Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

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

Table of Contents

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.

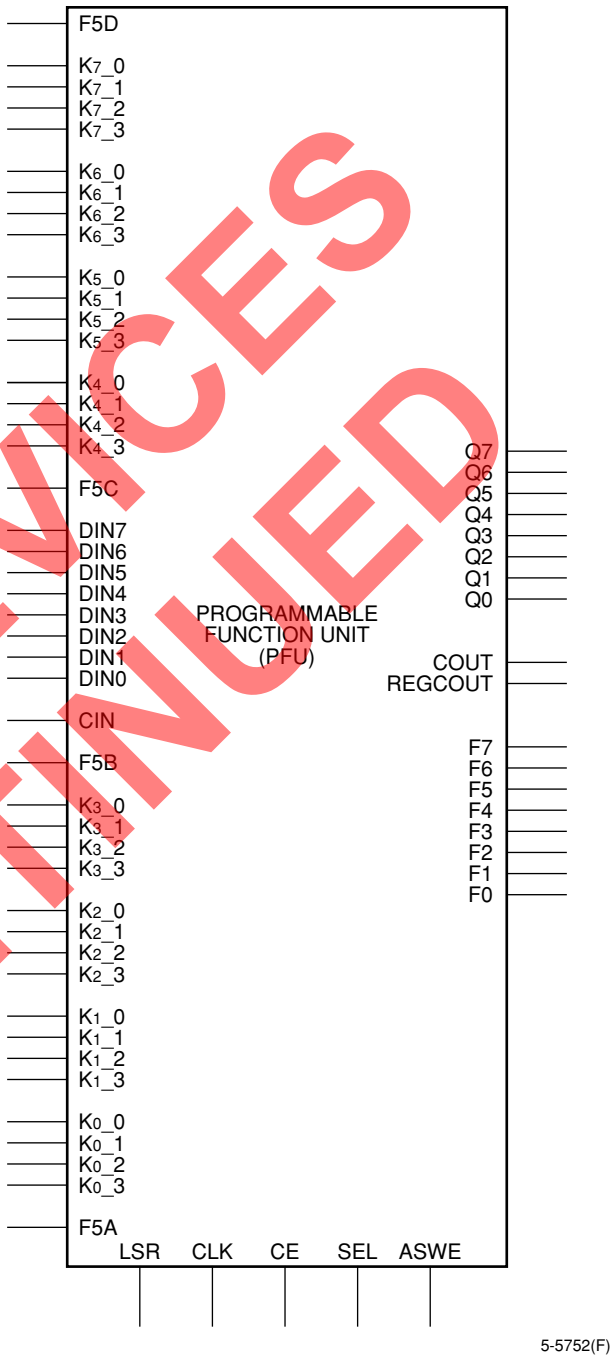
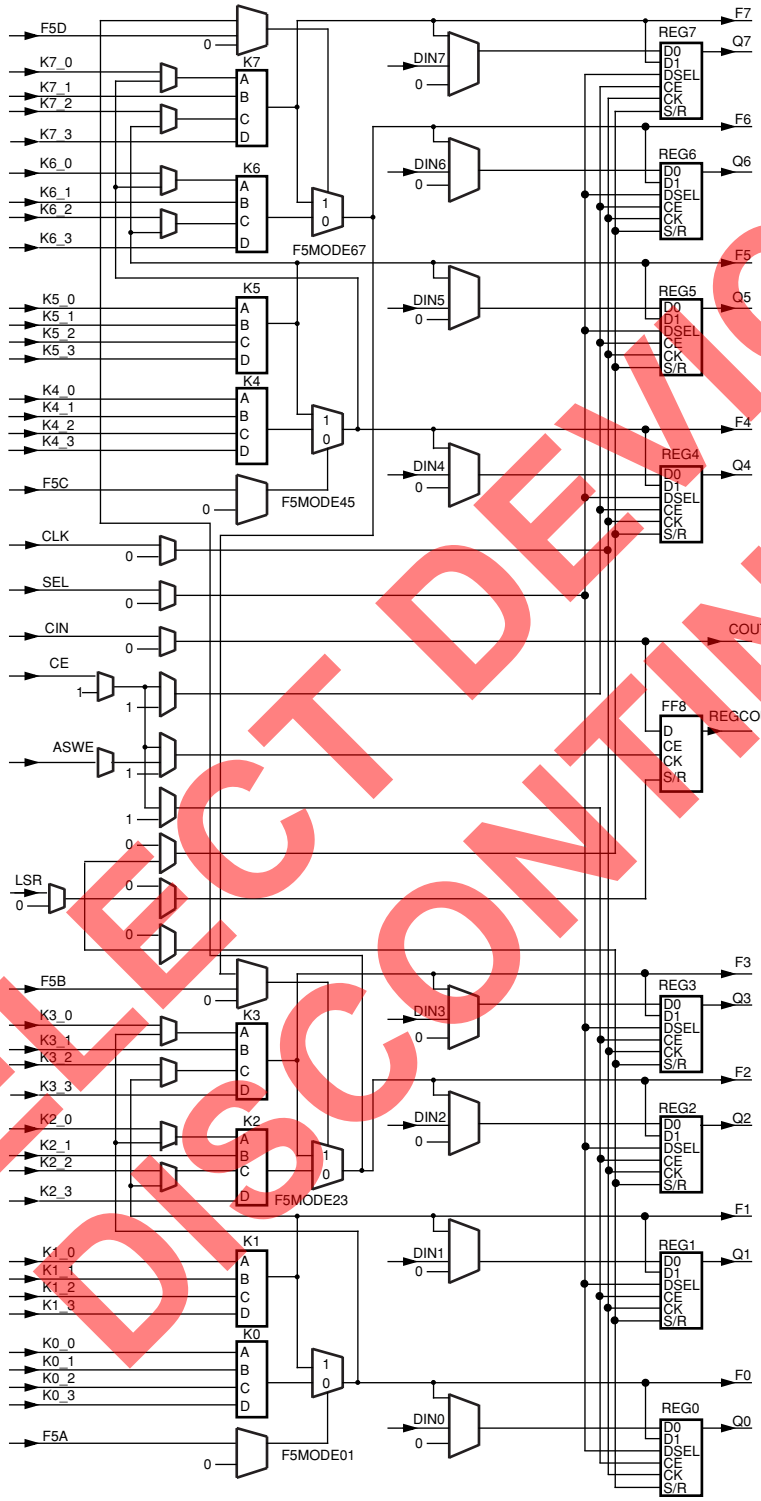


Figure 2. PFU Ports

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

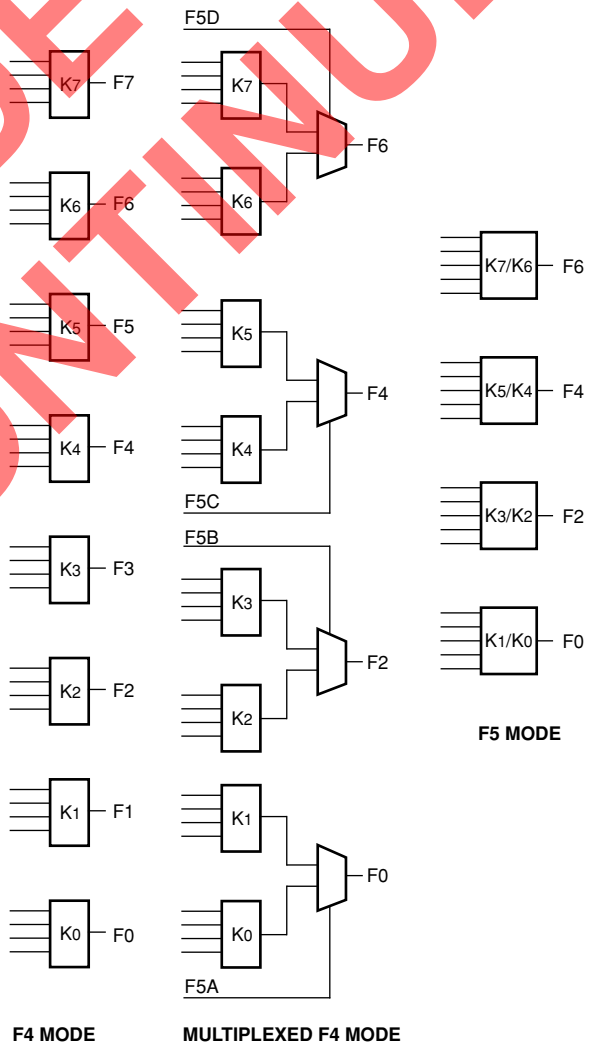
Mode	CLK	LSR	CE	ASWE	SEL
Logic	CLK to all latches/ FFs	LSR to all latches/ FFs, enabled per nibble and for ninth FF	CE to all latches/FFs, selectable per nibble and for ninth FF	CE to all latches/FFs, selectable per nibble and for ninth FF	Select between LUT input and direct input for eight latches/FFs
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.

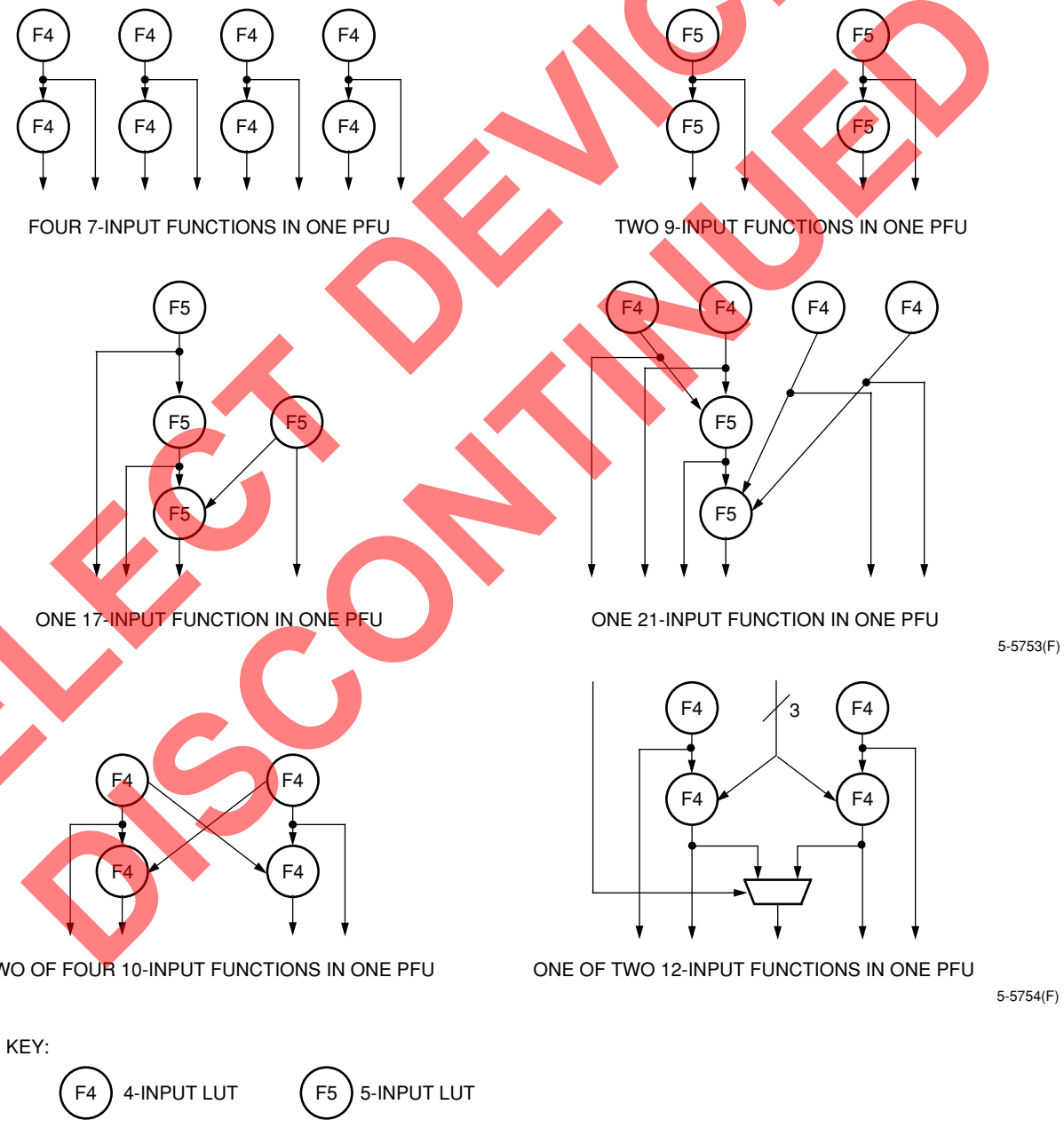


Figure 5. Softwired LUT Topology Examples

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.

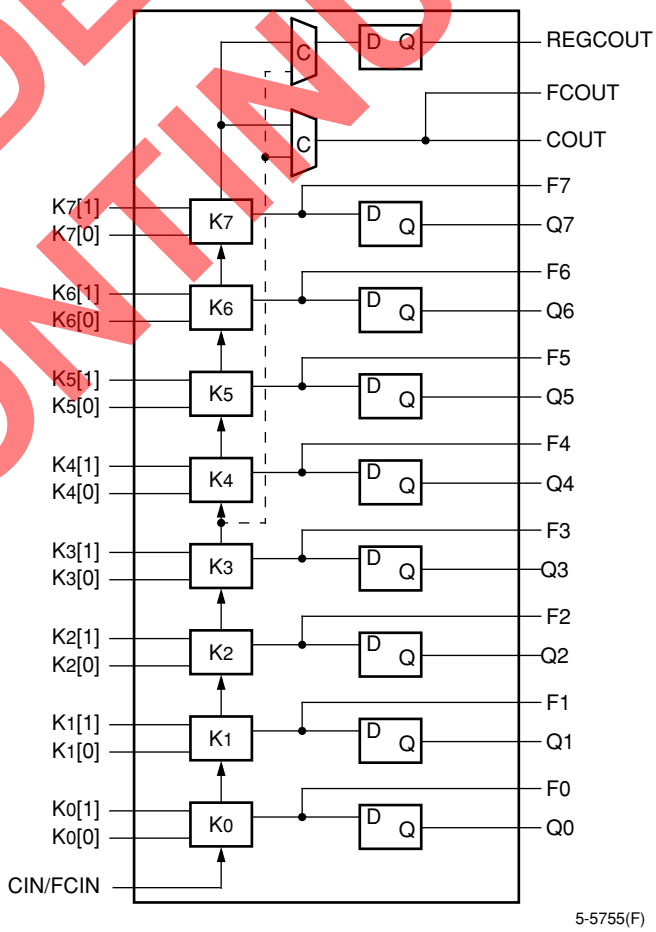


Figure 6. Ripple Mode

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.

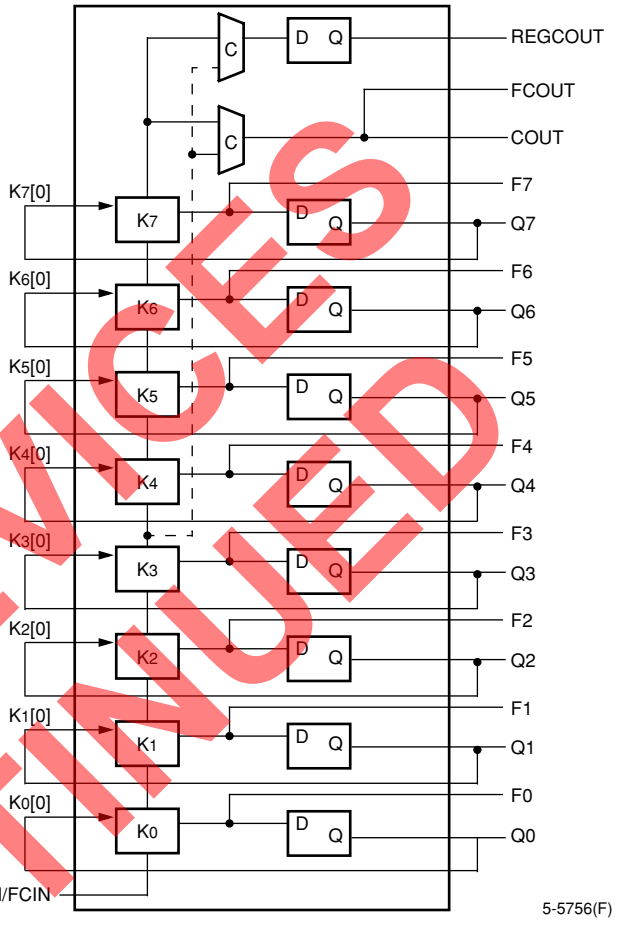


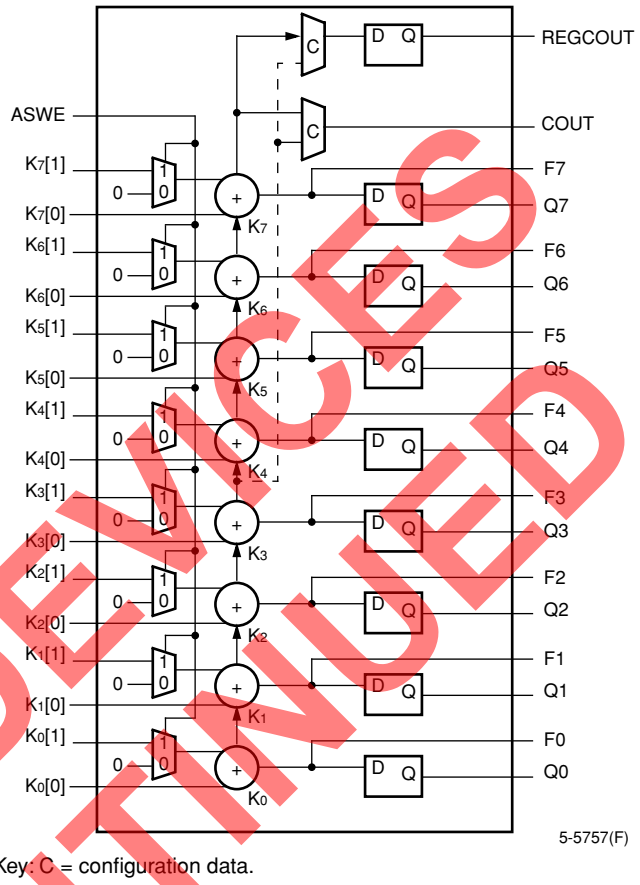
Figure 7. Counter Submode

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).

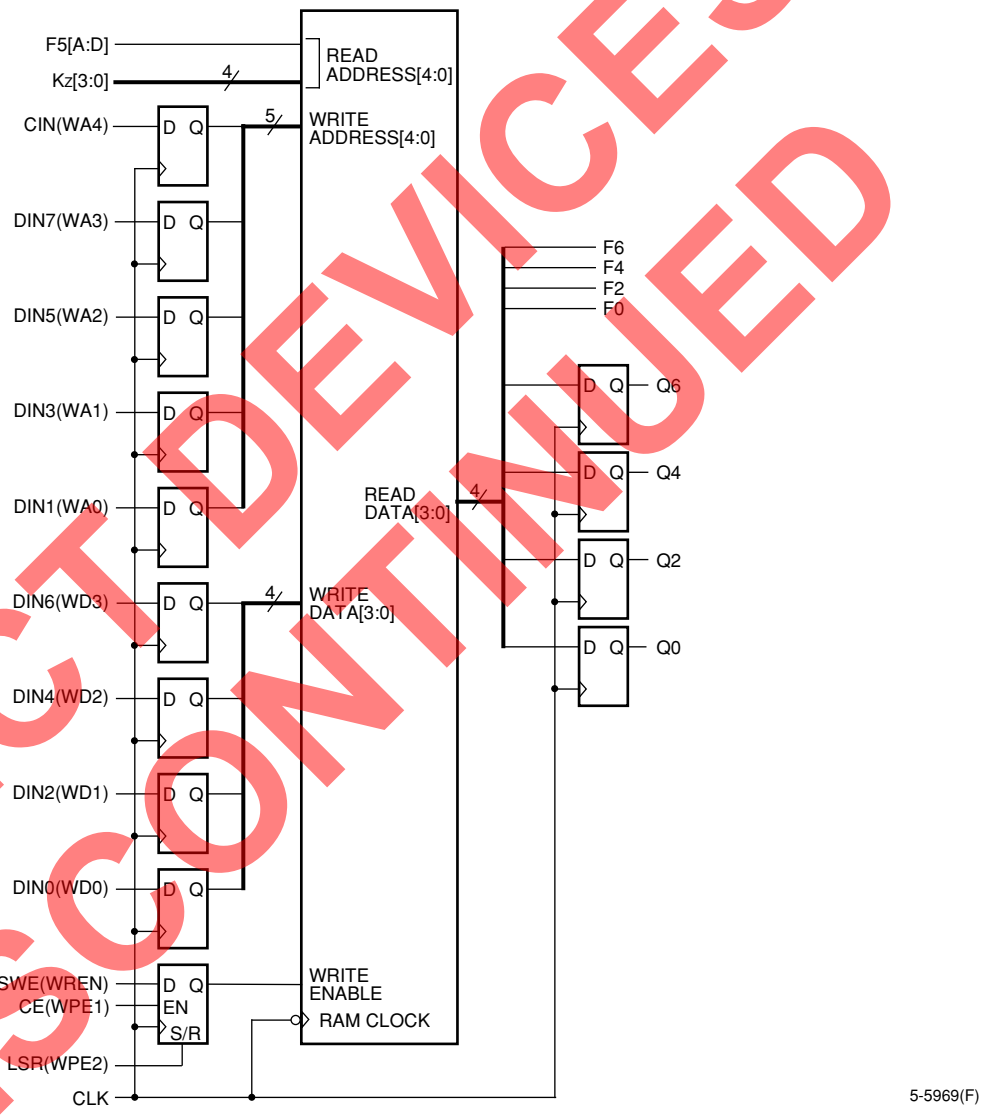


Figure 9. Memory Mode

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.

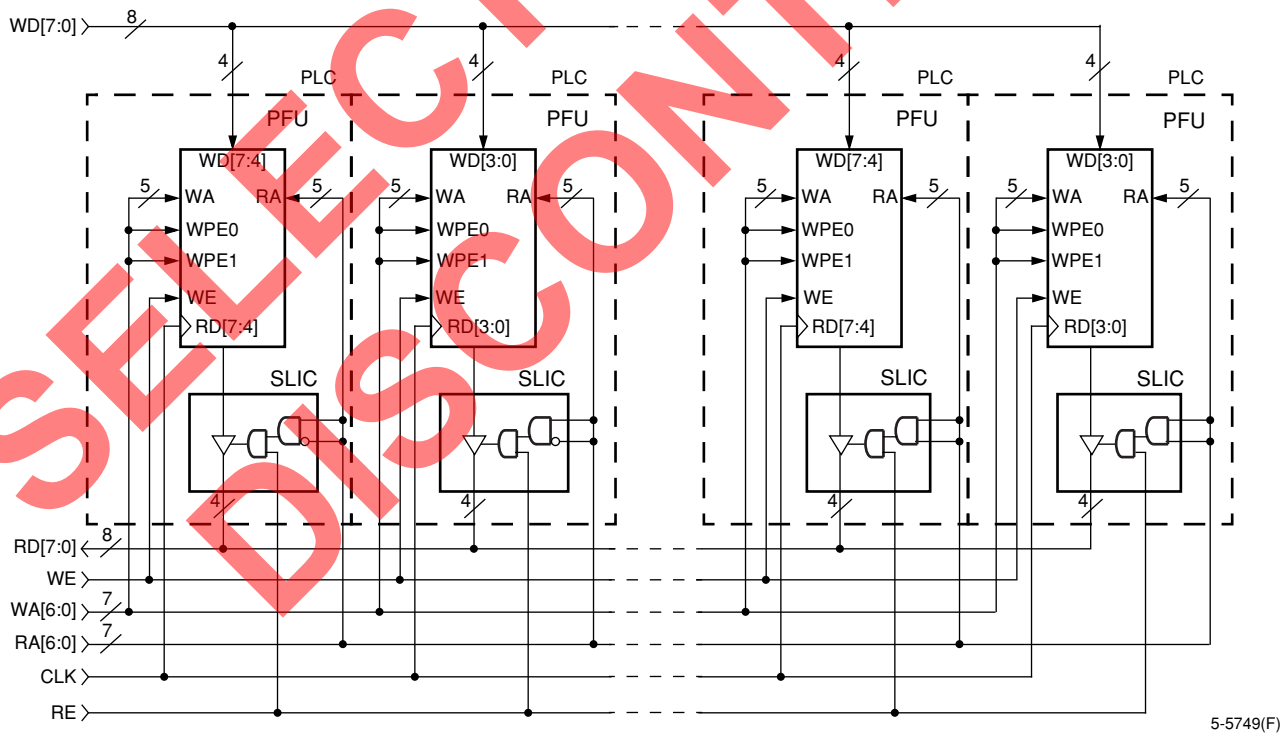


Figure 10. Memory Mode Expansion Example—128 x 8 RAM

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)

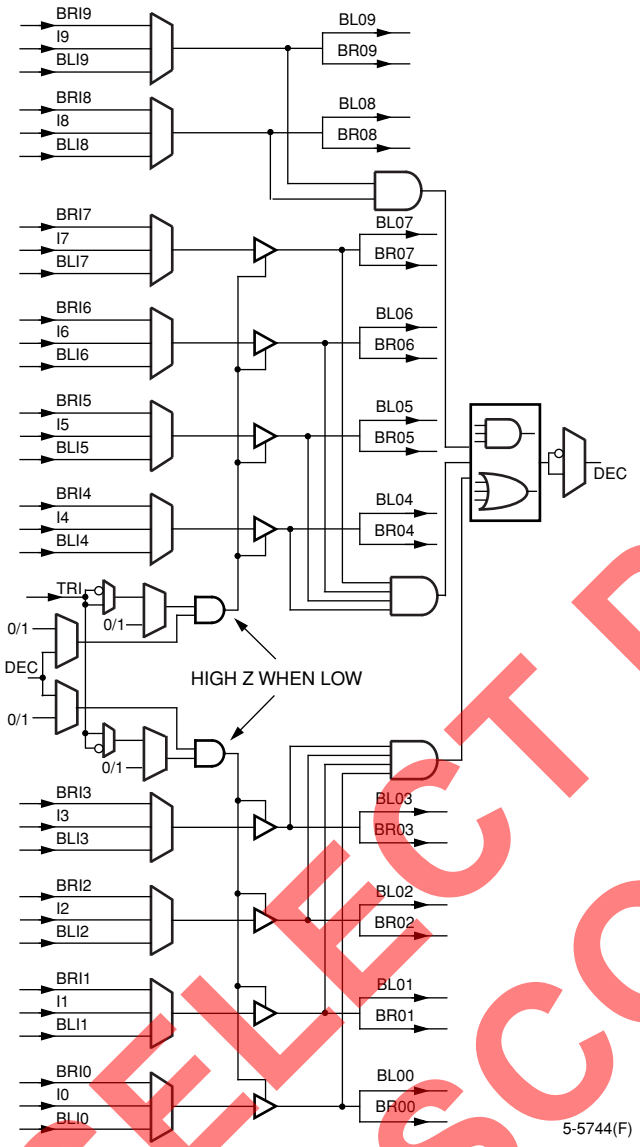


Figure 11. SLIC All Modes Diagram

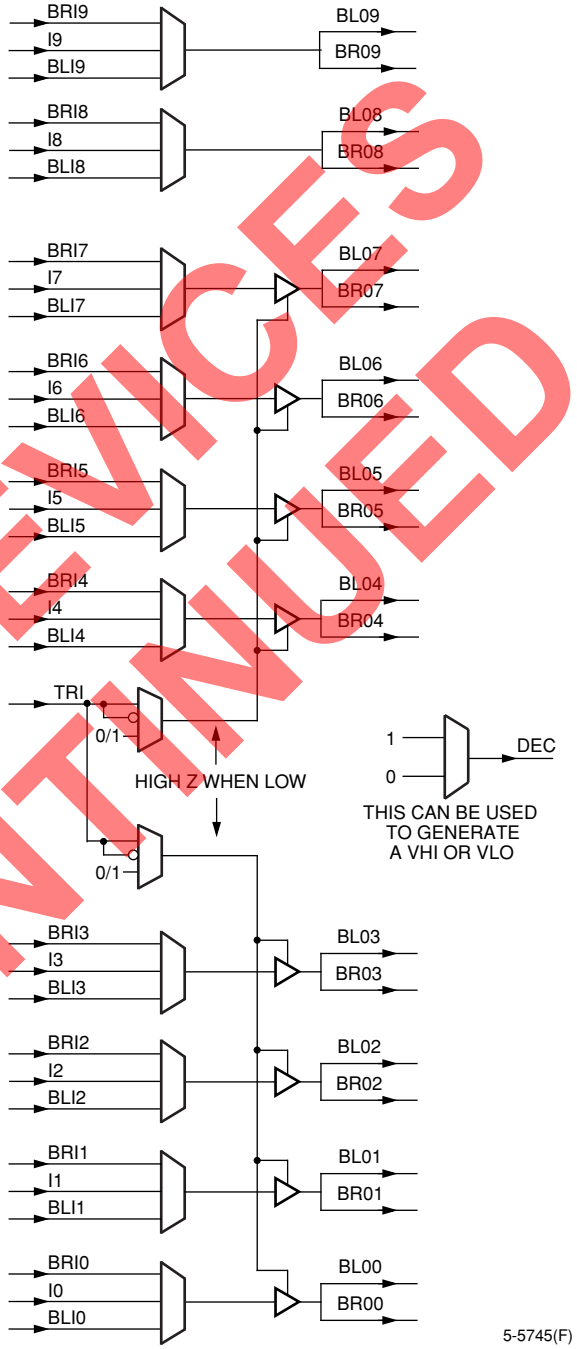


Figure 12. Buffer Mode

Programmable Logic Cells (continued)

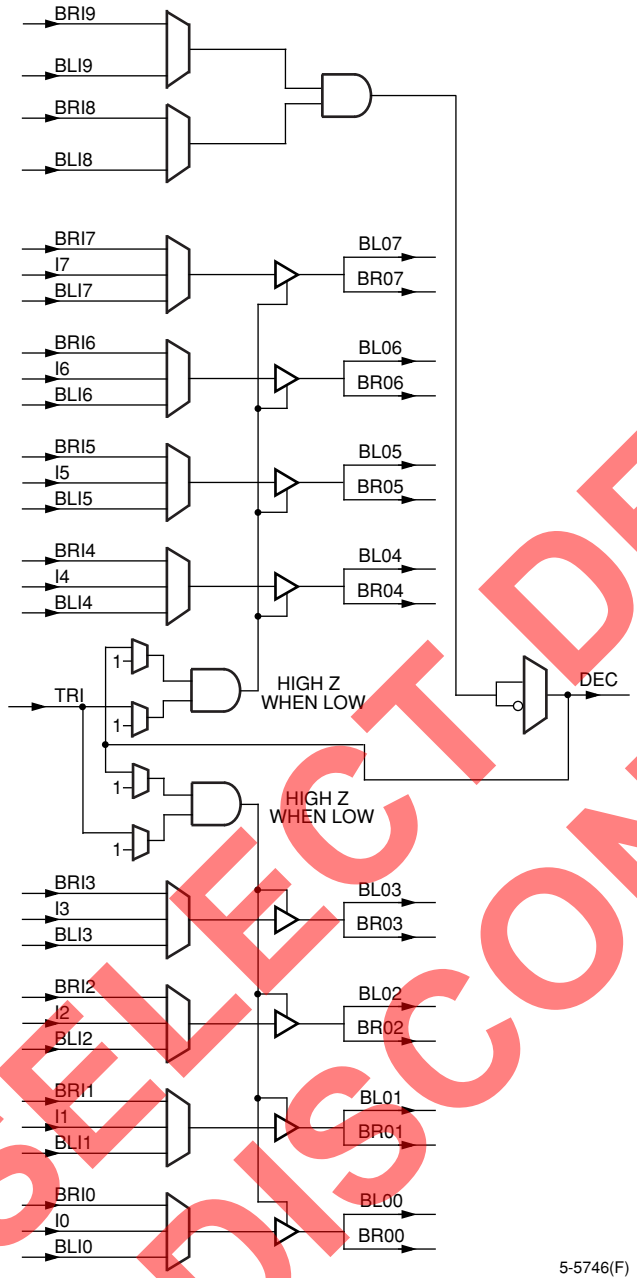


Figure 13. Buffer-Buffer-Decoder Mode

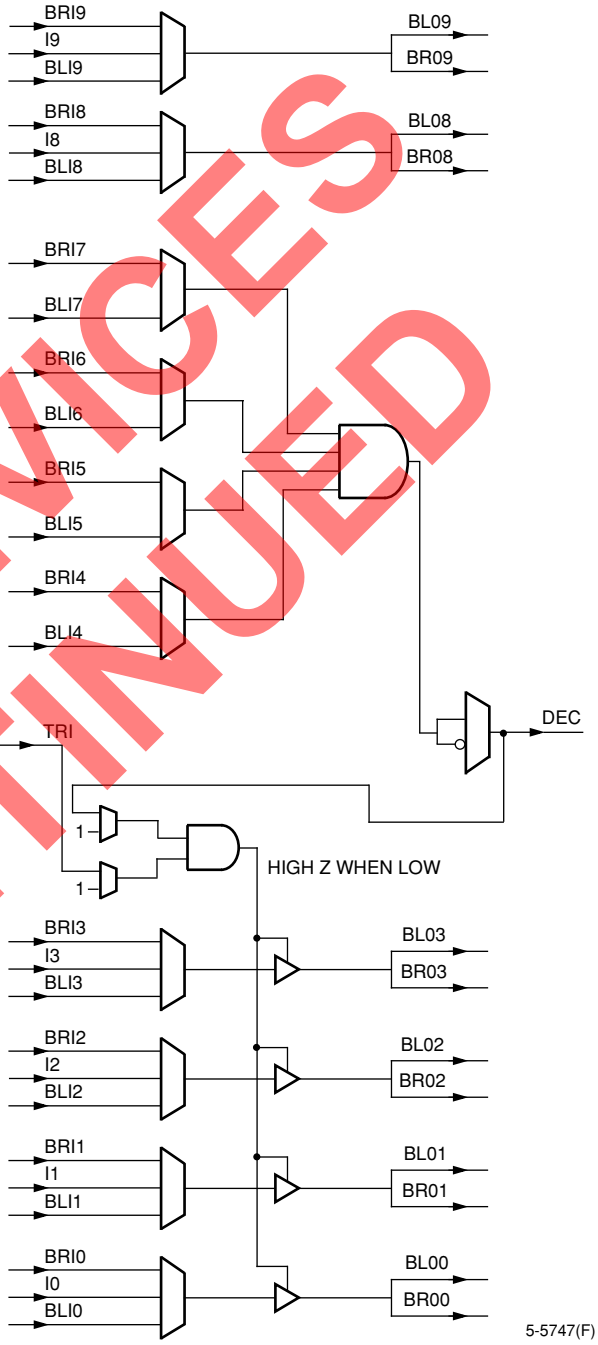


Figure 14. Buffer-Decoder-Buffer Mode

Programmable Logic Cells (continued)

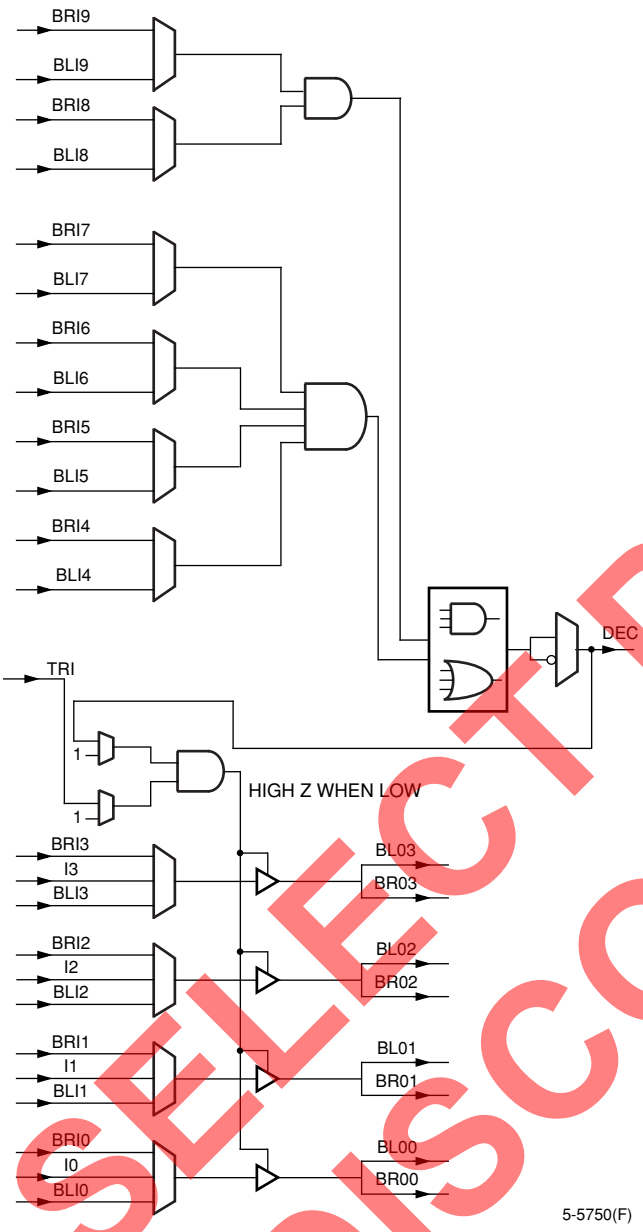


Figure 15. Buffer-Decoder-Decoder Mode

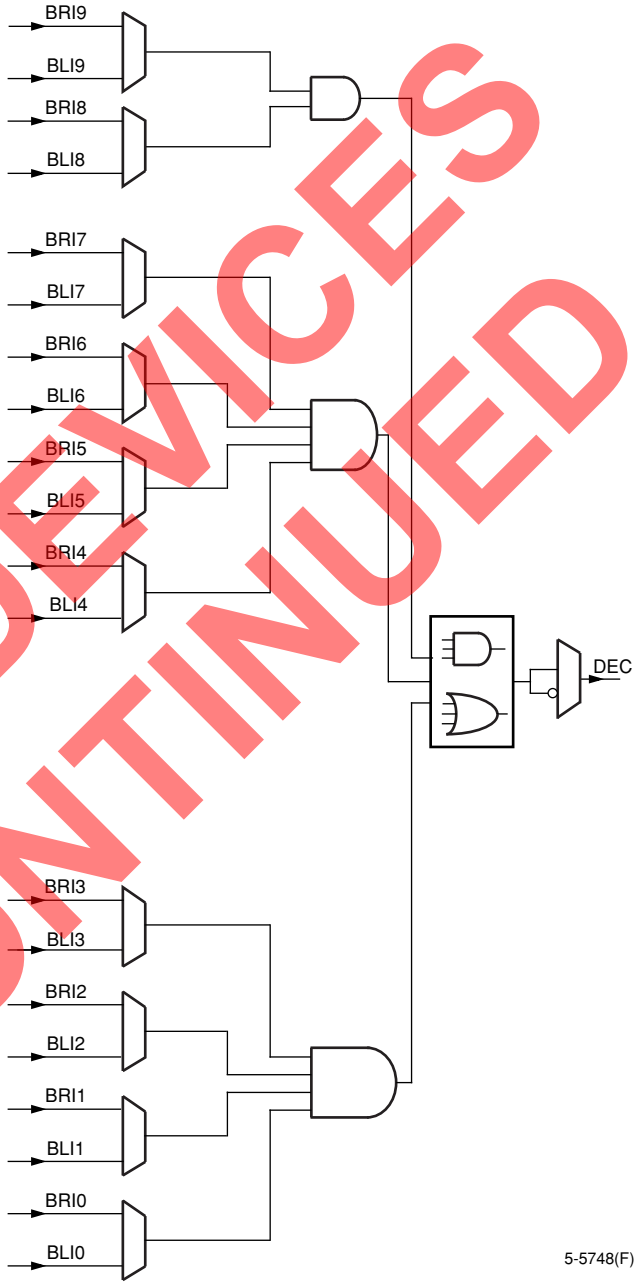


Figure 16. Decoder Mode

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.