ORCA™ Series 3C and 3T FPGA Device Datasheet

June 2010

Select Devices Discontinued!

Product Change Notifications (PCNs) have been issued to discontinue select devices in this data sheet.

The original datasheet pages have not been modified and do not reflect those changes. Please refer to the table below for reference PCN and current product status.

Product Line	Ordering Part Number	Product Status	Reference PCN
OR3C80	OR3C805PS208-DB	**Discontinued**	PCN#02-06
	OR3C804PS208-DB		
	OR3C804PS208I-DB		
	OR3C804BA352-DB		
OR3T20	OR3T206T144-DB	**Discontinued**	PCN#09-10
	OR3T207S208-DB		
	OR3T206S208-DB		
	OR3T206S208I-DB		
	OR3T207BA256-DB		
	OR3T206BA256-DB		
OR3T30	OR3T307S208-DB	Active / Orderable	
	OR3T306S208-DB		
	OR3T306S208I-DB		
	OR3T307S240-DB	**Discontinued**	PCN#12A-09
	OR3T306S240-DB		
	OR3T306S240I-DB		
	OR3T307BA256-DB	Active / Orderable	
	OR3T306BA256-DB		
	OR3T306BA256I-DB		
OR3T55	OR3T557S208-DB	Active / Orderable	
	OR3T556S208-DB		
	OR3T556S208I-DB		
	OR3T557PS240-DB	**Discontinued**	PCN#06-07
	OR3T556PS240-DB		
	OR3T556PS240I-DB		

5555 N.E. Moore Ct. ● Hillsboro, Oregon 97124-6421 ● Phone (503) 268-8000 ● FAX (503) 268-8347
Internet: http://www.latticesemi.com

Product Line	Ordering Part Number	Product Status	Reference PCN
OR3T55 (Cont'd)	OR3T557BA256-DB	Active / Orderable	
	OR3T556BA256-DB		
	OR3T556BA256I-DB		
	OR3T557BA352-DB	**Discontinued**	PCN#09-10
	OR3T556BA352-DB		
	OR3T556BA352I-DB		
OR3T80	OR3T807S208-DB	**Discontinued**	PCN#09-10
	OR3T806S208-DB		
	OR3T806S208I-DB		
	OR3T807PS240-DB	**Discontinued**	PCN#06-07
	OR3T806PS240-DB		
	OR3T806PS240I-DB		
	OR3T807BA352-DB	**Discontinued**	PCN#09-10
	OR3T806BA352-DB		
	OR3T806BA352I-DB		
	OR3T807BC432-DB		
	OR3T806BC432-DB		
	OR3T806BC432I-DB		
OR3T125	OR3T1257PS208-DB	**Discontinued**	PCN#06-07
	OR3T1256PS208-DB		
	OR3T1256PS208I-DB		
	OR3T1257PS240-DB		
	OR3T1256PS240-DB		
	OR3T1256PS240I-DB		
	OR3T1257BA352-DB		PCN#09-10
	OR3T1256BA352-DB		
	OR3T1256BA352I-DB		
	OR3T1257BC432-DB		
	OR3T1256BC432-DB		
	OR3T1256BC432I-DB		

5555 N.E. Moore Ct. ● Hillsboro, Oregon 97124-6421 ● Phone (503) 268-8000 ● FAX (503) 268-8347
Internet: http://www.latticesemi.com

Data Sheet
November 2006

ORCA® Series 3C and 3T Field-Programmable Gate Arrays

Features

- High-performance, cost-effective, 0.35 µm (OR3C) and 0.3 µm (OR3T) 4-level metal technology, (4- or 5-input look-up table delay of 1.1 ns with -7 speed grade in 0.3 µm).
- Same basic architecture as lower-voltage, advanced process technology Series 3 architectures. (See *ORCA* Series 3L FPGA documentation.)
- Up to 186,000 usable gates.
- Up to 342 user I/Os. (OR3Txxx I/Os are 5 V tolerant to allow interconnection to both 3.3 V and 5 V devices, selectable on a per-pin basis.)
- Pin selectable I/O clamping diodes provide 5 V or 3.3 V PCI compliance and 5 V tolerance on OR3Txxx devices.
- Twin-quad programmable function unit (PFU) architecture with eight 16-bit look-up tables (LUTs) per PFU, organized in two nibbles for use in nibble- or byte-wide functions. Allows for mixed arithmetic and logic functions in a single PFU.
- Nine user registers per PFU, one following each LUT, plus one extra. All have programmable clock enable and local set/reset, plus a global set/reset that can be disabled per PFU.
- Flexible input structure (FINS) of the PFUs provides a routability enhancement for LUTs with shared inputs and the logic flexibility of LUTs with independent inputs.
- Fast-carry logic and routing to adjacent PFUs for nibble-, byte-wide, or longer arithmetic functions, with the option to register the PFU carry-out.
- Softwired LUTs (SWL) allow fast cascading of up to three levels of LUT logic in a single PFU for up to 40% speed improvement.
- Supplemental logic and interconnect cell (SLIC) provides 3-statable buffers, up to 10-bit decoder, and *PAL**-like AND-OR with optional INVERT in each programmable logic cell (PLC), with over 50% speed improvement typical.
- Abundant hierarchical routing resources based on routing two data nibbles and two control lines per set provide for faster place and route implementations and less routing delay.
- TTL or CMOS input levels programmable per pin for the OR3Cxx (5.0 V) devices.
- Individually programmable drive capability: 12 mA sink/6 mA source or 6 mA sink/3 mA source.
- Built-in boundary scan (*IEEE*†1149.1 JTAG) and TS_ALL testability function to 3-state all I/O pins.
- Enhanced system clock routing for low skew, high-speed clocks originating on-chip or at any I/O.
- Up to four ExpressCLK inputs allow extremely fast clocking of signals on- and off-chip plus access to internal general clock routing.
- StopCLK feature to glitchlessly stop/start ExpressCLKs independently by user command.
- Programmable I/O (PIO) has:
  - — Fast-capture input latch and input flip-flop (FF) latch for reduced input setup time and zero hold time.
  - — Capability to (de)multiplex I/O signals.
  - — Fast access to SLIC for decodes and *PAL*-like functions.
  - — Output FF and two-signal function generator to reduce CLK to output propagation delay.
  - — Fast open-drain dive capability
  - — Capability to register 3-state enable signal.
- Baseline FPGA family used in Series 3+ FPSCs (field programmable system chips) which combine FPGA logic and standard cell logic on one device.

* *PAL* is a trademark of Advanced Micro Devices, Inc.
† *IEEE* is a registered trademark of The Institute of Electrical and Electronics Engineers, Inc.

Table 1. *ORCA* Series 3 (3C and 3T) FPGAs

Device	System Gates‡	LUTs	Registers	Max User RAM	Max User I/Os	Array Size	Process Technology
OR3T20	36K	1152	1872	18K	192	12 x 12	0.3 µm/4 LM
OR3T30	48K	1568	2436	25K	221	14 x 14	0.3 µm/4 LM
OR3T55	80K	2592	3780	42K	288	18 x 18	0.3 µm/4 LM
OR3C/3T80	116K	3872	5412	62K	342	22 x 22	0.3 µm/4 LM
OR3T125	186K	6272	8400	100K	342	28 x 28	0.3 µm/4 LM

‡ The system gate counts range from a logic-only gate count to a gate count assuming 30% of the PFUs/SLICs being used as RAMs. The logic-only gate count includes each PFU/SLIC (counted as 108 gates per PFU/SLIC), including 12 gates per LUT/FF pair (eight per PFU), and 12 gates per SLIC/FF pair (one per PFU). Each of the four PIOs per PIC is counted as 16 gates (two FFs, fast-capture latch, output logic, CLK drivers, and I/O buffers). PFUs used as RAM are counted at four gates per bit, with each PFU capable of implementing a 32 x 4 RAM (or 512 gates) per PFU.

Table of Contents

Contents	Page
Features	1
System-Level Features	4
Description	5
FPGA Overview	5
PLC Logic	5
Description (continued)	6
PIC Logic	6
System Features	6
Routing	6
Configuration	6
Description (continued)	7
ispLEVER Development System	7
Architecture	7
Programmable Logic Cells	9
Programmable Function Unit	9
Look-Up Table Operating Modes	11
Supplemental Logic and Interconnect Cell (SLIC)	19
PLC Latches/Flip-Flops	23
PLC Routing Resources	25
PLC Architectural Description	32
rogrammable Input/Output Cells	34
5 V Tolerant I/O	35
PCI Compliant I/O	35
Inputs	36
Outputs	39
PIC Routing Resources	42
PIC Architectural Description	43
High-Level Routing Resources	45
Interquad Routing	45
Programmable Corner Cell Routing	46
PIC Interquad (MID) Routing	47
Clock Distribution Network	48
PFU Clock Sources	48
Clock Distribution in the PLC Array	49
Clock Sources to the PLC Array	50
Clocks in the PICs	50
ExpressCLK Inputs	51
Selecting Clock Input Pins	51
Special Function Blocks	52
Single Function Blocks	52
Boundary Scan	55
Microprocessor Interface (MPI)	62
PowerPC System	63
i960 System	64
MPI Interface to FPGA	65
MPI Setup and Control	66
Programmable Clock Manager (PCM)	70
PCM Registers	71
Delay-Locked Loop (DLL) Mode	73
Phase-Locked Loop (PLL) Mode	74
PCM/FPGA Internal Interface	77
PCM Operation	77
PCM Detailed Programming	78
PCM Applications	81
PCM Cautions	82
FPGA States of Operation	83
Initialization	83
Configuration	84
Start-Up	85
Reconfiguration	86
Partial Reconfiguration	86
Other Configuration Options	86
Using ispLEVER to Generate Configuration RAM Data	87
Configuration Data Frame	87
Bit Stream Error Checking	89
FPGA Configuration Modes	90
Master Parallel Mode	90
Master Serial Mode	91
Asynchronous Peripheral Mode	92
Microprocessor Interface (MPI) Mode	92
Slave Serial Mode	95
Slave Parallel Mode	95
Daisy-Chaining	96
Daisy-Chaining with Boundary Scan	97
Absolute Maximum Ratings	98
Recommended Operating Conditions	98
Electrical Characteristics	99
Timing Characteristic Description	101
Description	101
PFU Timing	102
PLC Timing	109
SLIC Timing	109
PIO Timing	110
Special Function Blocks Timing	113
Clock Timing	121
Configuration Timing	131
Readback Timing	140
Input/Output Buffer Measurement Conditions	141
Output Buffer Characteristics	142
OR3Cxx	142
OR3Txxx	143
Estimating Power Dissipation	144
OR3Cxx	144
OR3Txxx	145
Pin Information	147
Pin Descriptions	147
Package Compatibility	151
Compatibility with OR2C/TxxA Series	152
Package Thermal Characteristics	188
FPGA Maximum Junction Temperature	190
Package Coplanarity	191
Package Parasitics	191
Package Outline Diagrams	192

Table of Contents

Contents	Page	Contents	Page
Terms and Definitions	192		
144-Pin TQFP	193		
208-Pin SQFP	194		
208-Pin SQFP2	195		
240-Pin SQFP	196		
240-Pin SQFP2	197		
256-Pin PBGA	198		
352-Pin PBGA	199		
432-Pin EBGA	200		
Ordering Information	201		

System-Level Features

System-level features reduce glue logic requirements and make a system on a chip possible. These features in the *ORCA* Series 3 include:

- Full PCI local bus compliance.
- Dual-use microprocessor interface (MPI) can be used for configuration, readback, device control, and device status, as well as for a general-purpose interface to the FPGA. Glueless interface to *i960** and *PowerPC*† processors with user-configurable address space provided.
- Parallel readback of configuration data capability with the built-in microprocessor interface.
- Programmable clock manager (PCM) adjusts clock phase and duty cycle for input clock rates from 5 MHz to 120 MHz. The PCM may be combined with FPGA logic to create complex functions, such as digital phase-locked loops (DPLL), frequency counters, and frequency synthesizers or clock doublers. Two PCMs are provided per device.
- True, internal, 3-state, bidirectional buses with simple control provided by the SLIC.
- 32 x 4 RAM per PFU, configurable as single- or dual-port at >176 MHz. Create large, fast RAM/ROM blocks (128 x 8 in only eight PFUs) using the SLIC decoders as bank drivers.

* *i960* is a registered trademark of Intel Corporation.
† *PowerPC* is a registered trademark of International Business Machines Corporation.

Table 2. *ORCA* Series 3 System Performance

Parameter	# PFUs	Speed				Unit
		-4	-5	-6	-7	
16-bit Loadable Up/Down Counter	2	78	102	131	168	MHz
16-bit Accumulator	2	78	102	131	168	MHz
8 x 8 Parallel Multiplier:						
Multiplier Mode, Unpipelined[1]	11.5	19	25	30	38	MHz
ROM Mode, Unpipelined[2]	8	51	66	80	102	MHz
Multiplier Mode, Pipelined[3]	15	76	104	127	166	MHz
32 x 16 RAM (synchronous):						
Single-port, 3-state Bus[4]	4	97	127	151	192	MHz
Dual-port[5]	4	127	166	203	253	MHz
128 x 8 RAM (synchronous):						
Single-port, 3-state Bus[4]	8	88	116	139	176	MHz
Dual-port[5]	8	88	116	139	176	MHz
8-bit Address Decode (internal):						
Using Softwired LUTs	0.25	4.87	3.66	2.58	2.03	ns
Using SLICs[6]	0	2.35	1.82	1.23	0.99	ns
32-bit Address Decode (internal):						
Using Softwired LUTs	2	16.06	12.07	9.01	7.03	ns
Using SLICs[7]	0	6.91	5.41	4.21	3.37	ns
36-bit Parity Check (internal)	2	16.06	12.07	9.01	7.03	ns

1. Implemented using 8 x 1 multiplier mode (unpipelined), register-to-register, two 8-bit inputs, one 16-bit output.
2. Implemented using two 32 x 12 ROMs and one 12-bit adder, one 8-bit input, one fixed operand, one 16-bit output.
3. Implemented using 8 x 1 multiplier mode (fully pipelined), two 8-bit inputs, one 16-bit output (7 of 15 PFUs contain only pipelining registers).
4. Implemented using 32 x 4 RAM mode with read data on 3-state buffer to bidirectional read/write bus.
5. Implemented using 32 x 4 dual-port RAM mode.
6. Implemented in one partially occupied SLIC with decoded output set up to CE in same PLC.
7. Implemented in five partially occupied SLICs.

Description

FPGA Overview

The *ORCA* Series 3 FPGAs are a new generation of SRAM-based FPGAs built on the successful OR2C/TxxA FPGA Series, with enhancements and innovations geared toward today's high-speed designs and tomorrow's systems on a single chip. Designed from the start to be synthesis friendly and to reduce place and route times while maintaining the complete routability of the *ORCA* 2C/2T devices, Series 3 more than doubles the logic available in each logic block and incorporates system-level features that can further reduce logic requirements and increase system speed. *ORCA* Series 3 devices contain many new patented enhancements and are offered in a variety of packages, speed grades, and temperature ranges.

The *ORCA* Series 3 FPGAs consist of three basic elements: programmable logic cells (PLCs), programmable input/output cells (PICs), and system-level features. An array of PLCs is surrounded by PICs. Each PLC contains a programmable function unit (PFU), a supplemental logic and interconnect cell (SLIC), local routing resources, and configuration RAM. Most of the FPGA logic is performed in the PFU, but decoders, *PAL*-like functions, and 3-state buffering can be performed in the SLIC. The PICs provide device inputs and outputs and can be used to register signals and to perform input demultiplexing, output multiplexing, and other functions on two output signals. Some of the system-level functions include the new microprocessor interface (MPI) and the programmable clock manager (PCM).

PLC Logic

Each PFU within a PLC contains eight 4-input (16-bit) look-up tables (LUTs), eight latches/flip-flops (FFs), and one additional flip-flop that may be used independently or with arithmetic functions.

The PFU is organized in a twin-quad fashion: two sets of four LUTs and FFs that can be controlled independently. LUTs may also be combined for use in arithmetic functions using fast-carry chain logic in either 4-bit or 8-bit modes. The carry-out of either mode may be registered in the ninth FF for pipelining. Each PFU may also be configured as a synchronous 32 x 4 single- or dual-port RAM or ROM. The FFs (or latches) may obtain input from LUT outputs or directly from invertible PFU inputs, or they can be tied high or tied low. The FFs also have programmable clock polarity, clock enables, and local set/reset.

The SLIC is connected to PLC routing resources and to the outputs of the PFU. It contains 3-state, bidirectional buffers and logic to perform up to a 10-bit AND function for decoding, or an AND-OR with optional INVERT (AOI) to perform *PAL*-like functions. The 3-state drivers in the SLIC and their direct connections to the PFU outputs make fast, true 3-state buses possible within the FPGA, reducing required routing and allowing for real-world system performance.

Description (continued)

PIC Logic

Series 3 PIC addresses the demand for ever-increasing system clock speeds. Each PIC contains four programmable inputs/outputs (PIOs) and routing resources. On the input side, each PIO contains a fast-capture latch that is clocked by an ExpressCLK. This latch is followed by a latch/FF that is clocked by a system clock from the internal general clock routing. The combination provides for very low setup requirements and zero hold times for signals coming on-chip. It may also be used to demultiplex an input signal, such as a multiplexed address/data signal, and register the signals without explicitly building a demultiplexer. Two input signals are available to the PLC array from each PIO, and the *ORCA* 2C/2T capability to use any input pin as a clock or other global input is maintained.

On the output side of each PIO, two outputs from the PLC array can be routed to each output flip-flop, and logic can be associated with each I/O pad. The output logic associated with each pad allows for multiplexing of output signals and other functions of two output signals.

The output FF in combination with output signal multiplexing, is particularly useful for registering address signals to be multiplexed with data, allowing a full clock cycle for the data to propagate to the output. The I/O buffer associated with each pad is very similar to the *ORCA* 2C/2T Series buffer with a new, fast, open-drain option for ease of use on system buses.

System Features

Series 3 also provides system-level functionality by means of its dual-use microprocessor interface and its innovative programmable clock manager. These functional blocks allow for easy glueless system interfacing and the capability to adjust to varying conditions in today's high-speed systems.

Routing

The abundant routing resources of the *ORCA* Series 3 FPGAs are organized to route signals individually or as buses with related control signals. Clocks are routed on a low-skew, high-speed distribution network and may be sourced from PLC logic, externally from any I/O pad, or from the very fast ExpressCLK pins. ExpressCLKs may be glitchlessly and independently enabled and disabled with a programmable control signal using the new StopCLK feature. The improved PIC routing resources are now similar to the patented intra-PLC routing resources and provide great flexibility in moving signals to and from the PIOs. This flexibility translates into an improved capability to route designs at the required speeds when the I/O signals have been locked to specific pins.

Configuration

The FPGA's functionality is determined by internal configuration RAM. The FPGA's internal initialization/configuration circuitry loads the configuration data at powerup or under system control. The RAM is loaded by using one of several configuration modes. The configuration data resides externally in an EEPROM or any other storage media. Serial EEPROMs provide a simple, low pin count method for configuring FPGAs. A new, easy method for configuring the devices is through the microprocessor interface.

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.

Architecture (continued)

5-4489(F)

Figure 1. OR3T55 Array

Programmable Logic Cells

The programmable logic cell (PLC) consists of a programmable function unit (PFU), a supplemental logic and interconnect cell (SLIC), and routing resources. All PLCs in the array are functionally identical with only minor differences in routing connectivity for improved routability. The PFU, which contains eight 4-input LUTs, eight latches/FFs, and one FF for logic implementation, is discussed in the next section, followed by discussions of the SLIC and PLC routing resources.

Programmable Function Unit

The PFUs are used for logic. Each PFU has 50 external inputs and 18 outputs and can operate in several modes. The functionality of the inputs and outputs depends on the operating mode.

The PFU uses 36 data input lines for the LUTs, eight data input lines for the latches/FFs, five control inputs (ASWE, CLK, CE, LSR, SEL), and a carry input (CIN) for fast arithmetic functions and general-purpose data input for the ninth FF. There are eight combinatorial data outputs (one from each LUT), eight latched/registered outputs (one from each latch/FF), a carry-out (COUT), and a registered carry-out (REGCOUT) that comes from the ninth FF. The carry-out signals are used principally for fast arithmetic functions.

Figure 2 and Figure 3 show high-level and detailed views of the ports in the PFU, respectively. The eight sets of LUT inputs are labeled as K0 through K7 with each of the four inputs to each LUT having a suffix of _x, where x is a number from 0 to 3. There are four F5 inputs labeled A through D. These inputs are used for a fifth LUT input for 5-input LUTs or as a selector for multiplexing two 4-input LUTs. The eight direct data inputs to the latches/FFs are labeled as DIN[7:0]. Registered LUT outputs are shown as Q[7:0], and combinatorial LUT outputs are labeled as F[7:0].

The PFU implements combinatorial logic in the LUTs and sequential logic in the latches/FFs. The LUTs are static random access memory (SRAM) and can be used for read/write or read-only memory.

Each latch/FF can accept data from its associated LUT. Alternatively, the latches/FFs can accept direct data from DIN[7:0], eliminating the LUT delay if no combinatorial function is needed. Additionally, the CIN input can be used as a direct data source for the ninth FF. The LUT outputs can bypass the latches/FFs, which reduces the delay out of the PFU. It is possible to use the LUTs and latches/FFs more or less independently, allowing, for instance, a comparator function in the LUTs simultaneously with a shift register in the FFs.

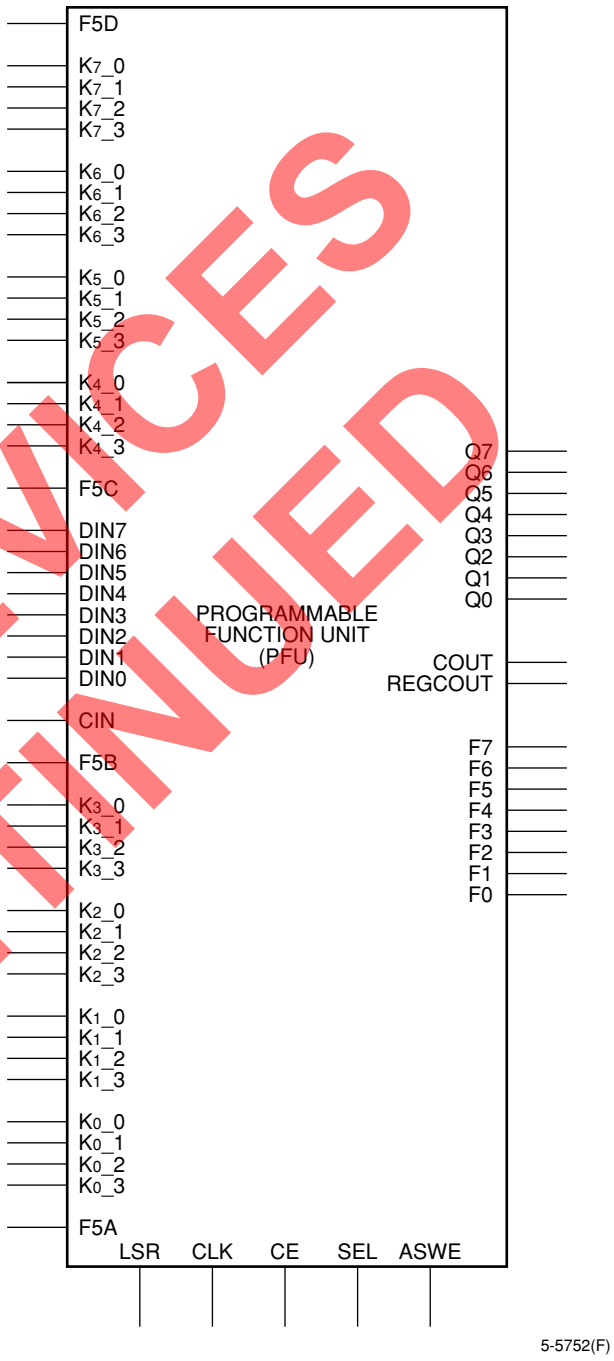
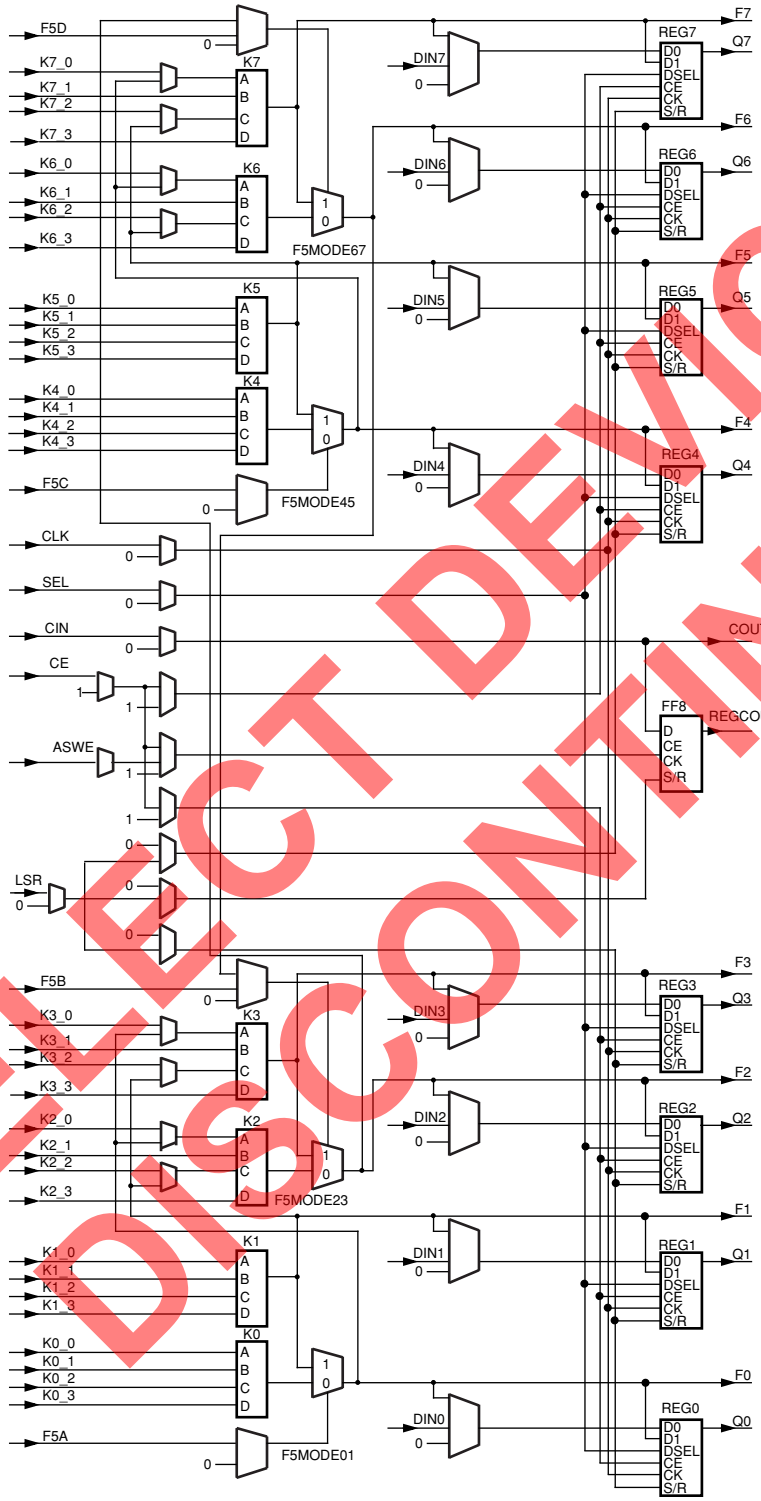


Figure 2. PFU Ports

The PFU can be configured to operate in four modes: logic mode, half-logic mode, ripple mode, and memory (RAM/ROM) mode. In addition, ripple mode has four submodes and RAM mode can be used in either a single- or dual-port memory fashion. These submodes of operation are discussed in the following sections.

Programmable Logic Cells (continued)



5-5743(F)

Note: All multiplexers without select inputs are configuration selector multiplexers.

Figure 3. Simplified PFU Diagram

Programmable Logic Cells (continued)

Look-Up Table Operating Modes

The operating mode affects the functionality of the PFU input and output ports and internal PFU routing. For example, in some operating modes, the DIN[7:0] inputs are direct data inputs to the PFU latches/FFs. In memory mode, the same DIN[7:0] inputs are used as a 4-bit write data input bus and a 4-bit write address input bus into LUT memory.

Table 3 lists the basic operating modes of the LUT. Figure 4—Figure 10 show block diagrams of the LUT operating modes. The accompanying descriptions demonstrate each mode's use for generating logic.

Table 3. Look-Up Table Operating Modes

Mode	Function
Logic	4- and 5-input LUTs; softwired LUTs; latches/FFs with direct input or LUT input; CIN as direct input to ninth FF or as pass through to COUT.
Half Logic/ Half Ripple	Upper four LUTs and latches/FFs in logic mode; lower four LUTs and latches/FFs in ripple mode; CIN and ninth FF for logic or ripple functions.
Ripple	All LUTs combined to perform ripple-through data functions. Eight LUT registers available for direct-in use or to register ripple output. Ninth FF dedicated to ripple out, if used. The submodes of ripple mode are adder/subtractor, counter, multiplier, and comparator.
Memory	All LUTs and latches/FFs used to create a 32 x 4 synchronous dual-port RAM. Can be used as single-port or as ROM.

PFU Control Inputs

Each PFU has five routable control inputs and an active-low, asynchronous global set/reset (GSRN) signal that affects all latches and FFs in the device. The five control inputs are CLK, LSR, CE, ASWE, and SEL, and their functionality for each logic mode of the PFU (discussed subsequently) is shown in Table 4. The clock signal to the PFU is CLK, CE stands for clock enable, which is its primary function. LSR is the local set/reset signal that can be configured as synchronous or asynchronous. The selection of set or reset is made for each latch/FF and is not a function of the signal itself. ASWE stands for add/subtract/write enable, which are its functions, along with being an optional clock enable, and SEL is used to dynamically select between direct PFU input and LUT output data as the input to the latches/FFs.

All of the control signals can be disabled and/or inverted via the configuration logic. A disabled clock enable indicates that the clock is always enabled. A disabled LSR indicates that the latch/FF never sets/resets (except from GSRN). A disabled SEL input indicates that DIN[7:0] PFU inputs are routed to the latches/FFs. For logic and ripple modes of the PFU, the LSR, CE, and ASWE (as a clock enable) inputs can be disabled individually for each nibble (latch/FF[3:0], latch/FF[7:4]) and for the ninth FF.

Programmable Logic Cells (continued)

Table 4. Control Input Functionality

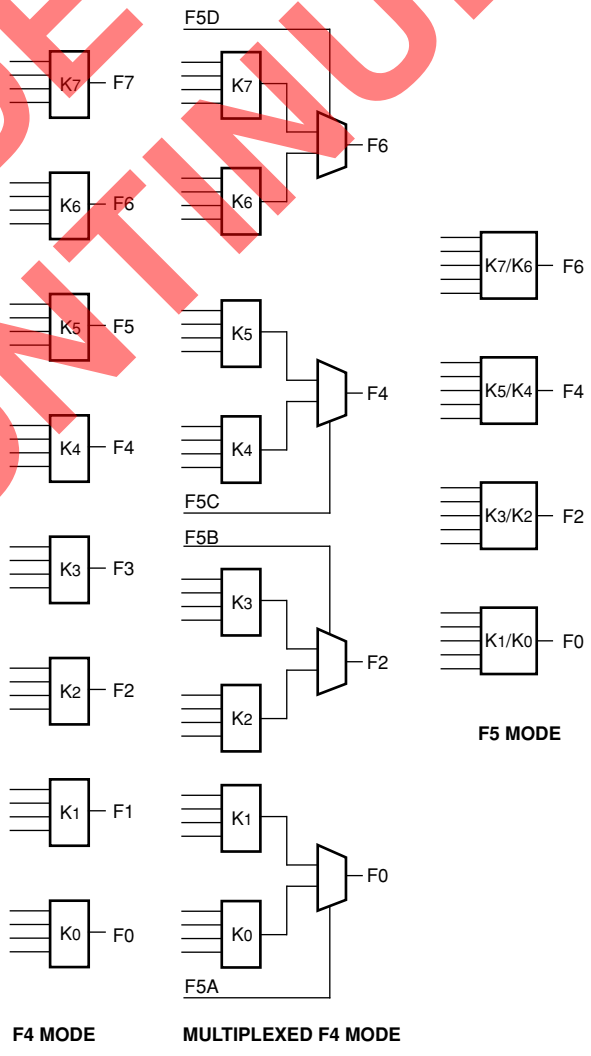
Mode	CLK	LSR	CE	ASWE	SEL
Logic	CLK to all latches/ FFs	LSR to all latches/ FFs, enabled per nibble and for ninth FF	CE to all latches/FFs, selectable per nibble and for ninth FF	CE to all latches/FFs, selectable per nibble and for ninth FF	Select between LUT input and direct input for eight latches/FFs
Half Logic/ Half Ripple	CLK to all latches/ FFs	LSR to all latches/FF, enabled per nibble and for ninth FF	CE to all latches/FFs, selectable per nibble and for ninth FF	Ripple logic control input	Select between LUT input and direct input for eight latches/FFs
Ripple	CLK to all latches/ FFs	LSR to all latches/ FFs, enabled per nibble and for ninth FF	CE to all latches/FFs, selectable per nibble and for ninth FF	Ripple logic control input	Select between LUT input and direct input for eight latches/FFs
Memory (RAM)	CLK to RAM	Port enable 2	Port enable 1	Write enable	Not used
Memory (ROM)	Optional for sync. outputs	Not used	Not used	Not used	Not used

Logic Mode

The PFU diagram of Figure 3 represents the logic mode of operation. In logic mode, the eight LUTs are used individually or in flexible groups to implement user logic functions. The latches/FFs may be used in conjunction with the LUTs or separately with the direct PFU data inputs. There are three basic submodes of LUT operation in PFU logic mode: F4 mode, F5 mode, and softwired LUT (SWL) mode. Combinations of these submodes are possible in each PFU.

F4 mode, shown simplified in Figure 4, illustrates the uses of the basic 4-input LUTs in the PFU. The output of an F4 LUT can be passed out of the PFU, captured at the LUTs associated latch/FF, or multiplexed with the adjacent F4 LUT output using one of the F5[A:D] inputs to the PFU. Only adjacent LUT pairs (K_0 and K_1, K_2 and K_3, K_4 and K_5, K_6 and K_7) can be multiplexed, and the output always goes to the even-numbered output of the pair.

The F5 submode of the LUT operation, shown simplified in Figure 4, indicates the use of 5-input LUTs to implement logic. 5-input LUTs are created from two 4-input LUTs and a multiplexer. The F5 LUT is the same as the multiplexing of two F4 LUTs described previously with the constraint that the inputs to the F4 LUTs be the same. The F5[A:D] input is then used as the fifth LUT input. The equations for the two F4 LUTs will differ by the assumed value for the F5[A:D] input, one F4 LUT assuming that the F5[A:D] input is zero, and the other assuming it is a one. The selection of the appropriate F4 LUT output in the F5 MUX by the F5[A:D] signal creates a 5-input LUT. Any combination of F4 and F5 LUTs is allowed per PFU using the eight 16-bit LUTs. Examples are eight F4 LUTs, four F5 LUTs, and a combination of four F4 plus two F5 LUTs.



5-5970(F)

Figure 4. Simplified F4 and F5 Logic Modes

Programmable Logic Cells (continued)

Softwired LUT submode uses F4 and F5 LUTs and internal PFU feedback routing to generate complex logic functions up to three LUT-levels deep. Figure 3 shows multiplexers between the Kz[3:0] inputs to the PFU and the LUTs. These multiplexers can be independently configured to route certain LUT outputs to the input of other LUTs. In this manner, very complex logic functions, some of up to 21 inputs, can be implemented in a single PFU at greatly enhanced speeds.

Figure 5 shows several softwired LUT topologies. In this figure, each circle represents either an F4 or F5 LUT. It is important to note that an LUT output that is fed back for softwired use is still available to be registered or output from the PFU. This means, for instance, that a logic equation that is needed by itself and as a term in a larger equation need only be generated once and PLC routing resources will not be required to use it in the larger equation.

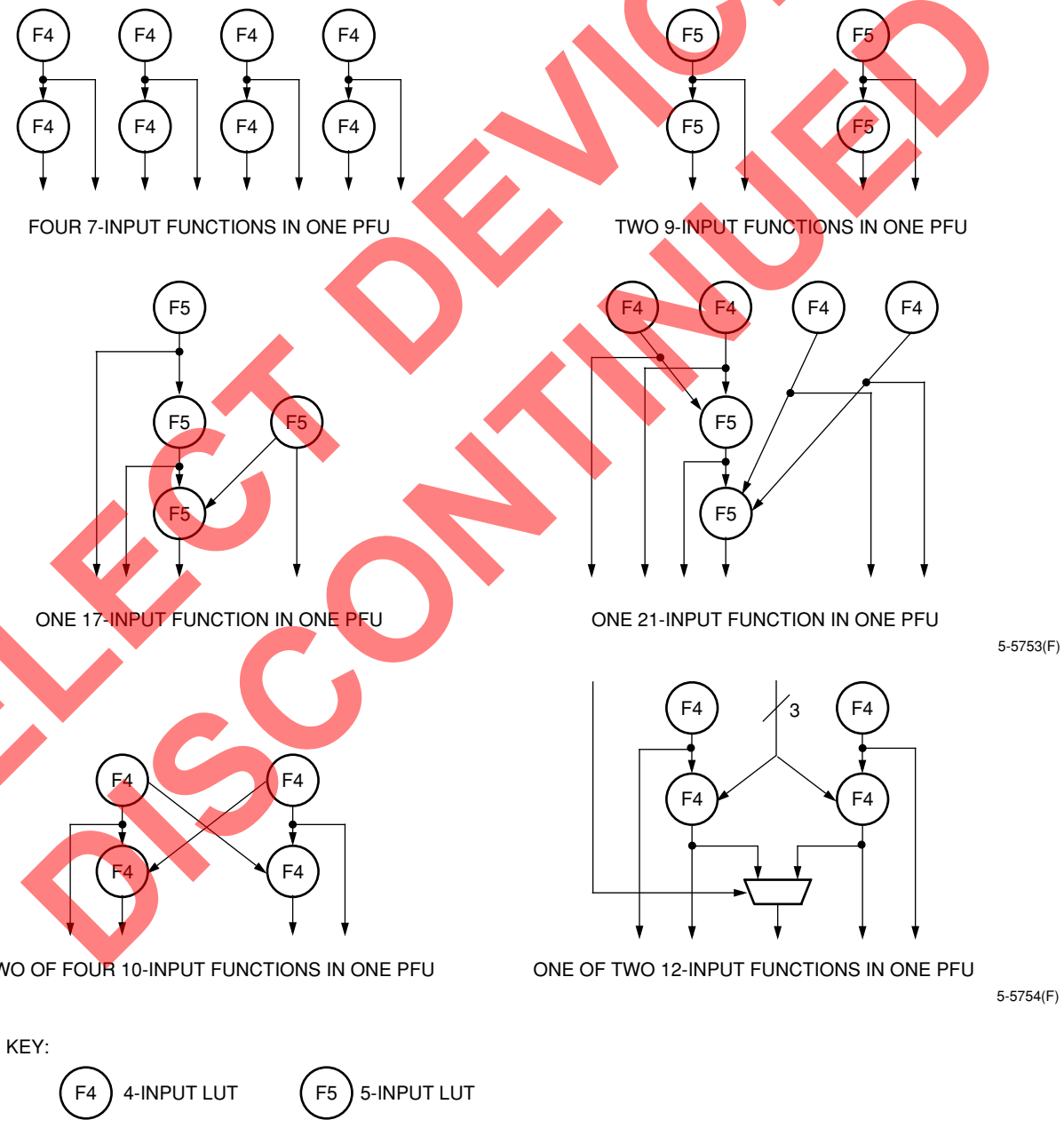


Figure 5. Softwired LUT Topology Examples

Programmable Logic Cells (continued)

Half-Logic Mode

Series 3 FPGAs are based upon a twin-quad architecture in the PFUs. The byte-wide nature (eight LUTs, eight latches/FFs) may just as easily be viewed as two nibbles (two sets of four LUTs, four latches/FFs). The two nibbles of the PFU are organized so that any nibble-wide feature (excluding some softwired LUT topologies) can be swapped with any other nibble-wide feature in another PFU. This provides for very flexible use of logic and for extremely flexible routing. The half-logic mode of the PFU takes advantage of the twin-quad architecture and allows half of a PFU, K[7:4] and associated latches/FFs, to be used in logic mode while the other half of the PFU, K[3:0] and associated latches/FFs, is used in ripple mode. In half-logic mode, the ninth FF may be used as a general-purpose FF or as a register in the ripple mode carry chain.

Ripple Mode

The PFU LUTs can be combined to do byte-wide ripple functions with high-speed carry logic. Each LUT has a dedicated carry-out net to route the carry to/from any adjacent LUT. Using the internal carry circuits, fast arithmetic, counter, and comparison functions can be implemented in one PFU. Similarly, each PFU has carry-in (CIN, FCIN) and carry-out (COUT, FCOUT) ports for fast-carry routing between adjacent PFUs.

The ripple mode is generally used in operations on two data buses. A single PFU can support an 8-bit ripple function. Data buses of 4 bits and less can use the nibble-wide ripple chain that is available in half-logic mode. This nibble-wide ripple chain is also useful for longer ripple chains where the length modulo 8 is four or less. For example, a 12-bit adder (12 modulo 8 = 4) can be implemented in one PFU in ripple mode (8 bits) and one PFU in half-logic mode (4 bits), freeing half of a PFU for general logic mode functions.

Each LUT has two operands and a ripple (generally carry) input, and provides a result and ripple (generally carry) output. A single bit is rippled from the previous LUT and is used as input into the current LUT. For LUT K0, the ripple input is from the PFU CIN or FCIN port. The CIN/FCIN data can come from either the fast-carry routing (FCIN) or the PFU input (CIN), or it can be tied to logic 1 or logic 0.

In the following discussions, the notations LUT K7/K3 and F[7:0]/F[3:0] are used to denote the LUT that provides the carry-out and the data outputs for full PFU ripple operation (K7, F[7:0]) and half-logic ripple operation (K3, F[3:0]), respectively. The ripple mode diagram in Figure 6 shows full PFU ripple operation, with half-logic ripple connections shown as dashed lines.

The result output and ripple output are calculated by using generate/propagate circuitry. In ripple mode, the two operands are input into Kz[1] and Kz[0] of each LUT. The result bits, one per LUT, are F[7:0]/F[3:0] (see Figure 6). The ripple output from LUT K7/K3 can be routed on dedicated carry circuitry into any of four adjacent PLCs, and it can be placed on the PFU COUT/FCOUT outputs. This allows the PLCs to be cascaded in the ripple mode so that nibble-wide ripple functions can be expanded easily to any length.

Result outputs and the carry-out may optionally be registered within the PFU. The capability to register the ripple results, including the carry output, provides for improved counter performance and simplified pipelining in arithmetic functions.

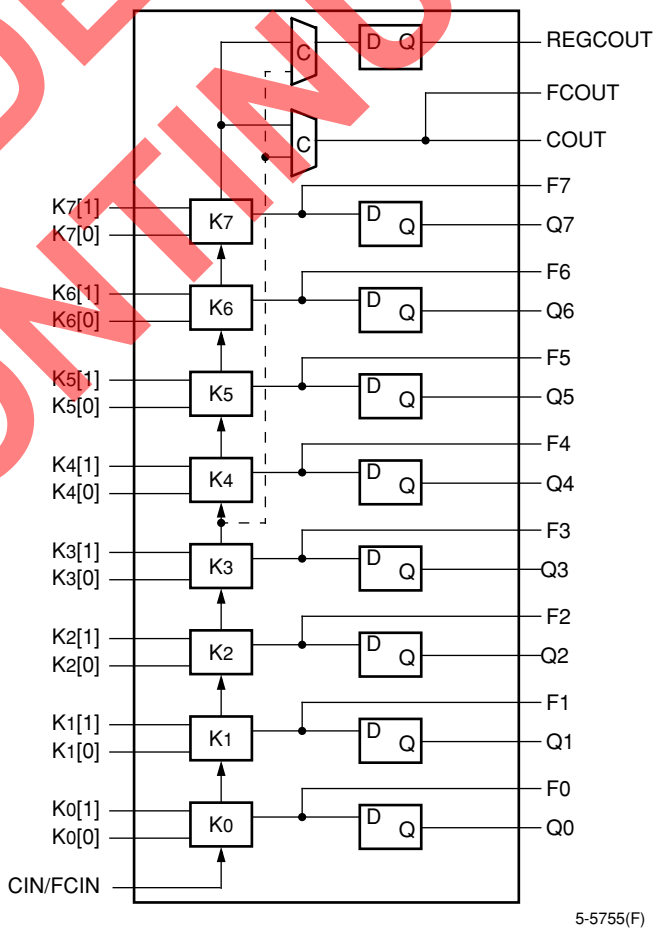


Figure 6. Ripple Mode

Programmable Logic Cells (continued)

The ripple mode can be used in one of four submodes. The first of these is **adder-subtractor submode**. In this submode, each LUT generates three separate outputs. One of the three outputs selects whether the carry-in is to be propagated to the carry-out of the current LUT or if the carry-out needs to be generated. If the carry-out needs to be generated, this is provided by the second LUT output. The result of this selection is placed on the carry-out signal, which is connected to the next LUT carry-in or the COUT/FCOUT signal, if it is the last LUT (K_7/K_3). Both of these outputs can be any equation created from Kz[1] and Kz[0], but in this case, they have been set to the propagate and generate functions.

The third LUT output creates the result bit for each LUT output connected to F[7:0]/F[3:0]. If an adder/subtractor is needed, the control signal to select addition or subtraction is input on ASWE, with a logic 0 indicating subtraction and a logic 1 indicating addition. The result bit is created in one-half of the LUT from a single bit from each input bus Kz[1:0], along with the ripple input bit.

The second submode is the **counter submode** (see Figure 7). The present count, which may be initialized via the PFU DIN inputs to the latches/FFs, is supplied to input Kz[0], and then output F[7:0]/F[3:0] will either be incremented by one for an up counter or decremented by one for a down counter. If an up/down counter is needed, the control signal to select the direction (up or down) is input on ASWE with a logic 1 indicating an up counter and a logic 0 indicating a down counter. Generally, the latches/FFs in the same PFU are used to hold the present count value.

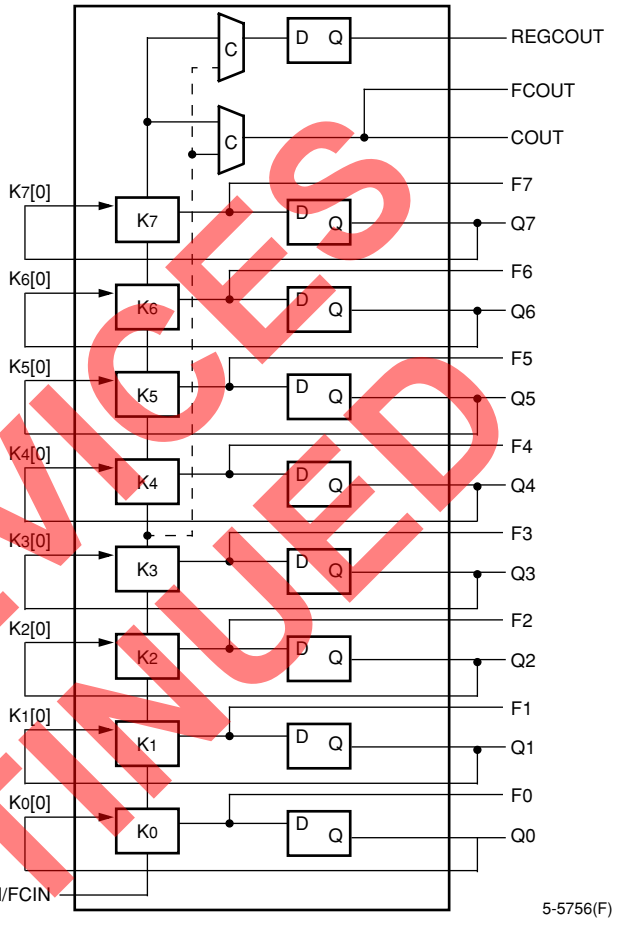


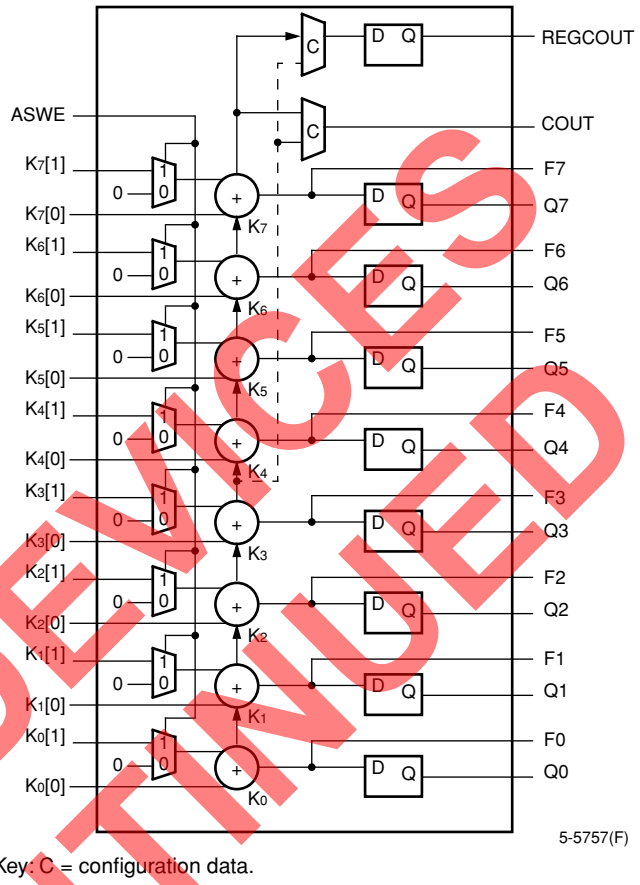
Figure 7. Counter Submode

Programmable Logic Cells (continued)

In the third submode, **multiplier submode**, a single PFU can affect an 8 x 1 bit (4 x 1 for half-ripple mode) multiply and sum with a partial product (see Figure 8). The multiplier bit is input at ASWE, and the multiplicand bits are input at Kz[1], where K7[1] is the most significant bit (MSB). Kz[0] contains the partial product (or other input to be summed) from a previous stage. If ASWE is logical 1, the multiplicand is added to the partial product. If ASWE is logical 0, 0 is added to the partial product, which is the same as passing the partial product. CIN/FCIN can bring the carry-in from the less significant PFUs if the multiplicand is wider than 8 bits, and COUT/FCOUT holds any carry-out from the multiplication, which may then be used as part of the product or routed to another PFU in multiplier mode for multiplicand width expansion.

Ripple mode's fourth submode features **equality comparators.** The functions that are explicitly available are A > B, A ≠ B, and A < B, where the value for A is input on Kz[0], and the value for B is input on Kz[1]. A value of 1 on the carry-out signals valid argument. For example, a carry-out equal to 1 in AB submode indicates that the value on Kz[0] is greater than or equal to the value on Kz[1]. Conversely, the functions A < B, A + B, and A > B are available using the same functions but with a 0 output expected. For example, A > B with a 0 output indicates A < B. Table 5 shows each function and the output expected.

If larger than 8 bits, the carry-out signal can be cascaded using fast-carry logic to the carry-in of any adjacent PFU. The use of this submode could be shown using Figure 6, except that the CIN/FCIN input for the least significant PFU is controlled via configuration.



Key: C = configuration data.

Figure 8. Multiplier Submode

Table 5. Ripple Mode Equality Comparator Functions and Outputs

Equality Function	ispLEVER Submode	True, if Carry-Out Is:
A > B	A > B	1
A < B	A < B	1
A ≠ B	A ≠ B	1
A < B	A > B	0
A > B	A < B	0
A = B	A ≠ B	0

Programmable Logic Cells (continued)

Memory Mode

The Series 3 PFU can be used to implement a 32 x 4 (128-bit) synchronous, dual-port random access memory (RAM). A block diagram of a PFU in memory mode is shown in Figure 9. This RAM can also be configured to work as a single-port memory and because initial values can be loaded into the RAM during configuration, it can also be used as a read-only memory (ROM).

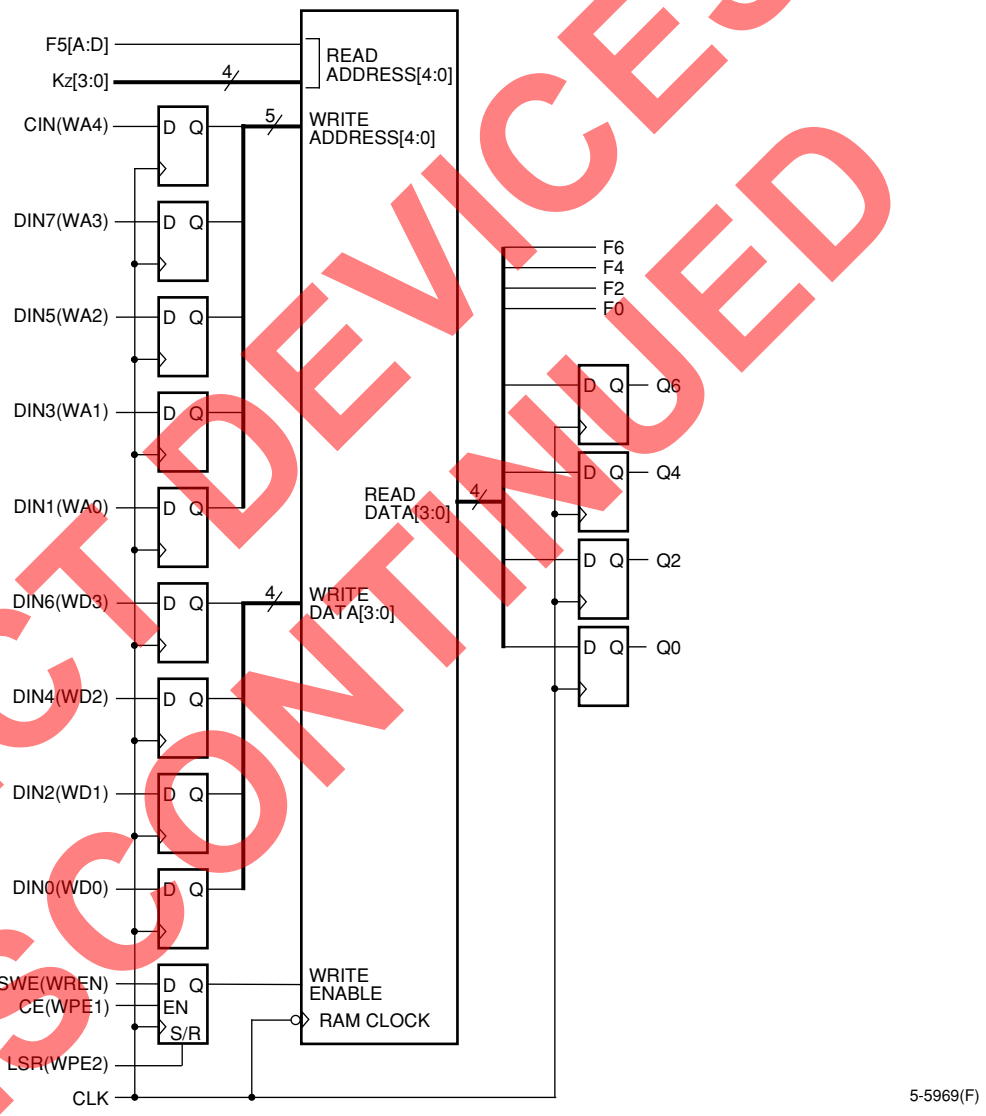


Figure 9. Memory Mode

The PFU memory mode uses all LUTs and latches/FFs including the ninth FF in its implementation as shown in Figure 9. The read address is input at the Kz[3:0] and F5[A:D] inputs where Kz[0] is the LSB and F5[A:D] is the MSB, and the write address is input on CIN (MSB) and DIN[7, 5, 3, 1], with DIN[1] being the LSB. Write data is input on DIN[6, 4, 2, 0], where DIN[6] is the MSB, and read data is available combinatorially on F[6, 4, 2, 0] and registered on Q[6, 4, 2, 0] with F[6] and Q[6] being the MSB. The write enable signal is input at ASWE, and two write port enables are input on CE and LSR. The PFU CLK signal is used to synchronously write the data. The polarities of the clock, write enable, and port enables are all programmable. Write-port enables may be disabled if they are not to be used.

Programmable Logic Cells (continued)

Data is written to the write data, write address, and write enable registers on the active edge of the clock, but data is not written into the RAM until the next clock edge one-half cycle later. The read port is actually asynchronous, providing the user with read data very quickly after setting the read address, but timing is also provided so that the read port may be treated as fully synchronous for write then read applications. If the read and write address lines are tied together (maintaining MSB to MSB, etc.), then the dual-port RAM operates as a synchronous single-port RAM. If the write enable is disabled, and an initial memory contents is provided at configuration time, the memory acts as a ROM (the write data and write address ports and write port enables are not used).

Wider memories can be created by operating two or more memory mode PFUs in parallel, all with the same address and control signals, but each with a different nibble of data. To increase memory word depth above 32, two or more PLCs can be used. Figure 10 shows a 128 x 8 dual-port RAM that is implemented in eight PLCs. This figure demonstrates data path width expansion by placing two memories in parallel to achieve an 8-bit data path. Depth expansion is applied to achieve 128 words deep using the 32-word deep PFU memories. In addition to the PFU in each PLC, the SLIC (described in the next section) in each PLC is used for read address decodes and 3-state drivers. The 128 x 8 RAM shown could be made to operate as a single-port RAM by tying (bit-for-bit) the read and write addresses.

To achieve depth expansion, one or two of the write address bits (generally the MSBs) are routed to the write port enables as in Figure 10. For 2 bits, the bits select which 32-word bank of RAM of the four available from a decode of two WPE inputs is to be written. Similarly, 2 bits of the read address are decoded in the SLIC and are used to control the 3-state buffers through which the read data passes. The write data bus is common, with separate nibbles for width expansion, across all PLCs, and the read data bus is common (again, with separate nibbles) to all PLCs at the output of the 3-state buffers.

Figure 10 also shows a new optional capability to provide a read enable for RAMs/ROMs in Series 3 using the SLIC cell. The read enable will 3-state the read data bus when inactive, allowing the write data and read data buses to be tied together if desired.

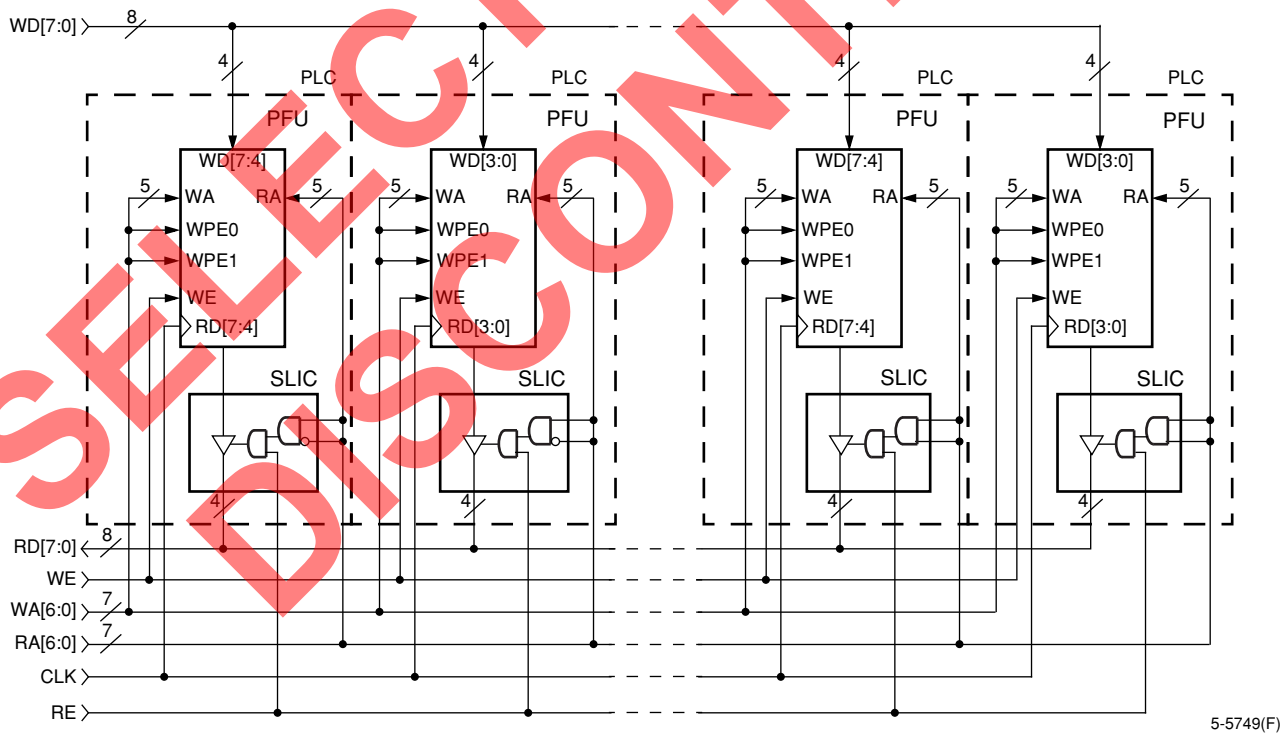


Figure 10. Memory Mode Expansion Example—128 x 8 RAM

Programmable Logic Cells (continued)

Supplemental Logic and Interconnect Cell (SLIC)

Each PLC contains a supplemental logic and interconnect cell (SLIC) embedded within the PLC routing, outside of the PFU. As its name indicates, the SLIC performs both logic and interconnect (routing) functions. Its main features are 3-statable, bidirectional buffers, and a *PAL*-like decoder capability. Figure 11 shows a diagram of a SLIC with all of its features shown. All modes of the SLIC are not available at one time.

Each SLIC contains ten bidirectional (BIDI) buffers, each buffer capable of driving left and/or right out of the SLIC. These BIDI buffers are twin-quad in nature and are segregated into two groups of four (nibbles) and a third group of two for control. Each of these groups of BIDIs can drive from the left (BLI[9:0]) to the right (BRO[9:0]), the right (BRI[9:0]) to the left (BLO[9:0]), or from the central input (I[9:0]) to the left and/or right. This central input comes directly from the PFU outputs (O[9:0]). Each of the BIDIs in the nibble-wide groups also has a 3-state buffer capability, but not the third group.

There is one 3-state control (TRI) for each SLIC, with the capability to invert or disable the 3-state control for each group of four BIDIs. Separate 3-state control for each nibble-wide group is achievable by using the SLIC's decoder (DEC) output, driven by the group of two BIDIs, to control the 3-state of one BIDI nibble while using the TRI signal to control the 3-state of the other BIDI nibble. Figure 12 and Figure 13 show the SLIC in buffer mode with available 3-state control from the TRI and DEC signals. If the entire SLIC is acting in a buffer capacity, the DEC output may be used to generate a constant logic 1 (VHI) or logic 0 (VLO) signal for general use.

The SLIC may also be used to generate *PAL*-like AND-OR with optional INVERT (AOI) functions or a decoder of up to 10 bits. Each group of buffers can feed into an AND gate (4-input AND for the nibble groups and 2-input AND for the other two buffers). These AND gates then feed into a 3-input gate that can be configured as either an AND gate or an OR gate. The output of the 3-input gate is invertible and is output at the DEC output of the SLIC. Figure 16 shows the SLIC in full decoder mode.

The functionality of the SLIC is parsed by the two nibble-wide groups and the 2-bit buffer group. Each of these groups may operate independently as BIDI buffers (with or without 3-state capability for the nibble-wide groups) or as a *PAL*/decoder.

As discussed in the memory mode section, if the SLIC is placed into one of the modes where it contains both buffers and a decode or AOI function (e.g., BUF_BUF_DEC mode), the DEC output can be gated with the 3-state input signal. This allows up to a 6-input decode (e.g., BUF_DEC_DEC mode) plus the 3-state input to control the enable/disable of up to four buffers per SLIC. Figure 12—Figure 16 show several configurations of the SLIC, while Table 6 shows all of the possible modes.

Table 6. SLIC Modes

Mode #	Mode	BUF [3:0]	BUF [7:4]	BUF [9:8]
1	BUFFER	Buffer	Buffer	Buffer
2	BUF_BUF_DEC	Buffer	Buffer	Decoder
3	BUF_DEC_BUF	Buffer	Decoder	Buffer
4	BUF_DEC_DEC	Buffer	Decoder	Decoder
5	DEC_BUF_BUF	Decoder	Buffer	Buffer
6	DEC_BUF_DEC	Decoder	Buffer	Decoder
7	DEC_DEC_BUF	Decoder	Decoder	Buffer
8	DECODER	Decoder	Decoder	Decoder

Programmable Logic Cells (continued)

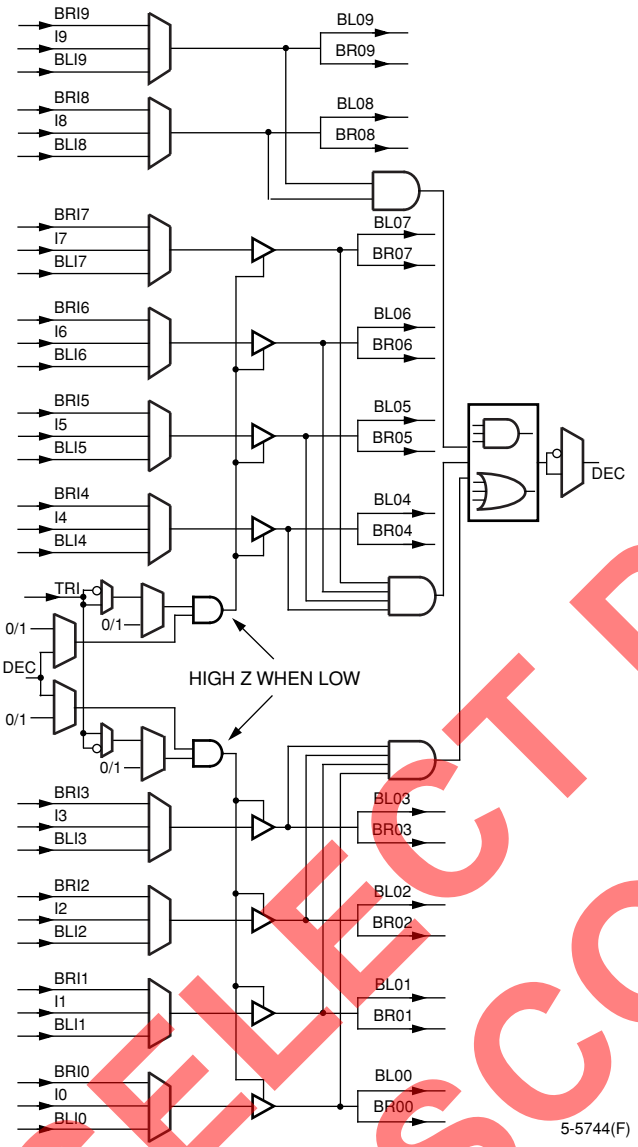


Figure 11. SLIC All Modes Diagram

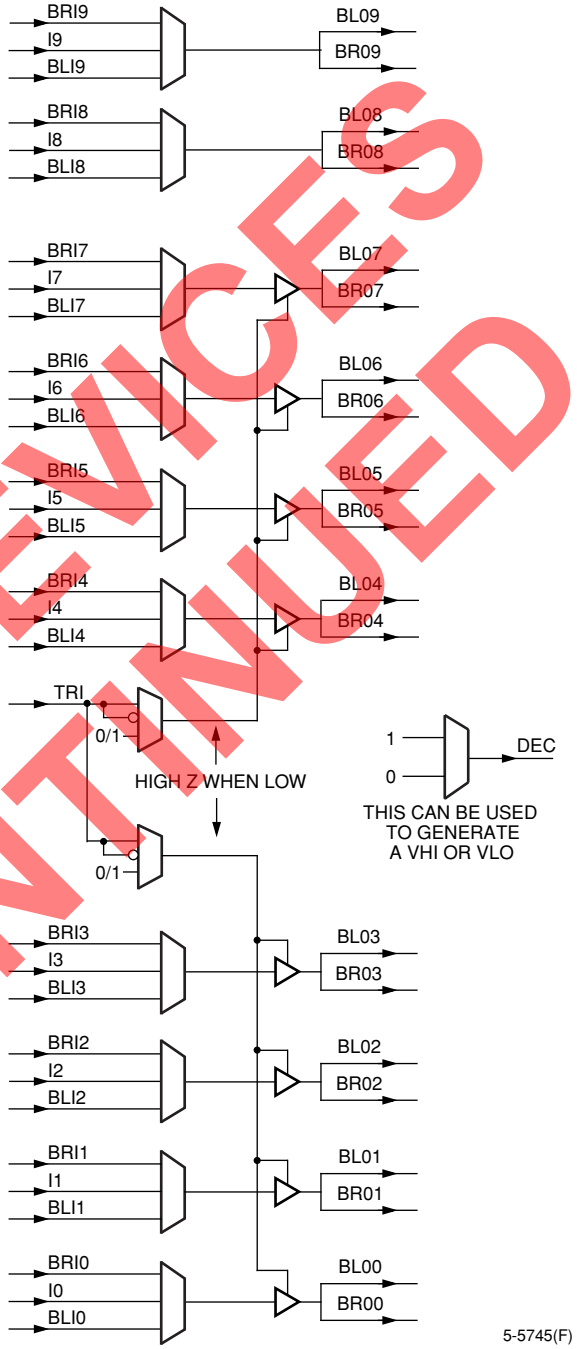


Figure 12. Buffer Mode

Programmable Logic Cells (continued)

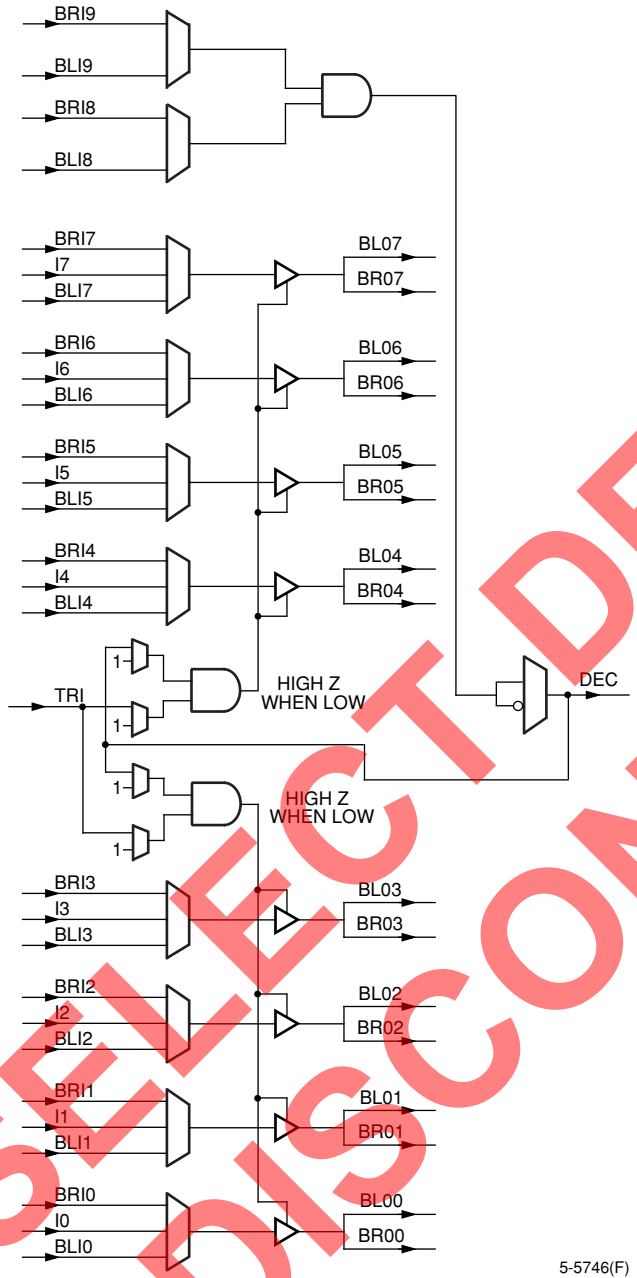


Figure 13. Buffer-Buffer-Decoder Mode

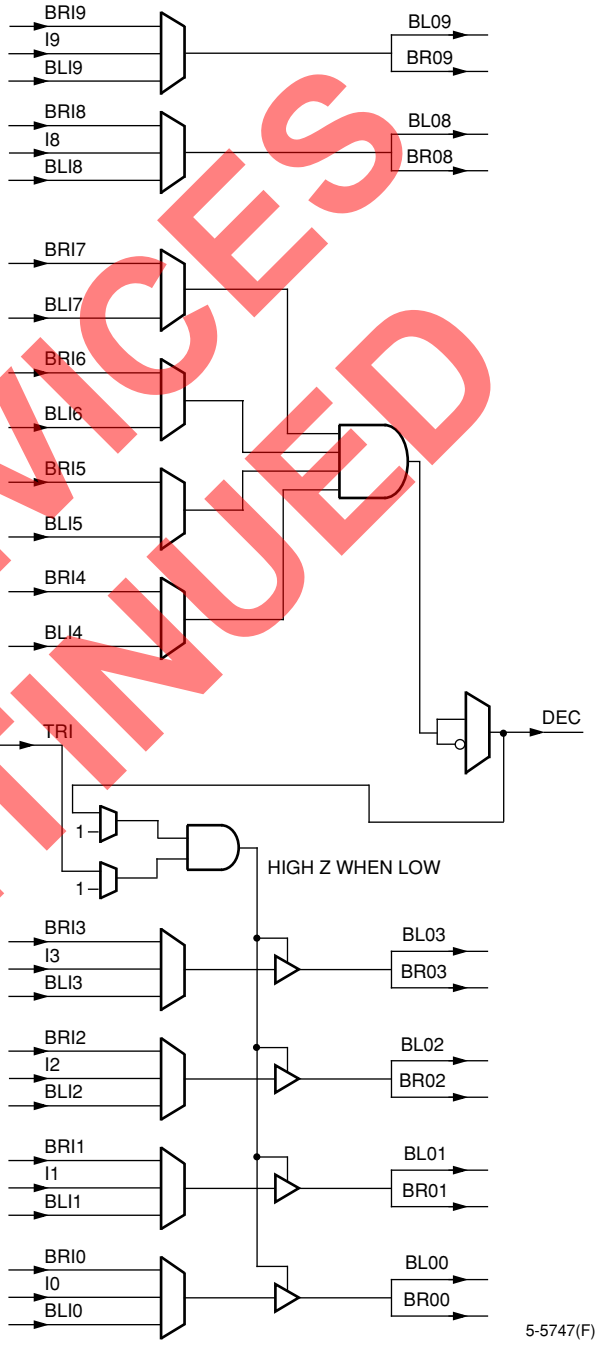


Figure 14. Buffer-Decoder-Buffer Mode

Programmable Logic Cells (continued)

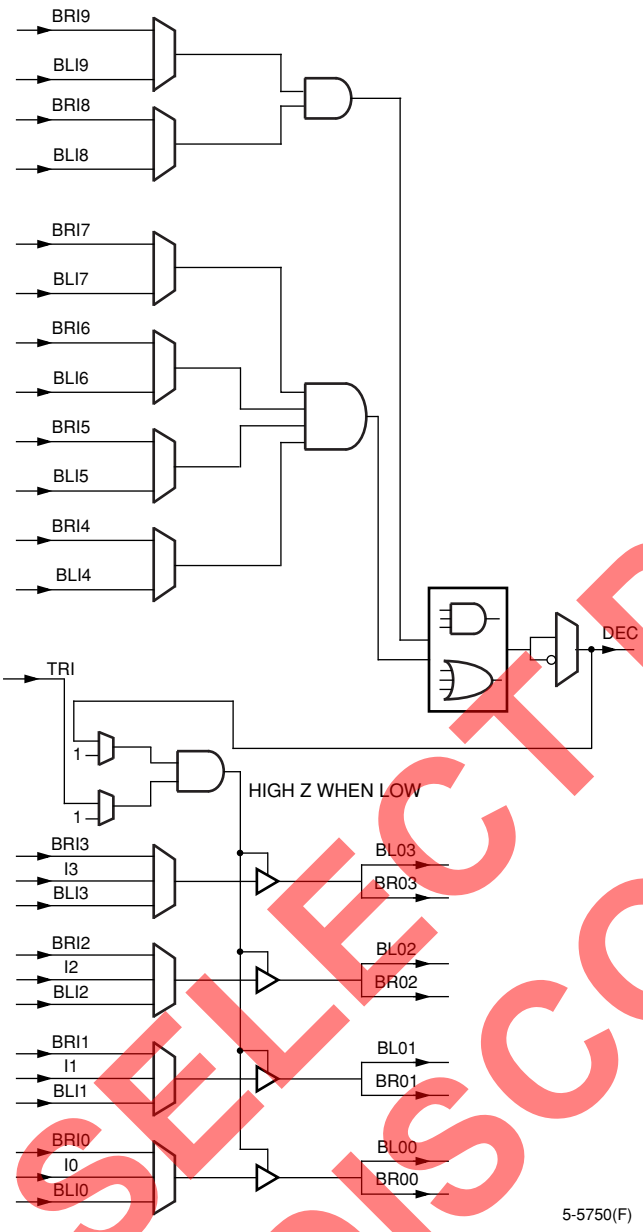


Figure 15. Buffer-Decoder-Decoder Mode

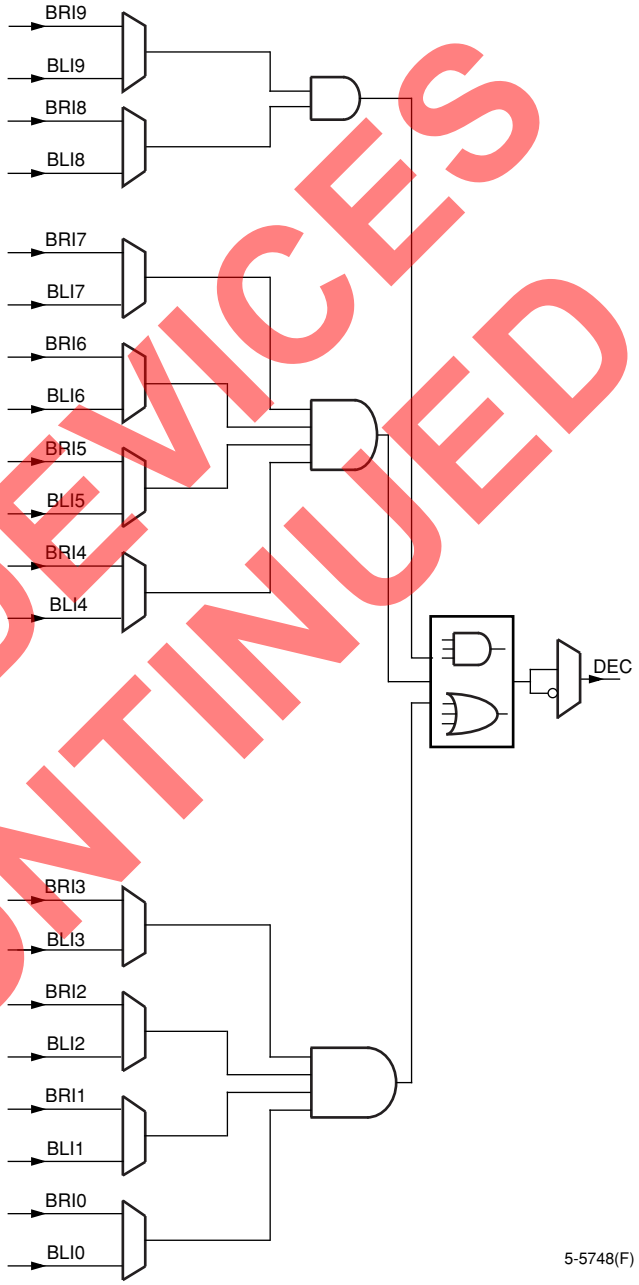


Figure 16. Decoder Mode

Programmable Logic Cells (continued)

PLC Latches/Flip-Flops

The eight general-purpose latches/FFs in the PFU can be used in a variety of configurations. In some cases, the configuration options apply to all eight latches/FFs in the PFU and some apply to the latches/FFs on a nibble-wide basis where the ninth FF is considered independently. For other options, each latch/FF is independently programmable. In addition, the ninth FF can be used for a variety of functions.

Table 7 summarizes these latch/FF options. The latches/FFs can be configured as either positive- or negative-level sensitive latches, or positive or negative edge-triggered flip-flops (the ninth register can only be FF). All latches/FFs in a given PFU share the same clock, and the clock to these latches/FFs can be inverted. The input into each latch/FF is from either the corresponding LUT output (F[7:0]) or the direct data input (DIN[7:0]). The latch/FF input can also be tied to logic 1 or to logic 0, which is the default.

Table 7. Configuration RAM Controlled Latch/Flip-Flop Operation

Function	Options
Common to All Latches/FFs in PFU	
LSR Operation	Asynchronous or synchronous
Clock Polarity	Noninverted or inverted
Front-end Select*	Direct (DIN[7:0]) or from LUT (F[7:0])
LSR Priority	Either LSR or CE has priority
Latch/FF Mode	Latch or flip-flop
Enable GSRN	GSRN enabled or has no effect on PFU latches/FFs
Set Individually in Each Latch/FF in PFU	
Set/Reset Mode	Set or reset
By Group (Latch/FF[3:0], Latch/FF[7:4], and FF[8])	
Clock Enable	CE or ASWE or none
LSR Control	LSR or none

* Not available for FF[8].

The eight latches/FFs in a PFU share the clock (CLK) and options for clock enable (CE), local set/reset (LSR), and front-end data select (SEL) inputs. When CE is disabled, each latch/FF retains its previous value when clocked. The clock enable, LSR, and SEL inputs can be inverted to be active-low.

The set/reset operation of the latch/FF is controlled by two parameters: reset mode and set/reset value. When the global set/reset (GSRN) and local set/reset (LSR) signals are not asserted, the latch/FF operates normally. The reset mode is used to select a synchronous or asynchronous LSR operation. If synchronous, LSR has the option to be enabled only if clock enable (CE or ASWE) is active or for LSR to have priority over the clock enable input, thereby setting/resetting the FF independent of the state of the clock enable. The clock enable is supported on FFs, not latches. It is implemented by using a 2-input multiplexer on the FF input, with one input being the previous state of the FF and the other input being the new data applied to the FF. The select of this 2-input multiplexer is clock enable (CE or ASWE), which selects either the new data or the previous state. When the clock enable is inactive, the FF output does not change when the clock edge arrives.

Programmable Logic Cells (continued)

The GSRN signal is only asynchronous, and it sets/resets all latches/FFs in the FPGA based upon the set/reset configuration bit for each latch/FF. The set/reset value determines whether GSRN and LSR are set or reset inputs. The set/reset value is independent for each latch/FF. A new option is available to disable the GSRN function per PFU after initial device configuration.

The latch/FF can be configured to have a data front-end select. Two data inputs are possible in the front-end select mode, with the SEL signal used to select which data input is used. The data input into each latch/FF is from the output of its associated LUT, F[7:0], or direct from DIN[7:0], bypassing the LUT. In the front-end data select mode, both signals are available to the latches/FFs.

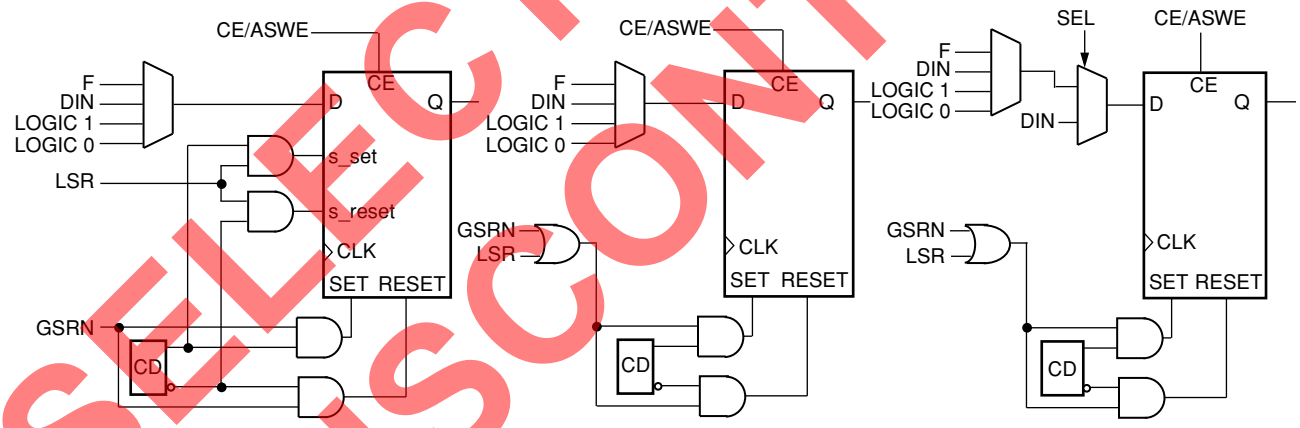
If either or both of these inputs is unused or is unavailable, the latch/FF data input can be tied to a logic 0 or logic 1 instead (the default is logic 0).

The latches/FFs can be configured in three basic modes:

1. Local synchronous set/reset: the input into the PFU's LSR port is used to synchronously set or reset each latch/FF.
2. Local asynchronous set/reset: the input into LSR asynchronously sets or resets each latch/FF.
3. Latch/FF with front-end select, LSR either synchronous or asynchronous: the data select signal selects the input into the latches/FFs between the LUT output and direct data in.

For all three modes, each latch/FF can be independently programmed as either set or reset. Figure 17 provides the logic functionality of the front-end select, global set/reset, and local set/reset operations.

The ninth PFU FF, which is generally associated with registering the carry-out signal in ripple mode functions, can be used as a general-purpose FF. It is only an FF and is not capable of being configured as a latch. Because the ninth FF is not associated with an LUT, there is no front-end data select. The data input to the ninth FF is limited to the CIN input, logic 1, logic 0, or the carry-out in ripple and half-logic modes.



Key: C = configuration data.

Figure 17. Latch/FF Set/Reset Configurations

Programmable Logic Cells (continued)

PLC Routing Resources

Generally, the ispLEVER Development System is used to automatically route interconnections. Interactive routing with the ispLEVER design editor (EPIC) is also available for design optimization. To use EPIC for interactive layout, an understanding of the routing resources is needed and is provided in this section.

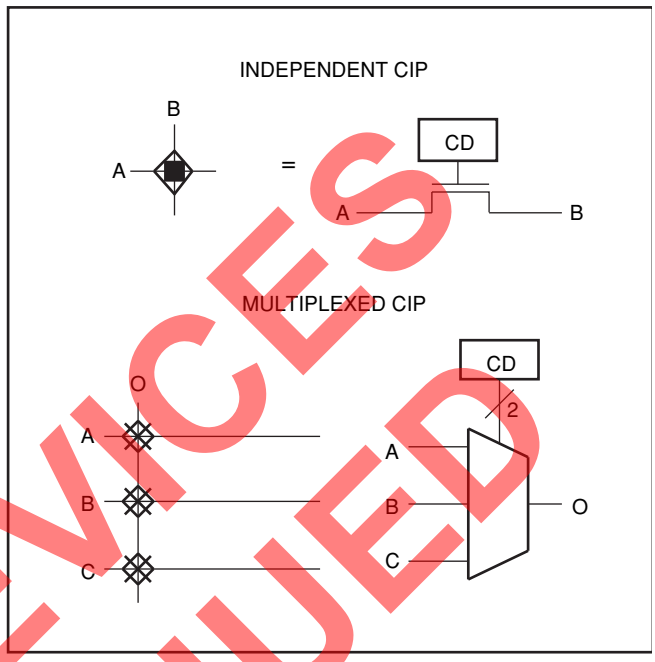
The routing resources consist of switching circuitry and metal interconnect segments. Generally, the metal lines which carry the signals are designated as routing segments. The switching circuitry connects the routing segments, providing one or more of three basic functions: signal switching, amplification, and isolation. A net running from a PFU or PIC output (source) to a PLC or PIC input (destination) consists of one or more routing segments, connected by switching circuitry called configurable interconnect points (CIPs).

The following sections discuss PLC, PIC, and interquad routing resources. This section discusses the PLC switching circuitry, intra-PLC routing, inter-PLC routing, and clock distribution.

Configurable Interconnect Points

The process of connecting routing segments uses three basic types of switching circuits: two types of configurable interconnect points (CIPs) and bidirectional buffers (BIDIs). The basic element in CIPs is one or more pass transistors, each controlled by a configuration RAM bit. The two types of CIPs are the mutually exclusive (or multiplexed) CIP and the independent CIP.

A mutually exclusive set of CIPs contains two or more CIPs, only one of which can be on at a time. An independent CIP has no such restrictions and can be on independent of the state of other CIPs. Figure 18 shows an example of both types of CIPs.



Key: C = configuration data.

5-5973(C)

Figure 18. Configurable Interconnect Point

3-Statable Bidirectional Buffers

Bidirectional buffers, previously described in the SLIC section of the programmable logic cell discussion, provide isolation as well as amplification for signals routed a long distance. Bidirectional buffers are also used to route signals diagonally in the PLC (described later in the subsection entitled Intra-PLC Routing), and BIDIs can be used to indirectly route signals through the switching routing (xSW) segments. Any number from zero to ten BIDIs can be used in a given PLC.

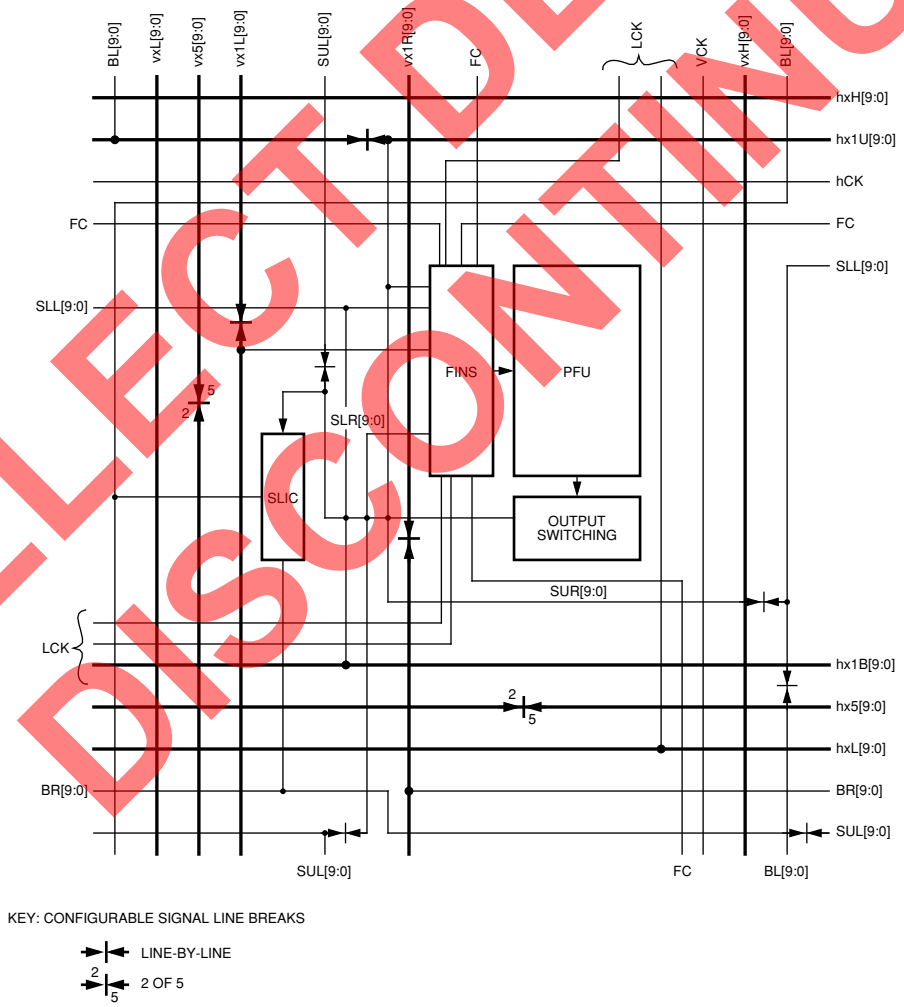
Programmable Logic Cells (continued)

General Routing Structure

Routing resources in Series 3 FPGAs generally consist of routing segments in groups of ten, with varying lengths and connectivity to logic and other routing resources. The varying lengths of routing segments provides a hierarchy of routing capability from chip-length routes to routes within a PLC. The hierarchical nature of the routing provides the ispLEVER development tools with the necessary resources to route a design completely and to optimize the routing for system speed while reducing the overall power required by the device.

Within each group of ten routing segments there is an equivalency of connectivity between pairs of segments. These pairs are segments: [0, 4] and [1, 5] and [2, 6] and [3, 7] and [8, 9]. The equivalency in connectivity ensures that signals on either segment in a pair have the same capability to get to a given destination. This, in turn, allows for signal distribution from a source to varying destinations without using special routing. It also provides for routing flexibility by ensuring that one segment position will not become so congested as to preclude routing a bus or group of signals and allows easy connectivity from either of the twin quads in a source PFU to either of the twin quads in any destination PFU.

Having ten segments in a group is significant in that it provides for routing a byte of data and two control signals or parity. Due to the equivalent pairs of segments, this can also be viewed as routing two nibbles each with a control signal. Figure 19 is an overview of the routing for a single PLC.



5-5766(F)

Figure 19. Single PLC View of Inter-PLC Route Segments

Programmable Logic Cells (continued)

Intra-PLC Routing

The function of the intra-PLC routing resources is to connect the PFU's input and output ports to the routing resources used for entry to and exit from the PLC. This routing provides PFU feedback, corner turning, or switching from one type of routing resource to another.

Flexible Input Structure (FINS)

The flexible input switching structure (FINS) in each PLC of the *ORCA* Series 3 provides for the flexibility of a crossbar switch from the routing resources to the PFU inputs while taking advantage of the routability of shared inputs. Connectivity between the PLC routing resources and the PFU inputs is provided in two stages. The primary FINS switch has 50 inputs that connect the PLC routing to the 35 inputs on the secondary switch. The outputs of the second switch connect to the 50 PFU inputs. The switches are implemented to provide connectivity for bused signals and individual connections.

PFU Output Switching

The PFU outputs are switched onto PLC routing resources via the PFU output multiplexer (OMUX). The PFU output switching segments from the output multiplexer provide ten connections to the PLC routing out of 18 possible PFU outputs (F[7:0], Q[7:0], DOUT, REGCOUT). These output switching segments connect segment for segment to the SUR, SUL, SLR, and SLL switching segments described below (e.g., O4 connects only to SUR4, not SUR5). The output switching segments also feed directly into the SLIC on a segment-by-segment basis. This connectivity is also described below.

Switching Routing Segments (xSW)

There are four sets of switching routing segments in each PLC. Each set consists of ten switching elements: SUL[9:0], SUR[9:0], SLL[9:0], and SLR[9:0], traditionally labeled for the upper-left, upper-right, lower-left, and lower-right sections of the PFUs, respectively. The xSW routing segments connect to the PFU inputs and outputs as well as the BIDI routing segments, to be described later. They also connect to both the horizontal and vertical x1 and x5 routing segments (inter-PLC routing resources, described later) in their specific corner. xSW segments can be used for fast connections between adjacent PLCs or PICs without requiring the use of inter-PLC routing resources. This capability not only increases signal speed on adjacent PLC routing, but also reduces routing congestion on the principal inter-PLC routing resources. The SLL and SUR segments combine to provide connectivity to the PLCs to the left and right of the current PLC; the SLR and SUL segments combine to provide connectivity to the PLCs above and below the current PLC.

Fast routes on switching segments to diagonally adjacent PLCs/PICs are possible using the BIDI routing segments (discussed below) and the SLL and SLR switching segments. The BR BIDI routing segments combine with the SUL switching segments of the PLC below and to the right of the current PLC to connect to that PLC. The BL BIDI routing segments combine with the SLL switching segments of the PLC above and to the right of the current PLC to connect to that PLC. These fast diagonal connections provide a great amount of flexibility in routing congested areas of logic and in shifting data on a per-PLC basis such as performing implicit multiplications/divisions in routing between functional logic elements.

Switching routing segments are also the chief means by which signals are transferred between the inter-PLC routing resources and the PFU. Each set of switching segments has connectivity to the x1 routing segments, and there is varying connectivity to the x5, xH, and xL inter-PLC routing segments. Detailed information on switching segment/inter-PLC routing connectivity is provided later in this section in the Inter-PLC Routing Resources subsection.

Programmable Logic Cells (continued)

BIDI Routing and SLIC Connectivity

The SLIC is connected to the rest of the PLC by the bidirectional (BIDI) routing segments and the PFU output switching segments coming from the PFU output multiplexer. The BIDI routing segments (xBID) are labeled as BL for BIDI-left and BR for BIDI-right. Each set of BR and BL xBID segments is composed of ten bidirectional lines (note that these lines are diagramed as ten input lines to the SLIC and ten output lines from the SLIC that can be used in a mutually exclusive fashion). Because the SLIC is connected directly to the outputs of the PFU, it provides great flexibility in routing via the xBID segments. The PFU routing segments, O[9:0], only connect to their respective line in the SLL, SUL, SUR, and SLR switching segment groups. That is, O9 only connects to SLL9, SUL9, SUR9, and SLR9. The BIDI lines provide the capability to connect to the other member of the routing set. That means, for example, that O9 can be routed to BR8 or BL8. This connectivity can be used as a means to distribute or gather signals on intra-PLC routing without disturbing inter-PLC resources. As described in the Switching Routing Segments subsection, the BIDI routing segments are also used for routes to a diagonally adjacent PFU.

In addition to the intra-PLC connections, the xBID and output switching segments also have connectivity to the x1, x5, and xL inter-PLC routing resources, providing an alternate routing path rather than using PLC xSW segments. These connections also provide a path to the 3-state buffers in the SLIC without encumbering the xSW segments. In this manner, buffering or 3-state control can be added to inter-PLC routing without disturbing local functionality within a PFU.

Control Signal and Fast-Carry Routing

PFU control signal and the fast-carry routing are performed using the FINS structure and several dedicated routing paths. The fast-carry (FC) routing resources consist of a dedicated bidirectional segment between each orthogonal pair of PLCs. This means that a fast-carry can go to or come from each PLC to the right or left, above or below the subject PLC. The FINS structure is used to control the switching of these fast-carry paths between the fast-carry input (FCIN) and fast-carry output (FCOUT) ports of the PFU.

The PFU control inputs (CE, SEL, LSR, ASWE) and CIN can be reached via the FINS by two special routing segments, E1 and E2. The E1 routing segment provides connectivity between all of the xBID routing segments and the FINS. It is unidirectional from the BIDI routing to the FINS. E1 also provides connectivity to the PFU clock input via FINS for a local clock signal. The E2 segment connects the SLIC DEC output to the FINS and to a group of CIPS that provide bidirectional connectivity with all of the BIDI routing segments. This allows the DEC signal to be used in the PFU and/or routed on the BIDI segments. It also allows signals to be routed to the PFU on the xBID segments if the SLIC DEC output is not used.

There is also a dedicated routing segment from the FINS to the SLIC TRI input used for BIDI buffer 3-state control.

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 Logic Cells (continued)

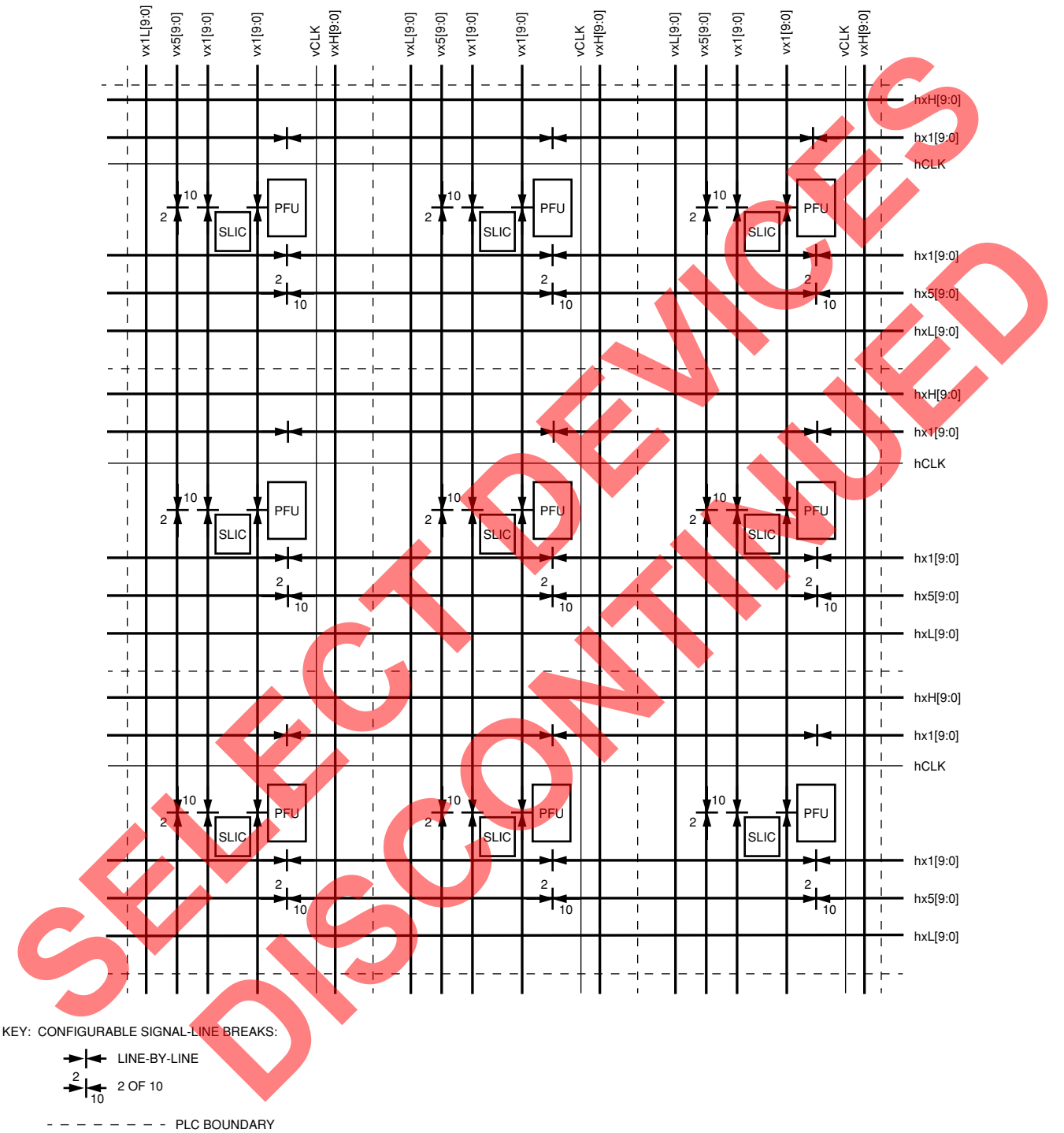


Figure 20. Multiple PLC View of Inter-PLC Routing

Programmable Logic Cells (continued)

xL Routing Lines. The xL routing lines run vertically and horizontally the height and width of the array, respectively. There are a total of 20 xL routing lines per PLC: ten horizontal (hxL[9:0]) and ten vertical (vxL[9:0]). Each of the xL lines connects to the PIC routing at either end. The xL lines are intended primarily for global signals that must travel long distances and require minimum delay and/or skew, such as clocks or 3-state buses.

Each xL line (also called a long line) drives a buffer in each PLC that can drive onto the horizontal and vertical local clock routing segments (ICLK) in the PLC. Also, two out of each group of ten xL segments in each PLC can be driven by a buffer attached to a clock spine (described later) allowing local distribution of global clock signals. More general-purpose connections to the long lines can be made through the xBID segments in a PLC. Each long line is connected to an xBID segment on a bit-by-bit basis. These BIDI connections allow corner turning from horizontal to vertical long lines, and connection between long lines and x1 or x5 segments.

xH Routing Segments. Ten by-half (xH) routing segments run horizontally (hxH[9:0]) and ten xH routing segments run vertically (vxH[9:0]) in each row and column in the array. These routing segments travel a distance of one-half the PLC array before being broken in the middle of the array in the interquad area (discussed later). They also connect at the periphery of the FPGA to the PICs, like the xL lines. xH routing segments connect to the PLCs only by switching segments. They are intended for fast signal interconnect.

Clock (and Global CE and LSR) Routing Segments. For a very fast and low-skew clock (or other global signal tree), clock routing segments run the entire height and width of the PLC array. There are two clock routing segments per PLC: one horizontal (hCLK) and one vertical (vCLK). The source for these clock routing segments can be any of the I/O buffers in the PIC, the Series 3 ExpressCLK inputs, user logic, or the programmable clock manager (PCM). The horizontal clock routing segments (hCLK) are alternately driven by the left and right PICs. The vertical clock routing segments (vCLK) are alternately driven by the top and bottom PICs.

The clock routing segments are designed to be a clock spine. In each PLC, there is a fast connection available from the clock segment to a long-line driver (described earlier). With this connection, one of the clock routing segments in each PLC can be used to drive one of the ten xL routing segments perpendicular to it, which, in turn, creates a clock spine tree. This feature is discussed in detail in the Clock Distribution Network section.

Special connectivity is provided in each PLC to connect the clock enable signals (CE and ASWE) and the LSR signal to the clock network for fast global control signal distribution. CE and ASWE have a special connection to the horizontal clock spine, and LSR has a special connection to the vertical clock spine. This allows both signals to be routed globally within the same PLC, if desired; however, this will consume some of the resources available for clock signal routing.

If using these spines, the clock enable signal must come from the right or left edge of the device, and the LSR signal must come from the top or bottom of the device due to their horizontal and vertical connectivity, respectively, to the clock network.

Minimizing Routing Delay

The CIP is an active element used to connect two routing segments. As an active element, it adds significantly to the resistance and capacitance of a routing network (net), thus increasing the net's delay. The advantage of the x1 segment over an x5 segment is routing flexibility. A net from one PLC to the next is easily routed by using x1 routing segments. As more CIPs are added to a net, the delay increases. To increase speed, routes that are greater than two PLCs away are routed on the x5 routing segments because a CIP is located only in every fifth PLC. A net that spans eight PLCs requires seven x1 routing segments and six CIPs. Using x5 routing segments, the same net uses two routing segments and one CIP.

Programmable Logic Cells (continued)

PLC Architectural Description

Figure 21 is an architectural drawing of the PLC (as seen in ispLEVER) that reflects the PFU, the routing segments, and the CIPs. A discussion of each of the letters in the drawing follows.

A. These are switching routing segments (xSW) that give the router flexibility. In general switching theory, the more levels of indirection there are in the routing, the more routable the network is. The xSW segments can also connect to the xSW lines in adjacent PLCs.

B. These CIPs connect the x1 routing. These are located in the middle of the PLC to allow the block to connect to either the left end of the horizontal x1 segment from the right or the right end of the horizontal x1 segment from the left, or both. By symmetry, the same principle is used in the vertical direction.

C. This set of CIPs is used to connect the x1 and x5 nets to the xSW segments or to other x1 and x5 nets. The CIPs on the major diagonal allow data to be transmitted on a bit-by-bit basis from x1 nets to the xSW segments and between the x1 and x5 nets.

D. This structure is the supplemental logic and interconnect cell, or SLIC. It contains 3-statable bidirectional buffers and logic for building decoders and AND-OR-INVERT type structures.

E. These are the primary and secondary elements of the flexible input structure or FINS. FINS is a switch matrix that provides high connectivity while retaining routing capability. FINS also includes feedback paths for softwired LUT implementation.

F. This is the PFU output switch matrix. It is a complex switch network which, like the FINS at the input, provides high connectivity and maintains routability.

G. This set of CIPs allows an xBID segment to transfer a signal to/from xSW segments on each side. The BIDIs can access the PFU through the xSW segments. These CIPs allow data to be routed through the BIDIs for amplification or 3-state control and continue to another PLC. They also provide an alternative routing resource to improve routability.

H. These CIPs are used to transfer data from/to the xBID segments to/from the x1 and xL routing segments. These CIPs have been optimized to allow the BIDI buffers to drive the loads usually seen when using each type of routing segment.

I. Clock input to PFU.

J. These are the ten switched output routing segments from the PFU. They connect to the PLC switching segments and are input to the SLIC.

K. These lines deliver the auxiliary signals clock enable (CE), local set/reset (LSR), front-end select (SEL), add/subtract/write enable (ASWE), as well as the carry signals (CIN and FCIN) to the latches/FFs.

L. This is the local clock buffer. Any of the horizontal and vertical xL lines can drive the clock input of the PLC latches/FFs. The clock routing segments (vCLK and hCLK) and multiplexers/drivers are used to connect to the xL routing segments for low-skew, low-delay global signals.

M. These routing segments are used to route the fast-carry signal to/from the neighboring four PLCs. The carry-out (COUT) and registered carry-out (REGCOUT) can also be routed out of the PFU.

N. This is the E2 control routing segment. It runs from the SLIC DEC output to the FINS and also provides connectivity to all xBID segments.

O. The xH routing segments run one-half the length (width) of the array before being broken by a CIP.

P. These CIPs connect the xH segments to the xSW segments.

Q. The xBID segments are used to connect the SLIC to the xSW segments, x1 segments, x5 segments, and xL lines, as well as providing for diagonal PLC to PLC connections.

R. These CIPs provide connections from the xBID segments to the E1/E2 routing segments that feed PFU control inputs CE, LSR, CIN, ASWE, SEL, and the clock input. Alternatively, these CIPs connect the BIDI lines to the decoder (DEC) output of the SLIC, for routing the DEC signal.

S. These are clock spines (vCLK and hCLK) with the multiplexers and drivers to connect to the xL routing segments.

T. These CIPs connect xBID segments to switching segments in diagonally and orthogonally adjacent PFUs.

U. These CIPs connect xSW segments to the PFU output segments.

V. These CIPS connect xSW segments in orthogonally adjacent PFUs.

W. This is the SLIC 3-state control routing segment from the FINS to the SLIC 3-state control.

X. This is the E1 control routing segment. It provides a PFU input path from all xBID segments.

Y. These CIPs are used to select which xBID segments are connected to the E1/E2 signal as described in (R).

Programmable Logic Cells (continued)

5-5758(F)

Figure 21. PLC Architecture

Programmable Input/Output Cells

The programmable input/output cells (PICs) are located along the perimeter of the device. The PIC's name is represented by a two-letter designation to indicate on which side of the device it is located followed by a number to indicate in which row or column it is located. The first letter, P, designates that the cell is a PIC and not a PLC. The second letter indicates the side of the array where the PIC is located. The four sides are left (L), right (R), top (T), and bottom (B). The individual I/O pad is indicated by a single letter (either A, B, C, or D) placed at the end of the PIC name. As an example, PL10A indicates a pad located on the left side of the array in the tenth row.

Each PIC interfaces to four bond pads and contains the necessary routing resources to provide an interface between I/O pads and the PLCs. Each PIC is composed of four programmable I/Os (PIOs) and significant routing resources. Each PIO contains input buffers, output buffers, routing resources, latches/FFs, and logic and can be configured as an input, output, or bidirectional I/O.

PICs in the Series 3 FPGAs have significant local routing resources, similar to routing in the PLCs. This new routing increases the ability to fix user pinouts prior to placement and routing of a design and still maintain routability. The flexibility provided by the routing also provides for increased signal speed due to a greater variety of signal paths possible.

Included in the PIC routing is a fast path from the input pins to the SLICs in each of the three adjacent PLCs (one orthogonal and two diagonal). This feature allows for input signals to be very quickly processed by the SLIC decoder function and used on-chip or sent back off of the FPGA. Also new to the Series 3 PIOs are latches and FFs and options for using fast, dedicated clocks called ExpressCLKs. These features will all be discussed in subsequent sections.

A diagram of a single PIO (one of four in a PIC) is shown in Figure 22. Table 9 provides an overview of the programmable functions in an I/O cell.

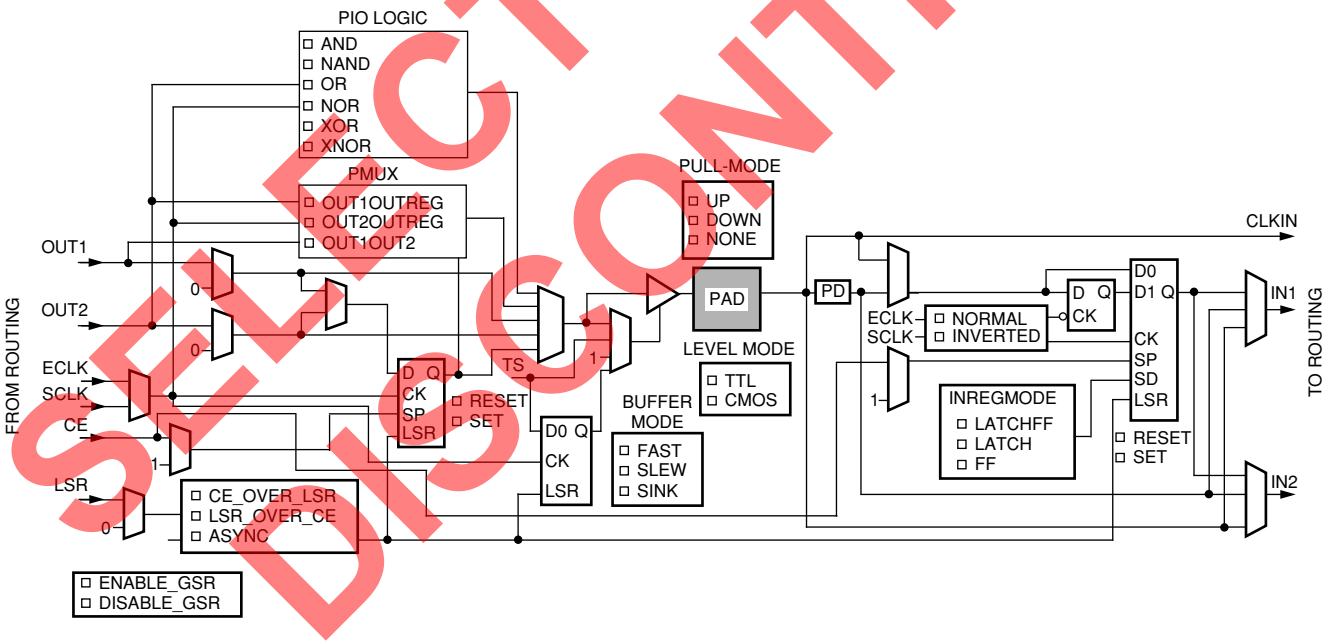


Figure 22. OR3C/Txxx Programmable Input/Output (PIO) Image from ispLEVER

Programmable Input/Output Cells

(continued)

Table 9. PIO Options

Input	Option
Input Level	TTL, OR3Cxx only CMOS, OR3Cxx or OR3Txxx 3.3 V PCI Compliant, OR3Txxx 5 V PCI Compliant, OR3Txxx
Input Speed	Fast, Delayed
Float Value	Pull-up, Pull-down, None
Register Mode	Latch, FF, Fast Zero Hold FF, None (direct input)
Clock Sense	Inverted, Noninverted
Input Selection	Input 1, Input 2, Clock Input
Output	**Option**
Output Drive Current	12 mA/6 mA or 6 mA/3 mA
Output Function	Normal, Fast Open Drain
Output Speed	Fast, Slewlim, Sinklim
Output Source	FF Direct-out, General Routing
Output Sense	Active-high, Active-low
3-State Sense	Active-high, Active-low (3-state)
FF Clocking	ExpressCLK, System Clock
Clock Sense	Inverted, Noninverted
Logic Options	See Table 10.
I/O Controls	**Option**
Clock Enable	Active-high, Active-low, Always Enabled
Set/Reset Level	Active-high, Active-low, No Local Reset
Set/Reset Type	Synchronous, Asynchronous
Set/Reset Priority	CE over LSR, LSR over CE
GSR Control	Enable GSR, Disable GSR

5 V Tolerant I/O

The I/O on the OR3Txxx Series devices allow interconnection to both 3.3 V and 5 V devices (selectable on a per-pin basis).

The OR3Txxx devices will drive the pin to the 3.3 V levels when the output buffer is enabled. If the other device being driven by the OR3Txxx device has TTL-compatible inputs, then the device will not dissipate much input buffer power. This is because the OR3Txxx output is being driven to a higher level than the TTL level required. If the other device has a CMOS-compatible input, the amount of input buffer power will also be small. Both of these power values are dependent upon the input buffer characteristics of the other device when driven at the OR3Txxx output buffer voltage levels.

The OR3Txxx device has internal programmable pull-ups on the I/O buffers. These pull-up voltages are always referenced to V_{DD} and are always sufficient to pull the input buffer of the OR3Txxx device to a high state. The pin on the OR3Txxx device will be at a level 1.0 V below V_{DD} (minimum of 2.0 V with a minimum V_{DD} of 3.0 V). This voltage is sufficient to pull the external pin up to a 3.3 V CMOS high input level (1.8 V, min) or a TTL high input level (2.0 V, min) in a 5 V tolerant system. Therefore, in a 5 V tolerant system using 5 V CMOS parts, care must be taken to evaluate the use of these pull-ups to pull the pin of the OR3Txxx device to a typical 5 V CMOS high input level (2.2 V, min).

PCI Compliant I/O

The I/O on the OR3Txxx Series devices allows compliance with PCI Local Bus (Rev. 2.2) 5 V and 3.3 V signaling environments. The signaling environment used for each input buffer can be selected on a per-pin basis. The selection provides the appropriate I/O clamping diodes for PCI compliance. Choosing an IBT input buffer will provide PCI compliance in OR3Txxx devices. OR3Cxx devices have PCI Local Bus compliant I/Os for 5 V signaling.

Programmable Input/Output Cells

(continued)

Inputs

As outlined earlier in Table 9, there are six major options on the PIO inputs that can be selected in the ispLEVER tools. For OR3Cxx devices, the inputs and bidirectional buffers can be configured as either TTL or CMOS compatible. OR3Txxx devices support CMOS levels only for input or bidirectional buffers, have 5 V tolerant I/Os as previously explained, but can optionally be selected on a pin-by-pin basis to be PCI bus 3.3 V signaling compliant (PCI bus 5 V signaling compliance occurs in 5 V tolerant operation). The default buffer upon powerup for the unused sites is 5 V tolerant/5 V PCI compliant. Consult the *ORCA* macro library, Series 3 I/O cells, for the appropriate buffers. Inputs may have a pull-up or pull-down resistor selected on an input for signal stabilization and power management. Input signals in a PIO can be passed to PIC routing on any of three paths, two general signal paths into PIC routing, and/or a fast route into the clock routing system.

There is also a programmable delay available on the input. When enabled, this delay affects the IN1 and IN2 signals of each PIO, but not the clock input. The delay allows any signal to have a guaranteed zero hold time when input. This feature is discussed subsequently.

Inputs should have transition times of less than 500 ns and should not be left floating. If any pin is not used, it is 3-stated with an internal pull-up resistor enabled automatically after configuration.

Warning: During configuration, all OR3Txxx inputs have internal pull-ups enabled. If these inputs are driven to 5 V, they will draw substantial current (≅ 5 mA). This is due to the fact that the inputs are pulled up to 3 V.

Floating inputs increase power consumption, produce oscillations, and increase system noise. The OR3Cxx inputs have a typical hysteresis of approximately 280 mV (200 mV for the OR3Txxx) to reduce sensitivity to input noise. The PIC contains input circuitry which provides protection against latch-up and electrostatic discharge.

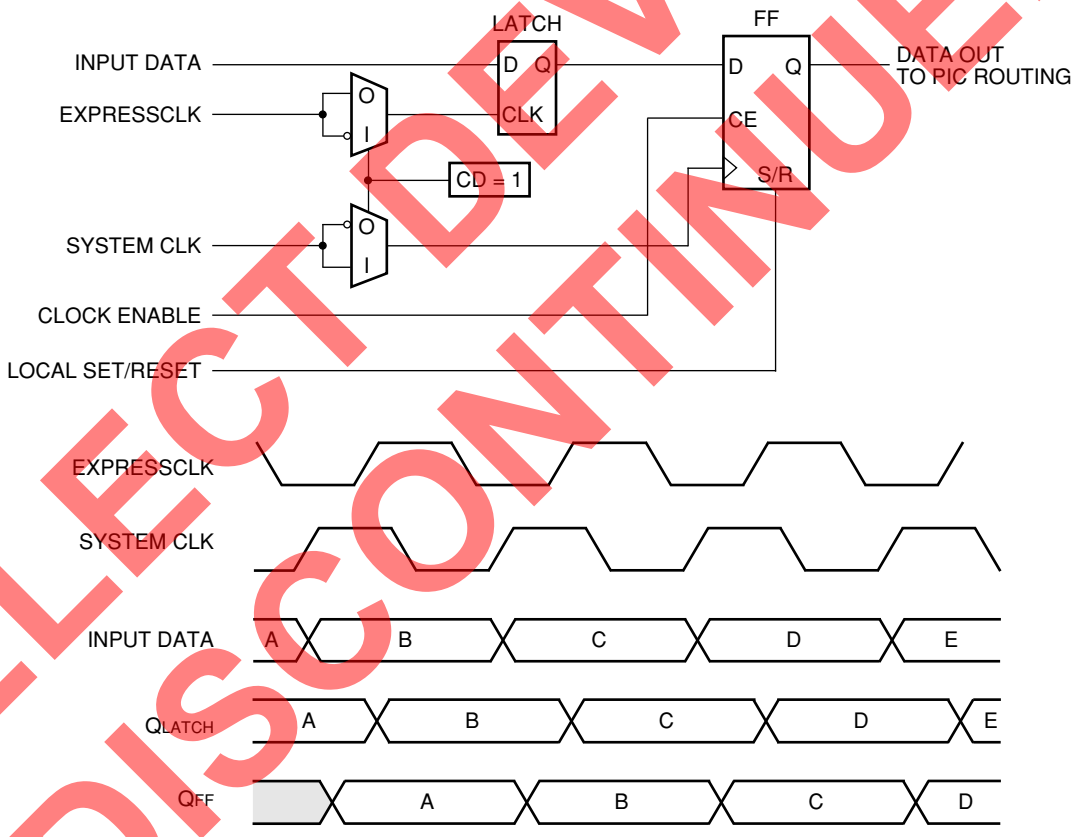
The other features of the PIO inputs relate to the new latch/FF structure in the input path. As shown in Figure 23, the input is optionally passed to a register or latch/register pair. These structures can operate in the modes listed in Table 9. In latch mode, the input signal is fed to a latch that is clocked by a system clock signal. The clock may be inverted or noninverted from its sense in the PIC routing. There is also a local set/reset signal to the latch from the PIC routing. The senses of these signals are also programmable as well as the capability to enable or disable the global set/reset signal and select the set/reset priority. The same control signals may also be used to control the input latch/FF when it is configured as a FF instead of a latch, with the addition of another control signal used as a clock enable.

Programmable Input/Output Cells (continued)

Zero-Hold Input

There are two options for zero-hold input capture in the PIO. If input delay mode is selected to delay the signal from the input pin, data can be either registered or latched with guaranteed zero-hold time in the PIO using a system clock.

To guarantee zero hold, the system clock spine structure must be used for clocking, as will be discussed later. The fast zero-hold mode of the PIO input takes advantage of the latch/FF combination and sources the input FF data from a dedicated latch that is clocked by the ExpressCLK from the PIC. The ExpressCLK is a clock from a dedicated input pin designed for fast, low-skew operation at the I/Os and is described more fully in the Clock Distribution Network and PIC Interquad (MID) Routing sections that follow. The combination of ExpressCLK latch and system clock FF guarantees a zero-hold capture of input data in the PIO FF, while at the same time reducing input setup time. Figure 23 shows a schematic of the fast-capture latch/FF and a sample timing diagram.



5-5974(F)

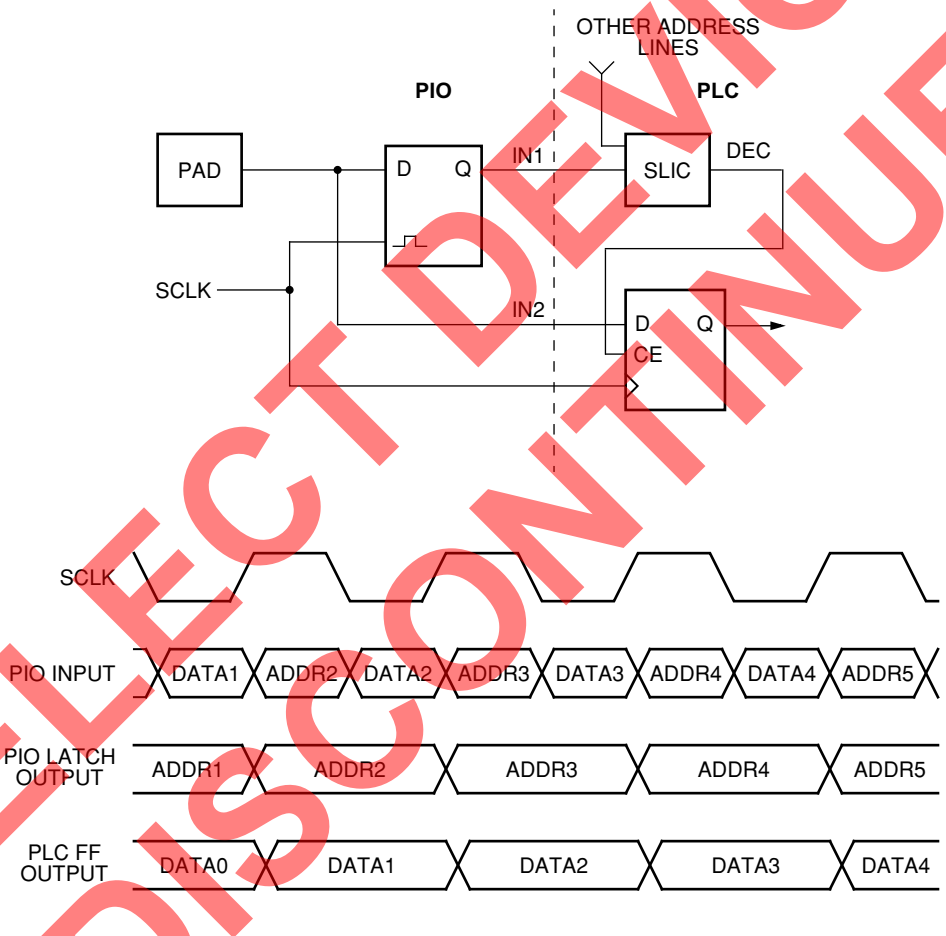
Note: CE and LSR signals not shown.

Figure 23. Fast-Capture Latch and Timing

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)

Outputs

The PIC's output drivers have programmable drive capability and slew rates. Three propagation delays (fast, slewlim, sinklim) are available on output drivers. The sinklim mode has the longest propagation delay and is used to minimize system noise and minimize power consumption. The fast and slewlim modes allow critical timing to be met.

The drive current is 12 mA sink/6 mA source for the slewlim and fast output speed selections and 6 mA sink/3 mA source for the sinklim output. Two adjacent outputs can be interconnected to increase the output sink/source current to 24 mA/12 mA.

All outputs that are not speed critical should be configured as sinklim to minimize power and noise. The number of outputs that switch simultaneously in the same direction should be limited to minimize ground bounce. To minimize ground bounce problems, locate heavily loaded output buffers near the ground pads. Ground bounce is generally a function of the driving circuits, traces on the printed-circuit board, and loads and is best determined with a circuit simulation.

At powerup, the output drivers are in slewlim mode, and the input buffers are configured as TTL-level compatible (CMOS for OR3Txxx) with a pull-up. If an output is not to be driven in the selected configuration mode, it is 3-stated.

The output buffer signal can be inverted, and the 3-state control signal can be made active-high, active-low, or always enabled. In addition, this 3-state signal can be registered or nonregistered. Additionally, there is a fast, open-drain output option that directly connects the output signal to the 3-state control, allowing the output buffer to either drive to a logic 0 or 3-state, but never to drive to a logic 1. Because there is no explicit route required to create the open-drain output, its response is very fast. Like the input side of the PIO, there are two output connections from PIC routing to the output side of the PIO, OUT1, and OUT2. These connections provide for flexible routing and can be used in data manipulation in the PIO as described in subsequent paragraphs.

An FF has been added to the output path of the PIO. The register has a local set/reset and clock enable. The LSR has the option to be synchronous or asynchronous and have priority set as clock enable over LSR or LSR over clock enable. Clocking to the output FF can come from either the system clock or the ExpressCLK associated with the PIC. The input to the FF can come from either OUT1 or OUT2, or it can be tied to VDD or GND. Additionally, the input to the FF can be inverted.

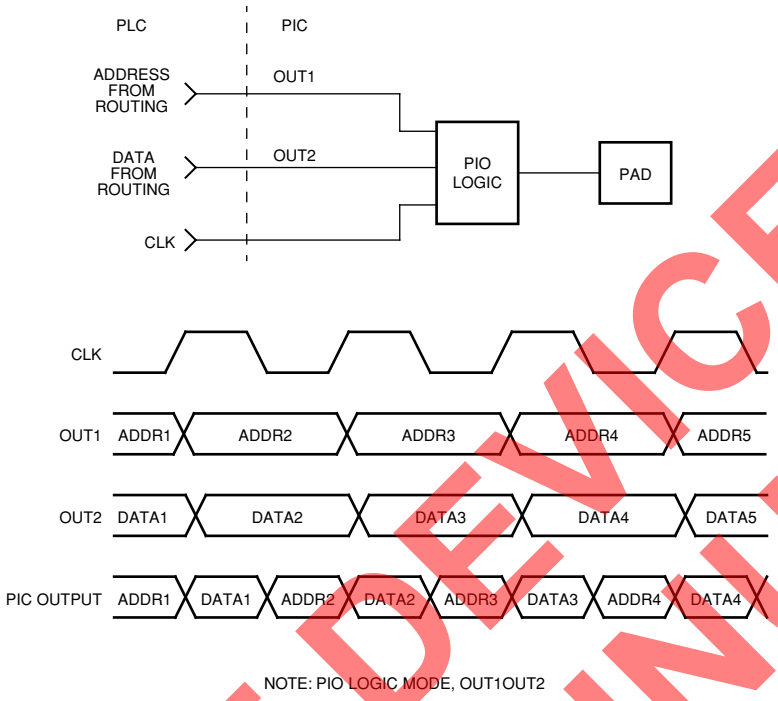
Output Multiplexing

The Series 3 PIO output FF can be combined with the new PIO logic block to perform output data multiplexing with no PLC resources required. The PIO logic block has three multiplexing modes: OUT1OUTREG, OUT2OUTREG, and OUT1OUT2. OUT1OUTREG and OUT2OUTREG are equivalent except that either OUT1 or OUT2 is MUXed with the FF, where the FF data is output on the clock phase after the active edge. The simplest multiplexing mode is OUT1OUT2. In this mode, the signal at OUT1 is output to the pad while the clock is low, and the signal on OUT2 is output to the pad when the clock is high. Figure 25 shows a simple schematic of a PIO in OUT1OUT2 mode and a general timing diagram for multiplexing an address and data signal.

Often an address will be used to generate or read a data sample from memory with the goal of multiplexing the data onto a single line. In this case, the address often precedes the data by one clock cycle. OUT1OUTREG and OUT2OUTREG modes of the PIO logic can be used to address this situation.

Because OUT1OUTREG mode is equivalent to OUT2OUTREG, only OUT2OUTREG mode is described here. Figure 26 shows a simple PIO schematic in OUT2OUTREG mode and general timing for multiplexing data with a leading address. The address signal on OUT1 is registered in the PIO FF. This delays the address so that it aligns with the data signal. The PIO logic block then sends the OUTREG signal (address) to the pad when the clock is high and the OUT2 signal (data) to the pad when the clock is low, resulting in an aligned, multiplexed signal.

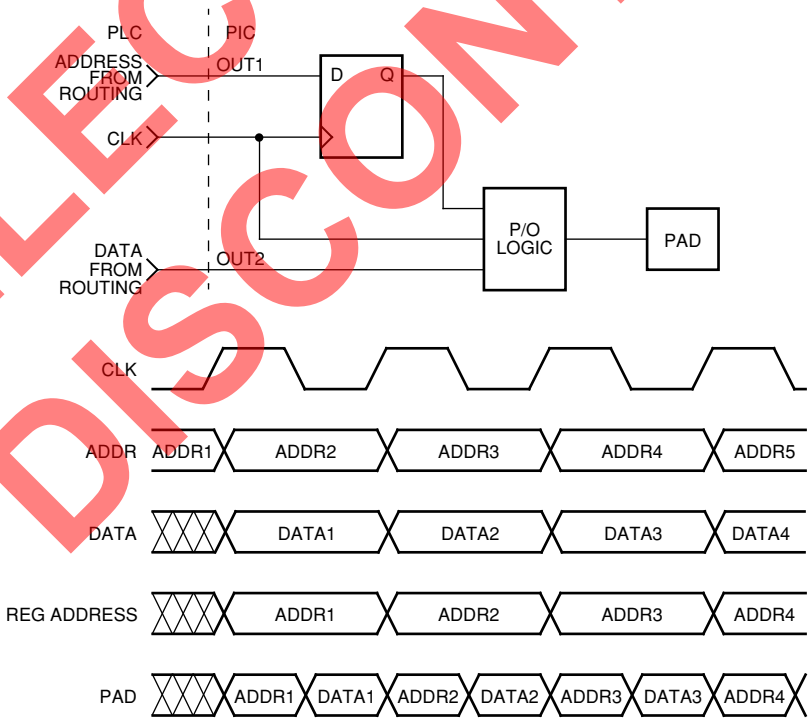
Programmable Input/Output Cells (continued)



NOTE: PIO LOGIC MODE, OUT1OUT2

5-5799(F)

Figure 25. Output Multiplexing (OUT1OUT2 Mode)



NOTE: PIO LOGIC MODE, OUT1OUT2

5-5797(F)

Figure 26. Output Multiplexing (OUT2OUTREG Mode)

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.

Programmable Input/Output Cells

(continued)

PIC Routing Resources

The PIC routing borrows many of the concepts and constructs from the PLC routing. It is designed to be able to gather an 8-bit bidirectional bus from any eight consecutive I/O pads and route them to either or both of the two adjacent PLCs. The eight I/O bits do not need to start at a PIC boundary; that is, they may start at one of the middle two PIOs in a PIC and span three PICs.

Substantial routing has been added to the PIC to offload PLC routing from being used to move signals around the PLC array perimeter. This saves PLC routing for logic purposes and provides greater flexibility for locking design pinouts prior to final placement and routing of the device, or allowing a change in the pinout late in the design cycle. The PIC routing has also been increased substantially to allow routing to the complex PIO cells that now allow multiple inputs and outputs per device pin, along with new sequential control signals, such as clock enable, LSR, and clock.

PICs are grouped in pairs for purposes of discussing PIC routing. On the sides of a device, the PICs in a pair are referred to as top and bottom. On the top or bottom of a device, the PICs in a pair are referred to as left or right. For example, on the top edge of the device, the leftmost PIC, PT1, is the left PIC of a pair, and PIC PT2 is the right PIC of that pair. The next PIC to the right, PT3, is the left PIC of the next pair, and so on.

The need for PIC pairs stems from the routing of switching segments and PLC half- and long-line drivers. As described below, the connectivity for these types of routing is grouped across pairs of PICs to provide complete and fast routing of I/O signals between a given PIC and the three adjacent PLCs: one orthogonal and two diagonal.

PIC routing segments use the same terminology as PLC routing segments, but are prefixed with a p to distinguish them as belonging to the PICs.

PIC Switching Segments. Each PIC has two groups of switching segments (pSW), each group having eight lines with connectivity to the PIOs in groups of four. One set of switching segments connects to the PIC to the left (above), and the other set connects to the switching segments of the PIC to the right (below). This means of connectivity between PICs using staggered connections of groups of switching segments allows a given PIC to route signals to both adjacent PICs and all adjacent PLCs efficiently. This provides single signal routing flexibility and routing of multiple buses on groups of I/Os without tying up global routing resources.

px1 Routing Segments. There are five px1 routing segments in each PIC that run parallel to the edge of the chip on which the PIC resides, each broken by a CIP in each PIC. The px1 segments have connectivity to the pSW segments and to the x1 routing segments of the two adjacent PLCs.

px2 Routing Segments. There are five px2 routing segments in each PIC that run parallel to the edge of the chip on which the PIC resides. To provide greater routing flexibility, the CIPs that break the px2 segments every two PICs are staggered across the two PICs in a pair. One PIC of the pair has break CIPs on the even-numbered px2 segments, and the other has them on the odd-numbered px2 segments. The px2 segments have connectivity to the pSW segments and to the x1 routing segments of the two adjacent PLCs.

px5 Routing Segments. There are ten px5 routing segments in each PIC that run parallel to the edge of the chip on which the PIC resides. Two of the ten segments are broken in each PIC so that each segment is broken every five PICs. All ten px5 segments break at the corners of the chip, allowing independent px5 routing on each edge of the chip. The px5 routing segments connect to the pSW segments and the x5 and xH routing segments of the two adjacent PLCs.

pxH Routing Segments. Each PIC contains eight pxH routing segments that run parallel to the edge of the chip on which the PIC resides. The pxH segments have connectivity with the xL, xH, and one set of xBID routing segments in the immediately adjacent PLC.

pxL Routing Segments. There are ten pxL routing segments in each PIC that run parallel to the edge of the chip on which the PIC resides. Each of the xL lines makes a connection to an xL line from the adjacent PLC. PIC long lines (xL) can be used for global signal distribution just as PLC xL lines can.

Programmable Input/Output Cells

(continued)

PIC Architectural Description

The PIC architecture as seen in ispLEVER is shown in Figure 27. The figure is the left PIC of a PIC pair on the top edge of a Series 3 array. Both PICs in a pair are similar, with the differences mainly lying in the connections between the PIC switching segments (pSW), the IN2 connections across PIC boundaries, and the system clock spine driver residing in only one PIC of a pair.

A. This is a programmable input/output (PIO). There are four PIOs per PIC. The PIOs contain the PIC logic and I/O buffers.

B. This is the PIC output switching block. It connects the PIC switching segments and local clock lines to the PIO output and control signals.

C. This is the system clock spine switching block and buffer. There is only one system clock spine per pair of PICs. Its inputs can come from the PIC switching segments or any of the eight PIO inputs in a PIC pair.

D. PIC switching segments (pSW). These routing segments are used to interconnect routing resources within the PIC and to a lesser degree, between PICs.

E. px1 routing segments. The PIC x1 routing segments traverse one PIC and break at a CIP in the middle of each PIC.

F. px2 routing segments. The PICs have routing that traverses two PICs between breaks. The breaks are staggered among the five px2 segments.

G. px5 routing segments. Each of the ten PIC x5 routing segments traverses five PICs in between breaks at a CIP. Two px5 segments break in each PIC.

H. pxH routing segments. The eight PIC xH routing segments traverse half of the array and break at CIPs in the interquad routing region that is in the middle of the array.

I. (Not used intentionally for clarity.)

J. pxL routing segments. The PIC long lines run the entire length of the side of the array.

K. x5 routing segments from the adjacent PLC routing.

L. xL routing segments from the adjacent PLC routing.

M. x1 routing segments from the adjacent PLC routing.

N. Switching segments from the adjacent PLC routing.

O. xH routing segments from the adjacent PLC routing.

P. BIDI routing segments from the adjacent PLC routing.

Q. These are the IN2 routing segments. There is one IN2 line from each PIO, and all eight IN2 lines from each PIC pair are present in both PICs of a pair.

R. These CIPs connect the IN1 and IN2 routing segments from the PIOs to the PIC switching segments.

S. These CIPs break the PIC switching segments at the interface between a PIC pair.

T. These CIPs connect adjacent PLC routing resources to the PIC switching segments.

U. These CIPs connect inter-PIC routing with the PIC switching segments.

V. These CIPs break the px1, px2, and px5 routing at the middle of a PIC. The px2 and px5 CIP placement varies depending on the PLC.

W. These mutually exclusive buffers can drive one long line signal onto a PIC local clock routing segment.

X. These mutually exclusive buffers can select a source from one of the local system clock routes to drive the PIO 3-state control signal.

Y. These are the four local system clock routing segments. Two come from connections within the PIC, one from the other PIC in the pair, and one from the adjacent PLC.

Z. These mutually exclusive buffers allow a signal on the PIC switching segments to be routed to a system clock spine or to a PIO system clock.

AA. ExpressCLK routing line.

AB. System clock spine.

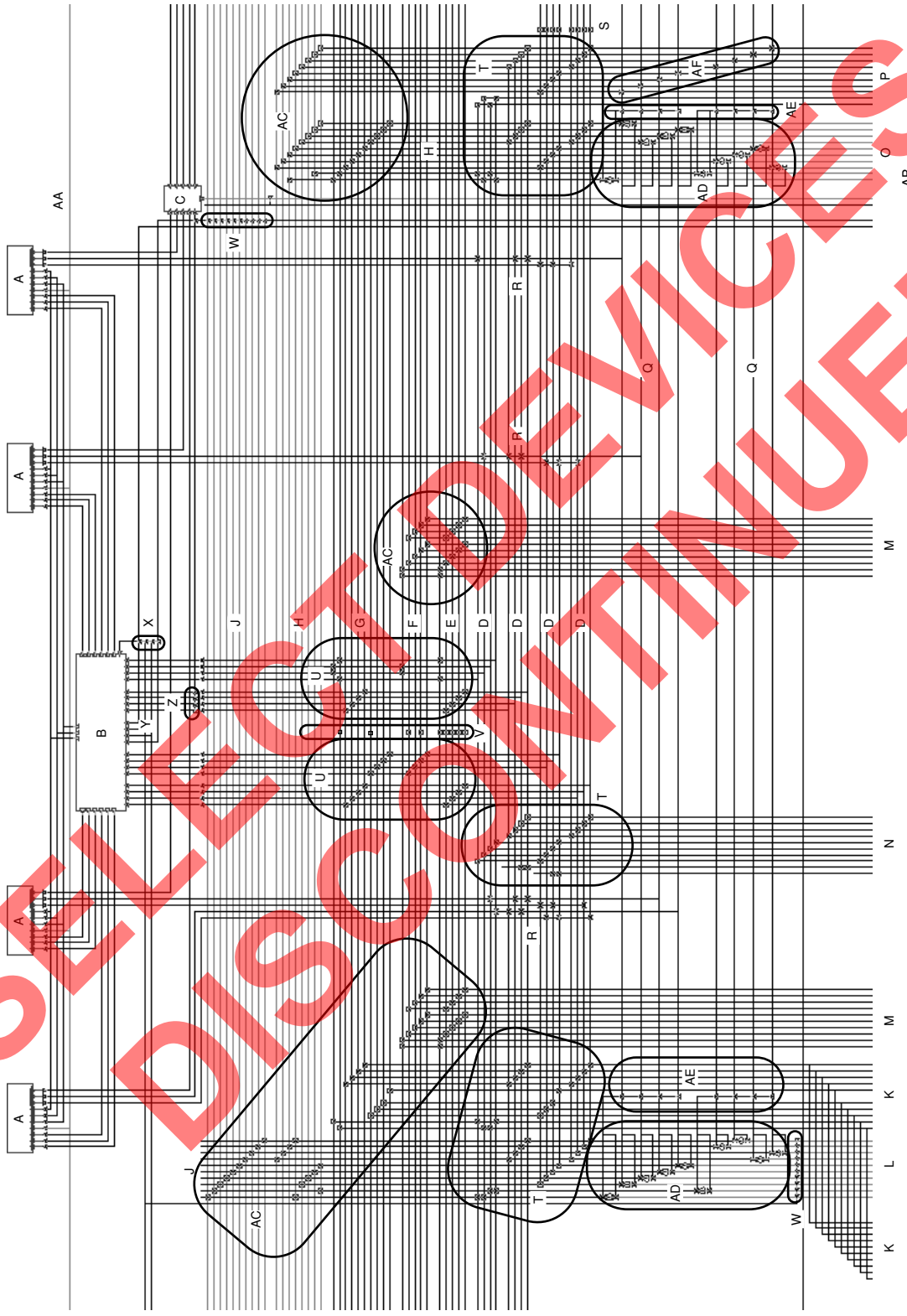
AC. These various groups of CIPs connect routing resources from the adjacent PLC to the inter-PIC routing resources.

AD. These buffers provide connectivity between the PLC xL (xH) lines and the PIC xL (xH) lines **or** connectivity between one of the IN2 routing segments and the PIC and/or PLC xL (xH) routing segments.

AE. These mutually exclusive buffers and CIPs provide connectivity to the PLC xL and xH lines from one of the IN2 input segments.

AF. These buffers allow the IN2 signals to drive onto the BIDI routing of the adjacent PLC, or the BIDI routing of the adjacent PLC, and the PIC switching segments and/or PIC half lines may be connected.

Programmable Input/Output Cells (continued)



5-5823(F)

Figure 27. PIC Architecture

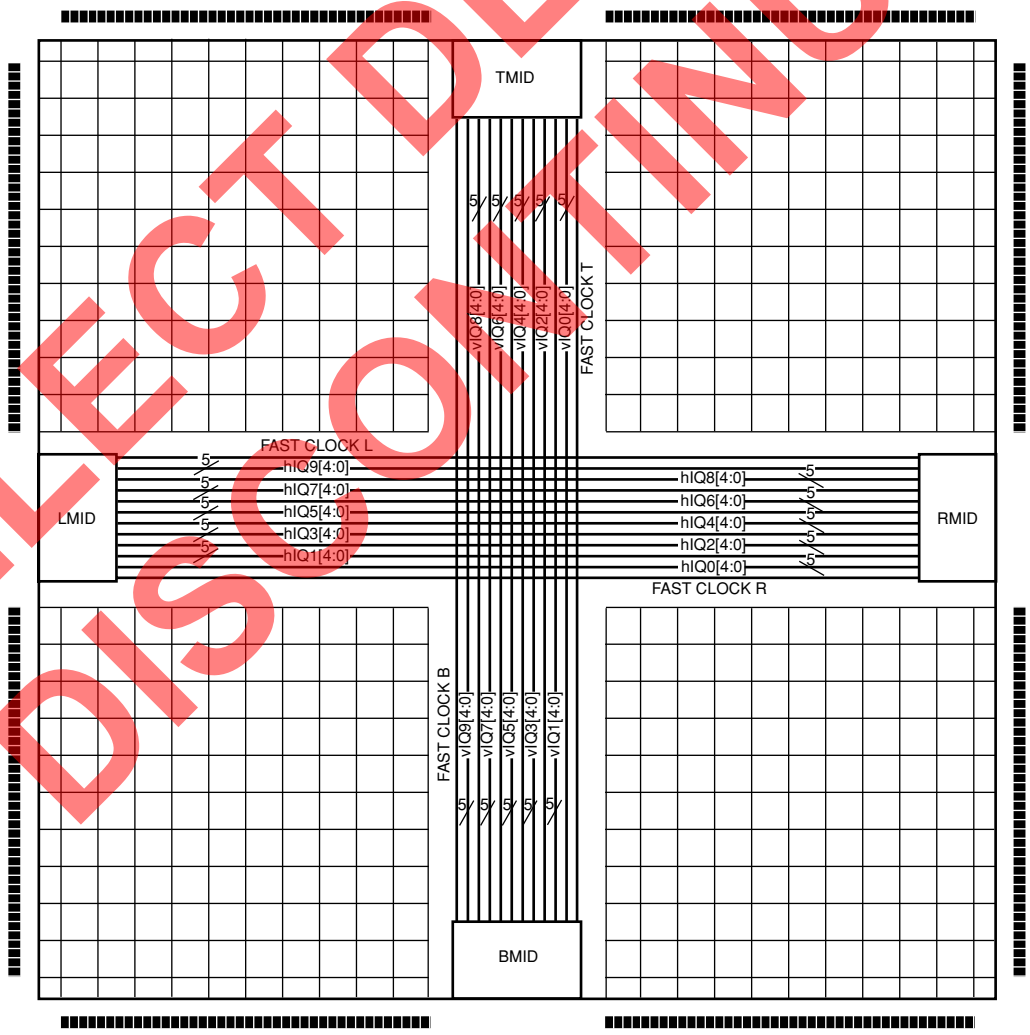
High-Level Routing Resources

The high-level routing resources in the *ORCA* Series 3 devices are interquad routing, corner cell routing, and PIC interquad routing. These resources and their related structures are discussed in the following subsections.

Interquad Routing

In the *ORCA* Series 3 devices, the PLC array is split into four equal quadrants. In between these quadrants, routing has been added to route signals between the quadrants and distribute clocks. In addition to general routing, there are four specialized clock routing spines. The general routing is discussed below, followed by the special clock routing.

One of the main purposes of interquad routing is to distribute internally generated signals, such as clocks and control signals. There are two types of interquad blocks: vertical and horizontal. Vertical interquad blocks (vIQ) run between quadrants on the left and right, while horizontal interquad blocks (hIQ) run between top and bottom quadrants. Interquad lines begin and end in the MID cells that are discussed later. Since hIQ and vIQ blocks have the same logic, only the hIQ block is described below. The interquad routing connects to x5 and xH segments. It does not affect other local routing (xsw, x1, fast carry), so local routing is the same, whether PLC-PLC connections cross quadrants or not. Figure 28 presents a (not to scale) view of interquad routing.



5-4538(F)

Figure 28. Interquad Routing

High-Level Routing Resources (continued)

Figure 29 shows the connections from the interquad routing to the inter-PLC routing for a block of the horizontal interquad. The vertical interquad has similar connections. The connections shown in Figure 29 are made with PLCs located above and below the routing shown in the figure. The interquad routing segments, prefixed IH for interquad horizontal, are in ten groups of five lines. Any one line from each group can be routed to one of the xH segments from the top of the device (left for vertical interquad), one of the xH segments from the bottom of the device (right for vertical interquad), and one of the x5 segments crossing the interquad.

Figure 28 shows four fast middle clock (fast clock) signals with the suffixes T (top), B (bottom), R (right), and L (left), respectively. Figure 29 also shows the fast clock R and fast clock L lines; these are dedicated interquad clock spines. They originate in the CLKCNTRL special function blocks in the middle of each edge of the device, with the name referencing the edge of origin. For example, fast clock R originates in the CLKCNTRL block on the right edge of a device. Fast clock spines traverse the entire PLC array but do not connect to the PICs on the edge of the device opposite to the source. Each fast clock line connects to two of the xL lines in each PLC that run orthogonally to the fast clock. These connections allow the fast clock lines to generate a clock tree that can reach any PLC in the device. Fast clocks and other clock resources are discussed in the Clock Distribution Network section.

Programmable Corner Cell Routing

Programmable Routing

The programmable corner cell (PCC) contains the circuitry to connect the routing of the two PICs in each corner of the device. The PIC px1 and px2 segments and eight PIC switching segments are directly connected together from one PIC to another. The px5 lines are all broken with CIPs and the PIC pxL and pxH segments are connected from one block to another through programmable buffers.

Corner Cell Special Functions

In addition to routing functions, special-purpose functions are located in each FPGA corner. The upper-left PCC contains connections to the boundary-scan logic and microprocessor interface. The upper-right PCC contains connections to the readback logic, connectivity to the global 3-state signal (TS_ALL), and a programmable clock manager. The lower-left PCC contains connections to the internal oscillator and a programmable clock manager. The lower-right PCC contains connections to the start-up and global reset logic. These functions are all more completely described in the Special Function Blocks section of this data sheet.

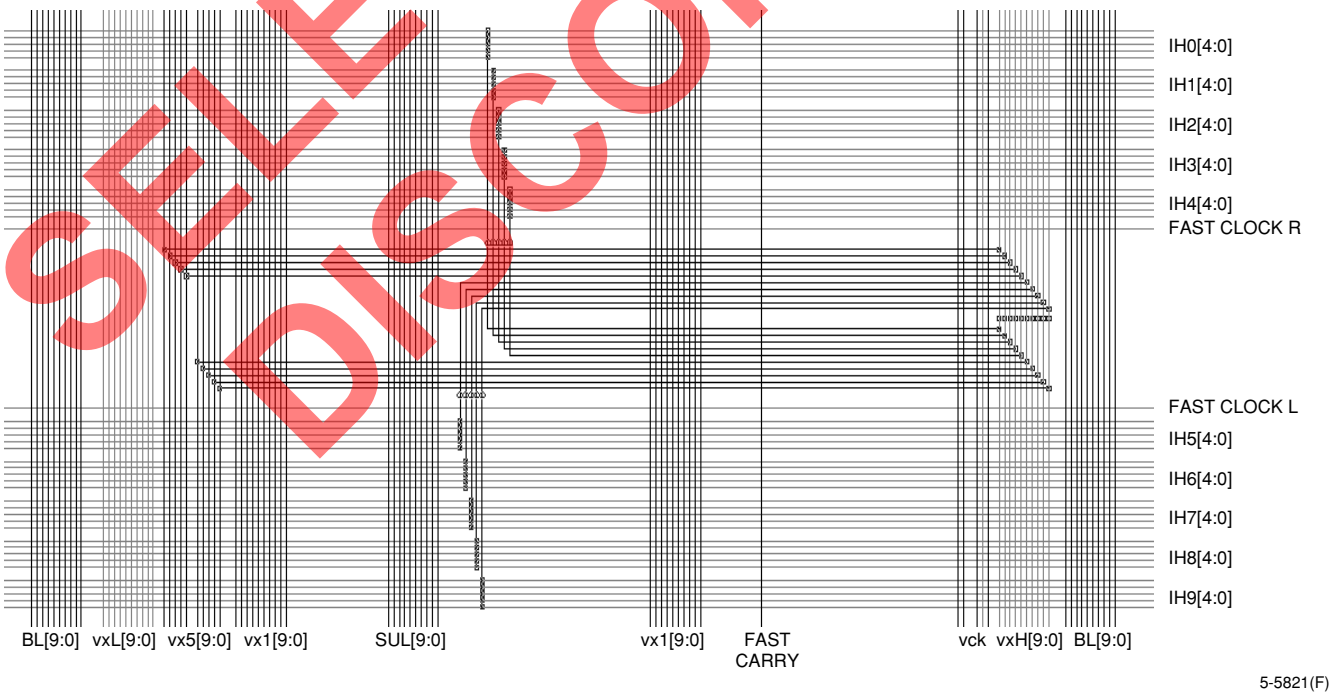


Figure 29. hIQ Block Detail

High-Level Routing Resources (continued)

PIC Interquad (MID) Routing

There is also connectivity between the PICs in each quadrant, as well as a clock control (CLKCNTRL) module (discussed in the Special Function Blocks section) between the PIC routing and the interquad routing. These blocks are called LMID (left), TMID (top), RMID (right), and BMID (bottom). The TMID routing is shown in Figure 30. As with the hIQ and vIQ blocks, the only connectivity to the PIC routing is to the global pxH and px5 segments.

The pxH segments from the one quadrant can be connected through a CIP to its counterpart in the opposite quadrant, providing a path that spans the array of PICs. Since a passive CIP is used to connect the two pxH segments, a 3-state signal can be routed on the two pxH segments in the opposite quadrants, and then connected through this CIP. As with the hIQ and vIQ blocks, CIPs and buffers allow nibble-wide connections between the interquad segments, the xH segments, and the x5 segments.

5-5822(F)

Figure 30. Top (TMID) Routing

Clock Distribution Network

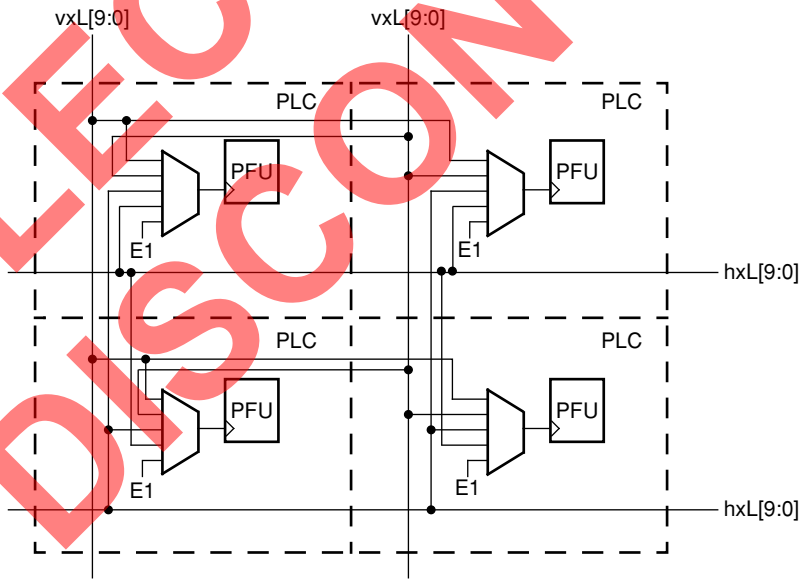
The Series 3 FPGAs provide three types of high-speed, low-skew clock distributions: system clock, fast middle clock (fast clock), and ExpressCLK. Because of the great variety of sources and distribution for clock signals in the *ORCA* Series 3, the clock mechanisms will be described here from the inside out. The clock connections to the PFU will be described, followed by clock distribution to the PLC array, clock sources to the PLC array, and finally ending with clock sources and distribution in the PICs. The ExpressCLK inputs are new, dedicated clock inputs in Series 3 FPGAs. They are mentioned in several of the clock network descriptions and are described fully later in this section.

PFU Clock Sources

Within a PLC there are five sources for the clock signal of the latches/FFs in the PFU. Two of the signals are generated off of the long lines (xL) within the PLC: one from the set of vertical long lines and one from the set of horizontal long lines. For each of these signals, any one of the ten long lines of each set, vertical or horizontal, can generate the clock signal. Two of the five PFU clock sources come from neighboring PLCs. One clock is generated from the PLC to the left or right of the current PLC, and one is generated from the PLC above or below the current PLC. The selection decision as to where these signals come from, above/below and left/right, is based on the position of the PLC in the array and has to do with the alternating nature of the source of the system clock spines (discussed later). The last of the five clock sources is also generated within the PLC. The E1 control signal, described in the PLC Routing Resources section, can drive the PFU clock. The E1 signal can come from any xBID routing resource in the PLC. The selection and switching of clock signals in a PLC is performed in the FINS. Figure 31 shows the PFU clock sources for a set of four adjacent PLCs.

Global Control Signals

The four clock signals in each PLC that are generated from the long lines (xL) in the current PLC or an adjacent PLC can also be used to drive the PFU clock enable (CE), local set/reset (LSR) and add/subtract/write enable (ASWE) signals. The clock signals generated from vertical long lines can drive CE and ASWE, and the clocks generated from horizontal long lines can drive LSR. This allows for low-skew global distribution of two of these three control signals with the clock routing while still allowing a global clock route to occur.



5-6054(F)

Figure 31. PFU Clock Sources

Clock Distribution Network (continued)

Clock Distribution in the PLC Array

System Clock (SCLK)

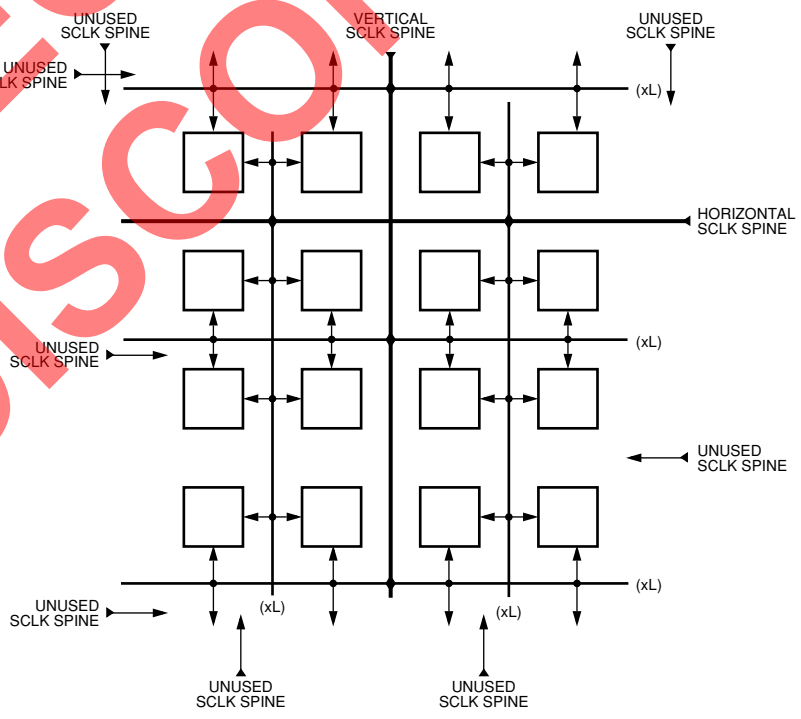
The clock distribution network, or clock spine network, within the PLC array is designed to minimize clock skew while maximizing clock flexibility. Clock flexibility is expressed in two ways: the ease with which a single clock is routed to the entire array, and the capability to provide multiple clocks to the PLC array.

There is one horizontal and one vertical clock spine passing through each PLC. The horizontal clock spine is sourced from the PIC in the same row on either the left- or right-hand side of the array, with the source side (left or right) alternating for each row. The vertical clock spines are similarly sourced from the PICs alternating from the top or bottom of a column. Each clock spine is capable of driving one of the ten xL routing segments that run orthogonal to it within each PLC. Full connectivity to all PFUs is maintained due to the connectivity from the xL lines to the PFU clock signals described in the previous section; however, only an xL line in every other row (column) needs to be driven to allow the given clock signal to be distributed to every PFU. Figure 32 is a high-level diagram of the Series 3 system clock spine network with sample xL line connections for a 4 x 4 array of PLCs.

The clock spine structure previously described provides for complete distribution of a clock from any I/O pin to the entire PLC array by means of a single clock spine and long lines (xL). This distribution system also provides a means to have many different clocks routed to many different and dispersed locations in the PLC array. Each spine can carry a different clock signal, so for the OR3T55 (which has an 18 x 18 array of PLCs, implying nine clock spines per side), 36 input clock signals can be supported using the system clock network.

Fast Clock

Fast clocks are high-speed, low-skew clock spines that originate from the CLKCNTRL special function blocks (described later). There are four fast clock spines—one originating on the middle of each edge of the array. The spines run in the interquad region of the PLC array from their source side of the device to the last row or column on the opposite side of the device. The fast clocks connect to two long lines, xL[8] and xL[9], that run orthogonal to the spine direction in each PLC. These long lines can then be connected to the PFU clock input in the same manner as the general system clocks, and, like the system clock connections, xL lines are only needed in every other row (column) to distribute a clock to every PFU. The limited number of long-line connections and the low skew of the CLKCNTRL source combine to make the fast clocks a very robust, low-skew clock source.



5-5801(F).a

Figure 32. *ORCA* Series 3 System Clock Distribution Overview

Clock Distribution Network (continued)

Clock Sources to the PLC Array

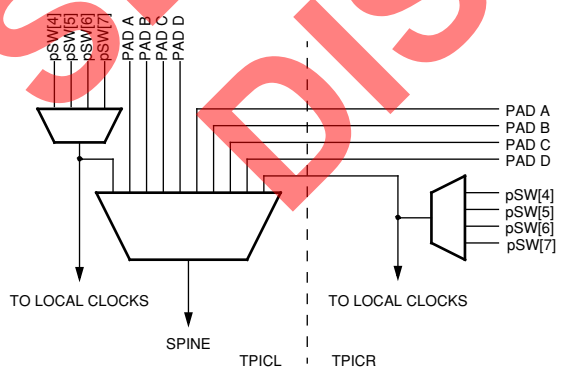
The source of a clock that is globally available to the PLC array can be from any user I/O pad, any of the ExpressCLK pads, or an internally generated source.

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.

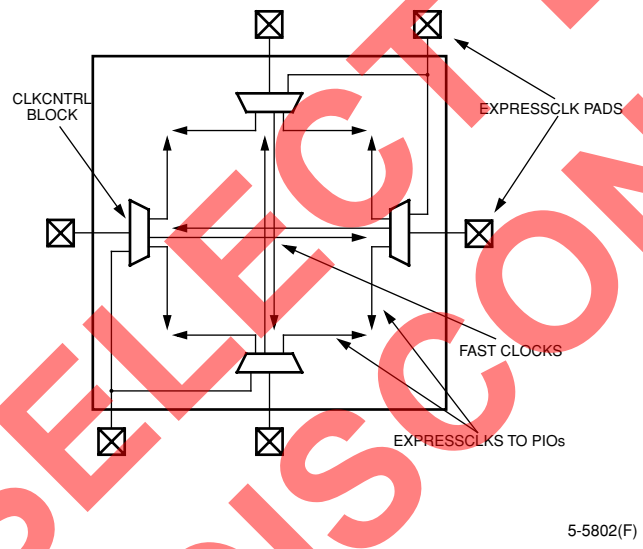
PIC ExpressCLK

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.

Clock Distribution Network (continued)

ExpressCLK Inputs

There are four dedicated ExpressCLK pads on each Series 3 device: one in the middle of each side. Two other user I/O pads can also be used as corner ExpressCLK inputs, one on the lower-left corner, and one on the upper-right corner. The corner ExpressCLK pads feed the ExpressCLK to the two sides of the array that are adjacent to that corner, always driving the same signal in both directions. The ExpressCLK route from the middle pad and from the corner pad associated with that side are multiplexed and can be glitchlessly stopped/started under user control using the StopCLK feature of the CLKCNTRL function block (described under Special Function Blocks) on that side. The ExpressCLK output of the programmable clock manager (PCM) is programmably connected to the corner ExpressCLK routes. PCM blocks are found in the same corners as the corner ExpressCLK signals and are described in the Special Function Blocks section. The ExpressCLK structure is shown in Figure 34 (PCM blocks are not shown).



Note: All multiplexers are set during configuration.

Figure 34. ExpressCLK and Fast Clock Distribution

Selecting Clock Input Pins

Any user I/O pin on an *ORCA* FPGA can be used as a fast, low-skew system clock input. Since the four dedicated ExpressCLK inputs can only be used to distribute global signals into the FPGA, these pins should be selected first as clock pins. Within the interquad region of the device, these clocks sourced by the ExpressCLK inputs are called fast clocks. Choosing the next clock pin is completely arbitrary, but using a pin that is near the center of an edge of the device will provide the lowest skew system clock network. The pin-to-pin timing numbers in the Timing Characteristics section assume that the clock pin is in one of the PICs at the center of any side of the device next to an ExpressCLK pad. For actual timing characteristics for a given clock pin, use the timing analyzer results from ispLEVER.

To select subsequent clock pins, certain rules should be followed. As discussed in the Programmable Input/Output Cells section, PICs are grouped into adjacent pairs. Each of these pairs contains eight I/Os, but only one of the eight I/Os in a PIC pair can be routed directly onto a system clock spine. Therefore, to achieve top performance, the next clock input chosen should not be one of the pins from a PIC pair previously used for a clock input. If it is necessary to have a second input in the same PIC pair route onto global system clock routing, the input can be routed to a free clock spine using the PIC switching segment (pSW) connections to the clock spine network at some small sacrifice in speed. Alternatively, if global distribution of the secondary clock is not required, the signal can be routed on long lines (xL) and input to the PFU clock input without using a clock spine.

Another rule for choosing clock pins has to do with the alternating nature of clock spine connections to the xL and pxL routing segments. Starting at the left side of the device, the first vertical clock spine from the top connects to hxL[0] (horizontal xL[0]), and the first vertical clock spine from the bottom connects to hxL[5] in all PLC rows. The next vertical clock spine from the top connects to hxL[1], and the next one from the bottom connects to hxL[6]. This progression continues across the device, and after a spine connects to hxL[9], the next spine connects to hxL[0] again. Similar connections are made from horizontal clock spines to vxL (vertical xL) lines from the top to the bottom of the device. Because the *ORCA* Series 3 clock routing only requires the use of an xL line in every other row or column, even two inputs chosen 20 PLCs apart on the same xL line will not conflict, but it is always better to avoid these choices, if possible. The fast clock spines in the interquad routing region also connect to xL[8] and xL[9] for each set of xL lines, so it is better to avoid user I/Os that connect to xL[8] or xL[9] when a fast clock is used that might share one of these connections. Another reason to use the fast clock spines is that since they use only the xL[9:8] lines, they will not conflict with internal data buses which typically use xL[7:0]. For more details on clock selection, refer to application notes on clock distribution in *ORCA* Series 3 devices.

Special Function Blocks

Special function blocks in the Series 3 provide extra capabilities beyond general FPGA operation. These blocks reside in the corners and MIDs (middle inter-quad areas) of the FPGA array.

Single Function Blocks

Most of the special function blocks perform a specific dedicated function. These functions are data/configuration readback control, global 3-state control (TS_ALL), internal oscillator generation, global set/reset (GSRN), and start-up logic.

Readback Logic

The readback logic is located in the upper right corner of the FPGA and can be enabled via a bit stream option or by instantiation of a library readback component.

Readback is used to read back the configuration data and, optionally, the state of the PFU outputs. A readback operation can be done while the FPGA is in normal system operation. The readback operation cannot be daisy-chained. To use readback, the user selects options in the bit stream generator in the ispLEVER Development System.

Table 12 provides readback options selected in the bit stream generator tool. The table provides the number of times that the configuration data can be read back. This is intended primarily to give the user control over the security of the FPGA's configuration program. The user can prohibit readback (0), allow a single readback (1), or allow unrestricted readback (U).

Table 12. Readback Options

Option	Function
0	Prohibit Readback
1	Allow One Readback Only
U	Allow Unrestricted Number of Readbacks

Readback can be performed via the Series 3 microprocessor interface (MPI) or by using dedicated FPGA readback controls. If the MPI is enabled, readback via the dedicated FPGA readback logic is disabled. Readback using the MPI is discussed in the Microprocessor Interface (MPI) section.

The pins used for dedicated readback are readback data (RD_DATA), read configuration ($\overline{\text{RD\_CFG}}$), and configuration clock (CCLK). A readback operation is initiated by a high-to-low transition on $\overline{\text{RD\_CFG}}$. The $\overline{\text{RD\_CFG}}$ input must remain low during the readback operation. The readback operation can be restarted at frame 0 by driving the $\overline{\text{RD\_CFG}}$ pin high, applying at least two rising edges of CCLK, and then driving $\overline{\text{RD\_CFG}}$ low again. One bit of data is shifted out on RD_DATA at the rising edge of CCLK. The first start bit of the readback frame is transmitted out several cycles after the first rising edge of CCLK after $\overline{\text{RD\_CFG}}$ is input low (see the Readback Timing Characteristics table in the Timing Characteristics section). To be certain of the start of the readback frame, the data can be monitored for the 01 frame start bit pair.

Readback can be initiated at an address other than frame 0 via the new microprocessor interface (MPI) control registers (see the Microprocessor Interface (MPI) section for more information). In all cases, readback is performed at sequential addresses from the start address.

It should be noted that the RD_DATA output pin is also used as the dedicated boundary-scan output pin, TDO. If this pin is being used as TDO, the RD_DATA output from readback can be routed internally to any other pin desired. The $\overline{\text{RD\_CFG}}$ input pin is also used to control the global 3-state (TS_ALL) function. Before and during configuration, the TS_ALL signal is always driven by the $\overline{\text{RD\_CFG}}$ input and readback is disabled. After configuration, the selection as to whether this input drives the readback or global 3-state function is determined by a set of bit stream options. If used as the $\overline{\text{RD\_CFG}}$ input for readback, the internal TS_ALL input can be routed internally to be driven by any input pin.

Special Function Blocks (continued)

The readback frame contains the configuration data and the state of the internal logic. During readback, the value of all registered PFU and PIC outputs can be captured. The following options are allowed when doing a capture of the PFU outputs.

1. Do not capture data (the data written to the RAMs, usually 0, will be read back).
2. Capture data upon entering readback.
3. Capture data based upon a configurable signal internal to the FPGA. If this signal is tied to logic 0, capture RAMs are written continuously.
4. Capture data on either options 2 or 3 above.

The readback frame has an identical format to that of the configuration data frame, which is discussed later in the Configuration Data Format section. If LUT memory is not used as RAM and there is no data capture, the readback data (not just the format) will be identical to the configuration data for the same frame. This eases a bitwise comparison between the configuration and readback data. The configuration header, including the length count field, is not part of the readback frame. The readback frame contains bits in locations not used in the configuration. These locations need to be masked out when comparing the configuration and readback frames. The development system optionally provides a readback bit stream to compare to readback data from the FPGA. Also note that if any of the LUTs are used as RAM and new data is written to them, these bits will not have the same values as the original configuration data frame either.

Global 3-State Control (TS_ALL)

To increase the testability of the *ORCA* Series FPGAs, the global 3-state function (TS_ALL) disables the device. The TS_ALL signal is driven from either an external pin or an internal signal. Before and during configuration, the TS_ALL signal is driven by the input pad $\overline{\text{RD\_CFG}}$. After configuration, the TS_ALL signal can be disabled, driven from the $\overline{\text{RD\_CFG}}$ input pad, or driven by a general routing signal in the upper right corner. Before configuration, TS_ALL is active-low; after configuration, the sense of TS_ALL can be inverted.

The following occur when TS_ALL is activated:

1. All of the user I/O output buffers are 3-stated, the user I/O input buffers are pulled up (with the pull-down disabled), and the input buffers are configured with TTL input thresholds (OR3Cxx only).
2. The TDO/RD_DATA output buffer is 3-stated.
3. The $\overline{\text{RD\_CFG}}$, $\overline{\text{RESET}}$, and $\overline{\text{PRGM}}$ input buffers remain active with a pull-up.
4. The DONE output buffer is 3-stated, and the input buffer is pulled up.

Internal Oscillator

The internal oscillator resides in the lower left corner of the FPGA array. It has output clock frequencies of 1.25 MHz and 10 MHz. The internal oscillator is the source of the internal CCLK used for configuration. It may also be used after configuration as a general-purpose clock signal.

Global Set/Reset (GSRN)

The GSRN logic resides in the lower right corner of the FPGA. GSRN is an invertible, default, active-low signal that is used to reset all of the user-accessible latches/FFs on the device. GSRN is automatically asserted at powerup and during configuration of the device.

The timing of the release of GSRN at the end of configuration can be programmed in the start-up logic described below. Following configuration, GSRN may be connected to the $\overline{\text{RESET}}$ pin via dedicated routing, or it may be connected to any signal via normal routing. Within each PFU and PIO, individual FFs and latches can be programmed to either be set or reset when GSRN is asserted. A new option in Series 3 allows individual PFUs and PIOs to turn off the GSRN signal to its latches/FFs after configuration.

The $\overline{\text{RESET}}$ input pad has a special relationship to GSRN. During configuration, the $\overline{\text{RESET}}$ input pad always initiates a configuration abort, as described in the FPGA States of Operation section. After configuration, the global set/reset signal (GSRN) can either be disabled (the default), directly connected to the $\overline{\text{RESET}}$ input pad, or sourced by a lower-right corner signal. If the $\overline{\text{RESET}}$ input pad is not used as a global reset after configuration, this pad can be used as a normal input pad.

Special Function Blocks (continued)

Start-Up Logic

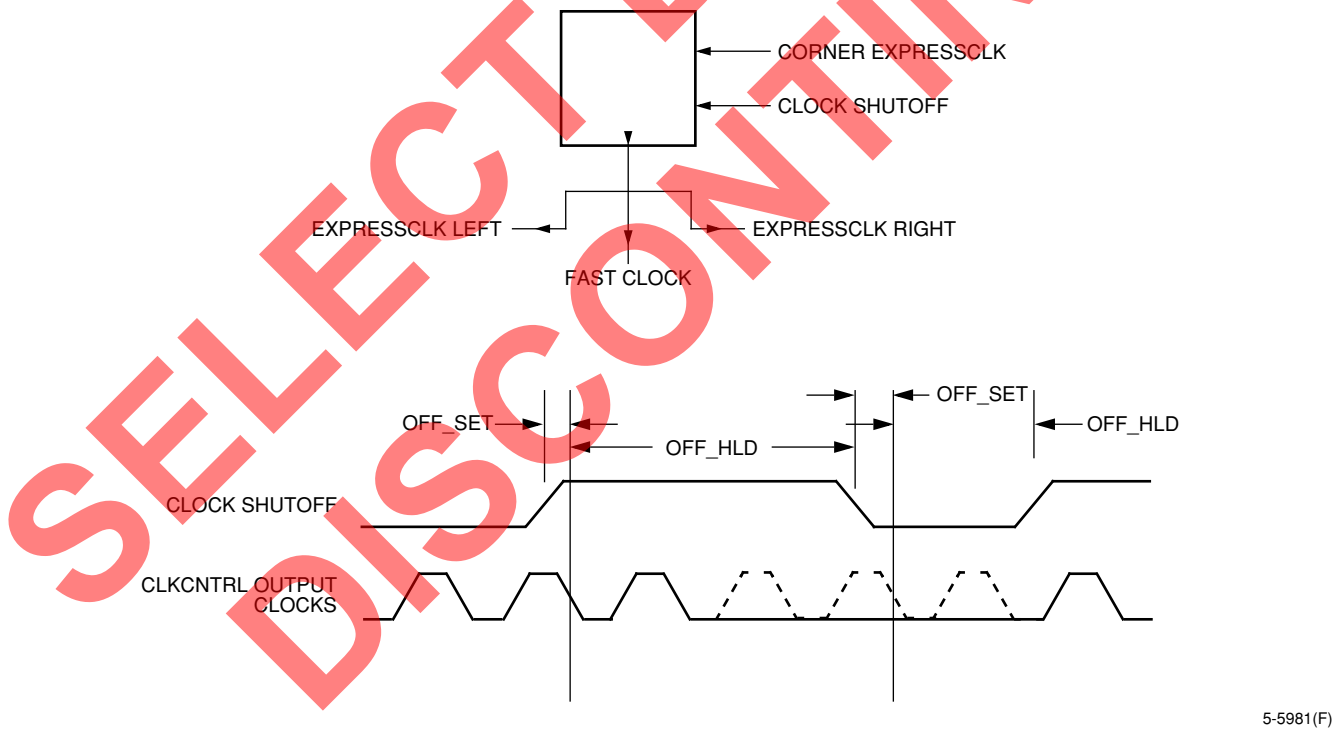
The start-up logic block is located in the lower right corner of the FPGA. This block can be configured to coordinate the relative timing of the release of GSRN, the activation of all user I/Os, and the assertion of the DONE signal at the end of configuration. If a start-up clock is used to time these events, the start-up clock can come from CCLK, or it can be routed into the start-up block using lower right corner routing resources. These signals are described in the Start-Up subsection of the FPGA States of Operation section.

Clock Control (CLKCNTRL) and StopCLK

There is one CLKCNTRL block in the MID section of the interquad routing on each side of the FPGA. This block is used to selectively distribute the fast clock to the PLC array and the left (top) and right (bottom) ExpressCLKs (ECKL and ECKR) to the side of the array on which the CLKCNTRL block resides.

The source clock for the CLKCNTRL block comes either from the ExpressCLK pad at the middle of the side of the FPGA or from the corner ExpressCLK route that comes from the corner ExpressCLK pad (at the lower left or upper right of the device, whichever is closer). The programmable clock manager ExpressCLK output can also be sourced to this corner routing for distribution at the two closest CLKCNTRL blocks.

Each CLKCNTRL block also features an invertible StopCLK shutoff input that is available from local routing. This feature may be used to glitchlessly stop and start the clock at the three outputs of each CLKCNTRL block and has the option of doing so on either the rising or falling edge of the clock. When the clock is halted based on its rising edge, it stops and stays at VDD. When it is stopped based on its falling edge, it stops and stays at GND. If the StopCLK shutoff signal meets the CLKCNTRL setup and hold times, the clock is stopped on the second clock cycle after the shutoff signal. A diagram of the bottom CLKCNTRL block and StopCLK timing is shown in Figure 35.



Notes:
CLKCNTRL output clocks are ExpressCLK left and right and fast clock.

Clock shutoff shown active-high acting on clock falling edge.

Figure 35. Top CLKCNTRL Function Block

Special Function Blocks (continued)

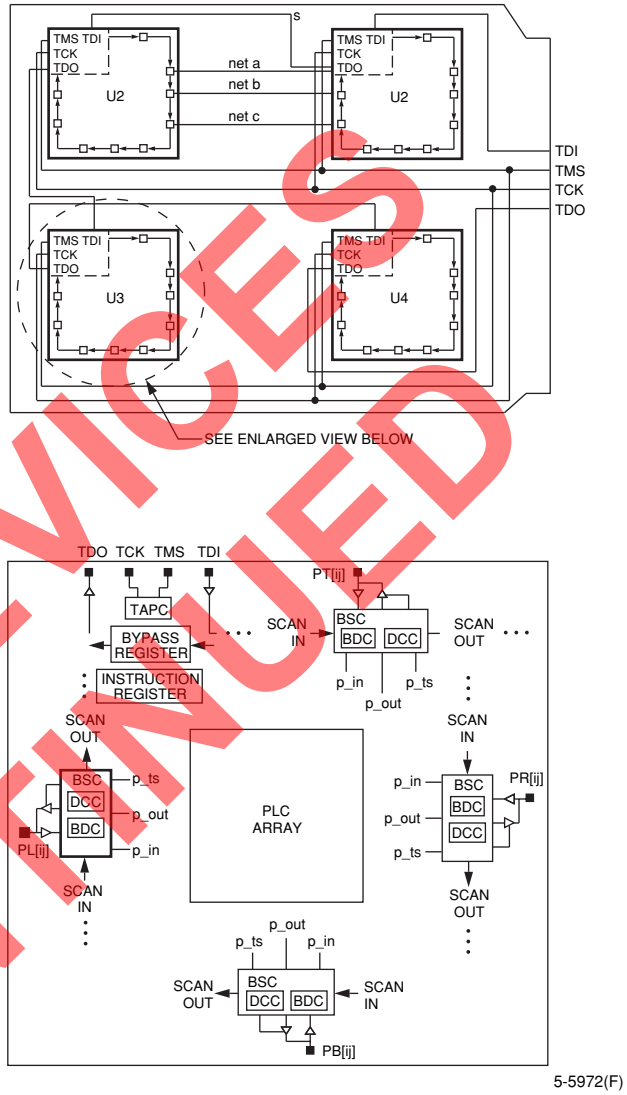
Boundary Scan

The increasing complexity of integrated circuits (ICs) and IC packages has increased the difficulty of testing printed-circuit boards (PCBs). To address this testing problem, the *IEEE* standard 1149.1/D1 (*IEEE* Standard Test Access Port and Boundary-Scan Architecture) is implemented in the *ORCA* series of FPGAs. It allows users to efficiently test the interconnection between integrated circuits on a PCB as well as test the integrated circuit itself. The *IEEE* 1149.1/D1 standard is a well-defined protocol that ensures interoperability among boundary-scan (BSCAN) equipped devices from different vendors.

The *IEEE* 1149.1/D1 standard defines a test access port (TAP) that consists of a four-pin interface with an optional reset pin for boundary-scan testing of integrated circuits in a system. The *ORCA* Series FPGA provides four interface pins: test data in (TDI), test mode select (TMS), test clock (TCK), and test data out (TDO). The $\overline{\text{PRGM}}$ pin used to reconfigure the device also resets the boundary-scan logic.

The user test host serially loads test commands and test data into the FPGA through these pins to drive outputs and examine inputs. In the configuration shown in Figure 36, where boundary scan is used to test ICs, test data is transmitted serially into TDI of the first BSCAN device (U1), through TDO/TDI connections between BSCAN devices (U2 and U3), and out TDO of the last BSCAN device (U4). In this configuration, the TMS and TCK signals are routed to all boundary-scan ICs in parallel so that all boundary-scan components operate in the same state. In other configurations, multiple scan paths are used instead of a single ring. When multiple scan paths are used, each ring is independently controlled by its own TMS and TCK signals.

Figure 37 provides a system interface for components used in the boundary-scan testing of PCBs. The three major components shown are the test host, boundary-scan support circuit, and the devices under test (DUTs). The DUTs shown here are *ORCA* Series FPGAs with dedicated boundary-scan circuitry. The test host is normally one of the following: automatic test equipment (ATE), a workstation, a PC, or a microprocessor.



5-5972(F)

Key: BSC = boundary-scan cell, BDC = bidirectional data cell, and DCC = data control cell.

Figure 36. Printed-Circuit Board with Boundary-Scan Circuitry

Special Function Blocks (continued)

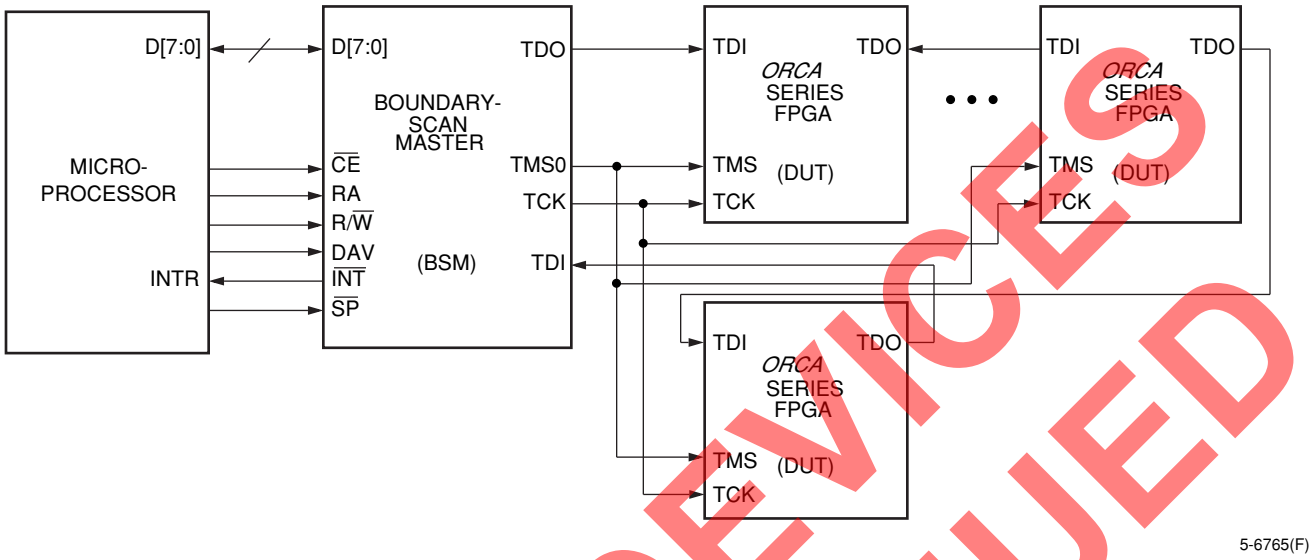


Figure 37. Boundary-Scan Interface

The boundary-scan support circuit shown in Figure 37 is the 497AA Boundary-Scan Master (BSM). The BSM off-loads tasks from the test host to increase test throughput. To interface between the test host and the DUTs, the BSM has a general microprocessor interface and provides parallel-to-serial/serial-to-parallel conversion, as well as three 8K data buffers. The BSM also increases test throughput with a dedicated automatic test-pattern generator and with compression of the test response with a signature analysis register. The PC-based boundary-scan test card/software allows a user to quickly prototype a boundary-scan test setup.

Boundary-Scan Instructions

The *ORCA* Series boundary-scan circuitry is used for three mandatory *IEEE* 1149.1/D1 tests (EXTEST, SAMPLE/PRELOAD, BYPASS), the optional *IEEE* 1149.1/D1 IDCODE instruction, and five *ORCA*-defined instructions. The 3-bit wide instruction register supports the nine instructions listed in Table 13, where the use of PSR1 or USERCODE is selectable by a bit stream option.

Table 13. Boundary-Scan Instructions

Code	Instruction
000	EXTEST
001	PLC Scan Ring 1 (PSR1)/USERCODE
010	RAM Write (RAM_W)
011	IDCODE
100	SAMPLE/PRELOAD
101	PLC Scan Ring 2 (PSR2)
110	RAM Read (RAM_R)
111	BYPASS

Special Function Blocks (continued)

The external test (EXTEST) instruction allows the interconnections between ICs in a system to be tested for opens and stuck-at faults. If an EXTEST instruction is performed for the system shown in Figure 36, the connections between U1 and U2 (shown by nets a, b, and c) can be tested by driving a value onto the given nets from one device and then determining whether the same value is seen at the other device. This is determined by shifting 2 bits of data for each pin (one for the output value and one for the 3-state value) through the BSR until each one aligns to the appropriate pin. Then, based upon the value of the 3-state signal, either the I/O pad is driven to the value given in the BSR, or the BSR is updated with the input value from the I/O pad, which allows it to be shifted out TDO.

The SAMPLE/PRELOAD instruction is useful for system debugging and fault diagnosis by allowing the data at the FPGA's I/Os to be observed during normal operation or written during test operation. The data for all of the I/Os is captured simultaneously into the BSR, allowing them to be shifted-out TDO to the test host. Since each I/O buffer in the PICs is bidirectional, two pieces of data are captured for each I/O pad: the value at the I/O pad and the value of the 3-state control signal. For preload operation, data is written from the BSR to all of the I/Os simultaneously.

There are five *ORCA*-defined instructions. The PLC scan rings 1 and 2 (PSR1, PSR2) allow user-defined internal scan paths using the PLC latches/FFs. The RAM_Write Enable (RAM_W) instruction allows the user to serially configure the FPGA through TDI. The RAM_Read Enable (RAM_R) allows the user to read back RAM contents on TDO after configuration. The IDCODE instruction allows the user to capture a 32-bit identification code that is unique to each device and serially output it at TDO. The IDCODE format is shown in Table 14.

Table 14. Boundary-Scan ID Code

Device	Version (4 bits)	Part* (10 bits)	Family (6 bits)	Manufacturer (11 bits)	LSB (1 bit)
OR3T20	0000	0011000000	110000	00000011101	1
OR3T30	0000	0111000000	110000	00000011101	1
OR3T55	0000	0100100000	110000	00000011101	1
OR3C/T80	0000	0110100000	110000	00000011101	1
OR3T125	0000	0011100000	110000	00000011101	1

* PLC array size of FPGA, reverse bit order.

Note: Table assumes version 0.

Special Function Blocks (continued)

ORCA Boundary-Scan Circuitry

The *ORCA* Series boundary-scan circuitry includes a test access port controller (TAPC), instruction register (IR), boundary-scan register (BSR), and bypass register. It also includes circuitry to support the four predefined instructions.

Figure 38 shows a functional diagram of the boundary-scan circuitry that is implemented in the *ORCA* Series. The input pins' (TMS, TCK, and TDI) locations vary depending on the part, and the output pin is the dedicated TDO/RD_DATA output pad. Test data in (TDI) is the serial input data. Test mode select (TMS) controls the boundary-scan test access port controller (TAPC). Test clock (TCK) is the test clock on the board.

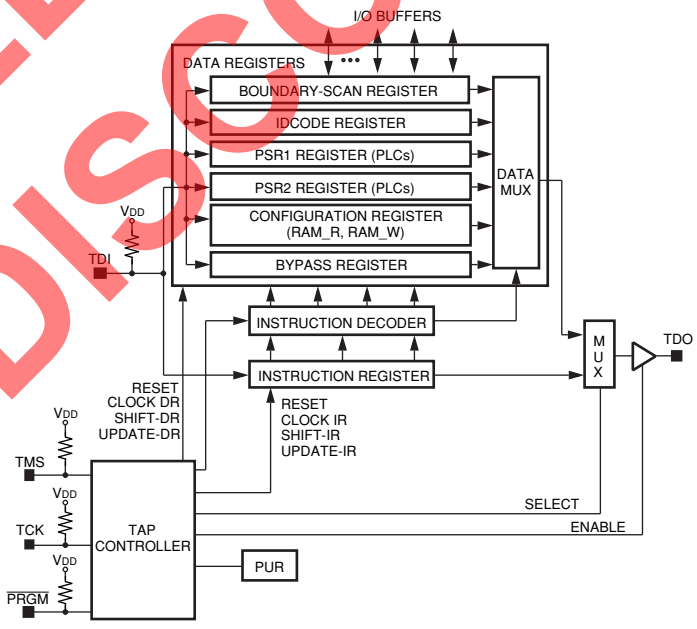
The BSR is a series connection of boundary-scan cells (BSCs) around the periphery of the IC. Each I/O pad on the FPGA, except for CCLK, DONE, and the boundary-scan pins (TCK, TDI, TMS, and TDO), is included in the BSR. The first BSC in the BSR (connected to TDI) is located in the first PIC I/O pad on the left of the top side of the FPGA (PTA PIC). The BSR proceeds clockwise around the top, right, bottom, and left sides of the array. The last BSC in the BSR (connected to TDO) is located on the top of the left side of the array (PL1D).

The bypass instruction uses a single FF, which resynchronizes test data that is not part of the current scan operation. In a bypass instruction, test data received on TDI is shifted out of the bypass register to TDO. Since the BSR (which requires a two FF delay for each pad) is bypassed, test throughput is increased when devices that are not part of a test operation are bypassed.

The boundary-scan logic is enabled before and during configuration. After configuration, a configuration option determines whether or not boundary-scan logic is used.

The 32-bit boundary-scan identification register contains the manufacturer's ID number, unique part number, and version (as described earlier). The identification register is the default source for data on TDO after RESET if the TAP controller selects the shift-data-register (SHIFT-DR) instruction. If boundary scan is not used, TMS, TDI, and TCK become user I/Os, and TDO is 3-stated or used in the readback operation.

An optional USERCODE is available if the boundary-scan PSR1 instruction is not used. The selection between PSR1 and USERCODE is a configuration option and can be performed in ispLEVER. The USERCODE is an 11-bit value that the user can set during device configuration and can be written to and read from the FPGA via the boundary-scan logic. The USERCODE value replaces the manufacturer field of the boundary-scan ID code when the USERCODE instruction is issued, allowing users to have configured devices identified in a user-defined manner. The manufacturer ID field remains available when the IDCODE instruction is issued.



5-5768(F)

Figure 38. *ORCA* Series Boundary-Scan Circuitry Functional Diagram

Special Function Blocks (continued)

ORCA Series TAP Controller (TAPC)

The *ORCA* Series TAP controller (TAPC) is a 1149.1/D1 compatible test access port controller. The 16 JTAG state assignments from the *IEEE* 1149.1/D1 specification are used. The TAPC is controlled by TCK and TMS. The TAPC states are used for loading the IR to allow three basic functions in testing: providing test stimuli (Update-DR), test execution (Run-Test/Idle), and obtaining test responses (Capture-DR). The TAPC allows the test host to shift in and out both instructions and test data/results. The inputs and outputs of the TAPC are provided in the table below. The outputs are primarily the control signals to the instruction register and the data register.

Table 15. TAP Controller Input/Outputs

Symbol	I/O	Function
TMS	I	Test Mode Select
TCK	I	Test Clock
PUR	I	Powerup Reset
$\overline{\text{PRGM}}$	I	BSCAN Reset
TRESET	O	Test Logic Reset
Select	O	Select IR (High); Select-DR (Low)
Enable	O	Test Data Out Enable
Capture-DR	O	Capture/Parallel Load-DR
Capture-IR	O	Capture/Parallel Load-IR
Shift-DR	O	Shift Data Register
Shift-IR	O	Shift Instruction Register
Update-DR	O	Update/Parallel Load-DR
Update-IR	O	Update/Parallel Load-IR

The TAPC generates control signals that allow capture, shift, and update operations on the instruction and data registers. In the capture operation, data is loaded into the register. In the shift operation, the captured data is shifted out while new data is shifted in. In the update operation, either the instruction register is loaded for instruction decode, or the boundary-scan register is updated for control of outputs.

The test host generates a test by providing input into the *ORCA* Series TMS input synchronous with TCK. This sequences the TAPC through states in order to perform the desired function on the instruction register or a data register. Figure 39 provides a diagram of the state transitions for the TAPC. The next state is determined by the TMS input value.

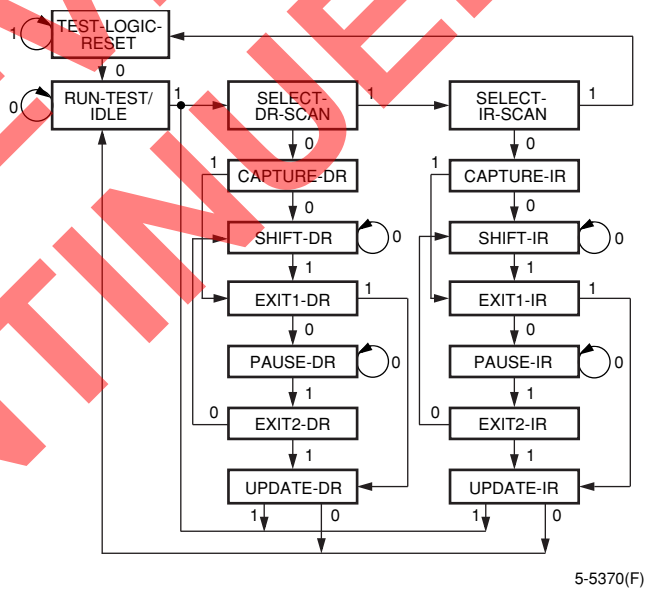


Figure 39. TAP Controller State Transition Diagram

Special Function Blocks (continued)

Boundary-Scan Cells

Figure 40 is a diagram of the boundary-scan cell (BSC) in the *ORCA* series PICs. There are four BSCs in each PIC: one for each pad, except as noted above. The BSCs are connected serially to form the BSR. The BSC controls the functionality of the in, out, and 3-state signals for each pad.

The BSC allows the I/O to function in either the normal or test mode. Normal mode is defined as when an output buffer receives input from the PLC array and provides output at the pad or when an input buffer provides input from the pad to the PLC array. In the test mode, the BSC executes a boundary-scan operation, such as shifting in scan data from an upstream BSC in the BSR, providing test stimuli to the pad, capturing test data at the pad, etc.

The primary functions of the BSC are shifting scan data serially in the BSR and observing input (p_in), output (p_out), and 3-state (p_ts) signals at the pads. The BSC consists of two circuits: the bidirectional data cell is used to access the input and output data, and the direction control cell is used to access the 3-state value. Both cells consist of a flip-flop used to shift scan data which feeds a flip-flop to control the I/O buffer. The bidirectional data cell is connected serially to the direction control cell to form a boundary-scan shift register.

The TAPC signals (capture, update, shiftn, treset, and TCK) and the MODE signal control the operation of the BSC. The bidirectional data cell is also controlled by the high out/low in (HOLI) signal generated by the direction control cell. When HOLI is low, the bidirectional data cell receives input buffer data into the BSC. When HOLI is high, the BSC is loaded with functional data from the PLC.

The MODE signal is generated from the decode of the instruction register. When the MODE signal is high (EXTEST), the scan data is propagated to the output buffer. When the MODE signal is low (BYPASS or SAMPLE), functional data from the FPGA's internal logic is propagated to the output buffer.

The boundary-scan description language (BSDL) is provided for each device in the *ORCA* Series of FPGAs on the ispLEVER CD. The BSDL is generated from a device profile, pinout, and other boundary-scan information.

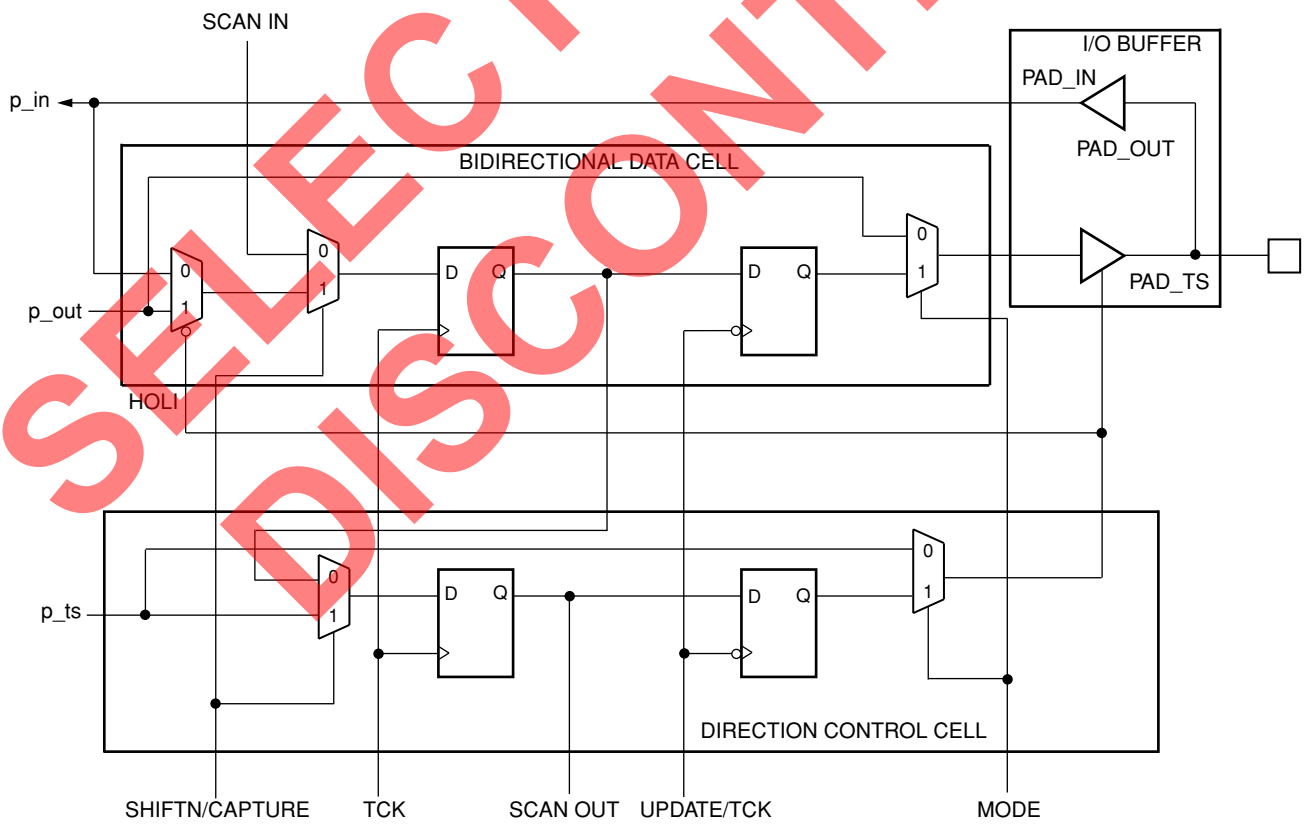


Figure 40. Boundary-Scan Cell

Special Function Blocks (continued)

Boundary-Scan Timing

To ensure race-free operation, data changes on specific clock edges. The TMS and TDI inputs are clocked in on the rising edge of TCK, while changes on TDO occur on the falling edge of TCK. In the execution of an EXTEST instruction, parallel data is output from the BSR to the FPGA pads on the falling edge of TCK. The maximum frequency allowed for TCK is 10 MHz.

Figure 41 shows timing waveforms for an instruction scan operation. The diagram shows the use of TMS to sequence the TAPC through states. The test host (or BSM) changes data on the falling edge of TCK, and it is clocked into the DUT on the rising edge.

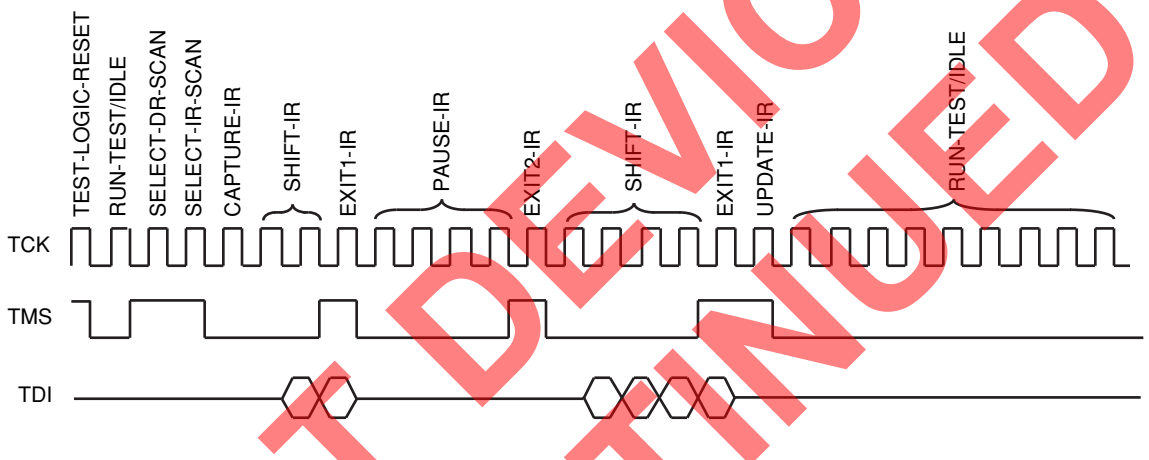


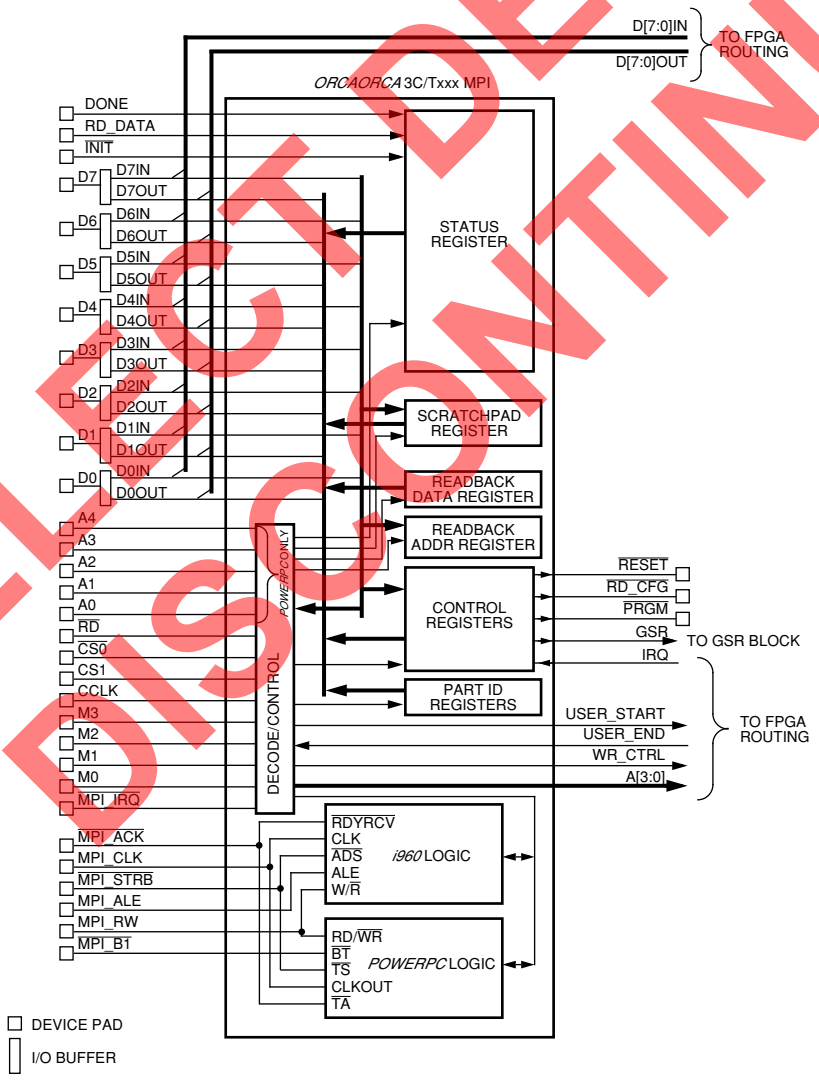
Figure 41. Instruction Register Scan Timing Diagram

Microprocessor Interface (MPI)

The Series 3 FPGAs have a dedicated synchronous microprocessor interface function block (see Figure 42). The MPI is programmable to operate with *PowerPC* MPC800 series microprocessors and *Intel* i960** J core processors; see Table 16 and Table 17, respectively, for compatible processors. The MPI implements an 8-bit interface to the host processor (*PowerPC* or *i960*) that can be used for configuration and readback of the FPGA as well as for user-defined data processing and general monitoring of FPGA function. In addition to dedicated-function registers, the microprocessor interface allows for the control of up to 16 user registers (RAM or flip-flops) in the FPGA logic. A synchronous/asynchronous handshake procedure is used to control transactions with user logic in the FPGA array. There is also capability for the FPGA logic to interrupt the host processor either by a hard interrupt or by having the host processor poll the microprocessor interface.

The control portion of the microprocessor interface is available following powerup of the FPGA if the mode pins specify MPI mode, even if the FPGA is not yet configured. The mode pin (M[2:0]) settings can be found in the FPGA Configuration Modes section of this data sheet, and the setup and use of the MPI for configuration is discussed in the MPI Setup and Control subsection. For postconfiguration use, the MPI must be included in the configuration bit stream by using an MPI library element in your design from the *ORCA* macro library, or by setting the MP_USER bit of the MPI configuration control register prior to the start of configuration (MPI registers are discussed later).

* *Intel* and *i960* are registered trademarks of Intel Corporation.



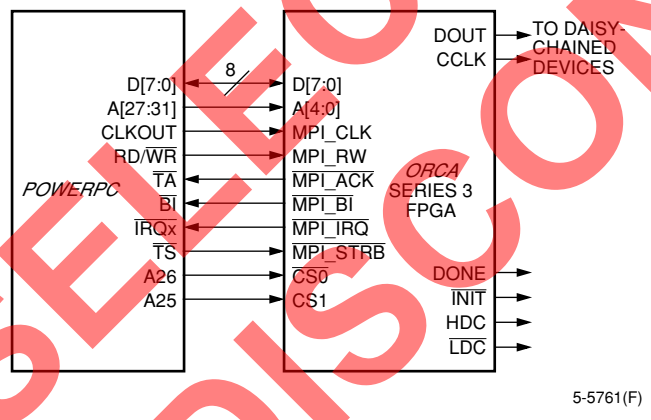
5-5806(F)

Figure 42. MPI Block Diagram

Microprocessor Interface (MPI) (continued)

PowerPC System

In Figure 43, the *ORCA* FPGA is a memory-mapped peripheral to the *PowerPC* processor. The *PowerPC* interface uses separate address and data buses and has several control lines. The *ORCA* chip select lines, $\overline{CS0}$ and CS1, are each connected to an address line coming from the *PowerPC*. In this manner, the FPGA is capable of a transaction with the *PowerPC* whenever the address line connected to $\overline{CS0}$ is low, the address line for CS1 is high, and there is a valid address on *PowerPC* address lines A[27:31]. Other forms of selection are possible by using the FPGA chip selects in a different way. For example, *PowerPC* address bits A[0:26] could be decoded to select $\overline{CS0}$ and CS1, or if the FPGA is the only peripheral to the *PowerPC*, CS0 and CS1 could be tied low and high, respectively, to cause them to always be selected. If the MPI is not used for FPGA configuration, decoding logic can be implemented internal or external to the FPGA. If logic internal to the FPGA is used, the chip selects must be routed out on an output pin and then connected externally to $\overline{CS0}$ and/or CS1. If the MPI is to be used for configuration, any decode logic used must be implemented external to the FPGA since the FPGA logic has not been configured yet.



Note: FPGA shown as a memory-mapped peripheral using $\overline{CS0}$ and CS1. Other decoding schemes are possible using $\overline{CS0}$ and/or CS1.

Figure 43. *PowerPC*/MPI

The basic flow of a transaction on the *PowerPC*/MPI interface is given below. Pin descriptions are shown in Table 16 and timing is shown in the Timing Characteristics section of this data sheet. For both read and write transactions, the address, chip select, and read/write (read high, write low) signals are set up at the FPGA pins by the *PowerPC*. The *PowerPC* then asserts its transfer start signal ($\overline{TS}$) low. Data is available to the MPI during a write at the rising clock edge after the clock cycle during which $\overline{TS}$ is low. The transfer is acknowledged to the *PowerPC* by the low assertion of the $\overline{TA}$ signal. The MPI *PowerPC* interface does not support burst transfers, so the burst inhibit signal, $\overline{BI}$, is also asserted low during the transfer acknowledge. The same process applies to a read from the MPI except that the read data is expected at the FPGA data pins by the *PowerPC* at the rising edge of the clock when $\overline{TA}$ is low. The MPI only drives $\overline{TA}$ low for one clock cycle.

Interrupt requests can be sent to the *PowerPC* asynchronously to the read/write process. Interrupt requests are sourced by the user-logic in the FPGA. The MPI will assert the request to the *PowerPC* as a direct interrupt signal and/or a pollable bit in the MPI status register (discussed in the MPI Setup and Control section). The MPI will continue to assert the interrupt request until the user-logic deasserts its interrupt request signal.

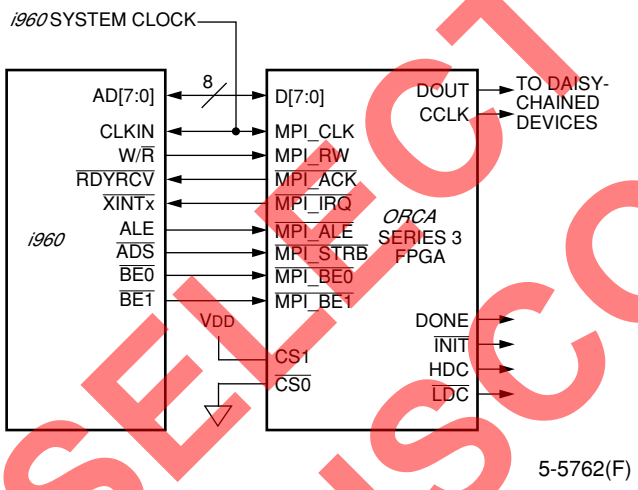
Table 16. *PowerPC*/MPI Configuration

PowerPC Signal	*ORCA* Pin Name	MPI I/O	Function
D[0:7]	D[7:0]	I/O	8-bit data bus
A[27:31]	A[4:0]	I	5-bit MPI address bus
$\overline{TS}$	$\overline{RD/MPI\_STRB}$	I	Transfer start signal
—	$\overline{CS0}$	I	Active-low MPI select
—	CS1	I	Active-high MPI select
CLKOUT	A7/MPI_CLK	I	*PowerPC* interface clock
RD/$\overline{WR}$	A8/MPI_RW	I	Read (high)/write (low) signal
$\overline{TA}$	A9/$\overline{MPI\_ACK}$	O	Active-low transfer acknowledge signal
$\overline{BI}$	A10/$\overline{MPI\_BI}$	O	Active-low burst transfer inhibit signal
Any of $\overline{IRQ}$[7:0]	A11/$\overline{MPI\_IRQ}$	O	Active-low interrupt request signal

Microprocessor Interface (MPI) (continued)

i960 System

Figure 44 shows a schematic for connecting the *ORCA* MPI to supported *i960* processors. In the figure, the FPGA is shown as the only peripheral, with the FPGA chip select lines, $\overline{CS0}$ and CS1, tied low and high, respectively. The *i960* address and data are multiplexed onto the same bus. This precludes memory mapping of the FPGA in the *i960* memory space of a multiperipheral system without some form of address latching to capture and hold the address signals to drive the $\overline{CS0}$ and/or CS1 signals. Multiple address signals could also be decoded and latched to drive the $\overline{CS0}$ and/or CS1 signals. If the MPI is not used for FPGA configuration, decoding/latching logic can be implemented internal or external to the FPGA. If logic internal to the FPGA is used, the chip selects must be routed out an output pin and then connected externally to $\overline{CS0}$ and/or CS1. If the MPI is to be used for configuration, any decode/latch logic used must be implemented external to the FPGA since the FPGA logic has not been configured yet.



Note: FPGA shown as only system peripheral with fixed-chip select signals. For multiperipheral systems, address decoding and/or latching can be used to implement chip selects.

Figure 44. *i960*/MPI

The basic flow of a transaction on the *i960*/MPI interface is given below. Pin descriptions are shown in Table 17, and timing is shown in the *ORCA* Timing Characteristics section of this data sheet. For both read and write transactions, the address latch enable (ALE) is set up by the *i960* at the FPGA to the falling edge of the clock. The address, byte enables, chip selects, and read/write (read low, write high) signals are normally set up at the FPGA pins by the *i960* at the next rising edge of the clock. At this same rising clock edge, the *i960* asserts its address/data strobe ($\overline{ADS}$) low. Data is available to the MPI during a write at the rising clock edge of the following clock cycle. The transfer is acknowledged to the *i960* by the low assertion of the ready/recover ($\overline{RDYRCV}$) signal. The same process applies to a read from the MPI except that the read data is expected at the FPGA data pins by the *i960* at the rising edge of the clock when $\overline{RDYRCV}$ is low. The MPI only drives $\overline{RDYRCV}$ low for one clock cycle.

Interrupts can be sent to the *i960* asynchronously to the read/write process. Interrupt requests are sourced by the user-logic in the FPGA. The MPI will assert the request to the *i960* as a direct interrupt signal and/or a pollable bit in the MPI status register (discussed in the MPI Setup and Control section). The MPI will continue to assert the interrupt request until the user-logic deasserts its interrupt request signal.

Table 17. *i960*/MPI Configuration

i960 Signal	*ORCA* Pin Name	MPI I/O	Function
AD[7:0]	D[7:0]	I/O	Multiplexed 5-bit address/ 8-bit data bus. The address appears on D[4:0].
ALE	RDY/RCLK/ $\overline{MPI\_ALE}$	I	Address latch enable used to capture address from AD[4:0] on falling edge of clock.
$\overline{ADS}$	$\overline{RD}$/ $\overline{MPI\_STRB}$	I	Address/data strobe to indicate start of transaction.
—	$\overline{CS0}$	I	Active-low MPI select.
—	CS1	I	Active-high MPI select.
System Clock	A7/ MPI_CLK	I	*i960* system clock. This clock is sourced by the system and not the *i960*.
W/$\overline{R}$	A8/MPI_RW	I	Write (high)/read (low) signal.
$\overline{RDYRCV}$	A9/ $\overline{MPI\_ACK}$	O	Active-low ready/recover signal indicating acknowledgment of the transaction.
Any of $\overline{XINT}$[7:0]	A11/ $\overline{MPI\_IRQ}$	O	Active-low interrupt request signal.
$\overline{BE0}$	A0/ $\overline{MPI\_BE0}$	I	Byte-enable 0 used as address bit 0 in *i960* 8-bit mode.
$\overline{BE1}$	A1/ $\overline{MPI\_BE1}$	I	Byte-enable 1 used as address bit 1 in *i960* 8-bit mode.

Microprocessor Interface (MPI) (continued)

MPI Interface to FPGA

The MPI interfaces to the user-programmable FPGA logic using a 4-bit address, read/write control signal, interrupt request signal, and user start and user end handshake signals. Timing numbers are provided so that the user-logic data transfers can be performed synchronously with the host processor (*PowerPC* or *i960*) interface clock or asynchronously. Table 18 shows the internal interface signals between the MPI and the FPGA user-programmable logic. All of the signals are connected to the MPI in the upper-left corner of the device except for the D[7:0] and CLK signals that come directly from the I/O pin.

The 4-bit addressing from the MPI to the PLCs allows for up to 16 locations to be addressed by the host processor. The user address space of the MPI does not address any hard register. Rather, the user is free to construct registers from FFs, latches, or RAM that can be selected by the addressing. Alternately, the decoded address signals may be used as control signals for other functions such as state machines or timers.

The transaction sequence between the MPI and the user-logic is as follows. When the host processor initiates a transaction as discussed in the preceding sections, the MPI outputs the 4-bit user address (UA[3:0]) and the read/write control signal ($\overline{\text{URDWR}}$, which is read-high, write-low regardless of host processor), and then asserts the user start signal, USTART. During a write from the host processor, the user logic can accept data written by the host processor from the D[7:0] pins once the USTART signal is asserted. The user logic ends a transaction by asserting an active-high user end (UEND) signal to the MPI.

The MPI will insert wait-states in the host processor bus cycles, holding the host processor until the user-logic completes its task and returns a UEND signal, upon which the MPI generates an acknowledge signal. If the host processor is reading from the FPGA, the user logic must have the read data available on the D[7:0] pins of the FPGA when the UEND signal is asserted. If the user logic is fast or if the MPI user address is being decoded for use as a control signal, the MPI transaction time can be minimized by routing the USTART signal directly to the UEND input of the MPI. The timing section of this data sheet contains a parameter table with delay, setup, and hold timing requirements to operate the user-logic either synchronously or asynchronously with the MPI host interface clock.

The user-logic may also assert an active-low interrupt request ($\overline{\text{UIRQ}}$) to the MPI, which, in turn, asserts an interrupt to the host processor. Assertion of an interrupt request is asynchronous to the host processor clock and any read or write transaction occurring in the MPI. The user-logic is responsible for providing any required interrupt vectors for the host processor, and the user-logic must deassert the interrupt request once serviced. If the interrupt request is not deasserted in the user logic, it will continue to be asserted to the host processor via the $\overline{\text{MPI\_IRQ}}$ pin.

Table 18. MPI Internal Interface Signals

Signal	MPI I/O	Function
UA[3:0]	O	**User Logic Address**. Addresses up to 16 unique user registers or use as control signals.
URDWRN	O	**User Logic Read/Write Control Signal**. High indicates a read from user logic by the host processor, low indicates a write to user-logic by the host processor.
USTART	O	**Active-High User Start Signal**. Indicates the start of an MPI transaction between the host processor and the user logic.
UEND	I	**Active-High User End Signal**. Indicates that the user-logic is finished with the current MPI transaction.
$\overline{\text{UIRQ}}$	I	**Active-Low Interrupt.** Sends request from the user-logic to the host processor.
D[7:0]	FPGA I/O	**User Data.** Eight data bits come directly from the FPGA pins—not through the MPI.
MPI_CLK	FPGA I	**MPI Clock.** The MPI clock is sourced by the host processor and comes directly from the FPGA pin—not through the MPI.

Microprocessor Interface (MPI) (continued)

MPI Setup and Control

The MPI has a series of addressable registers that provide MPI control and status, configuration and readback data transfer, FPGA device identification, and a dedicated user scratchpad register. All registers are 8 bits wide. The address map for these registers and the user-logic address space are shown in Table 19, followed by descriptions of the register and bit functions. Note that for all registers, the most significant bit is bit 7, and the least significant bit is bit 0.

Table 19. MPI Setup and Control Registers

Address (Hex)	Register
00	Control Register 1.
01	Control Register 2.
02	Scratchpad Register.
03	Status Register.
04	Configuration/Readback Data Register.
05	Readback Address Register 1 (bits [7:0]).
06	Readback Address Register 2 (bits [15:8]).
07	Device ID Register 1 (bits [7:0]).
08	Device ID Register 2 (bits [15:8]).
09	Device ID Register 3 (bits [23:16]).
0A	Device ID Register 4 (bits [31:24]).
0B—0F	Reserved.
10—1F	User-definable Address Space.

Control Register 1

The MPI control register 1 is a read/write register. The host processor writes a control byte to configure the MPI. It is readable by the host processor to verify the status of control bits previously written.

Table 20. MPI Setup and Control Registers Descriptions

Bit #	Description
Bit 0	**GSR Input.** Setting this bit to a 1 invokes a global set/reset on the FPGA. The host processor must return this bit to a 0 to remove the GSR signal. GSR does not affect the registers at MPI addresses 0 through F hexadecimal or any configuration registers. Default state = 0.
Bit 1	**Reserved.**
Bit 2	**Reserved.**
Bit 3	**Reserved.**
Bit 4	**Reserved.**
Bit 5	**$\overline{\text{RD\_CFG}}$ Input.** Changing this bit to a 0 after configuration will initiate readback. The host processor must return this bit to a 1 to remove the $\overline{\text{RD\_CFG}}$ signal. Since this bit works exactly like the $\overline{\text{RD\_CFG}}$ input pin, please see the FPGA pin descriptions for more information on this signal. Default state = 1.
Bit 6	**Reserved.**
Bit 7	**$\overline{\text{PRGM}}$ Input.** Setting this bit to a 0 causes the FPGA to begin configuration and resets the boundary-scan circuitry. The host processor must return this bit to a 1 to remove the $\overline{\text{PRGM}}$ signal. Since this bit works exactly like the $\overline{\text{PRGM}}$ input pin (except that it does not reset the MPI), please see the FPGA pin descriptions for more information on this signal. Default state = 1.

Microprocessor Interface (MPI) (continued)

Scratchpad Register

The MPI scratchpad register is an 8-bit read/write register with no defined operation. It may be used for any user-defined function.

Control Register 2

The MPI control register 2 is a read/write register. The host processor writes a control byte to configure the MPI. It is readable by the host processor to verify the status of control bits it had previously written.

Table 21. MPI Control Register 2

Bit #	Bit Name	Description
Bit 0	EN_IRQ_CFG	**Enable $\overline{\text{IRQ}}$ for Configuration Data Request in Daisy-Chain Configuration Mode**. Setting this bit to a 1 prior to configuration enables the $\overline{\text{IRQ}}$ signal to go active when new data is requested for configuration writes or is available for configuration reads to/from the configuration data register. A 0 clears the $\overline{\text{IRQ}}$ enable. This bit is only valid for daisy-chain configuration. Default = 0.
Bit 1	EN_IRQ_ERR	**Enable $\overline{\text{IRQ}}$ for Bit Stream Error**. Setting this bit to a 1 prior to configuration enables the $\overline{\text{IRQ}}$ signal to go active on the occurrence of a bit stream error during configuration. A 0 clears the $\overline{\text{IRQ}}$ enable. This bit only has effect while in configuration mode. Default = 0.
Bit 2	EN_IRQ_USR	**Enable $\overline{\text{IRQ}}$ from the User FPGA Space**. Setting this bit to a 1 allows user-defined circuitry in the FPGA to generate an interrupt to the host processor by sourcing a logic low on the $\overline{\text{UIRQ}}$ signal in the user logic. Default = 0.
Bit 3	MP_DAISY	**MPI Daisy-Chain Output Enable**. Setting this bit to a 1 enables daisy-chain output of the configuration data. See the Configuration section of this data sheet for daisy-chain configuration details. Default = 0.
Bit 4	MP_HOLD_BUS	**Enable Bus Holding During Daisy-Chain Configuration Mode**. Setting this bit to a 1 will cause the MPI to wait until the FPGA configuration logic has serialized a byte of configuration data before acknowledging the transaction. The data is only serialized if the MP_DAISY (bit 3 above) control bit is set to 1. If MP_HOLD_BUS is set to 0, the MPI will immediately acknowledge a configuration data byte transfer. Immediate acknowledgment allows the host processor to perform other tasks during FPGA configuration by polling the MPI status register (or by interrupt) and only write configuration data when the FPGA is ready. Default = 0.
Bit 5	MP_USER	**MPI User Mode Enable**. Setting this bit to a 1 will enable the MPI for user mode operation. MP_USER must be set prior to the FPGA DONE signal going high during configuration. The MPI may also be enabled for user operation via the configuration bit stream. Default = 0.
Bit 6	Reserved	—
Bit 7	Reserved	—

Microprocessor Interface (MPI) (continued)

Status Register

The microprocessor interface status register is a read-only register, providing information to the host processor.

Table 22. Status Register

Bit #	Description
Bit 0	**Reserved**.
Bit 1	**Data Ready**. Set by the MPI, a 1 on this bit during configuration alerts the host processor that the FPGA is ready for another byte of configuration data. During byte-wide readback, the MPI sets this bit to a 1 to tell the host processor that a byte of configuration data is available for reading. This bit is cleared by a host processor access (read or write) to the configuration data register.
Bit 2	**$\overline{\text{IRQ}}$ Pending**. The MPI sets this bit to 1 to indicate to the host processor that the FPGA has a pending interrupt request. This bit may be used for the host processor to poll for interrupts if the $\overline{\text{MPI\_IRQ}}$ pin output of the FPGA has been masked at the host processor. This bit is set to 0 when the status register is read. Interrupt requests from the FPGA user space must be cleared in FPGA user logic in addition to reading this bit.
Bits [4:3]	**Bit Stream Error Flags**. Bits 3 and 4 are set by the MPI to indicate any error during FPGA configuration. See bit 2 of control register 2 for the capability to alert the host processor of an error via the $\overline{\text{IRQ}}$ signal during configuration. In the truth table below, bit 3 is the LSB (bit on right). These bits are cleared to 0 when $\overline{\text{PRGM}}$ goes active: 00 = No error 01 = ID error 10 = Checksum error 11 = Stop-bit/alignment error
Bit 5	**Reserved**.
Bit 6	**$\overline{\text{INIT}}$.** This bit reflects the binary value of the FPGA $\overline{\text{INIT}}$ pin.
Bit 7	**DONE**. This bit reflects the binary value of the FPGA DONE pin.

Configuration Data Register

The MPI configuration data register is a writable register in configuration mode and a readable register in readback mode. For FPGA configuration, this is where the configuration data bytes are sequentially written by the host processor. Similarly, for readback mode, the MPI provides the readback data bytes in this register for the host processor.

Readback Address Register 1

The MPI readback address register 1 is a writable register used to accept the least significant address byte (bits [7:0]) of the configuration data location to be read back.

Readback Address Register 2

The MPI readback address register 2 is a writable register used to accept the most significant address byte (bits [15:8]) of the configuration data location to be read back.

Microprocessor Interface (MPI) (continued)

Device ID Registers

The MPI device ID is broken into four registers holding 1 byte each. The device ID that is available through the MPI is the same as the boundary-scan ID code, except that the device ID in the MPI has a reverse bit order. There is no means to overwrite any of the device ID as can be done with the boundary-scan ID, but the MPI scratchpad register can be used as a personalization register. The format for the entire device ID is shown below followed by family and device values and the partitioning of the device ID into the four device ID registers.

Table 23. Device ID Code

Version	Part*	Family	Manufacturer	MSB
4 bits	10 bits	6 bits	11 bits	1 bit
Example: (First version of OR3C80)			0000 0110100000 110000 00000011101 1	

* PLC array size of FPGA.

Table 24 shows the family and device values for all parts covered by this data sheet.

Table 24. Series 3 Family and Device ID Values

Part Name	Family ID (Hex)	Device ID (Hex)
OR3T20	03	0C
OR3T30	03	0E
OR3T55	03	12
OR3C/T80	03	16
OR3T125	03	1C

Table 25 describes the device IDs for all parts covered by this data sheet as they are partitioned into the four registers found in the MPI.

Table 25. *ORCA* Series 3 Device ID Descriptions

Device ID Register 1	
Bit 0	Logic 1. This bit is always a one.
Bits [7:1]	0011101, the 7 least significant bits of the manufacturer ID.
Device ID Register 2	
Bits [3:0]	0000, the 4 most significant bits of the manufacturer ID.
Bits [7:4]	The 4 least significant bits of the 10-bit part number.
Device ID Register 3	
Bits [5:0]	The 6 most significant bits of the 10-bit part number.
Bits [7:6]	The 2 least significant bits of the device family code.
Device ID Register 4	
Bits [3:0]	The 4 most significant bits of the device family code.
Bits [7:4]	The 4-bit device version code.

Programmable Clock Manager (PCM)

The *ORCA* programmable clock manager (PCM) is a special function block that is used to modify or condition clock signals for optimum system performance. Some of the functions that can be performed with the PCM are clock skew reduction (both internal and board level), duty-cycle adjustment, clock delay reduction, clock phase adjustment, and clock frequency multiplication/division. Due to the different capabilities required by customer application, each PCM contains both a PLL (phase-locked loop) and a DLL (delayed-locked loop) mode. By using PLC logic resources in conjunction with the PCM, many other functions, such as frequency synthesis, are possible.

There are two PCMs on each Series 3 device, one in the lower left corner and one in the upper right corner. Each can drive two different, but interrelated clock networks inside the FPGA. Each PCM can take a clock input from the ExpressCLK pad in its corner or from general routing resources. There are also two input sources that provide feedback to the PCM from the PLC array. One of these is a dedicated corner ExpressCLK feedback, and the other is from general routing. Each PCM sources two clock outputs, one to the corner ExpressCLK that feeds the CLKCNTRL blocks on the two sides adjacent to the PCM, and one to the system clock spine network through general routing. Figure 45 shows a high-level block diagram of the PCM.

Functionality of the PCM is programmed during operation through a read/write interface internal to the FPGA array or via the configuration bit stream. The internal FPGA interface comprises write enable and read enable signals, a 3-bit address bus, an 8-bit input (to the PCM) data bus, and an 8-bit output data bus. There is also a PCM output signal, LOCK, that indicates a stable output clock state. These signals are used to program a series of registers to configure the PCM functional core for the desired functionality.

Operation of the PCM is divided into two modes, delay-locked loop (DLL) and phase-locked loop (PLL). Some operations can be performed by either mode and some are specific to a particular mode. These will be described in each individual mode section. In general, DLL mode is preferable to PLL mode for the same function because it is less sensitive to input clock noise.

In the discussions that follow, the duty cycle is the percent of the clock period during which the output clock is high.

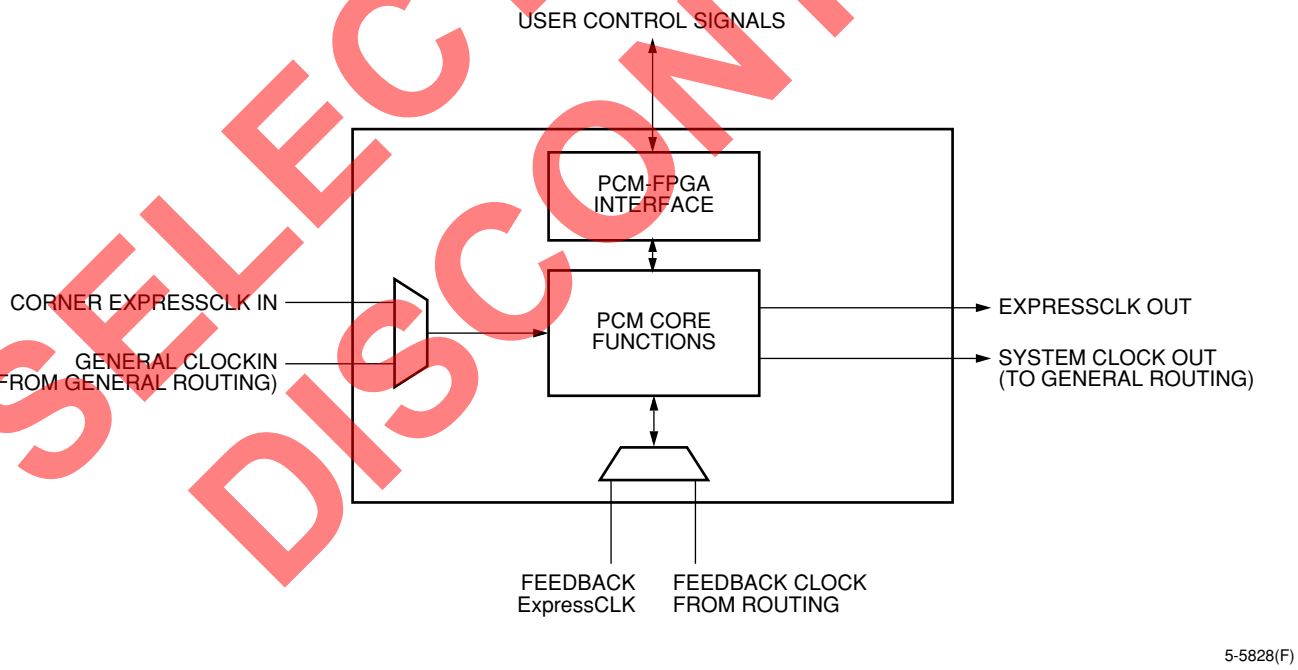


Figure 45. PCM Block Diagram

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)

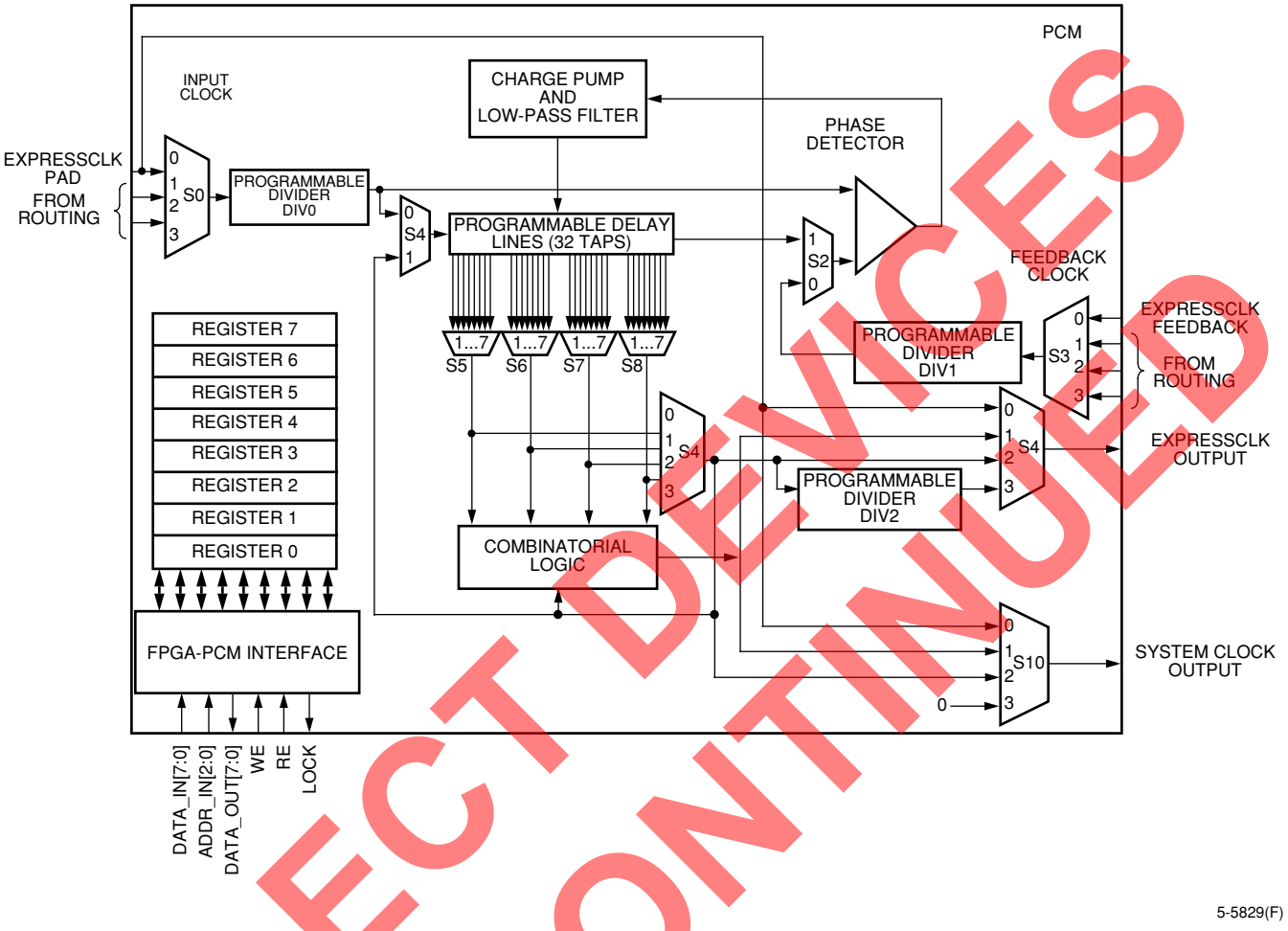


Figure 46. PCM Functional Block Diagram

Programmable Clock Manager (PCM)

(continued)

Delay-Locked Loop (DLL) Mode

DLL mode is used for implementing a delayed clock (phase adjustment), clock doubling, and duty cycle adjustment. All DLL functions stem from a delay line with 32 taps. The delayed input clock is pulled from various taps and processed to implement the desired result. There is no feedback clock in DLL mode, providing a very stable output and a fast lock time for the output clock.

DLL mode is selected by setting bit 0 in PCM register five to a 0. The settings for the various submodes of DLL mode are described in the following paragraphs. Divider DIV0 may be used with any of the DLL modes to divide the input clock by an integer factor of 1 to 8 prior to implementation of the DLL process.

Delayed Clock

A delayed version of the input clock can be constructed in DLL mode. The output clock can be delayed by increments of 1/32 of the input clock period. Express CLK and system CLK outputs in delay modes are selected by setting register six, bits [5:4] to 10 or 11 for ExpressCLK output, and/or bits [7:6] to 10 for system clock output. The delay value is entered in register four. See register four programming details for more information. Delay values are also shown in the second column of Table 27.

Note that when register six, bits [5:4] are set to 11, the ExpressCLK output is divided by an integer factor from 1 to 8 while the system clock cannot be divided. The ExpressCLK divider is provided so that the I/O clocking provided by the ExpressCLK can operate slower than the internal system clock. This allows for very fast internal processing while maintaining slower interface speeds off-chip for improved noise and power performance or to interoperate with slower devices in the system. The divisor of the ExpressCLK frequency is selected in register two. See the register two programming details for more information.

1x Clock Duty-Cycle Adjustment

A duty-cycle adjusted replica of the input clock can be constructed in DLL mode. The duty cycle can be adjusted in 1/32 (3.125%) increments of the input clock period. DLL 1x clock mode is selected by setting bit 4 of register five to a 1, and output clock source selection is selected by setting register six, bits [5:4] to 01 for ExpressCLK output, and/or bits [7:6] to 01 for system clock output. The duty-cycle percentage value is entered in register four. See register four programming details for more information. Duty cycle values are also shown in the third column of Table 27.

Table 27. DLL Mode Delay/1x Duty Cycle Programming Values

Register 4 [7:0] 7 6 5 4 3 2 1 0	Delay (CLK_IN/32)	Duty Cycle (% of CLK_IN)
0 0 X X X 0 0 0	1	3.125
0 0 X X X 0 0 1	2	6.250
0 0 X X X 0 1 0	3	9.375
0 0 X X X 0 1 1	4	12.500
0 0 X X X 1 0 0	5	15.625
0 0 X X X 1 0 1	6	18.750
0 0 X X X 1 1 0	7	21.875
0 0 X X X 1 1 1	8	25.000
0 1 X X X 0 0 0	9	28.125
0 1 X X X 0 0 1	10	31.250
0 1 X X X 0 1 0	11	34.375
0 1 X X X 0 1 1	12	37.500
0 1 X X X 1 0 0	13	40.625
0 1 X X X 1 0 1	14	43.750
0 1 X X X 1 1 0	15	46.875
0 1 1 1 1 X X X	16	50.000
1 0 0 0 0 X X X	17	53.125
1 0 0 0 1 X X X	18	56.250
1 0 0 1 0 X X X	19	59.375
1 0 0 1 1 X X X	20	62.500
1 0 1 0 0 X X X	21	65.625
1 0 1 0 1 X X X	22	68.750
1 0 1 1 0 X X X	23	71.875
1 0 1 1 1 X X X	24	75.000
1 1 0 0 0 X X X	25	78.125
1 1 0 0 1 X X X	26	81.250
1 1 0 1 0 X X X	27	84.375
1 1 0 1 1 X X X	28	87.500
1 1 1 0 0 X X X	29	90.625
1 1 1 0 1 X X X	30	93.750
1 1 1 1 0 X X X	31	96.875

Programmable Clock Manager (PCM)

(continued)

2x Clock Duty-Cycle Adjustment

A doubled-frequency, duty-cycle adjusted version of the input clock can be constructed in DLL mode. The first clock cycle of the 2x clock output occurs when the input clock is high, and the second cycle occurs when the input clock is low. The duty cycle can be adjusted in 1/32 (6.25%) increments of the input clock period. Additionally, each of the two doubled-clock cycles that occurs in a single input clock cycle may be adjusted to have different duty cycles. DLL 2x clock mode is selected by setting bit 4 of register five to a 1, and by setting register six, bits [5:4] to 01 for ExpressCLK output, and/or bits [7:6] to 01 for system clock output. The duty-cycle percentage value is entered in register three. See register three programming details for more information. Duty-cycle values where both cycles of the doubled clock have the same duty cycle are also shown in Table 28.

Table 28. DLL Mode Delay/2x Duty Cycle Programming Values

Register 3 [7:0] 7 6 5 4 3 2 1 0	Duty Cycle (%)
0 0 0 0 0 0 0 0	6.25
0 0 0 0 1 0 0 1	12.50
0 0 0 1 0 0 1 0	18.75
0 0 0 1 1 0 1 1	25.00
0 0 1 0 0 1 0 0	31.25
0 0 1 0 1 1 0 1	37.50
0 0 1 1 0 1 1 0	43.75
0 0 1 1 1 1 1 1	50.00
1 1 0 0 0 0 0 0	56.25
1 1 0 0 1 0 0 1	62.50
1 1 0 1 0 0 1 0	68.75
1 1 0 1 1 0 1 1	75.00
1 1 1 0 0 1 0 0	81.25
1 1 1 0 1 1 0 1	87.50
1 1 1 1 0 1 1 0	93.75

Phase-Locked Loop (PLL) Mode

The PLL mode of the PCM is used for clock multiplication (1/8x to 64x) and clock delay minimization functions. PLL functions make use of the PCM dividers and use feedback signals, often from the FPGA array. The use of feedback is discussed with each PLL submode. PLL mode is selected by setting bit 0 of register five to 1.

Clock Delay Minimization

PLL mode can be used to minimize the effects of the input buffer and input routing delay on the clock signal. PLL mode causes a feedback clock signal to align in phase with the input clock (refer back to the block diagram in Figure 45) so that the delay between them is effectively eliminated.

There is a dedicated feedback path from an adjacent middle CLKCNTRL block to the PCM. Using the corner ExpressCLK pad as the input to the PCM and using this dedicated feedback path, the clock from the ExpressCLK output of the PCM, as viewed at the CLKCNTRL block, will be phase-aligned with the ExpressCLK input to the PCM. These relationships are diagrammed in Figure 47.

A feedback clock can also be input to the PCM from general routing. This allows for compensating for delay between the PCM input and a point in the general routing. The use of this routed-feedback path is not generally recommended. Because compensation is based on the programmable routing, the amount of clock delay compensation can vary between FPGA lots and fabrication processes, and will vary each time that the feedback line is routed using different resources. Contact Lattice for application notes regarding the use of routed-feedback delay compensation.

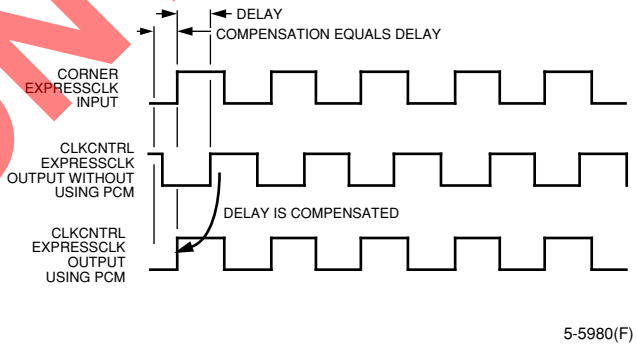


Figure 47. ExpressCLK Delay Minimization Using the PCM

Programmable Clock Manager (PCM)

(continued)

Clock Multiplication

An output clock that is a multiple (not necessarily an integer multiple) of the input clock can be generated in PLL mode. The multiplication ratio is programmed in the division registers DIV0, DIV1, and DIV2. Note that DIV2 applies only to the ExpressCLK output of the PCM and any reference to DIV2 is implicitly 1 for the system clock output of the PCM. The clock multiplication formulas when using ExpressCLK feedback are:

$$F_{ExpressCLK\_OUT} = F_{INPUT\_CLOCK} \bullet \frac{DIV1}{DIV0}$$

$$F_{SYSTEM\_CLOCK\_OUT} = F_{ExpressCLK\_OUT} \bullet DIV2$$

Where the values of DIV0, DIV1, and DIV2 range from 1 to 8.

The ExpressCLK multiplication range of output clock frequencies is, therefore, from 1/8x up to 8x, with the system clock range up to 8x the ExpressCLK frequency or 64x the input clock frequency. If system clock feedback is used, the formulas are:

$$F_{SYSTEM\_CLOCK\_OUT} = F_{INPUT\_CLOCK} \bullet \frac{DIV1}{DIV0}$$

$$F_{ExpressCLK\_OUT} = F_{SYSTEM\_CLOCK}/DIV2$$

The divider values, DIV0, DIV1, and DIV2 are programmed in registers zero, one, and two, respectively.

The multiplied output is selected by setting register six, bits [5:4] to 10 or 11 for ExpressCLK output and/or bits [7:6] to 10 for system clock output. Note that when register six, bits [5:4] are set to 11, the ExpressCLK output is divided by DIV2, while the system clock cannot be divided. The ExpressCLK divider is provided so that the I/O clocking provided by the ExpressCLK can operate slower than the internal system clock. This allows for very fast internal processing while maintaining slower interface speeds off-chip for improved noise and power performance or to interoperate with slower devices in the system.

It is also necessary to configure the internal PCM oscillator for operation in the proper frequency range. Table 29 and Table 30 show the settings required for register four for a given frequency range for Series 3C and 3T devices. In addition, the acquisition time is shown for each frequency range. This is the time that is required for the PCM to acquire LOCK. The PCM oscillator frequency range is chosen based on the desired output frequency at the system clock output. If using the ExpressCLK output, the equivalent system clock frequency can be selected by multiplying the expected ExpressCLK output frequency by the value for DIV2. Choose the nominal frequency from the table that is closest to the desired frequency, and use that value to program register four. Minor adjustments to match the exact input frequency are then performed automatically by the PCM.

Programmable Clock Manager (PCM)

(continued)

Table 29. PCM Oscillator Frequency Range 3Txxx

Register 4 76543210	Min (MHz)	System Clock Output Frequency (MHz) NOM	Max (MHz)	T Acquisition (µs)
00XXX010	17.00	58.50	100.00	36.00
00XXX011	16.10	52.50	89.00	37.00
00XXX100	15.17	49.00	82.80	38.00
00XXX101	14.25	45.00	76.50	39.00
00XXX110	13.33	41.50	70.30	40.00
00XXX111	12.40	38.00	64.00	41.00
01XXX000	12.20	36.75	61.30	43.75
01XXX001	12.10	35.00	58.00	46.50
01XXX010	11.90	33.00	54.30	49.25
01XXX011	11.70	31.30	51.00	52.00
01XXX100	11.10	30.00	49.40	54.75
01XXX101	10.50	29.15	47.80	57.50
01XXX110	10.00	28.10	46.20	60.25
01XXX111	9.40	27.00	44.60	63.00
10000XXX	9.20	26.25	43.30	65.40
10001XXX	9.00	25.65	42.30	67.80
10010XXX	8.80	25.00	41.30	70.10
10011XXX	8.60	24.45	40.30	72.50
10100XXX	8.40	23.70	39.00	74.90
10101XXX	8.10	22.90	37.70	77.30
10110XXX	7.90	22.20	36.50	79.60
10111XXX	7.70	21.50	35.20	82.00
11000XXX	7.60	20.80	34.00	84.30
11001XXX	7.45	20.10	32.80	86.50
11010XXX	7.30	19.45	31.60	88.80
11011XXX	7.20	18.85	30.50	91.00
11100XXX	6.60	18.30	30.00	93.30
11101XXX	6.00	17.70	29.40	95.50
11110XXX	5.50	17.10	28.60	97.80
11111XXX	5.00	16.50	28.00	100.00

Note: Use of settings in the first three rows is not recommended. X means don't care.

Table 30. PCM Oscillator Frequency Range 3Cxx

Register 4 76543210	Min (MHz)	System Clock Output Frequency (MHz) NOM	Max (MHz)	T Acquisition (µs)
00XXX010	10.50	73.00	135.00	36.00
00XXX011	10.00	68.00	126.00	37.00
00XXX100	9.50	63.00	117.00	38.00
00XXX101	9.10	58.50	108.00	39.00
00XXX110	8.60	53.80	99.00	40.00
00XXX111	8.10	49.00	90.00	41.00
01XXX000	7.80	47.70	87.50	43.80
01XXX001	7.60	46.30	85.00	46.50
01XXX010	7.30	45.00	82.50	49.30
01XXX011	7.10	43.60	80.00	52.00
01XXX100	6.80	42.10	77.50	55.00
01XXX101	6.50	40.75	75.00	57.50
01XXX110	6.30	39.40	72.50	60.30
01XXX111	6.00	38.00	70.00	63.00
10000XXX	5.90	37.40	68.80	65.40
10001XXX	5.90	36.70	67.50	67.80
10010XXX	5.80	36.00	66.30	70.10
10011XXX	5.80	35.40	65.00	72.50
10100XXX	5.70	35.00	63.80	74.90
10101XXX	5.60	34.10	62.50	77.30
10110XXX	5.60	33.50	61.30	79.60
10111XXX	5.50	32.80	60.00	82.00
11000XXX	5.40	32.10	58.80	84.30
11001XXX	5.40	31.50	57.50	86.50
11010XXX	5.30	30.70	56.30	88.80
11011XXX	5.30	30.10	55.00	91.00
11100XXX	5.20	29.50	53.80	93.30
11101XXX	5.10	28.80	52.50	95.50
11110XXX	5.10	28.20	51.30	97.80
11111XXX	5.00	27.50	50.00	100.00

Note: Use of settings in the first three rows is not recommended. X means don't care.

Programmable Clock Manager (PCM)

(continued)

PCM/FPGA Internal Interface

Writing and reading the PCM registers is done through a simple asynchronous interface that connects with the FPGA routing resources. Reads from the PCM by the FPGA logic are accomplished by setting up the 3-bit address, A[2:0], and then applying an active-high read enable (RE) pulse. The read data will be available as long as RE is held high. The address may be changed while RE is high, to read other addresses. When RE goes low, the data output bus is 3-stated.

Writes to the PCM by the FPGA logic are performed by applying the write data to the data input bus of the PCM, applying the 3-bit address to write to, and asserting the write enable (WE) signal high. Data will be written by the high-going transition of the WE pulse.

The read enable (RE) and write enable (WE) signals may not be active at the same time. For detailed timing information and specifications, see the Timing Characteristics section of this data sheet.

The LOCK signal output from the PCM to the FPGA routing indicates a stable output clock signal from the PCM. The LOCK signal is high when the PCM output clock parameters fall within the programmed values and the PCM specifications for jitter. Due to phase corrections that occur internal to the PCM, the LOCK signal might occasionally pulse low when the output clock is out of specification for only one or two clock cycles (high jitter due to temperature, voltage fluctuation, etc.) To accommodate these pulses, it is suggested that the user integrate the LOCK signal over a period suitable to their application to achieve the desired usage of the LOCK signal.

The LOCK signal will also pulse high and low during the acquisition time as the output clock stabilizes. True LOCK is only achieved when the LOCK signal is a solid high. Again, it is suggested that the user integrate the LOCK signal over a time period suitable to the subject application.

PCM Operation

Several features are available for the control of the PCM's overall operation. The PCM may be programmably enabled/disabled via bit 0 of register 7. When disabled, the analog power supply of the PCM is turned off, conserving power and eliminating the possibility of inducing noise into the system power buses. Individual bits (register 7, bits [2:1]) are provided to reset the DLL and PLL functions of the PCM. These resets affect only the logic generating the DLL or PLL function; they do not reset the divider values (DIV0, DIV1, DIV2) or registers [7:0]. The global set/reset (GSRN) is also programmably controlled via register 7, bit 7. If register 7, bit 7 is set to 1, GSRN will have no effect on the PCM logic, allowing the clock to operate during a global set/reset. This function allows the FPGA to be reset without affecting a clock that is sent off-chip and used elsewhere in the system. Bit 6 of register 7 affects the functionality of the PCM during configuration. If set to 1, this bit enables the PCM to operate during configuration, after the PCM has been configured. The PCM functionality is programmed via the bit stream. If register 7, bit 6 is 0, the PCM cannot function and its power supply is disabled until after the configuration DONE signal goes high.

When the PCM is powered up via register 7, bit 0, there is a wake-up time associated with its operation. Following the wake-up time, the PCM will begin to fully function, and, following an acquisition time during which the output clock may be unstable, the PCM will be in steady-state operation. There is also a shutdown time associated with powering off the PCM. The output clock will be unstable during this period. Waveforms and timing parameters can be found in the Timing Characteristics section of this data sheet.

Programmable Clock Manager (PCM) (continued)

PCM Detailed Programming

Descriptions of bit fields and individual control bits in the PCM control registers are provided in Table 31. Refer to Figure 46 for more information on the location of the PCM elements that are discussed. In the following discussion, the duty cycle is in the percentage of the clock period where the clock is high.

Table 31. PCM Control Registers

Bit #	Function
Register 0 Divider 0 Programming	
Bits [3:0]	**4-Bit Divider, DIV0, Value**. This value enables the input clock to immediately be divided by a value from 1 to 8. A 0 value (the default) indicates that DIV0 is bypassed (no division). Bypass incurs less delay than dividing by 1. Hexadecimal values greater than 8 for bits [3:0] yield their modulo 8 value. For example, if bits [3:0] are 1001 (9 hex), the result is divide by 1 (remainder 9/8 = 1).
Bits [6:4]	**Reserved**.
Bit 7	**DIV 0 Reset Bit**. DIV0 may not be reset by GSRN depending on the value of register 7, bit 7. This bit may be set to 1 to reset DIV0 to its default value. Bit 0 must be set to 0 (the default) to remove the reset.
Register 1 Divider 1 Programming	
Bits [3:0]	**4-Bit Divider, DIV1, Value**. This value enables the feedback clock to be divided by a value from 1 to 8. A 0 value (the default) indicates that DIV1 is bypassed (no division). Bypass incurs less delay than dividing by 1. Hexadecimal values greater than 8 for bits [3:0] yield their modulo 8 value. For example, if bits [3:0] are 1001 (9 hex), the result is divide by 1 (remainder 9/8 = 1).
Bits [6:4]	**Reserved**.
Bit 7	**DIV1 Reset Bit**. DIV1 may not be reset by GSRN, depending on the value of register 7, bit 7. This bit may be set to 1 to reset DIV1 to its default value. Bit 0 must be set to 0 (the default) to remove the reset.
Register 2 Divider 2 Programming	
Bits [3:0]	**4-Bit Divider, DIV2, Value**. This value enables the tapped delay line output clock driven onto ExpressCLK to be divided by a value from 1 to 8. A 0 value (the default) indicates that DIV2 is bypassed (no division). Bypass incurs less delay than dividing by 1. Hexadecimal values greater than 8 for bits [3:0] yield their modulo 8 value. For example, if bits [3:0] are 1001 (9 hex), the result is divide by 1 (remainder 9/8 = 1).
Bits [6:4]	**Reserved**.
Bit 7	**DIV2 Reset Bit**. DIV2 may not be reset by GSRN, depending on the value of register 7, bit 7. This bit may be set to 1 to reset DIV2 to its default value. Bit 7 must be set to 0 (the default) to remove the reset.
Register 3 DLL 2x Duty-Cycle Programming	
Bits [2:0]	Duty-cycle selection for the doubled clock period associated with the input clock high. The duty cycle is (value of bit 6) * 50% + ((value of bits [2:0]) + 1) * 6.25%. See the description for bit 6.
Bits [5:3]	Duty-cycle selection for the doubled clock period associated with the input clock low. The duty cycle is (value of bit 7) * 50% + ((value of bits [2:0]) + 1) * 6.25%. See the description for bit 7.
Bit 6	Master duty-cycle control for the first clock period of the doubled clock: 0 = less than or equal to 50%, 1 = greater than 50%.
Bit 7	Master duty-cycle control for the second clock period of the doubled clock: 0 = less than or equal to 50%, 1 = greater than 50%. Example: Both clock periods having a 62.5% duty cycle, bits [7:0] are 11 001 001.

Programmable Clock Manager (PCM) (continued)

Table 31. PCM Control Registers (continued)

Bit #	Function
Register 4 DLL 1x Duty-Cycle Programming	
Bits [2:0]	**Duty-Cycle/Delay Selection for Duty Cycle/Delays Less Than or Equal to 50%.** The duty-cycle/delay is (value of bits [7:6]) * 25% + ((value of bits [2:0]) + 1) * 3.125%. See the description for bits [7:6].
Bits [5:3]	**Duty-Cycle/Delay Selection for Duty Cycle/Delays Greater Than 50%.** The duty-cycle/delay is (value of bits [7:6]) * 25% + ((value of bits [5:3]) + 1) * 3.125%. See the description for bits [7:6].
Bits [7:6]	**Master Duty Cycle Control**: 00: duty cycle 3.125% to 25% 01: duty cycle 28.125% to 50% 10: duty cycle 53.125% to 75% 11: duty cycle 78.125% to 96.875% Example: A 40.625% duty cycle, bits [7:0] are 01 XXX 100, where X is a don't care because the duty cycle is not greater than 50%. Example: The PCM output clock should be delayed 96.875% (31/32) of the input clock period. Bits [7:0] are 11110XXX, which is 78.125% from bits [7:6] and 18.75% from bits [5:3]. Bits [2:0] are don't care (X) because the delay is greater than 50%.
Register 5 Mode Programming	
Bit 0	**DLL/PLL Mode Selection Bit**. 0 = DLL, 1 = PLL. Default is DLL mode.
Bit 1	**Reserved**.
Bit 2	**PLL Phase Detector Feedback Input Selection Bit**. 0 = feedback signal from routing/ExpressCLK, 1 = feedback from programmable delay line output. Default is 0. Has no effect in DLL mode.
Bit 3	**Reserved**.
Bit 4	**1x/2x Clock Selection Bit for DLL Mode**. 0 = 1x clock output, 1 = 2x clock output. Default is 1x clock output. Has no effect in PLL mode.
Bits [7:5]	**Reserved**.

Programmable Clock Manager (PCM) (continued)

Table 31. PCM Control Registers (continued)

Bit #	Function
Bits [5:4]	**ExpressCLK Output Source Selector**. Default is 00. 00: PCM input clock, bypass path through PCM 01: DLL output 10: tapped delay line output 11: divided (DIV2) delay line output
Bits [7:6]	**System Clock Output Source Selector**. Default is 00. 00: PCM input clock, bypass path through PCM 01: DLL output 10: tapped delay line output 11: reserved
Register 7 PCM Control Programming	
Bit 0	**PCM Analog Power Supply Switch**. 1 = power supply on, 0 = power supply off.
Bit 1	**PCM Reset**. A value of 1 resets all PCM logic for PLL and DLL modes.
Bit 2	**DLL Reset**. A value of 1 resets the clock generation logic for DLL mode. No dividers or user registers are affected.
Bits [5:3]	**Reserved**.
Bit 6	**PCM Configuration Operation Enable Bit**. 0 = normal configuration operation. During configuration (DONE = 0), the PCM analog power supply will be off, the PCM output data bus is 3-stated, and the LOCK signal is asserted to logic 0. The PCM will power up when DONE = 1. 1 = PCM operation during configuration. The PCM may be powered up (see bit 0) and begin operation, or continue operation. The setup of the PCM can be performed via the configuration bit stream.
Bit 7	**PCM GSRN Enable Bit**. 0 = normal GSRN operation. 1 = GSRN has no effect on PCM logic, so clock processing will not be interrupted by a chip reset. Default is 0.

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.

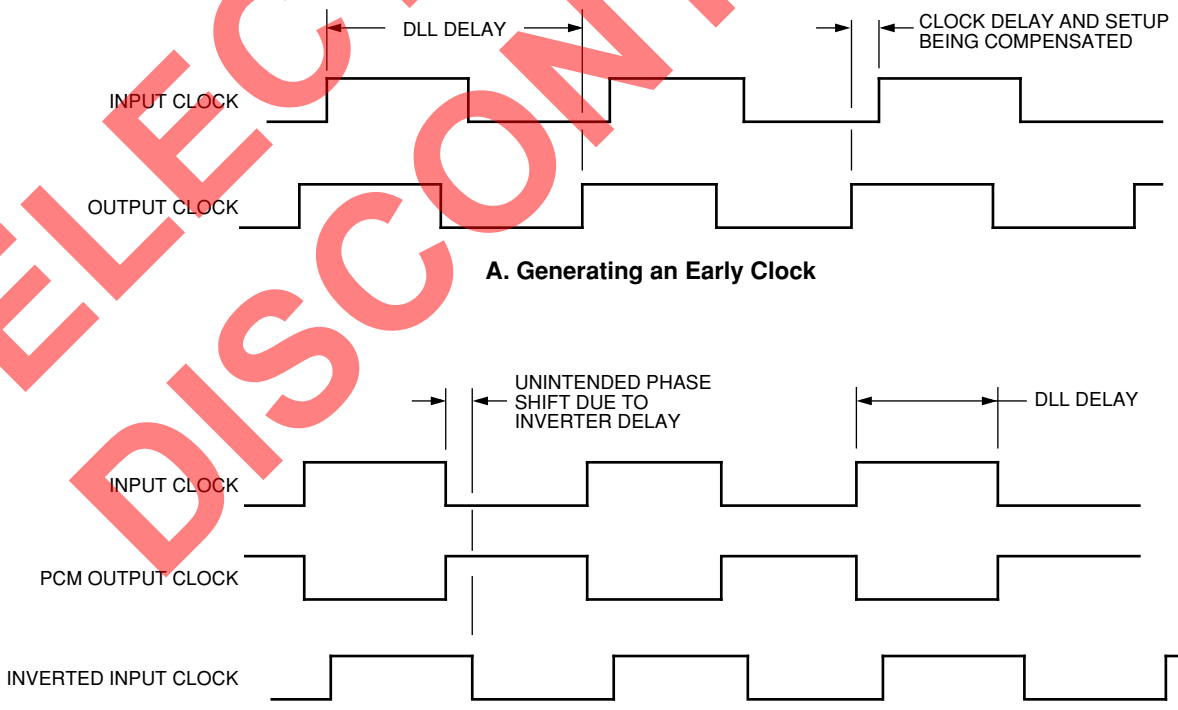
An output clock that precedes the input clock can be used to compensate for clock delay that is largely due to excessive loading. The preceding output clock is really not early relative to the input clock, but is delayed almost a full cycle. This is shown in Figure 48A. The amount of delay that is being 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

Programmable Clock Manager (PCM)
(continued)

High-Speed Internal Processing with Slow I/Os

The PCM PLL mode provides two outputs, one sent to the global system clock routing of the FPGA and the other to the ExpressCLK(s) that serve the FPGA I/Os. The ExpressCLK output of the PCM has a divide capability (DIV2) that the system clock output does not. This feature allows an input clock to be multiplied up to a higher frequency for high-speed internal processing, and also allows the ExpressCLK output to be divided down to a lower frequency to accommodate off-FPGA data transfers. For example, a 10 MHz input clock may be multiplied (see Clock Multiplication in the Phase-Locked Loop (PLL) Mode subsection) to 25 MHz (DIV0 = 4, DIV1 = 5, DIV2 = 2) and output to the FPGA ExpressCLK. This allows the I/Os of the circuit to run at 25 MHz ((2 * 5)/4 * 10 MHz). The system clock will run at DIV2 times the ExpressCLK rate, which is 2 times 25 MHz, or 50 MHz. This setup allows for internal processing to occur at twice the rate of on/off device I/O transfers.

PCM Cautions

Cautions do apply when using the PCM. There are a number of configurations that are possible in the PCM that are theoretically valid, but may not produce viable results. This section describes some of those situations, and should leave the user with an understanding of the types of pitfalls that must be avoided when modifying clock signals.

Resultant signals from the PCM must meet the FPGA timing specifications. It is possible to specify pulses by using duty-cycle adjustments that are too narrow to function in the FPGA. For instance, if a 40 MHz clock is doubled to 80 MHz and a 6.25% duty cycle is selected, the result will be a 780 ps pulse that repeats every 12.5 ns. This pulse falls outside of the clock pulse width specification and is not valid.

Using divider DIV2, it is possible to specify a clock multiplication factor of 64 between the input clock and the output system clock. As mentioned above, the resultant frequency must meet all FPGA timing specifications. The input clock must also meet the minimum specifications. An input clock rate that is below the PCM clock minimum cannot be used even if the multiplied output is within the allowable range.

The use of the PCM to tweak a clock signal to eliminate a particular problem, such as a single setup time violation, is discouraged. A small shift in delay, duty cycle, or phase to correct a single-point problem is in essence an asynchronous patch to a synchronous system, making the system less stable. This type of local problem, as opposed to a global clock control issue like device-wide clock delay, can usually be eliminated through more robust design practices. If this type of change is made, the designer must be aware that depending on the extent of the change made, the design may fail to operate correctly in a different speed grade or voltage grade (e.g., 3C vs. 3T), or even in a different production lot of the same device.

Divider DIV2 is available in DLL mode for the ExpressCLK output, but its use is not recommended with duty-cycle adjusted clocks.

FPGA States of Operation

Prior to becoming operational, the FPGA goes through a sequence of states, including initialization, configuration, and start-up. Figure 49 outlines these three FPGA states.

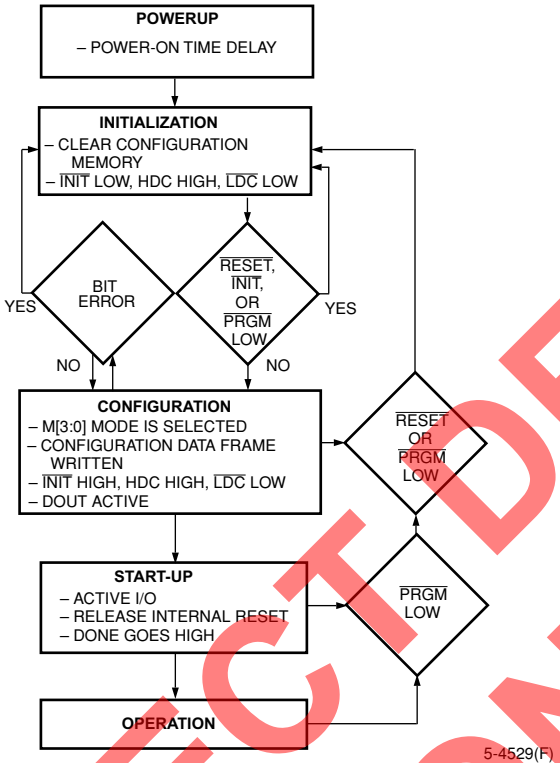


Figure 49. FPGA States of Operation

Initialization

Upon powerup, the device goes through an initialization process. First, an internal power-on-reset circuit is triggered when power is applied. When V_{DD} reaches the voltage at which portions of the FPGA begin to operate (2.5 V to 3 V for the OR3Cxx, 2.2 V to 2.7 V for the OR3Txxx), the I/Os are configured based on the configuration mode, as determined by the mode select inputs M[2:0]. A time-out delay is initiated when V_{DD} reaches between 3.0 V and 4.0 V (OR3Cxx) or 2.7 V to 3.0 V (OR3Txxx) to allow the power supply voltage to stabilize. The $\overline{INIT}$ and DONE outputs are low. At powerup, if V_{DD} does not rise from 2.0 V to V_{DD} in less than 25 ms, the user should delay configuration by inputting a low into $\overline{INIT}$, $\overline{PRGM}$, or $\overline{RESET}$ until V_{DD} is greater than the recommended minimum operating voltage (4.75 V for OR3Cxx commercial devices and 3.0 V for OR3Txxx devices).

At the end of initialization, the default configuration option is that the configuration RAM is written to a low state. This prevents shorts prior to configuration. As a configuration option, after the first configuration (i.e., at reconfiguration), the user can reconfigure without clearing the internal configuration RAM first. The active-low, open-drain initialization signal $\overline{INIT}$ is released and must be pulled high by an external resistor when initialization is complete. To synchronize the configuration of multiple FPGAs, one or more $\overline{INIT}$ pins should be wire-ANDed. If $\overline{INIT}$ is held low by one or more FPGAs or an external device, the FPGA remains in the initialization state. $\overline{INIT}$ can be used to signal that the FPGAs are not yet initialized. After $\overline{INIT}$ goes high for two internal clock cycles, the mode lines (M[3:0]) are sampled, and the FPGA enters the configuration state.

The high during configuration (HDC), low during configuration ($\overline{LDC}$), and DONE signals are active outputs in the FPGA's initialization and configuration states. HDC, $\overline{LDC}$, and DONE can be used to provide control of external logic signals such as reset, bus enable, or PROM enable during configuration. For parallel master configuration modes, these signals provide PROM enable control and allow the data pins to be shared with user logic signals.

FPGA States of Operation (continued)

If configuration has begun, an assertion of $\overline{\text{RESET}}$ or $\overline{\text{PRGM}}$ initiates an abort, returning the FPGA to the initialization state. The $\overline{\text{PRGM}}$ and $\overline{\text{RESET}}$ pins must be pulled back high before the FPGA will enter the configuration state. During the start-up and operating states, only the assertion of $\overline{\text{PRGM}}$ causes a reconfiguration.

In the master configuration modes, the FPGA is the source of configuration clock (CCLK). In this mode, the initialization state is extended to ensure that, in daisy-chain operation, all daisy-chained slave devices are ready. Independent of differences in clock rates, master mode devices remain in the initialization state an additional six internal clock cycles after $\overline{\text{INIT}}$ goes high.

When configuration is initiated, a counter in the FPGA is set to 0 and begins to count configuration clock cycles applied to the FPGA. As each configuration data frame is supplied to the FPGA, it is internally assembled into data words. Each data word is loaded into the internal configuration memory. The configuration loading process is complete when the internal length count equals the loaded length count in the length count field, and the required end of configuration frame is written.

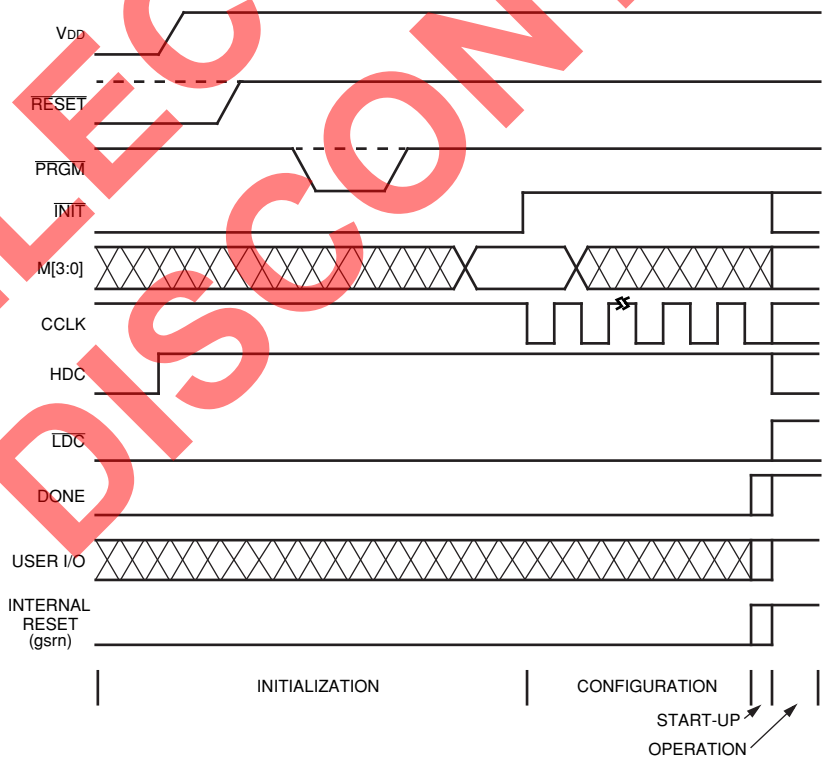
All OR3Cxx I/Os operate as TTL inputs during configuration (OR3Txxx I/Os are CMOS-only). All I/Os that are not used during the configuration process are 3-stated with internal pull-ups.

Warning: During configuration, all OR3Txxx inputs have internal pull-ups enabled. If these inputs are driven to 5V, they will draw substantial current (≅ 5 ma). This is due to the fact that the inputs are pulled up to 3V.

During configuration, the PIC and PLC latches/FFs are held set/reset and the internal BIDI buffers are 3-stated. The combinatorial logic begins to function as the FPGA is configured. Figure 50 shows the general waveform of the initialization, configuration, and start-up states.

Configuration

The *ORCA* Series FPGA functionality is determined by the state of internal configuration RAM. This configuration RAM can be loaded in a number of different modes. In these configuration modes, the FPGA can act as a master or a slave of other devices in the system. The decision as to which configuration mode to use is a system design issue. Configuration is discussed in detail, including the configuration data format and the configuration modes used to load the configuration data in the FPGA, following a description of the start-up state.



5-4482(F)

Figure 50. Initialization/Configuration/Start-Up Waveforms

FPGA States of Operation (continued)

Start-Up

After configuration, the FPGA enters the start-up phase. This phase is the transition between the configuration and operational states and begins when the number of CCLKs received after $\overline{\text{INIT}}$ goes high is equal to the value of the length count field in the configuration frame and when the end of configuration frame has been written. The system design issue in the start-up phase is to ensure the user I/Os become active without inadvertently activating devices in the system or causing bus contention. A second system design concern is the timing of the release of global set/reset of the PLC latches/FFs.

There are configuration options that control the relative timing of three events: DONE going high, release of the set/reset of internal FFs, and user I/Os becoming active. Figure 51 shows the start-up timing for *ORCA* FPGAs. The system designer determines the relative timing of the I/Os becoming active, DONE going high, and the release of the set/reset of internal FFs. In the *ORCA* Series FPGA, the three events can occur in any arbitrary sequence. This means that they can occur before or after each other, or they can occur simultaneously.

There are four main start-up modes: CCLK_NOSYNC, CCLK_SYNC, UCLK_NOSYNC, and UCLK_SYNC. The only difference between the modes starting with CCLK and those starting with UCLK is that for the UCLK modes, a user clock must be supplied to the start-up logic. The timing of start-up events is then based upon this user clock, rather than CCLK. The difference between the SYNC and NOSYNC modes is that for SYNC mode, the timing of two of the start-up events, release of the set/reset of internal FFs, and the I/Os becoming active is triggered by the rise of the external DONE pin followed by a variable number of rising clock edges (either CCLK or UCLK). For the NOSYNC mode, the timing of these two events is based only on either CCLK or UCLK.

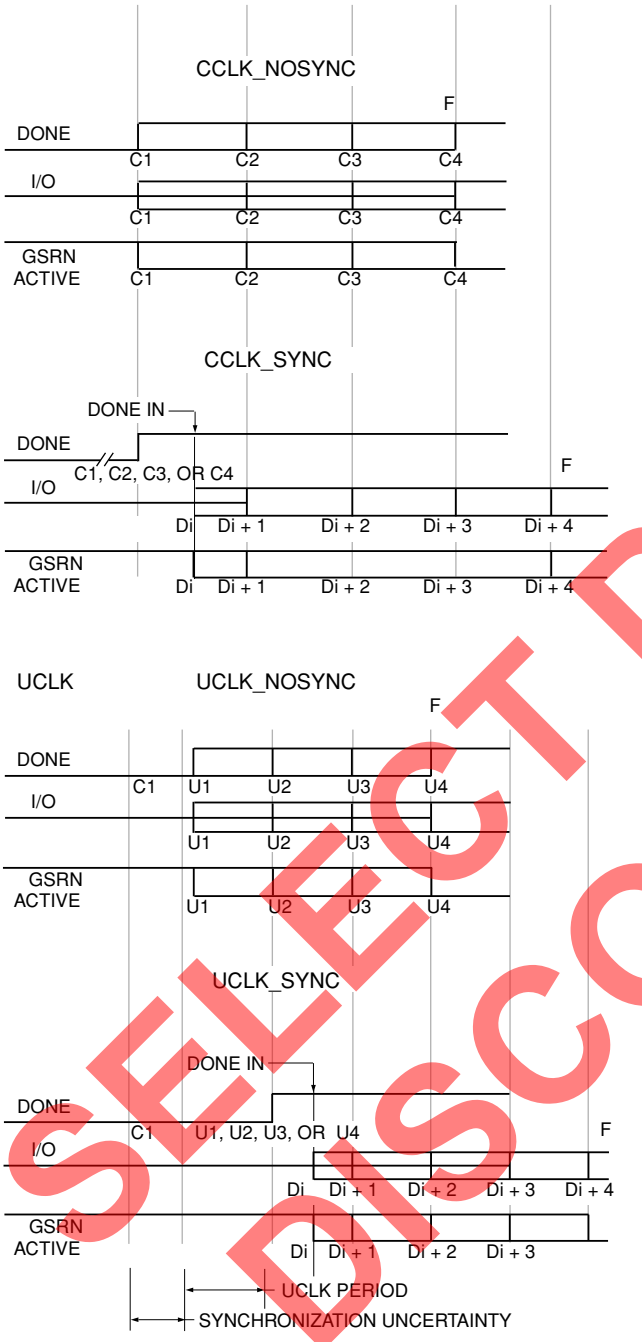
DONE is an open-drain bidirectional pin that may include an optional (enabled by default) pull-up resistor to accommodate wired ANDing. The open-drain DONE signals from multiple FPGAs can be tied together (ANDed) with a pull-up (internal or external) and used as an active-high ready signal, an active-low PROM enable, or a reset to other portions of the system. When used in SYNC mode, these ANDed DONE pins can be used to synchronize the other two start-up events, since they can all be synchronized to the same external signal. This signal will not rise until all FPGAs release their DONE pins, allowing the signal to be pulled high.

The default for *ORCA* is the CCLK_SYNC synchronized start-up mode where DONE is released on the first CCLK rising edge, C1 (see Figure 51). Since this is a synchronized start-up mode, the open-drain DONE signal can be held low externally to stop the occurrence of the other two start-up events. Once the DONE pin has been released and pulled up to a high level, the other two start-up events can be programmed individually to either happen immediately or after up to four rising edges of CCLK (Di, Di + 1, Di + 2, Di + 3, Di + 4). The default is for both events to happen immediately after DONE is released and pulled high.

A commonly used design technique is to release DONE one or more clock cycles before allowing the I/O to become active. This allows other configuration devices, such as PROMs, to be disconnected using the DONE signal so that there is no bus contention when the I/Os become active. In addition to controlling the FPGA during start-up, other start-up techniques that avoid contention include using isolation devices between the FPGA and other circuits in the system, reassigning I/O locations, and maintaining I/Os as 3-stated outputs until contentions are resolved.

Each of these start-up options can be selected during bit stream generation in ispLEVER, using Advanced Options. For more information, please see the ispLEVER documentation.

FPGA States of Operation (continued)



Note: F = finished, no more CLKs required.

5-2761(F)

Figure 51. Start-Up Waveforms

Reconfiguration

To reconfigure the FPGA when the device is operating in the system, a low pulse is input into $\overline{\text{PRGM}}$. The configuration data in the FPGA is cleared, and the I/Os not used for configuration are 3-stated. The FPGA then samples the mode select inputs and begins reconfiguration. When reconfiguration is complete, DONE is released, allowing it to be pulled high.

Partial Reconfiguration

All *ORCA* device families have been designed to allow a partial reconfiguration of the FPGA at any time. This is done by setting a bit stream option in the previous configuration sequence that tells the FPGA to not reset all of the configuration RAM during a reconfiguration. Then only the configuration frames that are to be modified need to be rewritten, thereby reducing the configuration time.

Other bit stream options are also available that allow one portion of the FPGA to remain in operation while a partial reconfiguration is being done. If this is done, the user must be careful to not cause contention between the two configurations (the bit stream resident in the FPGA and the partial reconfiguration bit stream) as the second reconfiguration bit stream is being loaded.

Other Configuration Options

There are many other configuration options available to the user that can be set during bit stream generation in ispLEVER. These include options to enable boundary scan and/or the microprocessor interface (MPI) and/or the programmable clock manager (PCM), readback options, and options to control and use the internal oscillator after configuration.

Other useful options that affect the next configuration (not the current configuration process) include options to disable the global set/reset during configuration, disable the 3-state of I/Os during configuration, and disable the reset of internal RAMs during configuration to allow for partial configurations (see above). For more information on how to set these and other configuration options, please see the ispLEVER documentation.

Configuration Data Format

The ispLEVER Development System interfaces with front-end design entry tools and provides tools to produce a fully configured FPGA. This section discusses using the ispLEVER Development System to generate configuration RAM data and then provides the details of the configuration frame format.

The *ORCA* OR3Cxx and OR3Txxx Series FPGAs are bit stream compatible.

Using ispLEVER to Generate Configuration RAM Data

The configuration data bit stream defines the I/O functionality, logic, and interconnections within the FPGA. The bit stream is generated by the development system. The bit stream created by the bit stream generation tool is a series of 1s and 0s used to write the FPGA configuration RAM. It can be loaded into the FPGA using one of the configuration modes discussed later.

In the bit stream generator, the designer selects options that affect the FPGA's functionality. Using the output of the bit stream generator, **circuit_name.bit**, the development system's download tool can load the configuration data into the *ORCA* series FPGA evaluation board from a PC or workstation.

Alternatively, a user can program a PROM (such as a Serial ROM or a standard EPROM) and load the FPGA from the PROM. The development system's PROM programming tool produces a file in .mks or .exo format.

Configuration Data Frame

Configuration data can be presented to the FPGA in two frame formats: autoincrement and explicit. A detailed description of the frame formats is shown in Figure 52, Figure 53, and Table 32. The two modes are similar except that autoincrement mode uses assumed address incrementation to reduce the bit stream size, and explicit mode requires an address for each data frame. In both cases, the header frame begins with a series of 1s and a preamble of 0010, followed by a 24-bit length count field representing the total number of configuration clocks needed to complete the loading of the FPGAs.

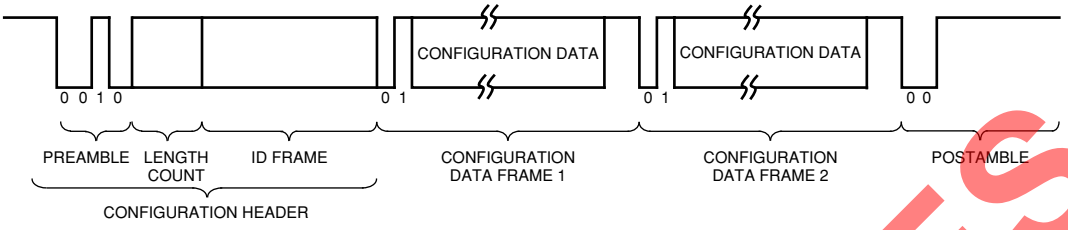
Following the header frame is a mandatory ID frame. (Note that the ID frame was optional in the *ORCA* 2C and 2C/TxxA Series.)

The ID frame contains data used to determine if the bit stream is being loaded to the correct type of *ORCA* FPGA (i.e., a bit stream generated for an OR3T55 is being sent to an OR3T55). Error checking is always enabled for Series 3 devices, through the use of an 8-bit checksum. One bit in the ID frame also selects between the autoincrement and explicit address modes for this load of the configuration data.

A configuration data frame follows the ID frame. A data frame starts with a 01-start bit pair and ends with enough 1-stop bits to reach a byte boundary. If using autoincrement configuration mode, subsequent data frames can follow. If using explicit mode, one or more address frames must follow each data frame, telling the FPGA at what addresses the preceding data frame is to be stored (each data frame can be sent to multiple addresses).

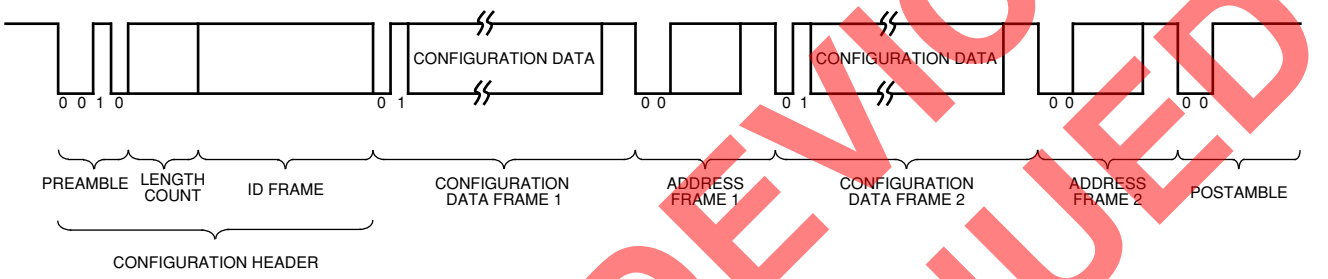
Following all data and address frames is the postamble. The format of the postamble is the same as an address frame with the highest possible address value with the checksum set to all ones.

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.

Configuration Data Format (continued)

The length and number of data frames and information on the PROM size for the Series 3 FPGAs are given in Table 33.

Table 33. Configuration Frame Size

Devices	OR3T20	OR3T30	OR3T55	OR3C/T80	OR3T125
# of Frames	856	984	1240	1496	1880
Data Bits/Frame	202	232	292	352	442
Configuration Data (# of frames x # of data bits/frame)	172,912	228,288	362,080	526,592	830,960
Maximum Total # Bits/Frame (align bits, 01 frame start, 8-bit checksum, 8 stop bits)	224	256	312	376	464
Maximum Configuration Data (# bits/frame x # of frames)	191,744	251,904	386,880	562,496	872,320
Maximum PROM Size (bits) (add configuration header and postamble)	191,912	252,072	387,048	562,664	872,488

Bit Stream Error Checking

There are three different types of bit stream error checking performed in the *ORCA* Series 3 FPGAs: ID frame, frame alignment, and CRC checking.

The ID data frame is sent to a dedicated location in the FPGA. This ID frame contains a unique code for the device for which it was generated. This device code is compared to the internal code of the FPGA. Any differences are flagged as an ID error. This frame is automatically created by the bit stream generation program in ispLEVER.

Each data and address frame in the FPGA begins with a frame start pair of bits and ends with eight stop bits set to 1. If any of the previous stop bits were a 0 when a frame start pair is encountered, it is flagged as a frame alignment error.

Error checking is also done on the FPGA for each frame by means of a checksum byte. If an error is found on evaluation of the checksum byte, then a checksum/parity error is flagged. The checksum is the XOR of all the data bytes, from the start of frame up to and including the bytes before the checksum. It applies to the ID, address, and data frames.

When any of the three possible errors occur, the FPGA is forced into an idle state, forcing $\overline{\text{INIT}}$ low. The FPGA will remain in this state until either the $\overline{\text{RESET}}$ or $\overline{\text{PRGM}}$ pins are asserted.

If using either of the MPI modes to configure the FPGA, the specific type of bit stream error is written to one of the MPI registers by the FPGA configuration logic. The $\overline{\text{PGRM}}$ bit of the MPI control register can also be used to reset out of the error condition and restart configuration.

FPGA Configuration Modes

There are eight methods for configuring the FPGA. Seven of the configuration modes are selected on the M0, M1, and M2 inputs. The eighth configuration mode is accessed through the boundary-scan interface. A fourth input, M3, is used to select the frequency of the internal oscillator, which is the source for CCLK in some configuration modes. The nominal frequencies of the internal oscillator are 1.25 MHz and 10 MHz. The 1.25 MHz frequency is selected when the M3 input is unconnected or driven to a high state.

There are three basic FPGA configuration modes: master, slave, and peripheral. The configuration data can be transmitted to the FPGA serially or in parallel bytes. As a master, the FPGA provides the control signals out to strobe data in. As a slave device, a clock is generated externally and provided into the CCLK input. In the three peripheral modes, the FPGA acts as a microprocessor peripheral. Table 34 lists the functions of the configuration mode pins. Note that two configuration modes previously available on the OR2Cxx and OR2C/TxxA devices (master parallel down and synchronous peripheral) have been removed for Series 3 devices.

Table 34. Configuration Modes

M2	M1	M0	CCLK	Configuration Mode	Data
0	0	0	Output	Master Serial	Serial
0	0	1	Input	Slave Parallel	Parallel
0	1	0	Output	Microprocessor: *Motorola* PowerPC*	Parallel
0	1	1	Output	Microprocessor: *Intel i960*	Parallel
1	0	0	Output	Master Parallel	Parallel
1	0	1	Output	Async Peripheral	Parallel
1	1	0	Reserved		
1	1	1	Input	Slave Serial	Serial

* *Motorola* is a registered trademark of Motorola, Inc.

Master Parallel Mode

The master parallel configuration mode is generally used to interface to industry-standard, byte-wide memory, such as the 2764 and larger EPROMs. Figure 54 provides the connections for master parallel mode. The FPGA outputs an 18-bit address on A[17:0] to memory and reads 1 byte of configuration data on the rising edge of RCLK. The parallel bytes are internally serialized starting with the least significant bit, D0. D[7:0] of the FPGA can be connected to D[7:0] of the microprocessor only if a standard prom file format is used. If a .bit or .rbt file is used from ispLEVER, then the user must mirror the bytes in the .bit or .rbt file OR leave the .bit or .rbt file unchanged and connect D[7:0] of the FPGA to D[0:7] of the microprocessor.

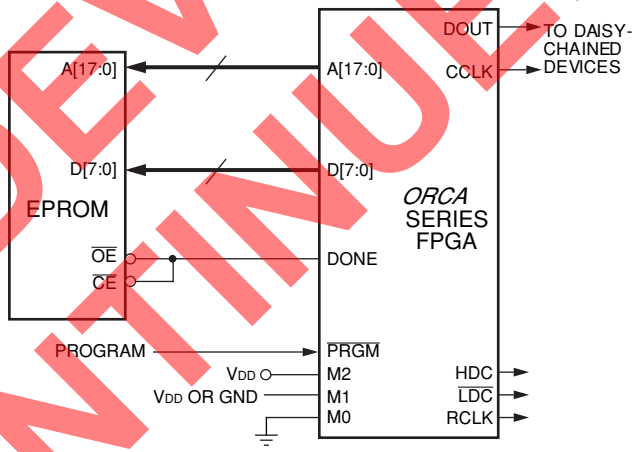


Figure 54. Master Parallel Configuration Schematic

In master parallel mode, the starting memory address is 00000 Hex, and the FPGA increments the address for each byte loaded.

One master mode FPGA can interface to the memory and provide configuration data on DOUT to additional FPGAs in a daisy-chain. The configuration data on DOUT is provided synchronously with the falling edge of CCLK. The frequency of the CCLK output is eight times that of RCLK.

FPGA Configuration Modes (continued)

Master Serial Mode

In the master serial mode, the FPGA loads the configuration data from an external serial ROM. The configuration data is either loaded automatically at start-up or on a $\overline{\text{PRGM}}$ command to reconfigure. The ATT1700A Series Serial PROMs can be used to configure the FPGA in the master serial mode. This provides a simple 4-pin interface in a compact package.

Configuration in the master serial mode can be done at powerup and/or upon a configure command. The system or the FPGA must activate the serial ROM's $\overline{\text{RESET}}$/OE and $\overline{\text{CE}}$ inputs. At powerup, the FPGA and serial ROM each contain internal power-on reset circuitry that allows the FPGA to be configured without the system providing an external signal. The power-on reset circuitry causes the serial ROM's internal address pointer to be reset. After powerup, the FPGA automatically enters its initialization phase.

The serial ROM/FPGA interface used depends on such factors as the availability of a system reset pulse, availability of an intelligent host to generate a configure command, whether a single serial ROM is used or multiple serial ROMs are cascaded, whether the serial ROM contains a single or multiple configuration programs, etc. Because of differing system requirements and capabilities, a single FPGA/serial ROM interface is generally not appropriate for all applications.

Data is read in the FPGA sequentially from the serial ROM. The DATA output from the serial ROM is connected directly into the DIN input of the FPGA. The CCLK output from the FPGA is connected to the CLK input of the serial ROM. During the configuration process, CCLK clocks one data bit on each rising edge.

Since the data and clock are direct connects, the FPGA/serial ROM design task is to use the system or FPGA to enable the $\overline{\text{RESET}}$/OE and $\overline{\text{CE}}$ of the serial ROM(s). There are several methods for enabling the serial ROM's $\overline{\text{RESET}}$/OE and $\overline{\text{CE}}$ inputs. The serial ROM's $\overline{\text{RESET}}$/OE is programmable to function with RESET active-high and $\overline{\text{OE}}$ active-low or $\overline{\text{RESET}}$ active-low and OE active-high.

In Figure 55, serial ROMs are cascaded to configure multiple daisy-chained FPGAs. The host generates a 500 ns low pulse into the FPGA's $\overline{\text{PRGM}}$ input. The FPGA's $\overline{\text{INIT}}$ input is connected to the serial ROMs' $\overline{\text{RESET}}$/OE input, which has been programmed to function with $\overline{\text{RESET}}$ active-low and OE active-high. The FPGA DONE is routed to the $\overline{\text{CE}}$ pin. The low on DONE enables the serial ROMs. At the completion of configuration, the high on the FPGA's DONE disables the serial ROM.

Serial ROMs can also be cascaded to support the configuration of multiple FPGAs or to load a single FPGA when configuration data requirements exceed the capacity of a single serial ROM. After the last bit from the first serial ROM is read, the serial ROM outputs $\overline{\text{CEO}}$ low and 3-states the DATA output. The next serial ROM recognizes the low on $\overline{\text{CE}}$ input and outputs configuration data on the DATA output. After configuration is complete, the FPGA's DONE output into $\overline{\text{CE}}$ disables the serial ROMs.

This FPGA/serial ROM interface is not used in applications in which a serial ROM stores multiple configuration programs. In these applications, the next configuration program to be loaded is stored at the ROM location that follows the last address for the previous configuration program. The reason the interface in Figure 55 will not work in this application is that the low output on the $\overline{\text{INIT}}$ signal would reset the serial ROM address pointer, causing the first configuration to be reloaded.

In some applications, there can be contention on the FPGA's DIN pin. During configuration, DIN receives configuration data, and after configuration, it is a user I/O. If there is contention, an early DONE at start-up (selected in ispLEVER) may correct the problem. An alternative is to use $\overline{\text{LDC}}$ to drive the serial ROM's $\overline{\text{CE}}$ pin. In order to reduce noise, it is generally better to run the master serial configuration at 1.25 MHz (M3 pin tied high), rather than 10 MHz, if possible.

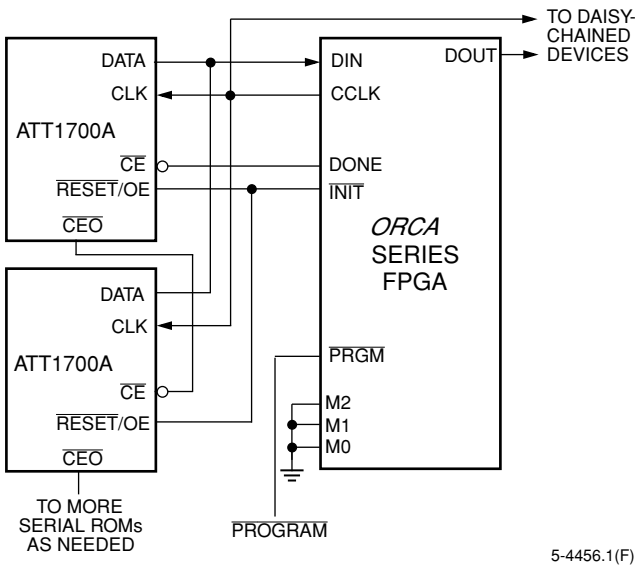


Figure 55. Master Serial Configuration Schematic

FPGA Configuration Modes (continued)

Asynchronous Peripheral Mode

Figure 56 shows the connections needed for the asynchronous peripheral mode. In this mode, the FPGA system interface is similar to that of a microprocessor-peripheral interface. The microprocessor generates the control signals to write an 8-bit byte into the FPGA. The FPGA control inputs include active-low $\overline{CS0}$ and active-high CS1 chip selects and $\overline{WR}$ and $\overline{RD}$ inputs. The chip selects can be cycled or maintained at a static level during the configuration cycle. Each byte of data is written into the FPGA's D[7:0] input pins. D[7:0] of the FPGA can be connected to D[7:0] of the microprocessor only if a standard prom file format is used. If a .bit or .rbt file is used from ispLEVER, then the user must mirror the bytes in the .bit or .rbt file OR leave the .bit or .rbt file unchanged and connect D[7:0] of the FPGA to D[0:7] of the microprocessor.

The FPGA provides an RDY/$\overline{BUSY}$ status output to indicate that another byte can be loaded. A low on RDY/$\overline{BUSY}$ indicates that the double-buffered hold/shift registers are not ready to receive data, and this pin must be monitored to go high before another byte of data can be written. The shortest time RDY/$\overline{BUSY}$ is low occurs when a byte is loaded into the hold register and the shift register is empty, in which case the byte is immediately transferred to the shift register. The longest time for RDY/$\overline{BUSY}$ to remain low occurs when a byte is loaded into the holding register and the shift register has just started shifting configuration data into configuration RAM.

The RDY/$\overline{BUSY}$ status is also available on the D7 pin by enabling the chip selects, setting $\overline{WR}$ high, and applying $\overline{RD}$ low, where the $\overline{RD}$ input provides an output enable for the D7 pin when $\overline{RD}$ is low. The D[6:0] pins are not enabled to drive when $\overline{RD}$ is low and, therefore, only act as input pins in asynchronous peripheral mode. Optionally, the user can ignore the RDY/$\overline{BUSY}$ status and simply wait until the maximum time it would take for the RDY/$\overline{BUSY}$ line to go high, indicating the FPGA is ready for more data, before writing the next data byte.

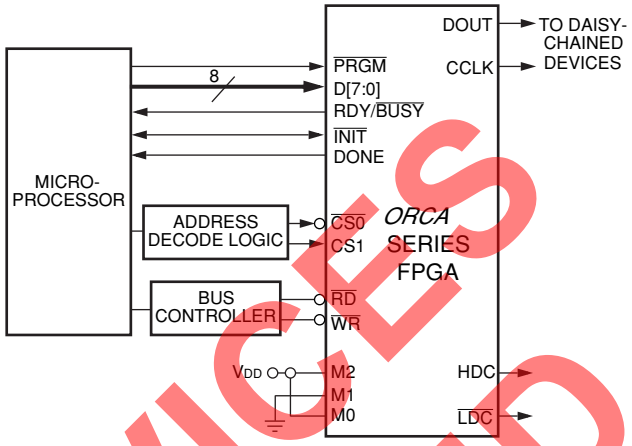


Figure 56. Asynchronous Peripheral Configuration

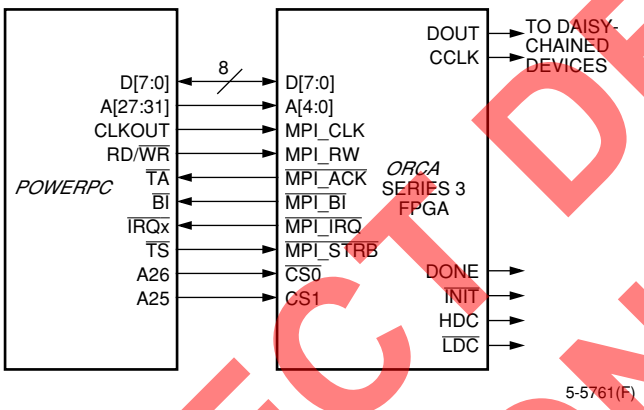
Microprocessor Interface (MPI) Mode

The built-in MPI in Series 3 FPGAs is designed for use in configuring the FPGA. Figure 57 and Figure 58 show the glueless interface for FPGA configuration and readback from the *PowerPC* and *i960* processors, respectively. When enabled by the mode pins, the MPI handles all configuration/readback control and handshaking with the host processor. For single FPGA configuration, the host sets the configuration control register $\overline{PRGM}$ bit to zero then back to a one and, after reading that the $\overline{INIT}$ signal is high in the MPI status register, transfers data 8 bits at a time to the FPGA's D[7:0] input pins.

If configuring multiple FPGAs through daisy-chain operation is desired, the MP_DAISY bit must be set in the configuration control register of the MPI. Because of the latency involved in a daisy-chain configuration, the MP_HOLD_BUS bit may be set to zero rather than one for daisy-chain operation. This allows the MPI to acknowledge the data transfer before the configuration information has been serialized and transferred on the FPGA daisy-chain. The early acknowledgment frees the host processor to perform other system tasks. Configuring with the MP_HOLD_BUS bit at zero requires that the host microprocessor poll the RDY/$\overline{BUSY}$ bit of the MPI status register and/or use the MPI interrupt capability to confirm the readiness of the MPI for more configuration data.

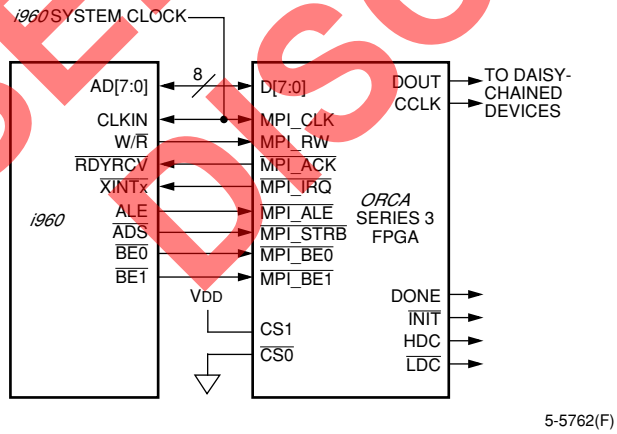
FPGA Configuration Modes (continued)

There are two options for using the host interrupt request in configuration mode. The configuration control register offers control bits to enable the interrupt on either a bit stream error or to notify the host processor when the FPGA is ready for more configuration data. The MPI status register may be used in conjunction with, or in place of, the interrupt request options. The status register contains a 2-bit field to indicate the bit stream error status. As previously mentioned, there is also a bit to indicate the MPI's readiness to receive another byte of configuration data. A flow chart of the MPI configuration process is shown in Figure 59. The MPI status and configuration register bit maps can be found in the Special Function Blocks section and MPI configuration timing information is available in the Timing Characteristics section of this data sheet.



Note: FPGA shown as a memory-mapped peripheral using CS0 and CS1. Other decoding schemes are possible using CS0 and/or CS1.

Figure 57. *PowerPC*/MPI Configuration Schematic



Note: FPGA shown as only system peripheral with fixed chip select signals. For multiperipheral systems, address decoding and/or latching can be used to implement chip selects.

Figure 58. *i960*/MPI Configuration Schematic

Configuration readback can also be performed via the MPI when it is in user mode. The MPI is enabled in user mode by setting the MP_USER bit to 1 in the configuration control register prior to the start of configuration or through a configuration option. To perform readback, the host processor writes the 14-bit readback start address to the readback address registers and sets the RD_CFG bit to 0 in the configuration control register. Readback data is returned 8 bits at a time to the readback data register and is valid when the DATA_RDY bit of the status register is 1. There is no error checking during readback. A flow chart of the MPI readback operation is shown in Figure 60. The RD_DATA pin used for dedicated FPGA readback is invalid during MPI readback.

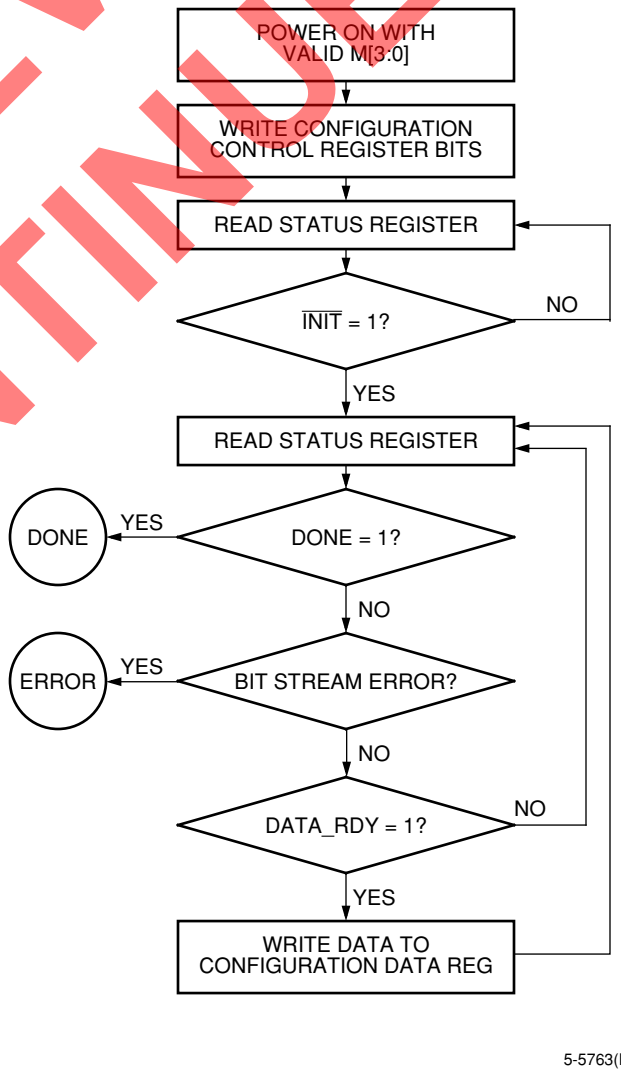
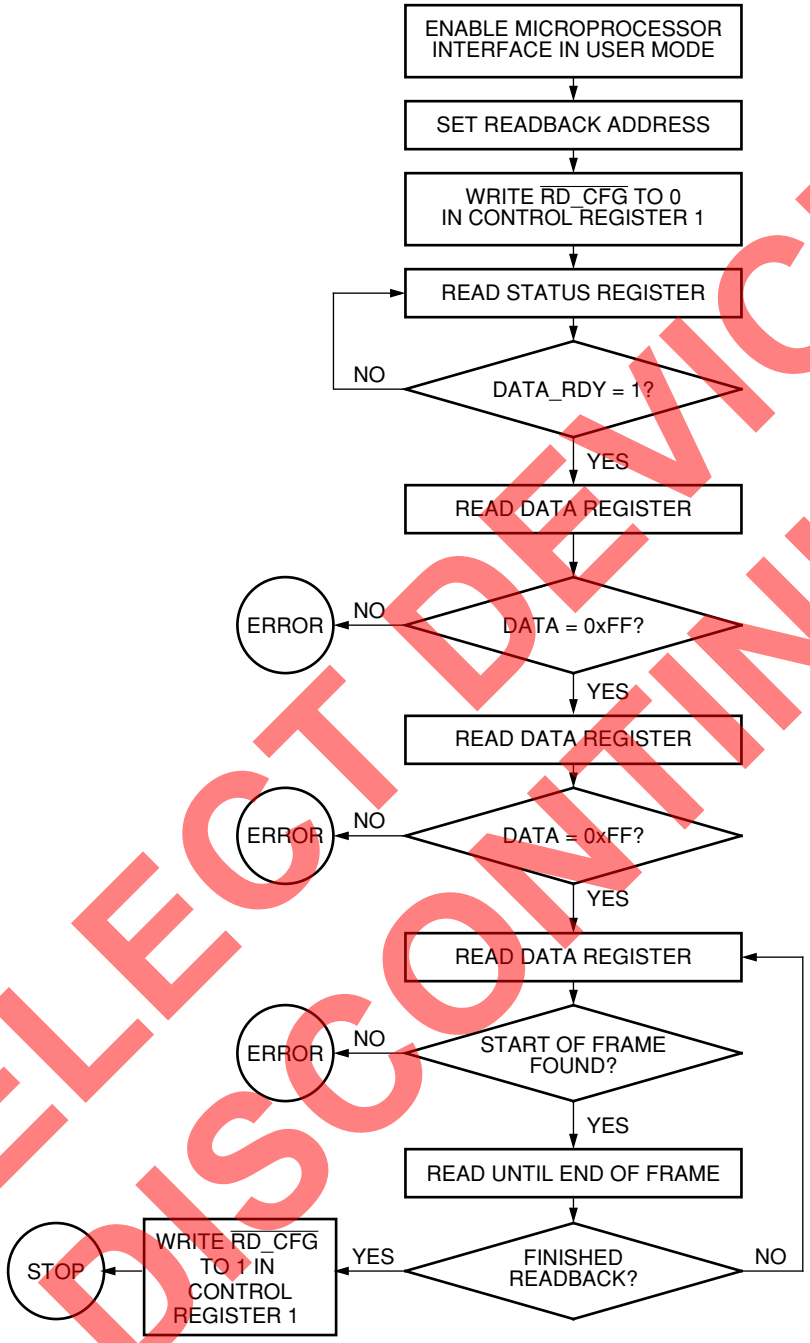


Figure 59. Configuration Through MPI

FPGA Configuration Modes (continued)



5-5764(F)

Figure 60. Readback Through MPI

FPGA Configuration Modes (continued)

Slave Serial Mode

The slave serial mode is primarily used when multiple FPGAs are configured in a daisy-chain (see the Daisy-Chaining section). It is also used on the FPGA evaluation board that interfaces to the download cable. A device in the slave serial mode can be used as the lead device in a daisy-chain. Figure 61 shows the connections for the slave serial configuration mode.

The configuration data is provided into the FPGA's DIN input synchronous with the configuration clock CCLK input. After the FPGA has loaded its configuration data, it retransmits the incoming configuration data on DOUT. CCLK is routed into all slave serial mode devices in parallel.

Multiple slave FPGAs can be loaded with identical configurations simultaneously. This is done by loading the configuration data into the DIN inputs in parallel.

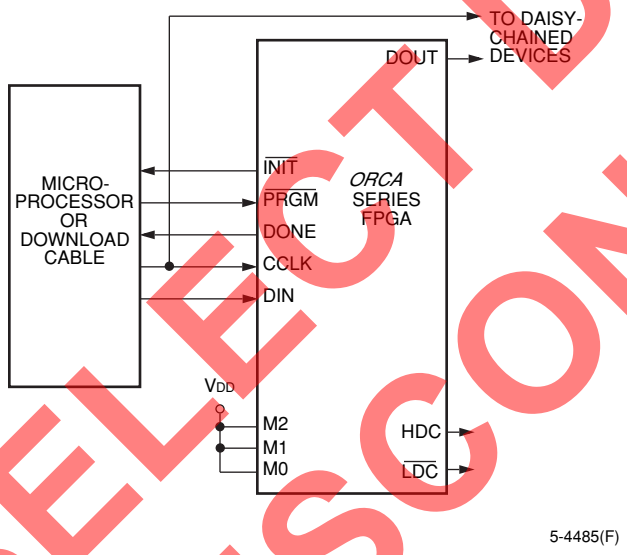


Figure 61. Slave Serial Configuration Schematic

Slave Parallel Mode

The slave parallel mode is essentially the same as the slave serial mode except that 8 bits of data are input on pins D[7:0] for each CCLK cycle. Due to 8 bits of data being input per CCLK cycle, the DOUT pin does not contain a valid bit stream for slave parallel mode. As a result, the lead device cannot be used in the slave parallel mode in a daisy-chain configuration.

Figure 62 is a schematic of the connections for the slave parallel configuration mode. $\overline{WR}$ and $\overline{CS0}$ are active-low chip select signals, and CS1 is an active-high chip select signal. These chip selects allow the user to configure multiple FPGAs in slave parallel mode using an 8-bit data bus common to all of the FPGAs. These chip selects can then be used to select the FPGA(s) to be configured with a given bit stream. The chip selects must be active for each valid CCLK cycle until the device has been completely programmed. They can be inactive between cycles but must meet the setup and hold times for each valid positive CCLK. D[7:0] of the FPGA can be connected to D[7:0] of the microprocessor only if a standard prom file format is used. If a .bit or .rbt file is used from ispLEVER, then the user must mirror the bytes in the .bit or .rbt file OR leave the .bit or .rbt file unchanged and connect D[7:0] of the FPGA to D[0:7] of the microprocessor.

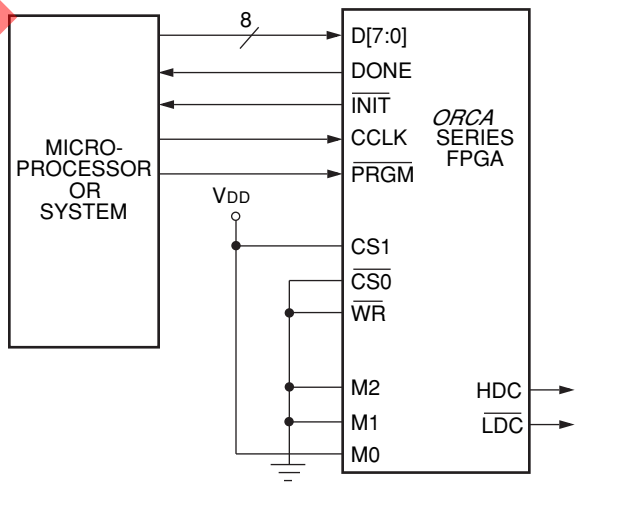


Figure 62. Slave Parallel Configuration Schematic

FPGA Configuration Modes (continued)

Daisy-Chaining

Multiple FPGAs can be configured by using a daisy-chain of the FPGAs. Daisy-chaining uses a lead FPGA and one or more FPGAs configured in slave serial mode. The lead FPGA can be configured in any mode except slave parallel mode. (Daisy-chaining is available with the boundary-scan ram_w instruction discussed later.)

All daisy-chained FPGAs are connected in series. Each FPGA reads and shifts the preamble and length count in on positive CCLK and out on negative CCLK edges.

An upstream FPGA that has received the preamble and length count outputs a high on DOUT until it has received the appropriate number of data frames so that downstream FPGAs do not receive frame start bit pairs. After loading and retransmitting the preamble and length count to a daisy-chain of slave devices, the lead device loads its configuration data frames.

The loading of configuration data continues after the lead device has received its configuration data if its internal frame bit counter has not reached the length count. When the configuration RAM is full and the number of bits received is less than the length count field, the FPGA shifts any additional data out on DOUT.

The configuration data is read into DIN of slave devices on the positive edge of CCLK, and shifted out DOUT on the negative edge of CCLK. Figure 63 shows the connections for loading multiple FPGAs in a daisy-chain configuration.

The generation of CCLK for the daisy-chained devices that are in slave serial mode differs depending on the configuration mode of the lead device. A master parallel mode device uses its internal timing generator to produce an internal CCLK at eight times its memory address rate (RCLK). The asynchronous peripheral mode device outputs eight CCLKs for each write cycle. If the lead device is configured in slave mode, CCLK must be routed to the lead device and to all of the daisy-chained devices.

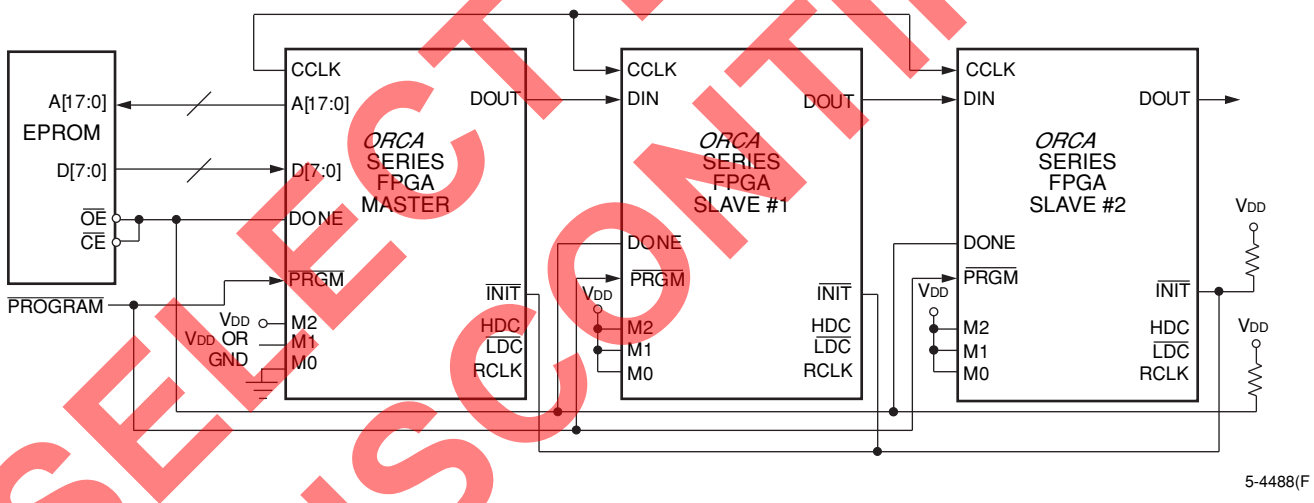


Figure 63. Daisy-Chain Configuration Schematic

As seen in Figure 63, the $\overline{\text{INIT}}$ pins for all of the FPGAs are connected together. This is required to guarantee that powerup and initialization will work correctly. In general, the DONE pins for all of the FPGAs are also connected together as shown to guarantee that all of the FPGAs enter the start-up state simultaneously. This may not be required, depending upon the start-up sequence desired.

FPGA Configuration Modes (continued)

Daisy-Chaining with Boundary Scan

Multiple FPGAs can be configured through the JTAG ports by using a daisy-chain of the FPGAs. This daisy-chaining operation is available upon initial configuration after powerup, after a power-on reset, after pulling the program pin to reset the chip, or during a reconfiguration if the EN_JTAG RAM has been set.

All daisy-chained FPGAs are connected in series. Each FPGA reads and shifts the preamble and length count in on the positive TCK and out on the negative TCK edges.

An upstream FPGA that has received the preamble and length count outputs a high on TDO until it has received the appropriate number of data frames so that downstream FPGAs do not receive frame start bit pairs. After loading and retransmitting the preamble and length count to a daisy-chain of downstream devices, the lead device loads its configuration data frames.

The loading of configuration data continues after the lead device had received its configuration read into TDI of downstream devices on the positive edge of TCK, and shifted out TDO on the negative edge of TCK. Figure 63 shows the connections for loading multiple FPGAs in a JTAG daisy-chain configuration.

Absolute Maximum Ratings

Stresses in excess of the absolute maximum ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operations sections of this data sheet. Exposure to absolute maximum ratings for extended periods can adversely affect device reliability.

The *ORCA* Series FPGAs include circuitry designed to protect the chips from damaging substrate injection currents and to prevent accumulations of static charge. Nevertheless, conventional precautions should be observed during storage, handling, and use to avoid exposure to excessive electrical stress.

Table 35. Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit
Storage Temperature	T_{stg}	–65	150	°C
Supply Voltage with Respect to Ground	V_{DD}	–0.5	7.0	V
Input Signal with Respect to Ground	—	–0.5	V_{DD} + 0.3	V
Signal Applied to High-impedance Output	—	–0.5	V_{DD} + 0.3	V
Maximum Package Body Temperature	—	—	220	°C

Recommended Operating Conditions

Table 36. Recommended Operating Conditions

Mode	OR3Cxx		OR3Txxx	
	Temperature Range (Ambient)	Supply Voltage (V_{DD})	Temperature Range (Ambient)	Supply Voltage (V_{DD})
Commercial	0 °C to 70 °C	5 V ± 5%	0 °C to 70 °C	3.0 V to 3.6 V
Industrial	–40 °C to +85 °C	5 V ± 10%	–40 °C to +85 °C	3.0 V to 3.6 V

Note: The maximum recommended junction temperature (T_J) during operation is 125 °C.

Electrical Characteristics

Table 37. Electrical 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	Test Conditions	OR3Cxx		OR3Txxx		Unit
			Min	Max	Min	Max	
Input Voltage: High Low	 V_{IH} V_{IL}	Input configured as CMOS (includes OR3Txxx)	 50% V_{DD} GND – 0.5	 V_{DD} + 0.5 20% V_{DD}	 50% V_{DD} GND – 0.5	 V_{DD} + 0.5 30% V_{DD}	 V V
Input Voltage: High Low	 V_{IH} V_{IL}	OR3Txxx 5 V Tolerant	 — —	 — —	 50% V_{DD} GND – 0.5	 5.8 V 30% V_{DD}	 V V
Input Voltage: High Low	 V_{IH} V_{IL}	Input configured as TTL (not valid for OR3Txxx)	 2.0 –0.5	 V_{DD} + 0.3 0.8	 — —	 — —	 V V
Output Voltage: High Low	 V_{OH} V_{OL}	 V_{DD} = min, I_{OH} = 6 mA or 3 mA V_{DD} = min, I_{OL} = 12 mA or 6 mA	 2.4 —	 — 0.4	 2.4 —	 — 0.4	 V V
Input Leakage Current	I_L	V_{DD} = max, V_{IN} = V_{SS} or V_{DD}	–10	10	–10	10	µA
Standby Current: OR3T20 OR3T30 OR3T55 OR3C/T80 OR3T125	I_{DDSB}	OR3Cxx (T_A = 25 °C, V_{DD} = 5.0 V) OR3Txxx (T_A = 25 °C, V_{DD} = 3.3 V) internal oscillator running, no output loads, inputs V_{DD} or GND (after configuration)	 — — — — —	 — — 4.06 4.56 —	 — — — — —	 4.70 4.90 5.30 5.80 6.70	 mA mA mA mA mA
Standby Current: OR3T20 OR3T30 OR3T55 OR3C/T80 OR3T125	I_{DDSB}	OR3Cxx (T_A = 25 °C, V_{DD} = 5.0 V) OR3Txxx (T_A = 25 °C, V_{DD} = 3.3 V) internal oscillator stopped, no output loads, inputs V_{DD} or GND (after configuration)	 — — — — —	 — — 3.05 3.42 —	 — — — — —	 3.52 3.68 3.98 4.35 5.02	 mA mA mA mA mA
Powerup Current: OR3T20 OR3T30 OR3T55 OR3C/T80 OR3T125	Ipp	Power supply current @ approximately 1 V, within a recommended power supply ramp rate of 1 ms—200 ms	 — — 3.2 5.4 —	 — — — — —	 1.2 1.6 2.7 4.0 6.5	 — — — — —	 mA mA mA mA mA
Data Retention Voltage	V_{DR}	T_A = 25 °C	2.3	—	2.3	—	V
Input Capacitance	C_{IN}	OR3Cxx (T_A = 25 °C, V_{DD} = 5.0 V) OR3Txxx (T_A = 25 °C, V_{DD} = 3.3 V) Test frequency = 1 MHz	—	9	—	8	pF
Output Capacitance	C_{OUT}	OR3Cxx (T_A = 25 °C, V_{DD} = 5.0 V) OR3Txxx (T_A = 25 °C, V_{DD} = 3.3 V) Test frequency = 1 MHz	—	9	—	8	pF

Electrical Characteristics (continued)

Table 37. Electrical Characteristics (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.

Parameter	Symbol	Test Conditions	OR3Cxx		OR3Txxx		Unit
			Min	Max	Min	Max	
DONE Pull-up Resistor*	R_{DONE}	—	100	—	100	—	kΩ
M[3:0] Pull-up Resistors*	R_M	—	100	—	100	—	kΩ
I/O Pad Static Pull-up Current*	I_{PU}	OR3Cxx (V_{DD} = 5.25 V, V_{IN} = V_{SS}, T_A = 0 °C) OR3Txxx (V_{DD} = 3.6 V, V_{IN} = V_{SS}, T_A = 0 °C)	14.4	50.9	14.4	50.9	µA
I/O Pad Static Pull-down Current	I_{PD}	OR3Cxx (V_{DD} = 5.25 V, V_{IN} = V_{SS}, T_A = 0 °C) OR3Txxx (V_{DD} = 3.6 V, V_{IN} = V_{SS}, T_A = 0 °C)	26	103	26	103	µA
I/O Pad Pull-up Resistor*	R_{PU}	V_{DD} = all, V_{IN} = V_{SS}, T_A = 0 °C	100	—	100	—	kΩ
I/O Pad Pull-down Resistor	R_{PD}	V_{DD} = all, V_{IN} = V_{DD}, T_A = 0 °C	50	—	50	—	kΩ

* On the OR3Txxx devices, the pull-up resistor will externally pull the pin to a level 1.0 V below V_{DD}.

Note: For 3T devices driven to 5 V.

Timing Characteristics

Description

To define speed grades, the *ORCA* Series part number designation (see Ordering Information) uses a single-digit number to designate a speed grade. This number is not related to any single ac parameter. Higher numbers indicate a faster set of timing parameters. The actual speed sorting is based on testing the delay in a path consisting of an input buffer, combinatorial delay through all PLCs in a row, and an output buffer. Other tests are then done to verify other delay parameters, such as routing delays, setup times to FFs, etc.

The most accurate timing characteristics are reported by the timing analyzer in the ispLEVER Development System. A timing report provided by the development system after layout divides path delays into logic and routing delays. The timing analyzer can also provide logic delays prior to layout. While this allows routing budget estimates, there is wide variance in routing delays associated with different layouts.

The logic timing parameters noted in the Electrical Characteristics section of this data sheet are the same as those in the design tools. In the PFU timing given in Table 41—Table 48, symbol names are generally a concatenation of the PFU operating mode (as defined in Table 3) and the parameter type. The setup, hold, and propagation delay parameters, defined below, are designated in the symbol name by the SET, HLD, and DEL characters, respectively.

The values given for the parameters are the same as those used during production testing and speed binning of the devices. The junction temperature and supply voltage used to characterize the devices are listed in the delay tables. Actual delays at nominal temperature and voltage for best-case processes can be much better than the values given.

It should be noted that the junction temperature used in the tables is generally 85 °C. The junction temperature for the FPGA depends on the power dissipated by the device, the package thermal characteristics (Θ_{JA}), and the ambient temperature, as calculated in the following equation and as discussed further in the Package Thermal Characteristics section:

$$T_{Jmax} = T_{Amax} + (P \bullet \Theta_{JA})\ ^{\circ}C$$

Note: The user must determine this junction temperature to see if the delays from ispLEVER should be derated based on the following derating tables.

Table 38 and Table 39 provide approximate power supply and junction temperature derating for OR3Cxx commercial and industrial devices. Table 40 provides the same information for the OR3Txxx devices (both commercial and industrial). The delay values in this data sheet and reported by ispLEVER are shown as **1.00** in the tables. The method for determining the maximum junction temperature is defined in the Package Thermal Characteristics section. Taken cumulatively, the range of parameter values for best-case vs. worst-case processing, supply voltage, and junction temperature can approach 3 to 1.

Table 38. Derating for Commercial Devices (OR3Cxx)

TJ (¡C)	Power Supply Voltage		
	4.75 V	5.0 V	5.25 V
0	0.81	0.79	0.77
25	0.85	0.83	0.81
85	**1.00**	0.97	0.95
100	1.05	1.02	1.00
125	1.12	1.09	1.07

Table 39. Derating for Industrial Devices (OR3Cxx)

TJ (¡C)	Power Supply Voltage				
	4.5 V	4.75 V	5.0 V	5.25 V	5.5 V
—40	0.71	0.70	0.68	0.66	0.65
0	0.80	0.78	0.76	0.74	0.73
25	0.84	0.82	0.80	0.78	0.77
85	**1.00**	0.97	0.94	0.93	0.91
100	1.05	1.01	0.99	0.97	0.95
125	1.12	1.09	1.06	1.04	1.02

Table 40. Derating for Commercial/Industrial Devices (OR3Txxx)

TJ (¡C)	Power Supply Voltage		
	3.0 V	3.3 V	3.6 V
—40	0.73	0.66	0.61
0	0.82	0.73	0.68
25	0.87	0.78	0.72
85	**1.00**	0.90	0.83
100	1.04	0.94	0.87
125	1.10	1.00	0.92

Note: The derating tables shown above are for a typical critical path that contains 33% logic delay and 66% routing delay. Since the routing delay derates at a higher rate than the logic delay, paths with more than 66% routing delay will derate at a higher rate than shown in the table. The approximate derating values vs. temperature are 0.26% per °C for logic delay and 0.45% per °C for routing delay. The approximate derating values vs. voltage are 0.13% per mV for both logic and routing delays at 25 °C.

Timing Characteristics (continued)

In addition to supply voltage, process variation, and operating temperature, circuit and process improvements of the *ORCA* Series FPGAs over time will result in significant improvement of the actual performance over those listed for a speed grade. Even though lower speed grades may still be available, the distribution of yield to timing parameters may be several speed grades higher than that designated on a product brand. Design practices need to consider best-case timing parameters (e.g., delays = 0), as well as worst-case timing.

The routing delays are a function of fan-out and the capacitance associated with the CIPs and metal interconnect in the path. The number of logic elements that can be driven (fan-out) by PFUs is unlimited, although the delay to reach a valid logic level can exceed timing requirements. It is difficult to make accurate routing delay estimates prior to design compilation based on fan-out. This is because the CAE software may delete redundant logic inserted by the designer to reduce fan-out, and/or it may also automatically reduce fan-out by net splitting.

The waveform test points are given in the Input/Output Buffer Measurement Conditions section of this data sheet. The timing parameters given in the electrical characteristics tables in this data sheet follow industry practices, and the values they reflect are described below.

Propagation Delay—The time between the specified reference points. The delays provided are the worst case of the tphh and tpll delays for noninverting functions, tplh and tphl for inverting functions, and tphz and tplz for 3-state enable.

Setup Time—The interval immediately preceding the transition of a clock or latch enable signal, during which the data must be stable to ensure it is recognized as the intended value.

Hold Time—The interval immediately following the transition of a clock or latch enable signal, during which the data must be held stable to ensure it is recognized as the intended value.

3-State Enable—The time from when a 3-state control signal becomes active and the output pad reaches the high-impedance state.

PFU Timing

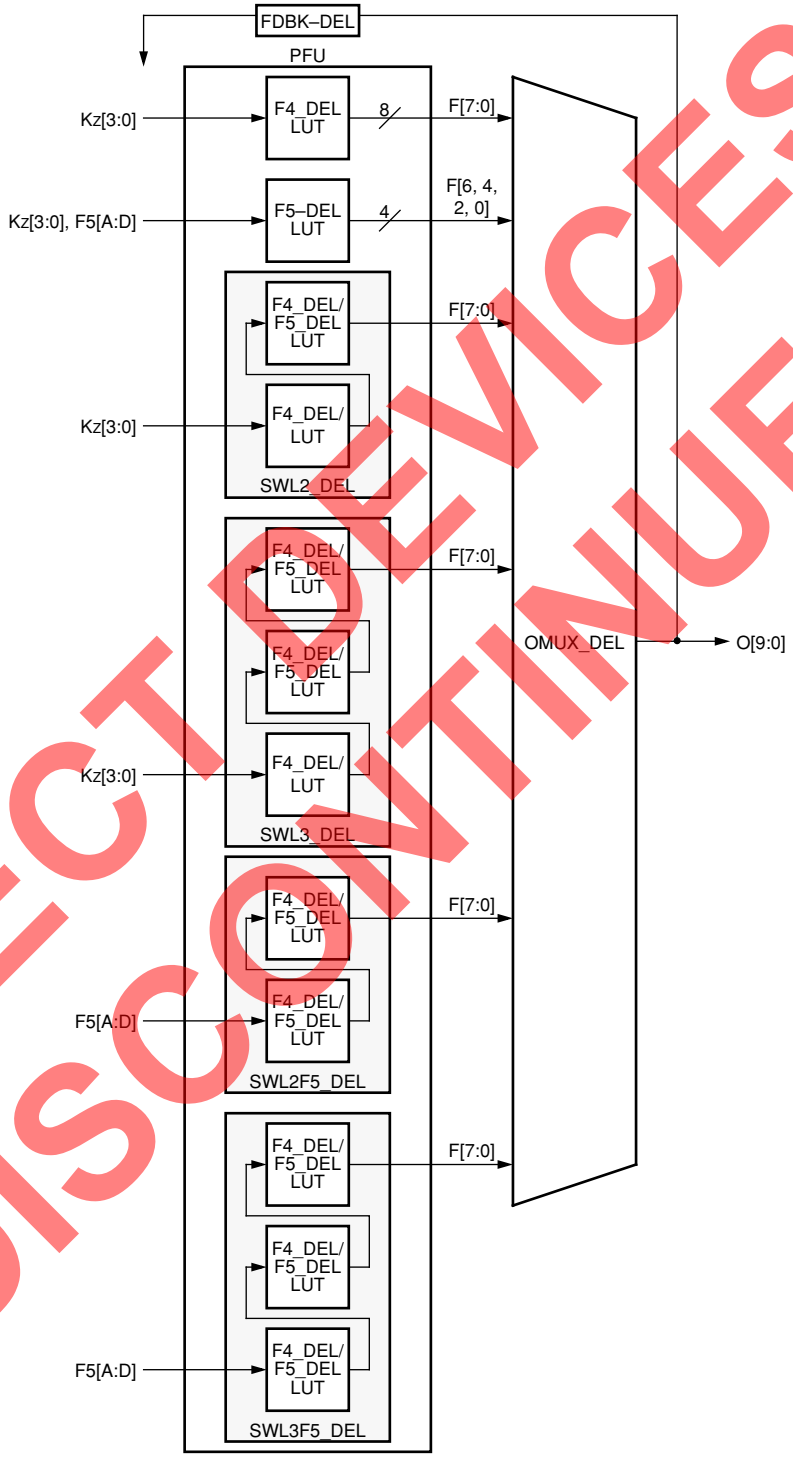
Table 41. Combinatorial PFU Timing Characteristics

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	
Combinatorial Delays (T_J = +85 °C, V_{DD} = min):										
Four-input Variables (Kz[3:0] to F[z])*	F4_DEL	—	2.34	—	1.80	—	1.32	—	1.05	ns
Five-input Variables (F5[A:D] to F[0, 2, 4, 6])	F5_DEL	—	2.11	—	1.57	—	1.23	—	0.99	ns
Two-level LUT Delay (Kz[3:0] to F w/feedbk)*	SWL2_DEL	—	4.87	—	3.66	—	2.58	—	2.03	ns
Two-level LUT Delay (F5[A:D] to F w/feedbk)	SWL2F5_DEL	—	4.69	—	3.51	—	2.48	—	1.94	ns
Three-level LUT Delay (Kz[3:0] to F w/feedbk)*	SWL3_DEL	—	6.93	—	5.15	—	3.63	—	2.82	ns
Three-level LUT Delay (F5[A:D] to F w/feedbk)	SWL3F5_DEL	—	6.89	—	5.08	—	3.54	—	2.75	ns
C_{IN} to C_{OUT} Delay (logic mode)	CO_DEL	—	3.47	—	2.65	—	1.79	—	1.43	ns

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

Timing Characteristics (continued)



Note: See Table 46 for an explanation of FDBK_DEL and OMUX_DEL.

5-5751(F)

Figure 64. Combinatorial PFU Timing

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 43. Ripple Mode 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 (T_J = +85 °C, V_{DD} = min)	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Full Ripple Setup Times (byte wide):										
Operands to Clock (Kz[1:0] to CLK)	RIP_SET	3.50	—	2.50	—	1.96	—	1.48	—	ns
Bitwise Operands to Clock (Kz[1:0] to CLK at F[z])	FRIP_SET	1.99	—	1.47	—	1.08	—	0.85	—	ns
Fast Carry-in to Clock (FCIN to CLK)	FCIN_SET	2.55	—	1.87	—	1.34	—	1.04	—	ns
Carry-in to Clock (CIN to CLK)	CIN_SET	3.80	—	2.79	—	1.97	—	1.56	—	ns
Add/Subtract to Clock (ASWE to CLK)	AS_SET	8.82	—	6.18	—	4.68	—	3.50	—	ns
Operands to Clock (Kz[1:0] to CLK at REGCOUT)	RIPRC_SET	2.09	—	1.61	—	1.19	—	0.93	—	ns
Fast Carry-in to Clock (FCIN to CLK at REGCOUT)	FCINRC_SET	2.29	—	1.76	—	1.28	—	1.02	—	ns
Carry-in to Clock (CIN to CLK at REGCOUT)	CINRC_SET	3.09	—	2.36	—	1.73	—	1.35	—	ns
Add/Subtract to Clock (ASWE to CLK at REGCOUT)	ASRC_SET	8.14	—	5.73	—	4.54	—	3.39	—	ns
Full Ripple Hold Times (T_J = all, V_{DD} = all):										
Fast Carry-in from Clock (FCIN from CLK at REGCOUT)	FCINRC_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
All Others	GENERIC_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Half Ripple Setup Times (nibble wide):										
Operands to Clock (Kz[1:0] to CLK)	HRIP_SET	3.91	—	2.81	—	2.21	—	1.66	—	ns
Bitwise Operands to Clock (Kz[1:0] to CLK at F[z])	HFRIP_SET	1.99	—	1.47	—	1.08	—	0.85	—	ns
Fast Carry-in to Clock (FCIN to CLK)	HFCIN_SET	2.55	—	1.87	—	1.34	—	1.04	—	ns
Carry-in to Clock (CIN to CLK)	HCIN_SET	3.80	—	2.79	—	1.97	—	1.56	—	ns
Add/Subtract to Clock (ASWE to CLK)	HAS_SET	8.82	—	6.18	—	4.68	—	3.50	—	ns
Operands to Clock (Kz[1:0] to CLK at REGCOUT)	HRIPRC_SET	3.03	—	2.31	—	1.68	—	1.32	—	ns
Fast Carry-in to Clock (FCIN to CLK at REGCOUT)	HFCINRC_SET	2.29	—	1.76	—	1.28	—	1.02	—	ns
Carry-in to Clock (CIN to CLK at REGCOUT)	HCINRC_SET	3.09	—	2.36	—	1.73	—	1.35	—	ns
Add/Subtract to Clock (ASWE to CLK at REGCOUT)	HASRC_SET	8.14	—	5.73	—	4.54	—	3.39	—	ns
Half Ripple Hold Times (T_J = all, V_{DD} = all):										
Fast Carry-in from Clock (HFCIN from CLK at REGCOUT)	HFCINRC_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
All Others	GENERIC_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns

Note: The table shows worst-case delay for the ripple chain. ispLEVER reports the delay for individual paths within the ripple chain that will be less than or equal to those listed above.

Timing Characteristics (continued)

Table 43. Ripple Mode PFU Timing Characteristics (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.

Parameter (T_J = +85 °C, V_{DD} = min)	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Full Ripple Delays (byte wide):										
Operands to Carry-out (Kz[1:0] to COUT)	RIPCO_DEL	—	5.32	—	4.11	—	2.98	—	2.32	ns
Operands to Carry-out (Kz[1:0] to FCOUT)	RIPFCO_DEL	—	5.30	—	4.10	—	2.98	—	2.32	ns
Operands to PFU Out (Kz[1:0] to F[7:0])	RIP_DEL	—	7.37	—	5.60	—	4.18	—	3.10	ns
Bitwise Operands to PFU Out (Kz[1:0] to F[z])	FRIP_DEL	—	2.34	—	1.80	—	1.32	—	1.05	ns
Fast Carry-in to Carry-out (FCIN to COUT)	FCINCO_DEL	—	2.59	—	1.99	—	1.43	—	1.14	ns
Fast Carry-in to Fast Carry-out (FCIN to FCOUT)	FCINFCO_DEL	—	2.57	—	1.98	—	1.41	—	1.13	ns
Carry-in to Carry-out (CIN to COUT)	CINCO_DEL	—	3.47	—	2.65	—	1.79	—	1.43	ns
Carry-in to Fast Carry-out (CIN to FCOUT)	CINFCO_DEL	—	3.46	—	2.64	—	1.78	—	1.43	ns
Fast Carry-in PFU Out (FCIN to F[7:0])	FCIN_DEL	—	6.03	—	4.55	—	3.21	—	2.51	ns
Carry-in PFU Out (CIN to F[7:0])	CIN_DEL	—	6.91	—	5.21	—	3.53	—	3.05	ns
Add/Subtract to Carry-out (ASWE to COUT)	ASCO_DEL	—	8.28	—	5.89	—	4.58	—	3.45	ns
Add/Subtract to Carry-out (ASWE to FCOUT)	ASFCO_DEL	—	8.11	—	5.78	—	4.48	—	3.38	ns
Add/Subtract to PFU Out (ASWE to F[7:0])	AS_DEL	—	10.66	—	7.55	—	5.85	—	4.38	ns
Half Ripple Delays (nibble wide):										
Operands to Carry-out (Kz[1:0] to COUT)	HRIPCO_DEL	—	5.32	—	4.11	—	2.98	—	2.32	ns
Operands to Fast Carry-out (Kz[1:0] to FCOUT)	HRIPFCO_DEL	—	5.30	—	4.10	—	2.98	—	2.32	ns
Operands to PFU Out (Kz[1:0] to F[3:0])	HRIP_DEL	—	5.50	—	4.07	—	3.20	—	2.40	ns
Bitwise Operands to PFU Out (Kz[1:0] to F[z])	HFRIP_DEL	—	2.34	—	1.80	—	1.32	—	1.05	ns
Fast Carry-in to Carry-out (FCIN to COUT)	HFCINCO_DEL	—	2.59	—	1.99	—	1.43	—	1.14	ns
Fast Carry-in to Fast Carry-out (FCIN to FCOUT)	HFCINFCO_DEL	—	2.57	—	1.98	—	1.41	—	1.13	ns
Carry-in to Carry-out (CIN to COUT)	HCINCO_DEL	—	3.47	—	2.65	—	1.79	—	1.43	ns
Carry-in to Carry-out (CIN to FCOUT)	HCINFCO_DEL	—	3.46	—	2.64	—	1.78	—	1.43	ns
Fast Carry-in PFU Out (FCIN to F[3:0])	HFCIN_DEL	—	3.76	—	2.84	—	2.01	—	1.58	ns
Carry-in PFU Out (CIN to F[3:0])	HCIN_DEL	—	4.65	—	3.50	—	2.33	—	2.12	ns
Add/Subtract to Carry-out (ASWE to COUT)	HASCO_DEL	—	8.28	—	5.89	—	4.58	—	3.45	ns
Add/Subtract to Carry-out (ASWE to FCOUT)	HASFCO_DEL	—	8.11	—	5.78	—	4.48	—	3.38	ns
Add/Subtract to PFU Out (ASWE to F[3:0])	HAS_DEL	—	9.12	—	6.49	—	4.86	—	3.69	ns

Note: The table shows worst-case delay for the ripple chain. ispLEVER reports the delay for individual paths within the ripple chain that will be less than or equal to those listed above.

Timing Characteristics (continued)

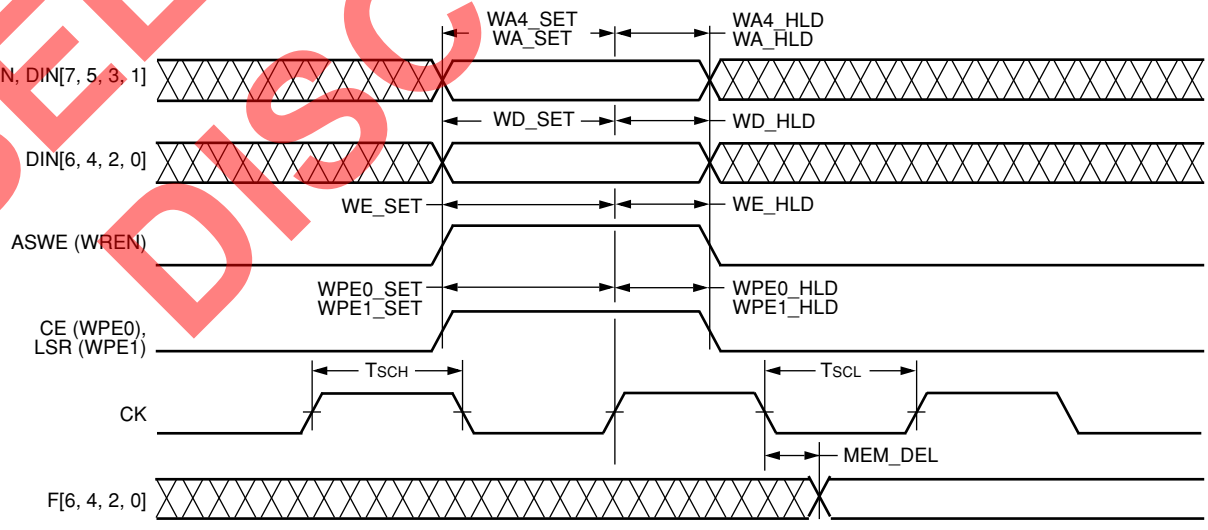
Table 44. Synchronous Memory Write Characteristics

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	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Write Operation for RAM Mode:										
Maximum Frequency	SMCLK_FRQ	—	151.00	—	197.00	—	254.00	—	315.00	MHz
Clock Low Time	SMCLKL_MPW	2.34	—	1.80	—	1.32	—	1.05		ns
Clock High Time	SMCLKH_MPW	3.79	—	2.77	—	2.13	—	1.62		ns
Clock to Data Valid (CLK to F[6, 4, 2, 0])*	MEM_DEL	—	10.00	—	7.14	—	5.00	—	4.08	ns
Write Operation Setup Time:										
Address to Clock (CIN to CLK)	WA4_SET	1.25	—	0.99	—	0.71	—	0.58	—	ns
Address to Clock (DIN[7, 5, 3, 1] to CLK)	WA_SET	0.72	—	0.52	—	0.35	—	0.28	—	ns
Data to Clock (DIN[6, 4, 2, 0] to CLK)	WD_SET	0.02	—	0.06	—	0.00	—	0.00	—	ns
Write Enable (WREN) to Clock (ASWE to CLK)	WE_SET	0.18	—	0.16	—	0.14	—	0.12	—	ns
Write-port Enable 0 (WPE0) to Clock (CE to CLK)	WPE0_SET	2.25	—	1.69	—	1.16	—	0.84	—	ns
Write-port Enable 1 (WPE1) to Clock (LSR to CLK)	WPE1_SET	2.79	—	2.13	—	1.58	—	1.31	—	ns
Write Operation Hold Time:										
Address from Clock (CIN from CLK)	WA4_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Address from Clock (DIN[7, 5, 3, 1] from CLK)	WA_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Data from Clock (DIN[6, 4, 2, 0] from CLK)	WD_HLD	0.59	—	0.42	—	0.40	—	0.32	—	ns
Write Enable (WREN) from Clock (ASWE from CLK)	WE_HLD	0.03	—	0.00	—	0.08	—	0.06	—	ns
Write-port Enable 0 (WPE0) from Clock (CE from CLK)	WPE0_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Write-port Enable 1 (WPE1) from Clock (LSR from CLK)	WPE1_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns

* The RAM is written on the inactive clock edge following the active edge that latches the address, data, and control signals.

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.



5-4621(F)

Figure 65. Synchronous Memory Write Characteristics

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.

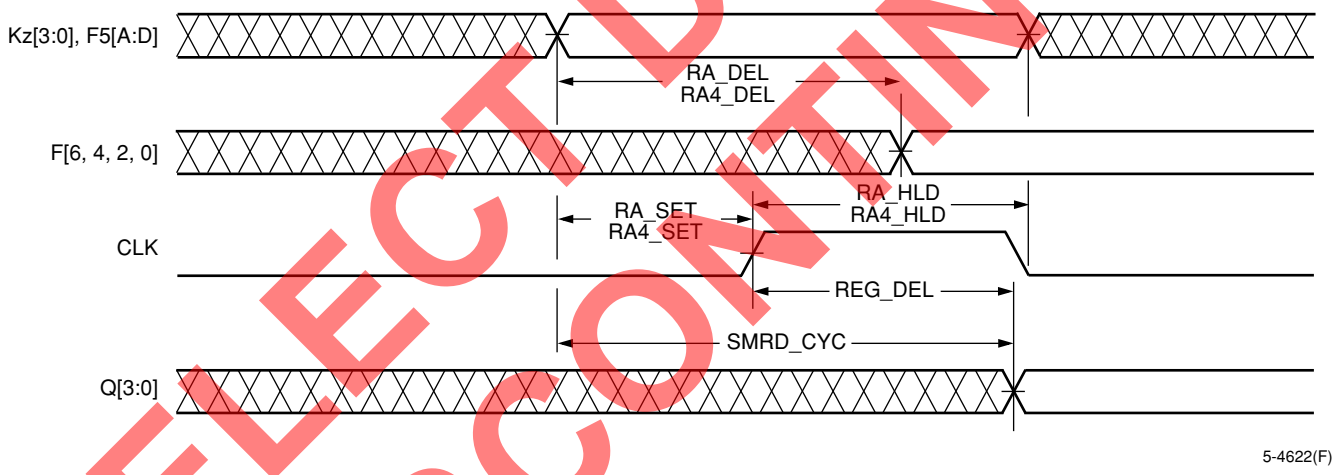


Figure 66. Synchronous Memory Read Cycle

Timing Characteristics (continued)

PLC Timing

Table 46. PFU Output MUX and Direct Routing 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 (TJ = 85 °C, VDD = min)	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
PFU Output MUX (Fan-out = 1)										
Output MUX Delay (F[7:0]/Q[7:0] to O[9:0])	OMUX_DEL	—	0.50	—	0.39	—	0.35	—	0.28	ns
Carry-out MUX Delay (COUT to O9)	COO9_DEL	—	0.34	—	0.26	—	0.24	—	0.18	ns
Registered Carry-out MUX Delay (REGCOUT to O8)	RCOO8_DEL	—	0.34	—	0.26	—	0.24	—	0.18	ns
Direct Routing										
PFU Feedback (xSW)*	FDBK_DEL	—	1.74	—	1.41	—	1.48	—	1.14	ns
PFU to Orthogonal PFU Delay (xSW to xSW)	ODIR_DEL	—	2.21	—	1.77	—	1.75	—	1.39	ns
PFU to Diagonal PFU Delay (xBID to xSW)	DDIR_DEL	—	2.69	—	2.19	—	2.53	—	1.98	ns

* This is general feedback using switching segments. See the combinatorial PFU timing table for softwired look-up table feedback timing.

SLIC Timing

Table 47. Supplemental Logic and Interconnect Cell (SLIC) 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 (TJ = 85 °C, VDD = min)	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
3-Statable BIDIs										
BIDI Delay (BRx to BLx, BLx to BRx)	BUF_DEL	—	0.84	—	0.70	—	0.94	—	0.77	ns
BIDI Delay (Ox to BRx, Ox to BLx)	OBUF_DEL	—	0.72	—	0.61	—	0.87	—	0.70	ns
BIDI 3-state Enable/Disable Delay (TRI to BL, BR)	TRI_DEL	—	2.55	—	1.90	—	1.31	—	1.01	ns
BIDI 3-state Enable/Disable Delay (BL, BR via DEC, TRI to BL, BR)	DECTRI_DEL	—	3.59	—	2.65	—	1.91	—	1.48	ns
Decoder										
Decoder Delay (BR[9:8], BL[9:8] to DEC)	DEC98_DEL	—	2.39	—	1.85	—	1.27	—	1.02	ns
Decoder Delay (BR[7:0], BL[7:0] to DEC)	DEC_DEL	—	2.35	—	1.82	—	1.23	—	0.99	ns

Timing Characteristics (continued)

PIO Timing

Table 48. Programmable I/O (PIO) 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 Delays (T_J = 85 °C, V_{DD} = min)										
Input Rise Time	IN_RIS	—	500	—	500	—	500	—	500	ns
Input Fall Time	IN_FAL	—	500	—	500	—	500	—	500	ns
PIO Direct Delays:										
Pad to In (pad to CLK IN)	CKIN_DEL	—	1.41	—	1.26	—	0.64	—	0.41	ns
Pad to In (pad to IN1, IN2)	IN_DEL	—	2.16	—	1.87	—	1.28	—	0.90	ns
Pad to In Delayed (pad to IN1, IN2)	IND_DEL	—	9.05	—	7.83	—	6.64	—	7.27	ns
PIO Transparent Latch Delays:										
Pad to In (pad to IN1, IN2)	LATCH_DEL	—	4.11	—	3.25	—	2.52	—	1.82	ns
Pad to In Delayed (pad to IN1, IN2)	LATCHD_DEL	—	10.58	—	9.05	—	7.67	—	7.65	ns
Input Latch/FF Setup Timing:										
Pad to ExpressCLK (fast-capture latch/FF)	INREGE_SET	5.93	—	4.82	—	3.63	—	3.23	—	ns
Pad Delayed to ExpressCLK (fast-capture latch/FF)	INREGED_SET	12.86	—	11.03	—	9.18	—	9.68	—	ns
Pad to Clock (input latch/FF)	INREG_SET	1.62	—	1.42	—	0.71	—	0.50	—	ns
Pad Delayed to Clock (input latch/FF)	INREGD_SET	8.57	—	7.36	—	5.91	—	7.06	—	ns
Clock Enable to Clock (CE to CLK)	INCE_SET	2.03	—	1.64	—	1.29	—	1.00	—	ns
Local Set/Reset (sync) to Clock (LSR to CLK)	INLSR_SET	1.79	—	1.45	—	1.14	—	0.89	—	ns
Input FF/Latch Hold Timing:										
Pad from ExpressCLK (fast-capture latch/FF)	INREGE_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Pad Delayed from ExpressCLK (fast-capture latch/FF)	INREGED_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Pad from Clock (input latch/FF)	INREG_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Pad Delayed from Clock (input latch/FF)	INREGD_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Clock Enable from Clock (CE from CLK)	INCE_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Local Set/Reset (sync) from Clock (LSR from CLK)	INLSR_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Clock-to-in Delay (FF CLK to IN1, IN2)	INREG_DEL	—	4.05	—	3.14	—	2.53	—	2.05	ns
Clock-to-in Delay (latch CLK to IN1, IN2)	INLTCH_DEL	—	4.08	—	3.19	—	2.62	—	2.14	ns
Local S/R (async) to IN (LSR to IN1, IN2)	INLSR_DEL	—	6.11	—	4.76	—	3.81	—	3.17	ns
Local S/R (async) to IN (LSR to IN1, IN2) LatchFF in Latch Mode	INLSRL_DEL	—	5.89	—	4.66	—	3.57	—	2.98	ns
Global S/R to In (GSRN to IN1, IN2)	INGSR_DEL	—	5.38	—	4.22	—	3.44	—	2.88	ns

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

Timing Characteristics (continued)

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

Parameter	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Output Delays (T_J = 85 °C, V_{DD} = min, C_L = 50 pF)										
Output to Pad (OUT2, OUT1 direct to pad):										
Fast	OUTF_DEL	—	5.09	—	4.21	—	2.63	—	2.17	ns
Slewlim	OUTSL_DEL	—	7.86	—	6.49	—	3.49	—	2.91	ns
Sinklim	OUTSI_DEL	—	9.41	—	7.98	—	8.08	—	7.32	ns
3-state Enable/Disable Delay (TS 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	OUTLSRF_DEL	—	9.03	—	7.25	—	4.96	—	3.94	ns
Slewlim	OUTLSRSL_DEL	—	11.79	—	9.53	—	5.82	—	4.67	ns
Sinklim	OUTLSRSI_DEL	—	13.35	—	11.02	—	10.38	—	9.10	ns
Global Set/Reset to Pad (GSRN to pad):										
Fast	OUTGSRF_DEL	—	8.30	—	6.69	—	4.39	—	3.46	ns
Slewlim	OUTGSRSL_DEL	—	11.06	—	8.97	—	5.07	—	3.99	ns
Sinklim	OUTGSRSI_DEL	—	12.62	—	10.46	—	10.02	—	8.81	ns
Output FF Setup Timing:										
Out to ExpressCLK (OUT[2:1] to ECLK)	OUTE_SET	0.00	—	0.00	—	0.00	—	0.00	—	ns
Out to Clock (OUT[2:1] to CLK)	OUT_SET	0.00	—	0.00	—	0.00	—	0.00	—	ns
Clock Enable to Clock (CE to CLK)	OUTCE_SET	0.91	—	0.67	—	0.56	—	0.45	—	ns
Local Set/Reset (sync) to Clock (LSR to CLK)	OUTLSR_SET	0.41	—	0.32	—	0.26	—	0.24	—	ns
Output FF Hold Timing:										
Out from ExpressCLK (OUT[2:1] from ECLK)	OUTE_HLD	0.73	—	0.58	—	0.36	—	0.29	—	ns
Out from Clock (OUT[2:1] from CLK)	OUT_HLD	0.73	—	0.58	—	0.36	—	0.29	—	ns
Clock Enable from Clock (CE from CLK)	OUTCE_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Local Set/Reset (sync) from Clock (LSR from CLK)	OUTLSR_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Clock to Pad Delay (ECLK, SCLK to pad):										
Fast	OUTREGF_DEL	—	6.71	—	5.44	—	3.56	—	2.78	ns
Slewlim	OUTREGSL_DEL	—	9.47	—	7.71	—	4.42	—	3.52	ns
Sinklim	OUTREGSI_DEL	—	11.03	—	9.20	—	8.98	—	7.94	ns
Additional Delay If Using Open Drain	OD_DEL	—	0.20	—	0.16	—	0.10	—	0.08	ns

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

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)

Special Function Blocks Timing

Table 49. Microprocessor Interface (MPI) 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	
***PowerPC* Interface Timing** (T_J = 85 °C, V_{DD} = min)										
Transfer Acknowledge Delay (CLK to $\overline{TA}$)	TA_DEL	—	11.6	—	9.3	—	8.0	—	6.8	ns
Burst Inhibit Delay (CLK to BIN)	BI_DEL	—	11.6	—	9.3	—	8.0	—	6.8	ns
Transfer Acknowledge Delay to High Impedance	TA_DELZ	—	(2)	—	(2)	—	(2)	—	(2)	ns
Burst Inhibit Delay to High Impedance	BI_DELZ	—	(2)	—	(2)	—	(2)	—	(2)	ns
Write Data Setup Time (data to $\overline{TS}$)	WD_SET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Write Data Hold Time (data from CLK while $\overline{MPI\_ACK}$ low)	WD_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Address Setup Time (addr to $\overline{TS}$)	A_SET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Address Hold Time (addr from CLK while $\overline{MPI\_ACK}$ low)	A_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Read/Write Setup Time (R/W to $\overline{TS}$)	RW_SET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Read/Write Hold Time (R/W from CLK while $\overline{MPI\_ACK}$ low)	RW_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Chip Select Setup Time ($\overline{CS0}$, CS1 to $\overline{TS}$)	CS_SET	0.3	—	.25	—	.14	—	.12	—	ns
Chip Select Hold Time ($\overline{CS0}$, CS1 from CLK)	CS_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
User Address Delay (pad to UA[3:0])	UA_DEL	—	3.3	—	2.6	—	2.3	—	1.9	ns
User Read/Write Delay (pad to URDWR_DEL)	URDWR_DEL	—	7.0	—	5.4	—	4.2	—	3.6	ns
***i960* Interface Timing** (T_J = 85 °C, V_{DD} = min)										
Addr/Data Select to ALE ($\overline{ADS}$, to ALE low)	ADSN_SET	2.0	—	1.8	—	1.6	—	1.4	—	ns
Addr/Data Select to ALE ($\overline{ADS}$, from ALE low)	ADSN_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Ready/Receive Delay (CLK to $\overline{RDYRCV}$)	RDYRCV_DEL	—	11.6	—	9.3	—	8.0	—	6.8	ns
Ready/Receive Delay to High Impedance	RDYRCV_DELZ	—	(2)	—	(2)	—	(2)	—	(2)	ns
Write Data Setup Time	WD_SET	(3)	—	(3)	—	(3)	—	(3)	—	ns
Write Data Hold Time	WD_HLD	(4)	—	(4)	—	(4)	—	(4)	—	ns
Address Setup Time (addr to ALE low)	A_SET	2.0	—	1.8	—	0.50	—	—	0.42	ns
Address Hold Time (addr from ALE low)	A_HLD	2.0	—	1.8	—	0.51	—	—	0.44	ns
Byte Enable Setup Time ($\overline{BE0}$, $\overline{BE1}$ to ALE low)	BE_SET	2.0	—	1.8	—	0.50	—	—	0.42	ns
Byte Enable Hold Time ($\overline{BE0}$, $\overline{BE1}$ from ALE low)	BE_HLD	2.0	—	1.8	—	0.51	—	—	0.44	ns
Read/Write Setup Time	RW_SET	(3)	—	(3)	—	(3)	—	(3)	—	ns
Read/Write Hold Time	RW_HLD	(4)	—	(4)	—	(4)	—	(4)	—	ns
Chip Select Setup Time ($\overline{CS0}$, CS1 to CLK)[1]	CS_SET	2.0	—	1.8	—	0.45	—	—	0.38	ns
Chip Select Hold Time ($\overline{CS0}$, CS1 from CLK)[1]	CS_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
User Address Delay (CLK low to UA[3:0])	UA_DEL	—	6.6	—	4.3	—	4.1	—	3.5	ns
User Read/Write Delay (pad to URDWR_DEL)	URDWR_DEL	—	7.0	—	5.4	—	4.2	—	3.6	ns

1. For user system flexibility, $\overline{CS0}$ and CS1 may be set up to any one of the three rising clock edges, beginning with the rising clock edge when $\overline{MPI\_STRB}$ is low. If both chip selects are valid and the setup time is met, the MPI will latch the chip select state, and $\overline{CS0}$ and CS1 may go inactive before the end of the read/write cycle.
2. 0.5 MPI_CLK.
3. Write data and W/$\overline{R}$ have to be valid starting from the clock cycle after both $\overline{ADS}$ and $\overline{CS0}$ and CS1 are recognized.
4. Write data and W/$\overline{R}$ have to be held until the microprocessor receives a valid $\overline{RDYRCV}$.

Notes:
Read and write descriptions are referenced to the host microprocessor; e.g., a read is a read by the host (*PowerPC*, *i960*) from the FPGA.

PowerPC and *i960* timings to/from the clock are relative to the clock at the FPGA microprocessor interface clock pin (MPI_CLK).

Timing Characteristics (continued)

Table 49. Microprocessor Interface (MPI) Timing Characteristics (continued)

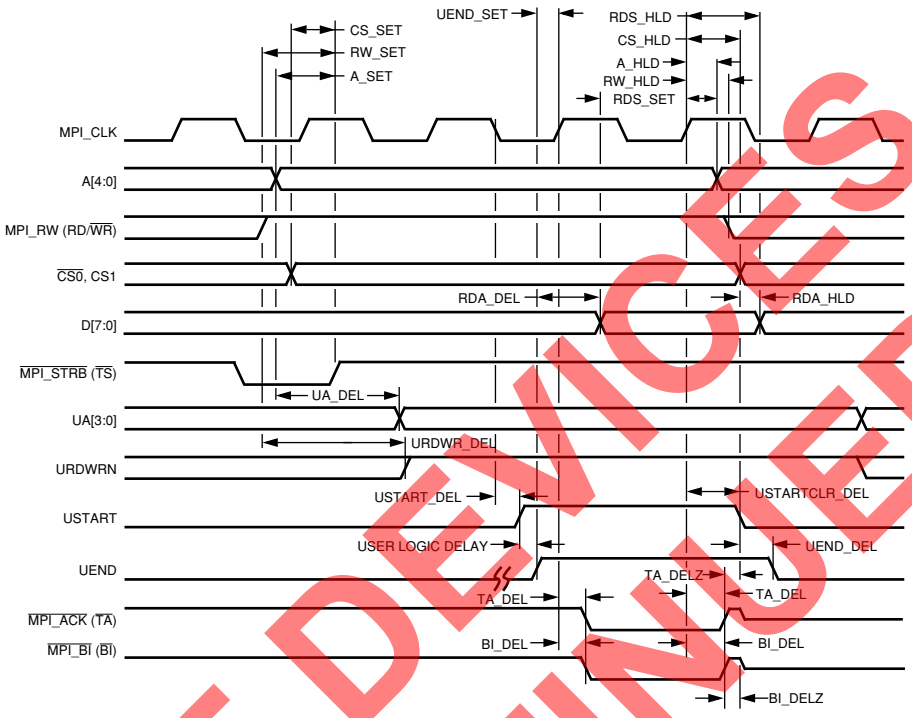
OR3Cxx Commercial: V_{DD} = 5.0 V ± 5%, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C < T_A < +85 °C.
OR3Txxx Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C < T_A < +85 °C.

Parameter	Symbol	Speed								Unit
		–4		–5		–6		–7		
		Min	Max	Min	Max	Min	Max	Min	Max	
User Logic Delay[5]	User Logic Delay	—	—	—	—	—	—	—	—	ns
User Start Delay (MPI_CLK falling to USTART)[6]	USTART_DEL	—	3.6	—	3.4	—	3.3	—	2.8	ns
User Start Clear Delay (MPI_CLK to USTART)	USTARTCLR_DEL	—	7.5	—	7.3	—	7.1	—	6.0	ns
User End Delay (USTART low to UEND low)[7]	UEND_DEL	—	—	—	—	—	—	—	—	ns
Synchronous User Timing:										
User End Setup (UEND to MPI_CLK)	UEND_SET	0.00	—	0.00	—	0.00	—	0.00	—	ns
User End Hold (UEND to MPI_CLK)	UEND_HLD	1.0	—	0.95	—	0.88	—	0.75	—	ns
Data Setup for Read (D[7:0] to MPI_CLK)[9]	RDS_SET	—	—	—	—	—	—	—	—	ns
Data Hold for Read (D[7:0] from MPI_CLK)[9]	RDS_HLD	—	—	—	—	—	—	—	—	ns
Asynchronous User Timing:										
User End to Read Data Delay (UEND to D[7:0])[10]	RDA_DEL	—	—	—	—	—	—	—	—	ns
Data Hold from User Start (low)[9]	RDA_HLD	—	—	—	—	—	—	—	—	ns
Interrupt Request Pulse Width[8]	TUIRQ_PW	—	—	—	—	—	—	—	—	ns

1. For user system flexibility, $\overline{CS0}$ and CS1 may be set up to any one of the three rising clock edges, beginning with the rising clock edge when $\overline{MPI\_STRB}$ is low. If both chip selects are valid and the setup time is met, the MPI will latch the chip select state, and $\overline{CS0}$ and CS1 may go inactive before the end of the read/write cycle.
2. 0.5 MPI_CLK.
3. Write data and W/$\overline{R}$ have to be valid starting from the clock cycle after both $\overline{ADS}$ and $\overline{CS0}$ and CS1 are recognized.
4. Write data and W/$\overline{R}$ have to be held until the microprocessor receives a valid $\overline{RDYRCV}$.
5. User Logic Delay has no predefined value. The user must generate a UEND signal to complete the cycle.
6. USTART_DEL is based on the falling clock edge.
7. There is no specific time associated with this delay. The user must assert UEND low to complete this cycle.
8. The user must assert interrupt request low until a service routine is executed.
9. This should be at least one MPI_CLK cycle.
10. User should set up read data so that RDS_SET and RDS_HLD can be met for the microprocessor timing.

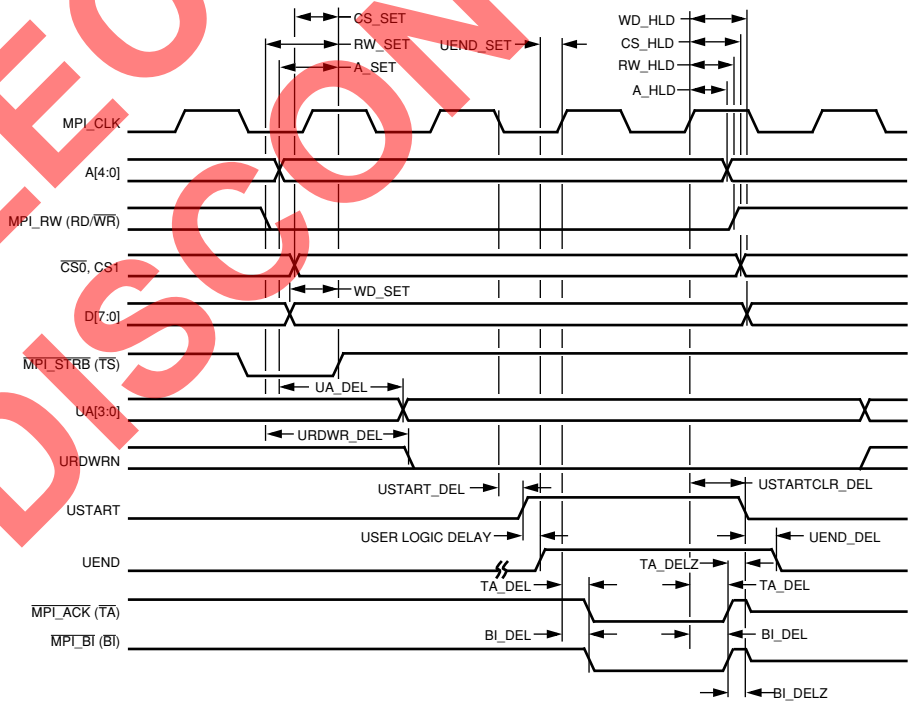
Notes:
Read and write descriptions are referenced to the host microprocessor; e.g., a read is a read by the host (*PowerPC*, *i960*) from the FPGA. *PowerPC* and *i960* timings to/from the clock are relative to the clock at the FPGA microprocessor interface clock pin (MPI_CLK).

Timing Characteristics (continued)



5-5832(F)

Figure 67. MPI *PowerPC* User Space Read Timing



5-5840(F)

Figure 68. MPI *PowerPC* User Space Write Timing

Timing Characteristics (continued)

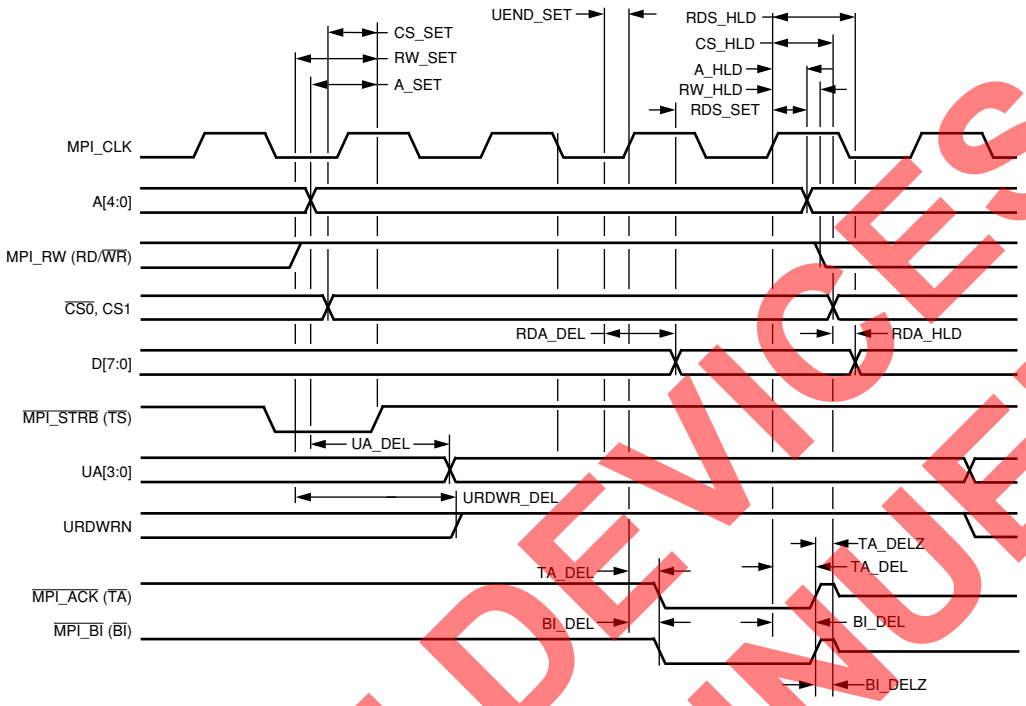


Figure 69. MPI *PowerPC* Internal Read Timing

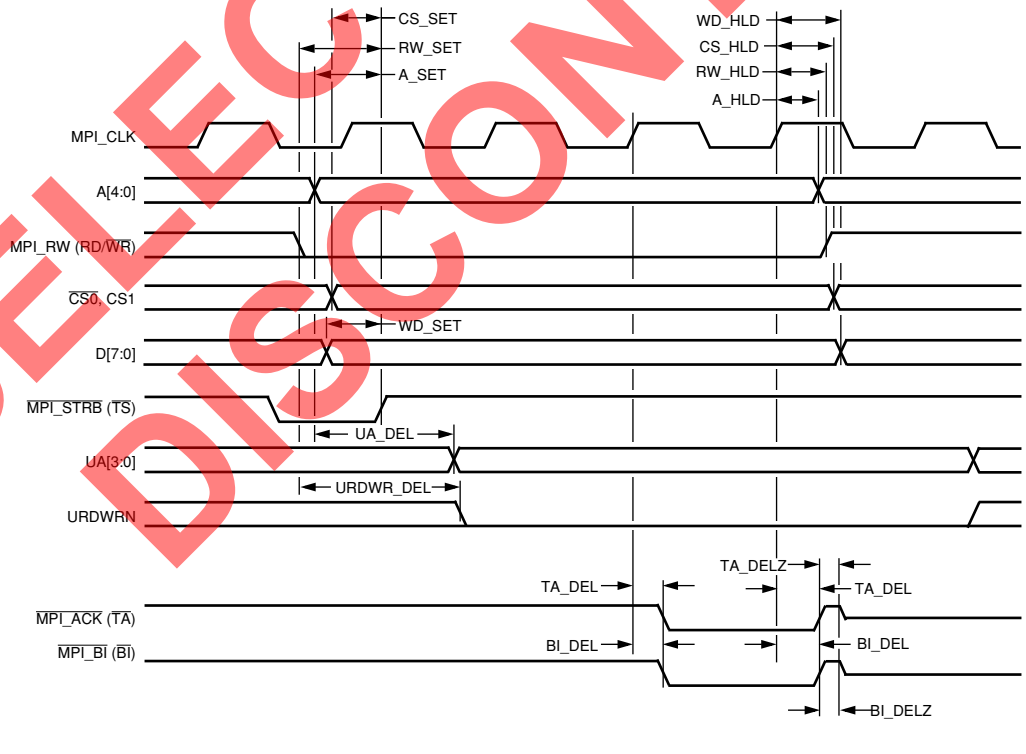
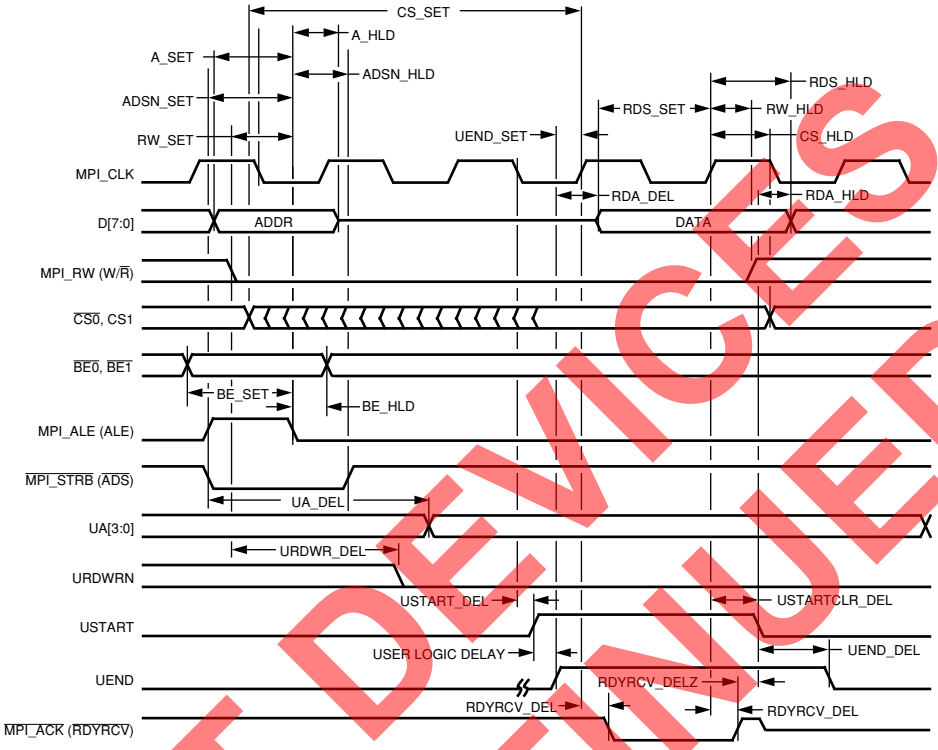


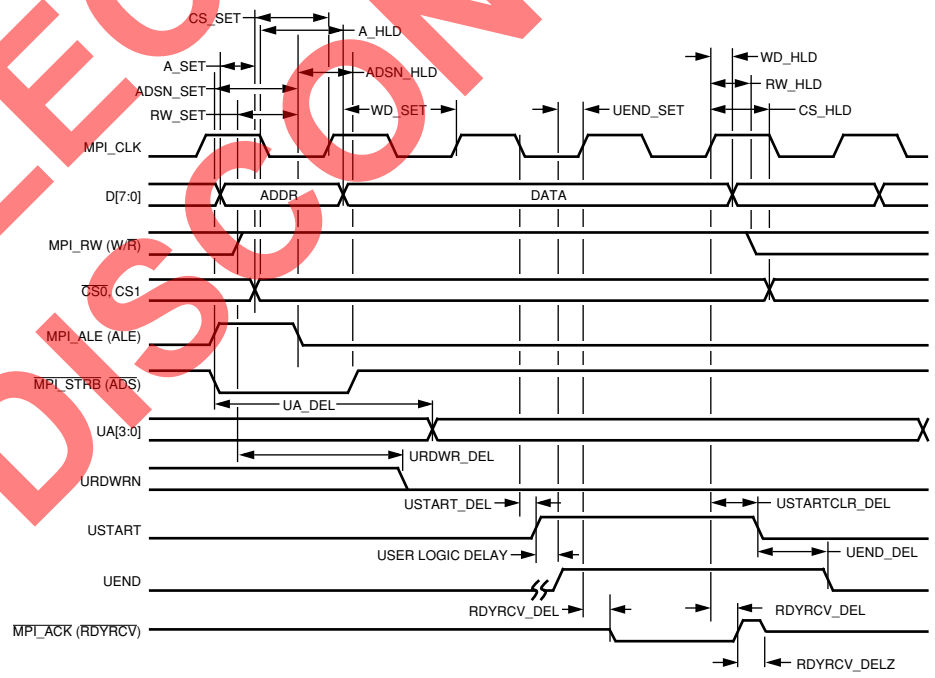
Figure 70. MPI *PowerPC* Internal Write Timing

Timing Characteristics (continued)



5-5831(F).b

Figure 71. MPI *i960* User Space Read Timing



5-5830(F).b

Figure 72. MPI *i960* User Space Write Timing

Timing Characteristics (continued)

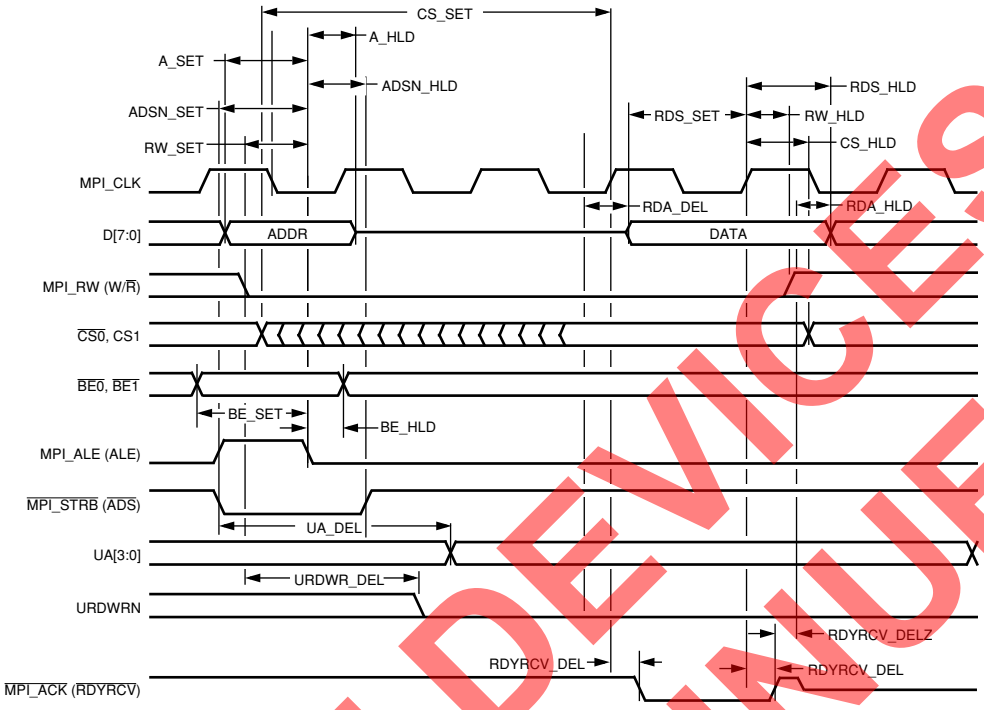


Figure 73. MPI *i960* Internal Read Timing

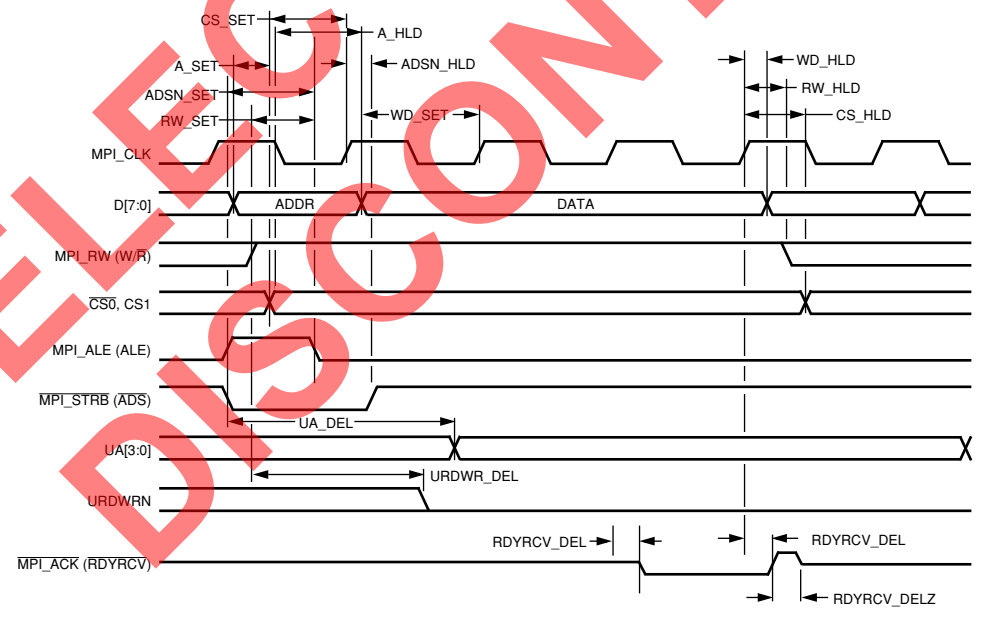


Figure 74. MPI *i960* Internal Write Timing

Timing Characteristics (continued)

Table 50. Programmable Clock Manager (PCM) 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	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Input Clock Frequency:	FPCMI									
OR3Cxx		5	133	5	133	—	—	—	—	MHz
OR3Txxx		—	—	5	133	5	133	5	133	MHz
Output Clock Frequency:	FPCMO									
OR3Cxx		5	135	5	135	—	—	—	—	MHz
OR3Txxx		—	—	5	100	5	100	5	100	MHz
Input Clock Duty Cycle	PCMI_DUTY	30.00	70.00	30.00	70.00	30.00	70.00	30.00	70.00	%
Output Clock Duty Cycle	PCMO_DUTY	3.13	96.90	3.13	96.90	3.13	96.90	3.13	96.90	%
Input Frequency Tolerance*	FTOL	—	26400	—	26400	—	26400	—	26400	ppm
PCM Acquisition Time (CLK In to LOCK)	PCM_ACQ†	36	100	36	100	36	100	36	100	µs
PCM Off Delay (config. Done-L, WE to PCM power off)	PCMOFF_DEL	—	100.0	—	100.0	—	100.0	—	100.0	ns
PCM Delay in DLL Mode (propagation delay)	PCMDLL-DEL	—	1.95	—	1.82	—	1.63	—	1.50	ns
PCM Delay in PLL Mode (propagation delay)	PCMPLL_DEL	—	0.00	—	0.00	—	0.00	—	0.00	ns
PCM Clock In to PCM Clock Out (CLK In to ECLK)‡	PCMBYE_DEL	—	0.47	—	0.36	—	0.26	—	0.24	ns
PCM Clock In to PCM Clock Out (CLK In to SCLK)‡	PCMBYS_DEL	—	0.47	—	0.36	—	0.26	—	0.24	ns
Routed Clock-in Delay (routing to PCM phase detect, using DIV0)	RTCKD_DEL	—	1.30	—	1.10	—	0.90	—	TBD	ns
System Clock-out Delay (PCM oscillator to SCLK output at PCM)	PCMSCK_DEL	—	2.70	—	2.20	—	1.90	—	TBD	ns

Parameter	Symbol	fOUT (MHz)	PLL Mode	DLL Mode	Unit
Output Jitter	OUTJIT	5—20	250	200	ps
		21—30	210	170	ps
		31—40	180	145	ps
		41—50	155	123	ps
		51—60	130	105	ps
		61—70	110	90	ps
		71—80	95	75	ps
		81—90	80	65	ps
		91—100	70	55	ps

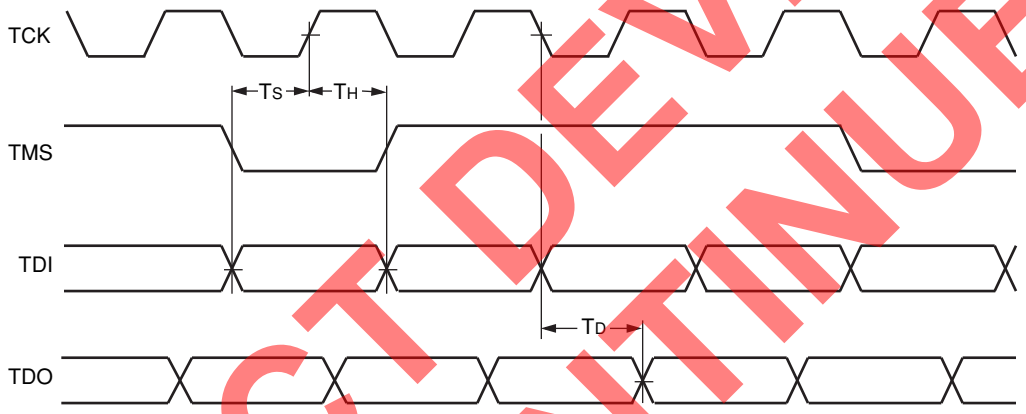
* Input frequency tolerance is the allowed input clock frequency change in parts per million.
† See Table 29 and Table 30 for acquisition times for individual frequencies.
‡ PLL mode, divider reg = 1111111 (input freq. = output freq.).

Timing Characteristics (continued)

Table 51. Boundary-Scan 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	Min	Max	Unit
TDI/TMS to TCK Setup Time	T_S	25.0	—	ns
TDI/TMS Hold Time from TCK	T_H	0.0	—	ns
TCK Low Time	T_{CL}	50.0	—	ns
TCK High Time	T_{CH}	50.0	—	ns
TCK to TDO Delay	T_D	—	20.0	ns
TCK Frequency	T_{TCK}	—	10.0	MHz



5-6764(F)

Figure 75. Boundary-Scan Timing Diagram

Timing Characteristics (continued)

Clock Timing

Table 52. ExpressCLK (ECLK) and Fast Clock (FCLK) 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.

Device (T_J = 85 °C, V_{DD} = min)	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Clock Control Timing Delay Through CLKCNTRL (input from corner)	ECLKC_DEL	0.31	—	0.31	—	0.31	—	0.31	—	ns
Delay Through CLKCNTRL (input from internal clock controller PAD)	ECLKM_DEL	1.54	—	1.17	—	1.00	—	0.92	—	ns
Clock Shutoff Timing:										
Setup from Middle ECLK (shut off to CLK)	OFFM_SET	0.77	—	0.51	—	0.44	—	0.41	—	ns
Hold from Middle ECLK (shut off from CLK)	OFFM_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
Setup from Corner ECLK (shut off to CLK)	OFFC_SET	0.77	—	0.51	—	0.44	—	0.41	—	ns
Hold from Corner ECLK (shut off from CLK)	OFFC_HLD	0.00	—	0.00	—	0.00	—	0.00	—	ns
ECLK Delay (middle pad):	ECLKM_DEL									
OR3T20		—	—	—	2.56	—	2.05	—	1.78	ns
OR3T30		—	—	—	2.62	—	2.08	—	1.80	ns
OR3T55		—	3.50	—	2.74	—	2.13	—	1.85	ns
OR3C/T80		—	3.67	—	2.86	—	2.19	—	1.90	ns
OR3T125		—	—	—	3.06	—	2.29	—	1.98	ns
ECLK Delay (corner pad):	ECLKC_DEL									
OR3T20		—	—	—	4.48	—	3.85	—	3.36	ns
OR3T30		—	—	—	4.53	—	3.97	—	3.47	ns
OR3T55		—	5.47	—	4.64	—	4.22	—	3.69	ns
OR3C/T80		—	5.64	—	4.77	—	4.47	—	3.92	ns
OR3T125		—	—	—	4.96	—	4.85	—	4.27	ns
FCLK Delay (middle pad):	FCLKM_DEL									
OR3T20		—	—	—	5.91	—	4.59	—	3.81	ns
OR3T30		—	—	—	6.12	—	4.66	—	3.89	ns
OR3T55		—	8.24	—	6.59	—	4.83	—	4.06	ns
OR3C/T80		—	8.87	—	7.11	—	5.01	—	4.26	ns
OR3T125		—	—	—	7.98	—	5.33	—	4.59	ns
FCLK Delay (corner pad):	FCLKC_DEL									
OR3T20		—	—	—	7.88	—	6.41	—	5.40	ns
OR3T30		—	—	—	8.11	—	6.58	—	5.58	ns
OR3T55		—	10.34	—	8.60	—	6.95	—	5.94	ns
OR3C/T80		—	11.01	—	9.15	—	7.34	—	6.33	ns
OR3T125		—	—	—	10.07	—	7.96	—	6.94	ns

Notes:
The ECLK delays are to all of the PICs 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 PIC clock input.

The FCLK delays are for a fully routed clock tree that uses the ExpressCLK input into the fast clock network. It includes both the input buffer delay and the clock routing to the PFU CLK input. The delay will be reduced if any of the clock branches are not used.

Timing Characteristics (continued)

Table 53. General-Purpose Clock Timing Characteristics (Internally Generated Clock)

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.

Device (T_J = 85 °C, V_{DD} = min)	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
OR3T20	CLK_DEL	—	—	—	4.22	—	3.46	—	2.84	ns
OR3T30	CLK_DEL	—	—	—	4.29	—	3.48	—	2.87	ns
OR3T55	CLK_DEL	—	5.34	—	4.41	—	3.53	—	2.93	ns
OR3C/T80	CLK_DEL	—	5.49	—	4.52	—	3.57	—	2.98	ns
OR3T125	CLK_DEL	—	—	—	4.80	—	3.71	—	3.13	ns

Notes:
This table represents the delay for an internally generated clock from the clock tree input in one of the four middle PICs (using pSW routing) on any side of the device which is then distributed to the PFU/PIO clock inputs. If the clock tree input used is located at any other PIC, see the results reported by ispLEVER.

This clock delay is for a fully routed clock tree that uses the general clock network. The delay will be reduced if any of the clock branches are not used. See pin-to-pin timing in Table 56 for clock delays of clocks input on general I/O pins.

Timing Characteristics (continued)

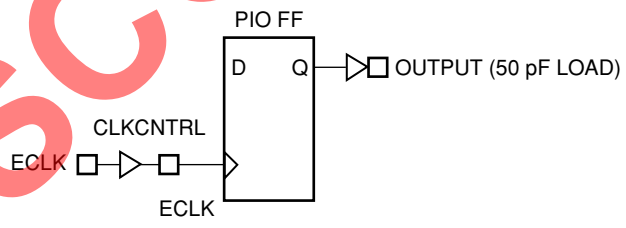
Table 54. OR3Cxx ExpressCLK to Output Delay (Pin-to-Pin)

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; C_L = 50 pF.
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; C_L = 50 pF.

Description (T_J = 85 °C, V_{DD} = min)	Device	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
ECLK Middle Input Pin→OUTPUT Pin (Fast)	OR3T20	—	—	—	7.78	—	5.40	—	4.38	ns
	OR3T30	—	—	—	7.84	—	5.43	—	4.40	ns
	OR3T55	—	9.93	—	7.96	—	5.48	—	4.44	ns
	OR3C/T80	—	10.10	—	8.08	—	5.54	—	4.49	ns
	OR3T125	—	—	—	8.28	—	5.64	—	4.58	ns
ECLK Middle Input Pin→OUTPUT Pin (Slewlim)	OR3T20	—	—	—	9.77	—	6.07	—	4.91	ns
	OR3T30	—	—	—	9.83	—	6.10	—	4.93	ns
	OR3T55	—	12.37	—	9.95	—	6.15	—	4.97	ns
	OR3C/T80	—	12.54	—	10.07	—	6.21	—	5.02	ns
	OR3T125	—	—	—	10.27	—	6.31	—	5.11	ns
ECLK Middle Input Pin→OUTPUT Pin (Sinklim)	OR3T20	—	—	—	11.12	—	10.92	—	9.65	ns
	OR3T30	—	—	—	11.18	—	10.95	—	9.67	ns
	OR3T55	—	13.73	—	11.30	—	11.00	—	9.71	ns
	OR3C/T80	—	13.90	—	11.42	—	11.06	—	9.76	ns
	OR3T125	—	—	—	11.62	—	11.16	—	9.85	ns
Additional Delay if ECLK Corner Pin Used	OR3T20	—	—	—	1.91	—	1.80	—	1.58	ns
	OR3T30	—	—	—	1.91	—	1.90	—	1.67	ns
	OR3T55	—	1.97	—	1.91	—	2.09	—	1.84	ns
	OR3C/T80	—	1.97	—	1.91	—	2.28	—	2.02	ns
	OR3T125	—	—	—	1.90	—	2.57	—	2.29	ns

Notes:
Timing is without the use of the programmable clock manager (PCM).

This clock delay is for a fully routed clock tree that uses the ExpressCLK network. It includes both the input buffer delay, the clock routing to the PIO CLK input, the clock→Q of the FF, and the delay through the output buffer. The given timing requires that the input clock pin be located at one of the six ExpressCLK inputs of the device, and that a PIO FF be used.



5-4846(F).a

Figure 76. ExpressCLK to Output Delay

Timing Characteristics (continued)

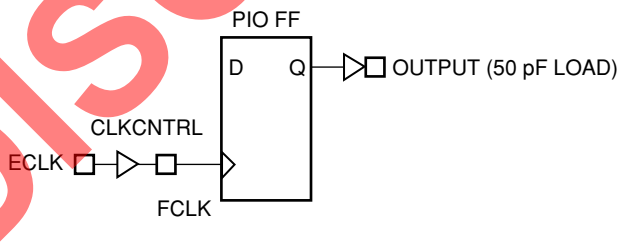
Table 55. OR3Cxx Fast Clock (FCLK) to Output Delay (Pin-to-Pin)

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; C_L = 50 pF.
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; C_L = 50 pF.

Description (T_J = 85 °C, V_{DD} = min)	Device	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Output Not on Same Side of Device As Input Clock (Fast Clock Delays Using ExpressCLK Inputs)										
ECLK Middle Input Pin →OUTPUT Pin (Fast)	OR3T20	—	—	—	11.13	—	7.94	—	6.40	ns
	OR3T30	—	—	—	11.35	—	8.01	—	6.48	ns
	OR3T55	—	14.68	—	11.81	—	8.18	—	6.66	ns
	OR3C/T80	—	15.30	—	12.33	—	8.36	—	6.85	ns
	OR3T125	—	—	—	13.20	—	8.68	—	7.19	ns
ECLK Middle Input Pin →OUTPUT Pin (Slewlim)	OR3T20	—	—	—	13.12	—	8.61	—	6.93	ns
	OR3T30	—	—	—	13.33	—	8.68	—	7.01	ns
	OR3T55	—	17.11	—	13.80	—	8.85	—	7.19	ns
	OR3C/T80	—	17.74	—	14.32	—	9.04	—	7.38	ns
	OR3T125	—	—	—	15.19	—	9.35	—	7.72	ns
ECLK Middle Input Pin →OUTPUT Pin (Sinklim)	OR3T20	—	—	—	14.47	—	13.46	—	11.67	ns
	OR3T30	—	—	—	14.68	—	13.53	—	11.75	ns
	OR3T55	—	18.47	—	15.15	—	13.70	—	11.93	ns
	OR3C/T80	—	19.10	—	15.67	—	13.88	—	12.12	ns
	OR3T125	—	—	—	16.54	—	14.20	—	12.46	ns
Additional Delay if ECLK Corner Pin Used	OR3T20	—	—	—	1.97	—	1.82	—	1.60	ns
	OR3T30	—	—	—	1.99	—	1.92	—	1.69	ns
	OR3T55	—	2.10	—	2.01	—	2.12	—	1.88	ns
	OR3C/T80	—	2.14	—	2.04	—	2.33	—	2.07	ns
	OR3T125	—	—	—	2.09	—	2.63	—	2.39	ns

Notes:
Timing is without the use of the programmable clock manager (PCM).

This clock delay is for a fully routed clock tree that uses the primary clock network. It includes both the input buffer delay, the clock→Q of the FF, and the delay through the output buffer. The delay will be reduced if any of the clock branches are not used. The given timing requires that the input clock pin be located at one of the six ExpressCLK inputs of the device and that a PIO FF be used.



5-4846(F).b

Figure 77. Fast Clock to Output Delay

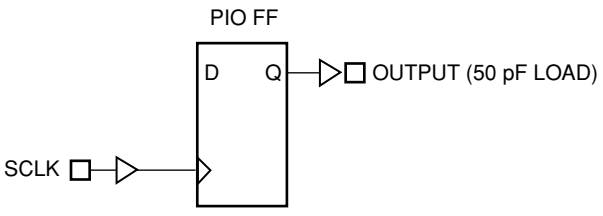
Timing Characteristics (continued)

Table 56. OR3Cxx General System Clock (SCLK) to Output Delay (Pin-to-Pin)

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; C_L = 50 pF.
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; C_L = 50 pF.

Description (T_J = 85 °C, V_{DD} = min)	Device	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
Output On Same Side of Device As Input Clock (System Clock Delays Using General User I/O Inputs)										
Clock Input Pin (mid-PIC) →OUTPUT Pin (Fast)	OR3T20	—	—	—	11.35	—	7.74	—	6.10	ns
	OR3T30	—	—	—	11.63	—	7.93	—	6.27	ns
	OR3T55	—	14.91	—	12.17	—	8.28	—	6.59	ns
	OR3C/T80	—	15.71	—	12.80	—	8.66	—	6.95	ns
	OR3T125	—	—	—	13.69	—	9.24	—	7.49	ns
Clock Input Pin (mid-PIC) →OUTPUT Pin (Slewlim)	OR3T20	—	—	—	13.34	—	8.42	—	6.63	ns
	OR3T30	—	—	—	13.62	—	8.60	—	6.80	ns
	OR3T55	—	17.34	—	14.16	—	8.95	—	7.12	ns
	OR3C/T80	—	18.14	—	14.79	—	9.34	—	7.48	ns
	OR3T125	—	—	—	15.68	—	9.91	—	8.02	ns
Clock Input Pin (mid-PIC) →OUTPUT Pin (Sinklim)	OR3T20	—	—	—	14.69	—	13.26	—	11.37	ns
	OR3T30	—	—	—	14.97	—	13.45	—	11.54	ns
	OR3T55	—	18.70	—	15.51	—	13.80	—	11.86	ns
	OR3C/T80	—	19.51	—	16.14	—	14.18	—	12.22	ns
	OR3T125	—	—	—	17.03	—	14.76	—	12.76	ns
Additional Delay if Non-mid-PIC Used as Clock Pin	OR3T20	—	—	—	0.16	—	0.18	—	0.17	ns
	OR3T30	—	—	—	0.20	—	0.21	—	0.20	ns
	OR3T55	—	0.41	—	0.36	—	0.37	—	0.35	ns
	OR3C/T80	—	0.63	—	0.55	—	0.57	—	0.55	ns
	OR3T125	—	—	—	1.11	—	1.05	—	1.02	ns
Output Not on Same Side of Device As Input Clock (System Clock Delays Using General User I/O Inputs)										
Additional Delay if Output Not on Same Side as Input Clock Pin	OR3T20	—	—	—	0.16	—	0.18	—	0.17	ns
	OR3T30	—	—	—	0.20	—	0.21	—	0.20	ns
	OR3T55	—	0.41	—	0.36	—	0.37	—	0.35	ns
	OR3C/T80	—	0.63	—	0.55	—	0.57	—	0.55	ns
	OR3T125	—	—	—	1.11	—	1.05	—	1.02	ns

Note:
This clock delay is for a fully routed clock tree that uses the primary clock network. It includes both the input buffer delay, the clock routing to the PIO CLK input, the clock→Q of the FF, and the delay through the output buffer. The delay will be reduced if any of the clock branches are not used. The given timing requires that the input clock pin be located at one of the four center PICs on any side of the device and that a PIO FF be used. For clock pins located at any other PIO, see the results reported by ispLEVER.



5-4846(F)

Figure 78. System Clock to Output Delay

Timing Characteristics (continued)

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

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 Setup Time (middle ECLK pin)	OR3T20	—	—	1.34	—	0.88	—	0.83	—	ns
	OR3T30	—	—	1.30	—	0.86	—	0.82	—	ns
	OR3T55	1.36	—	1.22	—	0.83	—	0.80	—	ns
	OR3C/T80	1.25	—	1.14	—	0.80	—	0.77	—	ns
	OR3T125	—	—	1.03	—	0.76	—	0.74	—	ns
Input to ECLK Setup Time (middle ECLK pin, delayed data input)	OR3T20	—	—	6.30	—	5.32	—	5.98	—	ns
	OR3T30	—	—	6.27	—	5.30	—	5.97	—	ns
	OR3T55	6.91	—	6.19	—	5.27	—	5.95	—	ns
	OR3C/T80	6.79	—	6.11	—	5.24	—	5.93	—	ns
	OR3T125	—	—	6.00	—	5.20	—	5.90	—	ns
Input to ECLK Setup 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	—	0.00	—	0.00	—	ns
	OR3C/T80	0.00	—	0.00	—	0.00	—	0.00	—	ns
	OR3T125	—	—	0.00	—	0.00	—	0.00	—	ns
Input to ECLK Setup Time (corner ECLK pin, delayed data input)	OR3T20	—	—	4.39	—	3.51	—	4.41	—	ns
	OR3T30	—	—	4.35	—	3.40	—	4.31	—	ns
	OR3T55	4.94	—	4.28	—	3.18	—	4.11	—	ns
	OR3C/T80	4.82	—	4.21	—	2.98	—	3.91	—	ns
	OR3T125	—	—	4.10	—	2.63	—	3.61	—	ns
Input to ECLK Hold Time (middle ECLK pin)	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
Input to ECLK Hold Time (middle 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

Note:
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.

Timing Characteristics (continued)

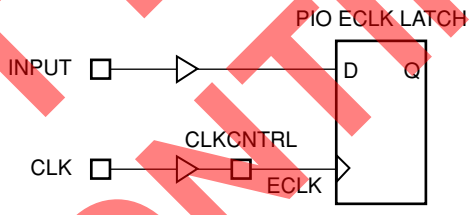
Table 57. OR3C/Txxx Input to ExpressCLK (ECLK) Fast-Capture Setup/Hold Time (Pin-to-Pin) (continued)

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 58. OR3C/Txxx Input to Fast Clock Setup/Hold Time (Pin-to-Pin)

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	
Output Not on Same Side of Device As Input Clock (Fast Clock Delays Using ExpressCLK Inputs)										
Input to FCLK Setup Time (middle ECLK pin)	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
Input to FCLK Setup Time (middle ECLK pin, delayed data input)	OR3T20	—	—	0.80	—	0.58	—	2.20	—	ns
	OR3T30	—	—	0.74	—	0.55	—	2.17	—	ns
	OR3T55	0.29	—	0.62	—	0.51	—	2.11	—	ns
	OR3C/T80	0.14	—	0.50	—	0.46	—	2.06	—	ns
	OR3T125	—	—	0.22	—	0.33	—	1.90	—	ns
Input to FCLK Setup 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	—	0.00	—	0.00	—	ns
	OR3C/T80	0.00	—	0.00	—	0.00	—	0.00	—	ns
	OR3T125	—	—	0.00	—	0.00	—	0.00	—	ns
Input to FCLK Setup 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
Input to FCLK Hold Time (middle ECLK pin)	OR3T20	—	—	4.29	—	3.72	—	3.27	—	ns
	OR3T30	—	—	4.50	—	3.80	—	3.35	—	ns
	OR3T55	6.33	—	4.97	—	3.96	—	3.52	—	ns
	OR3C/T80	6.95	—	5.49	—	4.15	—	3.72	—	ns
	OR3T125	—	—	6.36	—	4.47	—	4.05	—	ns

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

The FCLK delays are for a fully routed clock tree that uses the ExpressCLK input into the fast clock network. It includes both the input buffer delay and the clock routing to the PFU CLK input. The delay will be reduced if any of the clock branches are not used.

Timing Characteristics (continued)

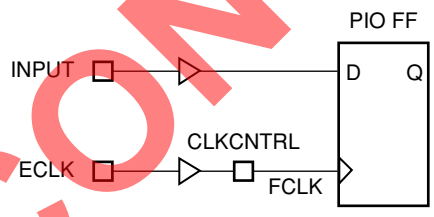
Table 58. OR3C/Txxx Input to Fast Clock Setup/Hold Time (Pin-to-Pin) (continued)

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 FCLK Hold Time (middle 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
Input to FCLK Hold Time (corner ECLK pin)	OR3T20	—	—	6.26	—	5.54	—	4.88	—	ns
	OR3T30	—	—	6.49	—	5.72	—	5.04	—	ns
	OR3T55	8.43	—	6.98	—	6.09	—	5.40	—	ns
	OR3C/T80	9.09	—	7.53	—	6.47	—	5.79	—	ns
	OR3T125	—	—	8.45	—	7.10	—	6.40	—	ns
Input to FCLK 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 FCLK delays are for a fully routed clock tree that uses the ExpressCLK input into the fast clock network. It includes both the input buffer delay and the clock routing to the PFU CLK input. The delay will be reduced if any of the clock branches are not used.



5-4847(F).a

Figure 80. Input to Fast Clock Setup/Hold Time

Timing Characteristics (continued)

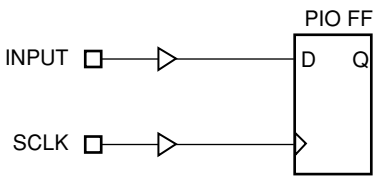
Table 59. OR3C/Txxx Input to General System Clock (SCLK) Setup/Hold Time (Pin-to-Pin)

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 SCLK Setup Time	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
Input to SCLK Setup Time (delayed data input)	OR3T20	—	—	1.33	—	1.47	—	3.09	—	ns
	OR3T30	—	—	1.22	—	1.40	—	3.03	—	ns
	OR3T55	0.99	—	1.09	—	1.33	—	2.97	—	ns
	OR3C/T80	0.79	—	0.93	—	1.26	—	2.91	—	ns
	OR3T125	—	—	0.78	—	1.19	—	2.86	—	ns
Input to SCLK Hold Time	OR3T20	—	—	4.74	—	3.64	—	3.04	—	ns
	OR3T30	—	—	5.01	—	3.83	—	3.22	—	ns
	OR3T55	6.82	—	5.56	—	4.18	—	3.54	—	ns
	OR3C/T80	7.62	—	6.19	—	4.56	—	3.89	—	ns
	OR3T125	—	—	7.07	—	5.14	—	4.44	—	ns
Input to SCLK Hold Time (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
Additional Hold Time if Non-mid-PIC Used as SCLK Pin (no delay on data input)	OR3T20	—	—	0.16	—	0.18	—	0.17	—	ns
	OR3T30	—	—	0.20	—	0.21	—	0.20	—	ns
	OR3T55	0.41	—	0.36	—	0.37	—	0.35	—	ns
	OR3C/T80	0.63	—	0.55	—	0.57	—	0.55	—	ns
	OR3T125	—	—	1.11	—	1.05	—	1.02	—	ns

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

This clock delay is for a fully routed clock tree that uses the clock network. It includes both the input buffer delay and the clock routing to the PIO FF CLK input. The delay will be reduced if any of the clock branches are not used. The given setup (delayed and no delay) and hold (delayed) timing allows the input clock pin to be located in any PIO on any side of the device, but a PIO FF must be used. The hold (no delay) timing assumes the clock pin is located at one of the four middle PICs on any side of the device and that a PIO FF is used. If the clock pin is located elsewhere, then the last parameter in the table must be added to the hold (no delay) timing.



5-4847(F)

Figure 81. Input to System Clock Setup/Hold Time

Timing Characteristics (continued)

Configuration Timing

Table 60. General Configuration Mode 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	Min	Max	Unit
All Configuration Modes				
M[3:0] Setup Time to $\overline{INIT}$ High	TSMODE	0.00	—	ns
M[3:0] Hold Time from $\overline{INIT}$ High	THMODE	600.00	—	ns
$\overline{RESET}$ Pulse Width Low to Start Reconfiguration	TRW	50.00	—	ns
$\overline{PRGM}$ Pulse Width Low to Start Reconfiguration	TPGW	50.00	—	ns
Master and Asynchronous Peripheral Modes				
Power-on Reset Delay	TPO	15.70	52.40	ms
CCLK Period (M3 = 0)	TCCLK	60.00	200.00	ns
(M3 = 1)		480.00	1600.00	ns
Configuration Latency (autoincrement mode):	TCL			
OR3T20 (M3 = 0)		11.50	38.40*	ms
(M3 = 1)		92.10	307.00*	ms
OR3T30 (M3 = 0)		15.10	50.40*	ms
(M3 = 1)		121.00	403.30*	ms
OR3T55 (M3 = 0)		23.20	77.40*	ms
(M3 = 1)		185.00	619.00*	ms
OR3C/T80 (M3 = 0)		33.70	113.00*	ms
(M3 = 1)		270.00	900.00*	ms
OR3T125 (M3 = 0)		52.30	175.00*	ms
(M3 = 1)		418.00	1395.00*	ms
Microprocessor (MPI) Mode				
Power-on Reset Delay	TPO	15.70	52.40	ms
Configuration Latency (autoincrement mode):	TCL			
OR3T20		27413	—	write cycles
OR3T30		35445	—	write cycles
OR3T55		53341	—	write cycles
OR3C/T80		76317	—	write cycles
OR3T125		116581	—	write cycles
Partial Reconfiguration (explicit mode):	TPR			
OR3T20		32	—	write cycles
OR3T30		36	—	write cycles
OR3T55		43	—	write cycles
OR3C/T80		51	—	write cycles
OR3T125		62	—	write cycles
Slave Serial Mode				
Power-on Reset Delay	TPO	3.90	13.10	ms
CCLK Period	TCCLK			
OR3Cxx		40	—	ns
OR3Txxx		15	—	ns
Configuration Latency (autoincrement mode):	TCL			
OR3T20		2.80	—	ms
OR3T30		3.80	—	ms
OR3T55		5.80	—	ms
OR3C80		22.50	—	ms
OR3T80		8.40	—	ms
OR3T125		13.09	—	ms

* Not applicable to asynchronous peripheral mode.

Timing Characteristics (continued)

Table 60. General Configuration Mode Timing Characteristics (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.

Parameter	Symbol	Min	Max	Unit
Slave Parallel Mode				
Power-on Reset Delay	T_{PO}	3.90	13.10	ms
CCLK Period:	T_{CCLK}			
OR3Cxx		40.00	—	ns
OR3Txxx		15.00	—	ns
Configuration Latency (normal mode):	T_{CL}			
OR3T20		0.36	—	ms
OR3T30		0.47	—	ms
OR3T55		0.72	—	ms
OR3C80		2.81	—	ms
OR3T80		1.05	—	ms
OR3T125		1.64	—	ms
Partial Reconfiguration (explicit mode):	T_{PR}			
OR3T20		0.48	—	µs/frame
OR3T30		0.54	—	µs/frame
OR3T55		0.65	—	µs/frame
OR3C80		2.04	—	µs/frame
OR3T80		0.77	—	µs/frame
OR3T125		0.93	—	µs/frame
$\overline{\text{INIT}}$ Timing				
$\overline{\text{INIT}}$ High to CCLK Delay:	T_{INIT_CCLK}			
Slave Parallel		1.00	—	µs
Slave Serial		1.00	—	µs
Master Serial:				
(M3 = 1)		1.00	3.40	µs
(M3 = 0)		0.50	2.00	µs
Master Parallel:				
(M3 = 1)		4.80	16.20	µs
(M3 = 0)		1.00	3.60	µs
Initialization Latency ($\overline{\text{PRGM}}$ high to $\overline{\text{INIT}}$ high):	T_{IL}			
OR3T20		0.21	0.68	ms
OR3T30		0.24	0.79	ms
OR3T55		0.30	1.00	ms
OR3C/T80		0.36	1.20	ms
OR3T125		0.45	1.50	ms
$\overline{\text{INIT}}$ High to $\overline{\text{WR}}$, Asynchronous Peripheral	T_{INIT_WR}	2.00	—	µs

Note: T_{PO} is triggered when V_{DD} reaches between 3.0 V to 4.0 V for the OR3Cxx and between 2.7 V and 3.0 V for the OR3Txxx.

Timing Characteristics (continued)

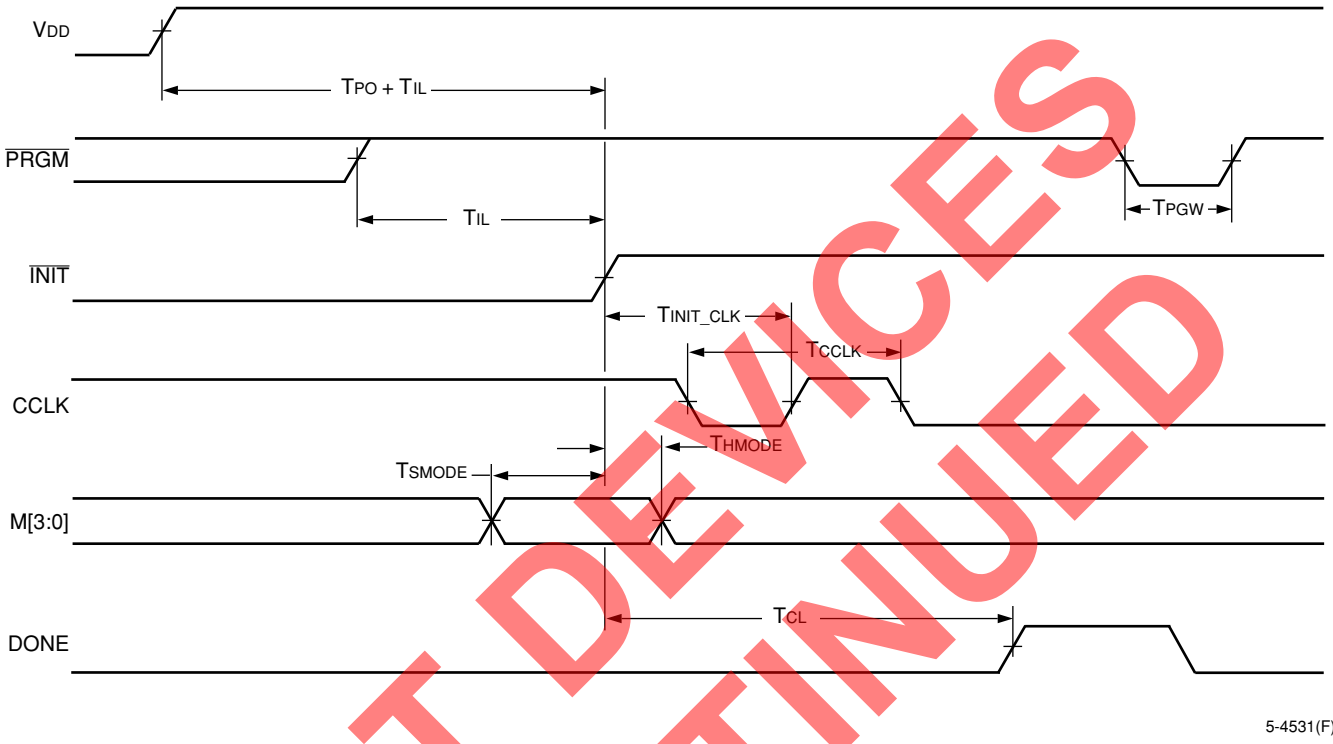


Figure 82. General Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 61. Master Serial Configuration Mode 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	Min	Max	Unit
DIN Setup Time*	T_S	60.00	—	ns
DIN Hold Time	T_H	0.00	—	ns
CCLK Frequency (M3 = 0)	F_C	5.00	16.67	MHz
CCLK Frequency (M3 = 1)	F_C	0.63	2.08	MHz
CCLK to DOUT Delay	T_D	—	5.00	ns

* Data gets clocked out from an external serial ROM. The clock to data delay of the serial ROM must be less than the CCLK frequency since the data available out of the serial ROM must be setup and waiting to be clocked into the FPGA before the next CCLK rising edge.

Note: Serial configuration data is transmitted out on DOUT on the falling edge of CCLK after it is input on DIN.

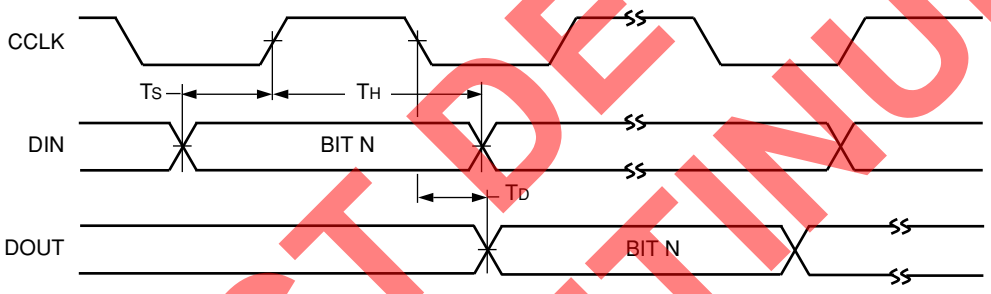


Figure 83. Master Serial Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 62. Master Parallel Configuration Mode 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	Min	Max	Unit
RCLK to Address Valid	T_{AV}	—	60.00	ns
D[7:0] Setup Time to RCLK High	T_S	60.00	—	ns
D[7:0] Hold Time to RCLK High	T_H	0.00	—	ns
RCLK Low Time (M3 = 0)	T_{CL}	7.00	7.00	CCLK cycles
RCLK High Time (M3 = 0)	T_{CH}	1.00	1.00	CCLK cycles
RCLK Low Time (M3 = 1)	T_{CL}	7.00	7.00	CCLK cycles
RCLK High Time (M3 = 1)	T_{CH}	1.00	1.00	CCLK cycles
CCLK to DOUT	T_D	—	5.00	ns

Notes:
The RCLK period consists of seven CCLKs for RCLK low and one CCLK for RCLK high.

Serial data is transmitted out on DOUT 1.5 CCLK cycles after the byte is input on D[7:0].

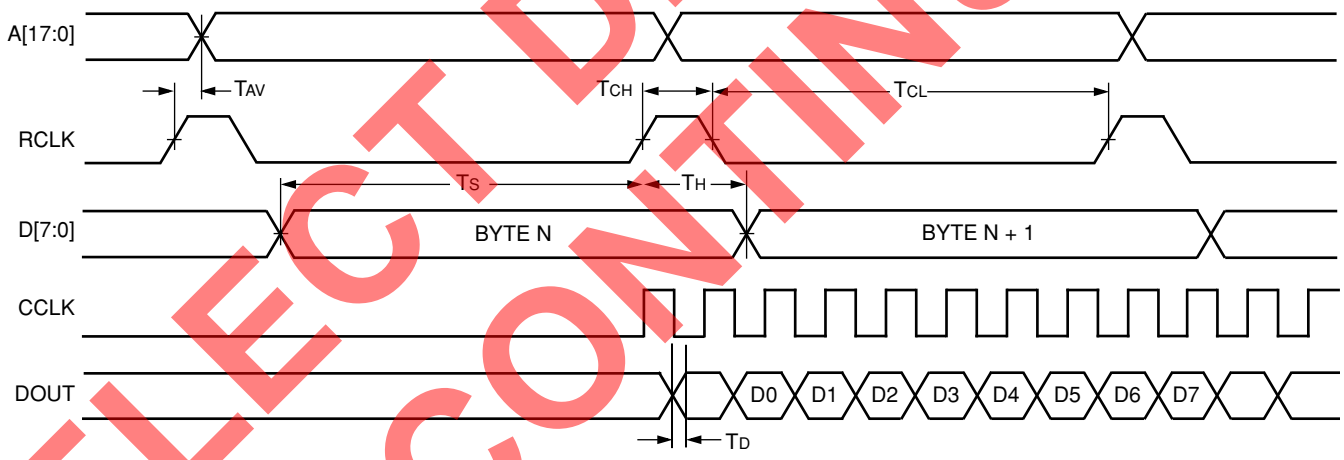


Figure 84. Master Parallel Configuration Mode Timing Diagram

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 $\overline{RD}$.

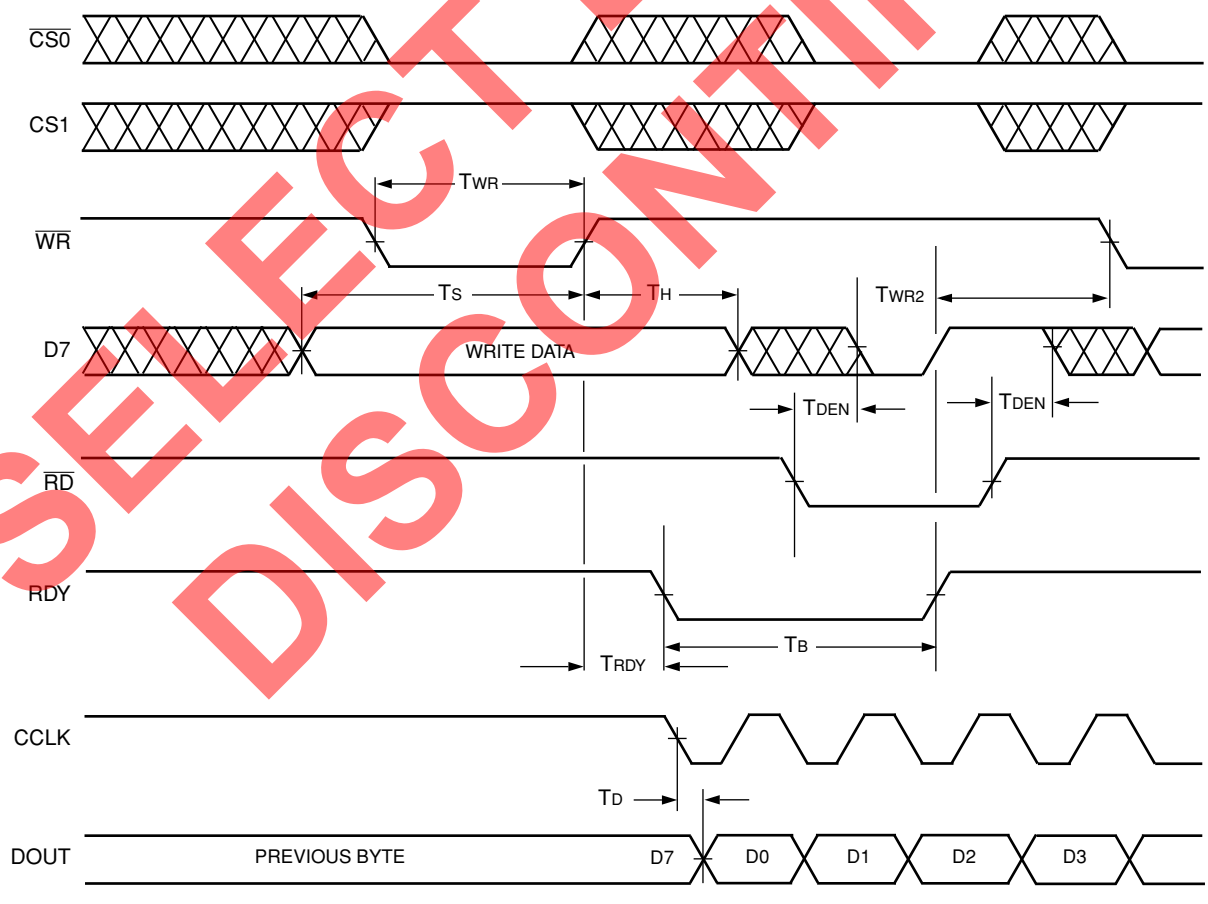


Figure 85. Asynchronous Peripheral Configuration Mode Timing Diagram

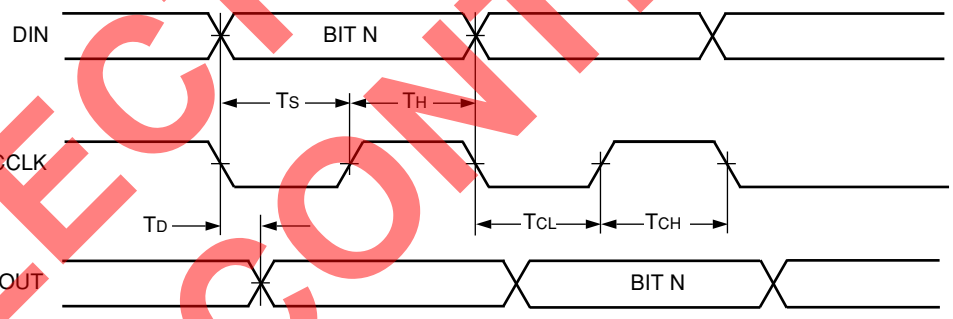
Timing Characteristics (continued)

Table 64. Slave Serial Configuration Mode 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	Min	Max	Unit
DIN Setup Time: 3Cxx 3Txxx	T_S	 20.00 10.50	 — —	 ns ns
DIN Hold Time	T_H	0.00	—	ns
CCLK High Time: 3Cxx 3Txxx	T_{CH}	 20.00 7.00	 — —	 ns ns
CCLK Low Time: 3Cxx 3Txxx	T_{CL}	 20.00 7.00	 — —	 ns ns
CCLK Frequency: 3Cxx 3Txxx	F_C	 — —	 25.00 66.00	 MHz MHz
CCLK to DOUT	T_D	—	20.00	ns

Note: Serial configuration data is transmitted out on DOUT on the falling edge of CCLK after it is input on DIN.



5-4535(F).

Figure 86. Slave Serial Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 65. Slave Parallel Configuration Mode 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	Min	Max	Unit
$\overline{CS0}$, CS1, $\overline{WR}$ Setup Time	T_{S1}	40.00	—	ns
$\overline{CS0}$, CS1, $\overline{WR}$ Hold Time	T_{H1}	20.00	—	ns
D[7:0] Setup Time: 3Cxx 3Txxx	T_{S2}	 20.00 7.00	 — —	 ns ns
D[7:0] Hold Time	T_{H2}	0.00	—	ns
CCLK High Time: 3Cxx 3Txxx	T_{CH}	 20.00 7.00	 — —	 ns ns
CCLK Low Time: 3Cxx 3Txxx	T_{CL}	 20.00 7.00	 — —	 ns ns
CCLK Frequency: 3Cxx 3Txxx	F_C	 — —	 25.00 66.00	 MHz MHz

Note: Daisy-chaining of FPGAs is not supported in this mode.

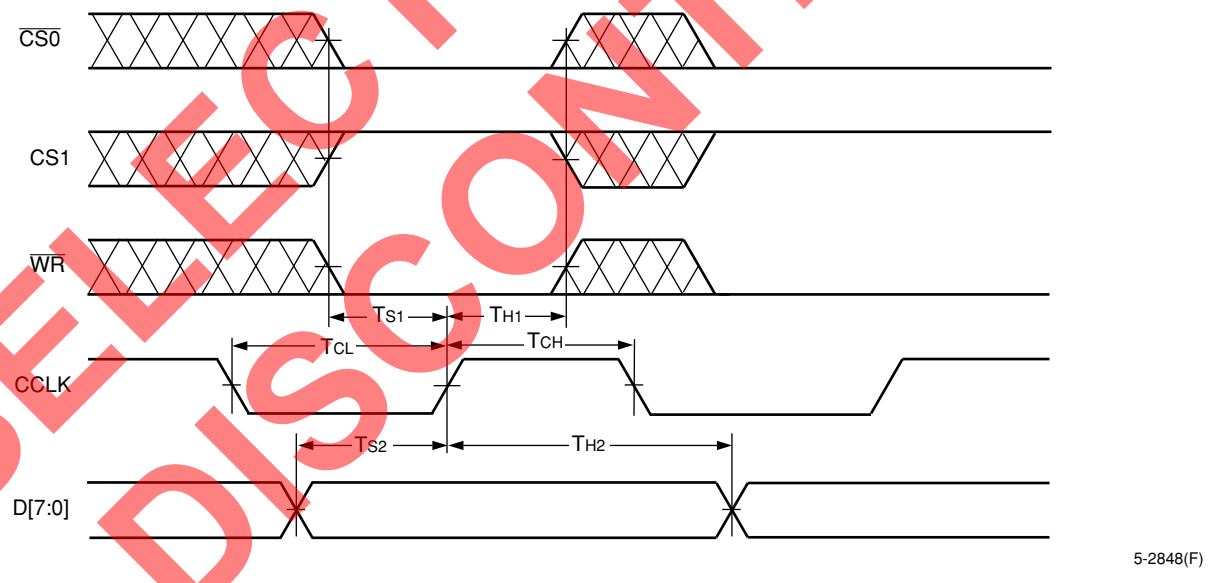


Figure 87. Slave Parallel Configuration Mode Timing Diagram

Timing Characteristics (continued)

Microprocessor Interface (MPI) Configuration Timing Characteristics

For configuration timing using the MPI, consult Table 49. See Figures 67 through 74 for MPI timing diagrams.

Timing Characteristics (continued)

Readback Timing

Table 66. Readback 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
$\overline{\text{RD\_CFG}}$ to CCLK Setup Time	TS	50.00	—	ns
$\overline{\text{RD\_CFG}}$ High Width to Abort Readback	TRBA	2	—	CCLK cycles
CCLK Low Time	TCL	40.00	—	ns
CCLK High Time	TCH	40.00	—	ns
CCLK Frequency	FC	—	12.50	MHz
CCLK to RD_DATA Delay	TD	—	40.00	ns

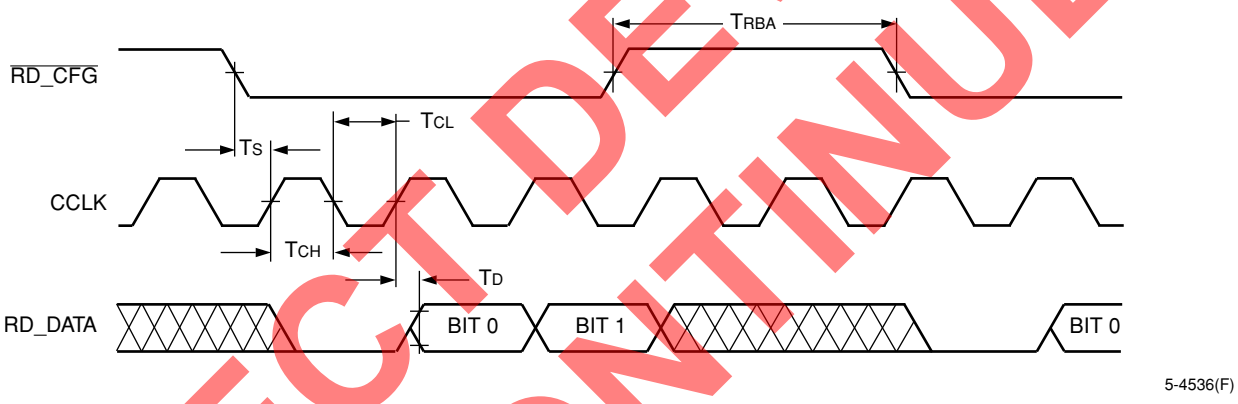
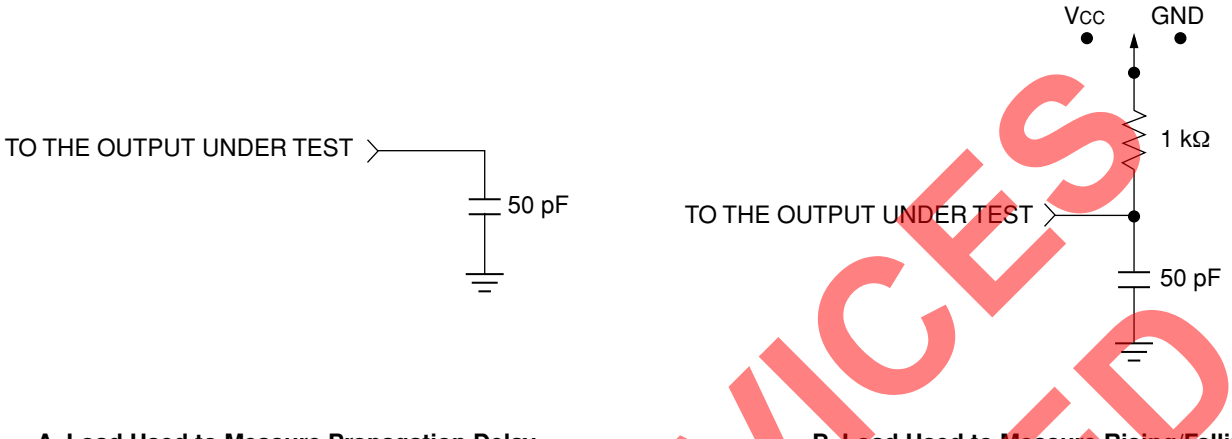


Figure 88. Readback Timing Diagram

Input/Output Buffer Measurement Conditions



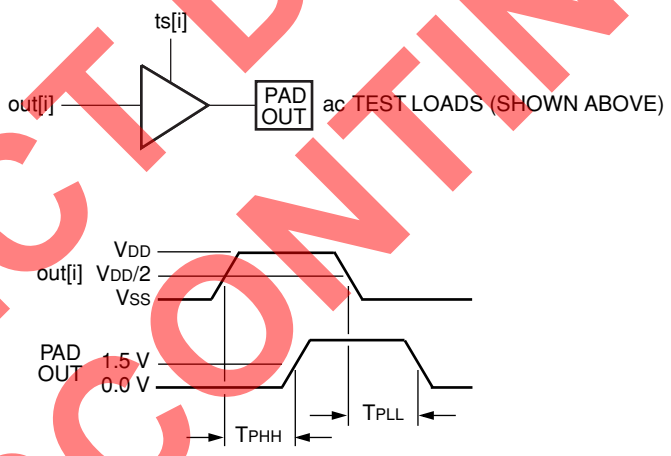
A. Load Used to Measure Propagation Delay

B. Load Used to Measure Rising/Falling Edges

Note: Switch to VDD for TPLZ/TPZL; switch to GND for TPHZ/TPZH.

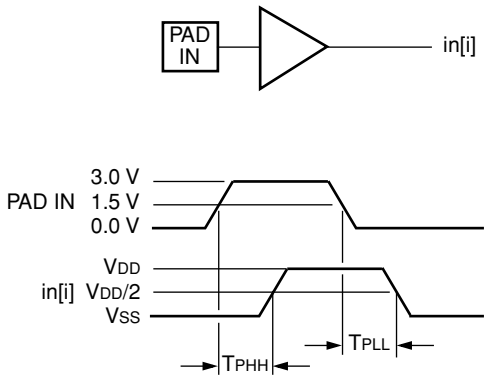
5-3234(F)

Figure 89. ac Test Loads



5-3233.a(F)

Figure 90. Output Buffer Delays



5-3235(F)

Figure 91. Input Buffer Delays

Output Buffer Characteristics

OR3Cxx

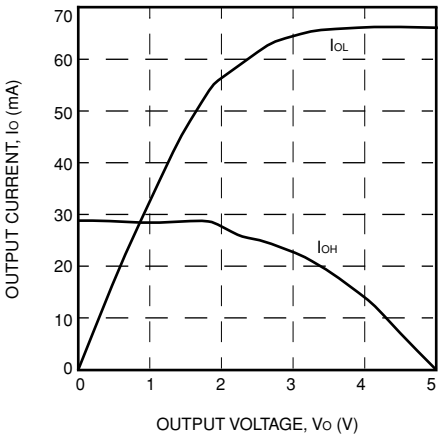


Figure 92. Sinklim (TJ = 25 ¡C, VDD = 5.0 V)

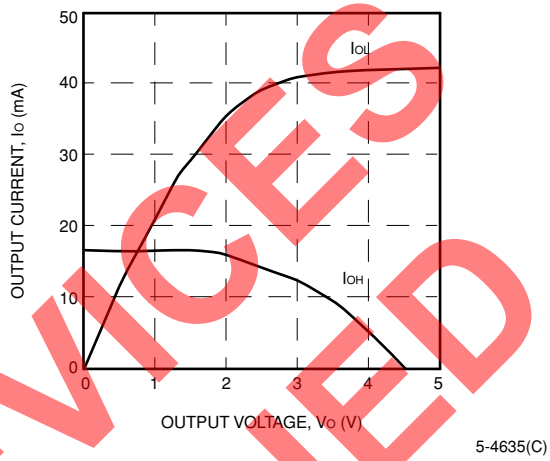


Figure 95. Sinklim (TJ = 125 ¡C, VDD = 4.5 V)

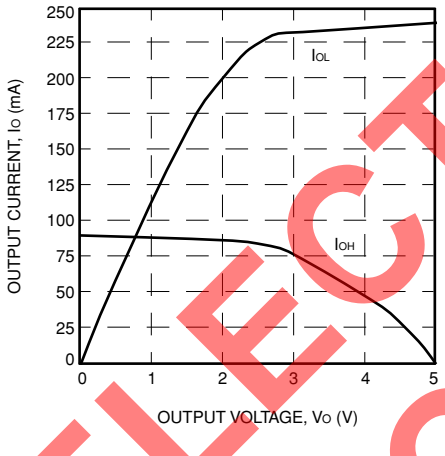


Figure 93. Slewlim (TJ = 25 ¡C, VDD = 5.0 V)

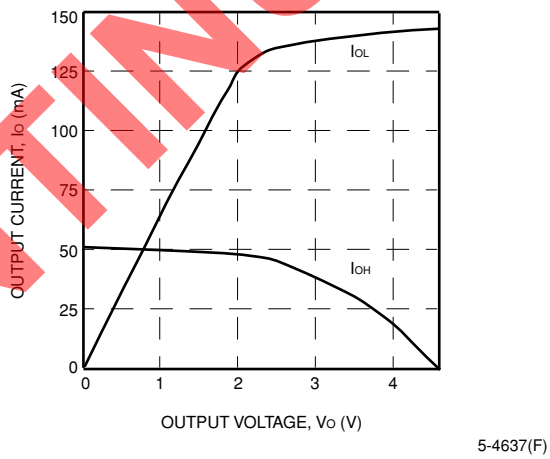


Figure 96. Slewlim (TJ = 125 ¡C, VDD = 4.5 V)

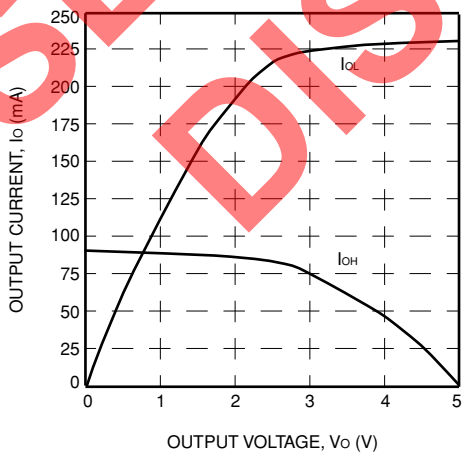


Figure 94. Fast (TJ = 25 ¡C, VDD = 5.0 V)

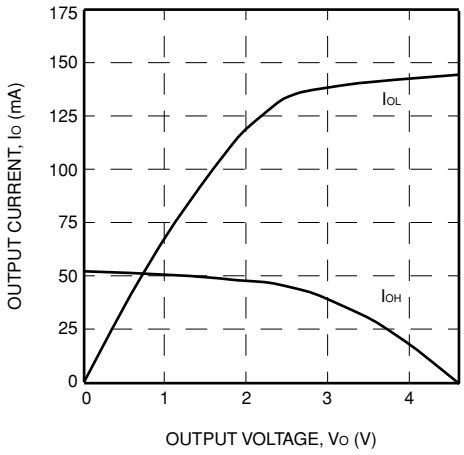


Figure 97. Fast (TJ = 125 ¡C, VDD = 4.5 V)

Output Buffer Characteristics (continued)

OR3Txxx

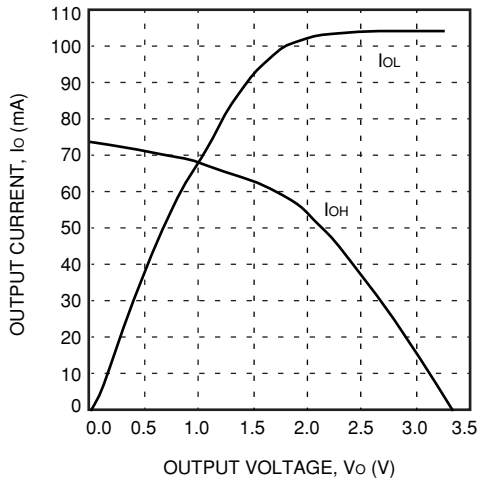


Figure 98. Sinklim (TJ = 25 ¡C, VDD = 3.3 V)

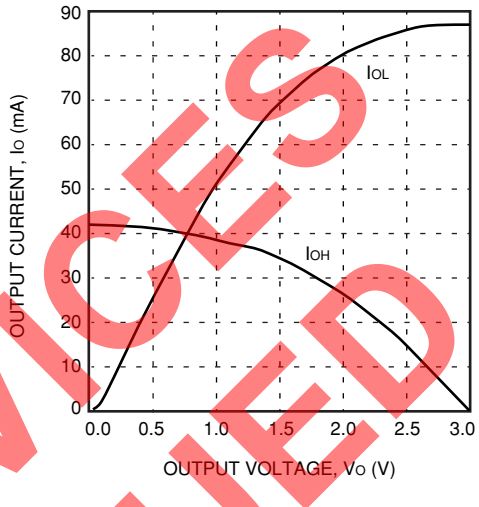


Figure 101. Sinklim (TJ = 125 ¡C, VDD = 3.0 V)

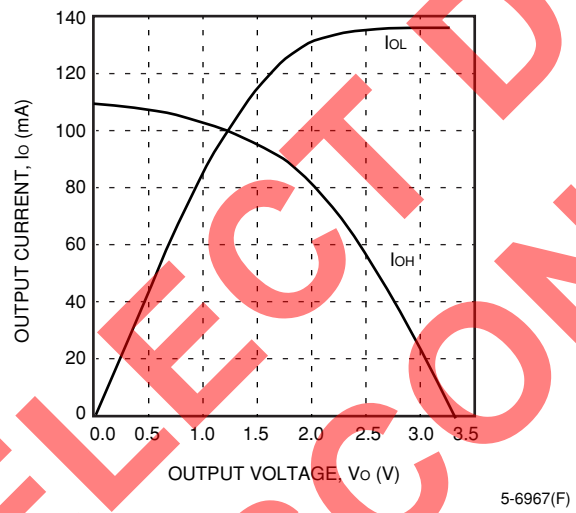


Figure 99. Slewlim (TJ = 25 ¡C, VDD = 3.3 V)

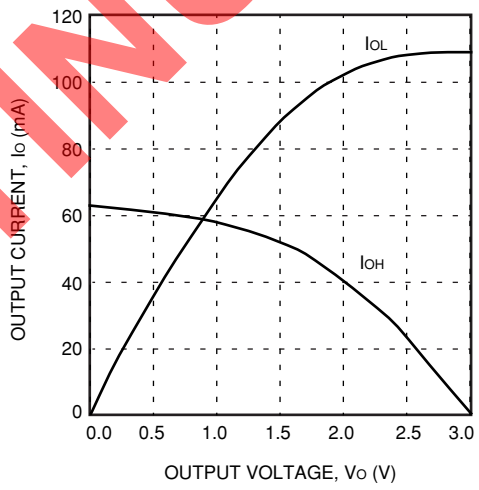


Figure 102. Slewlim (TJ = 125 ¡C, VDD = 3.0 V)

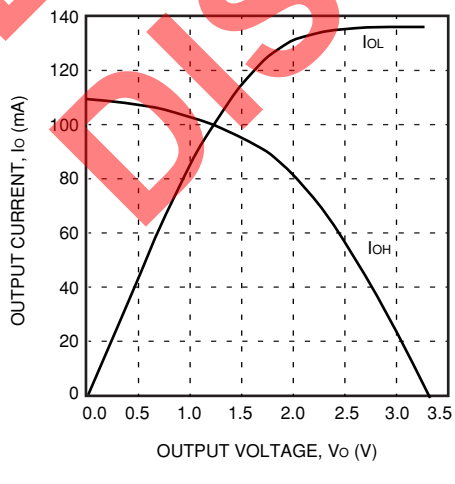


Figure 100. Fast (TJ = 25 ¡C, VDD = 3.3 V)

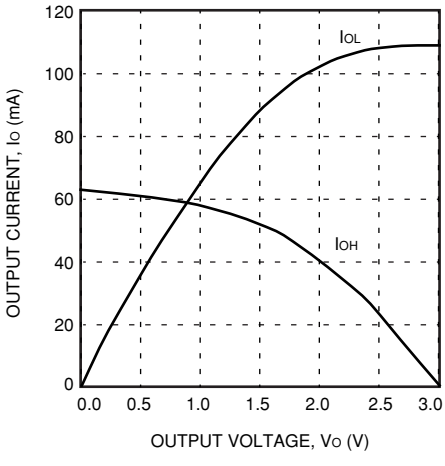


Figure 103. Fast (TJ = 125 ¡C, VDD = 3.0 V)

Estimating Power Dissipation

OR3Cxx

The total operating power dissipated is estimated by summing the standby (I_{DDSB}), internal, and external power dissipated. The internal and external power is the power consumed in the PLCs and PICs, respectively. In general, the standby power is small and may be neglected. The total operating power is as follows:

$$P_T = \Sigma P_{PLC} + \Sigma P_{PIC}$$

The internal operating power is made up of two parts: clock generation and PFU output power. The PFU output power can be estimated based upon the number of PFU outputs switching when driving an average fan-out of two:

$$P_{PFU} = 0.136 \text{ mW/MHz}$$

For each PFU output that switches, 0.136 mW/MHz needs to be multiplied times the frequency (in MHz) that the output switches. Generally, this can be estimated by using one-half the clock rate, multiplied by some activity factor; for example, 20%.

The power dissipated by the clock generation circuitry is based upon four parts: the fixed clock power, the power/clock branch row or column, the clock power dissipated in each PFU that uses this particular clock, and the power from the subset of those PFUs that are configured as synchronous memory. Therefore, the clock power can be calculated for the four parts using the following equations:

OR3C80 Clock Power

P = [0.224 mW/MHz
+ (0.288 mW/MHz/Branch) (# Branches)
+ (0.033 mW/MHz/PFU) (# PFUs)
+ (0.008 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3C80 clock power ≈ 21.06 mW/MHz.

The power dissipated in a PIC is the sum of the power dissipated in the four PIOs in the PIC. This consists of power dissipated by inputs and ac power dissipated by outputs. The power dissipated in each PIO depends on whether it is configured as an input, output, or input/output. If a PIO is operating as an output, then there is a power dissipation component for P_{IN}, as well as P_{OUT}. This is because the output feeds back to the input.

The power dissipated by a TTL input buffer is estimated as:

$$P_{TTL} = 2.2 \text{ mW} + 0.17 \text{ mW/MHz}$$

The power dissipated by an input buffer is estimated as:

$$P_{CMOS} = 0.17 \text{ mW/MHz}$$

The ac power dissipation from an output or bidirectional is estimated by the following:

$$P_{OUT} = (C_L + 8.8 \text{ pF}) \times V_{DD}^2 \times F \text{ Watts}$$

where the unit for C_L is farads, and the unit for F is Hz.

As an example of estimating power dissipation, suppose that a fully utilized OR3C80 has an average of six outputs for each of the 484 PFUs, that 10 clock brances are used so that the clock is driven to the entire PLC array, that 150 of the 484 PFUs have FFs clocked at 40 MHz, and that the PFUoutputs have an average activity factor of 20%.

Twenty TTL-configured inputs, 20 CMOS-configured inputs, 32 outputs driving 30 pF loads, and 16 biderectional I/Os driving 50 pF loads are also generated from the 40 MHz clock with an average activity factor of 20%. All of the ouptut PIOs are registered, and 30 of the input PIOs are registered. The worst-case (V_{DD} = 5.25 V) power dissipation is estimated as follows:

P_{PFU} = 484 x 6 (0.136 mW/MHz x 20 MHz x 20%)
= 1579.78 mW

P_{CLK} = [40 X [0.224 mW/MHz + (0.288 mW/MHz/Branch) (10 Branches)
+ (0.033 mW/MHz/PFU) (150 PFUs)
+ (0.008 mW/MHz/PIO) (58 PIOs)]
= 340.72 mW

P_{TTL} = 20 x [2.2 mW + (0.17 mW/MHz x 20 MHz x 20%)]
= 57.6 mW

P_{CMOS} = 20 x [0.17 mW x 20 MHz x 20%]
= 13.6 mW

P_{OUT} = 32 x [(30 pF + 8.8 pF) x $(5.25)^2$ x 20 MHz x 20%]
= 136.89 mW

P_{BID} = 16 x [(50 pF + 8.8 pF) x $(5.25)^2$ x 20 MHz x 20%]
= 103.72 mW

Total = 2.23 W

Estimating Power Dissipation (continued)

OR3Txxx

The total operating power dissipated is estimated by summing the standby (I_{DDSB}), internal, and external power dissipated. The internal and external power is the power consumed in the PLCs and PICs, respectively. In general, the standby power is small and may be neglected. The total operating power is as follows:

$$P_T = \Sigma P_{PLC} + \Sigma P_{PIC}$$

The internal operating power is made up of two parts: clock generation and PFU output power. The PFU output power can be estimated based upon the number of PFU outputs switching when driving an average fan-out of two:

$$P_{PFU} = 0.068 \text{ mW/MHz}$$

For each PFU output that switches, 0.068 mW/MHz needs to be multiplied times the frequency (in MHz) that the output switches. Generally, this can be estimated by using one-half the clock rate, multiplied by some activity factor; for example, 20%.

The power dissipated by the clock generation circuitry is based upon four parts: the fixed clock power, the power/clock branch row or column, the clock power dissipated in each PFU that uses this particular clock, and the power from the subset of those PFUs configured as synchronous memory. Therefore, the clock power can be calculated for the four parts using the following equations.

OR3T20 Clock Power

P = [0.38 mW/MHz
+ (0.045 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T20 clock power ≈ 2.92 mW/MHz.

OR3T30 Clock Power

P = [0.53 mW/MHz
+ (0.061 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T30 clock power ≈ 3.98 mW/MHz.

OR3T55 Clock Power

P = [0.88 mW/MHz
+ (0.102 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T55 clock power ≈ 6.58 mW/MHz.

OR3T80 Clock Power

P = [0.107 mW/MHz
+ (0.124 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T80 clock power ≈ 9.47 mW/MHz.

OR3T125 Clock Power

P = [0.167 mW/MHz
+ (0.193 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T125 clock power ≈ 15.44 mW/MHz.

The power dissipated in a PIC is the sum of the power dissipated in the four PIOs in the PIC. This consists of power dissipated by inputs and ac power dissipated by outputs. The power dissipated in each PIO depends on whether it is configured as an input, output, or input/output. If a PIO is operating as an output, then there is a power dissipation component for P_{IN}, as well as P_{OUT}. This is because the output feeds back to the input.

The power dissipated by an input buffer ($V_{IH} = V_{DD} –$ 0.3 V or higher) is estimated as:

$$P_{IN} = 0.09 \text{ mW/MHz}$$

The ac power dissipation from an output or bidirectional is estimated by the following:

$$P_{OUT} = (C_L + 8.8 \text{ pF}) \times V_{DD}^2 \times F \text{ Watts}$$

where the unit for C_L is farads, and the unit for F is Hz.

Estimating Power Dissipation (continued)

As an example of estimating power dissipation, suppose that a fully utilized OR3T80 has an average of six outputs for each of the 484 PFUs, that 12 clock branches are used so that the clock is driven to the entire PLC array, that 250 of the 484 PFUs have FFs clocked at 40 MHz, and that the PFU outputs have an average activity factor of 20%.

Eighty inputs, 40 of them used as 5 V tolerant inputs, 50 outputs driving 30 pF loads, and 30 bidirectional I/Os driving 50 pF loads are also generated from the 40 MHz clock with an average activity factor of 20%. All of the output PIOs are registered, and 30 of the input PIOs are registered.

The worst-case (V_{DD} = 3.6 V) power dissipation is estimated as follows:

P_{PFU} = 484 x 6 (0.068 mW/MHz x 20 MHz x 20%)
= 789.9 mW

P_{CLK} = [0.107 mW/MHz + (0.09 mW/MHz – Branch) (12 Branches)
+ (0.015 mW/MHz – PFU) (250 PFUs)
+ (0.004 mW/MHz/PIO) (110 PIOs)]
= 230.43 mW

P_{IN} = 80 x [0.09 mW/MHz x 20 MHz x 20%]
= 28.8 mW

P_{OUT} = 50 x [(30 pF + 8.8 pF) x $(3.6)^2$ x 20 MHz x 20%]
= 100.57 mW

P_{BID} = 30 x [(50 pF + 8.8 pF) x $(3.6)^2$ x 20 MHz x 20%]
= 91.45 mW

TOTAL = 1.241 W

Pin Information

Pin Descriptions

This section describes the pins found on the Series 3 FPGAs. Any pin not described in this table is a user-programmable I/O. During configuration, the user-programmable I/Os are 3-stated with an internal pull-up resistor enabled. If any pin is not used (or not bonded to a package pin), it is also 3-stated with an internal pull-up resistor enabled after configuration.

Table 67. Pin Descriptions

Symbol	I/O	Description
Dedicated Pins		
VDD	—	Positive power supply.
GND	—	Ground supply.
VDD5	—	5 V tolerant select. VDD5 pin locations are shown for package compatibility with OR2TxxA devices. Connections to 5 V power sources are not used for 5 V tolerant I/Os in the OR3Txxx devices.
$\overline{\text{RESET}}$	I	During configuration, $\overline{\text{RESET}}$ forces the restart of configuration and a pull-up is enabled. After configuration, $\overline{\text{RESET}}$ can be used as a general FPGA input or as a direct input, which causes all PLC latches/FFs to be asynchronously set/reset.
CCLK	I	In the master and asynchronous peripheral modes, CCLK is an output which strobes configuration data in. In the slave or synchronous peripheral mode, CCLK is input synchronous with the data on DIN or D[7:0]. In microprocessor mode, CCLK is used internally and output for daisy-chain operation.
DONE	I	As an input, a low level on DONE delays FPGA start-up after configuration (see Note).
	O	As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. DONE has an optional pull-up resistor.
$\overline{\text{PRGM}}$	I	$\overline{\text{PRGM}}$ is an active-low input that forces the restart of configuration and resets the boundary-scan circuitry. This pin always has an active pull-up.
$\overline{\text{RD\_CFG}}$	I	This pin must be held high during device initialization until the $\overline{\text{INIT}}$ pin goes high. This pin always has an active pull-up. During configuration, $\overline{\text{RD\_CFG}}$ is an active-low input that activates the TS_ALL function and 3-states all of the I/O. After configuration, $\overline{\text{RD\_CFG}}$ can be selected (via a bit stream option) to activate the TS_ALL function as described above, or, if readback is enabled via a bit stream option, a high-to-low transition on $\overline{\text{RD\_CFG}}$ will initiate readback of the configuration data, including PFU output states, starting with frame address 0.
RD_DATA/TDO	O	RD_DATA/TDO is a dual-function pin. If used for readback, RD_DATA provides configuration data out. If used in boundary scan, TDO is test data out.
Special-Purpose Pins		
M0, M1, M2	I	During powerup and initialization, M0—M2 are used to select the configuration mode with their values latched on the rising edge of $\overline{\text{INIT}}$; see Table 34 for the configuration modes. During configuration, a pull-up is enabled.
	I/O	After configuration, these pins are user-programmable I/O (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 Information (continued)

Table 67. Pin Descriptions (continued)

Symbol	I/O	Description
Special-Purpose Pins (continued)		
M3	I	During powerup and initialization, M3 is used to select the speed of the internal oscillator during configuration with their values latched on the rising edge of $\overline{\text{INIT}}$. When M3 is low, the oscillator frequency is 10 MHz. When M3 is high, the oscillator is 1.25 MHz. During configuration, a pull-up is enabled.
	I/O	After configuration, this pin is a user-programmable I/O pin (see Note).
TDI, TCK, TMS	I	If boundary scan is used, these pins are test data in, test clock, and test mode select inputs. If boundary scan is not selected, all boundary-scan functions are inhibited once configuration is complete. Even if boundary scan is not used, either TCK or TMS must be held at logic 1 during configuration. Each pin has a pull-up enabled during configuration.
	I/O	After configuration, these pins are user-programmable I/O (see Note).
RDY/RCLK/ MPI_ALE	O	During configuration in peripheral mode, RDY/RCLK indicates another byte can be written to the FPGA. If a read operation is done when the device is selected, the same status is also available on D7 in asynchronous peripheral mode.
	O	During the master parallel configuration mode, RCLK is a read output signal to an external memory. This output is not normally used.
	I	In *i960* microprocessor mode, this pin acts as the address latch enable (ALE) input.
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin (see Note).
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
	I/O	After configuration, this pin is a user-programmable I/O pin (see Note).
$\overline{\text{LDC}}$	O	$\overline{\text{Low During Configuration}}$ is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
	I/O	After configuration, this pin is a user-programmable I/O pin (see Note).
$\overline{\text{INIT}}$	I/O	$\overline{\text{INIT}}$ is a bidirectional signal before and during configuration. During configuration, a pull-up is enabled, but an external pull-up resistor is recommended. As an active-low open-drain output, $\overline{\text{INIT}}$ is held low during power stabilization and internal clearing of memory. As an active-low input, $\overline{\text{INIT}}$ holds the FPGA in the wait-state before the start of configuration.
	I/O	After configuration, this pin 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 Information (continued)

Table 67. Pin Descriptions (continued)

Symbol	I/O	Description
Special-Purpose Pins (continued)		
$\overline{CS0}$, CS1	I	$\overline{CS0}$ and CS1 are used in the asynchronous peripheral, slave parallel, and microprocessor configuration modes. The FPGA is selected when $\overline{CS0}$ is low and CS1 is high. During configuration, a pull-up is enabled.
	I/O	After configuration, these pins are user-programmable I/O pins (see Note).
$\overline{RD}$/ $\overline{MPI\_STRB}$	I	$\overline{RD}$ is used in the asynchronous peripheral configuration mode. A low on $\overline{RD}$ changes D7 into a status output. As a status indication, a high indicates ready, and a low indicates busy. $\overline{WR}$ and $\overline{RD}$ should not be used simultaneously. If they are, the write strobe overrides.
	I	This pin is also used as the microprocessor interface (MPI) data transfer strobe. For *PowerPC*, it is the transfer start (TS). For *i960*, it is the address/data strobe ($\overline{ADS}$).
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin (see Note).
$\overline{WR}$	I	$\overline{WR}$ is used in the asynchronous peripheral configuration mode. When the FPGA is selected, a low on the write strobe, $\overline{WR}$, loads the data on D[7:0] inputs into an internal data buffer. $\overline{WR}$ and $\overline{RD}$ should not be used simultaneously. If they are, the write strobe overrides.
	I/O	After configuration, this pin is a user-programmable I/O pin (see Note).
A[17:0]	O	During master parallel configuration mode, A[17:0] address the configuration EPROM. In microprocessor interface (MPI) mode, many of the A[n] pins have alternate uses as described below. See the Special Function Blocks section for more MPI information. During configuration, if not in master parallel or an MPI configuration mode, these pins are 3-stated with a pull-up enabled.
	I/O	After configuration, the pins are user-programmable I/O pins (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 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 Information (continued)

Package Compatibility

Table 68 provides the number of user I/Os available for the *ORCA* Series 3 FPGAs for each available package. Each package has six dedicated configuration pins.

Tables 70—75 provide the package pin and pin function for the *ORCA* Series 3 FPGAs and packages. The bond pad name is identified in the PIC nomenclature used in the ispLEVER design editor.

When the number of FPGA bond pads exceeds the number of package pins, bond pads are unused. When the number of package pins exceeds the number of bond pads, package pins are left unconnected (no connects). When a package pin is to be left as a no connect for a specific die, it is indicated as a note in the device pad column for the FPGA. The tables provide no information on unused pads.

Table 68. *ORCA* I/Os Summary

Device	144-Pin TQFP	208-Pin SQFP/SQPF2	240-Pin SQFP/SQFP2	256-Pin PBGA	352-Pin PBGA	432-Pin EBGA
OR3T20						
User I/Os*	114	171	—	192	—	—
VDD/VSS	24	31	—	26	—	—
Configuration	6	6	—	6	—	—
Unused	0	0	—	32	—	—
OR3T30						
User I/Os*	—	171	192	221	—	—
VDD/VSS	—	31	40	26	—	—
Configuration	—	6	6	6	—	—
Unused	—	0	2	3	—	—
OR3T55						
User I/Os*	—	171	192	223	288	—
VDD/VSS	—	31	42	26	48	—
Configuration	—	6	6	6	6	—
Unused	—	0	0	1	10	—
OR3C/T80						
User I/Os*	—	171	192	—	298	342
VDD/VSS	—	31	42	—	48	84
Configuration	—	6	6	—	6	6
Unused	—	0	0	—	0	0
OR3T125						
User I/Os*	—	171	192	—	298	342
VDD/VSS	—	31	42	—	48	84
Configuration	—	6	6	—	6	6
Unused	—	0	0	—	0	0

*User I/O count includes four ExpressCLK inputs.

Pin Information (continued)

Compatibility with OR2C/TxxA Series

The pinouts shown for the OR3Cxx and OR3Txxx devices are consistent with the OR2C/TxxA Series for all devices offered in the same packages. This includes the following pins: VDD, VSS, VDD5 (OR2TxxA Series only), and all configuration pins.

The following restrictions apply:

1. There are two configuration modes supported in the OR2C/TxxA Series that are **not** supported in Series 3: master parallel down and synchronous peripheral modes. The Series 3 FPGAs have two new microprocessor interface (MPI) configuration modes that are unavailable in the OR2C/TxxA Series.
2. There are four pins—one per each device side—that are user I/O in the OR2C/TxxA Series which can only be used as fast dedicated clocks or global inputs in Series 3. These pins are also used to drive the ExpressCLK to the I/O FFs on their given side of the device. These four middle ExpressCLK pins should not be used to connect to a programmable clock manager (PCM). A corner ExpressCLK input should be used instead (see item 3 below). See Table 69 for a list of these pins in each package.
3. There are two other pins that are user I/O in both the OR2C/TxxA and Series 3 but also have optional added functionality. Each of these pins drives the ExpressCLKs on two sides of the device. They also have fast connectivity to the programmable clock manager (PCM). See Table 69 for a list of these pins in each package.

Table 69. Series 3 ExpressCLK Pins

Pin Name/ Package	144-Pin TQFP	208-Pin SQFP/SQFP2	240-Pin SQFP/SQFP2	256-Pin PBGA	352-Pin PBGA	432-Pin EBGA
I-ECKL	15	22	26	K3	N2	R29
I-ECKB	55	80	91	W11	AE14	AH16
I-ECKR	92	131	152	K18	N23	T2
I-ECKT	124	178	207	B11	B14	C15
I/O-SECKLL	33	49	56	W1	AB4	AG29
I/O-SECKUR	111	159	184	A19	A25	D5

Table 70. OR3T20 144-Pin TQFP Pinout

Pin	OR3T20 Pad	Function
1	VDD	VDD
2	VSS	VSS
3	PL1A	I/O-A0/$\overline{\text{MPI\_BE0}}$
4	PL2D	I/O
5	PL2A	I/O-A1/$\overline{\text{MPI\_BE1}}$
6	PL3D	I/O-A2
7	PL3A	I/O-A3
8	PL4D	I/O
9	PL4C	I/O
10	PL4A	I/O-A4
11	PL5D	I/O-A5
12	PL5C	I/O
13	PL5A	I/O-A6
14	VSS	VSS
15	PECKL	I-ECKL
16	PL6C	I/O
17	PL6A	I/O-A7/MPI_CLK
18	VDD	VDD
19	PL7D	I/O
20	PL7C	I/O
21	PL7A	I/O-A8/MPI_RW
22	VSS	VSS
23	PL8D	I/O-A9/$\overline{\text{MPI\_ACK}}$
24	PL8A	I/O-A10/$\overline{\text{MPI\_BI}}$
25	PL9D	I/O
26	PL9C	I/O
27	PL9A	I/O-A11/$\overline{\text{MPI\_IRQ}}$
28	PL10D	I/O-A12
29	PL10C	I/O
30	PL10A	I/O-A13
31	PL11A	I/O-A14
32	PL12D	I/O
33	PL12B	I/O-SECKLL
34	PL12A	I/O-A15
35	VSS	VSS
36	PCCLK	CCLK
37	VDD	VDD
38	VSS	VSS
39	PB1A	I/O-A16
40	PB1D	I/O
41	PB2A	I/O-A17
42	PB3A	I/O

Pin	OR3T20 Pad	Function
43	PB3B	I/O
44	PB3D	I/O
45	VDD	VDD
46	PB4A	I/O
47	PB4D	I/O
48	PB5A	I/O
49	PB5C	I/O
50	PB5D	I/O
51	PB6A	I/O
52	PB6C	I/O
53	PB6D	I/O
54	VSS	VSS
55	PECKB	I-ECKB
56	PB7C	I/O
57	PB7D	I/O
58	PB8A	I/O
59	PB8D	I/O
60	PB9A	I/O-HDC
61	PB9C	I/O
62	PB9D	I/O
63	VDD	VDD
64	PB10A	I/O-$\overline{LDC}$
65	PB10C	I/O
66	PB10D	I/O
67	PB11A	I/O-$\overline{INIT}$
68	PB11D	I/O
69	PB12A	I/O
70	VSS	VSS
71	PDONE	DONE
72	VDD	VDD
73	VSS	VSS
74	PRESETN	$\overline{RESET}$
75	PPRGMN	$\overline{PRGM}$
76	PR12A	I/O-M0
77	PR12D	I/O
78	PR11A	I/O
79	PR10A	I/O-M1
80	PR10C	I/O
81	PR10D	I/O
82	PR9A	I/O-M2
83	PR9B	I/O
84	PR9D	I/O
85	PR8A	I/O-M3

Pin	OR3T20 Pad	Function
86	PR8D	I/O
87	VSS	VSS
88	PR7A	I/O
89	PR7C	I/O
90	PR7D	I/O
91	VDD	VDD
92	PECKR	I-ECKR
93	PR6C	I/O
94	PR6D	I/O
95	VSS	VSS
96	PR5A	I/O
97	PR5C	I/O
98	PR5D	I/O
99	PR4A	I/O-CS1
100	PR4D	I/O
101	PR3A	I/O-$\overline{CS0}$
102	PR3D	I/O
103	PR2A	I/O-$\overline{RD}$/MPI_$\overline{STRB}$
104	PR2C	I/O
105	PR2D	I/O
106	PR1A	I/O-$\overline{WR}$
107	VSS	VSS
108	PRD_CFGN	$\overline{RD\_CFG}$
109	VDD	VDD
110	VSS	VSS
111	PT12D	I/O-SECKUR
112	PT12A	I/O-RDY/RCLK/ MPI_ALE
113	PT11D	I/O
114	PT11A	I/O-D7
115	PT10D	I/O
116	PT10C	I/O
117	PT10A	I/O-D6
118	VDD	VDD
119	PT9D	I/O
120	PT9A	I/O-D5
121	PT8D	I/O
122	PT8B	I/O
123	PT8A	I/O-D4
124	PECKT	I-ECKT
125	PT7C	I/O
126	PT7A	I/O-D3
127	VSS	VSS
128	PT6D	I/O

Pin	OR3T20 Pad	Function
129	PT6C	I/O
130	PT6A	I/O-D2
131	PT5D	I/O-D1
132	PT5C	I/O
133	PT5A	I/O-D0/DIN
134	PT4D	I/O
135	PT4A	I/O-DOUT
136	VDD	VDD
137	PT3D	I/O
138	PT3C	I/O
139	PT3A	I/O-TDI
140	PT2A	I/O-TMS
141	PT1D	I/O
142	PT1A	I/O-TCK
143	VSS	VSS
144	PRD_DATA	RD_DATA/TDO

Table 71. OR3T20, OR3T30, OR3T55, OR3C/T80, and OR3T125 208-Pin SQFP/SQFP2 Pinout

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
1	VSS	VSS	VSS	VSS	VSS	VSS
2	VSS	VSS	VSS	VSS	VSS	VSS
3	PL1D	PL1D	PL1D	PL1D	PL1D	I/O
4	PL1A	PL2D	PL2D	PL2D	PL2D	I/O-A0/$\overline{\text{MPI\_BE0}}$
5	PL2D	PL3D	PL3D	PL4D	PL4D	I/O
6	PL2C	PL3C	PL3A	PL4A	PL5D	I/O
7	PL2A	PL3A	PL4A	PL5A	PL7D	I/O-A1/$\overline{\text{MPI\_BE1}}$
8	PL3D	PL4D	PL5A	PL6A	PL8A	I/O-A2
9	PL3C	PL4C	PL6D	PL7D	PL9D	I/O
10	PL3B	PL4B	PL6B	PL7B	PL9B	I/O
11	PL3A	PL4A	PL6A	PL7A	PL9A	I/O-A3
12	VDD	VDD	VDD	VDD	VDD	VDD
13	PL4D	PL5D	PL7D	PL8D	PL10D	I/O
14	PL4C	PL5C	PL7C	PL8A	PL10A	I/O
15	PL4B	PL5B	PL7B	PL9D	PL11D	I/O
16	PL4A	PL5A	PL7A	PL9B	PL11A	I/O-A4
17	PL5D	PL6D	PL8D	PL9A	PL12D	I/O-A5
18	PL5C	PL6C	PL8C	PL10C	PL12A	I/O
19	PL5B	PL6B	PL8B	PL10B	PL13D	I/O
20	PL5A	PL6A	PL8A	PL10A	PL13A	I/O-A6
21	VSS	VSS	VSS	VSS	VSS	VSS
22	PECKL	PECKL	PECKL	PECKL	PECKL	I-ECKL
23	PL6C	PL7C	PL9C	PL11C	PL14C	I/O
24	PL6B	PL7B	PL9B	PL11B	PL14B	I/O
25	PL6A	PL7A	PL9A	PL11A	PL14A	I/O-A7/MPI_CLK
26	VDD	VDD	VDD	VDD	VDD	VDD
27	PL7D	PL8D	PL10D	PL12D	PL15D	I/O
28	PL7C	PL8C	PL10C	PL12C	PL15C	I/O
29	PL7B	PL8B	PL10B	PL12B	PL15B	I/O
30	PL7A	PL8A	PL10A	PL12A	PL15A	I/O-A8/MPI_RW
31	VSS	VSS	VSS	VSS	VSS	VSS
32	PL8D	PL9D	PL11D	PL13D	PL16D	I/O-A9/$\overline{\text{MPI\_ACK}}$
33	PL8C	PL9C	PL11C	PL13B	PL16A	I/O
34	PL8B	PL9B	PL11B	PL13A	PL17D	I/O
35	PL8A	PL9A	PL11A	PL14C	PL17A	I/O-A10/$\overline{\text{MPI\_BI}}$
36	PL9D	PL10D	PL12D	PL14B	PL18D	I/O
37	PL9C	PL10C	PL12C	PL15C	PL18A	I/O
38	PL9B	PL10B	PL12B	PL15B	PL19D	I/O
39	PL9A	PL10A	PL12A	PL15A	PL19A	I/O-A11/$\overline{\text{MPI\_IRQ}}$
40	VDD	VDD	VDD	VDD	VDD	VDD
41	PL10D	PL11D	PL13D	PL16D	PL20D	I/O-A12
42	PL10C	PL11C	PL13B	PL16B	PL20B	I/O

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
43	PL10B	PL11B	PL14D	PL17D	PL21D	I/O
44	PL10A	PL11A	PL14B	PL17B	PL21B	I/O-A13
45	PL11D	PL12D	PL15D	PL18D	PL22D	I/O
46	PL11A	PL12A	PL16D	PL19D	PL24A	I/O-A14
47	PL12D	PL13D	PL17D	PL20D	PL26D	I/O
48	PL12C	PL13A	PL17A	PL21D	PL27D	I/O
49	PL12B	PL14C	PL18C	PL21A	PL27A	I/O-SECKLL
50	PL12A	PL14A	PL18A	PL22A	PL28A	I/O-A15
51	VSS	VSS	VSS	VSS	VSS	VSS
52	PCCLK	PCCLK	PCCLK	PCCLK	PCCLK	CCLK
53	VSS	VSS	VSS	VSS	VSS	VSS
54	VSS	VSS	VSS	VSS	VSS	VSS
55	PB1A	PB1A	PB1A	PB1A	PB1A	I/O-A16
56	PB1B	PB1D	PB1D	PB2A	PB2A	I/O
57	PB1C	PB2A	PB2A	PB2D	PB2D	I/O
58	PB1D	PB2D	PB2D	PB3D	PB3D	I/O
59	PB2A	PB3A	PB3D	PB4D	PB4D	I/O-A17
60	PB2D	PB3D	PB4D	PB5D	PB5D	I/O
61	PB3A	PB4A	PB5B	PB6B	PB6D	I/O
62	PB3B	PB4B	PB5D	PB6D	PB7D	I/O
63	PB3C	PB4C	PB6B	PB7B	PB8D	I/O
64	PB3D	PB4D	PB6D	PB7D	PB9D	I/O
65	VDD	VDD	VDD	VDD	VDD	VDD
66	PB4A	PB5A	PB7A	PB8A	PB10A	I/O
67	PB4B	PB5B	PB7B	PB8D	PB10D	I/O
68	PB4C	PB5C	PB7C	PB9A	PB11A	I/O
69	PB4D	PB5D	PB7D	PB9C	PB11D	I/O
70	PB5A	PB6A	PB8A	PB9D	PB12A	I/O
71	PB5B	PB6B	PB8B	PB10A	PB12D	I/O
72	PB5C	PB6C	PB8C	PB10B	PB13A	I/O
73	PB5D	PB6D	PB8D	PB10D	PB13D	I/O
74	VSS	VSS	VSS	VSS	VSS	VSS
75	PB6A	PB7A	PB9A	PB11A	PB14A	I/O
76	PB6B	PB7B	PB9B	PB11B	PB14B	I/O
77	PB6C	PB7C	PB9C	PB11C	PB14C	I/O
78	PB6D	PB7D	PB9D	PB11D	PB14D	I/O
79	VSS	VSS	VSS	VSS	VSS	VSS
80	PECKB	PECKB	PECKB	PECKB	PECKB	I-ECKB
81	PB7B	PB8B	PB10B	PB12B	PB15B	I/O
82	PB7C	PB8C	PB10C	PB12C	PB15C	I/O
83	PB7D	PB8D	PB10D	PB12D	PB15D	I/O
84	VSS	VSS	VSS	VSS	VSS	VSS
85	PB8A	PB9A	PB11A	PB13A	PB16A	I/O

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
86	PB8B	PB9B	PB11B	PB13B	PB16D	I/O
87	PB8C	PB9C	PB11C	PB13C	PB17A	I/O
88	PB8D	PB9D	PB11D	PB14A	PB17D	I/O
89	PB9A	PB10A	PB12A	PB14B	PB18A	I/O-HDC
90	PB9B	PB10B	PB12B	PB14D	PB18D	I/O
91	PB9C	PB10C	PB12C	PB15A	PB19A	I/O
92	PB9D	PB10D	PB12D	PB15D	PB19D	I/O
93	VDD	VDD	VDD	VDD	VDD	VDD
94	PB10A	PB11A	PB13A	PB16A	PB20A	I/O-$\overline{\text{LDC}}$
95	PB10B	PB11D	PB13D	PB16D	PB21D	I/O
96	PB10C	PB12A	PB14A	PB17A	PB22A	I/O
97	PB10D	PB12B	PB14D	PB17D	PB23D	I/O
98	PB11A	PB12C	PB15A	PB18A	PB24A	I/O-$\overline{\text{INIT}}$
99	PB11C	PB12D	PB16A	PB19A	PB25A	I/O
100	PB11D	PB13A	PB17A	PB20A	PB26A	I/O
101	PB12A	PB13D	PB18A	PB21D	PB27D	I/O
102	PB12D	PB14D	PB18D	PB22D	PB28D	I/O
103	VSS	VSS	VSS	VSS	VSS	VSS
104	PDONE	PDONE	PDONE	PDONE	PDONE	DONE
105	VSS	VSS	VSS	VSS	VSS	VSS
106	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{RESET}}$
107	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PRGM}}$
108	PR12A	PR14A	PR18A	PR22A	PR28A	I/O-M0
109	PR12D	PR13A	PR18D	PR21A	PR27A	I/O
110	PR11A	PR13D	PR17B	PR20A	PR26A	I/O
111	PR11B	PR12A	PR16A	PR19A	PR25A	I/O
112	PR10A	PR11A	PR15D	PR18D	PR22D	I/O-M1
113	PR10B	PR11B	PR14A	PR17A	PR21A	I/O
114	PR10C	PR11C	PR14D	PR17D	PR21D	I/O
115	PR10D	PR11D	PR13A	PR16A	PR20A	I/O
116	VDD	VDD	VDD	VDD	VDD	VDD
117	PR9A	PR10A	PR12A	PR15A	PR19A	I/O-M2
118	PR9B	PR10B	PR12B	PR15D	PR19D	I/O
119	PR9C	PR10C	PR12C	PR14A	PR18A	I/O
120	PR9D	PR10D	PR12D	PR14C	PR18D	I/O
121	PR8A	PR9A	PR11A	PR14D	PR17A	I/O-M3
122	PR8B	PR9B	PR11B	PR13A	PR17D	I/O
123	PR8C	PR9C	PR11C	PR13B	PR16A	I/O
124	PR8D	PR9D	PR11D	PR13D	PR16D	I/O
125	VSS	VSS	VSS	VSS	VSS	VSS
126	PR7A	PR8A	PR10A	PR12A	PR15A	I/O
127	PR7B	PR8B	PR10B	PR12B	PR15B	I/O

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

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
171	PT9B	PT10B	PT12B	PT15A	PT18D	I/O
172	PT9A	PT10A	PT12A	PT14C	PT18A	I/O-D5
173	PT8D	PT9D	PT11D	PT14B	PT17D	I/O
174	PT8C	PT9C	PT11C	PT13D	PT17A	I/O
175	PT8B	PT9B	PT11B	PT13C	PT16D	I/O
176	PT8A	PT9A	PT11A	PT13A	PT16A	I/O-D4
177	VSS	VSS	VSS	VSS	VSS	VSS
178	PECKT	PECKT	PECKT	PECKT	PECKT	I-ECKT
179	PT7C	PT8C	PT10C	PT12C	PT15C	I/O
180	PT7B	PT8B	PT10B	PT12B	PT15B	I/O
181	PT7A	PT8A	PT10A	PT12A	PT15A	I/O-D3
182	VSS	VSS	VSS	VSS	VSS	VSS
183	PT6D	PT7D	PT9D	PT11D	PT14D	I/O
184	PT6C	PT7C	PT9C	PT11C	PT14C	I/O
185	PT6B	PT7B	PT9B	PT11B	PT14B	I/O
186	PT6A	PT7A	PT9A	PT11A	PT14A	I/O-D2
187	VSS	VSS	VSS	VSS	VSS	VSS
188	PT5D	PT6D	PT8D	PT10D	PT13D	I/O-D1
189	PT5C	PT6C	PT8C	PT10B	PT13A	I/O
190	PT5B	PT6B	PT8B	PT10A	PT12D	I/O
191	PT5A	PT6A	PT8A	PT9C	PT12A	I/O-D0/DIN
192	PT4D	PT5D	PT7D	PT9B	PT11D	I/O
193	PT4C	PT5C	PT7C	PT8D	PT11A	I/O
194	PT4B	PT5B	PT7B	PT8C	PT10D	I/O
195	PT4A	PT5A	PT7A	PT8A	PT10A	I/O-DOUT
196	VDD	VDD	VDD	VDD	VDD	VDD
197	PT3D	PT4D	PT6D	PT7D	PT9D	I/O
198	PT3C	PT4C	PT6A	PT7A	PT8A	I/O
199	PT3B	PT4B	PT5C	PT6C	PT7A	I/O
200	PT3A	PT4A	PT5A	PT6A	PT6A	I/O-TDI
201	PT2D	PT3D	PT4A	PT5A	PT5A	I/O
202	PT2A	PT3A	PT3A	PT4A	PT4A	I/O-TMS
203	PT1D	PT2D	PT2C	PT3A	PT3A	I/O
204	PT1C	PT2A	PT2A	PT2A	PT2A	I/O
205	PT1B	PT1D	PT1D	PT1D	PT1D	I/O
206	PT1A	PT1A	PT1A	PT1A	PT1A	I/O-TCK
207	VSS	VSS	VSS	VSS	VSS	VSS
208	PRD_DATA	PRD_DATA	PRD_DATA	PRD_DATA	PRD_DATA	RD_DATA/TDO

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
42	PL10B	PL12B	PL15B	PL19D	I/O
43	PL10A	PL12A	PL15A	PL19A	I/O-A11/$\overline{\text{MPI\_IRQ}}$
44	VDD	VDD	VDD	VDD	VDD
45	PL11D	PL13D	PL16D	PL20D	I/O-A12
46	PL11C	PL13B	PL16B	PL20B	I/O
47	PL11B	PL14D	PL17D	PL21D	I/O
48	PL11A	PL14B	PL17B	PL21B	I/O-A13
49	PL12D	PL14A	PL17A	PL21A	I/O
50	PL12C	PL15D	PL18D	PL22D	I/O
51	PL12B	PL15B	PL18B	PL23D	I/O
52	PL12A	PL16D	PL19D	PL24A	I/O-A14
53	VSS	VSS	VSS	VSS	VSS
54	PL13D	PL17D	PL20D	PL26D	I/O
55	PL13A	PL17A	PL21D	PL27D	I/O
56	PL14C	PL18C	PL21A	PL27A	I/O-SECKLL
57	PL14A	PL18A	PL22A	PL28A	I/O-A15
58	VSS	VSS	VSS	VSS	VSS
59	PCCLK	PCCLK	PCCLK	PCCLK	CCLK
60	VDD	VDD	VDD	VDD	VDD
61	VSS	VSS	VSS	VSS	VSS
62	VSS	VSS	VSS	VSS	VSS
63	PB1A	PB1A	PB1A	PB1A	I/O-A16
64	PB1D	PB1D	PB2A	PB2A	I/O
65	PB2A	PB2A	PB2D	PB2D	I/O
66	PB2D	PB2D	PB3D	PB3D	I/O
67	VSS	VSS	VSS	VSS	VSS
68	PB3A	PB3D	PB4D	PB4D	I/O-A17
69	PB3B	PB4D	PB5D	PB5D	I/O
70	PB3C	PB5A	PB6A	PB6A	I/O
71	PB3D	PB5B	PB6B	PB6D	I/O
72	PB4A	PB5D	PB6D	PB7D	I/O
73	PB4B	PB6A	PB7A	PB8A	I/O
74	PB4C	PB6B	PB7B	PB8D	I/O
75	PB4D	PB6D	PB7D	PB9D	I/O
76	VDD	VDD	VDD	VDD	VDD
77	PB5A	PB7A	PB8A	PB10A	I/O
78	PB5B	PB7B	PB8D	PB10D	I/O
79	PB5C	PB7C	PB9A	PB11A	I/O
80	PB5D	PB7D	PB9C	PB11D	I/O
81	PB6A	PB8A	PB9D	PB12A	I/O
82	PB6B	PB8B	PB10A	PB12D	I/O
83	PB6C	PB8C	PB10B	PB13A	I/O
84	PB6D	PB8D	PB10D	PB13D	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

Pin	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
128	VSS	VSS	VSS	VSS	VSS
129	PR12A	PR16A	PR19A	PR25A	I/O
130	PR12B	PR16D	PR19D	PR24A	I/O
131	PR12C	PR15A	PR18A	PR23A	I/O
132	PR12D	PR15C	PR18C	PR23D	I/O
133	PR11A	PR15D	PR18D	PR22D	I/O-M1
134	PR11B	PR14A	PR17A	PR21A	I/O
135	PR11C	PR14D	PR17D	PR21D	I/O
136	PR11D	PR13A	PR16A	PR20A	I/O
137	VDD	VDD	VDD	VDD	VDD
138	PR10A	PR12A	PR15A	PR19A	I/O-M2
139	PR10B	PR12B	PR15D	PR19D	I/O
140	PR10C	PR12C	PR14A	PR18A	I/O
141	PR10D	PR12D	PR14C	PR18D	I/O
142	PR9A	PR11A	PR14D	PR17A	I/O-M3
143	PR9B	PR11B	PR13A	PR17D	I/O
144	PR9C	PR11C	PR13B	PR16A	I/O
145	PR9D	PR11D	PR13D	PR16D	I/O
146	VSS	VSS	VSS	VSS	VSS
147	PR8A	PR10A	PR12A	PR15A	I/O
148	PR8B	PR10B	PR12B	PR15B	I/O
149	PR8C	PR10C	PR12C	PR15C	I/O
150	PR8D	PR10D	PR12D	PR15D	I/O
151	VDD	VDD	VDD	VDD	VDD
152	PECKR	PECKR	PECKR	PECKR	I-ECKR
153	PR7B	PR9B	PR11B	PR14B	I/O
154	PR7C	PR9C	PR11C	PR14C	I/O
155	PR7D	PR9D	PR11D	PR14D	I/O
156	VSS	VSS	VSS	VSS	VSS
157	PR6A	PR8A	PR10A	PR13A	I/O
158	PR6B	PR8B	PR10C	PR13D	I/O
159	PR6C	PR8C	PR10D	PR12A	I/O
160	PR6D	PR8D	PR9B	PR12D	I/O
161	PR5A	PR7A	PR9C	PR11A	I/O-CS1
162	PR5B	PR7B	PR9D	PR11D	I/O
163	PR5C	PR7C	PR8A	PR10A	I/O
164	PR5D	PR7D	PR8D	PR10D	I/O
165	VDD	VDD	VDD	VDD	VDD
166	PR4A	PR6A	PR7A	PR9A	I/O-$\overline{CS0}$
167	PR4B	PR6B	PR7B	PR9B	I/O
168	PR4C	PR5B	PR6B	PR8B	I/O
169	PR4D	PR5D	PR6D	PR8D	I/O
170	PR3A	PR4A	PR5A	PR7A	I/O-$\overline{RD}$/MPI_$\overline{STRB}$

Pin	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
171	PR3B	PR4B	PR5B	PR6A	I/O
172	PR3C	PR4D	PR5D	PR5A	I/O
173	PR3D	PR3A	PR4A	PR4A	I/O
174	VSS	VSS	VSS	VSS	VSS
175	PR2A	PR2A	PR3A	PR3A	I/O-WR
176	PR2D	PR2C	PR2A	PR2A	I/O
177	PR1A	PR1A	PR1A	PR1A	I/O
178	PR1D	PR1D	PR1D	PR1D	I/O
179	VSS	VSS	VSS	VSS	VSS
180	PRD_CFGN	PRD_CFGN	PRD_CFGN	PRD_CFGN	RD_CFG
181	VSS	VSS	VSS	VSS	VSS
182	VDD	VDD	VDD	VDD	VDD
183	VSS	VSS	VSS	VSS	VSS
184	PT14D	PT18D	PT22D	PT28D	I/O-SECKUR
185	PT14C	PT18B	PT22A	PT28A	I/O
186	PT14A	PT18A	PT21D	PT27D	I/O
187	PT13D	PT17D	PT21A	PT27A	I/O-RDY/RCLK/MPI_ALE
188	—	VSS	VSS	VSS	VSS
189	PT13B	PT16D	PT19D	PT25D	I/O
190	PT13A	PT16C	PT19C	PT25C	I/O
191	PT12D	PT16A	PT19A	PT25A	I/O
192	PT12C	PT15D	PT18D	PT24D	I/O-D7
193	PT12A	PT14D	PT17D	PT23D	I/O
194	PT11D	PT14A	PT17A	PT22D	I/O
195	PT11C	PT13D	PT16D	PT21D	I/O
196	PT11B	PT13B	PT16B	PT20D	I/O-D6
197	VDD	VDD	VDD	VDD	VDD
198	PT10D	PT12D	PT15D	PT19D	I/O
199	PT10C	PT12C	PT15B	PT19A	I/O
200	PT10B	PT12B	PT15A	PT18D	I/O
201	PT10A	PT12A	PT14C	PT18A	I/O-D5
202	PT9D	PT11D	PT14B	PT17D	I/O
203	PT9C	PT11C	PT13D	PT17A	I/O
204	PT9B	PT11B	PT13C	PT16D	I/O
205	PT9A	PT11A	PT13A	PT16A	I/O-D4
206	VSS	VSS	VSS	VSS	VSS
207	PECKT	PECKT	PECKT	PECKT	I-ECKT
208	PT8C	PT10C	PT12C	PT15C	I/O
209	PT8B	PT10B	PT12B	PT15B	I/O
210	PT8A	PT10A	PT12A	PT15A	I/O-D3
211	VSS	VSS	VSS	VSS	VSS
212	PT7D	PT9D	PT11D	PT14D	I/O

Pin	OR3T30 Pad	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
213	PT7C	PT9C	PT11C	PT14C	I/O
214	PT7B	PT9B	PT11B	PT14B	I/O
215	PT7A	PT9A	PT11A	PT14A	I/O-D2
216	VSS	VSS	VSS	VSS	VSS
217	PT6D	PT8D	PT10D	PT13D	I/O-D1
218	PT6C	PT8C	PT10B	PT13A	I/O
219	PT6B	PT8B	PT10A	PT12D	I/O
220	PT6A	PT8A	PT9C	PT12A	I/O-D0/DIN
221	PT5D	PT7D	PT9B	PT11D	I/O
222	PT5C	PT7C	PT8D	PT11A	I/O
223	PT5B	PT7B	PT8C	PT10D	I/O
224	PT5A	PT7A	PT8A	PT10A	I/O-DOUT
225	VDD	VDD	VDD	VDD	VDD
226	PT4D	PT6D	PT7D	PT9D	I/O
227	PT4C	PT6A	PT7A	PT8A	I/O
228	PT4B	PT5C	PT6C	PT7A	I/O
229	PT4A	PT5A	PT6A	PT6A	I/O-TDI
230	PT3D	PT4D	PT5D	PT5D	I/O
231	PT3C	PT4A	PT5A	PT5A	I/O
232	PT3B	PT3D	PT4D	PT4D	I/O
233	PT3A	PT3A	PT4A	PT4A	I/O-TMS
234	VSS	VSS	VSS	VSS	VSS
235	PT2D	PT2C	PT3A	PT3A	I/O
236	PT2A	PT2A	PT2A	PT2A	I/O
237	PT1D	PT1D	PT1D	PT1D	I/O
238	PT1A	PT1A	PT1A	PT1A	I/O-TCK
239	VSS	VSS	VSS	VSS	VSS
240	PRD_DATA	PRD_DATA	PRD_DATA	PRD_DATA	RD_DATA/TDO

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	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	Function
P4	PL10A	PL11A	PL14B	I/O-A13
R3	PL11D	PL12D	PL14A	I/O
T2	PL11C	PL12C	PL15D	I/O
U1	PL11B	PL12B	PL15B	I/O
T3	PL11A	PL12A	PL16D	I/O-A14
U2	—	PL13D	PL17D	I/O
V1	PL12D	PL13C	PL17C	I/O
T4	PL12C	PL13B	PL17B	I/O
U3	—	PL13A	PL17A	I/O
V2	—	PL14D	PL18D	I/O
W1	PL12B	PL14C	PL18C	I/O-SECKLL
V3	—	PL14B	PL18B	I/O
W2	PL12A	PL14A	PL18A	I/O-A15
Y1	PCCLK	PCCLK	PCCLK	CCLK
W3	—	—	—	NC
Y2	PB1A	PB1A	PB1A	I/O-A16
W4	—	PB1C	PB1C	I/O
V4	PB1B	PB1D	PB1D	I/O
U5	PB1C	PB2A	PB2A	I/O
Y3	PB1D	PB2B	PB2B	I/O
Y4	—	PB2C	PB2C	I/O
V5	—	PB2D	PB2D	I/O
W5	PB2A	PB3A	PB3D	I/O-A17
Y5	PB2B	PB3B	PB4D	I/O
V6	PB2C	PB3C	PB5A	I/O
U7	PB2D	PB3D	PB5B	I/O
W6	PB3A	PB4A	PB5D	I/O
Y6	PB3B	PB4B	PB6A	I/O
V7	PB3C	PB4C	PB6B	I/O
W7	PB3D	PB4D	PB6D	I/O
Y7	PB4A	PB5A	PB7A	I/O
V8	PB4B	PB5B	PB7B	I/O
W8	PB4C	PB5C	PB7C	I/O
Y8	PB4D	PB5D	PB7D	I/O
U9	PB5A	PB6A	PB8A	I/O
V9	PB5B	PB6B	PB8B	I/O
W9	PB5C	PB6C	PB8C	I/O
Y9	PB5D	PB6D	PB8D	I/O
W10	PB6A	PB7A	PB9A	I/O
V10	PB6B	PB7B	PB9B	I/O
Y10	PB6C	PB7C	PB9C	I/O
Y11	PB6D	PB7D	PB9D	I/O
W11	PECKB	PECKB	PECKB	I-ECKB
V11	PB7B	PB8B	PB10B	I/O
U11	PB7C	PB8C	PB10C	I/O

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	Function
Y12	PB7D	PB8D	PB10D	I/O
W12	PB8A	PB9A	PB11A	I/O
V12	PB8B	PB9B	PB11B	I/O
U12	PB8C	PB9C	PB11C	I/O
Y13	PB8D	PB9D	PB11D	I/O
W13	PB9A	PB10A	PB12A	I/O-HDC
V13	PB9B	PB10B	PB12B	I/O
Y14	PB9C	PB10C	PB12C	I/O
W14	PB9D	PB10D	PB12D	I/O
Y15	PB10A	PB11A	PB13A	I/O-$\overline{\text{LDC}}$
V14	PB10B	PB11B	PB13B	I/O
W15	PB10C	PB11C	PB13C	I/O
Y16	PB10D	PB11D	PB13D	I/O
U14	—	PB12A	PB14A	I/O
V15	—	PB12B	PB14D	I/O
W16	PB11A	PB12C	PB15A	I/O-$\overline{\text{INIT}}$
Y17	—	—	PB15D	I/O
V16	—	PB12D	PB16A	I/O
W17	PB11B	PB13A	PB16D	I/O
Y18	PB11C	PB13B	PB17A	I/O
U16	PB11D	PB13C	PB17C	I/O
V17	PB12A	PB13D	PB17D	I/O
W18	PB12B	PB14A	PB18A	I/O
Y19	PB12C	PB14B	PB18B	I/O
V18	PB12D	PB14C	PB18C	I/O
W19	—	PB14D	PB18D	I/O
Y20	PDONE	PDONE	PDONE	DONE
W20	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{RESET}}$
V19	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PRGM}}$
U19	PR12A	PR14A	PR18A	I/O-M0
U18	—	PR14C	PR18C	I/O
T17	—	PR14D	PR18D	I/O
V20	—	PR13A	PR17A	I/O
U20	PR12B	PR13B	PR17B	I/O
T18	PR12C	PR13C	PR17C	I/O
T19	PR12D	PR13D	PR17D	I/O
T20	PR11A	PR12A	PR16A	I/O
R18	PR11B	PR12B	PR16D	I/O
P17	PR11C	PR12C	PR15A	I/O
R19	PR11D	PR12D	PR15C	I/O
R20	PR10A	PR11A	PR15D	I/O-M1
P18	PR10B	PR11B	PR14A	I/O
P19	PR10C	PR11C	PR14D	I/O
P20	PR10D	PR11D	PR13A	I/O
N18	PR9A	PR10A	PR12A	I/O-M2

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	Function
N19	PR9B	PR10B	PR12B	I/O
N20	PR9C	PR10C	PR12C	I/O
M17	PR9D	PR10D	PR12D	I/O
M18	PR8A	PR9A	PR11A	I/O-M3
M19	PR8B	PR9B	PR11B	I/O
M20	PR8C	PR9C	PR11C	I/O
L19	PR8D	PR9D	PR11D	I/O
L18	PR7A	PR8A	PR10A	I/O
L20	PR7B	PR8B	PR10B	I/O
K20	PR7C	PR8C	PR10C	I/O
K19	PR7D	PR8D	PR10D	I/O
K18	PECKR	PECKR	PECKR	I-ECKR
K17	PR6B	PR7B	PR9B	I/O
J20	PR6C	PR7C	PR9C	I/O
J19	PR6D	PR7D	PR9D	I/O
J18	PR5A	PR6A	PR8A	I/O
J17	PR5B	PR6B	PR8B	I/O
H20	PR5C	PR6C	PR8C	I/O
H19	PR5D	PR6D	PR8D	I/O
H18	PR4A	PR5A	PR7A	I/O-CS1
G20	PR4B	PR5B	PR7B	I/O
G19	PR4C	PR5C	PR7C	I/O
F20	PR4D	PR5D	PR7D	I/O
G18	PR3A	PR4A	PR6A	I/O-$\overline{CS0}$
F19	PR3B	PR4B	PR6B	I/O
E20	PR3C	PR4C	PR5B	I/O
G17	PR3D	PR4D	PR5D	I/O
F18	PR2A	PR3A	PR4A	I/O-$\overline{RD}$/MPI_$\overline{STRB}$
E19	PR2B	PR3B	PR4B	I/O
D20	PR2C	PR3C	PR4D	I/O
E18	PR2D	PR3D	PR3A	I/O
D19	PR1A	PR2A	PR2A	I/O-$\overline{WR}$
C20	PR1B	PR2B	PR2B	I/O
E17	PR1C	PR2C	PR2C	I/O
D18	PR1D	PR2D	PR2D	I/O
C19	—	PR1A	PR1A	I/O
B20	—	PR1B	PR1B	I/O
C18	—	PR1C	PR1C	I/O
B19	—	PR1D	PR1D	I/O
A20	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{RD\_CFG}$
A19	PT12D	PT14D	PT18D	I/O-SECKUR
B18	—	PT14C	PT18C	I/O
B17	PT12C	PT14B	PT18B	I/O
C17	PT12B	PT14A	PT18A	I/O
D16	PT12A	PT13D	PT17D	I/O-RDY/RCLK/MPI_ALE
A18	—	PT13C	PT17A	I/O

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	Function
A17	PT11D	PT13B	PT16D	I/O
C16	PT11C	PT13A	PT16C	I/O
B16	PT11B	PT12D	PT16A	I/O
A16	PT11A	PT12C	PT15D	I/O-D7
C15	—	PT12B	PT15A	I/O
D14	PT10D	PT12A	PT14D	I/O
B15	PT10C	PT11D	PT14A	I/O
A15	PT10B	PT11C	PT13D	I/O
C14	PT10A	PT11B	PT13B	I/O-D6
B14	PT9D	PT11A	PT13A	I/O
A14	PT9C	PT10D	PT12D	I/O
C13	—	PT10C	PT12C	I/O
B13	PT9B	PT10B	PT12B	I/O
A13	PT9A	PT10A	PT12A	I/O-D5
D12	PT8D	PT9D	PT11D	I/O
C12	PT8C	PT9C	PT11C	I/O
B12	PT8B	PT9B	PT11B	I/O
A12	PT8A	PT9A	PT11A	I/O-D4
B11	PECKT	PECKT	PECKT	I-ECKT
C11	PT7C	PT8C	PT10C	I/O
A11	PT7B	PT8B	PT10B	I/O
A10	PT7A	PT8A	PT10A	I/O-D3
B10	PT6D	PT7D	PT9D	I/O
C10	PT6C	PT7C	PT9C	I/O
D10	PT6B	PT7B	PT9B	I/O
A9	PT6A	PT7A	PT9A	I/O-D2
B9	PT5D	PT6D	PT8D	I/O-D1
C9	PT5C	PT6C	PT8C	I/O
D9	PT5B	PT6B	PT8B	I/O
A8	PT5A	PT6A	PT8A	I/O-D0/DIN
B8	PT4D	PT5D	PT7D	I/O
C8	PT4C	PT5C	PT7C	I/O
A7	PT4B	PT5B	PT7B	I/O
B7	PT4A	PT5A	PT7A	I/O-DOUT
A6	PT3D	PT4D	PT6D	I/O
C7	PT3C	PT4C	PT6A	I/O
B6	PT3B	PT4B	PT5C	I/O
A5	PT3A	PT4A	PT5A	I/O-TDI
D7	PT2D	PT3D	PT4D	I/O
C6	PT2C	PT3C	PT4A	I/O
B5	PT2B	PT3B	PT3D	I/O
A4	PT2A	PT3A	PT3A	I/O-TMS
C5	—	PT2D	PT2D	I/O
B4	PT1D	PT2C	PT2C	I/O
A3	PT1C	PT2B	PT2B	I/O
D5	PT1B	PT2A	PT2A	I/O

Pin	OR3T20 Pad	OR3T30 Pad	OR3T55 Pad	Function
C4	—	PT1D	PT1D	I/O
B3	—	PT1C	PT1C	I/O
B2	—	PT1B	PT1B	I/O
A2	PT1A	PT1A	PT1A	I/O-TCK
C3	PRD_DATA	PRD_DATA	PRD_DATA	RD_DATA/TDO
A1	VSS	VSS	VSS	VSS
D4	VSS	VSS	VSS	VSS
D8	VSS	VSS	VSS	VSS
D13	VSS	VSS	VSS	VSS
D17	VSS	VSS	VSS	VSS
H4	VSS	VSS	VSS	VSS
H17	VSS	VSS	VSS	VSS
N4	VSS	VSS	VSS	VSS
N17	VSS	VSS	VSS	VSS
U4	VSS	VSS	VSS	VSS
U8	VSS	VSS	VSS	VSS
U13	VSS	VSS	VSS	VSS
U17	VSS	VSS	VSS	VSS
J9	VSS	VSS	VSS	VSS*
J10	VSS	VSS	VSS	VSS*
J11	VSS	VSS	VSS	VSS*
J12	VSS	VSS	VSS	VSS*
K9	VSS	VSS	VSS	VSS*
K10	VSS	VSS	VSS	VSS*
K11	VSS	VSS	VSS	VSS*
K12	VSS	VSS	VSS	VSS*
L9	VSS	VSS	VSS	VSS*
L10	VSS	VSS	VSS	VSS*
L11	VSS	VSS	VSS	VSS*
L12	VSS	VSS	VSS	VSS*
M9	VSS	VSS	VSS	VSS*
M10	VSS	VSS	VSS	VSS*
M11	VSS	VSS	VSS	VSS*
M12	VSS	VSS	VSS	VSS*
D6	VDD	VDD	VDD	VDD
D11	VDD	VDD	VDD	VDD
D15	VDD	VDD	VDD	VDD
F4	VDD	VDD	VDD	VDD
F17	VDD	VDD	VDD	VDD
K4	VDD	VDD	VDD	VDD
L17	VDD	VDD	VDD	VDD
R4	VDD	VDD	VDD	VDD
R17	VDD	VDD	VDD	VDD
U6	VDD	VDD	VDD	VDD
U10	VDD	VDD	VDD	VDD
U15	VDD	VDD	VDD	VDD

Table 74. OR3T55, OR3C/T80, and OR3T125 352-Pin PBGA Pinout

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
B1	PL1D	PL1D	PL1D	I/O
C2	PL1C	PL1C	PL1C	I/O
C1	PL1B	PL1B	PL1B	I/O
D2	PL1A	PL1A	PL1A	I/O
D3	PL2D	PL2D	PL2D	I/O-A0/$\overline{\text{MPI\_BE0}}$
D1	PL2C	PL2A	PL2A	I/O
E2	PL2B	PL3D	PL3D	I/O
E4	—	PL3B	PL3B	I/O
E3	PL2A	PL3A	PL3A	I/O
E1	PL3D	PL4D	PL4D	I/O
F2	PL3C	PL4C	PL4C	I/O
G4	PL3B	PL4B	PL4B	I/O
F3	PL3A	PL4A	PL5D	I/O
F1	PL4D	PL5D	PL6D	I/O
G2	PL4C	PL5C	PL6C	I/O
G1	PL4B	PL5B	PL6B	I/O
G3	PL4A	PL5A	PL7D	I/O-A1/$\overline{\text{MPI\_BE1}}$
H2	PL5D	PL6D	PL8D	I/O
J4	PL5C	PL6C	PL8C	I/O
H1	PL5B	PL6B	PL8B	I/O
H3	PL5A	PL6A	PL8A	I/O-A2
J2	PL6D	PL7D	PL9D	I/O
J1	PL6C	PL7C	PL9C	I/O
K2	PL6B	PL7B	PL9B	I/O
J3	PL6A	PL7A	PL9A	I/O-A3
K1	PL7D	PL8D	PL10D	I/O
K4	PL7C	PL8A	PL10A	I/O
L2	PL7B	PL9D	PL11D	I/O
K3	PL7A	PL9B	PL11A	I/O-A4
L1	PL8D	PL9A	PL12D	I/O-A5
M2	PL8C	PL10C	PL12A	I/O
M1	PL8B	PL10B	PL13D	I/O
L3	PL8A	PL10A	PL13A	I/O-A6
N2	PECKL	PECKL	PECKL	I-ECKL
M4	PL9C	PL11C	PL14C	I/O
N1	PL9B	PL11B	PL14B	I/O
M3	PL9A	PL11A	PL14A	I/O-A7/MPI_CLK
P2	PL10D	PL12D	PL15D	I/O
P4	PL10C	PL12C	PL15C	I/O
P1	PL10B	PL12B	PL15B	I/O
N3	PL10A	PL12A	PL15A	I/O-A8/MPI_RW
R2	PL11D	PL13D	PL16D	I/O-A9/$\overline{\text{MPI\_ACK}}$
P3	PL11C	PL13B	PL16A	I/O
R1	PL11B	PL13A	PL17D	I/O
T2	PL11A	PL14C	PL17A	I/O-A10/$\overline{\text{MPI\_BI}}$

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
R3	PL12D	PL14B	PL18D	I/O
T1	PL12C	PL15C	PL18A	I/O
R4	PL12B	PL15B	PL19D	I/O
U2	PL12A	PL15A	PL19A	I/O-A11/$\overline{\text{MPI\_IRQ}}$
T3	PL13D	PL16D	PL20D	I/O-A12
U1	PL13C	PL16C	PL20C	I/O
U4	PL13B	PL16B	PL20B	I/O
V2	PL13A	PL16A	PL20A	I/O
U3	PL14D	PL17D	PL21D	I/O
V1	PL14C	PL17C	PL21C	I/O
W2	PL14B	PL17B	PL21B	I/O-A13
W1	PL14A	PL17A	PL21A	I/O
V3	PL15D	PL18D	PL22D	I/O
Y2	PL15C	PL18C	PL22C	I/O
W4	PL15B	PL18B	PL23D	I/O
Y1	PL15A	PL18A	PL24D	I/O
W3	PL16D	PL19D	PL24A	I/O-A14
AA2	PL16C	PL19C	PL25C	I/O
Y4	PL16B	PL19B	PL25B	I/O
AA1	PL16A	PL19A	PL25A	I/O
Y3	PL17D	PL20D	PL26D	I/O
AB2	PL17C	PL20C	PL26C	I/O
AB1	PL17B	PL20A	PL26A	I/O
AA3	PL17A	PL21D	PL27D	I/O
AC2	PL18D	PL21C	PL27C	I/O
AB4	PL18C	PL21A	PL27A	I/O-SECKLL
AC1	PL18B	PL22D	PL28D	I/O
AB3	—	PL22C	PL28C	I/O
AD2	—	PL22B	PL28B	I/O
AC3	PL18A	PL22A	PL28A	I/O-A15
AD1	PCCLK	PCCLK	PCCLK	CCLK
AF2	PB1A	PB1A	PB1A	I/O-A16
AE3	—	PB1B	PB1B	I/O
AF3	PB1B	PB1C	PB1C	I/O
AE4	PB1C	PB1D	PB1D	I/O
AD4	PB1D	PB2A	PB2A	I/O
AF4	PB2A	PB2D	PB2D	I/O
AE5	PB2B	PB3A	PB3A	I/O
AC5	PB2C	PB3C	PB3C	I/O
AD5	PB2D	PB3D	PB3D	I/O
AF5	PB3A	PB4A	PB4A	I/O
AE6	PB3B	PB4B	PB4B	I/O
AC7	PB3C	PB4C	PB4C	I/O
AD6	PB3D	PB4D	PB4D	I/O-A17
AF6	PB4A	PB5A	PB5A	I/O
AE7	PB4B	PB5B	PB5B	I/O

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
AF7	PB4C	PB5C	PB5C	I/O
AD7	PB4D	PB5D	PB5D	I/O
AE8	PB5A	PB6A	PB6A	I/O
AC9	PB5B	PB6B	PB6D	I/O
AF8	PB5C	PB6C	PB7A	I/O
AD8	PB5D	PB6D	PB7D	I/O
AE9	PB6A	PB7A	PB8A	I/O
AF9	PB6B	PB7B	PB8D	I/O
AE10	PB6C	PB7C	PB9A	I/O
AD9	PB6D	PB7D	PB9D	I/O
AF10	PB7A	PB8A	PB10A	I/O
AC10	PB7B	PB8D	PB10D	I/O
AE11	PB7C	PB9A	PB11A	I/O
AD10	PB7D	PB9C	PB11D	I/O
AF11	PB8A	PB9D	PB12A	I/O
AE12	PB8B	PB10A	PB12D	I/O
AF12	PB8C	PB10B	PB13A	I/O
AD11	PB8D	PB10D	PB13D	I/O
AE13	PB9A	PB11A	PB14A	I/O
AC12	PB9B	PB11B	PB14B	I/O
AF13	PB9C	PB11C	PB14C	I/O
AD12	PB9D	PB11D	PB14D	I/O
AE14	PECKB	PECKB	PECKB	I-ECKB
AC14	PB10B	PB12B	PB15B	I/O
AF14	PB10C	PB12C	PB15C	I/O
AD13	PB10D	PB12D	PB15D	I/O
AE15	PB11A	PB13A	PB16A	I/O
AD14	PB11B	PB13B	PB16D	I/O
AF15	PB11C	PB13C	PB17A	I/O
AE16	PB11D	PB14A	PB17D	I/O
AD15	PB12A	PB14B	PB18A	I/O-HDC
AF16	PB12B	PB14D	PB18D	I/O
AC15	PB12C	PB15A	PB19A	I/O
AE17	PB12D	PB15D	PB19D	I/O
AD16	PB13A	PB16A	PB20A	I/O-$\overline{\text{LDC}}$
AF17	PB13B	PB16B	PB20D	I/O
AC17	PB13C	PB16C	PB21A	I/O
AE18	PB13D	PB16D	PB21D	I/O
AD17	PB14A	PB17A	PB22A	I/O
AF18	PB14B	PB17B	PB23A	I/O
AE19	PB14C	PB17C	PB23C	I/O
AF19	PB14D	PB17D	PB23D	I/O
AD18	PB15A	PB18A	PB24A	I/O-$\overline{\text{INIT}}$
AE20	PB15B	PB18B	PB24B	I/O
AC19	PB15C	PB18C	PB24C	I/O
AF20	PB15D	PB18D	PB24D	I/O

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
AD19	PB16A	PB19A	PB25A	I/O
AE21	PB16B	PB19B	PB25B	I/O
AC20	PB16C	PB19C	PB25C	I/O
AF21	PB16D	PB19D	PB25D	I/O
AD20	PB17A	PB20A	PB26A	I/O
AE22	PB17B	PB20B	PB26B	I/O
AF22	PB17C	PB20D	PB26D	I/O
AD21	PB17D	PB21A	PB27A	I/O
AE23	—	PB21B	PB27B	I/O
AC22	PB18A	PB21D	PB27D	I/O
AF23	PB18B	PB22A	PB28A	I/O
AD22	PB18C	PB22B	PB28B	I/O
AE24	—	PB22C	PB28C	I/O
AD23	PB18D	PB22D	PB28D	I/O
AF24	PDONE	PDONE	PDONE	DONE
AE26	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{PRESETN}}$	$\overline{\text{RESET}}$
AD25	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PPRGMN}}$	$\overline{\text{PRGM}}$
AD26	PR18A	PR22A	PR28A	I/O-M0
AC25	PR18B	PR22C	PR28C	I/O
AC24	PR18C	PR22D	PR28D	I/O
AC26	PR18D	PR21A	PR27A	I/O
AB25	PR17A	PR21D	PR27D	I/O
AB23	PR17B	PR20A	PR26A	I/O
AB24	PR17C	PR20B	PR26B	I/O
AB26	PR17D	PR20D	PR26D	I/O
AA25	PR16A	PR19A	PR25A	I/O
Y23	PR16B	PR19B	PR25B	I/O
AA24	PR16C	PR19C	PR25C	I/O
AA26	PR16D	PR19D	PR24A	I/O
Y25	PR15A	PR18A	PR23A	I/O
Y26	PR15B	PR18B	PR23B	I/O
Y24	PR15C	PR18C	PR23D	I/O
W25	PR15D	PR18D	PR22D	I/O-M1
V23	PR14A	PR17A	PR21A	I/O
W26	PR14B	PR17B	PR21B	I/O
W24	PR14C	PR17C	PR21C	I/O
V25	PR14D	PR17D	PR21D	I/O
V26	PR13A	PR16A	PR20A	I/O
U25	PR13B	PR16B	PR20B	I/O
V24	PR13C	PR16C	PR20C	I/O
U26	PR13D	PR16D	PR20D	I/O
U23	PR12A	PR15A	PR19A	I/O-M2
T25	PR12B	PR15D	PR19D	I/O
U24	PR12C	PR14A	PR18A	I/O
T26	PR12D	PR14C	PR18D	I/O
R25	PR11A	PR14D	PR17A	I/O-M3

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
R26	PR11B	PR13A	PR17D	I/O
T24	PR11C	PR13B	PR16A	I/O
P25	PR11D	PR13D	PR16D	I/O
R23	PR10A	PR12A	PR15A	I/O
P26	PR10B	PR12B	PR15B	I/O
R24	PR10C	PR12C	PR15C	I/O
N25	PR10D	PR12D	PR15D	I/O
N23	PECKR	PECKR	PECKR	I-ECKR
N26	PR9B	PR11B	PR14B	I/O
P24	PR9C	PR11C	PR14C	I/O
M25	PR9D	PR11D	PR14D	I/O
N24	PR8A	PR10A	PR13A	I/O
M26	PR8B	PR10C	PR13D	I/O
L25	PR8C	PR10D	PR12A	I/O
M24	PR8D	PR9B	PR12D	I/O
L26	PR7A	PR9C	PR11A	I/O-CS1
M23	PR7B	PR9D	PR11D	I/O
K25	PR7C	PR8A	PR10A	I/O
L24	PR7D	PR8D	PR10D	I/O
K26	PR6A	PR7A	PR9A	I/O-$\overline{CS0}$
K23	PR6B	PR7B	PR9B	I/O
J25	PR6C	PR7C	PR9C	I/O
K24	PR6D	PR7D	PR9D	I/O
J26	PR5A	PR6A	PR8A	I/O
H25	PR5B	PR6B	PR8B	I/O
H26	PR5C	PR6C	PR8C	I/O
J24	PR5D	PR6D	PR8D	I/O
G25	PR4A	PR5A	PR7A	I/O-$\overline{RD}$/MPI_$\overline{STRB}$
H23	PR4B	PR5B	PR6A	I/O
G26	PR4C	PR5C	PR6C	I/O
H24	PR4D	PR5D	PR5A	I/O
F25	PR3A	PR4A	PR4A	I/O
G23	PR3B	PR4B	PR4B	I/O
F26	PR3C	PR4C	PR4C	I/O
G24	PR3D	PR4D	PR4D	I/O
E25	PR2A	PR3A	PR3A	I/O-$\overline{WR}$
E26	PR2B	PR3B	PR3B	I/O
F24	—	PR3D	PR3D	I/O
D25	PR2C	PR2A	PR2A	I/O
E23	PR2D	PR2D	PR2D	I/O
D26	PR1A	PR1A	PR1A	I/O
E24	PR1B	PR1B	PR1B	I/O
C25	PR1C	PR1C	PR1C	I/O
D24	PR1D	PR1D	PR1D	I/O
C26	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{RD\_CFG}$
A25	PT18D	PT22D	PT28D	I/O-SECKUR

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
B24	PT18C	PT22C	PT28C	I/O
A24	—	PT22B	PT28B	I/O
B23	PT18B	PT22A	PT28A	I/O
C23	PT18A	PT21D	PT27D	I/O
A23	PT17D	PT21A	PT27A	I/O-RDY/RCLK/ $\overline{\text{MPI\_ALE}}$
B22	PT17C	PT20D	PT26D	I/O
D22	PT17B	PT20C	PT26C	I/O
C22	PT17A	PT20A	PT26A	I/O
A22	PT16D	PT19D	PT25D	I/O
B21	PT16C	PT19C	PT25C	I/O
D20	PT16B	PT19B	PT25B	I/O
C21	PT16A	PT19A	PT25A	I/O
A21	PT15D	PT18D	PT24D	I/O-D7
B20	PT15C	PT18C	PT24C	I/O
A20	PT15B	PT18B	PT24B	I/O
C20	PT15A	PT18A	PT24A	I/O
B19	PT14D	PT17D	PT23D	I/O
D18	PT14C	PT17C	PT23C	I/O
A19	PT14B	PT17B	PT23B	I/O
C19	PT14A	PT17A	PT22D	I/O
B18	PT13D	PT16D	PT21D	I/O
A18	PT13C	PT16C	PT21A	I/O
B17	PT13B	PT16B	PT20D	I/O-D6
C18	PT13A	PT16A	PT20A	I/O
A17	PT12D	PT15D	PT19D	I/O
D17	PT12C	PT15B	PT19A	I/O
B16	PT12B	PT15A	PT18D	I/O
C17	PT12A	PT14C	PT18A	I/O-D5
A16	PT11D	PT14B	PT17D	I/O
B15	PT11C	PT13D	PT17A	I/O
A15	PT11B	PT13C	PT16D	I/O
C16	PT11A	PT13A	PT16A	I/O-D4
B14	PECKT	PECKT	PECKT	I-ECKT
D15	PT10C	PT12C	PT15C	I/O
A14	PT10B	PT12B	PT15B	I/O
C15	PT10A	PT12A	PT15A	I/O-D3
B13	PT9D	PT11D	PT14D	I/O
D13	PT9C	PT11C	PT14C	I/O
A13	PT9B	PT11B	PT14B	I/O
C14	PT9A	PT11A	PT14A	I/O-D2
B12	PT8D	PT10D	PT13D	I/O-D1
C13	PT8C	PT10B	PT13A	I/O
A12	PT8B	PT10A	PT12D	I/O
B11	PT8A	PT9C	PT12A	I/O-D0/DIN
C12	PT7D	PT9B	PT11D	I/O
A11	PT7C	PT8D	PT11A	I/O

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
D12	PT7B	PT8C	PT10D	I/O
B10	PT7A	PT8A	PT10A	I/O-DOUT
C11	PT6D	PT7D	PT9D	I/O
A10	PT6C	PT7C	PT9A	I/O
D10	PT6B	PT7B	PT8D	I/O
B9	PT6A	PT7A	PT8A	I/O
C10	PT5D	PT6D	PT7D	I/O
A9	PT5C	PT6C	PT7A	I/O
B8	PT5B	PT6B	PT6D	I/O
A8	PT5A	PT6A	PT6A	I/O-TDI
C9	PT4D	PT5D	PT5D	I/O
B7	PT4C	PT5C	PT5C	I/O
D8	PT4B	PT5B	PT5B	I/O
A7	PT4A	PT5A	PT5A	I/O
C8	PT3D	PT4D	PT4D	I/O
B6	PT3C	PT4C	PT4C	I/O
D7	PT3B	PT4B	PT4B	I/O
A6	PT3A	PT4A	PT4A	I/O-TMS
C7	PT2D	PT3D	PT3D	I/O
B5	PT2C	PT3A	PT3A	I/O
A5	PT2B	PT2D	PT2D	I/O
C6	—	PT2C	PT2C	I/O
B4	—	PT2B	PT2B	I/O
D5	PT2A	PT2A	PT2A	I/O
A4	PT1D	PT1D	PT1D	I/O
C5	PT1C	PT1C	PT1C	I/O
B3	PT1B	PT1B	PT1B	I/O
C4	PT1A	PT1A	PT1A	I/O-TCK
A3	PRD_DATA	PRD_DATA	PRD_DATA	RD_DATA/TDO
A1	Vss	Vss	Vss	Vss
A2	Vss	Vss	Vss	Vss
A26	Vss	Vss	Vss	Vss
AC13	Vss	Vss	Vss	Vss
AC18	Vss	Vss	Vss	Vss
AC23	Vss	Vss	Vss	Vss
AC4	Vss	Vss	Vss	Vss
AC8	Vss	Vss	Vss	Vss
AD24	Vss	Vss	Vss	Vss
AD3	Vss	Vss	Vss	Vss
AE1	Vss	Vss	Vss	Vss
AE2	Vss	Vss	Vss	Vss
AE25	Vss	Vss	Vss	Vss
AF1	Vss	Vss	Vss	Vss
AF25	Vss	Vss	Vss	Vss
AF26	Vss	Vss	Vss	Vss

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
B2	Vss	Vss	Vss	Vss
B25	Vss	Vss	Vss	Vss
B26	Vss	Vss	Vss	Vss
C24	Vss	Vss	Vss	Vss
C3	Vss	Vss	Vss	Vss
D14	Vss	Vss	Vss	Vss
D19	Vss	Vss	Vss	Vss
D23	Vss	Vss	Vss	Vss
D4	Vss	Vss	Vss	Vss
D9	Vss	Vss	Vss	Vss
H4	Vss	Vss	Vss	Vss
J23	Vss	Vss	Vss	Vss
N4	Vss	Vss	Vss	Vss
P23	Vss	Vss	Vss	Vss
V4	Vss	Vss	Vss	Vss
W23	Vss	Vss	Vss	Vss
L11	Vss	Vss	Vss	Vss*
L12	Vss	Vss	Vss	Vss*
L13	Vss	Vss	Vss	Vss*
L14	Vss	Vss	Vss	Vss*
L15	Vss	Vss	Vss	Vss*
L16	Vss	Vss	Vss	Vss*
M11	Vss	Vss	Vss	Vss*
M12	Vss	Vss	Vss	Vss*
M13	Vss	Vss	Vss	Vss*
M14	Vss	Vss	Vss	Vss*
M15	Vss	Vss	Vss	Vss*
M16	Vss	Vss	Vss	Vss*
N11	Vss	Vss	Vss	Vss*
N12	Vss	Vss	Vss	Vss*
N13	Vss	Vss	Vss	Vss*
N14	Vss	Vss	Vss	Vss*
N15	Vss	Vss	Vss	Vss*
N16	Vss	Vss	Vss	Vss*
P11	Vss	Vss	Vss	Vss*
P12	Vss	Vss	Vss	Vss*
P13	Vss	Vss	Vss	Vss*
P14	Vss	Vss	Vss	Vss*
P15	Vss	Vss	Vss	Vss*
P16	Vss	Vss	Vss	Vss*
R11	Vss	Vss	Vss	Vss*
R12	Vss	Vss	Vss	Vss*
R13	Vss	Vss	Vss	Vss*
R14	Vss	Vss	Vss	Vss*
R15	Vss	Vss	Vss	Vss*

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
R16	VSS	VSS	VSS	VSS*
T11	VSS	VSS	VSS	VSS*
T12	VSS	VSS	VSS	VSS*
T13	VSS	VSS	VSS	VSS*
T14	VSS	VSS	VSS	VSS*
T15	VSS	VSS	VSS	VSS*
T16	VSS	VSS	VSS	VSS*
AA23	VDD	VDD	VDD	VDD
AA4	VDD	VDD	VDD	VDD
AC11	VDD	VDD	VDD	VDD
AC16	VDD	VDD	VDD	VDD
AC21	VDD	VDD	VDD	VDD
AC6	VDD	VDD	VDD	VDD
D11	VDD	VDD	VDD	VDD
D16	VDD	VDD	VDD	VDD
D21	VDD	VDD	VDD	VDD
D6	VDD	VDD	VDD	VDD
F23	VDD	VDD	VDD	VDD
F4	VDD	VDD	VDD	VDD
L23	VDD	VDD	VDD	VDD
L4	VDD	VDD	VDD	VDD
T23	VDD	VDD	VDD	VDD
T4	VDD	VDD	VDD	VDD

*Thermally enhanced connection.

Table 75. OR3C/T80 and OR3T125 432-Pin EBGA Pinout

Pin	OR3C/T80 Pad	OR3T125 Pad	Function
E4	PRD_CFGN	PRD_CFGN	RD_CFG
D3	PR1D	PR1D	I/O
D2	PR1C	PR1C	I/O
D1	PR1B	PR1B	I/O
F4	PR1A	PR1A	I/O
E3	PR2D	PR2D	I/O
E2	PR2C	PR2C	I/O
E1	PR2B	PR2B	I/O
F3	PR2A	PR2A	I/O
F2	PR3D	PR3D	I/O
F1	PR3C	PR3C	I/O
H4	PR3B	PR3B	I/O
G3	PR3A	PR3A	I/O-WR
G2	PR4D	PR4D	I/O
G1	PR4C	PR4C	I/O
J4	PR4B	PR4B	I/O
H3	PR4A	PR4A	I/O
H2	PR5D	PR5A	I/O
J3	PR5C	PR6C	I/O
K4	PR5B	PR6A	I/O
J2	PR5A	PR7A	I/O-RD/MPI_STRB
J1	PR6D	PR8D	I/O
K3	PR6C	PR8C	I/O
K2	PR6B	PR8B	I/O
K1	PR6A	PR8A	I/O
L3	PR7D	PR9D	I/O
M4	PR7C	PR9C	I/O
L2	PR7B	PR9B	I/O
L1	PR7A	PR9A	I/O-CS0
M3	PR8D	PR10D	I/O
N4	PR8A	PR10A	I/O
M2	PR9D	PR11D	I/O
N3	PR9C	PR11A	I/O-CS1
N2	PR9B	PR12D	I/O
P4	PR9A	PR12C	I/O
N1	PR10D	PR12A	I/O
P3	PR10C	PR13D	I/O
P2	PR10B	PR13C	I/O
P1	PR10A	PR13A	I/O
R3	PR11D	PR14D	I/O
R2	PR11C	PR14C	I/O
R1	PR11B	PR14B	I/O
T2	PECKR	PECKR	I-ECKR
T4	PR12D	PR15D	I/O
T3	PR12C	PR15C	I/O
U1	PR12B	PR15B	I/O
U2	PR12A	PR15A	I/O
U3	PR13D	PR16D	I/O
V1	PR13C	PR16B	I/O
V2	PR13B	PR16A	I/O
V3	PR13A	PR17D	I/O
W1	PR14D	PR17A	I/O-M3
V4	PR14C	PR18D	I/O
W2	PR14B	PR18B	I/O
W3	PR14A	PR18A	I/O
Y2	PR15D	PR19D	I/O
W4	PR15A	PR19A	I/O-M2
Y3	PR16D	PR20D	I/O
AA1	PR16C	PR20C	I/O
AA2	PR16B	PR20B	I/O
Y4	PR16A	PR20A	I/O
AA3	PR17D	PR21D	I/O
AB1	PR17C	PR21C	I/O
AB2	PR17B	PR21B	I/O
AB3	PR17A	PR21A	I/O
AC1	PR18D	PR22D	I/O-M1
AC2	PR18C	PR23D	I/O
AB4	PR18B	PR23B	I/O
AC3	PR18A	PR23A	I/O
AD2	PR19D	PR24A	I/O
AD3	PR19C	PR25C	I/O
AC4	PR19B	PR25B	I/O
AE1	PR19A	PR25A	I/O
AE2	PR20D	PR26D	I/O
AE3	PR20C	PR26C	I/O
AD4	PR20B	PR26B	I/O
AF1	PR20A	PR26A	I/O
AF2	PR21D	PR27D	I/O
AF3	PR21C	PR27C	I/O
AG1	PR21B	PR27B	I/O
AG2	PR21A	PR27A	I/O
AG3	PR22D	PR28D	I/O
AF4	PR22C	PR28C	I/O
AH1	PR22B	PR28B	I/O
AH2	PR22A	PR28A	I/O-M0
AH3	PPRGMN	PPRGMN	PRGM
AG4	PRESETN	PRESETN	RESET
AH5	PDONE	PDONE	DONE
AJ4	PB22D	PB28D	I/O
AK4	PB22C	PB28C	I/O
AL4	PB22B	PB28B	I/O
AH6	PB22A	PB28A	I/O
AJ5	PB21D	PB27D	I/O
AK5	PB21C	PB27C	I/O
AL5	PB21B	PB27B	I/O
AJ6	PB21A	PB27A	I/O
AK6	PB20D	PB26D	I/O
AL6	PB20C	PB26C	I/O

Pin	OR3C/T80 Pad	OR3T125 Pad	Function
AH8	PB20B	PB26B	I/O
AJ7	PB20A	PB26A	I/O
AK7	PB19D	PB25D	I/O
AL7	PB19C	PB25C	I/O
AH9	PB19B	PB25B	I/O
AJ8	PB19A	PB25A	I/O
AK8	PB18D	PB24D	I/O
AJ9	PB18C	PB24C	I/O
AH10	PB18B	PB24B	I/O
AK9	PB18A	PB24A	I/O-$\overline{\text{INIT}}$
AL9	PB17D	PB23D	I/O
AJ10	PB17C	PB23C	I/O
AK10	PB17B	PB23A	I/O
AL10	PB17A	PB22A	I/O
AJ11	PB16D	PB21D	I/O
AH12	PB16C	PB21A	I/O
AK11	PB16B	PB20D	I/O
AL11	PB16A	PB20A	I/O-$\overline{\text{LDC}}$
AJ12	PB15D	PB19D	I/O
AH13	PB15B	PB19B	I/O
AK12	PB15A	PB19A	I/O
AJ13	PB14D	PB18D	I/O
AK13	PB14C	PB18B	I/O
AH14	PB14B	PB18A	I/O-HDC
AL13	PB14A	PB17D	I/O
AJ14	PB13D	PB17B	I/O
AK14	PB13C	PB17A	I/O
AL14	PB13B	PB16D	I/O
AJ15	PB13A	PB16A	I/O
AK15	PB12D	PB15D	I/O
AL15	PB12C	PB15C	I/O
AK16	PB12B	PB15B	I/O
AH16	PECKB	PECKB	I-ECKB
AJ16	PB11D	PB14D	I/O
AL17	PB11C	PB14C	I/O
AK17	PB11B	PB14B	I/O
AJ17	PB11A	PB14A	I/O
AL18	PB10D	PB13D	I/O
AK18	PB10C	PB13B	I/O
AJ18	PB10B	PB13A	I/O
AL19	PB10A	PB12D	I/O
AH18	PB9D	PB12A	I/O
AK19	PB9C	PB11D	I/O
AJ19	PB9B	PB11B	I/O
AK20	PB9A	PB11A	I/O
AH19	PB8D	PB10D	I/O
AJ20	PB8B	PB10B	I/O
AL21	PB8A	PB10A	I/O

Pin	OR3C/T80 Pad	OR3T125 Pad	Function
AK21	PB7D	PB9D	I/O
AH20	PB7C	PB9A	I/O
AJ21	PB7B	PB8D	I/O
AL22	PB7A	PB8A	I/O
AK22	PB6D	PB7D	I/O
AJ22	PB6C	PB7A	I/O
AL23	PB6B	PB6D	I/O
AK23	PB6A	PB6A	I/O
AH22	PB5D	PB5D	I/O
AJ23	PB5C	PB5C	I/O
AK24	PB5B	PB5B	I/O
AJ24	PB5A	PB5A	I/O
AH23	PB4D	PB4D	I/O-A17
AL25	PB4C	PB4C	I/O
AK25	PB4B	PB4B	I/O
AJ25	PB4A	PB4A	I/O
AH24	PB3D	PB3D	I/O
AL26	PB3C	PB3C	I/O
AK26	PB3B	PB3B	I/O
AJ26	PB3A	PB3A	I/O
AL27	PB2D	PB2D	I/O
AK27	PB2C	PB2C	I/O
AJ27	PB2B	PB2B	I/O
AH26	PB2A	PB2A	I/O
AL28	PB1D	PB1D	I/O
AK28	PB1C	PB1C	I/O
AJ28	PB1B	PB1B	I/O
AH27	PB1A	PB1A	I/O-A16
AG28	PCCLK	PCCLK	CCLK
AH29	PL22A	PL28A	I/O-A15
AH30	PL22B	PL28B	I/O
AH31	PL22C	PL28C	I/O
AF28	PL22D	PL28D	I/O
AG29	PL21A	PL27A	I/O-SECKLL
AG30	PL21B	PL27B	I/O
AG31	PL21C	PL27C	I/O
AF29	PL21D	PL27D	I/O
AF30	PL20A	PL26A	I/O
AF31	PL20B	PL26B	I/O
AD28	PL20C	PL26C	I/O
AE29	PL20D	PL26D	I/O
AE30	PL19A	PL25A	I/O
AE31	PL19B	PL25B	I/O
AC28	PL19C	PL25C	I/O
AD29	PL19D	PL24A	I/O-A14
AD30	PL18A	PL24D	I/O
AC29	PL18B	PL23D	I/O
AB28	PL18C	PL22C	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
B19	PT9B	PT11D	I/O
D18	PT9C	PT12A	I/O-D0/DIN
A19	PT9D	PT12C	I/O
C18	PT10A	PT12D	I/O
B18	PT10B	PT13A	I/O
A18	PT10C	PT13C	I/O
C17	PT10D	PT13D	I/O-D1
B17	PT11A	PT14A	I/O-D2
A17	PT11B	PT14B	I/O
B16	PT11C	PT14C	I/O
D16	PT11D	PT14D	I/O
C16	PT12A	PT15A	I/O-D3
A15	PT12B	PT15B	I/O
B15	PT12C	PT15C	I/O
C15	PECKT	PECKT	I-ECKT
A14	PT13A	PT16A	I/O-D4
B14	PT13B	PT16B	I/O
C14	PT13C	PT16D	I/O
A13	PT13D	PT17A	I/O
D14	PT14A	PT17B	I/O
B13	PT14B	PT17D	I/O
C13	PT14C	PT18A	I/O-D5
B12	PT14D	PT18B	I/O
D13	PT15A	PT18D	I/O
C12	PT15B	PT19A	I/O
A11	PT15D	PT19D	I/O
B11	PT16A	PT20A	I/O
D12	PT16B	PT20D	I/O-D6
C11	PT16C	PT21A	I/O
A10	PT16D	PT21D	I/O
B10	PT17A	PT22D	I/O
C10	PT17B	PT23B	I/O
A9	PT17C	PT23C	I/O
B9	PT17D	PT23D	I/O
D10	PT18A	PT24A	I/O
C9	PT18B	PT24B	I/O
B8	PT18C	PT24C	I/O
C8	PT18D	PT24D	I/O-D7
D9	PT19A	PT25A	I/O
A7	PT19B	PT25B	I/O
B7	PT19C	PT25C	I/O
C7	PT19D	PT25D	I/O
D8	PT20A	PT26A	I/O
A6	PT20B	PT26B	I/O
B6	PT20C	PT26C	I/O
C6	PT20D	PT26D	I/O
A5	PT21A	PT27A	I/O-RDY/RCLK/MPI_ALE
B5	PT21B	PT27B	I/O
C5	PT21C	PT27C	I/O
D6	PT21D	PT27D	I/O

Pin	OR3C/T80 Pad	OR3T125 Pad	Function
A4	PT22A	PT28A	I/O
B4	PT22B	PT28B	I/O
C4	PT22C	PT28C	I/O
D5	PT22D	PT28D	I/O-SECKUR
A12	VSS	VSS	VSS
A16	VSS	VSS	VSS
A2	VSS	VSS	VSS
A20	VSS	VSS	VSS
A24	VSS	VSS	VSS
A29	VSS	VSS	VSS
A3	VSS	VSS	VSS
A30	VSS	VSS	VSS
A8	VSS	VSS	VSS
AD1	VSS	VSS	VSS
AD31	VSS	VSS	VSS
AJ1	VSS	VSS	VSS
AJ2	VSS	VSS	VSS
AJ30	VSS	VSS	VSS
AJ31	VSS	VSS	VSS
AK1	VSS	VSS	VSS
AK29	VSS	VSS	VSS
AK3	VSS	VSS	VSS
AK31	VSS	VSS	VSS
AL12	VSS	VSS	VSS
AL16	VSS	VSS	VSS
AL2	VSS	VSS	VSS
AL20	VSS	VSS	VSS
AL24	VSS	VSS	VSS
AL29	VSS	VSS	VSS
AL3	VSS	VSS	VSS
AL30	VSS	VSS	VSS
AL8	VSS	VSS	VSS
B1	VSS	VSS	VSS
B29	VSS	VSS	VSS
B3	VSS	VSS	VSS
B31	VSS	VSS	VSS
C1	VSS	VSS	VSS
C2	VSS	VSS	VSS
C30	VSS	VSS	VSS
C31	VSS	VSS	VSS
H1	VSS	VSS	VSS
H31	VSS	VSS	VSS
M1	VSS	VSS	VSS
M31	VSS	VSS	VSS
T1	VSS	VSS	VSS
T31	VSS	VSS	VSS
Y1	VSS	VSS	VSS
Y31	VSS	VSS	VSS
A1	VDD	VDD	VDD
A31	VDD	VDD	VDD

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

Package Thermal Characteristics

There are four thermal parameters that are in common use: ΘJA, ΨJC, ΘJC, and ΘJB. It should be noted that all the parameters are affected, to varying degrees, by package design (including paddle size) and choice of materials, the amount of copper in the test board or system board, and system airflow.

Table 76 contains the currently available thermal specifications for FPGA packages mounted on both JEDEC and non-JEDEC test boards. The thermal values for the newer package types correspond to those packages mounted on a JEDEC four-layer board. The values for the older packages, however, correspond to those packages mounted on a non-JEDEC, single-layer, sparse copper board (see Note 2). It should also be noted that the values for the older packages are considered conservative.

ΘJA

This is the thermal resistance from junction to ambient (a.k.a. theta-JA, R-theta, etc.).

$$\Theta JA = \frac{TJ - TA}{Q}$$

where TJ is the junction temperature, TA is the ambient air temperature, and Q is the chip power.

Experimentally, ΘJA is determined when a special thermal test die is assembled into the package of interest, and the part is mounted on the thermal test board. The diodes on the test chip are separately calibrated in an oven. The package/board is placed either in a JEDEC natural convection box or in the wind tunnel, the latter for forced convection measurements. A controlled amount of power (Q) is dissipated in the test chip's heater resistor, the chip's temperature (TJ) is determined by the forward drop on the diodes, and the ambient temperature (TA) is noted. Note that ΘJA is expressed in units of °C/watt.

ψJC

This JEDEC designated parameter correlates the junction temperature to the case temperature. It is generally used to infer the junction temperature while the device is operating in the system. It is not considered a true thermal resistance, and it is defined by:

$$\psi_{JC} = \frac{T_J - T_C}{Q}$$

where T_C is the case temperature at top dead center, T_J is the junction temperature, and Q is the chip power. During the Θ_{JA} measurements described above, besides the other parameters measured, an additional temperature reading, T_C, is made with a thermocouple attached at top-dead-center of the case. ψ_{JC} is also expressed in units of °C/watt.

ΘJC

This is the thermal resistance from junction to case. It is most often used when attaching a heat sink to the top of the package. It is defined by:

$$\Theta_{JC} = \frac{T_J - T_C}{Q}$$

The parameters in this equation have been defined above. However, the measurements are performed with the case of the part pressed against a water-cooled heat sink so as to draw most of the heat generated by the chip out the top of the package. It is this difference in the measurement process that differentiates Θ_{JC} from ψ_{JC}. Θ_{JC} is a true thermal resistance and is expressed in units of °C/watt.

ΘJB

This is the thermal resistance from junction to board (a.k.a. Θ_{JL}). It is defined by:

$$\Theta_{JB} = \frac{T_J - T_B}{Q}$$

where T_B is the temperature of the board adjacent to a lead measured with a thermocouple. The other parameters on the right-hand side have been defined above. This is considered a true thermal resistance, and the measurement is made with a water-cooled heat sink pressed against the board so as to draw most of the heat out of the leads. Note that Θ_{JB} is expressed in units of °C/watt, and that this parameter and the way it is measured is still in JEDEC committee.

Package Thermal Characteristics (continued)

FPGA Maximum Junction Temperature

Once the power dissipated by the FPGA has been determined (see the Estimating Power Dissipation section), the maximum junction temperature of the FPGA can be found. This is needed to determine if speed derating of the device from the 85 °C junction temperature used in all of the delay tables is needed. Using the maximum ambient temperature, T_{Amax}, and the power dissipated by the device, Q (expressed in °C), the maximum junction temperature is approximated by:

$$T_{Jmax} = T_{Amax} + (Q \bullet \Theta_{JA})$$

Table 76 lists the plastic package thermal characteristics for the *ORCA* Series FPGAs.

Table 76. Plastic Package Thermal Characteristics for the *ORCA* Series[1]

Package	Θ_{JA} (°C/W)			T_A = 70 °C max T_J = 125 °C max @ 0 fpm (W)
	0 fpm	200 fpm	500 fpm	
144-Pin TQFP[1]	52.0	39.0	–	1.1
208-Pin SQFP[1]	26.5	23.0	21.0	2.1
208-Pin SQFP2[1]	12.8	10.3	9.1	4.3
240-Pin SQFP[1]	25.5	22.5	21.0	2.2
240-Pin SQFP2[1]	13.0	10.0	9.0	4.2
256-Pin PBGA[1, 2]	22.5	19.0	17.5	2.4
256-Pin PBGA[1, 3]	26.0	22.0	20.5	2.1
352-Pin PBGA[1, 2]	19.0	16.0	15.0	2.9
352-Pin PBGA[1, 3]	25.5	22.0	20.5	2.1
432-Pin EBGA[1]	11.0	8.5	7.5	5.0

1. Mounted on 4-layer JEDEC standard test board with two power/ground planes.
2. With thermal balls connected to board ground plane.
3. Without thermal balls connected to board ground plane.

Package Coplanarity

The coplanarity limits of the *ORCA* Series 3 packages are as follows.

Table 77. Package Coplanarity

Package Type	Coplanarity Limit (mils)
EBGA	8.0
PBGA	8.0
SQFP/SQFP2	4.0 3.15
TQFP	3.15

Package Parasitics

The electrical performance of an IC package, such as signal quality and noise sensitivity, is directly affected by the package parasitics. Table 78 lists eight parasitics associated with the *ORCA* packages. These parasitics represent the contributions of all components of a package, which include the bond wires, all internal package routing, and the external leads.

Four inductances in nH are listed: L_{SW} and L_{SL}, the self-inductance of the lead; and L_{MW} and L_{ML}, the mutual inductance to the nearest neighbor lead. These parameters are important in determining ground bounce noise and inductive crosstalk noise. Three capacitances in pF are listed: C_M, the mutual capacitance of the lead to the nearest neighbor lead; and C_1 and C_2, the total capacitance of the lead to all other leads (all other leads are assumed to be grounded). These parameters are important in determining capacitive crosstalk and the capacitive loading effect of the lead. The lead resistance value, R_W, is in MΩ.

The parasitic values in Table 78 are for the circuit model of bond wire and package lead parasitics. If the mutual capacitance value is not used in the designer's model, then the value listed as mutual capacitance should be added to each of the C_1 and C_2 capacitors.

Table 78. Package Parasitics

Package Type	L_{SW}	L_{MW}	R_W	C_1	C_2	C_M	L_{SL}	L_{ML}
144-Pin TQFP	3	1	140	1	1	0.6	4—6	2—2.5
208-Pin SQFP	4	2	200	1	1	1	7—10	4—6
208-Pin SQFP2	4	2	200	1	1	1	6—9	4—6
240-Pin SQFP	4	2	200	1	1	1	8—12	5—8
240-Pin SQFP2	4	2	200	1	1	1	7—11	4—7
256-Pin PBGA	5	2	220	1	1	1	5—8	2—4
352-Pin PBGA	5	2	220	1.5	1.5	1.5	7—12	3—6
432-Pin EBGA	4	1.5	500	1	1	0.3	3—5.5	0.5—1

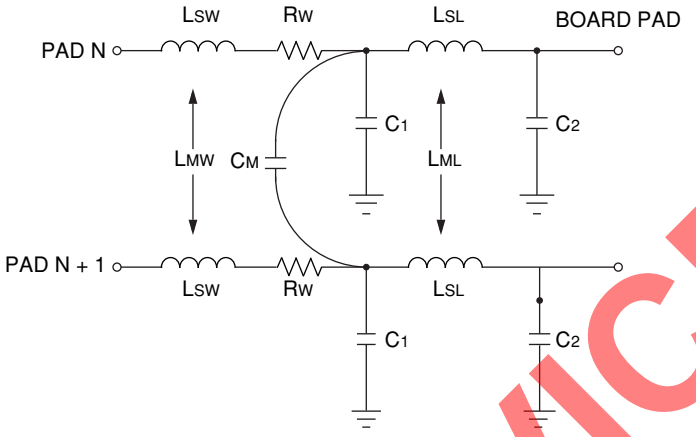


Figure 104. Package Parasitics

Package Outline Diagrams

Terms and Definitions

Basic Size (BSC): The basic size of a dimension is the size from which the limits for that dimension are derived by the application of the allowance and the tolerance.

Design Size: The design size of a dimension is the actual size of the design, including an allowance for fit and tolerance.

Typical (TYP): When specified after a dimension, this indicates the repeated design size if a tolerance is specified or repeated basic size if a tolerance is not specified.

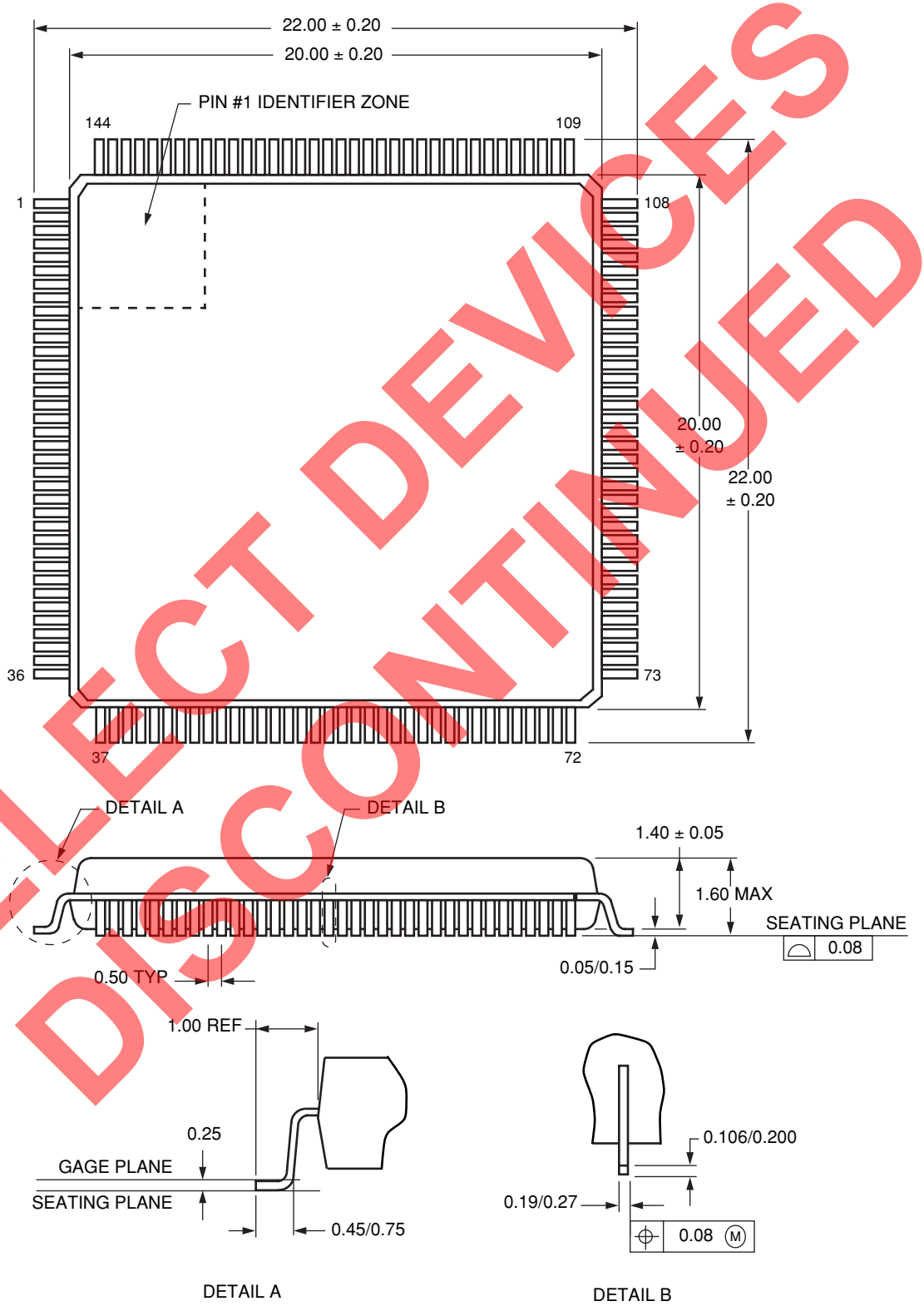
Reference (REF): The reference dimension is an untoleranced dimension used for informational purposes only. It is a repeated dimension or one that can be derived from other values in the drawing.

Minimum (MIN) or Maximum (MAX): Indicates the minimum or maximum allowable size of a dimension.

Package Outline Diagrams (continued)

144-Pin TQFP

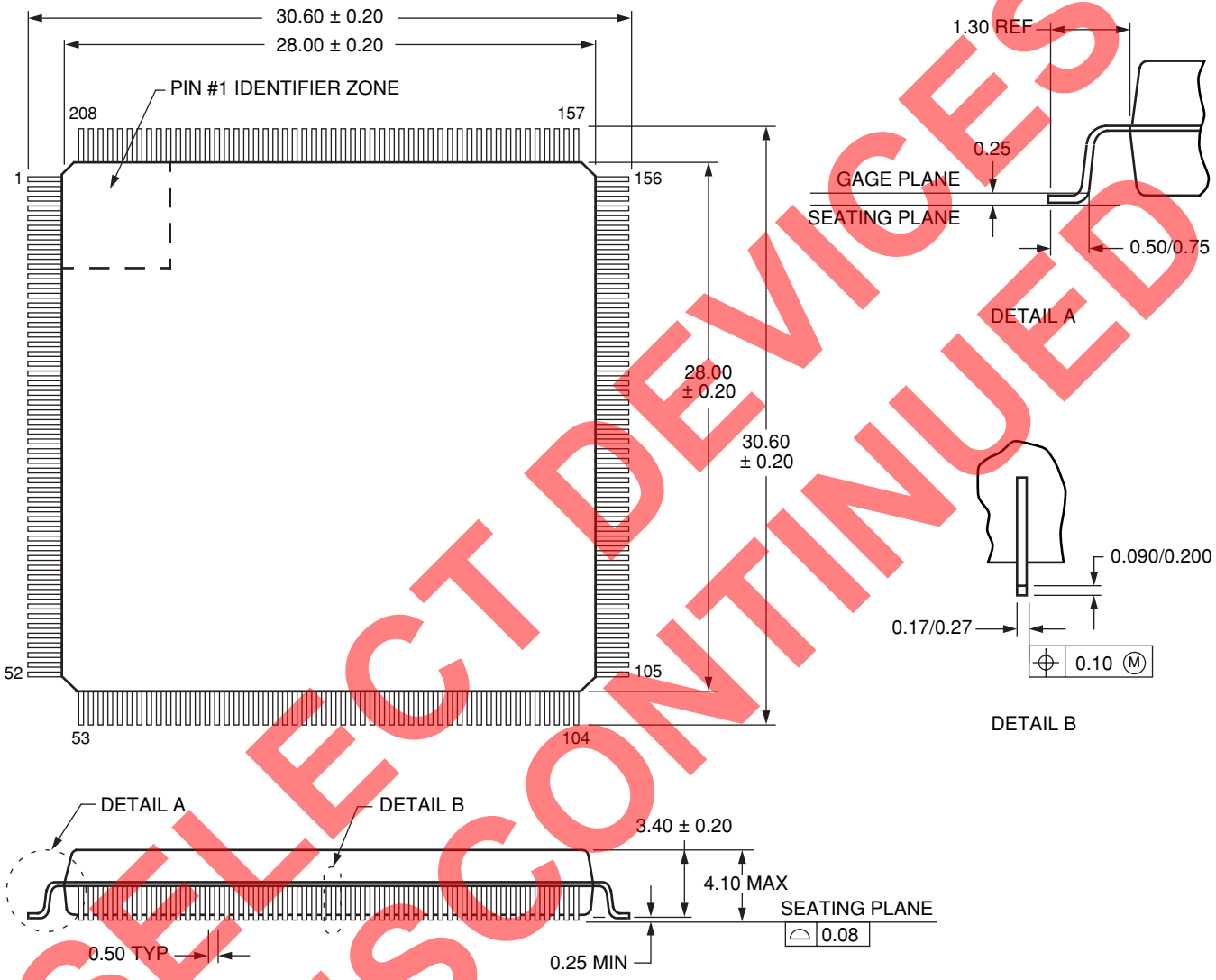
Dimensions are in millimeters.



Package Outline Diagrams (continued)

208-Pin SQFP

Dimensions are in millimeters.

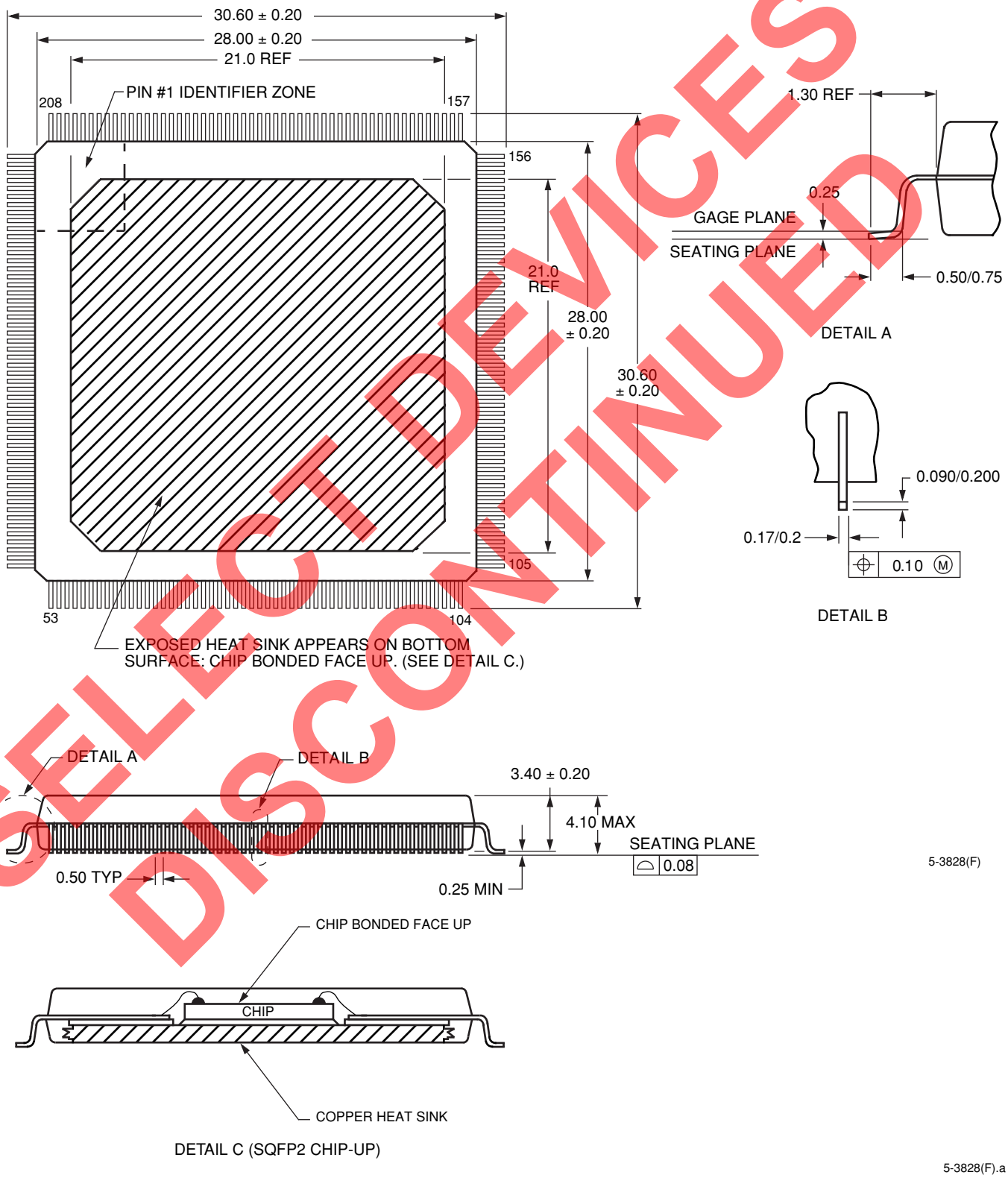


Note: The dimensions in this outline diagram are intended for informational purposes only.

Package Outline Diagrams (continued)

208-Pin SQFP2

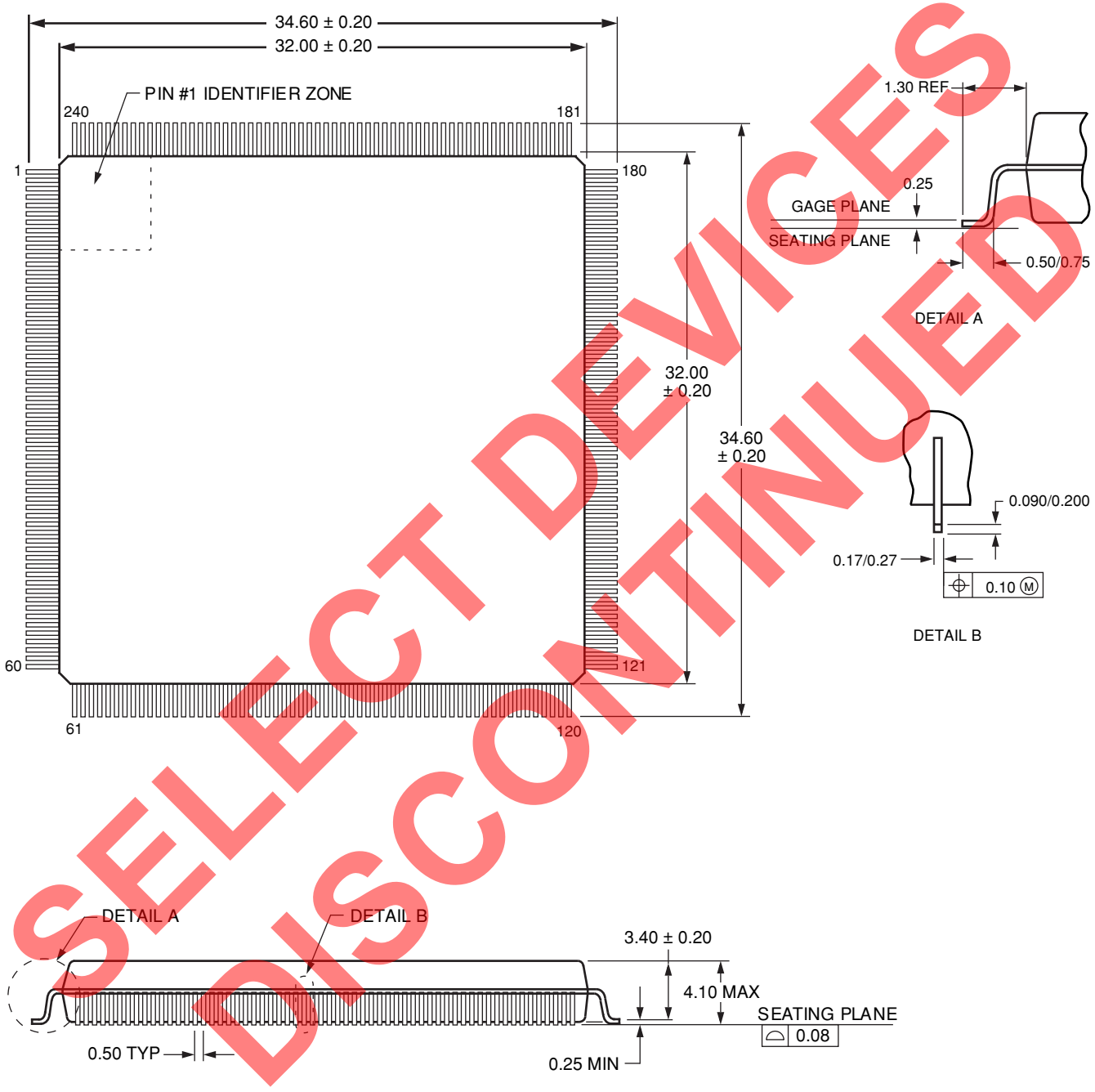
Dimensions are in millimeters.



Package Outline Diagrams (continued)

240-Pin SQFP

Dimensions are in millimeters.

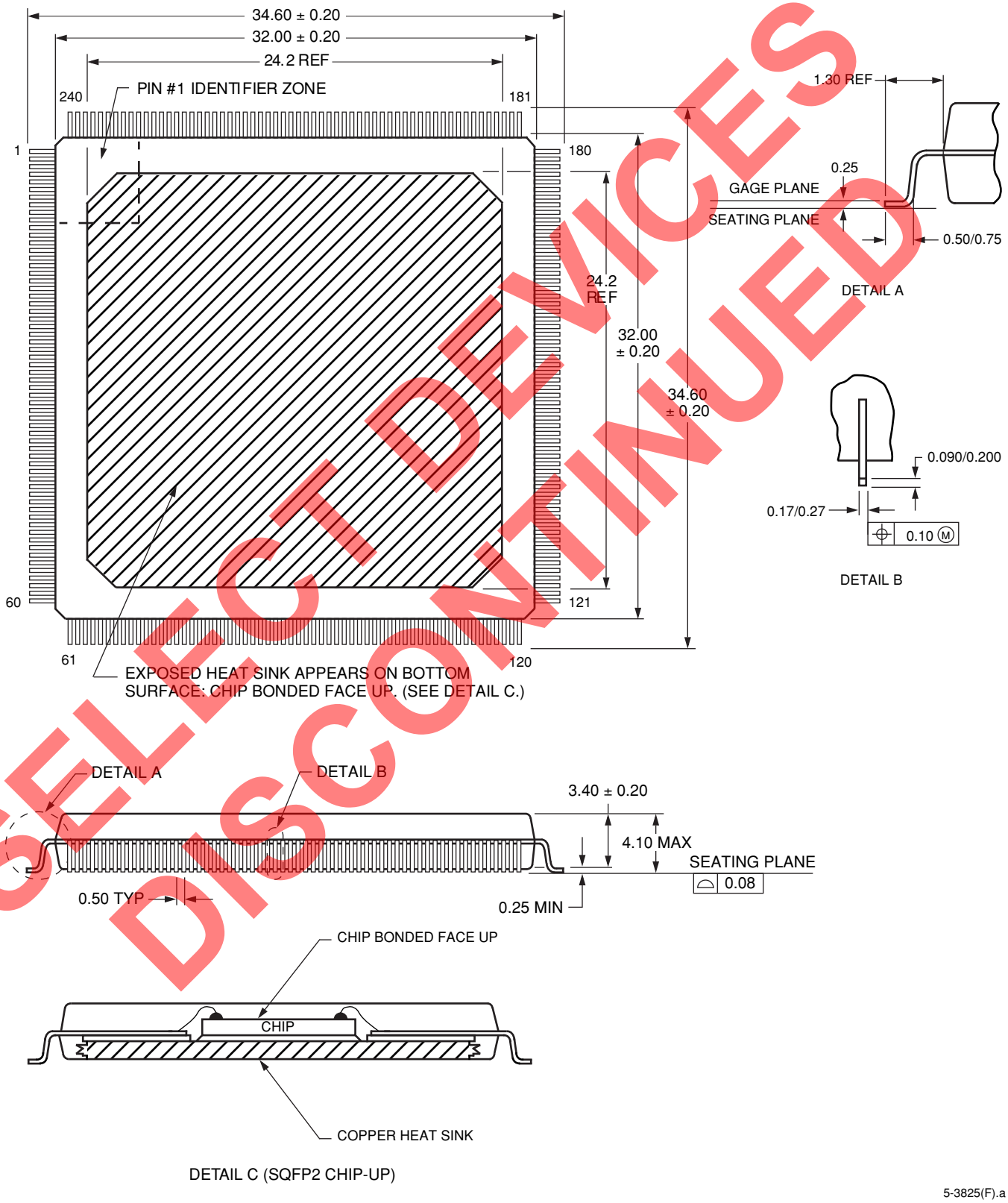


Note: The dimensions in this outline diagram are intended for informational purposes only.

Package Outline Diagrams (continued)

240-Pin SQFP2

Dimensions are in millimeters.



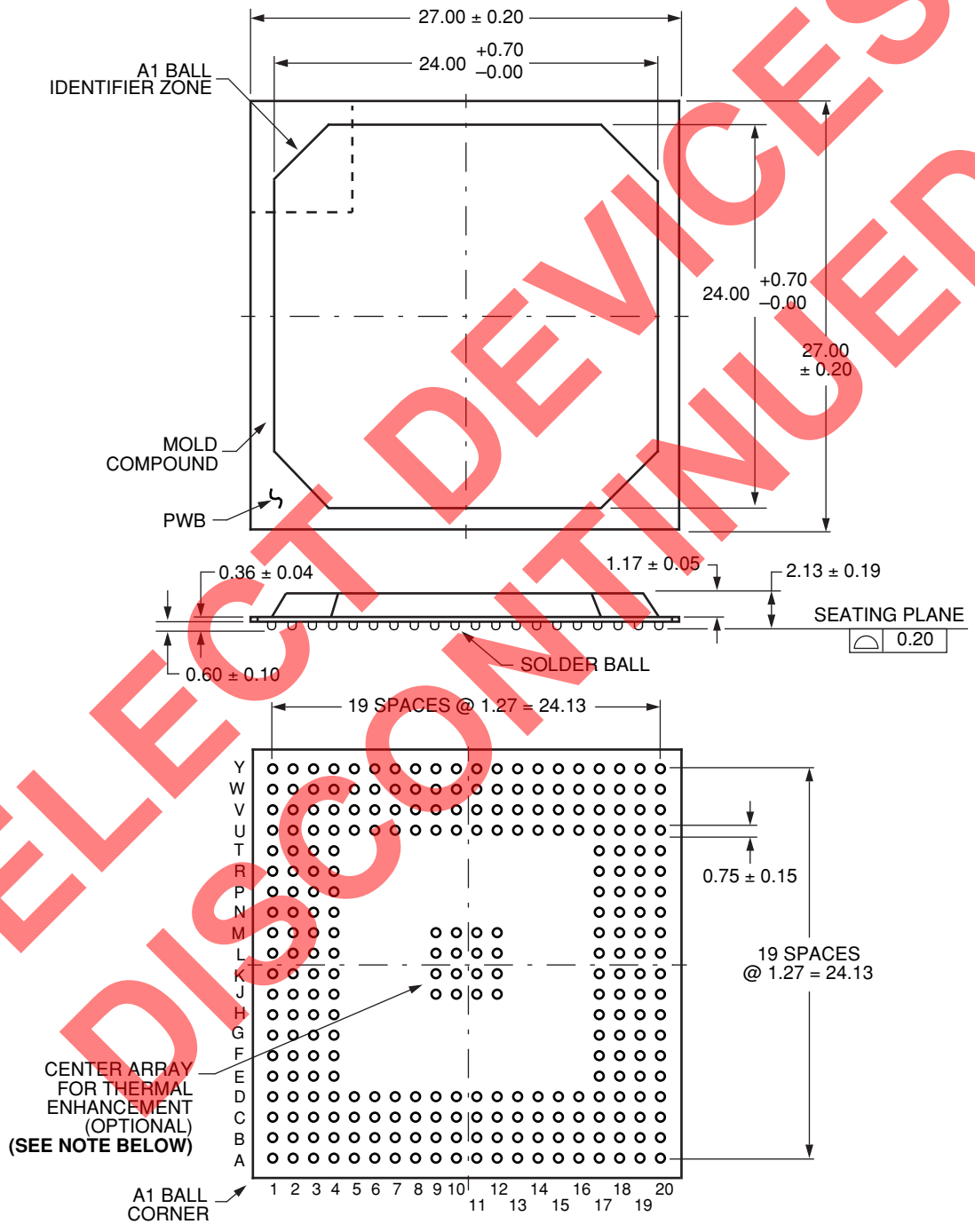
DETAIL C (SQFP2 CHIP-UP)

5-3825(F).a

Package Outline Diagrams (continued)

256-Pin PBGA

Dimensions are in millimeters.



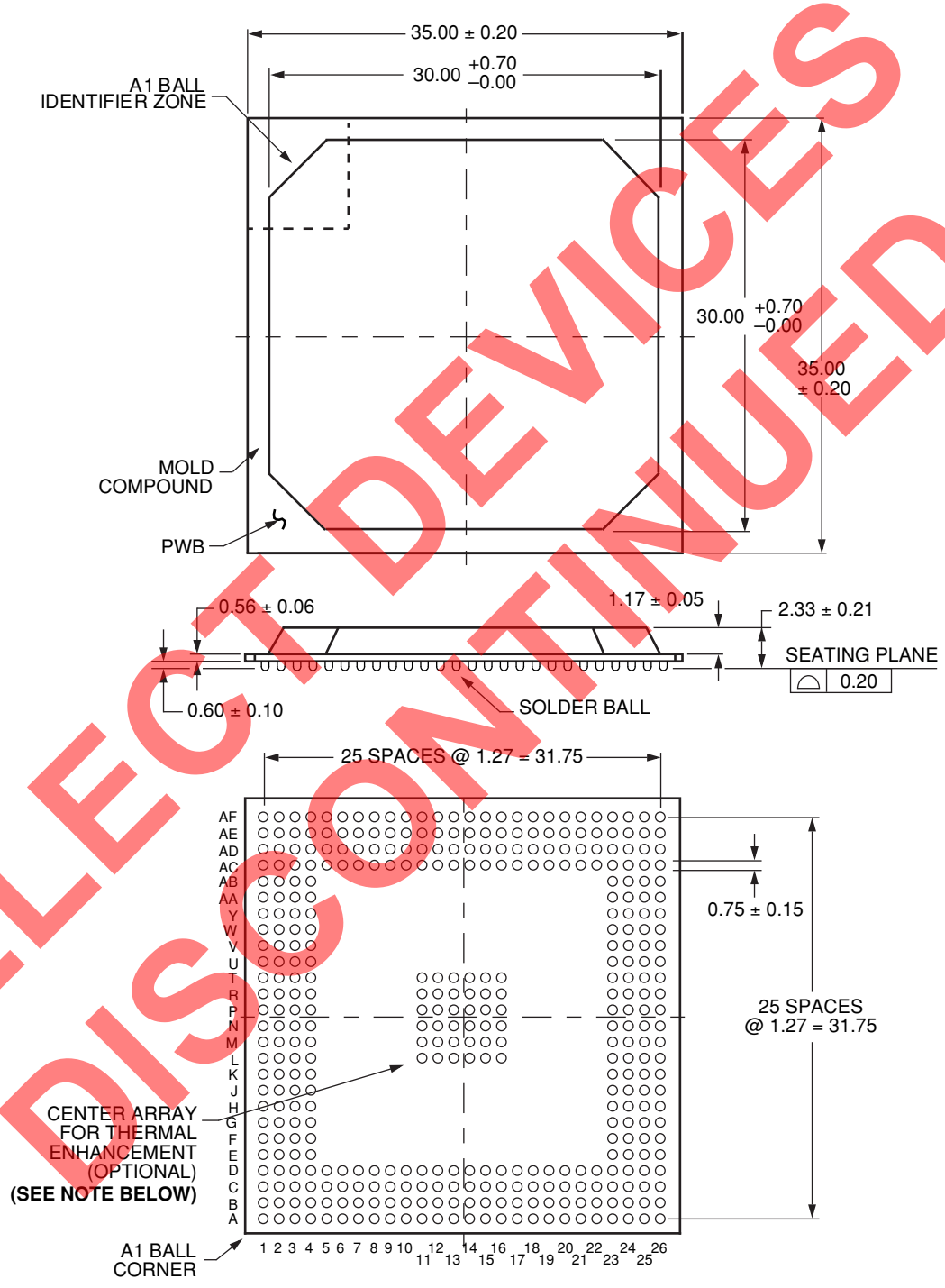
5-4406(F)

Note: Although the 16 thermal enhancement balls are stated as an option, they are standard on the 256 FPGA package.

Package Outline Diagrams (continued)

352-Pin PBGA

Dimensions are in millimeters.



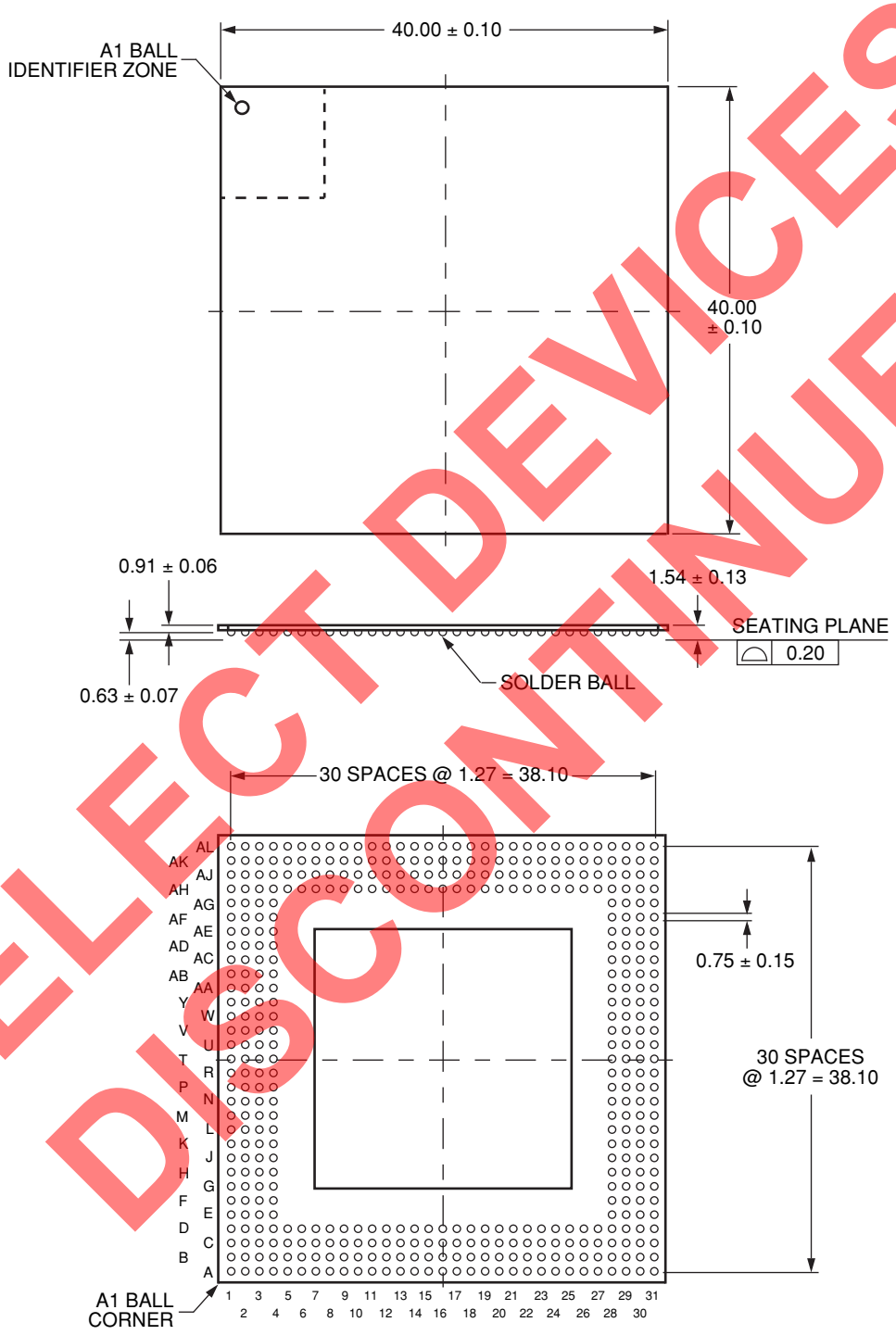
5-4407(F)

Note: Although the 36 thermal enhancement balls are stated as an option, they are standard on the 352 FPGA package.

Package Outline Diagrams (continued)

432-Pin EBGA

Dimensions are in millimeters.



5-4409(F)

Ordering Information

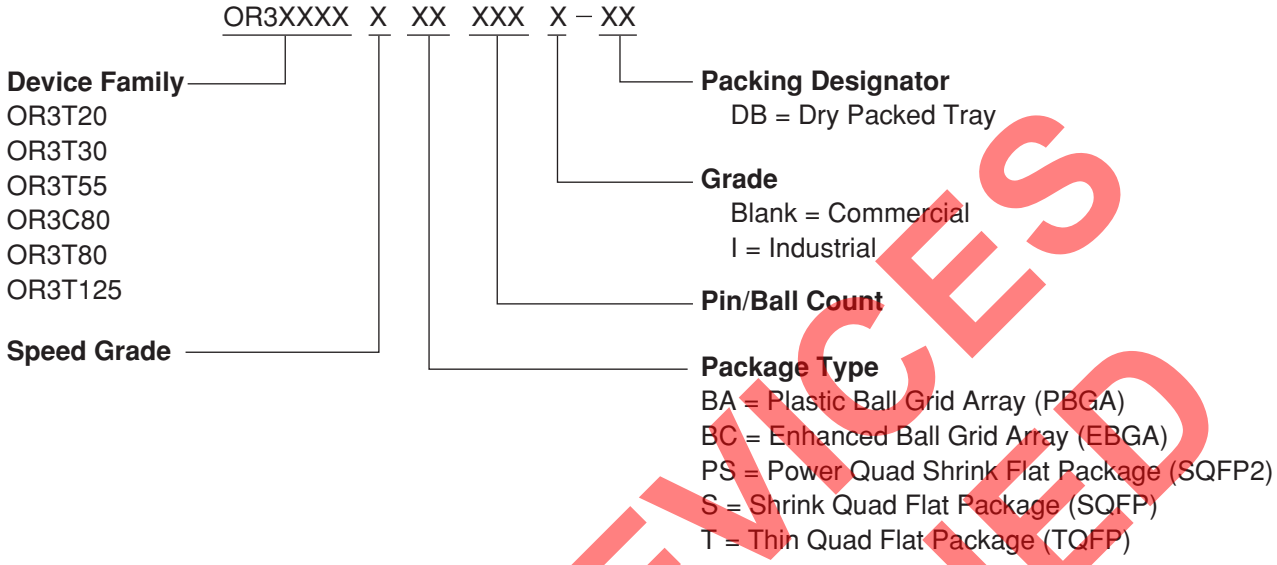


Table 79. Ordering Information

Commercial

Device Family	Part Number	Speed Grade	Package Type	Pin/Ball Count	Grade	Packing Designator
OR3C80	OR3C805PS208-DB[2]	5	SQFP2	208	C	DB
	OR3C805BA352-DB[2]	5	PBGA	352	C	DB
	OR3C804PS208-DB[2]	4	SQFP2	208	C	DB
	OR3C804BA352-DB[2]	4	PBGA	352	C	DB
OR3T20	OR3T207S208-DB	7	SQFP	208	C	DB
	OR3T207BA256-DB	7	PBGA	256	C	DB
	OR3T206S208-DB	6	SQFP	208	C	DB
	OR3T206T144-DB	6	TQFP	144	C	DB
	OR3T206BA256-DB	6	PBGA	256	C	DB
OR3T30	OR3T307S208-DB	7	SQFP	208	C	DB
	OR3T307S240-DB	7	SQFP	240	C	DB
	OR3T307BA256-DB	7	PBGA	256	C	DB
	OR3T306S208-DB	6	SQFP	208	C	DB
	OR3T306S240-DB	6	SQFP	240	C	DB
	OR3T306BA256-DB	6	PBGA	256	C	DB

Commercial

Device Family	Part Number	Speed Grade	Package Type	Pin/Ball Count	Grade	Packing Designator
OR3T55	OR3T557PS208-DB[1]	7	SQFP2	208	C	DB
	OR3T557S208-DB	7	SQFP	208	C	DB
	OR3T557PS240-DB[3]	7	SQFP2	240	C	DB
	OR3T557BA256-DB	7	PBGA	256	C	DB
	OR3T557BA352-DB	7	PBGA	352	C	DB
	OR3T556PS208-DB[1]	6	SQFP2	208	C	DB
	OR3T556S208-DB	6	SQFP	208	C	DB
	OR3T556PS240-DB[3]	6	SQFP2	240	C	DB
	OR3T556BA256-DB	6	PBGA	256	C	DB
	OR3T556BA352-DB	6	PBGA	352	C	DB
OR3T80	OR3T807PS208-DB[1]	7	SQFP2	208	C	DB
	OR3T807S208-DB	7	SQFP	208	C	DB
	OR3T807PS240-DB[3]	7	SQFP2	240	C	DB
	OR3T807BA352-DB	7	PBGA	352	C	DB
	OR3T807BC432-DB	7	EBGA	432	C	DB
	OR3T806PS208-DB[1]	6	SQFP2	208	C	DB
	OR3T806S208-DB	6	SQFP	208	C	DB
	OR3T806PS240-DB[3]	6	SQFP2	240	C	DB
	OR3T806BA352-DB	6	PBGA	352	C	DB
	OR3T806BC432-DB	6	EBGA	432	C	DB
OR3T125	OR3T1257PS208-DB[3]	7	SQFP2	208	C	DB
	OR3T1257PS240-DB[3]	7	SQFP2	240	C	DB
	OR3T1257BA352-DB	7	PBGA	352	C	DB
	OR3T1257BC432-DB	7	EBGA	432	C	DB
	OR3T1256PS208-DB[3]	6	SQFP2	208	C	DB
	OR3T1256PS240-DB[3]	6	SQFP2	240	C	DB
	OR3T1256BA352-DB	6	PBGA	352	C	DB
	OR3T1256BC432-DB	6	EBGA	432	C	DB

Industrial

Device Family	Part Number	Speed Grade	Package Type	Pin/Ball Count	Grade	Packing Designator
OR3C80	OR3C804PS208I-DB[2]	4	SQFP2	208	I	DB
	OR3C804BA352I-DB[2]	4	PBGA	352	I	DB
OR3T20	OR3T206S208I-DB	6	SQFP	208	I	DB
OR3T30	OR3T306S208I-DB	6	SQFP	208	I	DB
	OR3T306S240I-DB	6	SQFP	240	I	DB
	OR3T306BA256I-DB	6	PBGA	256	I	DB
OR3T55	OR3T556PS208I-DB[1]	6	SQFP2	208	I	DB
	OR3T556S208I-DB	6	SQFP	208	I	DB
	OR3T556PS240I-DB[3]	6	SQFP2	240	I	DB
	OR3T556BA256I-DB	6	PBGA	256	I	DB
	OR3T556BA352I-DB	6	PBGA	352	I	DB
OR3T80	OR3T806PS208I-DB[1]	6	SQFP2	208	I	DB
	OR3T806S208I-DB	6	SQFP	208	I	DB
	OR3T806PS240I-DB[3]	6	SQFP2	240	I	DB
	OR3T806BA352I-DB	6	PBGA	352	I	DB
	OR3T806BC432I-DB	6	EBGA	432	I	DB
OR3T125	OR3T1256PS208I-DB[3]	6	SQFP2	208	I	DB
	OR3T1256PS240I-DB[3]	6	SQFP2	240	I	DB
	OR3T1256BA352I-DB	6	PBGA	352	I	DB
	OR3T1256BC432I-DB	6	EBGA	432	I	DB

1. Converted to S208 package device per PCN#11A-06.
2. Discontinued per PCN#02-06. Contact Rochester Electronics for available inventory.
2. Discontinued per PCN#06-07. Contact Rochester Electronics for available inventory.