QL901M QuickMIPS™ Data Sheet

QuickMIPS Embedded Standard Product (ESP) Family

Device Highlights

CPU Core

- 32-bit MIPS 4Kc processor runs up to 133 MHz (173 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 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

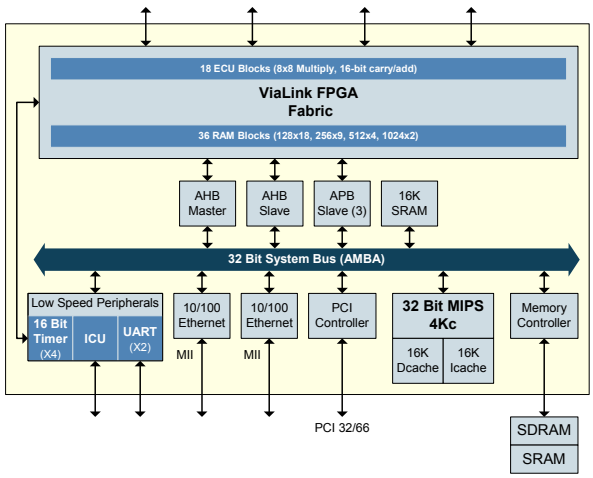
Four General Purpose 16-bit Timer/Counters

- 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

Figure 1: QL901M Block Diagram



High Performance 32-bit System Bus (AMBA Bus)

- Operates at one-half 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 (536 K system gates)
- 252 I/O pins
- 2.5 V Vcc, 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 Tco <3 ns
- Programmable Slew Rate Control
- Programmable I/O Standards:
  - LVTTL, LVCMOS, 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
- Two User-programmable Phase Locked Loop (PLL) circuits

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 QL901M is fabricated on a 0.25 µ, six layer metal CMOS process. The core voltage is 2.5 V V_{CC} supply and the I/Os are up to 3.3 V compliant. The QL901M is available in commercial and industrial temperature grades.

QL901M Architectural Overview

The QL901M 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 QL901M device.

CPU Core

The MIPS32 4Kc processor core is a high-performance, low-power, 32-bit MIPS RISC core capable of speeds up to 133 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 QL901M 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 (PL_CLOCKIN) is the input to a PLL that is fixed at a two times clock multiplication rate. For example, if the clock rate applied to PL_CLOCKIN is 50 MHz, the resultant clock that drives the MIPS core is 100 MHz. **Table 1** shows the maximum input clock rates for PL_CLOCKIN based upon the ASSP speed grade of the given QL901M device.

Table 1: Maximum Input Frequency for PL_CLOCKIN and MIPS Core Frequency Based on QL901M ASSP Speed Grade

QuickMIPS Device Part Number Prefix	Maximum Input Frequency for PL_CLOCKIN	Resultant Maximum MIPS Core Frequency
QL901M-100	50 MHz	100 MHz
QL901M-133	66 MHz	133 MHz

SDRAM Memory Controller

The QL901M 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

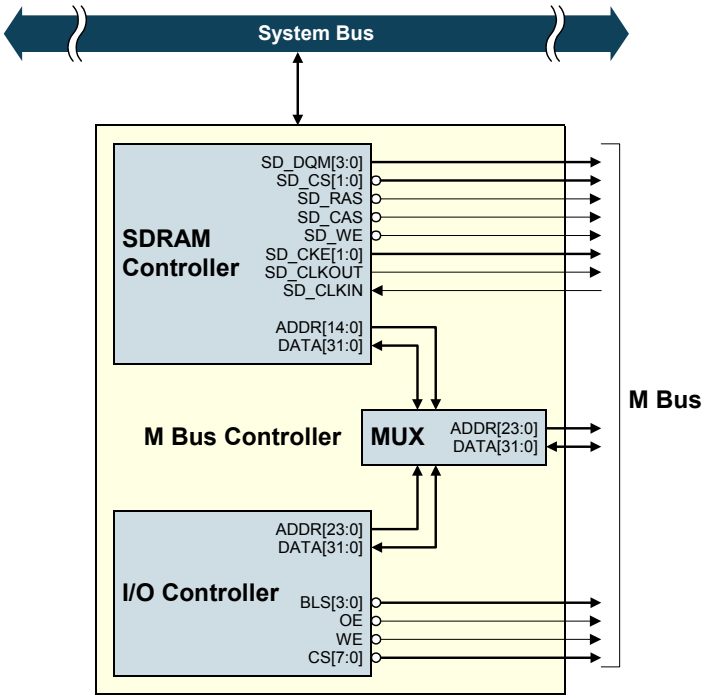
This section describes access to I/O and memory devices on the external M Bus (with the exception of SDRAM). The I/O Peripheral Controller (**Figure 2**) generates strobes and signals that can be used to interface the M Bus with common asynchronous peripheral devices.

The QL901M 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.

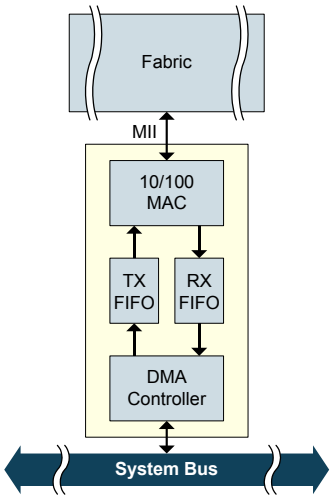
The PCI Controller is a bridge between the on-chip System Bus and the external PCI bus. There are two main modes of operation of the PCI function, each utilizing several resources:

- PCI Target
  - PCI Target to PCI configuration registers
  - PCI Target to extended registers (DMA and message/mailbox)
- PCI Master
  - System-to-PCI and PCI-to-System DMA

Ethernet Controllers

The QL901M 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). The 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 3** shows a block diagram of the QL901M Ethernet Controller.

Figure 3: 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.

EachThe 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. EachThe 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.

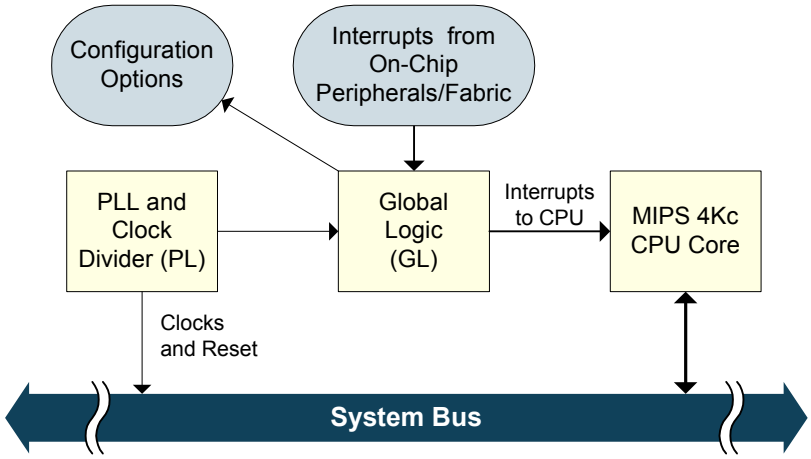
System SRAM

The QL901M contains 16 K bytes of SRAM internal to the ASSP portion of the device. 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.

Interrupt Controller

This section describes the function of the QL901M Interrupt Controller Unit (ICU). **Figure 4** shows a block diagram of the Interrupt Controller.

Figure 4: Simplified Interrupt Controller Block Diagram



The QL901M 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.

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 QL901M 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 QL901M 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 QL901M is 32 bits wide. All signals are synchronous to the rising clock edge of the bus clock (hclk).

The key features of the QL901M AHB include:

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

Table 2 lists the master and slave devices that connect to the AHB.

Table 2: 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 QL901M 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 QL901M employs fixed arbitration priority with preemption. 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 and then give up the bus after the next data transfer. 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 and will not be interrupted by the arbiter.

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 QL901M is supported through an AHB-to-APB bridge. Three separately decoded APB regions are available for APB devices implemented in the Fabric. **Table 3** lists the slave devices on the APB.

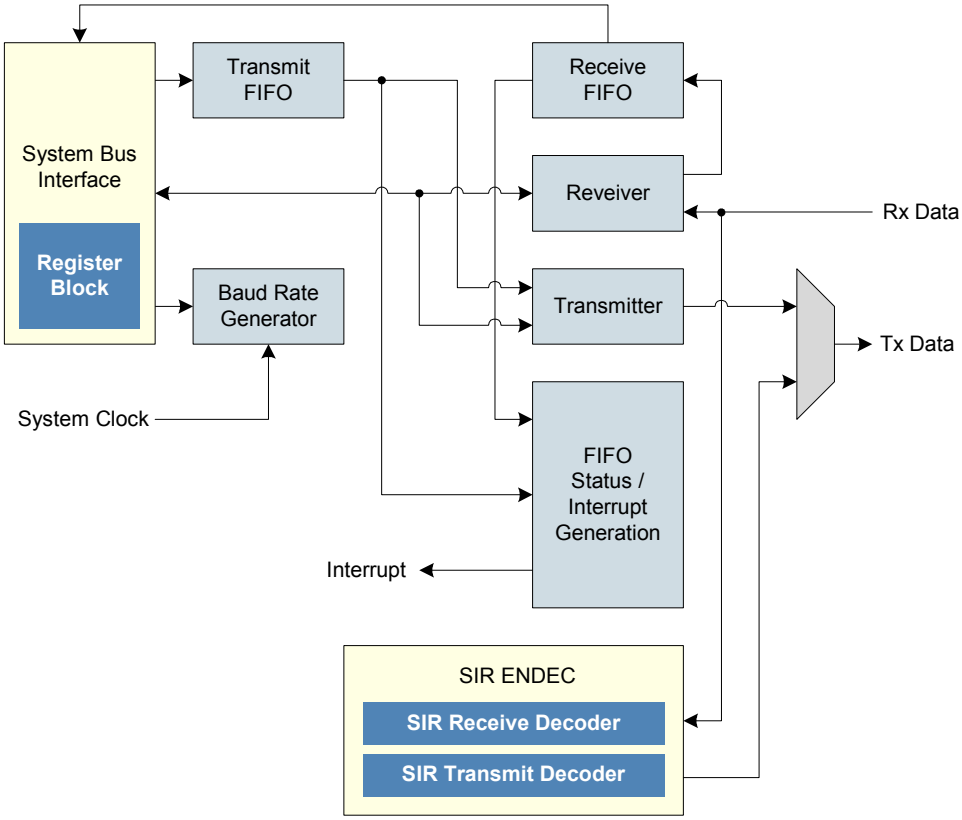
Table 3: Slave Devices on the APB

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

UARTs

The QL901M 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 5** shows a block diagram of the UART.

Figure 5: 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 QL901M Interrupt Controller.

The SIR Transmit Encoder modulates the Non-Return-to-Zero (NRZ) transmit bitstream output from the QL901M 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 QL901M 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 QL901M chip has four independent 16-bit timer/counter modules. The configuration registers for these modules are accessible through the System Bus (APB).

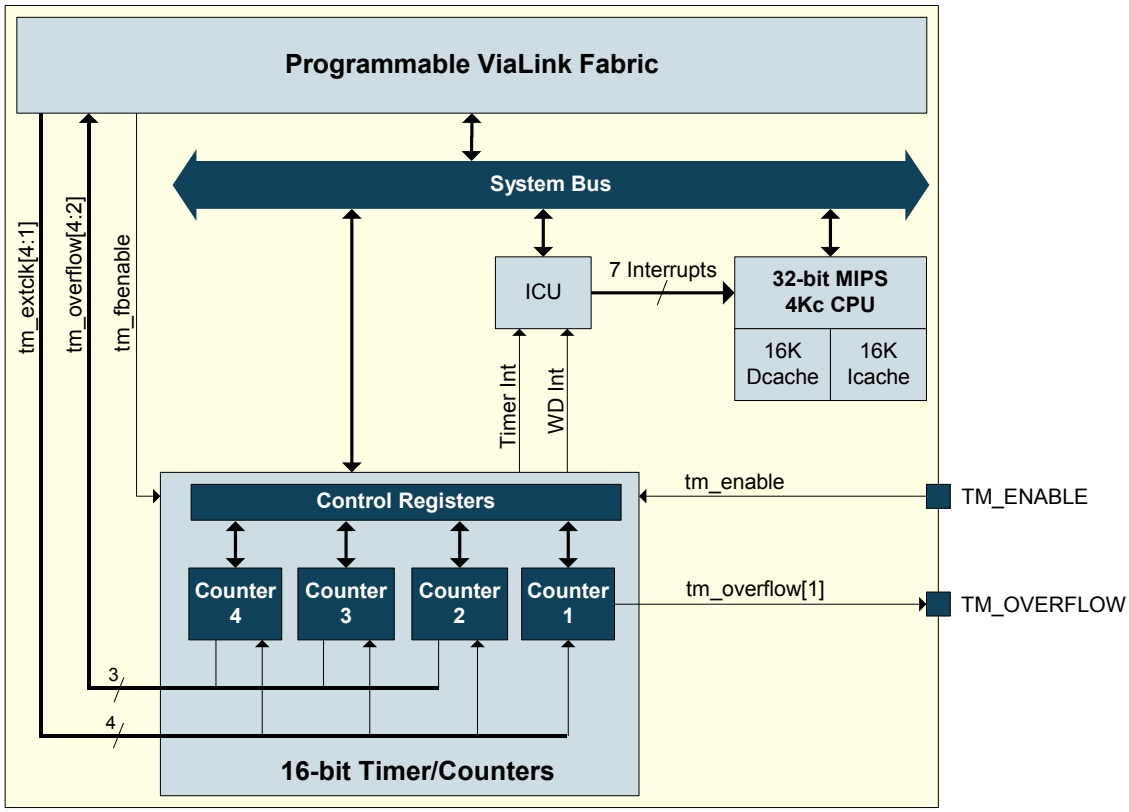
The System Bus clock (hclk), or an external clock supplied from the Fabric, drive the clock inputs on the timer/counter modules. These counters operate in one of four modes: decrement, increment, interval, or Pulse Width Modulation (PWM).

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 6 shows a functional block diagram of the timer module.

Figure 6: Timer Functional Block Diagram



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

- Up to 66 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 QL901M device.

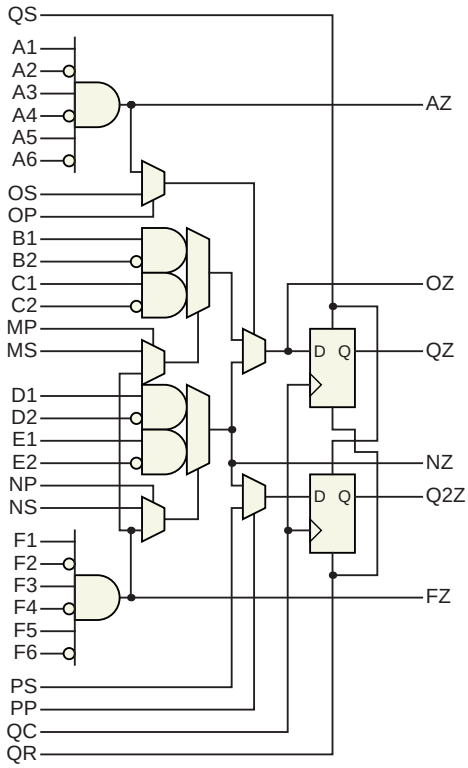
Logic Cells

The QL901M logic cell structure presented in **Figure 7** 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.

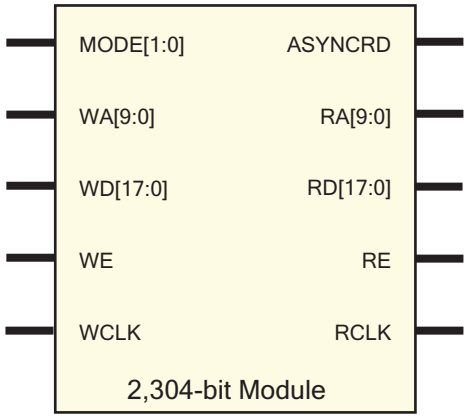
Figure 7: QL901M LogicCell



Dual-Port SRAM Modules

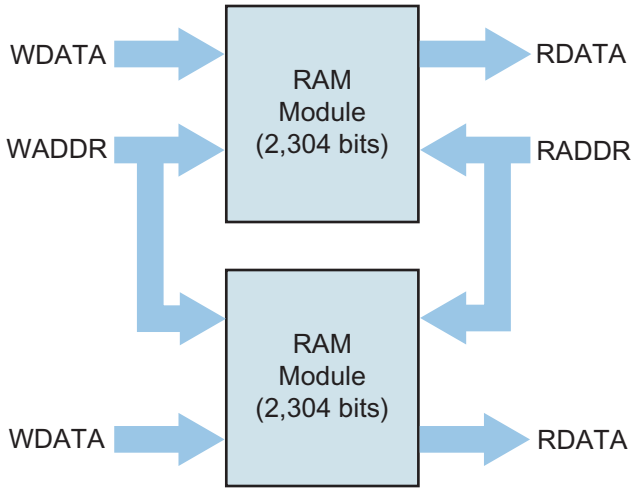
The QL901M includes 36 dual-port 2,304-bit RAM modules (shown in **Figure 8**) for implementing RAM, ROM, and FIFO functions. Each module is user-configurable into four different block organizations and can be cascaded vertically to increase their effective depth or horizontally to increase their effective width as shown in **Figure 9**.

Figure 8: 2,304-bit RAM Module



Using two *mode* pins, designers can configure each module into 128 x 18 (Mode 0), 256 x 9 (Mode1), 512 x 4 (Mode 2), or 1024 x 2 (Mode 3).

Figure 9: 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 10 address lines, allowing word lengths of up to 18 bits and address spaces of up to 1,024 words. Depending on the mode selected, however, some higher order data or address lines 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), or as a flow-through enable for asynchronous READ operation (ASYNCRD input 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 512 words. In this case address signals higher than the ninth 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 4**.

Table 4: 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(9:0)	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(9:0) is ignored when WE is low. Note that some higher order bits of WA(9:0) 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(9:0)	I	**Read Address.** This is the read address for data to be read from the RAM module. If ASYNCRD is low, RA(9:0) is sampled only on the rising edge of RCLK while RE is high. If ASYNCRD is high, RA(9:0) is continuously sampled by the RAM module and RE has no effect. Note that some higher order bits of RA(9:0) may not be used depending on the selected mode for the RAM module (see MODE(1:0) signal description).

Table 4: Dual-Port RAM Module Signal Descriptions (Continued)

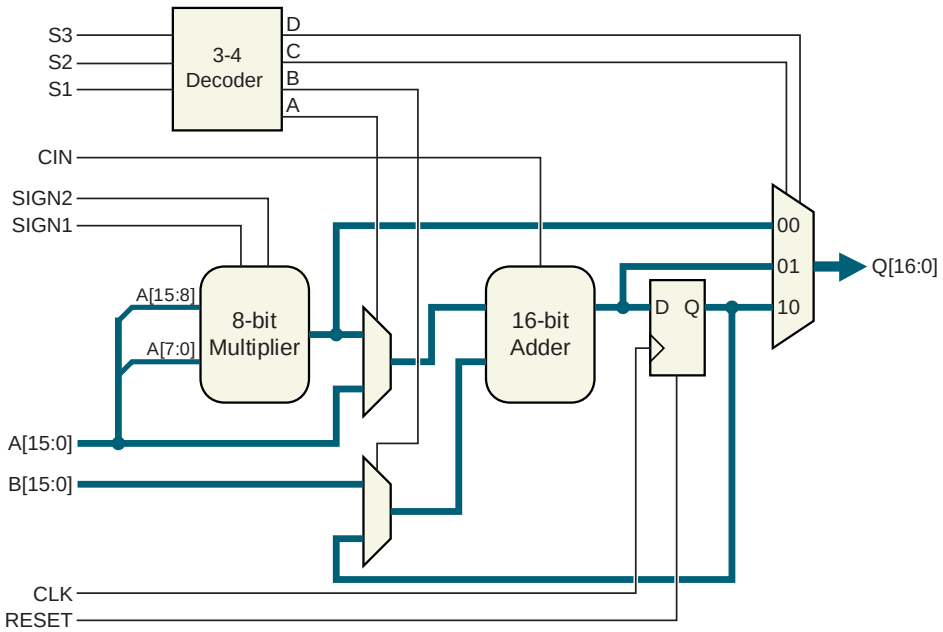
Signal Name	I/O	Description
RD(17:0)	O	**Read Data.** This is the read output data from the RAM module. If the RAM module is in synchronous read mode (ASYNCRD low), valid read data is output immediately following the rising edge of RCLK which sampled RE as high. If the RAM module is in asynchronous read 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) = "10" : 512 x 4 (locations x data bits) MODE(1:0) = "11" : 1024 x 2 (locations x data bits)

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 QL901M architecture allows for functionality above and beyond that achievable using programmable logic devices. By embedding a dynamically reconfigurable computational unit, the QL901M 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 10**.

Figure 10: ECU Block Diagram



ECU Signals

Table 5 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 5: 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 6**).
S2	I	**ECU Control S2.** One of three instruction signals that define the configuration mode of the ECU (see **Table 6**).
S3	I	**ECU Control S3.** One of three instruction signals that define the configuration mode of the ECU (see **Table 6**).
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)	I	**Augend Input.** 16-bit augend input of the 16-bit adder when the ECU is in any of the adder configuration modes.
A(15:8)		**Multiplicand Input.** 8-bit multiplicand input when the ECU is in any of the multiplier configuration modes.
A(7:0)		**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 QL901M 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 6**.

Table 6: 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)

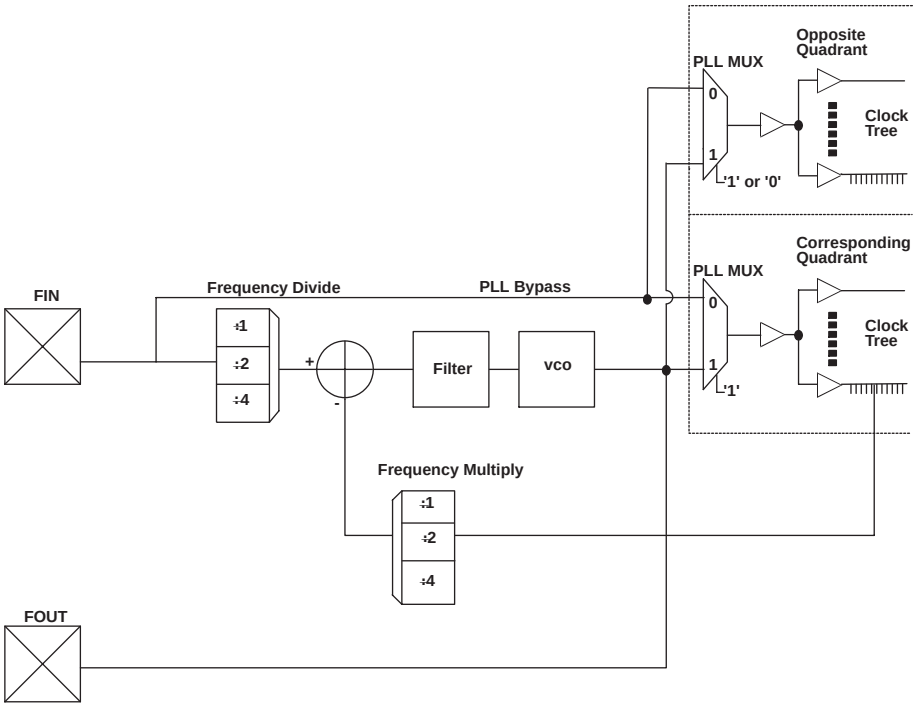
a. B (15:0) set to zero.

Fabric PLLs

Instead of requiring extra components, designers simply need to instantiate one of the pre-configured models (described in this section). The QuickLogic built-in PLLs support a wider range of frequencies than many other PLLs. Also, QuickLogic PLLs can be cascaded to support different ranges of frequency multiplications or divisions, driving the device at a faster or slower rate than the incoming clock frequency. Most importantly, they achieve a very short clock-to-out time—generally less than 3 ns. This low clock-to-out time is achieved by the PLL subtracting the clock tree delay through the feedback path, effectively making the clock tree delay zero.

The QL901M contains two fabric PLLs. PLL0 is in the upper left corner of the device (near the upper left quadrant) and PLL1 is in the upper right corner of the device (near the upper right quadrant). If a PLL is utilized, it must drive logic in its associated quadrant. The PLL can also drive the opposing quadrant as an option. The QuickWorks software handles PLL selection and placement of drive logic automatically. Designers also have the option to issue design constraints to QuickWorks, selecting specific PLLs and the location of driven logic. **Figure 11** shows a block diagram of a QL901M PLL.

Figure 11: PLL Block



F_{in} represents a very stable high-frequency input clock and produces an accurate signal reference. This signal can either bypass the PLL entirely, thus entering the clock tree directly, or it can pass through the PLL itself.

Within the PLL, a voltage-controlled oscillator (VCO) is added to the circuit. The external F_{in} signal and the local VCO form a control loop. The VCO is multiplied or divided down to the reference frequency, so that a phase detector (the crossed circle in **Figure 11**) can compare the two signals. If the phases of the external and local signals are not within the tolerance required, the phase detector sends a signal through the charge pump and loop filter (**Figure 11**). The charge pump generates an error voltage to bring the VCO back into alignment and the loop filter removes any high frequency noise before the error voltage enters the VCO. This new VCO signal enters the clock tree to drive the chip's circuitry.

F_{out} represents the clock signal that emerges from the output pad (the output signal PLLPAD_OUT is explained in **Table 8**). This clock signal is meaningful only when the PLL is configured for external use; otherwise, it remains in high Z state, as shown in the post-simulation waveform.

For more specific information on the Phase Locked Loops, please refer to Application Note 58 at http://www.quicklogic.com/images/appnote58.pdf.

Fabric PLL Modes of Operation

QuickLogic PLLs have eight modes of operation, based on the input frequency and desired output frequency—**Table 7** indicates the features of each mode.

Table 7: PLL Mode Frequencies

PLL Model	Output Frequency	Input Frequency Range[a]	Output Frequency Range
PLL_HF[b]	Same as input frequency	66 MHz–150 MHz	66 MHz–150 MHz
PLL_LF	Same as input frequency	25 MHz–133 MHz	25 MHz–133 MHz
PLL_MULT2HF	2 × input frequency	50 MHz–125 MHz	100 MHz–250 MHz
PLL_MULT2LF	2 × input frequency	16 MHz–50 MHz	32 MHz–100 MHz
PLL_DIV2HF	1/2 × input frequency	100 MHz–250 MHz	50 MHz–125 MHz
PLL_DIV2LF	1/2 × input frequency	50 MHz–100 MHz	25 MHz–50 MHz
PLL_MULT4	4 × input frequency	16 MHz–40 MHz	64 MHz–160 MHz
PLL_DIV4	1/4 × input frequency	100 MHz–300 MHz	25 MHz–75 MHz

a. The input frequency can range from 12.5 MHz to 500 MHz, while output frequency ranges from 25 MHz to 250 MHz. When you add PLLs to your top-level design, be sure that the PLL mode matches your desired input and output frequencies.
b. HF stands for high frequency and LF stands for low frequency.

Fabric PLL Signals

Table 8 summarizes the key signals in QuickLogic PLLs.

Table 8: PLL Signals

Signal Name	Description
PLLIN[a]	Input clock signal
PLLRST	**Active High Reset** If PLLRST is asserted, then CLKNET_OUT and PLLOUT are reset to 0. This signal must be asserted and then released in order for the LOCK_DETECT to work.
ONn_OFFCHIP	**PLL output** This signal selects whether the PLL will drive the internal clock network or be used off-chip. This is a static signal, not a dynamic signal. Tied to GND = outgoing signal drives internal gates. Tied to VCC = outgoing signal used off-chip.
CLKNET_OUT	**Out to internal gates** This signal bypasses the PLL logic before driving the internal gates. Note that this signal cannot be used in the same quadrant where the PLL signal is used (PLLCLK_OUT).
PLLCLK_OUT	**Out from PLL to internal gates** This signal can drive the internal gates after going through the PLL. For this to work, ONn_OFFCHIP must be tied to GND.
PLLOUT	**Out to off-chip** This outgoing signal is used off-chip. For this to work, ONn_OFFCHIP signal must be tied to VCC.
LOCK_DETECT	**Active High Lock detection signal** NOTE: For simulation purposes, this signal gets asserted after 10 clock cycles. However, it can take a maximum of 200 clock cycles to sync with the input clock upon release of the RESET signal.

a. Because PLLIN and PLLRST signals have INPAD, and PLLOUT has OUTPAD, you do not have to add additional pads to your design.

Advanced Clock Networks

The QL901M 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 QL901M has only two quadrants.) **Figure 12** shows a simplified view of the low-skew clock network architecture in the QL901M device.

Figure 12: Low Skew Clock Architecture

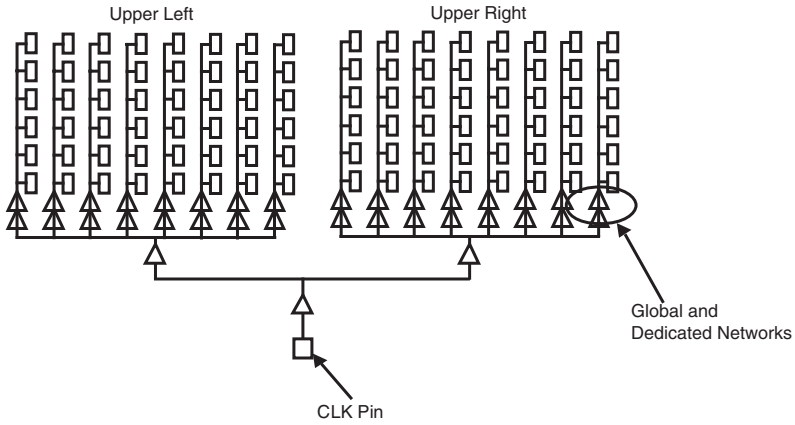


Table 9 shows the number of Global and Dedicated Networks available per quadrant in the QL901M.

Table 9: Number of QL901M 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 9** shows, there are a total of nine low-skew networks per quadrant in the QL901M, making a total of eighteen in the entire device. Each quadrant contains eight Global Networks and one Dedicated Network.

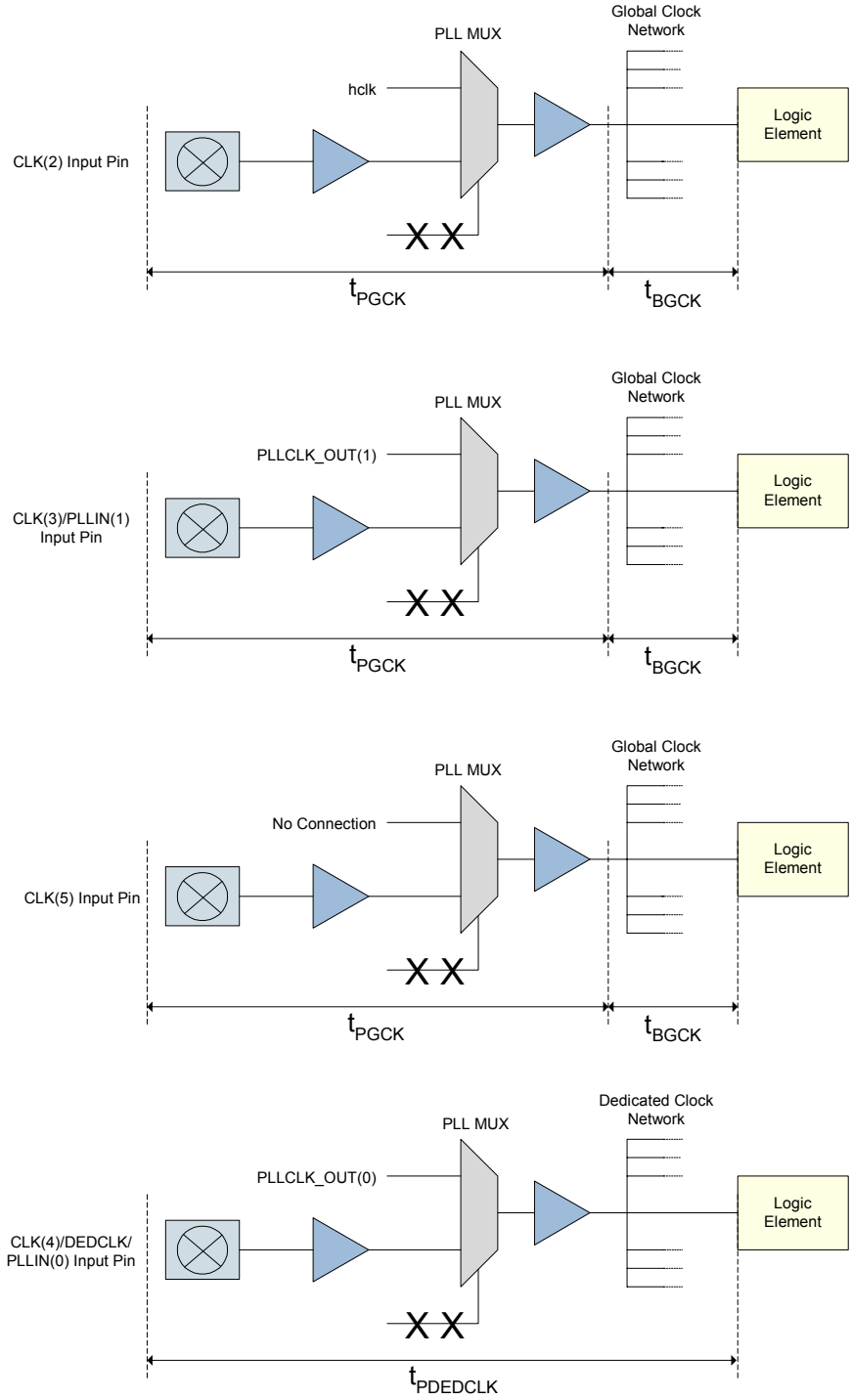
The Global Network and Dedicated Network differ slightly in performance and flexibility. The Dedicated Network offers superb low-skew and minimal pin to logic element delay performance, but can only drive the clock inputs of specific Fabric elements. The Global Network offers more flexibility to drive a variety of inputs in the Fabric as well as internal ASSP port inputs, but at a slight increase in skew and delay. **Table 10** outlines all allowable input destinations for each clock network type.

Table 10: 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 13** shows a simplified schematic diagram of the CLK input pins, the PLLMUX elements and the associated clock networks.

Figure 13: 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 the corresponding 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 11**.

Table 11: PLLMUX Input Signals and Output Type (Per Quadrant)

Input		Output Type
Pin	Internal Signals	
CLK(2)	hclk	Global clock network
CLK(3)	PLLCLK_OUT(1)	Global clock network
CLK(4)/DEDCLK	PLLCLK_OUT(0)	Dedicated clock network
CLK(5)	-	Global clock network

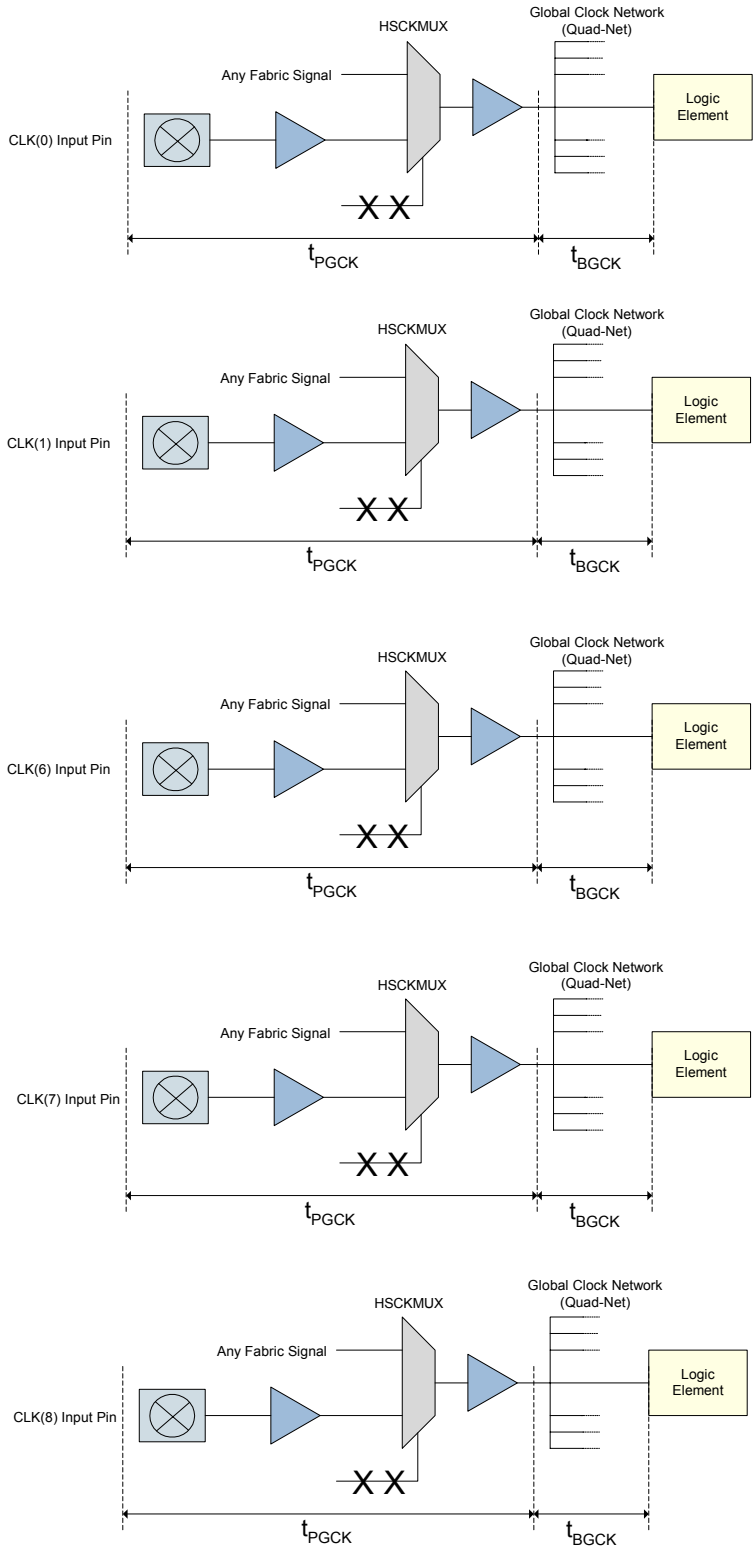
As **Table 11** indicates, once a PLLMUX is used to drive an internal signal onto the Global or Dedicated Networks, the corresponding CLK input pin is blocked from entering that quadrant.

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 14** shows a simplified schematic diagram of the CLK input pins, the HSCKMUX elements and the associated Quad-Net networks.

Figure 14: 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 12**.

Table 12: 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 12** 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 12** 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 QL901M 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 32**.

General Routing Network

QL901M 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 24. 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 QL901M 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 2.5 V and 3.3 V compliant 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 QL901M can also support the LVDS and LVPECL I/O standards with the use of external resistors (see **Table 13**).

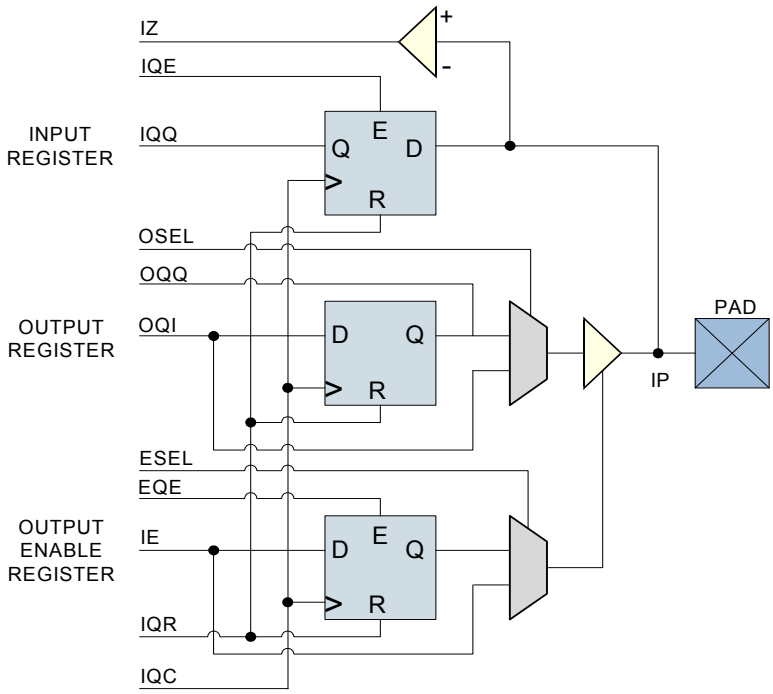
Table 13: I/O Standards and Applications

I/O Standard	Reference Voltage	Output Voltage	Application
LVTTL	n/a	3.3 V	General Purpose
LVCMOS25	n/a	2.5 V	General Purpose
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 QL901M 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 15**, 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 15: QL901M 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.

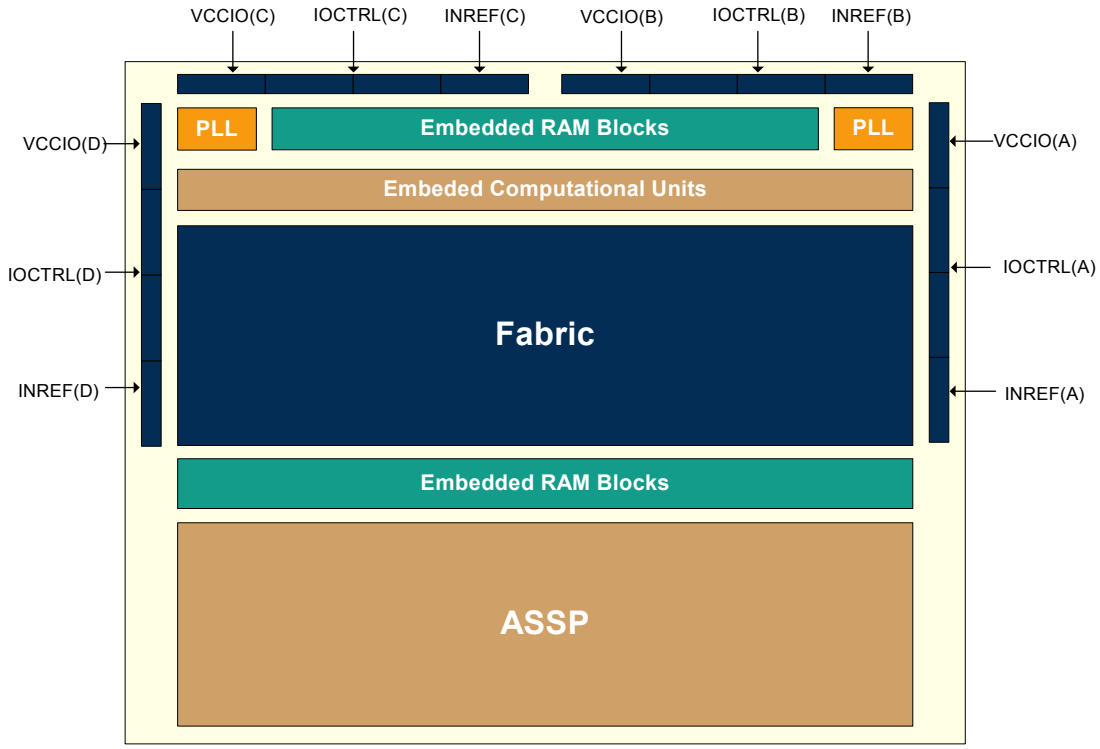
For registered control operation, the array logic drives the D input of the OE register which in turn drives the three-state control through a multiplexer. The multiplexer allows either a combinatorial or a registered signal to be driven to the three-state control.

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 QL901M contains four I/O banks. **Figure 16** 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 16: Multiple I/O Banks



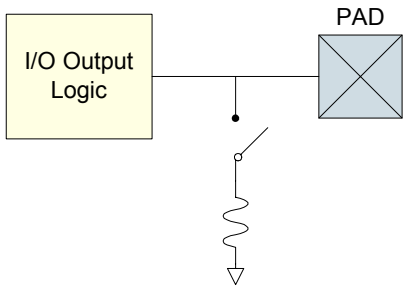
Programmable Slew Rate

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 38** and **Table 39** 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.

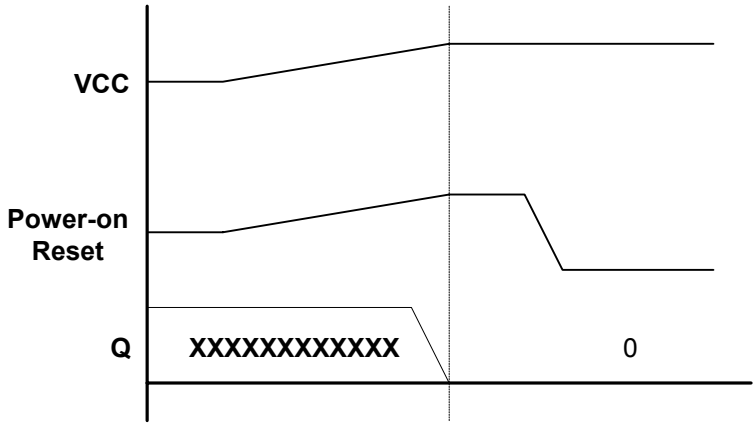
Figure 17: Programmable I/O Weak Pull-Down



Global Power-On Reset (POR)

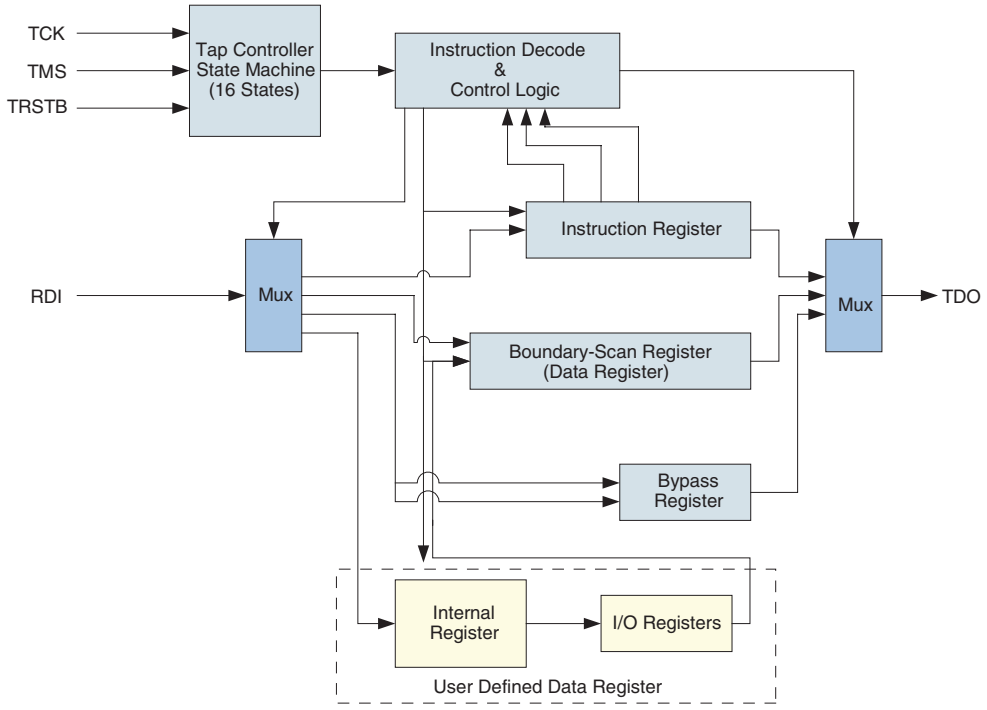
The QL901M family of devices features a global power-on reset. This reset is hardwired to all registers and resets them to Logic '0' upon power-up of the device. In QuickLogic devices, the asynchronous Reset input to flip-flops has priority over the Set input; therefore, the Global POR will reset all 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 2.5 V.

Figure 18: Power-On Reset



Joint Test Access Group (JTAG)

Figure 19: 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 Fuses

There are two security links: one to disable reading logic from the array, and the second 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 fuses is selectable through QuickWorks in the Tools/Options/Device Programming window in SpDE.

Flexibility Fuse

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 through 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 bit is selectable through QuickWorks in the Tools/Options/Device Programming window in SpDE. For more information on Power-Up Loading refer to QuickLogic Application Note 55 at http://www.quicklogic.com/images/appnote55.pdf.

NOTE: All JTAG inputs are clamped to the V_{CC} rail, not V_{CCIO}. Therefore, these pins can only be driven up to V_{CC} + 0.3 V. These input pins are LVCMOS2 compliant only (2.5 V). All JTAG outputs are driven by the V_{CC} rail, not V_{CCIO}. Therefore, these output pins can only drive up to V_{CC}. These output pins are LVCMOS2 compliant only (2.5 V).

Electrical Specifications

DC Characteristics

The DC Specifications are provided in **Table 14** through **Table 18**.

Table 14: Absolute Maximum Ratings

Parameter	Value	Parameter	Value
VCC Voltage	-0.5 V to 3.6 V	DC Input Current	±20 mA
VCCIO(A:D) Voltage	-0.5 V to 4.6 V	ESD Pad Protection	±2000 V
INREF(A:D) Voltage	2.7 V	BGA Package Storage Temperature	-55° C to +125° C
Input Voltage	-0.5 V to VCCIO(A:D) + 0.5 V	VCCIO Voltage	-0.5 V to 4.0 V
Latch-up Immunity	±100 mA		

Table 15: Operating Range

Symbol	Parameter		Industrial		Commercial		Unit
			Min.	Max.	Min.	Max.	
VCC	Supply Voltage for QL901M		2.3	2.7	2.3	2.7	V
VCCIO	ASSP I/O Supply Voltage for QL901M		3.0	3.6	3.0	3.6	V
VCCIO(A:D)	Fabric I/O Bank Reference Voltages		2.3	3.6	2.3	3.6	V
TC	Case Temperature		-40	85	0	70	°C
K	Delay Factor	100 MHz Speed Grade	0.43	1.80	0.46	1.76	n/a
		133 MHz Speed Grade	0.46	1.26	0.46	1.23	n/a

Table 16: DC Characteristics

Symbol	Parameter	Conditions	Min.	Max.	Units
I_{CC}	D.C. Supply Current	V_I, V_O = VCCIO, VCCIO(A:D) or GND	-	TBD	mA

Table 17: 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	-10	10	µA
I_{OZ}	3-State Output Leakage Current	V_I = VCCIO(A:D) or GND	-	10	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

Table 17: Fabric I/O DC Characteristics (Continued)

Symbol	Parameter	Conditions	Min.	Max.	Units
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	-	150	µA
I_{CCIO}	D.C. Supply Current on VCCIO(A:D)	VCCIO(A:D) = 3.6 V VCCIO(A:D) = 2.5 V	-	2	mA

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

Table 18: 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
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 18** are JEDEC and PCI Specifications. QuickLogic devices either meet or exceed these requirements.

NOTE: All CLK, DEDCLK, PLLIN, PLLRST, and IOCTRL input pins are clamped to the V_{CC} rail. Therefore, these pains can be driven up to V_{CC}+0.3 V.

AC Characteristics

The AC Specifications in this section are shown at V_{CC} = 1.8 V, TA = 25° C, Worst Case Corner, Fabric Speed Grade = -6 (-6 Fabric Speed Grade = 133 MHz CPU Speed, K = 1.07) unless otherwise indicated.

ASSP PLL

Table 19: ASSP PLL Timing Parameters

Jitter	Standby Current (µA)	Frequency Range	Minimum Lock Frequency	Duty Cycle	Crystal Accuracy	Lock Time
<200ps	157 µA	40-66.6 MHz	25 MHz	60/40	200 PPM	10 µs

SDRAM Controller

Figure 20: SDRAM Waveforms

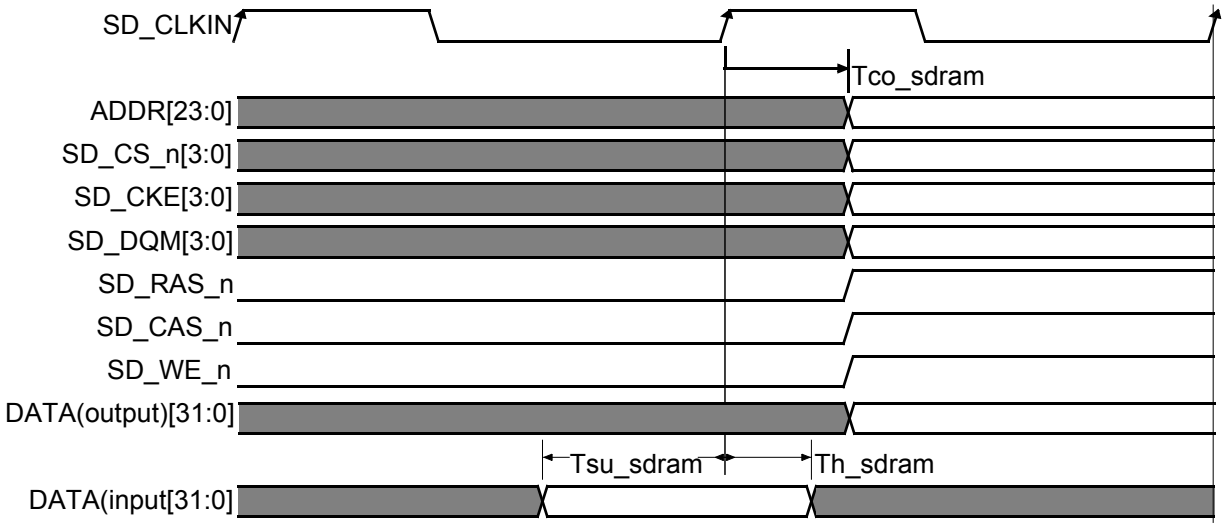


Table 20: SDRAM AC Timing

Parameter[a]		100 MHz		133 MHz		Units
		Min	Max	Min	Max	
Tco	DATA, ADDR, SD_RAS_n, SD_CAS_n, SD_CS_n[3:0], SD_DQM[3:0], SD_WE_n, SD_CKE[3:0]	2	8	2	6	ns
Tsu	DATA	12		9		ns
Th	DATA	2		2		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 21: SRAM Read Waveforms

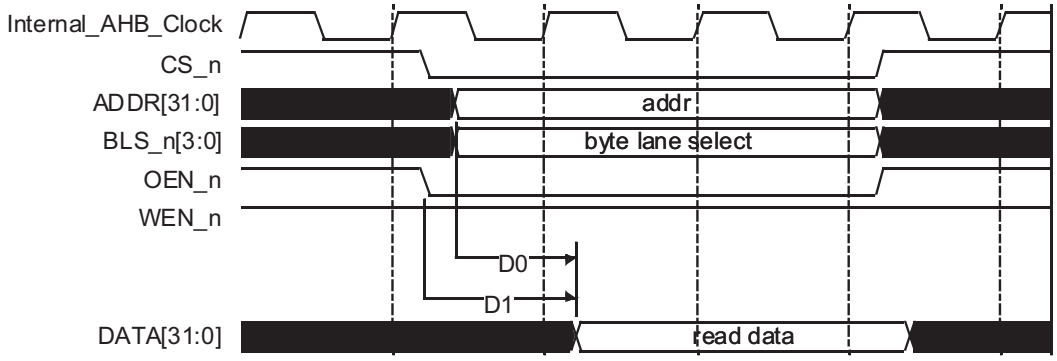


Table 21: SRAM AC Read Timing Requirements

Parameter		100 MHz		133 MHz		Units
		Min.	Max.	Min.	Max.	
D0	Access time, address and byte lane output to read data valid[a]	-	TBD	-	TBD	ns
D1	Access time, output enable to read data valid	-	TBD	-	TBD	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 22: SRAM Write Waveforms

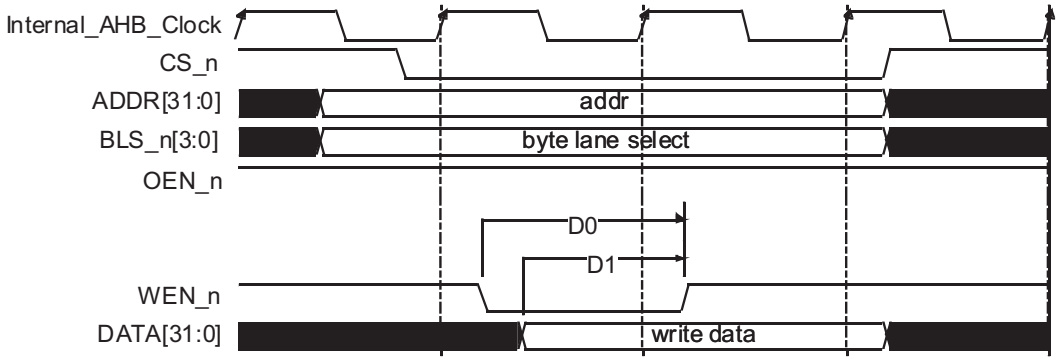


Table 22: SRAM AC Write Timing Characteristics

Parameter		100 MHz		133 MHz		Units
		Min.	Max.	Min.	Max.	
D0	Write enable low pulse width	TBD	-	TBD	-	hclk period
D1	Write output data valid before rising edge of write enable[a]	TBD	-	TBD	-	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 23: PCI Waveforms

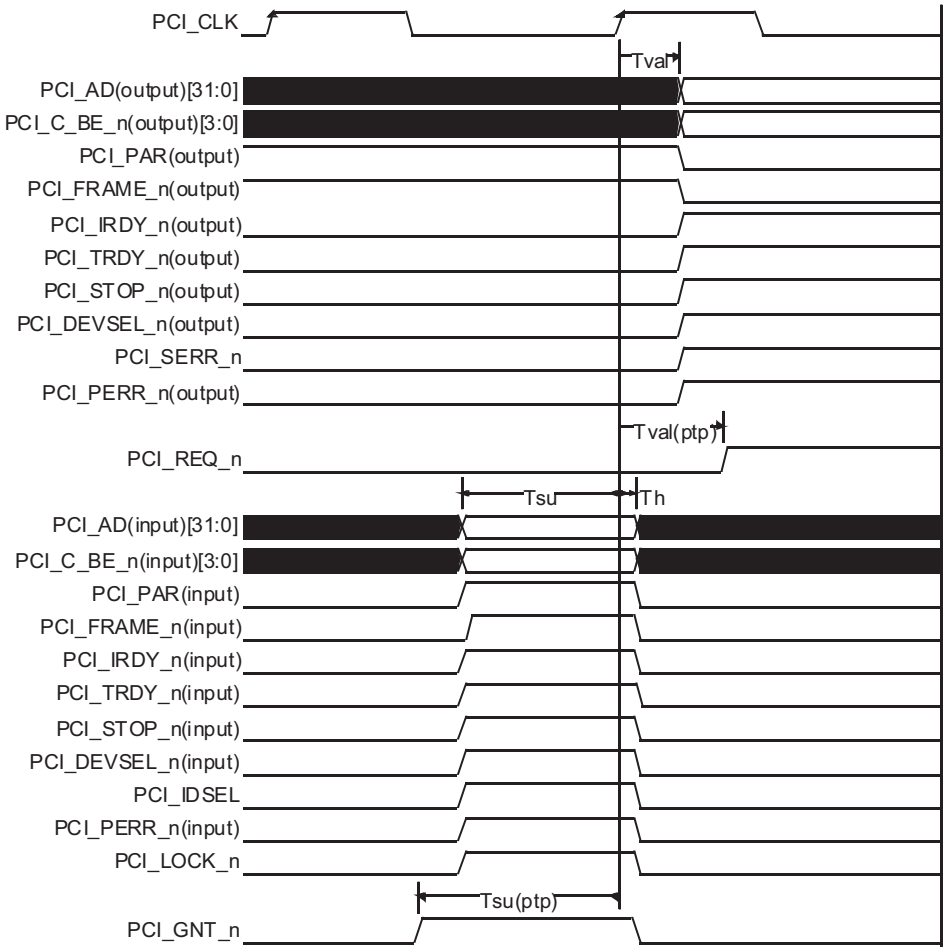


Table 23: PCI AC Timing

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
$t_{RST-CLK}$	Reset Active Time after PCI_CLK stable	100	-	100	-	µs
$t_{RST-OFF}$[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.

Ethernet Controllers

Figure 24: Ethernet MAC Transmit Interface Waveforms

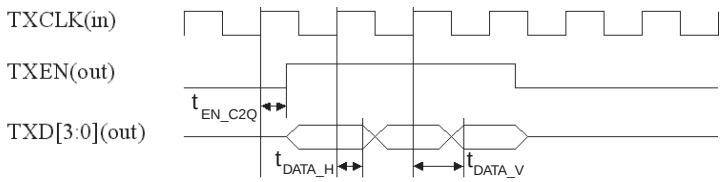


Table 24: Ethernet MAC Transmit Interface AC Timing

Parameter		Min.	Max.	Units
t_{EN_C2Q}	Time from the rising clock edge of TXCLK to the change in TXEN	-	8.0	ns
t_{DATA_V}	Time from the rising clock edge of TXCLK to all data signals having valid stable values	-	9.0	ns
t_{DATA_H}	Time in which the output data is still valid after the rising clock edge of TXCLK	0.0	-	ns

Figure 25: Ethernet MAC Receive Interface Waveforms

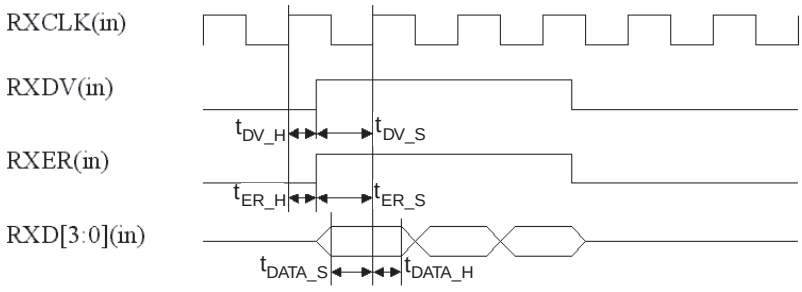


Table 25: Ethernet MAC Receive Interface AC Timing

Parameter		Min.	Max.	Units
t_{DV_S}	RXDV (receive data valid) to RXCLK setup time	2.0	-	ns
t_{DV_H}	RXDV (receive data valid) from RXCLK hold time	2.0	-	ns
t_{ER_S}	RXER (receive data error) to RXCLK setup time	2.0	-	ns
t_{ER_H}	RXER (receive data error) from RXCLK hold time	2.0	-	ns
t_{DATA_S}	RXD (receive data) to RXCLK setup time	2.0	-	ns
t_{DATA_H}	RXD (receive data) from RXCLK hold time	2.0	-	ns

The timing of the MII Management Interface listed below depends on the system clock frequency. The numbers displayed are correct for a processor clock frequency of 100 MHz and an AMBA bus system clock frequency of 50 MHz. Note that for a system clock of 133 MHz, the mandatory MDC minimum clock cycle of 400 ns for some PHY devices will not be met.

Figure 26: MII Management Interface Waveforms (1 of 2)

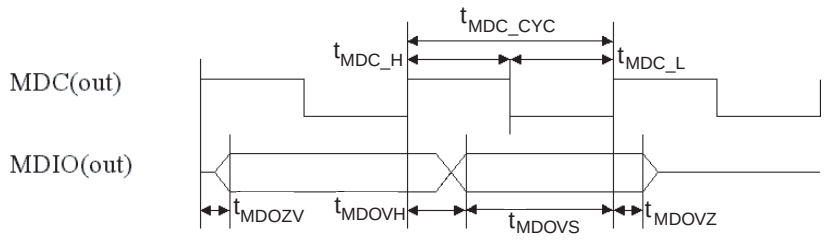


Table 26: MII Management Interface AC Timing (1 of 2)

Parameter		Min.	Max.	Units
t_{MDC_CYC}	MDC cycle time	520	-	ns
t_{MDC_H}	MDC high time	260	-	ns
t_{MDC_L}	MDC low time	260	-	ns
t_{MDOZV}	MDIO output high impedance to valid time from rising edge of MDC	40	-	ns
t_{MDOVZ}	MDIO output valid to high impedance time from rising edge of MDC	40	-	ns
t_{MDOS}	MDIO output valid before MDC rising edge	440	-	ns
t_{MDOVH}	MDIO output valid from MDC rising edge	40	-	ns

Figure 27: MII Management Interface Waveforms (2 of 2)

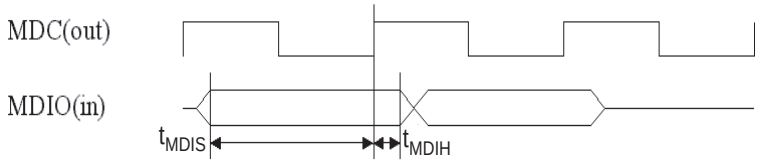


Table 27: MII Management Interface AC Timing (2 of 2)

Parameter		Min.	Max.	Units
t_{MDIS}	MDIO setup time to MDC	25	-	ns
t_{MDIH}	MDIO hold time to MDC	0	-	ns

Logic Cells

Figure 28: QL901M Logic Cell

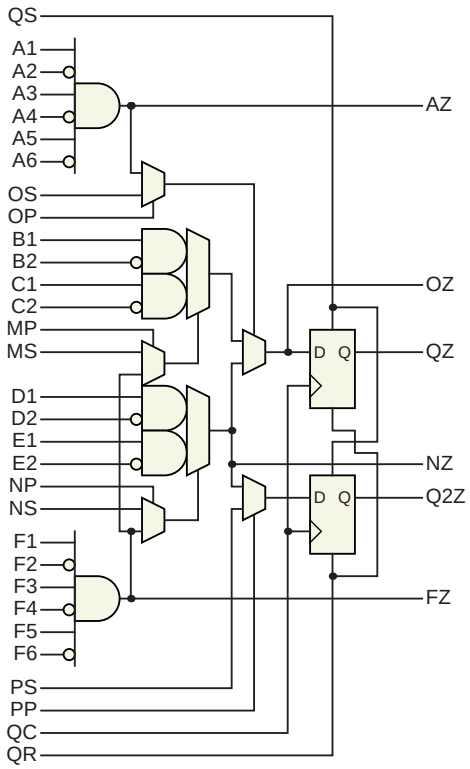


Figure 29: Logic Cell Flip-Flop

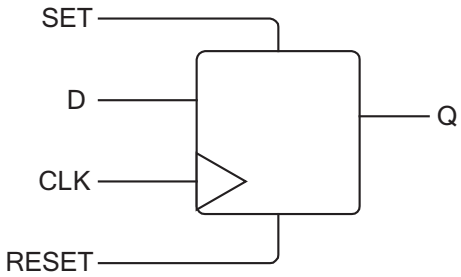


Figure 30: Logic Cell Flip-Flop Timings—First Waveform

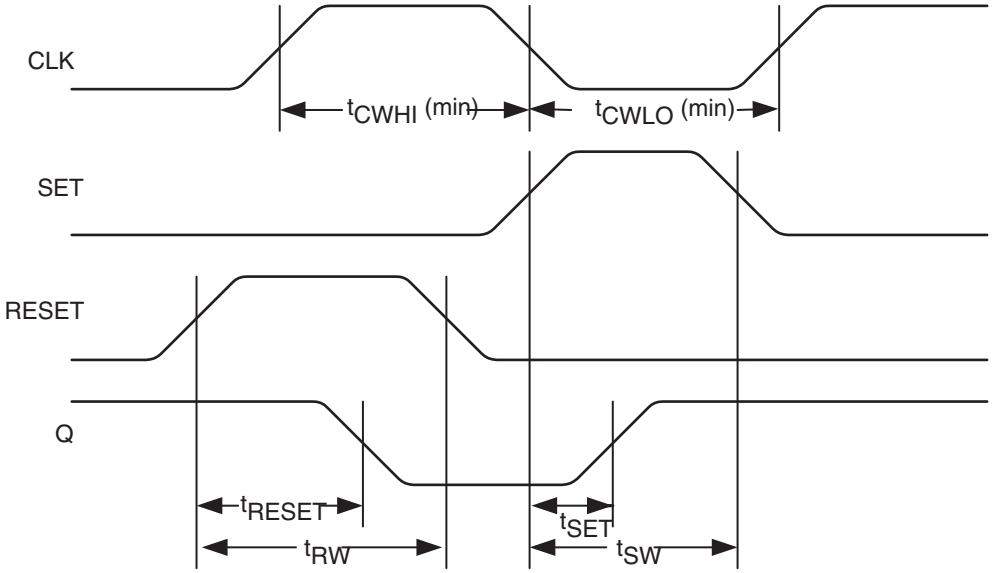


Figure 31: Logic Cell Flip-Flop Timings—Second Waveform

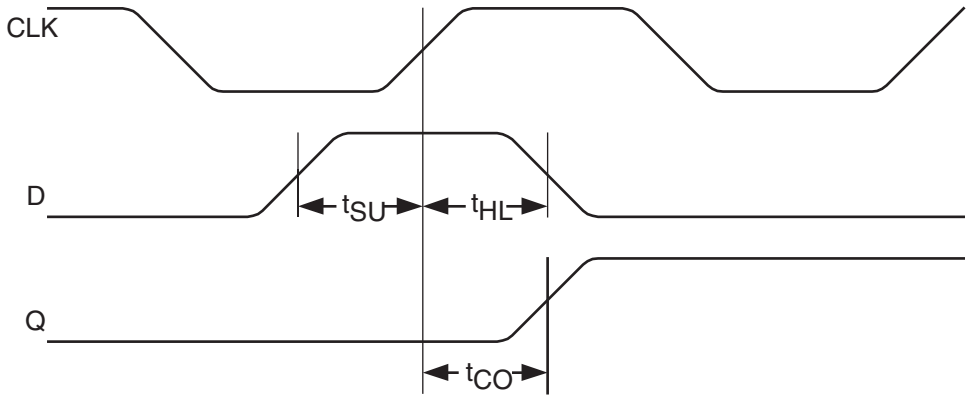


Table 28: Logic Cells

Symbol	Parameter	Value	
Logic Cells		Min.	Max.
t_{PD}	Combinatorial Delay of the longest path: time taken by the combinatorial circuit to output	0.205	1.01
t_{SU}	Setup time: time the synchronous input of the flip-flop must be stable before the active clock edge	0.231	-
t_{HL}	Hold time: time the synchronous input of the flip-flop must be stable after the active clock edge	0	-
t_{CO}	Clock-to-out delay: the amount of time taken by the flip-flop to output after the active clock edge.	-	0.427
t_{CWHI}	Clock High Time: required minimum time the clock stays high	0.46	-
t_{CWLO}	Clock Low Time: required minimum time that the clock stays low	0.46	-
t_{SET}	Set Delay: time between when the flip-flop is “set” (high and when the output is consequently “set” (high)	-	0.585
t_{RESET}	Reset Delay: time between when the flip-flop is ”reset” (low) and when the output is consequently “reset” (low)	-	0.658
t_{SW}	Set Width: time that the SET signal must remain high/low	0.3	-
t_{RW}	Reset Width: time that the RESET signal must remain high/low	0.3	-

Dual-Port SRAM Modules

Figure 32: RAM Module

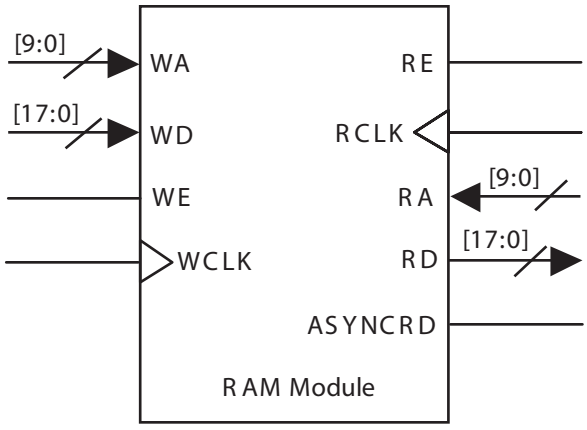


Figure 33: RAM Cell Synchronous Write Timing

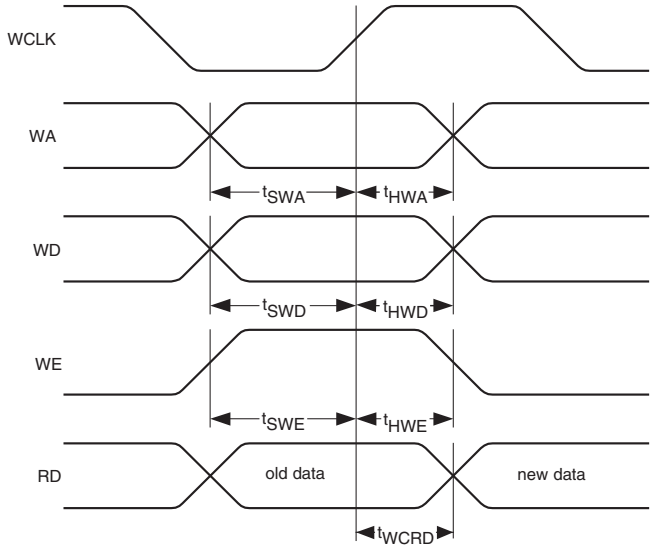


Table 29: RAM Cell Synchronous Write Timing

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	0.276 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	-	2.796 ns

Figure 34: RAM Cell Synchronous and Asynchronous Read Timing

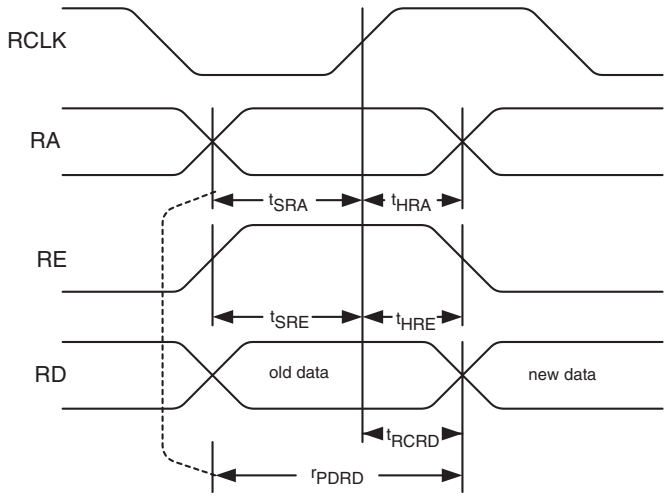


Table 30: RAM Cell Synchronous and Asynchronous Read Timing

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	-	2.225 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	-	2.405 ns

ECUs

Figure 35: ECU Block Diagram

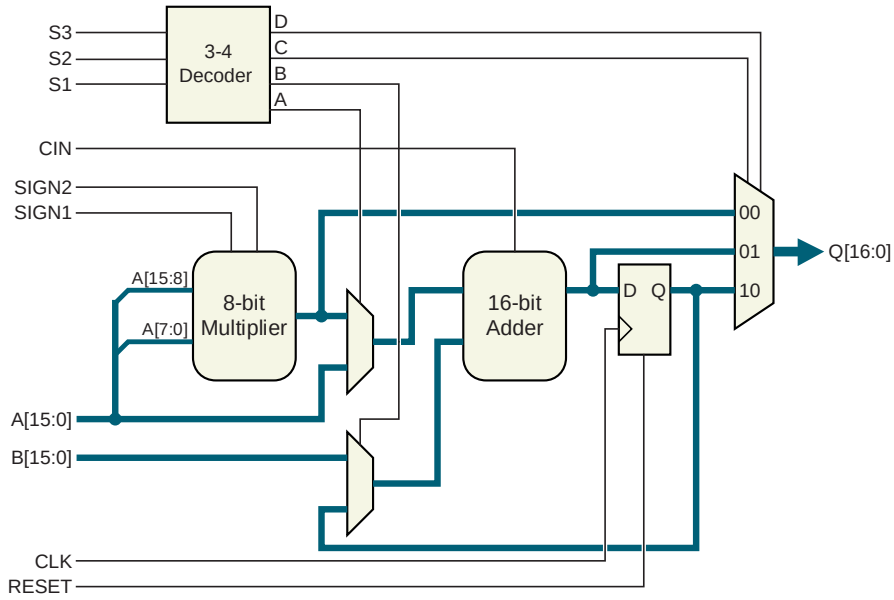


Table 31: ECU Mode Select Criteria

Instruction			Operation	ECU Performance[a], -6 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 31** represent -6 Worst Case Commercial conditions.

Clock Network

Table 32: 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 36: Dedicated Clock Structure Schematic

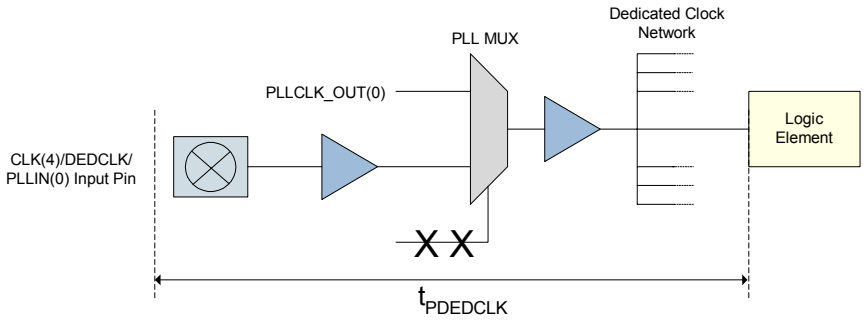


Table 33: Dedicated Clock Network Performance

Symbol	Parameters	Value	
		Min.	Max.
$t_{PDEDCLK}$	Delay from dedicated clock input pin to logic cell flip-flop	-	1.73 ns
$t_{SKEWDEDCLK}$	Skew on dedicated clock network	-	TBD ns

NOTE: When using the Fabric PLL, $t_{PDEDCLK}$ is effectively zero due to delay adjustment by PLL.

Figure 37: Global Clock Structure Schematic

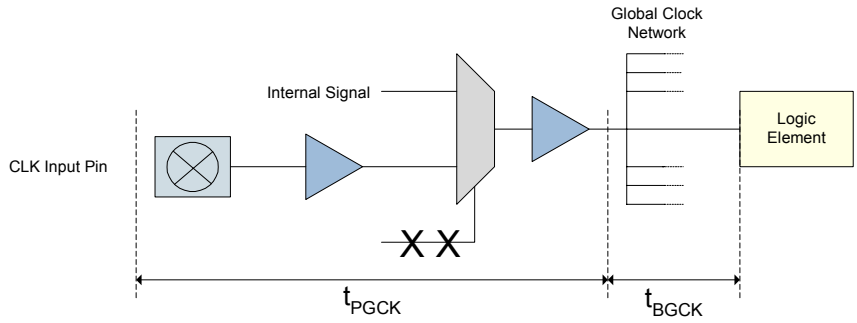


Table 34: Global Clock Network Performance

Symbol	Parameter	Value	
		Min.	Max.
t_{PGCK}	Global clock pin delay to quad net	-	1.34 ns
t_{BGCK}	Global clock tree delay (quad net to logic cell flip-flop)	-	0.56 ns
$t_{SKEWGCK}$	Skew on global clock network	-	TBD ns

NOTE: When using the Fabric PLL, t_{PGCK} and t_{BGCK} are effectively zero due to delay adjustment by PLL.

I/O Cells

Figure 38: QL901M Input Register Cell

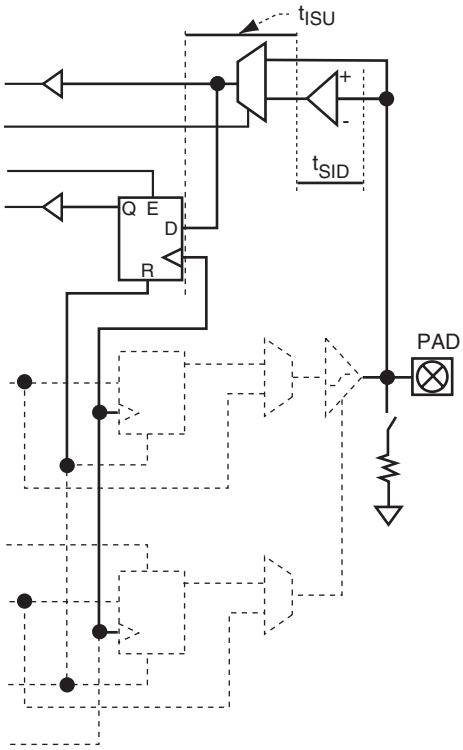


Table 35: Standard Input Delays

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	-	0.34 ns
t_{SID} (LVCMOS2)	LVCMOS2 input delay: Low Voltage CMOS for 2.5 V and lower applications	-	0.42 ns
t_{SID} (GTL+)	GTL+ input delay: Gunning Transceiver Logic	-	0.68 ns
t_{SID} (SSTL3)	SSTL3 input delay: Stub Series Terminated Logic for 3.3 V	-	0.55 ns
t_{SID} (SSTL2)	SSTL2 input delay: Stub Series Terminated Logic for 2.5 V	-	0.61 ns

a. See **Table 36** for t_{ISU} value.

Figure 39: QL901M Input Register Cell Timing

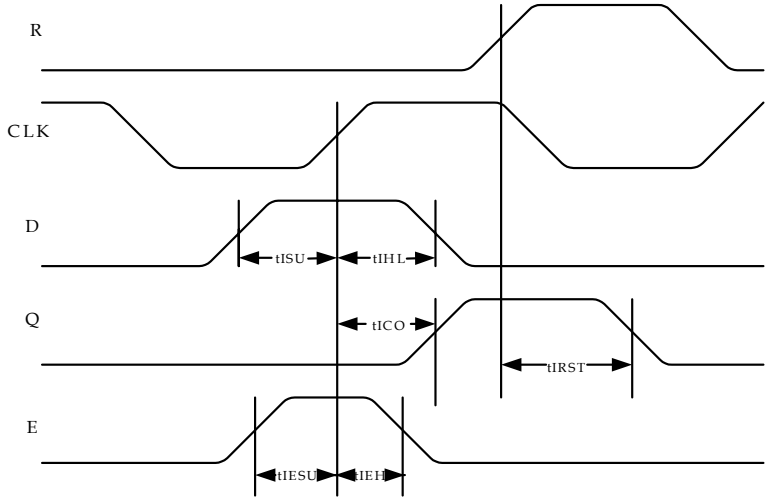


Table 36: Input Register Cell

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.	3.308 ns	3.526 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.	-	0.494 ns
t_{IRST}	**Input register reset delay:** time between when the flip-flop is “reset”(low) and when the output is consequently “reset” (low).	-	0.464 ns
t_{IESU}	**Input register clock enable setup time:** time “enable” must be stable before the active clock edge.	0.830 ns	0.987 ns
t_{IEH}	**Input register clock enable hold time:** time “enable” must be stable after the active clock edge.	0 ns	-

Figure 40: QL901M Output Register Cell

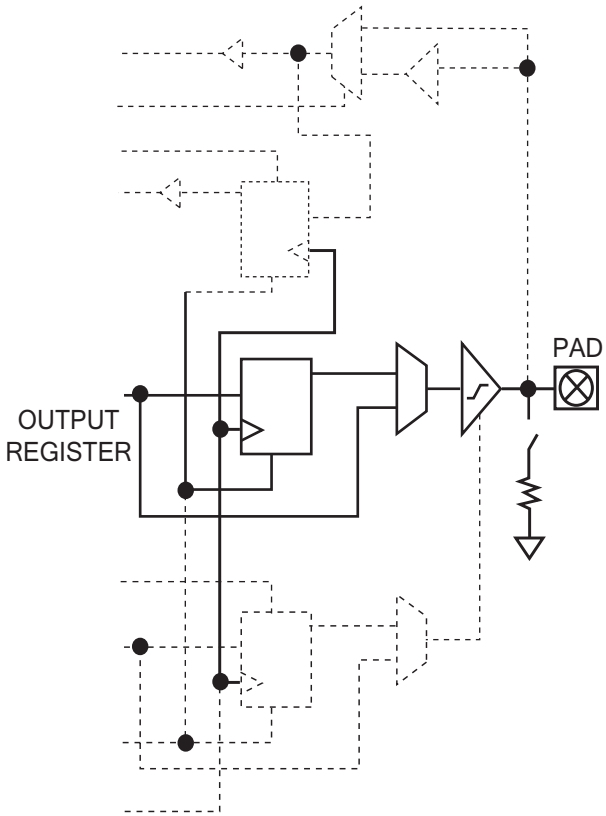


Figure 41: QL901M Output Register Cell Timing

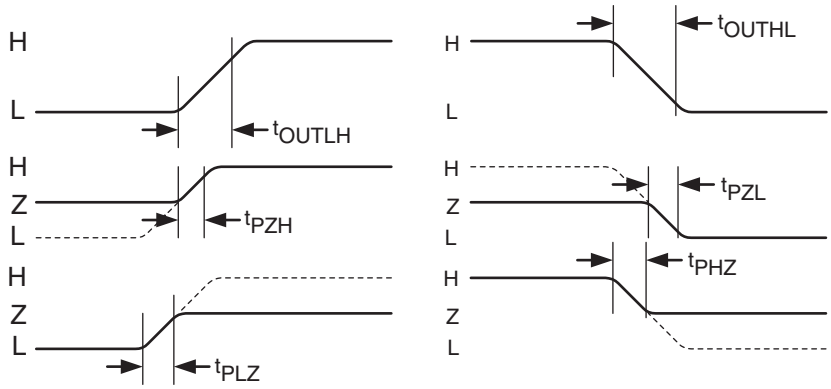


Table 37: Output Register Cell

Symbol	Parameter	Value	
		Min.	Max.
Output Register Cell Only			
t_{OUTLH}	**Output Delay low to high** (90% of H)	-	0.40 ns
t_{OUTHL}	**Output Delay high to low** (10% of L)	-	0.55 ns
t_{PZH}	**Output Delay tri-state to high** (90% of H)	-	2.94 ns
t_{PZL}	**Output Delay tri-state to low** (10% of L)	-	2.34 ns
t_{PHZ}	**Output Delay high to tri-state**	-	3.07 ns
t_{PLZ}	**Output Delay low to tri-state**	-	2.53 ns
t_{COP}	**Clock-to-out delay:** time taken by the flip-flop to output after the active clock edge. (Does not include clock tree delays.)	-	3.15 ns (fast slew) 10.2 ns (slow slew)

Table 38: 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 39: 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

ASSP to Fabric Timing

Table 40 and **Table 41** list the synchronous and asynchronous timing for the QL901M 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 twice the hclk frequency. This internal clock is not driven to the Fabric.
- All AF_PCI_* signals are static.

Table 40: 100MHz and 133MHz QuickMIPS Interface Port Synchronous Timing (to hclk)

Signal Name	100MHz			133MHz		
	Setup Time (Tsu)	Hold Time (Thold)	Clock-to-out Time (Tco)	Setup Time (Tsu)	Hold Time (Thold)	Clock-to-out Time (Tco)
hresetn	X[a]	X	5.11	X		3.84
Fabric AHB Slave Ports						
ahbs_hsel	X	X	10.73	X	X	8.07
ahbs_haddr	X	X	10.56	X	X	7.94
ahbs_htrans	X	X	11.35	X	X	8.53
ahbs_hwrite	X	X	8.32	X	X	6.26
ahbs_hsize	X	X	9.50	X	X	7.14
ahbs_hburst	X	X	9.12	X	X	6.86
ahbs_hprot	X	X	9.66	X	X	7.26
ahbs_hwdata	X	X	13.07	X	X	9.83
ahbs_hrdata	5.94	0	X	4.47	0	X
ahbs_hready_out	9.55	0	X	7.18	0	X
ahbs_hresp	10.39	0	X	7.81	0	X
Fabric AHB Master Ports						
ahbm_haddr	11.94	0	X	8.98	0	X
ahbm_htrans	11.33	0	X	8.52	0	X
ahbm_hwrite	10.39	0	X	7.81	0	X
ahbm_hsize	10.58	0	X	7.96	0	X
ahbm_hburst	10.79	0	X	8.11	0	X
ahbm_hprot[b]	–	–	X	–	–	X
ahbm_hwdata	10.39	0	X	7.81	0	X
ahbm_hrdata	X	X	16.28	X	X	12.24
ahb_hready_in	X	X	11.82	X	X	8.89
ahbm_hresp	X	X	9.02	X	X	6.78
ahbm_hbusreq	7.40	0	X	5.56	0	X
ahbm_hgrant	X	X	16.64	X	X	12.51

Table 40: 100MHz and 133MHz QuickMIPS Interface Port Synchronous Timing (to hclk) (Continued)

Signal Name	100MHz			133MHz		
	Setup Time (Tsu)	Hold Time (Thold)	Clock-to-out Time (Tco)	Setup Time (Tsu)	Hold Time (Thold)	Clock-to-out Time (Tco)
Fabric APB Slave Ports						
apbs_paddr	X	X	4.52	X	X	3.40
apbs_pwdata	X	X	4.66	X	X	3.50
apbs_penable	X	X	2.87	X	X	2.16
apbs_pwrite	X	X	4.13	X	X	3.11
apbs_psel0	X	X	3.80	X	X	2.86
apbs_psel1	X	X	3.43	X	X	2.58
apbs_psel2	X	X	3.25	X	X	2.44
apbs_prdata0	7.44	0	X	5.59	0	X
apbs_prdata1	6.79	0	X	5.11	0	X
apbs_prdata2	6.97	0	X	5.24	0	X
Timer Ports						
tm_fbenable	0.23	0	X	0.17	0	X
tm_overflow2	X	X	4.35	X	X	3.27
tm_overflow3	X	X	4.48	X	X	3.37
tm_overflow4	X	X	5.00	X	X	3.76

a. “x” indicates that this timing delay does not apply to the signal.
b. The ahbm_hprot signal is NOT used by any slave within the standard cell part of the chip. None of the masters besides the processor-AHB-bridge generates this signal. Therefore there is no setup or hold timing for ahbm_hprot.

Table 41: QuickMIPS Interface Port Asynchronous Timing

Start Port	End Port	Propagation Delay (Tprop)	
		100MHz	133MHz
ahbm_haddr	ahbs_haddr	8.39	6.31
ahbm_haddr	ahbs_hsel	7.71	5.80
ahbm_htrans	ahbs_htrans	6.48	4.87
ahbm_hwrite	ahbs_hwrite	6.21	4.67
ahbm_hsize	ahbs_hsize	6.04	4.54
ahbm_hburst	ahbs_hburst	5.70	4.29
ahbm_hprot	ahbs_hprot	7.07	5.32
ahbm_hwdata	ahbs_hwdata	8.15	6.13
ahbs_hrdata	ahbm_hrdata	5.78	4.35
ahbs_hready_out	ahb_hready_in	5.03	3.78
ahbs_hresp	ahbm_hresp	4.98	3.74
ahbm_hbusreq	ahbm_hgrant	10.14	7.62
ahbs_hresp	ahbm_hgrant	10.50	7.89
apbs_prdata0	ahbm_hrdata	8.28	6.23
apbs_prdata1	ahbm_hrdata	7.51	5.65
apbs_prdata2	ahbm_hrdata	7.57	5.69

Fabric Kv and Kt Graphs

Figure 42: Voltage Factor vs. Supply Voltage

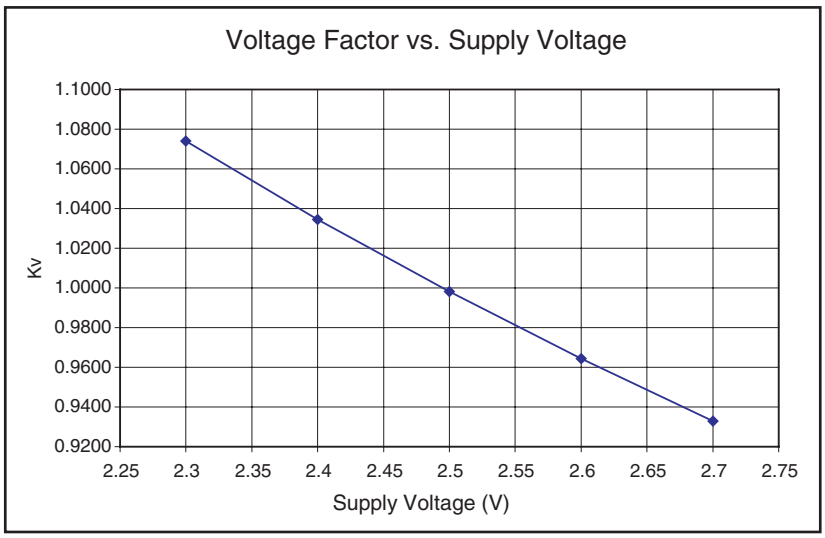
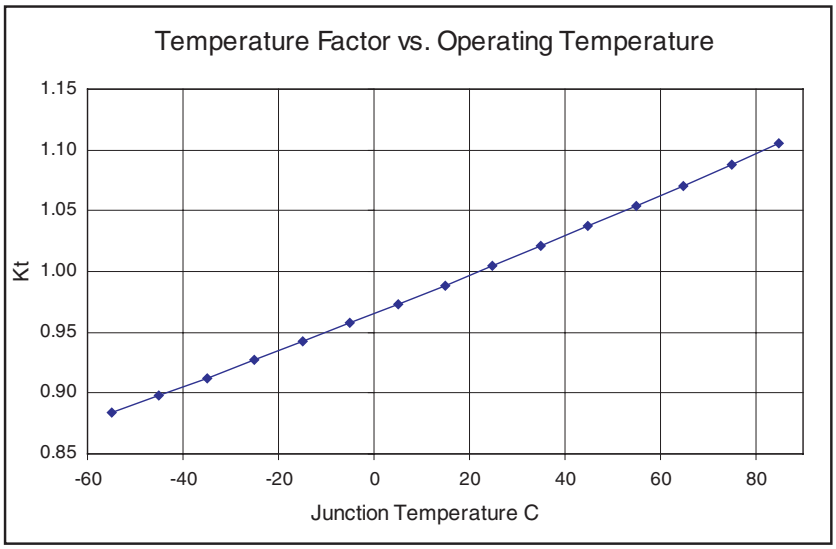


Figure 43: Temperature Factor vs. Operating Temperature



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 42**, pick an appropriate T_{AMAX} and use:

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

Table 42: Package Thermal Characteristics

Package Description		θ_{JA} (° C/W) @ Various Flow Rates (m/sec)				θ_{JC} (° C/W)
Pin Count	Package Type	0	0.5	1	2	
680	PBGA	22.0	20.0	19.0	18.0	8.0

Power vs. Operating Frequency

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

$$P_{TOTAL}\ (mW) = 0.350 + f_{Fabric}\ [0.0031\ \eta_{LC} + 0.0948\ \eta_{CKBF} + 0.01\ \eta_{CLBF} + 0.0263\ \eta_{CKLD} + 0.543\ \eta_{RAM} + 0.20\ \eta_{PLL} + 0.0035\ \eta_{INP} + 0.0257\ \eta_{OUTP}] + 28.1\ f_{asspio} + 10.7\ f_{mips}$$

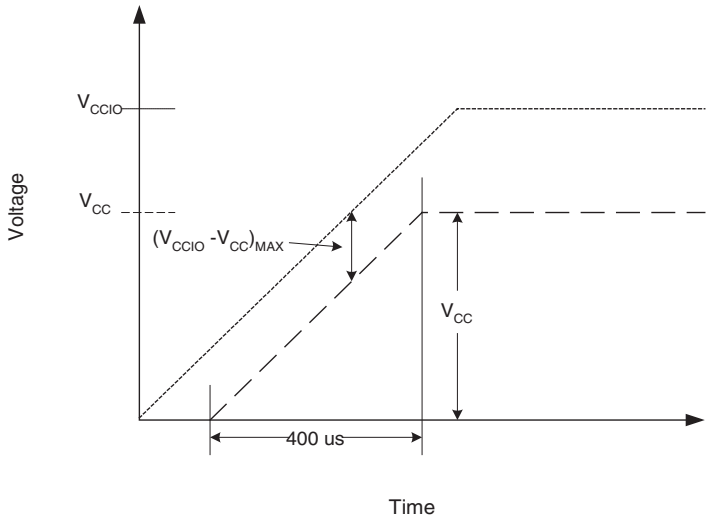
Where

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 44: Power-Up Sequencing



When powering up a device, the VCC/VCCIO rails must take 400 µs or longer to reach the maximum value (refer to **Figure 44**).

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

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

Signal Descriptions

Pin Descriptions

Table 43: Pin Descriptions

Pin	I/O	Function
PCI Signals		
PCI_AD(31:0)	I/O	**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.
PCI_C_BE_n(3:0)	I/O	**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). **PCI_C_BE_n(3:0)** — **Bus Command** 0000 — Interrupt Acknowledge 0001 — Special Cycle 0010 — I/O Read 0011 — I/O Write 0110 — Memory Read 0111 — Memory Write 1010 — Configuration Read 1011 — Configuration Write 1100 — Memory Read Multiple 1101 — Dual Address Cycle 1110 — Memory Read Line 1111 — Memory Write and Invalidate 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.
PCI_DEVSEL_n	I/O	**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.

Table 43: 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 43: 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.

Table 43: Pin Descriptions (Continued)

Pin	I/O	Function
Ethernet Controller 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.
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.
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_MDIO, M2_MDIO	I	**Ethernet Management Data In/Out.** 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. This is also 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_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.

Table 43: Pin Descriptions (Continued)

Pin	I/O	Function
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.
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.
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.

Table 43: Pin Descriptions (Continued)

Pin	I/O	Function
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.
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.
WEN_n	O	**SRAM Write Enable.** WEN_n indicates whether transactions between the QuickMIPS chip and the external SRAM are reads (WEN_n is high) or writes (WEN_n is low).
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_n	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.

Table 43: Pin Descriptions (Continued)

Pin	I/O	Function
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_n	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.
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.
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.
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.
EJTAG_TRST	I	**EJTAG Test Reset.** This signal is asserted asynchronously to reset the TAP Controller, Instruction register, and EJTAGBOOT indication.
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.
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)/PLLIN(1), CLK(2:0)	I	**Programmable Global Clock Pin.** This pin provides access to a global 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. In addition, this input also serves as the input clock reference to PLL1. This input pin is LVCMOS2 compliant only (2.5 V). Connect to 2.5 V or GND if unused.

Table 43: Pin Descriptions (Continued)

Pin	I/O	Function
CLK(4)/ DEDCLK/PLLIN(0)	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). In addition, this input also serves as the input clock reference to PLL0. This input pin is LVCMOS2 compliant only (2.5 V). Connect to 2.5 V 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 18** for the appropriate standard. Connect to GND when using TTL, PCI or LVCMOS.
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 Ground on this pin. This input pin is LVCMOS2 compliant only (2.5 V). This pin should be tied to Ground if it is not used. If tied to Vcc, 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 2.5 V 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 2.5 V if not used for JTAG.
TCK	I	**Test Clock for JTAG.** Hold HIGH or LOW during normal operation. Connect to 2.5 V 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.

Table 43: Pin Descriptions (Continued)

Pin	I/O	Function
Miscellaneous Signals		
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)). BOOT(1:0) = 00: 8-bit width BOOT(1:0) = 01: 16-bit width BOOT(1:0) = 10: 32-bit width 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.
STM	I	**QuickLogic Reserved Pin.** Tie to GND on the PCB.
PL_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, PL_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 PL_CLOCKIN clock cycles. For reliable operation, the power supply must be stable and the clock must be running before this signal is deasserted.
PL_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, PL_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 PL_CLOCKIN 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		
PLLRST(1:0)	I	**Fabric PLL Active High Reset.** If PLLRST is high, then the corresponding PLL outputs are driven to 0. This signal must be asserted and then released for the LOCK_DETECT signal to operate properly. This input pin is LVCMOS2 compliant only (2.5 V). Connect to 2.5 V if the Fabric PLL is not used.
PLLOUT(1:0)	O	**Fabric PLL Output Off Chip.** This is the Fabric PLL output clock driven off chip. Leave unconnected if unused.
ASSP PLL Signals		
PL_CLOCKIN	I	**Input Clock Signal.** This clock input is the reference clock used by the ASSP PLL. The frequency of the clock on this input is multiplied by two to drive the MIPS CPU. This input pin is LVCMOS2 compliant only (2.5 V).
PL_ENABLE	I	**Active High PLL Enable Signal.** This signal must be high to enable the ASSP-side PLL. If PL_ENABLE and PL_BYPASS are held low, the QL901M is put into a low power saving quiescent state.
PL_BYPASS	I	**PLL Bypass.** When high, the 2X multiplication of the input clock is not performed and the output clocks are equal to the input frequency.

Table 43: Pin Descriptions (Continued)

Pin	I/O	Function
PL_CLKOUT	O	**Output Clock Signal from the ASSP PLL.** This output operates at the same frequency that is driven to the MIPS CPU. Leave unconnected if unused.
PL_LOCK	O	**ASSP 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.
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 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 2.5 V or 3.3 V.
VCC	I	**Supply Pin.** Tie to 2.5 V supply.
GNDPLL(1:0)	I	**Fabric PLL Ground Pin.** Tie to analog GND on the PCB.
VCCPLL(1:0)	I	**Fabric PLL Voltage Supply Pin.** Tie to 2.5 V analog supply.
GNDPLL	I	**ASSP PLL Ground Pin.** Tie to analog GND on the PCB.
VCCPLL	I	**ASSP PLL Voltage Supply Pin.** Tie to 2.5 V analog supply.
VCCIO	I	**Voltage Supply Pin for ASSP I/O Signals.** This pin must be tied to 3.3 V.

ASSP Fabric Port Descriptions

Table 44: ASSP to Fabric Port Descriptions

Port	I/O[a]	Function
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 is fixed at 1/2 the CPU clock rate.
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.
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.

Table 44: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
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.
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.

Table 44: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
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.
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.

Table 44: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
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.
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.

Table 44: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
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.
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.

Table 44: ASSP to Fabric Port Descriptions (Continued)

Port	I/O[a]	Function
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 44: 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.

680 BGA Pinout Table

Table 45: 680 PBGA Pinout Table

680 PBGA	Function	680 PBGA	Function	680 PBGA	Function	680 PBGA	Function	680 PBGA	Function	680 PBGA	Function
A1	GND	B26	INREF<B>	D17	I/O<C>	F32	GND	M31	I/O<A>	R30	VCC
A2	GND	B27	I/O<B>	D18	I/O<B>	F33	I/O<B>	M32	IOCTRL<A>	R31	I/O<A>
A3	I/O<C>	B28	I/O<B>	D19	I/O<B>	F34	I/O<B>	M33	I/O<A>	R32	I/O<A>
A4	I/O<C>	B29	I/O<B>	D20	I/O<B>	G1	I/O<D>	M34	INREF<A>	R33	I/O<A>
A5	I/O<C>	B30	I/O<B>	D21	I/O<B>	G2	I/O<C>	N1	I/O<D>	R34	I/O<A>
A6	I/O<C>	B31	I/O<B>	D22	I/O<B>	G3	I/O<C>	N2	I/O<D>	T1	I/O<D>
A7	INREF<C>	B32	I/O<B>	D23	I/O<B>	G4	I/O<C>	N3	I/O<D>	T2	I/O<D>
A8	I/O<C>	B33	GND	D24	I/O<B>	G5	VCC	N4	I/O<D>	T3	I/O<D>
A9	I/O<C>	B34	GND	D25	I/O<B>	G30	VCC	N5	VCC	T4	I/O<D>
A10	I/O<C>	C1	GNDPLL<0>	D26	I/O<B>	G31	PLLOUT<0>	N13	GND	T5	I/O<D>
A11	I/O<C>	C2	I/O<C>	D27	I/O<B>	G32	I/O<B>	N14	VCCIO<C>	T13	VCCIO<D>
A12	I/O<C>	C3	GND	D28	I/O<B>	G33	I/O<B>	N15	VCCIO<C>	T14	GND
A13	I/O<C>	C4	I/O<C>	D29	I/O<B>	G34	I/O<B>	N16	VCCIO<C>	T15	GND
A14	I/O<C>	C5	I/O<C>	D30	I/O<B>	H1	I/O<D>	N17	VCCIO<C>	T16	GND
A15	I/O<C>	C6	I/O<C>	D31	GND	H2	I/O<D>	N18	VCCIO<B>	T17	GND
A16	I/O<C>	C7	I/O<C>	D32	I/O<B>	H3	I/O<C>	N19	VCCIO<B>	T18	GND
A17	GND	C8	I/O<C>	D33	I/O<B>	H4	I/O<C>	N20	VCCIO<B>	T19	GND
A18	GND	C9	I/O<C>	D34	I/O<B>	H5	VCCIO<D>	N21	VCCIO<B>	T20	GND
A19	CLK<5>	C10	IOCTRL<C>	E1	PLLRST<0>	H30	VCCIO<A>	N22	GND	T21	GND
A20	I/O<B>	C11	I/O<C>	E2	I/O<C>	H31	I/O<B>	N30	VCC	T22	VCCIO<A>
A21	I/O<B>	C12	I/O<C>	E3	I/O<C>	H32	I/O<B>	N31	IOCTRL<A>	T30	I/O<A>
A22	I/O<B>	C13	I/O<C>	E4	I/O<C>	H33	I/O<A>	N32	I/O<A>	T31	I/O<A>
A23	I/O<B>	C14	I/O<C>	E5	I/O<C>	H34	I/O<A>	N33	I/O<A>	T32	I/O<A>
A24	I/O<B>	C15	I/O<C>	E6	VCC	J1	I/O<D>	N34	I/O<A>	T33	I/O<A>
A25	I/O<B>	C16	I/O<C>	E7	VCC	J2	I/O<D>	P1	I/O<D>	T34	I/O<A>
A26	I/O<B>	C17	TMS	E8	VCCIO<C>	J3	I/O<D>	P2	I/O<D>	U1	GND
A27	IOCTRL<B>	C18	CLK<6>	E9	I/O<C>	J4	I/O<D>	P3	I/O<D>	U2	I/O<D>
A28	I/O<B>	C19	I/O<B>	E10	I/O<C>	J5	I/O<D>	P4	I/O<D>	U3	I/O<D>
A29	I/O<B>	C20	I/O<B>	E11	I/O<C>	J30	I/O<B>	P5	VCCIO<D>	U4	I/O<D>
A30	I/O<B>	C21	I/O<B>	E12	I/O<C>	J31	I/O<A>	P13	VCCIO<D>	U5	I/O<D>
A31	I/O<B>	C22	I/O<B>	E13	VCC	J32	I/O<A>	P14	GND	U13	VCCIO<D>
A32	I/O<B>	C23	I/O<B>	E14	VCCIO<C>	J33	I/O<A>	P15	GND	U14	GND
A33	GND	C24	I/O<B>	E15	VCC	J34	I/O<A>	P16	GND	U15	GND
A34	GND	C25	IOCTRL<B>	E16	I/O<C>	K1	I/O<D>	P17	GND	U16	GND
B1	GND	C26	I/O<B>	E17	I/O<C>	K2	I/O<D>	P18	GND	U17	GND
B2	GND	C27	I/O<B>	E18	I/O<B>	K3	I/O<D>	P19	GND	U18	GND
B3	I/O<C>	C28	I/O<B>	E19	I/O<B>	K4	I/O<D>	P20	GND	U19	GND
B4	I/O<C>	C29	I/O<B>	E20	VCC	K5	I/O<D>	P21	GND	U20	GND
B5	I/O<C>	C30	I/O<B>	E21	VCCIO<B>	K30	I/O<A>	P22	VCCIO<A>	U21	GND
B6	I/O<C>	C31	I/O<B>	E22	VCC	K31	I/O<A>	P30	VCCIO<A>	U22	VCCIO<A>
B7	I/O<C>	C32	GND	E23	I/O<B>	K32	I/O<A>	P31	I/O<A>	U30	I/O<A>
B8	IOCTRL<C>	C33	I/O<B>	E24	I/O<B>	K33	I/O<A>	P32	I/O<A>	U31	I/O<A>
B9	I/O<C>	C34	I/O<B>	E25	I/O<B>	K34	I/O<A>	P33	I/O<A>	U32	I/O<A>
B10	I/O<C>	D1	GND	E26	I/O<B>	L1	I/O<D>	P34	I/O<A>	U33	I/O<A>
B11	I/O<C>	D2	PLLOUT<1>	E27	VCCIO<B>	L2	I/O<D>	R1	I/O<D>	U34	GND
B12	I/O<C>	D3	I/O<C>	E28	VCC	L3	I/O<D>	R2	I/O<D>	V1	GND
B13	I/O<C>	D4	I/O<C>	E29	VCC	L4	I/O<D>	R3	I/O<D>	V2	I/O<D>
B14	I/O<C>	D5	I/O<C>	E30	I/O<B>	L5	I/O<D>	R4	I/O<D>	V3	I/O<D>
B15	I/O<C>	D6	I/O<C>	E31	I/O<B>	L30	I/O<A>	R5	VCC	V4	I/O<D>
B16	I/O<C>	D7	I/O<C>	E32	VCCPLL<1>	L31	I/O<A>	R13	VCCIO<D>	V5	I/O<D>
B17	CLK<8>	D8	I/O<C>	E33	PLLRST<1>	L32	I/O<A>	R14	GND	V13	VCCIO
B18	CLK<7>	D9	I/O<C>	E34	GNDPLL<1>	L33	I/O<A>	R15	GND	V14	GND
B19	I/O<B>	D10	I/O<C>	F1	I/O<C>	L34	I/O<A>	R16	GND	V15	GND
B20	I/O<B>	D11	I/O<C>	F2	I/O<C>	M1	IOCTRL<D>	R17	GND	V16	GND
B21	I/O<B>	D12	I/O<C>	F3	VCCPLL<0>	M2	INREF<D>	R18	GND	V17	GND
B22	I/O<B>	D13	I/O<C>	F4	I/O<C>	M3	IOCTRL<D>	R19	GND	V18	GND
B23	I/O<B>	D14	I/O<C>	F5	VCC	M4	I/O<D>	R20	GND	V19	GND
B24	I/O<B>	D15	I/O<C>	F30	VCC	M5	I/O<D>	R21	GND	V20	GND
B25	I/O<B>	D16	I/O<C>	F31	I/O<B>	M30	I/O<A>	R22	VCCIO<A>	V21	GND

(Sheet 1 of 2)

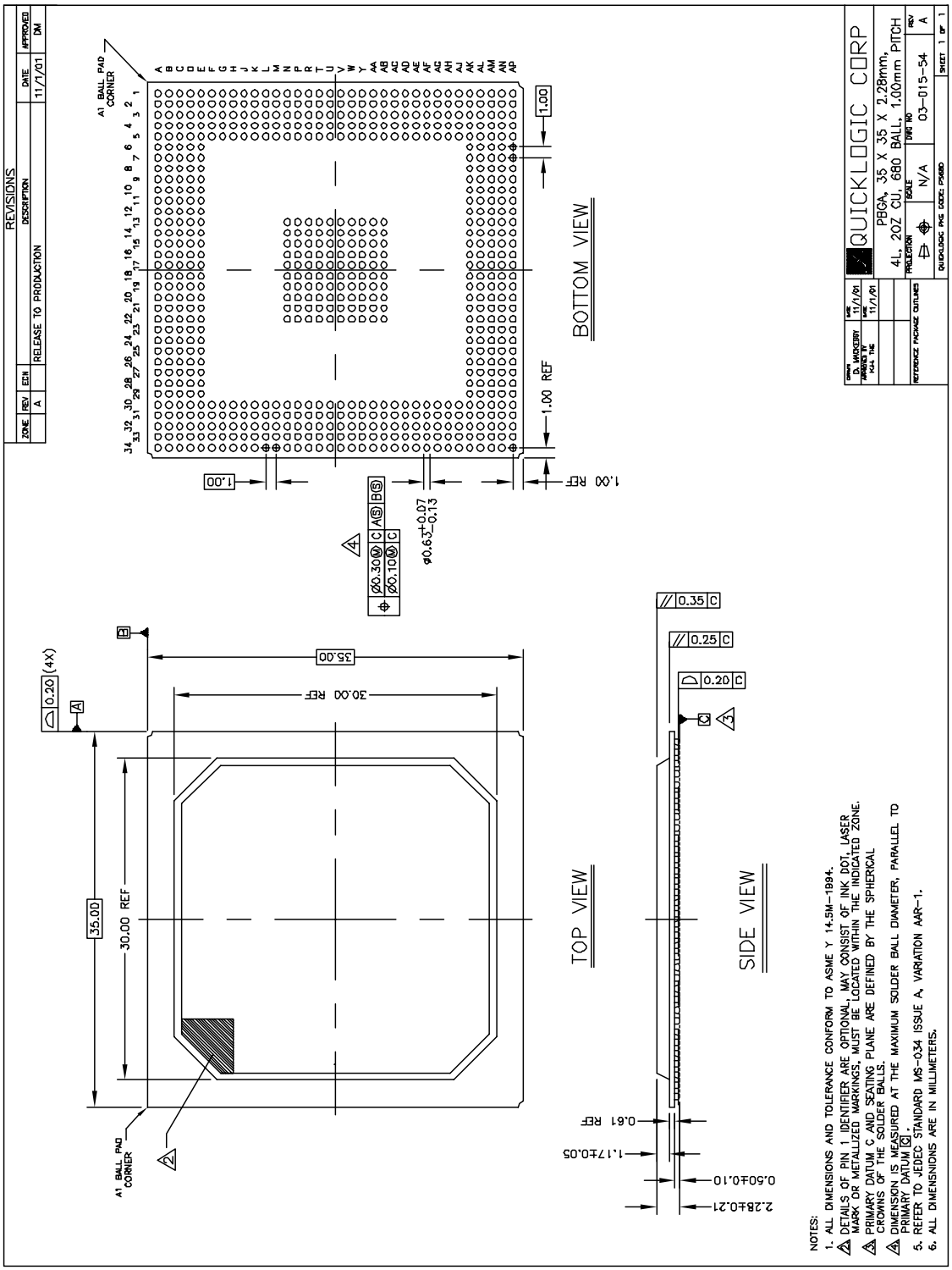
Table 45: 680 PBGA Pinout Table

680 PBGA	Function	680 PBGA	Function	680 PBGA	Function	680 PBGA	Function	680 PBGA	Function	680 PBGA	Function
V22	VCCIO	AA17	GND	AE5	M2_RXDV	AK10	PCI_AD<17>	AL31	GND	AN18	CPU_EXTINT_n<6>
V30	I/O<A>	AA18	GND	AE30	ADDR<2>	AK11	PCI_FRAME_n	AL32	ADDR<23>	AN19	CPU_EXTINT_n<2>
V31	I/O<A>	AA19	GND	AE31	OEN_n	AK12	PCI_PAR	AL33	ADDR<20>	AN20	U1_RTS_n
V32	I/O<A>	AA20	GND	AE32	BLS_n<0>	AK13	VCC	AL34	ADDR<15>	AN21	U1_TXD _SIROUT_n
V33	I/O<A>	AA21	GND	AE33	CS_n<6>	AK14	VCCIO	AM1	M1_RXDV	AN22	NC
V34	GND	AA22	VCCIO	AE34	CS_n<3>	AK15	EJTAG_TRST	AM2	NC	AN23	SD_CAS_n
W1	I/O<D>	AA30	VCCIO	AF1	M2_TXEN	AK16	EJTAG_TDO	AM3	GND	AN24	NC
W2	I/O<D>	AA31	EJTAG_DEBUGM	AF2	M2_TXD<1>	AK17	CPU_EXTINT _n<1>	AM4	M1_RXD<1>	AN25	SD_DQM<1>
W3	I/O<D>	AA32	CLK<3>/PLLIN<1>	AF3	M2_RXER	AK18	CPU_ BIGENDIAN	AM5	TM_OVERFLOW	AN26	DATA<30>
W4	I/O<D>	AA33	I/O<A>	AF4	M2_RXD<2>	AK19	U1_RI_n	AM6	PCI_GNT_n	AN27	DATA<23>
W5	GNDPLL	AA34	I/O<A>	AF5	M2_RXD<0>	AK20	U1_DCD_n	AM7	PCI_AD<30>	AN28	DATA<21>
W13	VCCIO	AB1	TCK	AF30	ADDR<7>	AK21	VCCIO	AM8	PCI_AD<26>	AN29	DATA<17>
W14	GND	AB2	TDI	AF31	ADDR<4>	AK22	VCC	AM9	PCI_AD<21>	AN30	DATA<12>
W15	GND	AB3	GND	AF32	ADDR<0>	AK23	SD_CLKIN	AM10	PCI_C_BE _n<3>	AN31	DATA<8>
W16	GND	AB4	STM	AF33	BLS_n<1>	AK24	DATA<28>	AM11	PCI_DEVSEL_n	AN32	DATA<4>
W17	GND	AB5	VCC	AF34	WEN_n	AK25	DATA<24>	AM12	PCI_SERR_n	AN33	GND
W18	GND	AB13	GND	AG1	M2_MDC	AK26	DATA<16>	AM13	PCI_AD<15>	AN34	GND
W19	GND	AB14	VCCIO	AG2	M2_TXCLK	AK27	VCCIO	AM14	PCI_AD<11>	AP1	GND
W20	GND	AB15	VCCIO	AG3	M2_RXD<1>	AK28	VCC	AM15	PCI_AD<5>	AP2	GND
W21	GND	AB16	VCCIO	AG4	M2_CRS	AK29	DATA<2>	AM16	PCI_AD<1>	AP3	PCI_INTA_n
W22	VCCIO	AB17	VCCIO	AG5	VCCIO	AK30	DATA<1>	AM17	EJTAG_TDI	AP4	PCI_AD<29>
W30	I/O<A>	AB18	VCCIO	AG30	VCCIO	AK31	ADDR<22>	AM18	CPU_EXTINT _n<5>	AP5	PCI_AD<27>
W31	I/O<A>	AB19	VCCIO	AG31	ADDR<12>	AK32	ADDR<18>	AM19	CPU_EXTINT_n <0>	AP6	PCI_AD<24>
W32	I/O<A>	AB20	VCCIO	AG32	ADDR<5>	AK33	ADDR<14>	AM20	U1_DTR_n	AP7	PCI_AD<22>
W33	I/O<A>	AB21	VCCIO	AG33	ADDR<1>	AK34	ADDR<11>	AM21	NC	AP8	PCI_AD<18>
W34	I/O<A>	AB22	GND	AG34	BLS_n<3>	AL1	M1_TXD<0>	AM22	SD_CKE<0>	AP9	PCI_IRDY_n
Y1	I/O<D>	AB30	VCC	AH1	M2_RXD<3>	AL2	M1_RXCLK	AM23	NC	AP10	VCCIO
Y2	I/O<D>	AB31	CS_n<0>	AH2	M2_RXCLK	AL3	M1_CRS	AM24	SD_DQM<2>	AP11	PCI_C_BE_n<1>
Y3	I/O<D>	AB32	GND	AH3	M1_MDIO	AL4	GND	AM25	SD_CLKOUT	AP12	PCI_AD<13>
Y4	I/O<D>	AB33	TRSTB	AH4	M1_MDC	AL5	M1_RXD<2>	AM26	DATA<26>	AP13	PCI_AD<10>
Y5	PL_CLOCKIN	AB34	CLK<2>	AH5	VCC	AL6	PCI_RST_n	AM27	DATA<20>	AP14	PCI_AD<6>
Y13	VCCIO	AC1	VCC	AH30	VCC	AL7	M1_RXD<0>	AM28	DATA<15>	AP15	PCI_AD<2>
Y14	GND	AC2	TDO	AH31	ADDR<16>	AL8	PCI_IDSEL	AM29	DATA<13>	AP16	GND
Y15	GND	AC3	PL_CLKOUT	AH32	ADDR<9>	AL9	PCI_CLK	AM30	DATA<9>	AP17	GND
Y16	GND	AC4	PL_BYPASS	AH33	ADDR<6>	AL10	PCI_AD<20>	AM31	DATA<5>	AP18	GND
Y17	GND	AC5	BOOT<0>	AH34	ADDR<3>	AL11	PCI_C_BE _n<2>	AM32	GND	AP19	CPU_EXTINT_n<3>
Y18	GND	AC30	CS_n<5>	AJ1	M2_COL	AL12	PCI_STOP_n	AM33	DATA<0>	AP20	U2_TXD _SIROUT_n
Y19	GND	AC31	CS_n<2>	AJ2	M1_TXEN	AL13	PCI_AD<14>	AM34	ADDR<19>	AP21	U1_DSR_n
Y20	GND	AC32	NC	AJ3	M1_TXD<1>	AL14	PCI_AD<8>	AN1	GND	AP22	U1_RXD_SIRIN
Y21	GND	AC33	EJTAG_DINT	AJ4	M1_TXCLK	AL15	PCI_AD<7>	AN2	GND	AP23	SD_CKE<1>
Y22	VCCIO	AC34	VCC	AJ5	M1_COL	AL16	PCI_AD<3>	AN3	TM_ENABLE	AP24	SD_RAS_n
Y30	CLK<4>/DEDCLK/ PLLIN<0>	AD1	PL_LOCK	AJ30	ADDR<21>	AL17	EJTAG_TMS	AN4	PCI_REQ_n	AP25	SD_CS_n<0>
Y31	I/O<A>	AD2	PL_ENABLE	AJ31	ADDR<17>	AL18	CPU_EXTINT _n<4>	AN5	PCI_AD<31>	AP26	SD_DQM<0>
Y32	I/O<A>	AD3	PL_WARMRESET_n	AJ32	ADDR<13>	AL19	U2_RXD_SIRIN	AN6	PCI_AD<28>	AP27	DATA<29>
Y33	I/O<A>	AD4	M2_MDIO	AJ33	ADDR<10>	AL20	U1_CTS_n	AN7	PCI_AD<25>	AP28	DATA<27>
Y34	I/O<A>	AD5	M2_TXD<2>	AJ34	ADDR<8>	AL21	SD_WE_n	AN8	PCI_AD<19>	AP29	DATA<22>
AA1	I/O<D>	AD30	BLS_n<2>	AK1	M1_TXD<3>	AL22	SD_CS_n<1>	AN9	PCI_AD<16>	AP30	DATA<18>
AA2	CLK<0>	AD31	CS_n<7>	AK2	M1_TXD<2>	AL23	SD_DQM<3>	AN10	PCI_TRDY_n	AP31	DATA<14>
AA3	CLK<1>	AD32	CS_n<4>	AK3	M1_RXER	AL24	DATA<31>	AN11	PCI_PERR_n	AP32	DATA<10>
AA4	VCCIO	AD33	CS_n<1>	AK4	M1_RXD<3>	AL25	DATA<25>	AN12	PCI_C_BE_n<0>	AP33	GND
AA5	VCCPLL	AD34	NC	AK5	NC	AL26	DATA<19>	AN13	PCI_AD<12>	AP34	GND
AA13	VCCIO	AE1	PL_RESET_n	AK6	GND	AL27	DATA<11>	AN14	PCI_AD<9>		
AA14	GND	AE2	BOOT<1>	AK7	VCC	AL28	DATA<6>	AN15	PCI_AD<4>		
AA15	GND	AE3	M2_TXD<3>	AK8	VCCIO	AL29	DATA<7>	AN16	PCI_AD<0>		
AA16	GND	AE4	M2_TXD<0>	AK9	PCI_AD<23>	AL30	DATA<3>	AN17	EJTAG_TCK		

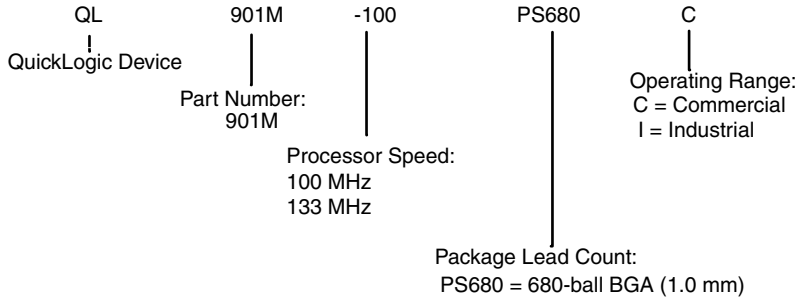
(Sheet 2 of 2)

680 BGA Pinout Drawing

Figure 45: 680 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	December 2001	First release
B	December 2001	Donna Chin - updated PLL information
C	July 2003	Judd Heape and Kathleen Murchek Modifications to all sections including part number information.
D	July 2003	Judd Heape and Kathleen Murchek Modifications to all sections including DC characteristics, signal descriptions, device pinout, and Fabric PLL information.
E	September 2003	Judd Heape and Jesse Blount Changed T_{jmax} number from 150°C to 125°C in Package Thermal Characteristics section. Removed Preliminary.
F	December 2003	Judd Heape and Kathleen Murchek Added an AHB Master and AHB Slave Interface Signals section with an ahb_hready_in row to ASSP to Fabric Port Descriptions table.

Copyright and Trademark Information