Type (or better)	Totals
VCC	8	8	16
VCCIO(A)	2	2	4
VCCIO(B)	2	2	4
VCCIO(C)	2	2	4
VCCIO(D)	2	2	4
V3V	8	8	16
Totals:	24	24	48

NOTE: All capacitors should be X7R type (or better) and 0603 package size is recommended.

Capacitor Placement

Eight capacitors (all for the V_{CC} supply) should be placed directly under the device (in the area between the 8x8 GND pads in the center of the device and the outer rows of signal pads) on the back side of the PCB. The remaining forty capacitors (for the remainder of V_{CC} and all other supplies) should be placed around the perimeter of the chip, also preferably on the back side of the board.

Listed below is the proximity location priority (1=highest). Higher priority filters and/or capacitors should be placed closest to the QL902M.

1. PLL bypass capacitors and filter circuitry
2. V_{CC} bypass capacitors
3. VCCIO/V3V bypass capacitors

Signal Descriptions

Pin Descriptions

Table 50: Pin Descriptions

<table>
<tr><th>Pin</th><th>I/O</th><th>Function</th></tr>
<tr><td colspan="3">PCI Signals</td></tr>
<tr><td>PCI_AD(31:0)</td><td>I/O</td><td>PCI Address and Data. PCI_AD(31:0) contain the multiplexed address and data. A bus transaction consists of a single address phase (or two address phases for 64-bit addresses) followed by one or more data phases. The QuickMIPS chip supports both read and write bursts.

The address phase occurs in the first clock cycle when PCI_FRAME_n is asserted. During the address phase, PCI_AD(31:0) contain a 32-bit physical address. For I/O, this is a byte address; for configuration and memory, it is a DWORD (32-bit) address. During data phases, PCI_AD(7:0) contain the least-significant byte, and PCI_AD(31:24) contain the most-significant byte.

Write data is stable and valid when PCI_IRDY_n is asserted; read data is stable and valid when PCI_TRDY_n is asserted. Data is transferred when both PCI_IRDY_n and PCI_TRDY_n are asserted.

Connect to GND if the PCI Controller is unused.</td></tr>
<tr><td>PCI_C_BE_n(3:0)</td><td>I/O</td><td>Bus Command and Byte Enables. Bus commands and byte enables are multiplexed on PCI_C_BE_n(3:0). During the address phase of a transaction (PCI_FRAME_n is asserted), PCI_C_BE_n(3:0) define the bus command as shown in the following table (only valid combinations are shown).

<table>
<tr><th>PCI_C_BE_n(3:0)</th><th>Bus Command</th></tr>
<tr><td>0000</td><td>Interrupt Acknowledge</td></tr>
<tr><td>0001</td><td>Special Cycle</td></tr>
<tr><td>0010</td><td>I/O Read</td></tr>
<tr><td>0011</td><td>I/O Write</td></tr>
<tr><td>0110</td><td>Memory Read</td></tr>
<tr><td>0111</td><td>Memory Write</td></tr>
<tr><td>1010</td><td>Configuration Read</td></tr>
<tr><td>1011</td><td>Configuration Write</td></tr>
<tr><td>1100</td><td>Memory Read Multiple</td></tr>
<tr><td>1101</td><td>Dual Address Cycle</td></tr>
<tr><td>1110</td><td>Memory Read Line</td></tr>
<tr><td>1111</td><td>Memory Write and Invalidate</td></tr>
</table>

During each data phase, PCI_C_BE_n(3:0) are byte enables. The byte enables are valid for the entire data phase and determine which byte lanes contain meaningful data. PCI_C_BE_n(0) applies to byte 0 (PCI_AD(7:0)) and PCI_C_BE_n(3) applies to byte 3 (PCI_AD(31:24)).

Connect to GND if the PCI Controller is unused.</td></tr>
<tr><td>PCI_DEVSEL_n</td><td>I/O</td><td>PCI Device Select. When asserted low, PCI_DEVSEL_n indicates the driving device has decoded its address as the target of the current access. As an input, PCI_DEVSEL_n indicates whether any device on the bus has responded.

Connect to 3.3 V if the PCI Controller is unused.</td></tr>
</table>

Table 50: Pin Descriptions (Continued)

Pin	I/O	Function
PCI_FRAME_n	I/O	**PCI Cycle Frame.** The current master asserts PCI_FRAME_n to indicate the beginning and duration of a bus transaction. While PCI_FRAME_n is asserted, data transfers continue. When PCI_FRAME_n is deasserted, the transaction is in the final data phase or has completed. Connect to 3.3 V if the PCI Controller is unused.
PCI_GNT_n	I	**PCI Grant.** A low assertion of PCI_GNT_n indicates to the agent that access to the bus has been granted. PCI_GNT_n is ignored while PCI_RST_n is asserted. Connect to 3.3 V if the PCI Controller is unused.
PCI_IDSEL	I	**PCI Initialization Device Select.** PCI_IDSEL is used as a chip select during configuration read and write transactions (PCI_C_BE_n(3:0) = 1010 or 1011). Connect to GND if the PCI Controller is unused.
PCI_INTA_n	O	**PCI Interrupt Acknowledge.** PCI_INTA_n is a level-sensitive interrupt driven by the QuickMIPS chip. PCI_INTA_n is asserted and deasserted asynchronously to the PCI_CLK. This interrupt remains asserted until the interrupt is cleared. Because the PCI interrupt controller is not built into the QuickMIPS ESP core, this pin is output only. However, such an interrupt controller can be built into the Fabric. Leave unconnected if the PCI Controller is unused.
PCI_IRDY_n	I/O	**PCI Initiator Ready.** PCI_IRDY_n is used in conjunction with PCI_TRDY_n. The bus master (initiator) asserts PCI_IRDY_n to indicate when there is valid data on PCI_AD(31:0) during a write, or that it is ready to accept data on PCI_AD(31:0) during a read. A data phase is completed when both PCI_IRDY_n and PCI_TRDY_n are asserted. During a write, a low assertion of PCI_IRDY_n indicates that valid data is present on PCI_AD(31:0). During a read, a low assertion of PCI_IRDY_n indicates the master is prepared to accept data. Wait cycles are inserted until both PCI_IRDY_n and PCI_TRDY_n are asserted together. Connect to 3.3 V if the PCI Controller is unused.
PCI_PAR	I/O	**PCI Parity.** Parity is driven high or low to create even parity across PCI_AD(31:0) and PCI_C_BE_n(3:0). The master drives PCI_PAR for address and write data phases; the target drives PCI_PAR for read data phases. Connect to 3.3 V if the PCI Controller is unused.
PCI_PERR_n	I/O	**PCI Parity Error.** PCI_PERR_n indicates the occurrence of a data parity error during all PCI transactions except a Special Cycle. The QuickMIPS chip drives PCI_PERR_n low two clocks following the data when a data parity error is detected. The minimum duration of the deassertion of PCI_PERR_n is one clock for each data phase that a data parity error is detected. (If sequential data phases each have a data parity error, the PCI_PERR_n signal is asserted for more than a single clock.) PCI_PERR_n is driven high for one clock before being 3-stated as with all sustained 3-state signals. Connect to 3.3 V if the PCI Controller is unused.

Table 50: Pin Descriptions (Continued)

Pin	I/O	Function
PCI_REQ_n	O	**PCI Request.** Assertion of PCI_REQ_n indicates to the arbiter that this agent desires use of the bus. PCI_REQ_n is 3-stated while PCI_RST_n is asserted. Leave unconnected if the PCI Controller is unused.
PCI_RST_n	I	**PCI Reset.** Asserting PCI_RST_n low resets the internal state of the QuickMIPS PCI block. When PCI_RST_n is asserted, all PCI output signals are asynchronously 3-stated. PCI_REQ_n and PCI_GNT_n must both be 3-stated (they cannot be driven low or high during reset). The assertion/deassertion of PCI_RST_n can be asynchronous to PCI_CLK. Connect to GND if the PCI Controller is unused.
PCI_SERR_n	O	**PCI System Error.** The QuickMIPS chip asserts PCI_SERR_n to indicate an address parity error, a data parity error on the Special Cycle command, or any other system error where the result is catastrophic. PCI_SERR_n is open drain and is actively driven for a single PCI clock. The assertion of PCI_SERR_n is synchronous to the clock and meets the setup and hold times of all bused signals. However, the restoring of PCI_SERR_n to the deasserted state is accomplished by a weak pull-up (same value as used for s/t/s), which is provided by the central resource not by the signaling agent. This pull-up can take two to three clock periods to fully restore PCI_SERR_n. Leave unconnected if the PCI Controller is unused.
PCI_STOP_n	I/O	**PCI Stop.** PCI_STOP_n is asserted low to indicate the current target is requesting the master to stop the current transaction. Connect to 3.3 V if the PCI Controller is unused.
PCI_TRDY_n	I/O	**PCI Target Ready.** PCI_TRDY_n is used in conjunction with PCI_IRDY_n. The current bus slave (target) asserts PCI_TRDY_n to indicate when there is valid data on PCI_AD(31:0) during a read, or that it is ready to accept data on PCI_AD(31:0) during a write. A data phase is completed when both PCI_TRDY_n and PCI_IRDY_n are asserted. During a read, a low assertion of PCI_TRDY_n indicates that valid data is present on PCI_AD(31:0). During a write, a low assertion indicates the target is prepared to accept data. Wait cycles are inserted until both PCI_IRDY_n and PCI_TRDY_n are asserted together. Connect to 3.3 V if the PCI Controller is unused.
PCI_CLK	I	**PCI Clock.** All PCI signals (except PCI_RST_n and PCI_INTA_n) are sampled on the rising edge of PCI_CLK. PCI_CLK operates at speeds up to either 33 MHz or 66 MHz. Connect to GND if the PCI Controller is unused.
Ethernet MAC Signals		
M1_RXCLK, M2_RXCLK	I	**Ethernet Receive Clock.** RXCLK is a continuous clock that provides the timing reference for the transfer of the RXDV and RXD(3:0) signals from the Ethernet PHY Controller to the MAC core. The Ethernet PHY Controller chip sources RXCLK. RXCLK has a frequency equal to 25% of the data rate of the received signal on the Ethernet cable. Connect to GND if the Ethernet Controller is unused.

Table 50: Pin Descriptions (Continued)

Pin	I/O	Function
M1_TXCLK, M2_TXCLK	I	**Ethernet Transmit Clock.** TXCLK is a continuous clock that provides a timing reference for the transfer of the TXEN and TXD signals from the MAC core to the Ethernet PHY Controller. The Ethernet PHY Controller chip sources TXCLK. The operating frequency of TXCLK is 25 MHz when operating at 100 Mbps and 2.5 MHz when operating at 10 Mbps. Connect to GND if the Ethernet Controller is unused.
Memory Controller Interface Signals		
BLS_n(3:0)	O	**SRAM Byte Enables.** BLS_n(3:0) indicates the validity of the bytes on DATA(31:0) for external SRAM read and write accesses. BLS_n(3) corresponds to DATA(31:24) BLS_n(2) corresponds to DATA(23:16) BLS_n(1) corresponds to DATA(15:8) BLS_n(0) corresponds to DATA(7:0)
CS_n(7:0)	O	**Chip Selects.** These signals are the active-low chip selects for the SRAM.
ADDR(23:0)	O	**Memory Address.** This 24-bit bus contains the memory address for external SRAM and SDRAM accesses.
DATA(31:0)	I/O	**Memory Data.** This 32-bit bus contains the memory read/write data for SRAM and SDRAM accesses.
OE_n	O	**SRAM Output Enable.** OE_n is the active-low output enable to the external SRAM.
SD_CAS_n	O	**SDRAM Column Address Strobe.** SD_CAS_n is the active-low column address strobe for the external SDRAM.
SD_CKE(1:0)	O	**SDRAM Clock Enables.** If low, these signals indicate to the externally connected SDRAM to enter the power-down state.
SD_CLKIN	I	**SDRAM Input Clock.** SD_CLKIN should be tied to SD_CLKOUT on the PCB. Internal to the QuickMIPS device, all SDRAM command, address and data signals are synchronized with SD_CLKIN. If a clock buffer is used to drive the SDRAM devices, this buffer should be a zero-delay type buffer, and SD_CLKIN should be tied to one of the buffer outputs.
SD_CLKOUT	O	**SDRAM Output Clock.** SD_CLKOUT is the clock source for the externally connected SDRAMs. This signal may be connected to a zero-delay buffer to drive multiple SDRAM devices. SD_CLKOUT is equal in frequency to the internal System Bus clock and hclk. See **Table 2** for additional details.
SD_CS_n(1:0)	O	**SDRAM Output Chip Select.** SD_CS_n(1:0) are the active-low chip selects for the external SDRAMs.
SD_DQM(3:0)	O	**SDRAM Data Mask.** SD_DQM(3:0) are the data masks for DATA(31:0) during SDRAM read and write accesses. SD_DQM(3) corresponds to DATA(31:24) SD_DQM(2) corresponds to DATA(23:16) SD_DQM(1) corresponds to DATA(15:8) SD_DQM(0) corresponds to DATA(7:0)
SD_RAS_n	O	**SDRAM Row Address Strobe.** SD_RAS_n is the active-low row address strobe for the external SDRAM.
SD_WE_n	O	**SDRAM Write Enable.** SD_WE_n is the active-low write enable to the SDRAMs.
WE_n	O	**SRAM Write Enable.** WE_n indicates whether transactions between the QuickMIPS chip and the external SRAM are reads (WE_n is high) or writes (WE_n is low).

Table 50: Pin Descriptions (Continued)

Pin	I/O	Function
UART Interface Signals		
U1_CTS_n	I	**UART1 Clear To Send.** A low on this signal indicates the external device is ready to transfer data. Connect to GND if the UART is unused.
U1_DCD_n	I	**UART1 Data Carrier Detect.** A low on this signal indicates the data carrier has been detected. Connect to GND if the UART is unused.
U1_DSR_n	I	**UART1 Data Set Ready.** A low on this signal indicates the modem or data set is ready to establish the link to the QuickMIPS UART. Connect to GND if the UART is unused.
U1_DTR_n	O	**UART1 Data Terminal Ready.** The QuickMIPS chip asserts this output low to indicate it is ready to establish the external communication link. Leave unconnected if the UART is unused.
U1_RI_n	I	**UART1 Ring Indicator.** This input is an active-low ring indicator. Connect to 3.3 V if the UART is unused.
U1_RTS_n	O	**UART1 Request to Send.** The QuickMIPS chip asserts this signal low to inform the external device that the UART is ready to send data. Leave unconnected if the UART is unused.
U1_RXD_SIRIN	I	**UART1 Receive Serial Data/SIR Receive Serial Data.** This input receives serial data from either the UART or the IrDA block. Connect to GND if the UART is unused.
U1_TXD_SIROUT	O	**UART1 Transmit Serial Data/SIR Transmit Serial Data.** This output transmits serial data to either the UART or IrDA block. Leave unconnected if the UART is unused.
U2_RXD_SIRIN	I	**UART2 Receive Serial Data/SIR Receive Serial Data.** This input receives serial data from either the UART or the IrDA block. Connect to GND if the UART is unused.
U2_TXD_SIROUT	O	**UART2 Transmit Serial Data/SIR Transmit Serial Data.** This output transmits serial data to either the UART or IrDA block. Leave unconnected if the UART is unused.
Test Interface Signals		
EJTAG_TCK	I	**EJTAG Test Clock.** This clock controls the updates to the TAP Controller and the shifts through the Instruction register or selected data registers. Both the rising and falling edges of EJTAG_TCK are used. Connect to 3.3 V through a 1 KΩ resistor.
EJTAG_TDI	I	**EJTAG Test Data In.** Serial test data is input on this pin and is shifted into the Instruction or data register. This input is sampled on the rising edge of EJTAG_TCK. Connect to 3.3 V through a 1 KΩ resistor.

Table 50: Pin Descriptions (Continued)

Pin	I/O	Function
EJTAG_TDO	O	**EJTAG Test Data Out.** The QuickMIPS chip outputs serial test data on this pin from the Instruction or data register. This signal changes on the falling edge of EJTAG_TCK. Connect to 3.3 V through a 10 KΩ resistor.
EJTAG_TMS	I	**EJTAG Test Mode Select.** This input is the control signal for the TAP Controller. It is sampled on the rising edge of EJTAG_TCK. Connect to 3.3 V through a 1 KΩ resistor.
EJTAG_TRST_n	I	**EJTAG Test Reset.** This signal is asserted asynchronously to reset the TAP Controller, Instruction register, and EJTAGBOOT indication. Connect to GND through a 1 KΩ resistor.
EJTAG_DEBUGM	O	**Debug Mode.** This bit is asserted high when the MIPS 4Kc core is in Debug Mode. This output can be used to bring the chip out of low power mode.
EJTAG_DINT	I	**Debug Exception Request.** Assertion high of this input indicates a debug exception request is pending. The request is cleared when debug mode is entered. Requests that occur while the chip is in debug mode are ignored. Connect to 3.3 V through a 1 KΩ resistor.
Fabric Interface Signals		
I/O(A)	I/O	**Programmable Input/Output/3-State/Bidirectional Pin in Bank A.** If an I/O is not used, SpDE (Quick*Works* Tool) provides the option of tying that pin to GND, $V_{CC,}$ or TriState during programming.
I/O(B)	I/O	**Programmable Input/Output/3-State/Bidirectional Pin in Bank B.** If an I/O is not used, SpDE (Quick*Works* Tool) provides the option of tying that pin to GND, $V_{CC,}$ or TriState during programming.
I/O(C)	I/O	**Programmable Input/Output/3-State/Bidirectional Pin in Bank C.** If an I/O is not used, SpDE (Quick*Works* Tool) provides the option of tying that pin to GND, $V_{CC,}$ or TriState during programming.
I/O(D)	I/O	**Programmable Input/Output/3-State/Bidirectional Pin in Bank D.** If an I/O is not used, SpDE (Quick*Works* Tool) provides the option of tying that pin to GND, $V_{CC,}$ or TriState during programming.
CLK(8:5) CLK(3:0)	I	**Programmable Global Clock Pin.** This pin provides access to a dedicated, distributed network capable of driving the CLOCK, SET, RESET, F1, and A2 inputs to the Logic Cell, READ, and WRITE CLOCKS, Read and Write Enables of the Embedded RAM Blocks, CLOCK of the ECUs, and Output Enables of the I/Os. Connect to 3.3V or GND if unused.
CLK(4)/ DEDCLK	I	**Low Skew Dedicated Clock.** This pin provides access to a dedicated, distributed clock network capable of driving the CLOCK inputs of all sequential elements of the device (e.g., RAM, flip-flops). Connect to 3.3V or GND if unused.
INREF(A:D)	I	**Differential I/O Reference Voltage.** INREF is the reference voltage pin for GTL+, SSTL2, and STTL3 standards. Follow the recommendations provided in **Table 23** for the appropriate standard. Connect to GND when using TTL, PCI or LVCMOS.

Table 50: Pin Descriptions (Continued)

Pin	I/O	Function
IOCTRL(A:D)	I	**High Drive I/O Control Pins.** This pin provides fast RESET, SET, CLOCK, and ENABLE access to the I/O cell flip-flops, providing fast clock-to-out and fast I/O response times. This pin can also double as a high-drive pin to the internal logic cells. There is an internal pull-down resistor to GND on this pin. This pin should be tied to GND if it is not used. If tied to 3.3 V, it will draw no more than 20 µA per IOCTRL pin due to the pull-down resistor.
Fabric JTAG Signals		
TDI/RSI	I	**Test Data In for JTAG/RAM Init. Serial Data In.** Hold HIGH during normal operation. Connects to serial PROM data in for RAM initialization. Connect to 3.3V if unused.
TRSTB/RRO	I	**Active low Reset for JTAG/RAM Init. Reset Out.** Hold LOW during normal operation. Connects to serial PROM reset for RAM initialization. Connect to GND if unused
TMS	I	**Test Mode Select for JTAG.** Hold HIGH during normal operation. Connect to 3.3V if not used for JTAG.
TCK	I	**Test Clock for JTAG.** Hold HIGH or LOW during normal operation. Connect to 3.3V or GND if not used for JTAG.
TDO/RCO	O	**Test Data Out for JTAG/RAM Init. Clock Out.** Connect to serial PROM clock for RAM initialization. Must be left unconnected if not used for JTAG or RAM initialization
Timer Interface Signals		
TM_OVERFLOW	O	**Timer Overflow.** When timer #1 is in PWM mode, it counts up to 0xFFFF and then back down to zero. This PWM output signal is asserted high when the value of the counter is less than or equal to the value programmed in timer #1's interval register. Conversely, this signal is asserted low when the counter is greater than the interval value.
TM_ENABLE	I	**Timer Enable.** This signal can be used to enable the timers internal to the QuickMIPS device. Internal timer setup registers determine how this signal is used by each timer block.
Miscellaneous Signals		
CPU_BOOT(1:0)	I	**Boot Memory Size.** These signals indicate to the QuickMIPS device the width of the boot memory (the width of the memory device connected to CS_n(7)). CPU_BOOT(1:0) = 00: 8-bit width CPU_BOOT(1:0) = 01: 16-bit width CPU_BOOT(1:0) = 10: 32-bit width CPU_BOOT(1:0) = 11: Reserved
CPU_BIGENDIAN	I	**Endian Setting.** A High on this input indicates big-endian byte ordering; a Low on this input indicates little-endian byte ordering.
CPU_EXTINT_n(6:0)	I	**CPU Interrupts.** Asserting low any of these inputs causes an interrupt to the QuickMIPS chip. These inputs are active low, level sensitive, and must be held low for at least two CPU pipeline clocks for the CPU to recognize the interrupt. CPU_EXTINT_n(6) is a Non Maskable Interrupt (NMI). Connect to 3.3 V if unused.

Table 50: Pin Descriptions (Continued)

Pin	I/O	Function
STM	I	**QuickLogic Reserved Pin.** Tie to GND on the PCB.
CPU_RESET_n	I	**Active Low CPU Reset.** Asserting this signal low resets the entire ASSP portion of the QuickMIPS device (except for the PCI Controller, which has its own reset input). When low, CPU_RESET_n causes a cold reset exception to the MIPS CPU and halts all internal system clocks. This signal should be asserted for at least five CPU_PLL_CLKIN clock cycles. For reliable operation, the power supply must be stable and the clock must be running before this signal is deasserted.
CPU_WARMRESET_n	I	**Active Low CPU Warm Reset.** Asserting this signal low resets the entire ASSP portion of the QuickMIPS device (except for the PCI Controller, which has its own reset input). When low, CPU_WARMRESET_n causes a warm reset exception to the MIPS CPU, but all system clocks continue to operate. This signal should be asserted for at least five CPU_PLL_CLKIN clock cycles. For reliable operation, the power supply must be stable and the clock must be running before this signal is deasserted.
Fabric PLL Signals		
FB_PLL_CLKIN	I	**Fabric PLL Input Clock Signal.** This is the input reference clock to the Fabric PLL circuit. Connect to GND if the Fabric PLL is not used.
FB_PLL_RESET_n	I	**Fabric PLL Active Low Reset.** If FB_PLL_RESET_n is low, then all PLL outputs are reset to zero. Connect to GND if the Fabric PLL is not used.
FB_PLL_PADOUT	O	**Fabric PLL Output Off Chip.** This is the PLL output clock driven off chip. This output runs at the PLL VCO (F_{VCO}) frequency determined by the M(1:0) input. Leave unconnected if unused.
ASSP PLL Signals		
CPU_PLL_CLKIN	I	**Input Clock Signal.** This clock input is the reference clock used by the CPU PLL. The frequency of the clock on this input is multiplied by eight to drive the MIPS CPU.
CPU_PLL_ENABLE_n	I	**Active Low PLL Enable Signal.** This signal must be low to enable the ASSP-side PLL. If CPU_PLL_ENABLE_n is held high and CPU_PLL_BYPASS is held low, the QL902M is put into a power saving quiescent state.
CPU_PLL_BYPASS	I	**PLL Bypass.** When high, the 8X multiplication of the input clock is not performed and the output clocks are equal to the input frequency.
CPU_PLL_CLKOUT	O	**Output Clock Signal from the PLL.** This output operates at the same frequency that is driven to the MIPS CPU. Leave unconnected if unused.
CPU_PLL_LOCK	O	**PLL Lock.** The lock output indicates when the PLL is locked to the input clock and is producing valid output clocks. Leave unconnected if unused.
CPU_PLL_DIV(1:0)	I	**System Bus Clock to MIPS CPU Clock Ratio Control.** See **Table 2** for more details.

Table 50: Pin Descriptions (Continued)

Pin	I/O	Function
Power and Ground Signals		
GND	I	**Ground Pin.** Tie to GND on the PCB.
VCCIO(A:D)	I	**Voltage Supply Pin for Each of the Four I/O Banks.** This pin provides the flexibility for the Fabric to interface with either a 1.8 V, 2.5 V, or 3.3 V device. Every I/O pin in the respective bank is tolerant of VCCIO(A:D) input signals and outputs VCCIO(A:D) level signals. This pin must be connected to either 1.8 V, 2.5 V, or 3.3 V.
VCC	I	**Supply Pin.** Tie to 1.95 V supply.
V3V	I	**Supply Pin.** Tie to 3.3 V supply.
GND_FB_PLL	I	**Fabric PLL Ground Pin.** Tie to analog GND on the PCB.
V3V_FB_PLL	I	**Fabric PLL 3.3 V Supply Pin.** Tie to 3.3 V analog supply.
GND_CPU_PLL	I	**CPU PLL Ground Pin.** Tie to analog GND on the PCB.
V3V_CPU_PLL	I	**CPU PLL 3.3 V Supply Pin.** Tie to 3.3 V analog supply.

ASSP Fabric Port Descriptions

Table 51: ASSP to Fabric Port Descriptions

Port	I/O[a]	Function
Ethernet Controller Signals		
M1_COL, M2_COL	I	**Ethernet Collision Detected.** The external Ethernet PHY Controller chip asserts COL high upon detection of a collision on the medium. COL remains asserted while the collision condition persists. The transitions on the COL signal are not synchronous to either the TXCLK or the RXCLK. The QuickMIPS MAC core ignores the COL signal when operating in the full-duplex mode.
M1_CRS, M2_CRS	I	**Ethernet Carrier Sense.** The external Ethernet PHY Controller chip asserts CRS high when either transmit or receive medium is non-idle. The PHY deasserts CRS low when both the transmit and receive medium are idle. The PHY must ensure that CRS remains asserted throughout the duration of a collision condition. The transitions on the CRS signal are not synchronous to either the TXCLK or the RXCLK.
M1_MDC, M2_MDC	O	**Ethernet Management Data Clock.** MDC is sourced by the MAC core to the Ethernet PHY Controller as the timing reference for transfer of information on the MDI/MDO signals. MDC is an aperiodic signal that has no maximum high or low times. The minimum high and low times for MDC are 160 ns each, and the minimum period for MDC is 400 ns, regardless of the nominal period of TXCLK and RXCLK.
M1_MDI, M2_MDI	I	**Ethernet Management Data In.** This is the data input signal from the Ethernet PHY Controller. The PHY drives the Read Data synchronously with respect to the MDC clock during the read cycles.
M1_MDO, M2_MDO	O	**Ethernet Management Data Out.** This is the data output signal from the MAC core that drives the control information during the Read/Write cycles to the External PHY Controller. The MAC core drives the MDO signal synchronously with respect to the MDC.
M1_MDO_EN_N, M2_MDO_EN_N	O	**Ethernet Management Data Output enable.** This signal, when low, enables the Fabric IO driver to drive Mn_MDO off chip.
M1_RXD(3:0), M2_RXD(3:0)	I	**Ethernet Receive Data.** RXD(3:0) transition synchronously with respect to RXCLK. The Ethernet PHY Controller chip drives RXD(3:0). For each RXCLK period in which RXDV is asserted, RXD(3:0) transfer four bits of recovered data from the PHY to the MAC core. RXD0 is the least-significant bit. While RXDV is deasserted low, RXD(3:0) has no effect on the MAC core.
M1_RXDV, M2_RXDV	I	**Ethernet Receive Data Valid.** The Ethernet PHY Controller asserts RXDV high to indicate to the MAC core that it is presenting the recovered and decoded data bits on RXD(3:0) and that the data on RXD(3:0) is synchronous to RXCLK. RXDV transitions synchronously with respect to RXCLK. RXDV remains asserted continuously from the first recovered nibble of the frame through the final recovered nibble, and is deasserted low prior to the first RXCLK that follows the final nibble.

Table 51: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
M1_RXER, M2_RXER	I	**Ethernet Receive Error.** The Ethernet PHY Controller chip asserts RXER high for one or more RXCLK periods to indicate to the MAC core that an error (a coding error or any error that the PHY is capable of detecting that is otherwise undetectable by the MAC) was detected somewhere in the frame presently being transferred from the PHY to the MAC core. RXER transitions synchronously with respect to RXCLK. While RXDV is deasserted low, RXER has no effect on the MAC core.
M1_TXD(3:0), M2_TXD(3:0)	O	**Ethernet Transmit Data.** The QuickMIPS MAC core drives TXD(3:0). TXD(3:0) transition synchronously with respect to TXCLK. For each TXCLK period in which TXEN is asserted, TXD(3:0) have the data to be accepted by the Ethernet PHY Controller chip. TXD0 is the least-significant bit. While TXEN is deasserted, ignore the data presented on TXD(3:0).
M1_TXEN, M2_TXEN	O	**Ethernet Transmit Enable.** A high assertion on TXEN indicates that the MAC core is presenting nibbles on the MII for transmission. The QuickMIPS MAC core asserts TXEN with the first nibble of the preamble and holds TXEN asserted while all nibbles to be transmitted are presented to the MII. TXEN is deasserted low prior to the first TXCLK following the final nibble of the frame. TXEN is transitions synchronously with respect to TXCLK.
On-Chip System SRAM Block 1 Signals		
SR1_ADDR(11:2)	I	**On-Chip SRAM Block 1 Address.** This 10-bit address is the memory address for block 1 of the On-Chip SRAM.
SR1_CLK	I	**On-Chip SRAM Block 1 Clock.** This clock drives block 1 of the On-Chip SRAM. When SRAM block 1 is connected to the Fabric, all SRAM block 1 IO signals are synchronous with this clock.
SR1_RDATA(31:0)	O	**On-Chip SRAM Block 1 Read Data.** This 32-bit bus is the read data from On-Chip SRAM block 1.
SR1_WDATA(31:0)	I	**On-Chip SRAM Block 1 Write Data.** This 32-bit bus is the write data to On-Chip SRAM block 1.
SR1_WEN	I	**On-Chip SRAM Block 1 Write Enable.** This is the active-low write enable to On-Chip SRAM block 1.
SR1_WENb(3:0)	I	**On-Chip SRAM Block 1 Byte Write Enable.** These control signals act as active low byte write enables to On-Chip SRAM block 1. During a write operation, if a bit of SR1_WENb is low, the corresponding data byte on the SR1_WDATA bus is written to the SRAM: SR1_WENb(3) enables SR1_WDATA(31:24) SR1_WENb(2) enables SR1_WDATA(23:16) SR1_WENb(1) enables SR1_WDATA(15:8) SR1_WENb(0) enables SR1_WDATA(7:0)
On-Chip System SRAM Block 2 Signals		
SR2_ADDR(11:2)	I	**On-Chip SRAM Block 2 Address.** This 10-bit address is the memory address for block 2 of the On-Chip SRAM.
SR2_CLK	I	**On-Chip SRAM Block 2 Clock.** This clock drives block 2 of the On-Chip SRAM. When SRAM block 2 is connected to the Fabric, all SRAM block 2 IO signals are synchronous with this clock.
SR2_RDATA(31:0)	O	**On-Chip SRAM Block 2 Read Data.** This 32-bit bus is the read data from On-Chip SRAM block 2.

Table 51: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
SR2_WDATA(31:0)	I	**On-Chip SRAM Block 2 Write Data.** This 32-bit bus is the write data to On-Chip SRAM block 2.
SR2_WEN	I	**On-Chip SRAM Block 2 Write Enable.** This is the active-low write enable to On-Chip SRAM block 2.
SR2_WENb(3:0)	I	**On-Chip SRAM Block 2 Byte Write Enable.** These control signals act as active low byte write enables to On-Chip SRAM block 2. During a write operation, if a bit of SR2_WENb is low, the corresponding data byte on the SR2_WDATA bus is written to the SRAM: SR2_WENb(3) enables SR2_WDATA(31:24) SR2_WENb(2) enables SR2_WDATA(23:16) SR2_WENb(1) enables SR2_WDATA(15:8) SR2_WENb(0) enables SR2_WDATA(7:0)
On-Chip System SRAM Block 3 Signals		
SR3_ADDR(11:2)	I	**On-Chip SRAM Block 3 Address.** This 10-bit address is the memory address for block 3 of the On-Chip SRAM.
SR3_CLK	I	**On-Chip SRAM Block 3 Clock.** This clock drives block 3 of the On-Chip SRAM. When SRAM block 3 is connected to the Fabric, all SRAM block 3 IO signals are synchronous with this clock.
SR3_RDATA(31:0)	O	**On-Chip SRAM Block 3 Read Data.** This 32-bit bus is the read data from On-Chip SRAM block 3.
SR3_WDATA(31:0)	I	**On-Chip SRAM Block 3 Write Data.** This 32-bit bus is the write data to On-Chip SRAM block 3.
SR3_WEN	I	**On-Chip SRAM Block 3 Write Enable.** This is the active-low write enable to On-Chip SRAM block 3.
SR3_WENb(3:0)	I	**On-Chip SRAM Block 3 Byte Write Enable.** These control signals act as active low byte write enables to On-Chip SRAM block 3. During a write operation, if a bit of SR3_WENb is low, the corresponding data byte on the SR3_WDATA bus is written to the SRAM: SR3_WENb(3) enables SR3_WDATA(31:24) SR3_WENb(2) enables SR3_WDATA(23:16) SR3_WENb(1) enables SR3_WDATA(15:8) SR3_WENb(0) enables SR3_WDATA(7:0)
On-Chip System SRAM Block 4 Signals		
SR4_ADDR(11:2)	I	**On-Chip SRAM Block 4 Address.** This 10-bit address is the memory address for block 4 of the On-Chip SRAM.
SR4_CLK	I	**On-Chip SRAM Block 4 Clock.** This clock drives block 4 of the On-Chip SRAM. When SRAM block 4 is connected to the Fabric, all SRAM block 4 IO signals are synchronous with this clock.
SR4_RDATA(31:0)	O	**On-Chip SRAM Block 4 Read Data.** This 32-bit bus is the read data from On-Chip SRAM block 4.
SR4_WDATA(31:0)	I	**On-Chip SRAM Block 4 Write Data.** This 32-bit bus is the write data to On-Chip SRAM block 4.
SR4_WEN	I	**On-Chip SRAM Block 4 Write Enable.** This is the active-low write enable to On-Chip SRAM block 4.

Table 51: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
SR4_WENb(3:0)	I	**On-Chip SRAM Block 4 Byte Write Enable.** These control signals act as active low byte write enables to On-Chip SRAM block 4. During a write operation, if a bit of SR4_WENb is low, the corresponding data byte on the SR4_WDATA bus is written to the SRAM: SR4_WENb(3) enables SR4_WDATA(31:24) SR4_WENb(2) enables SR4_WDATA(23:16) SR4_WENb(1) enables SR4_WDATA(15:8) SR4_WENb(0) enables SR4_WDATA(7:0)
AHB and APB Clock and Reset Signals		
hclk	O	**AMBA Bus Clock.** All AMBA bus transactions are synchronous with this clock. Upon entering the Fabric, hclk is automatically placed on a global clock net. hclk can be programmed to be 1/2, 1/3 or 1/4 the CPU clock rate. See **Table 2** for more details.
hresetn	O	**AMBA Bus Reset.** When low, this signal indicates to the programmable Fabric that the ASSP side of the device is in the reset state. This signal should be used to reset the Fabric AHB Master, AHB Slave or APB Slave interfaces.
AHB Master and AHB Slave Interface Signals		
ahb_hready_in	O	**AHB Ready Input.** This signal is used by an AHB master and /or an AHB slave implemented in the Fabric. For an AHB master implemented in the Fabric: When high, this signal indicates to the AHB master that the accessed AHB slave is ready to continue the current transfer. For an AHB slave implemented in the Fabric: An AHB slave must only sample the address and control signals and ahbs_hsel when ahb_hready_in is high, indicating that the current transfer is completing. Under certain circumstances it is possible that ahbs_hsel will be asserted when ahb_hready_in is low, but the selected slave will have changed by the time the current transfer completes.
AHB Master Interface Signals		
ahbm_haddr(31:0)	I	**AHB Master Address.** This bus contains the AHB address for the transfer initiated by the Fabric AHB master.
ahbm_hburst(2:0)	I	**AHB Master Burst Type.** These signals indicate the length of the Fabric AHB master burst transfer. Possible burst sizes are: 000: SINGLE 001: INCR (length unspecified) 010: WRAP4 011: INCR4 100: WRAP8 101: INCR8 110: WRAP16 111: INCR16
ahbm_hbusreq	I	**AHB Master Bus Request.** When high, this signal indicates to the AHB arbiter that the AHB master implemented in the Fabric is requesting ownership of the AHB.
ahbm_hgrant	O	**AHB Master Grant.** When high, this signal indicates that the AHB master implemented in the Fabric is the current AHB master.

Table 51: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
ahbm_hprot(3:0)	I	**AHB Master Protection.** Protected transfers are not supported by the QuickMIPS device. The AHB master implemented in the Fabric should tie all bits of this bus low.
ahbm_hrdata(31:0)	O	**AHB Master Read Data.** The AHB master implemented in the Fabric receives data for AHB reads on this bus. Data is received from the selected AHB slave.
ahbm_hresp(1:0)	O	**AHB Master Transfer Response.** The AHB master implemented in the Fabric receives these signals from the accessed AHB slave. For a given transfer, the slave may respond with: 00: OKAY 01: ERROR 10: RETRY 11: SPLIT (not supported in QuickMIPS)
ahbm_hsize(2:0)	I	**AHB Master Transfer Size.** The AHB master implemented in the Fabric drives these signals to indicate to the selected slave the size of the transfer taking place. Possible transfer sizes are: 000: 8 bits (byte) 001: 16 bits (halfword) 010: 32 bits (word) 011: 64 bits 100: 128 bits (4-word line) 101: 256 bits (8-word line) 110: 512 bits 111: 1024 bits
ahbm_htrans(1:0)	I	**AHB Master Transfer Type.** The AHB master implemented in the Fabric drives these signals to indicate to the selected slave the type of transfer taking place. Possible transfer types are: 00: IDLE 01: BUSY 10: NONSEQUENTIAL 11: SEQUENTIAL
ahbm_hwdata(31:0)	I	**AHB Master Write Data.** The AHB master implemented in the Fabric drives data for AHB writes on this bus. Data is received by the selected AHB slave.
ahbm_hwrite	I	**AHB Master Write.** The AHB master implemented in the Fabric drives this signal high during an AHB write operation and low during an AHB read.
AHB Slave Interface Signals		
ahbs_haddr(31:0)	O	**AHB Slave Address.** This bus contains the AHB address for the transfer intended for the AHB Fabric slave.
ahbs_hburst(2:0)	O	**AHB Slave Burst Type.** These signals indicate the length of the transfer intended for the AHB Fabric slave. Possible burst sizes are: 000: SINGLE 001: INCR (length unspecified) 010: WRAP4 011: INCR4 100: WRAP8 101: INCR8 110: WRAP16 111: INCR16
ahbs_hprot(3:0)	O	**AHB Slave Protection.** Protected transfers are not supported by the QuickMIPS device. The AHB slave implemented in the Fabric ignores these signals.

Table 51: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
ahbs_hrdata;(31:0)	I	**AHB Slave Read Data.** The AHB slave implemented in the Fabric drives data for AHB reads on this bus. Data is received by the initiating AHB master.
ahbs_hready_out	I	**AHB Slave Ready Output.** When high, this signal indicates to the initiating AHB master that the AHB slave implemented in the Fabric is ready to continue the current transfer.
ahbs_hresp(1:0)	I	**AHB Slave Transfer Response.** The initiating AHB master receives these signals from the AHB slave implemented in the Fabric. For a given transfer, the slave responds with: 00: OKAY 01: ERROR 10: RETRY 11: SPLIT (not supported in QuickMIPS)
ahbs_hsel	O	**AHB Slave Select.** When high, this signal indicates to the AHB slave implemented in the Fabric that it is the selected slave for the current AHB transfer.
ahbs_hsize(2:0)	O	**AHB Slave Transfer Size.** These signals indicate the size of the transfer intended for the AHB Fabric slave. Possible transfer sizes are: 000: 8 bits (byte) 001: 16 bits (halfword) 010: 32 bits (word) 011: 64 bits 100: 128 bits (4-word line) 101: 256 bits (8-word line) 110: 512 bits
ahbs_htrans(1:0)	O	**AHB Slave Transfer Type.** These signals indicate the type of transfer intended for the AHB Fabric slave. Possible transfer types are: 00: IDLE 01: BUSY 10: NONSEQUENTIAL 11: SEQUENTIAL
ahbs_hwdata(31:0)	O	**AHB Slave Write Data.** The initiating AHB master drives data for AHB writes on this bus. Data is intended for the AHB slave in the Fabric.
ahbs_hwrite	O	**AHB Slave Write.** During an AHB transfer, this signal is driven high during a write operation and low during a read. It is received by the AHB slave implemented in the Fabric.
APB Slave Interface Signals		
apbs_paddr(15:2)	O	**APB Slave Address.** This bus contains the APB address for the transfer intended for an APB Fabric slave.
apbs_penable	O	**APB Slave Enable.** This signal, when high, indicates the second phase (data phase) of an APB transfer intended for an APB Fabric slave.
apbs_prdata0(31:0)	I	**APB Slave 0 Read Data.** The APB slave 0 implemented in the Fabric drives data for APB reads on this bus.
apbs_prdata1(31:0)	I	**APB Slave 1 Read Data.** The APB slave 1 implemented in the Fabric drives data for APB reads on this bus.
apbs_prdata2(31:0)	I	**APB Slave 2 Read Data.** The APB slave 2 implemented in the Fabric drives data for APB reads on this bus.
apbs_psel0	O	**APB Slave 0 Select.** This signal, when high, indicates that the current transfer is intended for APB slave 0 implemented in the Fabric.

Table 51: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
apbs_psel1	O	**APB Slave 1 Select.** This signal, when high, indicates that the current transfer is intended for APB slave 1 implemented in the Fabric.
apbs_psel2	O	**APB Slave 2 Select.** This signal, when high, indicates that the current transfer is intended for APB slave 2 implemented in the Fabric.
apbs_pwdata(31:0)	O	**APB Slave Write Data.** All APB slaves implemented in the Fabric receive data for APB write transactions from this bus.
apbs_pwrite	O	**APB Slave Write.** During an APB transfer, this signal is driven high during a write operation and low during a read. It is received by all APB slaves implemented in the Fabric.
Timer/Counter Signals		
tm_extclk1	I	**Timer 1 External Clock.** This port allows a clock generated in the Fabric to drive Timer/Counter #1.
tm_extclk2	I	**Timer 2 External Clock.** This port allows a clock generated in the Fabric to drive Timer/Counter #2.
tm_extclk3	I	**Timer 3 External Clock.** This port allows a clock generated in the Fabric to drive Timer/Counter #3.
tm_extclk4	I	**Timer 4 External Clock.** This port allows a clock generated in the Fabric to drive Timer/Counter #4.
tm_fbenable	I	**Timer Enable from Fabric.** This signal, when high, indicates to the timer enable logic that the Fabric design has enabled the timer(s). Internal timer setup registers determine how this signal is used by each timer block.
tm_overflow2	O	**Timer 2 Overflow.** When timer #2 is in PWM mode, it counts up to 0xFFFF and then back down to zero. This PWM output signal is asserted high when the value of the counter is less than or equal to the value programmed in timer #2 interval register. Conversely, this signal is asserted Low when the counter is greater than the interval value.
tm_overflow3	O	**Timer 3 Overflow.** When timer #3 is in PWM mode, it counts up to 0xFFFF and then back down to zero. This PWM output signal is asserted high when the value of the counter is less than or equal to the value programmed in timer #3 interval register. Conversely, this signal is asserted Low when the counter is greater than the interval value.
tm_overflow4	O	**Timer 4 Overflow.** When timer #4 is in PWM mode, it counts up to 0xFFFF and then back down to zero. This PWM output signal is asserted high when the value of the counter is less than or equal to the value programmed in timer #4 interval register. Conversely, this signal is asserted Low when the counter is greater than the interval value.
MIPS CPU Signals		
fb_bigendian	O	**Big Endian Indicator to Fabric.** This signal, when high, indicates to the Fabric that the QuickMIPS device is in big endian mode.
fb_int	I	**Interrupt from Fabric.** This signal, when driven high by a design in the Fabric, causes an interrupt to the MIPS processor. This input is active high and level sensitive.
pm_dcachehit	O	**Performance Monitor Data Cache Hit.** This signal is asserted whenever there is a data cache hit. This signal is synchronous with CPU_PLL_CLKOUT.
pm_dcachemiss	O	**Performance Monitor Data Cache Miss.** This signal is asserted whenever there is a data-cache miss. This signal is synchronous with CPU_PLL_CLKOUT.

Table 51: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
pm_dtlbhit	O	**Performance Monitor Data TLB Hit.** This signal is asserted whenever there is a hit in the data TLB. This signal is synchronous with CPU_PLL_CLKOUT.
pm_dtlbmiss	O	**Performance Monitor data TLB Miss.** This signal is asserted whenever there is a miss in the data TLB. This signal is synchronous with CPU_PLL_CLKOUT.
pm_icachehit	O	**Performance Monitor Instruction Cache Hit.** This signal is asserted whenever there is an instruction-cache hit. This signal is synchronous with CPU_PLL_CLKOUT.
pm_icachemiss	O	**Performance Monitor Instruction Cache Miss.** This signal is asserted whenever there is an instruction-cache miss. This signal is synchronous with CPU_PLL_CLKOUT.
pm_instncomplete	O	**Performance Monitor Instruction Complete.** This signal is asserted each time an instruction completes in the pipeline. This signal is synchronous with CPU_PLL_CLKOUT.
pm_itlbhit	O	**Performance Monitor Instruction TLB Hit.** This signal is asserted whenever there is an instruction TLB hit. This signal is synchronous with CPU_PLL_CLKOUT.
pm_itlbmiss	O	**Performance Monitor Instruction TLB Miss.** This signal is asserted whenever there is an instruction TLB miss. This signal is synchronous with CPU_PLL_CLKOUT.
pm_jtlbhit	O	**Performance Monitor Joint TLB Hit.** This signal is asserted whenever there is a joint TLB hit. This signal is synchronous with CPU_PLL_CLKOUT.
pm_jtlbmiss	O	**Performance Monitor Joint TLB Miss.** This signal is asserted whenever there is a joint TLB miss. This signal is synchronous with CPU_PLL_CLKOUT.
pm_wtbmerge	O	**Performance Monitor Write-Through Merge.** This signal is asserted whenever there is a successful merge in the write-through buffer. This signal is synchronous with CPU_PLL_CLKOUT.
pm_wtbnomerge	O	**Performance Monitor Write-Through Non-Merge.** This signal is asserted whenever a non-merging store is written to the write-through buffer. This signal is synchronous with CPU_PLL_CLKOUT.
si_rp	O	**Reduce Power Indicator to Fabric.** This signal represents the state of the RP bit (27) in the MIPS CP0 Status register. Software can write this bit to indicate that the device can enter a reduced power mode. This signal is synchronous with CPU_PLL_CLKOUT.
si_sleep	O	**Sleep Indicator to Fabric.** This signal is asserted by the MIPS core whenever the WAIT instruction is executed. The assertion of this signal indicates that the clock has stopped and that the core is waiting for an interrupt. This signal is synchronous with CPU_PLL_CLKOUT.
CPU PLL Status		
io_mips_pll_lock	O	**PLL Lock Indicator.** This signal, when high, indicates to the Fabric that the ASSP PLL has achieved phase lock on the incoming reference clock signal.

Table 51: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
PCI Configuration Settings		
AF_PCI_CFGDONE	I	**PCI Configuration Done.** This signal represents the initial value (after reset) of the Config Done bit in the PCI DMA registers. After reset, the value of this register may be overwritten through the AHB. The purpose for this register is to disable the PCI interface until the MIPS processor is ready. This may be useful when the read-only ID registers in the PCI configuration space will be over-written by the MIPS processor, which will require some time. While this register is 0, retries will be signaled on the PCI bus, thus signaling that the QuickMIPS device is not ready, and the PCI transaction should be tried again at a later time. Note that the PCI Specification limits the length of time that a device can retry a transaction, and states the amount of time after the PCI reset is deasserted when a PCI configuration cycle may occur. In an embedded system, however, a designer may choose to violate certain PCI specifications if he/she knows it will not have a detrimental impact on the system. This signal is tie-low or tie-high in the Fabric only.
AF_PCI_CLASSCODE(23:0)	I	**PCI Class Code.** These signals represent the initial value (after reset) of the Class Code bits in the PCI Configuration Registers. After reset, the value of this register may be overwritten through the AHB. The Class Code register is used to identify the generic function of the device and, in some cases, a specific register-level programming interface. These signals are tie-low or tie-high in the Fabric only.
AF_PCI_DEVID(15:0)	I	**PCI Device ID.** These signals represent the initial value (after reset) of the Device ID bits in the PCI Configuration Registers. After reset, the value of this register may be overwritten through the AHB. This field identifies the particular PCI device. This identifier is allocated by the vendor. These signals are tie-low or tie-high in the Fabric only.
AF_PCI_HOST	I	**PCI Host.** This signal represents the initial value (after reset) of the Host Mode bit in the PCI DMA registers. After reset, the value of this register may be overwritten through the AHB. This register controls whether the QuickMIPS device acts as the PCI system host or is a satellite device. A system host must configure itself as well as all the devices on the PCI bus, whereas a satellite device will be configured by another device (the host) of the PCI system. Note that while in host mode, the PCI configuration registers may only be accessed by the AHB, but while in satellite (non-host) mode, the PCI configuration registers may only be accessed by the PCI bus. This signal is tie-low or tie-high in the Fabric only.
AF_PCI_MAXLAT(7:0)	I	**PCI Maximum Latency.** These signals represent the initial value (after reset) of the Max Latency bits in the PCI Configuration Registers. After reset, the value of this register may be overwritten through the AHB. This register is used for specifying how often the device needs to gain access to PCI bus. These signals are tie-low or tie-high in the Fabric only.
AF_PCI_MINGNT(7:0)	I	**PCI Minimum Grant.** These signals represent the initial value (after reset) of the Min Grant bits in the PCI Configuration Registers. After reset, the value of this register may be overwritten through the AHB. This register is used for specifying how long a burst period the device needs assuming a clock rate of 33 MHz.These signals are tie-low or tie-high in the Fabric only.
AF_PCI_REVID(7:0)	I	**PCI Revision ID.** These signals represent the initial value (after reset) of the Revision ID bits in the PCI Configuration Registers. After reset, the value of this register may be overwritten through the AHB. This register specifies a device specific revision identifier. The value is chosen by the vendor (zero is an acceptable value). This field should be viewed as a vendor defined extension to the Device ID. These signals are tie-low or tie-high in the Fabric only.

Table 51: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
AF_PCI_SUBSYSID(15:0)	I	**PCI Subsystem ID.** These signals represent the initial value (after reset) of the Subsystem ID bits in the PCI Configuration Registers. After reset, the value of this register may be overwritten through the AHB. These registers are used to uniquely identify the expansion board or subsystem where the PCI device resides. They provide a mechanism for expansion board vendors to distinguish their boards from one another even though the boards may have the same PCI controller on them. These signals are tie-low or tie-high in the Fabric only.
AF_PCI_SUBSYSVID(15:0)	I	**PCI Subsystem Vendor ID.** These signals represent the initial value (after reset) of the Subsystem Vendor ID bits in the PCI Configuration Registers. After reset, the value of this register may be overwritten through the AHB. These registers are used to uniquely identify the expansion board or subsystem where the PCI device resides. They provide a mechanism for expansion board vendors to distinguish their boards from one another even though the boards may have the same PCI controller on them. These signals are tie-low or tie-high in the Fabric only.
AF_PCI_VENID(15:0)	I	**PCI Vendor ID.** These signals represent the initial value (after reset) of the Vendor ID bits in the PCI Configuration Registers. After reset, the value of this register may be overwritten through the AHB. This field identifies the manufacturer of the device. Valid vendor identifiers are allocated by the PCI SIG to ensure uniqueness. These signals are tie-low or tie-high in the Fabric only.

a. Interface direction is specified with respect to the ASSP portion of the device. I designates an input to the ASSP and O designates an output from the ASSP.

544 BGA Pinout Table

Table 52: 544 BGA Pinout Table

Ball	Function	Ball	Function	Ball	Function	Ball	Function	Ball	Function
A1	GND	C1	I/O(D)	E1	I/O(D)	G1	I/O(D)	K23	I/O(A)
A2	I/O(D)	C2	VCCIO(D)	E2	I/O(D)	G2	I/O(D)	K24	I/O(A)
A3	I/O(D)	C3	I/O(D)	E3	I/O(D)	G3	I/O(D)	K25	GND
A4	I/O(C)	C4	I/O(D)	E4	IOCTRL(D)	G4	I/O(C)	K26	GND
A5	I/O(C)	C5	VCCIO(C)	E5	I/O(D)	G5	I/O(C)	L1	TCK
A6	I/O(C)	C6	I/O(C)	E6	I/O(C)	G6	GND	L2	CLK(0)
A7	INREF(C)	C7	I/O(C)	E7	I/O(C)	G21	VCC	L3	I/O(D)
A8	I/O(C)	C8	IOCTRL(C)	E8	I/O(C)	G22	I/O(B)	L4	I/O(D)
A9	VCCIO(C)	C9	IOCTRL(C)	E9	I/O(C)	G23	I/O(B)	L5	I/O(D)
A10	I/O(C)	C10	I/O(C)	E10	I/O(C)	G24	I/O(B)	L6	VCCIO(D)
A11	I/O(C)	C11	I/O(C)	E11	I/O(C)	G25	I/O(A)	L10	GND
A12	VCCIO(C)	C12	I/O(C)	E12	I/O(C)	G26	I/O(A)	L11	GND
A13	CLK(8)	C13	I/O(B)	E13	I/O(C)	H1	I/O(D)	L12	GND
A14	CLK(5)	C14	CLK(7)	E14	I/O(B)	H2	I/O(D)	L13	GND
A15	I/O(B)	C15	I/O(B)	E15	I/O(B)	H3	V3V	L14	GND
A16	I/O(B)	C16	I/O(B)	E16	I/O(B)	H4	GND	L15	GND
A17	I/O(B)	C17	INREF(B)	E17	I/O(B)	H5	I/O(C)	L16	GND
A18	I/O(B)	C18	IOCTRL(B)	E18	I/O(B)	H6	VCC	L17	GND
A19	IOCTRL(B)	C19	I/O(B)	E19	I/O(B)	H21	GND	L21	VCCIO(A)
A20	I/O(B)	C20	I/O(B)	E20	I/O(B)	H22	V3V	L22	I/O(A)
A21	I/O(B)	C21	I/O(B)	E21	I/O(B)	H23	I/O(B)	L23	I/O(A)
A22	I/O(A)	C22	GND	E22	I/O(B)	H24	VCCIO(A)	L24	I/O(A)
A23	I/O(A)	C23	I/O(A)	E23	I/O(B)	H25	I/O(A)	L25	V3V
A24	VCCIO(B)	C24	I/O(A)	E24	I/O(A)	H26	DEDCLK/CLK(4)	L26	GND
A25	IOCTRL(A)	C25	I/O(A)	E25	I/O(A)	J1	I/O(D)	M1	GND
A26	GND	C26	I/O(A)	E26	I/O(A)	J2	GND	M2	TDI
B1	I/O(D)	D1	I/O(D)	F1	I/O(D)	J3	VCCIO(D)	M3	I/O(D)
B2	I/O(C)	D2	I/O(D)	F2	I/O(D)	J4	I/O(C)	M4	I/O(D)
B3	I/O(C)	D3	GND	F3	I/O(D)	J5	I/O(C)	M5	I/O(D)
B4	I/O(C)	D4	IOCTRL(D)	F4	INREF(D)	J6	VCCIO(D)	M6	VCCIO(D)
B5	I/O(C)	D5	I/O(D)	F5	I/O(D))	J21	VCCIO(A)	M10	GND
B6	I/O(C)	D6	I/O(C)	F6	I/O(C)	J22	I/O(A)	M11	GND
B7	GND	D7	I/O(C)	F7	VCC	J23	I/O(B)	M12	GND
B8	I/O(C)	D8	I/O(C)	F8	VCC	J24	I/O(B)	M13	GND
B9	I/O(C)	D9	I/O(C)	F9	VCCIO(C)	J25	CLK(3)	M14	GND
B10	I/O(C)	D10	I/O(C)	F10	VCCIO(C)	J26	CLK(2)	M15	GND
B11	I/O(C)	D11	I/O(C)	F11	VCCIO(C)	K1	I/O(D)	M16	GND
B12	TMS	D12	I/O(C)	F12	GND	K2	I/O(D)	M17	GND
B13	GND	D13	IO(C)	F13	VCC	K3	I/O(D)	M21	GND
B14	CLK(6)	D14	I/O(B)	F14	VCC	K4	I/O(D)	M22	I/O(A)
B15	I/O(B)	D15	I/O(B)	F15	GND	K5	I/O(D)	M23	I/O(A)
B16	I/O(B)	D16	I/O(B)	F16	VCCIO(B)	K6	VCCIO(D)	M24	I/O(A)
B17	VCCIO(B)	D17	I/O(B)	F17	VCCIO(B)	K10	GND	M25	TRSTB
B18	GND	D18	I/O(B)	F18	VCCIO(B)	K11	GND	M26	NC
B19	I/O(B)	D19	I/O(B)	F19	VCC	K12	GND	N1	STM
B20	I/O(B)	D20	I/O(B)	F20	GND	K13	GND	N2	TDO
B21	VCCIO(B)	D21	I/O(B)	F21	VCC	K14	GND	N3	V3V
B22	I/O(A)	D22	I/O(B)	F22	I/O(B)	K15	GND	N4	CLK(1)
B23	I/O(A)	D23	I/O(A)	F23	I/O(A)	K16	GND	N5	V3V
B24	INREF(A)	D24	GND	F24	I/O(A)	K17	GND	N6	GND
B25	IOCTRL(A)	D25	VCCIO(A)	F25	I/O(A)	K21	VCCIO(A)	N10	GND
B26	I/O(A)	D26	I/O(A)	F26	I/O(A)	K22	I/O(A)	N11	GND

Table 52: 544 BGA Pinout Table **(Continued)**

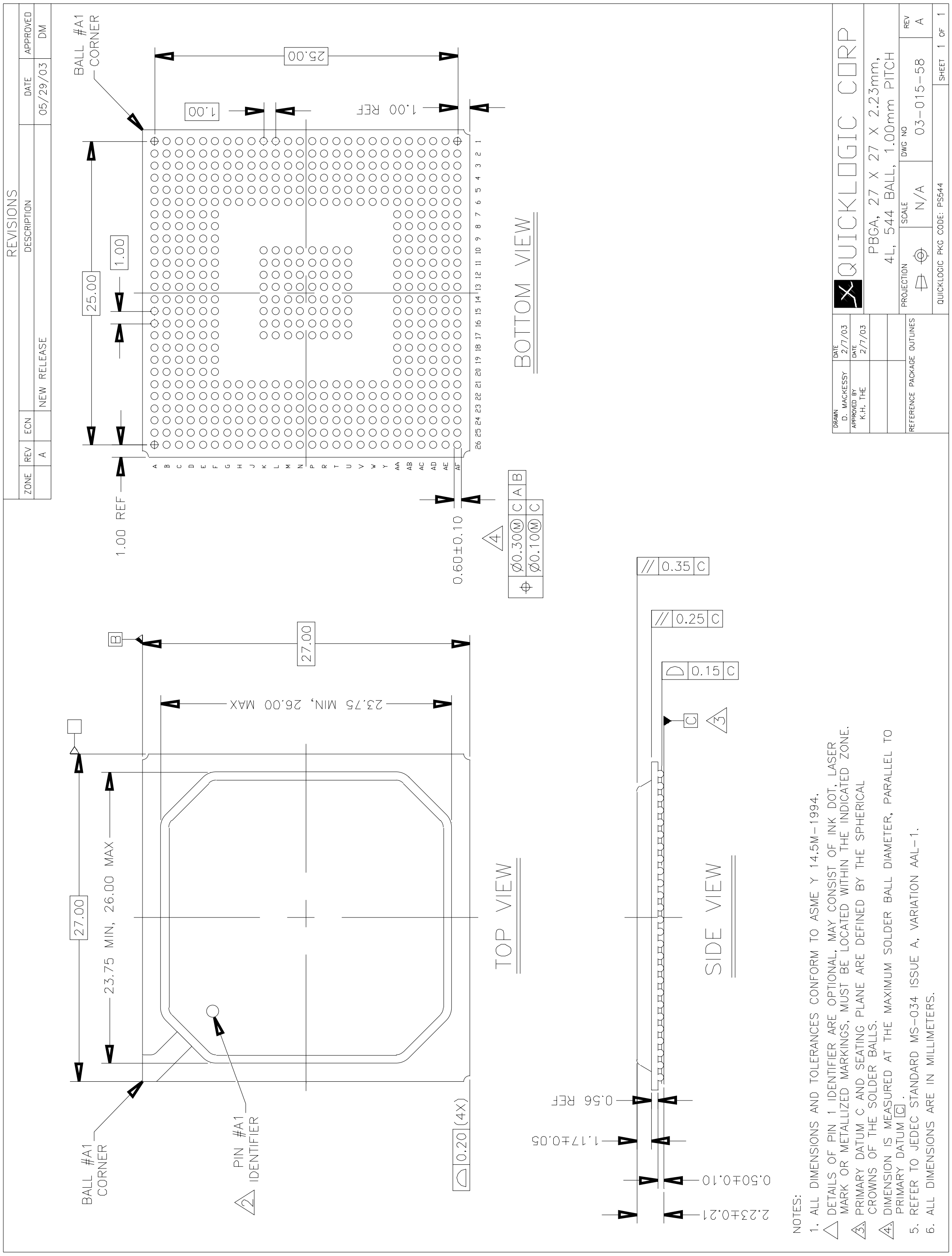
Ball	Function	Ball	Function	Ball	Function	Ball	Function	Ball	Function
N12	GND	T1	M1_TXCLK	W1	PCI_RST_n	AB3	PCI_AD(22)	AD3	PCI_AD(12)
N13	GND	T2	M1_RXCLK	W2	TM_OVERFLOW	AB4	PCI_AD(24)	AD4	PCI_AD(13)
N14	GND	T3	CPU_PLL_DIV(1)	W3	V3V	AB5	PCI_AD(18)	AD5	PCI_AD(5)
N15	GND	T4	TM_ENABLE	W4	PCI_AD(23)	AB6	PCI_C_BE_n(3)	AD6	PCI_PERR_n
N16	GND	T5	NC	W5	PCI_AD(17)	AB7	PCI_STOP_n	AD7	PCI_C_BE_n(2)
N17	GND	T6	CPU_PLL_CLKIN	W6	V3V	AB8	PCI_TRDY_n	AD8	PCI_DEVSEL_n
N21	VCC	T10	GND	W21	V3V	AB9	PCI_SERR_n	AD9	PCI_AD(15)
N22	I/O(A)	T11	GND	W22	SD_CLKIN	AB10	PCI_AD(0)	AD10	PCI_AD(4)
N23	I/O(A)	T12	GND	W23	ADDR(9)	AB11	EJTAG_TMS	AD11	PCI_AD(1)
N24	I/O(A)	T13	GND	W24	V3V	AB12	U1_DSR_n	AD12	EJTAG_TDO
N25	M2_TXCLK	T14	GND	W25	BLS_n(2)	AB13	U1_CTS_n	AD13	EJTAG_TDI
N26	M2_RXCLK	T15	GND	W26	ADDR(0)	AB14	SD_CS_n(0)	AD14	U1_RXD_SIRIN
P1	CPU_PLL_BYPASS	T16	GND	Y1	PCI_AD(30)	AB15	DATA(26)	AD15	SD_DQM(3)
P2	CPU_PLL_ENABLE_n	T17	GND	Y2	PCI_IDSEL	AB16	DATA(15)	AD16	DATA(24)
P3	VCCIO(D)	T21	GND_FB_PLL	Y3	PCI_GNT_n	AB17	DATA(4)	AD17	DATA(14)
P4	CPU_RESET_n	T22	CS_n(1)	Y4	PCI_AD(29)	AB18	ADDR(17)	AD18	ADDR(18)
P5	V3V_CPU_PLL	T23	EJTAG_DEBUGM	Y5	PCI_AD(16)	AB19	ADDR(19)	AD19	ADDR(23)
P6	VCC	T24	CS_n(4)	Y6	VCC	AB20	ADDR(20)	AD20	ADDR(14)
P10	GND	T25	GND	Y21	GND	AB21	ADDR(22)	AD21	DATA(25)
P11	GND	T26	CS_n(5)	Y22	ADDR(13)	AB22	DATA(0)	AD22	DATA(23)
P12	GND	U1	CPU_BOOT(1)	Y23	ADDR(2)	AB23	DATA(1)	AD23	DATA(16)
P13	GND	U2	GND	Y24	ADDR(3)	AB24	DATA(2)	AD24	DATA(13)
P14	GND	U3	CPU_BOOT(0)	Y25	ADDR(4)	AB25	DATA(3)	AD25	GND
P15	GND	U4	PCI_INTA_n	Y26	ADDR(6)	AB26	DATA(5)	AD26	DATA(12)
P16	GND	U5	PCI_REQ_n	AA1	PCI_AD(25)	AC1	PCI_AD(14)	AE1	PCI_AD(9)
P17	GND	U6	GND_CPU_PLL	AA2	PCI_AD(26)	AC2	V3V	AE2	PCI_AD(7)
P21	VCC	U10	GND	AA3	PCI_AD(28)	AC3	GND	AE3	PCI_AD(3)
P22	V3V_FB_PLL	U11	GND	AA4	PCI_AD(19)	AC4	PCI_C_BE_n(0)	AE4	GND
P23	V3V	U12	GND	AA5	PCI_CLK	AC5	V3V	AE5	CPU_EXTINT_n(6)
P24	VCCIO(A)	U13	GND	AA6	GND	AC6	PCI_PAR	AE6	CPU_EXTINT_n(3)
P25	NC	U14	GND	AA7	GND	AC7	PCI_IRDY_n	AE7	GND
P26	EJTAG_DINT	U15	GND	AA8	V3V	AC8	PCI_FRAME_n	AE8	EJTAG_TCK
R1	CPU_PLL_DIV(0)	U16	GND	AA9	V3V	AC9	V3V	AE9	V3V
R2	CPU_PLL_LOCK	U17	GND	AA10	V3V	AC10	PCI_AD(8)	AE10	U1_RTS_n
R3	CPU_PLL_CLKOUT	U21	V3V	AA11	GND	AC11	CPU_EXTINT_n(5)	AE11	GND
R4	NC	U22	OE_n	AA12	GND	AC12	U2_TXD_SIROUT	AE12	U1_DCD_n
R5	CPU_WARMRESET_n	U23	BLS_n(3)	AA13	VCC	AC13	U2_RXD_SIRIN	AE13	V3V
R6	VCC	U24	WE_n	AA14	V3V	AC14	SD_RAS_n	AE14	SD_CKE(1)
R10	GND	U25	CS_n(6)	AA15	V3V	AC15	DATA(30)	AE15	SD_WE_n
R11	GND	U26	CS_n(7)	AA16	GND	AC16	DATA(19)	AE16	NC
R12	GND	V1	GND	AA17	VCC	AC17	DATA(8)	AE17	V3V
R13	GND	V2	GND	AA18	V3V	AC18	ADDR(21)	AE18	GND
R14	GND	V3	PCI_AD(27)	AA19	V3V	AC19	ADDR(16)	AE19	SD_CLKOUT
R15	GND	V4	PCI_AD(31)	AA20	GND	AC20	ADDR(15)	AE20	V3V
R16	GND	V5	PCI_AD(20)	AA21	VCC	AC21	DATA(6)	AE21	DATA(31)
R17	GND	V6	GND	AA22	ADDR(12)	AC22	DATA(7)	AE22	DATA(29)
R21	FB_PLL_CLKIN	V21	V3V	AA23	ADDR(7)	AC23	DATA(9)	AE23	V3V
R22	FB_PLL_PADOUT	V22	ADDR(8)	AA24	GND	AC24	DATA(10)	AE24	DATA(20)
R23	FB_PLL_RESET_n	V23	ADDR(5)	AA25	ADDR(10)	AC25	V3V	AE25	DATA(18)
R24	CS_n(0)	V24	ADDR(1)	AA26	ADDR(11)	AC26	DATA(11)	AE26	DATA(17)
R25	CS_n(2)	V25	BLS_n(0)	AB1	PCI_C_BE_n(1)	AD1	PCI_AD(10)	AF1	GND
R26	CS_n(3)	V26	BLS_n(1)	AB2	PCI_AD(21)	AD2	PCI_AD(11)	AF2	PCI_AD(6)

Table 52: 544 BGA Pinout Table **(Continued)**

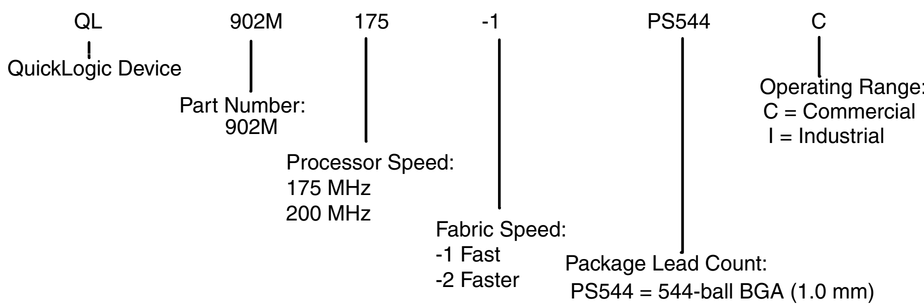
Ball	Function	Ball	Function	Ball	Function	Ball	Function	Ball	Function
AF3	PCI_AD(2)	AF8	CPU_EXTINT_n(0)	AF13	NC	AF18	SD_CS_n(1)	AF23	DATA(27)
AF4	EJTAG_TRST_n	AF9	CPU_BIGENDIAN	AF14	NC	AF19	SD_DQM(2)	AF24	DATA(22)
AF5	CPU_EXTINT_n(4)	AF10	U1_DTR_n	AF15	SD_CKE(0)	AF20	SD_DQM(1)	AF25	DATA(21)
AF6	CPU_EXTINT_n(2)	AF11	U1_RI_n	AF16	SD_CAS_n	AF21	SD_DQM(0)	AF26	GND
AF7	CPU_EXTINT_n(1)	AF12	U1_TXD_SIROUT	AF17	NC	AF22	DATA(28)		

544 BGA Pinout Drawing

Figure 57: 544 BGA Pinout Diagram



Ordering Information



Contact Information

Telephone:	(408) 990 4000 (US)
	(416) 497 8884 (Canada)
	+(44) 1932 57 9011 (Rest of Europe)
	+(49) 89 930 86 170 (Germany & Benelux)
	+(8621) 6867 0273 (Asia)
	+(81) 45 470 5525 (Japan)
E-mail:	info@quicklogic.com
Support:	http://www.quicklogic.com/support
Web site:	http://www.quicklogic.com/

Revision History

Revision	Date	Comments
A	May 2003	Judd Heape and Kathleen Murchek
B	June 2003	Judd Heape and Kathleen Murchek Modifications to all sections including pinout tables and AC timing parameters. Added section related to operation of on-chip SRAM.
C	June 2003	Judd Heape and Kathleen Murchek Modifications to all sections including pinout tables and AC timing parameters. Added Interrupt Controller section.
D	July 2003	Judd Heape and Kathleen Murchek General clean-up of document text and figures. Added ECU and Dual-Port RAM signals. Added text to ASSP I/O DC Characteristics and ASSP to Fabric Interface Output Timing tables. Added new section - Board Layout Recommendations. Replaced Advanced Clocks Network section.
E	July 2003	Judd Heape and Kathleen Murchek New PLL power supply filtering illustration and general clean-up of document text. Modified Multiple I/O Banks figure.
F	December 2003	Judd Heape and Kathleen Murchek Changed T_{jmax} number from 150°C to 125°C in Package Thermal Characteristics section. Added $I_{CC(STATIC)}$ row to DC Characteristics table. Added an AHB Master and AHB Slave Interface Signals section with an ahb_hready_in row to ASSP to Fabric Port Descriptions table. Changed V_{CC} supply specification to 1.95 ± 0.05 V.
G	March 2004	Judd Heape and Kathleen Murchek Removed Fabric RAM modes 512x4 and 1024x2. Updated AC and DC Characteristics including all Fabric timing and ASSP to Fabric interface timing. Added Fabric PLL timing data and modified clock network figures. Added System SRAM timing data and added timing diagrams. Modified Power-Up Sequencing section. Modified security and flexibility fuse descriptions in JTAG section. Modified EJTAG pin descriptions. Added Fabric I/O voltage and current graphs to DC Characteristics section.
H	September 2004	Judd Heape and Kathleen Murchek Removed I_{FAB_PLL} and I_{CPU_PLL} sections from DC Characteristics table. Changed all instances of VCC_FB_PLL and VCC_CPU_PLL to VCC. Removed VCC_FB_PLL and VCC_CPU_PLL from pin description table and PLL Power Supply Filtering Circuits figure. Removed instances of analog V_{CC}, FB_PLL_VCC and CPU_PLL_VCC from PLL Power Supply Filtering section. Updated text and tables in Fabric PLL Signals section

Copyright and Trademark Information