QL5632 Enhanced QuickPCI Device Data Sheet

33 MHz/32-bit PCI Master/Target with Embedded Programmable Logic and Dual Port SRAM

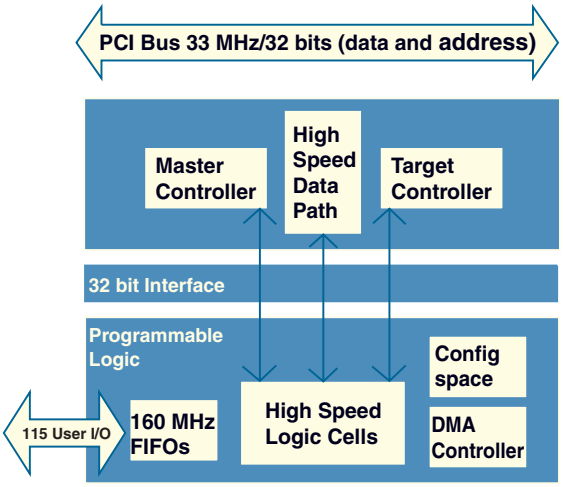


Figure 1: QL5632 Diagram

Device Highlights

High Performance PCI Controller

- 32-bit / 33 MHz PCI Master/Target
- Zero-wait state PCI Master provides 132 MBps transfer rates
- Zero-wait-state PCI Target Write/One-wait-state PCI Target Read interface
- Supports all PCI commands, including configuration and MWI
- Supports fully-customizable byte enable for master channels
- Target interface supports retry, disconnect with/without data transfer, and target abort
- Fully programmable back-end interface
- Independent PCI bus (33 MHz) and local bus (up to 160 MHz) clocks
- Fully customizable PCI Configuration Space
- Configurable FIFOs with depths up to 256 words
- Reference design with driver code (Win 95/98/Win 2000/NT4.0) available
- PCI v2.3 compliant
- Supports Type 0 Configuration Cycles in Target mode
- 3.3 V PCI signaling
- 2.5 V Supply Voltage
- 280-pin PBGA
- 208-pin PQTP
- Supports Extendable PCI functionality
- Unlimited/Continuous Burst Transfers supported

Extendable PCI Functionality

- Support for PCI host-bridge function
- Support for Configuration Space from 0×40 to $0 \times 3FF$
- Multi-Function, Expanded Capabilities, & Expansion ROM capable
- PCI v2.3 Power Management Spec compatible
- PCI v2.3 Vital Product Data (VPD) configuration support
- Programmable Interrupt Generator
- I_2O support with local processor
- Mailbox register support

Flexible Programmable Logic

- 772 Logic Cells
- 41,472 RAM bits
- Up to 115 I/O pins
- All back-end interface and glue-logic can be implemented on chip
- Six 32-bit busses interface between the PCI Controller and the Programmable Logic
- Eighteen 2,304 bit Dual Port High Performance SRAM Blocks
- 1,889 flip-flops available

Architecture Overview

The QL5632 device in the QuickLogic® QuickPCI ESP (Embedded Standard Product) family provides a complete and customizable PCI interface solution combined with programmable logic. This device eliminates any need for the designer to worry about PCI bus compliance, yet allows for the maximum 32-bit PCI bus bandwidth (132 MBps).

The programmable logic portion of the device contains 772 QuickLogic Logic Cells and 18 QuickLogic Dual-Port RAM Blocks. These configurable RAM blocks can be configured in many width/depth combinations. They can also be combined with logic cells to form FIFOs, or be initialized via Serial EEPROM on power-up and used as ROMs.

The QL5632 device meets PCI 2.3 electrical and timing specifications and has been fully hardware-tested. This device also supports the Win'98 and PC'98 standards. The QL5632 device features 2.5 V operation with multi-volt compatible I/Os. The device can easily operate in 3 V embedded systems and is fully compatible with 3.3 V applications.

PCI Controller

The PCI Controller is a 32-bit/33 MHz PCI 2.3 Compliant Master/Target Controller capable of infinite length Master Write and Read transactions at zero wait states (132 MBps).

The Master will never insert wait states during transfers, so data is supplied or received by FIFOs that can be configured in the programmable region of the device. The Master is capable of initiating any type of PCI commands, including configuration cycles and Memory Write and Invalidate (MWI). This enables the QL5632 device to act as a PCI host. The Master Controller will most often be operated by a DMA Controller in the programmable region of the device. DMA Controller reference design is available and will be included in the Quick*Works*® 9.3 design software.

The Target interface offers full PCI Configuration Space and flexible target addressing. It supports zero-wait-state target Write and one-wait-state target Read operations. It also supports retry, disconnect with/without data transfer, and target abort requested by the back end. Any number of 32-bit BARs may be configured as either memory or I/O space. All required and optional PCI 2.3 Configuration Space registers can be implemented within the programmable region of the device. A reference design of a Target Configuration and Addressing module is available and will be included in the Quick*Works* 9.3 design software.

The interface ports are divided into a set of ports for master transactions and a set for target transactions. The Master DMA controller and Target Configuration Space and Address Decoding are done in the programmable logic region of the device. These functions are not timing critical, so leaving these elements in the programmable region allows the greatest degree of flexibility to the designer. Reference DMA controller, Configuration Space, and Address Decoding blocks are readily available so that the design cycle can be minimized.

Configuration Space and Address Decode

The configuration space is completely customizable in the programmable region of the device.

PCI address and command decoding is performed by logic in the programmable section of the device. This allows support for any size of memory or I/O space for back end logic. It also allows the user to implement any subset of the PCI commands supported by the QL5632. In the reference design, QuickLogic provides a reference Address Register/Counter and Command Decode block.

DMA Master Target Controller

The customizable DMA controller included with the QuickWorks design software contains the following features:

- Configurable DMA count size for Reads and Writes (up to 30 bits)
- Configurable DMA burst size for PCI (including unlimited/continuous burst)
- Customizable PCI command to use by core
- Customizable Byte Enable signal
- Programmable Arbitration between DMA Read & Write transactions
- DMA Registers may be mapped to any area of Target Memory Space, including:
  - Read Address (32-bit register)
  - Write Address (32-bit register)
  - Read Length (16-bit register) / Write Length (16-bit register)
  - Control and Status (32-bit register, includes 8 bit Burst Length)
- DMA Registers are available to the local design or the PCI bus
- Programmable Interrupt Control to signal end of transfer or other event

Configurable FIFOs

FIFOs may be created with the RAM/FIFO wizard in the QuickWorks tools. **Figure 2** shows the graphical interface used to create these FIFOs. FIFOs may be designed up to 1,889 words deep. The 18 RAM cells available in the QL5632 allow for up to:

- 9 FIFOs at 128 words deep (36 wide)
- 4 FIFOs at 256 words deep (36 wide)
- 2 FIFOs at 512 words deep (36 wide)
- 1 FIFO at 1,024 words deep (36 wide)

Figure 2: Graphical Interface to create FIFO

PCI Interface Symbol

Figure 3 shows the interface symbol you have to use in your schematic design in order to attach the local interface programmable logic design to the PCI core. If you are designing with a top-level Verilog or VHDL file you must use a structural instantiation of this PCI32_25um block instead of a graphical symbol.

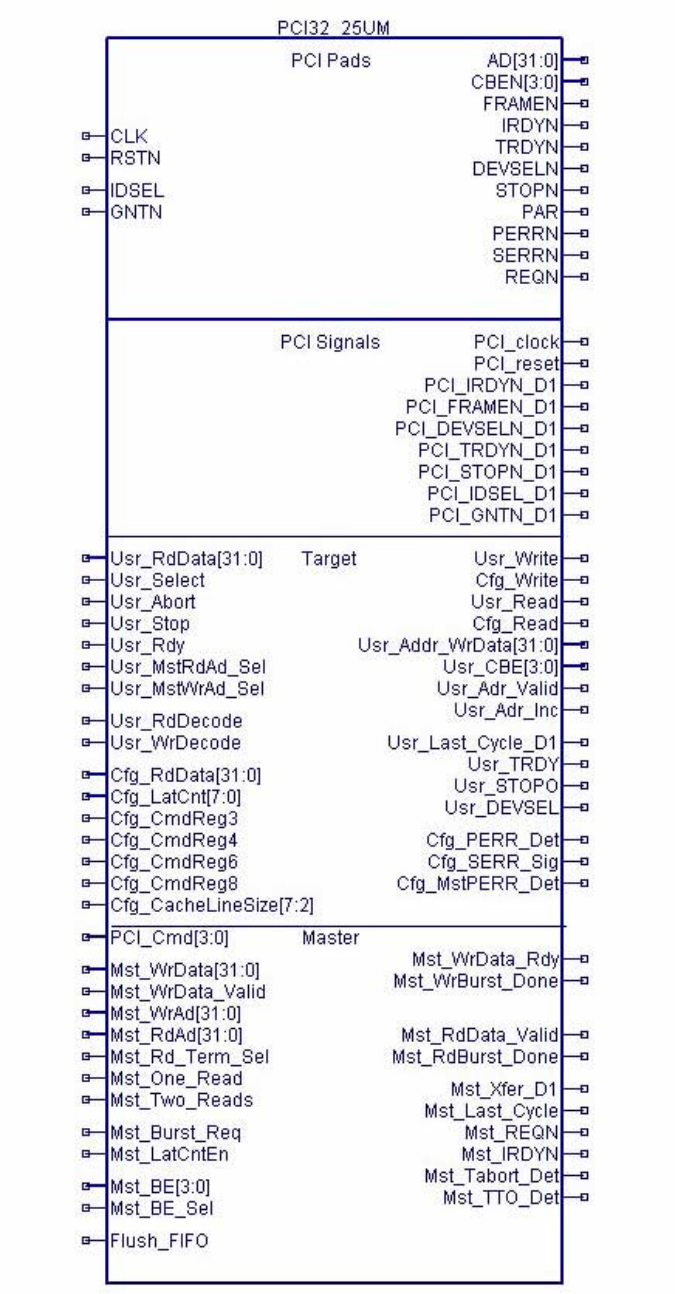


Figure 3: PCI Interface Symbol

PCI Master Interface

The internal signals used to interface with the PCI controller in the QL5632 are listed in **Table 1** along with a description of each signal. The direction of the signal indicates if the signal is an input provided by the local interface (I) or an output provided by the PCI controller (O).

NOTE: Signals that end with the character 'N' should be considered active-low (for example, Mst_IRDYN).

Table 1: PCI Master Interface

Signal		Description
PCI_cmd[3:0]	I	**PCI command to be used for the master transaction** This signal must remain unchanged throughout the period when Mst_Burst_Req is active. PCI commands considered as reads include Interrupt Acknowledge, I/O Read, Memory Read, Configuration Read, Memory Read Multiple, Memory Read Line. PCI commands considered as writes include Special Cycle, I/O Write, Memory Write, Configuration Write, Memory Write and Invalidate. Users should make sure that only valid PCI commands are supplied.
mst_burst_req	I	**Request use of the PCI bus** When this signal is active, the core requests the PCI bus and then generates a master transaction. This signal should be held active until all requested data are transferred on the PCI bus and deactivated in the 2nd clock cycle following the last data transfer on PCI (to avoid being considered as requesting a new transaction).
mst_wrAd[31:0]	I	**Address for master DMA writes** This address must be treated as valid from the beginning of a DMA Write until the DMA Write operation is complete. It should be incremented by 4 bytes each time data is transferred on the PCI bus.
mst_rdAd[31:0]	I	**Address for master DMA reads** This address must be treated as valid from the beginning of a DMA read until the DMA Read operation is complete. It should be incremented by 4 bytes each time data is transferred on the PCI bus.
Mst_WrData[31:0]	I	**Data for master DMA Writes (to PCI bus)**
Mst_BE[3:0]	I	**Byte enables for master DMA Reads and writes** Active-low.
Mst_WrData_Valid	I	**Data and byte enable valid on Mst_WrData[31:0] (for master Write only) and Mst_BE[3:0] (for both master Read and Write)**
Mst_WrData_Rdy	O	**Data receive acknowledge for Mst_WrData[31:0] (for master Write only) and Mst_BE[3:0] (for both)** This serves as the PUSH control for the internal FIFO and the POP control for the external FIFO (in FPGA region) which provides data and byte enables to the PCI32 core.
Mst_BE_Sel	I	**Byte enable select for master transactions** When low, Mst_BE[3:0] should remain constant throughout the entire transfer (when Mst_Burst_Req is active) and it is used for every data phase of the master transaction. When high, Mst_BE[3:0] pushed into internal FIFO (along with data in case of master Write) is used. Should be held constant throughout the transaction.
Mst_WrBurst_Done	O	**Master Write transaction is completed** Active for only one clock cycle.
Mst_Rd_Term_Sel	I	**Master Read termination mode select when Mst_BE_Sel is high** When both Mst_BE_Sel and Mst_Rd_Term_Sel are high, master Read termination happens when the internal FIFO is empty, and Mst_Two_Reads and Mst_One_Read are ignored. When either signal is low, Mst_Two_Reads and Mst_One_Read are used to signal end of master Read. Should be held constant throughout the transaction.
Mst_One_Read	I	This signals to the PCI32 core that only one data transfer remains to be read in the burst Read.
Mst_Two_Reads	I	**Two data transfers remain to be read in the burst Read** It is not used for single-data-phase master read transactions.

Table 1: PCI Master Interface

Mst_RdData_Valid	O	**Master Read data valid on Usr_Addr_WrData[31:0]** This serves as the PUSH control for the external FIFO (in FPGA region) that receives data from the PCI32 core.
Mst_RdBurst_Done	O	**Master Read transaction is completed** Active for only one clock cycle.
Flush_FIFO	I	**Internal FIFO flush** FIFO flushed immediately after it is active (synchronized with PCI clock).
Mst_LatCntEn	I	**Enable Latency Counter** Set to 0 to ignore the Latency Timer in the PCI configuration space (offset 0Ch). For full PCI compliance, this port should be always set to 1.
Mst_Xfer_D1	O	**Data was transferred on the previous PCI clock** Useful for updating DMA transfer counts on DMA Read operations
Signal		Description
Mst_Last_Cycle	O	**Active during the last data transfer of a master transaction**
Mst_REQN	O	**Copy of the PCI REQN signal generated by QL5632 as PCI master** Not usually used in the back-end design.
Mst_IRDYN	O	**Copy of the PCI IRDYN signal generated by QL5632 as PCI master** Valid only when QL5 × 33 is the PCI master. Kept low otherwise. Not usually used in the back-end design.
Mst_Tabort_Det	O	**Target abort detected during master transaction** This is normally an error condition to be handled in the DMA controller.
Mst_TTO_Det	O	**Target timeout detected (no response from target)** This is normally an error condition to be handled in the DMA controller.

PCI Target Interface

Table 2: PCI Target Interface

Signal		Description
Usr_Addr_WrData[31:0]	O	**Target address, and target Write data** During all target accesses, the address is presented on Usr_Addr_WrData[31:0] at the same time Usr_Adr_Valid is active. During target Write transactions, this port also presents valid Write data to the PCI configuration space or user logic when Usr_Adr_Inc is active.
Usr_CBE[3:0]	O	**PCI command and byte enables** During target accesses, the PCI command is presented on Usr_CBE[3:0] at the same time Usr_Adr_Valid is active. This port also presents active-low byte enables to the PCI configuration space or user logic.
Usr_Adr_Valid	O	Indicates the beginning of a PCI transaction, and that a target address is valid on Usr_Addr_WrData[31:0] and the PCI command is valid on Usr_CBE[3:0]. When this signal is active, the target address must be latched and decoded to determine if this address belongs to the device's memory or I/O space. Also, the PCI command must be decoded to determine the type of PCI transaction. On subsequent clocks of a target access, this signal is low, indicating that address is NOT present on Usr_Addr_WrData[31:0].
Usr_Adr_Inc	O	Indicates that the target address should be incremented, because the previous data transfer has been completed. During burst target accesses, the target address is only presented to the back-end logic at the beginning of the transaction (when Usr_Adr_Valid is active), and must therefore be latched and incremented by 4 for subsequent data transfers. Note that during target Write transactions, Usr_Adr_Inc indicates valid data on Usr_Addr_WrData[31:0] that must be accepted by the backend logic (regardless of the state of Usr_Rdy). During Read transactions, Usr_Adr_Inc signals to the backend that the PCI core has presented the read data on the PCI bus (TRDYN asserted).
Usr_RdDecode	I	This signal should be the combinatorial decode of the "user read" command from Usr_CBE[3:0]. This command may be mapped from any of the PCI "read" commands, such as Memory Read, Memory Read Line, Memory Read Multiple, I/O Read, etc. It is internally gated with Usr_Adr_Valid.
Usr_WrDecode	I	This signal should be the combinatorial decode of the "user write" command from Usr_CBE[3:0]. This command may be mapped from any of the PCI "write" commands, such as Memory Write or I/O Write. It is internally gated with Usr_Adr_Valid.
Usr_Select	I	This signal should be driven active when the address on Usr_Addr_WrData[31:0] has been decoded and determined to be within the address space of the device. Usr_Addr_WrData[31:0] must be compared to each of the valid Base Address Registers in the PCI configuration space. Also, this signal must be gated by the Memory Access Enable or I/O Access Enable registers in the PCI configuration space (Command Register bits 1 or 0 at offset 04h). Internally gated with Usr_Adr_Valid.
Usr_Write	O	This signal is active throughout a "user write" transaction, which has been decoded by Usr_WrDecode at the beginning of the transaction. The Write strobe for individual DWORDs of data (on Usr_Addr_WrData[31:0]) during a user Write transaction should be generated by logically ANDing this signal with Usr_Adr_Inc.
Cfg_Write	O	This signal is active throughout a "configuration write" transaction. The Write strobe for individual DWORDs of data (on Usr_Addr_WrData[31:0]) during a configuration Write transaction should be generated by logically ANDing this signal with Usr_Adr_Inc.
Usr_Read	O	This signal is active throughout a "user read" transaction, which has been decoded by Usr_RdDecode at the beginning of the transaction.
Cfg_Read	O	This signal is active throughout a "configuration read" transaction.
Cfg_RdData[31:0]	I	Data from the PCI configuration registers, required to be presented during PCI configuration reads.
Usr_RdData[31:0]	I	Data from the back-end user logic, required to be presented during PCI user reads.

Table 2: PCI Target Interface

Signal		Description
Cfg_CmdReg3	I	Bits 3 from the Command Register in the PCI configuration space (offset 04h). Enable Special Cycle monitoring. If high, the core reports data parity error in Special Cycles through SERRN if Cfg_CmdReg8 is active.
Cfg_CmdReg4	I	Bits 4 from the Command Register in the PCI configuration space (offset 04h). Memory Write and Invalidate (MWI) Enable. If high, the core generates MWI transactions as requested by the backend. Otherwise it uses Memory Write instead even if MWI is requested.
Cfg_CmdReg6	I	Bits 6 from the Command Register in the PCI configuration space (offset 04h). Parity Error Response. If high, the core uses PERRN to report data parity errors. Otherwise it never drives it.
Cfg_CmdReg8	I	Bits 8 from the Command Register in the PCI configuration space (offset 04h). SERRN Enable. If high, the cores uses SERRN to report address parity errors if Cfg_CmdReg6 is high.
Cfg_LatCnt[7:0]	I	8-bit value of the Latency Timer in the PCI configuration space (offset 0Ch).
Usr_MstRdAd_Sel	I	Used when a target read operation should return the value set on the Mst_RdAd[31:0] pins. This select pin saves on logic which would otherwise need to be used to multiplex Mst_RdAd[31:0] into the Usr_RdData[31:0] bus. When this signal is asserted, the data on Usr_RdData[31:0] is ignored.
Usr_MstWrAd_Sel	I	Used when a target read operation should return the value set on the Mst_WrAd[31:0] pins. This select pin saves on logic which would otherwise need to be used to multiplex Mst_WrAd[31:0] into the Usr_RdData[31:0] bus. When this signal is asserted, the data on Usr_RdData[31:0] is ignored.
Cfg_PERR_Det	O	Parity error detected on the PCI bus. When this signal is active, bit 15 of the Status Register must be set in the PCI configuration space (offset 04h).
Cfg_SERR_Sig	O	System error asserted on the PCI bus. When this signal is active, the Signalled System Error bit, bit 14 of the Status Register, must be set in the PCI configuration space (offset 04h).
Cfg_MstPERR_Det	O	Data parity error detected on the PCI bus by the master. When this signal is active, bit 8 of the Status Register must be set in the PCI configuration space (offset 04h).
Usr_TRDY	O	Inverted copy of the TRDYN signal as driven by the PCI target interface. Valid only within a target access.
Usr_STOPO	O	Inverted copy of the STOPN signal as driven by the PCI target interface. Valid only within a target access.
Usr_DEVSEL	O	Inverted copy of the DEVSELN signal as driven by the PCI target interface. Valid only within a target access.
Usr_Last_Cycle_D1	O	Active one clock cycle after the last data phase (may not with data transfer) occurs on PCI and inactive one clock cycle afterwards.
Usr_Rdy	I	Used to delay (add wait states to) a target PCI transaction when the backend needs additional time to provide data (read) or accept data (write). Subject to PCI latency restrictions.
Usr_Stop	I	Used to prematurely stop a PCI target access on the next PCI clock.
Usr_Abort	I	Used to signal Target Abort on PCI when the backend has fatal error and is unable to complete a transaction. Rarely used.

PCI Internal Signals

Table 3: PCI Internal Signal

Signal		Description
PCI_clock	O	PCI clock.
PCI_reset	O	PCI reset signal.
PCI_IRDYN_D1	O	Copy of the IRDYN signal from the PCI bus, delayed by one clock.
PCI_FRAMEN_D1	O	Copy of the FRAMEN signal from the PCI bus, delayed by one clock.
PCI_DEVSELN_D1	O	Copy of the DEVSELN signal from the PCI bus, delayed by one clock.
PCI_TRDYN_D1	O	Copy of the TRDYN signal from the PCI bus, delayed by one clock.
PCI_STOPN_D1	O	Copy of the STOPN signal from the PCI bus, delayed by one clock.
PCI_IDSEL_D1	O	Copy of the IDSEL signal from the PCI bus, delayed by one clock.

RAM Module Features

The QL5632 device has eighteen 2,304-bit RAM modules, for a total of 41,472 RAM bits. Using two "mode" pins, designers can configure each module into 128 × 18, 256 × 9, 512 × 4, or 1024 × 2 blocks (see **Figure 1**). The blocks are also easily cascadable to increase their effective width or depth.

The RAM modules are "dual-ported" 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. This approach allows up to 1,024-deep configurations as large as 44 bits wide in the QL5632 device.

A similar technique can be used to create depths greater than 1,024 words. In this case, address signals higher than the eighth bit are encoded onto the write enable (WE) input for Write operations. The Read data outputs are multiplexed together using encoded higher Read address bits for the multiplexer SELECT signals.

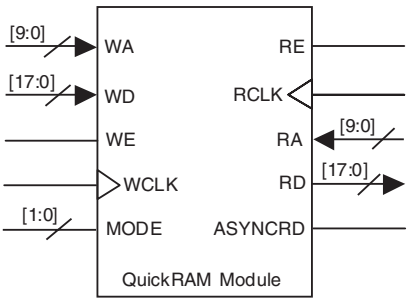


Figure 4: RAM Module

Embedded Computational Unit (ECU)

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 QL5632 architecture allows for functionality above and beyond that achievable using programmable logic devices. By embedding a dynamically reconfigurable computational unit, the QL5632 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 5**.

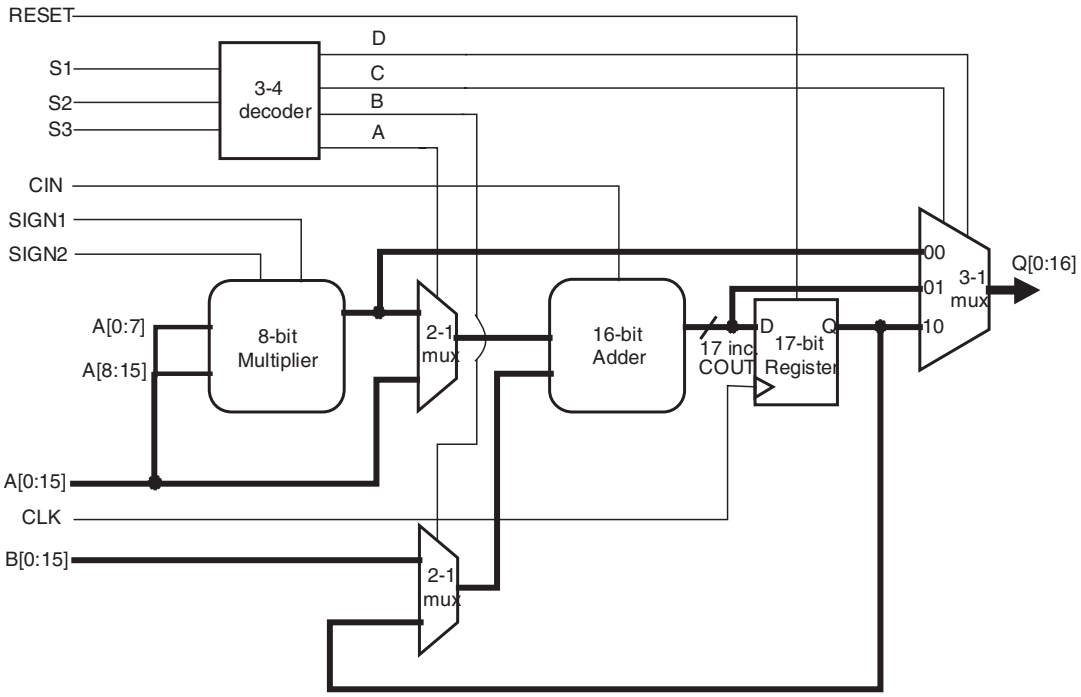


Figure 5: ECU Block Diagram

The 10 QL5632 ECU blocks are placed next to the SRAM circuitry for efficient memory/instruction fetch and addressing for DSP algorithmic implementations.

Ten 8-bit MAC functions can be implemented per cycle for a total of ~1 billion MACs/s when clocked at 98 MHz. Additional multiply-accumulate functions can be implemented in the programmable logic.

The modes for the ECU block are dynamically re-programmable through the programmable logic.

Table 4: ECU Mode Select Criteria

Instruction			Operation	ECU Performance[a], -B WCC		
S1	S2	S3		t_{PD}	t_{SU}	t_{CO}
0	0	0	Multiply	7.0 ns max		
0	0	1	Multiply-Add	9.4 ns max		
0	1	0	Accumulate[b]		4.1 ns min	1.2 ns max
0	1	1	Add	3.3 max		
1	0	0	Multiply (registered)[c]		10.2 ns min	1.2 ns max
1	0	1	Multiply- Add (registered)		10.2 ns min	1.2 ns max
1	1	0	Multiply - Accumulate		10.2 ns min	1.2 ns max
1	1	1	Add (registered)		4.1 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 224 MHz.
c. B [15:0] set to zero.

NOTE: Timing numbers in **Table 4** represent -B Worst Case Commercial conditions.

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. These PLLs also have the ability to 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 Phase Locked Loop subtracting the clock tree delay through the feedback path, effectively making the clock tree delay zero.

Figure 6 illustrates a typical QuickLogic ESP PLL.

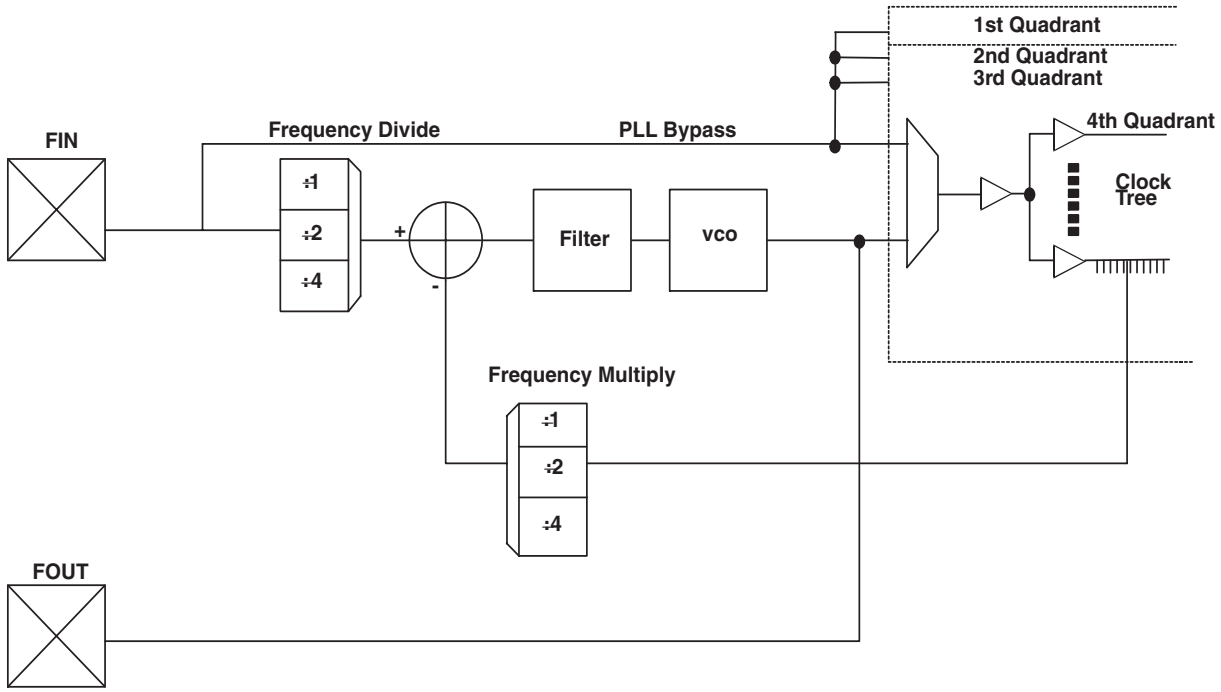


Figure 6: PLL Block Diagram

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 6**) 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 6**). 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 emerging from the output pad (the output signal PLLPAD_OUT is explained in **Table 6**). 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.

Most QuickLogic products contain four PLLs, one to be used in each quadrant. The PLL presented in **Figure 6** controls the clock tree in the fourth Quadrant of its ESP. QuickLogic PLLs compensate for the additional delay created by the clock tree itself, as previously noted, by subtracting the clock tree delay through the feedback path.

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

PLL Modes of Operation

QuickLogic PLLs have eight modes of operation, based on the input frequency and desired output frequency—**Table 5** indicates the features of each mode.

NOTE: "HF" stands for "high frequency" and "LF" stands for "low frequency."

Table 5: PLL Mode Frequencies

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

NOTE: The input frequency can range from 16 MHz to 300 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.

PLL Signals

Table 6 summarizes the key signals in QuickLogic's PLLs.

Table 6: PLL Signals

Signal Name	Description
PLLCLK_IN	**Input clock signal**
PLL_RESET	**Active High Reset** If PLL_RESET is asserted, then CLKNET_OUT and PLLPAD_OUT 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.
PLLPAD_OUT	**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.

NOTE: Because PLLCLK_IN and PLL_RESET signals have INPAD, and PLLPAD_OUT has OUTPAD, you do not have to add additional pads to your design.

JTAG Support

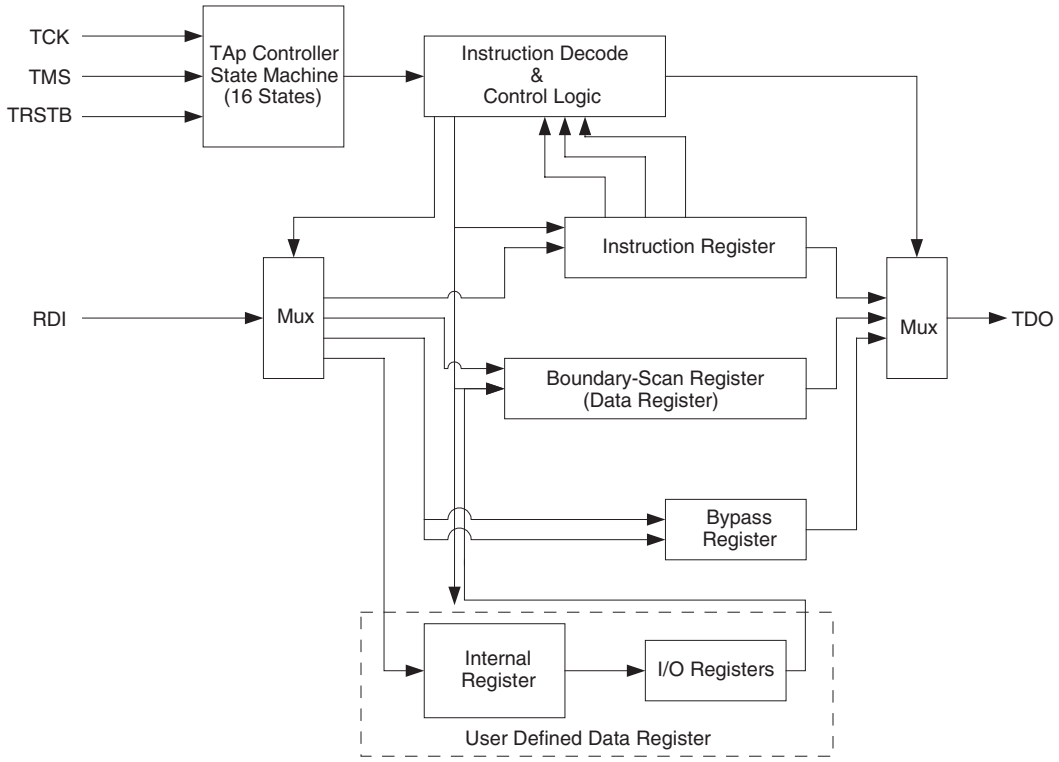


Figure 7: JTAG Block Diagram

The Joint Test Access Group (JTAG) pins support the IEEE Standard 1149.1a to provide boundary scan capability for the QL5632 device. Six pins are dedicated to JTAG and programming functions on each QL5632 device; these pins are unavailable for general design input and output signals. TDI, TDO, TCK, TMS, and TRSTB are JTAG pins. The sixth pin, STM, is used only for programming.

Microprocessors and Application Specific Integrated Circuits (ASICs) pose many design challenges. One of these challenges concerns the accessibility of test points. JTAG was 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); these 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 JTAG 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 TAP's Test Data In (TDI) and Test Data Out (TDO) pins. Boundary scan cells are preloaded with test patterns (via 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 via 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 connects the TDI and TDO pins, allowing serial data to be transferred through a device without affecting the operation of the device.

Development Tool Support

Software support for the QL5632 device is available through the Quick*Works* development package. This turnkey PC-based Quick*Works* package, shown in **Figure 8**, provides a complete ESP software solution with design entry, logic synthesis, place and route, and simulation. Quick*Works* includes VHDL, Verilog, schematic, and mixed-mode entry with fast and efficient logic synthesis provided by the integrated Synplicity Synplify Lite™ tool which is specially tuned to take advantage of the QL5632 architecture. Quick*Works* also provides functional and timing simulation for guaranteed timing and source-level debugging.

The UNIX-based Quick*Tools* package is a subset of Quick*Works* and provides a solution for designers who use schematic-only design flow third-party tools for design entry, synthesis, or simulation. Quick*Tools* Reads EDIF netlists and provides support for all QuickLogic devices. Quick*Tools* also supports a wide range of third-party modeling and simulation tools.

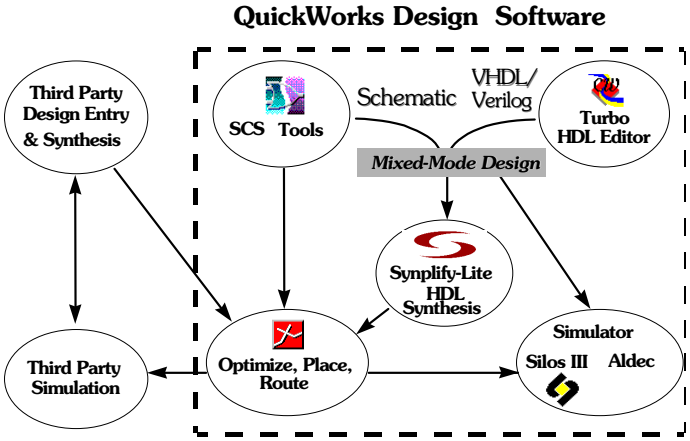


Figure 8: QuickWorks Tool Suite

Electrical Specifications

Table 7: Absolute Maximum Ratings

V_{CC} Voltage	-0.5 V to 3.6 V	DC Input Current	±20 mA
V_{CCIO} Voltage	-0.5 V to 4.6 V	ESD Pad Protection	±2000 V
V_{REF} Voltage	2.7 V	Leaded Package Storage Temperature	-65°C to +150°C
Input Voltage	-0.5 V to V_{CCIO} +0.5 V	Laminate Package (BGA) Storage Temperature	-55°C to +125°C

Table 8: Operating Range

Symbol	Parameter		Military		Industrial		Commercial		Unit
			Min	Max	Min	Max	Min	Max	
V_{CC}	Supply Voltage		2.3	2.7	2.3	2.7	2.3	2.7	V
V_{CCIO}	I/O Input Tolerance Voltage		2.3	3.6	2.3	3.6	2.3	3.6	V
TA	Ambient Temperature		-55		-40	85	0	70	°C
TC	Case Temperature		-	125	-	-	-	-	°C
K	Delay Factor	-B Speed Grade	0.42	1.35	0.43	1.26	0.46	1.23	n/a

DC Characteristics

Table 9: DC Characteristics

Symbol	Parameter	Conditions	Temperature		Unit
			Min	Max	
I_I	I or I/O Input Leakage Current	$V_I = V_{CCIO}$ or GND	-10	10	µA
I_{OZ}	3-State Output Leakage Current	$V_I = V_{CCIO}$ or GND	-10	10	µA
C_I	Input Capacitance [a]	-	-	8	pF
I_{OS}	Output Short Circuit Current [b]	V_o = GND $V_o = V_{CC}$	-15 40	-180 210	mA mA
I_{CC}	D.C. Supply Current	$V_I, V_o = V_{CCIO}$ or GND	0.50 (typ)	2	mA
I_{CCIO}	D.C. Supply Current on V_{CCIO}	-	0	2	mA
I_{CCIO}(DIF)	D.C. Supply Current on V_{CCIO} for Differential I/O	-	-	-	mA
I_{REF}	D.C. Supply Current on V_{REF}	-	-10	10	µA
I_{PD}	Pad Pull-down (programmable)	V_{CCIO} = 3.6 V	-	150	µA

a. Capacitance is sample tested only. Clock pins are 12 pF maximum.
b. Only one output at a time. Duration should not exceed 30 seconds.

Table 10: DC Input and Output Levels

	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	2.0	V_{CCIO} + 0.3	0.4	2.4	2.0	-2.0
LVCMOS 2	n/a	n/a	-0.3	0.7	1.7	V_{CCIO} + 0.3	0.7	1.7	2.0	-2.0
GTL+	0.88	1.12	-0.3	INREF - 0.2	INREF + 0.2	V_{CCIO} + 0.3	0.6	n/a	40	n/a
PCI	n/a	n/a	-0.3	0.3 x V_{CCIO}	0.5 x V_{CCIO}	V_{CCIO} + 0.5	0.1 x V_{CCIO}	0.9 x V_{CCIO}	1.5	-0.5
SSTL2	1.15	1.35	-0.3	INREF - 0.18	INREF + 0.18	V_{CCIO} + 0.3	0.74	1.76	7.6	-7.6
SSTL3	1.3	1.7	-0.3	INREF - 0.2	INREF + 0.2	V_{CCIO} + 0.3	1.10	1.90	8	-8

NOTE: The data provided in **Table 10** are JEDEC and PCI Specifications—QuickLogic devices either meet or exceed these requirements. For data specific to QuickLogic I/Os, see **Table 7** through **Table 20** and **Figure 9** through **Figure 21**.

NOTE: All CLK and IOCTRL pins are clamped to the V_{cc} rail, not the V_{ccio}. Therefore, these pins can only be driven up to V_{cc} + 0.3 V.

AC Characteristics

The AC characteristics are calculated at 2.5 V, TA = 25°C (K = 0.74). To calculate delays, multiply the appropriate K factor in **Table 8** by the numbers presented in **Table 11** through **Table 18.**

Table 11: Logic Cells

Symbol Logic Cells	Parameter	Value (ns)	
		Min	Max
t_{PD}	**Combinatorial Delay of the longest path**: time taken by the combinatorial circuit to output	-	0.257
t_{SU}	**Setup time**: time the synchronous input of the flip flop must be stable before the active clock edge	0.22	-
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.	-	0.255
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)	-	0.18
t_{RESET}	**Reset Delay**: time between when the flip flop is "reset" (low) and when the output is consequently "reset" (low)	-	0.09
t_{SW}	**Set Width**: time that the SET signal remains high/low	0.3	-
t_{RW}	**Reset Width**: time that the RESET signal remains high/low	0.3	-

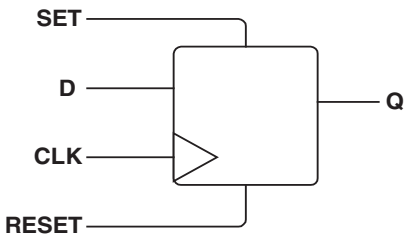


Figure 9: Logic Cell

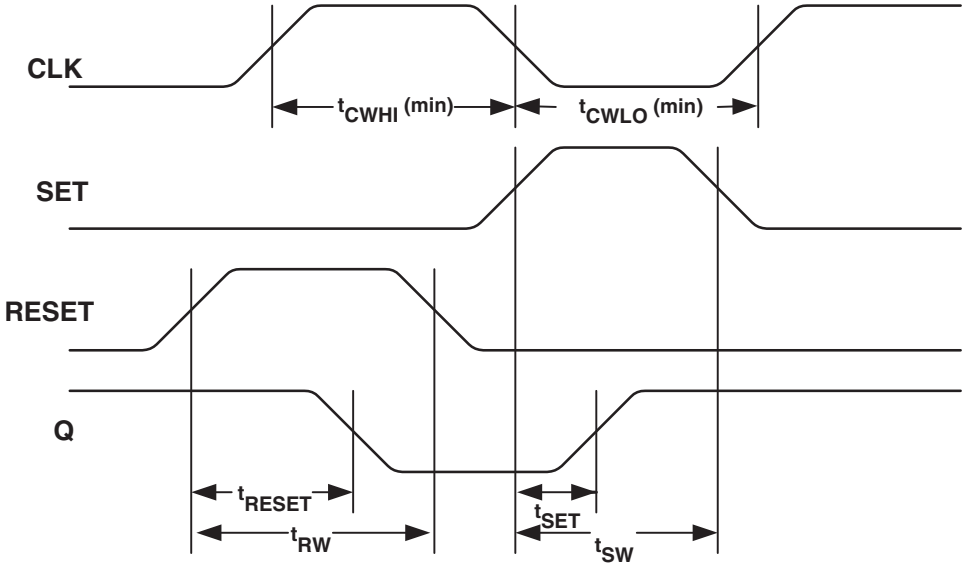


Figure 10: Logic Cell Flip Flop Timing - First Waveform

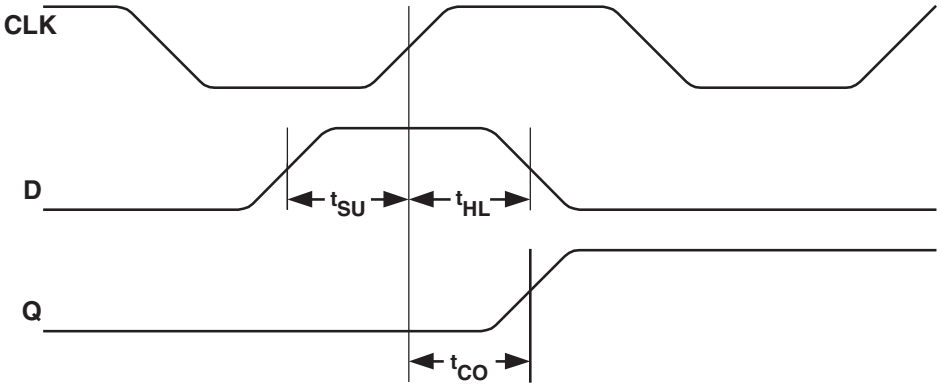


Figure 11: Logic Cell Flip Flop Timing - Second Waveform

Table 12: RAM Cell Synchronous Write Timing

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

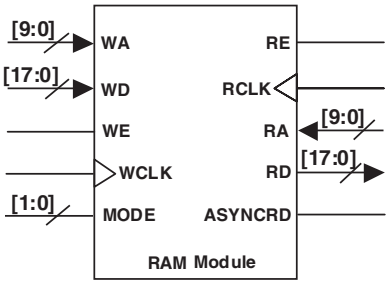


Figure 12: RAM Module

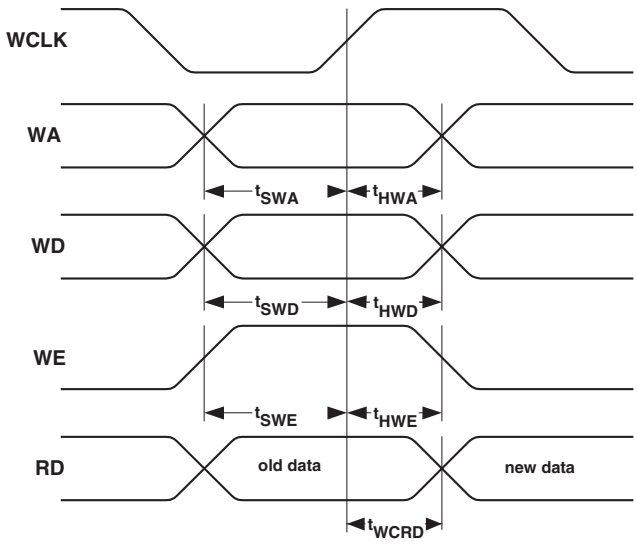


Figure 13: RAM Cell Synchronous Write Timing

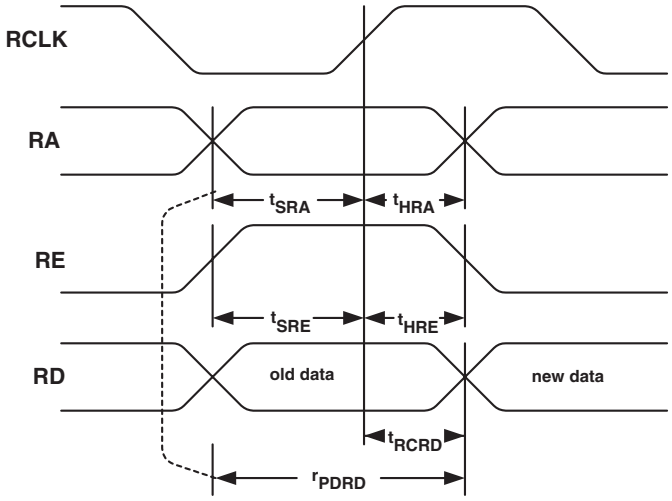


Figure 14: RAM Cell Synchronous & Asynchronous Read Timing

Table 13: RAM Cell Synchronous & Asynchronous Read Timing

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

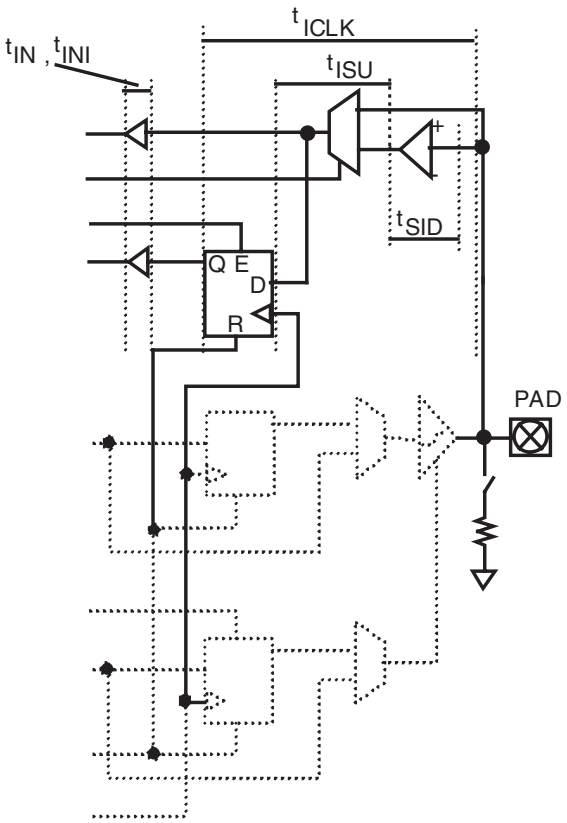


Figure 15: Input Register Cell

Table 14: Input Register Cell

Symbol	Parameter: Input Cell Register Only	Value(ns)	
		Min	Max
t_{ISU}	**Input register setup time**: time the synchronous input of the flip flop must be stable before the active clock edge	3.12	-
t_{IHL}	**Input register hold time**: time the synchronous input of the flip flop must be stable after the active clock edge	0	-
t_{ICLK}	**Input register clock to out**: time taken by the flip flop to output after the active clock edge	-	1.08
t_{IRST}	**Input register reset delay**: time between when the flip flop is "reset"(low) and when the output is consequently "reset" (low)	-	0.99
t_{IESU}	**Input register clock enable setup time**: time "enable" must be stable before the active clock edge	0.37	-
t_{IEH}	**Input register clock enable hold time**: time "enable" must be stable after the active clock edge	0	-

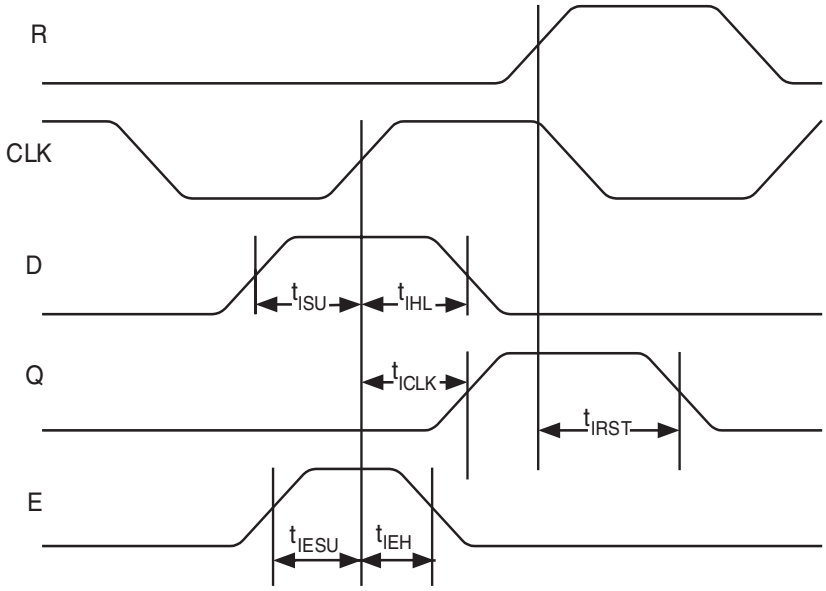


Figure 16: Input Register Cell Timings

Table 15: Standard Input Delays

Symbol Standard Input Delays	Parameter To get the total input delay add this delay to tISU	Value (ns) Min	 Max
$t_{SID(LVTTL)}$	**LVTTL input delay**: Low Voltage TTL for 3.3V applications	-	0.34
$t_{SID(LVCMOS2)}$	**LVCMOS2 input delay**: Low Voltage CMOS for 2.5V and lower applications	-	0.42
$t_{SID(GTL+)}$	**GTL+ input delay**: Gunning Transceiver Logic	-	0.68
$t_{SID(SSTL3)}$	**SSTL3 input delay**: Stub Series Terminated Logic for 3.3V	-	0.55
$t_{SID(SSTL2)}$	**SSTL2 input delay**: Stub Series Terminated Logic for 2.5V	-	0.61

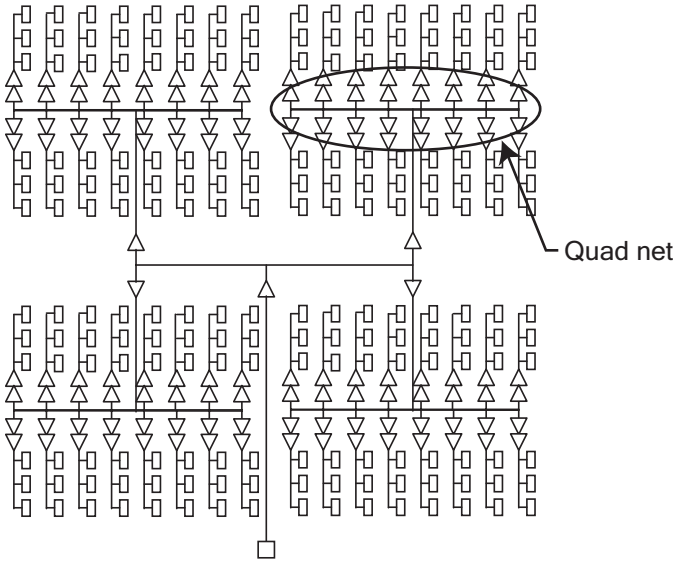


Figure 17: Global Clock Structure

Table 16: Clock Delay

Clock	Parameters	Clock Performance	
		Global	Dedicated
Logic Cells (Internal)	Clock signal generated internally	1.51 ns (max)	n/a
I/O's (External)	Clock signal generated externally	2.06 ns (max)	1.73 ns (max)

Table 17: Eclipse Global Clock Performance

Clock Segment	Parameter	Value (ns)	
		Min	Max
t_{PGCK}	Global clock pin delay to quad net	-	1.34
t_{BGCK}	Global clock buffer delay (quad net to flip flop)	-	0.56

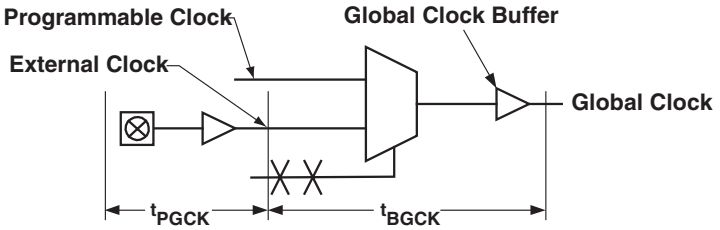


Figure 18: Global Clock Structure Schematic

Table 18: Output Register Cell

Symbol	Parameter: Output Register Cell Only	Min	Max
t_{OUTLH}	Output Delay low to high (90% of H)	-	0.40
t_{OUTHL}	Output Delay high to low (10% of L)	-	0.55
t_{PZH}	Output Delay tri-state to high (90% of H)	-	2.94
t_{PZL}	Output Delay tri-state to low (10% of L)	-	2.34
t_{PHZ}	Output Delay high to tri-State	-	3.07
t_{PLZ}	Output Delay low to tri-State	-	2.53
t_{COP}	Clock to out delay (does not include clock tree delays)	-	3.15 (fast slew) 10.2 (slow slew)

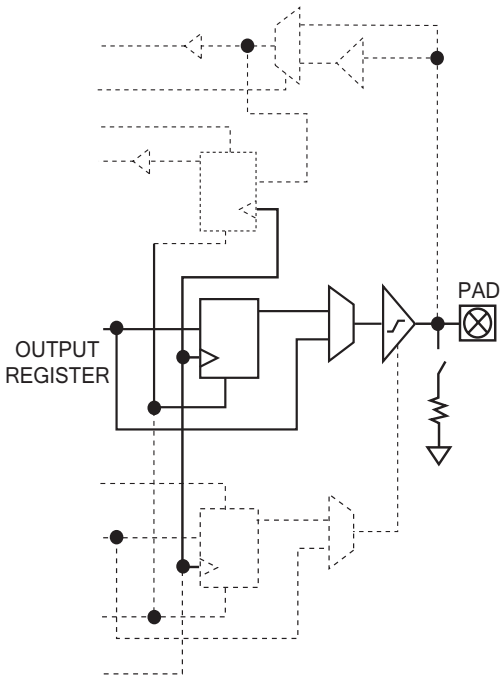


Figure 19: Output Register Cell

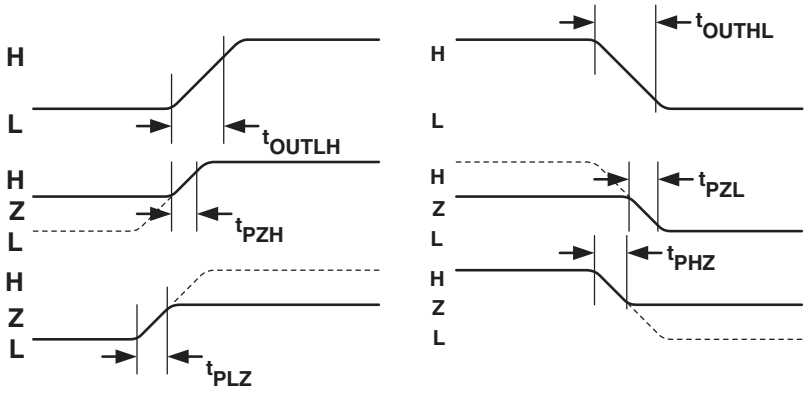


Figure 20: Output Register Cell Timing

Table 19: Output Slew Rates @ VCCIO = 3.3 V

	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 20: Output Slew Rates @ V_{CCIO} = 2.5 V

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

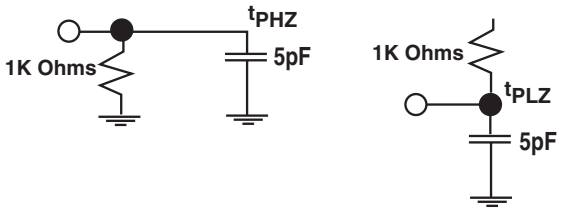


Figure 21: Loads for t_{PXZ}

Pin Type Descriptions

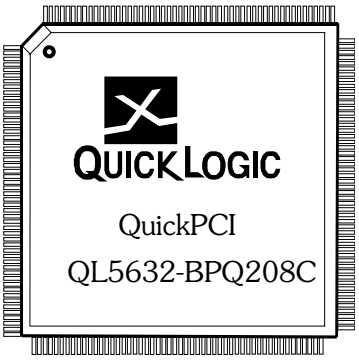
The QL5632 Device Pins are indicated in **Table 21**. Some of the pins presented in this table connect to the PCI bus, and others are programmable as user I/O.

Table 21: Pin Descriptions

Pin	Function	Description
TDI/RSI	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 V_{CC} if unused
TRSTB/RRO	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	Test Mode Select for JTAG	Hold HIGH during normal operation. Connect to V_{CC} if not used for JTAG
TCK	Test Clock for JTAG	Hold HIGH or LOW during normal operation. Connect to V_{CC} or ground if not used for JTAG
TDO/RCO	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
I/GCLK	High-drive input and/or global clock network driver	Can be configured as either input or global clock
I/O	Input/Output pin	Can be configured as an input and/or output
V_{CC}	Power supply pin	Connect to 2.5 V supply
$V_{CCIO<PCI>}$	Input voltage tolerance pin	Connect to 3.3 V supply
$V_{CCIO<A>-<E>}$	Input voltage tolerance pin	Connect to 3.3 V supply if 3.3 V input tolerance is required; otherwise, connect to 2.5 V supply
GND	Ground pin	Connect to ground
PLLIN	PLL clock input	Clock input for PLL
DEDCLK	Dedicated clock pin	Low skew global clock
GNDPLL	Ground pin for PLL	Connect to GND
INREF	Differential reference voltage	Connect to reference voltage or ground if used for non-differential input
PLLOUT	PLL output pin	Dedicated PLL output pin. Otherwise may be left unconnected
IOCTRL	Highdrive input	Can be used as highdrive input or clock to I/O register within the same bank. Tied low or high if unused

208 PQFP Pinout Diagram

Top



208 PQFP Pinout Table

Table 10: 208 PQFP Pinout Table

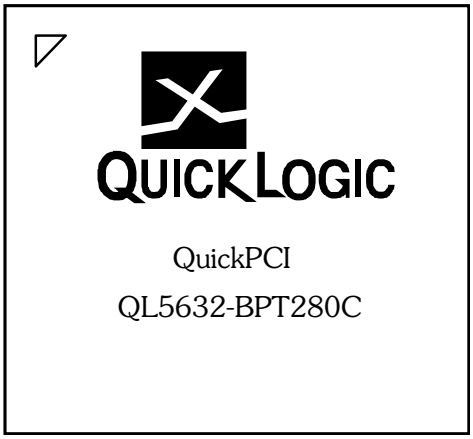
208 PQFP	Function	208 PQFP	Function	208 PQFP	Function	208 PQFP	Function	208 PQFP	Function	208 PQFP	Function
1	PLLRST(3)	36	IO(B)	71	IO(C)	106	VCCPLL(1)	141	AD [30]	176	CBEN [2]
2	VCCPLL(3)	37	IO(B)	72	VCCIO(C)	107	IO(E)	142	AD [29]	177	VCCIO(G)
3	GND	38	IO(B)	73	IO(C)	108	GND	143	AD [28]	178	GND
4	GND	39	IOCTRL(B)	74	IO(C)	109	IO(E)	144	IOCTRL(F)	179	FRAMEN
5	AD [8]	40	INREF(B)	75	GND	110	IO(E)	145	INREF(F)	180	IRDYN
6	CBEN [0]	41	IOCTRL(B)	76	VCC	111	VCCIO(E)	146	VCC	181	TRDYN
7	AD [7]	42	IO(B)	77	IO(C)	112	IO(E)	147	IOCTRL(F)	182	VCC
8	VCCIO(A)	43	IO(B)	78	TRSTB	113	VCC	148	AD [27]	183	TCK
9	AD [6]	44	VCCIO(B)	79	VCC	114	IO(E)	149	AD [26]	184	VCC
10	AD [5]	45	IO(B)	80	IO(D)	115	IO(E)	150	VCCIO(F)	185	DEVSELN
11	IOCTRL(A)	46	VCC	81	IO(D)	116	IO(E)	151	AD [25]	186	STOPN
12	VCC	47	IO(B)	82	IO(D)	117	IOCTRL(E)	152	AD [24]	187	PERRN
13	INREF(A)	48	IO(B)	83	GND	118	INREF(E)	153	GND	188	GND
14	IOCTRL(A)	49	GND	84	VCCIO(D)	119	IOCTRL(E)	154	CBEN [3]	189	VCCIO(H)
15	AD [4]	50	TDO	85	IO(D)	120	IO(E)	155	PLLOUT(3)	190	SERRN
16	AD [3]	51	PLLOUT(1)	86	VCC	121	IO(E)	156	GNDPLL(0)	191	PAR
17	AD [2]	52	GNDPLL(2)	87	IO(D)	122	VCCIO(E)	157	GND	192	IOCTRL(H)
18	AD [1]	53	GND	88	IO(D)	123	GND	158	VCCPLL(0)	193	CBEN [1]
19	VCCIO(A)	54	VCCPLL(2)	89	VCC	124	IO(E)	159	PLLRST(0)	194	INREF(H)
20	AD [0]	55	PLLRST(2)	90	IO(D)	125	IO(E)	160	GND	195	VCC
21	GND	56	VCC	91	IO(D)	126	IO(E)	161	IDSEL	196	IOCTRL(H)
22	IO(A)	57	IO(C)	92	IOCTRL(D)	127	CLK(5),PLLIN(3)	162	VCCIO(G)	197	AD [15]
23	TDI	58	GND	93	INREF(D)	128	CLK(6)	163	AD [23]	198	AD [14]
24	CLK(0)	59	IO(C)	94	IOCTRL(D)	129	VCC	164	AD [22]	199	AD [13]
25	CLK(1)	60	VCCIO(C)	95	IO(D)	130	CLK(7)	165	VCC	200	AD [12]
26	VCC	61	IO(C)	96	IO(D)	131	VCC	166	AD [21]	201	AD [11]
27	CLK(2),PLLIN(2)	62	IO(C)	97	IO(D)	132	CLK	167	AD [20]	202	AD [10]
28	CLK(3),PLLIN(1)	63	IO(C)	98	VCCIO(D)	133	TMS	168	AD [19]	203	VCCIO(H)
29	VCC	64	IO(C)	99	IO(D)	134	IO(F)	169	IOCTRL(G)	204	GND
30	CLK(4),DEDCLK,PLLIN(0)	65	IO(C)	100	IO(D)	135	RSTN	170	INREF(G)	205	AD [9]
31	IO(B)	66	IO(C)	101	GND	136	GNTN	171	IOCTRL(G)	206	PLLOUT(2)

Table 10: 208 PQFP Pinout Table

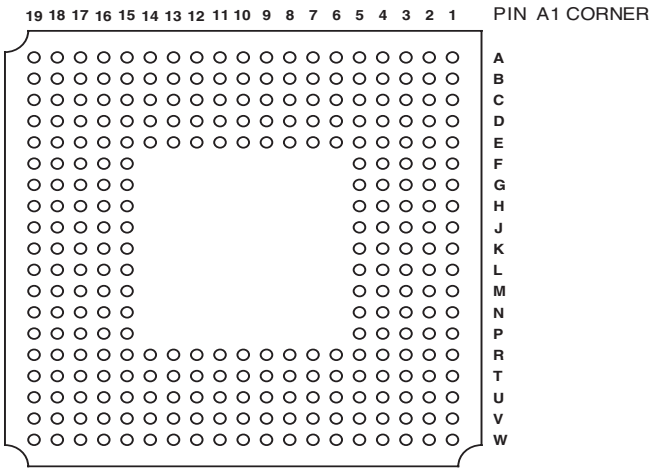
208 PQFP	Function	208 PQFP	Function	208 PQFP	Function	208 PQFP	Function	208 PQFP	Function	208 PQFP	Function
32	IO(B)	67	IOCTRL(C)	102	PLLOUT(0)	137	GND	172	AD [18]	207	GND
33	GND	68	INREF(C)	103	GND	138	VCCIO(F)	173	AD [17]	208	GNDPLL(3)
34	VCCIO(B)	69	IOCTRL(C)	104	GNDPLL(1)	139	REQN	174	AD [16]		
35	IO(B)	70	IO(C)	105	PLLRST(1)	140	AD [31]	175	VCC		

280 PBGA Pinout Diagram

Top



Bottom



280 PBGA Pinout Table

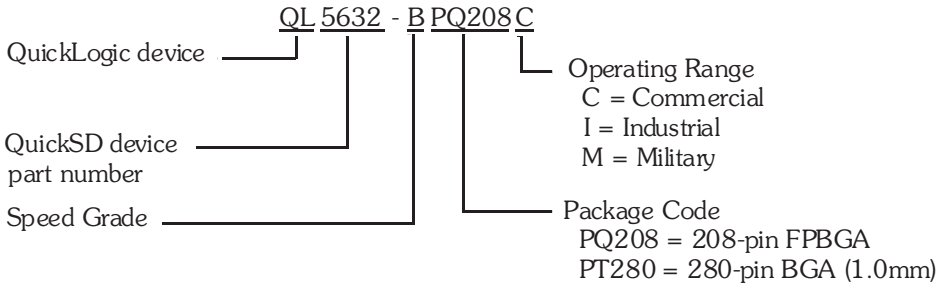
Table 11: 280 PBGA Pinout Table

280 PBGA	Function	280 PBGA	Function	280 PBGA	Function	280 PBGA	Function	280 PBGA	Function	280 PBGA	Function
A1	PLLOUT<3>	C10	CLK<5>/PLLIN <3>	E19	IOCTRL<D>	K16	I/O<C>	R4	I/O<H>	U13	I/O<B>
A2	GNDPLL<0>	C11	VCCIO<E>	F1	INREF<G>	K17	I/O<D>	R5	GND	U14	IOCTRL<B>
A3	AD [18]	C12	I/O<E>	F2	IOCTRL<G>	K18	I/O<C>	R6	GND	U15	VCCIO<B>
A4	AD [20]	C13	I/O<E>	F3	SERRN	K19	TRSTB	R7	VCC	U16	I/O<B>
A5	IDSEL	C14	I/O<E>	F4	DEVSELN	L1	AD [4]	R8	VCC	U17	TDO
A6	IOCTRL<F>	C15	VCCIO<E>	F5	GND	L2	AD [5]	R9	GND	U18	PLLRST<2>
A7	AD [26]	C16	I/O<E>	F15	VCC	L3	VCCIO<H>	R10	GND	U19	I/O<B>
A8	AD [30]	C17	I/O<E>	F16	IOCTRL<D>	L4	AD [6]	R11	VCC	V1	PLLOUT<2>
A9	RSTN	C18	I/O<E>	F17	I/O<D>	L5	VCC	R12	VCC	V2	GNDPLL<3>
A10	CLK<7>	C19	I/O<E>	F18	I/O<D>	L15	GND	R13	VCC	V3	GND
A11	I/O<E>	D1	TRDYN	F19	I/O<D>	L16	I/O<C>	R14	VCC	V4	I/O<A>

Table 11: 280 PBGA Pinout Table

280 PBGA	Function	280 PBGA	Function	280 PBGA	Function	280 PBGA	Function	280 PBGA	Function	280 PBGA	Function
A12	I/O<E>	D2	IRDYN	G1	AD [14]	L17	VCCIO<C>	R15	GND	V5	I/O<A>
A13	I/O<E>	D3	AD [16]	G2	AD [15]	L18	I/O<C>	R16	I/O<C>	V6	IOCTRL<A>
A14	IOCTRL<E>	D4	AD [23]	G3	IOCTRL<G>	L19	I/O<C>	R17	VCCIO<C>	V7	I/O<A>
A15	I/O<E>	D5	AD [24]	G4	PAR	M1	AD [0]	R18	I/O<C>	V8	I/O<A>
A16	I/O<E>	D6	AD [25]	G5	VCC	M2	AD [1]	R19	I/O<C>	V9	I/O<A>
A17	I/O<E>	D7	AD [29]	G15	VCC	M3	AD [2]	T1	I/O<H>	V10	CLK<1>
A18	PLLRST<1>	D8	GNTN	G16	I/O<D>	M4	AD [3]	T2	I/O<H>	V11	CLK<4>DEDCLK/PLLIN<0>
A19	GND	D9	CLK	G17	I/O<D>	M5	VCC	T3	I/O<A>	V12	I/O<B>
B1	PLLRST<0>	D10	I/O<E>	G18	I/O<D>	M15	VCC	T4	I/O<A>	V13	I/O<B>
B2	GND	D11	I/O<E>	G19	I/O<D>	M16	INREF<C>	T5	I/O<A>	V14	INREF<B>
B3	AD [19]	D12	I/O<E>	H1	AD [11]	M17	I/O<C>	T6	IOCTRL<A>	V15	I/O<B>
B4	AD [21]	D13	INREF<E>	H2	AD [12]	M18	I/O<C>	T7	I/O<A>	V16	I/O<B>
B5	CBEN [3]	D14	I/O<E>	H3	AD [13]	M19	I/O<C>	T8	I/O<A>	V17	I/O<B>
B6	INREF<F>	D15	I/O<E>	H4	I/O<G>	N1	IOCTRL<H>	T9	I/O<A>	V18	GNDPLL<2>
B7	AD [27]	D16	I/O<D>	H5	CBEN [1]	N2	I/O<H>	T10	I/O<A>	V19	GND
B8	AD [31]	D17	I/O<D>	H15	VCC	N3	I/O<H>	T11	CLK<3>/PLLIN<1>	W1	GND
B9	TMS	D18	I/O<D>	H16	VCC	N4	I/O<H>	T12	I/O<B>	W2	PLLRST<3>
B10	CLK<6>	D19	I/O<D>	H17	I/O<D>	N5	VCC	T13	I/O<B>	W3	I/O<A>
B11	I/O<E>	E1	PERRN	H18	I/O<D>	N15	VCC	T14	I/O<B>	W4	I/O<A>
B12	I/O<E>	E2	STOPN	H19	I/O<D>	N16	I/O<C>	T15	I/O<B>	W5	I/O<A>
B13	IOCTRL<E>	E3	VCCIO<G>	J1	AD [8]	N17	I/O<C>	T16	I/O<B>	W6	I/O<A>
B14	I/O<E>	E4	FRAMEN	J2	AD [9]	N18	IOCTRL<C>	T17	VCCPLL<2>	W7	I/O<A>
B15	I/O<E>	E5	GND	J3	VCCIO<G>	N19	IOCTRL<C>	T18	I/O<B>	W8	I/O<A>
B16	I/O<E>	E6	VCC	J4	AD [10]	P1	I/O<H>	T19	I/O<B>	W9	TDI
B17	VCCPLL<1>	E7	VCC	J5	GND	P2	I/O<H>	U1	I/O<A>	W10	CLK<2>/PLLIN<2>
B18	GNDPLL<1>	E8	VCC	J15	VCC	P3	IOCTRL<H>	U2	I/O<A>	W11	I/O<B>
B19	PLLOUT<0>	E9	VCC	J16	I/O<C>	P4	INREF<H>	U3	VCCPLL<3>	W12	I/O<B>
C1	CBEN [2]	E10	GND	J17	VCCIO<D>	P5	VCC	U4	I/O<A>	W13	I/O<B>
C2	VCCPLL<0>	E11	GND	J18	I/O<D>	P15	GND	U5	VCCIO<A>	W14	IOCTRL<B>
C3	AD [17]	E12	VCC	J19	I/O<D>	P16	I/O<C>	U6	INREF<A>	W15	I/O<B>
C4	AD [22]	E13	VCC	K1	VCC	P17	I/O<C>	U7	I/O<A>	W16	I/O<B>
C5	VCCIO<F>	E14	GND	K2	TCK	P18	I/O<C>	U8	I/O<A>	W17	I/O<B>
C6	IOCTRL<F>	E15	GND	K3	AD [7]	P19	I/O<C>	U9	VCCIO<A>	W18	I/O<B>
C7	AD[28]	E16	I/O<D>	K4	CBEN [0]	R1	I/O<H>	U10	CLK<0>	W19	PLLOUT<1>
C8	REQN	E17	VCCIO<D>	K5	GND	R2	I/O<H>	U11	VCCIO<B>		
C9	VCCIO<F>	E18	INREF<D>	K15	GND	R3	VCCIO<H>	U12	I/O<B>		

Ordering Information



Revision History

Table 12: Revision History

Revision	Date	Originator and Comments
Rev. A	August 2002	Bernhard Andretzky, Stacy Joseph, Andreea Rotaru, Paul Micallef

Copyright Information

Contact Information

Telephone: 408 990 4000 (US)
416 497 8884 (Canada)
44 1932 57 9011 (Europe)
49 89 930 86 170 (Germany)
852 8106 9091 (Asia)
81 45 470 5525 (Japan)

E-mail: info@quicklogic.com

Support: support@quicklogic.com

Web site: http://www.quicklogic.com/