



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





## **LatticeXP Family Data Sheet**

---

DS1001 Version 05.1, November 2007

## Features

### ■ Non-volatile, Infinitely Reconfigurable

- Instant-on – powers up in microseconds
- No external configuration memory
- Excellent design security, no bit stream to intercept
- Reconfigure SRAM based logic in milliseconds
- SRAM and non-volatile memory programmable through system configuration and JTAG ports

### ■ Sleep Mode

- Allows up to 1000x static current reduction

### ■ TransFR™ Reconfiguration (TFR)

- In-field logic update while system operates

### ■ Extensive Density and Package Options

- 3.1K to 19.7K LUT4s
- 62 to 340 I/Os
- Density migration supported

### ■ Embedded and Distributed Memory

- 54 Kbits to 396 Kbits sysMEM™ Embedded Block RAM
- Up to 79 Kbits distributed RAM
- Flexible memory resources:
  - Distributed and block memory

### ■ Flexible I/O Buffer

- Programmable sysIO™ buffer supports wide range of interfaces:
  - LVCMOS 3.3/2.5/1.8/1.5/1.2
  - LVTTTL
  - SSTL 18 Class I
  - SSTL 3/2 Class I, II
  - HSTL15 Class I, III
  - HSTL 18 Class I, II, III
  - PCI
  - LVDS, Bus-LVDS, LVPECL, RSDS

### ■ Dedicated DDR Memory Support

- Implements interface up to DDR333 (166MHz)

### ■ sysCLOCK™ PLLs

- Up to 4 analog PLLs per device
- Clock multiply, divide and phase shifting

### ■ System Level Support

- IEEE Standard 1149.1 Boundary Scan, plus ispTRACY™ internal logic analyzer capability
- Onboard oscillator for configuration
- Devices operate with 3.3V, 2.5V, 1.8V or 1.2V power supply

**Table 1-1. LatticeXP Family Selection Guide**

Device	LFXP3	LFXP6	LFXP10	LFXP15	LFXP20
PFU/PFF Rows	16	24	32	40	44
PFU/PFF Columns	24	30	38	48	56
PFU/PFF (Total)	384	720	1216	1932	2464
LUTs (K)	3	6	10	15	20
Distributed RAM (KBits)	12	23	39	61	79
EBR SRAM (KBits)	54	72	216	324	396
EBR SRAM Blocks	6	8	24	36	44
V <sub>CC</sub> Voltage	1.2/1.8/2.5/3.3V	1.2/1.8/2.5/3.3V	1.2/1.8/2.5/3.3V	1.2/1.8/2.5/3.3V	1.2/1.8/2.5/3.3V
PLLs	2	2	4	4	4
Max. I/O	136	188	244	300	340
<b>Packages and I/O Combinations:</b>					
100-pin TQFP (14 x 14 mm)	62				
144-pin TQFP (20 x 20 mm)	100	100			
208-pin PQFP (28 x 28 mm)	136	142			
256-ball fpBGA (17 x 17 mm)		188	188	188	188
388-ball fpBGA (23 x 23 mm)			244	268	268
484-ball fpBGA (23 x 23 mm)				300	340

---

## Introduction

The LatticeXP family of FPGA devices combine logic gates, embedded memory and high performance I/Os in a single architecture that is both non-volatile and infinitely reconfigurable to support cost-effective system designs.

The re-programmable non-volatile technology used in the LatticeXP family is the next generation ispXP™ technology. With this technology, expensive external configuration memories are not required and designs are secured from unauthorized read-back. In addition, instant-on capability allows for easy interfacing in many applications.

The ispLEVER® design tool from Lattice allows large complex designs to be efficiently implemented using the LatticeXP family of FPGA devices. Synthesis library support for LatticeXP is available for popular logic synthesis tools. The ispLEVER tool uses the synthesis tool output along with the constraints from its floor planning tools to place and route the design in the LatticeXP device. The ispLEVER tool extracts the timing from the routing and back-annotates it into the design for timing verification.

Lattice provides many pre-designed IP (Intellectual Property) ispLeverCORE™ modules for the LatticeXP family. By using these IPs as standardized blocks, designers are free to concentrate on the unique aspects of their design, increasing their productivity.

## Architecture Overview

The LatticeXP architecture contains an array of logic blocks surrounded by Programmable I/O Cells (PIC). Interspersed between the rows of logic blocks are rows of sysMEM Embedded Block RAM (EBR) as shown in Figure 2-1.

On the left and right sides of the PFU array, there are Non-volatile Memory Blocks. In configuration mode this non-volatile memory is programmed via the IEEE 1149.1 TAP port or the sysCONFIG™ peripheral port. On power up, the configuration data is transferred from the Non-volatile Memory Blocks to the configuration SRAM. With this technology, expensive external configuration memories are not required and designs are secured from unauthorized read-back. This transfer of data from non-volatile memory to configuration SRAM via wide busses happens in microseconds, providing an “instant-on” capability that allows easy interfacing in many applications.

There are two kinds of logic blocks, the Programmable Functional Unit (PFU) and Programmable Functional unit without RAM/ROM (PFF). The PFU contains the building blocks for logic, arithmetic, RAM, ROM and register functions. The PFF block contains building blocks for logic, arithmetic and ROM functions. Both PFU and PFF blocks are optimized for flexibility, allowing complex designs to be implemented quickly and efficiently. Logic Blocks are arranged in a two-dimensional array. Only one type of block is used per row. The PFU blocks are used on the outside rows. The rest of the core consists of rows of PFF blocks interspersed with rows of PFU blocks. For every three rows of PFF blocks there is a row of PFU blocks.

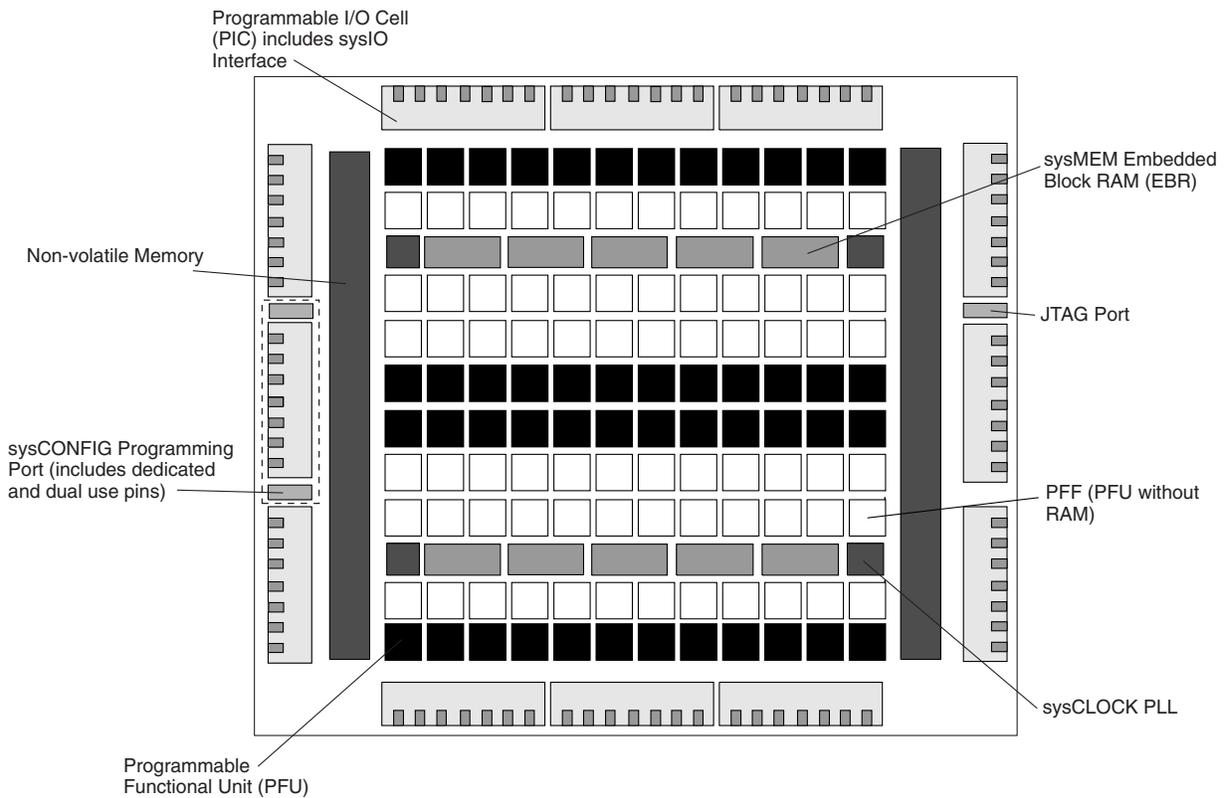
Each PIC block encompasses two PIOs (PIO pairs) with their respective sysIO interfaces. PIO pairs on the left and right edges of the device can be configured as LVDS transmit/receive pairs. sysMEM EBRs are large dedicated fast memory blocks. They can be configured as RAM or ROM.

The PFU, PFF, PIC and EBR Blocks are arranged in a two-dimensional grid with rows and columns as shown in Figure 2-1. The blocks are connected with many vertical and horizontal routing channel resources. The place and route software tool automatically allocates these routing resources.

At the end of the rows containing the sysMEM Blocks are the sysCLOCK Phase Locked Loop (PLL) Blocks. These PLLs have multiply, divide and phase shifting capability; they are used to manage the phase relationship of the clocks. The LatticeXP architecture provides up to four PLLs per device.

Every device in the family has a JTAG Port with internal Logic Analyzer (ispTRACY) capability. The sysCONFIG port which allows for serial or parallel device configuration. The LatticeXP devices are available for operation from 3.3V, 2.5V, 1.8V and 1.2V power supplies, providing easy integration into the overall system.

Figure 2-1. LatticeXP Top Level Block Diagram

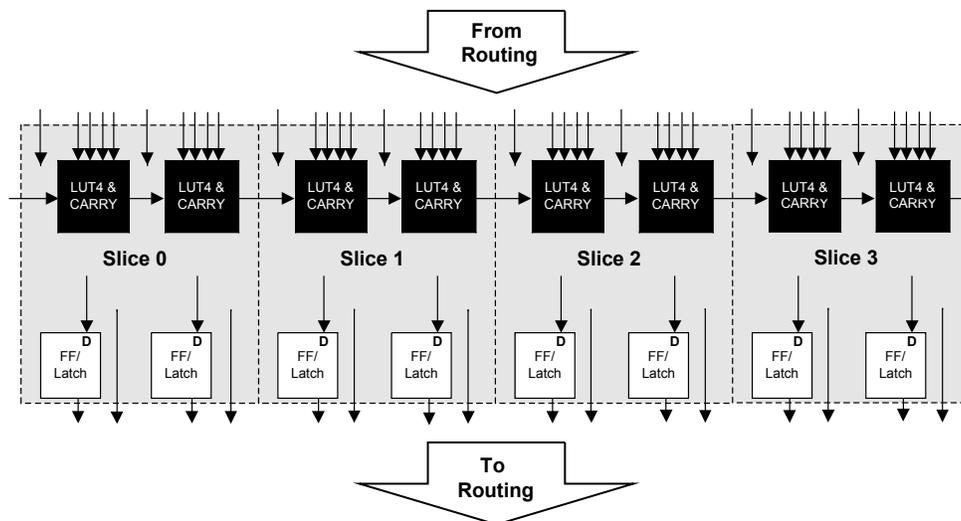


**PFU and PFF Blocks**

The core of the LatticeXP devices consists of PFU and PFF blocks. The PFUs can be programmed to perform Logic, Arithmetic, Distributed RAM and Distributed ROM functions. PFF blocks can be programmed to perform Logic, Arithmetic and ROM functions. Except where necessary, the remainder of the data sheet will use the term PFU to refer to both PFU and PFF blocks.

Each PFU block consists of four interconnected slices, numbered 0-3 as shown in Figure 2-2. All the interconnections to and from PFU blocks are from routing. There are 53 inputs and 25 outputs associated with each PFU block.

Figure 2-2. PFU Diagram

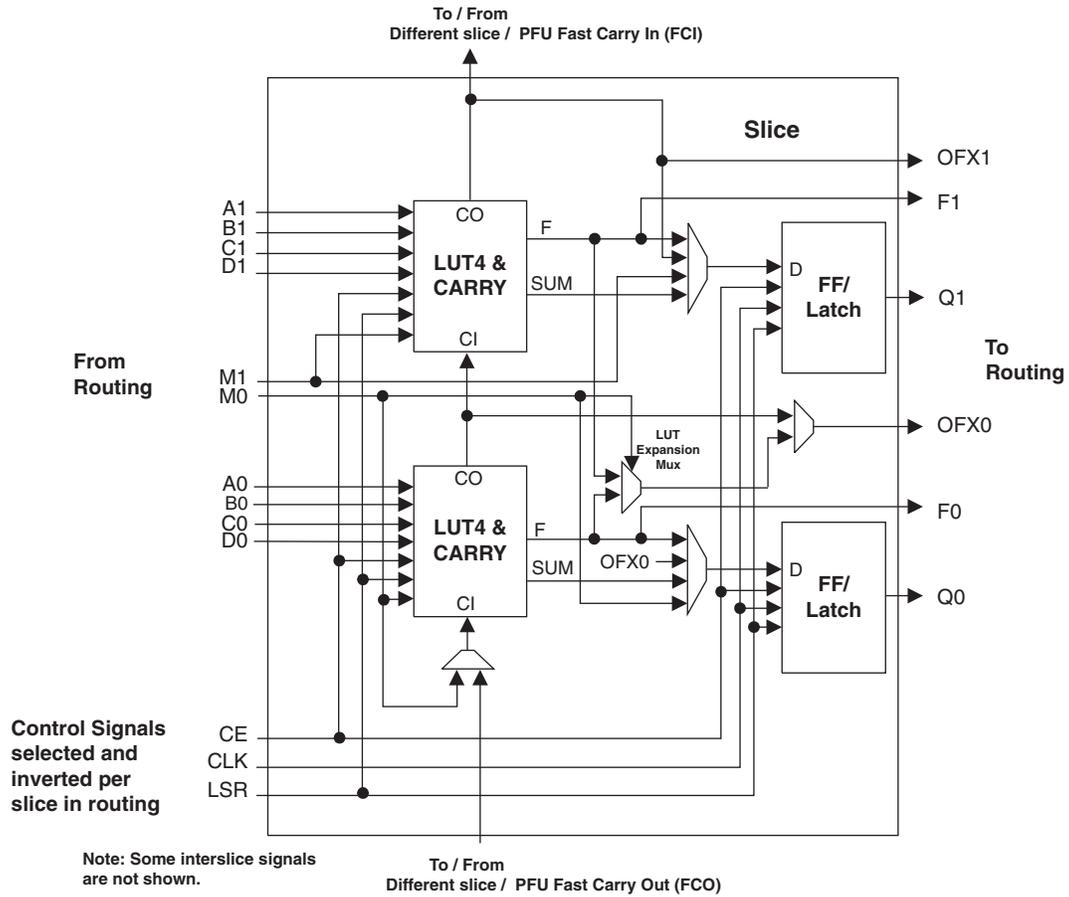


**Slice**

Each slice contains two LUT4 lookup tables feeding two registers (programmed to be in FF or Latch mode), and some associated logic that allows the LUTs to be combined to perform functions such as LUT5, LUT6, LUT7 and LUT8. There is control logic to perform set/reset functions (programmable as synchronous/asynchronous), clock select, chip-select and wider RAM/ROM functions. Figure 2-3 shows an overview of the internal logic of the slice. The registers in the slice can be configured for positive/negative and edge/level clocks.

There are 14 input signals: 13 signals from routing and one from the carry-chain (from adjacent slice or PFU). There are 7 outputs: 6 to routing and one to carry-chain (to adjacent PFU). Table 2-1 lists the signals associated with each slice.

**Figure 2-3. Slice Diagram**



**Table 2-1. Slice Signal Descriptions**

Function	Type	Signal Names	Description
Input	Data signal	A0, B0, C0, D0	Inputs to LUT4
Input	Data signal	A1, B1, C1, D1	Inputs to LUT4
Input	Multi-purpose	M0	Multipurpose Input
Input	Multi-purpose	M1	Multipurpose Input
Input	Control signal	CE	Clock Enable
Input	Control signal	LSR	Local Set/Reset
Input	Control signal	CLK	System Clock
Input	Inter-PFU signal	FCIN	Fast Carry In <sup>1</sup>
Output	Data signals	F0, F1	LUT4 output register bypass signals
Output	Data signals	Q0, Q1	Register Outputs
Output	Data signals	OFX0	Output of a LUT5 MUX
Output	Data signals	OFX1	Output of a LUT6, LUT7, LUT8 <sup>2</sup> MUX depending on the slice
Output	Inter-PFU signal	FCO	For the right most PFU the fast carry chain output <sup>1</sup>

1. See Figure 2-2 for connection details.
2. Requires two PFUs.

**Modes of Operation**

Each Slice is capable of four modes of operation: Logic, Ripple, RAM and ROM. The Slice in the PFF is capable of all modes except RAM. Table 2-2 lists the modes and the capability of the Slice blocks.

**Table 2-2. Slice Modes**

	Logic	Ripple	RAM	ROM
PFU Slice	LUT 4x2 or LUT 5x1	2-bit Arithmetic Unit	SP 16x2	ROM 16x1 x 2
PFF Slice	LUT 4x2 or LUT 5x1	2-bit Arithmetic Unit	N/A	ROM 16x1 x 2

**Logic Mode:** In this mode, the LUTs in each Slice are configured as 4-input combinatorial lookup tables. A LUT4 can have 16 possible input combinations. Any logic function with four inputs can be generated by programming this lookup table. Since there are two LUT4s per Slice, a LUT5 can be constructed within one Slice. Larger lookup tables such as LUT6, LUT7 and LUT8 can be constructed by concatenating other Slices.

**Ripple Mode:** Ripple mode allows the efficient implementation of small arithmetic functions. In ripple mode, the following functions can be implemented by each Slice:

- Addition 2-bit
- Subtraction 2-bit
- Add/Subtract 2-bit using dynamic control
- Up counter 2-bit
- Down counter 2-bit
- Ripple mode multiplier building block
- Comparator functions of A and B inputs
  - A greater-than-or-equal-to B
  - A not-equal-to B
  - A less-than-or-equal-to B

Two additional signals: Carry Generate and Carry Propagate are generated per Slice in this mode, allowing fast arithmetic functions to be constructed by concatenating Slices.

**RAM Mode:** In this mode, distributed RAM can be constructed using each LUT block as a 16x1-bit memory. Through the combination of LUTs and Slices, a variety of different memories can be constructed.

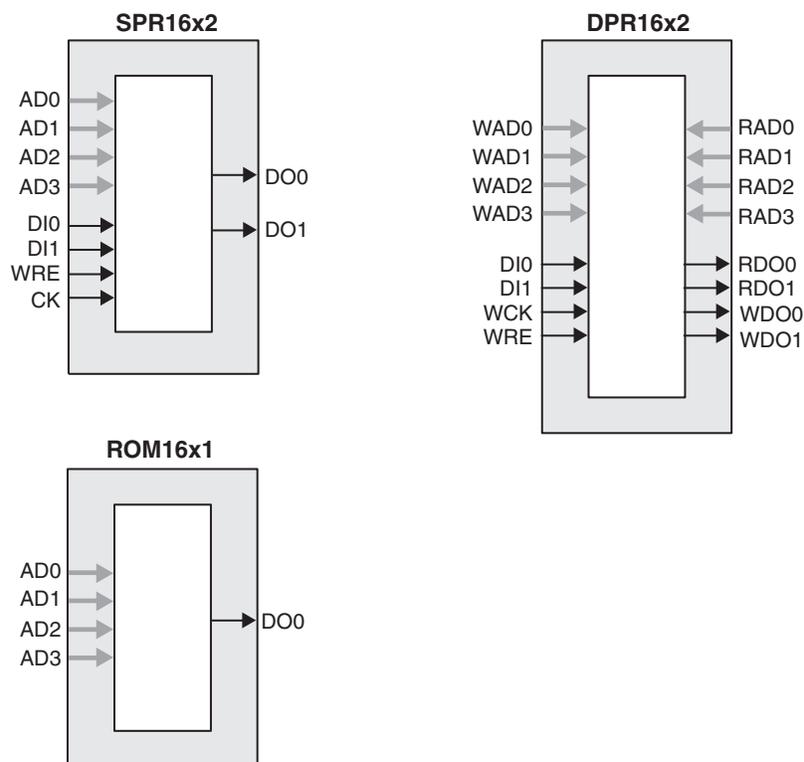
The Lattice design tools support the creation of a variety of different size memories. Where appropriate, the software will construct these using distributed memory primitives that represent the capabilities of the PFU. Table 2-3 shows the number of Slices required to implement different distributed RAM primitives. Figure 2-4 shows the distributed memory primitive block diagrams. Dual port memories involve the pairing of two Slices, one Slice functions as the read-write port. The other companion Slice supports the read-only port. For more information on RAM mode in LatticeXP devices, please see details of additional technical documentation at the end of this data sheet.

**Table 2-3. Number of Slices Required for Implementing Distributed RAM**

	SPR16x2	DPR16x2
Number of Slices	1	2

Note: SPR = Single Port RAM, DPR = Dual Port RAM

**Figure 2-4. Distributed Memory Primitives**



**ROM Mode:** The ROM mode uses the same principal as the RAM modes, but without the Write port. Pre-loading is accomplished through the programming interface during configuration.

**PFU Modes of Operation**

Slices can be combined within a PFU to form larger functions. Table 2-4 tabulates these modes and documents the functionality possible at the PFU level.

**Table 2-4. PFU Modes of Operation**

Logic	Ripple	RAM <sup>1</sup>	ROM
LUT 4x8 or MUX 2x1 x 8	2-bit Add x 4	SPR16x2 x 4 DPR16x2 x 2	ROM16x1 x 8
LUT 5x4 or MUX 4x1 x 4	2-bit Sub x 4	SPR16x4 x 2 DPR16x4 x 1	ROM16x2 x 4
LUT 6x 2 or MUX 8x1 x 2	2-bit Counter x 4	SPR16x8 x 1	ROM16x4 x 2
LUT 7x1 or MUX 16x1 x 1	2-bit Comp x 4		ROM16x8 x 1

1. These modes are not available in PFF blocks

## Routing

There are many resources provided in the LatticeXP devices to route signals individually or as buses with related control signals. The routing resources consist of switching circuitry, buffers and metal interconnect (routing) segments.

The inter-PFU connections are made with x1 (spans two PFU), x2 (spans three PFU) and x6 (spans seven PFU). The x1 and x2 connections provide fast and efficient connections in horizontal, vertical and diagonal directions. The x2 and x6 resources are buffered allowing both short and long connections routing between PFUs.

The ispLEVER design tool takes the output of the synthesis tool and places and routes the design. Generally, the place and route tool is completely automatic, although an interactive routing editor is available to optimize the design.

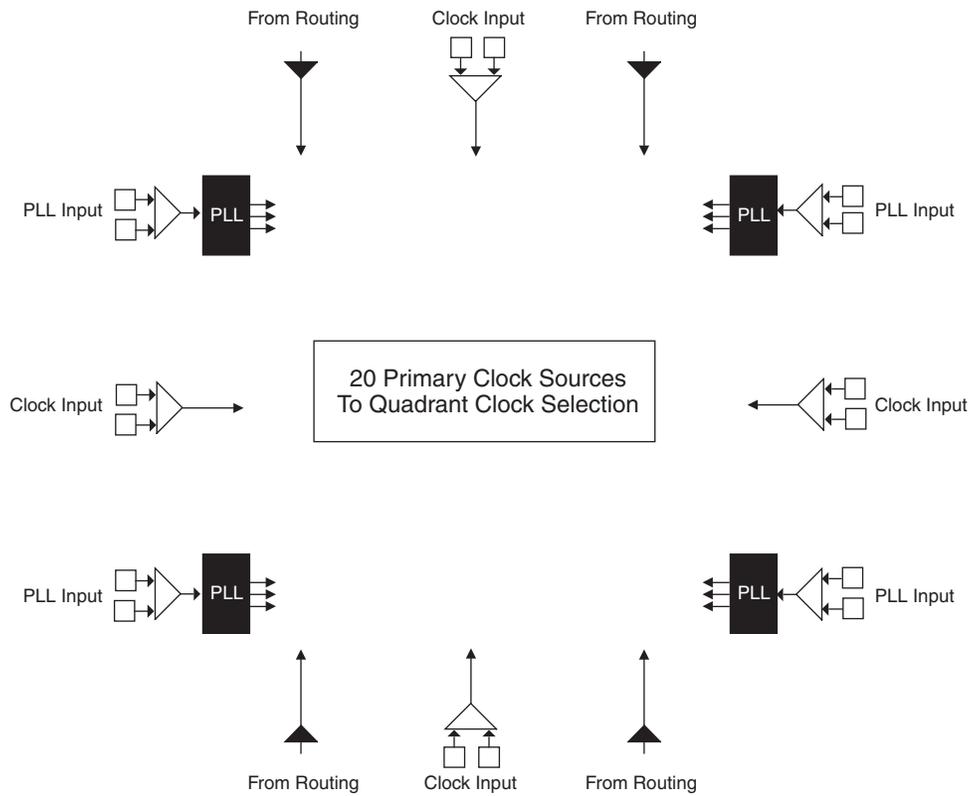
## Clock Distribution Network

The clock inputs are selected from external I/O, the sysCLOCK™ PLLs or routing. These clock inputs are fed through the chip via a clock distribution system.

### Primary Clock Sources

LatticeXP devices derive clocks from three primary sources: PLL outputs, dedicated clock inputs and routing. LatticeXP devices have two to four sysCLOCK PLLs, located on the left and right sides of the device. There are four dedicated clock inputs, one on each side of the device. Figure 2-5 shows the 20 primary clock sources.

Figure 2-5. Primary Clock Sources

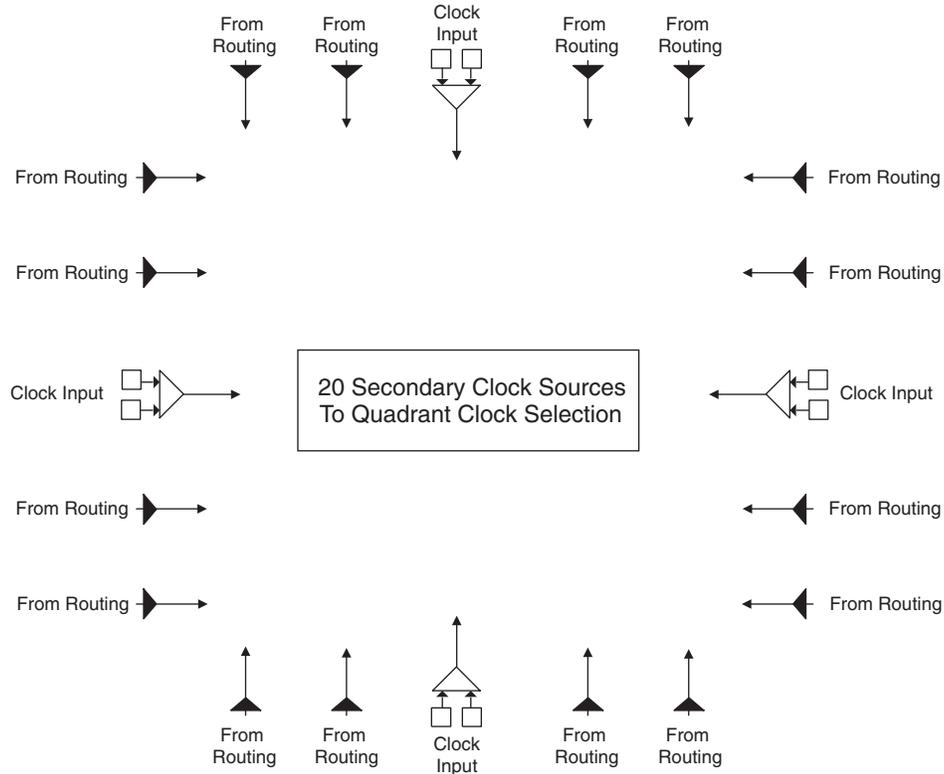


Note: Smaller devices have two PLLs.

### Secondary Clock Sources

LatticeXP devices have four secondary clock resources per quadrant. The secondary clock branches are tapped at every PFU. These secondary clock networks can also be used for controls and high fanout data. These secondary clocks are derived from four clock input pads and 16 routing signals as shown in Figure 2-6.

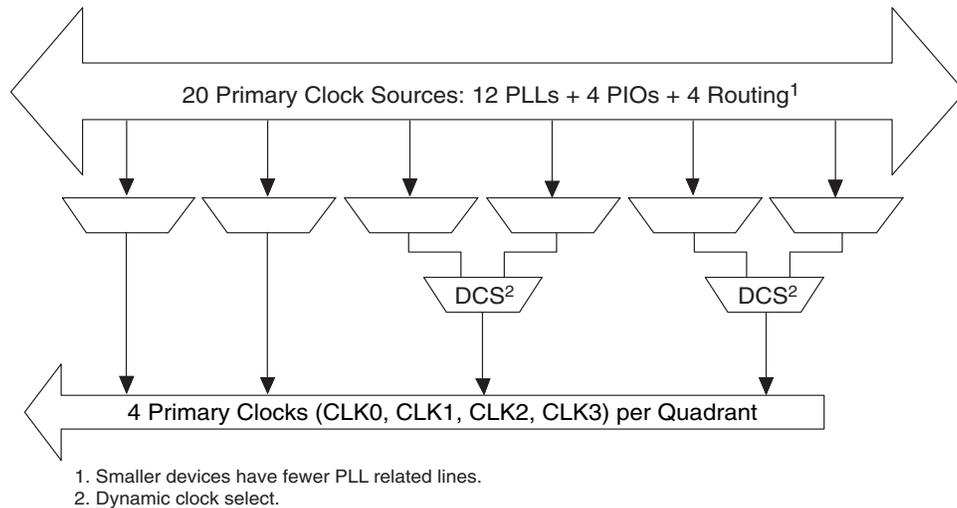
Figure 2-6. Secondary Clock Sources



**Clock Routing**

The clock routing structure in LatticeXP devices consists of four Primary Clock lines and a Secondary Clock network per quadrant. The primary clocks are generated from MUXs located in each quadrant. Figure 2-7 shows this clock routing. The four secondary clocks are generated from MUXs located in each quadrant as shown in Figure 2-8. Each slice derives its clock from the primary clock lines, secondary clock lines and routing as shown in Figure 2-9.

Figure 2-7. Per Quadrant Primary Clock Selection



1. Smaller devices have fewer PLL related lines.  
2. Dynamic clock select.

Figure 2-8. Per Quadrant Secondary Clock Selection

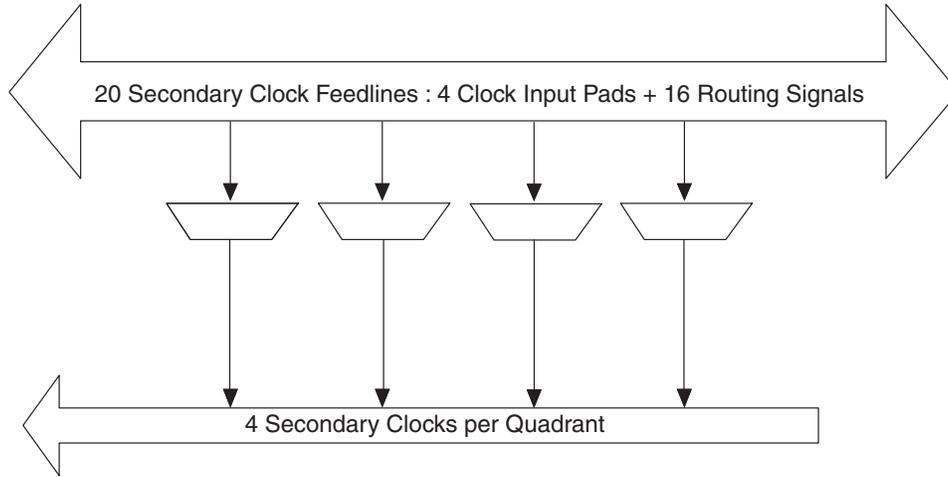
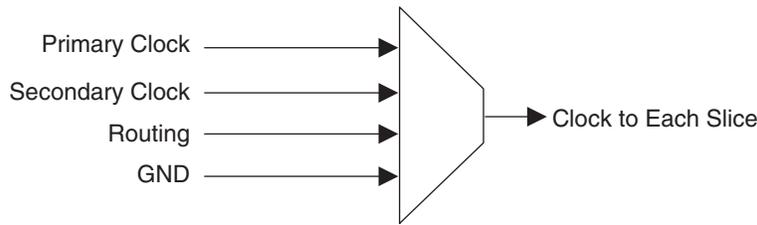


Figure 2-9. Slice Clock Selection



**sysCLOCK Phase Locked Loops (PLLs)**

The PLL clock input, from pin or routing, feeds into an input clock divider. There are three sources of feedback signals to the feedback divider: from CLKOP (PLL internal), from clock net (CLKOP or CLKOS) or from a user clock (PIN or logic). There is a PLL\_LOCK signal to indicate that VCO has locked on to the input clock signal. Figure 2-10 shows the sysCLOCK PLL diagram.

The setup and hold times of the device can be improved by programming a delay in the feedback or input path of the PLL which will advance or delay the output clock with reference to the input clock. This delay can be either programmed during configuration or can be adjusted dynamically. In dynamic mode, the PLL may lose lock after adjustment and not relock until the  $t_{LOCK}$  parameter has been satisfied. Additionally, the phase and duty cycle block allows the user to adjust the phase and duty cycle of the CLKOS output.

The sysCLOCK PLLs provide the ability to synthesize clock frequencies. Each PLL has four dividers associated with it: input clock divider, feedback divider, post scalar divider and secondary clock divider. The input clock divider is used to divide the input clock signal, while the feedback divider is used to multiply the input clock signal. The post scalar divider allows the VCO to operate at higher frequencies than the clock output, thereby increasing the frequency range. The secondary divider is used to derive lower frequency outputs.

Figure 2-10. PLL Diagram

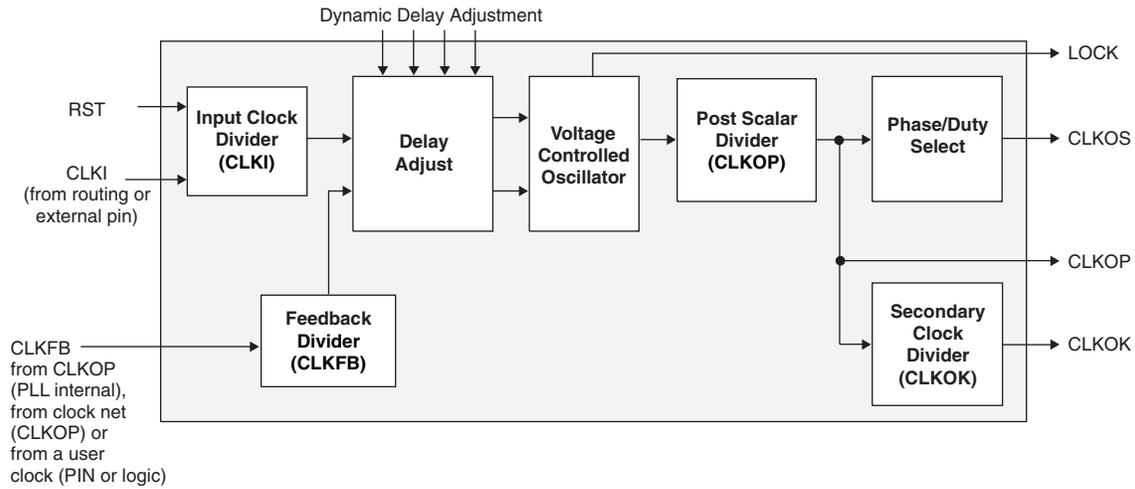


Figure 2-11 shows the available macros for the PLL. Table 2-11 provides signal description of the PLL Block.

Figure 2-11. PLL Primitive

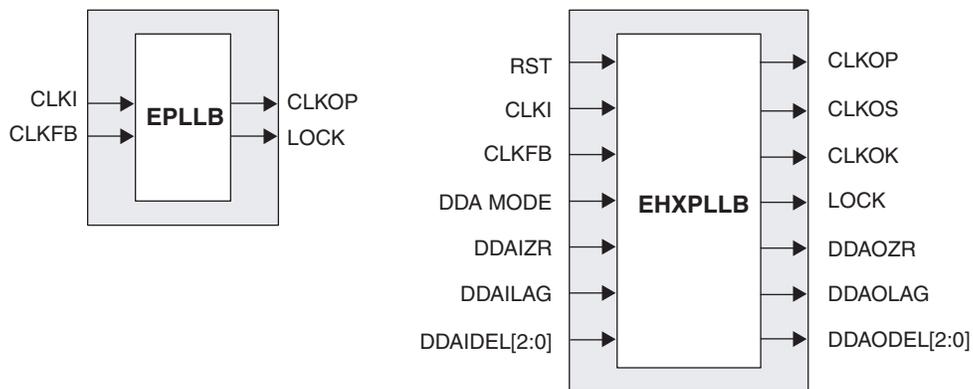


Table 2-5. PLL Signal Descriptions

Signal	I/O	Description
CLKI	I	Clock input from external pin or routing
CLKFB	I	PLL feedback input from CLKOP (PLL internal), from clock net (CLKOP) or from a user clock (PIN or logic)
RST	I	“1” to reset input clock divider
CLKOS	O	PLL output clock to clock tree (phase shifted/duty cycle changed)
CLKOP	O	PLL output clock to clock tree (No phase shift)
CLKOK	O	PLL output to clock tree through secondary clock divider
LOCK	O	“1” indicates PLL LOCK to CLKI
DDAMODE	I	Dynamic Delay Enable. “1” Pin control (dynamic), “0”: Fuse Control (static)
DDAIZR	I	Dynamic Delay Zero. “1”: delay = 0, “0”: delay = on
DDAILAG	I	Dynamic Delay Lag/Lead. “1”: Lag, “0”: Lead
DDAIDEL[2:0]	I	Dynamic Delay Input
DDAOZR	O	Dynamic Delay Zero Output
DDAOLAG	O	Dynamic Delay Lag/Lead Output
DDAODEL[2:0]	O	Dynamic Delay Output

For more information on the PLL, please see details of additional technical documentation at the end of this data sheet.

## Dynamic Clock Select (DCS)

The DCS is a global clock buffer with smart multiplexer functions. It takes two independent input clock sources and outputs a clock signal without any glitches or runt pulses. This is achieved irrespective of where the select signal is toggled. There are eight DCS blocks per device, located in pairs at the center of each side. Figure 2-12 illustrates the DCS Block Macro.

**Figure 2-12. DCS Block Primitive**

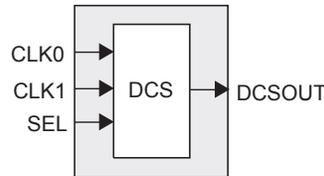
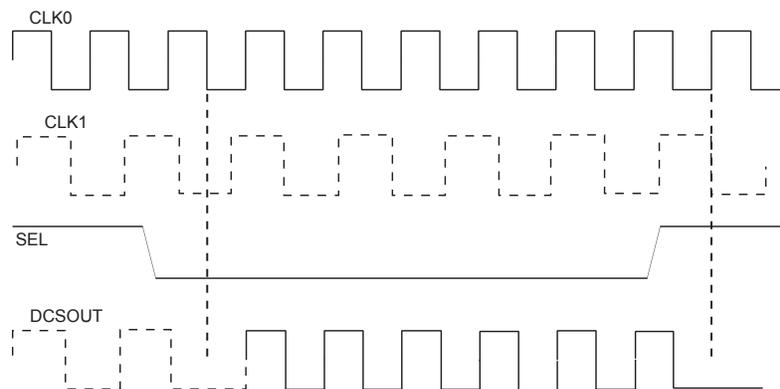


Figure 2-13 shows timing waveforms of the default DCS operating mode. The DCS block can be programmed to other modes. For more information on the DCS, please see details of additional technical documentation at the end of this data sheet.

**Figure 2-13. DCS Waveforms**



## sysMEM Memory

The LatticeXP family of devices contain a number of sysMEM Embedded Block RAM (EBR). The EBR consists of a 9-Kbit RAM, with dedicated input and output registers.

### sysMEM Memory Block

The sysMEM block can implement single port, dual port or pseudo dual port memories. Each block can be used in a variety of depths and widths as shown in Table 2-6.

**Table 2-6. sysMEM Block Configurations**

Memory Mode	Configurations
Single Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9 512 x 18 256 x 36
True Dual Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9 512 x 18
Pseudo Dual Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9 512 x 18 256 x 36

### Bus Size Matching

All of the multi-port memory modes support different widths on each of the ports. The RAM bits are mapped LSB word 0 to MSB word 0, LSB word 1 to MSB word 1 and so on. Although the word size and number of words for each port varies, this mapping scheme applies to each port.

### RAM Initialization and ROM Operation

If desired, the contents of the RAM can be pre-loaded during device configuration. By preloading the RAM block during the chip configuration cycle and disabling the write controls, the sysMEM block can also be utilized as a ROM.

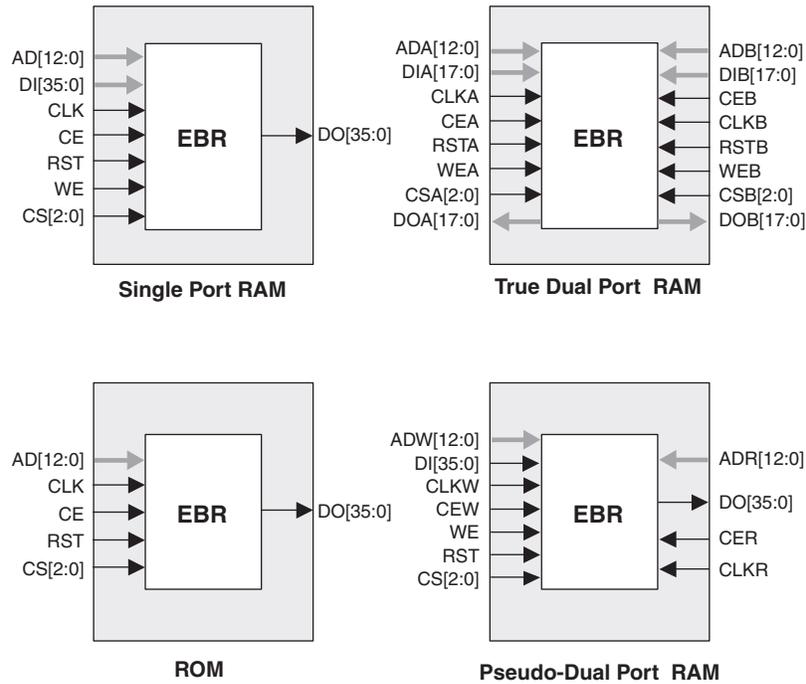
### Memory Cascading

Larger and deeper blocks of RAMs can be created using EBR sysMEM Blocks. Typically, the Lattice design tools cascade memory transparently, based on specific design inputs.

### Single, Dual and Pseudo-Dual Port Modes

Figure 2-14 shows the four basic memory configurations and their input/output names. In all the sysMEM RAM modes the input data and address for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the output.

Figure 2-14. sysMEM Memory Primitives



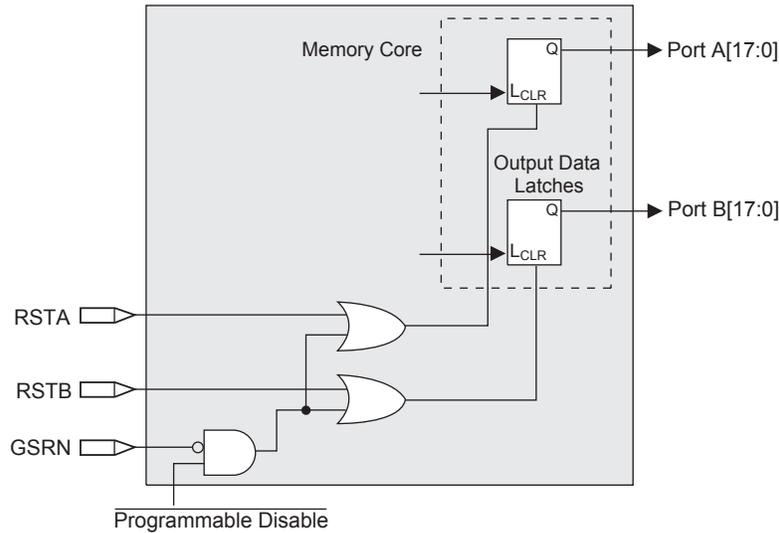
The EBR memory supports three forms of write behavior for single port or dual port operation:

1. **Normal** – data on the output appears only during read cycle. During a write cycle, the data (at the current address) does not appear on the output. This mode is supported for all data widths.
2. **Write Through** – a copy of the input data appears at the output of the same port during a write cycle. This mode is supported for all data widths.
3. **Read-Before-Write** – when new data is being written, the old content of the address appears at the output. This mode is supported for x9, x18 and x36 data widths.

### Memory Core Reset

The memory array in the EBR utilizes latches at the A and B output ports. These latches can be reset asynchronously.  $RSTA$  and  $RSTB$  are local signals, which reset the output latches associated with Port A and Port B respectively. The Global Reset (GSRN) signal resets both ports. The output data latches and associated resets for both ports are as shown in Figure 2-15.

Figure 2-15. Memory Core Reset

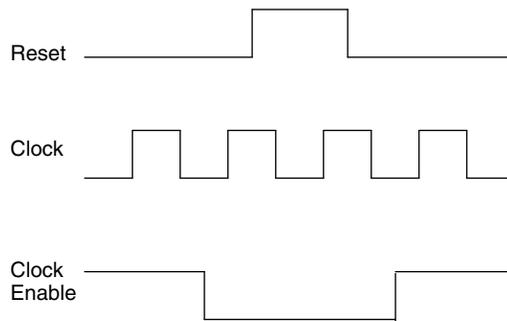


For further information on sysMEM EBR block, see the details of additional technical documentation at the end of this data sheet.

**EBR Asynchronous Reset**

EBR asynchronous reset or GSR (if used) can only be applied if all clock enables are low for a clock cycle before the reset is applied and released a clock cycle after the reset is released, as shown in Figure 2-16. The GSR input to the EBR is always asynchronous.

Figure 2-16. EBR Asynchronous Reset (Including GSR) Timing Diagram



If all clock enables remain enabled, the EBR asynchronous reset or GSR may only be applied and released after the EBR read and write clock inputs are in a steady state condition for a minimum of  $1/f_{MAX}$  (EBR clock). The reset release must adhere to the EBR synchronous reset setup time before the next active read or write clock edge.

If an EBR is pre-loaded during configuration, the GSR input must be disabled or the release of the GSR during device Wake Up must occur before the release of the device I/Os becoming active.

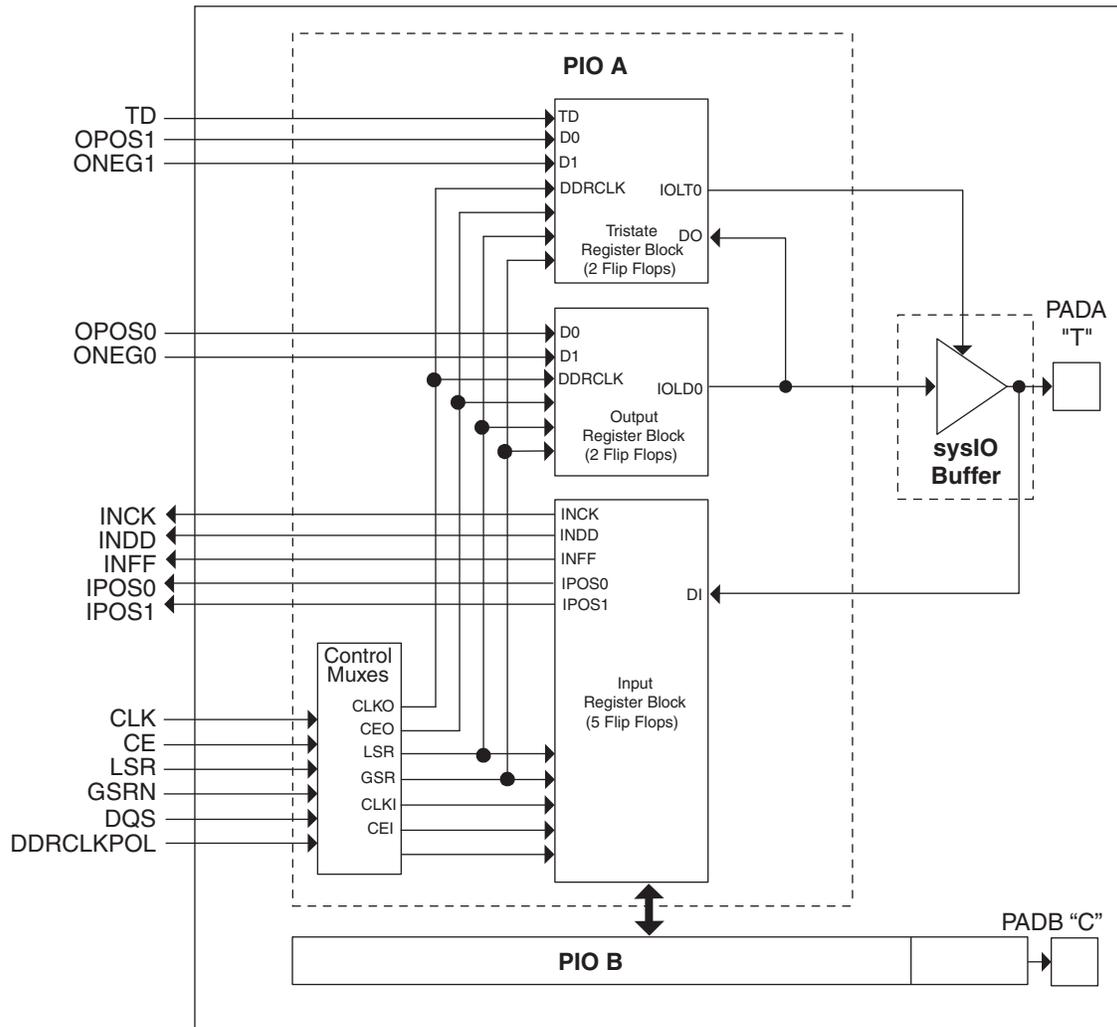
These instructions apply to all EBR RAM and ROM implementations.

Note that there are no reset restrictions if the EBR synchronous reset is used and the EBR GSR input is disabled.

**Programmable I/O Cells (PICs)**

Each PIC contains two PIOs connected to their respective sysIO Buffers which are then connected to the PADS as shown in Figure 2-17. The PIO Block supplies the output data (DO) and the Tri-state control signal (TO) to sysIO buffer, and receives input from the buffer.

Figure 2-17. PIC Diagram



In the LatticeXP family, seven PIOs or four (3.5) PICs are grouped together to provide two LVDS differential pairs, one PIC pair and one single I/O, as shown in Figure 2-18.

Two adjacent PIOs can be joined to provide a differential I/O pair (labeled as “T” and “C”). The PAD Labels “T” and “C” distinguish the two PIOs. Only the PIO pairs on the left and right edges of the device can be configured as LVDS transmit/receive pairs.

One of every 14 PIOs (a group of 8 PICs) contains a delay element to facilitate the generation of DQS signals as shown in Figure 2-19. The DQS signal feeds the DQS bus which spans the set of 13 PIOs (8 PICs). The DQS signal from the bus is used to strobe the DDR data from the memory into input register blocks. This interface is designed for memories that support one DQS strobe per eight bits of data.

The exact DQS pins are shown in a dual function in the Logic Signal Connections table in this data sheet. Additional detail is provided in the Signal Descriptions table in this data sheet.

Figure 2-18. Group of Seven PIOs

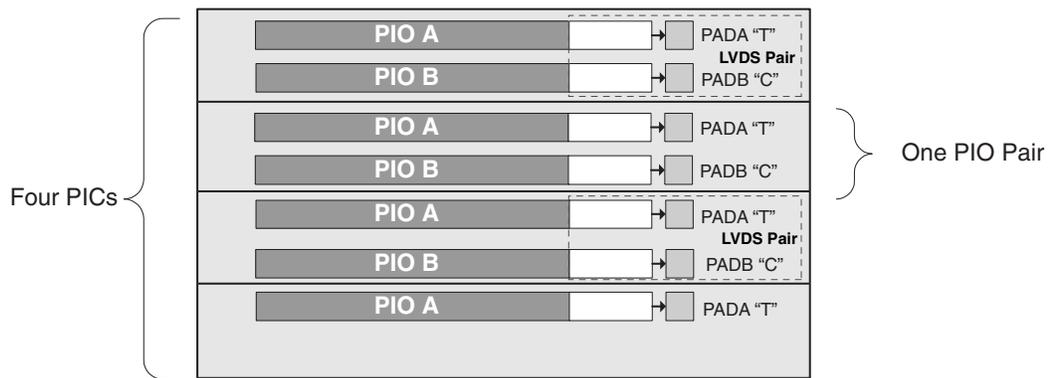
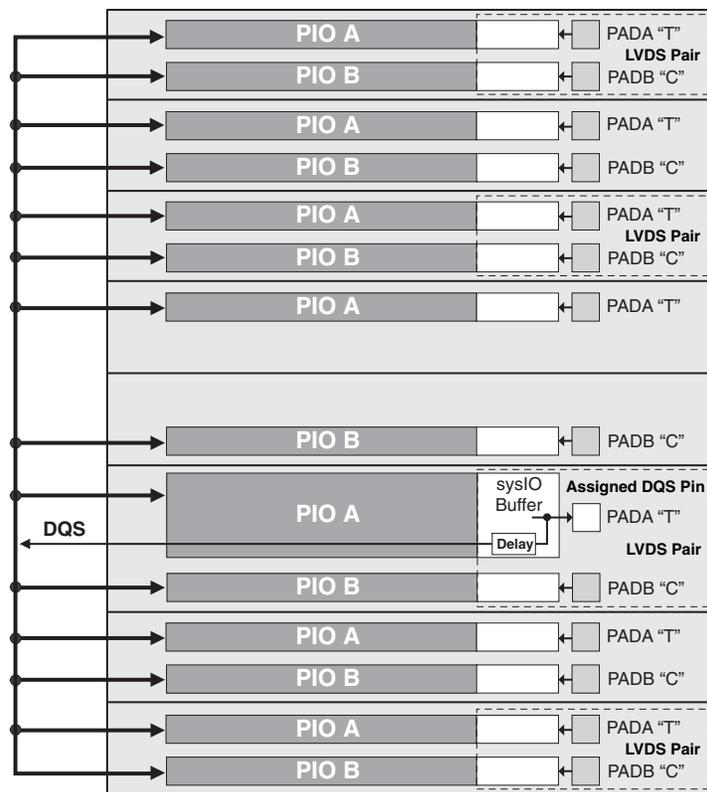


Figure 2-19. DQS Routing



## PIO

The PIO contains four blocks: an input register block, output register block, tristate register block and a control logic block. These blocks contain registers for both single data rate (SDR) and double data rate (DDR) operation along with the necessary clock and selection logic. Programmable delay lines used to shift incoming clock and data signals are also included in these blocks.

### Input Register Block

The input register block contains delay elements and registers that can be used to condition signals before they are passed to the device core. Figure 2-20 shows the diagram of the input register block.

Input signals are fed from the sysIO buffer to the input register block (as signal DI). If desired the input signal can bypass the register and delay elements and be used directly as a combinatorial signal (INDD), a clock (INCK) and

in selected blocks the input to the DQS delay block. If one of the bypass options is not chosen, the signal first passes through an optional delay block. This delay, if selected, ensures no positive input-register hold-time requirement when using a global clock.

The input block allows two modes of operation. In the single data rate (SDR) the data is registered, by one of the registers in the single data rate sync register block, with the system clock. In the DDR Mode two registers are used to sample the data on the positive and negative edges of the DQS signal creating two data streams, D0 and D2. These two data streams are synchronized with the system clock before entering the core. Further discussion on this topic is in the DDR Memory section of this data sheet.

Figure 2-21 shows the input register waveforms for DDR operation and Figure 2-22 shows the design tool primitives. The SDR/SYNC registers have reset and clock enable available.

The signal DDRCLKPOL controls the polarity of the clock used in the synchronization registers. It ensures adequate timing when data is transferred from the DQS to the system clock domain. For further discussion of this topic, see the DDR memory section of this data sheet.

**Figure 2-20. Input Register Diagram**

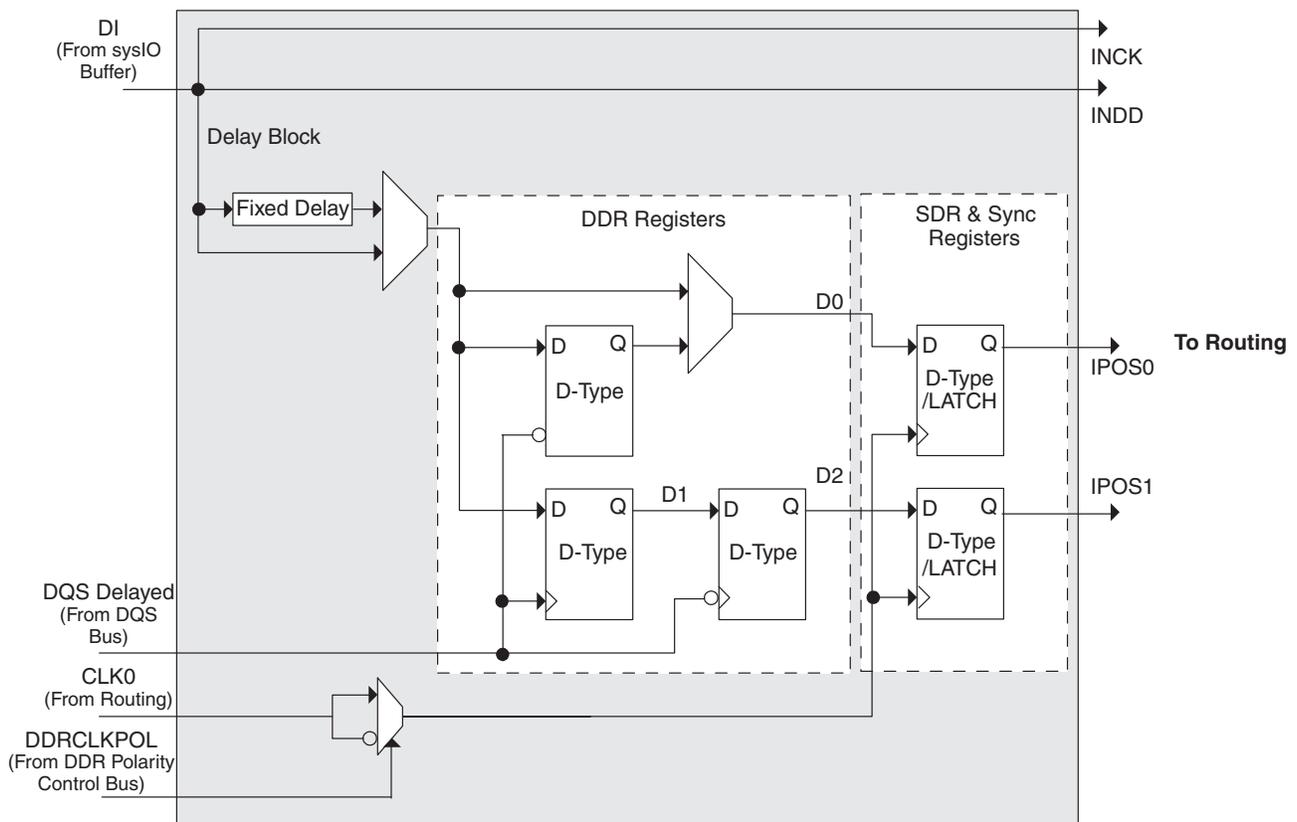


Figure 2-21. Input Register DDR Waveforms

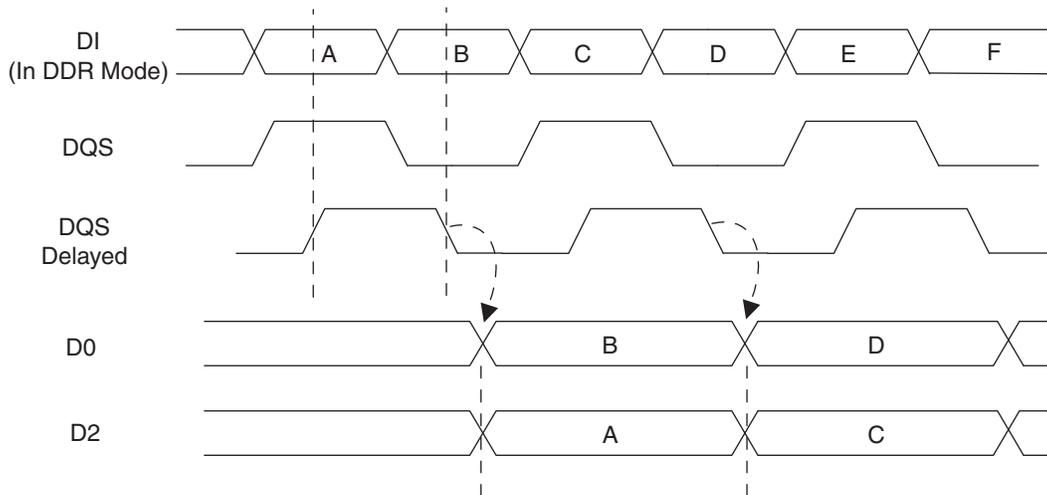
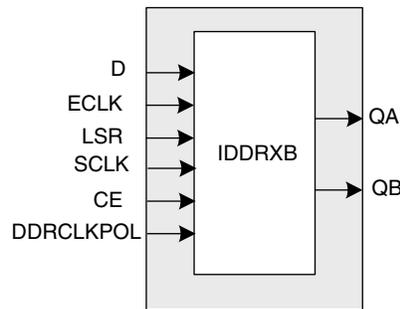


Figure 2-22. INDDRXB Primitive



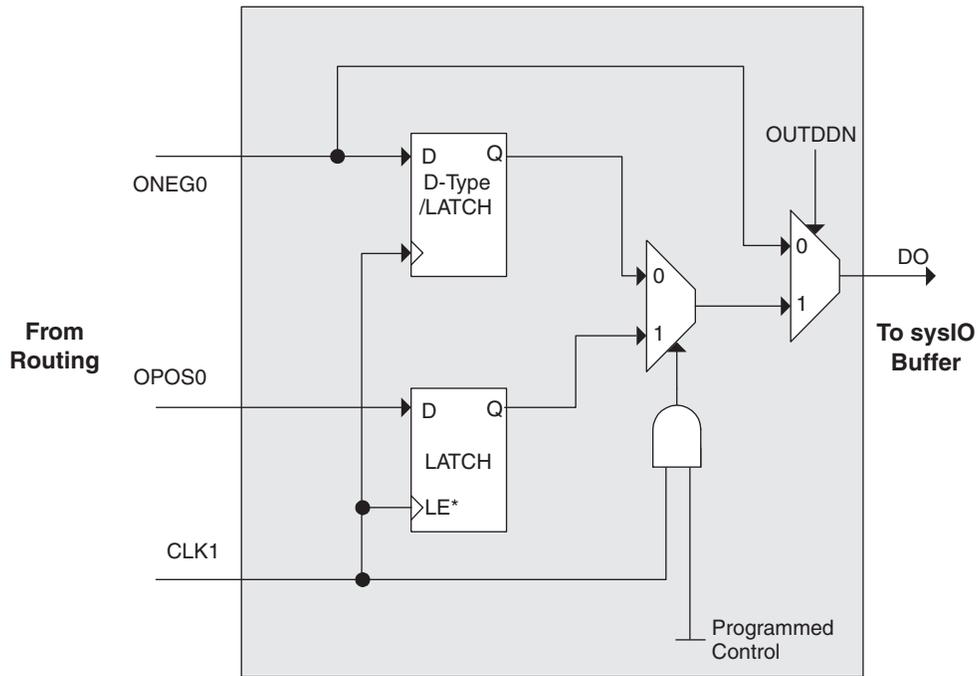
**Output Register Block**

The output register block provides the ability to register signals from the core of the device before they are passed to the sysIO buffers. The block contains a register for SDR operation that is combined with an additional latch for DDR operation. Figure 2-23 shows the diagram of the Output Register Block.

In SDR mode, ONEG0 feeds one of the flip-flops that then feeds the output. The flip-flop can be configured as a D-type or as a latch. In DDR mode, ONEG0 is fed into one register on the positive edge of the clock and OPOS0 is latched. A multiplexer running off the same clock selects the correct register for feeding to the output (D0).

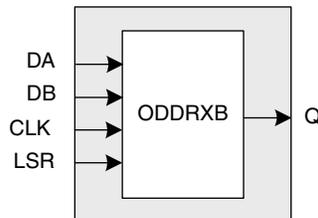
Figure 2-24 shows the design tool DDR primitives. The SDR output register has reset and clock enable available. The additional register for DDR operation does not have reset or clock enable available.

Figure 2-23. Output Register Block



\*Latch is transparent when input is low.

Figure 2-24. ODDRXB Primitive

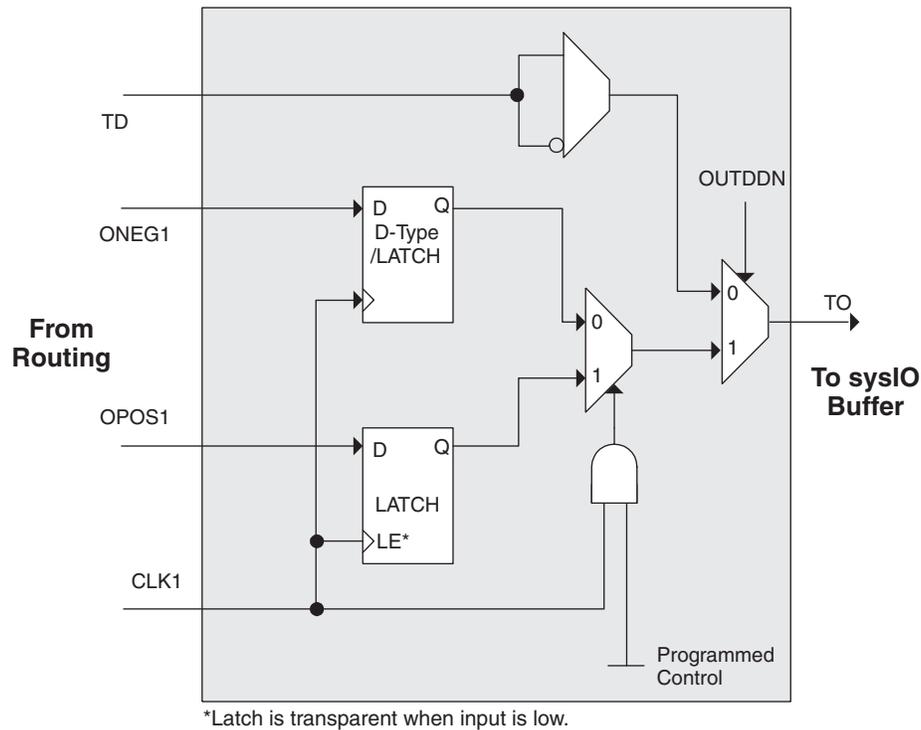


**Tristate Register Block**

The tristate register block provides the ability to register tri-state control signals from the core of the device before they are passed to the sysIO buffers. The block contains a register for SDR operation and an additional latch for DDR operation. Figure 2-25 shows the diagram of the Tristate Register Block.

In SDR mode, ONEG1 feeds one of the flip-flops that then feeds the output. The flip-flop can be configured a D-type or latch. In DDR mode, ONEG1 is fed into one register on the positive edge of the clock and OPOS1 is latched. A multiplexer running off the same clock selects the correct register for feeding to the output (D0).

Figure 2-25. Tristate Register Block



**Control Logic Block**

The control logic block allows the selection and modification of control signals for use in the PIO block. A clock is selected from one of the clock signals provided from the general purpose routing and a DQS signal provided from the programmable DQS pin. The clock can optionally be inverted.

The clock enable and local reset signals are selected from the routing and optionally inverted. The global tristate signal is passed through this block.

**DDR Memory Support**

Implementing high performance DDR memory interfaces requires dedicated DDR register structures in the input (for read operations) and in the output (for write operations). As indicated in the PIO Logic section, the LatticeXP devices provide this capability. In addition to these registers, the LatticeXP devices contain two elements to simplify the design of input structures for read operations: the DQS delay block and polarity control logic.

**DLL Calibrated DQS Delay Block**

Source Synchronous interfaces generally require the input clock to be adjusted in order to correctly capture data at the input register. For most interfaces a PLL is used for this adjustment, however in DDR memories the clock (referred to as DQS) is not free running so this approach cannot be used. The DQS Delay block provides the required clock alignment for DDR memory interfaces.

The DQS signal (selected PIOs only) feeds from the PAD through a DQS delay element to a dedicated DQS routing resource. The DQS signal also feeds the polarity control logic which controls the polarity of the clock to the sync registers in the input register blocks. Figures 2-26 and 2-27 show how the polarity control logic are routed to the PIOs.

The temperature, voltage and process variations of the DQS delay block are compensated by a set of calibration (6-bit bus) signals from two DLLs on opposite sides of the device. Each DLL compensates DQS Delays in its half of the device as shown in Figure 2-27. The DLL loop is compensated for temperature, voltage and process variations by the system clock and feedback loop.

Figure 2-26. DQS Local Bus

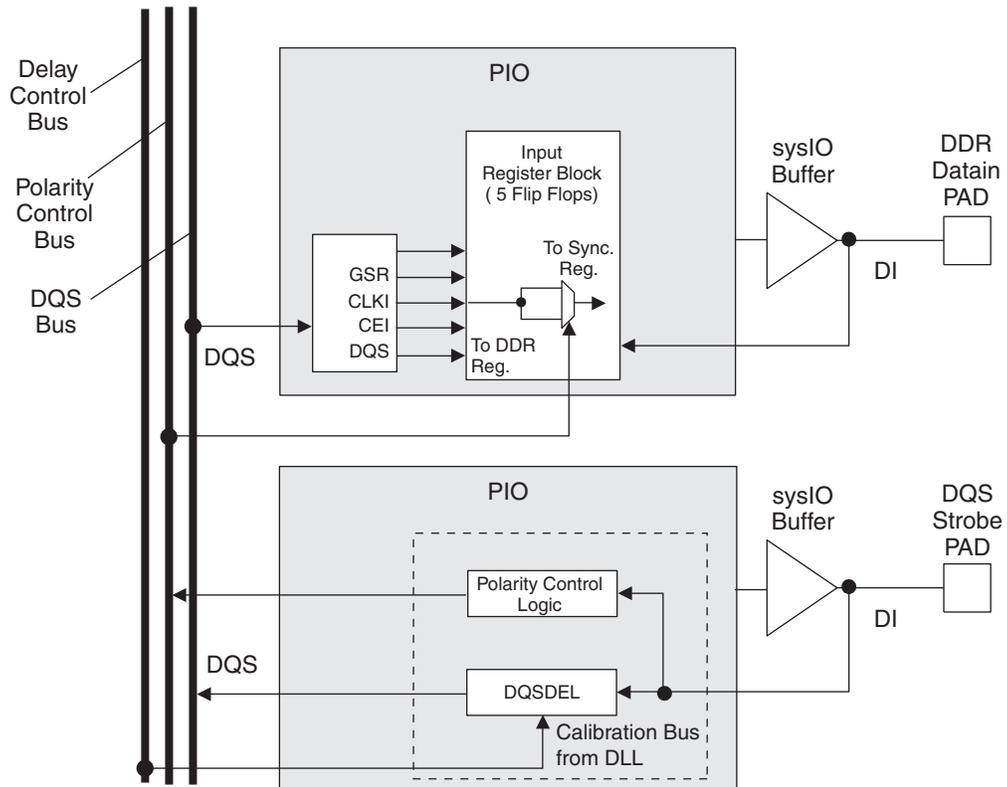
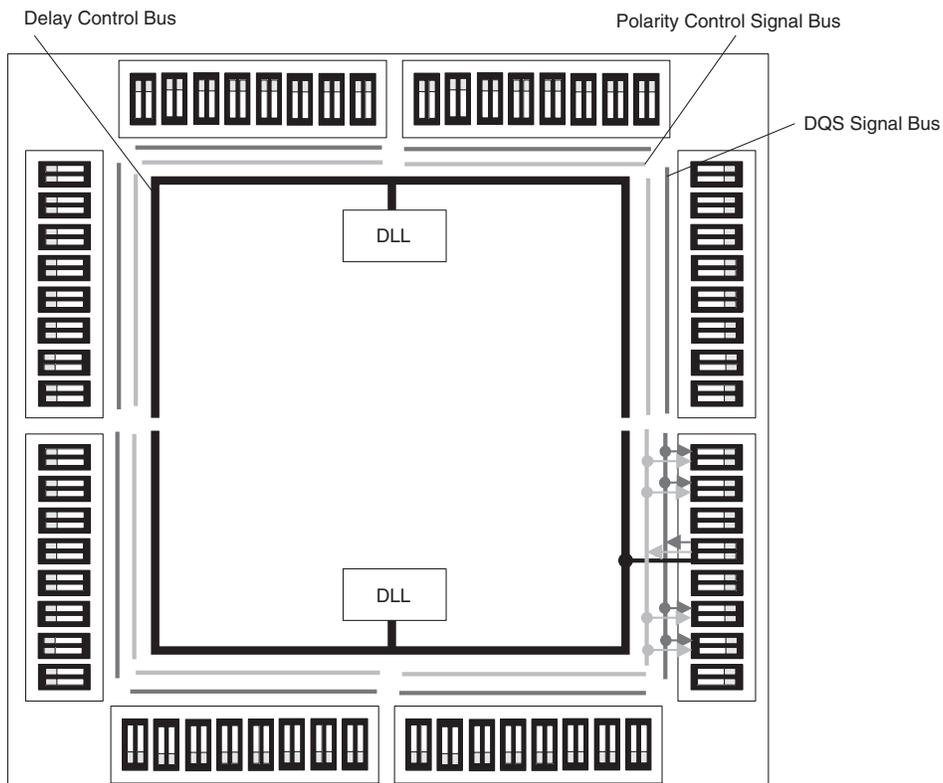


Figure 2-27. DLL Calibration Bus and DQS/DQS Transition Distribution



---

## Polarity Control Logic

In a typical DDR Memory interface design, the phase relation between the incoming delayed DQS strobe and the internal system Clock (during the READ cycle) is unknown.

The LatticeXP family contains dedicated circuits to transfer data between these domains. To prevent setup and hold violations at the domain transfer between DQS (delayed) and the system Clock a clock polarity selector is used. This changes the edge on which the data is registered in the synchronizing registers in the input register block. This requires evaluation at the start of the each READ cycle for the correct clock polarity.

Prior to the READ operation in DDR memories DQS is in tristate (pulled by termination). The DDR memory device drives DQS low at the start of the preamble state. A dedicated circuit detects this transition. This signal is used to control the polarity of the clock to the synchronizing registers.

## sysIO Buffer

Each I/O is associated with a flexible buffer referred to as a sysIO buffer. These buffers are arranged around the periphery of the device in eight groups referred to as Banks. The sysIO buffers allow users to implement the wide variety of standards that are found in today's systems including LVCMOS, SSTL, HSTL, LVDS and LVPECL.

### sysIO Buffer Banks

LatticeXP devices have eight sysIO buffer banks; each is capable of supporting multiple I/O standards. Each sysIO bank has its own I/O supply voltage ( $V_{CCIO}$ ), and two voltage references  $V_{REF1}$  and  $V_{REF2}$  resources allowing each bank to be completely independent from each other. Figure 2-28 shows the eight banks and their associated supplies.

In the LatticeXP devices, single-ended output buffers and ratioed input buffers (LVTTTL, LVCMOS, PCI and PCI-X) are powered using  $V_{CCIO}$ . LVTTTL, LVCMOS33, LVCMOS25 and LVCMOS12 can also be set as a fixed threshold input independent of  $V_{CCIO}$ . In addition to the bank  $V_{CCIO}$  supplies, the LatticeXP devices have a  $V_{CC}$  core logic power supply, and a  $V_{CCAUX}$  supply that power all differential and referenced buffers.

Each bank can support up to two separate VREF voltages, VREF1 and VREF2 that set the threshold for the referenced input buffers. In the LatticeXP devices, a dedicated pin in a bank can be configured to be a reference voltage supply pin. Each I/O is individually configurable based on the bank's supply and reference voltages.