FPGA Libraries Reference Guide

December 2013

Copyright

Trademarks

Disclaimers

Type Conventions Used in This Document

<table>
<tr><th>Convention</th><th>Meaning or Use</th></tr>
<tr><td>Bold</td><td>Items in the user interface that you select or click. Text that you type into the user interface.</td></tr>
<tr><td><<i>Italic</i>></td><td>Variables in commands, code syntax, and path names.</td></tr>
<tr><td>Ctrl+L</td><td>Press the two keys at the same time.</td></tr>
<tr><td><code>Courier</code></td><td>Code examples. Messages, reports, and prompts from the software.</td></tr>
<tr><td><code>...</code></td><td>Omitted material in a line of code.</td></tr>
<tr><td><code>.</code>
<code>.</code>
<code>.</code></td><td>Omitted lines in code and report examples.</td></tr>
<tr><td><code>[ ]</code></td><td>Optional items in syntax descriptions. In bus specifications, the brackets are required.</td></tr>
<tr><td><code>( )</code></td><td>Grouped items in syntax descriptions.</td></tr>
<tr><td><code>{ }</code></td><td>Repeatable items in syntax descriptions.</td></tr>
<tr><td><code>|</code></td><td>A choice between items in syntax descriptions.</td></tr>
</table>

Contents

FPGA Libraries Reference Guide

Lattice supports some libraries used in designing FPGAs with different device architectures in a number of CAE synthesis, schematic capture, and simulation platforms. These libraries are the main front-end design libraries for Lattice FPGAs. Logic design primitives in these libraries offer flexibility and efficiency to facilitate building specific applications with Lattice devices. Many primitives can be used in multiple Lattice device architectures. With Schematic Editor, you can place the primitive symbols from the Lattice Symbol Library, lattice.lib, which is composed of primitives compatible with most Lattice FPGA device families. For macro-sized primitives like architectural blocks, arithmetic, or memories, use IPexpress™ to configure and generate schematic symbols and files for implementation. You can also use physical macros that you create in EPIC. See the appropriate topic in the online help system for more information.

A specific primitive can be found according to the device family and functional category. Primitives available to each of the following device families are listed according to appropriate functional categories.

- Primitive Library - LatticeECP/EC and LatticeXP
- Primitive Library - LatticeECP2/M
- Primitive Library - LatticeECP3
- Primitive Library - LatticeSC/M
- Primitive Library - LatticeXP2
- Primitive Library - MachXO
- Primitive Library - MachXO2 and Platform Manager 2
- Primitive Library - MachXO3L
- Primitive Library - Platform Manager
- Primitive Library - Platform Manager 2

The “Alphanumeric Primitives List” section contains descriptions of all available primitives in their alphanumeric order. The following information is provided for each primitive, where applicable:

Table 1: Information Provided for Each Primitive

Fields	Description
Name	Primitive name
Definition	Brief description of primitive

Table 1: Information Provided for Each Primitive (Continued)

Fields	Description
Architectures Supported	Index of FPGA families supported by the primitive
Port Interface Symbol	Graphic to represent the primitive port interface. Data, address, clock, clock enable type ports appear on the left-hand side of the block. Synchronous control ports appear on the top of the symbol, asynchronous control ports on the bottom, and output ports on the right-hand side of the block. Some of the graphic symbols are shown in bus notation for proper layout. Those primitives must be instantiated in expanded bus notation format with each individual bit.
Attributes	Index of attributes compatible with the primitive. The first value is usually the default value, if it is not explicitly indicated. Attribute function, range, and port-to-attribute or attribute-to-attribute relationships for the primitive are noted. For descriptions of all the attributes used in primitives, see List of Primitive-Specific HDL Attributes.
Description	Detailed description of primitive function. Exceptions are identified by device family. This section sometimes includes truth table, waveform, state diagram, or other graphical methods to illustrate behavior. References to appropriate Lattice technical notes are listed.
Port Description	Index of port names, polarity, and function.
Connectivity Rules	Some primitive ports, such as DDR and DSP blocks, are connected to dedicated routing with source or loads related to other primitives. This section describes connection rules.

Naming Conventions

The table shows the convention for naming all of the sequential primitives. Each primitive is identified using up to seven characters.

Flip-Flop/Latch Naming Conventions (name = abcdef)

Table 2: Naming Conventions

a=	F - Static implementation
b=	D - D type flip-flop
	J - J/K type flip-flop
	L - Cells contain a positive select front end (loadable)
	N - Cells contain a negative select front end (loadable)
	S - R-S type flip-flop
	T - Toggle type flip-flop
c=	Value - Number of clocks
d=	This parameter identifies the enable capability.
	S - No enable input
	P - Positive-level enable
	N - Negative-level enable
e=	This parameter identifies the clock capability
	1 - Positive-level sense (latch)
	2 - Negative edge-triggered (flip-flop)
	3 - Positive edge-triggered (flip-flop)
	4 - Negative-level sense (latch)
f=	A - No clear or preset inputs
	B - Positive-level asynchronous preset
	D - Positive-level asynchronous clear
	I - Positive-level synchronous clear
	J - Positive-level synchronous preset
	X - Standard primitive where GSR asynchronously clears or presets the flip-flop depending upon the function of the local clear or preset. If no local clear or preset is present (f=A), then GSR clears the register element.
	Y - Primitive is preset using GSR rather than cleared
	Z - Primitive not compatible with similar primitives available in the standard cell library

Example:

FD1P3BX is a single clock, positive edge-triggered, static, D-type flip-flop with a positive-level enable and a positive-level asynchronous preset.

Table 3: Example: FD1P3BX

FD1P3BX	a = F	Static cell
	b = D	D type
	c = 1	Single clock
	d = P	Positive-level enable
	e = 3	Positive edge-triggered
	f = B	Positive-level asynchronous preset

When creating designs from schematic or synthesis, please observe the following rules:

- Component, net, site, or instance names should be unique and independent of case. For example, if a net in a design is named "d0", then you cannot have an instance of AND2 named "D0".
- Component, net, site, or instance names cannot be any cell name.
- Component, net, site, or instance names cannot be GND, PWR, VSS, VDD, GSR, GSRNET, TSALL, TSALLNET, and so forth.
- Component, net, site, or instance names cannot contain preference keywords such as DIN, DOUT, SITE, COMP, and so forth.
- Component, net, site, or instance names cannot contain names starting with 0–9, /, \, +, –, and other special characters.
- Component, net, site, or instance names cannot be named using VHDL or Verilog keywords such as IN, OUT, INOUT, and so forth.

Note the following about numbering used throughout this Help:

- Least significant bits (LSBs) on a primitive are always determined by the lowest integer value (usually zero) expressed in a pin name input or output in a pin grouping just as most significant bits (MSBs) are always determined by the highest integer value.

For example, out of the pin group containing pins A0, A1, A2, and A3, A0 is the LSB and A3 is the MSB. The LSB or MSB is not affected by the order in which pins are numbered (that is, whether or not they are in ascending or descending order).

Memory Primitives Overview

The architectures of various Lattice FPGAs provide resources for on-chip memory intensive applications. The sysMEM™ embedded block RAM (EBR) complements the distributed PFU-based memory. Single-port RAM, dual-port RAM, FIFO, and ROM memories can be constructed using the EBR. LUTs and PFU can implement distributed single-port RAM, dual-port RAM, and ROM. The Lattice Diamond software enables you to integrate the EBR- and PFU-based memories in various device families.

The EBR-based and PFU-based memory primitives are listed in this document. Designers can utilize the memory primitives in several ways via the IPexpress tool in the Diamond software. IPexpress allows you to specify the attributes of the memory application, such as required type and size. IPexpress takes the customized specification and constructs a netlist to implement the desired memory, using one or more of the memory primitives.

The available memory primitives include:

- RAM_DP (Dual Port RAM)
- RAM_DP_BE (Dual Port RAM with Byte Enable)
- RAM_DP_TRUE (True Dual Port RAM)
- RAM_DP_TRUE_BE (True Dual Port RAM with Byte Enable)
- RAM_DQ (Single Port RAM)
- RAM_DQ_BE (Single Port RAM with Byte Enable)
- ROM (Read Only Memory)
- Distributed_DPRAM (Distributed Dual Port RAM)
- Distributed_ROM (Distributed Read Only Memory)
- Distributed_SPRAM (Distributed Single Port RAM)
- FIFO (First In First Out Single Clock)
- FIFO_DC (First In First Out Dual Clock)
- Shift Registers (Distributed RAM Shift Register)

RAM_DP (Dual Port RAM)

Implementation: EBR

Table 4: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	WrAddress	Bus	(pmi_wr_addr_width – 1) : 0
I	RdAddress	Bus	(pmi_rd_addr_width – 1) : 0
I	Data	Bus	(pmi_wr_data_width – 1) : 0
I	RdClock	Bit	N/A
I	RdClockEn	Bit	N/A
I	Reset	Bit	N/A
I	WrClock	Bit	N/A
I	WrClockEn	Bit	N/A
I	WE	Bit	N/A
O	Q	Bus	(pmi_rd_data_width – 1) : 0

Table 5: Parameters

Name	Value	Default
pmi_rd_addr_depth [1]	2 to 65536	512
pmi_rd_addr_width [1]	1 to 16	9
pmi_rd_data_width	1 to 256	18
pmi_wr_addr_depth [1]	2 to 65536	512
pmi_wr_addr_width [1]	1 to 16	9
pmi_wr_data_width	1 to 256	18
pmi_regmode	"reg" | "noreg"	"reg"
pmi_gsr	"enable" | "disable"	"disable"
pmi_resetmode [2]	"async" | "sync"	"sync"
pmi_optimization	"area" | "speed"	"speed"
pmi_init_file	*<string>*	"none"
pmi_init_file_format	"binary" | "hex"	"binary"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "XO" | "XO2" | "ECP3" |"LPTM" |"LPTM2"	"EC"

Note

1. For SC, SCM, ECP2, ECP2M, XP2, and ECP3 FPGA device families, the read/write address depth ranges from 2 to 131072. The address width ranges from 1 to 17.
2. For ECP3, "async" is not a valid option.

RAM_DP_BE (Dual Port RAM with Byte Enable)

Implementation: EBR

Table 6: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	WrAddress	Bus	(pmi_wr_addr_width – 1) : 0
I	RdAddress	Bus	(pmi_rd_addr_width – 1) : 0
I	Data	Bus	(pmi_wr_data_width – 1) : 0
I	RdClock	Bit	N/A
I	RdClockEn	Bit	N/A
I	Reset	Bit	N/A
I	WrClock	Bit	N/A
I	WrClockEn	Bit	N/A
I	WE	Bit	N/A
O	Q	Bus	(pmi_rd_data_width – 1) : 0
I	ByteEn	Bus	((pmi_wr_data_width + pmi_byte_size – 1) / pmi_byte_size – 1) : 0

Table 7: Parameters

Name	Value	Default
pmi_rd_addr_depth [1]	2 to 65536	512
pmi_rd_addr_width [1]	1 to 16	9
pmi_rd_data_width	1 to 256	18
pmi_wr_addr_depth [1]	2 to 65536	512
pmi_wr_addr_width [1]	1 to 16	9
pmi_wr_data_width	1 to 256	18
pmi_regmode	"reg" | "noreg"	"reg"
pmi_gsr	"enable" | "disable"	"disable"
pmi_resetmode [2]	"async" | "sync"	"sync"
pmi_optimization	"area" | "speed"	"speed"
pmi_init_file	*<string>*	"none"
pmi_init_file_format	"binary" | "hex"	"binary"

Table 7: Parameters (Continued)

Name	Value	Default
pmi_family	"XP2" | "SC" | "SCM" | "ECP2" | "ECP2M" | "XO2" | "LPTM2" | "ECP3"	"ECP2"
pmi_byte_size	8 | 9	9

Note

1. For SC, SCM, ECP2, ECP2M, XP2, and ECP3 FPGA device families, the read/write address depth ranges from 2 to 131072. The address width ranges from 1 to 17.
2. For ECP3, "async" is not a valid option.

RAM_DP_TRUE (True Dual Port RAM)

Implementation: EBR

Table 8: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	DataInA	Bus	(pmi_data_width_a – 1) : 0
I	DataInB	Bus	(pmi_data_width_b – 1) : 0
I	AddressA	Bus	(pmi_addr_width_a – 1) : 0
I	AddressB	Bus	(pmi_addr_width_b – 1) : 0
I	ClockA	Bit	N/A
I	ClockB	Bit	N/A
I	ClockEnA	Bit	N/A
I	ClockEnB	Bit	N/A
I	WrA	Bit	N/A
I	WrB	Bit	N/A
I	ResetA	Bit	N/A
I	ResetB	Bit	N/A
O	QA	Bus	(pmi_data_width_a – 1) : 0
O	QB	Bus	(pmi_data_width_b – 1) : 0

Table 9: Parameters

Name	Value	Default
pmi_addr_depth_a [1]	2 to 65536	512
pmi_addr_width_a [1]	1 to 16	9
pmi_data_width_a	1 to 256	18
pmi_addr_depth_b [1]	2 to 65536	512
pmi_addr_width_b [1]	1 to 16	9
pmi_data_width_b	1 to 256	18
pmi_regmode_a	"reg" | "noreg"	"reg"
pmi_regmode_b	"reg" | "noreg"	"reg"
pmi_gsr	"enable" | "disable"	"disable"
pmi_resetmode [2]	"async" | "sync"	"sync"
pmi_optimization	"area" | "speed"	"speed"

Table 9: Parameters (Continued)

Name	Value	Default
pmi_init_file	*<string>*	"none"
pmi_init_file_format	"binary" | "hex"	"binary"
pmi_write_mode_a [3]	"normal" | "writethrough" | "readbeforewrite"	"normal"
pmi_write_mode_b [3]	"normal" | "writethrough" | "readbeforewrite"	"normal"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "XO" | "XO2" | "ECP3" | "LPTM" | "LPTM2"	"EC"

Note

1. For SC, SCM, ECP2, ECP2M, XP2, and ECP3 FPGA device families, the read/write address depth ranges from 2 to 131072. The address width ranges from 1 to 17.
2. For ECP3, "async" is not a valid option.
3. The "Readbeforewrite" option is not supported by ECP2, ECP2M, XP2, SC, SCM, or ECP3.

RAM_DP_TRUE_BE (True Dual Port RAM with Byte Enable)

Implementation: EBR

Table 10: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	DataInA	Bus	(pmi_data_width_a – 1) : 0
I	DataInB	Bus	(pmi_data_width_b – 1) : 0
I	AddressA	Bus	(pmi_addr_width_a – 1) : 0
I	AddressB	Bus	(pmi_addr_width_b – 1) : 0
I	ClockA	Bit	N/A
I	ClockB	Bit	N/A
I	ClockEnA	Bit	N/A
I	ClockEnB	Bit	N/A
I	WrA	Bit	N/A
I	WrB	Bit	N/A
I	ResetA	Bit	N/A
I	ResetB	Bit	N/A
O	QA	Bus	(pmi_data_width_a – 1) : 0
O	QB	Bus	(pmi_data_width_b – 1) : 0
I	ByteEnA	Bus	((pmi_data_width_a + pmi_byte_size – 1) / pmi_byte_size – 1) : 0
I	ByteEnB	Bus	((pmi_data_width_b + pmi_byte_size – 1) / pmi_byte_size – 1) : 0

Table 11: Parameters

Name	Value	Default
pmi_addr_depth_a [1]	2 to 65536	512
pmi_addr_width_a [1]	1 to 16	9
pmi_data_width_a	1 to 256	18
pmi_addr_depth_b [1]	2 to 65536	512
pmi_addr_width_b [1]	1 to 16	9

Table 11: Parameters (Continued)

Name	Value	Default
pmi_data_width_b	1 to 256	18
pmi_regmode_a	"reg" | "noreg"	"reg"
pmi_regmode_b	"reg" | "noreg"	"reg"
pmi_gsr	"enable" | "disable"	"disable"
pmi_resetmode [2]	"async" | "sync"	"sync"
pmi_optimization	"area" | "speed"	"speed"
pmi_init_file	*<string>*	"none"
pmi_init_file_format	"binary" | "hex"	"binary"
pmi_write_mode_a [3]	"normal" | "writethrough" | "readbeforewrite"	"normal"
pmi_write_mode_b [3]	"normal" | "writethrough" | "readbeforewrite"	"normal"
pmi_family	"XP2" | "SC" | "SCM" | "ECP2" | "ECP2M" | "XO2" | "LPTM2" | "ECP3"	"ECP2"
pmi_byte_size	8 | 9	9

Note

1. For SC, SCM, ECP2, ECP2M, XP2, and ECP3 FPGA device families, the read/write address depth ranges from 2 to 131072. The address width ranges from 1 to 17.
2. For ECP3, "async" is not a valid option.
3. The "Readbeforewrite" option is not supported by ECP2, ECP2M, XP2, SC, SCM, or ECP3.

RAM_DQ (Single Port RAM)

Implementation: EBR

Table 12: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	Data	Bus	(pmi_data_width – 1) : 0
I	Address	Bus	(pmi_addr_width – 1) : 0
I	Clock	Bit	N/A
I	ClockEn	Bit	N/A
I	Reset	Bit	N/A
I	WE	Bit	N/A
O	Q	Bus	(pmi_data_width – 1) : 0

Table 13: Parameter

Name	Value	Default
pmi_addr_depth [1]	2 to 65536	512
pmi_addr_width [1]	1 to 16	9
pmi_data_width	1 to 256	18
pmi_regmode	"reg" | "noreg"	"reg"
pmi_gsr	"enable" | "disable"	"disable"
pmi_resetmode [2]	"async" | "sync"	"sync"
pmi_optimization	"area" | "speed"	"speed"
pmi_init_file	*<string>*	"none"
pmi_init_file_format	"binary" | "hex"	"binary"
pmi_write_mode [3]	"normal" | "writethrough" | "readbeforewrite"	"normal"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "XO" | "XO2" | "ECP3" | "LPTM" | "LPTM2"	"EC"

Note

1. For SC, SCM, ECP2, ECP2M, XP2, and ECP3 FPGA device families, the read/write address depth ranges from 2 to 131072. The address width ranges from 1 to 17.
2. For ECP3, "async" is not a valid option.
3. The "Readbeforewrite" option is not supported by ECP2, ECP2M, XP2, SC, SCM, or ECP3.

RAM_DQ_BE (Single Port RAM with Byte Enable)

Implementation: EBR

Table 14: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	Data	Bus	(pmi_data_width – 1) : 0
I	Address	Bus	(pmi_addr_width – 1) : 0
I	Clock	Bit	N/A
I	ClockEn	Bit	N/A
I	Reset	Bit	N/A
I	WE	Bit	N/A
O	Q	Bus	(pmi_data_width – 1) : 0
I	ByteEn	Bus	((pmi_data_width + pmi_byte_size – 1) / pmi_byte_size – 1) : 0

Table 15: Parameters

Name	Value	Default
pmi_addr_depth [1]	2 to 65536	512
pmi_addr_width [1]	1 to 16	9
pmi_data_width	1 to 256	18
pmi_regmode	"reg" | "noreg"	"reg"
pmi_gsr	"enable" | "disable"	"disable"
pmi_resetmode [2]	"async" | "sync"	"sync"
pmi_optimization	"area" | "speed"	"speed"
pmi_init_file	*<string>*	"none"
pmi_init_file_format	"binary" | "hex"	"binary"
pmi_write_mode [3]	"normal" | "writethrough" | "readbeforewrite"	"normal"
pmi_family	"XP2" | "SC" | "SCM" | "ECP2" | "ECP2M" | "XO2" | "LPTM2" | "ECP3"	"ECP2"
pmi_byte_size	8 | 9	9

Note

1. For SC, SCM, ECP2, ECP2M, XP2, and ECP3 FPGA device families, the read/write address depth ranges from 2 to 131072. The address width ranges from 1 to 17.
2. For ECP3, "async" is not a valid option.
3. The "Readbeforewrite" option is not supported by ECP2, ECP2M, XP2, SC, SCM, or ECP3.

ROM (Read Only Memory)

Implementation: EBR

Table 16: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	Address	Bus	(pmi_addr_width – 1) : 0
I	OutClock	Bit	N/A
I	OutClockEn	Bit	N/A
I	Reset	Bit	N/A
O	Q	Bus	(pmi_data_width – 1) : 0

Table 17: Parameters

Name	Value	Default
pmi_addr_depth [1]	2 to 65536	512
pmi_addr_width [1]	1 to 16	9
pmi_data_width	1 to 256	18
pmi_regmode	"reg" | "noreg"	"reg"
pmi_gsr	"enable" | "disable"	"disable"
pmi_resetmode [2]	"async" | "sync"	"sync"
pmi_optimization	"area" | "speed"	"speed"
pmi_init_file	*<string>*	"none"
pmi_init_file_format	"binary" | "hex"	"binary"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "XO" | "XO2" | "ECP3" | "LPTM" | "LPTM2"	"EC"

Note

1. For SC, SCM, ECP2, ECP2M, XP2, and ECP3 FPGA device families, the read/write address depth ranges from 2 to 131072. The address width ranges from 1 to 17.
2. For ECP3, "async" is not a valid option.

Distributed_DPRAM (Distributed Dual Port RAM)

Implementation: LUT

Table 18: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	WrAddress	Bus	(pmi_addr_width – 1) : 0
I	Data	Bus	(pmi_data_width – 1) : 0
I	WrClock	Bit	N/A
I	WrClockEn	Bit	N/A
I	WE	Bit	N/A
I	RdAddress	Bus	(pmi_addr_width – 1) : 0
I	RdClock	Bit	N/A
I	RdClockEn	Bit	N/A
I	Reset	Bit	N/A
O	Q	Bus	(pmi_data_width – 1) : 0

Table 19: Parameters

Name	Value	Default
pmi_addr_depth	2 to 8192	32
pmi_addr_width	1 to 13	5
pmi_data_width	1 to 256	8
pmi_regmode	"reg" | "noreg"	"reg"
pmi_init_file [1]	*<string>*	"none"
pmi_init_file_format [1]	"binary" | "hex"	"binary"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "XO" | "XO2" | "ECP3" | "LPTM" | "LPTM2"	"EC"

Note

1. The pmi_init_file and pmi_init_file_format parameters are not supported for EC, XP, ECP, ECP2, ECP2M, LPTM, and XO device families in distributed mode.

Distributed_ROM (Distributed Read Only Memory)

Implementation: LUT

Table 20: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	Address	Bus	(pmi_addr_width – 1) : 0
I	OutClock	Bit	N/A
I	OutClockEn	Bit	N/A
I	Reset	Bit	N/A
O	Q	Bus	(pmi_data_width – 1) : 0

Table 21: Parameters

Name	Value	Default
pmi_addr_depth	2 to 8192	32
pmi_addr_width	1 to 13	5
pmi_data_width	1 to 128	8
pmi_regmode	"reg" | "noreg"	"reg"
pmi_init_file	*<string>*	"none"
pmi_init_file_format	"binary" | "hex"	"binary"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "XO" | "XO2" | "ECP3" | "LPTM" | "LPTM2"	"EC"

Distributed_SPRAM (Distributed Single Port RAM)

Implementation: LUT

Table 22: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	Address	Bus	(pmi_addr_width – 1) : 0
I	Data	Bus	(pmi_data_width – 1) : 0
I	Clock	Bit	N/A
I	ClockEn	Bit	N/A
I	WE	Bit	N/A
I	Reset	Bit	N/A
O	Q	Bus	(pmi_data_width – 1) : 0

Table 23: Parameters

Name	Value	Default
pmi_addr_depth	2 to 8192	32
pmi_addr_width	1 to 13	5
pmi_data_width	1 to 128	8
pmi_regmode	"reg" | "noreg"	"reg"
pmi_init_file [1]	*<string>*	"none"
pmi_init_file_format [1]	"binary" | "hex"	"binary"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "XO" | "XO2" | "ECP3" | "LPTM" | "LPTM2"	"EC"

Note

1. The pmi_init_file and pmi_init_file_format parameters are not supported for EC, XP, ECP, ECP2, ECP2M, LPTM, and XO device families in distributed mode.

FIFO (First In First Out Single Clock)

Implementation: EBR, LUT

Table 24: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	Data	Bus	(pmi_data_width – 1) : 0
I	Clock	Bit	N/A
I	WrEn	Bit	N/A
I	RdEn	Bit	N/A
I	Reset	Bit	N/A
O	Q	Bus	(pmi_data_width – 1) : 0
O	Empty	Bit	N/A
O	Full	Bit	N/A
O	AlmostEmpty	Bit	N/A
O	AlmostFull	Bit	N/A

Table 25: Parameters

Name	Value	Default
pmi_data_depth [1]	2 to 65536	256
pmi_data_width	1 to 256	8
pmi_almost_empty_flag	1 to 512	4
pmi_almost_full_flag	1 to 512	252
pmi_full_flag	1 to pmi_data_depth	256
pmi_empty_flag	0	0
pmi_regmode	"reg" | "noreg" | "outreg" | "outreg_rden"	"reg"
pmi_implementation	"EBR" | "LUT"	"EBR"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "ECP3" | "LPTM"	"EC"

Note

1. The device depth for the LUT based FIFO ranges from 2 to 8192. The XP2, ECP2, ECP2M, SC, SCM, and ECP3 device family address depths range from 2 to 131072. The EC, ECP, and XP families range from 2 to 65536.

FIFO_DC (First In First Out Dual Clock)

Implementation: EBR, LUT

Table 26: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	Data	Bus	(pmi_data_width_w – 1) : 0
I	WrClock	Bit	N/A
I	RdClock	Bit	N/A
I	WrEn	Bit	N/A
I	RdEn	Bit	N/A
I	Reset	Bit	N/A
I	RPReset	Bit	N/A
O	Q	Bus	(pmi_data_width_r – 1) : 0
O	Empty	Bit	N/A
O	Full	Bit	N/A
O	AlmostEmpty	Bit	N/A
O	AlmostFull	Bit	N/A

Table 27: Parameters

Name	Value	Default
pmi_data_depth_w [1]	2 to 131072	256
pmi_data_depth_r [1]	2 to 131072	256
pmi_data_width_w	1 to 256	18
pmi_data_width_r	1 to 256	18
pmi_full_flag	1 to pmi_data_depth_r	256
pmi_empty_flag	0	0
pmi_almost_full_flag	1 to pmi_data_depth_w	252
pmi_almost_empty_flag	1 to pmi_data_depth_r	4
pmi_regmode	"reg" | "noreg" | "outreg" | "outreg_rden"	"reg"
pmi_resetmode [2]	"async" | "sync"	"async"

Table 27: Parameters (Continued)

Name	Value	Default
pmi_implementation	"EBR" | "LUT"	"EBR"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "XO" | "XO2" | "ECP3" | "LPTM" | "LPTM2"	"EC"

Note

1. The device depth for the LUT-based FIFO_DC ranges from 2 to 8192. The XP2, ECP2, ECP2M, SC, SCM, and ECP3 device family data depths range from 2 to 131072. The EC, ECP, and XP families range from 2 to 65536. The XO, LPTM, LPTM2 and XO2 data depths range from 2 to 16384. The SC, SCM, XO, XO2, LPTM2 and LPTM device families support different read/write depths and different read/write data widths. The depth settings of SC and SCM are 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65535, and 131072. The depth settings for XO and LPTM are 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, and 16384.
2. The SC, SCM, XO, XO2, LPTM2 and LPTM device families support the pmi_resetmode parameter. The pmi_resetmode applies to the EBR memory implementation.

Shift Registers (Distributed RAM Shift Register)

Implementation: LUT, EBR [1]

Table 28: Pins

Input/Output	Port Name	Type	Size (Buses Only)
I	Din	Bus	(pmi_data_width – 1) : 0
I	Addr	Bus	(pmi_max_width – 1) : 0
I	Clock	Bit	N/A
I	ClockEn	Bit	N/A
O	Q	Bus	(pmi_data_width – 1) :0

Table 29: Parameters

Name	Value	Default
pmi_data_width	1 to 256	16
pmi_regmode	"reg" | "noreg"	"reg"
pmi_shiftreg_type [2]	"fixed" | "variable" | "lossless"	"fixed"
pmi_num_shift	2 to 1024	16
pmi_num_width	0 to 10	4
pmi_max_shift	2 to 1024	16
pmi_max_width	0 to 10	4
pmi_init_file [3]	*<string>*	"none"
pmi_init_file_format [3]	"binary" | "hex"	"binary"
pmi_family	"EC" | "XP" | "XP2" | "SC" | "SCM" | "ECP" | "ECP2" | "ECP2M" | "XO" | "XO2" | "ECP3" | "LPTM" | "LPTM2"	"EC"

Note

1. EBR implementation is not available in ispLEVER 6.0 or earlier.
2. The lossless mode is not supported in ispLEVER 6.0 or earlier.
3. The pmi_init_file and pmi_init_file_format parameters are not supported for EC, XP, ECP, ECP2, ECP2M, LPTM, and XO device families in distributed mode.

Primitive Library - LatticeECP/EC and LatticeXP

This library includes compatible FPGA primitives supported by the LatticeXP and LatticeECP/EC device families.

References

For further information, a variety of technical documents for the LatticeECP/EC family and LatticeXP family are available on the Lattice Web site.

- TN1049 - LatticeECP/EC and LatticeXP sysCLOCK PLL Design and Usage Guide
- TN1050 - LatticeECP/EC and XP DDR Usage Guide
- TN1051 - Memory Usage Guide for LatticeECP/EC and LatticeXP Devices
- TN1052 - Estimating Power Using Power Calculator for LatticeECP/EC and LatticeXP Devices
- TN1053 - LatticeECP/EC sysCONFIG Usage Guide

- TN1056 - LatticeECP/EC and LatticeXP sysIO Usage Guide
- TN1057 - LatticeECP sysDSP Usage Guide
- TN1082 - LatticeXP sysCONFIG Usage Guide
- IEEE 1149.1 Boundary Scan Testability in Lattice Devices

Table 30: Adders Subtractors

FADD2	2 Bit Fast Adder
FSUB2	2 Bit Fast Subtractor (two's complement)
FADSU2	2 Bit Fast Adder/Subtractor (two's complement)

Table 31: Comparators

AGEB2	A Greater Than Or Equal To B (2 bit)
ALEB2	A Less Than Or Equal To B (2 bit)
ANEB2	A Not Equal To B (2 bit)

Table 32: Counters

CB2	Combinational Logic for 2-Bit Bidirectional Counter
CD2	Combinational Logic for 2 Bit Down Counter
CU2	Combinational Logic for 2 Bit Up Counter

Table 33: Loadable Counters

LB2P3AX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB2P3AY	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset
LB2P3BX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LB2P3DX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB2P3IX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)

Table 33: Loadable Counters (Continued)

LB2P3JX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LB4P3AX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB4P3AY	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset
LB4P3BX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LB4P3DX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB4P3IX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LB4P3JX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LD2P3AX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear
LD2P3AY	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Preset
LD2P3BX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LD2P3DX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD2P3IX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LD2P3JX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LD4P3AX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear
LD4P3AY	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Preset
LD4P3BX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LD4P3DX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD4P3IX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)

Table 33: Loadable Counters (Continued)

LD4P3JX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LU2P3AX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU2P3AY	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU2P3BX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LU2P3DX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU2P3IX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU2P3JX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LU4P3AX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU4P3AY	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU4P3BX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LU4P3DX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU4P3IX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU4P3JX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 34: Flip-Flops

FD1P3AX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Clear
FD1P3AY	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Preset
FD1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset
FD1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear

Table 34: Flip-Flops (Continued)

FD1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)
FD1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)
FD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear
FD1S3AY	Positive Edge Triggered D Flip-Flop, GSR Used for Preset
FD1S3BX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset
FD1S3DX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear
FD1S3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear
FD1S3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset
FL1P3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Preset
FL1P3AZ	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Clear
FL1P3BX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable
FL1P3DX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Clear, and Positive Level Enable
FL1P3IY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Clear, and Positive Level Enable (Clear overrides Enable)
FL1P3JY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Preset, and Positive Level Enable (Preset overrides Enable)
FL1S3AX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Clear
FL1S3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Preset

Table 35: Input/Output Buffers

BB	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional
BBPD	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate and Pull-down BiDirectional
BBPU	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate and Pull-up BiDirectional
BBW	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional in keepermode
IB	CMOS Input Buffer
IBM	CMOS Input Buffer
IBMPD	CMOS Input Buffer with Pull-down
IBMPDS	CMOS Input Buffer with Pull-down and Delay
IBMPU	CMOS Input Buffer with Pull-up
IBMPUS	CMOS Input Buffer with Pull-up and Delay
IBMS	CMOS Input Buffer with Delay
IBPD	Input Buffer with Pull-down
IBPU	Input Buffer with Pull-up
ILVDS	LVDS Input Buffer
OB	Output Buffer
OB6	6mA Sink 3mA Source Sinklim Output Buffer
OBCO	Output Complementary Buffer
OBZ	6mA Sink 3mA Source Sinklim Output Buffer
OBZPD	12mA Sink 6mA Source Slewlim Output Buffer
OBZPU	12mA Sink 6mA Source Fast Output Buffer
OBW	6mA Sink 3mA Source Sinklim Output Buffer with Tristate
OLVDS	LVDS Output Buffer

Table 36: Latches

FD1S1A	Positive Level Data Latch with GSR Used for Clear
FD1S1AY	Positive Level Data Latch with GSR Used for Preset
FD1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset
FD1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear

Table 36: Latches (Continued)

FD1S1I	Positive Level Data Latch with Positive Level Synchronous Clear
FD1S1J	Positive Level Data Latch with Positive Level Synchronous Preset
FL1S1A	Positive Level Loadable Latch with Positive Select and GSR Used for Clear
FL1S1AY	Positive Level Loadable Latch with Positive Select and GSR Used for Preset
FL1S1B	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Preset
FL1S1D	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Clear
FL1S1I	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Clear
FL1S1J	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Preset

Table 37: LatticeECP DSP Block

MULT18X18	ECP 18X18 DSP Multiplier
MULT18X18ADDSUB	ECP 18X18 DSP Adder/Subtractor
MULT18X18ADDSUBSUM	ECP 18X18 DSP Adder/Subtractor/Sum
MULT18X18MAC	ECP 18X18 DSP MAC
MULT36X36	ECP 36X36 DSP Multiplier
MULT9X9	ECP 9X9 DSP Multiplier
MULT9X9ADDSUB	ECP 9X9 DSP Adder/Subtractor
MULT9X9ADDSUBSUM	ECP 9X9 DSP Adder/Subtractor/Sum
MULT9X9MAC	ECP 9X9 DSP MAC

Table 38: LatticeXP and LatticeEC Memory Primitives

DP8KA	8K Dual Port Block RAM
DPR16X2B	16 Word by 2 Dual Port RAM (within PFU)
PDP8KA	8K Pseudo Dual Port Block RAM
SP8KA	8K Single Port Block RAM
SPR16X2B	16 Word by 2 Single Port RAM (within PFU)

Table 39: Logic Gates

AND2	2 Input AND Gate
AND3	3 Input AND Gate
AND4	4 Input AND Gate
AND5	5 Input AND Gate
ND2	2 Input NAND Gate
ND3	3 Input NAND Gate
ND4	4 Input NAND Gate
ND5	5 Input NAND Gate
OR2	2 Input OR Gate
OR3	3 Input OR Gate
OR4	4 Input OR Gate
OR5	5 Input OR Gate
NR2	2 Input NOR Gate
NR3	3 Input NOR Gate
NR4	4 Input NOR Gate
NR5	5 Input NOR Gate
XNOR2	2 Input Exclusive NOR Gate
XNOR3	3 Input Exclusive NOR Gate
XNOR4	4 Input Exclusive NOR Gate
XNOR5	5 Input Exclusive NOR Gate
XOR11	11 Input Exclusive OR Gate
XOR2	2 Input Exclusive OR Gate
XOR21	21 Input Exclusive OR Gate
XOR3	3 Input Exclusive OR Gate
XOR4	4 Input Exclusive OR Gate
XOR5	5 Input Exclusive OR Gate

Table 40: Miscellaneous Logic

INV	Inverter

Table 40: Miscellaneous Logic

VHI	Logic High Generator
VLO	Logic Low Generator

Table 41: Multiplexer

L6MUX21	2 to 1 Mux
MUX161	16-Input Mux within the PFU (4 Slices)
MUX21	2 to 1 Mux
MUX321	32-Input Mux within the PFU (8 Slices)
MUX4	4-bit Multiplexer
MUX41	4 to 1 Mux
MUX81	8 to 1 Mux

Table 42: Multipliers (Not DSP)

MULT2	2X2 Multiplier

PIC Cells

Table 43: PIC Flip-Flops (Input)

IFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in input PIC area only)
IFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in input PIC area only)
IFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable, and System Clock (Clear overrides Enable) (used in input PIC area only)
IFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable, and System Clock (Preset overrides Enable) (used in input PIC area only)

Table 44: PIC Flip-Flops (Output)

OFE1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and Edge Clock (used in output PIC area only)
OFE1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and Edge Clock (used in output PIC area only)
OFE1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and Edge Clock (used in output PIC area only)
OFE1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and Edge Clock (used in output PIC area only)
OFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in output PIC area only)
OFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in output PIC area only)
OFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and System Clock (used in output PIC area only)
OFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and System Clock (used in output PIC area only)

Table 45: PIC Latches (Input)

IFS1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset and System Clock (used in input PIC area only)
IFS1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear and System Clock (used in input PIC area only)
IFS1S1I	Positive Level Data Latch with Positive Level Synchronous Clear and System Clock (used in input PIC area only)
IFS1S1J	Positive Level Data Latch with Positive Level Synchronous Preset and System Clock (used in input PIC area only)

Table 46: Read Only memory

ROM16X1	16 Word by 1 Bit Read-Only Memory
ROM32X1	32 Word by 1 Bit Read-Only Memory
ROM64X1	64 Word by 1 Bit Read-Only Memory
ROM128X1	128 Word by 1 Bit Read-Only Memory
ROM256X1	256 Word by 1 Bit Read-Only Memory

Special Cells

Table 47: Clock/PLL

DCS	Dynamic Clock Selection Multiplexer
EPLLB	Enhanced PLL
EHXPLLB	Enhanced High Performance with Dynamic Input Delay Control PLL

Table 48: Combinatorial Primitives

ORCALUT4	4-Input Look Up Table
ORCALUT5	5-Input Look Up Table
ORCALUT6	6-Input Look Up Table
ORCALUT7	7-Input Look Up Table
ORCALUT8	8-Input Look Up Table

Table 49: Dual Data Rate Cells

DQSBUFB	DDR DQS Buffer used as DDR memory DQS generator
DQSDLL	DLL used as DDR memory DQS DLL
IDDRXB	DDRX Input Cell
ODDRXB	Output DDR

Table 50: Miscellaneous

CCU2	Carry Chain
DELAY	Delay
GSR	Global Set/Reset
IBDDC	Dynamic Delay
JTAGB	JTAG (Joint Test Action Group) Controller
PFUMX	2-Input Mux within the PFU, C0 used for Selection with Positive Select
PUR	Power Up Set/Reset
SGSR	Synchronous Release Global Set/Reset Interface
STRTUP	Startup Controller

Primitive Library - LatticeECP2/M

This library includes compatible FPGA primitives supported by the LatticeECP2/M (including the “S-Series” LatticeECP2S and LatticeECP2MS) device families.

References

For further information, a variety of technical notes for the LatticeECP2/M family are available on the Lattice Web site.

- TN1102 - LatticeECP2/M sysIO Usage Guide
- TN1103 - LatticeECP2/M sysCLOCK PLL/DLL Design and Usage Guide
- TN1104 - LatticeECP2/M Memory Usage Guide
- TN1105 - LatticeECP2/M High-Speed I/O Interface
- TN1106 - LatticeECP2/M Power Estimation and Management
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1108 - LatticeECP2/M sysCONFIG Usage Guide
- TN1109 - LatticeECP2/M Configuration Encryption Usage Guide
- TN1113 - LatticeECP2/M Soft Error Detection (SED) Usage Guide
- TN1124 - LatticeECP2M SERDES/PCS Usage Guide

Table 51: Adders/Subtractors

FADD2B	2 Bit Fast Adders/Subtractors
FADSU2	2 Bit Fast Adder/Subtractor (two's complement)
FSUB2B	2 Bit Subtractor

Table 52: Comparators

AGEB2	A Greater Than Or Equal To B (2 bit)
ALEB2	A Less Than Or Equal To B (2 bit)
ANEB2	A Not Equal To B (2 bit)

Table 53: Counters

CB2	Combinational Logic for 2-Bit Bidirectional Counter
CD2	Combinational Logic for 2 Bit Down Counter
CU2	Combinational Logic for 2 Bit Up Counter

Table 54: Loadable Counters

LB2P3AX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB2P3AY	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset
LB2P3BX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LB2P3DX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB2P3IX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LB2P3JX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LD2P3AX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear

Table 54: Loadable Counters (Continued)

LD2P3AY	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Preset
LD2P3BX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LD2P3DX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD2P3IX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LD2P3JX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LU2P3AX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU2P3AY	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU2P3BX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LU2P3DX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU2P3IX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU2P3JX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 55: Flip-Flops

FD1P3AX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Clear
FD1P3AY	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Preset
FD1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset
FD1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear
FD1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)
FD1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)

Table 55: Flip-Flops (Continued)

FD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear
FD1S3AY	Positive Edge Triggered D Flip-Flop, GSR Used for Preset
FD1S3BX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset
FD1S3DX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear
FD1S3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear
FD1S3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset
FL1P3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Preset
FL1P3AZ	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Clear
FL1P3BX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable
FL1P3DX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Clear, and Positive Level Enable
FL1P3IY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Clear, and Positive Level Enable (Clear overrides Enable)
FL1P3JY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Preset, and Positive Level Enable (Preset overrides Enable)
FL1S3AX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Clear
FL1S3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Preset

Table 56: Input/Output Buffer

BB	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional
BBPD	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-down
BBPU	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-up

Table 56: Input/Output Buffer (Continued)

BBW	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional in keepermode
IB	CMOS Input Buffer
IBPD	Input Buffer with Pull-down
IBPU	Input Buffer with Pull-up
ILVDS	LVDS Input Buffer
OB	Output Buffer
OBCO	Output Complementary Buffer
OBW	Output Buffer with Tristate
OBZ	Output Buffer with Tristate
OBZPD	Output Buffer with Tristate and Pull-down
OBZPU	Output Buffer with Tristate and Pull-up
OLVDS	LVDS Output Buffer

Table 57: LatticeECP2/M Memory Primitive

DP16KB	True Dual Port Block RAM
DPR16X4A	Distributed Pseudo Dual Port RAM (within PFU)
PDPW16KB	Pseudo Dual Port Block RAM
SP16KB	Single Port Block RAM
SPR16X4A	Distributed Single Port RAM (within PFU)

Table 58: Logic Gates

AND2	2 Input AND Gate
AND3	3 Input AND Gate
AND4	4 Input AND Gate
AND5	5 Input AND Gate
ND2	2 Input NAND Gate
ND3	3 Input NAND Gate
ND4	4 Input NAND Gate
ND5	5 Input NAND Gate
OR2	2 Input OR Gate

Table 58: Logic Gates (Continued)

OR3	3 Input OR Gate
OR4	4 Input OR Gate
OR5	5 Input OR Gate
NR2	2 Input NOR Gate
NR3	3 Input NOR Gate
NR4	4 Input NOR Gate
NR5	5 Input NOR Gate
XNOR2	2 Input Exclusive NOR Gate
XNOR3	3 Input Exclusive NOR Gate
XNOR4	4 Input Exclusive NOR Gate
XNOR5	5 Input Exclusive NOR Gate
XOR2	2 Input Exclusive OR Gate
XOR3	3 Input Exclusive OR Gate
XOR4	4 Input Exclusive OR Gate
XOR5	5 Input Exclusive OR Gate
XOR11	11 Input Exclusive OR Gate
XOR21	21 Input Exclusive OR Gate

Table 59: Miscellaneous Logic

INV	Inverter
VHI	Logic High Generator
VLO	Logic Low Generator

Table 60: Multiplexers

L6MUX21	LUT-6 2 to 1 Multiplexer
MUX161	16-Input Mux within the PFU (4 Slices)
MUX21	2 to 1 Mux
MUX321	32-Input Mux within the PFU (8 Slices)
MUX41	4 to 1 Mux
MUX81	8 to 1 Mux

Table 61: Multipliers in DSP Blocks

MULT18X18ADDSUBB	18x18 Multiplier Add/Subtract Multipliers in DSP blocks
MULT18X18ADDSUBSUMB	18x18 Multiplier Add/Subtract and SUM Multipliers in DSP blocks
MULT18X18B	18x18 Multiplier in DSP blocks
MULT18X18MACB	18x18 Multiplier Accumulate Multipliers in DSP blocks
MULT36X36B	36x36 Multiplier Multipliers in DSP blocks
MULT9X9ADDSUBB	9x9 Multiplier Add/Subtract Multipliers in DSP blocks
MULT9X9ADDSUBSUMB	9x9 Multiplier Add/Subtract and SUM Multipliers in DSP blocks
MULT9X9B	9x9 Multiplier Multipliers in DSP blocks

PIC Cells

Table 62: PIC Flip-Flops (Input)

IFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in input PIC area only)
IFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in input PIC area only)
IFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable, and System Clock (Clear overrides Enable) (used in input PIC area only)
IFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable, and System Clock (Preset overrides Enable) (used in input PIC area only)

Table 63: PIC Flip-Flops (Output)

OFE1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and Edge Clock (used in output PIC area only)
OFE1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and Edge Clock (used in output PIC area only)
OFE1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and Edge Clock (used in output PIC area only)

Table 63: PIC Flip-Flops (Output) (Continued)

OFE1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and Edge Clock (used in output PIC area only)
OFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in output PIC area only)
OFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in output PIC area only)
OFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and System Clock (used in output PIC area only)
OFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and System Clock (used in output PIC area only)

Table 64: PIC Latches (Input)

IFS1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset and System Clock (used in input PIC area only)
IFS1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear and System Clock (used in input PIC area only)
IFS1S1I	Positive Level Data Latch with Positive Level Synchronous Clear and System Clock (used in input PIC area only)
IFS1S1J	Positive Level Data Latch with Positive Level Synchronous Preset and System Clock (used in input PIC area only)

Table 65: Read-only Memory

ROM128X1	128 Word by 1 bit read-only memory
ROM16X1	16 Word by 1 bit read-only memory
ROM256X1	256 Word by 1 bit read-only memory
ROM32X1	32 Word by 1 bit read-only memory
ROM64X1	64 Word by 1 bit read-only memory

Special Cells

Table 66: Clock Manager/PLL/DLL

CIDDLLA	Clock Injection Delay Removal
CLKDIVB	Clock Divider
DCS	Dynamic Clock Selection Multiplexer
DLLDELA	Slave Delay
EHXPLLD	Complex PLL
EPLLD	Enhanced PLL
OSCD	Oscillator for configuration clock
TRDLLA	Time Reference Delay

Table 67: Combinatorial Primitives

ORCALUT4	4-Input Look Up Table
ORCALUT5	5-Input Look Up Table
ORCALUT6	6-Input Look Up Table
ORCALUT7	7-Input Look Up Table
ORCALUT8	8-Input Look Up Table

Table 68: Dual Data Rate Cells

DQSBUFC	DQS Delay Function and Clock Polarity Selection Logic
DQSDLL	DLL Used as DDR Memory DQS DLL
IDDRFXA	DDR Generic Input with Full Clock Transfer (x1 Gearbox)
IDDRMFX1A	DDR Input and DQS to System Clock Transfer Registers with Full Clock Cycle Transfer
IDDRMX1A	DDR Input and DQS to System Clock Transfer Registers with Half Clock Cycle Transfer
IDDRX2B	DDR Generic Input with 2x Gearing Ratio
IDDRXC	DDR Generic Input
ODDRMXA	DDR Output Registers
ODDRX2B	DDR Generic Output with 2x Gearing Ratio
ODDRXC	DDR Generic Output

Table 69: Miscellaneous

CCU2B	Carry-Chain
DELAYB	Dynamic Delay in PIO
GSR	Global Set/Reset
JTAGC	JTAG (Joint Test Action Group) Controller
MULT2	2X2 Multiplier (not DSP)
PFUMX	2-Input Mux within the PFU, C0 used for Selection with Positive Select
PUR	Power Up Set/Reset
SEDAA	SED BASIC
SGSR	Synchronous Release Global Set/Reset Interface
SPIM	SPIM Primitive
STRTUP	Startup Controller

Primitive Library - LatticeECP3

This library includes compatible FPGA primitives supported by the LatticeECP3 device family.

- Adders/Subtractors
- Comparators
- Counters
- Loadable Counters
- Flip-Flops
- Input/Output Buffers
- LatticeECP3 Memory Primitives
- Logic Gates
- Miscellaneous Logic
- Multiplexer
- Multipliers in DSP Blocks
- PIC Cells
  - PIC Flip-Flops (Input)
  - PIC Flip-Flops (Output)
  - PIC Latches (Input)
- Read-Only Memory
- Special Cells
  - Clock Manager/PLL/DLL
  - Combinatorial Primitives
  - Dual Data Rate Cells
  - Miscellaneous

References

For further information on individual primitives, a variety of technical notes for the LatticeECP3 family are available on the Lattice Web site.

- TN1177 - LatticeECP3 sysIO Usage Guide
- TN1178 - LatticeECP3 sysCLOCK PLL/DLL Design and Usage Guide
- TN1179 - LatticeECP3 Memory Usage Guide
- TN1180 - LatticeECP3 High-Speed I/O Interface
- TN1181 - Power Consumption and Management for LatticeECP3 Devices
- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1169 - LatticeECP3 sysCONFIG Usage Guide
- TN1184 - LatticeECP3 Soft Error Detection (SED) Usage Guide
- TN1176 - LatticeECP3 SERDES/PCS Usage Guide

Table 70: Adders/Subtractors

FADD2B	2 Bit Fast Adders/Subtractors
FADSU2	2 Bit Fast Adder/Subtractor (two's complement)
FSUB2B	2 Bit Subtractor

Table 71: Comparators

AGEB2	A Greater Than Or Equal To B (2 bit)
ALEB2	A Less Than Or Equal To B (2 bit)
ANEB2	A Not Equal To B (2 bit)

Table 72: Counters

CB2	Combinational Logic for 2-Bit Bidirectional Counter
CD2	Combinational Logic for 2 Bit Down Counter
CU2	Combinational Logic for 2 Bit Up Counter

Table 73: Loadable Counters

LB2P3AX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB2P3AY	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset
LB2P3BX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LB2P3DX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB2P3IX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LB2P3JX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LD2P3AX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear

Table 73: Loadable Counters (Continued)

LD2P3AY	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Preset
LD2P3BX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LD2P3DX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD2P3IX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LD2P3JX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LU2P3AX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU2P3AY	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU2P3BX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LU2P3DX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU2P3IX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU2P3JX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 74: Flip-Flops

FD1P3AX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Clear
FD1P3AY	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Preset
FD1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset
FD1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear
FD1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)
FD1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)

Table 74: Flip-Flops (Continued)

FD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear
FD1S3AY	Positive Edge Triggered D Flip-Flop, GSR Used for Preset
FD1S3BX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset
FD1S3DX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear
FD1S3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear
FD1S3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset
FL1P3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Preset
FL1P3AZ	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Clear
FL1P3BX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable
FL1P3DX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Clear, and Positive Level Enable
FL1P3IY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Clear, and Positive Level Enable (Clear overrides Enable)
FL1P3JY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Preset, and Positive Level Enable (Preset overrides Enable)
FL1S3AX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Clear
FL1S3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Preset

Table 75: Input/Output Buffers

BB	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional
BBPD	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-down
BBPU	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-up

Table 75: Input/Output Buffers (Continued)

BBW	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional in keepermode
IB	CMOS Input Buffer
IBPD	Input Buffer with Pull-down
IBPU	Input Buffer with Pull-up
ILVDS	LVDS Input Buffer
OB	Output Buffer
OBCO	Output Complementary Buffer
OBZ	Output Buffer with Tristate
OBZPU	Output Buffer with Tristate and Pull-up
OLVDS	LVDS Output Buffer

Table 76: LatticeECP3 Memory Primitives

DP16KC	True Dual Port Block RAM
DPR16X4C	Distributed Pseudo Dual Port RAM
PDPW16KC	Pseudo Dual Port Block RAM
SP16KC	Single Port Block RAM
SPR16X4C	Distributed Single Port RAM

Table 77: Logic Gates

AND2	2 Input AND Gate
AND3	3 Input AND Gate
AND4	4 Input AND Gate
AND5	5 Input AND Gate
ND2	2 Input NAND Gate
ND3	3 Input NAND Gate
ND4	4 Input NAND Gate
ND5	5 Input NAND Gate
OR2	2 Input OR Gate
OR3	3 Input OR Gate
OR4	4 Input OR Gate

Table 77: Logic Gates (Continued)

OR5	5 Input OR Gate
NR2	2 Input NOR Gate
NR3	3 Input NOR Gate
NR4	4 Input NOR Gate
NR5	5 Input NOR Gate
XNOR2	2 Input Exclusive NOR Gate
XNOR3	3 Input Exclusive NOR Gate
XNOR4	4 Input Exclusive NOR Gate
XNOR5	5 Input Exclusive NOR Gate
XOR2	2 Input Exclusive OR Gate
XOR3	3 Input Exclusive OR Gate
XOR4	4 Input Exclusive OR Gate
XOR5	5 Input Exclusive OR Gate
XOR11	11 Input Exclusive OR Gate
XOR21	21 Input Exclusive OR Gate

Table 78: Miscellaneous Logic

INV	Inverter
VHI	Logic High Generator
VLO	Logic Low Generator

Table 79: Multiplexer

L6MUX21	LUT-6 2 to 1 Multiplexer
MUX161	16-Input Mux within the PFU (4 Slices)
MUX21	2 to 1 Mux
MUX321	32-Input Mux within the PFU (8 Slices)
MUX41	4 to 1 Mux
MUX81	8 to 1 Mux

Table 80: Multipliers in DSP Blocks

MULT18X18C	18x18 Multiplier in DSP blocks
MULT9X9C	9x9 Multiplier Multipliers in DSP blocks

PIC Cells

Table 81: PIC Flip-Flops (Input)

IFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in input PIC area only)
IFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in input PIC area only)
IFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable, and System Clock (Clear overrides Enable) (used in input PIC area only)
IFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable, and System Clock (Preset overrides Enable) (used in input PIC area only)

Table 82: PIC Flip-Flops (Output)

OFD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear. Used to Tri-State DDR/DDR2
OFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in output PIC area only)
OFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in output PIC area only)
OFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and System Clock (used in output PIC area only)
OFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and System Clock (used in output PIC area only)

Table 83: PIC Latches (Input)

IFS1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset and System Clock (used in input PIC area only)
IFS1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear and System Clock (used in input PIC area only)
IFS1S1I	Positive Level Data Latch with Positive Level Synchronous Clear and System Clock (used in input PIC area only)
IFS1S1J	Positive Level Data Latch with Positive Level Synchronous Preset and System Clock (used in input PIC area only)

Table 84: Read-Only Memory

ROM128X1A	128 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs
ROM16X1A	16 Word by 1 Bit Read-Only Memory
ROM256X1A	256 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs
ROM32X1A	32 Word by 1 Bit Read-Only Memory
ROM64X1A	64 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Special Cells

Table 85: Clock Manager/PLL/DLL

CIDDLLB	Clock Injection Delay Removal
CLKDIVB	Clock Divider
DCCA	*(For internal use only)* Dynamic Quadrant Clock Enable/Disable
DCS	Dynamic Clock Selection Multiplexer
DLLDELB	Slave Delay
EHXPLLF	Complex PLL
OSCF	Oscillator for Configuration Clock
TR1DLLB	Time Reference DLL with Dynamic Delay Adjustment
TRDLLB	Time Reference DLL

Table 86: Combinatorial Primitives

LUT4	4-Input Look Up Table
LUT5	5-Input Look Up Table
LUT6	6-Input Look Up Table
LUT7	7-Input Look Up Table
LUT8	8-Input Look Up Table

Table 87: Dual Data Rate Cells

DQSBUFD	DDR DQS Buffer Used for DDR3_MEM and DDR3_MEMGEN
DQSBUFE	DDR DQS Buffer Used for DDR_GENX2
DQSBUFE1	DDR DQS Buffer Used for DDR_GENX2
DQSBUFF	DDR DQS Buffer Used for DDR_MEM, DDR2_MEM, and DDR2_MEMGEN
DQSBUFG	DDR DQS Buffer Used for DDR_GENX1
DQSDLLB	DQS DLL for DDR_MEM, DDR2_MEM, and DDR3_MEM
ECLKSYNCA	ECLK Stop Block
IDDRX2D	Input DDR for DDR3_MEM, DDR_GENX2, and DDR3_MEMGEN
IDDRX2D1	Input DDR for DDR_GENX2
IDDRXD	Input DDR for DDR_MEM, DDR2_MEM, DDR_GENX1, and DDR2_MEMGEN
IDDRXD1	Input DDR for DDR_GENX1
ODDRTDQA	Tri-State for DQ: DDR3_MEM and DDR_GENX2
ODDRTDQSA	Tri-State for Single-Ended and Differential DQS: DDR_MEM, DDR2_MEM, and DDR3_MEM
ODDRX2D	Output DDR for DDR3_MEM and DDR_GENX2
ODDRX2DQSA	Output for Differential DQS: DDR3_MEM
ODDRXD	Output DDR for DDR_MEM, DDR2_MEM, DDR_GENX1, and DDR2_MEMGEN
ODDRXD1	Output DDR for DDR_GENX1
ODDRXDQSA	Output for Single-Ended and Differential DQS: DDR_MEM, DDR2_MEM, and DDR2_MEMGEN

Table 88: Miscellaneous

ALU24A	24 Bit Ternary Adder/Subtractor
ALU54A	54 Bit Ternary Adder/Subtractor
CCU2C	Carry-Chain
DELAYB	Dynamic Delay in PIO
DELAYC	Fixed Delay in PIO
GSR	Global Set/Reset
JTAGE	JTAG (Joint Test Action Group) Controller
MULT2	2X2 Multiplier (not DSP)
PERREGA	Persistent User Register
PFUMX	2-Input Mux within the PFU, C0 used for Selection with Positive Select
PUR	Power Up Set/Reset
SEDCA	Basic SED (Soft Error Detect)
SGSR	Synchronous Release Global Set/Reset Interface
SPIM	SPIM Primitive
START	Startup Controller

Primitive Library - LatticeSC/M

This library includes compatible FPGA primitives supported by the LatticeSC and LatticeSCM device families.

- Adders/Subtractors
- Comparators
- Counters
- Loadable Counters
- Flip-Flops
- Input/Output Buffers
- Latches
- LatticeSC/M Memory Primitives
- Logic Gates
- Miscellaneous Logic
- Multiplexers
- PIC Cells
  - PIC Flip-Flops (Input)
  - PIC Flip-Flops (Output)
  - PIC Flip-Flops (Latched)
  - PIC Latches (Input)
  - PIC Shift Registers
- Read-Only Memory
- Special Cells
  - Clock Manager/PLL/DLL
  - Dual Data Rate Cells
  - Miscellaneous

References

For further information, a variety of technical documents for the LatticeSC/M family are available on the Lattice Web site.

- TN1080 - LatticeSC sysCONFIG Usage Guide
- TN1085 - LatticeSC MPI/System Bus
- TN1088 - LatticeSC PURESPEED I/O Usage Guide
- TN1094 - On-Chip Memory Usage Guide for LatticeSC Devices
- TN1096 - LatticeSC QDR-II SRAM Memory Interface User’s Guide
- TN1098 - LatticeSC sysCLOCK and PLL/DLL User’s Guide
- TN1099 - LatticeSC DDR/DDR2 SDRAM Memory Interface User’s Guide
- TN1101 - Power Calculations and Considerations for LatticeSC Devices

Table 89: Adders/Subtractors

FADD2	2 Bit Fast Adder
FSUB2	2 Bit Fast Subtractor (two's complement)

Table 90: Comparators

AGEB2	A Greater Than Or Equal To B (2 bit)
ALEB2	A Less Than Or Equal To B (2 bit)
ANEB2	A Not Equal To B (2 bit)

Table 91: Counters

CB2	Combinational Logic for 2-Bit Bidirectional Counter
CU2	Combinational Logic for 2 Bit Up Counter
CD2	Combinational Logic for 2 Bit Down Counter

Table 92: Loadable Counters

LB2P3AX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB2P3AY	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset
LB2P3BX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LB2P3DX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB2P3IX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LB2P3JX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LB4P3AX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB4P3AY	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset

Table 92: Loadable Counters (Continued)

LB4P3BX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LB4P3DX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB4P3IX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LB4P3JX	4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LD2P3AX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear
LD2P3AY	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Preset
LD2P3BX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LD2P3DX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD2P3IX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LD2P3JX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LD4P3AX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear
LD4P3AY	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Preset
LD4P3BX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LD4P3DX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD4P3IX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LD4P3JX	4 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LU2P3AX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU2P3AY	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU2P3BX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset

Table 92: Loadable Counters (Continued)

LU2P3DX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU2P3IX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU2P3JX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LU4P3AX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU4P3AY	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU4P3BX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LU4P3DX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU4P3IX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU4P3JX	4 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 93: Flip-Flops

FD1P3AX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Clear
FD1P3AY	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Preset
FD1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset
FD1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear
FD1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)
FD1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)
FD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear
FD1S3AY	Positive Edge Triggered D Flip-Flop, GSR Used for Preset

Table 93: Flip-Flops (Continued)

FD1S3BX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset
FD1S3DX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear
FD1S3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear
FD1S3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset
FL1P3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Preset
FL1P3AZ	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Clear
FL1P3BX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable
FL1P3DX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Clear, and Positive Level Enable
FL1P3IY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Clear, and Positive Level Enable (Clear overrides Enable)
FL1P3JY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Preset, and Positive Level Enable (Preset overrides Enable)
FL1S3AX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Clear
FL1S3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Preset

Table 94: Input/Output Buffers

BB	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional
BBPD	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-down
BBPU	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-up
BBW	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional in keepermode
IB	Input Buffer

Table 94: Input/Output Buffers (Continued)

IBPD	Input Buffer with Pull-down
IBPU	Input Buffer with Pull-up
ILVDS	LVDS Input Buffer
OB	Output Buffer
OBW	Output Buffer with Tristate
OBZ	Output Buffer with Tristate
OBZPD	Output Buffer with Tristate and Pull-down
OBZPU	Output Buffer with Tristate and Pull-up
OLVDS	LVDS Output Buffer

Table 95: Latches

FD1S1A	Positive Level Data Latch with GSR Used for Clear
FD1S1AY	Positive Level Data Latch with GSR Used for Preset
FD1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset
FD1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear
FD1S1I	Positive Level Data Latch with Positive Level Synchronous Clear
FD1S1J	Positive Level Data Latch with Positive Level Synchronous Preset
FL1S1A	Positive Level Loadable Latch with Positive Select and GSR Used for Clear
FL1S1AY	Positive Level Loadable Latch with Positive Select and GSR Used for Preset
FL1S1B	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Preset
FL1S1D	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Clear
FL1S1I	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Clear
FL1S1J	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Preset

Table 96: LatticeSC/M Memory Primitives

DP16KA	16K Dual Port Block RAM
DPR16X2	16 Word by 2 Distributed Dual Port RAM (within PFU)

Table 96: LatticeSC/M Memory Primitives (Continued)

FIFO16KA	16K FIFO
PDP16KA	16K Pseudo Dual Port Block RAM
SP16KA	16 Word by 16 Bit Single Port Block RAM
SPR16X2	16 Word by 2 Distributed Single Port RAM (within PFU)

Table 97: Logic Gates

AND2	2 Input AND Gate
AND3	3 Input AND Gate
AND4	4 Input AND Gate
AND5	5 Input AND Gate
ND2	2 Input NAND Gate
ND3	3 Input NAND Gate
ND4	4 Input NAND Gate
ND5	5 Input NAND Gate
OR2	2 Input OR Gate
OR3	3 Input OR Gate
OR4	4 Input OR Gate
OR5	5 Input OR Gate
NR2	2 Input NOR Gate
NR3	3 Input NOR Gate
NR4	4 Input NOR Gate
NR5	5 Input NOR Gate
XNOR2	2 Input Exclusive NOR Gate
XNOR3	3 Input Exclusive NOR Gate
XNOR4	4 Input Exclusive NOR Gate
XNOR5	5 Input Exclusive NOR Gate
XOR2	2 Input Exclusive OR Gate
XOR3	3 Input Exclusive OR Gate
XOR4	4 Input Exclusive OR Gate
XOR5	5 Input Exclusive OR Gate
XOR11	11 Input Exclusive OR Gate
XOR21	21 Input Exclusive OR Gate

Table 98: Miscellaneous Logic

INV	Inverter
VHI	Logic High Generator
VLO	Logic Low Generator

Table 99: Multiplexers

L6MUX21	LUT-6 2 to 1 Multiplexer
MUX161	16-Input Mux within the PFU (4 Slices)
MUX21	2 to 1 Mux
MUX321	32-Input Mux within the PFU (8 Slices)
MUX41	4 to 1 Mux
MUX81	8 to 1 Mux

PIC Cells

Table 100: PIC Flip-Flops (Input)

IFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in input PIC area only)
IFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in input PIC area only)
IFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable, and System Clock (Clear overrides Enable) (used in input PIC area only)
IFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable, and System Clock (Preset overrides Enable) (used in input PIC area only)

Table 101: PIC Flip-Flops (Output)

OFE1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and Edge Clock (used in output PIC area only)
OFE1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and Edge Clock (used in output PIC area only)
OFE1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and Edge Clock (used in output PIC area only)
OFE1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and Edge Clock (used in output PIC area only)
OFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in output PIC area only)
OFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in output PIC area only)
OFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and System Clock (used in output PIC area only)
OFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and System Clock (used in output PIC area only)

Table 102: PIC Flip-Flops (Latched)

ILF2P3BX	Negative Level Edge Clocked (ECLK) Latch, Feeding Positive Edge Triggered System Clocked (SCLK) Flip-Flop, and Positive Level Asynchronous Preset (used in input PIC area only)
ILF2P3DX	Negative Level Edge Clocked (ECLK) Latch, Feeding Positive Edge Triggered System Clocked (SCLK) Flip-Flop, and Positive Level Asynchronous Clear (used in input PIC area only)
ILF2P3IX	Negative Level Edge Clocked (ECLK) Latch, Feeding Positive Edge Triggered System Clocked (SCLK) Flip-Flop, and Positive Level Synchronous Clear (Clear overrides Enable) (used in input PIC area only)
ILF2P3JX	Negative Level Edge Clocked (ECLK) Latch, Feeding Positive Edge Triggered System Clocked (SCLK) Flip-Flop, and Positive Level Synchronous Preset (Preset overrides Enable) (used in input PIC area only)

Table 103: PIC Latches (Input)

IFS1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset and System Clock (used in input PIC area only)
IFS1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear and System Clock (used in input PIC area only)
IFS1S1I	Positive Level Data Latch with Positive Level Synchronous Clear and System Clock (used in input PIC area only)
IFS1S1J	Positive Level Data Latch with Positive Level Synchronous Preset and System Clock (used in input PIC area only)

Table 104: PIC Shift Registers

ISRX1A	Input 1-Bit Shift Register
ISRX2A	Input 2-Bit Shift Register
ISRX4A	Input 4-Bit Shift Register
OSRX1A	Output 1-Bit Shift Register
OSRX2A	Output 2-Bit Shift Register
OSRX4A	Output 4-Bit Shift Register

Table 105: Read-Only Memory

ROM16X1	16 Word by 1 bit read-only memory
ROM32X1	32 Word by 1 bit read-only memory
ROM32X4	32 Word by 4 bit read-only memory
ROM64X1	64 Word by 1 bit read-only memory
ROM128X1	128 Word by 1 bit read-only memory
ROM256X1	256 Word by 1 bit read-only memory

Special Cells

Table 106: Clock Manager/PLL/DLL

CIDDLLA	Clock Injection Delay Removal
CIMDLLA	Clock Injection Match
CLKCNTL	Clock Controller
CLKDET	Clock Detect
CLKDIV	Clock Divider
DCS	Dynamic Clock Selection Multiplexer
EHXPLLA	Enhanced High Performance with Dynamic Input Delay Control PLL
OSCA	Internal Oscillator
SDCDLLA	Single Delay Cell DLL
TRDLLA	Time Reference Delay

Table 107: Dual Data Rate Cells

IDDRA	Input DDR
IDDRX1A	Input DDR
IDDRX2A	Input DDR
IDDRX4A	Input DDR
ODDRA	Output DDR
ODDRXA	Output DDR
ODDRX2A	Output DDR
ODDRX4A	Output DDR

Table 108: Miscellaneous

DELAY	Delay
GSR	Global Set/Reset
JTAGA	JTAG Logic Control Circuit
PFUMX	2-Input Mux within the PFU, C0 used for Selection with Positive Select
PUR	Power Up Set/Reset
PVTIOCTRL	PVT Monitor Circuit Controller

Table 108: Miscellaneous (Continued)

RDBK	Readback Controller
SGSR	Synchronous Release Global Set/Reset Interface
STRTUP	Startup Controller
TSALL	Global Tristate Interface

Primitive Library - LatticeXP2

This library includes compatible FPGA primitives supported by the LatticeXP2 device family.

- Arithmetic Functions
- Comparators
- Counters
- Loadable Counters
- Flip-Flops
- Input/Output Buffers
- LatticeXP2 Memory Primitives
- Logic Gates
- Miscellaneous Logic
- Multiplexers
- Multipliers in DSP Blocks
- PIC Cells
  - PIC Flip-Flops (Input)
  - PIC Flip-Flops (Output)
  - PIC Latches (Input)
- Read-Only Memory
- Special Cells
  - Clock Manager/PLL/DLL
  - Combinatorial Primitives
  - Dual Data Rate Cells
  - Miscellaneous

References

For further information, a variety of technical notes for the LatticeXP2 family are available on the Lattice web site.

- TN1142 - LatticeXP2 Configuration Encryption and Security Usage Guide
- TN1138 - LatticeXP2 High-Speed I/O Interface
- TN1137 - LatticeXP2 Memory Usage Guide
- TN1130 - LatticeXP2 Soft Error Detection (SED) Usage Guide
- TN1126 - LatticeXP2 sysCLOCK PLL Design and Usage Guide
- TN1141 - LatticeXP2 sysCONFIG Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide
- TN1136 - LatticeXP2 sysIO Usage Guide
- TN1139 - Power Estimation and Management for LatticeXP2 Devices

Table 109: Arithmetic Functions

FADD2B	2 Bit Fast Adders/Subtractors
FADSU2	2 Bit Fast Adder/Subtractor (two's complement)
FSUB2B	2 Bit Subtractor
MULT2	2X2 Multiplier (not DSP)

Table 110: Comparators

AGEB2	A Greater Than Or Equal To B (2 bit)
ALEB2	A Less Than Or Equal To B (2 bit)
ANEB2	A Not Equal To B (2 bit)

Table 111: Counters

CB2	Combinational Logic for 2-Bit Bidirectional Counter
CD2	Combinational Logic for 2 Bit Down Counter
CU2	Combinational Logic for 2 Bit Up Counter

Table 112: Loadable Counters

LB2P3AX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB2P3AY	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset
LB2P3BX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LB2P3DX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB2P3IX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LB2P3JX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LD2P3AX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear

Table 112: Loadable Counters (Continued)

LD2P3AY	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Preset
LD2P3BX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LD2P3DX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD2P3IX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LD2P3JX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LU2P3AX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU2P3AY	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU2P3BX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LU2P3DX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU2P3IX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU2P3JX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 113: Flip-Flops

FD1P3AX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Clear
FD1P3AY	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Preset
FD1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset
FD1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear
FD1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)
FD1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)

Table 113: Flip-Flops (Continued)

FD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear
FD1S3AY	Positive Edge Triggered D Flip-Flop, GSR Used for Preset
FD1S3BX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset
FD1S3DX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear
FD1S3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear
FD1S3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset
FL1P3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Preset
FL1P3AZ	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Clear
FL1P3BX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable
FL1P3DX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Clear, and Positive Level Enable
FL1P3IY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Clear, and Positive Level Enable (Clear overrides Enable)
FL1P3JY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Preset, and Positive Level Enable (Preset overrides Enable)
FL1S3AX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Clear
FL1S3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Preset

Table 114: Input/Output Buffers

BB	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional
BBPD	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-down
BBPU	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-up

Table 114: Input/Output Buffers (Continued)

BBW	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional in keepermode
IB	CMOS Input Buffer
IBPD	Input Buffer with Pull-down
IBPU	Input Buffer with Pull-up
ILVDS	LVDS Input Buffer
OB	Output Buffer
OBCO	Output Complementary Buffer
OBW	Output Buffer with Tristate
OBZ	Output Buffer with Tristate
OBZPD	Output Buffer with Tristate and Pull-down
OBZPU	Output Buffer with Tristate and Pull-up
OLVDS	LVDS Output Buffer

Table 115: LatticeXP2 Memory Primitives

DP16KB	True Dual Port Block RAM
DPR16X4A	Distributed Pseudo Dual Port RAM (within PFU)
DPR16X4B	Distributed Pseudo Dual Port RAM (within PFU)
PDPW16KB	Pseudo Dual Port Block RAM
SP16KB	Single Port Block RAM
SPR16X4A	Distributed Single Port RAM (within PFU)
SPR16X4B	Distributed Single Port RAM (within PFU)
SSPIA	SSPI TAG Memory
STFA	Store to Flash Primitive

Table 116: Logic Gates

AND2	2 Input AND Gate
AND3	3 Input AND Gate
AND4	4 Input AND Gate
AND5	5 Input AND Gate
ND2	2 Input NAND Gate

Table 116: Logic Gates (Continued)

ND3	3 Input NAND Gate
ND4	4 Input NAND Gate
ND5	5 Input NAND Gate
OR2	2 Input OR Gate
OR3	3 Input OR Gate
OR4	4 Input OR Gate
OR5	5 Input OR Gate
NR2	2 Input NOR Gate
NR3	3 Input NOR Gate
NR4	4 Input NOR Gate
NR5	5 Input NOR Gate
XNOR2	2 Input Exclusive NOR Gate
XNOR3	3 Input Exclusive NOR Gate
XNOR4	4 Input Exclusive NOR Gate
XNOR5	5 Input Exclusive NOR Gate
XOR2	2 Input Exclusive OR Gate
XOR3	3 Input Exclusive OR Gate
XOR4	4 Input Exclusive OR Gate
XOR5	5 Input Exclusive OR Gate
XOR11	11 Input Exclusive OR Gate
XOR21	21 Input Exclusive OR Gate

Table 117: Miscellaneous Logic

INV	Inverter
VHI	Logic High Generator
VLO	Logic Low Generator

Table 118: Multiplexers

L6MUX21	LUT-6 2 to 1 Multiplexer
MUX161	16-Input Mux within the PFU (4 Slices)
MUX21	2 to 1 Mux

Table 118: Multiplexers (Continued)

MUX321	32-Input Mux within the PFU (8 Slices)
MUX41	4 to 1 Mux
MUX81	8 to 1 Mux

Table 119: Multipliers in DSP Blocks

MULT18X18ADDSUBB	18x18 Multiplier Add/Subtract Multipliers in DSP blocks
MULT18X18ADDSUBSUMB	18x18 Multiplier Add/Subtract and SUM Multipliers in DSP blocks
MULT18X18B	18x18 Multiplier in DSP blocks
MULT18X18MACB	18x18 Multiplier Accumulate Multipliers in DSP blocks
MULT36X36B	36x36 Multiplier Multipliers in DSP blocks
MULT9X9ADDSUBB	9x9 Multiplier Add/Subtract Multipliers in DSP blocks
MULT9X9ADDSUBSUMB	9x9 Multiplier Add/Subtract and SUM Multipliers in DSP blocks
MULT9X9B	9x9 Multiplier Multipliers in DSP blocks

PIC Cells

Table 120: PIC Flip-Flops (Input)

IFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in input PIC area only)
IFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in input PIC area only)
IFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable, and System Clock (Clear overrides Enable) (used in input PIC area only)
IFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable, and System Clock (Preset overrides Enable) (used in input PIC area only)

Table 121: PIC Flip-Flops (Output)

OFE1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and Edge Clock (used in output PIC area only)
OFE1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and Edge Clock (used in output PIC area only)
OFE1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and Edge Clock (used in output PIC area only)
OFE1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and Edge Clock (used in output PIC area only)
OFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in output PIC area only)
OFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in output PIC area only)
OFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and System Clock (used in output PIC area only)
OFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and System Clock (used in output PIC area only)

Table 122: PIC Latches (Input)

IFS1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset and System Clock (used in input PIC area only)
IFS1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear and System Clock (used in input PIC area only)
IFS1S1I	Positive Level Data Latch with Positive Level Synchronous Clear and System Clock (used in input PIC area only)
IFS1S1J	Positive Level Data Latch with Positive Level Synchronous Preset and System Clock (used in input PIC area only)

Table 123: Read-Only Memory

ROM16X1	16 Word by 1 bit read-only memory
ROM32X1	32 Word by 1 bit read-only memory
ROM64X1	64 Word by 1 bit read-only memory
ROM128X1	128 Word by 1 bit read-only memory
ROM256X1	256 Word by 1 bit read-only memory

Special Cells

Table 124: Clock Manager/PLL/DLL

CLKDIVB	Clock Divider
DCS	Dynamic Clock Selection Multiplexer
EHXPLLE	Complex PLL
EHXPLLE1	Complex PLL
EPLLD	Enhanced PLL
EPLLD1	Enhanced PLL
OSCE	Oscillator for configuration clock

Table 125: Combinatorial Primitives

ORCALUT4	4-Input Look Up Table
ORCALUT5	5-Input Look Up Table
ORCALUT6	6-Input Look Up Table

Table 125: Combinatorial Primitives (Continued)

ORCALUT7	7-Input Look Up Table
ORCALUT8	8-Input Look Up Table

Table 126: Dual Data Rate Cells

DQSBUFC	DQS Delay Function and Clock Polarity Selection Logic
DQSDLL	DLL Used as DDR Memory DQS DLL
IDDRFXA	DDR Generic Input with Full Clock Transfer (x1 Gearbox)
IDDRMFX1A	DDR Input and DQS to System Clock Transfer Registers with Full Clock Cycle Transfer
IDDRMX1A	DDR Input and DQS to System Clock Transfer Registers with Half Clock Cycle Transfer
IDDRX2B	DDR Generic Input with 2x Gearing Ratio
IDDRXC	DDR Generic Input
ODDRMXA	DDR Output Registers
ODDRX2B	DDR Generic Output with 2x Gearing Ratio
ODDRXC	DDR Generic Output

Table 127: Miscellaneous

CCU2B	Carry-Chain
DELAYB	Dynamic Delay in PIO
GSR	Global Set/Reset
IOWAKEUPA	XP2 Wake-up Controller
JTAGE	JTAG (Joint Test Action Group) Controller
PFUMX	2-Input Mux within the PFU, C0 used for Selection with Positive Select
PUR	Power Up Set/Reset
SEDBA	XP2 SED BASIC
SEDBB	XP2 SED BASIC for One Shot Mode
SGSR	Synchronous Release Global Set/Reset Interface
SSPIA	XP2 SSPI TAG Memory
START	Startup Controller
STFA	XP2 Store to Flash Primitive

Primitive Library - MachXO and Platform Manager

This library includes compatible primitives supported by the MachXO and Platform Manager devices.

- Adders/Subtractors
- Comparators
- Counters
- Loadable Counters
- Flip-Flops
- Input/Output Buffers
- Latches
- Logic Gates
- MachXO and Platform Manager Memory Primitives
- Miselleaneous Logic
- Multiplexers
- Multipliers
- Read-Only Memory
- Combinatorial Primitives
- Miscellaneous

References

For further information, a variety of technical notes for the MachXO family are available on the Lattice Web site.

- TN1086 - MachXO JTAG Programming and Configuration User's Guide
- TN1087 - Minimizing System Interruption During Configuration Using TransFR Technology
- TN1089 - MachXO sysCLOCK PLL Design and Usage Guide
- TN1090 - Power Calculations and Considerations for MachXO Devices
- TN1091 - MachXO sysIO Usage Guide
- TN1092 - MachXO Memory Usage Guide
- TN1097 - MachXO Density Migration
- IEEE 1149.1 Boundary Scan Testability in Lattice Devices

Table 128: Adders/Subtractors

FADD2	2 Bit Fast Adder
FSUB2	2 Bit Fast Subtractor (two's complement)
FADSU2	2 Bit Fast Adder/Subtractor (two's complement)

Table 129: Comparators

AGEB2	A Greater Than Or Equal To B (2 bit)
ALEB2	A Less Than Or Equal To B (2 bit)
ANEB2	A Not Equal To B (2 bit)

Table 130: Counters

CB2	Combinational Logic for 2-Bit Bidirectional Counter
CD2	Combinational Logic for 2 Bit Down Counter
CU2	Combinational Logic for 2 Bit Up Counter

Table 131: Loadable Counters

LB2P3AX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB2P3AY	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset
LB2P3BX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LB2P3DX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB2P3IX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LB2P3JX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LD2P3AX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear
LD2P3AY	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Preset
LD2P3BX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LD2P3DX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD2P3IX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)

Table 131: Loadable Counters (Continued)

LD2P3JX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)
LU2P3AX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU2P3AY	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU2P3BX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LU2P3DX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU2P3IX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU2P3JX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 132: Flip-Flops

FD1P3AX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Clear
FD1P3AY	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Preset
FD1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset
FD1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear
FD1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)
FD1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)
FD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear
FD1S3AY	Positive Edge Triggered D Flip-Flop, GSR Used for Preset
FD1S3BX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset
FD1S3DX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear
FD1S3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear

Table 132: Flip-Flops (Continued)

FD1S3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset
FL1P3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Preset
FL1P3AZ	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Clear
FL1P3BX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable
FL1P3DX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Clear, and Positive Level Enable
FL1P3IY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Clear, and Positive Level Enable (Clear overrides Enable)
FL1P3JY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Preset, and Positive Level Enable (Preset overrides Enable)
FL1S3AX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Clear
FL1S3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Preset

Table 133: Input/Output Buffers

BB	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional
BBPD	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-down
BBPU	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-up
BBW	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional in keepermode
IB	Input Buffer
IBPD	Input Buffer with Pull-down
IBPU	Input Buffer with Pull-up
ILVDS	LVDS Input Buffer

Table 133: Input/Output Buffers

OB	Output Buffer
OBW	Output Buffer with Tristate
OBZ	Output Buffer with Tristate
OBZPD	Output Buffer with Tristate and Pull-down
OBZPU	Output Buffer with Tristate and Pull-up
OLVDS	LVDS Output Buffer

Table 134: Latches

FD1S1A	Positive Level Data Latch with GSR Used for Clear
FD1S1AY	Positive Level Data Latch with GSR Used for Preset
FD1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset
FD1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear
FD1S1I	Positive Level Data Latch with Positive Level Synchronous Clear
FD1S1J	Positive Level Data Latch with Positive Level Synchronous Preset
FL1S1A	Positive Level Loadable Latch with Positive Select and GSR Used for Clear
FL1S1AY	Positive Level Loadable Latch with Positive Select and GSR Used for Preset
FL1S1B	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Preset
FL1S1D	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Clear
FL1S1I	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Clear
FL1S1J	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Preset

Table 135: Logic Gates

AND2	2 Input AND Gate
AND3	3 Input AND Gate
AND4	4 Input AND Gate
AND5	5 Input AND Gate
ND2	2 Input NAND Gate

Table 135: Logic Gates (Continued)

ND3	3 Input NAND Gate
ND4	4 Input NAND Gate
ND5	5 Input NAND Gate
OR2	2 Input OR Gate
OR3	3 Input OR Gate
OR4	4 Input OR Gate
OR5	5 Input OR Gate
NR2	2 Input NOR Gate
NR3	3 Input NOR Gate
NR4	4 Input NOR Gate
NR5	5 Input NOR Gate
XNOR2	2 Input Exclusive NOR Gate
XNOR3	3 Input Exclusive NOR Gate
XNOR4	4 Input Exclusive NOR Gate
XNOR5	5 Input Exclusive NOR Gate
XOR2	2 Input Exclusive OR Gate
XOR3	3 Input Exclusive OR Gate
XOR4	4 Input Exclusive OR Gate
XOR5	5 Input Exclusive OR Gate
XOR11	11 Input Exclusive OR Gate
XOR21	21 Input Exclusive OR Gate

Table 136: MachXO and Platform Manager Memory Primitives

DP8KB	8K Dual Port Block RAM
DPR16X2B	16 Word by 2 Dual Port RAM (within PFU)
FIFO8KA	8K FIFO
PDP8KB	8K Pseudo Dual Port Block RAM
SP8KB	8 Word by 8 Bit Single Port Block RAM
SPR16X2B	16 Word by 2 Bit Positive Edge Triggered Write Synchronous Single Port RAM Memory with Positive Write Enable and Positive Write Port Enable (1-Slice)

Table 137: Miselleaneous Logic

INV	Inverter
VHI	Logic High Generator
VLO	Logic Low Generator

Table 138: Multiplexers

L6MUX21	LUT-6 2 to 1 Multiplexer
MUX161	16-Input Mux within the PFU (4 Slices)
MUX21	2 to 1 Mux
MUX321	32-Input Mux within the PFU (8 Slices)
MUX41	4 to 1 Mux
MUX81	8 to 1 Mux

Table 139: Multipliers

MULT2	2x2 Multiplier

Table 140: Read-Only Memory

ROM16X1	16 Word by 1 bit read-only memory
ROM32X1	32 Word by 1 bit read-only memory
ROM64X1	64 Word by 1 bit read-only memory
ROM128X1	128 Word by 1 bit read-only memory
ROM256X1	256 Word by 1 bit read-only memory

Special Cells

Table 141: Combinatorial Primitives

ORCALUT4	4-Input Look Up Table
ORCALUT5	5-Input Look Up Table
ORCALUT6	6-Input Look Up Table
ORCALUT7	7-Input Look Up Table
ORCALUT8	8-Input Look Up Table

Table 142: Miscellaneous

CCU2	Carry Chain
EHXPLLC	Enhanced Extended Performance PLL
GSR	Global Set/Reset
JTAGD	JTAG (Joint Test Action Group) Controller
OSCC	Internal Oscillator
PFUMX	2-Input Mux within the PFU, C0 used for Selection with Positive Select
PUR	Power Up Set/Reset
TSALL	Global Tristate Interface

Primitive Library - MachXO2 and Platform Manager 2

This library includes compatible primitives supported by the MachXO2 device family.

References

For further information, a variety of technical notes for the MachXO2 family are available on the Lattice Web site.

- TN1198 - Power Estimation and Management for MachXO2 Device
- TN1199 - MachXO2 sysCLOCK PLL Design and Usage Guide
- TN1201 - Memory Usage Guide for MachXO2 Devices
- TN1202 - MachXO2 sysIO Usage Guide
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices
- TN1204 - MachXO2 Programming and Configuration Usage Guide

- TN1205 - MachXO2 User Flash Memory and Hardened Control Functions
- TN1206 - MachXO2 Soft Error Detection (SED) Usage Guide

Table 143: Adders/Subtractors

FADD2B	Fast 2 Bit Adder
FSUB2B	2 Bit Subtractor
FADSU2	2 Bit Fast Adder/Subtractor (two's complement)

Table 144: Comparators

AGEB2	A Greater Than Or Equal To B (2 bit)
ALEB2	A Less Than Or Equal To B (2 bit)
ANEB2	A Not Equal To B (2 bit)

Table 145: Counters

CB2	Combinational Logic for 2-Bit Bidirectional Counter
CD2	Combinational Logic for 2 Bit Down Counter
CU2	Combinational Logic for 2 Bit Up Counter

Table 146: Bi-Directional Loadable Counters

LB2P3AX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB2P3AY	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset
LB2P3BX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LB2P3DX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB2P3IX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LB2P3JX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 147: Loadable Down Counters

LD2P3AX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear
LD2P3AY	2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable, GSR Used for Preset
LD2P3BX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LD2P3DX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD2P3IX	2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LD2P3JX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 148: Loadable Up Counters

LU2P3AX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU2P3AY	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU2P3BX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LU2P3DX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU2P3IX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU2P3JX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 149: Flip-Flops

FD1P3AX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Clear
FD1P3AY	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Preset

Table 149: Flip-Flops (Continued)

FD1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset
FD1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear
FD1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)
FD1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)
FD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear
FD1S3AY	Positive Edge Triggered D Flip-Flop, GSR Used for Preset
FD1S3BX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset
FD1S3DX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear
FD1S3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear
FD1S3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset
FL1P3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Preset
FL1P3AZ	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Clear
FL1P3BX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable
FL1P3DX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Clear, and Positive Level Enable
FL1P3IY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Clear, and Positive Level Enable (Clear overrides Enable)
FL1P3JY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Preset, and Positive Level Enable (Preset overrides Enable)
FL1S3AX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Clear
FL1S3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Preset

Table 150: Input/Output Buffer

BB	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional
BBPD	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-down
BBPU	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-up
BBW	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional in keepermode
IB	Input Buffer
IBPD	Input Buffer with Pull-down
IBPU	Input Buffer with Pull-up
ILVDS	LVDS Input Buffer
OB	Output Buffer
OBCO	Output Complementary Buffer
OBZ	Output Buffer with Tristate
OBZPU	Output Buffer with Tristate and Pull-up
OLVDS	LVDS Output Buffer

Table 151: Latches

FD1S1A	Positive Level Data Latch with GSR Used for Clear
FD1S1AY	Positive Level Data Latch with GSR Used for Preset
FD1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset
FD1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear
FD1S1I	Positive Level Data Latch with Positive Level Synchronous Clear
FD1S1J	Positive Level Data Latch with Positive Level Synchronous Preset
FL1S1A	Positive Level Loadable Latch with Positive Select and GSR Used for Clear
FL1S1AY	Positive Level Loadable Latch with Positive Select and GSR Used for Preset
FL1S1B	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Preset

Table 151: Latches (Continued)

FL1S1D	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Clear
FL1S1I	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Clear
FL1S1J	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Preset

Table 152: Logic Gates

AND2	2 Input AND Gate
AND3	3 Input AND Gate
AND4	4 Input AND Gate
AND5	5 Input AND Gate
ND2	2 Input NAND Gate
ND3	3 Input NAND Gate
ND4	4 Input NAND Gate
ND5	5 Input NAND Gate
OR2	2 Input OR Gate
OR3	3 Input OR Gate
OR4	4 Input OR Gate
OR5	5 Input OR Gate
NR2	2 Input NOR Gate
NR3	3 Input NOR Gate
NR4	4 Input NOR Gate
NR5	5 Input NOR Gate
XNOR2	2 Input Exclusive NOR Gate
XNOR3	3 Input Exclusive NOR Gate
XNOR4	4 Input Exclusive NOR Gate
XNOR5	5 Input Exclusive NOR Gate
XOR11	11 Input Exclusive OR Gate
XOR2	2 Input Exclusive OR Gate
XOR21	21 Input Exclusive OR Gate
XOR3	3 Input Exclusive OR Gate

Table 152: Logic Gates (Continued)

XOR4	4 Input Exclusive OR Gate
XOR5	5 Input Exclusive OR Gate

PIC Cells

Table 153: Flip-Flops (Input)

IFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in input PIC area only)
IFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in input PIC area only)
IFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable, and System Clock (Clear overrides Enable) (used in input PIC area only)
IFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable, and System Clock (Preset overrides Enable) (used in input PIC area only)

Table 154: Flip-Flops (Output)

OFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in output PIC area only)
OFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in output PIC area only)
OFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and System Clock (used in output PIC area only)
OFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and System Clock (used in output PIC area only)

Table 155: PIC Latches (Input)

IFS1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset and System Clock (used in input PIC area only)
IFS1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear and System Clock (used in input PIC area only)

Table 155: PIC Latches (Input) (Continued)

IFS1S1I	Positive Level Data Latch with Positive Level Synchronous Clear and System Clock (used in input PIC area only)
IFS1S1J	Positive Level Data Latch with Positive Level Synchronous Preset and System Clock (used in input PIC area only)

Table 156: MachXO2/Platform Manager 2 Memory Primitives

DP8KC	8K True Dual Port Block RAM
DPR16X4C	Distributed Pseudo Dual Port RAM
FIFO8KB	8K FIFO Block RAM
PDPW8KC	Pseudo Dual Port Block RAM
SP8KC	8K Single Port Block RAM
SPR16X4C	Distributed Single Port RAM

Table 157: Miscellaneous Logic

INV	Inverter
VHI	Logic High Generator
VLO	Logic Low Generator

Table 158: Multiplexers

L6MUX21	LUT-6 2 to 1 Multiplexer
MUX161	16-Input Mux within the PFU (4 Slices)
MUX21	2 to 1 Mux
MUX321	32-Input Mux within the PFU (8 Slices)
MUX41	4 to 1 Mux
MUX81	8 to 1 Mux

Table 159: Multipliers

MULT2	2x2 Multiplier

Table 160: Read-Only Memory

ROM128X1A	128 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs
ROM16X1A	16 Word by 1 Bit Read-Only Memory
ROM256X1A	256 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs
ROM32X1A	32 Word by 1 Bit Read-Only Memory
ROM64X1A	64 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Special Cells

Table 161: Clock/PLL/DLL

CLKDIVC	Clock Divider
DCCA	Dynamic Quadrant Clock Enable/Disable
DCMA	Dynamic Clock Mux
DLLDELC	Clock Shifting for ECLK or PCLK
ECLKBRIDGECS	ECLK Bridge Block Clock Select
ECLKSYNCA	ECLK Stop Block
EHXPLLJ	GPLL for MachXO2
OSCH	Oscillator for MachXO2
PLLREFCS	PLL Dynamic Reference Clock Switching

Table 162: Combinatorial Primitives

LUT4	4-Input Look Up Table
LUT5	5-Input Look Up Table
LUT6	6-Input Look Up Table
LUT7	7-Input Look Up Table
LUT8	8-Input Look Up Table

Table 163: Dual Data Rate Cells

DQSBUFH	DQS Circuit for DDR Memory
DQSDLLC	Master DLL for Generating Required Delay
IDDRXE	Input for Generic DDR X1 Using 1:2 Gearing
IDDRX2E	Input for Generic DDR X2 Using 1:4 Gearing
IDDRX4B	Input for Generic DDR X4 Using 1:8 Gearing
IDDRDQSX1A	Input for DDR1/2 Memory
IDDRX71A	7:1 LVDS Input Supporting 1:7 Gearing
ODDRXE	Output for Generic DDR X1 Using 2:1 Gearing
ODDRX2E	Output for Generic DDR X2 Using 4:1 Gearing
ODDRX4B	Output for Generic DDR X4 Using 8:1 Gearing
ODDRDQSX1A	Output for DDR1/2 Memory
ODDRX71A	7:1 LVDS Output
TDDRA	Tristate for DQ/DQS of PIC Cell

Table 164: Miscellaneous

BCINRD	Dynamic Bank Controller InRD
BCLVDSO	Dynamic Bank Controller LVDS
CCU2D	Carry Chain
CLKFBBUFA	Dummy Feedback Delay Between PLL clk Output and PLL fb Port
DELAYD	Dynamic Delay for Bottom Bank
DELAYE	Fixed Delay in PIO
EFB	Embedded Function Block
GSR	Global Set/Reset
INRDB	Input Reference and Differential Buffer
JTAGF	JTAG (Joint Test Action Group) Controller
LVDSOB	LVDS Output Buffer
PCNTR	Power Controller
PFUMX	2-Input Mux within the PFU, C0 used for Selection with Positive Select
PG	Power Guard

Table 164: Miscellaneous (Continued)

PUR	Power Up Set/Reset
SEDFA	Soft Error Detect in Basic Mode
SEDFB	Soft Error Detect in One Shot Mode
SGSR	Synchronous Release Global Set/Reset Interface
START	Startup Controller
TSALL	Global Tristate Interface

Primitive Library - MachXO3L

This library includes compatible primitives supported by the MachXO3L device family.

- Adders/Subtractors
- Comparators
- Counters
  - Bi-Directional Loadable Counters
  - Loadable Down Counters
  - Loadable Up Counters
- Flip-Flops
- Input/Output Buffer
- Latches
- Logic Gates
- PIC Cells
  - Flip-Flops (Input)
  - Flip-Flops (Output)
  - PIC Latches (Input)
- MachXO2 Memory Primitives
- Miscellaneous Logic
- Multiplexers
- Multipliers
- Read-Only Memory
- Special Cells
  - Clock/PLL/DLL
  - Combinatorial Primitives
  - Dual Data Rate Cells
  - Miscellaneous

Table 165: Adders/Subtractors

FADD2B	Fast 2 Bit Adder
FSUB2B	2 Bit Subtractor
FADSU2	2 Bit Fast Adder/Subtractor (two’s complement)

Table 166: Comparators

AGEB2	A Greater Than Or Equal To B (2 bit)
ALEB2	A Less Than Or Equal To B (2 bit)
ANEB2	A Not Equal To B (2 bit)

Table 167: Counters

CB2	Combinational Logic for 2-Bit Bidirectional Counter
CD2	Combinational Logic for 2 Bit Down Counter
CU2	Combinational Logic for 2 Bit Up Counter

Table 168: Bi-Directional Loadable Counters

LB2P3AX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear
LB2P3AY	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset
LB2P3BX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset
	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LB2P3IX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LB2P3JX	2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 169: Loadable Down Counters

LD2P3AX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable, GSR Used for Clear
LD2P3AY	2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable, GSR Used for Preset
LD2P3BX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Preset

Table 169: Loadable Down Counters

LD2P3DX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LD2P3IX	2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LD2P3JX	2 Bit Positive Edge Triggered Loadable Down Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 170: Loadable Up Counters

LU2P3AX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Clear
LU2P3AY	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable, GSR Used for Preset
LU2P3BX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Preset
LU2P3DX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Asynchronous Clear
LU2P3IX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)
LU2P3JX	2 Bit Positive Edge Triggered Loadable Up Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Table 171: Flip-Flops

FD1P3AX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Clear
FD1P3AY	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Preset
FD1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset
FD1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear
FD1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)

Table 171: Flip-Flops (Continued)

FD1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)
FD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear
FD1S3AY	Positive Edge Triggered D Flip-Flop, GSR Used for Preset
FD1S3BX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset
FD1S3DX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear
FD1S3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear
FD1S3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset
FL1P3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Preset
FL1P3AZ	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Clear
FL1P3BX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable
FL1P3DX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Clear, and Positive Level Enable
FL1P3IY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Clear, and Positive Level Enable (Clear overrides Enable)
FL1P3JY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Preset, and Positive Level Enable (Preset overrides Enable)
FL1S3AX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Clear
FL1S3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR used for Preset

Table 172: Input/Output Buffer

BB	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional
BBPD	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-down
BBPU	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional and Pull-up
BBW	CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate BiDirectional in keepermode
IB	Input Buffer
IBPD	Input Buffer with Pull-down
IBPU	Input Buffer with Pull-up
ILVDS	LVDS Input Buffer
OB	Output Buffer
OBCO	Output Complementary Buffer
OBZ	Output Buffer with Tristate
OBZPU	Output Buffer with Tristate and Pull-up
OLVDS	LVDS Output Buffer

Table 173: Latches

FD1S1A	Positive Level Data Latch with GSR Used for Clear
FD1S1AY	Positive Level Data Latch with GSR Used for Preset
FD1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset
FD1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear
FD1S1I	Positive Level Data Latch with Positive Level Synchronous Clear
FD1S1J	Positive Level Data Latch with Positive Level Synchronous Preset
FL1S1A	Positive Level Loadable Latch with Positive Select and GSR Used for Clear
FL1S1AY	Positive Level Loadable Latch with Positive Select and GSR Used for Preset
FL1S1B	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Preset

Table 173: Latches (Continued)

FL1S1D	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Clear
FL1S1I	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Clear
FL1S1J	Positive Level Loadable Latch with Positive Select and Positive Level Synchronous Preset

Table 174: Logic Gates

AND2	2 Input AND Gate
AND3	3 Input AND Gate
AND4	4 Input AND Gate
AND5	5 Input AND Gate
ND2	2 Input NAND Gate
ND3	3 Input NAND Gate
ND4	4 Input NAND Gate
ND5	5 Input NAND Gate
OR2	2 Input OR Gate
OR3	3 Input OR Gate
OR4	4 Input OR Gate
OR5	5 Input OR Gate
NR2	2 Input NOR Gate
NR3	3 Input NOR Gate
NR4	4 Input NOR Gate
NR5	5 Input NOR Gate
XNOR2	2 Input Exclusive NOR Gate
XNOR3	3 Input Exclusive NOR Gate
XNOR4	4 Input Exclusive NOR Gate
XNOR5	5 Input Exclusive NOR Gate
XOR11	11 Input Exclusive OR Gate
XOR2	2 Input Exclusive OR Gate
XOR21	21 Input Exclusive OR Gate
XOR3	3 Input Exclusive OR Gate

Table 174: Logic Gates (Continued)

XOR4	4 Input Exclusive OR Gate
XOR5	5 Input Exclusive OR Gate

PIC Cells

Table 175: Flip-Flops (Input)

IFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in input PIC area only)
IFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in input PIC area only)
IFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable, and System Clock (Clear overrides Enable) (used in input PIC area only)
IFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable, and System Clock (Preset overrides Enable) (used in input PIC area only)

Table 176: Flip-Flops (Output)

OFS1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in output PIC area only)
OFS1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in output PIC area only)
OFS1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and System Clock (used in output PIC area only)
OFS1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and System Clock (used in output PIC area only)

Table 177: PIC Latches (Input)

IFS1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset and System Clock (used in input PIC area only)
IFS1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear and System Clock (used in input PIC area only)

Table 177: PIC Latches (Input) (Continued)

IFS1S1I	Positive Level Data Latch with Positive Level Synchronous Clear and System Clock (used in input PIC area only)
IFS1S1J	Positive Level Data Latch with Positive Level Synchronous Preset and System Clock (used in input PIC area only)

Table 178: MachXO2 Memory Primitives

DP8KC	8K True Dual Port Block RAM
DPR16X4C	Distributed Pseudo Dual Port RAM
FIFO8KB	8K FIFO Block RAM
PDPW8KC	Pseudo Dual Port Block RAM
SP8KC	8K Single Port Block RAM
SPR16X4C	Distributed Single Port RAM

Table 179: Miscellaneous Logic

INV	Inverter
VHI	Logic High Generator
VLO	Logic Low Generator

Table 180: Multiplexers

L6MUX21	LUT-6 2 to 1 Multiplexer
MUX161	16-Input Mux within the PFU (4 Slices)
MUX21	2 to 1 Mux
MUX321	32-Input Mux within the PFU (8 Slices)
MUX41	4 to 1 Mux
MUX81	8 to 1 Mux

Table 181: Multipliers

MULT2	2x2 Multiplier

Table 182: Read-Only Memory

ROM128X1A	128 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs
ROM16X1A	16 Word by 1 Bit Read-Only Memory
ROM256X1A	256 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs
ROM32X1A	32 Word by 1 Bit Read-Only Memory
ROM64X1A	64 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Special Cells

Table 183: Clock/PLL/DLL

CLKDIVC	Clock Divider
DCCA	Dynamic Quadrant Clock Enable/Disable
DCMA	Dynamic Clock Mux
DLLDELC	Clock Shifting for ECLK or PCLK
ECLKBRIDGECS	ECLK Bridge Block Clock Select
ECLKSYNCA	ECLK Stop Block
EHXPLLJ	GPLL for MachXO2
OSCH	Oscillator for MachXO2
PLLREFCS	PLL Dynamic Reference Clock Switching

Table 184: Combinatorial Primitives

LUT4	4-Input Look Up Table
LUT5	5-Input Look Up Table
LUT6	6-Input Look Up Table
LUT7	7-Input Look Up Table
LUT8	8-Input Look Up Table

Table 185: Dual Data Rate Cells

DQSDLLC	Master DLL for Generating Required Delay
IDDRXE	Input for Generic DDR X1 Using 1:2 Gearing
IDDRX2E	Input for Generic DDR X2 Using 1:4 Gearing
IDDRX71A	7:1 LVDS Input Supporting 1:7 Gearing
ODDRXE	Output for Generic DDR X1 Using 2:1 Gearing
ODDRX2E	Output for Generic DDR X2 Using 4:1 Gearing
ODDRX71A	7:1 LVDS Output

Table 186: Miscellaneous

BCINRD	Dynamic Bank Controller InRD
BCLVDSO	Dynamic Bank Controller LVDS
CCU2D	Carry Chain
CLKFBBUFA	Dummy Feedback Delay Between PLL clk Output and PLL fb Port
DELAYD	Dynamic Delay for Bottom Bank
DELAYE	Fixed Delay in PIO
EFB	Embedded Function Block
GSR	Global Set/Reset
INRDB	Input Reference and Differential Buffer
JTAGF	JTAG (Joint Test Action Group) Controller
LVDSOB	LVDS Output Buffer
PCNTR	Power Controller
PFUMX	2-Input Mux within the PFU, C0 used for Selection with Positive Select
PG	Power Guard
PUR	Power Up Set/Reset
SEDFA	Soft Error Detect in Basic Mode
SEDFB	Soft Error Detect in One Shot Mode
SGSR	Synchronous Release Global Set/Reset Interface
START	Startup Controller
TSALL	Global Tristate Interface

Alphanumeric Primitives List

This section lists all the Lattice library primitives in alphanumeric order.

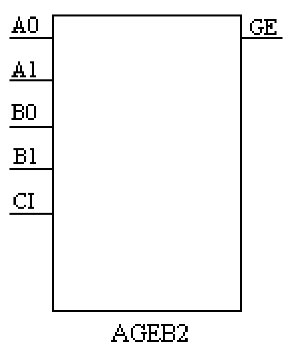
A

AGEB2

"A" Greater Than Or Equal To "B"

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A0, A1, B0, B1, CI

OUTPUT: GE

Description

AGEB2 is a 2-bit comparator that can be cascaded together to build larger comparators. It has two 2-bit inputs and a carry-in input. The carry-in (CI) on the first stage should be tied HIGH. The compare-out (GE) output is HIGH if A[1:0] >= B[1:0] and LOW if A[1:0] < B[1:0]. To build larger comparators, tie the GE on the lower stage to CI on the upper stage.

Note

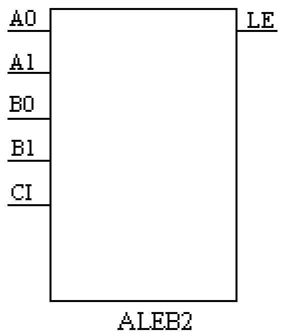
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

ALEB2

"A" Less Than Or Equal To "B"

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A0, A1, B0, B1, CI

OUTPUT: LE

Description

ALEB2 is a 2-bit comparator that can be cascaded together to build larger comparators. It has two 2-bit inputs and a carry-in input. The carry-in (CI) on the first stage should be tied HIGH. The compare-out (LE) output is HIGH if A[1:0] <= B[1:0] and LOW if A[1:0] > B[1:0]. To build larger comparators, tie the LE on the lower stage to CI on the upper stage.

Note

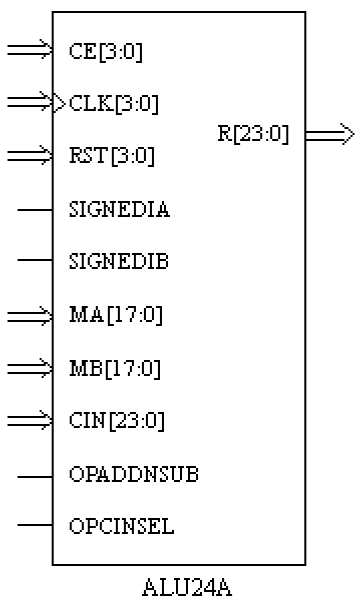
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

ALU24A

24 Bit Ternary Adder/Subtractor

Architectures Supported:

- LatticeECP3



INPUTS: MA17, MA16, MA15, MA14, MA13, MA12, MA11, MA10, MA9, MA8, MA7, MA6, MA5, MA4, MA3, MA2, MA1, MA0, MB17, MB16, MB15, MB14, MB13, MB12, MB11, MB10, MB9, MB8, MB7, MB6, MB5, MB4, MB3, MB2, MB1, MB0, CIN23, CIN22, CIN21, CIN20, CIN19, CIN18, CIN17, CIN16, CIN15, CIN14, CIN13, CIN12, CIN11, CIN10, CIN9, CIN8, CIN7, CIN6, CIN5, CIN4, CIN3, CIN2, CIN1, CIN0, CE3, CE2, CE1, CE0, CLK3, CLK2, CLK1, CLK0, RST3, RST2, RST1, RST0, SIGNEDIA, SIGNEDIB, OPADDNSUB, OPCINSEL

OUTPUTS: R23, R22, R21, R20, R19, R18, R17, R16, R15, R14, R13, R12, R11, R10, R9, R8, R7, R6, R5, R4, R3, R2, R1, R0

ATTRIBUTES:

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OPCODE_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OPCODE_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OPCODE_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OPCODE_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OPCODE_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OPCODE_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

RESETMODE: "SYNC" (default), "ASYNC"

ALU24A Attribute Description

Name	Description
REG_OUTPUT_CLK	ALU register for output clock selection
REG_OUTPUT_CLK	ALU register for output clock enable selection
REG_OUTPUT_RST	ALU register for output reset selection
REG_OPCODE_0_CLK	OPCODE register clock selection
REG_OPCODE_0_CE	OPCODE register clock enable selection
REG_OPCODE_0_RST	OPCODE register reset selection
REG_OPCODE_1_CLK	OPCODE pipeline register clock selection
REG_OPCODE_1_CE	OPCODE pipeline register clock enable selection
REG_OPCODE_1_RST	OPCODE pipeline register reset selection
GSR	Global set reset selection
RESETMODE	Reset mode selection

ALU24A Port Description

Input/Output	Port Name	Capture Name	Type	Size (Buses Only)	Description
I	CLK0	CLK0	Bit	N/A	Clock Input
I	CLK1	CLK1	Bit	N/A	Clock Input
I	CLK2	CLK2	Bit	N/A	Clock Input
I	CLK3	CLK3	Bit	N/A	Clock Input
I	CE0	CE0	Bit	N/A	Clock Enable Input
I	CE1	CE1	Bit	N/A	Clock Enable Input
I	CE2	CE2	Bit	N/A	Clock Enable Input
I	CE3	CE3	Bit	N/A	Clock Enable Input
I	RST0	RST0	Bit	N/A	Reset Input
I	RST1	RST1	Bit	N/A	Reset Input
I	RST2	RST2	Bit	N/A	Reset Input
I	RST3	RST3	Bit	N/A	Reset Input
I	SIGNEDIA[1]	SIGNEDIA	Bit	N/A	Input A Sign Selection
I	SIGNEDIB[1]	SIGNEDIB	Bit	N/A	Input A Sign Selection
I	MA[1]	MA[35:18]	Bus	17:0	Input A from Multiplier
I	MB[1]	MB[35:18]	Bus	17:0	Input B from Multiplier
I	CIN	CIN[50:27]	Bus	23:0	CIN Input
I	OPCINSEL	OP5	Bit	N/A	CIN Select selects CIN (010) or GND (000)
I	OPADDNSUB	OP7	Bit	N/A	Add/Subtract selection
O	R	R[43:34]	Bus	23:0	Output

Notes:
1. A and B refer to the first and second multiplier of the slice and not the Ax and Bx inputs to multiplier x.

Note

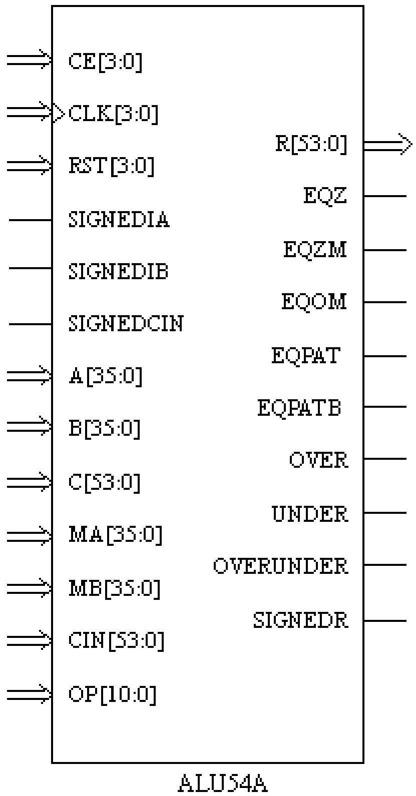
The synthesis tool will not use this block to inference DSP function.

ALU54A

54 Bit Ternary Adder/Subtractor

Architectures Supported:

- LatticeECP3



INPUTS: A35, A34, A33, A32, A31, A30, A29, A28, A27, A26, A25, A24, A23, A22, A21, A20, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0, B35, B34, B33, B32, B31, B30, B29, B28, B27, B26, B25, B24, B23, B22, B21, B20, B19, B18, B17, B16, B15, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1, B0, C53, C52, C51, C50, C49, C48, C47, C46, C45, C44, C43, C42, C41, C40, C39, C38, C37, C36, C35, C34, C33, C32, C31, C30, C29, C28, C27, C26, C25, C24, C23, C22, C21, C20, C19, C18, C17, C16, C15, C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1, C0, MA35, MA34, MA33, MA32, MA31, MA30, MA29, MA28, MA27, MA26, MA25, MA24, MA23, MA22, MA21, MA20, MA19, MA18, MA17, MA16, MA15, MA14, MA13, MA12, MA11, MA10, MA9, MA8, MA7, MA6, MA5, MA4, MA3, MA2, MA1, MA0, MB35, MB34, MB33, MB32, MB31, MB30, MB29, MB28, MB27, MB26, MB25, MB24, MB23, MB22, MB21, MB20, MB19, MB18, MB17, MB16, MB15, MB14, MB13, MB12, MB11, MB10, MB9, MB8, MB7, MB6, MB5, MB4, MB3, MB2, MB1, MB0, CIN53, CIN52, CIN51, CIN50, CIN49, CIN48, CIN47, CIN46, CIN45,

CIN44, CIN43, CIN42, CIN41, CIN40, CIN39, CIN38, CIN37, CIN36, CIN35, CIN34, CIN33, CIN32, CIN31, CIN30, CIN29, CIN28, CIN27, CIN26, CIN25, CIN24, CIN23, CIN22, CIN21, CIN20, CIN19, CIN18, CIN17, CIN16, CIN15, CIN14, CIN13, CIN12, CIN11, CIN10, CIN9, CIN8, CIN7, CIN6, CIN5, CIN4, CIN3, CIN2, CIN1, CIN0, CE3, CE2, CE1, CE0, CLK3, CLK2, CLK1, CLK0, RST3, RST2, RST1, RST0, SIGNEDIA, SIGNEDIB, SIGNEDCIN, OP10, OP9, OP8, OP7, OP6, OP5, OP4, OP3, OP2, OP1, OP0

OUTPUTS: R53, R52, R51, R50, R49, R48, R47, R46, R45, R44, R43, R42, R41, R40, R39, R38, R37, R36, R35, R34, R33, R32, R31, R30, R29, R28, R27, R26, R25, R24, R23, R22, R21, R20, R19, R18, R17, R16, R15, R14, R13, R12, R11, R10, R9, R8, R7, R6, R5, R4, R3, R2, R1, R0, EQZ, EQZM, EQOM, EQPAT, EQPATB, OVER, UNDER, OVERUNDER, SIGNEDR

ATTRIBUTES:

REG_INPUTC0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTC0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTC0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTC1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTC1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTC1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OPCODEOP0_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OPCODEOP0_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OPCODEOP0_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OPCODEOP1_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OPCODEOP0_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OPCODEOP0_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OPCODEOP0_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OPCODEOP1_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OPCODEIN_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OPCODEIN_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OPCODEIN_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OPCODEIN_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OPCODEIN_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OPCODEIN_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_FLAG_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_FLAG_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_FLAG_RST: "RST0" (default), "RST1", "RST2", "RST3"

MCPAT_SOURCE: "STATIC" (default), "DYNAMIC"

MASKPAT_SOURCE: "STATIC" (default), "DYNAMIC"

MASK01: any 14-bit hexadecimal value (default: all zeros)

MCPAT: any 14-bit hexadecimal value (default: all zeros)

MASKPAT: any 14-bit hexadecimal value (default: all zeros)

RNDPAT: any 14-bit hexadecimal value (default: all zeros)

GSR: "ENABLED" (default), "DISABLED"

RESETMODE: "SYNC" (default), "ASYNC"

MULT9_MODE: "DISABLED" (default), "ENABLED"

FORCE_ZERO_BARREL_SHIFT: "DISABLED" (default), "ENABLED"

LEGACY: "DISABLED" (default), "ENABLED"

ALU54A Attribute Description

Name	Description
REG_INPUTC0_CLK	Input C register selection for C[26:0]
REG_INPUTC0_CE	Input C clock enable selection for C[26:0]
REG_INPUTC0_RST	Input C reset selection for C[26:0]
REG_INPUTC1_CLK	Input C register selection for C[53:27]
REG_INPUTC1_CE	Input C clock enable selection for C[53:27]
REG_INPUTC1_RST	Input C reset selection for C[53:27]
REG_OPCODEOP0_0_CLK	OPCODE register clock selection for oper [0]
REG_OPCODEOP0_0_CE	OPCODE register clock enable selection for oper [3:0]
REG_OPCODEOP0_0_RST	OPCODE register reset selection for oper [3:0]
REG_OPCODEOP1_0_CLK	OPCODE register clock selection for oper [3:1]
REG_OPCODEOP0_1_CLK	OPCODE pipeline register clock selection for oper [3:1]
REG_OPCODEOP0_1_CE	OPCODE pipeline register clock enable selection for oper [3:0]
REG_OPCODEOP0_1_RST	OPCODE pipeline register reset selection for oper [3:0]
REG_OPCODEOP1_1_CLK	OPCODE pipeline register clock selection for oper [3:1]
REG_OPCODEIN_0_CLK	OPCODE input register clock for InA[1:0], InB[1:0], InC[2:0]
REG_OPCODEIN_0_CE	OPCODE input register clock enable for InA[1:0], InB[1:0], InC[2:0]
REG_OPCODEIN_0_RST	OPCODE input register reset for InA[1:0], InB[1:0], InC[2:0]
REG_OPCODEIN_1_CLK	OPCODE input pipeline register clock for InA[1:0], InB[1:0], InC[2:0]
REG_OPCODEIN_1_CE	OPCODE input pipeline register clock enable for InA[1:0], InB[1:0], InC[2:0]
REG_OPCODEIN_1_RST	OPCODE input pipeline register reset for InA[1:0], InB[1:0], InC[2:0]
REG_OUTPUT0_CLK	ALU register for LSB output 17:0 clock selection
REG_OUTPUT0_CE	ALU register for LSB output 17:0 clock enable selection
REG_OUTPUT0_RST	ALU register for LSB output 17:0 reset selection
REG_OUTPUT1_CLK	ALU register for MSB output 53:18 clock selection

Name	Description
REG_OUTPUT1_CE	ALU register for MSB output 53:18 clock enable selection
REG_OUTPUT1_RST	ALU register for MSB output 53:18 reset selection
REG_FLAG_CLK	Flag register clock selection
REG_FLAG_CE	Flag register clock enable selection
REG_FLAG_RST	Flag pipeline register reset selection
MASKPAT_SOURCE[1]	EQPAT/EQPATB source setting
MCPAT_SOURCE[1]	MEM Cell Pattern source setting
MASK01	Mask for EQZM/EQOM
MASKPAT	Mask for EQPAT/EQPATB
MCPAT	MEM Cell Pattern
RNDPAT	Rounding Pattern
GSR	Global set reset selection
RESETMODE	Reset mode selection
MULT9_MODE	Operation in Mult9 mode
FORCE_ZERO_BARREL_SHIFT	Forces zeros to 18 MSB of shift for barrel shift
LEGACY	Required to support LatticeECP2 backward compatibility

Notes:
1. MASKPAT_SOURCE and MCPAT_SOURCE cannot be DYNAMIC at the same time since both use C[53:0]. There should be a DRC in the software to check this.

ALU54A Port Description

Input/Output	Port Name	Capture Name	Type	Size (Buses Only)	Description
I	CLK0	CLK0	Bit	N/A	Clock Input
I	CLK1	CLK1	Bit	N/A	Clock Input
I	CLK2	CLK2	Bit	N/A	Clock Input
I	CLK3	CLK3	Bit	N/A	Clock Input
I	CE0	CE0	Bit	N/A	Clock Enable Input
I	CE1	CE1	Bit	N/A	Clock Enable Input
I	CE2	CE2	Bit	N/A	Clock Enable Input
I	CE3	CE3	Bit	N/A	Clock Enable Input

Input/Output	Port Name	Capture Name	Type	Size (Buses Only)	Description
I	RST0	RST0	Bit	N/A	Reset Input
I	RST1	RST1	Bit	N/A	Reset Input
I	RST2	RST2	Bit	N/A	Reset Input
I	RST3	RST3	Bit	N/A	Reset Input
I	SIGNEDIA[1]	SIGNEDIA	Bit	N/A	Input A Sign Selection
I	SIGNEDIB[1]	SIGNEDIB	Bit	N/A	Input A Sign Selection
I	A[1]	A	Bus	35:0	Input A from Multiplier
I	B[1]	B	Bus	35:0	Input B from Multiplier
I	C	C	Bus	53:0	C Input
I	MA[1]	MA	Bus	35:0	Input A from Multiplier
I	MB[1]	MB	Bus	35:0	Input B from Multiplier
I	CIN	CIN	Bus	53:0	CIN Input
I	OP	OP	Bus	10:0	Opcode for ALU Operation Selection
I	SIGNEDCIN	SIGNEDCIN	Bit	N/A	CIN Right Shift, Signed or Unsigned Control
O	R	R	Bus	53:0	Output
O	EQZ	F[7]	Bit	N/A	Equal to Zero
O	EQZM	F[6]	Bit	N/A	Equal to Zero with Mask
O	EQPOM	F[5]	Bit	N/A	Equal to One with Mask
O	EQPAT	F[4]	Bit	N/A	Equal to Pat with Mask
O	EQPATB	F[3]	Bit	N/A	Equal to Bit Inverted Pat with Mask
O	OVER	F[2]	Bit	N/A	Accumulator Overflow
O	UNDER	F[1]	Bit	N/A	Accumulator Underflow
O	OVERUNDER	F[0]	Bit	N/A	Either Over or Under Flow
O	SIGNEDR	SIGNEDR	Bit	N/A	Signed or Unsigned Output of ALU

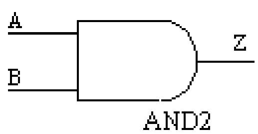
Notes:
1. A and B refer to the first and second multiplier of the slice and not the Ax and Bx inputs to multiplier x.

AND2

2 Input AND Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B

OUTPUT: Z

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

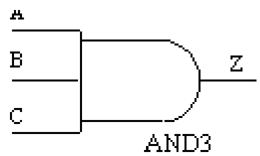
AND3

3 Input AND Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP

- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C

OUTPUT: Z

Note

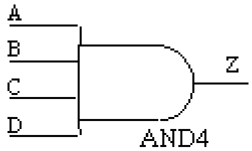
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

AND4

4 Input AND Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D

OUTPUT: Z

Note

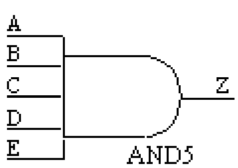
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

AND5

5 Input AND Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D, E

OUTPUT: Z

Note

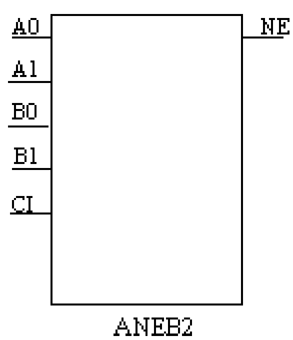
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

ANEB2

"A" Not Equal To "B"

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A0, A1, B0, B1, CI

OUTPUT: NE

Description

ANEB2 is a 2-bit comparator that can be cascaded together to build larger comparators. It has two 2-bit inputs and a carry-in input. The carry-in (CI) on the first stage should be tied LOW. The compare-out (NE) output is LOW if A[1:0] = B[1:0] and HIGH otherwise. To build larger comparators, tie the NE on the lower stage to CI on the upper stage.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

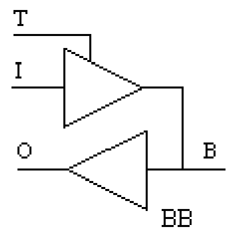
B

BB

CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate – BiDirectional

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: I, T

OUTPUT: O

IOPUT: B

Truth Table

INPUTS		OUTPUTS	BIDIRECTIONAL
I	T	O	B
X	1	U	Z
X	1	1	1
X	1	0	0
0	0	0	0
1	0	1	1

X = Don’t care

U = Unknown

When TSALL=0, O=U, B=Z

For PU/PD buffers, when TSALL=0, O and B will be pulled up or pulled down, respectively.

Note

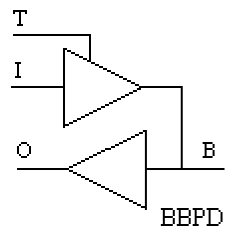
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

BBPD

CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate and Pull-down – BiDirectional

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: I, T

OUTPUT: O

IOPUT: B

Truth Table

INPUTS		OUTPUTS	BIDIRECTIONAL
I	T	O	B
X	1	U	Pull0
X	1	1	1
X	1	0	0
0	0	0	0
1	0	1	1

X = Don’t care

U = Unknown

When TSALL=0, O=U, B=Z

For PU/PD buffers, when TSALL=0, O and B will be pulled up or pulled down, respectively.

Note

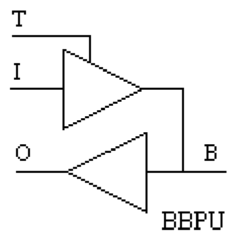
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

BBPU

CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate and Pull-up – BiDirectional

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: I, T

OUTPUT: O

IOPUT: B

Truth Table

INPUTS		OUTPUTS	BIDIRECTIONAL
I	T	O	B
X	1	U	Pull1
X	1	1	1
X	1	0	0
0	0	0	0
1	0	1	1

X = Don’t care

U = Unknown

When TSALL=0, O=U, B=Z

For PU/PD buffers, when TSALL=0, O and B will be pulled up or pulled down, respectively.

Note

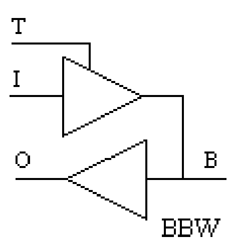
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

BBW

CMOS Input 6mA Sink 3mA Source Sinklim Output Buffer with Tristate – BiDirectional in keepermode

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: I, T

OUTPUT: O

IOPUT: B

Truth Table

INPUTS		OUTPUTS	BIDIRECTIONAL
I	T	O	B
0	1	0	weak 0
1	1	1	weak 1

X = Don't care

U = Unknown

Note

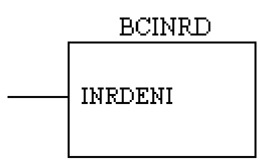
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

BCINRD

Dynamic Bank Controller InRD

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUT: INRDENI

ATTRIBUTES:

BANKID: 0 (default), 1, 2, 3, 4, 5

Description

The MachXO2/Platform Manager 2 dynamic bank controller is used to power down banks InRD (Input Referenced and Differential) and LVDS Outputs. The dynamic bank controller is represented with two primitives: BCINRD and BCLVDSO. The INRDENI input is the dynamic signal to enable and disable bank InRD.

For more information, refer to the following technical note on the Lattice web site:

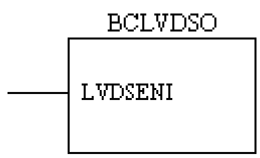
- TN1198 - Power Estimation and Management for MachXO2 Devices

BCLVDSO

Dynamic Bank Controller LVDS

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUT: LVDSENI

Description

The MachXO2/Platform Manager 2 dynamic bank controller is used to power down banks InRD (Input Referenced and Differential) and LVDS Outputs. The dynamic bank controller is represented with two primitives: BCLVDSO and BCINRD. The LVDSENI input is the dynamic signal to enable and disable bank InRD.

For more information, refer to the following technical note on the Lattice web site:

- TN1198 - Power Estimation and Management for MachXO2 Devices

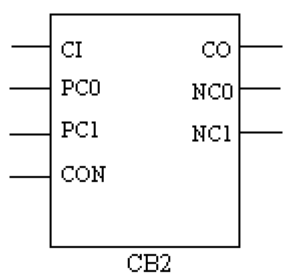
C

CB2

Combinational Logic for 2-Bit Bidirectional Counter

Architectures Supported:

- LatticeECP3
- LatticeXP2
- LatticeSC/M
- LatticeECP2/M
- LatticeECP/EC
- LatticeXP
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: CI, PC0, PC1, CON

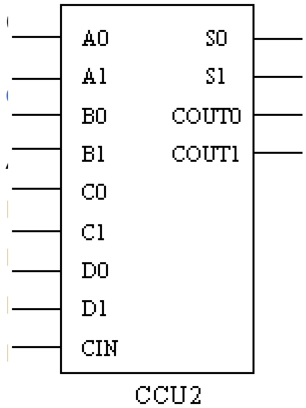
OUTPUTS: CO, NC0, NC1

Description

This primitive realizes the combinational logic needed to implement a 2-bit bidirectional counter by using ripple elements.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.



INPUT: CIN, A0, B0, C0, D0, A1, B1, C1, D1

OUTPUT: S0, S1, COUT0, COUT1

ATTRIBUTES:

INIT0: hexadecimal value (default: 16’h0000)

INIT1: hexadecimal value (default: 16’h0000)

INJECT1_0: "YES" (default), "NO"

INJECT1_1: "YES" (default), "NO"

Note

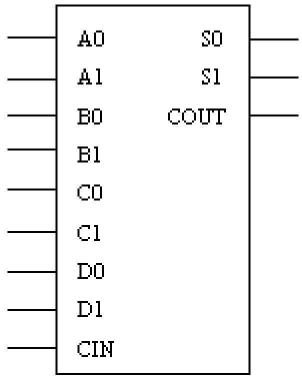
The attributes need to be defined when CCU2 is instantiated.

CCU2B

Carry-Chain

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



CCU2B

INPUTS: A0, B0, C0, D0, A1, B1, C1, D1, CIN

OUTPUTS: S0, S1, COUT

ATTRIBUTES:

INIT0: hexadecimal value (default: 16’h0000)

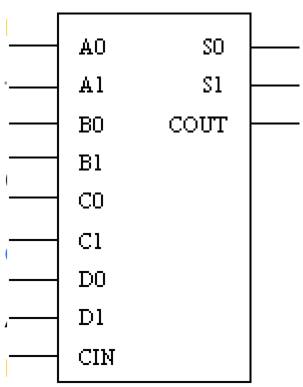
INIT1: hexadecimal value (default: 16’h0000)

INJECT1_0: "YES" (default), "NO"

INJECT1_1: "YES" (default), "NO"

Note

- The attributes need to be defined when CCU2B is instantiated.



ble as a schematic symbol. You can add it to your
Symbol command in the Schematic Editor.

CCU2C

INPUTS: CIN, A1, B1, C1, D1, A0, B0, C0, D0

OUTPUTS: S1, S0, COUT

ATTRIBUTES:

INIT0: hexadecimal value (default: 16’h0000)

INIT1: hexadecimal value (default: 16’h0000)

INJECT1_0: "YES" (default), "NO"

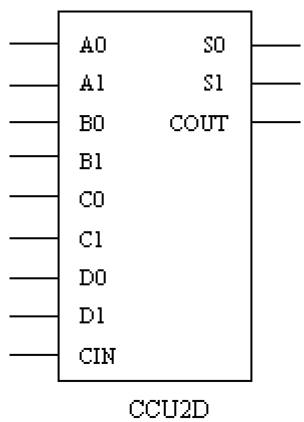
INJECT1_1: "YES" (default), "NO"

CCU2D

Carry Chain

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CIN, A1, B1, C1, D1, A0, B0, C0, D0

OUTPUTS: S1, S0, COUT

ATTRIBUTES:

INIT0: hexadecimal value (default: 16’h0000)

INIT1: hexadecimal value (default: 16’h0000)

INJECT1_0: "YES" (default), "NO"

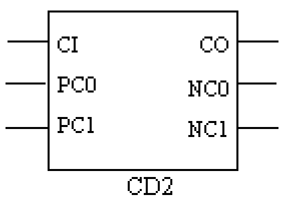
INJECT1_1: "YES" (default), "NO"

CD2

Combinational Logic for 2-Bit Down-Counter

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: CI, PC0, PC1

OUTPUTS: CO, NC0, NC1

Description

This primitive realizes the combinational logic needed to implement a 2-bit down-counter using ripple elements.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

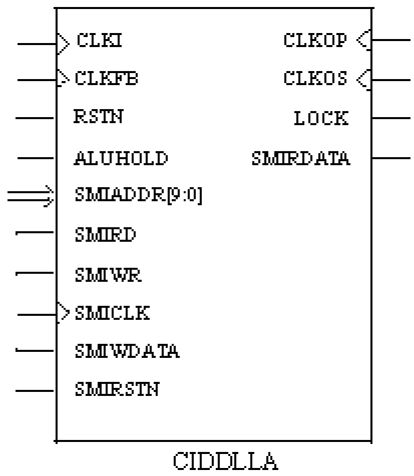
CIDDLLA

Clock Injection Delay Removal

Architectures Supported:

- LatticeECP2/M

▶ LatticeSC/M



INPUTS: CLKI, CLKFB, RSTN, ALUHOLD, SMIADDR9, SMIADDR8, SMIADDR7, SMIADDR6, SMIADDR5, SMIADDR4, SMIADDR3, SMIADDR2, SMIADDR1, SMIADDR0, SMIRD, SMIWR, SMICLK, SMIWDATA, SMIRSTN

OUTPUTS: CLKOP, CLKOS, LOCK, SMIRDATA

ATTRIBUTES:

CLKOP_PHASE: 0 (default), 90, 180, 270, 360

CLKOS_PHASE: 360 + (0, 11, 22, 45) (default: 360)

CLKOS_FPHASE: 0 (default), 11, 22, 45

CLKOP_DUTY50: "DISABLED" (default), "ENABLED"

CLKOS_DUTY50: "DISABLED" (default), "ENABLED"

CLKI_DIV: 1 (default), 2, 4

CLKOS_DIV: 1 (default), 2, 4

GSR: "DISABLED" (default), "ENABLED"

CLKOS_FDEL_ADJ: "DISABLED" (default), "ENABLED"

ALU_LOCK_CNT: integers 3~15 (default: 3)

ALU_UNLOCK_CNT: integers 3~15 (default: 3)

GLITCH_TOLERANCE: integers 0~7 (default: 2 for LatticeECP2/M; 0 for LatticeSC/M)

ALU_INIT_CNTVAL: 0 (default), 4, 8, 12, 16, 32, 48, 64, 72

LOCK_DELAY: integers 0~1000 (in ns) (default: 100)

(*LatticeSC/M only*) SMI_OFFSET: 0x400~0x7FF (default: 12'h410)

(*LatticeSC/M only*) MODULE_TYPE: "CIDDLLA"

(*LatticeSC/M only*) IP_TYPE: "CIDDLLA"

Description

CIDLLA removes the clock tree delay, aligning the external feedback clock to the reference clock. It has a single output coming from the fourth delay block. It features include clock tree insertion removal, N*Tcyc=4*Tdel + Tinj, lock achieved starting from minimum delay, and when it goes through all delay stages, the minimum frequency is 1/(4*Tdel). Its requirements include external feedback only, that you must use all delay cells, a maximum frequency of 700MHz, and a minimum frequency of 100MHz.

For more information, refer to the following technical notes on the Lattice web site:

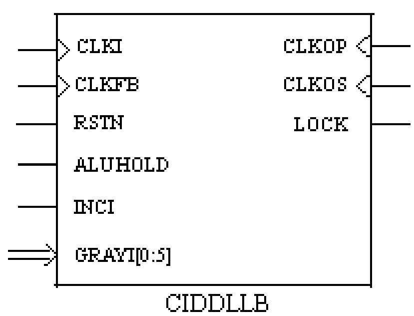
- TN1098 - LatticeSC sysCLOCK and PLL/DLL User's Guide
- TN1103 - LatticeECP2/M sysCLOCK PLL/DLL Design and Usage Guide

CIDDLLB

Clock Injection Delay Removal

Architectures Supported:

- LatticeECP3



INPUTS: CLKI, CLKFB, RSTN, ALUHOLD, INCI, GRAYI5, GRAYI4, GRAYI3, GRAYI2, GRAYI1, GRAYI0

OUTPUTS: CLKOP, CLKOS, LOCK

ATTRIBUTES:

CLKOP_PHASE: 0 (default), 90, 180, 270, 360

CLKOS_PHASE: 0 (default), 90, 180, 270, 360

CLKOS_FPHASE: 0 (default), 11, 22, 33, 45, 56, 67, 78, 90, 101, 112, 123, 135, 146, 157, 169, 191, 202, 214, 225, 236, 247, 259, 281, 292, 304, 315, 326, 337, 349

CLKI_DIV: 1 (default), 2, 4

CLKOS_DIV: 1 (default), 2, 4

GSR: "DISABLED" (default), "ENABLED"

ALU_LOCK_CNT: integers 3~15 (default: 3)

ALU_UNLOCK_CNT: integers 3~15 (default: 3)

GLITCH_TOLERANCE: integers 0~7 (default: 2)

ALU_INIT_CNTVAL: integers 0~31 (default: 0)

LOCK_DELAY: integers 0~1000 (in ns) (default: 100)

CLKOP_DUTY50: "DISABLED" (default), "ENABLED"

CLKOS_DUTY50: "DISABLED" (default), "ENABLED"

DEL0_GRAY: "DISABLED" (default), "ENABLED"

DEL1_GRAY: "DISABLED" (default), "ENABLED"

DEL2_GRAY: "DISABLED" (default), "ENABLED"

DEL3_GRAY: "DISABLED" (default), "ENABLED"

DEL4_GRAY: "DISABLED" (default), "ENABLED"

Description

CIDDLLB specifies the Clock Injection Delay Removal operation mode for the general purpose DLL (GDLL). In this mode, the feedback connection is supported and the CLKFB is captured on the CIDDLLB primitive.

Port Description

Port Name	Optional	Logical Capture Port Name
ALUHOLD	YES	HOLD
GRAYI[5:0]	YES	GRAY_IN[5:0]
INCI	YES	INC_IN
RSTN	YES	RSTN
CLKFB	NO	CLKFB
CLKI	NO	CLKI
CLK90	YES	CLK90
CLKOP	NO	CLKOP
CLKOS	YES	CLKOS
LOCK	NO	LOCK

For more information, refer to the following technical note on the Lattice web site:

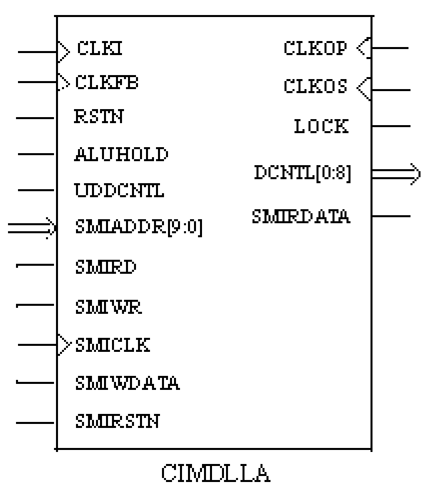
- TN1178 - LatticeECP3 sysCLOCK PLL/DLL Design and Usage Guide

CIMDLLA

Clock Injection Match

Architectures Supported:

- LatticeSC/M



INPUTS: CLKI, CLKFB, RSTN, ALUHOLD, UDDCNTL, SMIADDR9, SMIADDR8, SMIADDR7, SMIADDR6, SMIADDR5, SMIADDR4, SMIADDR3, SMIADDR2, SMIADDR1, SMIADDR0, SMIRD, SMIWR, SMICLK, SMIWDATA, SMIRSTN

OUTPUTS: CLKOP, CLKOS, LOCK, DCNTL0, DCNTL1, DCNTL2, DCNTL3, DCNTL4, DCNTL5, DCNTL6, DCNTL7, DCNTL8, SMIRDATA

ATTRIBUTES:

CLKOS_FPHASE: 0 (default), 11, 22, 45

CLKOS_DIV: 1 (default), 2, 4

GSR: "DISABLED" (default), "ENABLED"

ALU_LOCK_CNT: integers 3~15 (default: 3)

ALU_UNLOCK_CNT: integers 3~15 (default: 3)

GLITCH_TOLERANCE: integers 0~7 (default: 0)

DCNTL_ADJVAL: integers -127~127 (default: 0)

SMI_OFFSET: 0x400~0x7FF (default: 12'h410)

LOCK_DELAY: integers 0~1000 (in ns) (default: 100)

CLKOS_FDEL_ADJ: "DISABLED" (default), "ENABLED"

MODULE_TYPE: "CIMDLLA"

IP_TYPE: "CIMDLLA"

Description

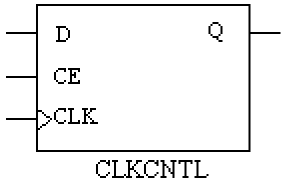
CIMDLLA matches the clock injection to one delay cell. This allows other inputs to take the registered ALU outputs and negate the clock injection delay. Its features include single clock output, lock achieved starting from minimum delay, output control bits, and allowance for +/- delay on these output control bits. Its requirements include external feedback (CLKOP) only, a maximum frequency of 700MHz, a minimum frequency of100MHz, and a maximum delay compensation of 3.9ns.

CLKCNTL

Clock Controller

Architectures Supported:

- LatticeSC/M



INPUTS: D, CE, CLK

OUTPUT: Q

ATTRIBUTES:

CLKMODE: "ECLK" (default), "SCLK"

Description

The CLKCNTL is the post-amble detect circuit required for the DQS input. The DQS generation will use the DELAY, TRDLLA and the CLKCNTL primitives.

The CLKCNTL primitive's instantiation rules allow the CLK input to be fed via secondary (local) routing paths, rather than constraining the routing to be via the Edge Clock tree. The Edge Clock tree is optimized for minimum skew rather than minimum delay. Although the Edge Clock delay is not a problem at the DDR/DDR2 clock frequencies originally targeted (300 MHz), the CLKCNTL is capable of operating at much higher frequencies (beyond 1 GHz). If the CLK input path utilizes the faster local routing resources, it is capable of properly gating an 800 MHz clock, as is required for testing of DDR3 memory devices. This requires changes to the mapper.

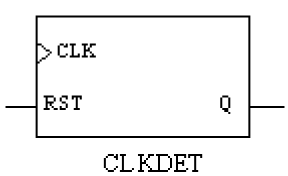
The CLKMODE attribute is supported for the CLKCNTL primitive. The legal values are ECLK (default) and SCLK.

CLKDET

Clock Detect

Architectures Supported:

- LatticeSC/M



INPUTS: CLK, RST

OUTPUT: Q

Truth Table

INPUTS		OUTPUTS
CK	RST	Q
X	1	0
↑	0	1

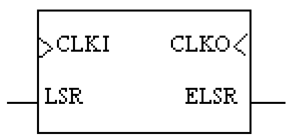
X = Don’t care

CLKDIV

Clock Divider

Architectures Supported:

- LatticeSC/M



INPUTS: CLKI, LSR

OUTPUTS: CLKO, ELSR

ATTRIBUTES:

DIV: 1 (default), 2, 4

GSR: "DISABLED" (default), "ENABLED"

Description

Clock divider. Refer to the following technical note on the Lattice web site for more details.

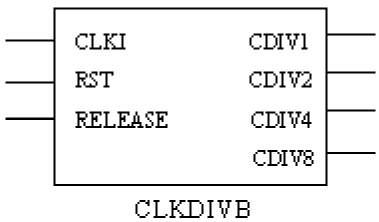
- TN1098 - LatticeSC sysCLOCK and PLL/DLL User’s Guide

CLKDIVB

Clock Divider

Architectures Supported:

- LatticeECP2/M
- LatticeECP3
- LatticeXP2



INPUTS: CLKI, RST, RELEASE

OUTPUTS: CDIV1, CDIV2, CDIV4, CDIV8

ATTRIBUTES:

GSR: "DISABLED" (default), "ENABLED"

Description

Clock divider. See the following table for port description.

Port Name	I/O	Definition
RELEASE	I	Asserting the RELEASE signal releases the divided outputs, synchronous to selected input source.
RST	I	Asserting the RST signal forces CDIV1 low synchronously, and forces CDIV2, CDIV4 and CDIV8 low asynchronously. De-asserting RST synchronously allows all outputs to toggle.
CLKI	I	Input clock.
CDIV1	O	Divide by 1 output port.
CDIV2	O	Divide by 2 output port.
CDIV4	O	Divide by 4 output port.
CDIV8	O	Divide by 8 output port.

Refer to the following technical notes on the Lattice web site for more details.

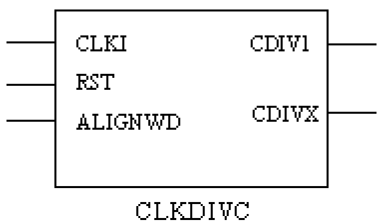
- TN1178 - LatticeECP3 sysCLOCK PLL/DLL Design and Usage Guide
- TN1126 - LatticeXP2 sysCLOCK PLL Design and Usage Guide
- TN1103 - LatticeECP2/M sysCLOCK PLL/DLL Design and Usage Guide

CLKDIVC

Clock Divider

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CLKI, RST, ALIGNWD

OUTPUTS: CDIV1, CDIVX

ATTRIBUTES:

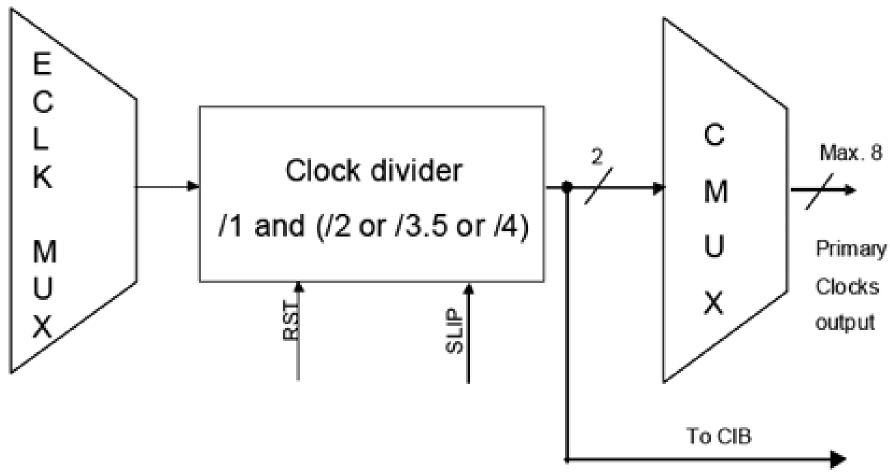
GSR: "DISABLED" (default), "ENABLED"

DIV: 2.0 (default), 3.5, 4.0

Description

A MachXO2/Platform Manager 2 device contains four CLKDIV with 1200 LUTs or more. The CLKDIV, or “clock divider” block, generates clock outputs one half, divided by three and a half or one quarter of the frequency of the input clock. It also generates an output clock of the same frequency as the input clock. All the outputs match input to output delay.

CLKDIV takes its inputs from the outputs of ECLK muxes. The divided outputs of the CLKDIV drive the primary clock center muxes directly and are also available on CIB ports for distribution to routing or secondary high fan out nets. The figure below represents the block diagram of CLKDIV:



The table below shows CLKDIVC IO description.

Port Name	I/O	Unused Nets	Description
RST	I	Tie low	Asserting the RST signal asynchronously forces all outputs low. De-asserting RST synchronously allows all outputs to toggle.
CLKI	I	Tie low	Input clock.
ALIGNWD	I	Tie low	This signal is used for word alignment.
CDIV1	O	Dangle	Divide by 1 output port.
CDIVX	O	Dangle	Divide by 2.0, 3.5, or 4.0 output port.

Refer to the following technical note on the Lattice web site for more details.

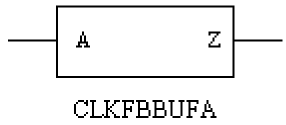
- TN1199 - MachXO2 sysCLOCK PLL Design and Usage Guide

CLKFBBUFA

Dummy Feedback Delay Between PLL clk Output and PLL fb Port

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUT: A

OUTPUT: Z

Description

The CLKFBBUFA is the dummy feedback path from the PLL clk output to the PLL feedback port to cancel out clock path variation over PVT.

The table below shows the IO description.

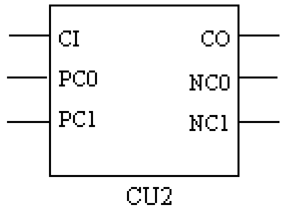
Port Name	I/O	Description
A	I	Clock input coming from the PLL CLKOP output
Z	O	Delayed output to the PLL fb port

CU2

Combinational Logic for 2-Bit Up-Counter

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: CI, PC0, PC1

OUTPUTS: CO, NC0, NC1

Description

This primitive realizes the combinational logic needed to implement a 2-bit up-counter using ripple elements.

When CI=0, NC[0:1]=PC[0:1] and CO=0

When CI=1, NC[0:1]=PC[0:1]+1, and CO=1 if PC[0:1]=11

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

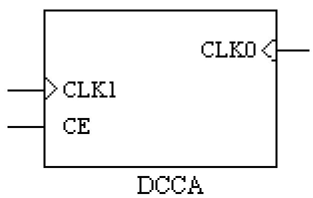
D

DCCA

Dynamic Quadrant Clock Enable/Disable

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CLKI, CE

OUTPUT: CLKO

Description

DCCA is the dynamic quadrant clock enable/disable primitive. In each quadrant, the dynamic quadrant clock enable/disable is available on the output of the center mux for the primary clocks CLK[5:0]. The dynamic quadrant clock enable/disable feature lets the internal logic control the quadrant primary clock network. When a clock network is disabled, all the logic fed by the clock network does not toggle, reducing the overall power consumption of the device. You need to instantiate a primitive (DCCA) in order to control the enable/disable function.

The DCCA IO description is shown below.

Port Name	I/O	Unused Nets	Description
CLKI	I	Tie low	Input clock
CE	I	Tie high	Clock enable
CLKO	O	Dangle	Output clock

DCCA Usage with VHDL

Library Instantiation

```
library lattice;
use lattice.components.all;
```

Component Declaration

```
component DCCA
   port (CLKI : in std_logic;
         CE   : in std_logic;
         CLKO : out std_logic);
end component;
```

DCCA Instantiation

```
I1: DCCA
   port map (CLKI => CLKI;
             CE   => CE;
             CLKO => CLKO);
end component;
```

DCCA Usage with Verilog HDL

Component Declaration

```
module DCCA (CLKI,
             CE,
             CLKO);
input CLKI;
input CE;
output CLKO;
endmodule
```

DCCA Instantiation

```
DCCA I1 (.CLKI (CLKI),
         .CE   (CE),
         .CLKO (CLKO));
```

For more information, refer to the following technical note on the Lattice web site:

- TN1199 - MachXO2 sysCLOCK PLL Design and Usage Guide

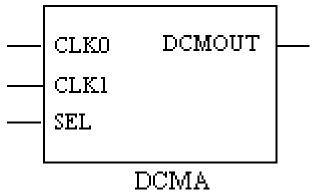
DCMA

Dynamic Clock Mux

Architectures Supported:

- MachXO2

- MachXO3L
- Platform Manager 2



INPUTS: CLK0, CLK1, SEL

OUTPUT: DCMOUT

Description

DCMA is a clock buffer incorporating a multiplexer function, whose output switches between the two clock inputs. It is not recommended that you route the PLL feedback signals through the multiplexer, because the PLL will loose lock.

DCMA IO description is shown in the below table.

Port Name	I/O	Unused Nets	Description
CLK0	I	Tie low	Input clock
CLKI	I	Tie low	Input clock
SEL	I	Tie low	Clock select from CIB
DCMOUT	O	Dangle	Output from the primary clock mux

DCMA Usage with VHDL

Library Instantiation

```
library lattice;
use lattice.components.all;
```

Component Declaration

```
component DCMA
   port (CLK0   : in std_logic;
         CLK1   : in std_logic;
         SEL    : in std_logic;
         DCMOUT :out std_logic);
end component;
```

DCMA Instantiation

```
I1: DCMA
   port map (CLK0 => CLK0,
             CLK1 => CLK1,
             SEL => SEL,
             DCMOUT => DCMOUT);
end component;
```

DCMA Usage with Verilog HDL

Component Declaration

```
module DCMA (CLK0, CLK1, SEL, DCMOUT);
input CLK0;
input CLK1;
input SEL;
output DCMOUT;
endmodule
```

DCMA Instantiation

```
DCMA I1 (.CLK0 (CLK0);
         .CLK1 (CLK1);
         .SEL (SEL);
         .DCMOUT (DCMOUT));
```

For additional information, see Lattice technical note on the web site:

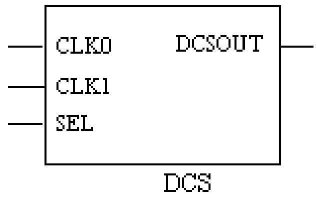
- TN1199 - MachXO2 sysCLOCK PLL Design and Usage Guide

DCS

Dynamic Clock Selection

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2



INPUTS: CLK0 CLK1 SEL

OUTPUT: DCSOUT

ATTRIBUTES:

DCSMODE: "NEG" (default), "POS", "HIGH_LOW", "HIGH_HIGH", "LOW_LOW", "LOW_HIGH","CLK0", "CLK1"

Note

For LatticeECP/EC, LatticeECP2/M, LatticeXP, and LatticeXP2 devices, the DCSMODE default has been changed from POS to NEG since ispLEVER 7.0 SP2. If your pre-7.0SP2 design included the DCS macro and you didn't specify the DCSMODE value, the design may behave differently in Diamond. To avoid the issue, you can manually set DCSMODE to POS in your old design.

Description

DCS is a global clock buffer incorporating a smart multiplexer function that takes two independent input clock sources and avoids glitches or runt pulses on the output clock, regardless of where the enable signal is toggled. Located in pairs at the center of each edge, there are eight DCS primitives per device.

Function	Pins
Input Clock Select	SEL
Primary Clock Input 0	CLK0
Primary Clock Input 1	CLK1
To Primary Clock Mid-Mux	DCSOUT

For each DCS:

- Inputs are from primary clocks at PLC routing block (one branch per DCS)
- Selects are from a routing block's LUT port (A0, B0, etc.)
- Outputs are connected to the DCS output sources of the feedline muxes

The outputs of the DCS then reach primary clock distribution via the feedlines. The connections from CIB to DCS are shown in the table below. The selected CIB interfaces to a PIC and is located at the center of each bank and next to the mid-muxes. Note that the CIB to DCS connections are merged with the existing PIC-CIB connections at that CIB. It is up to the hardware designer to select the most convenient CIB.

CIB to DCS Connections

CIB	DCS
C7	SEL
CLK0(jclk0)	CLK0
CLK1(jclk2)	CLK1

DCSMODE Values

Attribute Name	Description	Output		Value	DCS Fuse Settings				
		SEL=0	**SEL=1**		**0**	**1**	**2**	**3**	**4**
DCS MODE	Rising edge triggered, latched state is high	CLK0	CLK1	POS	1	0	0	0	0
	Falling edge triggered, latched state is low	CLK0	CLK1	NEG	0	0	0	0	0
	Sel is active high, Disabled output is low	0	CLK1	HIGH_LOW	0	1	0	0	0
	Sel is active high, Disabled output is high	1	CLK1	HIGH_HIGH	1	1	0	0	0
	Sel is active low, Disabled output is low	CLK0	0	LOW_LOW	0	0	1	0	0
	Sel is active low, Disabled output is high	CLK0	1	LOW_HIGH	1	0	1	0	0
	Buffer for CLK0	CLK0	CLK0	CLK0	0	0	1	0	1
	Buffer for CLK1	CLK1	CLK1	CLK1	0	1	0	1	0

For additional information, see Lattice technical notes on the web site:

- TN1178 - LatticeECP3 sysCLOCK and PLL/DLL Design and Usage Guide
- TN1098 - LatticeSC sysCLOCK and PLL/DLL User's Guide
- TN1103 - LatticeECP2 sysCLOCK PLL/DLL Design and Usage Guide
- TN1126 - LatticeXP2 sysCLOCK PLL Design and Usage Guide

- TN1049 - LatticeECP/EC and LatticeXP sysCLOCK PLL Design and Usage Guide

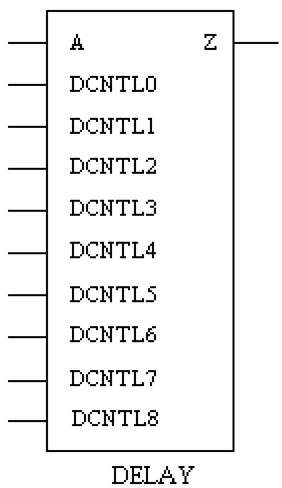
Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

DELAY

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: A, DCNTL0, DCNTL1, DCNTL2, DCNTL3, DCNTL4, DCNTL5, DCNTL6, DCNTL7, DCNTL8

OUTPUT: Z

Description

Sets the input delay for an input. You can choose either dynamic delay or the static delay. For more usage, see related technical notes or contact technical support.

Truth Table

INPUTS	OUTPUTS
A	Z
0	0
1	1

Note

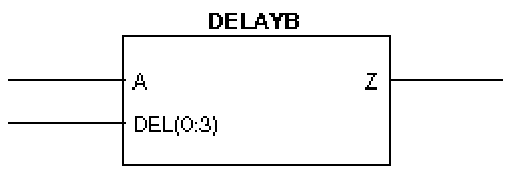
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

DELAYB

Dynamic Delay in PIO

Architectures Supported:

- LatticeECP2/M
- LatticeECP3
- LatticeXP2



INPUTS: A, DEL0, DEL1, DEL2, DEL3

OUTPUT: Z

Description

Data going to the DDR registers can be optionally delayed using the delay block. The DELAYB block receives a 4-bit delay value from the DLL. The user can also choose to implement a fixed delay value instead of using the delay generated by the DLL. The various fixed delay choices can be made in the IPexpress software tool.

The DELAYB block can also be used to delay non-DDR inputs that use the input PIO register.

DELAYB Port Definition

Port Name	I/O	Definition
A	I	DDR input from sysIO buffer.
DEL [0:3]	I	Delay inputs.
Z	O	Output with delay.

Refer to the following technical notes for more details:

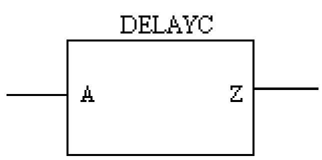
- TN1180 - LatticeECP3 High-Speed I/O Interface
- TN1105 - LatticeECP2/M High-Speed I/O Interface
- TN1138 - LatticeXP2 High-Speed I/O Interface

DELAYC

Fixed Delay in PIO

Architectures Supported:

- LatticeECP3



INPUT: A

OUTPUT: Z

DELAYC Port Definition

Port Name	I/O	Definition
A	I	DDR input from sysIO buffer.
Z	O	Output with delay.

Refer to the following technical note for more details:

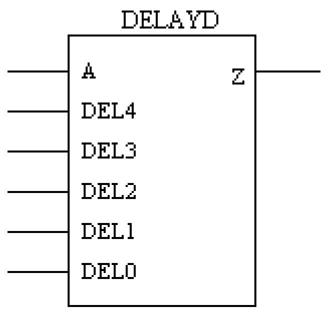
- TN1180 - LatticeECP3 High-Speed I/O Interface

DELAYD

Dynamic Delay for Bottom Bank

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUT: A, DEL4, DEL3, DEL2, DEL1, DEL0

OUTPUT: Z

Description

DELAYD is the dynamic delay for VPIC_RX, IOLA and IOLC cells in bottom side only. It can be used for x2, x4 and 7:1 applications. See the below table for its port description.

Port Name	I/O	Definition
A	I	Data input from IO buffer
DEL4, DEL3, DEL2, DEL1, DEL0	I	Dynamic delay inputs from CIB
Z	O	Output with delay

Refer to the following technical note for more details:

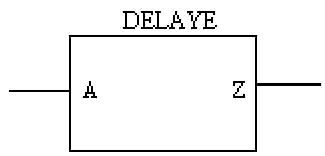
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

DELAYE

Fixed Delay in PIO

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUT: A

OUTPUT: Z

ATTRIBUTES:

DEL_MODE: "SCLK_ZEROHOLD", "ECLK_ALIGNED", "ECLK_CENTERED", "SCLK_ALIGNED", "SCLK_CENTERED", "USER_DEFINED" (default)

DEL_VALUE: "DELAY0" (default), "DELAY1", "DELAY2", ..., "DELAY31"

Description

DELAYE is the fix delay in PIO for all banks and all sides. It can be used for all IO registers and DDR types. See the below table for its IO port description.

Port Name	I/O	Definition
A	I	DDR input from sysIO buffer
Z	O	Output with delay

Refer to the following technical note for more details:

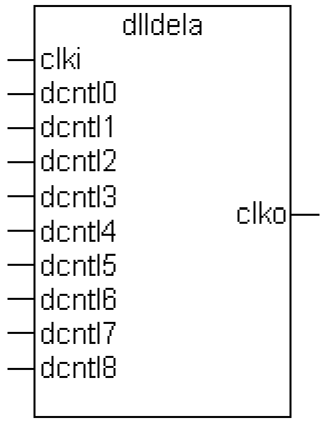
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

DLLDELA

Slave Delay

Architectures Supported:

- LatticeECP2/M



INPUTS: CLKI, DCNTL0, DCNTL1, DCNTL2, DCNTL3, DCNTL4, DCNTL5, DCNTL6, DCNTL7, DCNTL8

OUTPUT: CLKO

Description

The Slave Delay Line is designed to generate desired delay in DDR/SPI4 applications. The delay control inputs (DCNTL[8:0]) are fed from the general purpose DLL outputs. The following table shows its port description.

Name	I/O	Description
CLKI	I	Clock input
DCNTL[8:0]	I	Delay control bits
CLKO	O	Clock output

For more details such as application examples, refer to the following technical note on the Lattice web site:

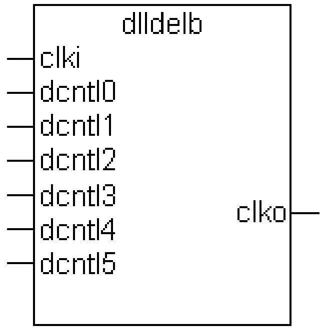
- TN1103 - LatticeECP2 sysCLOCK PLL/DLL Design and Usage Guide

DLLDELB

Slave Delay

Architectures Supported:

- LatticeECP3



INPUTS: CLKI, DCNTL5, DCNTL4, DCNTL3, DCNTL2, DCNTL1, DCNTL0

OUTPUT: CLKO

Description

DLLDELB is the LatticeECP3 slave delay line primitive. The DLLDELB port description is shown in the below table.

Port Name	I/O	Description
CLKI	I	Clock input
DCNTL0	I	Control bit 0 (hard wired to DLL)
DCNTL1	I	Control bit 1 (hard wired to DLL)
DCNTL2	I	Control bit 2 (hard wired to DLL)
DCNTL3	I	Control bit 3 (hard wired to DLL)
DCNTL4	I	Control bit 4 (hard wired to DLL)
DCNTL5	I	Control bit 5 (hard wired to DLL)
CLKO	O	Clock output

DLLDELB Usage with VHDL

```
COMPONENT DLLDELB
   PORT (CLKI   : IN  std_logic;
         DCNTL0 : IN  std_logic;
         DCNTL1 : IN  std_logic;
         DCNTL2 : IN  std_logic;
         DCNTL3 : IN  std_logic;
         DCNTL4 : IN  std_logic;
         DCNTL5 : IN  std_logic;
         CLKO   : OUT std_logic);
END COMPONENT;
```

```
begin
DLLDELAinst0: DLLDELB
   PORT MAP (
             CLKI   =>  clkisig,
             DCNTL0 =>  dcntl0sig,
             DCNTL1 =>  dcntl1sig,
             DCNTL2 =>  dcntl2sig,
             DCNTL3 =>  dcntl3sig,
             DCNTL4 =>  dcntl4sig,
             DCNTL5 =>  dcntl5sig,
             CLKO   =>  clkosig
            );
```

For more details such as application examples, refer to the following technical note on the Lattice web site:

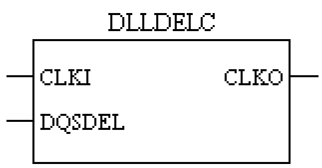
- TN1178 - LatticeECP3 sysCLOCK and PLL/DLL Design and Usage Guide

DLLDELC

Clock Shifting for ECLK or PCLK

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CLKI, DQSDEL

OUTPUT: CLKO

ATTRIBUTES:

DEL_ADJ: "PLUS" (default), MINUS

DEL_VAL: integers 0~127 (PLUS), 1~128 (MINUS) (default: 0)

Description

DLLDELC is the generic input clock shifting using DQSDEL from DQSDLL. The DLLDELC port description is shown in the below table.

Port Name	I/O	Description
CLKI	I	clk input from I/O buffer
DQSDEL	I	Dynamic delay inputs from DQSDLLC
CLKO	O	Clock output with delay

For more details such as application examples, refer to the following technical note on the Lattice web site:

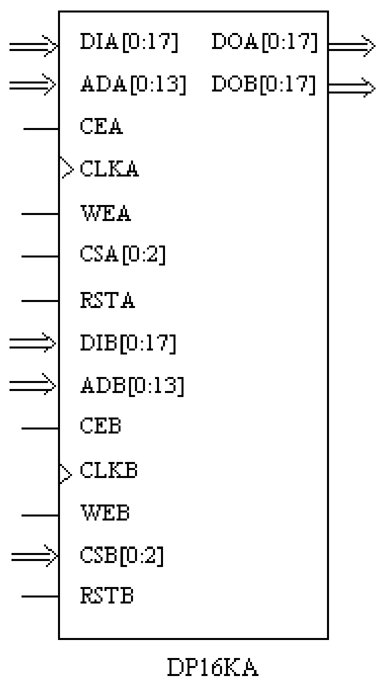
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

DP16KA

16K Dual Port Block RAM

Architectures Supported:

- LatticeSC/M



INPUTS: DIA0, DIA1, DIA2, DIA3, DIA4, DIA5, DIA6, DIA7, DIA8, DIA9, DIA10, DIA11, DIA12, DIA13, DIA14, DIA15, DIA16, DIA17, ADA0, ADA1,

ADA2, ADA3, ADA4, ADA5, ADA6, ADA7, ADA8, ADA9, ADA10, ADA11, ADA12, ADA13, CEA, CLKA, WEA, CSA0, CSA1, CSA2, RSTA, DIB0, DIB1, DIB2, DIB3, DIB4, DIB5, DIB6, DIB7, DIB8, DIB9, DIB10, DIB11, DIB12, DIB13, DIB14, DIB15, DIB16, DIB17, ADB0, ADB1, ADB2, ADB3, ADB4, ADB5, ADB6, ADB7, ADB8, ADB9, ADB10, ADB11, ADB12, ADB13, CEB, CLKB, WEB, CSB0, CSB1, CSB2, RSTB

OUTPUTS: DOA0, DOA1, DOA2, DOA3, DOA4, DOA5, DOA6, DOA7, DOA8, DOA9, DOA10, DOA11, DOA12, DOA13, DOA14, DOA15, DOA16, DOA17, DOB0, DOB1, DOB2, DOB3, DOB4, DOB5, DOB6, DOB7, DOB8, DOB9, DOB10, DOB11, DOB12, DOB13, DOB14, DOB15, DOB16, DOB17

ATTRIBUTES:

DATA_WIDTH_A: 1, 2, 4, 9, 18 (default)

DATA_WIDTH_B: 1, 2, 4, 9, 18 (default)

REGMODE_A: "NOREG" (default), "OUTREG"

REGMODE_B: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_A: any 3-bit binary value (default: 3’b000)

CSDECODE_B: any 3-bit binary value (default: 3’b000)

WRITEMODE_A: "NORMAL" (default), "WRITETHROUGH"

WRITEMODE_B: "NORMAL" (default), "WRITETHROUGH"

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_3F: (*Verilog*) 320'hXXX...X (80-bit hexadecimal value)
(*VHDL*) 0xXXX...X (80-bit hexadecimal value)
Default: all zeros

Description

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

- TN1094 - On-Chip Memory Usage Guide for LatticeSC Devices

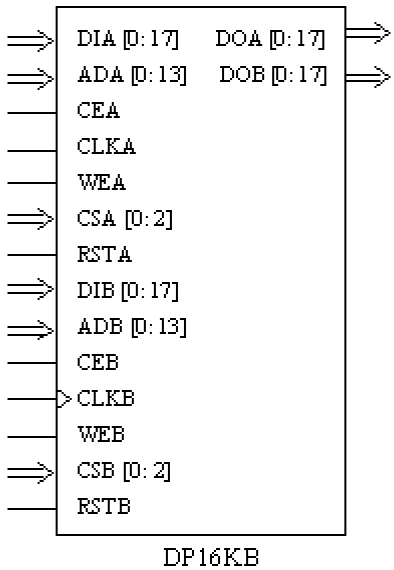
DP16KB

True Dual Port Block RAM

Architectures Supported:

- LatticeECP2/M

▶ LatticeXP2



INPUTS: DIA0, DIA1, DIA2, DIA3, DIA4, DIA5, DIA6, DIA7, DIA8, DIA9, DIA10, DIA11, DIA12, DIA13, DIA14, DIA15, DIA16, DIA17, ADA0, ADA1, ADA2, ADA3, ADA4, ADA5, ADA6, ADA7, ADA8, ADA9, ADA10, ADA11, ADA12, ADA13, CEA, CLKA, WEA, CSA0, CSA1, CSA2, RSTA, DIB0, DIB1, DIB2, DIB3, DIB4, DIB5, DIB6, DIB7, DIB8, DIB9, DIB10, DIB11, DIB12, DIB13, DIB14, DIB15, DIB16, DIB17, ADB0, ADB1, ADB2, ADB3, ADB4, ADB5, ADB6, ADB7, ADB8, ADB9, ADB10, ADB11, ADB12, ADB13, CEB, CLKB, WEB, CSB0, CSB1, CSB2, RSTB

OUTPUTS: DOA0, DOA1, DOA2, DOA3, DOA4, DOA5, DOA6, DOA7, DOA8, DOA9, DOA10, DOA11, DOA12, DOA13, DOA14, DOA15, DOA16, DOA17, DOB0, DOB1, DOB2, DOB3, DOB4, DOB5, DOB6, DOB7, DOB8, DOB9, DOB10, DOB11, DOB12, DOB13, DOB14, DOB15, DOB16, DOB17

ATTRIBUTES:

DATA_WIDTH_A: 1, 2, 4, 9, 18 (default)

DATA_WIDTH_B: 1, 2, 4, 9, 18 (default)

REGMODE_A: "NOREG" (default), "OUTREG"

REGMODE_B: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_A: any 3-bit binary value (default: 0b000)

CSDECODE_B: any 3-bit binary value (default: 0b000)

WRITEMODE_A: "NORMAL" (default), "WRITETHROUGH"

WRITEMODE_B: "NORMAL" (default), "WRITETHROUGH"

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_3F: 80-bit hexadecimal string (default: all zeros)

Description

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

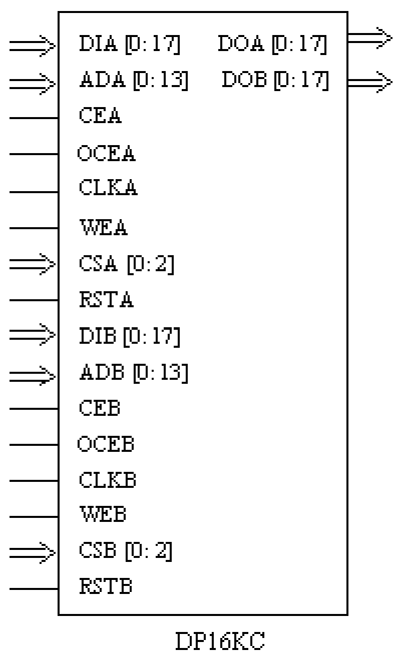
- TN1104 - LatticeECP2/M Memory Usage Guide

DP16KC

True Dual Port Block RAM

Architectures Supported:

- LatticeECP3



INPUTS: DIA17, DIA16, DIA15, DIA14, DIA13, DIA12, DIA11, DIA10, DIA9, DIA8, DIA7, DIA6, DIA5, DIA4, DIA3, DIA2, DIA1, DIA0, ADA13, ADA12, ADA11, ADA10, ADA9, ADA8, ADA7, ADA6, ADA5, ADA4, ADA3, ADA2, ADA1, ADA0, CEA, OCEA, CLKA, WEA, CSA2, CSA1, CSA0, RSTA, DIB17, DIB16, DIB15, DIB14, DIB13, DIB12, DIB11, DIB10, DIB9, DIB8, DIB7, DIB6, DIB5, DIB4, DIB3, DIB2, DIB1, DIB0, ADB13, ADB12, ADB11, ADB10, ADB9,

ADB8, ADB7, ADB6, ADB5, ADB4, ADB3, ADB2, ADB1, ADB0, CEB, OCEB, CLKB, WEB, CSB2, CSB1, CSB0, RSTB

OUTPUTS: DOA17, DOA16, DOA15, DOA14, DOA13, DOA12, DOA11, DOA10, DOA9, DOA8, DOA7, DOA6, DOA5, DOA4, DOA3, DOA2, DOA1, DOA0, DOB17, DOB16, DOB15, DOB14, DOB13, DOB12, DOB11, DOB10, DOB9, DOB8, DOB7, DOB6, DOB5, DOB4, DOB3, DOB2, DOB1, DOB0

ATTRIBUTES:

DATA_WIDTH_A: 1, 2, 4, 9, 18 (default)

DATA_WIDTH_B: 1, 2, 4, 9, 18 (default)

REGMODE_A: "NOREG" (default), "OUTREG"

REGMODE_B: "NOREG" (default), "OUTREG"

CSDECODE_A: any 3-bit binary value (default: all zeros)

CSDECODE_B: any 3-bit binary value (default: all zeros)

WRITEMODE_A: "NORMAL" (default), "WRITETHROUGH"

WRITEMODE_B: "NORMAL" (default), "WRITETHROUGH"

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_3F: "0xXXX....X" (80-bit hex string) (default: all zeros)

Description

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

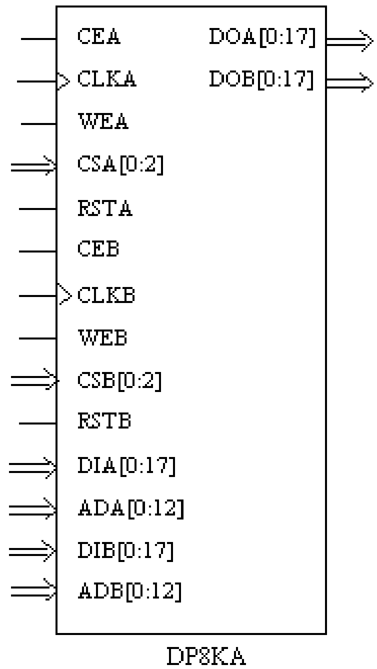
- TN1179 - LatticeECP3 Memory Usage Guide

DP8KA

8K Dual Port Block RAM

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: CEA, CLKA, WEA, CSA0, CSA1, CSA2, RSTA, CEB, CLKB, WEB, CSB0, CSB1, CSB2, RSTB, DIA0, DIA1, DIA2, DIA3, DIA4, DIA5, DIA6, DIA7, DIA8, DIA9, DIA10, DIA11, DIA12, DIA13, DIA14, DIA15, DIA16, DIA17, ADA0, ADA1, ADA2, ADA3, ADA4, ADA5, ADA6, ADA7, ADA8, ADA9, ADA10, ADA11, ADA12, DIB0, DIB1, DIB2, DIB3, DIB4, DIB5, DIB6, DIB7, DIB8, DIB9, DIB10, DIB11, DIB12, DIB13, DIB14, DIB15, DIB16, DIB17, ADB0, ADB1, ADB2, ADB3, ADB4, ADB5, ADB6, ADB7, ADB8, ADB9, ADB10, ADB11, ADB12

OUTPUTS: DOA0, DOA1, DOA2, DOA3, DOA4, DOA5, DOA6, DOA7, DOA8, DOA9, DOA10, DOA11, DOA12, DOA13, DOA14, DOA15, DOA16, DOA17, DOB0, DOB1, DOB2, DOB3, DOB4, DOB5, DOB6, DOB7, DOB8, DOB9, DOB10, DOB11, DOB12, DOB13, DOB14, DOB15, DOB16, DOB17

ATTRIBUTES:

DATA_WIDTH_A: 1, 2, 4, 9, 18 (default)

DATA_WIDTH_B: 1, 2, 4, 9, 18 (default)

REGMODE_A: "NOREG" (default), "OUTREG"

REGMODE_B: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_A: any 3-bit binary value (default: 111)

CSDECODE_B: any 3-bit binary value (default: 111)

WRITEMODE_A: "NORMAL" (default), "WRITETHROUGH"

WRITEMODE_B: "NORMAL" (default), "WRITETHROUGH"

INITVAL_00 to INITVAL_1F: (*Verilog*) 320'hXXX...X (80-bit hexadecimal value)
(*VHDL*) 0xXXX...X (80-bit hexadecimal value)
Default: all zeros

Description

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

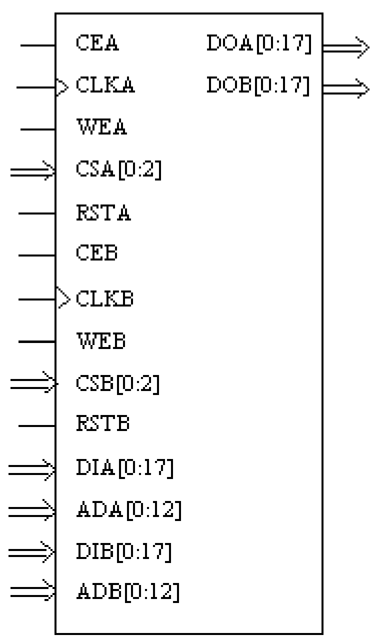
- TN1051 - Memory Usage Guide for LatticeECP/EC and LatticeXP Devices

DP8KB

8K Dual Port Block RAM

Architectures Supported:

- MachXO
- Platform Manager



DP8KB

INPUTS: CEA, CLKA, WEA, CSA0, CSA1, CSA2, RSTA, CEB, CLKB, WEB, CSB0, CSB1, CSB2, RSTB, DIA0, DIA1, DIA2, DIA3, DIA4, DIA5, DIA6, DIA7, DIA8, DIA9, DIA10, DIA11, DIA12, DIA13, DIA14, DIA15, DIA16, DIA17, ADA0, ADA1, ADA2, ADA3, ADA4, ADA5, ADA6, ADA7, ADA8, ADA9, ADA10, ADA11, ADA12, DIB0, DIB1, DIB2, DIB3, DIB4, DIB5, DIB6, DIB7, DIB8, DIB9, DIB10, DIB11, DIB12, DIB13, DIB14, DIB15, DIB16, DIB17, ADB0, ADB1, ADB2, ADB3, ADB4, ADB5, ADB6, ADB7, ADB8, ADB9, ADB10, ADB11, ADB12

OUTPUTS: DOA0, DOA1, DOA2, DOA3, DOA4, DOA5, DOA6, DOA7, DOA8, DOA9, DOA10, DOA11, DOA12, DOA13, DOA14, DOA15, DOA16, DOA17, DOB0, DOB1, DOB2, DOB3, DOB4, DOB5, DOB6, DOB7, DOB8, DOB9, DOB10, DOB11, DOB12, DOB13, DOB14, DOB15, DOB16, DOB17

ATTRIBUTES:

DATA_WIDTH_A: 1, 2, 4, 9, 18 (default)

DATA_WIDTH_B: 1, 2, 4, 9, 18 (default)

REGMODE_A: "NOREG" (default), "OUTREG"

REGMODE_B: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_A: any 3-bit binary value (default: all zeros)

CSDECODE_B: any 3-bit binary value (default: all zeros)

WRITEMODE_A: "NORMAL" (default), "WRITETHROUGH"

WRITEMODE_B: "NORMAL" (default), "WRITETHROUGH"

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_1F: (*Verilog*) 320'hXXX...X (80-bit hex value)
(*VHDL*) 0xXXX...X (80-bit hex value)
Default: all zeros

Description

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

- TN1092 - MachXO Memory Usage Guide

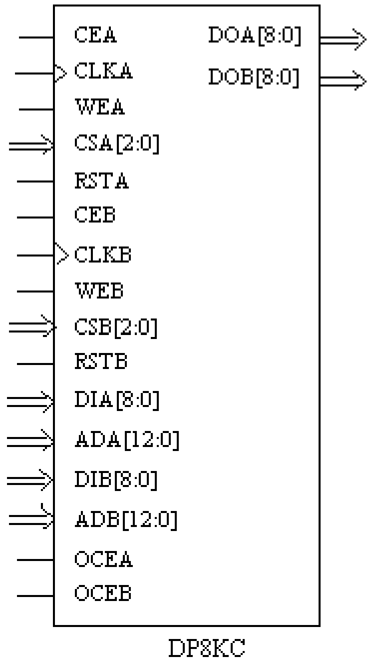
DP8KC

8K True Dual Port Block RAM

Architectures Supported:

- MachXO2

- MachXO3L
- Platform Manager 2



INPUTS: CEA, CLKA, WEA, CSA2, CSA1, CSA0, RSTA, CEB, CLKB, WEB, CSB2, CSB1, CSB0, RSTB, DIA8, DIA7, DIA6, DIA5, DIA4, DIA3, DIA2, DIA1, DIA0, ADA12, ADA11, ADA10, ADA9, ADA8, ADA7, ADA6, ADA5, ADA4, ADA3, ADA2, ADA1, ADA0, DIB8, DIB7, DIB6, DIB5, DIB4, DIB3, DIB2, DIB1, DIB0, ADB12, ADB11, ADB10, ADB9, ADB8, ADB7, ADB6, ADB5, ADB4, ADB3, ADB2, ADB1, ADB0, OCEA, OCEB

OUTPUTS: DOA8, DOA7, DOA6, DOA5, DOA4, DOA3, DOA2, DOA1, DOA0, DOB8, DOB7, DOB6, DOB5, DOB4, DOB3, DOB2, DOB1, DOB0

ATTRIBUTES:

DATA_WIDTH_A: 1, 2, 4, 9 (default)

DATA_WIDTH_B: 1, 2, 4, 9 (default)

REGMODE_A: "NOREG" (default), "OUTREG"

REGMODE_B: "NOREG" (default), "OUTREG"

CSDECODE_A: any 3-bit binary value (default: 3’b000)

CSDECODE_B: any 3-bit binary value (default: 3’b000)

WRITEMODE_A: "NORMAL" (default), "WRITETHROUGH", "READBEFORE"

WRITEMODE_B: "NORMAL" (default), "WRITETHROUGH", "READBEFORE"

GSR: "ENABLED" (default), "DISABLED"

RESETMODE: "SYNC" (default), "ASYNC"

ASYNC_RESET_RELEASE: "SYNC" (default), "ASYNC"

INIT_DATA: "STATIC" (default), "DYNAMIC"

INITVAL_00 to INITVAL_1F: (*Verilog*) 320'hXXX...X (80-bit hex value)
(*VHDL*) 0xXXX...X (80-bit hex value)
Default: all zeros

Description

The following table describes the I/O ports of the DP8KC primitive.

Port Name	I/O	Definition
CEA, CEB	I	Clock enable for port CLKA and CLKB
OCEA, OCEB	I	Output clock enable for port A and B
CLKA, CLKB	I	Clock for port A and B
RSTA, RSTB	I	Reset for port A and B
WEA, WEB	I	Write enable for port A and B
CSA[2:0], CSB[2:0]	I	Chip select for port A and B
DIA[8:0], DIB[8:0]	I	Input data port A and B (up to 9)
ADA[12:0], ADB[12:0]	I	Address bus port A and B (up to 13)
DOA[8:0], DOB[8:0]	O	Output data port A and B (up to 9)

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

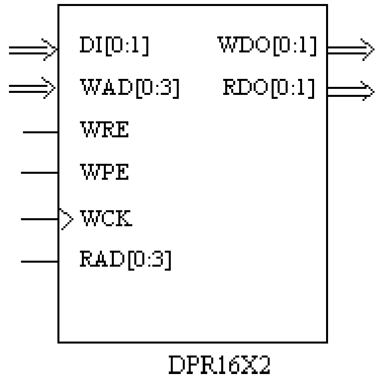
- TN1201 - Memory Usage Guide for MachXO2 Devices

DPR16X2

Distributed Dual Port RAM

Architectures Supported:

- LatticeSC/M



INPUTS: DI0, DI1, WAD0, WAD1, WAD2, WAD3, WRE, WPE, WCK, RAD0, RAD1, RAD2, RAD3

OUTPUTS: WDO0, WDO1, RDO0, RDO1

ATTRIBUTES:

INITVAL: (*Verilog*) 64'hXXXXXXXX (16-bit hexadecimal value)
(*VHDL*) 0xXXXXXXXX (16-bit hexadecimal value)
Default: all zeros

GSR: "ENABLED" (default), "DISABLED"

Description

The DPR16X2 symbol represents a 16-word by 2-bit distributed dual port RAM. You can refer to the following technical note on the Lattice web site for port definition, attribute definition and usage.

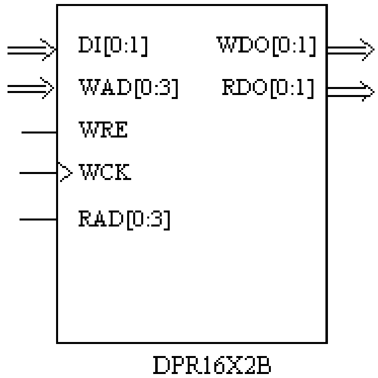
- TN1094 - On-Chip Memory Usage Guide for LatticeSC Devices

DPR16X2B

Distributed Dual Port RAM

Architectures Supported:

- LatticeECP/EC
- LatticeXP
- MachXO
- Platform Manager



INPUTS: DI0, DI1, WAD0, WAD1, WAD2, WAD3, WRE, WCK, RAD0, RAD1, RAD2, RAD3

OUTPUTS: WDO0, WDO1, RDO0, RDO1

ATTRIBUTES:

INITVAL: (*Verilog*) 64'hXXXXXXXXXXXXXXXX (16-bit hexadecimal value)
(*VHDL*) 0xXXXXXXXXXXXXXXXX (16-bit hexadecimal value)
Default: all zeros

Description

You can refer to the following technical notes on the Lattice web site for port definition, attribute definition and usage.

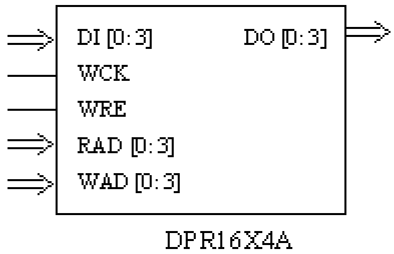
- TN1051 - Memory Usage Guide for LatticeECP/EC and LatticeXP Devices
- TN1092 - MachXO Memory Usage Guide

DPR16X4A

Distributed Pseudo Dual Port RAM

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: DI0, DI1, DI2, DI3, WCK, WRE, RAD0, RAD1, RAD2, RAD3, WAD0, WAD1, WAD2, WAD3

OUTPUTS: DO0, DO1, DO2, DO3

Description

PFU-based distributed Pseudo Dual-port RAM primitive. See Memory Primitives Overview for individual port description.

You can also refer to the following technical notes on the Lattice web site for port definition, attributes and usage.

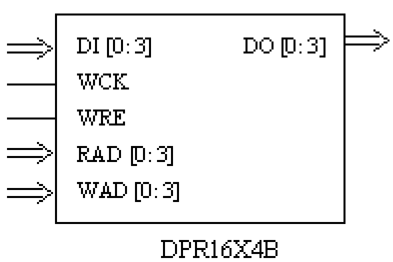
- TN1104 - LatticeECP2/M Memory Usage Guide
- TN1137 - LatticeXP2 Memory Usage Guide

DPR16X4B

Distributed Pseudo Dual Port RAM

Architectures Supported:

- LatticeXP2



INPUTS: WAD0, WAD1, WAD2, WAD3, DI0, DI1, DI2, DI3, WCK, WRE, RAD0, RAD1, RAD2, RAD3

OUTPUTS: DO0, DO1, DO2, DO3

ATTRIBUTES:

INITVAL: (*Verilog*) "64'hXXXXXXXXXXXXXXXX" (16-bit hex string)
(*VHDL*) "0xXXXXXXXXXXXXXXXX" (16-bit hex string)
Default: all zeros

Description

PFU-based distributed Pseudo Dual-port RAM primitive. See Memory Primitives Overview for individual port description.

You can also refer to the following technical note on the Lattice web site for port definition, attributes and usage.

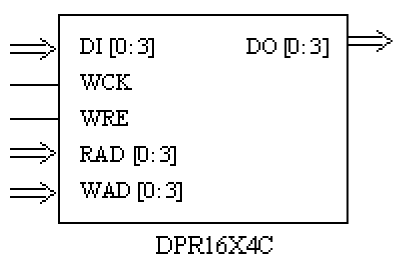
- TN1137 - LatticeXP2 Memory Usage Guide

DPR16X4C

Distributed Pseudo Dual Port RAM

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: DI3, DI2, DI1, DI0, WAD3,WAD2,WAD1,WAD0, WCK, WRE, RAD3,RAD2,RAD1,RAD0

OUTPUTS: DO3, DO2, DO1, DO0

ATTRIBUTES:

INITVAL: "0xXXXXXXXXXXXXXXXX" (16-bit hex string) (default: all zeros)

Description

PFU-based distributed Pseudo Dual-port RAM primitive. See Memory Primitives Overview for individual port description.

You can also refer to the following technical notes on the Lattice web site for port definition, attributes and usage.

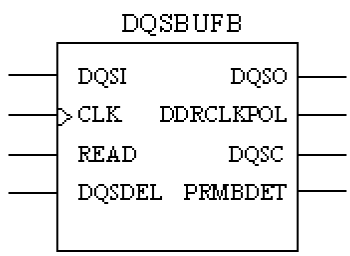
- TN1201 - Memory Usage Guide for MachXO2 Devices
- TN1179 - LatticeECP3 Memory Usage Guide

DQSBUFB

DDR DQS Buffer Used as DDR memory DQS generator

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: DQSI, CLK, READ, DQSDEL

OUTPUTS: DQSO, DDRCLKPOL, DQSC, PRMBDET

Description

This cell is used to indicate how many DDR I/Os need to be tied together, aligning the placement of the DDR cell. The input goes to the clock and the output goes to the clock on the DDR cell. For more usage, see related technical notes or contact technical support.

Refer to the following technical note on the Lattice web site for more details:

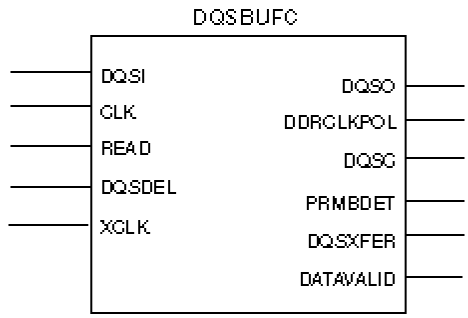
- TN1050 - LatticeECP/EC and LatticeXP DDR Usage Guide

DQSBUFC

DQS Delay Function and Clock Polarity Selection Logic

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: DQSI, CLK, XCLK, READ, DQSDEL

OUTPUTS: DQSO, DDRCLKPOL, DQSC, PRMBDET, DQSXFER, DATAVALID

ATTRIBUTES:

DQS_LI_DEL_VAL: integers 0~63 (PLUS), 1~64 (MINUS) (default: 4)

DQS_LI_DEL_ADJ: "MINUS" (default), "PLUS"

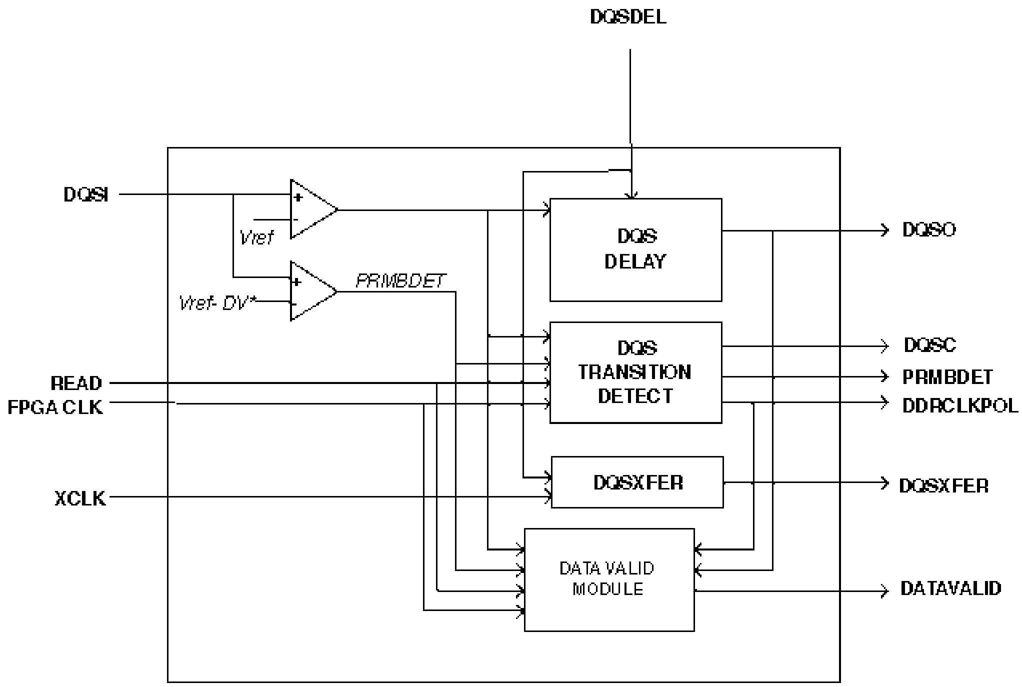
DQS_LO_DEL_VAL: integers 0~63 (PLUS), 1~64 (MINUS) (default: 0)

DQS_LO_DEL_ADJ: "PLUS" (default), "MINUS"

Description

DQSBUFC implements the DQS delay and the DQS transition detector logic. The primitive is composed of the DQS Delay, the DQS Transition Detect and the DQSXFER block as shown in the following figure. This block inputs the DQS and delays it by 90 degrees. It also generates the DDR Clock Polarity and the DQSXFER signal. The preamble detect (PRMBDET) signal is generated from the DQSI input using a voltage divider circuit.

DQSBUFC Function



*DV ~ 170mV for DDR1 (SSTL25 signaling)
*DV ~ 120mV for DDR2 (SSTL18 signaling)

DQS Delay Block: The DQS Delay block receives the digital control delay line (DQSDEL) coming from one of the two DQSDLL blocks. These control signals are used to delay the DQSI by 90 degrees. DQSO is the delayed DQS and is connected to the clock input of the first set of DDR registers.

DQS Transition Detect: The DQS Transition Detect block generates the DDR Clock Polarity signal based on the phase of the FPGA clock at the first DQS transition. The DDR READ control signal and FPGA CLK inputs to this coming and should be coming from the FPGA core.

DQSXFER: This block generates the 90-degree phase shifted clock to for the DDR Write interface. The input to this block is the XCLK. The user can choose to connect this either to the edge clock or the FPGA clocks. The DQSXFER is routed using the DQSXFER tree to all the I/Os spanned by that DQS.

Data Valid Module: The data valid module generates a DATAVALID signal. This signal indicates to the FPGA that valid data is transmitted out of the input DDR registers to the FPGA core.

The following table describes DQSBUFC I/O ports.

Port Name	I/O	Definition
DQSI	I	DQS strobe signal from memory.
CLK	I	System CLK.
READ	I	Read generated from the FPGA core.
DQSDEL	I	DQS delay from the DQSDLL primitive.
XCLK	I	Edge clock or system CLK.
DQSO	O	Delayed DQS strobe signal, to the input capture register block.
DQSC	O	DQS strobe signal before delay, going to the FPGA core logic.
DDRCLKPOL	O	DDR clock polarity signal.
PRMBDET	O	Preamble detect signal, going to the FPGA core logic.
DQSXFER	O	90 degree shifted clock going to the output DDR register block.
DTATVALID	O	Signal indicating transmission of valid data to the FPGA core.

READ Pulse Generation

The READ signal to the DQSBUFC block is internally generated in the FPGA core. The READ signal goes high when the READ command to control the DDR-SDRAM is initially asserted. This precedes the DQS preamble by one cycle, yet may overlap the trailing bits of a prior read cycle. The DQS Detect circuitry of the LatticeECP2 device requires the falling edge of the READ signal to be placed within the preamble stage.

The preamble state of the DQS can be detected using the CAS latency and the round trip delay for the signals between the FPGA and the memory device. Note that the internal FPGA core generates the READ pulse. The rise of the READ pulse should coincide with the initial READ command of the Read Burst and need to go low before the Preamble goes high.

Refer to the following technical notes on the Lattice web site for more details:

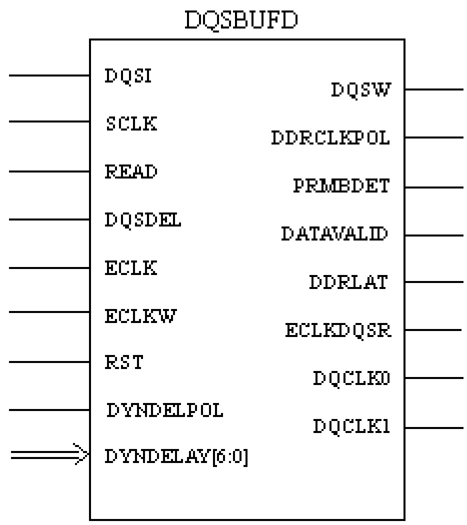
- TN1105 - LatticeECP2/M High-Speed I/O Interface
- TN1138 - LatticeXP2 High-Speed I/O Interface

DQSBUFD

DDR DQS Buffer Used for DDR3_MEM and DDR3_MEMGEN

Architectures Supported:

- LatticeECP3



INPUTS: DQSI, SCLK, READ, DQSDEL, ECLK, ECLKW, RST, DYNDELPOL, DYNDELAY6, DYNDELAY5, DYNDELAY4, DYNDELAY3, DYNDELAY2, DYNDELAY1, DYNDELAY0

OUTPUTS: DQSW, DDRCLKPOL, PRMBDET, DATAVALID, DDRLAT, ECLKDQSR, DQCLK0, DQCLK1

ATTRIBUTES:

DYNDEL_TYPE: "NORMAL" (default), "SHIFTED"

DYNDEL_VAL: integers 0~63 (PLUS), 1~64 (MINUS) (default: 0)

DYNDEL_CNTL: "DYNAMIC" (default), "STATIC"

(*EA only*) NRZMODE: "DISABLED" (default), "ENABLED"

Description

DQSBUFD is the DDR DQS buffer used for DDR3_MEM (DDR3 memory mode) and DDR3_MEMGEN.

- E and EA: DDR3_MEM and DDR3_MEMGEN (left/right)
- EA: DDR3_MEMGEN (top) – only input side

The table below describes the I/O ports.

Signal	I/O	Description
DQSI	I	DQS input coming from the pad.
SCLK	I	System clock.
READ	I	READ signal generated from the FPGA core.
RST	I	Reset input.
DQSDEL	I	Delay input from DQSDLL.
ECLK	I	Edge clock.
ECLKW	I	Edge clock used for the DDR write side.
DYNDELAY[6:0]	I	From user logic to DLL ADW & write clock generation.
DYNDELPOL	I	From user logic to DLL ADW & write clock generation. Will change the polarity of the clock depending on the clock frequency.
DQSW	O	DQS write clock.
DDRCLKPOL	O	DDR clock polarity signal.
PRMBDET	O	The preamble detect signal generated from the DQS signal going to the CIB. DQSI biased to go high when DQSI is tri-state.
DATAVALID	O	Signal indicating the transmission of valid data to the FPGA core.
DDRLAT	O	DDR latch control to input logic. Used to guarantee IDDRX2 gearing by selectively enabling a D flip-flop in the data path.
ECLKDQSR	O	Delay DQS used to capture the data.
DQCLK0	O	One clock edge, at half the frequency of ECLK, used in output gearing, 90 degree out of phase from DQCLK1.
DQCLK1	O	One clock edge, at the frequency of SCLK, used in output gearing.

Refer to the following technical note on the Lattice web site for more details:

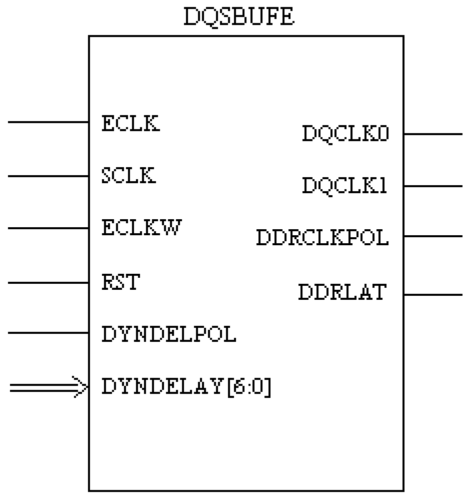
- TN1180 - LatticeECP3 High-Speed I/O Interface

DQSBUFE

DDR DQS Buffer Used for DDR_GENX2

Architectures Supported:

- LatticeECP3



INPUTS: ECLK, SCLK, ECLKW, RST, DYNDELPOL, DYNDELAY6, DYNDELAY5, DYNDELAY4, DYNDELAY3, DYNDELAY2, DYNDELAY1, DYNDELAY0

OUTPUTS: DQCLK0, DQCLK1, DDRCLKPOL, DDRLAT

ATTRIBUTES:

DYNDEL_TYPE: "NORMAL" (default), "SHIFTED"

DYNDEL_VAL: integers 0~63 (PLUS), 1~64 (MINUS) (default: 0)

DYNDEL_CNTL: "DYNAMIC" (default), "STATIC"

Description

DQSBUFE is the DDR DQS buffer used for DDR_GENX2 (DDR generic mode in X2 gearing).

- E: DDR_GENX2 (left/right/top)

The table below describes the I/O ports.

Signal	I/O	Description
SCLK	I	System clock.
ECLK	I	Edge clock.
ECLKW	I	Edge clock used for the DDR write side.
RST	I	Reset input.
DYNDELAY[6:0]	I	From user logic to DLL ADW & write clock generation.

Signal	I/O	Description
DYNDELPOL	I	From user logic to DLL ADW & write clock generation. Will change the polarity of the clock depending on the clock frequency.
DDRCLKPOL	O	DDR clock polarity signal.
DDRLAT	O	DDR latch control to input logic. Used to guarantee IDDRX2 gearing by selectively enabling a D flip-flop in the data path.
DQCLK0	O	One clock edge, at half the frequency of ECLK, used in output gearing, 90 degree out of phase from DQCLK1.
DQCLK1	O	One clock edge, at half the frequency of SCLK, used in output gearing.

Refer to the following technical note on the Lattice web site for more details:

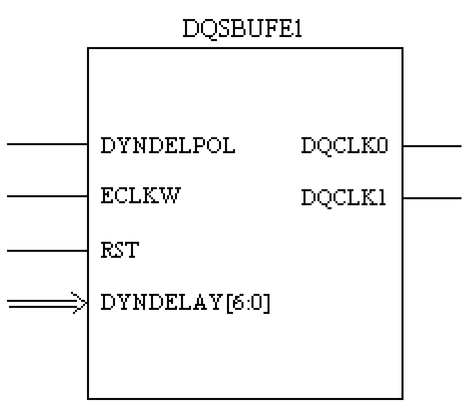
- TN1180 - LatticeECP3 High-Speed I/O Interface

DQSBUFE1

DDR DQS Buffer Used for DDR_GENX2

Architectures Supported:

- LatticeECP3



INPUTS: ECLKW, RST, DYNDELPOL, DYNDELAY6, DYNDELAY5, DYNDELAY4, DYNDELAY3, DYNDELAY2, DYNDELAY1, DYNDELAY0

OUTPUTS: DQCLK0, DQCLK1

ATTRIBUTES:

DYNDEL_TYPE: "NORMAL" (default), "SHIFTED"

DYNDEL_VAL: integers 0~63 (PLUS), 1~64 (MINUS) (default: 0)

DYNDEL_CNTL: "DYNAMIC" (default), "STATIC"

Description

DQSBUFE1 is the DDR DQS buffer used for DDR_GENX2 (DDR generic mode in X2 gearing).

- EA: DDR_GENX2 (left/right)
- EA: DDR_GENX2 (top)

The table below describes the I/O ports.

Signal	I/O	Description
ECLKW	I	Edge clock used for the DDR write side.
RST	I	Reset input.
DYNDELAY[6:0]	I	From user logic to DLL ADW & write clock generation.
DYNDELPOL	I	From user logic to DLL ADW & write clock generation. Will change the polarity of the clock depending on the clock frequency.
DQCLK0	O	One clock edge, at half the frequency of ECLK, used in output gearing, 90 degree out of phase from DQCLK1.
DQCLK1	O	One clock edge, at half the frequency of SCLK, used in output gearing.

Refer to the following technical note on the Lattice web site for more details:

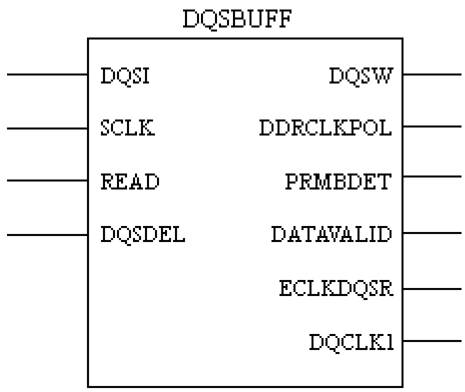
- TN1180 - LatticeECP3 High-Speed I/O Interface

DQSBUFF

DDR DQS Buffer Used for DDR_MEM, DDR2_MEM, and DDR2_MEMGEN

Architectures Supported:

- LatticeECP3



INPUTS: DQSI, SCLK, READ, DQSDEL

OUTPUTS: DQSW, DDRCLKPOL, PRMBDET, DATAVALID, ECLKDQSR, DQCLK1

Description

DQSBUFF is the DDR DQS buffer used for DDR_MEM (DDR memory mode), DDR2_MEM (DDR2 memory mode), and DDR2_MEMGEN.

- E and EA: DDR_MEM, DDR2_MEM, and DDR2_MEMGEN (left/right/top)

The table below describes the I/O ports.

Signal	I/O	Description
DQSI	I	DQS input coming from the pad.
SCLK	I	System clock.
READ	I	READ signal generated from the FPGA core.
DQSDEL	I	Delay input from DQSDLL.
DQSW	O	DQS write clock.
DDRCLKPOL	O	DDR clock polarity signal.
PRMBDET	O	The preamble detect signal generated from the DQS signal going to the CIB. DQSI biased to go high when DQSI is tri-state.
DATAVALID	O	Signal indicating the transmission of valid data to the FPGA core.
ECLKDQSR	O	Delay DQS used to capture the data.
DQCLK1	O	One clock edge, at the frequency of SCLK, used in output gearing.

Refer to the following technical note on the Lattice web site for more details:

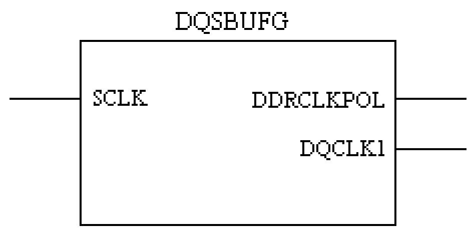
- TN1180 - LatticeECP3 High-Speed I/O Interface

DQSBUFG

DDR DQS Buffer Used for DDR_GENX1

Architectures Supported:

- LatticeECP3



INPUTS: SCLK

OUTPUT: DDRCLKPOL, DQCLK1

Description

DQSBUFG is the DDR DQS buffer used for DDR_GENX1 (DDR generic mode in X1 gearing).

- E: DDR_GENX1 (left/right/top)
- EA: Not allowed because it is not required

The table below describes the I/O ports.

Signal	I/O	Description
SCLK	I	System clock.
DDRCLKPOL	O	DDR clock polarity signal.
DQCLK1	O	One clock edge, at half the frequency of SCLK, used in output gearing.

Refer to the following technical note on the Lattice web site for more details:

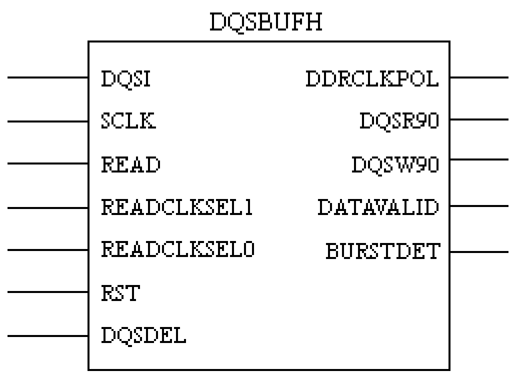
- TN1180 - LatticeECP3 High-Speed I/O Interface

DQSBUFH

DQS Circuit for DDR Memory

Architectures Supported:

- MachXO2
- Platform Manager 2



INPUTS: DQSI, SCLK, READ, READCLKSEL1, READCLKSEL0, RST, DQSDEL

OUTPUTS: DDRCLKPOL, DQSR90, DQSW90, DATAVALID, BURSTDET

ATTRIBUTES:

DQS_LI_DEL_ADJ: "PLUS" (default), "MINUS"

DQS_LI_DEL_VAL: integers 0~63 (PLUS), 1~64 (MINUS) (default: 0)

DQS_LO_DEL_ADJ: "PLUS" (default), "MINUS"

DQS_LO_DEL_VAL: integers 0~63 (PLUS), 1~64 (MINUS) (default: 0)

LPDDR: "DISABLED" (default), "ENABLED"

GSR: "ENABLED" (default), "DISABLED"

Description

DQSBUFH is the DQS circuit for DDR memory. It generates the 90 degree shift for DQS, and the DDRCLKPOL signal. It is used for right side only. The table below describes the I/O ports.

Signal	I/O	Description
DQSI	I	DQS signal from PIO.
READ	I	Signal for DDR read mode, coming from soft IP.
READCLKSEL1, READCLKSEL0	I	Select read clock source and polarity control for READ pulse position control in T/4 precision. The four positions are the rising/falling edges of SCLK or DQSW90. The signals come from soft IP.
SCLK	I	Clock from CIB.
RST	I	RESET for this block.
DQSDEL	I	DQS slave delay control from DQSDLLC.
DDRCLKPOL	O	SCLK polarity control.
DQSR90	O	DQS phase shifted by 90 degree output.
DQSW90	O	SCLK phase shifted by 90 degree output.
DATAVALID	O	Data valid signal for READ mode.
BURSTDET	O	Burst detection signal, moved from soft IP to hardware implementation.

Refer to the following technical note on the Lattice web site for more details:

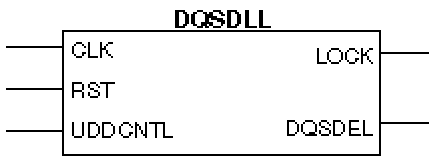
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

DQSDLL

DLL used as DDR memory DQS DLL

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeXP
- LatticeXP2



INPUTS: CLK, RST, UDDCNTL

OUTPUTS: LOCK, DQSDEL

ATTRIBUTES:

LOCK_SENSITIVITY: "LOW" (default), "HIGH"

Description

DQS delay calibration DLL. The primitive generates a 90-degree phase shift required for the DQS signal and implements the on-chip DQSDLL. Only one DQSDLL should be instantiated for all the DDR implementations on one half of the device. The clock input to this DLL should be at the same frequency as the DDR interface. DLL generates the delay based on this clock frequency and the update control input to this block. The DLL updates the dynamic delay control to the DQS delay block when this update control (UDDCNTL) input is asserted. The active low signal on UDDCNTL updates the DQS phase alignment and should be initiated at the beginning of READ cycles.

Port Name	I/O	Definition
CLK	I	System CLK should be at the frequency of the DDR interface from the FPGA core.
RST	I	Resets the DQSDLL
UDDCNTL	I	This is an active low port. It provides an update signal to the DLL that will update the dynamic delay. When held low, this signal will update the DQSDEL.
LOCK	O	Indicates when the DLL is in phase.
DQSDEL	O	The digital delay generated by the DLL, should be connected to the DQSBUF primitive.

DQSDLL Configuration: By default, this DLL generates a 90-degree phase shift for the DQS strobe based on the frequency of the input reference clock to the DLL. The user can control the sensitivity to jitter by using the LOCK_SENSITIVITY attribute. This configuration bit can be programmed to be either “HIGH” or “LOW”. The DLL Lock Detect circuit has two modes of operation controlled by the LOCK_SENSITIVITY bit, which selects more or less sensitivity to jitter. If this DLL is operated at or above 150 MHz, it is recommended that the LOCK_SENSITIVITY bit be programmed “HIGH” (more sensitive). When running at or below 100 MHz, it is recommended that the bit be programmed “LOW” (more tolerant). For 133 MHz, the LOCK_SENSITIVITY bit can go either way.

Refer to the following technical notes on the Lattice web site for more details

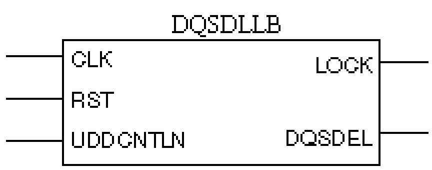
- TN1050 - LatticeECP/EC and LatticeXP DDR Usage Guide
- TN1105 - LatticeECP2/M High-Speed I/O Interface
- TN1138 - LatticeXP2 High-Speed I/O Interface

DQSDLLB

DQS DLL for DDR_MEM, DDR2_MEM, and DDR3_MEM

Architectures Supported:

- LatticeECP3



INPUTS: CLK, RST, UDDCNTLN

OUTPUTS: LOCK, DQSDEL

ATTRIBUTES:

LOCK_SENSITIVITY: "LOW" (default), "HIGH"

Description

DQSDLLB is the DLL used as DDR memory DQS DLL.

- E and EA: DDR_MEM, DDR2_MEM, and DDR3_MEM (left/right)

The table below describes the I/O ports.

Signal	I/O	Description
CLK	I	Clock from the CIB, coming from a PLL. The clock should run at the DDR memory frequency.
RST	I	RESET input to the master DLL.
UDDCNTLN	I	Update control generated from the core.

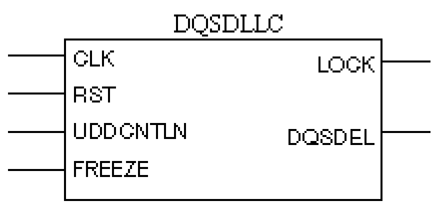
Signal	I/O	Description
DQSDEL	O	DQS delay generated by DQSDLL.
LOCK	O	Lock output of the DQSDLL to the CIB.

DQSDLLC

Master DLL for Generating Required Delay

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CLK, RST, UDDCNTLN, FREEZE

OUTPUTS: LOCK, DQSDEL

ATTRIBUTES:

DEL_ADJ: "PLUS" (default), MINUS

DEL_VAL: integers 0~127 (PLUS), 1~128 (MINUS) (default: 0)

LOCK_SENSITIVITY: "LOW" (default), "HIGH"

FIN: value range supported by DLL (default: "100.0")

FORCE_MAX_DELAY: "NO" (default), "YES"

GSR: "ENABLED" (default), "DISABLED"

Description

DQSDLLC is the master DLL to generate required delay. See the table below for I/O port descriptions.

Signal	I/O	Description
CLK	I	Clock from the CIB.
RST	I	DLL reset control.
UDDCNTLN	I	Hold/update control to delay code before adjustment.
FREEZE	I	Signal used to freeze or release DLL input CLK.
DQSDEL	O	DLL delay control code to slave delay cells, connected to DQSDEL of the DQSBUFH element.
LOCK	O	DLL lock signal.

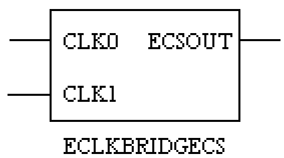
E

ECLKBRIDGECS

ECLK Bridge Block Clock Select

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CLK0, CLKI, SEL

OUTPUT: ECSOUT

Description

ECLK high speed bridge eases the design of high speed video or DDRX4 mode applications. It takes the high speed edge clocks sources from both sides (top and bottom). The bridge enhances the communication of ECLKs across the die. The bridge is supported for devices equal to or above 1200 LUTs.

The table below describes the I/O ports of the ECLKBRIDGECS primitive.

Port	I/O	Unused Port	Function
CLK0	I	Tie low	Input clock to the edge bridge clock select.
CLK1	I	Tie low	Input clock to the edge bridge clock select.
SEL	I	Tie low	From CIB, edge bridge clock select.
ECSOUT	O	Dangle	Output from edge bridge clock select.

ECLKBRIDGECS Usage with VHDL

Library Instantiation

```
library lattice;
```

```
use lattice.components.all;
```

Component Declaration

```
component ECLKBRIDGECS
   port (CLK0   : in std_logic;
         CLK1   : in std_logic;
         SEL    : in std_logic;
         ECSOUT : out std_logic);
end component;
```

ECLKBRIDGECS Instantiation

```
I1: ECLKBRIDGECS
   port map (CLK0   => CLK0;
             CLK1   => CLK1;
             SEL    => SEL;
             ECSOUT => ECSOUT);
end component;
```

ECLKBRIDGECS Usage with Verilog HDL

Component Declaration

```
module ECLKBRIDGECS (CLK0,
                     CLK1,
                     SEL,
                     ECSOUT);
input CLK0;
input CLK1;
input SEL;
output ECSOUT;
endmodule
```

ECLKBRIDGECS Instantiation

```
ECLKBRIDGECS I1 (.CLK0 (CLK0),
                 .CLK1 (CLK1),
                 .SEL   (SEL),
                 .ECSOUT (ECSOUT));
```

For more information and usage, refer to the following technical note on the Lattice web site.

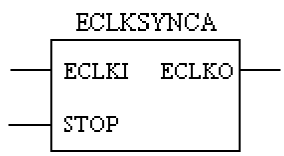
- TN1199 - MachXO2 sysCLOCK PLL Design and Usage Guide

ECLKSYNCA

ECLK Stop Block

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: ECLKI, STOP

OUTPUT: ECLKO

Description

ECLKSYNCA is the optional ECLK synchronization for DDR_MEM (DDR memory mode), DDR2_MEM (DDR2 memory mode), and DDR3_MEM (DDR3 memory mode).

- E and EA: DDR_MEM, DDR2_MEM, and DDR3_MEM (left/right/top)

The table below describes the I/O ports of the ECLKSYNCA primitive.

Port	I/O	Unused Port	Function
ECLKI	I	Tie low	Edge clock input to the stop clock
STOP	I	Tie low	Control signal to stop the edge clock to synchronize the signals derived from ECLK
ECLKO	O	Dangle	Edge clock output from the stop clock

ECLKSYNCA Usage with VHDL

Library Instantiation

```
library lattice;
use lattice.components.all;
```

Component Declaration

```
component ECLKSYNCA
   port (ECLKI : in std_logic;
```

```
         STOP  : in std_logic;
         ECLKO : out std_logic);
end component;
```

ECLKSYNCA Instantiation

```
I1: ECLKSYNCA
   port map (ECLKI => ECLKI,
             STOP => STOP,
             ECLKO => ECLKO);
end component;
```

ECLKSYNCA Usage with Verilog HDL

Component Declaration

```
module ECLKSYNCA (ECLKI,  STOP, ECLKO);
input ECLKI;
input STOP;
output ECLKO;
endmodule
```

ECLKSYNCA Instantiation

```
ECLKSYNCA I1 (.ECLKI (ECLKI);
              .STOP (STOP);
              .ECLKO (ECLKO));
```

For more information and usage, refer to the following technical notes on the Lattice web site.

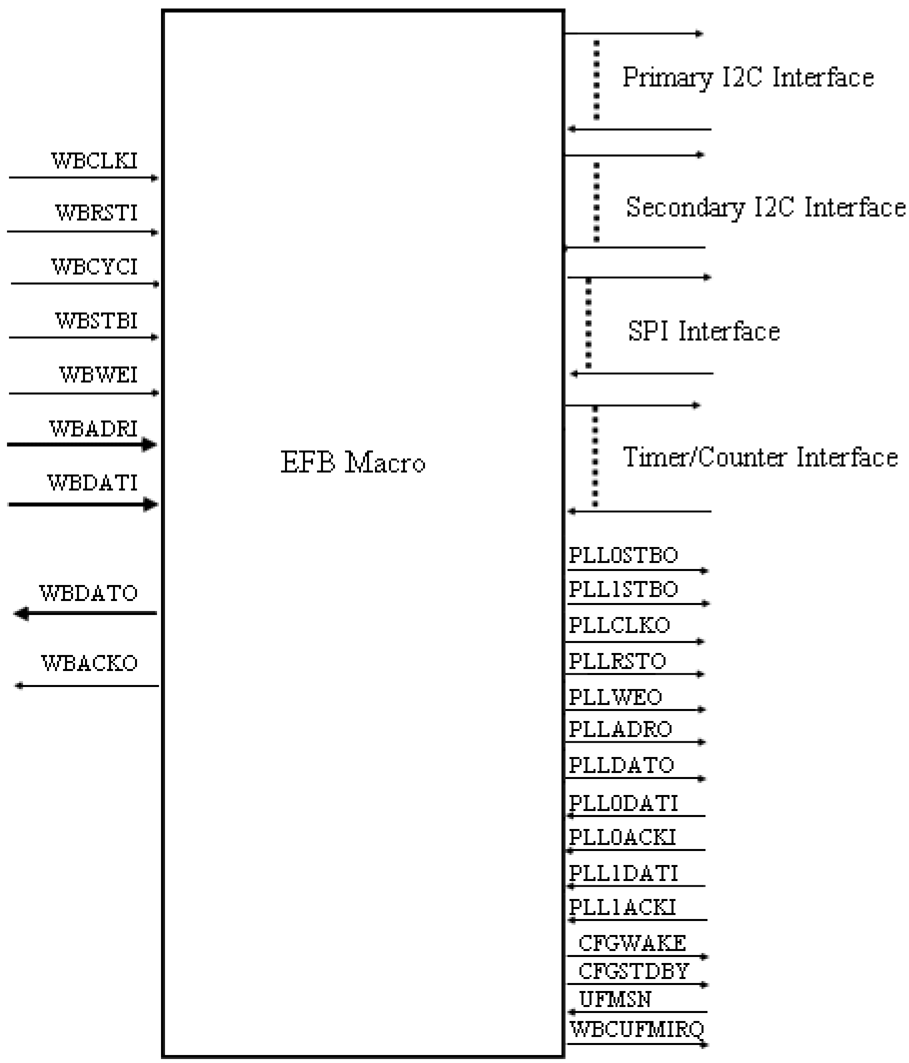
- TN1199 - MachXO2 sysCLOCK PLL Design and Usage Guide
- TN1177 - LatticeECP3 sysIO Usage Guide

EFB

Embedded Function Block

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: WBCLKI, WBRSTI, WBCYCI, WBSTBI, WBWEI, WBADRI7, WBADRI6, WBADRI5, WBADRI4, WBADRI3, WBADRI2, WBADRI1, WBADRI0, WBDATI7, WBDATI6, WBDATI5, WBDATI4, WBDATI3, WBDATI2, WBDATI1, WBDATI0, PLL0DATI7, PLL0DATI6, PLL0DATI5, PLL0DATI4, PLL0DATI3, PLL0DATI2, PLL0DATI1, PLL0DATI0, PLL0ACKI, PLL1DATI7, PLL1DATI6, PLL1DATI5, PLL1DATI4, PLL1DATI3, PLL1DATI2, PLL1DATI1, PLL1DATI0, PLL1ACKI, I2C1SCLI, I2C1SDAI, I2C2SCLI, I2C2SDAI, SPISCKI, SPIMISOI, SPIMOSII, SPISCSN, TCCLKI, TCRSTN, TCIC, UFMSN

OUTPUTS: WBDATO7, WBDATO6, WBDATO5, WBDATO4, WBDATO3, WBDATO2, WBDATO1, WBDATO0, WBACKO, PLLCLKO, PLLRSTO, PLL0STBO, PLL1STBO, PLLWEO, PLLADRO4, PLLADRO3, PLLADRO2, PLLADRO1, PLLADRO0, PLLDATO7, PLLDATO6, PLLDATO5, PLLDATO4,

PLLDATO3, PLLDATO2, PLLDATO1, PLLDATO0, I2C1SCLO, I2C1SCLOEN, I2C1SDAO, I2C1SDAOEN, I2C2SCLO, I2C2SCLOEN, I2C2SDAO, I2C2SDAOEN, I2C1IRQO, I2C2IRQO, SPISCKO, SPISCKEN, SPIMISOO, SPIMISOEN, SPIMOSIO, SPIMOSIEN, SPIMCSN0, SPIMCSN1, SPIMCSN2, SPIMCSN3, SPIMCSN4, SPIMCSN5, SPIMCSN6, SPIMCSN7, SPICSNEN, SPIIRQO, TCINT, TCOC, WBCUFMIRQ, CFGWAKE, CFGSTDBY

Description

The EFB primitive has seven explicit interfaces: WISHBONE, SPI, I2C (Primary), I2C (Secondary), Timer/Counter, and two PLLs.

For detailed information regarding the GUI, interface, and usage regarding each EFB interface, refer to the following technical note on the Lattice web site:

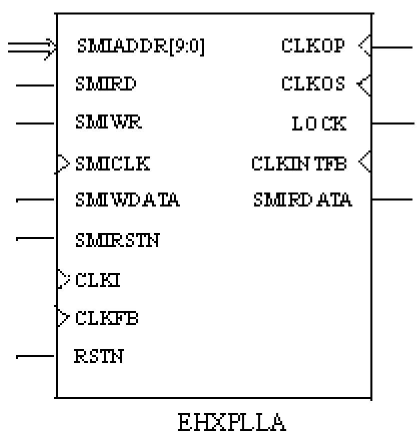
- TN1205 - Using User Flash Memory and Hardened Control Functions in MachXO2 Devices

EHXPLLA

Enhanced High Performance with Dynamic Input Delay Control PLL

Architectures Supported:

- LatticeSC/M



INPUTS: SMIADDR9, SMIADDR8, SMIADDR7, SMIADDR6, SMIADDR5, SMIADDR4, SMIADDR3, SMIADDR2, SMIADDR1, SMIADDR0, SMIRD, SMIWR, SMICLK, SMIWDATA, SMIRSTN, CLKI, CLKFB, RSTN

OUTPUTS: CLKOP, CLKOS, LOCK, CLKINTFB, SMIRDATA

ATTRIBUTES:

CLKI_DIV: integers 1~64 (default: 1)

CLKFB_DIV: integers 1~64 (default: 1)

CLKOP_DIV: integers 1~64 (default: 1)

CLKOS_DIV: integers 1~64 (default: 1)

CLKI_FDEL: 0 (default), 100, 200, ..., 700

CLKFB_FDEL: 0 (default), 100, 200, ..., 700

CLKOS_FDEL: 0 (default), 100, 200, ..., 700

CLKOP_MODE: "BYPASS" (default), "FDEL0", "VCO", "DIV"

CLKOS_MODE: "BYPASS" (default), "FDEL", "VCO", "DIV"

PHASEADJ: 0 (default), 45, 90, 135, 190, 225, 270, 315

GSR: "ENABLED" (default), "DISABLED"

SMI_OFFSET: 0x400~0x7FF (default: 12’h410)

LOCK_DELAY: integers 0~1000 (in ns) (default: 100)

CLKOS_VCODEL: integers 0~31 (default: 0)

MODULE_TYPE: "EHXPLLA"

IP_TYPE: "EHXPLLA"

Description

The Enhanced Extended Performance PLL (EHXPLLA) includes all features available in the PLL. This primitive includes SMI access so that you may configure the PLL as you require. The EHXPLLA primitive can be created through IPexpress. Note that Some combination of legal values are not allowed, due to other system limitations, such as the frequency of operation.

The following are descriptions of EHXPLLA port functions.

Port	I/O	Function
CLKI	I	CLKI[1:3]: from CIBs CLKI[0]: dedicated clock input pin Frequency: 2~1000 MHz
CLKFB	I	CLKFB[2,3]: from CIBs CLKFB[1]: dedicated external feedback pin CLKFB[0]: internal feedback from VCO output (CLKINTFB) Frequency: 2~1000 MHz

Port	I/O	Function
CLKOP	O	PLL output clock – main clock output Frequency: 1.5625~1000 MHz
CLKOS	O	PLL output clock – supplemental clock output Frequency: 1.5625~1000 MHz
LOCK	O	PLL locked to CLK1
CLKINTFB	O	CLKFB internal feedback source from VCO output
RSTN	I	Active low reset
SMIADDR[9:0]	I	SMI address bus
SMICLK	I	SMI clock signal
SMIRSTN	I	SMI reset signal
SMIRD	I	SMI read signal
SMIWDATA	I	SMI write data input
SMIWR	I	SMI write signal
SMIRDATA	O	SMI read data output

You can refer to the following technical note on the Lattice web site for more detailed description and usage.

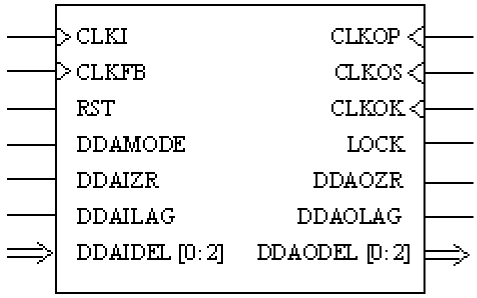
- TN1098 - LatticeSC sysCLOCK PLL/DLL User's Guide

EHXPLLB

Enhanced High Performance with Dynamic Input Delay Control PLL

Architectures Supported:

- LatticeECP/EC
- LatticeXP



EHXPLLB

INPUTS: CLKI, CLKFB, RST, DDAMODE, DDAIZR, DDAILAG, DDAIDEL0, DDAIDEL1, DDAIDEL2

OUTPUTS: CLKOP, CLKOS, CLKOK, LOCK, DDAOZR, DDAOLAG, DDAODEL0, DDAODEL1, DDAODEL2

ATTRIBUTES:

FIN: 20.0000~420.0000 (in MHz) (default: "100.0000")

CLKI_DIV: (*LatticeECP/EC*) integers 1~16 (default: 1); (*LatticeXP*) integers 1~15 (default: 1)

CLKFB_DIV: (*LatticeECP/EC*) integers 1~16 (default: 1); (*LatticeXP*) integers 1~15 (default: 1)

CLKOP_DIV: (*LatticeECP/EC*) even integers 2~32 if CLKOS is not used; 2, 4, 8, 16, 32 is CLKOS is used. (Default: 8)
(*LatticeXP*) even integers 2~30 if CLKOS is not used (default: 6); 2, 4, 8, 16 if CLKOS is used (default: 4).

CLKOK_DIV: 2 (default), 4, 6, 8, ..., 126, 128

FDEL: integers -8~8 (default: 0)

PHASEADJ: 0 (default), 45, 90, 135, 190, 225, 270, 315

DUTY: integers 1~7 (default: 4)

DELAY_CNTL: "STATIC" (default), "DYNAMIC"

Description

The following are descriptions of EHXPLLB port functions.

Port	I/O	Function
CLKI	I	Global clock input; frequency: 20~420 MHz.
CLKFB	I	External feedback, internal feedback from CLKOP divider; frequency: 20~420 MHz.
RST	I	"1" to reset M-divider.
DDAMODE	I	DDA mode. "1": pin control (dynamic); "0": fuse control (static).
DDAIZR	I	DDA delay zero. "1": delay = 0; "0": delay = on.
DDAILAG	I	DDA lag/lead. "1": lag; "0": lead.
DDAIDEL[0:2]	I	DDA delay.
CLKOP	O	PLL output clock to clock tree (no phase shift); frequency: 20~420 MHz.
CLKOS	O	PLL output clock to clock tree (phase shifted/duty cycle changed); frequency: 20~420 MHz.
CLKOK	O	PLL output to clock tree (K divider, low speed, output); frequency: 0.156~210 MHz.
LOCK	O	"1" indicates PLL LOCK to CLK_IN.
DDAOZR	O	DDA delay zero output.
DDAOLAG	O	DDA lag/lead output.
DDAODEL[0:2]	O	DDA delay output.

Dynamic Delay Adjustment

The Dynamic Delay Adjustment is controlled by DDAMODE input. This feature is available in EHXPLLB primitive only. When the DDAMODE input is set to “1,” the delay control is done through the inputs, DDAIZR, DDAILAG and DDAIDEL(2:0). For this mode, the attribute “DELAY_CNTL” must be set to “DYNAMIC.”

Equations for Generating Input and Output Frequency Ranges

These values of fIN, fOUT, fVCO are the absolute frequency ranges for the PLL. The values of fINMIN, fINMAX, fOUTMIN, and fOUTMAX, are the calculated frequency ranges based on the divider settings. These calculated frequency ranges become the limits for the specific divider settings used in the design.

fOUT = fIN * (N/M)

fVCO = fOUT * V = fIN * (N/M) * V

fIN = (fVCO /(V*N))*M

Where M = CLKI DIV

N = CLKFB DIV

V = CLKOP DIV

K = CLKOK DIV

fINMIN = ((fVCOMIN /(V*N))*M, if below 33 * M round up to 33 * M

fINMAX = (fVCOMAX/(V*N))*M, if above 420 round down to 420

fOUTMIN = fINMIN*(N/M), if below 33 * N round up to 33 * N

fOUTMAX = fINMAX*(N/M), if above 420 round down to 420

fOUTKMIN = fOUTMIN /K

fOUTKMAX = fOUTMAX /K

You can refer to the following technical note on the Lattice web site for more detailed description and usage.

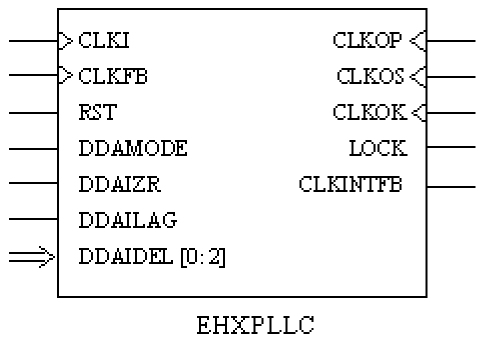
- TN1049 - LatticeECP/EC and LatticeXP sysCLOCK PLL Design and Usage Guide

EHXPLLC

Enhanced Extended Performance PLL

Architectures Supported:

- MachXO
- Platform Manager



INPUTS: CLKI, CLKFB, RST, DDAMODE, DDAIZR, DDAILAG, DDAIDEL0, DDAIDEL1, DDAIDEL2

OUTPUTS: CLKOP, CLKOS, CLKOK, LOCK, CLKINTFB

ATTRIBUTES:

FIN: 20.0000~420.0000 (in MHz) (default: "100.0000")

CLKFB_DIV: integers 1~16 (default: 1)

CLKI_DIV: integers 1~16 (default: 1)

CLKOP_DIV: even integers 2~32 if CLKOS is not used; 2, 4, 8, 16, 32 if CLKOS is used. (Default: 8)

CLKOK_DIV: 2 (default), 4, 6, 8, ..., 126, 128

DELAY_CNTL: "STATIC" (default), "DYNAMIC"

FDEL: integers -8~8 (default: 0)

PHASEADJ: 0 (default), 45, 90, 135, 190, 225, 270, 315

DUTY: integers 1~7 (default: 4)

Description

The EHXPLLC primitive is used for MachXO and Platform Manager PLL implementation. The definitions of the PLL I/O ports are shown in the following table. The EHXPLLC includes all features available in the MachXO or Platform Manager PLL.

Port	I/O	Function
CLKI	I	General routing or dedicated global clock input pad.
CLKFB	I	From general routing, clock tree, internal feedback from CLKOP or dedicated external feedback CLKFB Ipad.
RST	I	"1" to reset PLL counters.
CLKOP	O	PLL output clock to clock tree (no phase shift).
CLKOS	O	PLL output clock to clock tree (phase shifted/duty cycle changed).
CLKOK	O	PLL output to clock tree (CLKOK divider, low speed, output).
LOCK	O	"1" indicates PLL LOCK to CLKI; asynchronous signal.
CLKINTFB	O	Internal feedback source. CLKOP divider output before CLOCK TREE.
DDAMODE	I	DDA mode. "1": pin control (dynamic); "0": fuse control (static).
DDAIZR	I	DDA delay zero. "1": delay = 0; "0": delay = on.

Port	I/O	Function
DDAILAG	I	DDA lag/lead. "1": lead; "0": lag.
DDAIDEL[0:2]	I	DDA delay.

You can refer to the following technical note on the Lattice web site for more detailed description and usage.

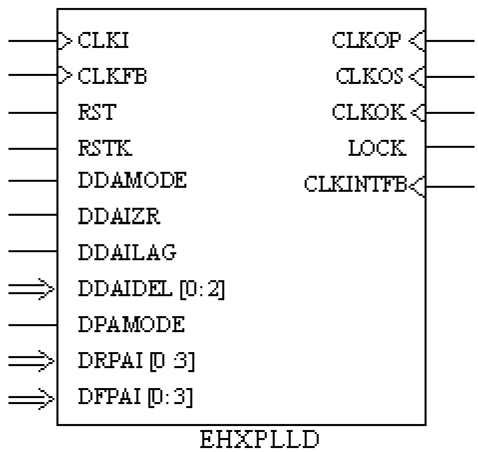
- TN1089 - MachXO sysCLOCK Design and Usage Guide

EHXPLLD

Complex PLL

Architectures Supported:

- LatticeECP2/M



INPUTS: CLKI, CLKFB, RST, RSTK, DDAMODE, DDAIZR, DDAILAG, DDAIDEL0, DDAIDEL1, DDAIDEL2, DPAMODE, DRPAI3, DRPAI2, DRPAI1, DRPAI0, DFPAI3, DFPAI2, DFPAI1, DFPAI0

OUTPUTS: CLKOP, CLKOK, CLKOS, LOCK, CLKINTFB

ATTRIBUTES:

FIN: 20.0000~420.0000 (in MHz) (default: "100.0000")

CLKI_DIV: integers 1~64 (default: 1)

CLKFB_DIV: integers 1~16 (default: 1)

CLKOP_DIV: 2, 4, 8 (default), 16, 32, 48, 64, 80, 96, 112, 128

CLKOK_DIV: 2 (default), 4, 6, 8, ..., 126, 128

FDEL: integers -8~8 (default: 0)

PHASEADJ: 0 (default), 22.5, 45, 67.5, 90, ..., 315, 337.5

DUTY: integers 2~14 (default: 8)

PHASE_CNTL: "STATIC" (default), "DYNAMIC"

DELAY_CNTL: "STATIC" (default), "DYNAMIC"

PLLCAP: "DISABLED" (default), "ENABLED", "AUTO"

CLKOP_BYPASS: "DISABLED" (default), "ENABLED"

CLKOS_BYPASS: "DISABLED" (default), "ENABLED"

CLKOK_BYPASS: "DISABLED" (default), "ENABLED"

Description

The ECP2 devices provide two type of PLLS: SPLL and GPLL. The SPLL is a baseline PLL. The GPLL includes all features of SPLL plus Dynamic Delay Adjustment. The primitive for the SPLL is EPLLD. The primitive for the GPLL are EPLLD and EHXPLLD. See the following table for GPLL and SPLL IO port description.

Port	I/O	Function
CLKI	I	Input clock.
CLKFB	I	Feedback clock.
RST	I	PLL reset (connect to CNTRST port). High active reset.
RSTK	I	Reset for K divider (connect to RESETK port). High active reset.
CLKOP	O	PLL output clock (no phase shift).
CLKOS	O	PLL output clock (phase shifted/duty cycle changed).
CLKOK	O	PLL output to clock tree (no phase shift, low speed).
LOCK	O	PLL LOCK to CLKI, asynchronous signal. Active high indicates PLL lock.
CLKINTFB	O	Internal feedback source. CLKOP divider output before CLOCK TREE.
DDAMODE	I	DDA mode. Active high indicates pin control (DYNAMIC) and active low indicates fuse control (STATIC).
DDAIZR	I	DDA delay zero. Active high indicates “delay = 0” and active low indicates “delay= on.”

Port	I/O	Function
DDAILAG	I	DDA lag/lead. Active high indicates “lead” and active low indicates “lag.”
DDAIDEL[2:0]	I	DDA delay.
DPAMODE	I	Dynamic phase adjust mode. Active high indicates pin control (DYNAMIC) and active low indicates fuse control (STATIC).
DRPAI[3:0]	I	Dynamic coarse phase shift, rising edge setting.
DFPAI[3:0]	I	Dynamic coarse phase shift, falling edge setting.

You can refer to the following technical note on the Lattice web site for more detailed description and usage.

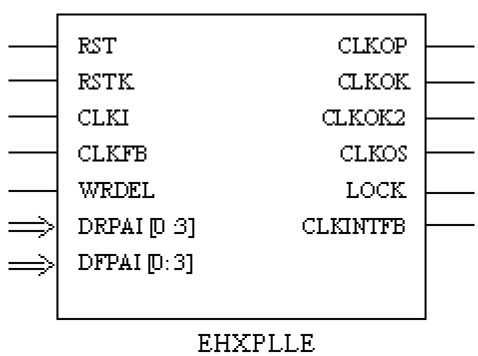
- TN1103 - LatticeECP2/M sysCLOCK PLL/DLL Design and Usage Guide

EHXPLLE

Complex PLL

Architectures Supported:

- LatticeXP2



INPUTS: CLKI, CLKFB, RST, RSTK, WRDEL, DRPAI3, DRPAI2, DRPAI1, DRPAI0, DFPAI3, DFPAI2, DFPAI1, DFPAI0

OUTPUTS: CLKOP, CLKOK, CLKOK2, CLKOS, LOCK, CLKINTFB

ATTRIBUTES:

FIN: 20.0000~420.0000 (in MHz) (default: "100.0000")

CLKFB_DIV: integers 1~64 (default: 1)

CLKI_DIV: integers 1~64 (default: 1)

CLKOP_DIV: 2, 4, 8 (default), 16, 32, 48, 64, 80, 96, 112, 128

CLKOK_DIV: 2 (default), 4, 6, ..., 126, 128

PHASE_CNTL: "STATIC" (default), "DYNAMIC"

PHASEADJ: 0 (default), 22.5, 45, 67.5, 90, ..., 315, 337.5

DUTY: integers 2~14 (default: 8)

CLKOP_BYPASS: "DISABLED" (default), "ENABLED"

CLKOS_BYPASS: "DISABLED" (default), "ENABLED"

CLKOK_BYPASS: "DISABLED" (default), "ENABLED"

CLKOP_TRIM_POL: "FALLING" (default), "RISING"

CLKOP_TRIM_DELAY: integers 0~7 (default: 0)

CLKOS_TRIM_POL: "RISING" (default), "FALLING"

CLKOS_TRIM_DELAY: integers 0~3 (default: 0)

Description

EHXPLLE and EPLLD are the two primitives defined for XP2 GPLL. The EPLLD is used for both ECP2 and XP2 to support design migration. See EPLLD for details on migration. It is recommended to use EPLLD for all PLL configurations except for configurations involving Duty Trim Options, CLKOK2 and the CLKOS Fine Delay Port (WRDEL). Those features are only supported by the EHXPLLE primitive.

See the following table for port description.

Port	I/O	Function
CLKI	I	Input clock.
CLKFB	I	Feedback clock.
RST	I	PLL reset (connect to CNTRST port). High active reset.
RSTK	I	Reset for K divider (connect to RESETK port). High active reset.
CLKOP	O	PLL output clock (no phase shift).
CLKOS	O	PLL output clock (phase shifted/duty cycle changed).
CLKOK	O	PLL output to clock tree (no phase shift, low speed).
CLKOK2	O	PLL output clock (no phase shift, CLKOP/3).

Port	I/O	Function
LOCK	O	PLL LOCK to CLKI, asynchronous signal. Active high indicates PLL lock.
CLKINTFB	O	Internal feedback source. CLKOP divider output before CLOCK TREE.
WRDEL	I	Fine delay adjust (0 = no delay; 1 = ~70ps).
DRPAI[3:0]	I	Dynamic coarse phase shift, rising edge setting.
DFPAI[3:0]	I	Dynamic coarse phase shift, falling edge setting.
DDAMODE	I	DDA mode. Active high indicates pin control (DYNAMIC) and active low indicates fuse control (STATIC).

You can refer to the following technical note on the Lattice web site for more detailed description and usage.

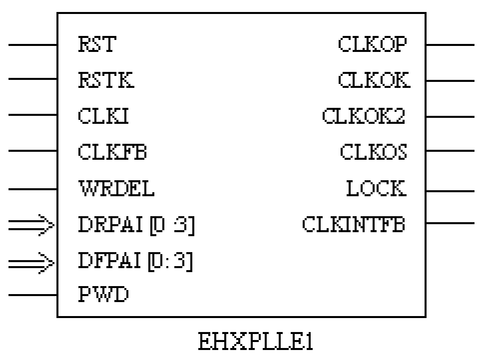
- TN1126 - LatticeXP2 sysCLOCK PLL Design and Usage Guide

EHXPLLE1

Complex PLL

Architectures Supported:

- LatticeXP2



INPUTS: CLKI, CLKFB, RST, RSTK, WRDEL, DRPAI3, DRPAI2, DRPAI1, DRPAI0, DFPAI3, DFPAI2, DFPAI1, DFPAI0, PWD

OUTPUTS: CLKOP, CLKOK, CLKOK2, CLKOS, LOCK, CLKINTFB

ATTRIBUTES:

FIN: 20.0000~420.0000 (in MHz) (default: "100.0000")

CLKFB_DIV: integers 1~64 (default: 1)

CLKI_DIV: integers 1~64 (default: 1)

CLKOP_DIV: 2, 4, 8 (default), 16, 32, 48, 64, 80, 96, 112, 128

CLKOK_DIV: 2 (default), 4, 6, ..., 126, 128

PHASE_CNTL: "STATIC" (default), "DYNAMIC"

PHASEADJ: 0 (default), 22.5, 45, 67.5, 90, ..., 315, 337.5

DUTY: integers 2~14 (default: 8)

CLKOP_BYPASS: "DISABLED" (default), "ENABLED"

CLKOS_BYPASS: "DISABLED" (default), "ENABLED"

CLKOK_BYPASS: "DISABLED" (default), "ENABLED"

CLKOP_TRIM_POL: "FALLING" (default), "RISING"

CLKOP_TRIM_DELAY: integers 0~7 (default: 0)

CLKOS_TRIM_POL: "RISING" (default), "FALLING"

CLKOS_TRIM_DELAY: integers 0~3 (default: 0)

Description

The following are descriptions of EHXPLLE1 port functions.

Port	I/O	Function
CLKI	I	Input clock.
CLKFB	I	Feedback clock.
RST	I	PLL reset (connect to CNTRST port). High active reset.
RSTK	I	Reset for K divider (connect to RESETK port). High active reset.
WRDEL	I	Fine delay adjust (0 = no delay; 1 = ~70ps).
DRPAI[3:0]	I	Dynamic coarse phase shift, rising edge setting.
DFPAI[3:0]	I	Dynamic coarse phase shift, falling edge setting.
DDAMODE	I	DDA mode. Active high indicates pin control (DYNAMIC) and active low indicates fuse control (STATIC).
PWD	I	Dynamic power down signal.
CLKOP	O	PLL output clock (no phase shift).
CLKOS	O	PLL output clock (phase shifted/duty cycle changed).

Port	I/O	Function
CLKOK	O	PLL output to clock tree (no phase shift, low speed).
CLKOK2	O	PLL output clock (no phase shift, CLKOP/3).
LOCK	O	PLL LOCK to CLKI, asynchronous signal. Active high indicates PLL lock.
CLKINTFB	O	Internal feedback source. CLKOP divider output before CLOCK TREE.

You can refer to the following technical note on the Lattice web site for more detailed description and usage.

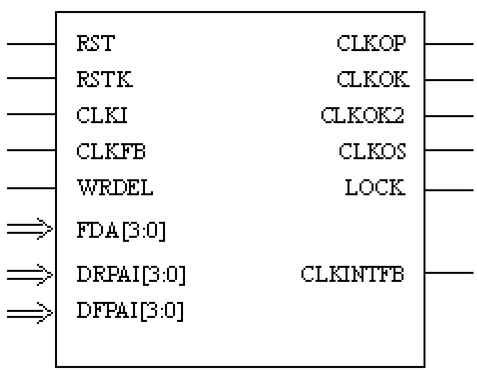
- TN1126 - LatticeXP2 sysCLOCK PLL Design and Usage Guide

EHXPLLF

Complex PLL

Architectures Supported:

- LatticeECP3



EHXPLLF

INPUTS: CLKI, CLKFB, RST, RSTK, DRPAI3, DRPAI2, DRPAI1, DRPAI0, DFPAI3, DFPAI2, DFPAI1, DFPAI0, FDA3, FDA2, FDA1, FDA0, WRDEL

OUTPUTS: CLKOP, CLKOS, CLKOK, CLKOK2, LOCK, CLKINTFB

ATTRIBUTES:

FIN: 20.0000~420.0000 (in MHz) (default: "100.0000")

CLKI_DIV: integers 1~64 (default: 1)

CLKFB_DIV: integers 1~64 (default: 1)

CLKOP_DIV: 2, 4, 8 (default), 16, 32, 48, 64, 80, 96,112, 128

CLKOK_DIV: 2 (default), 4, 6, 8, ...,126,128

PHASEADJ: 0 (default), 22.5, 45, 67.5, 90, ..., 315, 337.5

DUTY: integers 2~14 (default: 8)

PHASE_DELAY_CNTL: "STATIC" (default), "DYNAMIC"

CLKOP_BYPASS: "DISABLED" (default), "ENABLED"

CLKOS_BYPASS: "DISABLED" (default), "ENABLED"

CLKOK_BYPASS: "DISABLED" (default), "ENABLED"

CLKOP_TRIM_POL: "RISING" (default), "FALLING"

CLKOP_TRIM_DELAY: integers 0~7 (default: 0)

CLKOS_TRIM_POL: "RISING" (default), "FALLING"

CLKOS_TRIM_DELAY: integers 0~3 (default: 0)

DELAY_VAL: integers 0~15 (default: 0)

DELAY_PWD: "DISABLED" (default), "ENABLED"

CLKOK_INPUT: "CLKOP" (default), "CLKOS"

Description

EHXPLLF and EPLLD are the two primitives defined for the LatticeECP3 GPLL. The EPLLD primitive is used for both ECP3 and XP2 for design migration. For the ECP3 new configurations, only EHXPLLF will be supported.

The following table describes EHXPLLF IO port functions.

Port	I/O	Function
CLKI	I	Input clock.
CLKFB	I	Feedback clock.
RST	I	PLL reset (connect to CNTRST port). High active reset.
RSTK	I	Reset for K divider (connect to RESETK port). High active reset.
CLKOP	O	PLL output clock (no phase shift).
CLKOS	O	PLL output clock (phase shifted/duty cycle changed).

Port	I/O	Function
CLKOK	O	PLL output clock (no phase shift, low speed).
CLKOK2	O	PLL output clock (no phase shift, CLKOP/3).
LOCK	O	PLL LOCK to CLKI, asynchronous signal. Active high indicates PLL lock.
CLKINTFB	O	Internal feedback source. CLKOP divider output before CLOCK TREE.
WRDEL	I	Dynamic CLKOS single step fine delay adjust (0 = no delay; 1 = ~70ps).
FDA[3:0]	I	Dynamic CLKOS 16 step fine delay adjustment on CLKOS (each increment is ~125ps).
DRPAI[3:0]	I	Dynamic coarse phase shift, rising edge setting.
DFPAI[3:0]	I	Dynamic duty cycle, falling edge setting.

You can refer to the following technical note on the Lattice web site for EHXPLLF port definition, attribute definition, and usage.

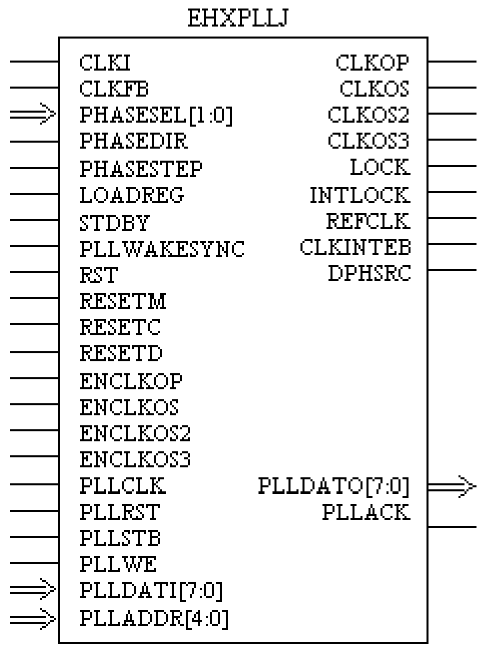
- TN1178 - LatticeECP3 sysCLOCK PLL/DLL Design and Usage Guide

EHXPLLJ

GPLL for MachXO2 and Platform Manager 2

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CLKI, CLKFB, PHASESEL1, PHASESEL0, PHASEDIR, PHASESTEP, LOADREG, STDBY, PLLWAKESYNC, RST, RESETM, RESETC, RESETD, ENCLKOP, ENCLKOS, ENCLKOS2, ENCLKOS3, PLLCLK, PLLRST, PLLSTB, PLLWE, PLLDATI7, PLLDATI6, PLLDATI5, PLLDATI4, PLLDATI3, PLLDATI2, PLLDATI1, PLLDATI0, PLLADDR4, PLLADDR3, PLLADDR2, PLLADDR1, PLLADDR0

OUTPUTS: CLKOP, CLKOS, CLKOS2, CLKOS3, LOCK, INTLOCK, REFCLK, CLKINTFB, DPHSRC, PLLDATO7, PLLDATO6, PLLDATO5, PLLDATO4, PLLDATO3, PLLDATO2, PLLDATO1, PLLDATO0, PLLACK

ATTRIBUTES:

The EHXPLLJ primitive utilizes many attributes that allow the configuration of the PLL through source constraints. The following table details these attributes:

Attribute	Type	Allowed Values	Default	Description
FREQ_PIN_CLKI	String	10 to 400	100	CLKI frequency (MHz)
FREQ_PIN_CLKOP	String	3.125 to 400	100	CLKOP frequency (MHz)
CLKOP_FTOL	String	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0	CLKOP frequency tolerance
CLKOP_AFREQ	String		-	CLKOP actual frequency (MHz)
FREQ_PIN_CLKOS	String	0.024 to 400	100	CLKOS frequency (MHz)

Attribute	Type	Allowed Values	Default	Description
CLKOS_FTOL	String	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0	CLKOS frequency tolerance
CLKOS_AFREQ	String		-	CLKOS actual frequency (MHz)
FREQUENCY_PIN_CLKOS2	String	0.024 to 400	100	CLKOS2 frequency (MHz)
CLKOS2_FTOL	String	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0	CLKOS2 frequency tolerance
CLKOS2_AFREQ	String		-	CLKOS2 actual frequency (MHz)
FREQUENCY_PIN_CLKOS3	String	0.024 to 400	100	CLKOS3 frequency (MHz)
CLKOS3_FTOL	String	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0	CLKOS3 frequency tolerance
CLKOS3_AFREQ	String		-	CLKOS3 actual frequency (MHz)
CLKI_DIV	Integer	1 to 128	1	CLKI divider setting
CLKFB_DIV	Integer	1 to 128	1	CLKFB divider setting
CLKOP_DIV	Integer	1 to 128	8	CLKOP divider setting
CLKOS_DIV	Integer	1 to 128	8	CLKOS divider setting
CLKOS2_ DIV	Integer	1 to 128	8	CLKOS2 divider setting
CLKOS3_ DIV	Integer	1 to 128	8	CLKOS3 divider setting
CLOCK_ENABLE_PORTS	Boolean	ENABLED, DISABLED	DISABLED	Clock enable ports
CLKOP_ENABLE	Boolean	ENABLED, DISABLED	ENABLED	CLKOP enable
CLKOS_ENABLE	Boolean	ENABLED, DISABLED	ENABLED	CLKOS enable
CLKOS2_ENABLE	Boolean	ENABLED, DISABLED	ENABLED	CLKOS2 enable
CLKOS3_ENABLE	Boolean	ENABLED, DISABLED	ENABLED	CLKOS3 enable
VCO_BYPASS_A0	Boolean	ENABLE, DISABLED	DISABLED	VCO bypass A0
VCO_BYPASS_B0	Boolean	ENABLE, DISABLED	DISABLED	VCO bypass B0
VCO_BYPASS_C0	Boolean	ENABLE, DISABLED	DISABLED	VCO bypass C0
VCO_BYPASS_D0	Boolean	ENABLE, DISABLED	DISABLED	VCO bypass D0

Attribute	Type	Allowed Values	Default	Description
CLKOP _PHASEADJ	String	0, 45, 90, 135, 180, 225, 270, 315	0	CLKOP desired phase shift selection (O) in static mode
CLKOS_PHASEADJ	String	0, 45, 90, 135, 180, 225, 270, 315	0	CLKOS desired phase shift selection (O) in static mode
CLKOS2_PHASEADJ	String	0, 45, 90, 135, 180, 225, 270, 315	0	CLKOS2 desired phase shift selection (O) in static mode
CLKOS3_PHASEADJ	String	0, 45, 90, 135, 180, 225, 270, 315	0	CLKOS3 desired phase shift selection (O) in static mode
CLKOP_CPHASE	Integer	0 to 127	N/A	CLKOP coarse phase adjust
CLKOS_CPHASE	Integer	0 to 127	N/A	CLKOS coarse phase adjust
CLKOS2_CPHASE	Integer	0 to 127	N/A	CLKOS2 coarse phase adjust
CLKOS3_CPHASE	Integer	0 to 127	N/A	CLKOS3 coarse phase adjust
CLKOP_FPHASE	Integer	0 to 7	N/A	CLKOP fine phase adjust
CLKOS_FPHASE	Integer	0 to 7	N/A	CLKOS fine phase adjust
CLKOS2_FPHASE	Integer	0 to 7	N/A	CLKOS2 fine phase adjust
CLKOS3_FPHASE	Integer	0 to 7	N/A	CLKOS3 fine phase adjust
FEEDBK_PATH	String	CLKOP, CLKOS, CLKOS2, CLKOS3, INT_DIVA, INT_DIVB, INT_DIVC, INT_DIVD, USERCLOCK	CLKOP	Feedback mode
KVCO	Integer	0 to 7	0	VCO gain - Kvco
LPF_CAPACITOR	Integer	0 to 3	0	LPF capacitor
LPF_RESISTOR	Integer	0 to 127	0	LPF resistor
ICP_CURRENT	Integer	0 to 31	0	ICP current
FRACN_ENABLE	Boolean	ENABLE, DISABLED	DISABLED	Fractional-N divider enable
FRACN_DIV	Integer	0 to 65535	0	Fractional-N divider
FRACN_ORDER	Integer	0 to 3	0	Fractional-N noise shaping order
CLKOP_TRIM_POL	String	RISING, FALLING	RISING	CLKOP duty trim polarity
CLKOP_TRIM_DELAY	Integer	0, 1, 2, 4	0	CLKOP duty trim polarity delay
CLKOS_TRIM_POL	String	RISING, FALLING	RISING	CLKOS duty trim polarity
CLKOS_TRIM_DELAY	Integer	0, 1, 2, 4	0	CLKOS duty trim polarity delay

Attribute	Type	Allowed Values	Default	Description
PLL_EXPERT	Boolean	ENABLE, DISABLED	DISABLED	
PLL_USE_WB	Boolean	ENABLE, DISABLED	DISABLED	
PREDIVIDER_MUXA1	Integer	0 to 3	0	
PREDIVIDER_MUXB1	Integer	0 to 3	0	
PREDIVIDER_MUXC1	Integer	0 to 3	0	
PREDIVIDER_MUXD1	Integer	0 to 3	0	
OUTDIVIDER_MUXA2	String	DIVA, REFCLK	DIVA	
OUTDIVIDER_MUXB2	String	DIVB, REFCLK	DIVB	
OUTDIVIDER_MUXC2	String	DIVC, REFCLK	DIVC	
OUTDIVIDER_MUXD2	String	DIVD, REFCLK	DIVD	
FREQ_LOCK_ACCURACY	Integer	0 to 3	0	
PLL_LOCK_MODE	Integer	0 to 7	0	
PLL_LOCK_DELAY	Integer	1600, 800, 400, 200 (in ns)	200	
GMC_GAIN	Integer	0 to 7	0	GM/C gain
GMC_TEST	Integer	0 to 15	14	GM/C test mode
MFG1_TEST	Integer	0 to 7	0	
MFG2_TEST	Integer	0 to 7	0	
MFG_FORCE_VFILTER	Integer	0, 1	0	
MFG_ICP_TEST	Integer	0, 1	0	
MFG_ EN_UP	Integer	0, 1	0	
MFG_ FLOAT_ICP	Integer	0, 1	0	
MFG_GMC_PRESET	Integer	0, 1	0	
MFG_LF_PRESET	Integer	0, 1	0	
MFG_GMC_RESET	Integer	0, 1	0	
MFG_LF_RESET	Integer	0, 1	0	
MFG_LF_RESGRND	Integer	0, 1	0	
MFG_ GMCREF_SEL	Integer	0 to 3	2	
MFG_EN_FILTEROPAMP	Integer	0, 1	1	
STDBY_ENABLE	Boolean	ENABLED, DISABLED	DISABLED	

Attribute	Type	Allowed Values	Default	Description
REFIN_RESET	Boolean	ENABLED, DISABLED	DISABLED	
SYNC_ENABLE	Boolean	ENABLED, DISABLED	DISABLED	
INT_LOCK_STICKY	Boolean	ENABLED, DISABLED	DISABLED	
DPHASE_SOURCE	Boolean	ENABLED, DISABLED	DISABLED	
INTFB_WAKE	Boolean	ENABLED, DISABLED	DISABLED	
PLLRST_ENA	Boolean	ENABLED, DISABLED	DISABLED	
MRST_ENA	Boolean	ENABLED, DISABLED	DISABLED	
DCRST_ENA	Boolean	ENABLED, DISABLED	DISABLED	
DDRST_ENA	Boolean	ENABLED, DISABLED	DISABLED	

Description

EHXPLLJ is the GPLL primitive for MachXO2 and Platform Manager 2. A wrapper will be used around the primitive for configurations without the dynamic control or other ports. The EHXPLLJ primitive uses a single reference clock input.

For detailed information, refer to the following technical note on the Lattice web site.

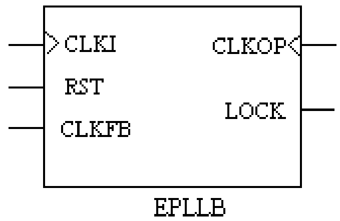
- TN1199 - MachXO2 sysCLOCK PLL Design and Usage Guide

EPLLB

Enhanced PLL

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: CLKI, RST, CLKFB

OUTPUTS: CLKOP, LOCK

ATTRIBUTES:

FIN: 20.0000~420.0000 (in MHz) (default: "100.0000")

CLKI_DIV: (*LatticeECP/EC*) integers 1~16 (default: 1);
(*LatticeXP*) integers 1~15 (default: 1)

CLKFB_DIV: (*LatticeECP/EC*) integers 1~16 (default: 1);
(*LatticeXP*) integers 1~15 (default: 1)

CLKOP_DIV: (*LatticeECP/EC*) even integers 2~32 if CLKOS is not used; 2, 4, 8, 16, 32 is CLKOS is used. (Default: 8)
(*LatticeXP*) even integers 2~30 if CLKOS is not used (default: 6); 2, 4, 8, 16 if CLKOS is used (default: 4).

FDEL: integers -8~8 (default: 0)

WAKE_ON_LOCK: "OFF" (default), "ON"

FB_MODE: "CLOCKTREE" (default), "INTERNAL", "EXTERNAL"

LOCK_CYC: integer (default: 2)

Description

The following are descriptions of EPLLB port functions.

Port	I/O	Function
CLKI	I	Global clock input; frequency: 20~420 MHz.
RST	I	PLL reset.
CLKFB	I	External feedback, internal feedback from CLKOP divider; frequency: 20~420 MHz.
CLKOP	O	PLL output clock to clock tree (no phase shift); frequency: 20~420 MHz.
LOCK	O	"1" indicates PLL LOCK to CLK_IN.

You can refer to the following technical note on the Lattice web site for more detailed description and usage.

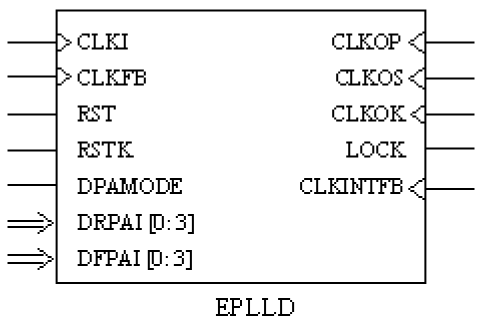
- TN1049 - LatticeECP/EC and LatticeXP sysCLOCK PLL Design and Usage Guide

EPLLD

Enhanced PLL

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: CLKI, CLKFB, RST, RSTK, DPAMODE, DRPAI3, DRPAI2, DRPAI1, DRPAI0, DFPAI3, DFPAI2, DFPAI1, DFPAI0

OUTPUTS: CLKOP, CLKOS, CLKOK, LOCK, CLKINTFB

ATTRIBUTES:

FIN: 20.0000~420.0000 (in MHz) (default: "100.0000")

CLKI_DIV: integers 1~64 (default: 1)

CLKFB_DIV: integers 1~64 (default: 1)

CLKOP_DIV: 2, 4, 8 (default), 16, 32, 48, 64, 80, 96,112, 128

CLKOK_DIV: 2 (default), 4, 6, 8, ..., 126, 128

PHASEADJ: 0 (default), 22.5, 45, 67.5, 90, ..., 315, 337.5

DUTY: integers 2~14 (default: 8)

PHASE_CNTL: "STATIC" (default), "DYNAMIC"

PLLCAP: "DISABLED" (default), "ENABLED", "AUTO"

CLKOP_BYPASS: "DISABLED" (default), "ENABLED"

CLKOS_BYPASS: "DISABLED" (default), "ENABLED"

CLKOK_BYPASS: "DISABLED" (default), "ENABLED"

PLLTYPE: "AUTO" (default), "SPLL", "GPLL"

Description

The following are descriptions of EPLLD port functions.

Port	I/O	Function
CLKI	I	Input clock.
CLKFB	I	Feedback clock.
RST	I	PLL reset (connected to the CNTRST port). High active reset.
RSTK	I	Reset for K divider (connected to the RESETK port). High active reset.
DPAMODE	I	Dynamic phase adjust mode. Active high indicates pin control (DYNAMIC) and active low indicates fuse control (STATIC).
DRPAI[3:0]	I	Dynamic coarse phase shift; rise edge setting.
DFPAI[3:0]	I	Dynamic coarse phase shift; falling edge setting.
CLKOP	O	PLL output clock (no phase shift).
CLKOS	O	PLL output clock (phase shifted/duty cycle changed).
CLKOK	O	PLL output to clock tree (no phase shift, low speed).
LOCK	O	PLL LOCK to CLKI, asynchronous signal. Active high indicates PLL lock.
CLKINTFB	O	Internal feedback source. CLKOP divider output before CLOCK TREE.

The EPLLD primitive can be used for design migration between LatticeECP2 and LatticeXP2, which can be divided into four situations:

Design Migration From LatticeECP2 to LatticeXP2 (EPLLD Configurations):

- EPLLD Configurations without Dynamic Phase & Duty Options: The LatticeECP2 PLL configuration migrates to LatticeXP2 without any changes.
- EPLLD Configurations with Dynamic Phase & Duty Options: The LatticeECP2 PLL configuration has DPAMODE port in the top-level port list. To migrate this configuration to LatticeXP2, the user has to tie the DPAMODE port to GND.

Design Migration From LatticeECP2 to LatticeXP2 (EHXPLLD Configurations): This configuration cannot be migrated to LatticeXP2 because LatticeXP2 does not support Delay Adjust.

Design Migration From LatticeXP2 to LatticeECP2 (EPLLD Configurations):

- EPLLD Configurations without Dynamic Phase & Duty Options: The LatticeXP2 PLL configuration migrates to LatticeECP2 without any changes.
- EPLLD Configurations with Dynamic Phase & Duty Options (No Duty Trim): The LatticeXP2 PLL configuration has no DPAMODE port in the top-level port list. Two options for migration:
  a. Regenerate the PLL configuration for LatticeECP2.
  b. Modify the LatticeXP2 PLL configuration to bring the DPAMODE to top-level port list.

Design Migration From LatticeXP2 to LatticeECP2 (EHXPLLE Configurations): This configuration cannot be migrated to LatticeECP2 because LatticeECP2 does not support Duty Trim Options, CLKOK2 and the CLKOS Fine Delay Port (WRDEL).

You can refer to the following technical notes on the Lattice web site for more detailed description and usage.

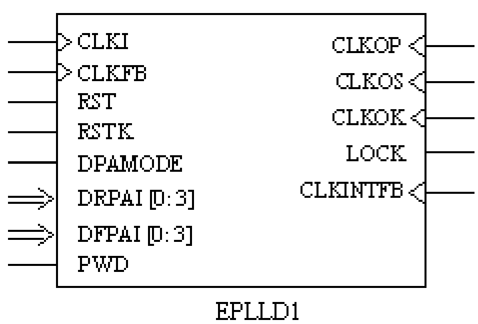
- TN1103 - LatticeECP2 sysCLOCK PLL/DLL Design and Usage Guide
- TN1126 - LatticeXP2 sysCLOCK PLL Design and Usage Guide

EPLLD1

Enhanced PLL

Architectures Supported:

- LatticeXP2



INPUTS: CLKI, CLKFB, RST, RSTK, DPAMODE, DRPAI3, DRPAI2, DRPAI1, DRPAI0, DFPAI3, DFPAI2, DFPAI1, DFPAI0, PWD

OUTPUTS: CLKOP, CLKOS, CLKOK, LOCK, CLKINTFB

ATTRIBUTES:

FIN: 20.0000~420.0000 (in MHz) (default: "100.0000")

CLKI_DIV: integers 1~64 (default: 1)

CLKFB_DIV: integers 1~64 (default: 1)

CLKOP_DIV: 2, 4, 8 (default), 16, 32, 48, 64, 80, 96,112, 128

CLKOK_DIV: 2 (default), 4, 6, 8, ..., 126, 128

PHASEADJ: 0 (default), 22.5, 45, 67.5, 90, ..., 315, 337.5

DUTY: integers 2~14 (default: 8)

PHASE_CNTL: "STATIC" (default), "DYNAMIC"

PLLCAP: "DISABLED" (default), "ENABLED", "AUTO"

CLKOP_BYPASS: "DISABLED" (default), "ENABLED"

CLKOS_BYPASS: "DISABLED" (default), "ENABLED"

CLKOK_BYPASS: "DISABLED" (default), "ENABLED"

PLLTYPE: "AUTO" (default), "SPLL", "GPLL"

Description

The following are descriptions of EPLLD1 port functions.

Port	I/O	Function
CLKI	I	Input clock.
CLKFB	I	Feedback clock.
RST	I	PLL reset (connected to the CNTRST port). High active reset.
RSTK	I	Reset for K divider (connected to the RESETK port). High active reset.
DPAMODE	I	Dynamic phase adjust mode. Active high indicates pin control (DYNAMIC) and active low indicates fuse control (STATIC).
DRPAI[3:0]	I	Dynamic coarse phase shift; rise edge setting.
DFPAI[3:0]	I	Dynamic coarse phase shift; falling edge setting.

Port	I/O	Function
PWD	I	Dynamic power down signal.
CLKOP	O	PLL output clock (no phase shift).
CLKOS	O	PLL output clock (phase shifted/duty cycle changed).
CLKOK	O	PLL output to clock tree (no phase shift, low speed).
LOCK	O	PLL LOCK to CLKI, asynchronous signal. Active high indicates PLL lock.
CLKINTFB	O	Internal feedback source. CLKOP divider output before CLOCK TREE.

You can refer to the following technical note on the Lattice web site for more detailed description and usage.

- TN1126 - LatticeXP2 sysCLOCK PLL Design and Usage Guide

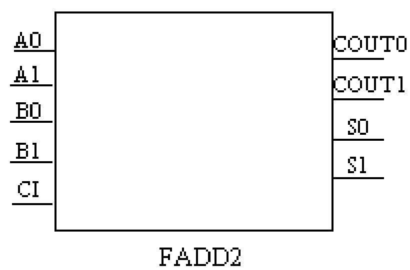
F

FADD2

2 Bit Fast Adder

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- Platform Manager



INPUTS: A1, A0, B1, B0, CI

OUTPUTS: COUT1, COUT0, S1, S0

Description

FADD2 is a 2-bit adder. It has a carry-in input (CI) and two 2-bit input (A0, A1 and B0, B1). The FADD2 produces a 2-bit sum output (S0, S1) along with a 2-bit carry-out output (COUT1, COUT).

Example pin functions:

Function	Pins
input	A1, A0, B1, B0
output	S1, S0
carry-in input	CI

Function	Pins
carry-out output (Bit-0)	COUT0
carry-out output (Bit-1)	COUT1

Truth Table

INPUTS					OUTPUTS			
A0	A1	B0	B1	CI	S0	COUT0	S1	COUT1
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	1
0	0	1	1	0	0	1	0	1
1	1	1	1	0	1	0	1	0
0	0	0	0	1	0	1	0	1
1	1	0	0	1	1	0	1	0
0	0	1	1	1	1	0	1	0
1	1	1	1	1	1	1	1	1

Note

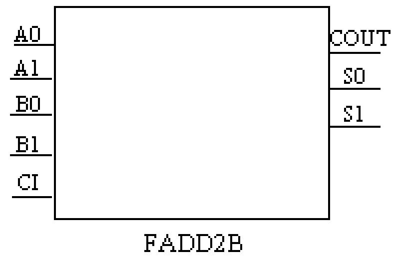
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FADD2B

Fast 2 Bit Adder

Architectures Supported:

- LatticeECP2/M
- LatticeECP3
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: A0, A1, B0, B1, CI

OUTPUTS: COUT, S0, S1

Note

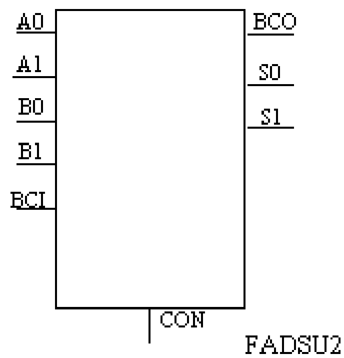
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FADSU2

2 Bit Fast Adder/Subtractor (two's complement)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A0, A1, B0, B1, BCI, CON

OUTPUTS: BCO, S0, S1

Description

FADSU2 is a 2 bit adder/subtractor. When the control signal (CON) is high FADSU2 functions as a 2 bit adder with a carry-in input (BCI) and two 2 bit inputs (A0:A1 and B0:B1), producing a 2 bit SUM output (S0:S1) along with a carry-out output (BCO).

When the control signal (CON) is low, FADSU2 functions as a 2 bit two's complement subtractor with a borrow-in input (BCI) and two 2 bit inputs (A0:A1 and B0:B1), producing a 2 bit two's complement output of A minus B (S0:S1) along with a borrow-out output (BCO).

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

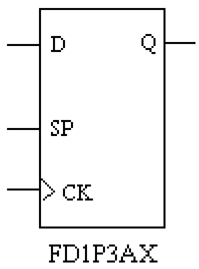
FD1P3AX

Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO

- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, SP, CK

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	SP	CK	Q
X	0	X	Q
0	1	↑	0
1	1	↑	1

X = Don't care

When GSR=0, Q=0 (D=SP=CK=X)

Note

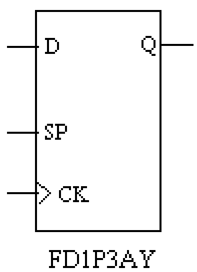
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1P3AY

Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, SP, CK

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	SP	CK	Q
X	0	X	Q
0	1	↑	0
1	1	↑	1

X = Don’t care

When GSR=0, Q=1 (D=SP=CK=X)

Note

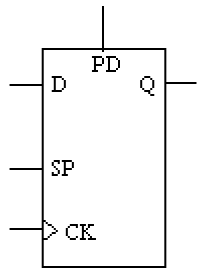
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1P3BX

Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



FD1P3BX

INPUTS: D, SP, CK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS				**OUTPUTS**
D	SP	CK	PD	Q
X	0	X	0	Q
X	X	X	1	1
0	1	↑	0	0
1	1	↑	0	1

X = Don't care

When GSR=0, Q=1 (D=SP=CK=PD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

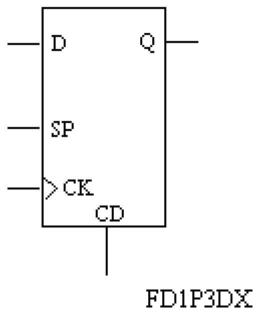
FD1P3DX

Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC

- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, SP, CK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS				OUTPUTS
D	SP	CK	CD	Q
X	0	X	0	Q
X	X	X	1	0
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

When GSR=0, Q=0 (D=SP=CK=CD=X)

Note

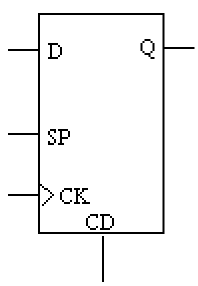
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1P3IX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



FD1P3IX

INPUTS: D, SP, CK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS				OUTPUTS
D	SP	CK	CD	Q
X	0	X	0	Q
X	X	↑	1	0
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

When GSR=0, Q=0 (D=SP=CK=CD=X)

Note

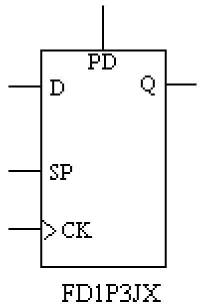
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1P3JX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, SP, CK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS				**OUTPUTS**
D	SP	CK	PD	Q
X	0	X	0	Q
X	X	↑	1	1
0	1	↑	0	0
1	1	↑	0	1

X = Don't care

When GSR=0, Q=1 (D=SP=CK=PD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

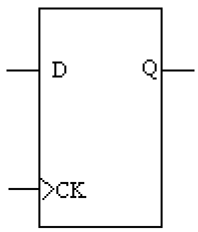
FD1S1A

Positive Level Data Latch with GSR Used for Clear

Architectures Supported:

- LatticeECP/EC

- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



FD1S1A

INPUTS: D, CK

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS		OUTPUTS
D	CK	Q
X	0	Q
0	1	0
1	1	1

X = Don't care

When GSR=0, Q=0 (D=CK=X)

Note

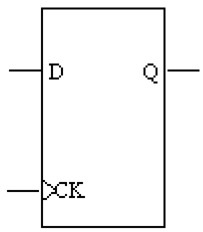
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1S1AY

Positive Level Data Latch with GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, CK

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS		OUTPUTS
D	CK	Q
X	0	Q
0	1	0
1	1	1

X = Don’t care

When GSR=0, Q=1 (D=CK=X)

Note

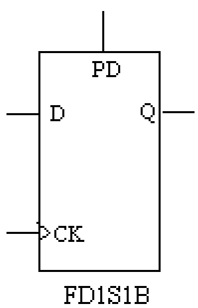
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1S1B

Positive Level Data Latch with Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, CK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	CK	PD	Q
X	0	0	Q
X	X	1	1
0	1	0	0
1	1	0	1

X= Don't care

When GSR=0, Q=1 (D=CK=PD=X)

Note

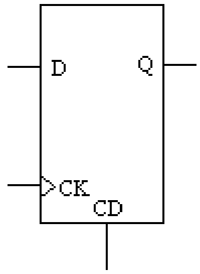
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1S1D

Positive Level Data Latch with Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



FD1S1D

INPUTS: D, CK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	CK	CD	Q
X	0	0	Q
X	X	1	0
0	1	0	0
1	1	0	1

X = Don’t care

When GSR=0, Q=0 (D=CK=CD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

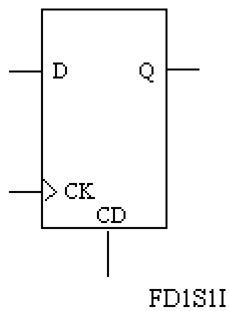
FD1S1I

Positive Level Data Latch with Positive Level Synchronous Clear

Architectures Supported:

- LatticeECP/EC

- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, CK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	CK	CD	Q
X	0	0	Q
X	1	1	0
0	1	0	0
1	1	0	1

X = Don't care

When GSR=0, Q=0 (D=CK=CD=X)

Note

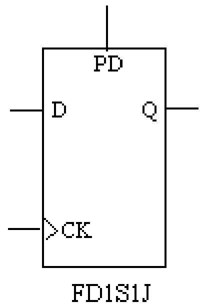
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1S1J

Positive Level Data Latch with Positive Level Synchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, CK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	CK	PD	Q
X	0	0	Q
X	1	1	1
0	1	0	0
1	1	0	1

X = Don't care

When GSR=0, Q=1 (D=CK=PD=X)

Note

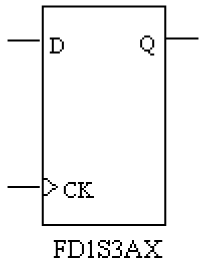
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1S3AX

Positive Edge Triggered D Flip-Flop, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, CK

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS		OUTPUTS
D	CK	Q
0	↑	0
1	↑	1

X = Don’t care

When GSR=0, Q=0 (D=CK=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

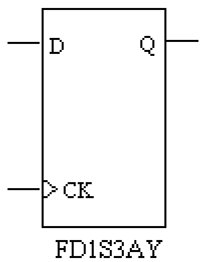
FD1S3AY

Positive Edge Triggered D Flip-Flop, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M

- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, CK

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS		OUTPUTS
D	CK	Q
0	↑	0
1	↑	1

X = Don't care

When GSR=0, Q=1 (D=CK=X)

Note

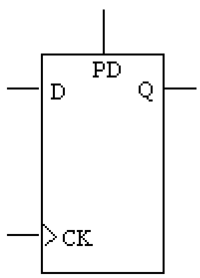
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1S3BX

Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



FD1S3BX

INPUTS: D, CK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	CK	PD	Q
X	X	1	1
0	↑	0	0
1	↑	X	1

X = Don’t care

When GSR=0, Q=1 (D=CK=PD=X)

Note

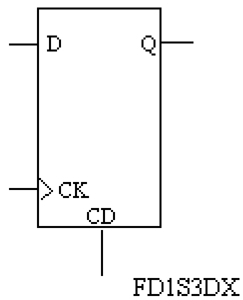
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1S3DX

Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, CK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	CK	CD	Q
X	X	1	0
0	↑	0	0
1	↑	0	1

X = Don’t care

When GSR=0, Q=0 (D=CK=CD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

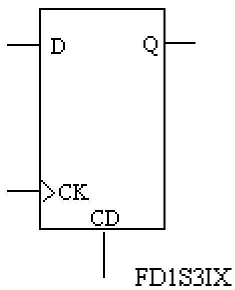
FD1S3IX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M

- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D, CK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	CK	CD	Q
X	↑	1	0
0	↑	0	0
1	↑	0	1

X = Don’t care

When GSR=0, Q=0 (D=CK=CD=X)

Note

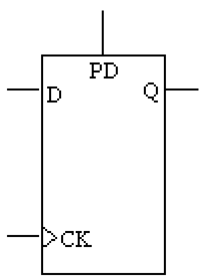
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FD1S3JX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



FD1S3JX

INPUTS: D, CK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS			OUTPUTS
D	CK	PD	Q
X	↑	1	1
0	↑	0	0
1	↑	X	1

X = Don't care

When GSR=0, Q=1 (D=CK=PD=X)

Note

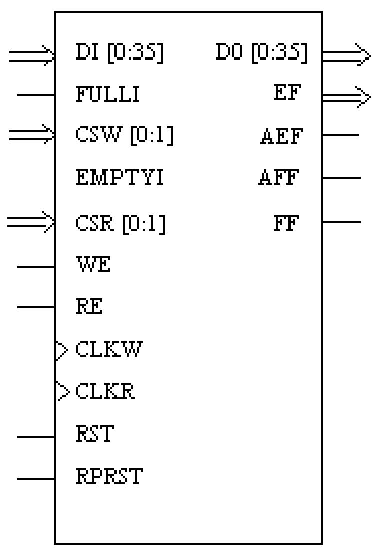
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FIFO16KA

16K FIFO

Architectures Supported:

- LatticeSC/M



FIFO16KA

INPUTS: DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, DI18, DI19, DI20, DI21, DI22, DI23, DI24, DI25, DI26, DI27, DI28, DI29, DI30, DI31, DI32, DI33, DI34, DI35, FULLI, CSW0, CSW1, EMPTYI, CSR0, CSR1, WE, RE, CLKW, CLKR, RST, RPRST

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17, DO18, DO19, DO20, DO21, DO22, DO23, DO24, DO25, DO26, DO27, DO28, DO29, DO30, DO31, DO32, DO33, DO34, DO35, EF, AEF, AFF, FF

ATTRIBUTES:

DATA_WIDTH_W: 1, 2, 4, 9, 18 (default), 36

DATA_WIDTH_R: 1, 2, 4, 9, 18 (default), 36

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_W: any 2-bit binary value (default: 2'b00)

CSDECODE_R: any 2-bit binary value (default: 2'b00)

GSR: "DISABLED" (default), "ENABLED"

AEPOINTER: any 15-bit binary value (default: all zeros)

AEPOINTER1: any 15-bit binary value (default: all zeros)

AFPOINTER: any 15-bit binary value (default: all zeros)

AFPOINTER1: any 15-bit binary value (default: all zeros)

FULLPOINTER: any 15-bit binary value (default: all zeros)

FULLPOINTER1: any 15-bit binary value (default: all zeros)

Description

You can refer to the following technical note on the Lattice web site on details of EBR port definition, attribute definition and usage.

- TN1094 - On-Chip Memory Usage Guide for LatticeSC Devices

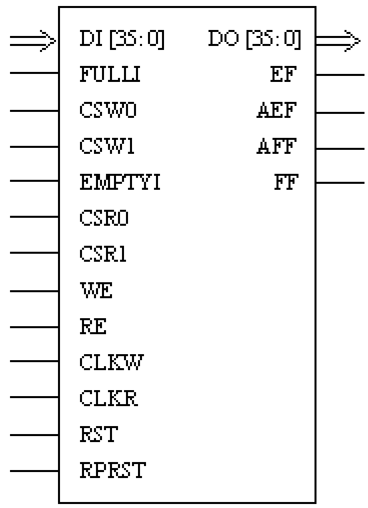
FIFO8KA

8K FIFO

Architectures Supported:

- MachXO

- Platform Manager



INPUTS: DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, DI18, DI19, DI20, DI21, DI22, DI23, DI24, DI25, DI26, DI27, DI28, DI29, DI30, DI31, DI32, DI33, DI34, DI35, FULLI, CSW0, CSW1, EMPTYI, CSR0, CSR1, WE, RE, CLKW, CLKR, RST, RPRST

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17, DO18, DO19, DO20, DO21, DO22, DO23, DO24, DO25, DO26, DO27, DO28, DO29, DO30, DO31, DO32, DO33, DO34, DO35, EF, AEF, AFF, FF

ATTRIBUTES:

DATA_WIDTH_W: 1, 2, 4, 9, 18 (default), 36

DATA_WIDTH_R: 1, 2, 4, 9, 18 (default), 36

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_W: any 2-bit binary value (default: 2’b00)

CSDECODE_R: any 2-bit binary value (default: 2’b00)

AEPOINTER: any 14-bit binary value (default: all zeros)

AEPOINTER1: any 14-bit binary value (default: all zeros)

AFPOINTER: any 14-bit binary value (default: all zeros)

AFPOINTER1: any 14-bit binary value (default: all zeros)

FULLPOINTER: any 14-bit binary value (default: all zeros)

FULLPOINTER1: any 14-bit binary value (default: all zeros)

GSR: "DISABLED" (default), "ENABLED"

Description

You can refer to the following technical note on the Lattice web site on detailed information and usage.

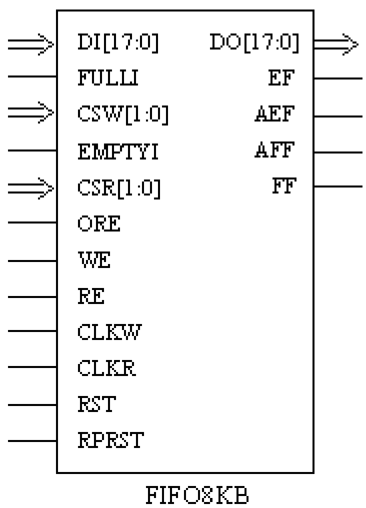
- TN1092 - MachXO Memory Usage Guide

FIFO8KB

8K FIFO Block RAM

Architecture Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: DI17, DI16, DI15, DI14, DI13, DI12, DI11, DI10, DI9, DI8, DI7, DI6, DI5, DI4, DI3, DI2, DI1, DI0, FULLI, CSW1, CSW0, EMPTYI, CSR1, CSR0, ORE, WE, RE, CLKW, CLKR, RST, RPRST

OUTPUTS: DO17, DO16, DO15, DO14, DO13, DO12, DO11, DO10, DO9, DO8, DO7, DO6, DO5, DO4, DO3, DO2, DO1, DO0, EF, AEF, AFF, FF

ATTRIBUTES:

DATA_WIDTH_W: 1, 2, 4, 9, 18 (default)

DATA_WIDTH_R: 1, 2, 4, 9, 18 (default)

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_W: any 2-bit binary value (default: zeros)

CSDECODE_R: any 2-bit binary value (default: zeros)

GSR: "DISABLED" (default), "ENABLED"

RESETMODE: "ASYNC" (default), "SYNC"

ASYNC_RESET_RELEASE: "SYNC" (default), "ASYNC"

AEPOINTER: any 14-bit binary value (default: all zeros)

AEPOINTER1: any 14-bit binary value (default: all zeros)

AFPOINTER: any 14-bit binary value (default: all zeros)

AFPOINTER1: any 14-bit binary value (default: all zeros)

FULLPOINTER: any 14-bit binary value (default: all zeros)

FULLPOINTER1: any 14-bit binary value (default: all zeros)

Description

The following table describes the I/O ports of the FIFO8KB primitive.

Port Name	I/O	Definition
DI[17:0]	I	Write data (up to 18)
CLKW	I	Write clock
WE	I	Write clock enable
RST	I	Reset write pointers
FULLI	I	Chip select write
CSW[1:0]	I	Chip select write
CLKR	I	Read clock
RE	I	Read clock enable
ORE	I	Read output clock enable
EMPTYI	I	Chip select read

Port Name	I/O	Definition
CSR[1:0]	I	Chip select read
RPRST	I	Reset read pointers
DO[17:0]	O	Read data (up to 18)
AFF	O	Almost full flag
FF	O	Full flag
AEF	O	Almost empty flag
EF	O	Empty flag

You can refer to the following technical note on the Lattice web site on detailed information and usage.

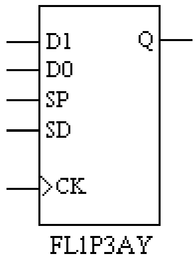
- TN1201 - Memory Usage Guide for MachXO2 Devices

FL1P3AY

Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, SP, CK, SD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS					**OUTPUTS**
D0	D1	SP	SD	CK	Q
X	X	0	X	X	Q
0	X	1	0	↑	0
1	X	1	0	↑	1
X	0	1	1	↑	0
X	1	1	1	↑	1

X = Don’t care

When GSR=0, Q=1 (D0=D1=SP=SD=CK=X)

Note

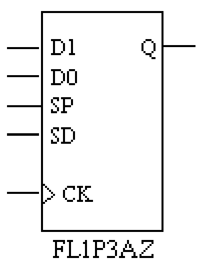
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1P3AZ

Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, SP, CK, SD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS					OUTPUTS
D0	D1	SP	SD	CK	Q
X	X	0	X	X	Q
0	X	1	0	↑	0
1	X	1	0	↑	1
X	0	1	1	↑	0
X	1	1	1	↑	1

X = Don't care

When GSR=0, Q=0 (D0=D1=SP=SD=CK=X)

Note

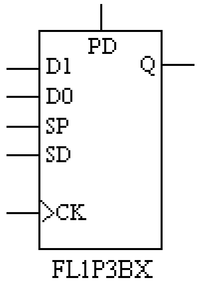
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1P3BX

Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, SP, CK, SD, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS						**OUTPUTS**
D0	D1	SP	SD	CK	PD	Q
X	X	0	X	X	0	Q
X	X	X	X	X	1	1
0	X	1	0	↑	0	0
1	X	1	0	↑	X	1
X	0	1	1	↑	0	0
X	1	1	1	↑	X	1

X = Don’t care

When GSR=0, Q=1 (D0=D1=SP=SD=CK=PD=X)

Note

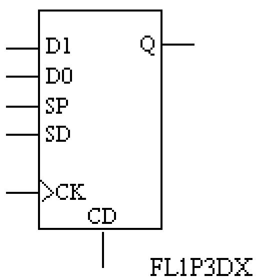
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1P3DX

Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Clear, and Positive Level Enable

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, SP, CK, SD, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS						OUTPUTS
D0	D1	SP	SD	CK	CD	Q
X	X	0	X	X	0	Q
X	X	X	X	X	1	0
0	X	1	0	↑	0	0
1	X	1	0	↑	X	1
X	0	1	1	↑	0	0
X	1	1	1	↑	X	1

X = Don't care

When GSR=0, Q=0 (D0=D1=SP=SD=CK=CD=X)

Note

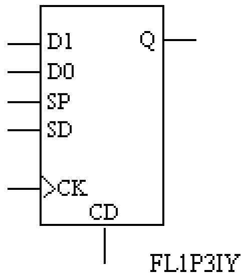
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1P3IY

Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Clear, and Positive Level Enable (Clear overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, SP, SD, CK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS						OUTPUTS
D0	D1	SP	SD	CK	CD	Q
X	X	0	X	X	0	Q
X	X	X	X	↑	1	0
0	X	1	0	↑	X	0
1	X	1	0	↑	0	1
X	0	1	1	↑	X	0
X	1	1	1	↑	0	1

X = Don't care

When GSR=0, Q=1 (D0=D1=SP=SD=CK=CD=X)

Note

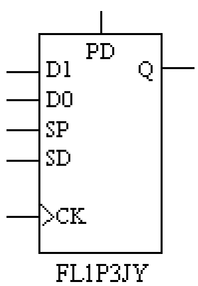
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1P3JY

Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Synchronous Preset, and Positive Level Enable (Preset overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, SP, CK, SD, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS						OUTPUTS
D0	D1	SP	SD	CK	PD	Q
X	X	0	X	X	0	Q
X	X	X	X	↑	1	1
0	X	1	0	↑	0	0
1	X	1	0	↑	X	1
X	0	1	1	↑	0	0
X	1	1	1	↑	X	1

X = Don't care

When GSR=0, Q=1 (D0=D1=SP=SD=CK=PD=X)

Note

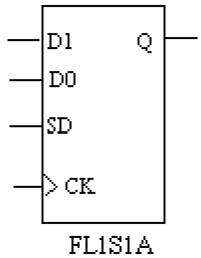
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1S1A

Positive Level Loadable Latch with Positive Select, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CK, SD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS				OUTPUTS
D0	D1	SD	CK	Q
0	X	0	1	0
1	X	0	1	1
X	0	1	1	0
X	1	1	1	1

X = Don’t care

When GSR=0, Q=0 (D0=D1=SD=CK=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

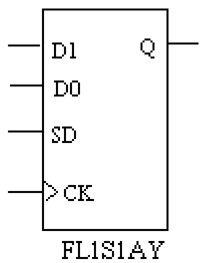
FL1S1AY

Positive Level Loadable Latch with Positive Select, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC

- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CK, SD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS				OUTPUTS
D0	D1	SD	CK	Q
0	X	0	1	0
1	X	0	1	1
X	0	1	1	0
X	1	1	1	1

X = Don't care

When GSR=0, Q=1 (D0=D1=SD=CK=X)

Note

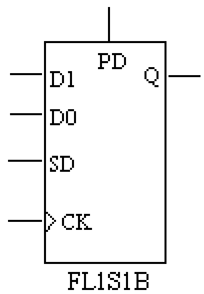
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1S1B

Positive Level Loadable Latch with Positive Level Data Select and Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CK, SD, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS					OUTPUTS
D0	D1	SD	CK	PD	Q
X	X	X	0	0	Q
X	X	X	X	1	1
0	X	0	1	0	0
1	X	0	1	X	1
X	0	1	1	0	0
X	1	1	1	X	1

X= Don't care

When GSR=0, Q=1 (D0=D1=SD=CK=PD=X)

Note

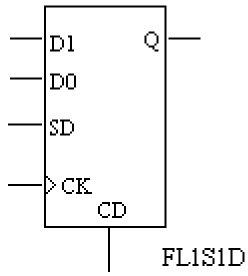
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1S1D

Positive Level Loadable Latch with Positive Level Data Select and Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CK, SD, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS					OUTPUTS
D0	D1	SD	CK	CD	Q
X	X	X	0	0	Q
X	X	X	X	1	0
0	X	0	1	X	0
1	X	0	1	0	1
X	0	1	1	X	0
X	1	1	1	0	1

X = Don’t care

When GSR=0, Q=0 (D0=D1=SD=CK=CD=X)

Note

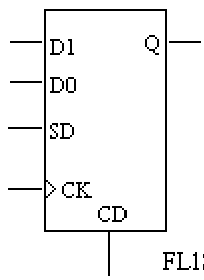
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1S1I

Positive Level Loadable Latch with Positive Level Data Select and Positive Level Synchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- Platform Manager
- Platform Manager 2



FL1S1I

INPUTS: D0, D1, CK, SD, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS					OUTPUTS
D0	D1	SD	CK	CD	Q
X	X	X	0	0	Q
X	X	X	1	1	0
0	X	0	1	X	0
1	X	0	1	0	1
X	0	1	1	X	0
X	1	1	1	0	1

X = Don't care

When GSR=0, Q=0 (D0=D1=SD=CK=CD=X)

Note

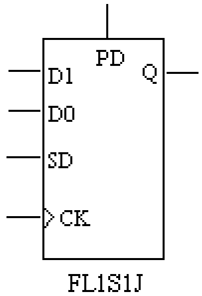
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1S1J

Positive Level Loadable Latch with Positive Level Data Select and Positive Level Synchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP
- MachXO
- MachXO2
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CK, SD, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS					OUTPUTS
D0	D1	SD	CK	PD	Q
X	X	X	0	0	Q
X	X	X	1	1	1
0	X	0	1	0	0
1	X	0	1	X	1
X	0	1	1	0	0
X	1	1	1	X	1

X = Don’t care

When GSR=0, Q=1 (D0=D1=SD=CK=PD=X)

Note

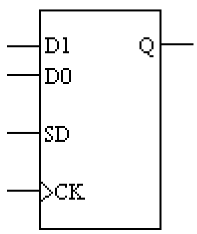
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1S3AX

Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



FL1S3AX

INPUTS: D0, D1, CK, SD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS				OUTPUTS
D0	D1	SD	CK	Q
0	X	0	↑	0
1	X	0	↑	1
X	0	1	↑	0
X	1	1	↑	1

X = Don’t care

When GSR=0, Q=0 (D0=D1=SD=CK=X)

Note

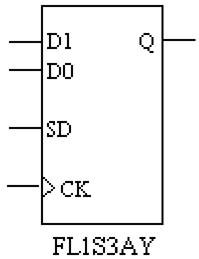
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FL1S3AY

Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CK, SD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS				**OUTPUTS**
D0	D1	SD	CK	Q
0	X	0	↑	0
1	X	0	↑	1
X	0	1	↑	0
X	1	1	↑	1

X = Don’t care

When GSR=0, Q=1 (D0=D1=SD=CK=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

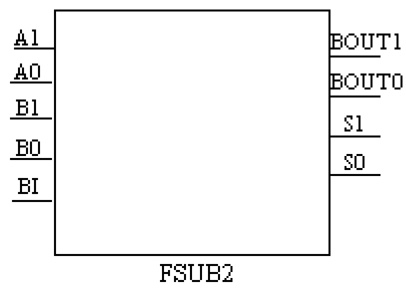
FSUB2

2 Bit Fast Subtractor (two's complement)

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M

- LatticeXP
- MachXO
- Platform Manager



INPUTS: A1, A0, B1, B0, BI

OUTPUTS: BOUT1, BOUT0, S1, S0

Description

FSUB2 is a 2-bit two's complement subtractor. It has a borrow-in input (BI) and two 2-bit input (A0, A1 and B0, B1). The FSUB2 produces a 2-bit difference output (S0, S1) along with a 2-bit borrow-out output (BOUT1, BOUT).

Example pin functions:

Function	Pins
input	A1, A0, B1, B0
output	S1, S0
borrow-in input	BI
borrow-out output (Bit-0)	BOUT0
borrow-out output (Bit-1)	BOUT1

Truth Table

INPUTS					OUTPUTS			
A0	A1	B0	B1	BI	S0	BOUT0	S1	BOUT1
0	0	0	0	0	0	0	1	0
1	1	0	0	0	1	1	0	1
0	0	1	1	0	0	0	0	0
1	1	1	1	0	0	0	1	0
0	0	0	0	1	1	1	0	1
1	1	0	0	1	1	1	1	1
0	0	1	1	1	0	0	1	0
1	1	1	1	1	1	1	0	1

Note

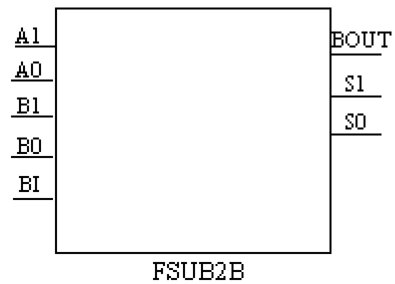
- BI and BO are inverse from standard two's complement behavior.
- This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

FSUB2B

2 Bit Subtractor

Architectures Supported:

- LatticeECP2/M
- LatticeECP3
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: A0, A1, B0, B1, BI

OUTPUTS: BOUT, S0, S1

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

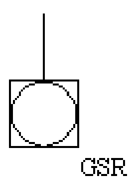
G

GSR

Global Set/Reset Interface

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUT: GSR

Description

GSR is used to reset or set all register elements in your design. The GSR component can be connected to a net from an input buffer or an internally generated net. It is active LOW and when pulsed will set or reset all flip-flops, latches, registers, and counters to the same state as the local set or reset functionality.

It is not necessary to connect signals for GSR to any register elements explicitly. The function will be implicitly connected globally. The functionality of the GSR for sequential cells without a local set or reset are described in the appropriate library help page.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

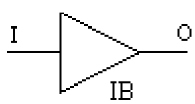
I

IB

Input Buffer

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUT: I

OUTPUT: O

Truth Table

INPUTS	OUTPUTS
I	O
1	1
0	0
Z	U

U = Unknown

Note

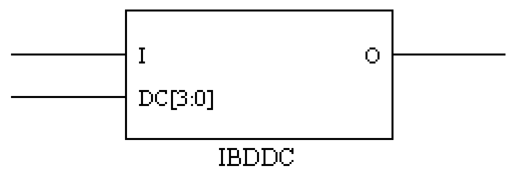
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IBDDC

Dynamic Delay

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: I, DC3, DC2, DC1, DC0

OUTPUT: O

Description

The IBDDC primitive is used to control the input delay dynamically. See the below table for the I/O description.

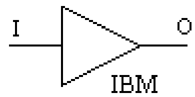
Port Name	I/O	Definition
I	Input	Input
DC[3:0]	Input	Dynamic delay control
O	Output	Output

IBM

CMOS Input Buffer

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUT: I

OUTPUT: O

Truth Table

INPUTS	OUTPUTS
I	O
1	1
0	0
Z	U

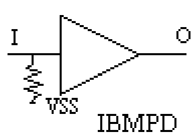
U = Unknown

IBMPD

CMOS Input Buffer with Pull-down

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUT: I

OUTPUT: O

Truth Table

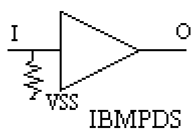
INPUTS	OUTPUTS
I	O
1	1
0	0
Z	0

IBMPDS

CMOS Input Buffer with Pull-down and Delay

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUT: I

OUTPUT: O

Truth Table

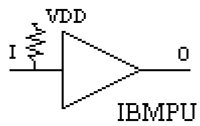
INPUTS	OUTPUTS
I	O
1	1
0	0
Z	0

IBMPU

CMOS Input Buffer with Pull-up

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUT: I

OUTPUT: O

Truth Table

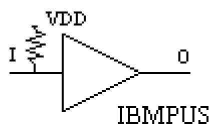
INPUTS	OUTPUTS
I	O
1	1
0	0
Z	1

IBMPUS

CMOS Input Buffer with Pull-up and Delay

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUT: I

OUTPUT: O

Truth Table

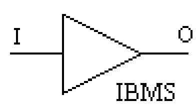
INPUTS	OUTPUTS
I	O
1	1
0	0
Z	1

IBMS

CMOS Input Buffer with Delay

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUT: I

OUTPUT: O

Truth Table

INPUTS	OUTPUTS
I	O
1	1
0	0
Z	U

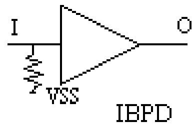
U = Unknown

IBPD

Input Buffer with Pull-down

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUT: I

OUTPUT: O

Truth Table

INPUTS	OUTPUTS
I	O
1	1
0	0
Z	0

Note

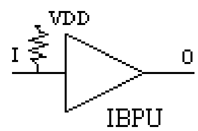
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IBPU

Input Buffer with Pull-up

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUT: I

OUTPUT: O

Truth Table

INPUTS	OUTPUTS
I	O
1	1
0	0
Z	1

Note

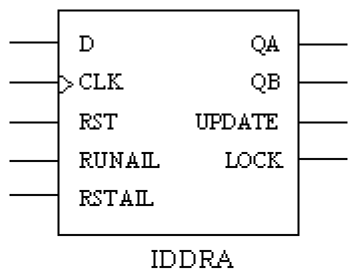
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IDDRA

Input DDR

Architectures Supported:

- LatticeSC/M

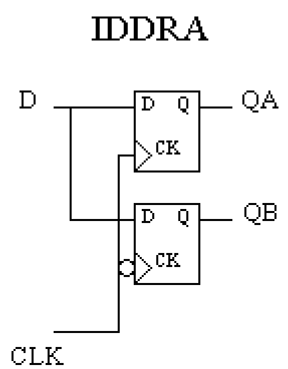


INPUTS: D, CLK, RST, RUNAIL, RSTAIL

OUTPUTS: QA, QB, UPDATE, LOCK

Description

Double Data Rate input logic. The following symbolic diagram shows the flip-flop structure of this primitive.



For more usage, see related technical notes or contact technical support.

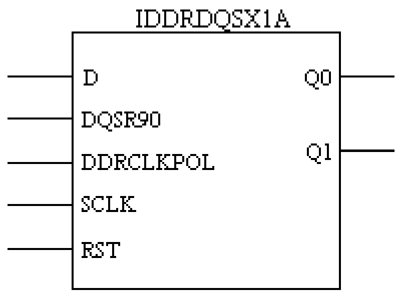
IDDRDQSX1A

Input for DDR1/2 Memory

Architectures Supported:

- MachXO2

- Platform Manager 2



INPUTS: D, DQSR90, DDRCLKPOL, SCLK, RST

OUTPUTS: Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

IDDR supports two clock domains (right side only). IDDRDQSX1A is the input for DDR1/2 memory. It is used for right bank only.

See the following table for port description of the IDDRDQSX1A primitive.

Port Name	I/O	Definition
D	I	DDR input from sysIO buffer.
DQSR90	I	Shifted DQS input for read.
DDRCLKPOL	I	DQS clock polarity. This signal is used to connect to the DDRCLKPOL output of DQSBUFH.
SCLK	I	System clock.
RST	I	RESET to this block from CIB.
Q0	O	Data at the positive edge of the clock.
Q1	O	Data at the negative edge of the clock.

For more information and usage, refer to the following technical note on the Lattice web site.

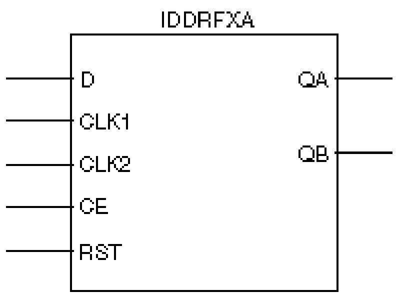
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

IDDRFXA

DDR Generic Input with Full Clock Transfer (x1 Gearbox)

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: D, CLK1, CLK2, RST, CE

OUTPUTS: QA, QB

Description

This primitive inputs DDR data at both edges of clock CLK1 and generates two streams of data aligned to clock CLK2. CLK1 is used to register the DDR registers and the first set of synchronization registers. CLK2 is used by the third stage of registers. Both CLK1 and CLK2 should be clocked by the FPGA clock. The LatticeECP2/M Family Data Sheet explains the input register block in more detail.

Note that LSR is only for second stage register/latch. For supporting DDR modes configured for bidirectional use, software will tie LSR LOW for input registers. The default for LSR is HIGH.

See the following table for port description of the IDDRFXA primitive.

Port Name	I/O	Definition
D	I	DDR data
CLK1	I	Clock connected to the FPGA clock
CLK2	I	Clock connected to the FPGA clock
CE	I	Clock enable signal
RST	I	Signal used to reset the DDR register
QA	O	Data at the positive edge of the CLK
QB	O	Data at the negative edge of the CLK

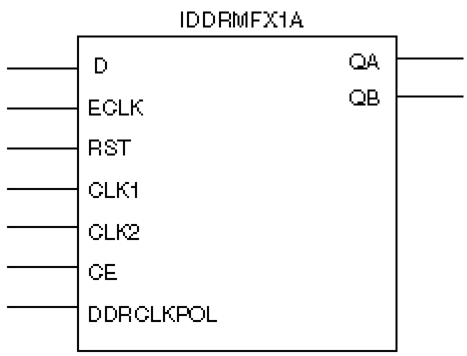
For more usage, see related technical notes or contact technical support.

IDDRMFX1A

DDR Input and DQS to System Clock Transfer Registers with Full Clock Cycle Transfer

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: D, ECLK, CLK1, CLK2, RST, CE, DDRCLKPOL

OUTPUTS: QA, QB

Description

This primitive implements a full clock cycle transfer as compared the to the IDDRMX1A primitive that will only implement a half clock cycle transfer. The DDR registers are designed to use edge clock routing on the I/O side and the primary clock on the FPGA side. The ECLK input is used to connect to the DQS strobe coming from the DQS delay block (DQSBUFC primitive). The CLK1 and CLK2 inputs should be connected to the slow system (FPGA) clock. DDRCLKPOL is an input from the DQS Clock Polarity tree. This signal is generated by the DQS Transition detect circuit in the hardware.

Note that the DDRCLKPOL input to IDDRMFX1A should be connected to the DDRCLKPOL output of DQSBUFC. LSR is only for second stage register/ latch. For supporting DDR modes configured for bidirectional use, software will tie LSR LOW for input registers. The default for LSR is HIGH.

See the following table for port description of the IDDRMFX1A primitive.

Port Name	I/O	Definition
D	I	DDR data.
ECLK	I	The phase shifted DQS should be connected to this input.
RST	I	Reset.
CLK1	I	Slow FPGA CLK.
CLK2	I	Slow FPGA CLK.
CE	I	Clock enable.
DDRCLKPOL	I	DDR clock polarity signal.
QA	O	Data at the positive edge of the CLK.
QB	O	Data at the negative edge of the CLK.

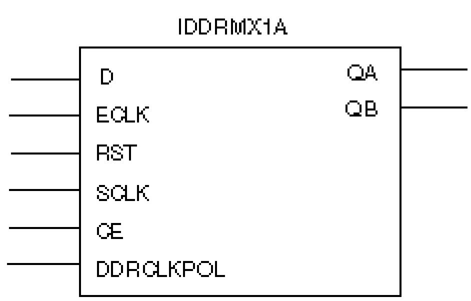
For more usage, see related technical notes or contact technical support.

IDDRMX1A

DDR Input and DQS to System Clock Transfer Registers with Half Clock Cycle Transfer

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: D, ECLK, SCLK, RST, CE, DDRCLKPOL

OUTPUTS: QA, QB

Description

This primitive implements the input register block. The DDR registers are designed to use edge clock routing on the I/O side and the primary clock on the FPGA side. The ECLK input is used to connect to the DQS strobe coming from the DQS delay block (DQSBUFC primitive). The SCLK input should be connected to the system (FPGA) clock. DDRCLKPOL is an input from the DQS Clock Polarity tree. This signal is generated by the DQS Transition detect circuit in the hardware. The DDRCLKPOL signal is used to choose the polarity of the SCLK to the synchronization registers.

Note that the DDRCLKPOL input to IDDRMX1A should be connected to the DDRCLKPOL output of DQSBUFC. LSR is only for second stage register/ latch. For supporting DDR modes configured for bidirectional use, software will tie LSR LOW for input registers. The default for LSR is HIGH.

See the following table for port description of the IDDRMX1A primitive.

Port Name	I/O	Definition
D	I	DDR data.
ECLK	I	The phase shifted DQS should be connected to this input.
RST	I	Reset.
SCLK	I	System CLK.
CE	I	Clock enable.
DDRCLKPOL	I	DDR clock polarity signal.
QA	O	Data at the positive edge of the CLK.
QB	O	Data at the negative edge of the CLK.

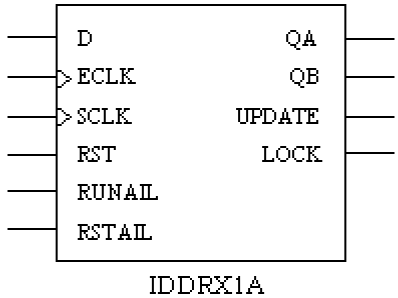
For more usage, see related technical notes or contact technical support.

IDDRX1A

Input DDR

Architectures Supported:

- LatticeSC/M

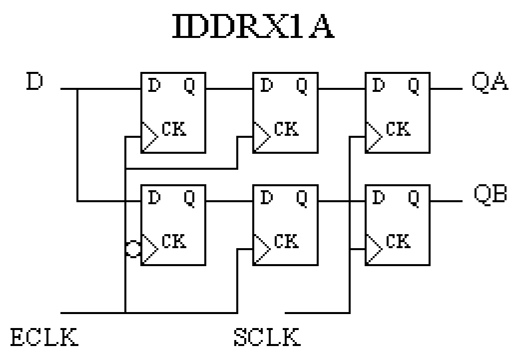


INPUTS: D, ECLK, SCLK, RST, RUNAIL, RSTAIL

OUTPUTS: QA, QB, UPDATE, LOCK

Description

Double Data Rate input logic. The input register block captures DDR input data using edge clock to primary clock domain transfer. It can also be set to perform the same functions as in the shift mode. The following symbolic diagram shows the flip-flop structure of this primitive.



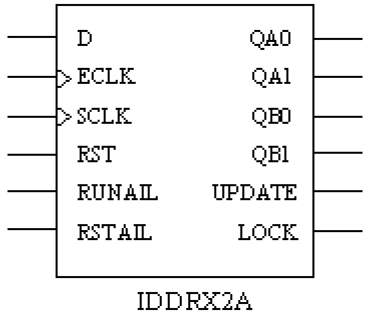
For more usage, see related technical notes or contact technical support.

IDDRX2A

Input DDR

Architectures Supported:

- LatticeSC/M

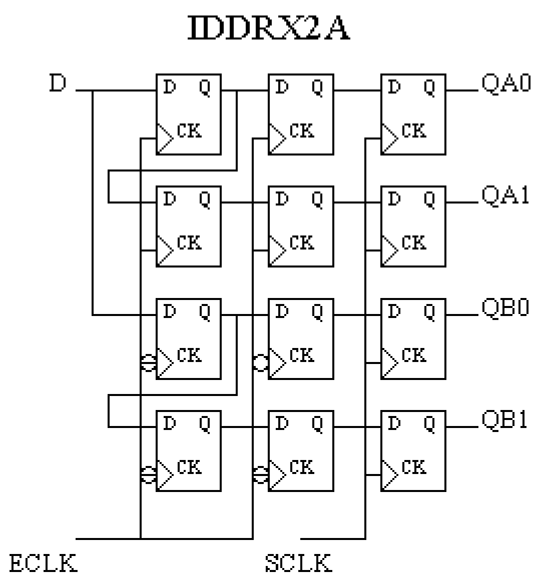


INPUTS: D, ECLK, SCLK, RST, RUNAIL, RSTAIL

OUTPUTS: QA0, QA1, QB0, QB1, UPDATE, LOCK

Description

Double Data Rate input logic. The input register block captures DDR input data using edge clock to primary clock domain transfer. It can also be set to perform the same functions as in the shift mode. The following symbolic diagram shows the flip-flop structure of this primitive.



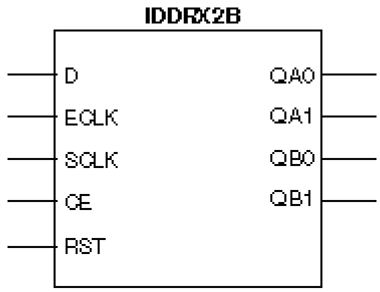
For more usage, see related technical notes or contact technical support.

IDDRX2B

DDR Generic Input with 2x Gearing Ratio

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: D, ECLK, SCLK, CE, RST

OUTPUTS: QA0, QA1, QB0, QB1

Description

This primitive is used when a gearing function is required. This primitive inputs DDR data at both edges of the edge and generates four streams of data. The DDR registers and the first set of synchronization registers are clocked by the ECLK input of the primitive, which should be connected to the fast ECLK. SCLK is used to clock the third stage of registers and should be connected to the FPGA clock. This primitive outputs four streams of data. Two of these data streams are generated using the complementary PIO registers.

Note that LSR is only for second stage register/latch. For supporting DDR modes configured for bidirectional use, software will tie LSR LOW for input registers. The default for LSR is HIGH.

See the following table for port description of the IDDRX2B primitive.

IDDRX2B Ports

Port Name	I/O	Definition
D	I	DDR data
ECLK	I	Clock connected to the fast edge clock
SCLK	I	Clock connected to the FPGA clock
CE	I	Clock enable signal
RST	I	Signal used to reset the DDR register
QA0, QA1	O	Data at the positive edge of the CLK
QB0, QB1	O	Data at the negative edge of the CLK

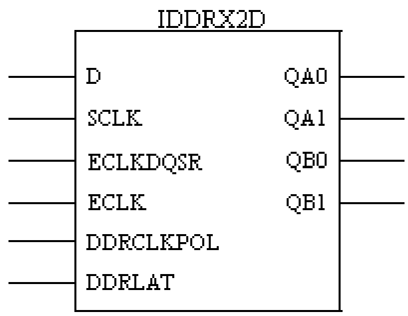
For more usage, see related technical notes or contact technical support.

IDDRX2D

Input DDR for DDR3_MEM, DDR_GENX2, and DDR3_MEMGEN

Architectures Supported:

- LatticeECP3



INPUTS: D, SCLK, ECLKDQSR, ECLK, DDRLAT, DDRCLKPOL

OUTPUTS: QA0, QA1, QB0, QB1

ATTRIBUTES:

(*EA only*) SCLKLATENCY: 1 (default), 2

Description

IDDRX2D is the input DDR for DDR3_MEM, DDR_GENX2 (DDR generic mode in X2 gearing), and DDR3_MEMGEN.

- E: DDR_GENX2 (left/right/top)
- E and EA: DDR3_MEM (left/right)
- E and EA: DDR3_MEMGEN (left/right/top)

See the below table for its port description.

Signal	I/O	Description
D	I	DDR input from sysIO buffer.
ECLK	I	Edge clock. Goes to the second stage of DDR registers.
SCLK	I	System clock. Clock used to transfer from the ECLK to the SCLK domain. SCLK = 1/2 ECLK rate. Goes to the third stage of registers.

Signal	I/O	Description
ECLKDQSR	I	Phase shifted DQS in case of DDR memory interface. Edge clock for generic DDR interfaces. Connects to DQSBUF. For EA devices, ECLKDQSR should be used only for the DQS strobe.
DDRCLKPOL	I	DDR clock polarity signal.
DDRLAT	I	DDR latch control to input logic. Used to guarantee IDDRX2 gearing by selectively enabling a D flip-flop in the data path.
QA0	O	Data at the positive edge of the clock (IPA).
QA1	O	Data at the positive edge of the clock (IPB).
QB0	O	Data at the negative edge of the clock (INA).
QB1	O	Data at the negative edge of the clock (INB).

For more information and usage, refer to the following technical note on the Lattice web site.

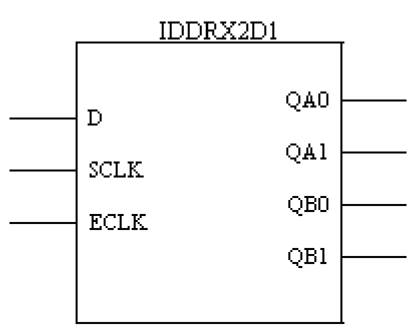
- TN1177 - LatticeECP3 sysIO Usage Guide

IDDRX2D1

Input DDR for DDR_GENX2

Architectures Supported:

- LatticeECP3



INPUTS: D, SCLK, ECLK

OUTPUTS: QA0, QA1, QB0, QB1

ATTRIBUTES:

(*EA only*) DR_CONFIG: "DISABLED" (default), "ENABLED"

Description

IDDRX2D1 is the input DDR for DDR_GENX2 (DDR generic mode in X2 gearing).

- EA: DDR_GENX2 (left/right/top)

See the below table for its port description.

Signal	I/O	Description
D	I	DDR input from sysIO buffer.
ECLK	I	Edge clock. Goes to the first and second stage of DDR registers.
SCLK	I	System clock. Clock used to transfer from the ECLK to the SCLK domain. SCLK = 1/2 ECLK rate. Goes to the third stage of registers.
QA0	O	Data at the positive edge of the clock (IPA).
QA1	O	Data at the positive edge of the clock (IPB).
QB0	O	Data at the negative edge of the clock (INA).
QB1	O	Data at the negative edge of the clock (INB).

For more information and usage, refer to the following technical note on the Lattice web site.

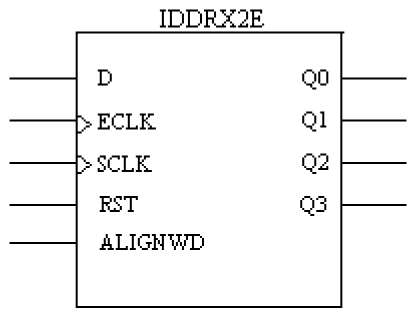
- TN1177 - LatticeECP3 sysIO Usage Guide

IDDRX2E

Input for Generic DDR X2 Using 1:4 Gearing

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, ECLK, SCLK, RST, ALIGNWD

OUTPUTS: Q0, Q1, Q2, Q3

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

IDDRX2E is the input for generic DDR X2 using 1:4 gearing. It uses the VPIC_RX hardware cell. It is used for bottom bank only.

See the below table for port description.

Signal	I/O	Description
D	I	DDR input from sysIO buffer.
ECLK	I	Clock connected to the high speed edge clock tree.
SCLK	I	Clock connected to the system clock.
RST	I	RESET to this block from CIB.
ALIGNWD	I	Data alignment signal used for word alignment. Each operation shifts word alignment by 1 bit.
Q0, Q2	O	Data available at the same edge of the clock.
Q1, Q3	O	Data available at the same edge of the clock.

For more information and usage, refer to the following technical note on the Lattice web site.

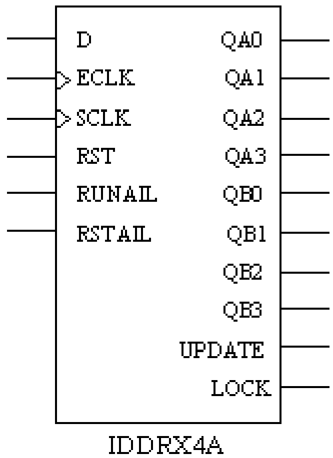
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

IDDRX4A

Input DDR

Architectures Supported:

- LatticeSC/M

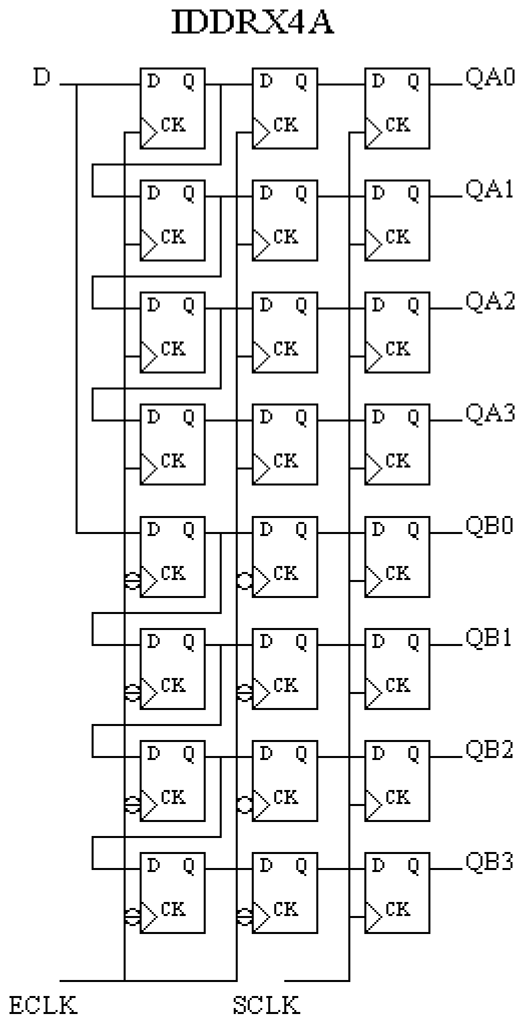


INPUTS: D, ECLK, SCLK, RST, RUNAIL, RSTAIL

OUTPUTS: QA0, QA1, QA2, QA3, QB0, QB1, QB2, QB3, UPDATE, LOCK

Description

Double Data Rate input logic. The input register block captures DDR input data using edge clock to primary clock domain transfer. It can also be set to perform the same functions as in the shift mode. The following symbolic diagram shows the flip-flop structure of this primitive.



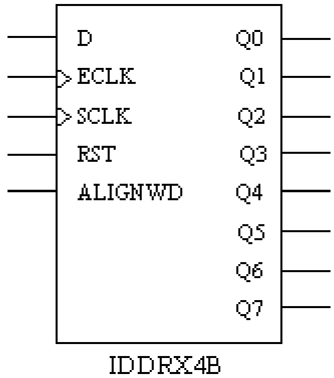
For more information, see related technical notes or contact technical support.

IDDRX4B

Input for Generic DDR X4 Using 1:8 Gearing

Architectures Supported:

- MachXO2
- Platform Manager 2



INPUTS: D, ECLK, SCLK, RST, ALIGNWD

OUTPUTS: Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

IDDRX4B is the input for generic DDR X4 using 1:8 gearing. It uses the VPIC_RX hardware cell. It is used for bottom bank only.

See the below table for IDDRX4B I/O description.

Signal	I/O	Description
D	I	DDR input from sysIO buffer.
ECLK	I	Clock connected to the high speed edge clock tree.
SCLK	I	Clock connected to the system clock.
RST	I	RESET to this block from CIB.
ALIGNWD	I	Data alignment signal used for word alignment. Each operation shifts alignment by 1 bit.
Q0, Q2, Q4, Q6	O	Data available at the same edge of the clock.
Q1, Q3, Q5, Q7	O	Data available at the same edge of the clock.

For more information and usage, refer to the following technical note on the Lattice web site.

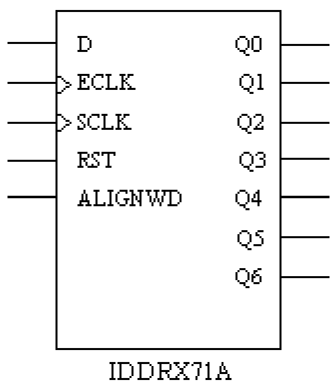
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

IDDRX71A

7:1 LVDS Input Supporting 1:7 Gearing

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, ECLK, SCLK, RST, ALIGNWD

OUTPUTS: Q0, Q1, Q2, Q3, Q4, Q5, Q6

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

IDDRX71A is the 7:1 LVDS input supporting 1:7 gearing. It is used for bottom bank only. IDDRX71A includes the SLIP circuitry.

See the below table for IDDRX71A I/O description.

Signal	I/O	Description
D	I	DDR input from sysIO buffer.
ECLK	I	Edge clock.
SCLK	I	Clock connected to the system clock.
RST	I	RESET for this block.

Signal	I/O	Description
ALIGNWD	I	Data alignment signal used for 7:1 LVDS. Each operation shifts alignment by 2 bits.
Q0, Q1, Q2, Q3, Q4, Q5, Q6	O	1:7 LVDS output signals.

For more information and usage, refer to the following technical note on the Lattice web site.

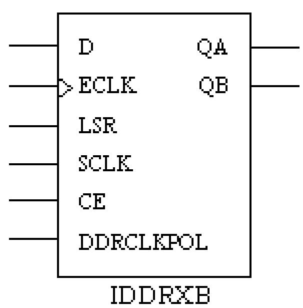
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

IDDRXB

Input DDR

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: D, ECLK, LSR, SCLK, CE, DDRCLKPOL

OUTPUTS: QA, QB

ATTRIBUTES:

REGSET: "RESET" (default), "SET"

Description

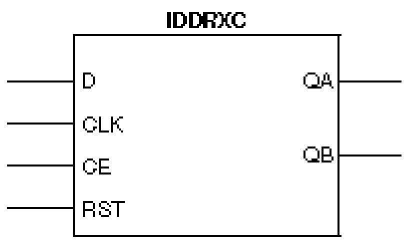
Double Data Rate input logic with half cycle clock domain transfer for the negative edge captured data (both edges of captured data enter core with positive edge flip-flops. For more information, see related technical notes or contact technical support.

IDDRXC

DDR Generic Input

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: D, CLK, RST, CE

OUTPUTS: QA, QB

Description

This primitive inputs the DDR data at both edges of the edge and generates two streams of data. CLK of this module can be connected to either the edge clock or the primary FPGA clock.

Note that the DDRCLKPOL input to IDDRXC should be connected to the DDRCLKPOL output of DQSBUFC. LSR is only for second stage register/ latch. To support DDR modes configured for bidirectional use, software will tie LSR LOW for input registers. The default for LSR is HIGH.

See the following table for port description of the IDDRXC primitive.

Port Name	I/O	Definition
D	I	DDR input from sysIO buffer
CLK	I	Clock from the CIB
RST	I	RESET to this block from CIB
CE	I	Clock enable signal
QA	O	Data at the positive edge of the CLK
QB	O	Data at the negative edge of the CLK

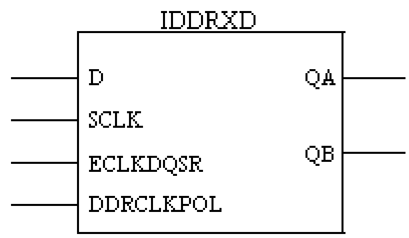
For more usage, see related technical notes or contact technical support.

IDDRXD

Input DDR for DDR_MEM, DDR2_MEM, DDR_GENX1, and DDR2_MEMGEN

Architectures Supported:

- LatticeECP3



INPUTS: D, SCLK, ECLKDQSR, DDRCLKPOL

OUTPUTS: QA, QB

ATTRIBUTES:

(*EA only*) SCLKLATENCY: 1 (default), 2

Description

IDDRXD is the input DDR for DDR_MEM, DDR2_MEM, DDR_GENX1 (DDR generic mode in X1 gearing), and DDR2_MEMGEN.

- E: DDR_MEM, DDR2_MEM, DDR_GENX1, and DDR2_MEMGEN (left/right/top)
- EA: DDR_MEM, DDR2_MEM, and DDR2_MEMGEN (left/right/top)

See the below table for its port description.

Signal	I/O	Description
D	I	DDR input from sysIO buffer.
SCLK	I	System clock.
ECLKDQSR	I	Phase shifted DQS in case of DDR memory interface. Connects to DQSBUF. For EA devices, ECLKDQSR should be used only for the DQS strobe.
DDRCLKPOL	I	DDR clock polarity signal.
QA	O	Data at the positive edge of the clock (mapped to IPB).
QB	O	Data at the negative edge of the clock (mapped to INB).

For more information and usage, refer to the following technical note on the Lattice web site.

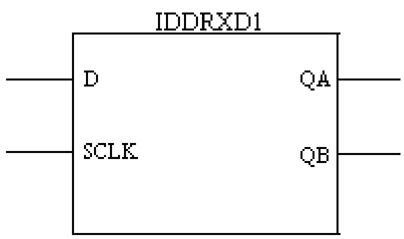
- TN1177 - LatticeECP3 sysIO Usage Guide

IDDRXD1

Input DDR for DDR_GENX1

Architectures Supported:

- LatticeECP3



INPUTS: D, SCLK

OUTPUTS: QA, QB

Description

IDDRXD1 is the input DDR for DDR_GENX1 (DDR generic mode in X1 gearing).

- EA: DDR_GENX1 (left/right/top)

See the below table for its port description.

Signal	I/O	Description
D	I	DDR input from sysIO buffer.
SCLK	I	System clock.
QA	O	Data at the positive edge of the clock (mapped to IPB).
QB	O	Data at the negative edge of the clock (mapped to INB).

For more information and usage, refer to the following technical note on the Lattice web site.

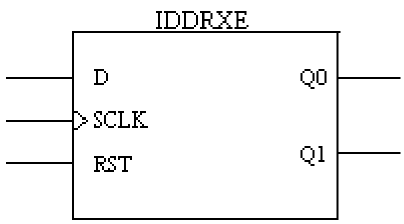
- TN1177 - LatticeECP3 sysIO Usage Guide

IDDRXE

Input for Generic DDR X1 Using 1:2 Gearing

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, SCLK, RST

OUTPUTS: Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

IDDRXE is the input for generic DDR X1 using 1:2 gearing. It uses the mPIC (base pic) or PIC (memory pic) hardware cell. It is used for all sides.

See the below table for port description.

Signal	I/O	Description
D	I	DDR input from sysIO buffer
SCLK	I	Clock connected to the system clock
RST	I	RESET for this block
Q0	O	Data at the positive edge of the clock
Q1	O	Data at the negative edge of the clock

For more information and usage, refer to the following technical note on the Lattice web site.

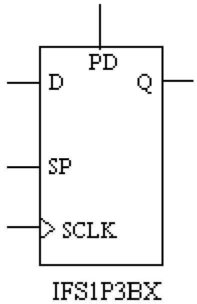
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

IFS1P3BX

Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in input PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, SP, SCLK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				OUTPUTS
D	SP	SCLK	PD	Q
X	0	X	0	Q
X	X	X	1	1
0	1	↑	0	0
1	1	↑	0	1

X = Don't care

When GSR=0, Q=1 (D=SP=SCLK=PD=X)

Note

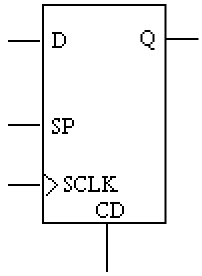
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IFS1P3DX

Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in input PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



IFS1P3DX

INPUTS: D, SP, SCLK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				**OUTPUTS**
D	SP	SCLK	CD	Q
X	0	X	0	Q
X	X	X	1	0
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

When GSR=0, Q=0 (D=SP=SCLK=CD=X)

Note

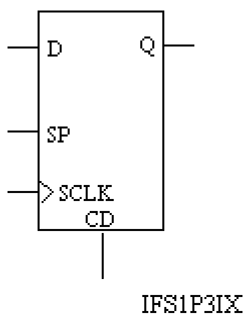
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IFS1P3IX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable, and System Clock (Clear overrides Enable) (used in input PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, SP, SCLK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				OUTPUTS
D	SP	SCLK	CD	Q
X	0	X	0	Q
X	X	↑	1	0
0	1	↑	0	0
1	1	↑	0	1

X = Don't care

When GSR=0, Q=0 (D=SP=SCLK=CD=X)

Note

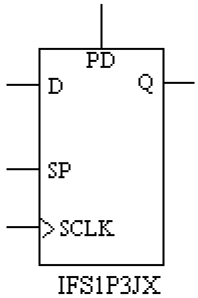
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IFS1P3JX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable, and System Clock (Preset overrides Enable) (used in input PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, SP, SCLK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				**OUTPUTS**
D	SP	SCLK	PD	Q
X	0	X	0	Q
X	X	↑	1	1
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

When GSR=0, Q=1 (D=SP=SCLK=PD=X)

Note

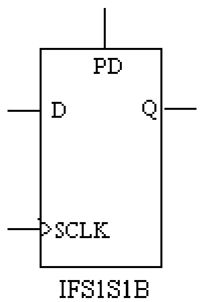
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IFS1S1B

Positive Level Data Latch with Positive Level Asynchronous Preset and System Clock (used in input PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, SCLK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS			OUTPUTS
D	SCLK	PD	Q
X	0	0	Q
X	X	1	1
0	1	0	0
1	1	0	1

X= Don't care

When GSR=0, Q=1 (D=SCLK=PD=X)

Note

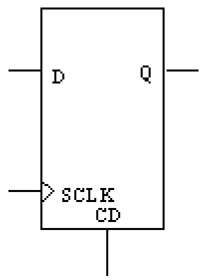
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IFS1S1D

Positive Level Data Latch with Positive Level Asynchronous Clear and System Clock (used in input PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



IFS1S1D

INPUTS: D, SCLK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS			**OUTPUTS**
D	SCLK	CD	Q
X	0	0	Q
X	X	1	0
0	1	0	0
1	1	0	1

X = Don’t care

When GSR=0, Q=0 (D=SCLK=CD=X)

Note

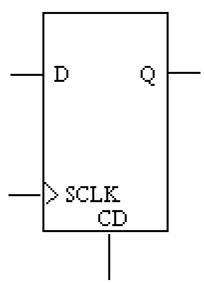
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IFS1S1I

Positive Level Data Latch with Positive Level Synchronous Clear and System Clock (used in input PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



IFS1S1I

INPUTS: D, SCLK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS			OUTPUTS
D	SCLK	CD	Q
X	0	0	Q
X	1	1	0
0	1	0	0
1	1	0	1

X = Don't care

When GSR=0, Q=0 (D=SCLK=CD=X)

Note

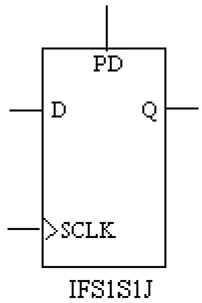
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IFS1S1J

Positive Level Data Latch with Positive Level Synchronous Preset and System Clock (used in input PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, SCLK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS			OUTPUTS
D	SCLK	PD	Q
X	0	0	Q
X	1	1	1
0	1	0	0
1	1	0	1

X = Don’t care

When GSR=0, Q=1 (D=SCLK=PD=X)

Note

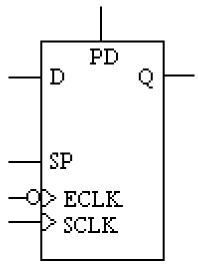
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

ILF2P3BX

Negative Level Edge Clocked (ECLK) Latch, Feeding Positive Edge Triggered System Clocked (SCLK) Flip-Flop, and Positive Level Asynchronous Preset (used in input PIC area only)

Architectures Supported:

- LatticeSC/M



ILF2P3BX

INPUTS: D, SP, ECLK, SCLK, PD

OUTPUT: Q

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS					OUTPUTS	
D	SP	ECLK	SCLK	PD	LATCH_Q	Q
X	X	X	X	1	LATCH_Q	1
X	0	1	X	0	LATCH_Q	Q
0	X	0	B	0	0	Q
1	X	0	B	0	1	Q
X	1	1	↑	0	0	0
X	1	1	↑	0	1	1
0	1	0	↑	0	0	0
1	1	0	↑	0	1	1

X = Don’t care

LATCH_Q = Output data from latch

B = Not rising edge

When GSR=0, Q=1 (D=SP=ECLK=SCLK=PD=X)

Note

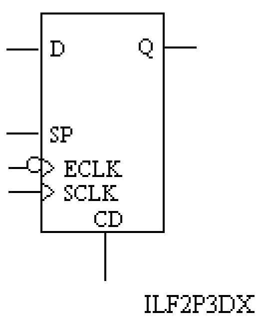
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

ILF2P3DX

Negative Level Edge Clocked (ECLK) Latch, Feeding Positive Edge Triggered System Clocked (SCLK) Flip-Flop, and Positive Level Asynchronous Clear (used in input PIC area only)

Architectures Supported:

- LatticeSC/M



INPUTS: D, SP, ECLK, SCLK, CD

OUTPUT: Q

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS					OUTPUTS	
D	SP	ECLK	SCLK	CD	LATCH_Q	Q
X	X	X	X	1	LATCH_Q	0
X	0	1	X	0	LATCH_Q	Q
0	X	0	B	0	0	Q
1	X	0	B	0	1	Q
X	1	1	↑	0	0	0
X	1	1	↑	0	1	1
0	1	0	↑	0	0	0
1	1	0	↑	0	1	1

X = Don't care

LATCH_Q = Output data from latch

B = Not rising edge

When GSR=0, Q=0 (D=SP=ECLK=SCLK=CD=X)

Note

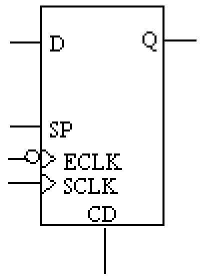
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

ILF2P3IX

Negative Level Edge Clocked (ECLK) Latch, Feeding Positive Edge Triggered System Clocked (SCLK) Flip-Flop, with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable) (used in input PIC area only)

Architectures Supported:

- LatticeSC/M



ILF2P3IX

INPUTS: D, SP, ECLK, SCLK, CD

OUTPUT: Q

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS					**OUTPUTS**	
D	SP	ECLK	SCLK	CD	LATCH_Q	Q
X	X	X	$\uparrow$	1	LATCH_Q	0
X	0	1	X	0	LATCH_Q	Q
0	X	0	B	0	0	Q
1	X	0	B	0	1	Q
X	1	1	$\uparrow$	0	0	0
X	1	1	$\uparrow$	0	1	1
0	1	0	$\uparrow$	0	0	0
1	1	0	$\uparrow$	0	1	1

X = Don’t care

LATCH_Q = Output data from latch

B = Not rising edge

When GSR = 0, Q = 0 (D = SP = ECLK = SCLK = CD = X)

Note

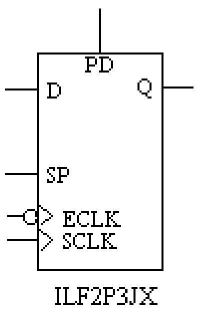
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

ILF2P3JX

Negative Level Edge Clocked (ECLK) Latch, Feeding Positive Edge Triggered System Clocked (SCLK) Flip-Flop, and Positive Level Synchronous Preset (Preset overrides Enable) (used in input PIC area only)

Architectures Supported:

- LatticeSC/M



INPUTS: D, SP, ECLK, SCLK, PD

OUTPUT: Q

Description

This primitive must be paired with an input or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS					OUTPUTS	
D	SP	ECLK	SCLK	PD	LATCH_Q	Q
X	X	X	↑	1	LATCH_Q	1
X	0	1	X	0	LATCH_Q	Q
0	X	0	B	0	0	Q
1	X	0	B	0	1	Q
X	1	1	↑	0	0	0
X	1	1	↑	0	1	1
0	1	0	↑	0	0	0
1	1	0	↑	0	1	1

X = Don't care

LATCH_Q = Output data from latch

B = Not rising edge

When GSR = 0, Q = 1 (D = SP = ECLK = SCLK = PD = X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

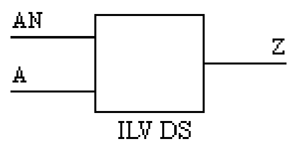
ILVDS

LVDS Input Buffer

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L

- Platform Manager
- Platform Manager 2



INPUTS: A, AN

OUTPUT: Z

Truth Table

INPUTS		OUTPUTS
A	AN	Z
0	1	0
1	0	1

Note

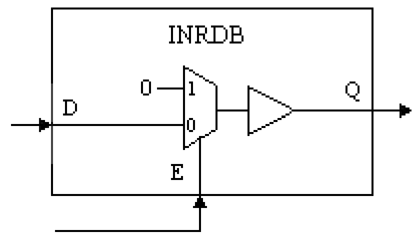
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

INRDB

Input Reference and Differential Buffer

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, E

OUTPUT: Q

Description

The INRDB primitive is used to support post-PAR simulation of Dynamic Bank Controller. The Dynamic Bank Controller signals to the IO are hardwired and cannot be changed for the simulation, so the INRDB and LVDSOB primitives are defined to support the simulation.

For more information, refer to the following technical note on the Lattice web site:

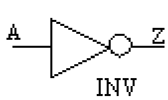
- TN1198 - Power Estimation and Management for MachXO2 Device

INV

Inverter

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUT: A

OUTPUT: Z

Note

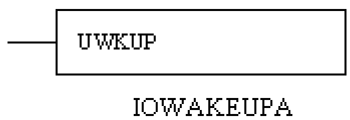
- It is possible that this primitive will be optimized away.
- This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

IOWAKEUPA

XP2 Wake-up Controller

Architectures Supported:

- LatticeXP2



INPUT: UWKUP

Description

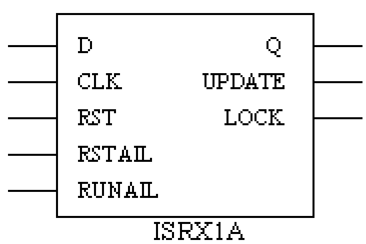
LatticeXP2 Wake-up controller.

ISRX1A

Input 1-Bit Shift Register

Architectures Supported:

- LatticeSC/M

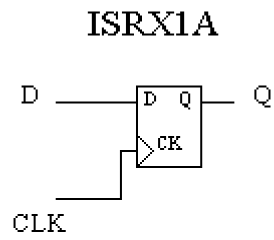


INPUTS: D, CLK, RST, RSTAIL, RUNAIL

OUTPUTS: Q, UPDATE, LOCK

Description

Shift register input logic that uses adaptive FF to capture input data. The following symbolic diagram shows the flip-flop structure of this primitive.

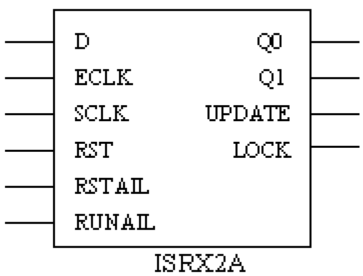


ISRX2A

Input 2-Bit Shift Register

Architectures Supported:

- LatticeSC/M

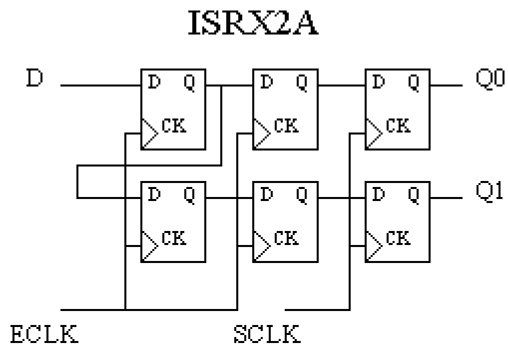


INPUTS: D, ECLK, SCLK, RST, RSTAIL, RUNAIL

OUTPUTS: Q0, Q1, UPDATE, LOCK

Description

Shift register input logic that allows clock domain transfer from edge clock to primary clock and parallel transfer to the core of incoming serial data. The following symbolic diagram shows the flip-flop structure of this primitive.

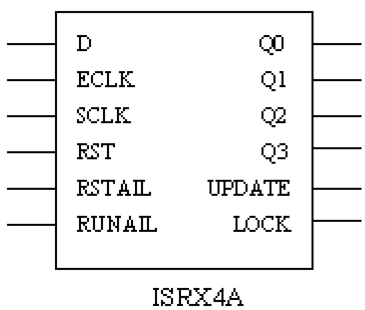


ISRX4A

Input 4-Bit Shift Register

Architectures Supported:

- LatticeSC/M

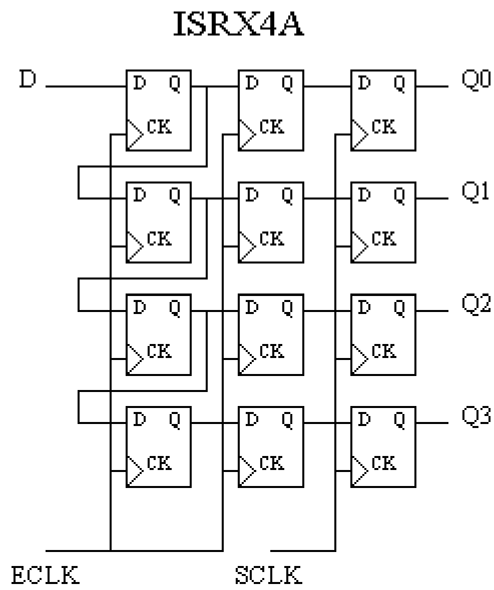


INPUTS: D, ECLK, SCLK, RST, RSTAIL, RUNAIL

OUTPUTS: Q0, Q1, Q2, Q3, UPDATE, LOCK

Description

Shift register input logic that allows clock domain transfer from edge clock to primary clock and parallel transfer to the core of incoming serial data. The following symbolic diagram shows the flip-flop structure of this primitive.


ISRX4A
D
Q0
Q1
Q2
Q3
ECLK
SCLK

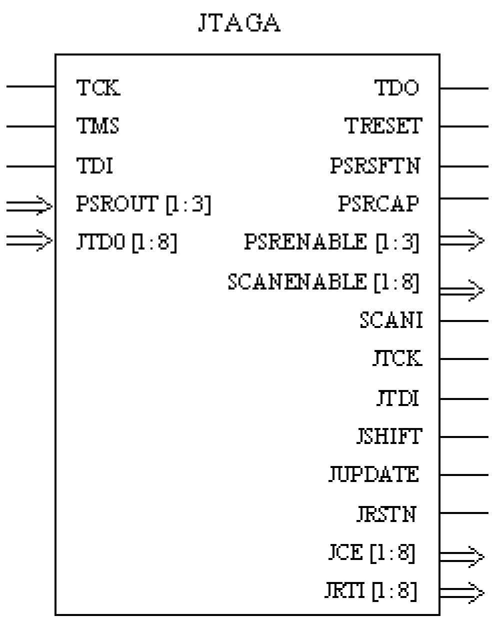
J

JTAGA

JTAG (Joint Test Action Group) Controller

Architectures Supported:

- LatticeSC/M



INPUTS: TCK, TMS, TDI, PSROUT1, PSROUT2, PSROUT3, JTDO1, JTDO2, JTDO3, JTDO4, JTDO5, JTDO6, JTDO7, JTDO8

OUTPUTS: TDO, TRESET, PSRSFTN, PSRCAP, PSRENABLE1, PSRENABLE2, PSRENABLE3, SCANENABLE1, SCANENABLE2, SCANENABLE3, SCANENABLE4, SCANENABLE5, SCANENABLE6, SCANENABLE7, SCANENABLE8, SCANI, JTCK, JTDI, JSHIFT, JUPDATE, JRSTN, JCE1, JCE2, JCE3, JCE4, JCE5, JCE6, JCE7, JCE8, JRTI1, JRTI2, JRTI3, JRTI4, JRTI5, JRTI6, JRTI7, JRTI8

Description

The JTAGA primitive provides the control and interconnect circuit used by the boundary scan function. This function allows the testing of increasingly complex ICs and IC packages. The LatticeSC/M device has enhanced its interface capability to the PLC array with increased scan chain connectivity and tap state machine flags such as shift capture update, reset, run test idle.

Example pin functions:

Pins	I/O	Function	Description
TCK TMS TDI TDO	I I I O	interface pins	Test clock (TCK), Test mode select (TMS), Test data in (TDI) and Test data out (TDO) are four interface pins for this primitive. The TDI, TMS and TCK pins are connected to the dedicated IO pads on the device.
PSROUT[1:3] JTDO[1:8]	I I	user boundary scan-ring outputs	These are the outputs of the last registers of the user scan rings to the boundary scan block. Inputs to the boundary scan macro are based on the instruction loaded.
TRESET	O	reset	Active high output of the boundary scan macro to the routing. The output is high when the boundary scan macro is in test logic reset state.
PSRSFTN	O	shift_not data register	Active low output of the boundary scan macro. The output is low when the boundary scan macro is in the shift data state and the programmable scan ring instructions are loaded.
PSRCAP	O	capture data register	Active high output of the boundary scan macro. The output is high when the boundary scan macro is in the capture data state and the programmable scan ring instructions are loaded.
PSRENABLE[1:3] SCANENABLE[1:8]	O	enable flag	Active high outputs of the boundary scan macro to the routing. The output equals a high upon update of the specific instructions, PLC_SCAN_RING[1:3] and SCAN[1:8], respectively.
SCANI	O	scan in	Private pin used for testing of the system bus that multiplexes the TDI and SCANOUT[11:14].
JTCK	O	test clock	The boundary scan clock which is output from the boundary scan macro to the scan rings.
JTDI	O	test data in	The output of the boundary scan macro from where the test data is output to the scan rings.
JSHIFT	O	shift data register	Active high output of the boundary scan macro. The output is high when the boundary scan macro is in the shift data state and the scan instructions are loaded.
JUPDATE	O	update data register	Active high output of the boundary scan macro. The output is high when the boundary scan macro is in the update data state and when the PLC_SCAN_RING or SCAN instructions are loaded.

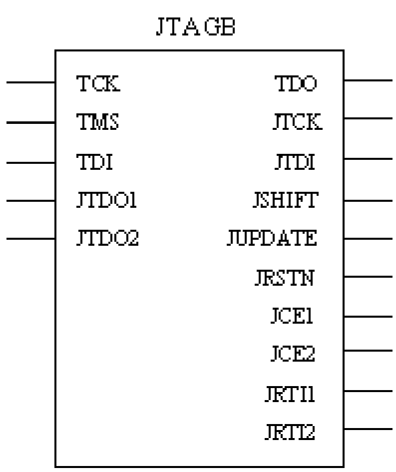
Pins	I/O	Function	Description
JRSTN	O	reset_not	Active low output of the boundary scan macro to the routing. The output is low when the boundary scan macro is in test logic reset state.
JCE[1:8]	O	clock enable	Active high output of the boundary scan macro to the routing. The output is high when the boundary scan macro is in the SHIFT or CAPTURE state during the SCAN[1:8] instructions, respectively.
JRTI[1:8]	O	run test idle	Active high output of the boundary scan macro. The output is high when the boundary scan macro is in the run test idle state and during the SCAN[1:8] instructions.

JTAGB

JTAG (Joint Test Action Group) Controller

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: TCK, TMS, TDI, JTDO1, JTDO2

OUTPUTS: TDO, JTCK, JTDI, JSHIFT, JUPDATE, JRSTN, JCE1, JCE2, JRTI1, JRTI2

ATTRIBUTES:

ER1: "ENABLED" (default), "DISABLED"

ER2: "ENABLED" (default), "DISABLED"

Description

Signal	I/O	Description
TCK	I	Test Clock, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TCK pin. Clocks registers and TAP Controller.
TMS	I	Test Mode Select, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TMS pin. Controls state machine switching for TAP Controller.
TDI	I	Test Data Input, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TDI pin.
JTDO[2:1]	I	JTAG Test Data Output one (scans output bus entering JTAG block), for internal logic to control non-disruptive re-configuration through JTAG port, JTAG serial interface to the device. Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO1. If ER2 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO2.
TDO	O	Test Data Output, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly to the pad of device TDO pin
JTCK	O	JTAG Test Clock (connects to TCK), for internal logic to control non-disruptive re-configuration through JTAG port. Signal is coming from TCK input and going into the FPGA fabric.
JTDI	O	JTAG Test Data Input, for internal logic to control non-disruptive re-configuration through JTAG port. Signal is coming from TDI input and going into the FPGA fabric.
JSHIFT	O	JTAG Shift. Signal goes high when TAP Controller State is Shift-DR.
JUPDATE	O	JTAG Update. Signal goes high when TAP controller state is Update-DR.
JRSTN	O	JTAG Reset (active low). Signal goes low when TAP controller state is Test-Logic-Reset.

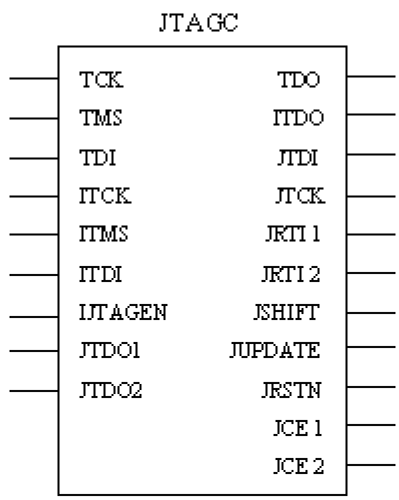
Signal	I/O	Description
JCE[2:1]	O	JTAG Clock Enable one (BS is to boundary scan ring (L2). Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JCE1 will go high when TAP controller is in Capture-DR or Shift-DR states. If ER2 instruction is shifted into the JTAG instruction register, JCE2 will go high when TAP controller is in Capture-DR or Shift-DR states.
JRTI[2:1]	O	JTAG Run-Test/Idle. Lattice supports two private JTAG instructions ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JRTI1 will go high when TAP controller is in Run-Test/Idle state. If ER2 instruction is shifted into the JTAG instruction register, JRTI2 will go high when TAP controller is in Run-Test/Idle state.

JTAGC

JTAG (Joint Test Action Group) Controller

Architectures Supported:

- LatticeECP2/M



INPUTS: TCK, TMS, TDI, ITCK, ITMS, ITDI, IJTAGEN, JTDO1, JTDO2

OUTPUTS: TDO, ITDO, JTDI, JTCK, JRTI1, JRTI2, JSHIFT, JUPDATE, JRSTN, JCE1, JCE2

Note

- The internal JTAG mode is not supported. The ITCK, ITMS, ITDI, and ITDO ports are non-operations and hence do not use them.
- The IJTAGEN port should always be tied to VCC.

ATTRIBUTES:

ER1: "ENABLED" (default), "DISABLED"

ER2: "ENABLED" (default), "DISABLED"

Description

Signal	I/O	Description
TCK	I	Test Clock, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TCK pin. Clocks registers and TAP Controller.
TMS	I	Test Mode Select, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TMS pin. Controls state machine switching for TAP Controller.
TDI	I	Test Data Input, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TDI pin.
ITCK	I	Internal Test Clock for internal logic to control non-disruptive re-configuration through JTAG port, comes from configuration block CIB.
ITMS	I	Internal Test Mode Select, for internal logic to control non-disruptive re-configuration through JTAG port, comes from configuration block CIB.
ITDI	I	Internal Test Data Input, for internal logic to control non-disruptive re-configuration through JTAG port, comes from configuration block CIB.
IJTAGEN	I	Internal JTAG Enable (active low), for internal logic to control non-disruptive re-configuration through JTAG port, comes from configuration block CIB.
JTDO[2:1]	I	JTAG Test Data Output one (scans output bus entering JTAG block), for internal logic to control non-disruptive re-configuration through JTAG port, JTAG serial interface to the device. Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO1. If ER2 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO2.
TDO	O	Test Data Output, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly to the pad of device TDO pin.

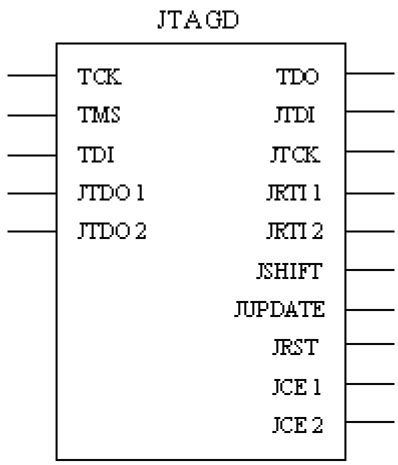
Signal	I/O	Description
ITDO	O	Internal Test Data Output for internal logic to control non-disruptive re-configuration through JTAG port, comes from configuration block CIB.
JTDI	O	JTAG Test Data Input, for internal logic to control non-disruptive re-configuration through JTAG port Signal is coming from TDI input and going into the FPGA fabric.
JTCK	O	JTAG Test Clock (connects to TCK), for internal logic to control non-disruptive re-configuration through JTAG port Signal is coming from TCK input and going into the FPGA fabric.
JRTI[2:1]	O	JTAG Run-Test/Idle. Lattice supports two private JTAG instructions ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JRTI1 will go high when TAP controller is in Run-Test/Idle state. If ER2 instruction is shifted into the JTAG instruction register, JRTI2 will go high when TAP controller is in Run-Test/Idle state.
JSHIFT	O	JTAG Shift. Signal goes high when TAP Controller State is Shift-DR.
JUPDATE	O	JTAG Update. Signal goes high when TAP controller state is Update-DR.
JRSTN	O	JTAG Reset (active low). Signal goes low when TAP controller state is Test-Logic-Reset.
JCE[2:1]	O	JTAG Clock Enable one (BS is to boundary scan ring (L2). Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JCE1 will go high when TAP controller is in Capture-DR or Shift-DR states. If ER2 instruction is shifted into the JTAG instruction register, JCE2 will go high when TAP controller is in Capture-DR or Shift-DR states.

JTAGD

JTAG (Joint Test Action Group) Controller

Architectures Supported:

- MachXO
- Platform Manager



INPUTS: TCK, TMS, TDI, JTDO1, JTDO2

OUTPUTS: TDO, JTDI, JTCK, JRTI1, JRTI2, JSHIFT, JUPDATE, JRST, JCE1, JCE2

ATTRIBUTES:

ER1: "ENABLED" (default), "DISABLED"

ER2: "ENABLED" (default), "DISABLED"

Description

Signal	I/O	Description
TCK	I	Test Clock, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TCK pin Clocks registers and TAP Controller
TMS	I	Test Mode Select, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TMS pin Controls state machine switching for TAP Controller
TDI	I	Test Data Input, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TDI pin
JTDO[2:1]	I	JTAG Test Data Output one (scans output bus entering JTAG block), for internal logic to control non-disruptive re-configuration through JTAG port, JTAG serial interface to the device. Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO1. If ER2 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO2.

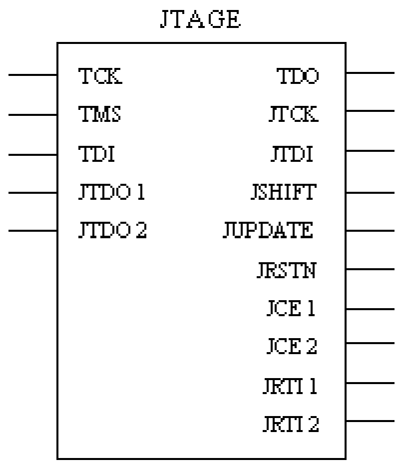
Signal	I/O	Description
TDO	O	Test Data Output, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly to the pad of device TDO pin
JTDI	O	JTAG Test Data Input, for internal logic to control non-disruptive re-configuration through JTAG port Signal is coming from TDI input and going into the FPGA fabric.
JTCK	O	JTAG Test Clock (connects to TCK), for internal logic to control non-disruptive re-configuration through JTAG port Signal is coming from TCK input and going into the FPGA fabric.
JRTI[2:1]	O	JTAG Run-Test/Idle Lattice supports two private JTAG instructions ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JRTI1 will go high when TAP controller is in Run-Test/Idle state. If ER2 instruction is shifted into the JTAG instruction register, JRTI2 will go high when TAP controller is in Run-Test/Idle state.
JSHIFT	O	JTAG Shift Signal goes high when TAP Controller State is Shift-DR.
JUPDATE	O	JTAG Update Signal goes high when TAP controller state is Update-DR.
JRST	O	JTAG Reset (active high) Signal goes high when TAP controller state is Test-Logic-Reset.
JCE[2:1]	O	JTAG Clock Enable one (BS is to boundary scan ring (L2). Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JCE1 will go high when TAP controller is in Capture-DR or Shift-DR states. If ER2 instruction is shifted into the JTAG instruction register, JCE2 will go high when TAP controller is in Capture-DR or Shift-DR states.

JTAGE

JTAG (Joint Test Action Group) Controller

Architectures Supported:

- LatticeECP3
- LatticeXP2



INPUTS: TCK, TMS, TDI, JTDO1, JTDO2

OUTPUTS: TDO, JTCK, JTDI, JSHIFT, JUPDATE, JRSTN, JCE1, JCE2, JRTI1, JRTI2

ATTRIBUTES:

ER1: "ENABLED" (default), "DISABLED"

ER2: "ENABLED" (default), "DISABLED"

Description

Signal	I/O	Description
TCK	I	Test Clock, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TCK pin. Clocks registers and TAP Controller.
TMS	I	Test Mode Select, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TMS pin. Controls state machine switching for TAP Controller.
TDI	I	Test Data Input, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TDI pin.
JTDO[2:1]	I	JTAG Test Data Output one (scans output bus entering JTAG block), for internal logic to control non-disruptive re-configuration through JTAG port, JTAG serial interface to the device. Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO1. If ER2 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO2.

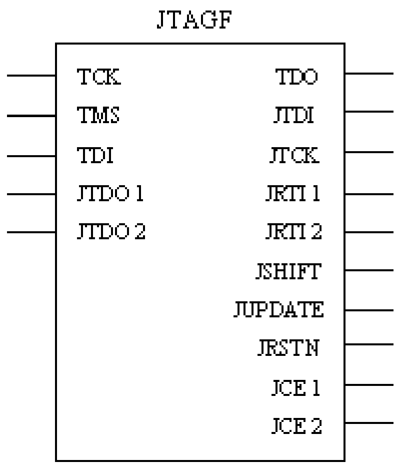
Signal	I/O	Description
TDO	O	Test Data Output, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly to the pad of device TDO pin.
JTDI	O	JTAG Test Data Input, for internal logic to control non-disruptive re-configuration through JTAG port Signal is coming from TDI input and going into the FPGA fabric.
JTCK	O	JTAG Test Clock (connects to TCK), for internal logic to control non-disruptive re-configuration through JTAG port Signal is coming from TCK input and going into the FPGA fabric.
JRTI[2:1]	O	JTAG Run-Test/Idle. Lattice supports two private JTAG instructions ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JRTI1 will go high when TAP controller is in Run-Test/Idle state. If ER2 instruction is shifted into the JTAG instruction register, JRTI2 will go high when TAP controller is in Run-Test/Idle state.
JSHIFT	O	JTAG Shift. Signal goes high when TAP Controller State is Shift-DR.
JRSTN	O	JTAG Reset (active low) Signal goes low when TAP controller state is Test-Logic-Reset.
JUPDATE	O	JTAG Update. Signal goes high when TAP controller state is Update-DR.
JCE[2:1]	O	JTAG Clock Enable one (BS is to boundary scan ring (L2). Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JCE1 will go high when TAP controller is in Capture-DR or Shift-DR states. If ER2 instruction is shifted into the JTAG instruction register, JCE2 will go high when TAP controller is in Capture-DR or Shift-DR states.

JTAGF

JTAG (Joint Test Action Group) Controller

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: TCK, TMS, TDI, JTDO1, JTDO2

OUTPUTS: TDO, JTDI, JTCK, JRTI1, JRTI2, JSHIFT, JUPDATE, JRSTN, JCE1, JCE2

ATTRIBUTES:

ER1: "ENABLED" (default), "DISABLED"

ER2: "ENABLED" (default), "DISABLED"

Description

The JTAGF primitive is used to provide access to internal JTAG signals from within the FPGA fabric. This is used for some cores, such as REVEAL, and other purposes. It is not a component an ordinary user would normally use directly.

Signal	I/O	Description
TCK	I	Test Clock, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TCK pin. Clocks registers and TAP Controller.
TMS	I	Test Mode Select, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TMS pin. Controls state machine switching for TAP Controller.
TDI	I	Test Data Input, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly from the pad of device TDI pin.

Signal	I/O	Description
JTDO[2:1]	I	JTAG Test Data Output one (scans output bus entering JTAG block), for internal logic to control non-disruptive re-configuration through JTAG port, JTAG serial interface to the device. Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO1. If ER2 instruction is shifted into the JTAG instruction register, TDO output will come from JTDO2.
TDO	O	Test Data Output, one of the pins for the Test Access Port (TAP), JTAG serial interface to the device, connected directly to the pad of device TDO pin.
JTDI	O	JTAG Test Data Input, for internal logic to control non-disruptive re-configuration through JTAG port Signal is coming from TDI input and going into the FPGA fabric.
JTCK	O	JTAG Test Clock (connects to TCK), for internal logic to control non-disruptive re-configuration through JTAG port Signal is coming from TCK input and going into the FPGA fabric.
JRTI[2:1]	O	JTAG Run-Test/Idle. Lattice supports two private JTAG instructions ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JRTI1 will go high when TAP controller is in Run-Test/Idle state. If ER2 instruction is shifted into the JTAG instruction register, JRTI2 will go high when TAP controller is in Run-Test/Idle state.
JSHIFT	O	JTAG Shift. Signal goes high when TAP Controller State is Shift-DR.
JRSTN	O	JTAG Reset (active low) Signal goes low when TAP controller state is Test-Logic-Reset.
JUPDATE	O	JTAG Update. Signal goes high when TAP controller state is Update-DR.
JCE[2:1]	O	JTAG Clock Enable one (BS is to boundary scan ring (L2). Lattice supports two private JTAG instructions, ER1 (0x32) and ER2 (0x38). If ER1 instruction is shifted into the JTAG instruction register, JCE1 will go high when TAP controller is in Capture-DR or Shift-DR states. If ER2 instruction is shifted into the JTAG instruction register, JCE2 will go high when TAP controller is in Capture-DR or Shift-DR states.

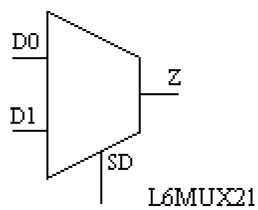
L

L6MUX21

2 to 1 Mux

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, SD

OUTPUT: Z

Truth Table

INPUTS			OUTPUTS
D0	D1	SD	Z
0	X	0	0
1	X	0	1
X	0	1	0
X	1	1	1

Note

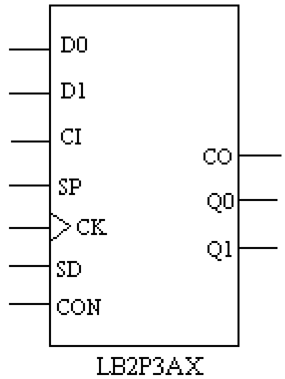
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LB2P3AX

2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, CON

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:1]	SD	CI	SP	CK	CON	CO	Q[0:1]
D[0:1]	1	0	1	↑	1	0	D[0:1]
D[0:1]	1	1	1	↑	1	*	D[0:1]
X	0	0	X	X	1	0	Q[0:1]
X	X	0	0	X	1	0	Q[0:1]
X	X	1	0	X	1	*	Q[0:1]
X	0	1	1	↑	1	*	count+1
D[0:1]	1	1	1	↑	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	**	D[0:1]
X	0	1	X	X	0	1	Q[0:1]
X	X	1	0	X	0	1	Q[0:1]
X	X	0	0	X	0	**	Q[0:1]
X	0	0	1	↑	0	**	count-1

X = Don't care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

** When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=!CON•CI, Q[0:1]=0 (D[0:1]=SP=CK=SD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

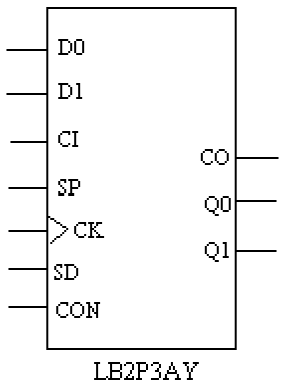
LB2P3AY

2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M

- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, CON

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:1]	SD	CI	SP	CK	CON	CO	Q[0:1]
D[0:1]	1	0	1	↑	1	0	D[0:1]
D[0:1]	1	1	1	↑	1	*	D[0:1]
X	0	0	X	X	1	0	Q[0:1]
X	X	0	0	X	1	0	Q[0:1]
X	X	1	0	X	1	*	Q[0:1]
X	0	1	1	↑	1	*	count+1
D[0:1]	1	1	1	↑	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	**	D[0:1]
X	0	1	X	X	0	1	Q[0:1]
X	X	1	0	X	0	1	Q[0:1]
X	X	0	0	X	0	**	Q[0:1]
X	0	0	1	↑	0	**	count-1

X = Don't care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

** When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, Q[0:1]=1, CO=!CON+CI (D[0:1]=SP=CK=SD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

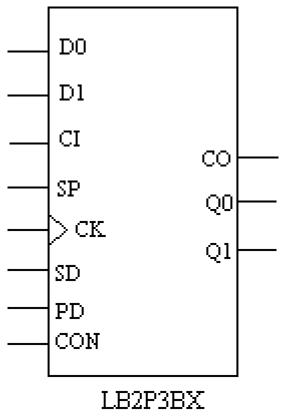
LB2P3BX

2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M

- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, PD, CON

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS							OUTPUTS	
D[0:1]	SD	CI	SP	CK	CON	PD	CO	Q[0:1]
X	X	0	X	X	1	1	0	1
X	X	1	X	X	1	1	1	1
D[0:1]	1	0	1	↑	1	0	0	D[0:1]
D[0:1]	1	1	1	↑	1	0	*	D[0:1]
X	0	0	X	X	1	0	0	Q[0:1]
X	X	0	0	X	1	0	0	Q[0:1]
X	X	1	0	X	1	0	*	Q[0:1]
X	0	1	1	↑	1	0	*	count+1
X	X	X	X	X	0	1	1	1
D[0:1]	1	1	1	↑	0	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	0	**	D[0:1]
X	0	1	X	X	0	0	1	Q[0:1]
X	X	1	0	X	0	0	1	Q[0:1]
X	X	0	0	X	0	0	**	Q[0:1]
X	0	0	1	↑	0	0	**	count-1

X = Don't care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

** When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=!CON+CI, Q[0:1]=1 (D[0:1]=SP=CK=SD=PD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

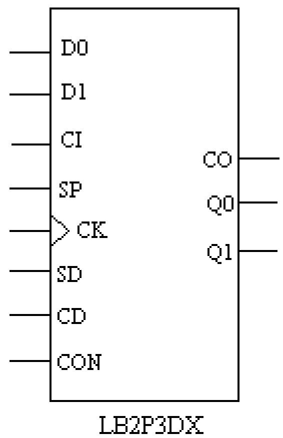
LB2P3DX

2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC

- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, CD, CON

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS							OUTPUTS	
D[0:1]	SD	CI	SP	CK	CON	CD	CO	Q[0:1]
X	X	X	X	X	1	1	0	0
D[0:1]	1	0	1	↑	1	0	0	D[0:1]
D[0:1]	1	1	1	↑	1	0	*	D[0:1]
X	0	0	X	X	1	0	0	Q[0:1]
X	X	0	0	X	1	0	0	Q[0:1]
X	X	1	0	X	1	0	*	Q[0:1]
X	0	1	1	↑	1	0	*	count+1
X	X	0	X	X	0	1	0	0
X	X	1	X	X	0	1	1	0
D[0:1]	1	1	1	↑	0	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	0	**	D[0:1]
X	0	1	X	X	0	0	1	Q[0:1]
X	X	1	0	X	0	0	1	Q[0:1]
X	X	0	0	X	0	0	**	Q[0:1]
X	0	0	1	↑	0	0	**	count-1

X = Don’t care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

** When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=!CON"CI, Q[0:1]=0 (D[0:1]=SP=CK=SD=CD=X)

Note

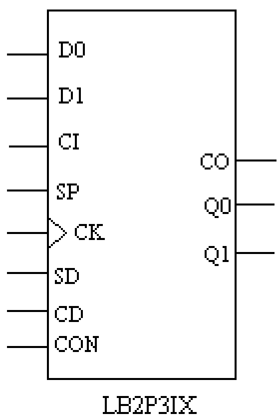
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LB2P3IX

2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, CD, CON

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS							OUTPUTS	
D[0:1]	SD	CI	SP	CK	CON	CD	CO	Q[0:1]
X	X	X	X	↑	1	1	0	0
D[0:1]	1	0	1	↑	1	0	0	D[0:1]
D[0:1]	1	1	1	↑	1	0	*	D[0:1]
X	0	0	X	X	1	0	0	Q[0:1]
X	X	0	0	X	1	0	0	Q[0:1]
X	X	1	0	X	1	0	*	Q[0:1]
X	0	1	1	↑	1	0	*	count+1
X	X	0	X	↑	0	1	0	0
X	X	1	X	↑	0	1	1	0
D[0:1]	1	1	1	↑	0	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	0	**	D[0:1]
X	0	1	X	X	0	0	1	Q[0:1]
X	X	1	0	X	0	0	1	Q[0:1]
X	X	0	0	X	0	0	**	Q[0:1]
X	0	0	1	↑	0	0	**	count-1

X = Don’t care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

** When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=!CON"CI, Q[0:1]=0 (D[0:1]=SP=CK=SD=CD=X)

Note

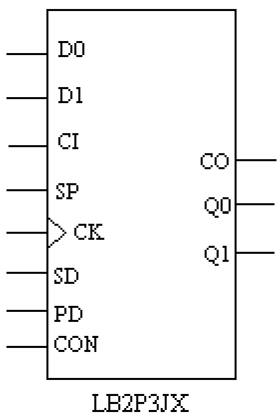
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LB2P3JX

2 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, PD, CON

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS							OUTPUTS	
D[0:1]	SD	CI	SP	CK	CON	PD	CO	Q[0:1]
X	X	0	X	↑	1	1	0	1
X	X	1	X	↑	1	1	1	1
D[0:1]	1	0	1	↑	1	0	0	D[0:1]
D[0:1]	1	1	1	↑	1	0	*	D[0:1]
X	0	0	X	X	1	0	0	Q[0:1]
X	X	0	0	X	1	0	0	Q[0:1]
X	X	1	0	X	1	0	*	Q[0:1]
X	0	1	1	↑	1	0	*	count+1
X	X	X	X	↑	0	1	1	1
D[0:1]	1	1	1	↑	0	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	0	**	D[0:1]
X	0	1	X	X	0	0	1	Q[0:1]
X	X	1	0	X	0	0	1	Q[0:1]
X	X	0	0	X	0	0	**	Q[0:1]
X	0	0	1	↑	0	0	**	count-1

X = Don't care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

** When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=!CON+CI, Q[0:1]=1 (D[0:1]=SP=CK=SD=PD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

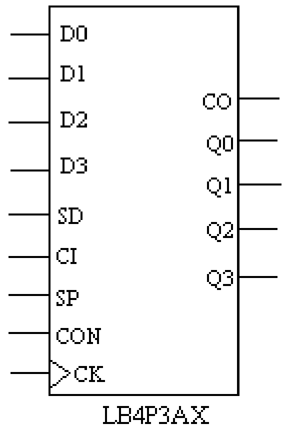
LB4P3AX

4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC

- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, CON

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS					OUTPUTS		
D[0:3]	SD	CI	SP	CK	CON	CO	Q[0:3]
D[0:3]	1	0	1	↑	1	0	D[0:3]
D[0:3]	1	1	1	↑	1	*	D[0:3]
X	0	0	X	X	1	0	Q[0:3]
X	X	0	0	X	1	0	Q[0:3]
X	X	1	0	X	1	*	Q[0:3]
X	0	1	1	↑	1	*	count+1
D[0:3]	1	1	1	↑	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	**	D[0:3]
X	0	1	X	X	0	1	Q[0:3]
X	X	1	0	X	0	1	Q[0:3]
X	X	0	0	X	0	**	Q[0:3]
X	0	0	1	↑	0	**	count-1

X = Don’t care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

** When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

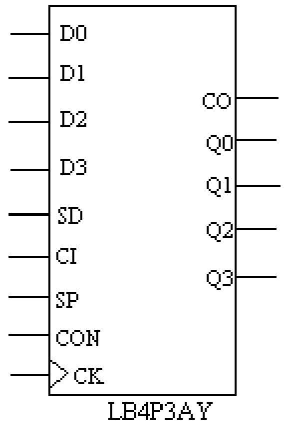
When GSR=0, CO=!CON•CI, Q[0:3]=0 (D[0:3]=SP=CK=SD=X)

LB4P3AY

4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, CON

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:3]	SD	CI	SP	CK	CON	CO	Q[0:3]
D[0:3]	1	0	1	↑	1	0	D[0:3]
D[0:3]	1	1	1	↑	1	*	D[0:3]
X	0	0	X	X	1	0	Q[0:3]
X	X	0	0	X	1	0	Q[0:3]
X	X	1	0	X	1	*	Q[0:3]
X	0	1	1	↑	1	*	count+1
D[0:3]	1	1	1	↑	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	**	D[0:3]
X	0	1	X	X	0	1	Q[0:3]
X	X	1	0	X	0	1	Q[0:3]
X	X	0	0	X	0	**	Q[0:3]
X	0	0	1	↑	0	**	count-1

X = Don’t care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

** When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

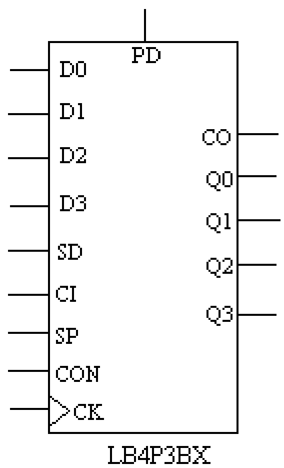
When GSR=0, Q[0:3]=1, CO=!CON+CI (D[0:3]=SP=CK=SD=X)

LB4P3BX

4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, PD, CON

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS							OUTPUTS	
D[0:3]	SD	CI	SP	CK	CON	PD	CO	Q[0:3]
X	X	0	X	X	1	1	0	1
X	X	1	X	X	1	1	1	1
D[0:3]	1	0	1	↑	1	0	0	D[0:3]
D[0:3]	1	1	1	↑	1	0	*	D[0:3]
X	0	0	X	X	1	0	0	Q[0:3]
X	X	0	0	X	1	0	0	Q[0:3]
X	X	1	0	X	1	0	*	Q[0:3]
X	0	1	1	↑	1	0	*	count+1
X	X	X	X	X	0	1	1	1
D[0:3]	1	1	1	↑	0	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	0	**	D[0:3]
X	0	1	X	X	0	0	1	Q[0:3]
X	X	1	0	X	0	0	1	Q[0:3]
X	X	0	0	X	0	0	**	Q[0:3]
X	0	0	1	↑	0	0	**	count-1

X = Don’t care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

** When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

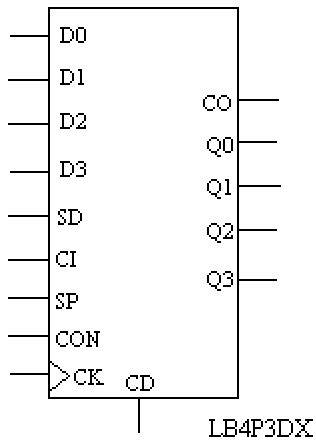
When GSR=0, CO=!CON+CI, Q[0:3]=1 (D[0:3]=SP=CK=SD=PD=X)

LB4P3DX

4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, CD, CON

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS							OUTPUTS	
D[0:3]	SD	CI	SP	CK	CON	CD	CO	Q[0:3]
X	X	X	X	X	1	1	0	0
D[0:3]	1	0	1	↑	1	0	0	D[0:3]
D[0:3]	1	1	1	↑	1	0	*	D[0:3]
X	0	0	X	X	1	0	0	Q[0:3]
X	X	0	0	X	1	0	0	Q[0:3]
X	X	1	0	X	1	0	*	Q[0:3]
X	0	1	1	↑	1	0	*	count+1
X	X	0	X	X	0	1	0	0
X	X	1	X	X	0	1	1	0
D[0:3]	1	1	1	↑	0	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	0	**	D[0:3]
X	0	1	X	X	0	0	1	Q[0:3]
X	X	1	0	X	0	0	1	Q[0:3]
X	X	0	0	X	0	0	**	Q[0:3]
X	0	0	1	↑	0	0	**	count-1

X = Don’t care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

** When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

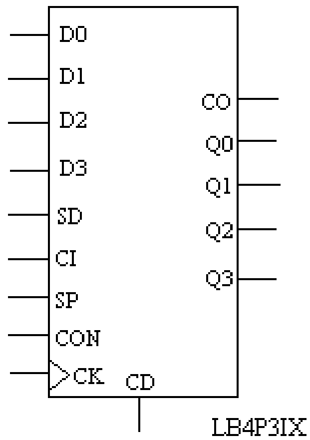
When GSR=0, CO=!CON"CI, Q[0:3]=0 (D[0:3]=SP=CK=SD=CD=X)

LB4P3IX

4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, CD, CON

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS							OUTPUTS	
D[0:3]	SD	CI	SP	CK	CON	CD	CO	Q[0:3]
X	X	X	X	↑	1	1	0	0
D[0:3]	1	0	1	↑	1	0	0	D[0:3]
D[0:3]	1	1	1	↑	1	0	*	D[0:3]
X	0	0	X	X	1	0	0	Q[0:3]
X	X	0	0	X	1	0	0	Q[0:3]
X	X	1	0	X	1	0	*	Q[0:3]
X	0	1	1	↑	1	0	*	count+1
X	X	0	X	↑	0	1	0	0
X	X	1	X	↑	0	1	1	0
D[0:3]	1	1	1	↑	0	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	0	**	D[0:3]
X	0	1	X	X	0	0	1	Q[0:3]
X	X	1	0	X	0	0	1	Q[0:3]
X	X	0	0	X	0	0	**	Q[0:3]
X	0	0	1	↑	0	0	**	count-1

X = Don't care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

** When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

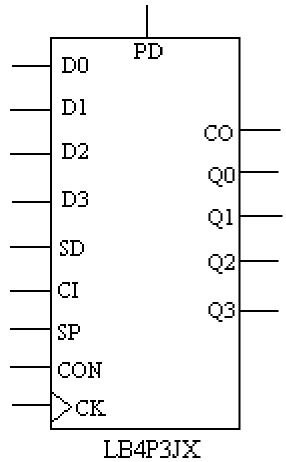
When GSR=0, CO=!CON"CI, Q[0:3]=0 (D[0:3]=SP=CK=SD=CD=X)

LB4P3JX

4 Bit Positive Edge Triggered Loadable Bidirectional Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, PD, CON

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

When CON = 0, CI and CO are active LOW

Truth Table

INPUTS							OUTPUTS	
D[0:3]	SD	CI	SP	CK	CON	PD	CO	Q[0:3]
X	X	0	X	↑	1	1	0	1
X	X	1	X	↑	1	1	1	1
D[0:3]	1	0	1	↑	1	0	0	D[0:3]
D[0:3]	1	1	1	↑	1	0	*	D[0:3]
X	0	0	X	X	1	0	0	Q[0:3]
X	X	0	0	X	1	0	0	Q[0:3]
X	X	1	0	X	1	0	*	Q[0:3]
X	0	1	1	↑	1	0	*	count+1
X	X	X	X	↑	0	1	1	1
D[0:3]	1	1	1	↑	0	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	0	**	D[0:3]
X	0	1	X	X	0	0	1	Q[0:3]
X	X	1	0	X	0	0	1	Q[0:3]
X	X	0	0	X	0	0	**	Q[0:3]
X	0	0	1	↑	0	0	**	count-1

X = Don’t care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

** When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=!CON+CI, Q[0:3]=1 (D[0:3]=SP=CK=SD=PD=X)

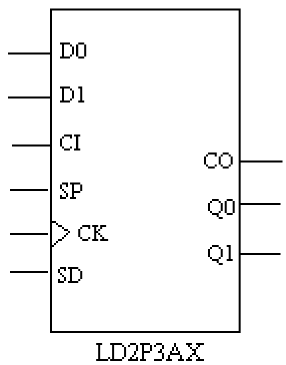
LD2P3AX

2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP

- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

CI and CO are active LOW

Truth Table

INPUTS					OUTPUTS	
D[0:1]	SD	CI	SP	CK	CO	Q[0:1]
D[0:1]	1	1	1	↑	1	D[0:1]
D[0:1]	1	0	1	↑	*	D[0:1]
X	0	1	X	X	1	Q[0:1]
X	X	1	0	X	1	Q[0:1]
X	X	0	0	X	*	Q[0:1]
X	0	0	1	↑	*	count-1

X = Don't care

* When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=CI, Q[0:1]=0 (D[0:1]=SP=CK=SD=X)

Note

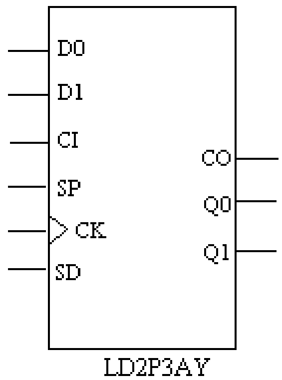
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LD2P3AY

2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

CI and CO are active LOW

Truth Table

INPUTS					OUTPUTS	
D[0:1]	SD	CI	SP	CK	CO	Q[0:1]
D[0:1]	1	1	1	↑	1	D[0:1]
D[0:1]	1	0	1	↑	*	D[0:1]
X	0	1	X	X	1	Q[0:1]
X	X	1	0	X	1	Q[0:1]
X	X	0	0	X	*	Q[0:1]
X	0	0	1	↑	*	count-1

X = Don’t care

* When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=1, Q[0:1]=1 (D[0:1]=CI=SP=CK=SD=X)

Note

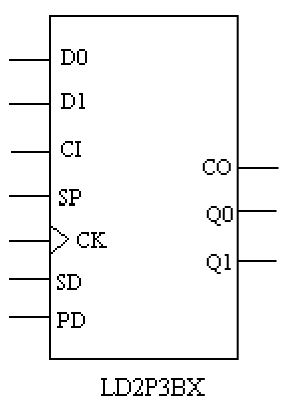
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LD2P3BX

2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, PD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:1]	SD	CI	SP	CK	PD	CO	Q[0:1]
X	X	X	X	X	1	1	1
D[0:1]	1	1	1	↑	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	*	D[0:1]
X	0	1	X	X	0	1	Q[0:1]
X	X	1	0	X	0	1	Q[0:1]
X	X	0	0	X	0	*	Q[0:1]
X	0	0	1	↑	0	*	count-1

X = Don’t care

* When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=1, Q[0:1]=1 (D[0:1]=SP=CK=SD=PD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

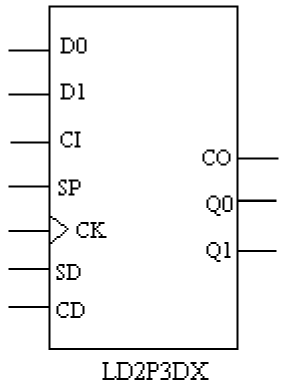
LD2P3DX

2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2

- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, CD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:1]	SD	CI	SP	CK	CD	CO	Q[0:1]
X	X	0	X	X	1	0	0
X	X	1	X	X	1	1	0
D[0:1]	1	1	1	↑	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	*	D[0:1]
X	0	1	X	X	0	1	Q[0:1]
X	X	1	0	X	0	1	Q[0:1]
X	X	0	0	X	0	*	Q[0:1]
X	0	0	1	↑	0	*	count-1

X = Don’t care

* When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=CI, Q[0:1]=0 (D[0:1]=SP=CK=SD=CD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

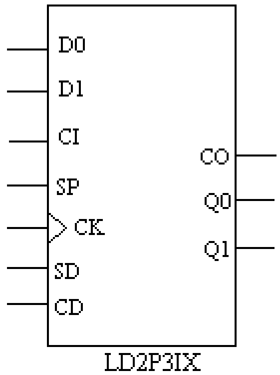
LD2P3IX

2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2

- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, CD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:1]	SD	CI	SP	CK	CD	CO	Q[0:1]
X	X	0	X	↑	1	0	0
X	X	1	X	↑	1	1	0
D[0:1]	1	1	1	↑	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	*	D[0:1]
X	0	1	X	X	0	1	Q[0:1]
X	X	1	0	X	0	1	Q[0:1]
X	X	0	0	X	0	*	Q[0:1]
X	0	0	1	↑	0	*	count-1

X = Don’t care

* When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=CI, Q[0:1]=0 (D[0:1]=SP=CK=SD=CD=X)

Note

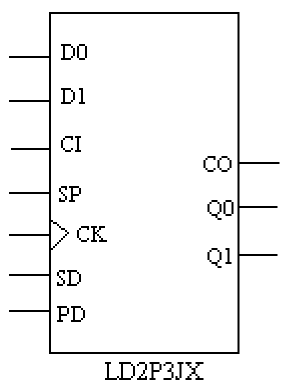
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LD2P3JX

2 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, PD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:1]	SD	CI	SP	CK	PD	CO	Q[0:1]
X	X	X	X	↑	1	1	1
D[0:1]	1	1	1	↑	0	1	D[0:1]
D[0:1]	1	0	1	↑	0	*	D[0:1]
X	0	1	X	X	0	1	Q[0:1]
X	X	1	0	X	0	1	Q[0:1]
X	X	0	0	X	0	*	Q[0:1]
X	0	0	1	↑	0	*	count-1

X = Don’t care

* When Q[0:1] is 00, CO will be 0; otherwise, CO will be 1

When GSR=0, CO=1, Q[0:1]=1 (D[0:1]=SP=CK=SD=PD=X)

Note

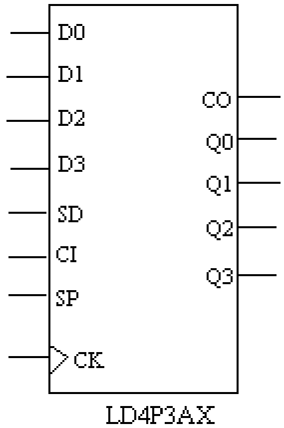
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LD4P3AX

4 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

CI and CO are active LOW

Truth Table

INPUTS					OUTPUTS	
D[0:3]	SD	CI	SP	CK	CO	Q[0:3]
D[0:3]	1	1	1	↑	1	D[0:3]
D[0:3]	1	0	1	↑	*	D[0:3]
X	0	1	X	X	1	Q[0:3]
X	X	1	0	X	1	Q[0:3]
X	X	0	0	X	*	Q[0:3]
X	0	0	1	↑	*	count-1

X = Don't care

* When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

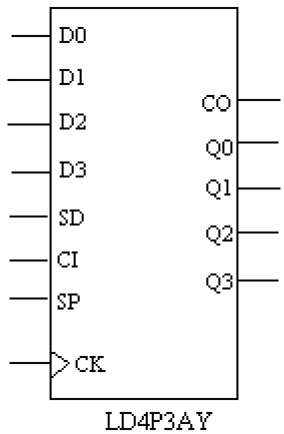
When GSR=0, CO=CI, Q[0:3]=0 (D[0:3]=SP=CK=SD=X)

LD4P3AY

4 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

CI and CO are active LOW

Truth Table

INPUTS					OUTPUTS	
D[0:3]	SD	CI	SP	CK	CO	Q[0:3]
D[0:3]	1	1	1	↑	1	D[0:3]
D[0:3]	1	0	1	↑	*	D[0:3]
X	0	1	X	X	1	Q[0:3]
X	X	1	0	X	1	Q[0:3]
X	X	0	0	X	*	Q[0:3]
X	0	0	1	↑	*	count-1

X = Don't care

* When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

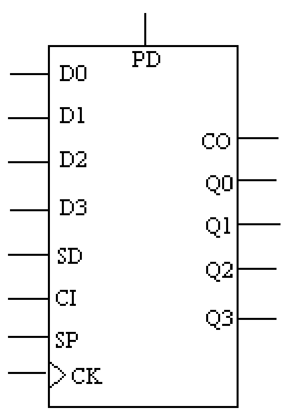
  When GSR=0, CO=1, Q[0:3]=1 (D[0:3]=CI=SP=CK=SD=X)

LD4P3BX

4 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



LD4P3BX

INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, PD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:3]	SD	CI	SP	CK	PD	CO	Q[0:3]
X	X	X	X	X	1	1	1
D[0:3]	1	1	1	↑	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	*	D[0:3]
X	0	1	X	X	0	1	Q[0:3]
X	X	1	0	X	0	1	Q[0:3]
X	X	0	0	X	0	*	Q[0:3]
X	0	0	1	↑	0	*	count-1

X = Don’t care

* When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

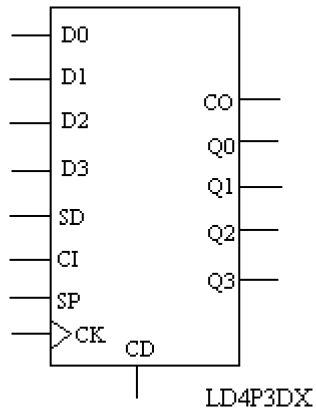
When GSR=0, CO=1, Q[0:3]=1 (D[0:3]=SP=CK=SD=PD=X)

LD4P3DX

4 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, CD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:3]	SD	CI	SP	CK	CD	CO	Q[0:3]
X	X	0	X	X	1	0	0
X	X	1	X	X	1	1	0
D[0:3]	1	1	1	↑	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	*	D[0:3]
X	0	1	X	X	0	1	Q[0:3]
X	X	1	0	X	0	1	Q[0:3]
X	X	0	0	X	0	*	Q[0:3]
X	0	0	1	↑	0	*	count-1

X = Don't care

* When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

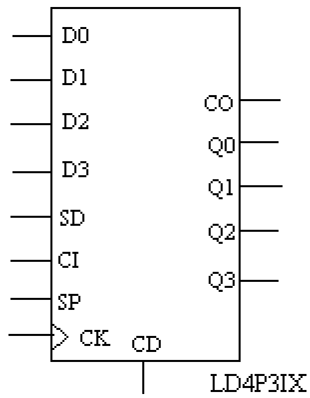
When GSR=0, CO=CI, Q[0:3]=0 (D[0:3]=SP=CK=SD=CD=X)

LD4P3IX

4 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, CD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:3]	SD	CI	SP	CK	CD	CO	Q[0:3]
X	X	0	X	↑	1	0	0
X	X	1	X	↑	1	1	0
D[0:3]	1	1	1	↑	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	*	D[0:3]
X	0	1	X	X	0	1	Q[0:3]
X	X	1	0	X	0	1	Q[0:3]
X	X	0	0	X	0	*	Q[0:3]
X	0	0	1	↑	0	*	count-1

X = Don’t care

* When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

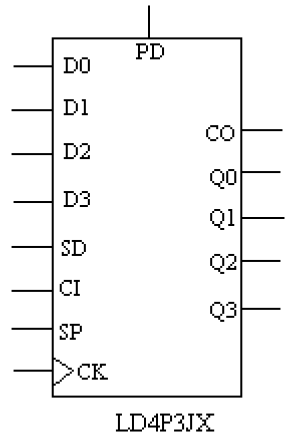
When GSR=0, CO=CI, Q[0:3]=0 (D[0:3]=SP=CK=SD=CD=X)

LD4P3JX

4 Bit Positive Edge Triggered Loadable Down-Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, PD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Note

CI and CO are active LOW

Truth Table

INPUTS						OUTPUTS	
D[0:3]	SD	CI	SP	CK	PD	CO	Q[0:3]
X	X	X	X	↑	1	1	1
D[0:3]	1	1	1	↑	0	1	D[0:3]
D[0:3]	1	0	1	↑	0	*	D[0:3]
X	0	1	X	X	0	1	Q[0:3]
X	X	1	0	X	0	1	Q[0:3]
X	X	0	0	X	0	*	Q[0:3]
X	0	0	1	↑	0	*	count-1

X = Don’t care

* When Q[0:3] is 0000, CO will be 0; otherwise, CO will be 1

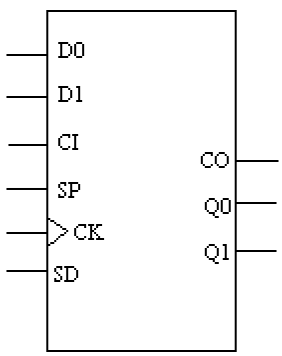
When GSR=0, CO=1, Q[0:3]=1 (D[0:3]=SP=CK=SD=PD=X)

LU2P3AX

2 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS					OUTPUTS	
D[0:1]	SD	CI	SP	CK	CO	Q[0:1]
D[0:1]	1	0	1	↑	0	D[0:1]
D[0:1]	1	1	1	↑	*	D[0:1]
X	0	0	X	X	0	Q[0:1]
X	X	0	0	X	0	Q[0:1]
X	X	1	0	X	*	Q[0:1]
X	0	1	1	↑	*	count+1

X = Don't care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

When GSR=0, CO=0, Q[0:1]=0 (D[0:1]=SP=CK=SD=X)

Note

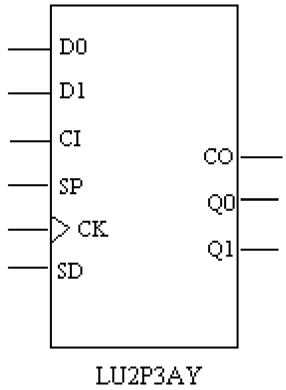
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LU2P3AY

2 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS					OUTPUTS	
D[0:1]	SD	CI	SP	CK	CO	Q[0:1]
D[0:1]	1	0	1	↑	0	D[0:1]
D[0:1]	1	1	1	↑	*	D[0:1]
X	0	0	X	X	0	Q[0:1]
X	X	0	0	X	0	Q[0:1]
X	X	1	0	X	*	Q[0:1]
X	0	1	1	↑	*	count+1

X = Don’t care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

When GSR=0, CO=CI, Q[0:1]=1 (D[0:1]=CI=SP=CK=SD=X)

Note

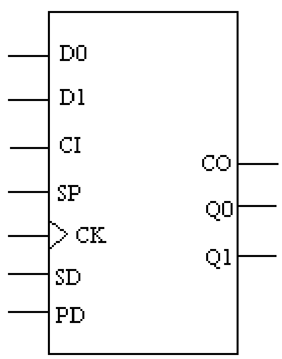
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LU2P3BX

2 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable and Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



LU2P3BX

INPUTS: D0, D1, CI, SP, CK, SD, PD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS						OUTPUTS	
D[0:1]	SD	CI	SP	CK	PD	CO	Q[0:1]
X	X	0	X	X	1	0	1
X	X	1	X	X	1	1	1
D[0:1]	1	0	1	↑	0	0	D[0:1]
D[0:1]	1	1	1	↑	0	*	D[0:1]
X	0	0	X	X	0	0	Q[0:1]
X	X	0	0	X	0	0	Q[0:1]
X	X	1	0	X	0	*	Q[0:1]
X	0	1	1	↑	0	*	count+1

X = Don’t care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

When GSR=0, CO=CI, Q[0:1]=1 (D[0:1]=SP=CK=SD=PD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

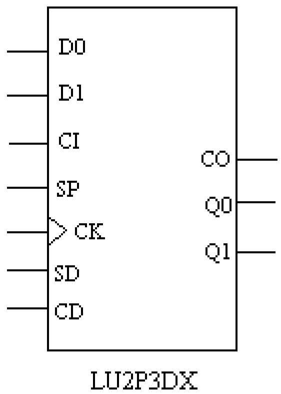
LU2P3DX

2 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable and Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L

- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, CD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS						OUTPUTS	
D[0:1]	SD	CI	SP	CK	CD	CO	Q[0:1]
X	X	X	X	X	1	0	0
D[0:1]	1	0	1	↑	0	0	D[0:1]
D[0:1]	1	1	1	↑	0	*	D[0:1]
X	0	0	X	X	0	0	Q[0:1]
X	X	0	0	X	0	0	Q[0:1]
X	X	1	0	X	0	*	Q[0:1]
X	0	1	1	↑	0	*	count+1

X = Don’t care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

When GSR=0, CO=0, Q[0:1]=0 (D[0:1]=SP=CK=SD=CD=X)

Note

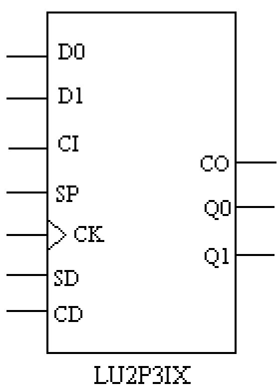
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LU2P3IX

2 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, CI, SP, CK, SD, CD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS						OUTPUTS	
D[0:1]	SD	CI	SP	CK	CD	CO	Q[0:1]
X	X	X	X	↑	1	0	0
D[0:1]	1	0	1	↑	0	0	D[0:1]
D[0:1]	1	1	1	↑	0	*	D[0:1]
X	0	0	X	X	0	0	Q[0:1]
X	X	0	0	X	0	0	Q[0:1]
X	X	1	0	X	0	*	Q[0:1]
X	0	1	1	↑	0	*	count+1

X = Don't care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

When GSR=0, CO=0, Q[0:1]=0 (D[0:1]=SP=CK=SD=CD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

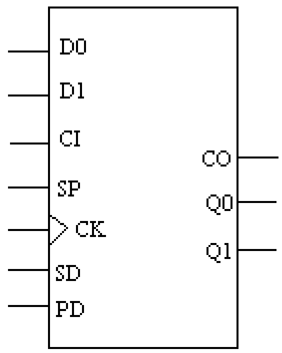
LU2P3JX

2 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L

- Platform Manager
- Platform Manager 2



LU2P3JX

INPUTS: D0, D1, CI, SP, CK, SD, PD

OUTPUTS: CO, Q0, Q1

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Truth Table

INPUTS						**OUTPUTS**	
D[0:1]	SD	CI	SP	CK	PD	CO	Q[0:1]
X	X	0	X	↑	1	0	1
X	X	1	X	↑	1	1	1
D[0:1]	1	0	1	↑	0	0	D[0:1]
D[0:1]	1	1	1	↑	0	*	D[0:1]
X	0	0	X	X	0	0	Q[0:1]
X	X	0	0	X	0	0	Q[0:1]
X	X	1	0	X	0	*	Q[0:1]
X	0	1	1	↑	0	*	count+1

X = Don't care

* When Q[0:1] is 11, CO will be 1; otherwise, CO will be 0

When GSR=0, CO=CI, Q[0:1]=1 (D[0:1]=SP=CK=SD=PD=X)

Note

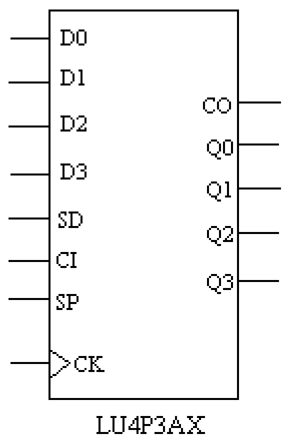
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

LU4P3AX

4 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable, GSR Used for Clear

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Truth Table

INPUTS					OUTPUTS	
D[0:3]	SD	CI	SP	CK	CO	Q[0:3]
D[0:3]	1	0	1	↑	0	D[0:3]
D[0:3]	1	1	1	↑	*	D[0:3]
X	0	0	X	X	0	Q[0:3]
X	X	0	0	X	0	Q[0:3]
X	X	1	0	X	*	Q[0:3]
X	0	1	1	↑	*	count+1

X = Don't care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

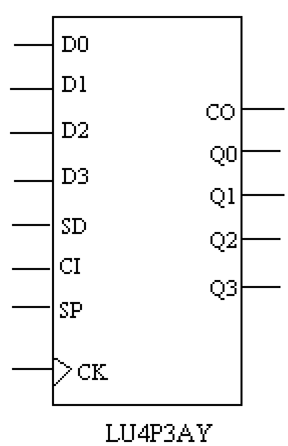
When GSR=0, CO=0, Q[0:3]=0 (D[0:3]=SP=CK=SD=X)

LU4P3AY

4 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable, GSR Used for Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Truth Table

INPUTS					OUTPUTS	
D[0:3]	SD	CI	SP	CK	CO	Q[0:3]
D[0:3]	1	0	1	↑	0	D[0:3]
D[0:3]	1	1	1	↑	*	D[0:3]
X	0	0	X	X	0	Q[0:3]
X	X	0	0	X	0	Q[0:3]
X	X	1	0	X	*	Q[0:3]
X	0	1	1	↑	*	count+1

X = Don't care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

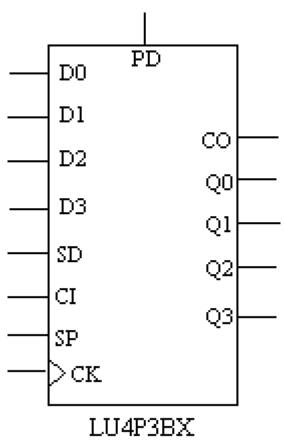
When GSR=0, CO=CI, Q[0:3]=1 (D[0:3]=CI=SP=CK=SD=X)

LU4P3BX

4 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable and Positive Level Asynchronous Preset

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, PD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Truth Table

INPUTS						OUTPUTS	
D[0:3]	SD	CI	SP	CK	PD	CO	Q[0:3]
X	X	0	X	X	1	0	1
X	X	1	X	X	1	1	1
D[0:3]	1	0	1	↑	0	0	D[0:3]
D[0:3]	1	1	1	↑	0	*	D[0:3]
X	0	0	X	X	0	0	Q[0:3]
X	X	0	0	X	0	0	Q[0:3]
X	X	1	0	X	0	*	Q[0:3]
X	0	1	1	↑	0	*	count+1

X = Don't care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

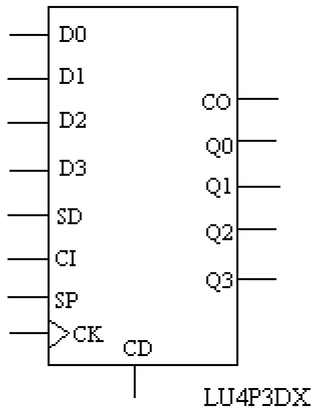
When GSR=0, CO=CI, Q[0:3]=1 (D[0:3]=SP=CK=SD=PD=X)

LU4P3DX

4 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable and Positive Level Asynchronous Clear

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, CD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Truth Table

INPUTS						OUTPUTS	
D[0:3]	SD	CI	SP	CK	CD	CO	Q[0:3]
X	X	X	X	X	1	0	0
D[0:3]	1	0	1	↑	0	0	D[0:3]
D[0:3]	1	1	1	↑	0	*	D[0:3]
X	0	0	X	X	0	0	Q[0:3]
X	X	0	0	X	0	0	Q[0:3]
X	X	1	0	X	0	*	Q[0:3]
X	0	1	1	↑	0	*	count+1

X = Don’t care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

When GSR=0, CO=0, Q[0:3]=0 (D[0:3]=SP=CK=SD=CD=X)

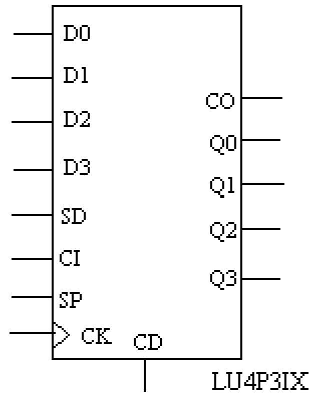
LU4P3IX

4 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable and Positive Level Synchronous Clear (Clear overrides Enable)

Architectures Supported:

- LatticeECP/EC

- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, CD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Truth Table

INPUTS						OUTPUTS	
D[0:3]	SD	CI	SP	CK	CD	CO	Q[0:3]
X	X	X	X	↑	1	0	0
D[0:3]	1	0	1	↑	0	0	D[0:3]
D[0:3]	1	1	1	↑	0	*	D[0:3]
X	0	0	X	X	0	0	Q[0:3]
X	X	0	0	X	0	0	Q[0:3]
X	X	1	0	X	0	*	Q[0:3]
X	0	1	1	↑	0	*	count+1

X = Don’t care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

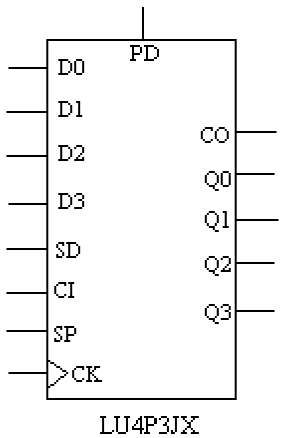
When GSR=0, CO=0, Q[0:3]=0 (D[0:3]=SP=CK=SD=CD=X)

LU4P3JX

4 Bit Positive Edge Triggered Loadable Up-Counter with Positive Clock Enable and Positive Level Synchronous Preset (Preset overrides Enable)

Architectures Supported:

- LatticeECP/EC
- LatticeSC/M
- LatticeXP



INPUTS: D0, D1, D2, D3, CI, SP, CK, SD, PD

OUTPUTS: CO, Q0, Q1, Q2, Q3

Truth Table

INPUTS						OUTPUTS	
D[0:3]	SD	CI	SP	CK	PD	CO	Q[0:3]
X	X	0	X	↑	1	0	1
X	X	1	X	↑	1	1	1
D[0:3]	1	0	1	↑	0	0	D[0:3]
D[0:3]	1	1	1	↑	0	*	D[0:3]
X	0	0	X	X	0	0	Q[0:3]
X	X	0	0	X	0	0	Q[0:3]
X	X	1	0	X	0	*	Q[0:3]
X	0	1	1	↑	0	*	count+1

X = Don't care

* When Q[0:3] is 1111, CO will be 1; otherwise, CO will be 0

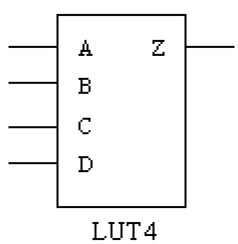
When GSR=0, CO=CI, Q[0:3]=1 (D[0:3]=SP=CK=SD=PD=X)

LUT4

4-Input Look Up Table

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: A, B, C, D

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 16'h0000)

Description

LUT4 defines the programmed state of a LUT4 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT4 programming. The contents of the look up table are addressed by the 4 input pins to access 1 of 16 locations.

The programming of the LUT4 (that is, the 0 or 1 value of each memory location within the LUT4) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

For example, hex value BF80 produces these 16 memory locations and values:

1011 1111 1000 0000

Memory location 0 (D=0, C=0, B=0, A=0) contains a 0, memory location 2 (D=0, C=0, B=1, A=0) contains a 0. Memory location 15 (D=1, C=1, B=1, A=1) contains a 1, etc.

The LUT4 may encode the Boolean logic for any Boolean expression of 4 input variables. For example, if the required expression was:

Z = (D*C) + (B*!A)

then the INIT value can be derived from the truth table resulting from the expression:

```
D C B A : Z
0 0 0 0 : 0
0 0 0 1 : 0
0 0 1 0 : 1
0 0 1 1 : 0

0 1 0 0 : 0
0 1 0 1 : 0
0 1 1 0 : 1
0 1 1 1 : 0

1 0 0 0 : 0
1 0 0 1 : 0
1 0 1 0 : 1
1 0 1 1 : 0

1 1 0 0 : 1
1 1 0 1 : 1
1 1 1 0 : 1
1 1 1 1 : 1
```

INIT = F444 (16)

LUT4 Usage with Verilog HDL

```
// LUT4 module instantiation
LUT4
   #(.init  (16'hF444))
   I1 ( .A (A), .B (B), .C (C), .D (D), .Z (Q[0]) );
```

LUT4 Usage with VHDL

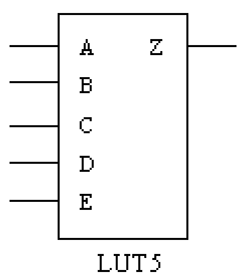
```
-- LUT4 component instantiation
I1 : LUT4
   Generic Map (INIT=>b"1111_0100_0100_0100")
   Port Map ( A=>A, B=>B, C=>C, D=>D, Z=>N_1 );
```

LUT5

5-Input Look Up Table

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: A, B, C, D, E

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 32'h00000000)

Description

LUT5 defines the programmed state of a LUT5 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT5 programming. The contents of the look up table are addressed by the 5 input pins to access 1 of 32 locations.

The programming of the LUT5 (that is, the 0 or 1 value of each memory location within the LUT5) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

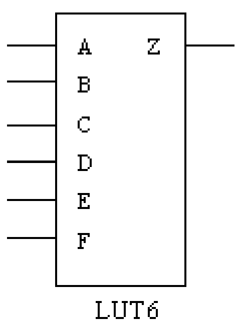
For more information on INIT attribute usage, see the LUT4 topic.

LUT6

6-Input Look Up Table

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: A, B, C, D, E, F

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 64'h0000000000000000)

Description

LUT6 defines the programmed state of a LUT6 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT6 programming. The contents of the look up table are addressed by the 6 input pins to access 1 of 64 locations.

The programming of the LUT6 (that is, the 0 or 1 value of each memory location within the LUT6) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

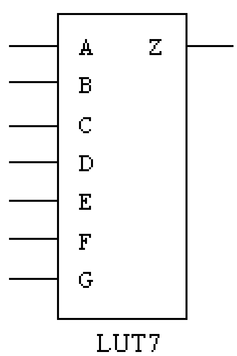
For more information on INIT attribute usage, see the LUT4 topic.

LUT7

7-Input Look Up Table

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: A, B, C, D, E, F, G

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 128'00000000000000000000000000000000)

Description

LUT7 defines the programmed state of a LUT7 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT7 programming. The contents of the look up table are addressed by the 7 input pins to access 1 of 128 locations.

The programming of the LUT7 (that is, the 0 or 1 value of each memory location within the LUT7) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

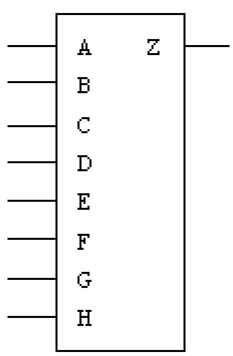
For more information on INIT attribute usage, see the LUT4 topic.

LUT8

8-Input Look Up Table

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



LUT8

INPUTS: A, B, C, D, E, F, G, H

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 256'h0000000000000000000000000000000000000000000000000000000000000000)

Description

LUT8 defines the programmed state of a LUT8 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT8 programming. The contents of the look up table are addressed by the 8 input pins to access 1 of 256 locations.

The programming of the LUT8 (that is, the 0 or 1 value of each memory location within the LUT8) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

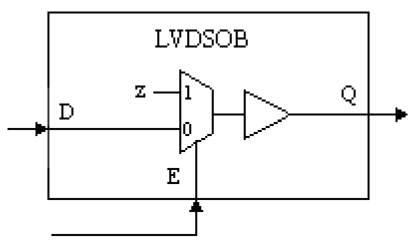
For more information on INIT attribute usage, see the LUT4 topic.

LVDSOB

LVDS Output Buffer

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, E

OUTPUT: Q

Description

The LVDSOB primitive is used to support post-PAR simulation of Dynamic Bank Controller. The Dynamic Bank Controller signals to the IO are hardwired and cannot be changed for the simulation, so the LVDSOB and INRDB primitives are defined to support the simulation.

For more information, refer to the following technical note on the Lattice web site:

- TN1198 - Power Estimation and Management for MachXO2 Device

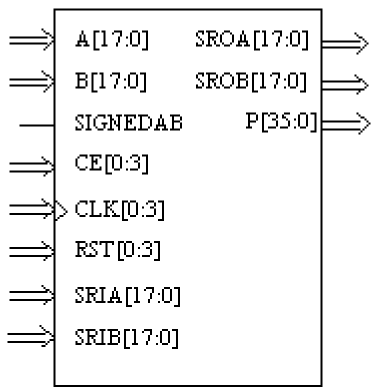
M

MULT18X18

MUX161ECP DSP Multiplier

Architectures Supported:

- LatticeECP (DSP Blocks Only)



MULT18X18

INPUTS: A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0, B17, B16, B15, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1, B0, SIGNEDAB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA17, SRIA16, SRIA15, SRIA14, SRIA13, SRIA12, SRIA11, SRIA10, SRIA9, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB17, SRIB16, SRIB15, SRIB14, SRIB13, SRIB12, SRIB11, SRIB10, SRIB9, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA17, SROA16, SROA15, SROA14, SROA13, SROA12, SROA11, SROA10, SROA9, SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB17, SROB16, SROB15, SROB14, SROB13, SROB12, SROB11, SROB10, SROB9, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, P35, P34, P33, P32, P31, P30, P29, P28, P27, P26, P25, P24, P23, P22, P21, P20, P19, P18, P17, P16, P15, P14, P13, P12, P11, P10, P9, P8, P7, P6, P5, P4, P3, P2, P1, P0

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

SHIFT_IN_A: "FALSE" (default), "TRUE"

SHIFT_IN_B: "FALSE" (default), "TRUE"

GSR: "ENABLED" (default), "DISABLED"

Description

LatticeECP DSP Block Multiplier. MULT18X18 is a combinational signed 18-bit by 18-bit multiplier used in the DSP block. The value represented in the 18-bit input A is multiplied by the value represented in the 18-bit input B. Output P is the 36-bit product of A and B. MULT18X18 may be represented as either unsigned or two's complement signed. The primitive consists of three types of optional pipeline registers:

- Input registers, located before the multipliers and registering the operands
- Multiplier pipeline registers, located after the multipliers and product registration

- Output registers, located before leaving the block, and registering the mode-specific output.

There are 12 register controls signals that enter the DSP block: CLK[0:3], CE0[0:3], and RST[0:3]. Each incoming signal also has the option of being tied high, tied low, or inverted. These 12 control signals are used to control the register banks in the DSP block. Dynamic control signals must match the register pipelining of the datapath. To facilitate this, the following bank control for each individual dynamic signal's input register and pipeline register (total 8 signals x 2 registers / signal = 16 registers) is as follows:

- Bypass or no-bypass of the registers.
- Clock is selected from CLK[0:3], one of four sources available to the DSP block.
- Clock enable is selected from CE0[0:3], one of four sources available to the DSP block.
- Reset is selected from RST[0:3], one of four sources available to the DSP block.

You can turn registers off or on via attribute settings. For example, setting REG_INPUTA_CLK= "CLK0" means the input A register is used, and the clock drive A register is coming from CLK0 of the DSP block. Setting REG_INPUTA_CE= "CE1" means input A register CE control is coming from CE1 of the DSP block. Setting REG_INPUTA_RST= "RST3" means input A register reset control is coming from RST3 of the DSP block. If REG_INPUT_A_CLK="NONE", this means the input A register is bypassed, therefore, REG_INPUT_A_RST or REG_INPUT_A_CE becomes irrelevant.

In case you want to use the register but do not care about the clock enable (CE), then this pin needs to be tied to VCC (always enabled). In this case you could set REG_INPUT_A_CE="CE3", then tie CE3 of the to VCC. If you want to use the register but do not care about the reset (RST), then this pin must to be tied to GND (always do not reset). In this case, you you could set REG_INPUT_A_RST="RST2", then tie RST2 of the to GND.

SIGNEDAB is a pin which controls whether the multiplier performs the signed or unsigned operation. It applied to both operand A and B. It can be tied to VCC (signed) or GROUND (unsigned). There are also two delay registers associated with this control pin, in order to match with incoming data. Setting REG_SIGNEDAB_0_CLK= "CLK0 | CLK1 | CLK2 | CLK3"will turn on the pipeline register for SIGNEDAB. Setting REG_SIGNEDAB_0_CLK= "NONE" will turn off the first pipeline register for SIGNEDAB. Setting REG_SIGNEDAB_1_CLK= "NONE" will turn off the second pipeline register for SIGNEDAB.

Input registers receive operand values from a serial shift chain or routing input. There is separate control for A and B operands. When in shift chain mode, multiplier operands may be bypassed using the bank bypass feature. The shift chain supports one chain of two 18-bit operands or two chains of two 9-bit operands. GSR "DISABLED" attribute disables the asynchronous global set reset input when in user mode.

You can refer to the following technical note on the Lattice web site for more details.

- TN1057 - LatticeECP sysDSP Usage Guide

MULT18X18 pin functions:

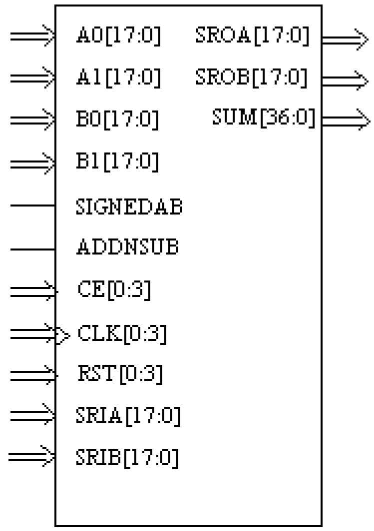
Function	Pins
input data A and B	A[17:0], B[17:0]0
signed input	SIGNEDAB
clock enable	CE[0:3]
clock input	CLK[0:3]
reset	RST[0:3]
shifted input A and B (from previous stage)	SRIA[17:0], SRIB[17:0]
shifted output A and B (from previous stage)	SROA[17:0], SROB[17:0]
output product data	P[35:0]

MULT18X18ADDSUB

ECP DSP Adder/Subtractor

Architectures Supported:

- LatticeECP (DSP Blocks Only)



MULT18X18ADDSUB

INPUTS: A017, A016, A015, A014, A013, A012, A011, A010, A09, A08, A07, A06, A05, A04, A03, A02, A01, A00, A117, A116, A115, A114, A113, A112, A111, A110, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, B017, B016, B015, B014, B013, B012, B011, B010, B09, B08, B07, B06, B05, B04, B03, B02, B01, B00, B117, B116, B115, B114, B113, B112, B111, B110, B19, B18, B17, B16, B15, B14, B13, B12, B11, B10, SIGNEDAB, ADDNSUB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA17, SRIA16, SRIA15, SRIA14, SRIA13, SRIA12, SRIA11, SRIA10, SRIA9, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB17, SRIB16, SRIB15, SRIB14, SRIB13, SRIB12, SRIB11, SRIB10, SRIB9, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA17, SROA16, SROA15, SROA14, SROA13, SROA12, SROA11, SROA10, SROA9, SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB17, SROB16, SROB15, SROB14, SROB13, SROB12, SROB11, SROB10, SROB9, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, SUM36, SUM35, SUM34, SUM33, SUM32, SUM31, SUM30, SUM29, SUM28, SUM27, SUM26, SUM25, SUM24, SUM23, SUM22, SUM21, SUM20, SUM19, SUM18, SUM17, SUM16, SUM15, SUM14, SUM13, SUM12, SUM11, SUM10, SUM9, SUM8, SUM7, SUM6, SUM5, SUM4, SUM3, SUM2, SUM1, SUM0

ATTRIBUTES:

REG_INPUTA0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB01_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

SHIFT_IN_A0: "FALSE" (default), "TRUE"

SHIFT_IN_B0: "FALSE" (default), "TRUE"

SHIFT_IN_A1: "FALSE" (default), "TRUE"

SHIFT_IN_B1: "FALSE" (default), "TRUE"

GSR: "ENABLED" (default), "DISABLED"

Description

LatticeECP Block Adder/Subtractor. MULT18X18ADDSUB can be configured to either add or subtract its inputs, adding or subtracting the inputs from two multiplier products. The add/subtract control is either configured as a static HIGH (Vcc), LOW (GND). In Lattice Diamond, the static settings are implemented by setting Vcc or GND in the CIB ISB.

The primitive consists of three types of optional pipeline registers:

- Input registers, located before the multipliers and registering the operands
- Multiplier pipeline registers, located after the multipliers and product registration
- Output registers, located before leaving the block, and registering the mode-specific output.

See the description of MULT18X18 for more information on control signals for DSP blocks and attributes.

You can also refer to the following technical note on the Lattice web site for more details.

- TN1057 - LatticeECP sysDSP Usage Guide

MULT18X18ADDSUB pin functions:

Function	Pins
input data A and B	A0|1[17:0], B0|1[17:0]
signed input (0 = unsigned, 1 = signed)	SIGNEDAB
add/subtract (0 = add, 1 = subtract)	ADDNSUB
clock enable	CE[0:3]
clock input	CLK[0:3]
reset	RST[0:3]
shifted input A and B (from previous stage)	SRIA[17:0], SRIB[17:0]

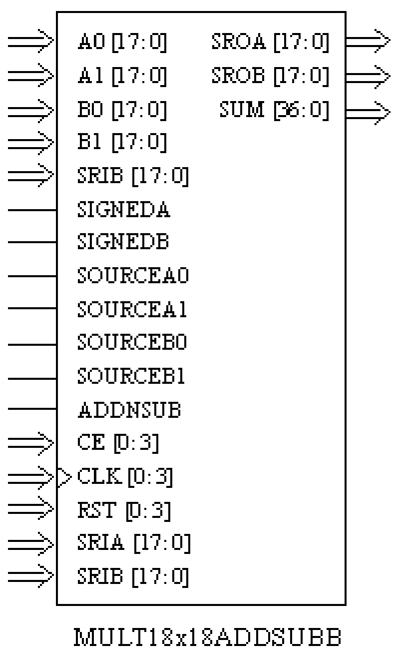
Function	Pins
shifted output A and B (from previous stage)	SROA[17:0], SROB[17:0]
output product sum data	SUM[36:0]

MULT18X18ADDSUBB

DSP Multiplier Add/Subtract

Architectures Supported:

- LatticeECP2/M
- LatticeECP3 (*for LatticeECP2/M backward compatibility only*)
- LatticeXP2



INPUTS: A017, A016, A015, A014, A013, A012, A011, A010, A09, A08, A07, A06, A05, A04, A03, A02, A01, A00, A117, A116, A115, A114, A113, A112, A111, A110, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, B017, B016, B015, B014, B013, B012, B011, B010, B09, B08, B07, B06, B05, B04, B03, B02, B01, B00, B117, B116, B115, B114, B113, B112, B111, B110, B19, B18, B17, B16, B15, B14, B13, B12, B11, B10, SIGNEDA, SIGNEDB, SOURCEA0, SOURCEA1, SOURCEB0, SOURCEB1, ADDNSUB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA17, SRIA16, SRIA15, SRIA14, SRIA13, SRIA12, SRIA11, SRIA10, SRIA9, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0,

SRIB17, SRIB16, SRIB15, SRIB14, SRIB13, SRIB12, SRIB11, SRIB10, SRIB9, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA17, SROA16, SROA15, SROA14, SROA13, SROA12, SROA11, SROA10, SROA9, SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB17, SROB16, SROB15, SROB14, SROB13, SROB12, SROB11, SROB10, SROB9, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, SUM36, SUM35, SUM34, SUM33, SUM32, SUM31, SUM30, SUM29, SUM28, SUM27, SUM26, SUM25, SUM24, SUM23, SUM22, SUM21, SUM20, SUM19, SUM18, SUM17, SUM16, SUM15, SUM14, SUM13, SUM12, SUM11, SUM10, SUM9, SUM8, SUM7, SUM6, SUM5, SUM4, SUM3, SUM2, SUM1, SUM0

ATTRIBUTES:

REG_INPUTA0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

Description

Refer to the following technical notes on the Lattice web site.

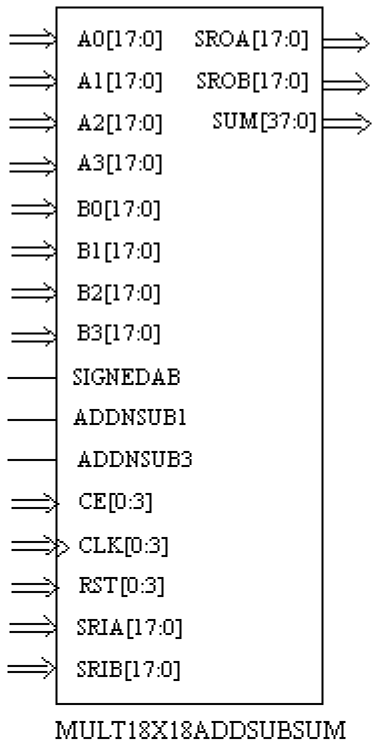
- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide

MULT18X18ADDSUBSUM

ECP DSP Adder/Subtractor/Sum

Architectures Supported:

- LatticeECP (DSP Blocks Only)



INPUTS: A017, A016, A015, A014, A013, A012, A011, A010, A09, A08, A07, A06, A05, A04, A03, A02, A01, A00, A117, A116, A115, A114, A113, A112, A111, A110, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, A217, A216, A215, A214, A213, A212, A211, A210, A29, A28, A27, A26, A25, A24, A23, A22, A21, A20, A317, A316, A315, A314, A313, A312, A311, A310, A39, A38, A37, A36, A35, A34, A33, A32, A31, A30, B017, B016, B015, B014, B013, B012, B011, B010, B09, B08, B07, B06, B05, B04, B03, B02, B01, B00, B117, B116, B115, B114, B113, B112, B111, B110, B19, B18, B17, B16, B15, B14, B13, B12, B11, B10, B217, B216, B215, B214, B213, B212, B211, B210, B29, B28, B27, B26, B25, B24, B23, B22, B21, B20, B317, B316, B315, B314, B313, B312, B311, B310, B39, B38, B37, B36, B35, B34, B33, B32, B31, B30, SIGNEDAB, ADDNSUB1, ADDNSUB3, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA17, SRIA16, SRIA15, SRIA14, SRIA13, SRIA12, SRIA11, SRIA10, SRIA9, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB17, SRIB16, SRIB15, SRIB14, SRIB13, SRIB12, SRIB11, SRIB10, SRIB9, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA17, SROA16, SROA15, SROA14, SROA13, SROA12, SROA11, SROA10, SROA9, SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB17, SROB16, SROB15, SROB14, SROB13, SROB12, SROB11, SROB10, SROB9, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, SUM37, SUM36, SUM35, SUM34, SUM33, SUM32, SUM31, SUM30, SUM29, SUM28, SUM27, SUM26, SUM25, SUM24, SUM23, SUM22, SUM21, SUM20, SUM19, SUM18, SUM17, SUM16, SUM15, SUM14, SUM13, SUM12, SUM11, SUM10, SUM9, SUM8, SUM7, SUM6, SUM5, SUM4, SUM3, SUM2, SUM1, SUM0

ATTRIBUTES:

REG_INPUTA0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE0_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE1_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB1_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB1_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB1_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB1_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB1_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB1_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB3_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB3_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB3_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB3_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB3_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB3_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

SHIFT_IN_A0: "FALSE" (default), "TRUE"

SHIFT_IN_B0: "FALSE" (default), "TRUE"

SHIFT_IN_A1: "FALSE" (default), "TRUE"

SHIFT_IN_B1: "FALSE" (default), "TRUE"

SHIFT_IN_A2: "FALSE" (default), "TRUE"

SHIFT_IN_B2: "FALSE" (default), "TRUE"

SHIFT_IN_A3: "FALSE" (default), "TRUE"

SHIFT_IN_B3: "FALSE" (default), "TRUE"

GSR: "ENABLED" (default), "DISABLED"

Description

LatticeECP DSP Block Adder/Subtractor. MULT18X18ADDSUBSUM can be configured to either add or subtract its inputs, adding or subtracting the inputs from two multiplier products. The add/subtract control is either configured as a static HIGH (Vcc), LOW (GND), or as dynamic control signals ADDNSUB1 and ADDNSUB3. In Lattice Diamond, the static settings are implemented by setting ADDNSUB1 and ADDNSUB3 signal to Vcc or GND in the CIB ISB.

The primitive consists of three types of optional pipeline registers:

- Input registers, located before the multipliers and registering the operands
- Multiplier pipeline registers, located after the multipliers and product registration
- Output registers, located before leaving the block, and registering the mode-specific output.

See the description of MULT18X18 for more information on control signals for DSP blocks and attributes.

Refer to the following technical note on the Lattice web site.

- TN1057 - LatticeECP sysDSP Usage Guide

MULT18X18ADDSUBSUM pin functions:

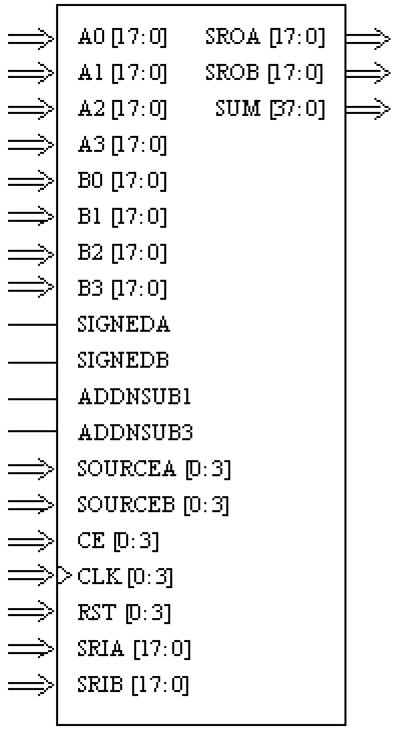
Function	Pins
input data A and B	A0|1|2|3[17:0], B0|1|2|3[17:0]
signed input (0 = unsigned, 1 = signed)	SIGNEDAB
add/subtract (0 = add, 1 = subtract)	ADDNSUB1, ADDNSUB3
clock enable	CE[0:3]
clock input	CLK[0:3]
reset	RST[0:3]
shifted input A and B (from previous stage)	SRIA[17:0], SRIB[17:0]
shifted output A and B (from previous stage)	SROA[17:0], SROB[17:0]
output product sum data	SUM[37:0]

MULT18X18ADDSUBSUMB

DSP Multiplier Add/Subtract/Sum

Architectures Supported:

- LatticeECP2/M
- LatticeECP3 (*for LatticeECP2/M backward compatibility only*)
- LatticeXP2



MULT18x18ADDSUBSUMB

INPUTS: A017, A016, A015, A014, A013, A012, A011, A010, A09, A08, A07, A06, A05, A04, A03, A02, A01, A00, A117, A116, A115, A114, A113, A112, A111, A110, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, A217, A216, A215, A214, A213, A212, A211, A210, A29, A28, A27, A26, A25, A24, A23, A22, A21, A20, A317, A316, A315, A314, A313, A312, A311, A310, A39, A38, A37, A36, A35, A34, A33, A32, A31, A30, B017, B016, B015, B014, B013, B012, B011, B010, B09, B08, B07, B06, B05, B04, B03, B02, B01, B00, B117, B116, B115, B114, B113, B112, B111, B110, B19, B18, B17, B16, B15, B14, B13, B12, B11, B10, B217, B216, B215, B214, B213, B212, B211, B210, B29, B28, B27, B26, B25, B24, B23, B22, B21, B20, B317, B316, B315, B314, B313, B312, B311, B310, B39, B38, B37, B36, B35, B34, B33, B32, B31, B30, SIGNEDA, SIGNEDB, ADDNSUB1, ADDNSUB3, SOURCEA0, SOURCEA1, SOURCEA2, SOURCEA3, SOURCEB0, SOURCEB1, SOURCEB2, SOURCEB3, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA17, SRIA16, SRIA15, SRIA14, SRIA13, SRIA12, SRIA11, SRIA10, SRIA9, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB17, SRIB16, SRIB15, SRIB14, SRIB13, SRIB12, SRIB11, SRIB10, SRIB9, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA17, SROA16, SROA15, SROA14, SROA13, SROA12, SROA11, SROA10, SROA9, SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB17, SROB16, SROB15, SROB14, SROB13, SROB12, SROB11, SROB10, SROB9, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, SUM37, SUM36, SUM35, SUM34, SUM33, SUM32, SUM31, SUM30, SUM29, SUM28, SUM27, SUM26, SUM25, SUM24, SUM23, SUM22, SUM21, SUM20,

SUM19, SUM18, SUM17, SUM16, SUM15, SUM14, SUM13, SUM12, SUM11, SUM10, SUM9, SUM8, SUM7, SUM6, SUM5, SUM4, SUM3, SUM2, SUM1, SUM0

ATTRIBUTES:

REG_INPUTA0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB1_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB1_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB1_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB1_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB1_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB1_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB3_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB3_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB3_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB3_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB3_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB3_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

Description

Refer to the following technical notes on the Lattice web site.

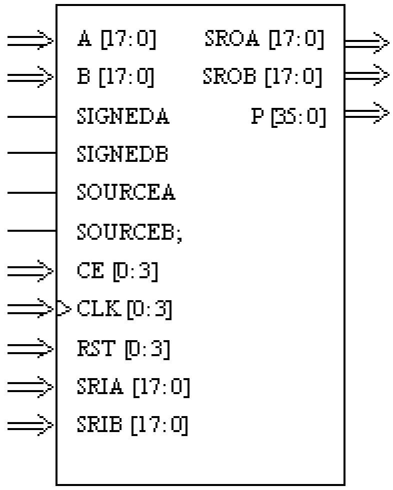
- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide

MULT18X18B

DSP Multiplier

Architectures Supported:

- LatticeECP2/M
- LatticeECP3 (*for LatticeECP2/M backward compatibility only*)
- LatticeXP2



MULT18x18B

INPUTS: A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0, B17, B16, B15, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1, B0, SIGNEDA, SIGNEDB, SOURCEA, SOURCEB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA17, SRIA16, SRIA15, SRIA14, SRIA13, SRIA12, SRIA11, SRIA10, SRIA9, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB17, SRIB16, SRIB15, SRIB14, SRIB13, SRIB12, SRIB11, SRIB10, SRIB9, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA17, SROA16, SROA15, SROA14, SROA13, SROA12, SROA11, SROA10, SROA9, SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB17, SROB16, SROB15, SROB14, SROB13, SROB12, SROB11, SROB10, SROB9, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, P35, P34, P33, P32, P31, P30, P29, P28, P27, P26, P25, P24, P23, P22, P21, P20, P19, P18, P17, P16, P15, P14, P13, P12, P11, P10, P9, P8, P7, P6, P5, P4, P3, P2, P1, P0

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

Description

Refer to the following technical notes on the Lattice web site.

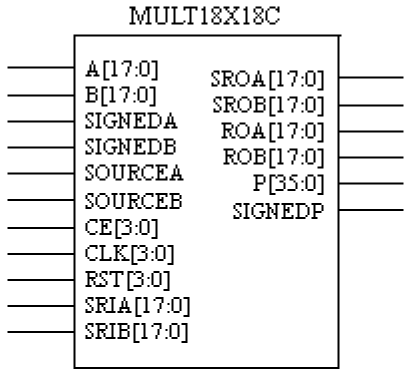
- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide

MULT18X18C

DSP Multiplier

Architectures Supported:

- LatticeECP3\



INPUTS: A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0, B17, B16, B15, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1, B0, SIGNEDA, SIGNEDB, SOURCEA, SOURCEB, CE3, CE2, CE1, CE0, CLK3, CLK2, CLK1, CLK0, RST3, RST2, RST1, RST0, SRIA17, SRIA16, SRIA15, SRIA14, SRIA13, SRIA12, SRIA11, SRIA10, SRIA9, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB17, SRIB16, SRIB15, SRIB14, SRIB13, SRIB12, SRIB11, SRIB10, SRIB9, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA17, SROA16, SROA15, SROA14, SROA13, SROA12, SROA11, SROA10, SROA9, SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB17, SROB16, SROB15, SROB14, SROB13, SROB12, SROB11, SROB10, SROB9, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, ROA17, ROA16, ROA15, ROA14, ROA13, ROA12, ROA11, ROA10, ROA9, ROA8, ROA7, ROA6, ROA5, ROA4, ROA3, ROA2, ROA1, ROA0, ROB17, ROB16, ROB15, ROB14, ROB13, ROB12, ROB11, ROB10, ROB9, ROB8, ROB7, ROB6, ROB5, ROB4, ROB3, ROB2, ROB1, ROB0, P35, P34, P33, P32, P31, P30, P29, P28, P27, P26, P25, P24, P23, P22, P21, P20, P19, P18, P17, P16, P15, P14, P13, P12, P11, P10, P9, P8, P7, P6, P5, P4, P3, P2, P1, P0, SIGNEDP

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

CAS_MATCH_REG: "FALSE" (default), "TRUE"

MULT_BYPASS: "DISABLED" (default), "ENABLED"

GSR: "ENABLED" (default), "DISABLED"

RESETMODE: "SYNC" (default), "ASYNC"

MULT18X18C Port Description

I/O	Port Name	Capture Name	Type	Size (Buses Only)	Description
I	A	A	Bus	17:0	A input
I	B	B	Bus	17:0	B input
I	SRIA	SRIA	Bus	17:0	Shift input A
I	SRIB	SRIB	Bus	17:0	Shift input B
I	SIGNEDA	SIGNEDA	Bit	N/A	Input A sign selection
I	SIGNEDB	SIGNEDB	Bit	N/A	Input B sign selection
I	SOURCEA	SOURCEA	Bit	N/A	Source A selection
I	SOURCEB	SOURCEB	Bit	N/A	Source B selection
I	CLK0	CLK0	Bit	N/A	Clock Input
I	CLK1	CLK1	Bit	N/A	Clock Input
I	CLK2	CLK2	Bit	N/A	Clock Input
I	CLK3	CLK3	Bit	N/A	Clock Input
I	CE0	CE0	Bit	N/A	Clock Enable Input
I	CE1	CE1	Bit	N/A	Clock Enable Input
I	CE2	CE2	Bit	N/A	Clock Enable Input
I	CE3	CE3	Bit	N/A	Clock Enable Input
I	RST0	RST0	Bit	N/A	Reset Input
I	RST1	RST1	Bit	N/A	Reset Input
I	RST2	RST2	Bit	N/A	Reset Input

I/O	Port Name	Capture Name	Type	Size (Buses Only)	Description
I	RST3	RST3	Bit	N/A	Reset Input
O	P	P	Bus	35:0	Product
O	SROA	SROA	Bus	17:0	Shift A Output
O	SROB	SROB	Bus	17:0	Shift B Output
O	ROA	ROA	Bus	17:0	Registered Output A from Multiplier
O	ROB	ROB	Bus	17:0	Registered Output B from Multiplier
O	SIGNEDP	SIGNEDP	Bit	N/A	Output Sign Bit (result of SignedA or SignedB)

MULT18X18C Attribute Description

Name	Value	Default	Description
REG_INPUTA_CLK	"NONE", "CLK0", "CLK1", "CLK2", "CLK3"	"NONE"	Input A clock selection
REG_INPUTA_CE	"CE0", "CE1", "CE2", "CE3"	"CE0"	Input A clock enable selection
REG_INPUTA_RST	"RST0", "RST1", "RST2", "RST3"	"RST0"	Input A reset selection
REG_INPUTB_CLK	"NONE", "CLK0", "CLK1", "CLK2", "CLK3"	"NONE"	Input B clock selection
REG_INPUTB_CE	"CE0", "CE1", "CE2", "CE3"	"CE0"	Input B clock enable selection
REG_INPUTB_RST	"RST0", "RST1", "RST2", "RST3"	"RST0"	Input B reset selection
REG_PIPELINE_CLK	"NONE", "CLK0", "CLK1", "CLK2", "CLK3"	"NONE"	Pipeline clock selection
REG_PIPELINE_CE	"CE0", "CE1", "CE2", "CE3"	"CE0"	Pipeline clock enable selection
REG_PIPELINE_RST	"RST0", "RST1", "RST2", "RST3"	"RST0"	Pipeline reset selection
REG_OUTPUT_CLK	"NONE", "CLK0", "CLK1", "CLK2", "CLK3"	"NONE"	Output clock selection
REG_OUTPUT_CE	"CE0", "CE1", "CE2", "CE3"	"CE0"	Output clock enable selection
REG_OUTPUT_RST	"RST0", "RST1", "RST2", "RST3"	"RST0"	Output reset selection

Name	Value	Default	Description
CAS_MATCH_REG	"FALSE", "TRUE"	"FALSE"	Cascade match register option
MULT_BYPASS	"DISABLED", "ENABLED"	"ENABLED"	Multiplier bypass option
RESETMODE	"SYNC", "ASYNC"	"SYNC"	Global set reset selection
GSR	"ENABLED", "DISABLED"	"ENABLED"	Reset mode selection

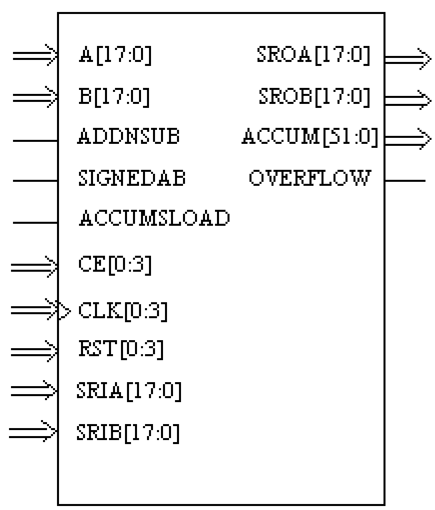
Note:
The multipliers linked by SROA/SRIA or SROB/SRIB pin pair can only be implemented in a continuous shift chain with multipliers placed next to each other. However, the length of the shift chain is limited on each DSP row and different depending on the device selected. For example, with the LatticeECP3-35 device, each DSP row has 32 MULT18s. So for the shift chained MULT18, it can support a continuous chain of length 32 for MULT18. If a chain is more than 32 MULT18s, you have to employ SRO/A,B pin pair for connecting the 32nd and 33rd MULT18s for implementing the chain in two DSP rows. But this capability is only limited to port A on SRO side in LatticeECP3. That is to say, only SROA is allowed to connect to A,B input pin of another MULT18, but not SROB. So if a chain is linked by SROB/SRIB pin pair, the length cannot exceed 32 for the LatticeECP3-35 device.

MULT18X18MAC

ECP DSP Multiplier Accumulate

Architectures Supported:

- LatticeECP (DSP Blocks Only)



MULT18X18MAC

INPUTS: A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0, B17, B16, B15, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1, B0, ADDNSUB, SIGNEDAB, ACCUMSLOAD, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA17, SRIA16, SRIA15, SRIA14, SRIA13, SRIA12, SRIA11, SRIA10, SRIA9, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB17, SRIB16, SRIB15, SRIB14, SRIB13, SRIB12, SRIB11, SRIB10, SRIB9, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA17, SROA16, SROA15, SROA14, SROA13, SROA12, SROA11, SROA10, SROA9, SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB17, SROB16, SROB15, SROB14, SROB13, SROB12, SROB11, SROB10, SROB9, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, ACCUM51, ACCUM50, ACCUM49, ACCUM48, ACCUM47, ACCUM46, ACCUM45, ACCUM44, ACCUM43, ACCUM42, ACCUM41, ACCUM40, ACCUM39, ACCUM38, ACCUM37, ACCUM36, ACCUM35, ACCUM34, ACCUM33, ACCUM32, ACCUM31, ACCUM30, ACCUM29, ACCUM28, ACCUM27, ACCUM26, ACCUM25, ACCUM24, ACCUM23, ACCUM22, ACCUM21, ACCUM20, ACCUM19, ACCUM18, ACCUM17, ACCUM16, ACCUM15, ACCUM14, ACCUM13, ACCUM12, ACCUM11, ACCUM10, ACCUM9, ACCUM8, ACCUM7, ACCUM6, ACCUM5, ACCUM4, ACCUM3, ACCUM2, ACCUM1, ACCUM0, OVERFLOW

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ACCUMSLOAD_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ACCUMSLOAD_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ACCUMSLOAD_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ACCUMSLOAD_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ACCUMSLOAD_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ACCUMSLOAD_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

SHIFT_IN_A: "FALSE" (default), "TRUE"

SHIFT_IN_B: "FALSE" (default), "TRUE"

GSR: "ENABLED" (default), "DISABLED"

Description

MULT18X18MAC supports operand bit widths for 18X18 multiplication and accumulates the output up to 52 bits. The DSP block includes optional registers for the input and intermediate pipeline stage. Pipeline stages may be set using a pipeline attribute. The output registers are required for the accumulator. Signed and unsigned arithmetic are supported. The OVERFLOW bit is also provided when the accumulated results are in the overflow condition. ACCUMSLOAD determines the mode of operation for either loading the multiplier product or to accumulate.

The primitive consists of three types of optional pipeline registers:

- Input registers, located before the multipliers and registering the operands
- Multiplier pipeline registers, located after the multipliers and product registration
- Output registers, located before leaving the block, and registering the mode-specific output.

See the description of MULT18X18 for more information on control signals for DSP blocks and attributes.

Refer to the following technical note on the Lattice web site.

- TN1057 - LatticeECP sysDSP Usage Guide

MULT18X18MAC pin functions:

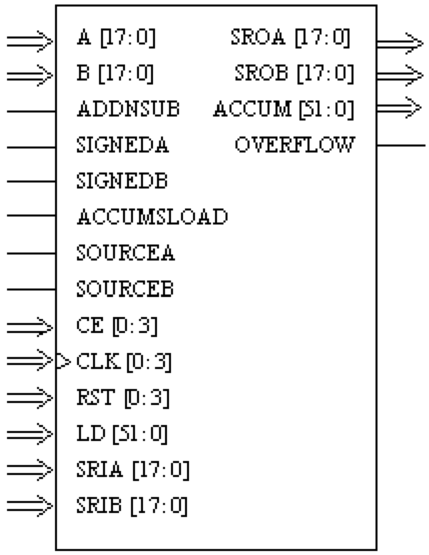
Function	Pins
input data A and B	A[17:0], B[17:0]
signed input (0 = unsigned, 1 = signed)	SIGNEDAB
add/subtract (0 = add, 1 = subtract)	ADDNSUB
Accumulate (HIGH) /Load (LOW) Mode	ACCUMSLOAD
clock enable	CE[0:3]
clock input	CLK[0:3]
clock reset	RST[0:3]
shifted input A and B (from previous stage)	SRIA[17:0], SRIB[17:0]
shifted output A and B (from previous stage)	SROA[17:0], SROB[17:0]
output data	ACCUM[51:0]
overflow	OVERFLOW

MULT18X18MACB

DSP Multiplier Accumulate

Architectures Supported:

- LatticeECP2/M
- LatticeECP3 (*for LatticeECP2/M backward compatibility only*)
- LatticeXP2



MULT18x18MACB

INPUTS: A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0, B17, B16, B15, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1, B0, ADDNSUB, SIGNEDA, SIGNEDB, ACCUMSLOAD, SOURCEA, SOURCEB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, LD51, LD50, LD49, LD48, LD47, LD46, LD45, LD44, LD43, LD42, LD41, LD40, LD39, LD38, LD37, LD36, LD35, LD34, LD33, LD32, LD31, LD30, LD29, LD28, LD27, LD26, LD25, LD24, LD23, LD22, LD21, LD20, LD19, LD18, LD17, LD16, LD15, LD14, LD13, LD12, LD11, LD10, LD9, LD8, LD7, LD6, LD5, LD4, LD3, LD2, LD1, LD0, SRIA17, SRIA16, SRIA15, SRIA14, SRIA13, SRIA12, SRIA11, SRIA10, SRIA9, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB17, SRIB16, SRIB15, SRIB14, SRIB13, SRIB12, SRIB11, SRIB10, SRIB9, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA17, SROA16, SROA15, SROA14, SROA13, SROA12, SROA11, SROA10, SROA9, SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB17, SROB16, SROB15, SROB14, SROB13, SROB12, SROB11, SROB10, SROB9, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, ACCUM51, ACCUM50, ACCUM49, ACCUM48, ACCUM47, ACCUM46, ACCUM45, ACCUM44, ACCUM43, ACCUM42, ACCUM41, ACCUM40, ACCUM39, ACCUM38, ACCUM37, ACCUM36, ACCUM35, ACCUM34, ACCUM33, ACCUM32, ACCUM31, ACCUM30, ACCUM29, ACCUM28, ACCUM27, ACCUM26, ACCUM25, ACCUM24, ACCUM23, ACCUM22, ACCUM21, ACCUM20, ACCUM19, ACCUM18, ACCUM17, ACCUM16, ACCUM15, ACCUM14, ACCUM13, ACCUM12, ACCUM11, ACCUM10, ACCUM9, ACCUM8, ACCUM7, ACCUM6, ACCUM5, ACCUM4, ACCUM3, ACCUM2, ACCUM1, ACCUM0, OVERFLOW

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ACCUMSLOAD_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ACCUMSLOAD_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ACCUMSLOAD_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ACCUMSLOAD_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ACCUMSLOAD_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ACCUMSLOAD_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

Description

Refer to the following technical notes on the Lattice web site.

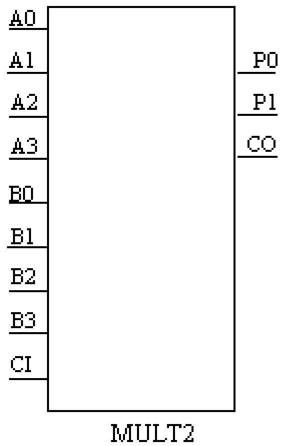
- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide

MULT2

2x2 Multiplier

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A0, A1, A2, A3, B0, B1, B2, B3, CI

OUTPUTS: P0, P1, CO

Description

MULT2 is a 2x2 multiplier. This primitive is useful for implementing an array multiplier using a dedicated carry chain. With this unique configuration, the two separate “and” inputs can be added together to perform the add and shift operations within a single slice. MULT2 must be cascaded together when performing the multiply function. Here are descriptions of the MULT2 pins:

Function	Pins
multiplicand	A0, A1, A2, A3
multiplier	B0, B1, B2, B3
carry in	CI
carry out	CO
output	P0, P1

The equations for this primitive are shown in the table below:

Equations
CO_int, P0 = A0*B0 + A1*B1 + CI
CO, P1 = A2*B2 + A3*B3 + CO_int

CO_int and P0 are the carry-out and sum of a full adder with inputs (A0 AND B0), (A1 AND B1), and CI. CO and P1 are the carry-out and sum of a full adder with inputs (A2 AND B2), (A3 AND B3), and CO_int.

Refer to the following technical notes on the Lattice web site.

- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide
- TN1057 - LatticeECP sysDSP Usage Guide

Note

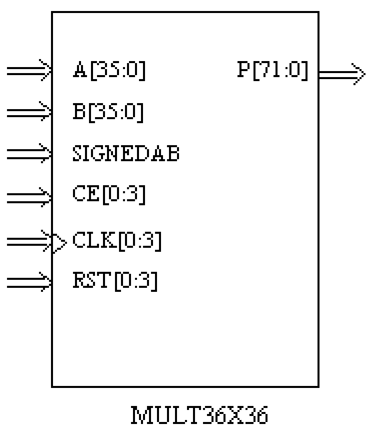
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

MULT36X36

ECP DSP Multiplier

Architectures Supported:

- LatticeECP (DSP Blocks Only)



INPUTS: A35, A34, A33, A32, A31, A30, A29, A28, A27, A26, A25, A24, A23, A22, A21, A20, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0, B35, B34, B33, B32, B31, B30, B29, B28, B27, B26, B25, B24, B23, B22, B21, B20, B19, B18, B17, B16, B15, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1, B0, SIGNEDAB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3

OUTPUTS: P71, P70, P69, P68, P67, P66, P65, P64, P63, P62, P61, P60, P59, P58, P57, P56, P55, P54, P53, P52, P51, P50, P49, P48, P47, P46, P45, P44, P43, P42, P41, P40, P39, P38, P37, P36, P35, P34, P33, P32, P31, P30, P29, P28, P27, P26, P25, P24, P23, P22, P21, P20, P19, P18, P17, P16, P15, P14, P13, P12, P11, P10, P9, P8, P7, P6, P5, P4, P3, P2, P1, P0

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

Description

LatticeECP Block Multiplier. MULT36X36 is a combinational signed 36-bit by 36-bit multiplier used in the DSP block. The value represented in the 36-bit input A is multiplied by the value represented in the 36-bit input B. Output P is the 72-bit product of A and B. MULT36X36 may be represented as either unsigned or two's complement signed. The primitive consists of three types of optional pipeline registers:

- Input registers, located before the multipliers and registering the operands
- Multiplier pipeline registers, located after the multipliers and product registration

- Output registers, located before leaving the block, and registering the mode-specific output.

See the description of MULT18X18 for more information on control signals for DSP blocks and attributes.

Input registers receive operand values from a serial shift chain or routing input. There is separate control for A and B operands. When in shift chain mode, multiplier operands may be bypassed using the bank bypass feature. The shift chain supports one chain of two 18-bit operands or two chains of two 9-bit operands. GSR "DISABLED" attribute disables the asynchronous global set reset input when in user mode.

Refer to the following technical note on the Lattice web site.

- TN1057 - LatticeECP sysDSP Usage Guide

MULT36X36 pin functions:

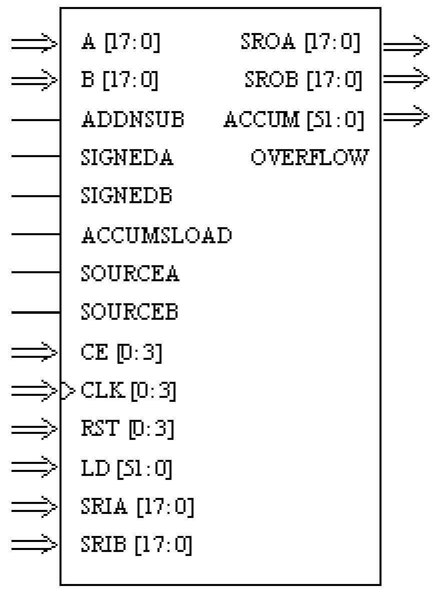
Function	**Pins**
input data A and B	A[35:0], B[35:0]
signed input (0 = unsigned, 1 = signed)	SIGNEDAB
clock enable	CE[0:3]
clock input	CLK[0:3]
clock reset	RST[0:3]
output data	P[71:0]

MULT36X36B

DSP Multiplier

Architectures Supported:

- LatticeECP2/M
- LatticeECP3 (*for LatticeECP2/M backward compatibility only*)
- LatticeXP2



MULT36x36B

INPUTS: A35, A34, A33, A32, A31, A30, A29, A28, A27, A26, A25, A24, A23, A22, A21, A20, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0, B35, B34, B33, B32, B31, B30, B29, B28, B27, B26, B25, B24, B23, B22, B21, B20, B19, B18, B17, B16, B15, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1, B0, SIGNEDA, SIGNEDB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3

OUTPUTS: P71, P70, P69, P68, P67, P66, P65, P64, P63, P62, P61, P60, P59, P58, P57, P56, P55, P54, P53, P52, P51, P50, P49, P48, P47, P46, P45, P44, P43, P42, P41, P40, P39, P38, P37, P36, P35, P34, P33, P32, P31, P30, P29, P28, P27, P26, P25, P24, P23, P22, P21, P20, P19, P18, P17, P16, P15, P14, P13, P12, P11, P10, P9, P8, P7, P6, P5, P4, P3, P2, P1, P0

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

Description

Refer to the following technical notes on the Lattice web site.

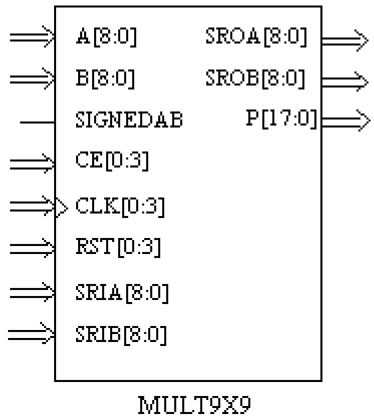
- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide

MULT9X9

ECP DSP Multiplier

Architectures Supported:

- LatticeECP (DSP Blocks Only)



INPUTS: A8, A7, A6, A5, A4, A3, A2, A1, A0, B8, B7, B6, B5, B4, B3, B2, B1, B0, SIGNEDAB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, P17, P16, P15, P14, P13, P12, P11, P10, P9, P8, P7, P6, P5, P4, P3, P2, P1, P0

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

SHIFT_IN_A: "FALSE" (default), "TRUE"

SHIFT_IN_B: "FALSE" (default), "TRUE"

GSR: "ENABLED" (default), "DISABLED"

Description

LatticeECP DSP Block Multiplier. MULT9X9 is a combinational signed 9-bit by 9-bit multiplier used in the DSP block. The value represented in the 9-bit input A is multiplied by the value represented in the 9-bit input B. Output P is the 18-bit product of A and B. MULT9X9 may be represented as either unsigned or two's complement signed. The primitive consists of three types of optional pipeline registers:

- Input registers, located before the multipliers and registering the operands
- Multiplier pipeline registers, located after the multipliers and product registration
- Output registers, located before leaving the block, and registering the mode-specific output.

See the description of MULT18X18 for more information on control signals for DSP blocks and attributes.

Refer to the following technical note on the Lattice web site.

- TN1057 - LatticeECP sysDSP Usage Guide

MULT9X9 pin functions:

Function	Pins
input data A and B	A[8:0], B[8:0]
signed input (0 = unsigned, 1 = signed)	SIGNEDAB
clock enable	CE[0:3]
clock input	CLK[0:3]
reset	RST[0:3]

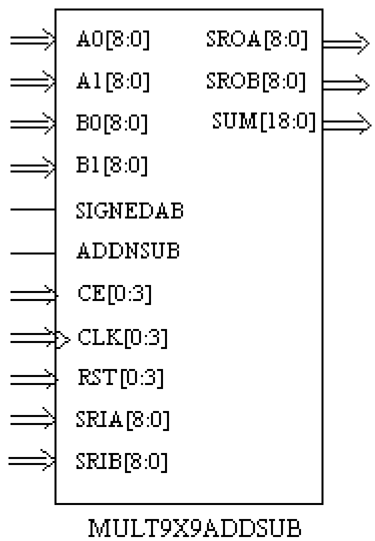
Function	Pins
shifted input A and B (from previous stage)	SRIA[8:0], SRIB[8:0]
shifted output A and B (from previous stage)	SROA[8:0], SROB[8:0]
output product data	P[17:0]

MULT9X9ADDSUB

ECP DSP Multiplier Add/Subtract

Architectures Supported:

- LatticeECP (DSP Blocks Only)



INPUTS: A08, A07, A06, A05, A04, A03, A02, A01, A00, A18, A17, A16, A15, A14, A13, A12, A11, A10, B08, B07, B06, B05, B04, B03, B02, B01, B00, B18, B17, B16, B15, B14, B13, B12, B11, B10, SIGNEDAB, ADDNSUB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, SUM18, SUM17, SUM16, SUM15, SUM14, SUM13, SUM12, SUM11, SUM10, SUM9, SUM8, SUM7, SUM6, SUM5, SUM4, SUM3, SUM2, SUM1, SUM0

ATTRIBUTES:

REG_INPUTA0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

SHIFT_IN_A0: "FALSE" (default), "TRUE"

SHIFT_IN_B0: "FALSE" (default), "TRUE"

SHIFT_IN_A1: "FALSE" (default), "TRUE"

SHIFT_IN_B1: "FALSE" (default), "TRUE"

GSR: "ENABLED" (default), "DISABLED"

Description

LatticeECP DSP Block Adder/Subtractor. MULT18X18ADDSUB can be configured to either add or subtract its inputs, adding or subtracting the inputs from two multiplier products. The add/subtract control is either configured as a static HIGH (Vcc), LOW (GND), or as dynamic control signals ADDNSUB1 and ADDNSUB3. In Lattice Diamond, the static settings are implemented by setting ADDNSUB1 and ADDNSUB3 signal to Vcc or GND in the CIB ISB.

The primitive consists of three types of optional pipeline registers:

- Input registers, located before the multipliers and registering the operands
- Multiplier pipeline registers, located after the multipliers and product registration
- Output registers, located before leaving the block, and registering the mode-specific output.

See the description of MULT18X18 for more information on control signals for DSP blocks and attributes.

Refer to the following technical note on the Lattice web site.

- TN1057 - LatticeECP sysDSP Usage Guide

MULT9X9ADDSUB pin functions:

Function	Pins
input data A and B	A0|1[8:0], B0|1[8:0]
signed input (0 = unsigned, 1 = signed)	SIGNEDAB
add/subtract (0 = add, 1 = subtract)	ADDNSUB
clock enable	CE[0:3]
clock input	CLK[0:3]

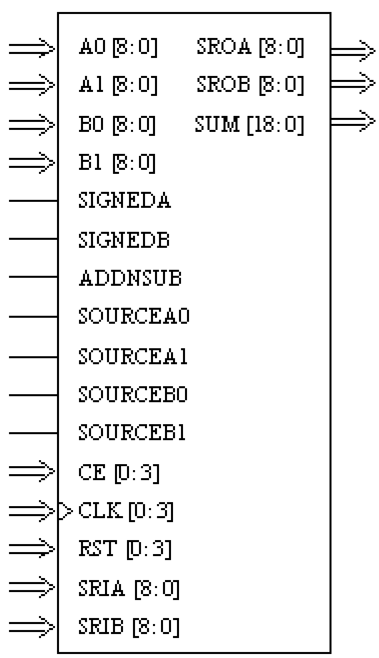
Function	Pins
reset	RST[0:3]
shifted input A and B (from previous stage)	SRIA[8:0], SRIB[8:0]
shifted output A and B (from previous stage)	SROA[8:0], SROB[8:0]
output product sum data	SUM[18:0]

MULT9X9ADDSUBB

DSP Multiplier Add/Subtract

Architectures Supported:

- LatticeECP2/M
- LatticeECP3 (*for LatticeECP2/M backward compatibility only*)
- LatticeXP2



MULT9X9ADDSUBB

INPUTS: A08, A07, A06, A05, A04, A03, A02, A01, A00, A18, A17, A16, A15, A14, A13, A12, A11, A10, B08, B07, B06, B05, B04, B03, B02, B01, B00, B18, B17, B16, B15, B14, B13, B12, B11, B10, SIGNEDA, SIGNEDB, ADDNSUB, SOURCEA0, SOURCEA1, SOURCEB0, SOURCEB1, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, SUM18, SUM17, SUM16, SUM15, SUM14, SUM13, SUM12, SUM11, SUM10, SUM9, SUM8, SUM7, SUM6, SUM5, SUM4, SUM3, SUM2, SUM1, SUM0

ATTRIBUTES:

REG_INPUTA0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

Description

Refer to the following technical notes on the Lattice web site.

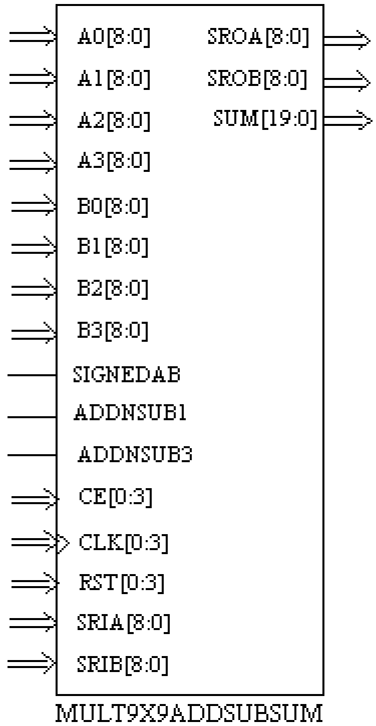
- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide

MULT9X9ADDSUBSUM

ECP DSP Adder/Subtractor/Sum

Architectures Supported:

- LatticeECP (DSP Blocks Only)



INPUTS: A08, A07, A06, A05, A04, A03, A02, A01, A00, A18, A17, A16, A15, A14, A13, A12, A11, A10, A28, A27, A26, A25, A24, A23, A22, A21, A20, A38, A37, A36, A35, A34, A33, A32, A31, A30, B08, B07, B06, B05, B04, B03, B02, B01, B00, B18, B17, B16, B15, B14, B13, B12, B11, B10, B28, B27, B26, B25, B24, B23, B22, B21, B20, B38, B37, B36, B35, B34, B33, B32, B31, B30, SIGNEDAB, ADDNSUB1, ADDNSUB3, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, SUM19, SUM18, SUM17, SUM16, SUM15, SUM14, SUM13, SUM12, SUM11, SUM10, SUM9, SUM8, SUM7, SUM6, SUM5, SUM4, SUM3, SUM2, SUM1, SUM0

ATTRIBUTES:

REG_INPUTA0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE0_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE1_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB1_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB1_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB1_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB1_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB1_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB1_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB3_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB3_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB3_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB3_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB3_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB3_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

SHIFT_IN_A0: "FALSE" (default), "TRUE"

SHIFT_IN_B0: "FALSE" (default), "TRUE"

SHIFT_IN_A1: "FALSE" (default), "TRUE"

SHIFT_IN_B1: "FALSE" (default), "TRUE"

SHIFT_IN_A2: "FALSE" (default), "TRUE"

SHIFT_IN_B2: "FALSE" (default), "TRUE"

SHIFT_IN_A3: "FALSE" (default), "TRUE"

SHIFT_IN_B3: "FALSE" (default), "TRUE"

GSR: "ENABLED" (default), "DISABLED"

Description

LatticeECP DSP Block Adder/Subtractor. MULT9X9ADDSUBSUM can be configured to either add or subtract its inputs, adding or subtracting the inputs from two multiplier products. The add/subtract control is either configured as a static HIGH (Vcc), LOW (GND), or as dynamic control signals ADDNSUB1 and ADDNSUB3. In Lattice Diamond, the static settings are implemented by setting ADDNSUB1 and ADDNSUB3 signal to Vcc or GND in the CIB ISB.

The primitive consists of three types of optional pipeline registers:

- Input registers, located before the multipliers and registering the operands
- Multiplier pipeline registers, located after the multipliers and product registration
- Output registers, located before leaving the block, and registering the mode-specific output.

See the description of MULT18X18 for more information on control signals for DSP blocks and attributes.

Refer to the following technical note on the Lattice web site.

- TN1057 - LatticeECP sysDSP Usage Guide

MULT9X9ADDSUBSUM pin functions:

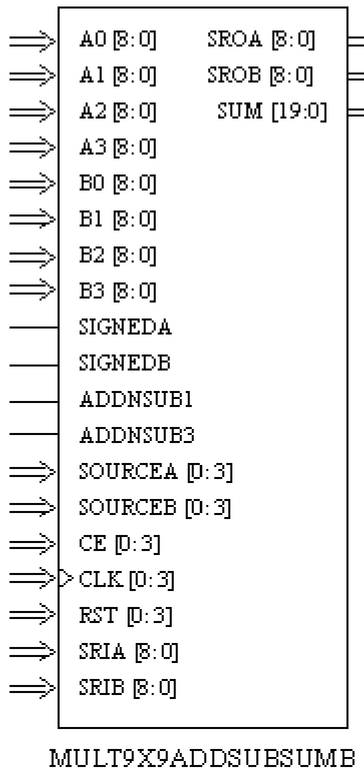
Function	Pins
input data A and B	A0|1|2|3[8:0], B0|1|2|3[8:0]
signed input (0 = unsigned, 1 = signed)	SIGNEDAB
add/subtract (0 = add, 1 = subtract)	ADDNSUB1, ADDNSUB3
clock enable	CE[0:3]
clock input	CLK[0:3]
reset	RST[0:3]
shifted input A and B (from previous stage)	SRIA[8:0], SRIB[8:0]
shifted output A and B (from previous stage)	SROA[8:0], SROB[8:0]
output product sum data	SUM[19:0]

MULT9X9ADDSUBSUMB

DSP Multiplier Add/Subtract/Sum

Architectures Supported:

- LatticeECP2/M
- LatticeECP3 (*for LatticeECP2/M backward compatibility only*)
- LatticeXP2



INPUTS: A08, A07, A06, A05, A04, A03, A02, A01, A00, A18, A17, A16, A15, A14, A13, A12, A11, A10, A28, A27, A26, A25, A24, A23, A22, A21, A20, A38, A37, A36, A35, A34, A33, A32, A31, A30, B08, B07, B06, B05, B04, B03, B02, B01, B00, B18, B17, B16, B15, B14, B13, B12, B11, B10, B28, B27, B26, B25, B24, B23, B22, B21, B20, B38, B37, B36, B35, B34, B33, B32, B31, B30, SIGNEDA, SIGNEDB, ADDNSUB1, ADDNSUB3, SOURCEA0, SOURCEA1, SOURCEA2, SOURCEA3, SOURCEB0, SOURCEB1, SOURCEB2, SOURCEB3, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, SUM19, SUM18, SUM17, SUM16, SUM15,

SUM14, SUM13, SUM12, SUM11, SUM10, SUM9, SUM8, SUM7, SUM6, SUM5, SUM4, SUM3, SUM2, SUM1, SUM0

ATTRIBUTES:

REG_INPUTA0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTA3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE2_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE2_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE2_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE3_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE3_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE3_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB1_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB1_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB1_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB1_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB1_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB1_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB3_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB3_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB3_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ADDNSUB3_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ADDNSUB3_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ADDNSUB3_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

Description

Refer to the following technical notes on the Lattice web site.

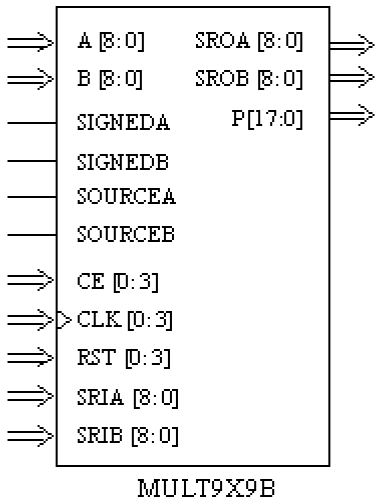
- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide

MULT9X9B

DSP Multiplier

Architectures Supported:

- LatticeECP2/M
- LatticeECP3 (*for LatticeECP2/M backward compatibility only*)
- LatticeXP2



INPUTS: A8, A7, A6, A5, A4, A3, A2, A1, A0, B8, B7, B6, B5, B4, B3, B2, B1, B0, SIGNEDA, SIGNEDB, SOURCEA, SOURCEB, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, P17, P16, P15, P14, P13, P12, P11, P10, P9, P8, P7, P6, P5, P4, P3, P2, P1, P0

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDB_RST: "RST0" (default), "RST1", "RST2", "RST3"

GSR: "ENABLED" (default), "DISABLED"

Description

Refer to the following technical notes on the Lattice web site.

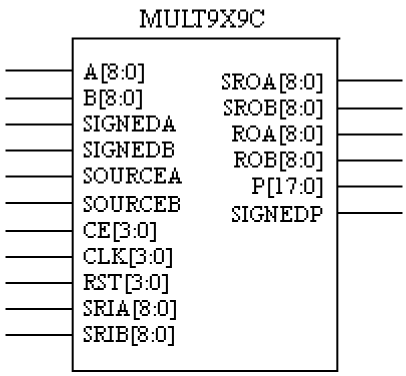
- TN1182 - LatticeECP3 sysDSP Usage Guide
- TN1107 - LatticeECP2/M sysDSP Usage Guide
- TN1140 - LatticeXP2 sysDSP Usage Guide

MULT9X9C

DSP Multiplier

Architectures Supported:

- LatticeECP3



INPUTS: A8, A7, A6, A5, A4, A3, A2, A1, A0, B8, B7, B6, B5, B4, B3, B2, B1, B0, SIGNEDA, SIGNEDB, SOURCEA, SOURCEB, CE3, CE2, CE1, CE0, CLK3, CLK2, CLK1, CLK0, RST3, RST2, RST1, RST0, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, ROA8, ROA7, ROA6, ROA5, ROA4, ROA3, ROA2, ROA1, ROA0, ROB8, ROB7, ROB6, ROB5, ROB4, ROB3, ROB2, ROB1, ROB0, P17, P16, P15, P14, P13, P12, P11, P10, P9, P8, P7, P6, P5, P4, P3, P2, P1, P0, SIGNEDP

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

CAS_MATCH_REG: "FALSE" (default), "TRUE"

MULT_BYPASS: "DISABLED" (default), "ENABLED"

GSR: "ENABLED" (default), "DISABLED"

RESETMODE: "SYNC" (default), "ASYNC"

MULT9X9C Port Description

I/O	Port Name	Capture Name	Type	Size (Buses Only)	Description
I	A	A[8:0]	Bus	8:0	A input
I	B	B[8:0]	Bus	8:0	B input
I	SRIA	SRIA[8:0]	Bus	8:0	Shift input A
I	SRIB	SRIB[8:0]	Bus	8:0	Shift input B
I	SIGNEDA	SIGNEDA	Bit	N/A	Input A sign selection

I/O	Port Name	Capture Name	Type	Size (Buses Only)	Description
I	SIGNEDB	SIGNEDB	Bit	N/A	Input B sign selection
I	SOURCEA	SOURCEA	Bit	N/A	Source A selection
I	SOURCEB	SOURCEB	Bit	N/A	Source B selection
I	CLK0	CLK0	Bit	N/A	Clock Input
I	CLK1	CLK1	Bit	N/A	Clock Input
I	CLK2	CLK2	Bit	N/A	Clock Input
I	CLK3	CLK3	Bit	N/A	Clock Input
I	CE0	CE0	Bit	N/A	Clock Enable Input
I	CE1	CE1	Bit	N/A	Clock Enable Input
I	CE2	CE2	Bit	N/A	Clock Enable Input
I	CE3	CE3	Bit	N/A	Clock Enable Input
I	RST0	RST0	Bit	N/A	Reset Input
I	RST1	RST1	Bit	N/A	Reset Input
I	RST2	RST2	Bit	N/A	Reset Input
I	RST3	RST3	Bit	N/A	Reset Input
O	P	P[17:0]	Bus	17:0	Product
O	SROA	SROA[8:0]	Bus	8:0	Shift A Output
O	SROB	SROB[8:0]	Bus	8:0	Shift B Output
O	ROA	ROA[8:0]	Bus	8:0	Registered Output A from Multiplier
O	ROB	ROB[8:0]	Bus	8:0	Registered Output B from Multiplier
O	SIGNEDP	SIGNEDP	Bit	N/A	Output Sign Bit (result of SignedA or SignedB)

MULT9X9C Attribute Description

Name	Value	Default	Description
REG_INPUTA_CLK	"NONE", "CLK0", "CLK1", "CLK2", "CLK3"	"NONE"	Input A clock selection
REG_INPUTA_CE	"CE0", "CE1", "CE2", "CE3"	"CE0"	Input A clock enable selection
REG_INPUTA_RST	"RST0", "RST1", "RST2", "RST3"	"RST0"	Input A reset selection

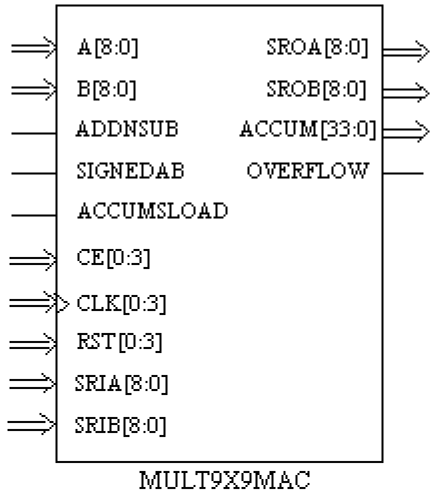
Name	Value	Default	Description
REG_INPUTB_CLK	"NONE", "CLK0", "CLK1", "CLK2", "CLK3"	"NONE"	Input B clock selection
REG_INPUTB_CE	"CE0", "CE1", "CE2", "CE3"	"CE0"	Input B clock enable selection
REG_INPUTB_RST	"RST0", "RST1", "RST2", "RST3"	"RST0"	Input B reset selection
REG_PIPELINE_CLK	"NONE", "CLK0", "CLK1", "CLK2", "CLK3"	"NONE"	Pipeline clock selection
REG_PIPELINE_CE	"CE0", "CE1", "CE2", "CE3"	"CE0"	Pipeline clock enable selection
REG_PIPELINE_RST	"RST0", "RST1", "RST2", "RST3"	"RST0"	Pipeline reset selection
REG_OUTPUT_CLK	"NONE", "CLK0", "CLK1", "CLK2", "CLK3"	"NONE"	Output clock selection
REG_OUTPUT_CE	"CE0", "CE1", "CE2", "CE3"	"CE0"	Output clock enable selection
REG_OUTPUT_RST	"RST0", "RST1", "RST2", "RST3"	"RST0"	Output reset selection
CAS_MATCH_REG	"FALSE", "TRUE"	"FALSE"	Cascade match register option
MULT_BYPASS	"DISABLED", "ENABLED"	"ENABLED"	Multiplier bypass option
RESETMODE	"SYNC", "ASYNC"	"SYNC"	Global set reset selection
GSR	"ENABLED", "DISABLED"	"ENABLED"	Reset mode selection

MULT9X9MAC

ECP DSP Multiplier

Architectures Supported:

- LatticeECP (DSP Blocks Only)



INPUTS: A8, A7, A6, A5, A4, A3, A2, A1, A0, B8, B7, B6, B5, B4, B3, B2, B1, B0, ADDNSUB, SIGNEDAB, ACCUMSLOAD, CE0, CE1, CE2, CE3, CLK0, CLK1, CLK2, CLK3, RST0, RST1, RST2, RST3, SRIA8, SRIA7, SRIA6, SRIA5, SRIA4, SRIA3, SRIA2, SRIA1, SRIA0, SRIB8, SRIB7, SRIB6, SRIB5, SRIB4, SRIB3, SRIB2, SRIB1, SRIB0

OUTPUTS: SROA8, SROA7, SROA6, SROA5, SROA4, SROA3, SROA2, SROA1, SROA0, SROB8, SROB7, SROB6, SROB5, SROB4, SROB3, SROB2, SROB1, SROB0, ACCUM33, ACCUM32, ACCUM31, ACCUM30, ACCUM29, ACCUM28, ACCUM27, ACCUM26, ACCUM25, ACCUM24, ACCUM23, ACCUM22, ACCUM21, ACCUM20, ACCUM19, ACCUM18, ACCUM17, ACCUM16, ACCUM15, ACCUM14, ACCUM13, ACCUM12, ACCUM11, ACCUM10, ACCUM9, ACCUM8, ACCUM7, ACCUM6, ACCUM5, ACCUM4, ACCUM3, ACCUM2, ACCUM1, ACCUM0, OVERFLOW

ATTRIBUTES:

REG_INPUTA_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTA_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTA_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_INPUTB_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_INPUTB_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_INPUTB_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_PIPELINE_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_PIPELINE_CE: "CE0" (default), "CE1", "CE1", "CE3"

REG_PIPELINE_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_OUTPUT_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_OUTPUT_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_OUTPUT_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_SIGNEDAB_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_SIGNEDAB_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_SIGNEDAB_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ACCUMSLOAD_0_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ACCUMSLOAD_0_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ACCUMSLOAD_0_RST: "RST0" (default), "RST1", "RST2", "RST3"

REG_ACCUMSLOAD_1_CLK: "NONE" (default), "CLK0", "CLK1", "CLK2", "CLK3"

REG_ACCUMSLOAD_1_CE: "CE0" (default), "CE1", "CE2", "CE3"

REG_ACCUMSLOAD_1_RST: "RST0" (default), "RST1", "RST2", "RST3"

SHIFT_IN_A: "FALSE" (default), "TRUE"

SHIFT_IN_B: "FALSE" (default), "TRUE"

GSR: "ENABLED" (default), "DISABLED"

Description

MULT9X9MAC supports operand bit widths for 9X9 multiplication and accumulates the output up to 34 bits. The DSP block includes optional registers for the input and intermediate pipeline stage. Pipeline stages may be set using a pipeline attribute. The output registers are required for the accumulator. Signed and unsigned arithmetic are supported. The OVERFLOW bit is also provided when the accumulated results are in the overflow condition. ACCUMSLOAD determines the mode of operation for either loading the multiplier product or to accumulate.

The primitive consists of three types of optional pipeline registers:

- Input registers, located before the multipliers and registering the operands

- Multiplier pipeline registers, located after the multipliers and product registration
- Output registers, located before leaving the block, and registering the mode-specific output.

See the description of MULT18X18 for more information on control signals for DSP blocks.

Refer to the following technical note on the Lattice web site.

- TN1057 - LatticeECP sysDSP Usage Guide

MULT9X9MAC pin functions:

Function	Pins
input data A and B	A[8:0], B[8:0]
signed input (0 = unsigned, 1 = signed)	SIGNEDAB
add/subtract (0 = add, 1 = subtract)	ADDNSUB
Accumulate (HIGH) /Load (LOW) Mode	ACCUMSLOAD
clock enable	CE[0:3]
clock input	CLK[0:3]
reset	RST[0:3]
shifted input A and B (from previous stage)	SRIA[8:0], SRIB[8:0]
shifted output A and B (from previous stage)	SROA[8:0], SROB[8:0]
output data	ACCUM[33:0]
overflow	OVERFLOW

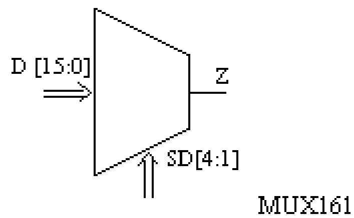
MUX161

16-Input Mux within the PFU (4 Slices)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO

- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, SD1, SD2, SD3, SD4

OUTPUT: Z

Description

For more usage, see related technical notes or contact technical support.

Truth Table

INPUTS	OUTPUTS	INPUTS	OUTPUTS
SD[4:1]	Z	SD[4:1]	Z
0000	D0	1000	D8
0001	D1	1001	D9
0010	D2	1010	D10
0011	D3	1011	D11
0100	D4	1100	D12
0101	D5	1101	D13
0110	D6	1110	D14
0111	D7	1111	D15

Note

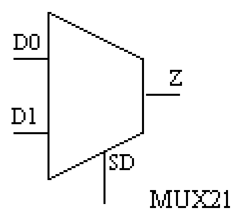
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

MUX21

2-to-1 Mux

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, SD

OUTPUT: Z

Description

For more usage, see related technical notes or contact technical support.

Truth Table

INPUTS			OUTPUTS
D0	D1	SD	Z
0	X	0	0
1	X	0	1
X	0	1	0
X	1	1	1

X = Don’t care

Note

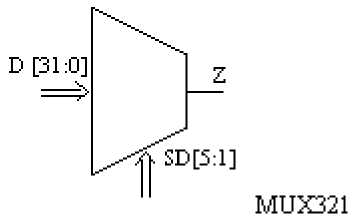
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

MUX321

32-Input Mux within the PFU (8 Slices)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16, D17, D18, D19, D20, D21, D22, D23, D24, D25, D26, D27, D28, D29, D30, D31, SD1, SD2, SD3, SD4, SD5

OUTPUT: Z

Description

For more usage, see related technical notes or contact technical support.

Truth Table

INPUTS	OUTPUTS	INPUTS	OUTPUTS
SD[5:1]	Z	SD[5:1]	Z
00000	D0	10000	D16
00001	D1	10001	D17
00010	D2	10010	D18
00011	D3	10011	D19
00100	D4	10100	D20
00101	D5	10101	D21
00110	D6	10110	D22
00111	D7	10111	D23
01000	D8	11000	D24
01001	D9	11001	D25
01010	D10	11010	D26
01011	D11	11011	D27
01100	D12	11100	D28
01101	D13	11101	D29
01110	D14	11110	D30
01111	D15	11111	D31

Note

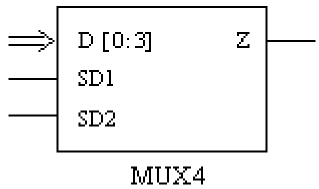
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

MUX4

4-bit Multiplexer

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: D0, D1, D2, D3, SD1, SD2

OUTPUT: Z

Description

For more usage, see related technical notes or contact technical support.

Note

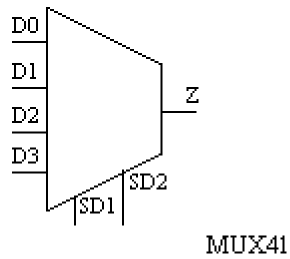
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

MUX41

4 to 1 Mux

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: D0, D1, D2, D3, SD1, SD2

OUTPUT: Z

Description

For more usage, see related technical notes or contact technical support.

Truth Table

INPUTS						OUTPUTS
D0	D1	D2	D3	SD1	SD2	Z
0	X	X	X	0	0	0
1	X	X	X	0	0	1
X	0	X	X	1	0	0
X	1	X	X	1	0	1
X	X	0	X	0	1	0
X	X	1	X	0	1	1
X	X	X	0	1	1	0
X	X	X	1	1	1	1

X = Don’t care

Note

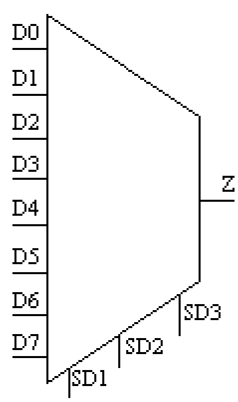
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

MUX81

8 to 1 Mux

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



MUX81

INPUTS: D0, D1, D2, D3, D4, D5, D6, D7, SD1, SD2, SD3

OUTPUT: Z

Description

For more usage, see related technical notes or contact technical support.

Truth Table

INPUTS									OUTPUTS
D0	D1	D2	D3	D4	D5	D6	D7	SD[1:3]	Z
0	X	X	X	X	X	X	X	0 0 0	0
1	X	X	X	X	X	X	X	0 0 0	1
X	0	X	X	X	X	X	X	1 0 0	0
X	1	X	X	X	X	X	X	1 0 0	1

INPUTS									OUTPUTS
X	X	0	X	X	X	X	X	0 1 0	0
X	X	1	X	X	X	X	X	0 1 0	1
X	X	X	0	X	X	X	X	1 1 0	0
X	X	X	1	X	X	X	X	1 1 0	1
X	X	X	X	0	X	X	X	0 0 1	0
X	X	X	X	1	X	X	X	0 0 1	1
X	X	X	X	X	0	X	X	1 0 1	0
X	X	X	X	X	1	X	X	1 0 1	1
X	X	X	X	X	X	0	X	0 1 1	0
X	X	X	X	X	X	1	X	0 1 1	1
X	X	X	X	X	X	X	0	1 1 1	0
X	X	X	X	X	X	X	1	1 1 1	1

X = Don’t care

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

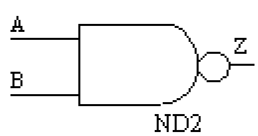
N

ND2

2 Input NAND Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B

OUTPUT: Z

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

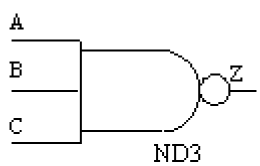
ND3

3 Input NAND Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M

- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C

OUTPUT: Z

Note

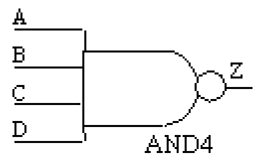
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

ND4

4 Input NAND Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D

OUTPUT: Z

Note

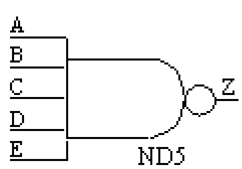
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

ND5

5 Input NAND Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D, E

OUTPUT: Z

Note

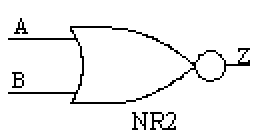
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

NR2

2 Input NOR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B

OUTPUT: Z

Note

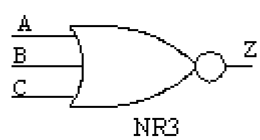
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

NR3

3 Input NOR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C

OUTPUT: Z

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

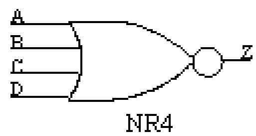
NR4

4 Input NOR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M

- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D

OUTPUT: Z

Note

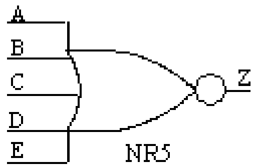
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

NR5

5 Input NOR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D, E

OUTPUT: Z

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

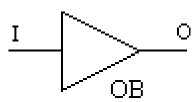
O

OB

Output Buffer

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUT: I

OUTPUT: O

Truth Table

INPUTS	OUTPUTS
I	O
1	1
0	0
Z	U

U = Unknown

Note

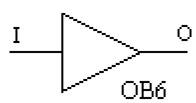
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OB6

6mA Sink 3mA Source Sinklim Output Buffer

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUT: I

OUTPUT: O

Truth Table

INPUTS	OUTPUTS
I	O
1	1
0	0

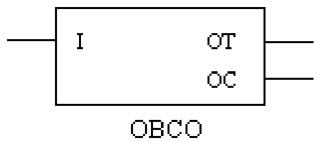
OBCO

Output Complementary Buffer

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeXP
- LatticeXP2
- MachXO2

- MachXO3L
- Platform Manager 2



INPUT: I

OUTPUTS: OT, OC

Note

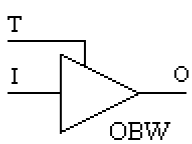
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OBW

Output Buffer with Tristate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



INPUTS: I, T

OUTPUT: O

Truth Table

INPUTS		OUTPUTS
I	T	O
0	1	weak 0
1	1	weak 1

Note

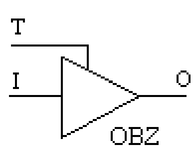
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OBZ

Output Buffer with Tristate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: I, T

OUTPUT: O

Truth Table

INPUTS		OUTPUTS
I	T	O
X	1	Z
0	0	0
1	0	1

X = Don't care

When TSALL=0, O=Z

Note

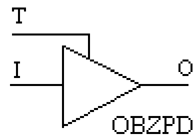
- For PU/PD buffers, when TSALL=0, O will be pulled up or pulled down, respectively. The letters PU (PD) in the buffer name indicate that a pull-up (pull-down) primitive is available. This is used to generate a logic high level (low level) for nodes that may be floating.
- This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OBZPD

Output Buffer with Tristate and Pull-down

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



INPUTS: I, T

OUTPUT: O

Truth Table

INPUTS		OUTPUTS
I	T	O
X	1	Z
0	0	0
1	0	1

X = Don't care

When TSALL=0, O=Z

Note

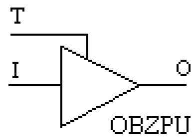
- For PU/PD buffers, when TSALL=0, O will be pulled up or pulled down, respectively. The letters PU (PD) in the buffer name indicate that a pull-up (pull-down) primitive is available. This is used to generate a logic high level (low level) for nodes that may be floating.
- This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OBZPU

Output Buffer with Tristate and Pull-up

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: I, T

OUTPUT: O

Truth Table

INPUTS		OUTPUTS
I	T	O
X	1	Z
0	0	0
1	0	1

X = Don’t care

When TSALL=0, O=Z

Note

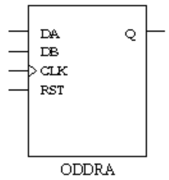
- For PU/PD buffers, when TSALL=0, O will be pulled up or pulled down, respectively. The letters PU (PD) in the buffer name indicate that a pull-up (pull-down) primitive is available. This is used to generate a logic high level (low level) for nodes that may be floating.
- This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

ODDRA

Output DDR

Architectures Supported:

- LatticeSC/M

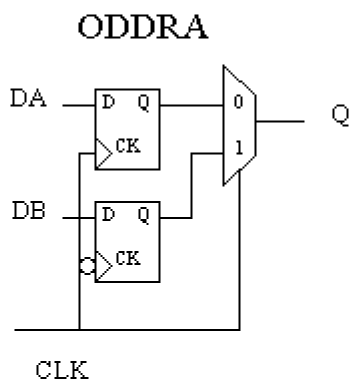


INPUTS: DA, DB, CLK, RST

OUTPUT: Q

Description

Output DDR data (positive edge and negative edge data) to the buffer. The following symbolic diagram shows the flip-flop structure of this primitive.

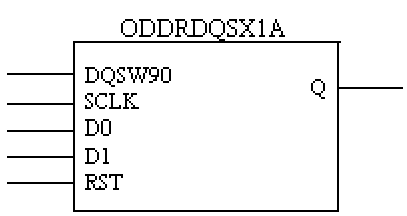


ODDRDQSX1A

Output for DDR1/2 Memory

Architectures Supported:

- MachXO2
- Platform Manager 2



INPUTS: DQSW90, SCLK, D0, D1, RST

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

ODDRDQSX1A is the output for DDR1/2 memory using the PIC hardware cell. It is used for right side only. See the below table for the port description.

Signal	I/O	Description
DQSW90	I	Shifts the DQS signal by 90 degree, from DQSBUFH
SCLK	I	Clock from the CIB
D0	I	Data A primary phase of data (first out)
D1	I	Data B secondary phase of data (second out)
RST	I	RESET to this block from the CIB
Q	O	DDR output data

For more information and usage, refer to the following technical note on the Lattice web site.

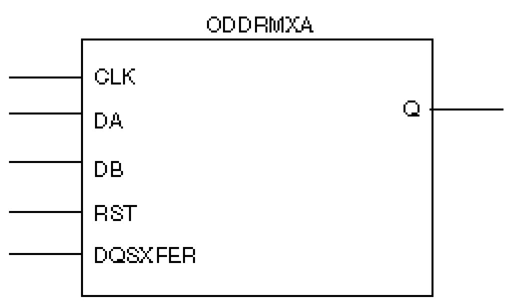
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

ODDRMXA

DDR Output Registers

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: CLK, DA, DB, RST, DQSXFER

OUTPUT: Q

Description

The ODDRMXA primitive implements the output register for both write and tristate functions. This primitive is used to output DDR data and DQS strobe to the memory. All DDR output tristate functions are also implemented using this primitive.

The following table provides description of all I/O ports associated with the ODDRMXA primitive.

Port Name	I/O	Definition
CLK	I	System CLK or ECLK
DA	I	Data at the negative edge of the clock
DB	I	Data at the positive edge of the clock
RST	I	Reset
DQSXFER	I	90-degree phase shifted clock coming from the DQSBUFC block
Q	O	DDR data to the memory

Notes:

- RST should be held low during DDR Write operation. By default the software will implement CE High and RST low.
- DDR output and tristate registers do not have CE support. RST is available for tristate DDRX mode (while reading). LSR will default to set when used in tristate mode.
- When asserting reset during DDR writes, it is important to know that it resets only the flip-flops but not the latches.

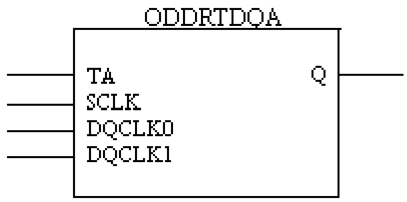
For more usage, see related technical notes or contact technical support.

ODDRTDQA

Tri-State for DQ: DDR3_MEM and DDR_GENX2

Architectures Supported:

- LatticeECP3



INPUTS: TA, SCLK, DQCLK1, DQCLK0

OUTPUT: Q

Description

ODDRTDQA is the tri-state for DQ used for DDR3_MEM (DDR3 memory mode) and DDR_GENX2.

- E and EA: DDR3_MEM and DDR_GENX2 (left/right)

See the below table for the port description.

Signal	I/O	Description
TA	I	Tri-state input.
SCLK	I	System clock.
DQCLK0	I	One clock edge, at the frequency of SCLK, used in output gearing, 90 degree out of phase from DQCLK1.
DQCLK1	I	One clock edge, at the frequency of SCLK, used in output gearing.
Q	O	Tri-state output.

For more information and usage, refer to the following technical note on the Lattice web site.

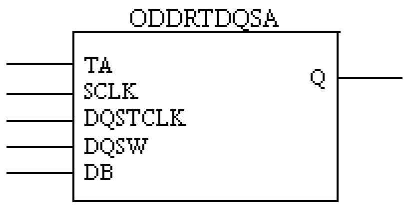
- TN1177 - LatticeECP3 sysIO Usage Guide

ODDRTDQSA

Tri-State for Single-Ended and Differential DQS: DDR_MEM, DDR2_MEM, and DDR3_MEM

Architectures Supported:

- LatticeECP3



INPUTS: TA, SCLK, DQSTCLK, DQSW, DB

OUTPUT: Q

Description

ODDRTDQSA is the tri-state for single-ended and differential DQS, used in DDR_MEM (DDR memory mode), DDR2_MEM (DDR2 memory mode), and DDR3_MEM (DDR3 memory mode).

- E and EA: DDR_MEM and DDR2_MEM (left/right/top)
- E and EA: DDR3_MEM (left/right)

See the below table for the port description.

Signal	I/O	Description
TA	I	Tri-state input
SCLK	I	System clock
DQSW	I	DQS write clock
DQSTCLK	I	DQS tri-state clock
DB	I	Data input (ONEGB)
Q	O	DQS tri-state output

For more information and usage, refer to the following technical note on the Lattice web site.

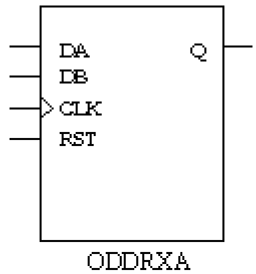
- TN1177 - LatticeECP3 sysIO Usage Guide

ODDRXA

Output DDR

Architectures Supported:

- LatticeSC/M

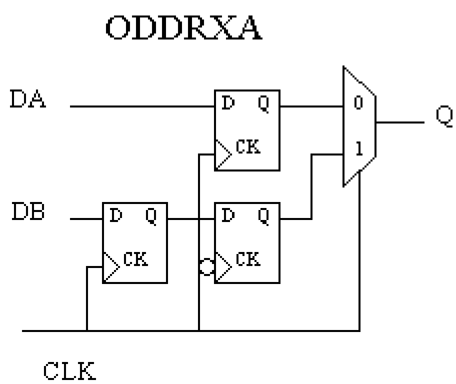


INPUTS: DA, DB, CLK, RST

OUTPUT: Q

Description

Output DDR data with half cycle clock domain transfer. The following symbolic diagram shows the flip-flop structure of this primitive.

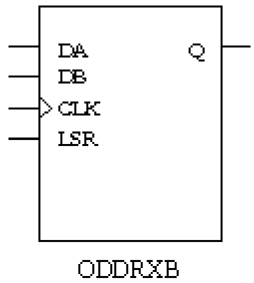


ODDRXB

Output DDR

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: DA, DB, CLK, LSR

OUTPUT: Q

ATTRIBUTES:

REGSET: "RESET" (default), "SET"

Description

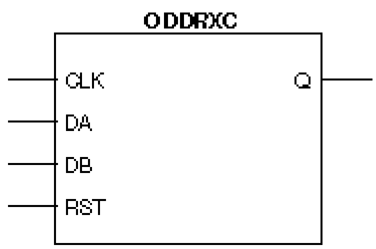
Output DDR data with half cycle clock domain transfer.

ODDRXC

DDR Generic Output

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: DA, DB, CLK, RST

OUTPUT: Q

Description

This DDR output module inputs two data streams and multiplexes them together to generate a single stream of data going to the sysIO™ buffer. CLK to this module can be connected to the edge clock or to the FPGA clock. This primitive is also used when DDR function is required for the tristate signal. See the following table for port description.

Port Name	I/O	Definition
DA	I	Data at the negative edge of the clock
DB	I	Data at the positive edge of the clock
CLK	I	Clock from CIB
RST	I	Reset signal
Q	O	DDR data to the memory

Notes:

- LSR should be held low during DDR Write operation. By default, the software will be implemented with CE High and LSR low.
- DDR output and tristate registers do not have CE support. LSR is available for the tristate DDRX mode (while reading). The LSR will default to set when used in the tristate mode.
- CE and LSR support is available for the regular (non-DDR) output mode.
- When asserting reset during DDR writes, it is important to keep in mind that this would only reset the flip-flops but not the latches.

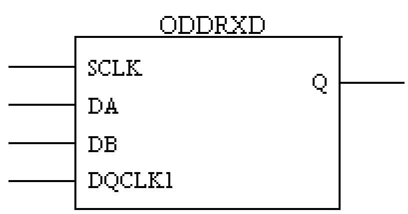
For more usage, see related technical notes or contact technical support.

ODDRXD

Output DDR for DDR_MEM, DDR2_MEM, DDR_GENX1, and DDR2_MEMGEN

Architectures Supported:

- LatticeECP3



INPUTS: SCLK, DA, DB, DQCLK1

OUTPUT: Q

ATTRIBUTES:

(*EA only*) MEMMODE: "DISABLED" (default), "ENABLED"

Description

ODDRXD is the output DDR for DDR_MEM (DDR memory mode), DDR2_MEM (DDR2 memory mode), DDR_GENX1 (DDR generic mode in X1 gearing), and DDR2_MEMGEN.

- E and EA: DDR_MEM, DDR2_MEM, and DDR2_MEMGEN (left/right/top)
- E: DDR_GENX1 (left/right/top)

The port information is described in the below table.

Signal	I/O	Description
SCLK	I	System clock.
DA	I	Data at the positive edge of the clock (OPOSA).
DB	I	Data at the negative edge of the clock (ONEGB).
DQCLK1	I	One clock edge, at the frequency of SCLK, used in output gearing.
Q	O	DDR data output.

For more information and usage, refer to the following technical note on the Lattice web site.

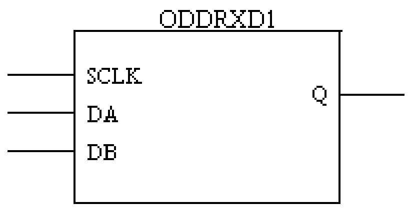
- TN1177 - LatticeECP3 sysIO Usage Guide

ODDRXD1

Output DDR for DDR_GENX1

Architectures Supported:

- LatticeECP3



INPUTS: SCLK, DA, DB

OUTPUT: Q

Description

ODDRXD1 is the output DDR for DDR_GENX1 (DDR generic mode in X1 gearing).

- EA: DDR_GENX1 (left/right/top)

The port information is described in the below table.

Signal	I/O	Description
SCLK	I	System clock
DA	I	Data at the positive edge of the clock (OPOSA)
DB	I	Data at the negative edge of the clock (ONEGB)
Q	O	DDR data output

For more information and usage, refer to the following technical note on the Lattice web site.

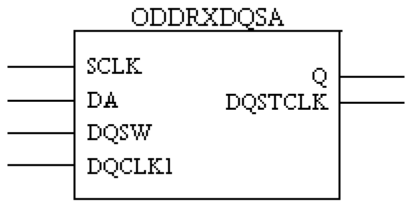
- TN1177 - LatticeECP3 sysIO Usage Guide

ODDRXDQSA

Output for Single-Ended and Differential DQS: DDR_MEM, DDR2_MEM, and DDR2_MEMGEN

Architectures Supported:

- LatticeECP3



INPUTS: SCLK, DA, DQSW, DQCLK1

OUTPUTS: Q, DQSTCLK

ATTRIBUTES:

(*EA only*) MEMMODE: "DISABLED" (default), "ENABLED"

Description

ODDRXDQSA is the output for single-ended and differential DQS, used for DDR_MEM (DDR memory mode), DDR2_MEM (DDR2 memory mode), and DDR2_MEMGEN.

- E and EA: DDR_MEM, DDR2_MEM, and DDR2_MEMGEN (left/right/top)

See the below table for the port description.

Signal	I/O	Description
SCLK	I	System clock.
DA	I	Data input (OPOSA).
DQSW	I	DQS write clock.
DQCLK1	I	One clock edge, at the frequency of SCLK, used in output gearing.
Q	O	DQS data output.
DQSTCLK	O	DQS Tri-state clock.

For more information and usage, refer to the following technical note on the Lattice web site.

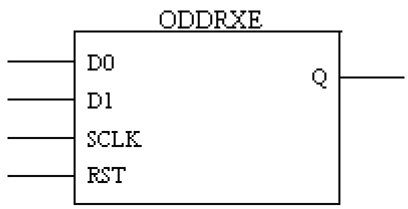
- TN1177 - LatticeECP3 sysIO Usage Guide

ODDRXE

Output for Generic DDR X1 Using 2:1 Gearing

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D0, D1, SCLK, RST

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

ODDRXE is the output for generic DDR X1 using 2:1 gearing. It uses the mPIC or PIC hardware cell. It is used for all sides.

The port information is described in the below table.

Signal	I/O	Description
D0	I	Data A primary phase of data (first out)
D1	I	Data B secondary phase of data (second out)
SCLK	I	Clock from the CIB
RST	I	RESET to this block from the CIB
Q	O	DDR output data

For more information and usage, refer to the following technical note on the Lattice web site.

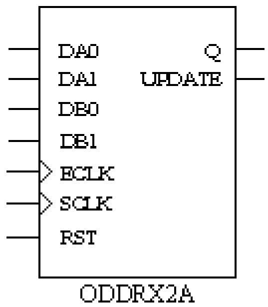
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

ODDRX2A

Output DDR

Architectures Supported:

▶ LatticeSC/M

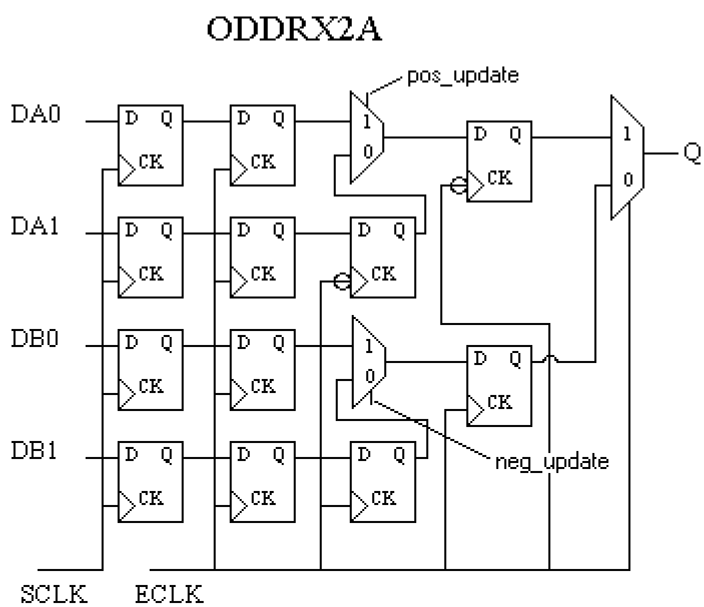


INPUTS: DA0, DA1, DB0, DB1, ECLK, SCLK, RST

OUTPUTS: Q, UPDATE

Description

Outputs DDR data to the buffer through the shift register and clock domain transfer from primary clock to edge clock. The following symbolic diagram shows the flip-flop structure of this primitive.



The pos_update and neg_update pins are select pins of the MUXes' internal

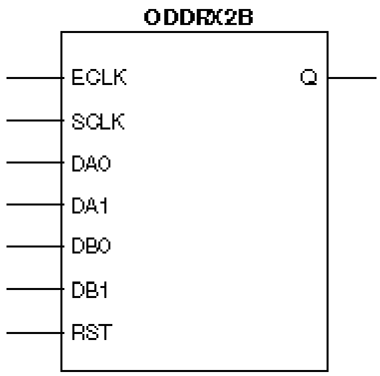
signals which go out as UPDATE signals depending upon the value of the UPDT parameter.

ODDRX2B

DDR Generic Output with 2x Gearing Ratio

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: DA0, DB0, DA1, DB1, ECLK, SCLK, RST

OUTPUT: Q

Description

This DDR output module can be used when a gearbox function is required. This primitive inputs four data streams and multiplexed them together to generate a single stream of data going to the sysIO buffer.

DDR registers of the complementary PIO is used when using this mode. The complementary PIO register can no longer be used to perform the DDR function. There are two clocks going to this primitive. ECLK is connected to the faster edge clock, while SCLK is connected to the slower FPGA clock. The DDR data output of this primitive is aligned to the faster edge clock.

Note that LSR should be held low during DDR Write operation. By default, the software will be implemented CE High and LSR low.

The following table lists port names and descriptions for the ODDRX2B primitive.

Port Name	I/O	Definition
DA0, DB0	I	Data at the negative edge of the clock
DA1, DB1	I	Data at the positive edge of the clock
ECLK	I	Clock connected to the faster edge clock

Port Name	I/O	Definition
SCLK	I	Clock connected to the slower edge clock
RST	I	Reset
Q	O	DDR data output

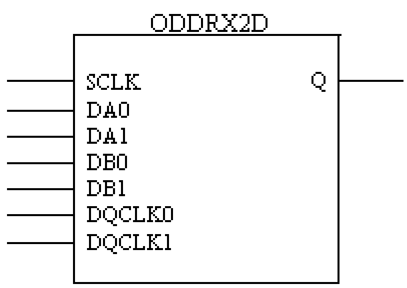
For more usage, see related technical notes or contact technical support.

ODDRX2D

Output DDR for DDR3_MEM and DDR_GENX2

Architectures Supported:

- LatticeECP3



INPUTS: SCLK, DA1, DB1, DA0, DB0, DQCLK1, DQCLK0

OUTPUT: Q

ATTRIBUTES:

ISI_CAL: "BYPASS" (default), "DEL1", "DEL2", "DEL3", "DEL4", "DEL5", "DEL6", "DEL7"

(*EA only*) MEMMODE: "DISABLED" (default), "ENABLED"

Description

ODDRX2D is the output DDR for DDR3_MEM (DDR3 memory mode) and DDR_GENX2 (DDR generic mode in X2 gearing).

- E and EA: DDR3_MEM and DDR_GENX2 (left/right)

The below table describes the port information.

Signal	I/O	Description
SCLK	I	System clock.
DA0	I	First data at the positive edge of the clock (OPOSA).
DA1	I	First data at the negative edge of the clock (OPOSB).
DB0	I	Second data at the positive edge of the clock (ONEGA).
DB1	I	Second data at the negative edge of the clock (ONEGB).
DQCLK0	I	One clock edge, at half the frequency of ECLK, used in output gearing, 90 degree out of phase from DQCLK1.
DQCLK1	I	One clock edge, at half the frequency of ECLK, used in output gearing.
Q	O	DDR data output.

For more information and usage, refer to the following technical note on the Lattice web site.

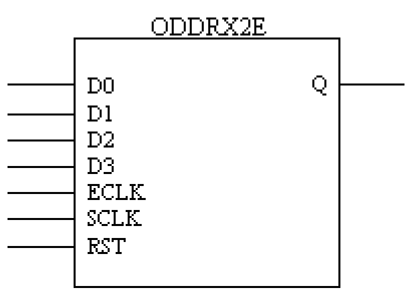
- TN1177 - LatticeECP3 sysIO Usage Guide

ODDRX2E

Output for Generic DDR X2 Using 4:1 Gearing

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D0, D1, D2, D3, ECLK, SCLK, RST

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

ODDRX2E is the output for generic DDR X2 using 4:1 gearing. It uses the VPIC_TX hardware cell. It is used for top bank only. See the below table for port information.

Signal	I/O	Description
D0, D2	I	Data at the same edge of the clock
D1, D3	I	Data at the same edge of the clock
ECLK	I	Edge clock
SCLK	I	Clock from the CIB
RST	I	RESET to this block from the CIB
Q	O	DDR output data

For more information and usage, refer to the following technical note on the Lattice web site.

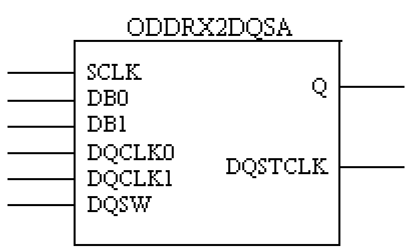
- TN1203 - Implementing High-Speed Interfaces with MachXO2 Devices

ODDRX2DQSA

Output for Differential DQS: DDR3_MEM

Architectures Supported:

- LatticeECP3



INPUTS: SCLK, DB1, DB0, DQCLK1, DQCLK0, DQSW

OUTPUTS: Q, DQSTCLK

ATTRIBUTES:

ISI_CAL: "BYPASS" (default), "DEL1", "DEL2", "DEL3", "DEL4", "DEL5", "DEL6", "DEL7"

(*EA only*) MEMMODE: "DISABLED" (default), "ENABLED"

Description

ODDRX2DQSA is the output for differential DQS used for DDR3_MEM (DDR3 memory mode).

- E and EA: DDR3_MEM (left/right)

See the below table for the port description.

Signal	I/O	Description
SCLK	I	System clock.
DB0	I	Data input (OPOSA).
DB1	I	Data input (OPOSB).
DQSW	I	DQS write clock.
DQCLK0	I	One clock edge, at half the frequency of ECLK, used in output gearing, 90 degree out of phase from DQCLK1.
DQCLK1	I	One clock edge, at half the frequency of SCLK, used in output gearing.
Q	O	DDR data output.
DQSTCLK	O	DQS Tri-state clock.

For more information and usage, refer to the following technical note on the Lattice web site.

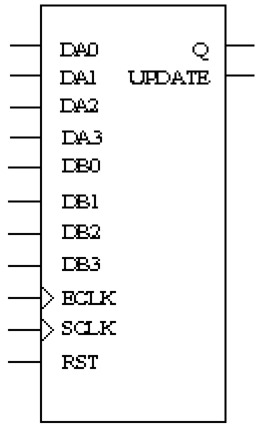
- TN1177 - LatticeECP3 sysIO Usage Guide

ODDRX4A

Output DDR

Architectures Supported:

- LatticeSC/M



ODDRX4A

INPUTS: DA0, DA1, DA2, DA3, DB0, DB1, DB2, DB3, ECLK, SCLK, RST

OUTPUTS: Q, UPDATE

ATTRIBUTES:

LSRMODE: "LOCAL" (default), "EDGE"

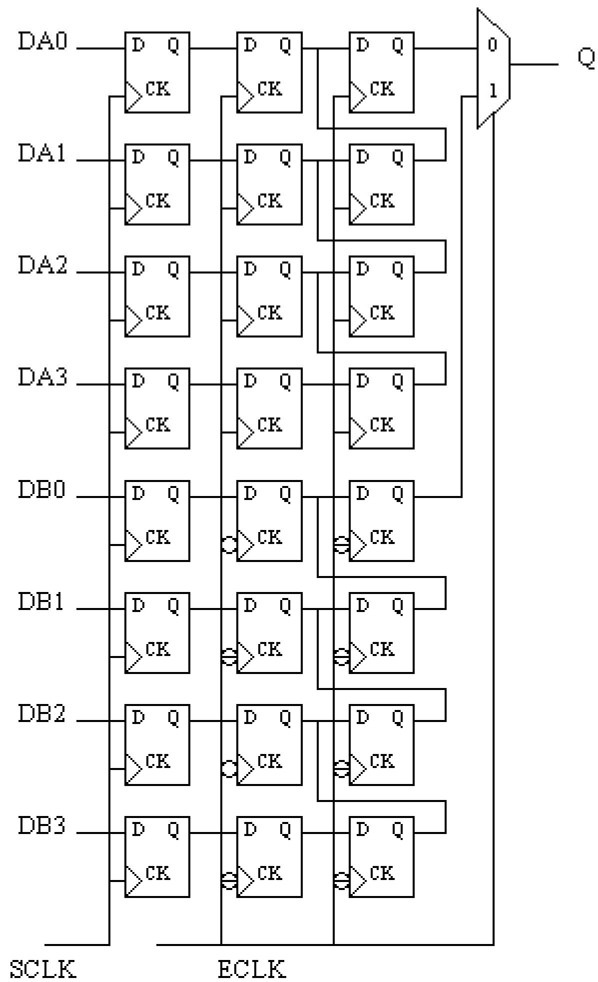
UPDT: "POS" (default), "NEG"

REGSET: "RESET" (default), "SET"

Description

Outputs DDR data to the buffer through the shift register and clock domain transfer from primary clock to edge clock. The following symbolic diagram shows the flip-flop structure of this primitive.

ODDRX4A

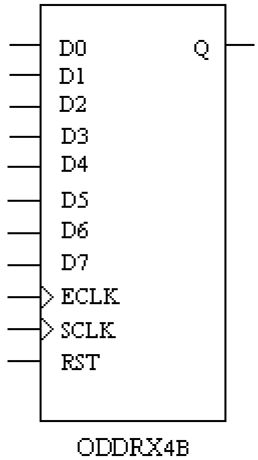


ODDRX4B

Output for Generic DDR X4 Using 8:1 Gearing

Architectures Supported:

- MachXO2
- Platform Manager 2



INPUTS: D0, D1, D2, D3, D4, D5, D6, D7, ECLK, SCLK, RST

OUTPUTS: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

ODDRX4B is the output for generic DDR X4 using 8:1 gearing. It uses the VPIC_TX hardware cell. It is used for top bank only. See the below table for the port description.

Signal	I/O	Description
D0, D2, D4, D6	I	Data at the same edge of the clock
D1, D3, D5, D7	I	Data at the same edge of the clock
ECLK	I	Edge clock
SCLK	I	Clock from the CIB
RST	I	RESET to this block from the CIB
Q	O	DDR output data

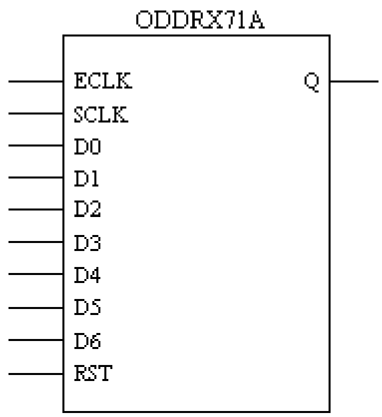
ODDRX71A

7:1 LVDS Output

Architectures Supported:

- MachXO2

- MachXO3L
- Platform Manager 2



INPUTS: ECLK, SCLK, D0, D1, D2, D3, D4, D5, D6, RST

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Description

ODDRX71A is the 7:1 LVDS output that supports 7:1 gearing. It is used for top bank only. See the below table for the port description.

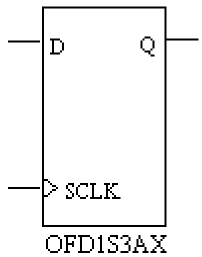
Signal	I/O	Description
ECLK	I	Edge clock
SCLK	I	Clock connected to the system clock
D0, D1, D2, D3, D4, D5, D6	I	Data available for 7:1 muxing
RST	I	RESET for this block
Q	O	7:1 LVDS signal output

OFD1S3AX

Positive Edge Triggered D Flip-Flop, GSR Used for Clear. Used to Tri-State DDR/DDR2

Architectures Supported:

- LatticeECP3



INPUTS: D, SCLK

OUTPUT: Q

ATTRIBUTES:

GSR: "DISABLED" (default), "ENABLED"

Description

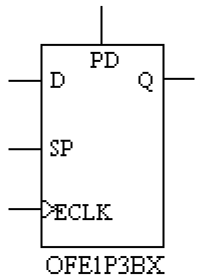
OFD1S3AX is a primitive used to implement DDR and DDR2 DQ tri-state. This primitive is functionally equivalent to the FD1S3AX primitive.

OFE1P3BX

Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and Edge Clock (used in output PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2



INPUTS: D, SP, ECLK, PD

OUTPUT: Q

Note

This primitive must be paired with an output or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				OUTPUTS
D	SP	ECLK	PD	Q
X	0	X	0	Q
X	X	X	1	1
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

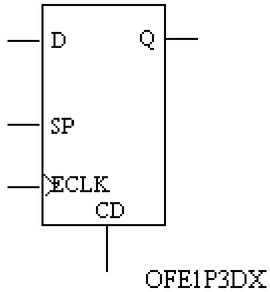
When GSR=0, Q=1 (D=SP=ECLK=PD=X)

OFE1P3DX

Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and Edge Clock (used in output PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2



INPUTS: D, SP, ECLK, CD

OUTPUT: Q

Note

This primitive must be paired with an output or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				OUTPUTS
D	SP	ECLK	CD	Q
X	0	X	0	Q
X	X	X	1	0
0	1	↑	0	0
1	1	↑	0	1

X = Don't care

When GSR=0, Q=0 (D=SP=ECLK=CD=X)

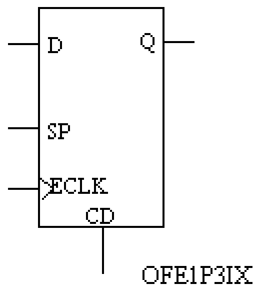
OFE1P3IX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and Edge Clock (used in output PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M

- LatticeSC/M
- LatticeXP
- LatticeXP2



INPUTS: D, SP, ECLK, CD

OUTPUT: Q

Note:

This primitive must be paired with an output or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				**OUTPUTS**
D	SP	ECLK	CD	Q
X	0	X	0	Q
X	X	↑	1	0
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

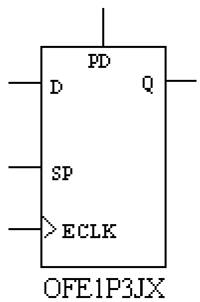
When GSR=0, Q=0 (D=SP=ECLK=CD=X)

OFE1P3JX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and Edge Clock (used in output PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2



INPUTS: D, SP, ECLK, PD

OUTPUT: Q

Note

This primitive must be paired with an output or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				OUTPUTS
D	SP	ECLK	PD	Q
X	0	X	0	Q
X	X	↑	1	1
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

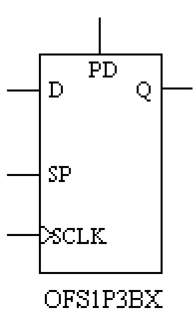
When GSR=0, Q=1 (D=SP=ECLK=PD=X)

OFS1P3BX

Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Preset, and System Clock (used in output PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, SP, SCLK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

This primitive must be paired with an output or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				OUTPUTS
D	SP	SCLK	PD	Q
X	0	X	0	Q
X	X	X	1	1
0	1	↑	0	0
1	1	↑	0	1

X = Don't care

When GSR=0, Q=1 (D=SP=SCLK=PD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

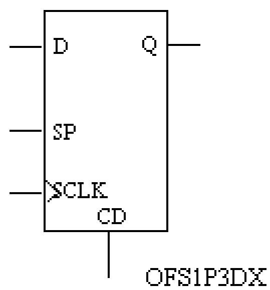
OFS1P3DX

Positive Edge Triggered D Flip-Flop with Positive Level Enable, Positive Level Asynchronous Clear, and System Clock (used in output PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3

- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, SP, SCLK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

This primitive must be paired with an output or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				**OUTPUTS**
D	SP	SCLK	CD	Q
X	0	X	0	Q
X	X	X	1	0
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

When GSR=0, Q=0 (D=SP=SCLK=CD=X)

Note

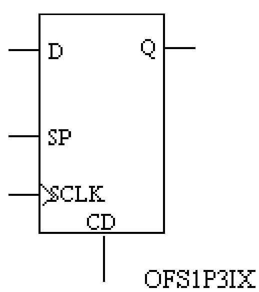
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OFS1P3IX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear, Positive Level Enable (Clear overrides Enable), and System Clock (used in output PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, SP, SCLK, CD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

This primitive must be paired with an output or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				OUTPUTS
D	SP	SCLK	CD	Q
X	0	X	0	Q
X	X	↑	1	0
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

When GSR=0, Q=0 (D=SP=SCLK=CD=X)

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

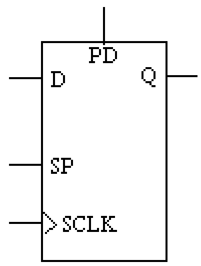
OFS1P3JX

Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset, Positive Level Enable (Preset overrides Enable), and System Clock (used in output PIC area only)

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2

- MachXO3L
- Platform Manager 2



INPUTS: D, SP, SCLK, PD

OUTPUT: Q

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

Note

This primitive must be paired with an output or bidirectional buffer. The mapper automatically assigns the primitive and its buffer to the same PIC. Use the PIN or LOC properties, or the LOCATE COMP preference, on the buffer per normal use, but not on this primitive.

Truth Table

INPUTS				**OUTPUTS**
D	SP	SCLK	PD	Q
X	0	X	0	Q
X	X	↑	1	1
0	1	↑	0	0
1	1	↑	0	1

X = Don’t care

When GSR=0, Q=1 (D=SP=SCLK=PD=X)

Note

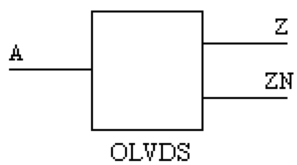
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OLVDS

LVDS Output Buffer

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUT: A

OUTPUTS: Z, ZN

Truth Table

INPUTS	OUPUTS	
A	Z	ZN
0	0	1
1	1	0

Note

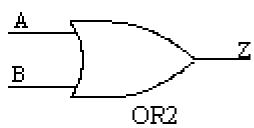
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OR2

2 Input OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B

OUTPUT: Z

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

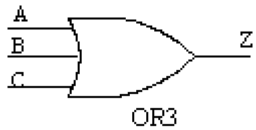
OR3

3 Input OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP

- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C

OUTPUT: Z

Note

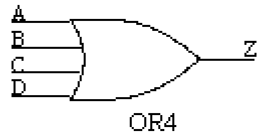
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OR4

4 Input OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D

OUTPUT: Z

Note

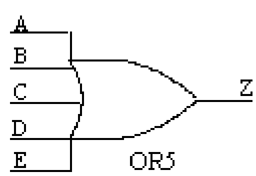
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OR5

5 Input OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D, E

OUTPUT: Z

Note

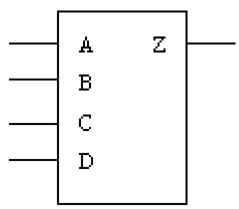
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

ORCALUT4

4-Input Look Up Table

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



ORCALUT4

INPUTS: A, B, C, D

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 16'h0000)

Description

ORCALUT4 defines the programmed state of a LUT4 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT4 programming. The contents of the look up table are addressed by the 4 input pins to access 1 of 16 locations.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

The programming of the ORCALUT4 (that is, the 0 or 1 value of each memory location within the LUT4) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

For example, hex value BF80 produces these 16 memory locations and values:

1011 1111 1000 0000

Memory location 0 (D=0, C=0, B=0, A=0) contains a 0, memory location 2 (D=0, C=0, B=1, A=0) contains a 0. Memory location 15 (D=1, C=1, B=1, A=1) contains a 1, etc.

The ORCALUT4 may encode the Boolean logic for any Boolean expression of 4 input variables. For example, if the required expression was:

Z = (D*C) + (B*!A)

then the INIT value can be derived from the truth table resulting from the expression:

```
D C B A : Z
0 0 0 0 : 0
0 0 0 1 : 0
0 0 1 0 : 1
0 0 1 1 : 0

0 1 0 0 : 0
0 1 0 1 : 0
0 1 1 0 : 1
0 1 1 1 : 0

1 0 0 0 : 0
1 0 0 1 : 0
1 0 1 0 : 1
1 0 1 1 : 0

1 1 0 0 : 1
1 1 0 1 : 1
1 1 1 0 : 1
1 1 1 1 : 1
```

INIT = F444 (16)

Adding INIT to HDL

INIT can be used as an HDL attribute. The following examples demonstrate how to use INIT with the ORCALUT4 primitive in your Verilog or VHDL source. INIT takes binary value in HDL.

Verilog Example

```
// synopsys translate_off
// parameter definition
defparam I1.init = 16'hF444 ;
// synopsys translate_on
```

```
// ORCALUT4 module instantiation
ORCALUT4 I1 (.A(A), .B (B), .C(C), .D(D), .Z(Q[0]))
/* synthesis init = "16'hF444" */;
```

VHDL Example

```
-- component definition
component ORCALUT4
   port (
         A,B,C,D: In std_logic;
               Z: Out std_logic
         );
end component;


-- attribute definition
  attribute INIT: string;
  attribute INIT of I1: label is "1000000000000000";--Z=A*B*C*D
...
-- ORCALUT4 component instantiation
I1 : ORCALUT4
   Port Map ( A=>A, B=>B, C=>C, D=>D, Z=>N_1 );
```

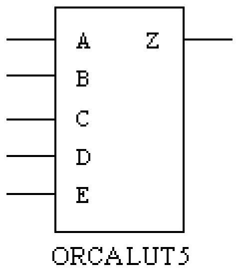
For generic examples of how to use HDL attributes, see “Adding FPGA Attributes to HDL” in online Help.

ORCALUT5

5-Input Look Up Table

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



INPUTS: A, B, C, D, E

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 32'h0000_0000)

Description

ORCALUT5 defines the programmed state of a LUT5 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT5 programming. The contents of the look up table are addressed by the 5 input pins to access 1 of 32 locations.

The programming of the ORCALUT5 (that is, the 0 or 1 value of each memory location within the LUT5) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

For more information on INIT attribute usage, see the ORCALUT4 topic.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

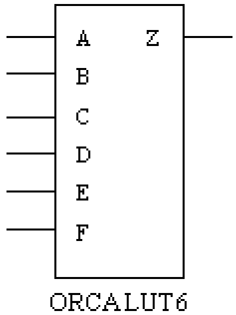
ORCALUT6

6-Input Look Up Table

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeXP
- LatticeXP2
- MachXO

- Platform Manager



INPUTS: A, B, C, D, E, F

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 64'h0000_0000_0000_0000)

Description

ORCALUT6 defines the programmed state of a LUT6 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT6 programming. The contents of the look up table are addressed by the 6 input pins to access 1 of 64 locations.

The programming of the ORCALUT6 (that is, the 0 or 1 value of each memory location within the LUT6) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

For more information on INIT attribute usage, see the ORCALUT4 topic.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

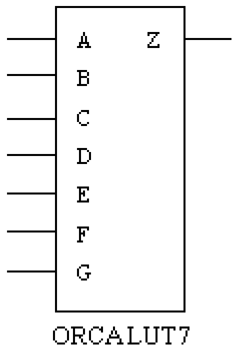
ORCALUT7

7-Input Look Up Table

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeXP

- LatticeXP2
- MachXO
- Platform Manager



INPUTS: A, B, C, D, E, F, G

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 128'h0000_0000_0000_0000_0000_0000_0000_0000)

Description

ORCALUT7 defines the programmed state of a LUT7 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT7 programming. The contents of the look up table are addressed by the 7 input pins to access 1 of 128 locations.

The programming of the ORCALUT7 (that is, the 0 or 1 value of each memory location within the LUT7) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

For more information on INIT attribute usage, see the ORCALUT4 topic.

Note

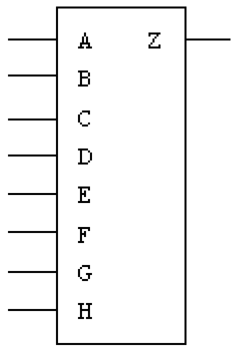
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

ORCALUT8

8-Input Look Up Table

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



ORCALUT8

INPUTS: A, B, C, D, E, F, G, H

OUTPUT: Z

ATTRIBUTES:

INIT: hexadecimal value (default: 256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000)

Description

ORCALUT8 defines the programmed state of a LUT8 primitive of a Slice. While this primitive is typically targeted by logic synthesis tools, it can also be instantiated in HDL source for intimate control over LUT8 programming. The contents of the look up table are addressed by the 8 input pins to access 1 of 256 locations.

The programming of the ORCALUT8 (that is, the 0 or 1 value of each memory location within the LUT8) is determined by the value assigned with INIT. The value is expressed in hexadecimal code. Highest memory locations are in the most significant hex digit, the lowest in the least significant digit.

For more information on INIT attribute usage, see the ORCALUT4 topic.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OSCA

Internal Oscillator

Architectures Supported:

- LatticeSC/M

OUTPUT: OSC

ATTRIBUTES:

DIV: 1 (default), 2, 4, 8, 16, 32, 64, 128

Description

The OSCA is the source of the internal clock for configuration. After configuration this oscillator is disabled by default. If needed, it can be enabled by instantiating the OSCA or the Sysbus (with sys_clk_sel=OSC option). OSCA may be used as a general-purpose clock to drive FPGA logic.

The internal clock frequency can be one of eight values: 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 of the oscillator frequency (about 128 MHz). During start-up, the clock divider is set to 1/128 (about 1 MHz). During initialization, it is set to 1/8 (about 16 MHz). After initialization, if OSCA is enabled, it is set to the user-specified value.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OSCC

Internal Oscillator

Architectures Supported:

- MachXO
- Platform Manager

OUTPUT: OSC

Description

OSCC is a dedicated oscillator in the MachXO and Platform Manager device and the source of the internal clock for configuration. The oscillator frequency range is 18 to 26 MHz. The output of the oscillator can also be routed as an input clock to the clock tree. The oscillator frequency output can be further divided by internal logic (user logic) for lower frequencies, if desired. The oscillator is powered down when not in use. The example below illustrates proper usage for instantiating the OSCC primitive in VHDL.

```
COMPONENT OSCC
   PORT (OSC:OUT std_logic);
END COMPONENT;


begin


OSCInst0: OSCC
   PORT MAP (
             OSC => osc_int
         );
```

Note

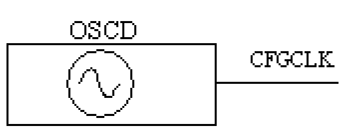
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OSCD

Oscillator for Configuration Clock

Architectures Supported:

- LatticeECP2/M



OUTPUT: CFGCLK

ATTRIBUTES:

NOM_FREQ: 2.5 (default), 4.3, 5.4, 6.9, 8.1, 9.2, 10.0, 13.0, 15.0, 20.0, 26.0, 30.0, 34.0, 41.0, 45.0, 55.0, 60.0, 130.0 (in MHz)

Description

OSCD is the primitive name of the ECP2/M oscillator. The internal oscillator is the source of the internal clock for configuration and is nominally running at 130 MHz. The oscillator is configurable by the user.

During configuration, the internal clock frequency is selected with bits 5-0 in configuration control register 0. The default frequency is 2.5 MHz, where the default values of bits 5-0 are all zeros. IO description and attribute descriptions are shown in the following tables.

OSCD Port Definition

Port Name	I/O	Description
CFGCLK	Output	Oscillator clock output

OSCD Usage with VHDL

```
COMPONENT OSCD
-- synthesis translate_off
   GENERIC(NOM_FREQ: string:= "2.5");
-- synthesis translate_on
   PORT (CFGCLK: OUT std_logic);
END COMPONENT;


   attribute NOM_FREQ : string;
   attribute NOM_FREQ of OSCins0 : label is "2.5";


begin


OSCInst0: OSCD
-- synthesis translate_off
   GENERIC MAP (NOM_FREQ => "2.5")
-- synthesis translate_on
   PORT MAP ( CFGCLK => osc_int);
```

OSCD Usage with Verilog HDL

```
module OSC_TOP(OSC_CLK);
output OSC_CLK;
OSCD OSCinst0 (.CFGCLK(OSC_CLK));
defparam OSCinst0.NOM_FREQ = "2.5";
```

```
endmodule
```

Note

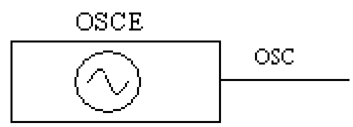
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OSCE

Oscillator for Configuration Clock

Architectures Supported:

- LatticeXP2



OUTPUT: OSC

ATTRIBUTES:

NOM_FREQ: 2.5 (default), 3.1, 4.3, 5.4, 6.9, 8.1, 9.2, 10.0, 13.0, 15.0, 20.0, 26.0, 32.0, 40.0, 54.0, 80.0, 163.0 (in MHz)

Description

OSCE is the primitive name of the XP2 oscillator. The internal oscillator is the source of the internal clock and is configurable by the user.

During configuration, the internal clock frequency is selected with bits 5-0 in configuration control register 0. The default frequency is 2.5 MHz, where the default values of bits 5-0 are all zeros. IO description and attribute descriptions are shown in the following tables.

OSCE Port Definition

Port Name	I/O	Description
OSC	Output	Oscillator clock output

OSCE Usage with VHDL

```
COMPONENT OSCE
-- synthesis translate_off
   GENERIC (NOM_FREQ: string := "2.5");
-- synthesis translate_on
   PORT (OSC :OUT std_logic);
```

```
END COMPONENT;

   attribute NOM_FREQ : string;
   attribute NOM_FREQ of OSCinst0 : label is "2.5";

begin

OSCInst0: OSCE
-- synthesis translate_off
   GENERIC MAP (
                NOM_FREQ => "2.5"
               )
-- synthesis translate_on
   PORT MAP (
             OSC => osc_int
             );
```

Note

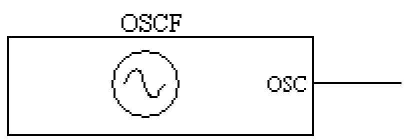
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

OSCF

Oscillator for Configuration Clock

Architectures Supported:

- LatticeECP3



OUTPUT: OSC

ATTRIBUTES:

NOM_FREQ: 2.5 (default), 4.3, 5.4, 6.9, 8.1, 9.2, 10.0, 13.0, 15.0, 20.0, 26.0, 30.0, 34.0, 41.0, 45.0, 55.0, 60.0, 130.0 (in MHz)

Description

OSCF is the primitive name of the LatticeECP3 oscillator. The internal oscillator is the source of the internal clock and is configurable by the user.

During configuration, the internal clock frequency is selected with bits 5-0 in configuration control register 0. The default frequency is 2.5 MHz, where the default values of bits 5-0 are all zeros. IO description and attribute descriptions are shown in the following tables.

OSCF Port Definition

Port Name	I/O	Description
OSC	Output	Oscillator clock output

OSCF Usage with VHDL

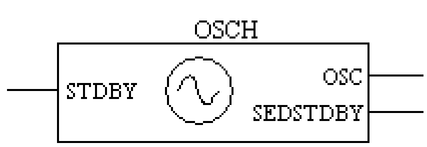
```
COMPONENT OSCF
-- synthesis translate_off
   GENERIC  (NOM_FREQ: string :=  "2.5");
-- synthesis translate_on
   PORT (OSC: OUT std_logic);
END COMPONENT;


   attribute NOM_FREQ : string;
   attribute NOM_FREQ of OSCinst0 : label is "2.5";


begin


OSCInst0: OSCF
-- synthesis translate_off
   GENERIC MAP (
                NOM_FREQ => "2.5"
               )
-- synthesis translate_on
   PORT MAP (
             OSC => osc_int
            );
```

OSCH

Oscillator for MachXO2/Platform Manager 2

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUT: STDBY

OUTPUT: OSC, SEDSTDBY

ATTRIBUTES:

NOM_FREQ: 2.08 (default), 2.15, 2.22, 2.29, 2.38, 2.46, 2.56, 2.66, 2.77, 2.89, 3.02, 3.17, …, 19.0, 20.46, 22.17, 24.18, 26.6, 29.56, 33.25, 38.0, 44.33, 53.2, 66.5, 88.67, 133.0 (in MHz)

Description

OSCH is the primitive name of the MachXO2/Platform Manager 2 oscillator. See the IO port description in the below table.

Port Name	I/O	Description
STDBY	Input	Standby – power down oscillator
OSC	Output	Oscillator clock output
SEDSTDBY	Output	Standby – power down SED clock

There is a fuse setting to enable the STDBY port of the oscillator. The fuse name is MC1_CIB_CTRL_OSC and its default setting is to disable the STDBY port. You must program this fuse when you want to make a connection to the STDBY port of the oscillator. After programing the fuse, a high value on the STDBY port will disable the oscillator and make the output to be driven low. The STDBY port defaults to low, which allows the oscillator to be active. There is a separate signal (OSC_ENA) that controls the oscillator operation during configuration.

OSCH Usage with VHDL

```
COMPONENT OSCH
-- synthesis translate_off
   GENERIC  (NOM_FREQ: string :=  "2.56");
-- synthesis translate_on
   PORT (STDBY    : IN  std_logic;
         OSC      : OUT std_logic;
         SEDSTDBY: OUT std_logic);
END COMPONENT;


   attribute NOM_FREQ : string;
   attribute NOM_FREQ of OSCinst0 : label is "2.56";


begin


OSCInst0: OSCH
-- synthesis translate_off
   GENERIC MAP (
                NOM_FREQ => "2.5"
               )
```

```
-- synthesis translate_on
   PORT MAP (STDBY => stdby,
             OSC    => osc_int,
             SEDSTDBY => stdby_sed
            );
```

For more information, refer to the following technical note on the Lattice web site:

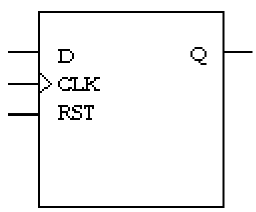
- TN1199 - MachXO2 sysCLOCK PLL Design and Usage Guide

OSRX1A

Output 1-Bit Shift Register

Architectures Supported:

- LatticeSC/M



OSRX1A

INPUTS: D, CLK, RST

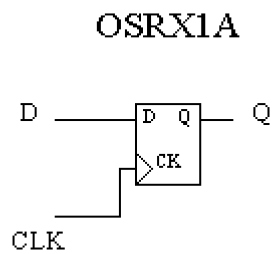
OUTPUT: Q

ATTRIBUTES:

REGSET: "RESET" (default), "SET"

Description

Outputs data through the shift register to the output data. The following symbolic diagram shows the flip-flop structure of this primitive.

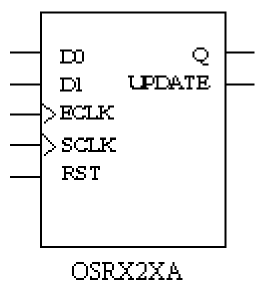


OSRX2A

Output 2-Bit Shift Register

Architectures Supported:

- LatticeSC/M

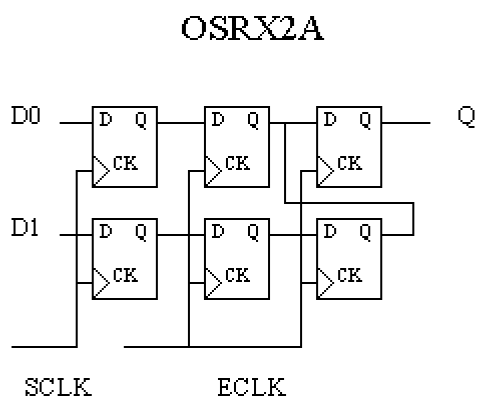


INPUTS: D0, D1, ECLK, SCLK, RST

OUTPUTS: Q, UPDATE

Description

Outputs data through the shift register to the output data. The following symbolic diagram shows the flip-flop structure of this primitive.

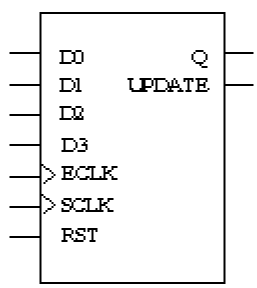


OSRX4A

Output 4-Bit Shift Register

Architectures Supported:

- LatticeSC/M



OSRX4A

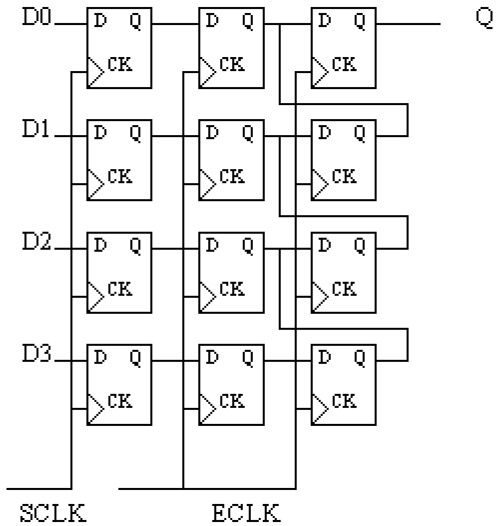
INPUTS: D0, D1, D2, D3, ECLK, SCLK, RST

OUTPUTS: Q, UPDATE

Description

Outputs data through the shift register to the output data. The following symbolic diagram shows the flip-flop structure of this primitive.

OSRX4A



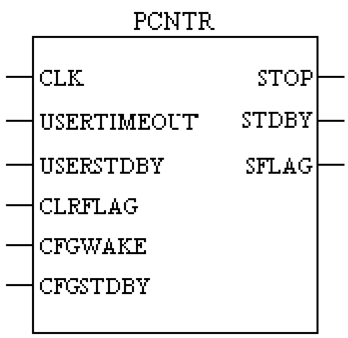
P

PCNTR

Power Controller

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CLK, USERTIMEOUT, USERSTDBY, CLRFLAG, CFGWAKE, CFGSTDBY

OUTPUTS: STOP, STDBY, SFLAG

ATTRIBUTES:

STDBYOPT: "USER_CFG" (default), "USER", "CFG"

TIMEOUT: "BYPASS" (default), "USER", "COUNTER"

WAKEUP: "USER_CFG" (default), "USER", "CFG"

POROFF: "FALSE" (default), "TRUE"

BGOFF: "FALSE" (default), "TRUE"

Description

PCNTR is the MachXO2/Platform Manager 2 power controller primitive.

For more information, refer to the following technical note on the Lattice web site:

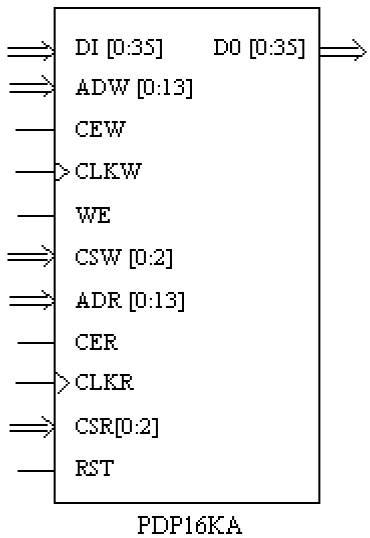
- TN1198 - Power Estimation and Management for MachXO2 Device

PDP16KA

16K Pseudo Dual Port Block RAM

Architectures Supported:

- LatticeSC/M



INPUTS: DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, DI18, DI19, DI20, DI21, DI22, DI23, DI24, DI25, DI26, DI27, DI28, DI29, DI30, DI31, DI32, DI33, DI34, DI35, ADW0, ADW1, ADW2, ADW3, ADW4, ADW5, ADW6, ADW7, ADW8, ADW9, ADW10, ADW11, ADW12, ADW13, CEW, CLKW, WE, CSW0, CSW1, CSW2, ADR0, ADR1, ADR2, ADR3, ADR4, ADR5, ADR6, ADR7, ADR8, ADR9, ADR10, ADR11, ADR12, ADR13, CER, CLKR, CSR0, CSR1, CSR2, RST

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17, DO18, DO19, DO20, DO21, DO22, DO23, DO24, DO25, DO26, DO27, DO28, DO29, DO30, DO31, DO32, DO33, DO34, DO35

ATTRIBUTES:

DATA_WIDTH_W: 1, 2, 4, 9, 18 (default), 36

DATA_WIDTH_R: 1, 2, 4, 9, 18 (default), 36

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_W: any 3-bit binary value (default: 3’b000)

CSDECODE_R: any 3-bit binary value (default: 3’b000)

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_3F: (*Verilog*) 320'hXXX...X (80-bit hexadecimal value)

(*VHDL*) 0xXXX...X (80-bit hexadecimal value)
Default: all zeros

Description

You can refer to the following technical note on the Lattice web site on details of EBR port definition, attribute definition and usage.

- TN1094 - On-Chip Memory Usage Guide for LatticeSC Devices

Note

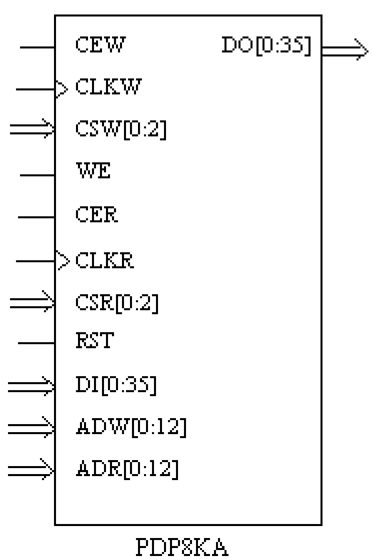
When the write data width (DATA_WIDTH_W) is set to 36, the WE port is invalid, that is, it has no effect on the data output.

PDP8KA

8K Pseudo Dual Port Block RAM

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: CEW, CLKW, CSW0, CSW1, CSW2, WE, CER, CLKR, CSR0, CSR1, CSR2, RST, DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, DI18, DI19, DI20, DI21, DI22, DI23, DI24, DI25, DI26, DI27, DI28, DI29, DI30, DI31, DI32, DI33, DI34, DI35, ADW0, ADW1, ADW2, ADW3, ADW4, ADW5, ADW6, ADW7, ADW8, ADW9,

ADW10, ADW11, ADW12, ADR0, ADR1, ADR2, ADR3, ADR4, ADR5, ADR6, ADR7, ADR8, ADR9, ADR10, ADR11, ADR12

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17, DO18, DO19, DO20, DO21, DO22, DO23, DO24, DO25, DO26, DO27, DO28, DO29, DO30, DO31, DO32, DO33, DO34, DO35

ATTRIBUTES (LatticeXP/EC):

DATA_WIDTH_W: 1, 2, 4, 9, 18, 36 (default)

DATA_WIDTH_R: 1, 2, 4, 9, 18, 36 (default)

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_W: any 3-bit binary value (default: 111)

CSDECODE_R: any 3-bit binary value (default: 111)

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_1F: (*Verilog*) 320'hXXX...X (80-bit hexadecimal value)
(*VHDL*) 0xXXX...X (80-bit hexadecimal value)
Default: all zeros

Description

You can refer to the following technical note on the Lattice web site on details of EBR port definition, attribute definition and usage.

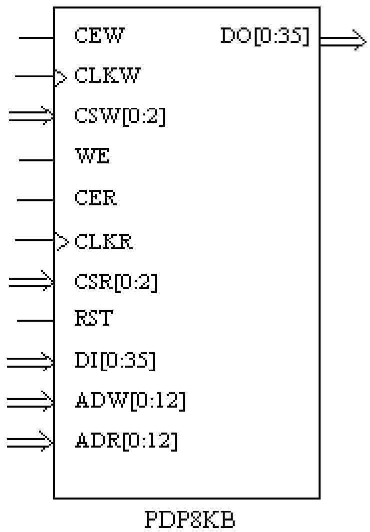
- TN1051 - Memory Usage Guide for LatticeECP/EC and LatticeXP Devices

PDP8KB

8K Pseudo Dual Port Block RAM

Architectures Supported:

- MachXO
- Platform Manager



INPUTS: CEW, CLKW, CSW0, CSW1, CSW2, WE, CER, CLKR, CSR0, CSR1, CSR2, RST, DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, DI18, DI19, DI20, DI21, DI22, DI23, DI24, DI25, DI26, DI27, DI28, DI29, DI30, DI31, DI32, DI33, DI34, DI35, ADW0, ADW1, ADW2, ADW3, ADW4, ADW5, ADW6, ADW7, ADW8, ADW9, ADW10, ADW11, ADW12, ADR0, ADR1, ADR2, ADR3, ADR4, ADR5, ADR6, ADR7, ADR8, ADR9, ADR10, ADR11, ADR12

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17, DO18, DO19, DO20, DO21, DO22, DO23, DO24, DO25, DO26, DO27, DO28, DO29, DO30, DO31, DO32, DO33, DO34, DO35

ATTRIBUTES:

DATA_WIDTH_W: 1, 2, 4, 9, 18 (default), 36

DATA_WIDTH_R: 1, 2, 4, 9, 18 (default), 36

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_W: any 3-bit binary value (default: 3’b000)

CSDECODE_R: any 3-bit binary value (default: 3’b000)

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_1F: (*Verilog*) "320'hXXX...X" (80-bit hex string)
(*VHDL*) "0xXXX...X" (80-bit hex string)
Default: all zeros

Description

You can refer to the following technical note on the Lattice web site on details of EBR port definition, attribute definition and usage.

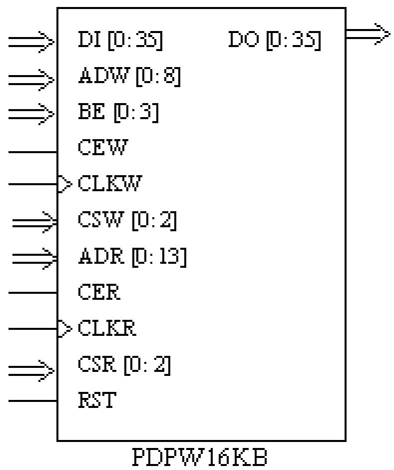
- TN1092 - MachXO Memory Usage Guide

PDPW16KB

Pseudo Dual Port Block RAM

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, DI18, DI19, DI20, DI21, DI22, DI23, DI24, DI25, DI26, DI27, DI28, DI29, DI30, DI31, DI32, DI33, DI34, DI35, ADW0, ADW1, ADW2, ADW3, ADW4, ADW5, ADW6, ADW7, ADW8, BE0, BE1, BE2, BE3, CEW, CLKW, CSW0, CSW1, CSW2, ADR0, ADR1, ADR2, ADR3, ADR4, ADR5, ADR6, ADR7, ADR8, ADR9, ADR10, ADR11, ADR12, ADR13, CER, CLKR, CSR0, CSR1, CSR2, RST

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17, DO18, DO19, DO20, DO21, DO22, DO23, DO24, DO25, DO26, DO27, DO28, DO29, DO30, DO31, DO32, DO33, DO34, DO35

ATTRIBUTES:

DATA_WIDTH_W: 1, 2, 4, 9, 18, 36 (default)

DATA_WIDTH_R: 1, 2, 4, 9, 18, 36 (default)

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_W: any 3-bit binary value (default: 0b000)

CSDECODE_R: any 3-bit binary value (default: 0b000)

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_3F: "0xXXX...X" (80-bit hex string) (default: all zeros)

Description

You can refer to the following technical notes on the Lattice web site on details of EBR port definition, attribute definition and usage.

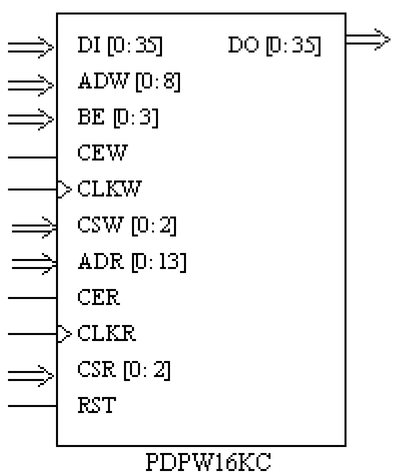
- TN1104 - LatticeECP2/M Memory Usage Guide
- TN1137 - LatticeXP2 Memory Usage Guide

PDPW16KC

Pseudo Dual Port Block RAM

Architectures Supported:

- LatticeECP3



INPUTS: DI35, DI34, DI33, DI32, DI31, DI30, DI29, DI28, DI27, DI26, DI25, DI24, DI23, DI22, DI21, DI20, DI19, DI18, DI17, DI16, DI15, DI14, DI13, DI12, DI11, DI10, DI9, DI8, DI7, DI6, DI5, DI4, DI3, DI2, DI1, DI0, ADW8, ADW7, ADW6, ADW5, ADW4, ADW3, ADW2, ADW1, ADW0, BE3, BE2, BE1, BE0, CEW, CLKW, CSW2, CSW1, CSW0, ADR13, ADR12, ADR11, ADR10,

ADR9, ADR8, ADR7, ADR6, ADR5, ADR4, ADR3, ADR2, ADR1, ADR0, CER, CLKR, CSR2, CSR1, CSR0, RST

OUTPUTS: DO35, DO34, DO33, DO32, DO31, DO30, DO29, DO28, DO27, DO26, DO25, DO24, DO23, DO22, DO21, DO20, DO19, DO18, DO17, DO16, DO15, DO14, DO13, DO12, DO11, DO10, DO9, DO8, DO7, DO6, DO5, DO4, DO3, DO2, DO1, DO0

ATTRIBUTES:

DATA_WIDTH_W: 1, 2, 4, 9, 18, 36 (default)

DATA_WIDTH_R: 1, 2, 4, 9, 18 (default), 36

REGMODE: "NOREG" (default), "OUTREG"

CSDECODE_W: any 3-bit binary value (default: all zeros)

CSDECODE_R: any 3-bit binary value (default: all zeros)

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_3F: "0xXXX...X" (80-bit hex string) (default: all zeros)

Description

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

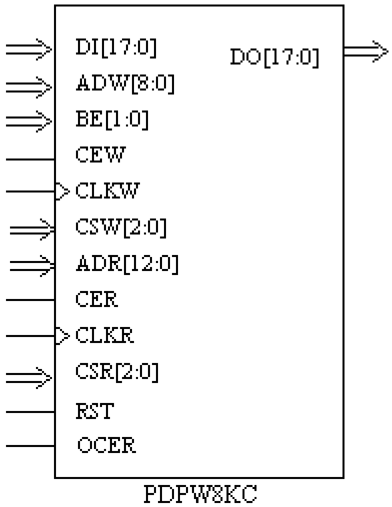
- TN1179 - LatticeECP3 Memory Usage Guide

PDPW8KC

Pseudo Dual Port Block RAM

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: DI17, DI16, DI15, DI14, DI13, DI12, DI11, DI10, DI9, DI8, DI7, DI6, DI5, DI4, DI3, DI2, DI1, DI0, ADW8, ADW7, ADW6, ADW5, ADW4, ADW3, ADW2, ADW1, ADW0, BE1, BE0, CEW, CLKW, CSW2, CSW1, CSW0, ADR12, ADR11, ADR10, ADR9, ADR8, ADR7, ADR6, ADR5, ADR4, ADR3, ADR2, ADR1, ADR0, CER, CLKR, CSR2, CSR1, CSR0, RST, OCER

OUTPUTS: DO17, DO16, DO15, DO14, DO13, DO12, DO11, DO10, DO9, DO8, DO7, DO6 DO5, DO4, DO3, DO2, DO1, DO0

ATTRIBUTES:

DATA_WIDTH_W: 18 (default)

DATA_WIDTH_R: 1, 2, 4, 9, 18 (default: 9)

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE_W: any 3-bit binary value (default: all zeros)

CSDECODE_R: any 3-bit binary value (default: all zeros)

GSR: "ENABLED" (default), "DISABLED"

RESETMODE: "SYNC" (default), "ASYNC"

ASYNC_RESET_RELEASE: "SYNC" (default), "ASYNC"

INIT_DATA: "STATIC" (default), "DYNAMIC"

INITVAL_00 to INITVAL_1F: (*Verilog*) "320'hXXX...X" (80-bit hex string)
(*VHDL*) "0xXXX...X" (80-bit hex string)
Default: all zeros

Description

The following table describes the I/O ports of the PDPW8KC primitive.

Port Name	I/O	Definition
CLKR	I	Read clock
CLKW	I	Write clock
RST	I	Reset
CSW[1:0]	I	Chip select write
CSR[1:0]	I	Chip select read
CER	I	Read clock enable
CEW	I	Write clock enable
OCER	I	Read output clock enable
BE[1:0]	I	Byte enable
DI[17:0]	I	Write data (up to 18)
ADW[8:0]	I	Write address (up to 9)
ADR[12:0]	I	Read address (up to 13)
DO[17:0]	O	Read data (up to 18)

You can refer to the following technical note on the Lattice web site on details of EBR port definition, attribute definition and usage.

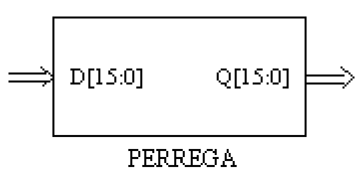
- TN1201 - Memory Usage Guide for MachXO2 Devices

PERREGA

Persistent User Register

Architectures Supported:

- LatticeECP3



INPUTS: D15, D14, D13, D12, D11, D10, D9, D8, D7, D6, D5, D4, D3, D2, D1, D0

OUTPUTS: Q15, Q14, Q13, Q12, Q11, Q10, Q9, Q8, Q7, Q6, Q5, Q4, Q3, Q2, Q1, Q0

Description

The PERREGA primitive enables you to use 16-bit registers to store information when the device goes into the configuration mode. The data on these registers will stay intact during the reconfiguration and be available for use after then. For example, you can use these registers to capture the last pattern that causes an error to reboot from the golden source when using SED.

These latches are available through the internal CIB. The below table describes the I/O ports of the PERREGA primitive.

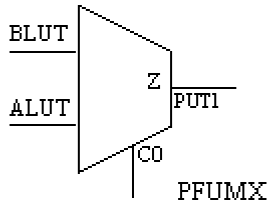
Signal	Type	Description	Function
D[15:0]	CIB Input	Parallel Data In	Provides parallel data from the FPGA fabric for the latch.
Q[15:0]	CIB Output	Parallel Data Out	Provides parallel data to the FPGA fabric for the latch.
PROGRAMN	Control Signal	High to Low Edge on the PROGRAMN pin	Latch in the data from D[15:0].
DONE	Internal Signal	Done signal	Prohibits multiple latching due to transitional glitches on the LE CIB or PROGRAMN pin.

PFUMX

2-Input Mux within the PFU, C0 Used for Selection with Positive Select

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: ALUT, BLUT, C0

OUTPUT: Z

Truth Table

INPUTS			**OUTPUTS**
BLUT	ALUT	C0	Z
X	1	1	1
X	0	1	0
1	X	0	1
0	X	0	0

X = Don't care

Note

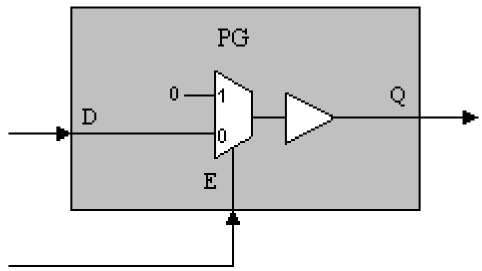
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

PG

Power Guard

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: D, E

OUTPUT: Q

Description

In the power guard mode, the receiver input is disconnected from the pad and held low with a weak pull down. The pad can be toggling and no receiver power will be used. Power guard is enabled on a bank by bank basis. Each bank has a CIB input signal to enable power guard. In addition, each PIO has individually programmable bit control to disable or enable the power guard capability.

PG is the power guard primitive for MachXO2/Platform Manager 2 devices. You can generate a primitive for a group of I/O pins and then connect it in the design. The PG primitive D port must be connected to the IO pin. You are required to connect the enable input of the primitive to the signal that is used to activate the power guard function. Only one enable signal can be used for all I/Os within an individual I/O bank using power guard. An individual I/O pin is only allowed to connect to one PG primitive module but there could be more than one PG module active in a single I/O bank. Once an I/O signal is connected to the power guard primitive, the software sets the individually programmable bit for that PIO to enable power guard. The default setting for this bit is to disable the power guard function for a PIO.

The following table describes the IO ports of the PG primitive.

Port Name	I/O	Description
D	I	This is the signal coming from an input or I/O pad. This pin cannot have any fanout—only connects between pad and PG.
E	I	This is the “ENABLE” input that is tied to BIE (Block Input Enable). When E=0, Q is connected to D. When E=1, Q is isolated from D.
Q	O	This is the output of the power guard that drives towards the core. When E=0, the output Q is driven by the input D.

For more information, refer to the following technical note on the Lattice web site:

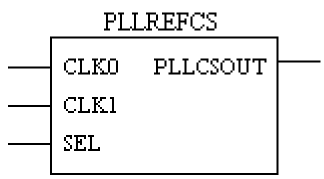
- TN1198 - Power Estimation and Management for MachXO2 Device

PLLREFCS

PLL Dynamic Reference Clock Switching

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CLK0, CLK1, SEL

OUTPUT: PLLCSOUT

Description

The PLLREFCS primitive is used to support the dynamic reference clock switching using the SEL signal. This primitive does not require a wrapper.

You only need to instantiate the PLLREFCS primitive if you want to select between two different reference clocks in your application. If you only use one reference clock for the PLL, the software will automatically make the correct connections.

The following table describes the I/O ports of the PLLREFCS primitive.

Port Name	I/O	Description
CLK0, CLK1	I	There are two clock input MUXES (8 inputs each) which are labeled as REFCLK1 and REFCLK2. CLK0 is the output of REFCLK1. CLK1 is the output of REFCLK2.
SEL	I	Selects which signal goes to the input clock MUX.
PLLCSOUT	O	

For more information, refer to the following technical note on the Lattice web site:

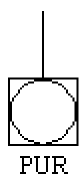
- TN1199 - MachXO2 sysCLOCK PLL Design and Usage Guide

PUR

Power Up Set/Reset

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUT: PUR

ATTRIBUTES:

RST_PULSE: integer (default: 1)

Description

PUR is used to reset or set all register elements in your design upon device configuration/startup. The PUR component can be connected to a net from an input buffer or an internally generated net. It is active LOW and when pulsed will set or reset all register bits to the same state as the local set or reset functionality.

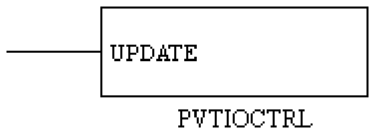
It is not necessary to connect signals for PUR to any register elements explicitly. The function will be implicitly connected globally.

PVTIOCTRL

PVT Monitor Circuit Controller

Architectures Supported:

- LatticeSC/M



INPUT: UPDATE

Description

The PVTIOCTRL primitive is used to generate a signal to control the PVT (Process, Voltage, and Temperature) monitor circuit. When UPDATE is 1, the PVT monitor circuit output will be updated. When UPDATE is 0, the last value of the PVT monitor circuit output is latched in.

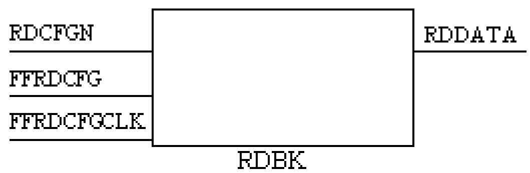
R

RDBK

Readback Controller

Architectures Supported:

- LatticeSC/M



INPUTS: RDCFGN, FFRDCFG, FFRDCFGCLK

OUTPUT: RDDATA

Description

RDBK is used to read back the configuration data and optionally the state of the PFU outputs. RDBK can be done while the FPGA is in normal system operation. To use RDBK, select options in the bit stream generator within the place and route tool.

- You can choose the option to prohibit readback, allow a single readback, or allow unrestricted readback. For more information on RDBK, refer to applicable technical notes or contact technical support.

RDCFGN: A high-to-low transition on this input initiates a readback operation. This pin must remain low during the readback operation.

RDDATA: Readback data is available at this output, which in turn is connected to the same pad as that used by TDO for boundary scan.

Note

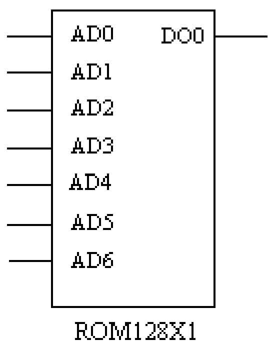
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

ROM128X1

128 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



INPUTS: AD0, AD1, AD2, AD3, AD4, AD5, AD6

OUTPUT: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 128'hXXX...X (32-bit hex string for LatticeECP2 and LatticeXP2; 32-bit hex value for other devices)
(*VHDL*) 0xXXX...X (32-bit hex string for LatticeECP2 and LatticeXP2; 32-bit hex value for other devices)
Default: all zeros

Description

The ROM128X1 symbol represents a 128 word by1-bit read-only memory. This ROM can be used to implement a ORCALUT7 in a design. The read operation is asynchronous and is always active. The memory is always being read.

The INITVAL=<*value*> attribute is used to initialize the ROM. The <*value*> should consist of 128 one-bit binary or 32 hexadecimal data written into the

ROM from the highest address to the lowest address. For example, if the following attribute is specified:

INITVAL=0x0123456789ABCDEF0123456789ABCDEF (in hex)

or

INITVAL=0000000100100011010001010110011110001001101010111100110
111101111000000010010001101000101011001111000100110101011110011
0111101111 (in binary)

it implies that the above data is loaded sequentially from location 127 to 0 (where location 127 would contain value 0 and location 0 value 1).

Truth Table

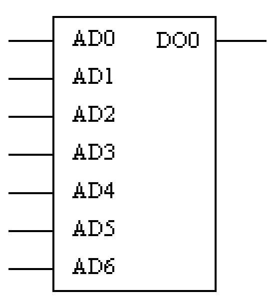
INPUTS	OUTPUTS	OPERATION
AD[5:0]	DO0	
AD[5:0]	MEM[AD[5:0]]	Read MEM[AD[5:0]]

ROM128X1A

128 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



ROM128X1A

INPUTS: AD6, AD5, AD4, AD3, AD2, AD1, AD0

OUTPUT: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 128'hXXX...X (32-bit hexadecimal value)
(*VHDL*) 0xXXX...X (32-bit hexadecimal value)
Default: all zeros

Description

PFU based distributed 128 Word by 1 Bit ROM primitive. See Memory Primitives Overview for individual port description.

You can refer to the following technical notes on the Lattice web site for port definition, attribute definition and usage.

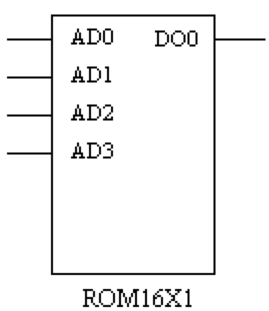
- TN1201 - Memory Usage Guide for MachXO2 Devices
- TN1179 - LatticeECP3 Memory Usage Guide

ROM16X1

16 Word by 1 Bit Read-Only Memory

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



INPUTS: AD0, AD1, AD2, AD3

OUTPUT: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 16'hXXXX (4-bit hex string for LatticeECP2 and LatticeXP2; 4-bit hex value for other devices)
(*VHDL*) 0xXXXX (4-bit hex string for LatticeECP2 and LatticeXP2; 4-bit hex value for other devices)
Default: all zeros

Description

The ROM16X1 symbol represents a 16 word by 1 bit read-only memory. This ROM can be used to implement a ORCALUT4 in a design. The read operation is asynchronous and is always active. The memory is always being read.

The INITVAL=<*value*> attribute is used to initialize the ROM. The <*value*> should consist of 16 one-bit binary or 4 hexadecimal data written into the ROM from the highest address to the lowest address. For example, if the following attribute is specified:

INITVAL=0xF30A (in hex)

or

INITVAL=1111001100001010 (in binary)

it implies that the above data is loaded sequentially from location 15 to 0 (where location 15 would contain value 1 and location 0 value 0).

Truth Table

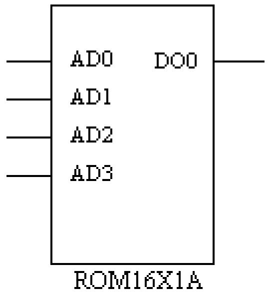
INPUTS	OUTPUTS	OPERATION
AD[3:0]	DO0	
AD[3:0]	MEM[AD[3:0]]	Read MEM[AD[3:0]]

ROM16X1A

16 Word by 1 Bit Read-Only Memory

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: AD3, AD2, AD1, AD0

OUTPUT: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 16'hXXXX (4-bit hexadecimal value)
(*VHDL*) 0xXXXX (4-bit hexadecimal value)
Default: all zeros

Description

PFU based distributed 16 Word by 1 Bit ROM primitive. See Memory Primitives Overview for individual port description.

You can refer to the following technical notes on the Lattice web site for port definition, attribute definition, and usage.

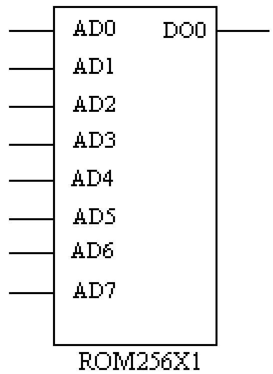
- TN1201 - Memory Usage Guide for MachXO2 Devices
- TN1179 - LatticeECP3 Memory Usage Guide

ROM256X1

256 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



INPUTS: AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7

OUTPUT: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 256'hXXX...X (64-bit hex string for LatticeECP2 and LatticeXP2; 64-bit hex value for other devices)
(*VHDL*) 0xXXX...X (64-bit hex string for LatticeECP2 and LatticeXP2; 64-bit hex value for other devices)
Default: all zeros

Description

The ROM256X1 symbol represents a 256 word by1-bit read-only memory. This ROM can be used to implement a ORCALUT8 in a design. The read operation is asynchronous and is always active. The memory is always being read.

The INITVAL=<*value*> attribute is used to initialize the ROM. The <*value*> should consist of 256 one-bit binary or 64 hexadecimal data written into the ROM from the highest address to the lowest address. For example, if the following attribute is specified:

INITVAL=0x0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
0123456789ABCDEF (in hex)

or

INITVAL=0000000100100011010001010110011110001001101010111100110
1111011110000000100100011010001010110011110001001101010111100 11
0111101111000000001001000110100010101100111100010011010101111001
10111101111000000001001000110100010101100111100010011010101111 00
110111101111 (in binary)

it implies that the above data is loaded sequentially from location 255 to 0 (where location 255 would contain value 0 and location 0 value 1).

Truth Table

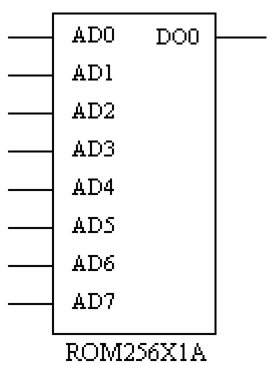
INPUTS	OUTPUTS	OPERATION
AD[5:0]	DO0	
AD[5:0]	MEM[AD[5:0]]	Read MEM[AD[5:0]]

ROM256X1A

256 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: AD7, AD6, AD5, AD4, AD3, AD2, AD1, AD0

OUTPUTS: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 256'hXXX...X (64-bit hexadecimal value)
(*VHDL*) 0xXXX...X (64-bit hexadecimal value)
Default: all zeros

Description

PFU based distributed 256 Word by 1 Bit ROM primitive. See Memory Primitives Overview for individual port description.

You can refer to the following technical notes on the Lattice web site for port definition, attribute definition, and usage.

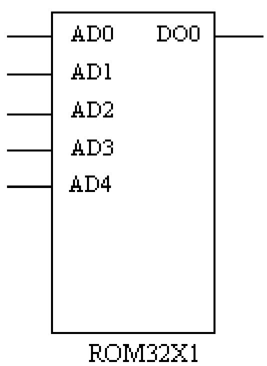
- TN1201 - Memory Usage Guide for MachXO2 Devices
- TN1179 - LatticeECP3 Memory Usage Guide

ROM32X1

32 Word by 1 Bit Read-Only Memory

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



INPUTS: AD0, AD1, AD2, AD3, AD4

OUTPUT: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 32'hXXXXXXXX (8-bit hex string for LatticeECP2 and LatticeXP2; 8-bit hex value for other devices)
(*VHDL*) 0xXXXXXXXX (8-bit hex string for LatticeECP2 and LatticeXP2; 8-bit hex value for other devices)
Default: all zeros

Description

The ROM32X1 symbol represents a 32 word by 1 bit read-only memory. This ROM can be used to implement a ORCALUT5 in a design. The read operation is asynchronous and is always active. The memory is always being read.

The INITVAL=<*value*> attribute is used to initialize the ROM. The <*value*> should consist of 32 one-bit binary or 8 hexadecimal data written into the ROM from the highest address to the lowest address. For example, if the following attribute is specified:

INITVAL=0xF30A1234 (in hex)

or

INITVAL=11110011000010100001001000110100 (in binary)

it implies that the above data is loaded sequentially from location 31 to 0 (where location 31 would contain value 1 and location 0 value 0).

Truth Table

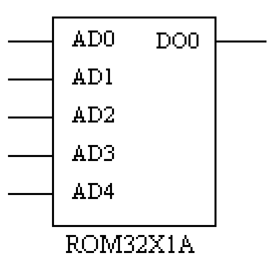
INPUTS	OUTPUTS	OPERATION
AD[4:0]	DO0	
AD[4:0]	MEM[AD[4:0]]	Read MEM[AD[4:0]]

ROM32X1A

32 Word by 1 Bit Read-Only Memory

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: AD4, AD3, AD2, AD1, AD0

OUTPUT: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 32'hXXXXXXXX (8-bit hexadecimal value)
(*VHDL*) 0xXXXXXXXX (8-bit hexadecimal value)
Default: all zeros

Description

PFU based distributed 32 Word by 1 Bit ROM primitive. See Memory Primitives Overview for individual port description.

You can refer to the following technical notes on the Lattice web site for port definition, attribute definition, and usage.

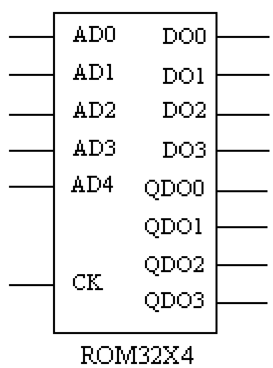
- TN1201 - Memory Usage Guide for MachXO2 Devices
- TN1179 - LatticeECP3 Memory Usage Guide

ROM32X4

32 Word by 4 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Architectures Supported:

- LatticeSC/M



INPUTS: AD0, AD1, AD2, AD3, AD4, CK

OUTPUTS: DO0, DO1, DO2, DO3, QDO0, QDO1, QDO2, QDO3

ATTRIBUTES:

INITVAL: (*Verilog*) 128'hXXX...X (32-bit hexadecimal value)
(*VHDL*) 0xXXX...X (32-bit hexadecimal value)
Default: all zeros

Description

The ROM32X4 symbol represents a 32 word by 4 bit read-only memory. The read operation is asynchronous and is always active. The memory is always being read.

This ROM also has registered data outputs (QDO[0:3]), which are registered on the rising edge of the clock.

The INITVAL=<*value*> attribute is used to initialize the ROM. The <*value*> should consist of 32 four-bit binary or 32 hexadecimal data written into the ROM from the highest address to the lowest address. For example, if the following attribute is specified:

INITVAL=0x0123456789ABCDEF0123456789ABCDEF (in hex)

or

INITVAL=0000000100100011010001010110011110001001101010111100110
1111011110000000010010001101000101011001111000100110101011110011
0111101111 (in binary)

it implies that the above data is loaded sequentially from location 127 to 0 (where location 127 would contain value 0 and location 0 value 1).

Truth Table

INPUTS		OUTPUTS		OPERATION
AD[4:0]	CK	DO[3:0]	QDO[3:0]	
AD[4:0]	X	MEM[AD[4:0]]	QDO[3:0]	Read MEM[AD[4:0]]
AD[4:0]	↑	MEM[AD[4:0]]	MEM[AD[4:0]]	MEM[AD[4:0]] Register data outputs

X = Don’t care

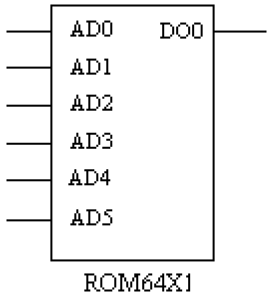
When GSR=0, QDO[3:0]=0

ROM64X1

64 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- Platform Manager



INPUTS: AD0, AD1, AD2, AD3, AD4, AD5

OUTPUT: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 64'hXXXXXXXXXXXXXXXX (16-bit hex string for LatticeECP2 and LatticeXP2; 16-bit hex value for other devices)
(*VHDL*) 0xXXXXXXXXXXXXXXXX (16-bit hex string for LatticeECP2 and LatticeXP2; 16-bit hex value for other devices)
Default: all zeros

Description

The ROM64X1 symbol represents a 64 word by 1-bit read-only memory. This ROM can be used to implement a ORCALUT6 in a design. The read operation is asynchronous and is always active. The memory is always being read.

The INITVAL=<*value*> attribute is used to initialize the ROM. The <*value*> should consist of 64 one-bit binary or 16 hexadecimal data written into the ROM from the highest address to the lowest address. For example, if the following attribute is specified:

INITVAL=0x0123456789ABCDEF (in hex)

or

INITVAL=0000000100100011010001010110011110001001101010111100110111101111 (in binary)

it implies that the above data is loaded sequentially from location 63 to 0 (where location 63 would contain value 0 and location 0 value 1).

Truth Table

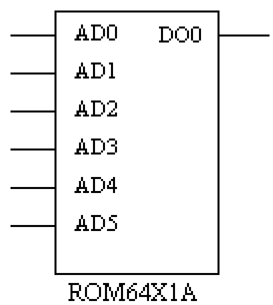
INPUTS	OUTPUTS	OPERATION
AD[5:0]	DO0	
AD[5:0]	MEM[AD[5:0]]	Read MEM[AD[5:0]]

ROM64X1A

64 Word by 1 Bit Positive Edge Triggered Read-Only Memory with Registered Data Outputs

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: AD5, AD4, AD3, AD2, AD1, AD0

OUTPUT: DO0

ATTRIBUTES:

INITVAL: (*Verilog*) 64'hXXXXXXXXXXXXXXXX (16-bit hexadecimal value)
(*VHDL*) 0xXXXXXXXXXXXXXXXX (16-bit hexadecimal value)
Default: all zeros

Description

PFU based distributed 64 Word by 1 Bit ROM primitive. See Memory Primitives Overview for individual port description.

You can also refer to the following technical notes on the Lattice web site for port definition, attribute definition, and usage.

- TN1201 - Memory Usage Guide for MachXO2 Devices

- TN1179 - LatticeECP3 Memory Usage Guide

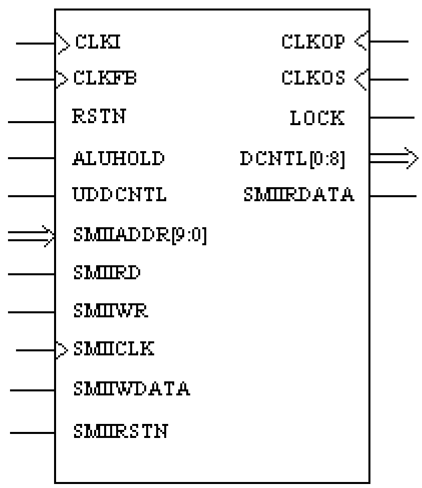
S

SDCDLLA

Single Delay Cell DLL

Architectures Supported:

- LatticeSC/M



SDCDLLA

INPUTS: CLKI, CLKFB, RSTN, ALUHOLD, UDDCNTL, SMIADDR9, SMIADDR8, SMIADDR7, SMIADDR6, SMIADDR5, SMIADDR4, SMIADDR3, SMIADDR2, SMIADDR1, SMIADDR0, SMIRD, SMIWR, SMICLK, SMIWDATA, SMIRSTN

OUTPUTS: CLKOP, CLKOS, LOCK, DCNTL0, DCNTL1, DCNTL2, DCNTL3, DCNTL4, DCNTL5, DCNTL6, DCNTL7, DCNTL8, SMIRDATA

ATTRIBUTES:

CLKOS_FPHASE: 0 (default), 11, 22, 45

CLKI_DIV: 1 (default), 2, 4

CLKOS_DIV: 1 (default), 2, 4

GSR: "DISABLED" (default), "ENABLED"

CLKOS_FDEL_ADJ: "DISABLED" (default), "ENABLED’

ALU_LOCK_CNT: integers 3~15 (default: 3)

ALU_UNLOCK_CNT: integers 3~15 (default: 3)

GLITCH_TOLERANCE: integers 0~7 (default: 0)

DCNTL_ADJVAL: integers -127~127 (default: 0)

ALU_INIT_CNTVAL: 0, 4, 8, 12, 16, 32, 48, 64, 72 (default: 0)

LOCK_DELAY: integers 0~1000 (in ns) (default: 100)

SMI_OFFSET: 0x400~0x7FF (default: 12'h410)

MODULE_TYPE: "SDCDLLA"

IP_TYPE: "SDCDLLA"

Description

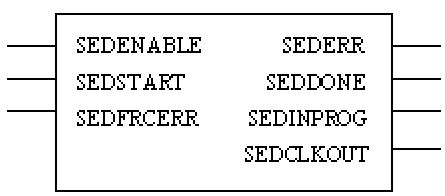
The single delay cell corrects for clock injection and enables the 9-bit ALU output. The primitive features a single clock output, lock achieved starting from minimum delay, output control bits, and allows +/- delay on output control bits. Its requirements are a maximum frequency of 700 MHz and a minimum frequency of 300 MHz.

SEDAA

SED (Soft Error Detect) Basic

Architectures Supported:

- LatticeECP2/M



SEDAA

INPUTS: SEDENABLE, SEDSTART, SEDFRCERR

OUTPUTS: SEDERR, SEDDONE, SEDINPROG, SEDCLKOUT

ATTRIBUTES:

OSC_DIV: 1 (default), 2, 4, 8, 16, 32, 64, 128, 256

CHECKALWAYS: "DISABLED" (default), "ENABLED"

MCCLK_FREQ: "2.5" (default), "4.3", "5.4", "6.9", "8.1", "9.2", "10.0", "13", "15", "20", "26", "30", "34", "41", "45", "55", "60", "130"

DEV_DENSITY: (ECP2) "35K" (default), "6K", "12K", "20K", "50K", "70K"

DEV_DENSITY: (ECP2M) "M35K" (default), "M20K", "M50K", "M70K", "M100K"

ENCRYPTION: "OFF" (default), "ON"

Description

SEDAA is the basic SED primitive for LatticeECP2 devices. Soft errors occur when high-energy charged particles alter the stored charge in a memory cell in an electronic circuit. There are customer selectable software features in Lattice Diamond to support Soft Error Detect (SED) IP features. This is only applicable to devices that support SED in their architecture.

This primitive should be instantiated in the user's source code in VHDL or Verilog.

See the following table for port definition.

Port Name	I/O	Description
SEDSTART	I	Error detection start (sampled at positive clock edge).
SEDENABLE	I	SED enable (a Low clears the SED).
SEDFRCERR	I	Force an SED error flag (e.g. for testing), active high.
SEDINPROG	O	SED cycle is in progress, asserts high.
SEDDONE	O	SED cycle is complete, asserts high.
SEDERR	O	SED error flag, asserts high.
SEDCLKOUT	O	Optional clock output.

You can refer to the following technical note on the Lattice web site for more information and usage.

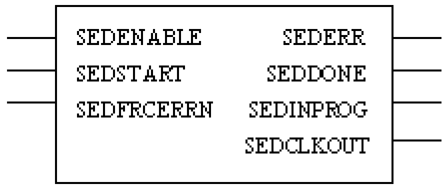
- TN1113 - LatticeECP2/M Soft Error Detection (SED) Usage Guide

SEDBA

Basic SED (Soft Error Detect)

Architectures Supported:

- LatticeXP2



SEDBA

INPUTS: SEDENABLE, SEDSTART, SEDFRCERRN

OUTPUTS: SEDERR, SEDDONE, SEDINPROG, SEDCLKOUT

ATTRIBUTES:

OSC_DIV: 1 (default), 2, 4, 8, 16, 32, 64, 128, 256

CHECKALWAYS: "DISABLED" (default), "ENABLED"

MCCLK_FREQ: "3.1" (default), "2.5", "4.3", "5.4", "6.9", "8.1", "9.2", "10.0", "13", "15", "26", "32", "40", "54"

DEV_DENSITY: "17K" (default), "5K", "8K", "30K", "40K"

BOOT_OPTION: "INTERNAL" (default), "EXTERNAL"

Description

SEDBA is the basic soft error detect (SED) primitive for LatticeXP2 devices. Soft errors occur when high-energy charged particles alter the stored charge in a memory cell in an electronic circuit. There are customer selectable software features in Lattice Diamond to support SED IP features. This is only applicable to devices that support SED in their architecture. LatticeXP2 devices have built-in SED circuitry to detect soft errors.

The SED primitive should be instantiated in user's VHDL or Verilog source code. There are two types of LatticeXP2 modules: Basic SED (SEDBA) and One Shot SED (SEDBB).

See the following table for port description

Port Name	I/O	Description
SEDSTART	I	Error detection start (sampled at positive clock edge).
SEDENABLE	I	SED enable (a Low clears the SED).
SEDFRCERRN	I	Force an SED error flag (e.g. for testing), active low.
SEDINPROG	O	SED cycle is in progress, asserts high.
SEDDONE	O	SED cycle is complete, asserts high.

Port Name	I/O	Description
SEDERR	O	SED error flag, asserts high.
SEDCLKOUT	O	Optional clock output.

You can refer to the following technical note on the Lattice web site for more information and usage.

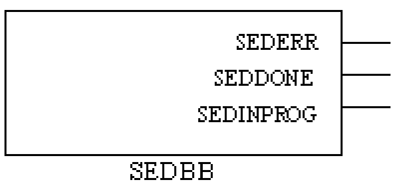
- TN1130 - LatticeXP2 Soft Error Detection (SED) Usage Guide

SEDBB

One Shot SED (Soft Error Detect)

Architectures Supported:

- LatticeXP2



OUTPUTS: SEDERR, SEDDONE, SEDINPROG

Description

SEDBB is the one shot soft error detect (SED) primitive for LatticeXP2 devices. Soft errors occur when high-energy charged particles alter the stored charge in a memory cell in an electronic circuit. There are customer selectable software features in Lattice Diamond to support SED IP features. This is only applicable to devices that support SED in their architecture. LatticeXP2 devices have built-in SED circuitry to detect soft errors.

The SED primitive should be instantiated in user's VHDL or Verilog source code. There are two types of LatticeXP2 modules: Basic SED (SEDBA) and One Shot SED (SEDBB).

See the following table for port description.

Port Name	I/O	Description
SEDINPROG	O	SED cycle is in progress, asserts high.
SEDDONE	O	SED cycle is complete, asserts high.
SEDERR	O	SED error flag, asserts high.

You can refer to the following technical note on the Lattice web site for more information and usage.

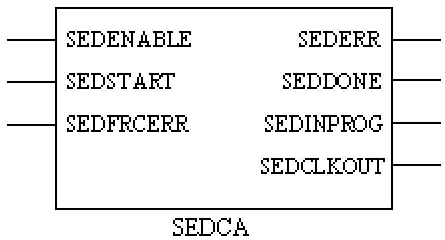
- TN1130 - LatticeXP2 Soft Error Detection (SED) Usage Guide

SEDCA

Basic SED (Soft Error Detect)

Architectures Supported:

- LatticeECP3



INPUTS: SEDENABLE, SEDSTART, SEDFRCERR

OUTPUTS: SEDERR, SEDDONE, SEDINPROG, SEDCLKOUT

ATTRIBUTES:

OSC_DIV: 1 (default), 2, 4, 8, 16, 32, 64, 128, 256

CHECKALWAYS: "DISABLED" (default), "ENABLED"

MCCLK_FREQ: "2.5" (default), "4.3", "5.4", "6.9", "8.1", "9.2", "10.0", "13", "15", "20", "26", "30", "34", "41", "45", "55", "60", "130"

DEV_DENSITY: "95K" (default)

Description

SEDCA is the basic Soft Error Detect primitive. Basic SED runs on all bits only. It checks the CRC for all the bitstreams that include both “Care” and “Don’t Care” bits.

See the following table for port definition.

Port Name	I/O	Description
SEDENABLE	I	SED enable (a Low clears the SED).
SEDSTART	I	Error detection start (sampled at positive clock edge).

Port Name	I/O	Description
SEDFRCERR	I	Force an SED error flag (e.g. for testing), active high.
SEDINPROG	O	SED cycle is in progress, asserts High.
SEDDONE	O	SED cycle is complete, asserts High.
SEDERR	O	SED error flag, asserts High.
SEDCLKOUT	O	Optional clock output.

You can refer to the following technical note on the Lattice web site for more information and usage.

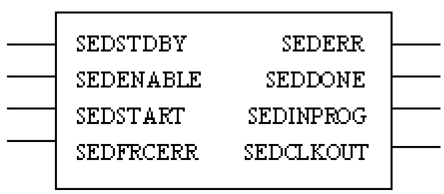
- TN1184 - LatticeECP3 Soft Error Detection (SED) Usage Guide

SEDFA

Soft Error Detect in Basic Mode

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



SEDFA

INPUTS: SEDSTDBY, SEDENABLE, SEDSTART, SEDFRCERR

OUTPUTS: SEDERR, SEDDONE, SEDINPROG, SEDCLKOUT

ATTRIBUTES:

SED_CLK_FREQ: "2.08", "2.15", "2.22", "2.29", "2.38", "2.46", "2.56", "2.66", "2.77", "2.89", "3.02", "3.17", "3.33", "3.5" (default), "3.69", "3.91", "4.16", "4.29", "4.43", "4.59", "4.75", "4.93", "5.12", "5.32", "5.54", "5.78", "6.05", "6.33", "6.65", "7", "7.39", "7.82", "8.31", "8.58", "8.87", "9.17", "9.5", "9.85", "10.23", "10.64", "11.08", "11.57", "12.09", "12.67", "13.3", "14", "14.78", "15.65", "16.63", "17.73", "19", "20.46", "22.17", "24.18", "26.6", "29.56", "33.25", "38", "44.33", "53.2", "66.5", "88.67", "133"

CHECKALWAYS: "DISABLED" (default), "ENABLED"

DEV_DENSITY: "256L", "640L", "1200L" (default), "2000L", "4000L", "7000L", "10000L"

Description

The Soft-Error Detect (SED) circuitry provides a method for the device to check its configuration data for soft-errors. SEDFA is used for SED basic mode.

See the following table for port description.

Port Name	I/O	Description
SEDSTDBY	I	SED standby.
SEDENABLE	I	SED enable (a Low clears the SED).
SEDSTART	I	Error detection start (sampled at positive clock edge).
SEDFRCERR	I	Force an SED error flag (e.g. for testing), active high.
SEDINPROG	O	SED cycle is in progress, asserts High.
SEDDONE	O	SED cycle is complete, asserts High.
SEDERR	O	SED error flag, asserts High.
SEDCLKOUT	O	Optional clock output.

You can refer to the following technical note on the Lattice web site for more information and usage.

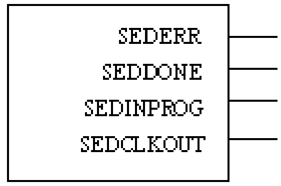
- TN1206 - MachXO2 Soft Error Detection (SED) Usage Guide

SEDFB

Soft Error Detect in One Shot Mode

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



SEDFB

OUTPUTS: SEDERR, SEDDONE, SEDINPROG, SEDCLKOUT

Description

The Soft-Error Detect (SED) circuitry provides a method for the device to check its configuration data for soft-errors. SEDFB is used for One Shot SED mode.

See the following table for port description.

Port Name	I/O	Description
SEDINPROG	O	SED cycle is in progress, asserts High.
SEDDONE	O	SED cycle is complete, asserts High.
SEDERR	O	SED error flag, asserts High.
SEDCLKOUT	O	Optional clock output.

You can refer to the following technical note on the Lattice web site for more information and usage.

- TN1206 - MachXO2 Soft Error Detection (SED) Usage Guide

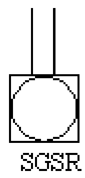
SGSR

Synchronous Release Global Set/Reset Interface

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO2
- MachXO3L

- Platform Manager 2



INPUTS: GSR, CLK

Description

SGSR is used to reset or set all register elements in your design. The SGSR component can be connected to a net from an input buffer or an internally generated net. It is active LOW and when pulsed will set or reset all flip-flops, latches, registers, and counters to the same state as the local set or reset functionality. When input GSR is HIGH, the global signal is released at the positive edge of the clock (CLK).

It is not necessary to connect signals for SGSR to any register elements explicitly. The function will be implicitly connected globally. The functionality of the SGSR for sequential cells without a local set or reset are described in the appropriate library help page.

Note

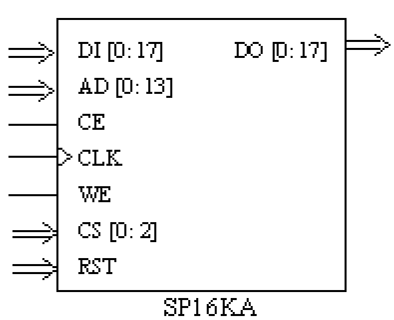
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

SP16KA

16K Single Port Block RAM

Architectures Supported:

- LatticeSC/M



INPUTS: DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7, AD8, AD9, AD10, AD11, AD12, AD13, CE, CLK, WE, CS0, CS1, CS2, RST

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17

ATTRIBUTES:

DATA_WIDTH: 1, 2, 4, 9, 18 (default)

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE: any 3-bit binary value (default: 3'b000)

WRITEMODE: "NORMAL" (default), "WRITETHROUGH"

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_3F: (*Verilog*) 320'hXXX...X (80-bit hexadecimal value)
(*VHDL*) 0xXXX...X (80-bit hexadecimal value)
Default: all zeros

Description

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

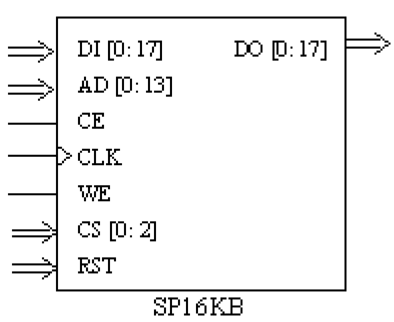
- TN1094 - On-Chip Memory Usage Guide for LatticeSC Devices

SP16KB

Single Port Block RAM

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7, AD8, AD9, AD10, AD11, AD12, AD13, CE, CLK, WE, CS0, CS1, CS2, RST

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17

ATTRIBUTES:

DATA_WIDTH:1, 2, 4, 9, 18 (default)

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE: any 3-bit binary value (default: 0b000)

WRITEMODE: "NORMAL" (default), "WRITETHROUGH"

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_3F: "0xXXX...X" (80-bit hex string) (default: all zeros)

Description

Single Port Block RAM primitive. See Memory Primitives Overview for more information.

You can also refer to the following technical notes on the Lattice web site for port definition, attributes and usage.

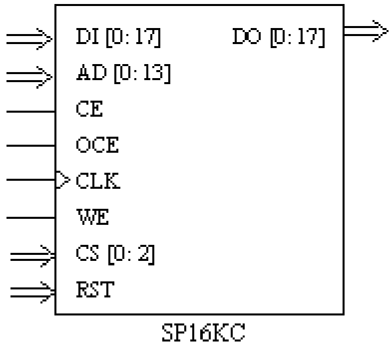
- TN1104 - LatticeECP2/M Memory Usage Guide
- TN1137 - LatticeXP2 Memory Usage Guide

SP16KC

Single Port Block RAM

Architectures Supported:

- LatticeECP3



INPUTS: DI17, DI16, DI15, DI14, DI13, DI12, DI11, DI10, DI9, DI8, DI7, DI6, DI5, DI4, DI3, DI2, DI1, DI0, AD13, AD12, AD11, AD10, AD9, AD8, AD7, AD6, AD5, AD4, AD3, AD2, AD1, AD0, CE, OCE, CLK, WE, CS2, CS1, CS0, RST

OUTPUTS: DO17, DO16, DO15, DO14, DO13, DO12, DO11, DO10, DO9, DO8, DO7, DO6, DO5, DO4, DO3, DO2, DO1, DO0

ATTRIBUTES:

DATA_WIDTH: 1, 2, 4, 9, 18 (default)

REGMODE: "NOREG" (default), "OUTREG"

CSDECODE: any 3-bit binary string (default: "0b000")

WRITEMODE: "NORMAL" (default), "WRITETHROUGH"

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_3F: "0xXXX...X" (80-bit hex string) (default: all zeros)

Description

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

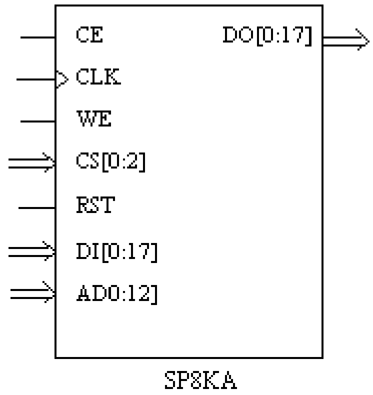
- TN1179 - LatticeECP3 Memory Usage Guide

SP8KA

8K Single Port Block RAM

Architectures Supported:

- LatticeECP/EC
- LatticeXP



INPUTS: CE, CLK, WE, CS0, CS1, CS2, RST, DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7, AD8, AD9, AD10, AD11, AD12

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17

ATTRIBUTES:

REGMODE: "NOREG" (default), "OUTREG"

GSR: "DISABLED" (default), "ENABLED"

WRITEMODE: "NORMAL" (default), "WRITETHROUGH"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE: any 3-bit binary value (default: 111)

DATA_WIDTH: 1, 2, 4, 9, 18 (default)

INITVAL_00 to INITVAL_1F: (*Verilog*) 320'hXXX...X (80-bit hexadecimal value)
(*VHDL*) 0xXXX...X (80-bit hexadecimal value)
Default: all zeros

Description

You can refer to the following technical note on the Lattice web site for EBR port definition, attribute definition and usage.

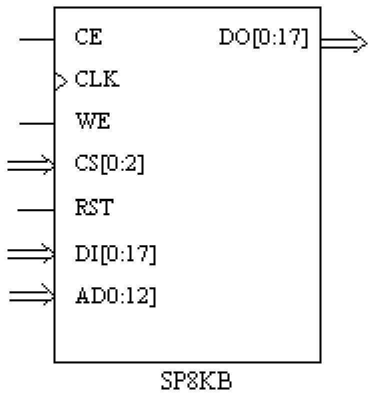
- TN1051 - Memory Usage Guide for LatticeECP/EC and LatticeXP Devices

SP8KB

8K Single Port Block RAM

Architectures Supported:

- MachXO
- Platform Manager



INPUTS: CE, CLK, WE, CS0, CS1, CS2, RST, DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7, AD8, AD9, AD10, AD11, AD12

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17

ATTRIBUTES:

DATA_WIDTH: 1, 2, 4, 9, 18 (default)

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE: any 3-bit binary value (default: 3’b000)

WRITEMODE: "NORMAL" (default), "WRITETHROUGH"

GSR: "DISABLED" (default), "ENABLED"

INITVAL_00 to INITVAL_1F: (*Verilog*) "320'hXXX...X" (80-bit hex string)
(*VHDL*) "0xXXX...X" (80-bit hex string)
Default: all zeros

Description

8K Single Port Block RAM primitive. See Memory Primitives Overview for more information.

You can also refer to the following technical note on the Lattice web site for port definition, attributes and usage.

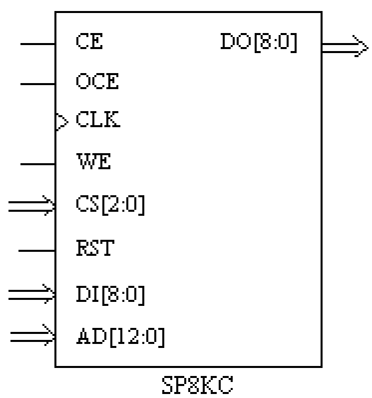
- TN1092 - MachXO Memory Usage Guide

SP8KC

8K Single Port Block RAM

Architectures Supported:

- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: CE, OCE, CLK, WE, CS0, CS1, CS2, RST, DI0, DI1, DI2, DI3, DI4, DI5, DI6, DI7, DI8, DI9, DI10, DI11, DI12, DI13, DI14, DI15, DI16, DI17, AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7, AD8, AD9, AD10, AD11, AD12

OUTPUTS: DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7, DO8, DO9, DO10, DO11, DO12, DO13, DO14, DO15, DO16, DO17

ATTRIBUTES:

DATA_WIDTH: 1, 2, 4, 9 (default)

REGMODE: "NOREG" (default), "OUTREG"

RESETMODE: "SYNC" (default), "ASYNC"

CSDECODE: any 3-bit binary value (default: all zeros)

WRITEMODE: "NORMAL" (default), "WRITETHROUGH", "READBEFOREWRITE"

GSR: "ENABLED" (default), "DISABLED"

INIT_DATA: "STATIC" (default), "DYNAMIC"

INITVAL_00 to INITVAL_1F: (*Verilog*) "320'hXXX...X" (80-bit hex string)
(*VHDL*) "0xXXX...X" (80-bit hex string)
Default: all zeros

Description

8K Single Port Block RAM primitive. See the below table for I/O port description.

Port Name	I/O	Definition
CLK	I	Clock
CE	I	Clock enable
OCE	I	Output clock enable
AD[12:0]	I	Address bus
DI[8:0]	I	Input data
WE	I	Write enable
CS[2:0]	I	Chip select
RST	I	Reset
DO[8:0]	O	Output data

You can also refer to the following technical note on the Lattice web site for port definition, attributes and usage.

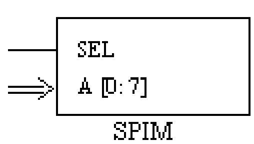
- TN1201 - Memory Usage Guide for MachXO2 Devices

SPIM

SPIM Primitive Distributed RAM

Architectures Supported:

- LatticeECP2/M
- LatticeECP3



INPUTS: SEL, A0, A1, A2, A3, A4, A5, A6, A7

Description

Some devices have a built-in SPI CIB interface that is embedded into the hardware. The SPI CIB interface allows the user to control the dual-boot flash support based on the user’s logic. Users can interface with this hardwired SPI controller through the use of the SPIM primitive. The SPIM primitive have 9 input ports only. The SEL line indicates which SPI Flash device to boot from and the signals [A0..A7] indicate the address from where the device will reboot during reconfiguration.

For further information on SPIm Mode, refer to the following technical notes on the Lattice web site:

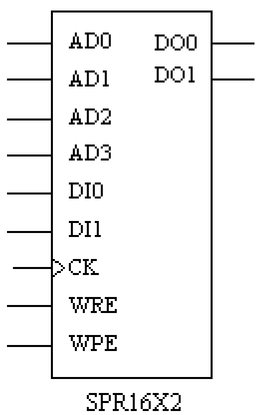
- TN1169 - LatticeECP3 sysCONFIG Usage Guide
- TN1108 - LatticeECP2/M sysCONFIG Usage Guide

SPR16X2

Distributed Single Port RAM

Architectures Supported:

- LatticeSC/M



INPUTS: AD0, AD1, AD2, AD3, DI0, DI1, CK, WRE, WPE

OUTPUTS: DO0, DO1

ATTRIBUTES:

INITVAL: (*Verilog*) 64'hXXXXXXXXXXXXXXXX (16-bit hexadecimal value)
(*VHDL*) 0xXXXXXXXXXXXXXXXX (16-bit hexadecimal value)
Default: all zeros

GSR: "ENABLED" (default), "DISABLED"

Description

The SPR16X2 symbol represents a 16 word by 2 bit asynchronous single port RAM. It has two data inputs DI[1:0], a positive Write Enable (WRE), one positive Write Port Enables (WPE), and one set of address inputs.

The WRE and WPE must be HIGH for the rising clock edge if the write to the RAM is occurring on the falling edge. The data is written into the locations specified by the write address lines AD[3:0] on the next negative clock (CK) edge. The data read operation is always performed asynchronously, with the memory contents specified by the address inputs AD[3:0] output on the data output signals DO[1:0].

In other words, the read operation is asynchronous and is always active. The write operation is synchronous and only occurs when there is a falling edge of the clock and the write enables are high prior to that falling edge.

If desired, the contents of the SPR16X2 can be assigned an initial value, which is loaded into the RAM during configuration. The INITVAL=<*value*> attribute is used to assign the initial value. The <*value*> should consist of 16 hexadecimal data written into the RAM from the highest address to the lowest address. For example, if the following attribute is specified:

INITVAL=0x0123012301230123 (in hex)

it implies that the above data is loaded sequentially from location FH to 0H (where FH would contain the 0 and 0H the 3). If no INITVAL attribute is specified, the RAM is initialized with zeros on configuration.

If the INITVAL is specified as a hex string of 16 values, the values should not be greater than 3, since 3 is setting both memory locations to 1. If a value greater than 3 is used for synthesis or mapping, only the last two (least significant) bits are used for initialization by the mapper.For example,

INITVAL=0x00000000ffffffff

is equivalent to

INITVAL=0x0000000033333333

for mapping purposes, since the first two bits of the "f" are ignored.

You can refer to the following technical note on the Lattice web site for more information and usage.

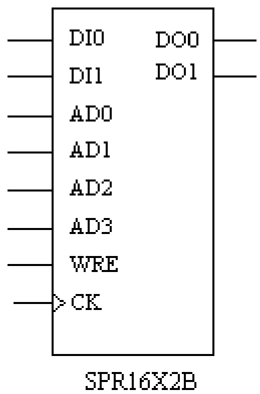
- TN1094 - On-Chip Memory Usage Guide for LatticeSC Devices

SPR16X2B

16 Word by 2 Bit Positive Edge Triggered Write Synchronous Single Port RAM Memory with Positive Write Enable and Positive Write Port Enable (1-Slice)

Architectures Supported:

- LatticeECP/EC
- LatticeXP
- MachXO
- Platform Manager



INPUTS: DI0, DI1, AD0, AD1, AD2, AD3, WRE, CK

OUTPUTS: DO0, DO1

ATTRIBUTES:

INITVAL: (*Verilog*) 64'hXXXXXXXXXXXXXXXX (16-bit hexadecimal value)
(*VHDL*) 0xXXXXXXXXXXXXXXXX (16-bit hexadecimal value)
Default: all zeros

Description

Refer to SPR16X2 for functionality. You can also refer to the following technical notes on the Lattice web site for EBR port definition, attribute definition, and use.

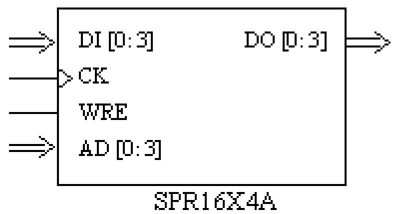
- TN1051 - Memory Usage Guide for LatticeECP/EC and LatticeXP Devices
- TN1092 - MachXO Memory Usage Guide

SPR16X4A

Distributed Single Port RAM

Architectures Supported:

- LatticeECP2/M
- LatticeXP2



INPUTS: DI0, DI1, DI2, DI3, AD0, AD1, AD2, AD3, CK, WRE

OUTPUTS: DO0, DO1, DO2, DO3

Description

PFU based distributed pseudo single port RAM primitive. See Memory Primitives Overview for more information.

You can also refer to the following technical notes on the Lattice web site for port definition, attributes, and use.

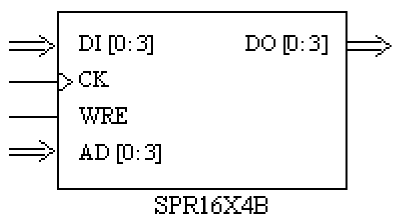
- TN1104 - LatticeECP2/M Memory Usage Guide
- TN1137 - LatticeXP2 Memory Usage Guide

SPR16X4B

Distributed Single Port RAM

Architectures Supported:

- LatticeXP2



INPUTS: DI0, DI1, DI2, DI3, AD0, AD1, AD2, AD3, CK, WRE

OUTPUTS: DO0, DO1, DO2, DO3

ATTRIBUTES:

INITVAL: (*Verilog*) "64'hXXXXXXXXXXXXXXXX" (16-bit hex string)
(*VHDL*) "0xXXXXXXXXXXXXXXXX" (16-bit hex string)
Default: all zeros

Description

PFU based distributed pseudo single port RAM primitive. See Memory Primitives Overview for more information.

You can also refer to the following technical note on the Lattice web site for port definition, attributes, and use.

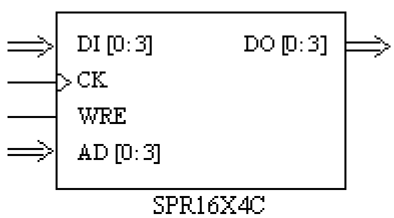
- TN1137 - LatticeXP2 Memory Usage Guide

SPR16X4C

Distributed Single Port RAM

Architectures Supported:

- LatticeECP3
- MachXO2
- MachXO3L
- Platform Manager 2



INPUTS: DI3, DI2, DI1, DI0, AD3, AD2, AD1, AD0, CK, WRE

OUTPUTS: DO3, DO2, DO1, DO0

ATTRIBUTES:

INITVAL: "0xXXXXXXXXXXXXXXXX" (16-bit hex string) (default: all zeros)

Description

PFU based distributed Single Port RAM primitive. See Memory Primitives Overview for individual port description.

You can also refer to the following technical notes on the Lattice web site for port definition, attribute definition, and use.

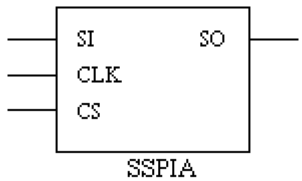
- TN1201 - Memory Usage Guide for MachXO2 Devices
- TN1179 - LatticeECP3 Memory Usage Guide

SSPIA

SSPI TAG Memory

Architectures Supported:

- LatticeXP2



INPUTS: SI, CLK, CS

OUTPUT: SO

ATTRIBUTES:

TAG_INITSIZE: 448, 632, 768, 2184 (default), 2488, 2640, 3384, 3608

TAG_INITIALIZATION: "DISABLED" (default), "ENABLED"

TAG_INITVAL_00 to TAG_INITVAL_0C: "0xXXX...X" (80-bit hex string) (default: all zeros)

Description

Implements the dedicated “TAG Memory” block, which is a one page FLASH non-volatile memory accessible by the hardwired Serial Peripheral Interface port or the JTAG port. This stand-alone TAG memory is ideal for scratch pad memory for mission critical data, board serialization, board revision log and programmed pattern identification.

The XP2 family of devices provides dedicated TAG memory ranging from 632 to 3384 bits depending on device density. The user can read and write to the TAG memory either through the SPI port (External Slave SPI) or the CIB interface (Internal Slave SPI) by setting the SLAVE_SPI_PORT attribute. The software default for access to the Tag memory is the CIB interface. The TAG memory initialization file format is similar to that of EBR.

- The TAG interface through the SPI port (External Slave SPI): When the TAG interface is set for the SPI port, four SPI pins will be reserved for the TAG access through the SPI port.

- The TAG interface through the CIB interface (Internal Slave SPI): When the TAG interface is set for the CIB interface, the user can select any four general purpose IO pins to access the TAG memory using the SPI commands. The four SPI pins on each XP2 device are considered general purpose IOs if they are not reserved using SLAVE_SPI_PORT attribute.

SSPIA Port Description

Port Name	I/O	Description
SI	I	Data input
CLK	I	Clock
CS	I	Chip select
SO	O	Data output

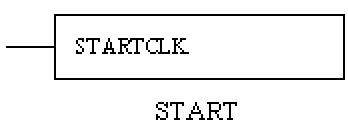
You can refer to LatticeXP2 technical notes on the Lattice web site for more details.

START

Startup Controller

Architectures Supported:

- LatticeECP3
- LatticeXP2
- MachXO2
- MachXO3L
- Platform Manager 2



INPUT: STARTCLK

Description

This primitive determines the user clock for the Wake up sequence. You can instantiate this module in your HDL source to tie a specific user clock to be used in the wake-up sequence instead of the TCK (JTAG), BCLK (SDM), or MCLK/CCLK (sysCONFIG).

START Usage with Verilog HDL

```
module START (STARTCLK);
input  STARTCLK;
endmodule
```

START Usage with VHDL

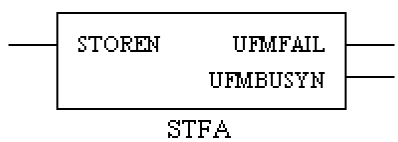
```
COMPONENT START
   PORT(
        STARTCLK     :  IN STD_ULOGIC
       );
END COMPONENT;
```

STFA

Store to Flash Primitive

Architectures Supported:

- LatticeXP2



INPUT: STOREN

OUTPUTS: UFMFAIL, UFMBUSYN

Description

The LatticeXP2 store-to-flash primitive is for user flash module (UFM) operations. The main function of a UFM is to protect the user data from being lost when the system is powered OFF. The user data in the UFM will be used by the system for initialization when the power comes back. To emulate the UFM capability, the users can use the EBR (shadow flash) memory configurations and then transfer the data to UFM through Store-to-Flash operation. The store-to-flash operation is a single-command-two-operation process. When the store-to-flash operation is initiated, an erase-UFM-Flash CIB signal will be enabled to erase the Flash, followed by the transfer-to-flash operation. Once the transfer is done, the flash controller will send a transfer-done signal back to the user logic. During the Store-to-Flash operation, the EBR's are not accessible. There is no difference between regular EBR RAM configuration and shadow flash (UFM) EBR RAM configuration in Lattice Diamond GUI. The presence of a STFA primitive in the design determines EBR RAM configuration. Due to a silicon limitation, the user cannot use the Store-to-Flash operation if the SED is operating in an Always mode. Only one STFA instance in the design is allowed.

STFA Port Description

Port Name	Corresponding Hardware Port Name	I/O	Description
STOREN	storecmdn	I	Initiates to store the EBR content to Flash
UFMFAIL	ufm_fail	O	Store to Flash operation failed
UFMBUSYN	fl_busyn	O	Tells the user whether the FLASH is in busy state or not

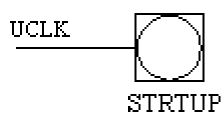
You can refer to LatticeXP2 technical notes on the Lattice web site for more information.

STRTUP

Startup Controller

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeSC/M



INPUT: UCLK

Description

After configuration, the FPGA enters the start-up phase, which is the transition between the configuration and operational states. Normally, the relative timing of the following three events is triggered by the configuration clock (CCLK): DONE going high, release of the set/reset of internal FFs, and activation of user I/Os. The three events can also be triggered by a user clock, UCLK. This allows the start-up to be synchronized by a known system clock. For more detailed information refer to an available data book or contact technical support.

Another set of bitstream options for the STRTUP block allows the DONE pin to be held low and then released to be used with either CCLK or UCLK to control the release of the set/reset of internal FFs and the activation of user I/Os. This allows the synchronization of the start-up of multiple FPGAs.

UCLK: User defined clock to trigger DONE going high, release of set/reset of internal FFs, and activation of user I/Os.

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in Schematic Editor.

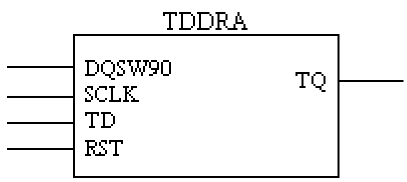
T

TDDRA

Tristate for DQ/DQS of PIC Cell

Architectures Supported:

- MachXO2
- Platform Manager 2



INPUTS: DQSW90, SCLK, TD, RST

OUTPUT: TQ

ATTRIBUTES:

GSR: "ENABLED" (default), "DISABLED"

DQSW90_INVERT: "DISABLED" (default), "ENABLED"

Description

TDDRA is the tristate for DQ/DQS of the PIC cell. It is used for right side only. See the below table for the port description.

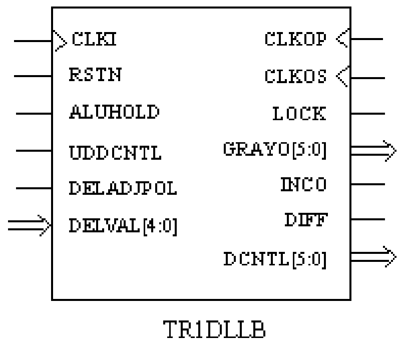
Signal	I/O	Description
DQSW90	I	Shifts the DQS signal by 90 degree
SCLK	I	Clock from the CIB
TD	I	Tristate signal
RST	I	RESET to this block from the CIB
TQ	O	Tristate output for DQ

TR1DLLB

Time Reference DLL with Dynamic Delay Adjustment

Architectures Supported:

- LatticeECP3



INPUTS: CLKI, RSTN, ALUHOLD, UDDCNTL, DELADJPOL, DELVAL4, DELVAL3, DELVAL2, DELVAL1, DELVAL0

OUTPUTS: CLKOP, CLKOS, LOCK, INCO, DIFF, GRAYO5, GRAYO4, GRAYO3, GRAYO2, GRAYO1, GRAYO0, DCNTL5, DCNTL4, DCNTL3, DCNTL2, DCNTL1, DCNTL0;

ATTRIBUTES:

CLKOP_PHASE: 0 (default), 90, 180, 270, 360

CLKOS_PHASE: 0 (default), 90, 180, 270, 360

CLKOS_FPHASE: 0, 11, 22, 33, 45, 56, 67, 78, 90, 101 (default), 112, 123, 135, 146, 157, 169, 191, 202, 214, 225, 236, 247, 259, 281, 292, 304, 315, 326, 337, 349

CLKOS_DIV: 1 (default), 2, 4

GSR: "DISABLED" (default), "ENABLED"

CLKOS_FPHASE _ADJVAL: integers -20~20 (default: 0)

ALU_LOCK_CNT: integers 3~15 (default: 3)

ALU_UNLOCK_CNT: integers 3~15 (default: 3)

GLITCH_TOLERANCE: integers 0~7 (default: 2)

LOCK_DELAY: integers 0~1000 (in ns) (default: 100)

CLKOP_DUTY50: "DISABLED" (default), "ENABLED"

CLKOS_DUTY50: "DISABLED" (default), "ENABLED"

Description

TRDLLB specifies the Time Reference operation mode for the general purpose DLL (GDLL). It supports the 1G SPI4.2 interface. The feedback connection is not required for this mode and hence the CLKFB is not captured on the TRDLLB primitive. In this mode, the CLKFB should be tied to GND.

Port Description

Port Name	Optional	Logical Capture Port Name
ALUHOLD	YES	HOLD
RSTN	YES	RSTN
UDDCNTL	YES	UDDCNTL
CLKI	NO	CLKI
CLKOP	YES	CLKOP
CLKOS	YES	CLKOS
LOCK	NO	LOCK
GRAYO[5:0]	YES	GRAY_OUT[5:0]
INCO	YES	INC_OUT
DIFF	YES	DIFF
DCNTL[5:0]	NO	DCNTL[5:0]
DELADJPOL	YES	DELADJPOL
DELVAL[4:0]	YES	DELVAL[4:0]

For more information, refer to the following technical note on the Lattice web site:

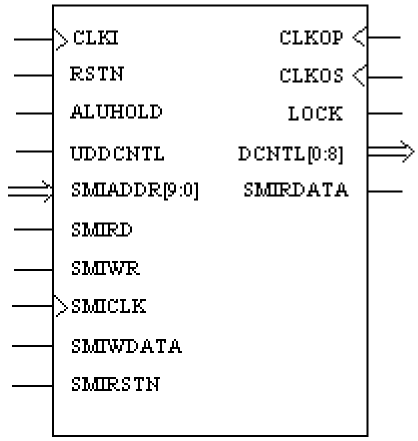
- TN1178 - LatticeECP3 sysCLOCK PLL/DLL Design and Usage Guide

TRDLLA

Time Reference Delay

Architectures Supported:

- LatticeECP2/M
- LatticeSC/M



TRDLLA

INPUTS: CLKI, RSTN, ALUHOLD, UDDCNTL, SMIADDR9, SMIADDR8, SMIADDR7, SMIADDR6, SMIADDR5, SMIADDR4, SMIADDR3, SMIADDR2, SMIADDR1, SMIADDR0, SMIRD, SMIWR, SMICLK, SMIWDATA, SMIRSTN

OUTPUTS: CLKOP, CLKOS, LOCK, DCNTL0, DCNTL1, DCNTL2, DCNTL3, DCNTL4, DCNTL5, DCNTL6, DCNTL7, DCNTL8, SMIRDATA

ATTRIBUTES:

CLKOP_PHASE: 0 (default), 90, 180, 270, 360

CLKOS_PHASE: (0, 90, 180, 270, 360) + (0, 11, 22, 45) (default: 0)

CLKOS_FPHASE: 0 (default), 11, 22, 45

CLKOP_DUTY50: "DISABLED" (default), "ENABLED"

CLKOS_DUTY50: "DISABLED" (default), "ENABLED"

CLKOS_DIV: 1 (default), 2, 4

GSR: "DISABLED" (default), "ENABLED"

CLKOS_FDEL_ADJ: "DISABLED" (default), "ENABLED"

CLKOS_FPHASE _ADJVAL: integers -127~127 (default: 0)

ALU_LOCK_CNT: integers 3~15 (default: 3)

ALU_UNLOCK_CNT: integers 3~15 (default: 3)

GLITCH_TOLERANCE: integers 0~7 (default: 2 for LatticeECP2/M, 0 for LatticeSC/M)

LOCK_DELAY: integers 0~1000 (in ns) (default: 100)

(*LatticeSC/M only*) DCNTL_ADJVAL: integers -127~127 (default: 0)

(*LatticeSC/M only*) SMI_OFFSET: 0x400~0x7FF (default: 12'h410)

(*LatticeSC/M only*) MODULE_TYPE: "TRDLLA"

(*LatticeSC/M only*) IP_TYPE: "TRDLLA"

Description

TRDLLA will generate four phases of the clock, 0, 90, 180, 270 degrees, along with the control setting used to generate these phases. This mode features registered control bit output with separate enable, addition and subtraction on the outgoing control bits, lock achieved starting from minimum delay which guarantees lock to first harmonic (fundamental frequency), and four available output phases (0, 90, 180, 270) degrees. This requires internal feedback only, a maximum frequency 700MHz, and a minimum frequency 100MHz.

For more information, see the following technical notes on the Lattice web site:

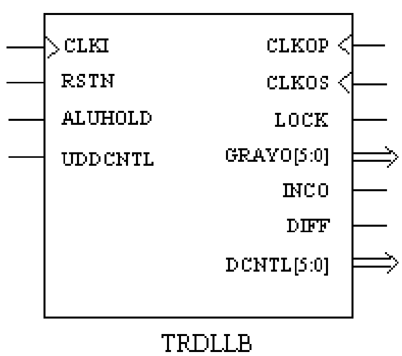
- TN1098 - LatticeSC sysCLOCK and PLL/DLL User’s Guide
- TN1103 - LatticeECP2 sysCLOCK PLL Design and Usage Guide

TRDLLB

Time Reference DLL

Architectures Supported:

- LatticeECP3



INPUTS: CLKI, RSTN, ALUHOLD, UDDCNTL

OUTPUTS: CLKOP, CLKOS, LOCK, INCO, DIFF, GRAYO5, GRAYO4, GRAYO3, GRAYO2, GRAYO1, GRAYO0, DCNTL5, DCNTL4, DCNTL3, DCNTL2, DCNTL1, DCNTL0

ATTRIBUTES:

CLKOP_PHASE: 0 (default), 90, 180, 270, 360

CLKOS_PHASE: 0 (default), 90, 180, 270, 360

CLKOS_FPHASE: 0, 11, 22, 33, 45, 56, 67, 78, 90, 101 (default), 112, 123, 135, 146, 157, 169, 191, 202, 214, 225, 236, 247, 259, 281, 292, 304, 315, 326, 337, 349

CLKOS_DIV: 1 (default), 2, 4

GSR: "DISABLED" (default), "ENABLED"

CLKOS_FPHASE _ADJVAL: integers -20~20 (default: 0)

ALU_LOCK_CNT: integers 3~15 (default: 3)

ALU_UNLOCK_CNT: integers 3~15 (default: 3)

GLITCH_TOLERANCE: integers 0~7 (default: 2)

LOCK_DELAY: integers 0~1000 (in ns) (default: 100)

CLKOP_DUTY50: "DISABLED" (default), "ENABLED"

CLKOS_DUTY50: "DISABLED" (default), "ENABLED"

Description

TRDLLB specifies the Time Reference operation mode for the general purpose DLL (GDLL). The feedback connection is not required for this mode and hence the CLKFB is not captured on the TRDLLB primitive. In this mode, the CLKFB should be tied to GND.

Port Description

Port Name	Optional	Logical Capture Port Name
ALUHOLD	YES	HOLD
RSTN	YES	RSTN
UDDCNTL	YES	UDDCNTL
CLKI	NO	CLKI
CLKOP	YES	CLKOP
CLKOS	YES	CLKOS
LOCK	NO	LOCK
GRAYO[5:0]	YES	GRAY_OUT[5:0]
INCO	YES	INC_OUT
DIFF	YES	DIFF
DCNTL[5:0]	NO	DCNTL[5:0]

For more information, refer to the following technical note on the Lattice web site:

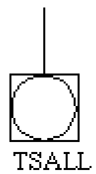
- TN1178 - LatticeECP3 sysCLOCK PLL/DLL Design and Usage Guide

TSALL

Global Tristate Interface

Architectures Supported:

- LatticeSC/M
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUT: TSALL (TSALLN for LatticeSC/M)

Description

TSALL is used to tristate buffers in your design. The TSALL component is connected to a net to drive all output and bidirectional buffers into a HIGH impedance state when active LOW.

It is not necessary to connect signals to buffers explicitly. The function will be implicitly connected globally.

Note

- The TSALL component may be driven by general FPGA logic or by the read-configuration block. In the latter case, the TSALL block must be driven by a buffer located at the RDCGFN pin. When locating the TSALL to the RDCFGN, you must do this by explicitly designating “RDCFGN” in the attribute. Check with customer support or with FAEs for more details.
- This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

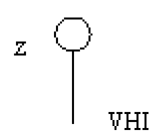
V

VHI

Logic High Generator

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



OUTPUT: Z

Note

- It is possible that this primitive will be optimized by the back-end tool before place and route.
- This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

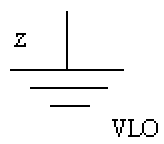
VLO

Logic Low Generator

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M

- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- Platform Manager
- Platform Manager 2



OUTPUT: Z

Note

- It is possible that this primitive will be optimized by the back-end tool before place and route.
- This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

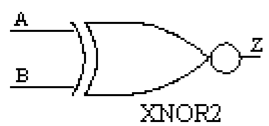
X

XNOR2

2-Input Exclusive NOR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B

OUTPUT: Z

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

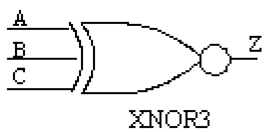
XNOR3

3-Input Exclusive NOR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M

- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C

OUTPUT: Z

Note

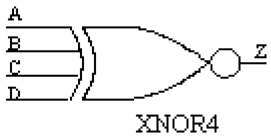
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

XNOR4

4-Input Exclusive NOR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D

OUTPUT: Z

Note

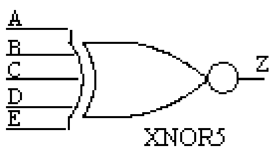
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

XNOR5

5-Input Exclusive NOR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D, E

OUTPUT: Z

Note

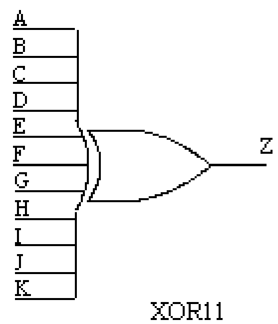
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

XOR11

11-Input Exclusive OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D, E, F, G, H, I, J, K

OUTPUT: Z

Note

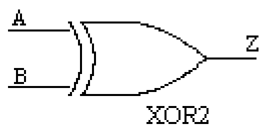
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

XOR2

2-Input Exclusive OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B

OUTPUT: Z

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

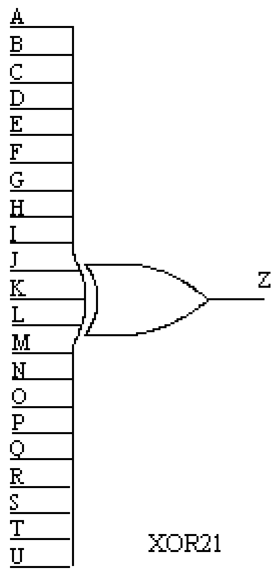
XOR21

21-Input Exclusive OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2

- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U

OUTPUT: Z

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

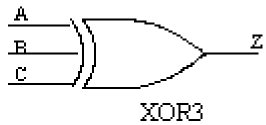
XOR3

3-Input Exclusive OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2

- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C

OUTPUT: Z

Note

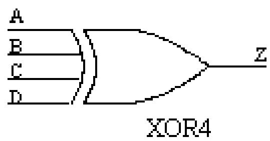
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

XOR4

4-Input Exclusive OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- MachXO3L
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D

OUTPUT: Z

Note

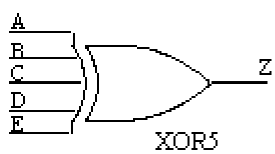
This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

XOR5

5-Input Exclusive OR Gate

Architectures Supported:

- LatticeECP/EC
- LatticeECP2/M
- LatticeECP3
- LatticeSC/M
- LatticeXP
- LatticeXP2
- MachXO
- MachXO2
- Platform Manager
- Platform Manager 2



INPUTS: A, B, C, D, E

OUTPUT: Z

Note

This primitive is also available as a schematic symbol. You can add it to your schematic using the Add > Symbol command in the Schematic Editor.

Primitive-Specific HDL Attributes

The following is a comprehensive list of HDL attributes that are commonly set automatically during module generation and are associated with specific primitives. We list these attributes here mainly for purposes of identification.

In almost all cases, it is not recommended that any of these attributes be edited manually. They should appear in your source as a result of module generation in IPexpress only. There are many interdependencies that exist between certain related attributes and their valid values on an architecture or device basis. These interdependencies make it impractical to simply edit the source HDL. If you were to do so, its very likely an invalid value may result in a failure in your design.

List of Primitive-Specific HDL Attributes

The below table lists all the primitive-specific attributes in alphabetic order.

Attribute	Type	Allowed Values	Default	Description
AEPOINTER	Binary	(*LatticeSCM*) 15-bit binary value; (*MachXO/Platform Manager*) 14-bit binary value	All zeros	Specifies the Almost Empty Flag Pointer. AEPOINTER1 refers to the Almost Empty Flag Pointer 1.
AFPOINTER	Binary	(*LatticeSCM*) 15-bit binary value; (*MachXO/Platform Manager*) 14-bit binary value	All zeros	Specifies the Almost Full Flag Pointer. AFPOINTER1 refers to the Almost Full Flag Pointer 1.
ALU_INIT_CNTVAL	Integer	(*LatticeECP3*) 0 to 31; (*Others*) 0, 4, 8, 12, 16, 32, 48, 64, 72 (0 = DISABLED)	0	Specifies the minimum number of delay taps that ALU will count for. This forces the ALU to count for a minimum number of delay taps before it can find lock, and prevents the DLL finding lock at the minimum possible delay setting and then "falling off the end" of the delay chain if the input clock has jitter. Used in all clock injection cancellation modes where the lock point is not predictable.
ALU_LOCK_CNT	Integer	3 to 15	3	Specifies the Lock Count Cycles. Attached to a DLL-type primitive.
ALU_UNLOCK_CNT	Integer	3 to 15	3	Specifies the Unlock Count Cycles. Attached to a DLL-type primitive.
ASYNC_RESET_RELEASE	String	SYNC, ASYNC	SYNC	Specifies reset release when the reset mode is ASYNC.

Attribute	Type	Allowed Values	Default	Description
BANKID	Integer	0, 1, 2, 3, 4, 5	0	Specifies the ID of the bank that enables dynamic InRD control for the BCINRD primitive.
BGOFF	Boolean	TRUE, FALSE	FALSE	Turns on or off Bandgap when in standby.
BOOT_OPTION	String	INTERNAL, EXTERNAL	INTERNAL	Specifies device boot from external SPI flash or internal flash.
CAS_MATCH_REG	Boolean	TRUE, FALSE	FALSE	Specifies the Cascade Match Register option. Attached to DSP primitives such as MULT9X9C and MULT18X18C.
CHECKALWAYS	Boolean	ENABLED, DISABLED	DISABLED	When set to ENABLED, makes the SED (Soft Error Detect) run automatically every time upon power up and after device configuration. The software will set signals from the SED IP that puts it in an “Always Running” state.
CLKFB_DIV	Integer	Vary	1	Specifies the CLKFB N Divider setting. Attached to a PLL-type primitive (such as EHXPLLB).
CLKFB_FDEL	Integer	0, 100, 200, ..., 700	0	Specifies the CLKFB Fine Delay setting for the EHXPLLA primitive.
CLKI_DIV	Integer	Vary	1	Specifies the CLKI M Divider setting. Attached to a PLL or DLL primitive (such as EHXPLLB and CIDDLLA).
CLKI_FDEL	Integer	0, 100, 200, ..., 700	0	Specifies the CLKI Fine Delay setting for the EHXPLLA primitive.
CLKMODE	String	ECLK, SCLK	ECLK	Specifies the edge clock or system clock for the CLKCNTL primitive.
CLKOK_BYPASS	Boolean	ENABLED, DISABLED	DISABLED	Enables or disables the GPLL clock bypass feature. When enabled, this feature allows the input reference clock (CLK) to bypass the PLL and directly drive CLKOP, CLKOS, and CLKOK. If CLKOP is used for bypass, the PLL is no longer functional and cannot be used as a PLL. The CLKOS and CLKOK can be used for bypass without affecting the operation of the loop. IPexpress includes selections for CLKOP, CLKOS, and CLKOK bypass.
CLKOK_DIV	Integer	Even integers from 2 to 128	2	Specifies the CLKOK K Divider setting. Attached to a PLL-type primitive (such as EHXPLLB).

Attribute	Type	Allowed Values	Default	Description
CLKOK_INPUT	String	CLKOP, CLKOS	CLKOP	Specifies the CLKOK divider input. Attached to a PLL-type primitive. (such as EHXPLLF).
CLKOP_BYPASS	Boolean	ENABLED, DISABLED	DISABLED	Enables or disables the GPLL clock bypass feature. When enabled, this feature allows the input reference clock (CLK) to bypass the PLL and directly drive CLKOP, CLKOS, and CLKOK. If CLKOP is used for bypass, the PLL is no longer functional and cannot be used as a PLL. The CLKOS and CLKOK can be used for bypass without affecting the operation of the loop. IPexpress includes selections for CLKOP, CLKOS, and CLKOK bypass.
CLKOP_DIV	Integer	Vary	1 or 8	Specifies the CLKOP V Divider setting. Attached to a PLL-type primitive (such as EHXPLLB). **Note**: CLKOP_DIV value must be calculated to maximize the FVCO within the specified range based on CLKI_DIV and CLKFB_DIV values for optimum performance.
CLKOP_DUTY50	Boolean	ENABLED, DISABLED	DISABLED	Enables or disables the CLKOP Duty Cycle. Attached to a PLL-type primitive (such as EHXPLLA).
CLKOP_MODE	String	BYPASS, FDEL0, VCO, DIV	BYPASS	Specifies the CLKOP Select for the EHXPLLA primitive.
CLKOP_PHASE	Integer	0, 90, 180, 270, 360	0	Specifies the CLKOP Phase setting. Attached to a DLL-type primitive (such as TRDLLA).
CLKOP_TRIM_DELAY	Integer	0 to 7	0	Specifies the CLKOP Duty Trim Polarity Delay. Attached to a PLL-type primitive.
CLKOP_TRIM_POL	String	FALLING, RISING	FALLING for LatticeXP2; RISING for LatticeECP3	Specifies the CLKOP Duty Trim Polarity. Attached to a PLL-type primitive.

Attribute	Type	Allowed Values	Default	Description
CLKOS_BYPASS	Boolean	ENABLED, DISABLED	DISABLED	Enables or disables the GPLL clock bypass feature. When enabled, this feature allows the input reference clock (CLK) to bypass the PLL and directly drive CLKOP, CLKOS, and CLKOK. If CLKOP is used for bypass, the PLL is no longer functional and cannot be used as a PLL. The CLKOS and CLKOK can be used for bypass without affecting the operation of the loop. IPexpress includes selections for CLKOP, CLKOS, and CLKOK bypass.
CLKOS_DIV	Integer	1, 2, 4 for DLL primitives; 1 to 64 for PLL primitives	1	Specifies the CLKOS Divider setting. Attached to a PLL- or DLL-type primitive (such as EHXPLLA and CIDDLLA).
CLKOS_DUTY50	Boolean	ENABLED, DISABLED	DISABLED	Enables or disables the CLKOS Duty Cycle. Attached to a PLL-type primitive (such as EHXPLLA).
CLKOS_FDEL	Integer	0, 100, 200, ..., 700	0	Specifies the CLKOS Fine Delay setting for the EHXPLLA primitive.
CLKOS_FDEL_ADJ	Boolean	ENABLED, DISABLED	DISABLED	Specifies the CLKOS DEL Manual Setting Adjust Value. Attached to a DLL-type primitive (such as TRDLLA).
CLKOS_FPHASE	Integer	Vary	0	Specifies the CLKOS Fine Phase setting. Attached to a DLL-type primitive (such as TRDLLA).
CLKOS_FPHASE_ADJVAL	Integer	-20 to 20	0	Specifies the CLKOS Fine Phase Adjust Value. Attached to a DLL-type primitive (such as TRDLLA).
CLKOS_MODE	String	BYPASS, FDEL, VCO, DIV	BYPASS	Specifies the CLKOS Select for the EHXPLLA primitive.
CLKOS_PHASE	Integer	Vary	0	Specifies the CLKOS Phase setting. Attached to a DLL-type primitive (such as TRDLLA).
CLKOS_TRIM_DELAY	Integer	0 to 3	0	Specifies the CLKOS Duty Trim Polarity Delay. Attached to a PLL-type primitive.
CLKOS_TRIM_POL	String	RISING, FALLING	RISING	Specifies the CLKOS Duty Trim Polarity Delay. Attached to a PLL-type primitive.
CLKOS_VCODEL	Integer	0 to 31	0	Specifies the CLKOS VCO Delay setting for the EHXPLLA primitive.

Attribute	Type	Allowed Values	Default	Description
CRUDIV	Integer	1.0, 2.0, 3.5, 4.0, 5.0	5.0	Sets the divider setting for the DIVCLK output.
CSDECODE	Binary	2- or 3-bit binary value	Vary	Attached to a single-port block RAM primitive. The CSDECODE value determines the decoding value of CS[2:0]. A value set to “000” means that the memory is selected if CS[2:0]=1’B000.
DATA_WIDTH	Integer	1, 2, 4, 9, 18 for DATA_WIDTH, DATA_WIDTH_A, and DATA_WIDTH_B; 1, 2, 4, 9, 18, 36 for DATA_WIDTH_R and DATA_WIDTH_W	9, 18, or 36	Specifies the Data Word Width. Attached to a memory type primitive.
DCNTL_ADJVAL	Integer	-127 to 127	0	Specifies the Adjust Delay Control. Attached to a DLL-type primitive.
DCSMODE	String	NEG, POS, HIGH_LOW, HIGH_HIGH, LOW_LOW, LOW_HIGH, CLK0, CLK1	NEG	Sets the particular mode for the DCS primitive. Refer to DCSMODE Values for more information.
DEL_ADJ	String	PLUS, MINUS	PLUS	Specifies the delay adjustment sign bit.
DEL_VAL	Integer	0 to 127 if DEL_ADJ=PLUS; 1 to 128 if DEL_ADJ=MINUS	0	Specifies the delay adjustment offset.
DEL[0,1,2,3,4]_GRAY	Boolean	ENABLED, DISABLED	DISABLED	Specifies gray in for DEL0, DEL1, DEL2, DEL3, and DEL4. Attached to a DLL-type primitive.
DEL_MODE	String	SCLK_ZEROHOLD, ECLK_ALIGNED, ECLK_CENTERED, SCLK_ALIGNED, SCLK_CENTERED, USER_DEFINED	USER_DEFINED	Controls whether the fixed delay value is dependent on a certain interface or user-defined delay value.
DEL_VALUE	String	DELAY0, DELAY1, DELAY2, ..., DELAY31	DELAY0	Specifies user-defined delay value.

Attribute	Type	Allowed Values	Default	Description
DELAY_CNTL	String	DYNAMIC, STATIC	STATIC	Specifies the Delay Control mode. Attached to a PLL-type primitive (such as EHXPLLB). The DYNAMIC mode switches delay control between DYNAMIC and STATIC depending upon the input logic of the DDAMODE pin. In the STATIC mode, delay inputs are ignored.
DELAY_PWD	Boolean	ENABLED, DISABLED	DISABLED	Enables or disables the CLKOS Fine Delay Powerdown. Attached to a PLL-type primitive. When set to ENABLED, the f_fdelay_pwd fuse will be set to HIGH to disable the fine delay circuitry for power saving. When set to DISABLED, the f_fdelay_pwd fuse will be set to LOW to enable the fine delay circuitry.
DELAY_VAL	Integer	0 to 15	0	Specifies the CLKOS Fine Delay Value. Attached to a PLL-type primitive.
DEV_DENSITY	String	Vary	Vary	Specifies the device density.
DIV	Integer	1, 2, 4 (1:off) for the CLKDIV primitive; 2.0, 3.5, 4.0 for the CLKDIVC primitive; 1, 2, 4, 8, 16, 32, 64, 128 for the OSCA primitive	1 or 2 or 2.0	Specifies the Divider setting.
DQS_LI_DEL_ADJ	String	PLUS, MINUS	Vary	Adjusts the sign delay offset direction for input DDR. For DQSBUFH, it adjusts the sign bit for the READ delay.
DQS_LI_DEL_VAL	Integer	0 to 63 or 0 to 127 if DQS_LI_DEL_ADJ=PLUS; 1 to 64 or 1 to 128 if DQS_LI_DEL_ADJ =MINUS	Vary	Specifies the delay value for input DDR.
DQS_LO_DEL_ADJ	String	PLUS, MINUS	PLUS	Adjusts the sign delay offset direction for output DDR. For DQSBUFH, it adjusts the sign bit for the WRITE delay.

Attribute	Type	Allowed Values	Default	Description
DQS_LO_DEL_VAL	Integer	0 to 63 or 0 to 127 if DQS_LO_DEL_ADJ= PLUS; 1 to 64 or 1 to 128 if DQS_LO_DEL_ADJ= MINUS	0	Specifies the delay value for output DDR.
DQSW90_INVERT	String	DISABLED, ENABLED	DISABLED	Selects the clock polarity for the second FF of the tristate cell for the DQS pin. Only used for the DQS during DDR write.
DR_CONFIG	String	DISABLED, ENABLED	DISABLED	Indicates whether the primitive is used for data recovery configuration or not. If it is for data recovery configuration then the correct GBB timing will be set. This attribute is required for mapper and is not required for simulation.
DUTY	Integer	Vary	4 or 8	Specifies the duty cycle floating point percentage value. Used to control the duty cycle modes of PLL primitives such as EHXPLLB.
DYNDEL_CNTL	String	STATIC, DYNAMIC	DYNAMIC	Enables the static or dynamic delay. Attached to a DQSBUF primitive (such as DQSBUFB).
DYNDEL_TYPE	String	NORMAL, SHIFTED	NORMAL	Specifies the value of the Static Delay input to the write portion of the DQSBUFD or DQSBUFE module that controls the clock inversion. NORMAL: 0-degree phase shift; SHIFTED: 180-degree phase shift through clock inversion.
DYNDEL_VAL	Integer	0 to 127	0	Specifies the value of the Static Delay input to the write portion of the DQSBUFD or DQSBUFE module.
ENCRYPTION	String	ON, OFF	OFF	Specifies the encryption feature.
ER1, ER2	Boolean	ENABLED, DISABLED	ENABLED	Lattice supports two private JTAG instructions ER1 (0x32) and ER2 (0x38). If the ER1 instruction is shifted into the JTAG instruction register, JRTI1 will go high when the TAP controller is in the Run-Test/Idle state. If the ER2 instruction is shifted into the JTAG instruction register, JRTI2 will go high when the TAP controller is in the Run-Test/Idle state.

Attribute	Type	Allowed Values	Default	Description
FB_MODE	String	INTERNAL, CLOCKTREE, EXTERNAL	CLOCKTREE	Defines PLL clock resources in the feedback mode.
FDEL	Integer	-8 to 8	0	Specifies the fine delay adjust setting. Attached to a PLL-type primitive (such as EHXPLLB).
FIN	Real	Vary (in MHz)	100.0	Specifies the input frequency (MHz) designation for a PLL/DLL primitive (such as EHXPLLB, DQSDLLC).
FORCE_MAX_DELAY	Boolean	YES, NO	NO	Used to bypasses the DLL-locking procedure at low frequency. When FIN $\leq$ 30 MHz (pending PDE result), the software sets this attribute to YES. Then DQSDLL will not go through the locking procedure but will be locked to the maximum delay steps.
FORCE_ZERO_BARREL_SHIFT	Boolean	ENABLED, DISABLED	DISABLED	When set to ENABLED, forces zeros to 18 MSB of shift for barrel shift. Attached to the ALU54A primitive.
FWFT	Boolean	ENABLED, DISABLED	DISABLED	First word fall through.
FULLPOINTER	Binary	(*LatticeSC/M*) 15-bit binary value; (*MachXO/Platform Manager*) 14-bit binary value	All zeros	Specifies the Full Flag Pointer. Attached to a FIFO primitive (such as FIFO8KA). FULLPOINTER1 refers to the Full Flag Pointer 1.
GLITCH_TOLERANCE	Integer	0 to 7	2	Specifies the Programmable Glitch Tolerance. Attached to a DLL-type primitive.
GSR	Boolean	ENABLED, DISABLED	Vary	Enables or disables the Global Set/Reset (GSR) for all registered primitives. Applicable to registers, PLLs, and memories such as SP8KA, DP8KA and the like.
INIT	Hexadecimal	Hex value or string	All zeros	Initializes the look-up table values. INIT is required to specify the look-up table values for the LUT primitives (ORCALUT4, 5, 6, 7, or 8). See ORCALUT4 for more information on INIT attribute usage.

Attribute	Type	Allowed Values	Default	Description
INIT_DATA	String	STATIC, DYNAMIC	STATIC	Defines whether or not the memory file can be updated. STATIC – Memory values are stored in the User Flash Memory (UFM) or bitstream but can be shared and can not be updated. DYNAMIC– Memory values are stored in the UFM and can be updated by user logic knowing the EBR address locations.
INITVAL	Hexadecimal	Hex value or string	All zeros	Specifies the initialization value for RAM type primitives (such as SPR16X2 and DPR16X2). These primitives carry prescribed initialization values, for example, DPR16X2 has an initialization value of 0x0000000000000000.
INJECT	Boolean	YES, NO	YES	This injection attribute is not a user selection. It is for software use only. Attached to a Carry Chain primitive.
IP_TYPE	String	EHXPLLA, CIDDLLA, CIMDLLA, TRDLLA, SDCDLLA	Vary	This attribute is not a user selection. It is for software use only. Attached to a PLL- or DLL-type primitive.
ISI_CAL	String	BYPASS, DEL1, DEL2, DEL3, DEL4, DEL5, DEL6, DEL7	BYPASS	Sets the ISI correction values in the ODDRX2D and ODDRX2DQSA blocks.
JTAG_FLASH_PRGRM	Boolean	ENABLED, DISABLED	ENABLED	When set to ENABLED, enables the use of the ispJTAG interface to program or write to Flash devices. Refer to TN1100 - SPI Serial Flash Programming Using ispJTAG in LatticeSC Devices on the Lattice Web site for more information.
LEGACY	Boolean	ENABLED, DISABLED	DISABLED	This attribute is required to support LatticeECP2 to LatticeECP3 mapping. Attached to the ALU54A primitive.
LOCK_CYC	Integer	Integer value	2	This lock cycle attribute is not a user selection. It is for software use only. Attached to a PLL- or DLL-type primitive.

Attribute	Type	Allowed Values	Default	Description
LOCK_DELAY	Integer	0 to 1000 (in ns)	100	This is a PLL-lock time attribute used for simulation. If you wish to enter other than the default value of 100 ns, it can be done by adding this attribute in the DEFPARAM section of the code generated by IPexpress. You can also set this attribute in the Spreadsheet view.
LOCK_SENSITIVITY	String	HIGH, LOW	LOW	This DLL configuration attribute selects greater or less sensitivity to the jitter. **Note**: There is a known issue for the LatticeXP family. The DQSDLL attribute LOCK_SENSITIVITY will always be set to LOW even if you attempt to set it to HIGH. Devices impacted are LFXP20E/C, LFXP15E/C, LFXP10E/C, LFXP6E/C and LFXP3E/C.
LPDDR	String	ENABLED, DISABLED	DISABLED	Turns the LPDDR feature on or off.
LSRMODE	String	EDGE, LOCAL	LOCAL	Attached to DDR and ISR primitives (such as IDDRA and ISRX1A), this attribute takes the EDGE and LOCAL mode options, which allows you to choose Local Set Reset or the Edge Set Reset.
MASK_ADDR	Hexadecimal	Any 4-bit hex value	All zeros	Specifies the starting mask address for the “care bits” mask. Attached to a SED-type primitive.
MASK01	Hexadecimal	Any 14-bit hex value	All zeros	Specifies the mask for EQZM/EQOM. Attached to the ALU54A primitive.
MASKPAT	Hexadecimal	Any 14-bit hex value	All zeros	Specifies the mask for EQPAT/EQPATB. Attached to the ALU54A primitive.
MASKPAT_SOURCE	String	STATIC, DYNAMIC	STATIC	Specifies the EQPAT/EQPATB source setting. Attached to the ALU54A primitive. MASKPAT_SOURCE and MCPAT_SOURCE cannot be DYANMIC at the same time.
MCCLK_FREQ	String	Vary	2.5 or 3.1	Controls the master clock frequency.
MCPAT	Hexadecimal	Any 14-bit hex value	All zeros	Specifies the MEM Cell Pattern. Attached to the ALU54A primitive.

Attribute	Type	Allowed Values	Default	Description
MCPAT_SOURCE	String	STATIC, DYNAMIC	STATIC	Specifies the MEM Cell Pattern source setting. Attached to the ALU54A primitive. MASKPAT_SOURCE and MCPAT_SOURCE cannot be DYANMIC at the same time.
MEMMODE	String	DISABLED, ENABLED	DISABLED	Indicates the memory mode or generic mode.
MODULE_TYPE	String	EHXPLLA, CIDDLLA, CIMDLLA, TRDLLA, SDCDLLA	Vary	This attribute is not a user selection. It is for software use only. Attached to a PLL- or DLL-type primitive.
MULT_BYPASS	Boolean	ENABLED, DISABLED	DISABLED	Enables or disables Multiplier Output Bypass. Attached to DSP primitives such as MULT9X9C and MULT18X18C.
MULT9_MODE	Boolean	ENABLED, DISABLED	DISABLED	Enables or disables the operation in the Mult9 mode. Attached to the ALU54A primitive.
NOM_FREQ	Real	Vary	Vary	Specifies the nominal frequency (in MHz) for oscillator primitives.
NRZMODE	String	DISABLED, ENABLED	DISABLED	Specifies NRZMODE for DDR3_MEM mode for the DQSBUFD primitive.
OSC_DIV	Integer	1, 2, 4, 8, 16, 32, 64, 128, 256	1	Used for the Soft Error Detect (SED). As an attribute for the internal oscillator, OSC_DIV specifies the divisor of the CCLK frequency to be used in the SED module or corresponding primitives.
PHASE_CNTL	String	DYNAMIC, STATIC	STATIC	Specifies the Phase Adjustment Select mode. When this is set to DYNAMIC, the Phase Adjustment Select control switches between Dynamic and Static depending upon the input logic of the DPAMODE pin. If the attribute is set to STATIC, Dynamic Phase Adjustment Select inputs are ignored.
PHASE_DELAY_CNTL	String	DYNAMIC, STATIC	STATIC	Specifies the CLKOS Phase and Duty Control/Duty Trimming mode. Attached to a PLL-type primitive.
PHASE_SHIFT	Integer	45, 57, 68, 79, 90, 101, 112, 123, 135	90	Phase shift value used. This is required for Simulation. DRC Check: This should match the PHASE_SHIFT value set in the DDRDLLA.

Attribute	Type	Allowed Values	Default	Description
PHASEADJ	Real	Vary	0	Specifies the Coarse Phase Shift setting. Attached to a PLL-type primitive (such as EHXPLLB).
PLLCAP	String	ENABLED, DISABLED, AUTO	DISABLED	Enables or disables the external capacitor pin. Attached to a PLL-type primitive. This attribute value will be usedand updated by MPAR. MPAR converts AUTO to DISABLED or ENABLED as per the placement. This attribute has no impact on the simulation.
PLLTYPE	String	AUTO, SPLL, GPLL	AUTO	Specifies the PLL configuration mode. Applicable only for the EPLLD primitive. This attribute value will be used by MPAR. It has no impact on simulation or bit generation.
POROFF	Boolean	TRUE, FALSE	FALSE	Turns on or off POR when in standby.
REG_*<RegisterType>*_*<RegisterName>*	String	Vary	Vary	This attribute applies to DSP block multipliers to enable various registers, such as Input Registers, Pipeline Registers, Output Registers, Signed Registers, Signed Pipeline Registers, and Accumulator Load Pipeline Registers. Attribute value is the register name. For clocks, you can also assign “NONE” to the attribute. Refer to appropriate DSP User Guide on the Lattice Web site for more details.
REGMODE	String	NOREG, OUTREG	NOREG	Specifies the register mode for pipelining.
REGSET	String	SET, RESET	RESET	Sets the output to either SET or RESET for input and output DDR and shift register elements.
RESETMODE	String	ASYNC, SYNC	Vary	Specifies the reset type. This attribute is attached to a block RAM-type primitive that has a memory size smaller than 9 bits. When set to SYNC, the memory reset is synchronized with the clock. When set to ASYNC, the memory reset is asynchronous to the clock.
RNDPAT	Hexadecimal	Any 14-bit hex value	All zeros	Specifies the Rounding Pattern. Attached to the ALU54A primitive.
RST_PULSE	Integer	Integer value	1	Specifies the required reset pulse length. Attached to the Power Up Reset (PUR) primitive.

Attribute	Type	Allowed Values	Default	Description
SCLKLATENCY	Integer	1, 2 (on the top only 1 is valid)	1	Adjusts SCLK latency. For simulation only.
SED_CLK_FREQ	String	2.08, 2.15, 2.22, 2.29, 2.38, 2.46, 2.56, 2.66, 2.77, 2.89, 3.02, 3.17, 3.33, 3.5, 3.69, 3.91, 4.16, 4.29, 4.43, 4.59, 4.75, 4.93, 5.12, 5.32, 5.54, 5.78, 6.05, 6.33, 6.65, 7, 7.39, 7.82, 8.31, 8.58, 8.87, 9.17, 9.5, 9.85, 10.23, 10.64, 11.08, 11.57, 12.09, 12.67, 13.3, 14, 14.78, 15.65, 16.63, 17.73, 19, 20.46, 22.17, 24.18, 26.6, 29.56, 33.25, 38, 44.33, 53.2, 66.5, 88.67, 133	3.5	Specifies the SED clock frequency.
SHIFT_IN	Boolean	TRUE, FALSE	FALSE	Specifies shift for data. Attached to a DSP multiplier primitive.
SMI_OFFSET	Hexadecimal	Hex value	0x410, 12'h410	Specifies Serial Management Interface offset. Attached to a PLL or DLL primitive.
STDBYOPT	String	USER, CFG, USER_CFG	USER_CFG	Specifies the entry option for entry signals.
TAG_INITIALIZATION	Boolean	ENABLED, DISABLED	DISABLED	Attached to the SSPIA primitive. When this attribute is set to DISABLED, the TAG configuration will be generated without an initialization file, and TAG_INITVAL_* will be all zeros. When this is set to ENABLED, the TAG configuration will be generated with an initialization file, and TAG_INITVAL_* will contain the initialization data. Any un-initialized byte word will default to 00000000 (software default).
TAG_INITSIZE	Integer	448, 632, 768, 2184, 2488, 2640, 3384, 3608	2184	Specifies the TAG Memory size. Attached to the SSPIA primitive.

Attribute	Type	Allowed Values	Default	Description
TAG_INITVAL	Hexadecimal	Any 80-bit hex value	All zeros	Specifies the TAG initialization value. The 80-bit hex string corresponds to the 320 TAG bits. Attached to the SSPIA primitive.
TIMEOUT	String	BYPASS, USER, COUNTER	BYPASS	Specifies the stop to standby delay.
UPDT	String	POS, NEG	POS	Attached to a DDR-type primitive (such as ODDRX4A), the UPDT attribute takes the POS and NEG options, which allows you to update block output.
WAKE_ON_LOCK	Boolean	ON, OFF	ON, OFF	This is a legacy attribute and not supported for new configurations. Attached to a PLL-type primitive.
WAKEUP	String	USER, CFG, USER_CFG	USER_CFG	Specifies the wake option for wake signals. There are three options. ▶ USER: In this case, MAP checks for the connection to the USERSTDBY pin only. If the pin is not driven by a signal that can be toggled, MAP issues error with DRC for this case only. ▶ CFG: In this case, MAP checks for JTAG, I2C, and SLAVE_SPI. If all are disabled, MAP errors out with config mode error. ▶ USER_CFG: In this case, MAP checks the USERSTDBY pin connection and JTAG, I2C, and SLAVE_SPI. Map errors out only when the USERSTDBY pin is not driven by live signal and all settings are disabled.
WRITEMODE	String	NORMAL, WRITETHROUGH, READBEFORE	NORMAL	Specifies Read/Write mode. Attached to a dual- and single-port RAM primitive. WRITEMODE_A and WRITEMODE_B are used for dual-port RAM primitives and refer to the A and B ports.