Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details	
Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	1568
Total RAM Bits	25600
Number of I/O	221
Number of Gates	48000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	256-BGA
Supplier Device Package	256-FPBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/lattice-semiconductor/or3t307ba256-db

Description (continued)

ispLEVER Development System

The ispLEVER Development System is used to process a design from a netlist to a configured FPGA. This system is used to map a design onto the *ORCA* architecture and then place and route it using ispLEVER's timing-driven tools. The development system also includes interfaces to, and libraries for, other popular CAE tools for design entry, synthesis, simulation, and timing analysis.

The ispLEVER Development System interfaces to front-end design entry tools and provides the tools to produce a configured FPGA. In the design flow, the user defines the functionality of the FPGA at two points in the design flow: at design entry and at the bit stream generation stage.

Following design entry, the development system's map, place, and route tools translate the netlist into a routed FPGA. A static timing analysis tool is provided to determine device speed and a back-annotated netlist can be created to allow simulation. Timing and simulation output files from ispLEVER are also compatible with many third-party analysis tools. Its bit stream generator is then used to generate the configuration data which is loaded into the FPGA's internal configuration RAM. When using the bit stream generator, the user selects options that affect the functionality of the FPGA. Combined with the front-end tools, ispLEVER produces configuration data that implements the various logic and routing options discussed in this data sheet.

Architecture

The *ORCA* Series 3 FPGA comprises three basic elements: PLCs, PICs, and system-level functions. Figure 1 shows an array of programmable logic cells (PLCs) surrounded by programmable input/output cells (PICs). Also shown are the interquad routing blocks (hIQ, vIQ) present in Series 3. System-level functions (located in the corners of the array) and the routing resources and configuration RAM are not shown in Figure 1.

The OR3T55 array in Figure 1 has PLCs arranged in an array of 18 rows and 18 columns. The location of a PLC is indicated by its row and column so that a PLC in the second row and the third column is R2C3. PICs are located on all four sides of the FPGA between the PLCs and the device edge. PICs are indicated using PT and PB to designate PICs on the top and bottom sides of the array, respectively, and PL and PR to designate PICs along the left and right sides of the array, respectively. The position of a PIC on an edge of the array is indicated by a number, counting from left to right for PT and PB and top to bottom for PL and PR PICs.

Each PIC contains routing resources and four programmable I/Os (PIOs). Each PIO contains the necessary I/O buffers to interface to bond pads. PIOs in Series 3 FPGAs also contain input and output FFs, fast open-drain capability on output buffers, special output logic functions, and signal multiplexing/demultiplexing capabilities.

PLCs comprise a programmable function unit (PFU), a supplemental logic and interconnect cell (SLIC), and routing resources. The PFU is the main logic element of the PLC, containing elements for both combinatorial and sequential logic. Combinatorial logic is done in look-up tables (LUTs) located in the PFU. The PFU can be used in different modes to meet different logic requirements. The LUT's twin-quad architecture provides a configurable medium-/large-grain architecture that can be used to implement from one to eight independent combinatorial logic functions or a large number of complex logic functions using multiple LUTs. The flexibility of the LUT to handle wide input functions, as well as multiple smaller input functions, maximizes the gate count per PFU while increasing system speed.

The LUTs can be programmed to operate in one of three modes: combinatorial, ripple, or memory. In combinatorial mode, the LUTs can realize any 4- or 5-input logic function and many multilevel logic functions using *ORCA*'s softwired LUT (SWL) connections. In ripple mode, the high-speed carry logic is used for arithmetic functions, comparator functions, or enhanced data path functions. In memory mode, the LUTs can be used as a 32 x 4 synchronous read/write or read-only memory, in either single- or dual-port mode.

Programmable Logic Cells (continued)

Inter-PLC Routing Resources

The inter-PLC routing is used to route signals between PLCs. The routing segments occur in groups of ten, and differ in the numbers of PLCs spanned. The x1 routing segments span one PLC, the x5 routing segments span five PLCs, the xH routing segments span one-half the width (height) of the PLC array, and the xL routing segments span the width (height) of the PLC array. All types of routing segments run in both horizontal and vertical directions.

Table 8 shows the groups of inter-PLC routing segments in each PLC. In the table, there are two rows/columns for x1 lines. They are differentiated by a T for top, B for bottom, L for left, and R for right. In the ispLEVER design editor representation, the horizontal x1 routing segments are located above and below the PFU. The two groups of vertical segments are located on the left side of the PFU. The xL and x5 routing segments only run below and to the left of the PFU, while the xH segments only run above and to the right of the PFU. The indexes specify individual routing segments within a group. For example, the vx5[2] segment runs vertically to the left of the PFU, spans five PLCs, and is the third line in the 10-bit wide group.

PLCs are arranged like tiles on the *ORCA* device. Breaks in routing occur at the middle of the tile (e.g., x1 lines break in the middle of each PLC) and run across tiles until the next break.

Table 8. Inter-PLC Routing Resources

Horizontal Routing Segments	Vertical Routing Segments	Distance Spanned
hx1U[9:0]	vx1R[9:0]	One PLC
hx1B[9:0]	vx1L[9:0]	One PLC
hx5[9:0]	vx5[9:0]	Five PLCs
hx5[9:0]	vx5[9:0]	Five PLCs
hxL[9:0]	vxL[9:0]	PLC Array
hxH[9:0]	vxH[9:0]	1/2 PLC Array
hCLK	vCLK	PLC Array

Figure 20 provides a global view of inter-PLC routing resources across multiple PLCs.

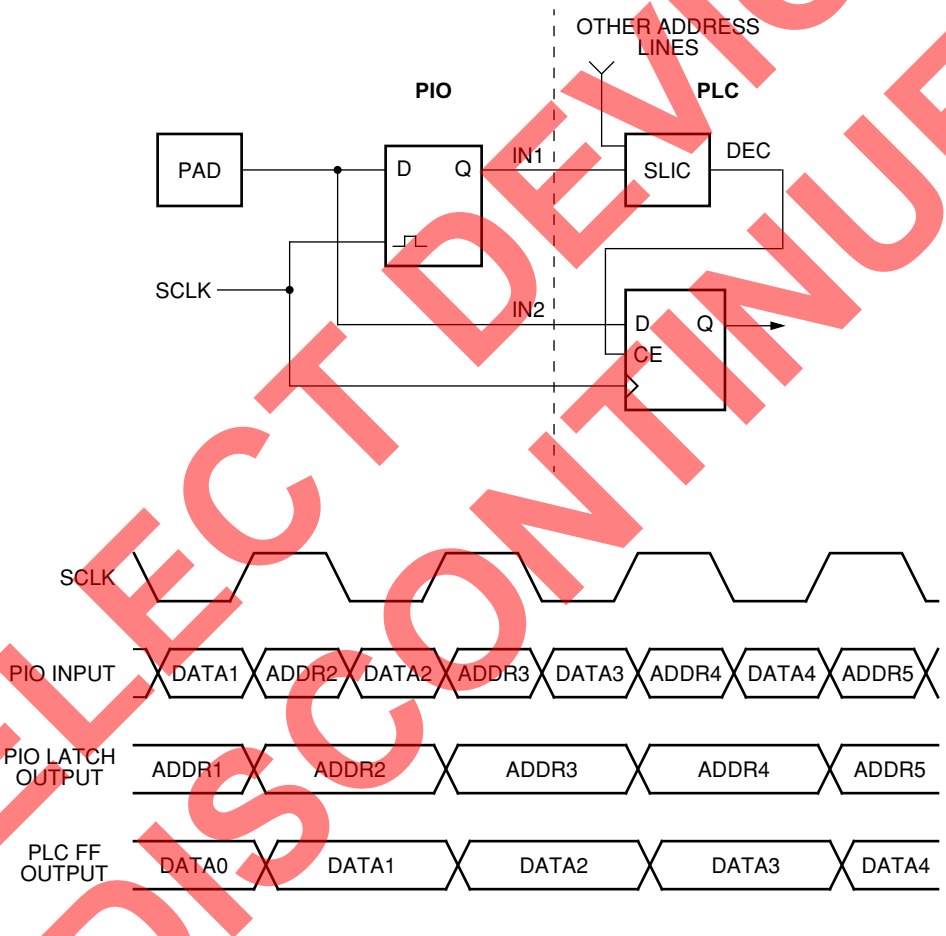
x1 Routing Segments. There are a total of 40 x1 routing segments per PLC: 20 vertical and 20 horizontal. Each of these are subdivided into two, 10-bit wide buses: hx1T[9:0], hx1B[9:0], vx1L[9:0], and vx1R[9:0]. An x1 segment is one PLC long. If a signal net is longer than one PLC, an x1 segment can be lengthened to n times its length by turning on n – 1 CIPs. A signal is routed onto an x1 route segment via the switching routing segments or BIDI routing segments which also allows the x1 route segment to be connected to other inter-PLC segments of different lengths. Corner turning between x1 segments is provided through direct connections, xSW segments, and xBID segments.

x5 Routing Segments. There are two sets of ten x5 routing segments per PLC. One set (vx5[9:0]) runs vertically, and the other (hx5[9:0]) runs horizontally. Each x5 segment traverses five PLCs before it is broken by a CIP. Two x5 segments in each group break in each PLC. The two that break are in an equivalent pair; for example, x5[0] and x5[4]. The x5 segments that break shift by one at the next PLC. For example, if hx5[0] and hx5[4] are broken at the current PLC, hx5[1] and hx5[5] will be broken at the PLC to the right of the current PLC. There are direct connections to the BIDI routing segments in the PLC at which the x5 segments break, on both sides of the break. Signal corner turning is enabled by CIPs in each PLC that allow the broken x5 segments to directly connect to the broken x5 segments that run in the orthogonal direction. x5 corner turning can also be accomplished via the xSW and xBID segments in a PLC. In addition, the x5 segments are connected to the FINS and PFU outputs on a bit-by-bit basis by the xSW segments. x5 segments can be connected for signal runs in multiples of five PLCs, or they can be combined with x1 and xH routing segments for runs of varying distances.

Programmable Input/Output Cells (continued)

Input Demultiplexing

The combination of input register capability and the two inputs, IN1 and IN2, from each PIO to the internal routing provides for input signal demultiplexing without any additional resources. Figure 24 shows the input configuration and general timing for demultiplexing a multiplexed address and data signal. The PIO input signal is sent to both the input latch and directly to IN2. The signal is latched on the falling edge of the clock and output to routing at IN1. The address and data are then both available at the rising edge of the system clock. These signals may be registered or otherwise processed in the PLCs at that clock edge. Figure 24 also shows the possible use of the SLIC decoder to perform an address decode to enable which registers are to receive the input data. Although the timing shown is for using the input register as a latch, it may also be used in the same way as an FF. Also note that the signals found in PIO inputs IN1 and IN2 can be interchanged.



5-5798(F)

Figure 24. PIO Input Demultiplexing

Programmable Input/Output Cells

(continued)

PIO Logic Function Generator

The PIO logic block can also generate logic functions based on the signals on the OUT2 and CLK ports of the PIO. The functions are AND, NAND, OR, NOR, XOR, and XNOR. Table 10 is provided as a summary of the PIO logic options.

Table 10. PIO Logic Options

Option	Description
OUT1OUTREG	Data at OUT1 output when clock low, data at FF out when clock high.
OUT2OUTREG	Data at OUT2 output when clock low, data at FF out when clock high.
OUT1OUT2	Data at OUT1 output when clock low, data at OUT2 when clock high.
AND	Output logical AND of signals on OUT2 and clock.
NAND	Output logical NAND of signals on OUT2 and clock.
OR	Output logical OR of signals on OUT2 and clock.
NOR	Output logical NOR of signals on OUT2 and clock.
XOR	Output logical XOR of signals on OUT2 and clock.
XNOR	Output logical XNOR of signals on OUT2 and clock.

PIO Register Control Signals

As discussed in the Inputs and Outputs subsections, the PIO latches/FFs have various clock, clock enable (CE), local set/reset (LSR), and global set/reset (GSRN) controls. Table 11 provides a summary of these control signals and their effect on the PIO latches/FFs. Note that all control signals are optionally invertible.

Table 11. PIO Register Control Signals

Control Signal	Effect/Functionality
ExpressCLK	Clocks input fast-capture latch; optionally clocks output FF, or 3-state FF.
System Clock (SCLK)	Clocks input latch/FF; optionally clocks output FF, or 3-state FF.
Clock Enable (CE)	Optionally enables/disables input FF (not available for input latch mode); optionally enables/disables output FF; separate CE inversion capability for input and output.
Local Set/Reset (LSR)	Option to disable; affects input latch/FF, output FF, and 3-state FF if enabled.
Global Set/Reset (GSRN)	Option to enable or disable per PIO after initial configuration.
Set/Reset Mode	The input latch/FF, output FF, and 3-state FF are individually set or reset by both the LSR and GSRN inputs.

Clock Distribution Network (continued)

Clock Sources to the PLC Array

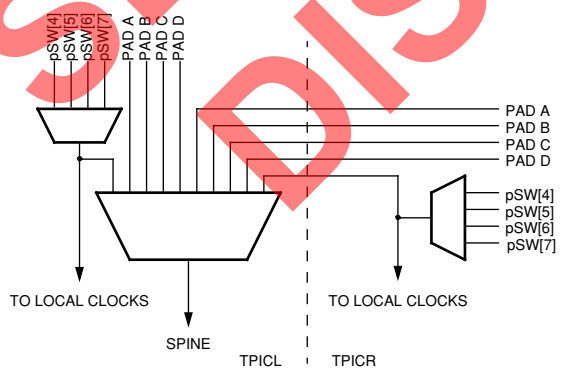
The source of a clock that is globally available to the PLC array can be from any user I/O pad, any of the ExpressCLK pads, or an internally generated source.

System Clock

As described in the Programmable Input/Output Cells section, PICs are grouped in adjacent pairs. Any one of the eight pads in a PIC pair can drive a clock spine in a row or column. For PIC pairs on the top of the chip, the column associated with the left PIC has the clock spine, for pairs on the bottom, the right PIC column has the spine. The top PIC of the pair sources the spine from the left side of the array, and the bottom PIC of the pair sources the spine from the right side of the array. Clock delay and skew are minimized by having a single clock buffer per pair of PICs. The clock spine for each pair can also be driven by one of the four PIC switching segments (pSW) in each PIC of the pair. This allows a signal generated in the PLC array to be routed onto the global clock spine network. The system clock output of the programmable clock manager (PCM) may also be routed to the global system clock spines via the pSW segments. Figure 33 shows the clock spine multiplexing structure for a pair of PICs on the top of the array.

Fast Clock

The fast clock spines are sourced to the PLC array from each side of the device by the ExpressCLK pads via the CLKCNTRL function block (described in the Special Function Blocks section). The ExpressCLK and fast clock source from the pads is shown in Figure 34 and will be described further in the ExpressCLK Inputs subsection.



5-5800(F)

Figure 33. PIC System Clock Spine Generation

Clocks in the PICs

Because the Series 3 FPGAs have latches and FFs in the I/Os, it is necessary to have clock signal distribution to the PIOs as well as in the PLC array. The system clock, the fast clock, and the ExpressCLK are available for PIO clocking.

PIC System Clock

There are five local system clock lines in each PIC. Much like the sources for a clock in the PFU, two of the local PIC clocks are generated within the PIC from long lines. One is generated from the set of ten PIC long lines (pxL) that runs parallel to the PICs on a side, and the other is generated from the set of ten long lines (xL) from the PLC array that terminate in the PIC. Another local PIC system clock route comes from the set of ten xL lines in the adjacent PLC that is parallel to the side of the array on which the PIC resides. The fourth local PIC system clock route comes from the set of ten long lines (xL) from the PLC array that terminate in the adjacent PIC that is not part of the same PIC pair. Much like the E1 signals in the PLCs that are used to distribute a local clock to the PFU source, the fifth local clock line in each PIC comes from local pSW signals. This clock signal for each PIC is shown in Figure 33. One of these five local PIC system clocks is selected for the system clock signal in the PIO. It is used as the PIO system clock for both input and output clocking as selected within the PIO. All PIOs in a PIC share the same system clock.

PIC ExpressCLK

The ExpressCLK signal used at the PIC latches/FFs comes from the CLKCNTRL function block that resides in the middle of the side on which the PIC resides. A single signal comes from the CLKCNTRL and is driven by separate buffers onto two ExpressCLK long wires. One of these ExpressCLK signals goes to the PICs on the right of (above) the CLKCNTRL block, and the other ExpressCLK signal goes to the PICs on the left of (below) the CLKCNTRL block on that side.

Programmable Clock Manager (PCM) (continued)

PCM Registers

The PCM contains eight user-programmable registers used for configuring the PCM's functionality. Table 26 shows the mapping of the registers and their functions. See Figure 46 for more information on the location of PCM elements that are discussed in the table. The PCM registers are referenced in the discussions that follow. Detailed explanations of all register bits are supplied following the functional description of the PCM.

Table 26. PCM Registers

Address	Function
0	**Divider 0 Programming**. Programmable divider, DIV0, value and DIV0 reset bit. DIV0 can divide the input clock to the PCM or can be bypassed.
1	**Divider 1 Programming**. Programmable divider, DIV1, value and DIV1 reset bit. DIV1 can divide the feedback clock input to the PCM or can be bypassed. Valid only in PLL mode.
2	**Divider 2 Programming**. Programmable divider, DIV2, value and DIV2 reset bit. DIV2 can divide the output of the tapped delay line or can be bypassed and is only valid for the ExpressCLK output.
3	**DLL 2x Duty-Cycle Programming**. DLL mode clock doubler (2x) duty-cycle selection.
4	**DLL 1x Duty-Cycle Programming**. Depending on the settings in other registers, this register is for: a. PLL mode phase/delay selection; b. DLL mode 1x duty cycle selection; and c. DLL mode programmable delay.
5	**Mode Programming**. DLL/PLL mode selection, DLL 1x/2x clock selection, phase detector feedback selection.
6	**Clock Source Status/Output Clock Selection Programming**. Input clock selection, feedback clock selection, ExpressCLK output source selection, system clock output source selection.
7	**PCM Control Programming**. PCM power, reset, and configuration control.

Programmable Clock Manager (PCM)
(continued)

PCM Applications

The applications discussed below are only a small sampling of the possible uses for the PCM. Check the Lattice website for additional application notes.

Clock Phase Adjustment

The PCM may be used to adjust the phase of the input clock. The result is an output clock which has its active edge either preceding or following the active edge of the input clock. Clock phase adjustment is accomplished in DLL mode by delaying the clock. This is discussed in the Delay-Locked Loop (DLL) Mode section. Examples of using the delayed clock as an early or late phase-adjusted clock are outlined in the following paragraphs.

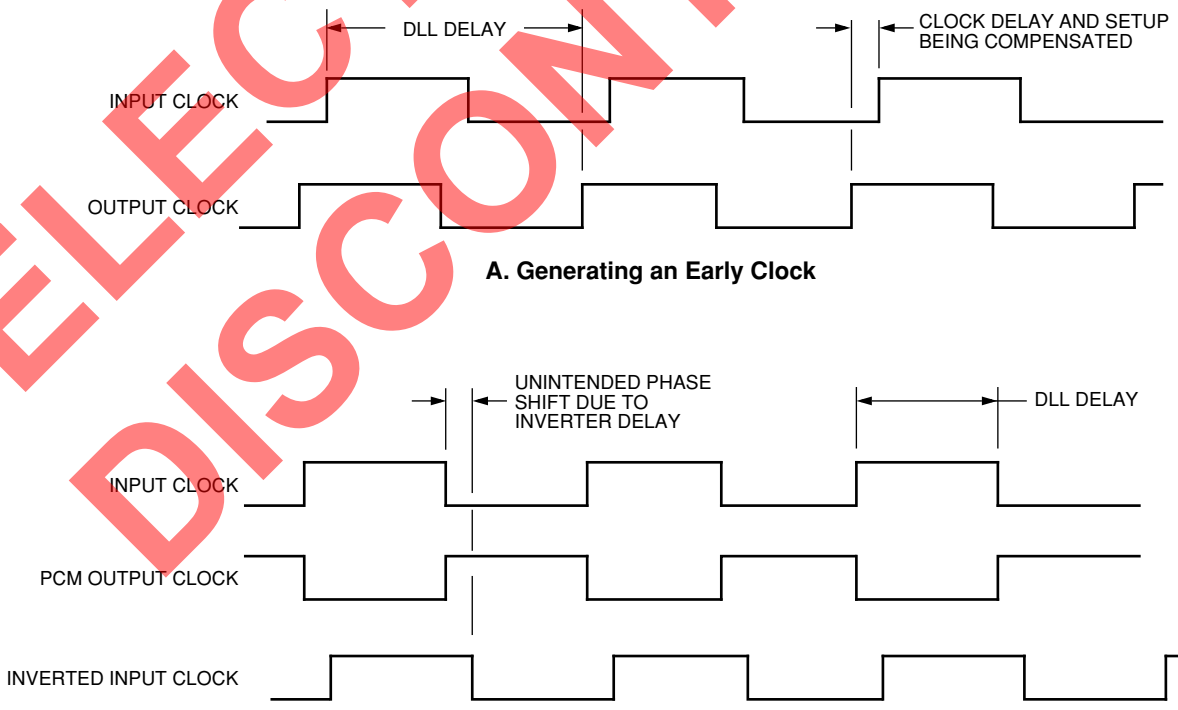
An output clock that precedes the input clock can be used to compensate for clock delay that is largely due to excessive loading. The preceding output clock is really not early relative to the input clock, but is delayed almost a full cycle. This is shown in Figure 48A. The amount of delay that is being compensated for, plus clock setup time and some margin, is the amount **less** than one full clock cycle that the output clock is delayed from the input clock.

In some systems, it is desirable to operate logic from several clocks that operate at different phases. This technique is often used in microprocessor-based systems to transfer and process data synchronously between functional areas, but without incurring excessive delays. Figure 48B shows an input clock and an output clock operating 180° out of phase. It also shows a version of the input clock that was shifted approximately 180° using logic gates to create an inverter. Note that the inverted clock is really shifted more than 180° due to the propagation delay of the inverter. The PCM output clock does not suffer from this delay. Additionally, the 180° shifted PCM output could be shifted by some smaller amount to effect an early 180° shifted clock that also accounts for loading effects.

In terms of degrees of phase shift, the phase of a clock is adjustable in DLL mode with resolution relative to the delay increment (see Table 27):

Phase Adjustment = (Delay)* 11.25, Delay < 16

Phase Adjustment = ((Delay)* 11.25) – 360, Delay > 16



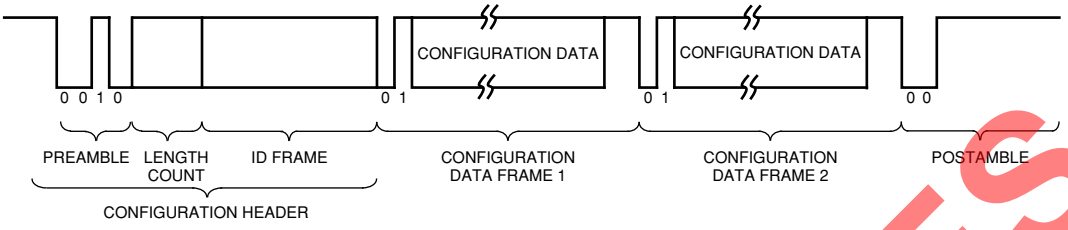
A. Generating an Early Clock

B. Multiphase Clock Generation Using the DLL

5-5979(F)

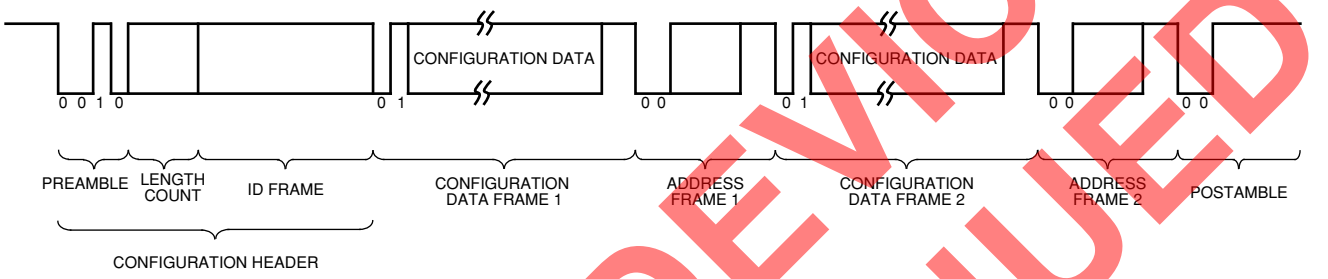
Figure 48. Clock Phase Adjustment Using the PCM

Configuration Data Format (continued)



5-5759(F)

Figure 52. Serial Configuration Data Format—Autoincrement Mode



5-5760(F)

Figure 53. Serial Configuration Data Format—Explicit Mode

Table 32. Configuration Frame Format and Contents

Header	11110010	Preamble
	24-bit Length Count	Configuration frame length.
	11111111	Trailing header—8 bits.
ID Frame	0101 1111 1111 1111	ID frame header.
	Configuration Mode	00 = autoincrement, 01 = explicit.
	Reserved [41:0]	Reserved bits set to 0.
	ID	20-bit part ID.
	Checksum	8-bit checksum.
	11111111	Eight stop bits (high) to separate frames.
Configuration Data Frame (repeated for each data frame)	01	Data frame header.
	Data Bits	Number of data bits depends upon device.
	Alignment Bits = 0	String of 0 bits added to bit stream to make frame header, plus data bits reach a byte boundary.
	Checksum	8-bit checksum.
	11111111	Eight stop bits (high) to separate frames.
Configuration Address Frame	00	Address frame header.
	14 Address Bits	14-bit address of location to start data storage.
	Checksum	8-bit checksum.
	11111111	Eight stop bits (high) to separate frames.
Postamble	00	Postamble header.
	11111111 111111	Dummy address.
	1111111111111111	16 stop bits.*

* In MPI configuration mode, the number of stop bits = 32.

Note: For slave parallel mode, the byte containing the preamble must be 11110010. The number of leading header dummy bits must be (n * 8) + 4, where n is any nonnegative integer and the number of trailing dummy bits must be (n * 8), where n is any positive integer. The number of stop bits/frame for slave parallel mode must be (x * 8), where x is a positive integer. Note also that the bit stream generator tool supplies a bit stream that is compatible with all configuration modes, including slave parallel mode.

Timing Characteristics (continued)

Table 42. Sequential PFU Timing Characteristics

OR3Cxx Commercial: V_{DD} = 5.0 V ± 5%, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C < T_A < +85 °C.
OR3Txxx Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C < T_A < +85 °C.

Parameter	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Input Requirements										
Clock Low Time	CLKL_MPW	3.36	—	2.07	—	0.94	—	0.72	—	ns
Clock High Time	CLKH_MPW	1.61	—	1.06	—	0.54	—	0.45	—	ns
Global S/R Pulse Width (GSRN)	GSR_MPW	3.36	—	2.07	—	0.94	—	0.72	—	ns
Local S/R Pulse Width	LSR_MPW	3.36	—	2.07	—	0.94	—	0.72	—	ns
Combinatorial Setup Times (T_J = +85 °C, V_{DD} = min):										
Four-input Variables to Clock (Kz[3:0] to CLK)*	F4_SET	1.99	—	1.47	—	1.08	—	0.85	—	ns
Five-input Variables to Clock (F5[A:D] to CLK)	F5_SET	1.79	—	1.33	—	1.03	—	0.81	—	ns
Data In to Clock (DIN[7:0] to CLK)	DIN_SET	0.47	—	0.32	—	0.18	—	0.16	—	ns
Carry-in to Clock, DIRECT to REGCOUT (CIN to CLK)	CINDIR_SET	1.25	—	0.99	—	0.71	—	0.58	—	ns
Clock Enable to Clock (CE to CLK)	CE1_SET	2.86	—	2.15	—	1.80	—	1.37	—	ns
Clock Enable to Clock (ASWE to CLK)	CE2_SET	1.68	—	1.30	—	0.95	—	0.77	—	ns
Local Set/Reset to Clock (SYNC) (LSR to CLK)	LSR_SET	1.86	—	1.36	—	0.86	—	0.68	—	ns
Data Select to Clock (SEL to CLK)	SEL_SET	1.37	—	1.00	—	0.92	—	0.70	—	ns
Two-level LUT to Clock (Kz[3:0] to CLK w/feedbk)*	SWL2_SET	3.98	—	2.99	—	2.13	—	1.63	—	ns
Two-level LUT to Clock (F5[A:D] to CLK w/feedbk)	SWL2F5_SET	4.06	—	2.97	—	2.29	—	1.68	—	ns
Three-level LUT to Clock (Kz[3:0] to CLK w/feedbk)*	SWL3_SET	6.49	—	4.81	—	3.42	—	2.64	—	ns
Three-level LUT to Clock (F5[A:D] to CLK w/feedbk)	SWL3F5_SET	6.39	—	4.73	—	3.34	—	2.57	—	ns
Combinatorial Hold Times (T_J = all, V_{DD} = all):										
Data In (DIN[7:0] from CLK)	DIN_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Carry-in from Clock, DIRECT to REGCOUT (CIN from CLK)	CINDIR_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Clock Enable (CE from CLK)	CE1_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Clock Enable from Clock (ASWE from CLK)	CE2_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Local Set/Reset from Clock (sync) (LSR from CLK)	LSR_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Data Select from Clock (SEL from CLK)	SEL_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
All Others	—	0.00	—	0.00	—	0.00	—	0.00	—	ns
Output Characteristics										
Sequential Delays (T_J = +85 °C, V_{DD} = min):										
Local S/R (async) to PFU Out (LSR to Q[7:0], REGCOUT)	LSR_DEL	—	7.02	—	5.29	—	3.64	—	2.90	ns
Global S/R to PFU Out (GSRN to Q[7:0], REGCOUT)	GSR_DEL	—	5.21	—	3.90	—	2.55	—	2.00	ns
Clock to PFU Out—Register (CLK to Q[7:0], REGCOUT)	REG_DEL	—	2.38	—	1.75	—	1.26	—	0.97	ns
Clock to PFU Out—Latch (CLK to Q[7:0])	LTCH_DEL	—	2.51	—	1.88	—	1.21	—	0.96	ns
Transparent Latch (DIN[7:0] to Q[7:0])	LTCHD_DEL	—	2.73	—	2.10	—	1.38	—	1.12	ns

* Four-input variables' (Kz[3:0]) setup times are valid for LUTs in both F4 (four-input LUT) and F5 (five-input LUT) modes.

Note: The table shows worst-case delays. ispLEVER reports the delays for individual paths within a group of paths representing the same timing parameter and may accurately report delays that are less than those listed.

Timing Characteristics (continued)

Table 45. Synchronous Memory Read Characteristics

OR3Cxx Commercial: VDD = 5.0 V ± 5%, 0 °C < TA < 70 °C; Industrial: VDD = 5.0 V ± 10%, –40 °C < TA < +85 °C.
OR3Txxx Commercial: VDD = 3.0 V to 3.6 V, 0 °C < TA < 70 °C; Industrial: VDD = 3.0 V to 3.6 V, –40 °C < TA < +85 °C.

Parameter (TJ = 85 °C, VDD = min)	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Read Operation:										
Data Valid After Address (Kz[3:0] to F[6, 4, 2, 0])	RA_DEL	—	2.34	—	1.80	—	1.32	—	1.05	ns
Data Valid After Address (F5[A:D] to F[6, 4, 2, 0])	RA4_DEL	—	2.11	—	1.57	—	1.23	—	0.99	ns
Read Operation, Clocking Data into Latch/FF:										
Address to Clock Setup Time (Kz[3:0] to CLK)	RA_SET	1.99	—	1.47	—	1.08	—	0.85	—	ns
Address to Clock Setup Time (F5[A:D] to CLK)	RA4_SET	1.79	—	1.33	—	1.03	—	0.81	—	ns
Address from Clock Hold Time (Kz[3:0] from CLK)	RA_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Address from Clock Hold Time (F5[A:D] from CLK)	RA4_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Clock to PFU Output—Register (CLK to Q[6, 4, 2, 0])	REG_DEL	—	2.38	—	1.75	—	1.26	—	0.97	ns
Read Cycle Delay	SMRD_CYC	—	10.48	—	7.66	—	7.53	—	5.78	ns

Note: The table shows worst-case delays. ispLEVER reports the delays for individual paths within a group of paths representing the same timing parameter and may accurately report delays that are less than those listed.

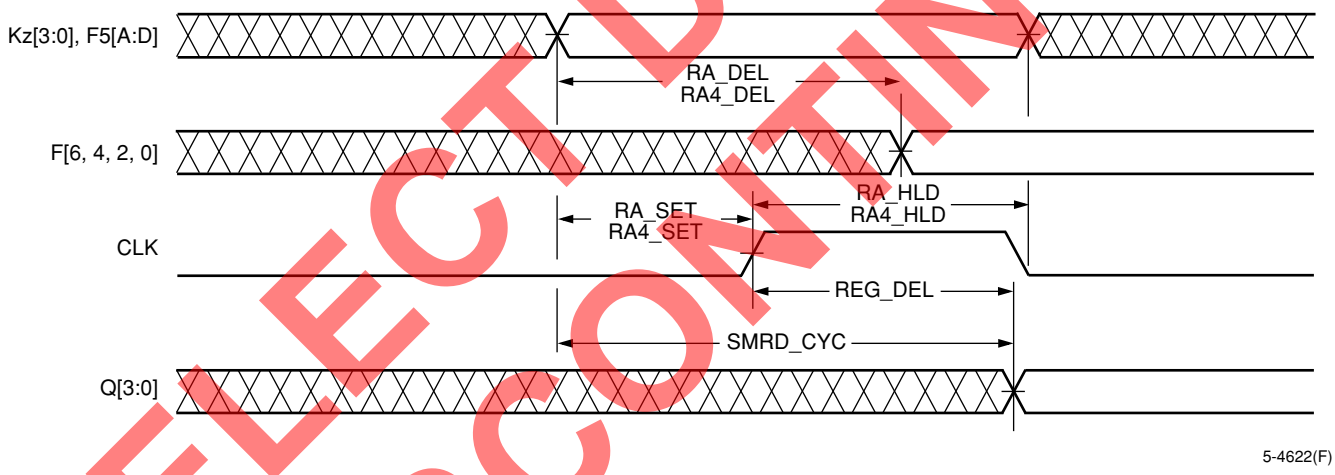


Figure 66. Synchronous Memory Read Cycle

Timing Characteristics (continued)

Table 48. Programmable I/O (PIO) Timing Characteristics (continued)

OR3Cxx Commercial: VDD = 5.0 V ± 5%, 0 °C < TA < 70 °C; Industrial: VDD = 5.0 V ± 10%, –40 °C < TA < +85 °C.
OR3Txxx Commercial: VDD = 3.0 V to 3.6 V, 0 °C < TA < 70 °C; Industrial: VDD = 3.0 V to 3.6 V, –40 °C < TA < +85 °C.

Parameter	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
PIO Logic Block Delays										
Out to Pad (OUT[2:1] via logic to pad):										
Fast	OUTLF_DEL	—	5.09	—	4.21	—	2.63	—	2.17	ns
Slewlim	OUTLSL_DEL	—	7.86	—	6.49	—	3.49	—	2.91	ns
Sinklim	OUTLSI_DEL	—	9.41	—	7.98	—	8.08	—	7.32	ns
Outreg to Pad (OUTREG via logic to pad):										
Fast	OUTRF_DEL	—	6.71	—	5.44	—	3.56	—	2.78	ns
Slewlim	OUTRSL_DEL	—	9.47	—	7.71	—	4.42	—	3.52	ns
Sinklim	OUTRSI_DEL	—	11.03	—	9.20	—	8.98	—	7.94	ns
Clock to Pad (ECLK, CLK via logic to pad):										
Fast	OUTCF_DEL	—	6.97	—	5.68	—	3.71	—	2.91	ns
Slewlim	OUTCSL_DEL	—	9.74	—	7.96	—	4.57	—	3.64	ns
Sinklim	OUTCSI_DEL	—	11.29	—	9.45	—	9.13	—	8.07	ns
3-State FF Delays										
3-state Enable/Disable Delay (TS direct to pad):										
Fast	TSF_DEL	—	4.93	—	4.09	—	2.33	—	1.88	ns
Slewlim	TSSL_DEL	—	7.70	—	6.37	—	3.00	—	2.41	ns
Sinklim	TSSI_DEL	—	9.25	—	7.86	—	7.95	—	7.23	ns
Local Set/Reset (async) to Pad (LSR to pad):										
Fast	TSLSRF_DEL	—	8.25	—	6.65	—	4.24	—	3.39	ns
Slewlim	TSLSRSL_DEL	—	11.01	—	8.92	—	4.92	—	3.92	ns
Sinklim	TSLSRSI_DEL	—	12.57	—	10.41	—	9.87	—	8.74	ns
Global Set/Reset to Pad (GSRN to pad):										
Fast	TSGSRF_DEL	—	7.52	—	6.09	—	3.88	—	3.11	ns
Slewlim	TSGSRSL_DEL	—	10.28	—	8.36	—	4.55	—	3.64	ns
Sinklim	TSGSRSI_DEL	—	11.84	—	9.85	—	9.51	—	8.45	ns
3-State FF Setup Timing:										
TS to ExpressCLK (TS to ECLK)	TSE_SET	0.00	—	0.00	—	0.00	—	0.00	—	ns
TS to Clock (TS to CLK)	TS_SET	0.00	—	0.00	—	0.00	—	0.00	—	ns
Local Set/Reset (sync) to Clock (LSR to CLK)	TSLSR_SET	0.28	—	0.21	—	0.17	—	0.18	—	ns
3-State FF Hold Timing:										
TS from ExpressCLK (TS from ECLK)	TSE_HLD	0.85	—	0.68	—	0.44	—	0.34	—	ns
TS from Clock (TS from CLK)	TS_HLD	0.85	—	0.68	—	0.44	—	0.34	—	ns
Local Set/Reset (sync) from Clock (LSR from CLK)	TSLSR_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Clock to Pad Delay (ECLK, SCLK to pad):										
Fast	TSREGF_DEL	—	5.94	—	4.82	—	2.84	—	2.23	ns
Slewlim	TSREGSL_DEL	—	8.70	—	7.10	—	3.52	—	2.76	ns
Sinklim	TSREGSI_DEL	—	10.26	—	8.59	—	8.47	—	7.58	ns

Note: The delays for all input buffers assume an input rise/fall time of <1 V/ns.

Timing Characteristics (continued)

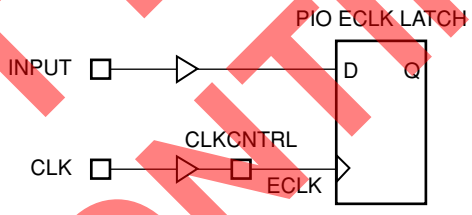
Table 57. OR3C/Txxx Input to ExpressCLK (ECLK) Fast-Capture Setup/Hold Time (Pin-to-Pin) (continued)

OR3Cxx Commercial: V_{DD} = 5.0 V ± 5%, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C < T_A < +85 °C.
OR3Txxx Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C < T_A < +85 °C.

Description (T_J = 85 °C, V_{DD} = min)	Device	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Input to ECLK Hold Time (corner ECLK pin)	OR3T20	—	—	0.00	—	0.00	—	0.00	—	ns
	OR3T30	—	—	0.00	—	0.00	—	0.00	—	ns
	OR3T55	0.00	—	0.00	—	0.80	—	1.10	—	ns
	OR3C/T80	0.00	—	0.00	—	0.00	—	0.00	—	ns
	OR3T125	—	—	0.00	—	0.00	—	0.00	—	ns
Input to ECLK Hold Time (corner ECLK pin, delayed data input)	OR3T20	—	—	0.00	—	0.00	—	0.00	—	ns
	OR3T30	—	—	0.00	—	0.00	—	0.00	—	ns
	OR3T55	0.00	—	0.00	—	0.00	—	0.00	—	ns
	OR3C/T80	0.00	—	0.00	—	0.00	—	0.00	—	ns
	OR3T125	—	—	0.00	—	0.00	—	0.00	—	ns

Notes:
The pin-to-pin timing parameters in this table should be used instead of results reported by ispLEVER.

The ECLK delays are to all of the PIOs on one side of the device for middle pin input, or two sides of the device for corner pin input. The delay includes both the input buffer delay and the clock routing to the PIO clock input.



5-4847(F).b

Figure 79. Input to ExpressCLK Setup/Hold Time

Timing Characteristics (continued)

Table 63. Asynchronous Peripheral Configuration Mode Timing Characteristics

OR3Cxx Commercial: VDD = 5.0 V ± 5%, 0 °C < TA < 70 °C; Industrial: VDD = 5.0 V ± 10%, –40 °C < TA < +85 °C.
OR3Txxx Commercial: VDD = 3.0 V to 3.6 V, 0 °C < TA < 70 °C; Industrial: VDD = 3.0 V to 3.6 V, –40 °C < TA < +85 °C.

Parameter	Symbol	Min	Max	Unit
$\overline{WR}$, $\overline{CS0}$, and CS1 Pulse Width	TWR	50.00	—	ns
D[7:0] Setup Time: 3Cxx 3Txxx	TS	 20.00 10.50	 — —	 ns ns
D[7:0] Hold Time	TH	0.00	—	ns
RDY Delay	TRDY	—	40.00	ns
RDY Low	TB	1.00	8.00	CCLK Periods
Earliest $\overline{WR}$ After RDY Goes High*	TWR2	0.00	—	ns
$\overline{RD}$ to D7 Enable/Disable	TDEN	—	40.00	ns
CCLK to DOUT	TD	—	5.00	ns

* This parameter is valid whether the end of not RDY is determined from the RDY pin or from the D7 pin.

Notes:
Serial data is transmitted out on DOUT on the falling edge of CCLK after the byte is input on D[7:0].
D[6:0] timing is the same as the write data portion of the D7 waveform because D[6:0] are not enabled by $\overline{RD}$.

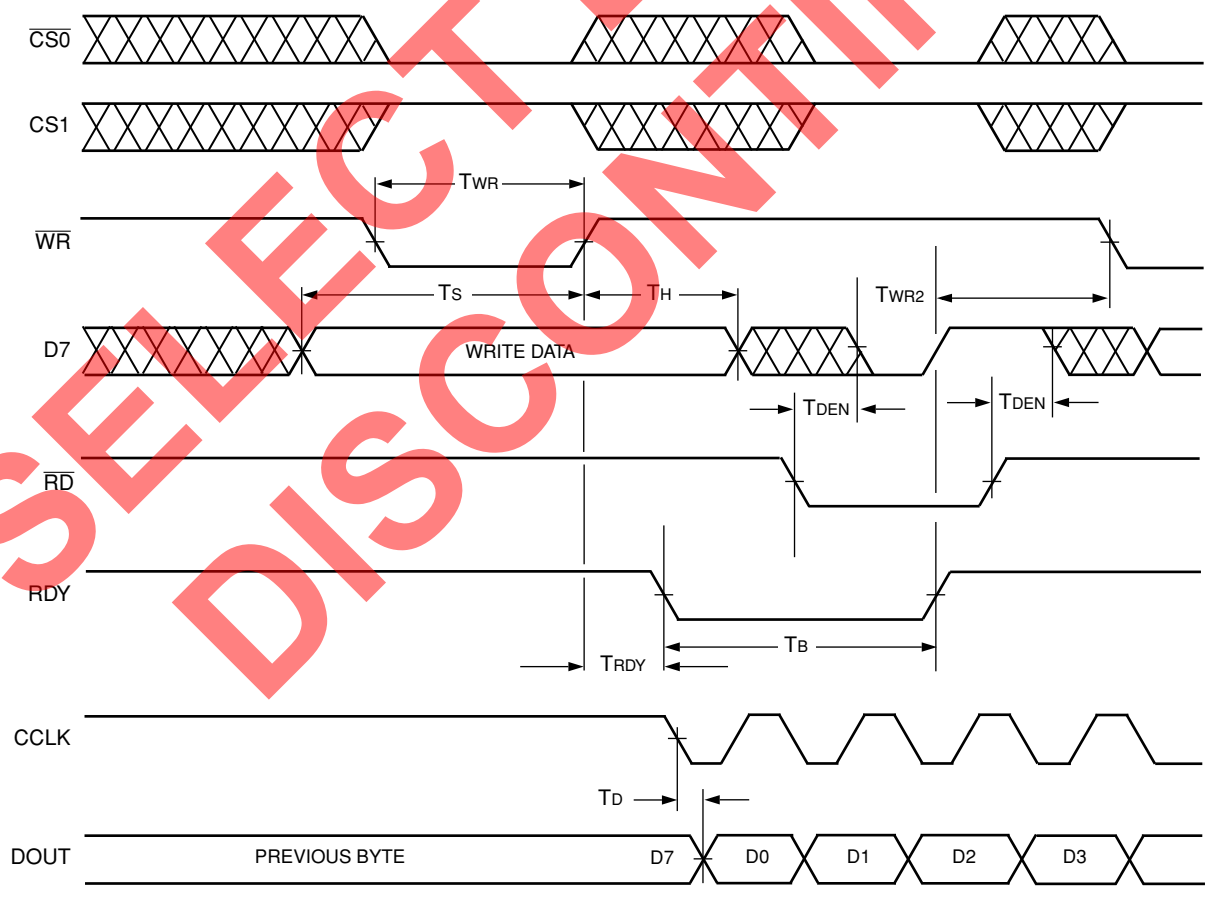


Figure 85. Asynchronous Peripheral Configuration Mode Timing Diagram

Pin Information (continued)

Table 67. Pin Descriptions (continued)

Symbol	I/O	Description
Special-Purpose Pins (continued)		
A11/$\overline{\text{MPI\_IRQ}}$	O	MPI active-low interrupt request output.
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin (see Note).
A10/$\overline{\text{MPI\_BI}}$	O	*PowerPC* mode MPI burst inhibit output.
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin (see Note).
A9/$\overline{\text{MPI\_ACK}}$	O	In *PowerPC* mode MPI operation, this is the active-high transfer acknowledge ($\overline{\text{TA}}$) output. For *i960* MPI operation, it is the active-low ready/record ($\overline{\text{RDYRCV}}$) output.
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin (see Note).
A8/MPI_RW	I	In *PowerPC* mode MPI operation, this is the active-low write/active-high read control signals. For *i960* operation, it is the active-high write/active-low read control signal.
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin (see Note).
A7/MPI_CLK	I	This is the clock used for the synchronous MPI interface. For *PowerPC*, it is the CLKOUT signal. For *i960*, it is the system clock that is chosen for the *i960* external bus interface.
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin (see Note).
A[4:0]	I	For *PowerPC* operation, these are the *PowerPC* address inputs. The address bit mapping (in *PowerPC*/FPGA notation) is A[31]/A[0], A[30]/A[1], A[29]/A[2], A[28]/A[3], A[27]/A[4]. Note that A[27]/A[4] is the MSB of the address. The A[4:2] inputs are not used in *i960* MPI mode.
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin (see Note).
A[1:0]/ $\overline{\text{MPI\_BE[1:0]}}$	I	For *i960* operation, $\overline{\text{MPI\_BE[1:0]}}$ provide the *i960* byte enable signals, $\overline{\text{BE[1:0]}}$, that are used as address bits A[1:0] in *i960* byte-wide operation.
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin (see Note).
D[7:0]	I	During master parallel, peripheral, and slave parallel configuration modes, D[7:0] receive configuration data, and each pin has a pull-up enabled. During serial configuration modes, D0 is the DIN input. D[7:0] are also the data pins for *PowerPC* microprocessor mode and the address/data pins for *i960* microprocessor mode.
	I/O	After configuration, the pins are user-programmable I/O pins (see Note).
DIN	I	During slave serial or master serial configuration modes, DIN accepts serial configuration data synchronous with CCLK. During parallel configuration modes, DIN is the D0 input. During configuration, a pull-up is enabled.
	I/O	After configuration, this pin is a user-programmable I/O pin (see Note).
DOUT	O	During configuration, DOUT is the serial data output that can drive the DIN of daisy-chained slave LCA devices. Data out on DOUT changes on the falling edge of CCLK.
	I/O	After configuration, DOUT is a user-programmable I/O pin (see Note).

Note: The FPGA States of Operation section contains more information on how to control these signals during start-up. The timing of DONE release is controlled by one set of bit stream options, and the timing of the simultaneous release of all other configuration pins (and the activation of all user I/Os) is controlled by a second set of options.

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
128	PR7C	PR8C	PR10C	PR12C	PR15C	I/O
129	PR7D	PR8D	PR10D	PR12D	PR15D	I/O
130	VDD	VDD	VDD	VDD	VDD	VDD
131	PECKR	PECKR	PECKR	PECKR	PECKR	I-ECKR
132	PR6B	PR7B	PR9B	PR11B	PR14B	I/O
133	PR6C	PR7C	PR9C	PR11C	PR14C	I/O
134	PR6D	PR7D	PR9D	PR11D	PR14D	I/O
135	VSS	VSS	VSS	VSS	VSS	VSS
136	PR5A	PR6A	PR8A	PR10A	PR13A	I/O
137	PR5B	PR6B	PR8B	PR10C	PR13D	I/O
138	PR5C	PR6C	PR8C	PR10D	PR12A	I/O
139	PR5D	PR6D	PR8D	PR9B	PR12D	I/O
140	PR4A	PR5A	PR7A	PR9C	PR11A	I/O-CS1
141	PR4B	PR5B	PR7B	PR9D	PR11D	I/O
142	PR4C	PR5C	PR7C	PR8A	PR10A	I/O
143	PR4D	PR5D	PR7D	PR8D	PR10D	I/O
144	VDD	VDD	VDD	VDD	VDD	VDD
145	PR3A	PR4A	PR6A	PR7A	PR9A	I/O-$\overline{CS0}$
146	PR3B	PR4B	PR6B	PR7B	PR9B	I/O
147	PR3C	PR4C	PR5B	PR6B	PR8B	I/O
148	PR3D	PR4D	PR5D	PR6D	PR8D	I/O
149	PR2A	PR3A	PR4A	PR5A	PR7A	I/O-$\overline{RD}$/$\overline{MPI\_STRB}$
150	PR2C	PR3C	PR4D	PR5D	PR5A	I/O
151	PR2D	PR3D	PR3A	PR4A	PR4A	I/O
152	PR1A	PR2A	PR2A	PR3A	PR3A	I/O-$\overline{WR}$
153	PR1C	PR2D	PR2C	PR2A	PR2A	I/O
154	PR1D	PR1A	PR1A	PR1A	PR1A	I/O
155	VSS	VSS	VSS	VSS	VSS	VSS
156	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{RD\_CFG}$
157	VSS	VSS	VSS	VSS	VSS	VSS
158	VSS	VSS	VSS	VSS	VSS	VSS
159	PT12D	PT14D	PT18D	PT22D	PT28D	I/O-SECKUR
160	PT12A	PT13D	PT17D	PT21A	PT27A	I/O-RDY/RCLK/MPI_ALE
161	PT11D	PT13A	PT16D	PT19D	PT25D	I/O
162	PT11C	PT12D	PT16A	PT19A	PT25A	I/O
163	PT11A	PT12C	PT15D	PT18D	PT24D	I/O-D7
164	PT10D	PT12A	PT14D	PT17D	PT23D	I/O
165	PT10C	PT11D	PT14A	PT17A	PT22D	I/O
166	PT10B	PT11C	PT13D	PT16D	PT21D	I/O
167	PT10A	PT11B	PT13B	PT16B	PT20D	I/O-D6
168	VDD	VDD	VDD	VDD	VDD	VDD
169	PT9D	PT10D	PT12D	PT15D	PT19D	I/O
170	PT9C	PT10C	PT12C	PT15B	PT19A	I/O

Table 72. OR3T30, OR3T55, OR3C/T80, and OR3T125 240-Pin SQFP/SQFP2 Pinout

Pin	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
1	VSS	VSS	VSS	VSS	VSS
2	VDD	VDD	VDD	VDD	VDD
3	PL1D	PL1D	PL1D	PL1D	I/O
4	PL1B	PL1C	PL1C	PL1C	I/O
5	PL1A	PL1B	PL1B	PL1B	I/O
6	PL2D	PL2D	PL2D	PL2D	I/O-A0/$\overline{\text{MPI\_BE0}}$
7	VSS	VSS	VSS	VSS	VSS
8	PL3D	PL3D	PL4D	PL4D	I/O
9	PL3C	PL3A	PL4A	PL5D	I/O
10	PL3B	PL4D	PL5D	PL6D	I/O
11	PL3A	PL4A	PL5A	PL7D	I/O-A1/$\overline{\text{MPI\_BE1}}$
12	PL4D	PL5A	PL6A	PL8A	I/O-A2
13	PL4C	PL6D	PL7D	PL9D	I/O
14	PL4B	PL6B	PL7B	PL9B	I/O
15	PL4A	PL6A	PL7A	PL9A	I/O-A3
16	VDD	VDD	VDD	VDD	VDD
17	PL5D	PL7D	PL8D	PL10D	I/O
18	PL5C	PL7C	PL8A	PL10A	I/O
19	PL5B	PL7B	PL9D	PL11D	I/O
20	PL5A	PL7A	PL9B	PL11A	I/O-A4
21	PL6D	PL8D	PL9A	PL12D	I/O-A5
22	PL6C	PL8C	PL10C	PL12A	I/O
23	PL6B	PL8B	PL10B	PL13D	I/O
24	PL6A	PL8A	PL10A	PL13A	I/O-A6
25	VSS	VSS	VSS	VSS	VSS
26	PECKL	PECKL	PECKL	PECKL	I-ECKL
27	PL7C	PL9C	PL11C	PL14C	I/O
28	PL7B	PL9B	PL11B	PL14B	I/O
29	PL7A	PL9A	PL11A	PL14A	I/O-A7/MPI_CLK
30	VDD	VDD	VDD	VDD	VDD
31	PL8D	PL10D	PL12D	PL15D	I/O
32	PL8C	PL10C	PL12C	PL15C	I/O
33	PL8B	PL10B	PL12B	PL15B	I/O
34	PL8A	PL10A	PL12A	PL15A	I/O-A8/MPI_RW
35	VSS	VSS	VSS	VSS	VSS
36	PL9D	PL11D	PL13D	PL16D	I/O-A9/$\overline{\text{MPI\_ACK}}$
37	PL9C	PL11C	PL13B	PL16A	I/O
38	PL9B	PL11B	PL13A	PL17D	I/O
39	PL9A	PL11A	PL14C	PL17A	I/O-A10/$\overline{\text{MPI\_BI}}$
40	PL10D	PL12D	PL14B	PL18D	I/O
41	PL10C	PL12C	PL15C	PL18A	I/O

Pin	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
85	VSS	VSS	VSS	VSS	VSS
86	PB7A	PB9A	PB11A	PB14A	I/O
87	PB7B	PB9B	PB11B	PB14B	I/O
88	PB7C	PB9C	PB11C	PB14C	I/O
89	PB7D	PB9D	PB11D	PB14D	I/O
90	VSS	VSS	VSS	VSS	VSS
91	PECKB	PECKB	PECKB	PECKB	I-ECKB
92	PB8B	PB10B	PB12B	PB15B	I/O
93	PB8C	PB10C	PB12C	PB15C	I/O
94	PB8D	PB10D	PB12D	PB15D	I/O
95	VSS	VSS	VSS	VSS	VSS
96	PB9A	PB11A	PB13A	PB16A	I/O
97	PB9B	PB11B	PB13B	PB16D	I/O
98	PB9C	PB11C	PB13C	PB17A	I/O
99	PB9D	PB11D	PB14A	PB17D	I/O
100	PB10A	PB12A	PB14B	PB18A	I/O-HDC
101	PB10B	PB12B	PB14D	PB18D	I/O
102	PB10C	PB12C	PB15A	PB19A	I/O
103	PB10D	PB12D	PB15D	PB19D	I/O
104	VDD	VDD	VDD	VDD	VDD
105	PB11A	PB13A	PB16A	PB20A	I/O-$\overline{\text{LDC}}$
106	PB11D	PB13D	PB16D	PB21D	I/O
107	PB12A	PB14A	PB17A	PB22A	I/O
108	PB12B	PB14D	PB17D	PB23D	I/O
109	PB12C	PB15A	PB18A	PB24A	I/O-$\overline{\text{INIT}}$
110	PB12D	PB15D	PB18D	PB24D	I/O
111	PB13A	PB16A	PB19A	PB25A	I/O
112	PB13B	PB16D	PB19D	PB25D	I/O
113	—	VSS	VSS	VSS	VSS
114	PB13D	PB17A	PB20A	PB26A	I/O
115	PB14A	PB17D	PB21A	PB27A	I/O
116	PB14B	PB18A	PB21D	PB27D	I/O
117	PB14D	PB18D	PB22D	PB28D	I/O
118	VSS	VSS	VSS	VSS	VSS
119	PDONE	PDONE	PDONE	PDONE	DONE
120	VDD	VDD	VDD	VDD	VDD
121	VSS	VSS	VSS	VSS	VSS
122	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{RESET}}$
123	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PRGM}}$
124	PR14A	PR18A	PR22A	PR28A	I/O-M0
125	PR14D	PR18C	PR22D	PR28D	I/O
126	PR13A	PR18D	PR21A	PR27A	I/O
127	PR13D	PR17B	PR20A	PR26A	I/O

Table 73. OR3T20, OR3T30, and OR3T55 256-Pin PBGA Pinout

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	Function
B1	VDD	VDD	VDD	VDD
C2	PL1D	PL1D	PL1D	I/O
D2	PL1C	PL1B	PL1C	I/O
D3	PL1B	PL1A	PL1B	I/O
E4	PL1A	PL2D	PL2D	I/O-A0/$\overline{\text{MPI\_BE0}}$
C1	—	PL2C	PL2C	I/O
D1	—	PL2B	PL2B	I/O
E3	—	PL2A	PL2A	I/O
E2	PL2D	PL3D	PL3D	I/O
E1	PL2C	PL3C	PL3A	I/O
F3	PL2B	PL3B	PL4D	I/O
G4	PL2A	PL3A	PL4A	I/O-A1/$\overline{\text{MPI\_BE1}}$
F2	—	—	PL5D	I/O
F1	PL3D	PL4D	PL5A	I/O-A2
G3	PL3C	PL4C	PL6D	I/O
G2	PL3B	PL4B	PL6B	I/O
G1	PL3A	PL4A	PL6A	I/O-A3
H3	PL4D	PL5D	PL7D	I/O
H2	PL4C	PL5C	PL7C	I/O
H1	PL4B	PL5B	PL7B	I/O
J4	PL4A	PL5A	PL7A	I/O-A4
J3	PL5D	PL6D	PL8D	I/O-A5
J2	PL5C	PL6C	PL8C	I/O
J1	PL5B	PL6B	PL8B	I/O
K2	PL5A	PL6A	PL8A	I/O-A6
K3	PECKL	PECKL	PECKL	I-ECKL
K1	PL6C	PL7C	PL9C	I/O
L1	PL6B	PL7B	PL9B	I/O
L2	PL6A	PL7A	PL9A	I/O-A7/MPI_CLK
L3	PL7D	PL8D	PL10D	I/O
L4	PL7C	PL8C	PL10C	I/O
M1	PL7B	PL8B	PL10B	I/O
M2	PL7A	PL8A	PL10A	I/O-A8/MPI_RW
M3	PL8D	PL9D	PL11D	I/O-A9/$\overline{\text{MPI\_ACK}}$
M4	PL8C	PL9C	PL11C	I/O
N1	PL8B	PL9B	PL11B	I/O
N2	PL8A	PL9A	PL11A	I/O-A10/$\overline{\text{MPI\_BI}}$
N3	PL9D	PL10D	PL12D	I/O
P1	PL9C	PL10C	PL12C	I/O
P2	PL9B	PL10B	PL12B	I/O
R1	PL9A	PL10A	PL12A	I/O-A11/$\overline{\text{MPI\_IRQ}}$
P3	PL10D	PL11D	PL13D	I/O-A12
R2	PL10C	PL11C	PL13B	I/O
T1	PL10B	PL11B	PL14D	I/O

Pin	OR3C/T80 Pad	OR3T125 Pad	Function
AC30	PL18D	PL22D	I/O
AC31	PL17A	PL21A	I/O
AB29	PL17B	PL21B	I/O-A13
AB30	PL17C	PL21C	I/O
AB31	PL17D	PL21D	I/O
AA29	PL16A	PL20A	I/O
Y28	PL16B	PL20B	I/O
AA30	PL16C	PL20C	I/O
AA31	PL16D	PL20D	I/O-A12
Y29	PL15A	PL19A	I/O-A11/$\overline{\text{MPI\_IRQ}}$
W28	PL15B	PL19D	I/O
Y30	PL15C	PL18A	I/O
W29	PL14A	PL18C	I/O
W30	PL14B	PL18D	I/O
V28	PL14C	PL17A	I/O-A10/$\overline{\text{MPI\_BI}}$
W31	PL14D	PL17C	I/O
V29	PL13A	PL17D	I/O
V30	PL13B	PL16A	I/O
V31	PL13C	PL16C	I/O
U29	PL13D	PL16D	I/O-A9/$\overline{\text{MPI\_ACK}}$
U30	PL12A	PL15A	I/O-A8/MPI_RW
U31	PL12B	PL15B	I/O
T30	PL12C	PL15C	I/O
T28	PL12D	PL15D	I/O
T29	PL11A	PL14A	I/O-A7/MPI_CLK
R31	PL11B	PL14B	I/O
R30	PL11C	PL14C	I/O
R29	PECKL	PECKL	I-ECKL
P31	PL10A	PL13A	I/O-A6
P30	PL10B	PL13D	I/O
P29	PL10C	PL12A	I/O
N31	PL10D	PL12C	I/O
P28	PL9A	PL12D	I/O-A5
N30	PL9B	PL11A	I/O-A4
N29	PL9C	PL11C	I/O
M30	PL9D	PL11D	I/O
N28	PL8A	PL10A	I/O
M29	PL8C	PL10C	I/O
L31	PL8D	PL10D	I/O
L30	PL7A	PL9A	I/O-A3
M28	PL7B	PL9B	I/O
L29	PL7C	PL9C	I/O
K31	PL7D	PL9D	I/O
K30	PL6A	PL8A	I/O-A2
K29	PL6B	PL8B	I/O
J31	PL6C	PL8C	I/O
J30	PL6D	PL8D	I/O
K28	PL5A	PL7D	I/O-A1/$\overline{\text{MPI\_BE1}}$
J29	PL5B	PL6B	I/O
H30	PL5C	PL6C	I/O
H29	PL5D	PL6D	I/O
J28	PL4A	PL5D	I/O
G31	PL4B	PL4B	I/O
G30	PL4C	PL4C	I/O
G29	PL4D	PL4D	I/O
H28	PL3A	PL3A	I/O
F31	PL3B	PL3B	I/O
F30	PL3C	PL3C	I/O
F29	PL3D	PL3D	I/O
E31	PL2A	PL2A	I/O
E30	PL2B	PL2B	I/O
E29	PL2C	PL2C	I/O
F28	PL2D	PL2D	I/O-A0/$\overline{\text{MPI\_BE0}}$
D31	PL1A	PL1A	I/O
D30	PL1B	PL1B	I/O
D29	PL1C	PL1C	I/O
E28	PL1D	PL1D	I/O
D27	PRD_DATA	PRD_DATA	RD_DATA/TDO
C28	PT1A	PT1A	I/O-TCK
B28	PT1B	PT1B	I/O
A28	PT1C	PT1C	I/O
D26	PT1D	PT1D	I/O
C27	PT2A	PT2A	I/O
B27	PT2B	PT2B	I/O
A27	PT2C	PT2C	I/O
C26	PT2D	PT2D	I/O
B26	PT3A	PT3A	I/O
A26	PT3B	PT3B	I/O
D24	PT3C	PT3C	I/O
C25	PT3D	PT3D	I/O
B25	PT4A	PT4A	I/O-TMS
A25	PT4B	PT4B	I/O
D23	PT4C	PT4C	I/O
C24	PT4D	PT4D	I/O
B24	PT5A	PT5A	I/O
C23	PT5B	PT5B	I/O
D22	PT5C	PT5C	I/O
B23	PT5D	PT5D	I/O
A23	PT6A	PT6A	I/O-TDI
C22	PT6B	PT6D	I/O
B22	PT6C	PT7A	I/O
A22	PT6D	PT7D	I/O
C21	PT7A	PT8A	I/O
D20	PT7B	PT8D	I/O
B21	PT7C	PT9A	I/O
A21	PT7D	PT9D	I/O
C20	PT8A	PT10A	I/O-DOUT
D19	PT8C	PT10D	I/O
B20	PT8D	PT11A	I/O
C19	PT9A	PT11C	I/O

Pin	OR3C/T80 Pad	OR3T125 Pad	Function
AA28	VDD	VDD	VDD
AA4	VDD	VDD	VDD
AE28	VDD	VDD	VDD
AE4	VDD	VDD	VDD
AH11	VDD	VDD	VDD
AH15	VDD	VDD	VDD
AH17	VDD	VDD	VDD
AH21	VDD	VDD	VDD
AH25	VDD	VDD	VDD
AH28	VDD	VDD	VDD
AH4	VDD	VDD	VDD
AH7	VDD	VDD	VDD
AJ29	VDD	VDD	VDD
AJ3	VDD	VDD	VDD
AK2	VDD	VDD	VDD
AK30	VDD	VDD	VDD
AL1	VDD	VDD	VDD
AL31	VDD	VDD	VDD
B2	VDD	VDD	VDD
B30	VDD	VDD	VDD
C29	VDD	VDD	VDD
C3	VDD	VDD	VDD
D11	VDD	VDD	VDD
D15	VDD	VDD	VDD
D17	VDD	VDD	VDD
D21	VDD	VDD	VDD
D25	VDD	VDD	VDD
D28	VDD	VDD	VDD
D4	VDD	VDD	VDD
D7	VDD	VDD	VDD
G28	VDD	VDD	VDD
G4	VDD	VDD	VDD
L28	VDD	VDD	VDD
L4	VDD	VDD	VDD
R28	VDD	VDD	VDD
R4	VDD	VDD	VDD
U28	VDD	VDD	VDD
U4	VDD	VDD	VDD