Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

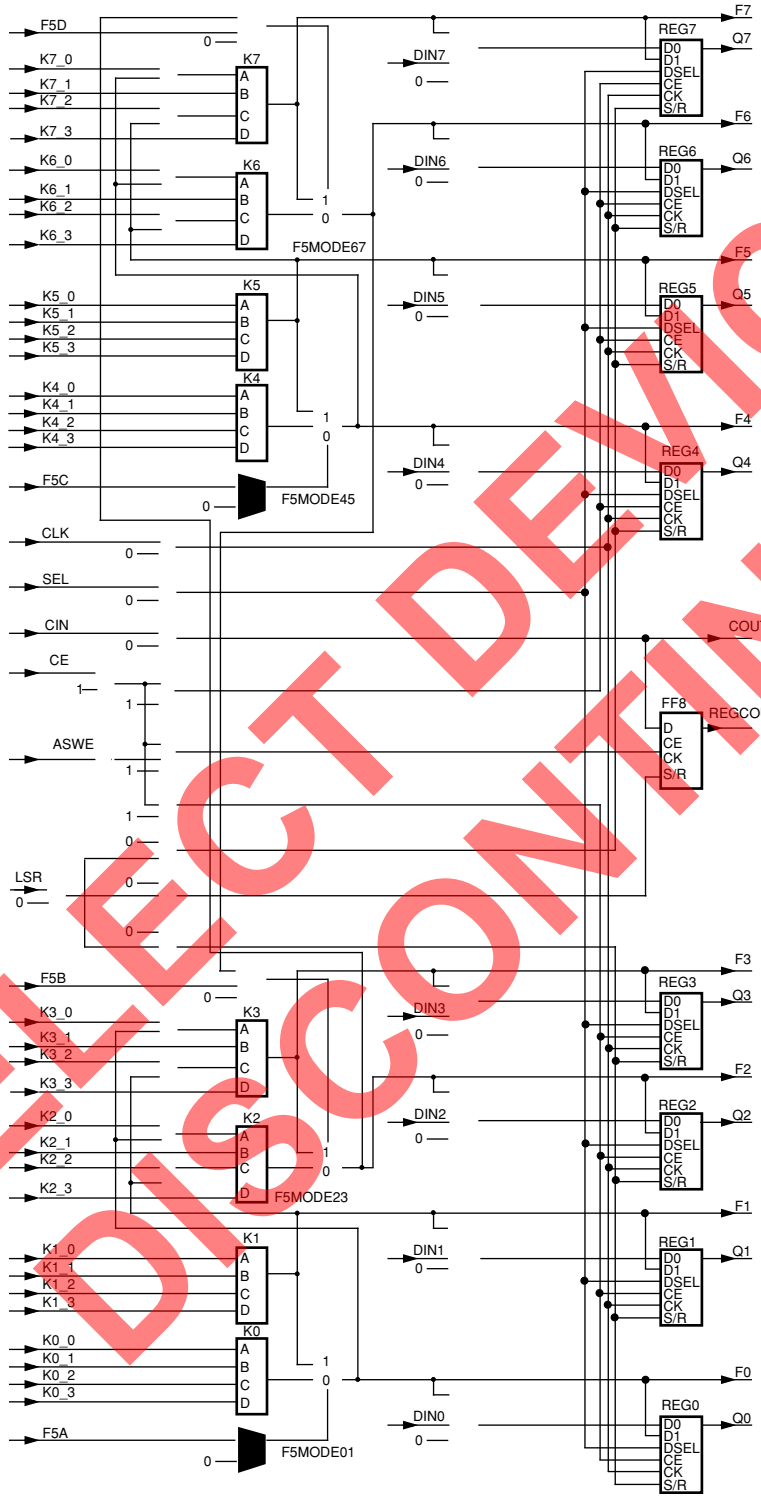
Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling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

Applications of Embedded - FPGAs

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

Details	
Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	2592
Total RAM Bits	43008
Number of I/O	223
Number of Gates	80000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 85°C (TA)
Package / Case	256-BGA
Supplier Device Package	256-FPBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/lattice-semiconductor/or3t556ba256i-db

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)

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.

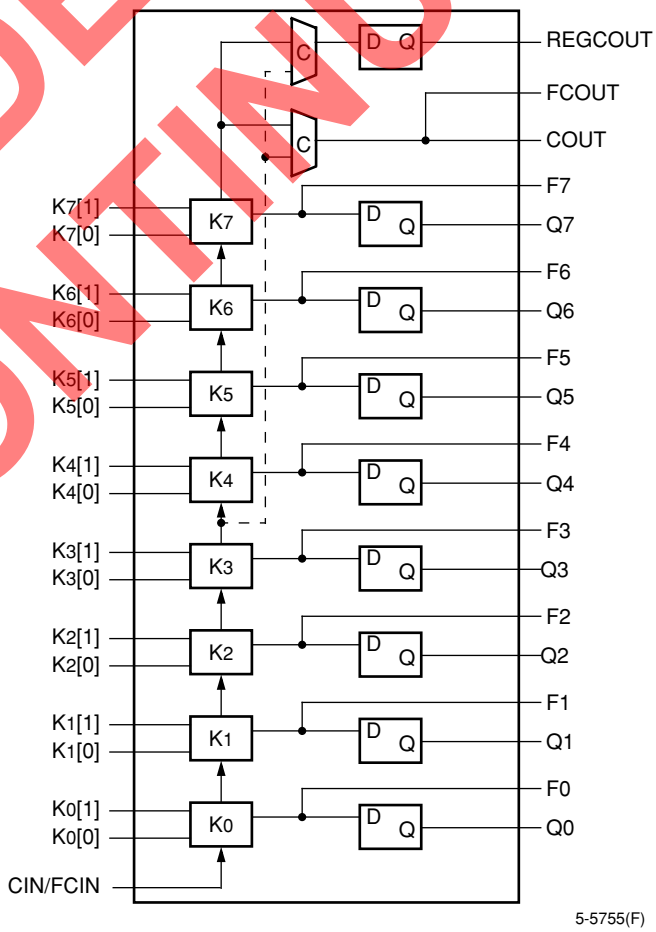


Figure 6. Ripple Mode

Programmable Logic Cells (continued)

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.

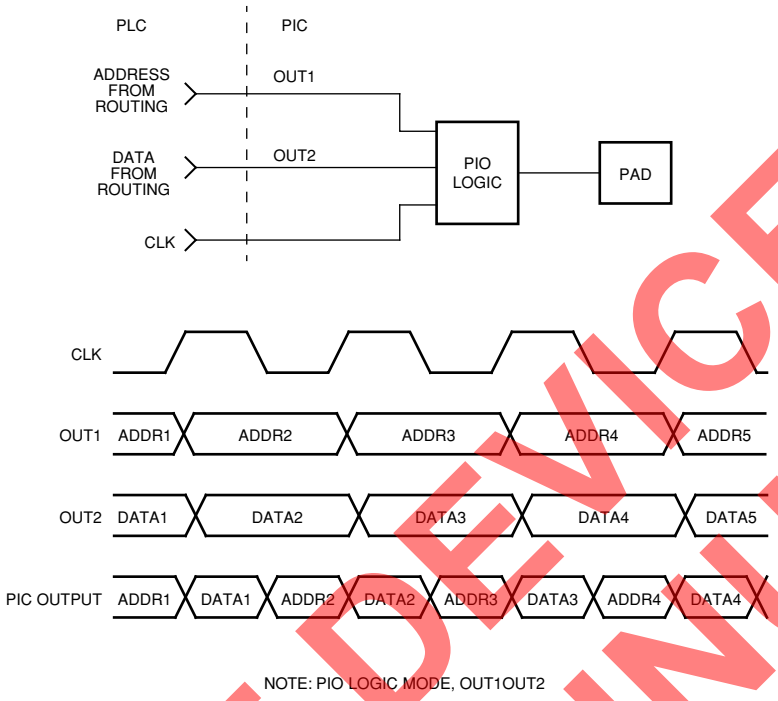
Switching Routing Segments (xSW)

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.

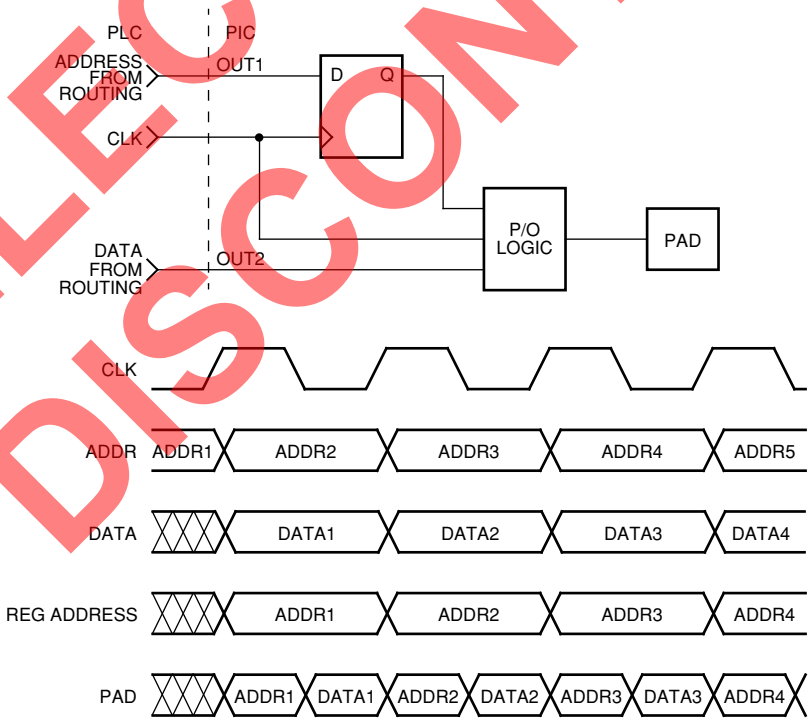
Programmable Input/Output Cells (continued)



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.

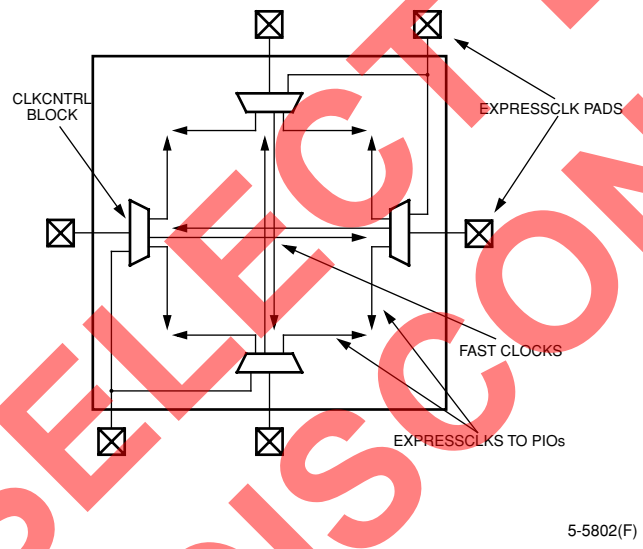
Table 11. PIO Register Control Signals

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.

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 (continued)

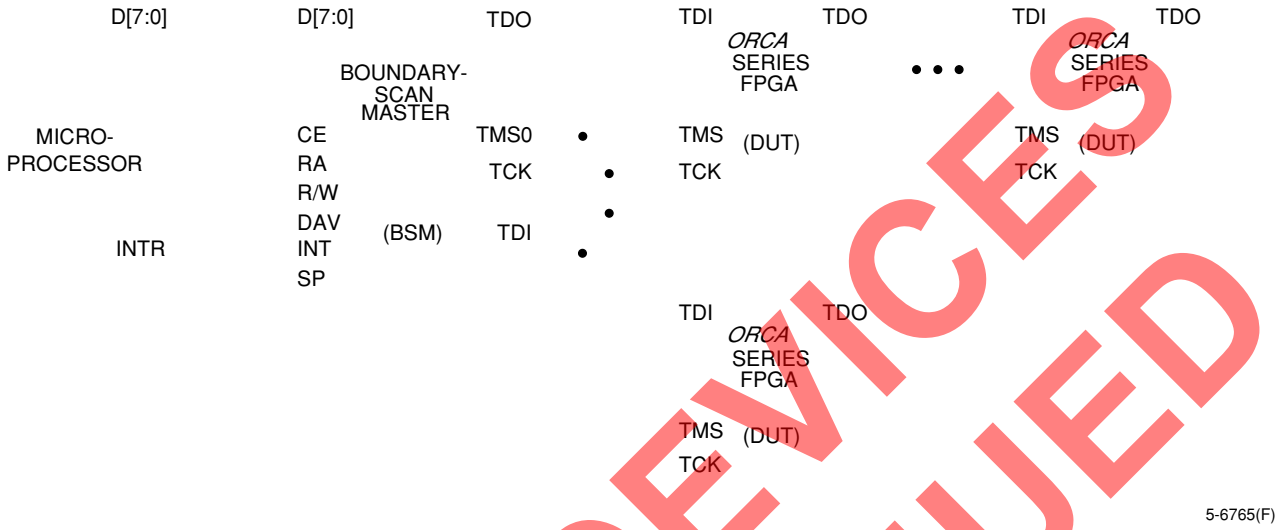


Figure 37. Boundary-Scan Interface

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)

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) (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.

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.

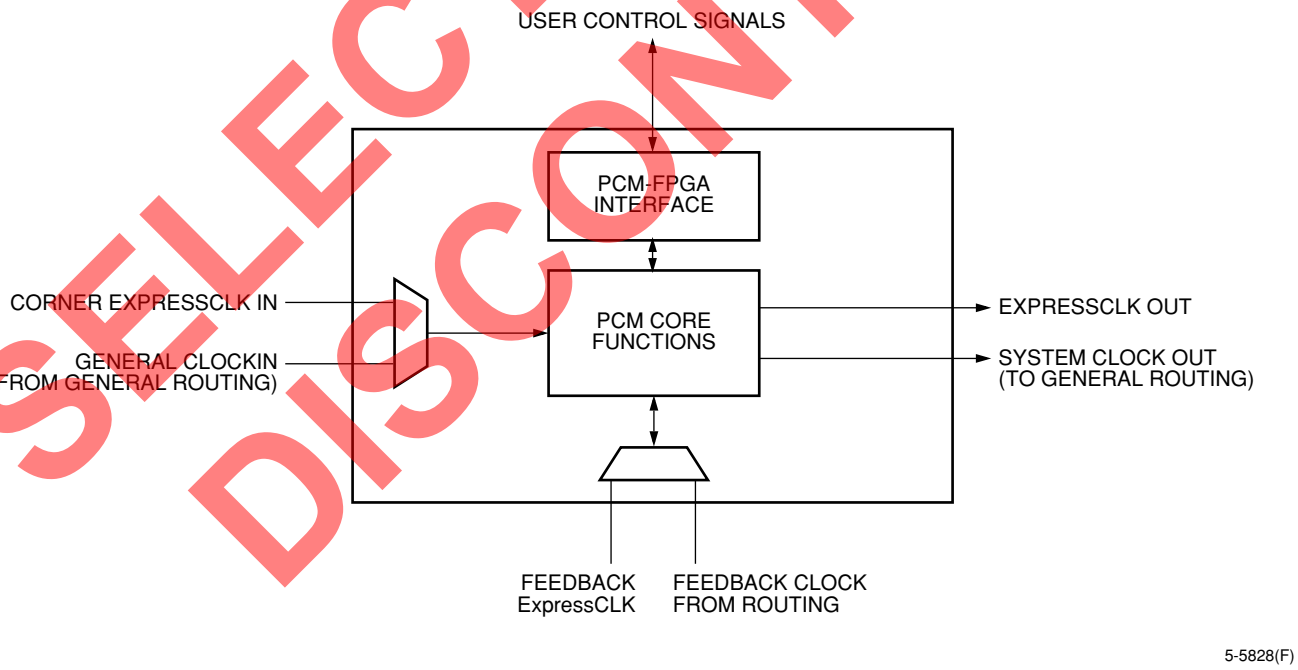


Figure 45. PCM Block Diagram

Programmable Clock Manager (PCM) (continued)

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**.

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)

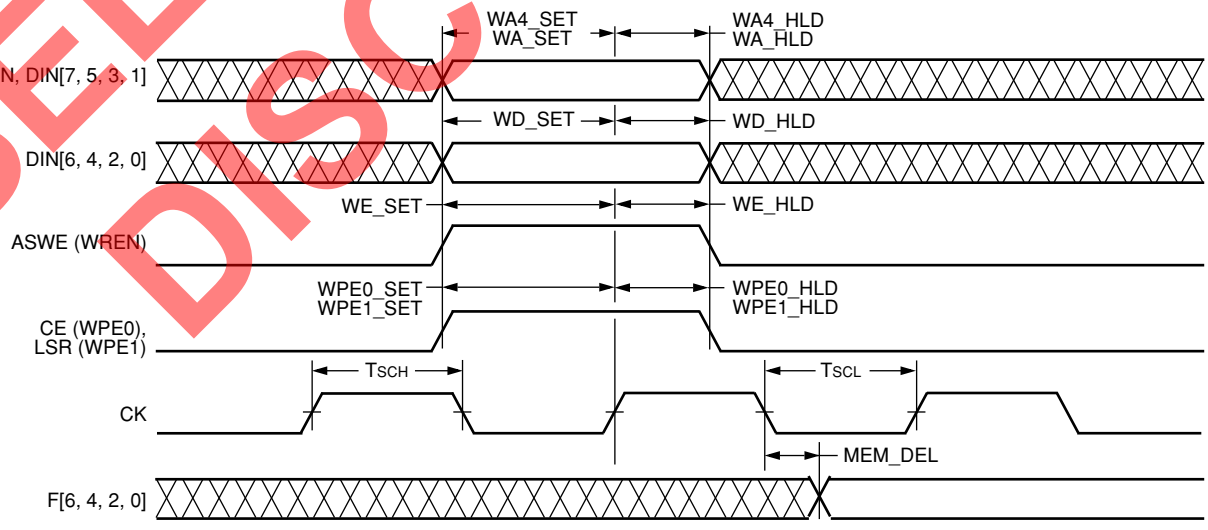
Table 44. Synchronous Memory Write Characteristics

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)

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)

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

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

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

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

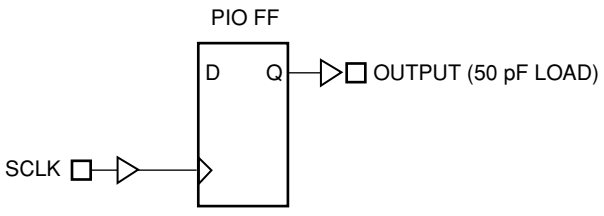
Timing Characteristics (continued)

Table 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

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	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

Package Outline Diagrams (continued)

208-Pin SQFP2

Dimensions are in millimeters.

5-3828(F)

DETAIL C (SQFP2 CHIP-UP)

5-3828(F).a