Lattice Radiant Software Guide for Diamond MachXO3LF to Radiant MachXO4 Device

December 11, 2025

Copyright

Trademarks

Disclaimers

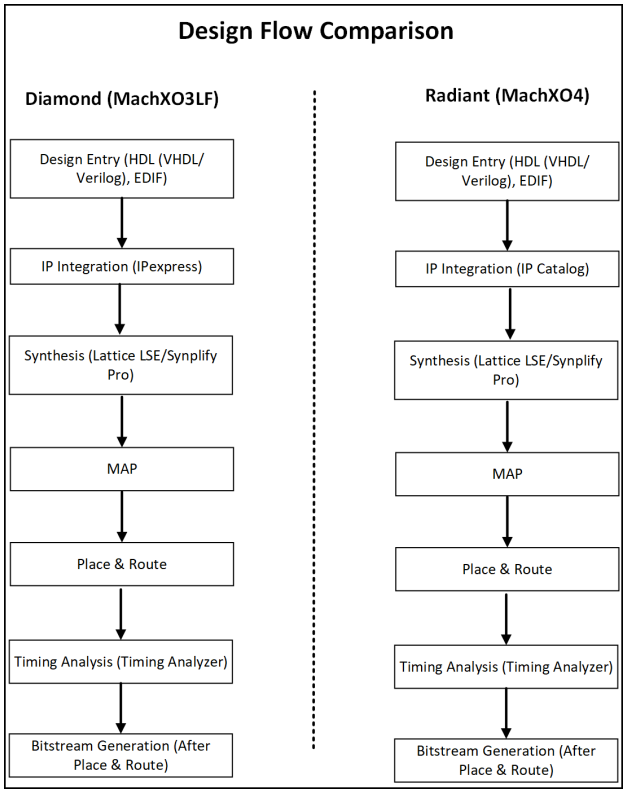
Type Conventions Used in This Document

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

Contents

Migrating Designs from Diamond MachXO3LF to MachXO4 Device on Radiant

When migrating MachXO3LF to MachXO4 device on the Radiant™ software, you should expect to go through the normal design process, such as design entry, design analysis, debug, simulation, and testing. While Diamond and the Radiant software are very similar, there are substantial differences. This chapter provides tips on how to rebuild your design using the Radiant MachXO4 IP, PMI, primitives, and constraints.



This diagram compares the design flows for Lattice Diamond (MachXO3LF) and Lattice Radiant (MachXO4). Both flows follow similar steps—starting with design entry and ending with bitstream generation—but differ in toolsets and formats. Diamond uses IPexpress and .lpf constraints, while Radiant uses the IP Catalog and .sdc constraints.

IP and Modules

IP are basic, configurable modules that provide a variety of functions including I/O, arithmetic, memory, and more. The Radiant IP Catalog works similarly to the IP configuration in Diamond's IPexpress. IP Catalog offers a collection of IP that are similar to those found in IPexpress.

In the Radiant software, signal names of the generated components have been converted to lower case and "_i," "_o," and "_io" suffixes added. Some signals have been renamed. For example: DataA_Re to data_re_i and Cout to overflow_o.

For differences in specific IP, see:

- "Architecture IP" on page 9
- "Arithmetic IP" on page 13
- "Memory IP" on page 20

For more information on the IP, see the Radiant Help under **References > Lattice Module Reference Guide**.

PMI

PMI (Parameterized Module Instantiation) is an alternate way to use some of the components that come with IP Catalog. Instead of using IP Catalog, PMI can directly instantiate a component into your HDL and customize it by setting parameters in the HDL. The Radiant software has a collection of PMI similar to Diamond's. To help you with PMI, templates for instantiating the modules are available in the Radiant Source Template view, which is similar to the Diamond Template Editor. See "PMI" on page 23.

Primitives

Lattice library primitives are very basic functions, such as logic gates and flip-flops. Usually primitives are simply inferred in synthesis, but they can be directly instantiated as HDL into designs. See "Primitives" on page 26.

Preferences and Constraints

Constraints are instructions applied to design elements that guide the design toward desired results and performance goals. The most common constraints are those for timing and pin assignments, but constraints are also available for placement, routing, and many other functions.

In Lattice Diamond, a Logical Preference File (.lpf) is used to constrain a design. In the Radiant software, preferences have been replaced by the

industry standard Synopsys Design Constraints for ease of use and improved compatibility with third-party vendor tools such as Synopsys Synplify Pro.

This is one of the bigger differences between Diamond and Radiant designs. See “Constraints” on page 28.

Architecture IP

- This section shows the feature compatibility of MachXO3LF and MachXO4 IPs.

Embedded Functional Block (EFB)

Table 1: Feature Compatibility (EFB)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
I2C	Yes	Yes (Primary I2C, Secondary I2C)
SPI	Yes	No
Timer/Counter	Yes	No
Timer/Counter Use	Yes	No
PLL (Dynamic Access)	Yes	No
User Flash Memory	Yes	Yes
Wishbone	Yes	Yes
Wishbone Clock Frequency	Yes	Yes

If the Primary or Secondary I2C is enabled, the following options will be available.

Table 2: Feature Compatibility

Feature (I2C)	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
General Call Enable	Yes	Yes
Wakeup Enable	Yes	No
I2C Bus Performance	Yes	Yes
Clock Prescale Value	Yes	Yes
I2C Addressing	Yes	Yes
Slave Address	Yes	Yes (I2C Device Address)
Primary I2C Device Address	No	Yes

PLL

Table 3: Feature Compatibility (PLL)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
General		
Configuration	Yes	Yes (Configuration Mode)
VCO Frequency	Yes	Yes
Estimate Bandwidth	Yes	Yes
Reference Clock (CLKI)		
Frequency	Yes	Yes (CLKI: Frequency (MHz))
Divider	Yes	Yes (CLKI: Divider Value)
Feedback		
FBK Mode	Yes (CLKOP, CLKOS, CLKOS2, CLKOS3, INT_OP, INT_OS, INT_OS2, INT_OS3, UserClock)	Yes (CLKFB: Feedback Mode – CLKOP, CLKOP_INT, CLKOS, CLKOS_INT, CLKOS2, CLKOS2_INT, CLKOS3, CLKOS3_INT, USERCLOCK)
Divider	Yes	Yes (CLKFB: Divider Value)
Fractional-N Divider	Yes	Yes (CLKFB: Feedback Fractional-N Divider Value)
Enable	Yes	Yes (CLKFB: Feedback Fractional-N Divider Enable)
CLKOP		
Bypass	Yes	Yes (CLKOP: Bypass)
Divider	Yes	Yes (CLKOP: Clock Divider Enable)
Desired Frequency	Yes	Yes (CLKOP: Frequency Desired Value)
Tolerance	Yes	Yes (CLKOP: Tolerance)
Divider	Yes	Yes (CLKOP: Divider Value)
Actual Frequency	Yes	Yes (CLKOP: Frequency Actual Value)
Static Phase Shift -degrees	Yes	Yes (CLKOP: Static Phase Shift)
Duty Trim Options	Yes	Yes (CLKOP: Duty Trim Enable) Yes (CLKOP: Duty Trim Options Mode) Yes (CLKOP: Duty Trim Options Delay Multiplier)
CLKOS		
Enable	Yes	Yes (CLKOS: Enable)
Bypass	Yes	Yes (CLKOS: Bypass)

Table 3: Feature Compatibility (PLL) (Continued)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Clock Divider	Yes	Yes (CLKOS: Clock Divider Enable)
Desired Frequency	Yes	Yes (CLKOS: Frequency Desired Value)
Tolerance	Yes	Yes (CLKOS: Tolerance)
Divider	Yes	Yes (CLKOS: Divider Value)
Actual Frequency	Yes	Yes (CLKOS: Frequency Actual Value)
Static Phase Shift -degrees	Yes	Yes (CLKOS: Static Phase Shift)
Duty Trim Options	Yes	Yes (CLKOP: Duty Trim Enable) Yes (CLKOP: Duty Trim Options Mode) Yes (CLKOP: Duty Trim Options Delay Multiplier)
CLKOS2		
Enable	Yes	Yes (CLKOS2: Enable)
Bypass	Yes	Yes (CLKOS2: Bypass)
Clock Divider	Yes	Yes (CLKOS2: Clock Divider Enable)
Desired Frequency	Yes	Yes (CLKOS2: Frequency Desired Value)
Tolerance	Yes	Yes (CLKOS2: Tolerance)
Divider	Yes	Yes (CLKOS2: Divider Value)
Actual Frequency	Yes	Yes (CLKOS2: Frequency Actual Value)
Static Phase Shift -degrees	Yes	Yes (CLKOS2: Static Phase Shift)
CLKOS3		
Enable	Yes	Yes (CLKOS3: Enable)
Bypass	Yes	Yes (CLKOS3: Bypass)
Clock Divider	Yes	Yes (CLKOS3: Clock Divider Enable)
Desired Frequency	Yes	Yes (CLKOS3: Frequency Desired Value)
Tolerance	Yes	Yes (CLKOS3: Tolerance)
Divider	Yes	Yes (CLKOS3: Divider Value)
Actual Frequency	Yes	Yes (CLKOS3: Frequency Actual Value)
Static Phase Shift -degrees	Yes	Yes (CLKOS3: Static Phase Shift)

Table 3: Feature Compatibility (PLL) (Continued)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Optional Port		
Dynamic Phase Ports	Yes	Yes
Clock Enable Ports	Yes	Yes
Standby Port	Yes	Yes
Enable Clock Select	Yes	Yes
PLL Reset Options		
Provide PLL Reset	Yes	Yes
Provide PLLM Reset	Yes	Yes
PLL CLKOS2 Reset	Yes	Yes
PLL CLKOS3Reset	Yes	Yes
Lock Settings		
Provide PLL Lock Signal	Yes	Yes
PLL Lock is Sticky		
Wishbone BUS		
Provide WB Ports Requires instantiation of EFB block	Yes	Yes

Arithmetic IP

The Radiant software has IP similar to all of Diamond's arithmetic modules except for fft_butterfly. The arithmetic IP of Diamond and the Radiant software are very similar except for a couple of differences:

- Data input widths often have a larger range. In Diamond, input widths are sometimes no more than 32 bits. In Radiant, input widths can be up to 64 bits.
- The Bus Ordering Style option is not available in the Radiant software.
  - All Radiant Arithmetic Foundation IPs use fixed MSB:LSB bus ordering, except for the Barrel Shifter Foundation IP, which supports configurable bus ordering (MSB:LSB or LSB:MSB).

Adder

Table 4: Feature Compatibility (Adder)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Specify the Data Width of the Adder	Yes	Yes
Specify the Representation of the Adder	Yes	Yes
Complex Inputs	Yes	Yes
Use Carry-in port	Yes	Yes
Specify the Carry-out Port	Yes	Yes
Enable Output Register	Yes	Yes
Specify number of pipeline stages	Yes	Yes
Bus Ordering Style	Yes	No

Adder Subtractor

Table 5: Feature Compatibility (Adder Subtractor)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Specify the Data Width of the Adder_Subtractor	Yes	Yes
Specify the Representation of the Adder_Subtractor	Yes	Yes
Complex Inputs	Yes	Yes

Table 5: Feature Compatibility (Adder Subtractor) (Continued)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Use Carry-in port	Yes	Yes
Specify the Carry-out Port	Yes	Yes
Enable Output Register	Yes	Yes
Specify number of pipeline stages	Yes	Yes
Bus Ordering Style	Yes	No

Comparator

Table 6: Feature Compatibility (Comparator)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Specify the data width of the comparator	Yes	Yes
Specify the representation of comparator	Yes	Yes
Specify the output port compare function	Yes	Yes
Use LUT based implementation (to use lesser resources)	Yes	No
Enable Output Register	Yes	Yes
Specify number of pipeline stages	Yes	Yes
Bus Ordering Style	Yes	No

Complex Multiplier

Table 7: Feature Compatibility (Complex Multiplier)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Block Implementation	Yes	Yes
Input A Width	Yes	Yes
Input B Width	Yes	Yes
Representation	Yes	Yes

Table 7: Feature Compatibility (Continued)(Complex Multiplier)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Specify the Number of Pipeline Stages	Yes	Yes
Enable Input Registers	Yes	Yes
Enable Output Registers	Yes	Yes
Implementation	Yes	No
Bus Ordering Style	Yes	No

Counter

Table 8: Feature Compatibility (Counter)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Specify the data width of the counter	Yes	Yes
Specify the direction of the counter	Yes	Yes
Optimized for speed	Yes	No
Lower count value	Yes	Yes
Upper count value	Yes	Yes
Enable load input	Yes	Yes
Bus Ordering Style	Yes	No

LFSR

Table 9: Feature Compatibility (LSFR)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
LFSR Type	Yes	Yes
Gate Type	Yes	Yes
Number of Bits	Yes	Yes
Feedback Polynomial	Yes	Yes
Initial Value	Yes	Yes
Enable Parallel Output	Yes	Yes

Table 9: Feature Compatibility (Continued)(LSFR)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Use Reloadable Seed Values	Yes	Yes
Bus Ordering Style	Yes	No

Multiply Accumulate

Table 10: Feature Compatibility (Multiply Accumulate)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Block Implementation	Yes	Yes
Add/Sub Operation	Yes	Yes
Input A Width	Yes	Yes
Representation	Yes	Yes
Input B Width	Yes	Yes
Representation	Yes	Yes
Accumulator Width	Yes	Yes
Input A Signed	No	Yes
Input B Signed	No	Yes
Specify the Number of Pipeline Stages	Yes	Yes
Enable Input Registers	Yes	No
Enable Output Registers	Yes	No
Bus Ordering Style	Yes	No
Result Width	No	Yes
I -> O Clocks	No	Yes

Multiply Add Subtract

Table 11: Feature Compatibility

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Block Implementation	Yes	Yes
Add/Sub Operation	Yes	Yes
Input A0/A1 Width	Yes	Yes

Table 11: Feature Compatibility (Continued)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Representation	Yes	Yes
Input B0/B1 Width	Yes	Yes
Representation	Yes	Yes
Specify the Number of Pipeline Stages	Yes	Yes
Enable Input Registers	Yes	No
Enable Output Registers	Yes	No
Bus Ordering Style	Yes	No
Result Width	No	Yes
I -> O Clocks	No	Yes

Multiply Add Subtract Sum

Table 12: Feature Compatibility (Multiply Add Subtract Sum)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Block Implementation	Yes	Yes
Add/Sub 0 Operation	Yes	Yes
Add/Sub 1 Operation	Yes	Yes
Input A0/A1/A2/A3 Width	Yes	Yes
Representation	Yes	Yes
Input B0/B1/B2/B3 Width	Yes	Yes
Representation	Yes	Yes
Product Bit Width	Yes	Yes
Specify the Number of Pipeline Stages	Yes	Yes
Enable Input Registers	Yes	Yes
Enable Output Registers	Yes	Yes
Bus Ordering Style	Yes	No
Result Width	No	Yes
I -> O Clocks	No	Yes

Multiplier

Table 13: Feature Compatibility (Multiplier)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Use a Constant Coefficient	Yes	Yes
Constant Coefficient Value	Yes	Yes
Use RAM Based Multiplier	Yes	No
Block Implementation	Yes	Yes
Input A Width	Yes	Yes
Representation	Yes	Yes
Input B Width	Yes	Yes
Representation	Yes	Yes
Specify the Number of Pipeline Stages	Yes	Yes
Enable Input Registers	Yes	Yes
Enable Output Registers	Yes	Yes
Bus Ordering Style	Yes	No
Result Width	No	Yes
I -> O Clocks	No	Yes

Subtract

Table 14: Feature Compatibility (Subtract)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Specify the Data Width of the Subtractor	Yes	Yes
Specify the Representation of the Subtractor	Yes	Yes
Complex Inputs	Yes	Yes
Use Carry-in port	Yes	Yes
Specify the Carry-out Port	Yes	Yes
Enable Output Register	Yes	Yes
Specify number of pipeline stages	Yes	Yes
Bus Ordering Style	Yes	No

Table 14: Feature Compatibility (Subtract) (Continued)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Result Width	No	Yes
I -> O Clocks	No	Yes

Memory IP

The types of memory available in the Radiant software are very similar to what is available in Diamond. All the IP available in Diamond have equivalents in the Radiant IP Catalog. Design differences are just in the names and a few options.

If a memory initialization file is needed, create one before configuring the IP. Each row includes the value to be stored in a particular memory location. The file name for the memory initialization file is *.mem (<*file_name*>.mem).

RAM DP

Table 15: Feature Compatibility (RAM DP)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Read Port		
Address Depth	Yes	Yes[1]
Data Width	Yes	Yes[1]
Write Port		
Address Depth	Yes	Yes[1]
Data Width	Yes	Yes[1]
Total Memory bits	Yes	Yes
Byte Enable	Yes	Yes
Byte Size	Yes	Yes
Enable Output Register	Yes	Yes
Enable Output ClockEn	Yes	Yes
Optimization	Yes (Area or Speed)	No[2]
Reset Mode	Yes	Yes
Release	Yes	No
Initialization	Yes	Yes
Memory File Format	Yes	Yes
Enable ECC (not supported for Data Width > 64)	Yes	No
Bus Ordering Style	Yes	No

Note:

1. As of this release, the Address Depth and Data Width must be the same for both the Read and Write ports. A legality check has been implemented to ensure this condition is always met.

2. Although there is no selectable option, the configuration is internally optimized for Speed.

Table 16: RAM_DP Port Mapping

Diamond Port Name	Radiant Port Name
WrAddress	wr_addr_i
RdAddress	rd_addr_i
Data	wr_data_i
ByteEn	ben_i
WE	wr_en_i
RdClock	rd_clk_i
RdClockEn	rd_clk_en_i
ORdClockEn	rd_out_clk_en_i
Reset	rst_i
WrClock	wr_clk_i
WrClockEn	wr_clk_en_i
Q	rd_data_o

RAM DQ

.

Table 17: Feature Compatibility (RAM DQ)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Address Depth	Yes	Yes
Data Width	Yes	Yes
Total Memory bits	Yes	Yes
Byte Enable	Yes	Yes

Table 17: Feature Compatibility (RAM DQ) (Continued)

Feature	Supported in MachXO3LF (Diamond)	Supported in MachXO4 (Radiant 2025.1)
Byte Size	Yes	Yes
Enable Output Register	Yes	Yes
Enable Output ClockEn	Yes	Yes
Optimization	Yes (Area or Speed)	No[1]
Reset Mode	Yes	Yes
Release	Yes	No
Initialization	Yes	Yes
Memory File Format	Yes	Yes
Enable ECC (not supported for Data Width > 64)	Yes	No
Bus Ordering Style	Yes	No
Memory Write/Read Mode	Yes (Normal, Write Through, Read before write)	Yes (Normal, Write Through, Read before write)

Note:

1. Although there is no selectable option, the configuration is internally optimized for Speed.

Table 18: RAM_DQ Port Mapping

Diamond Port Name	Radiant Port Name
Address	addr_i
ByteEn	ben_i
ClockEn	clk_en_i
clock	clk_i
OClockEn	rd_out_clk_en_i
Reset	rst_i
Data	wr_data_i
WE	wr_en_i
Q	rd_data_o

PMI

This section shows changes to MachXO4 PMI as compared to MachXO3LF PMI.

Table 19: Changes to PMI for MachXO4 and MachXO3LF

PMI Module Name	Corresponding Foundation IP	Available in Diamond MachXO3LF	Available in Radiant 2025.1 MachXO4	Difference of Diamond IP with Radiant IP
pmi_add	Adder	Yes (has an extra pmi_result_width parameter)	Yes	Does not support single bit.
pmi_addsub	Adder Subtractor	Yes	Yes	N/A
pmi_complex_mult	Complex Mult	Yes	Yes	Does not support 3 or 4 multiplier selection, but does support larger input width and number of pipeline stage settings.
pmi_constant_mult	N/A	Yes	No	N/A
pmi_counter	Counter	Yes	Yes	Does not support "optimize for speed setting" with high data width.
pmi_distributed_dpram	Distributed DPRAM	Yes	Yes	Supports higher data width and address depth, additional settings like rset assertion synchronization.
pmi_distributed_rom	Distributed ROM	Yes	Yes	Supports higher data width and address depth, additional settings like rset assertion synchronization.
pmi_distributed_shift_reg	Shift Register	Yes	Yes	N/A
pmi_distributed_spram	Distributed SPRAM	Yes	Yes	Supports additional settings like reset assertion synchronization.
pmi_fifo_dc	FIFO DC	Yes	No	N/A
pmi_mac	Mult Accumulate	Yes	Yes	Supports larger input data widths.
pmi_mult	Multiplier	Yes	Yes	Does not support RAM based multiplier, supports larger data widths.
pmi_multaddsub	Mult Add Sub	Yes	Yes	Supports larger input data widths.
pmi_multaddsubsum	Mult Add Sub Sum	Yes	Yes	Supports larger input data widths.
pmi_pll	N/A	Yes	No	N/A
pmi_pll_fp	N/A	Yes	No	N/A

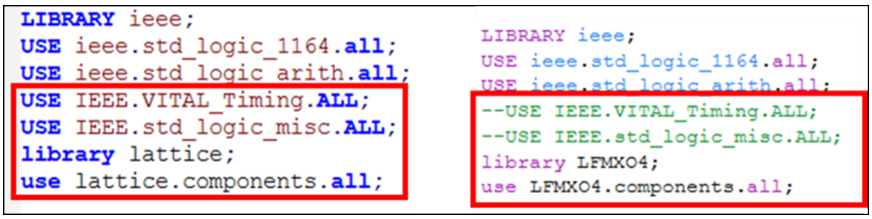
Table 19: Changes to PMI for MachXO4 and MachXO3LF (Continued)

PMI Module Name	Corresponding Foundation IP	Available in Diamond MachXO3LF	Available in Radiant 2025.1 MachXO4	Difference of Diamond IP with Radiant IP
pmi_ram_dp	RAM DP	Yes	Yes	Supports higher data width and address depth, additional settings like reset assertion synchronization, byte enable, and output clockEn buffer for read port. It has a separate read and write clocks.
pmi_ram_dp_be	RAM DP	Yes	Yes	Supports higher data width and address depth, additional settings like reset assertion synchronization, byte enable, and output clockEn buffer for read port. It has a separate read and write clocks.
pmi_ram_dp_true	RAM DP True	Yes	No	N/A
pmi_ram_dp_true_be	N/A	Yes	No	N/A
pmi_ram_dq	RAM DQ	Yes	Yes	Missing area vs speed optimization, UFM storage initialization, write mode, and enable ECC settings.
pmi_ram_dq_be	RAM DQ	Yes	Yes	Missing area vs speed optimization, UFM storage initialization, write mode, and enable ECC settings.
pmi_rom	ROM	Yes	No	N/A
pmi_sub	Subtractor	Yes	Yes	Does not support single bit.

VHDL Support

This section gives a general overview of how VHDL designs are supported during migration to the MachXO4 family.

1. Replace all references to “machxo2”, “machxo3”, “lattice”, and similar VHDL libraries with “lfmxo4”, as shown in the example below.



```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE IEEE.VITAL_Timing.ALL;
USE IEEE.std_logic_misc.ALL;
library lattice;
use lattice.components.all;
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
--USE IEEE.VITAL_Timing.ALL;
--USE IEEE.std_logic_misc.ALL;
library LFMXO4;
use LFMXO4.components.all;
```

2. Ensure that the correct VHDL library name is set for their source files. While the default is usually “work”, some original Diamond projects used custom library names that also need to be updated in Radiant.

Name	Value
VHDL Library Name	work
File Type	VHDL
Include for	Synthesis and Simulation

Primitives

This section shows the changes between MachXO3LF and MachXO4 primitives.

For more information on the Radiant primitives, see the Radiant Help under **References > FPGA Libraries Reference Guide**.

Parameters

The parameter value format for the following primitives has been updated from integer to string. For MachXO4 primitives, the format is now string, whereas MachXO3LF primitives continue to use integer values.

Table 20: Primitives List

Name	MachXO4	MachXO3LF
DP8KC	String	Integer
FIFO8KB	String	Integer
PDPW8KC	String	Integer
SP8KC	String	Integer
ROM128X1A	String	Integer
ROM16X1A	String	Integer
ROM256X1A	String	Integer
ROM32X1A	String	Integer
ROM64X1A	String	Integer
EHXPLLJ	String	Integer
LUT4	String	Integer
LUT5	String	Integer
LUT6	String	Integer
LUT7	String	Integer
LUT8	String	Integer
DQSDLLC	String	Integer
CCU2D	String	Integer
EFB	String	Integer

Primitive Names

The names of the following MachXO3LF primitives have been changed in MachXO4.

Table 21: Primitives

MachXO3LF in Diamond	MachXO4 in Radiant
DCCA	DCCB
ECLKSYNCA	ECLKSYNCB
PLLREFCS	PLLREFCSB
DELAYD	DELAYG
DELAYE	DELAYH
GSR	GSRB
PFMUX	SLICEMUX
SGSR	SGSRA

EFB and SEDFA

DEV_DENSITY port values of EFB and SEDFA primitives have been updated for the MachXO4 device. The following table shows how they compare to MachXO3LF.

Table 22: Mapping Table for MachXO3LF and MachXO4

MachXO3LF	MachXO4
640L	/
640L_121P	/
1300L	010
1300L	015
1300L_256P	015_256P
2100L	025
2100L_324P	/
4300L	050
4300L_400P	050_400P
6900L	080
9400L	110

Constraints

One of the more prominent differences between Lattice Diamond software and the Radiant software is the process of constraining a design. In Lattice Diamond, a Logical Preference File (.lpf) was used to constrain a design by tuning all aspects of logical, timing and physical constraints to improve performance.

In the Radiant software, preferences have been replaced by the industry standard Synopsys Design Constraints for ease of use and improved compatibility with third-party vendor tools such as Synopsys Synplify Pro.

Radiant Constraint Tools

In Lattice Diamond, timing and physical preferences are applied using Spreadsheet View, Package View, Netlist View, and Device View.

In the Radiant software:

- Timing constraints are applied using the Timing Constraint Editors (Pre- and Post-Synthesis) or by directly modifying the .sdc file.
- Physical constraints are applied using the Device Constraint Editor (DCE) or by directly modifying the .pdc file.

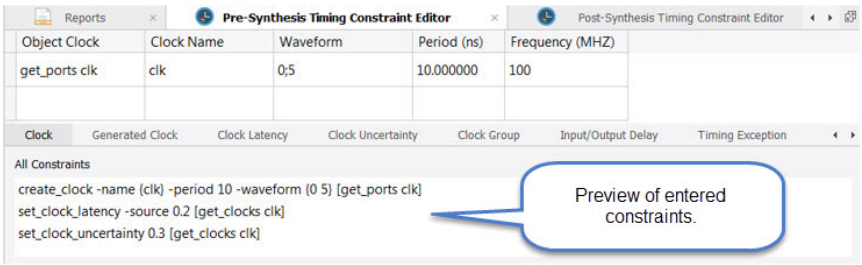
Timing Constraints

Timing constraints are managed in SDC format in both .sdc/.fdc files for Synopsys Synplify Pro.

The new Radiant software tools for pre- and post-synthesis timing constraints are:

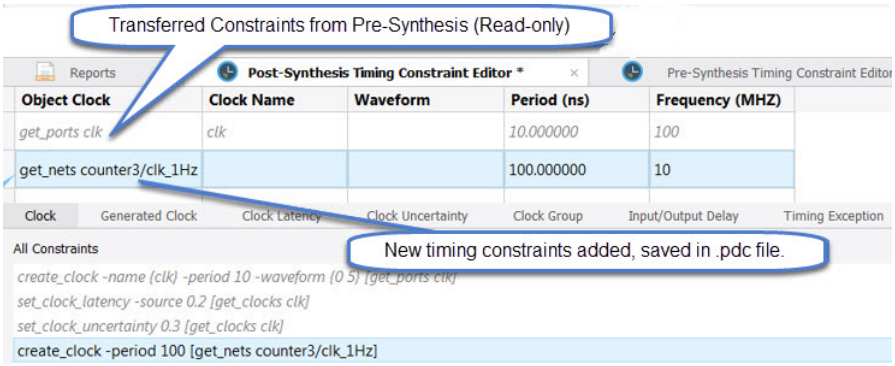
- Pre-Synthesis Timing Constraint Editor (Figure 1) —Reads the HDL designs and helps you to create timing constraints based on HDL signal, port, and object names. The constraints are saved in an .ldc file.

Figure 1: Pre-Synthesis Timing Constraint Editor



- Post-Synthesis Timing Constraint Editor (Figure 2)—Reads the post-synthesis netlists and helps you to create timing constraints based on post-synthesis netlist signal, port, and object names. The constraints are saved in a .pdc file. This flow allows further tuning of timing constraints for the Place & Route process.

Figure 2: Post-Synthesis Timing Constraint Editor



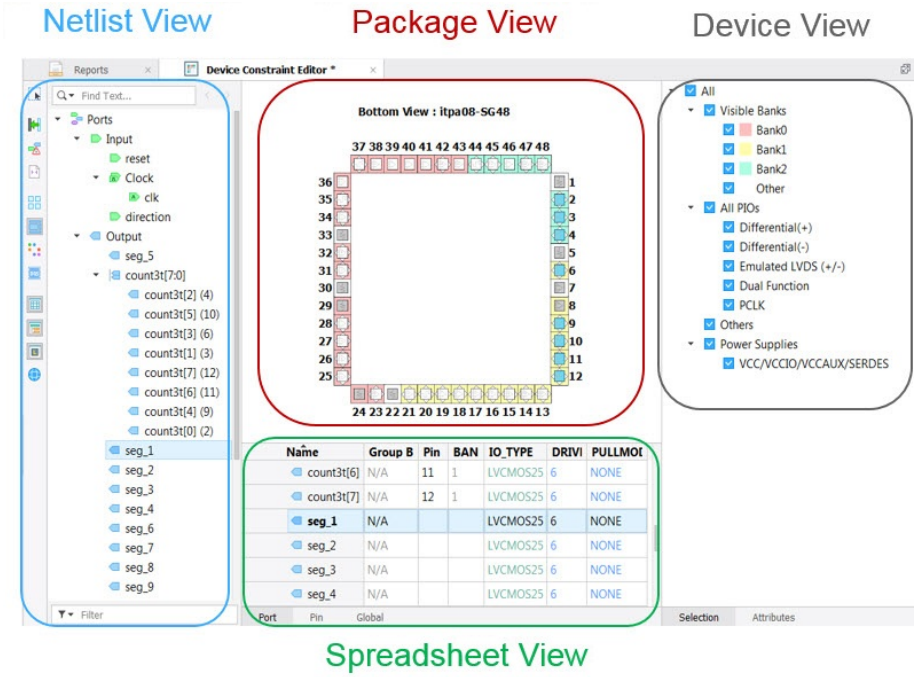
The updated post-synthesis timing constraints override pre-synthesis constraints. This happens only when the same constraint is applied in post-synthesis.

In general, physical constraints are entered in the .pdc file. Alternatively, through a text editor, the physical constraints may also be entered in the .ldc file. Synthesis does not consume them and transfer them to a .pdc file for back-end consumption.

Physical Constraints

The Radiant Device Constraint Editor tool now combines the Netlist, Package, Device, and Spreadsheet views into one GUI for the primary purpose of entering physical (.pdc) constraints. This makes it easier to manage multiple tools and perform such actions as cross probing between multiple views. See Figure 3.

Figure 3: Device Constraint Editor



Each of the views are used to apply constraints such as prohibiting pins and assigning IO_TYPEs.

Note that the Floorplan View is used for creating GROUPs and REGIONs. For more information on Device Constraint Editor, see the Radiant software online Help under **User Guides > Applying Design Constraints > Using Radiant Software Tools > Device Constraint Editor.**

Preferences to Constraints

Table 23 shows the most commonly used Lattice Diamond physical preference keywords and the equivalent Radiant SDC commands to create a physical constraint.

Table 23: Lattice Diamond Preference Keywords Compared to the Radiant SDC Commands

Lattice Diamond Physical Preference	Radiant Software Physical Constraint	Description
Global, Net, and Clock Attributes	ldc_set_attribute	Sets global attributes or specific attributes to the selected object.
UGROUP	ldc_create_group	Defines a user group.
IOBUF	ldc_set_port	Sets constraint attributes to the selected port.
LOCATE	ldc_set_location	Places an object on a site or a user group into a region.

Table 23: Lattice Diamond Preference Keywords Compared to the Radiant SDC Commands (Continued)

Lattice Diamond Physical Preference	Radiant Software Physical Constraint	Description
LOCATE VREF	ldc_create_vref	Defines a voltage reference site.
PROHIBIT	ldc_prohibit	Prohibits use of a site.
REGION	ldc_create_region	Defines a rectangular region.
SYSCONFIG	ldc_set_sysconfig	Sets SysConfig attributes.
VCC_NOMINAL VCC_DERATE VOLTAGE	ldc_set_vcc	Sets a voltage to a bank or derates the core voltage.

All physical constraints and post-synthesis timing constraints are stored in the .pdc file.

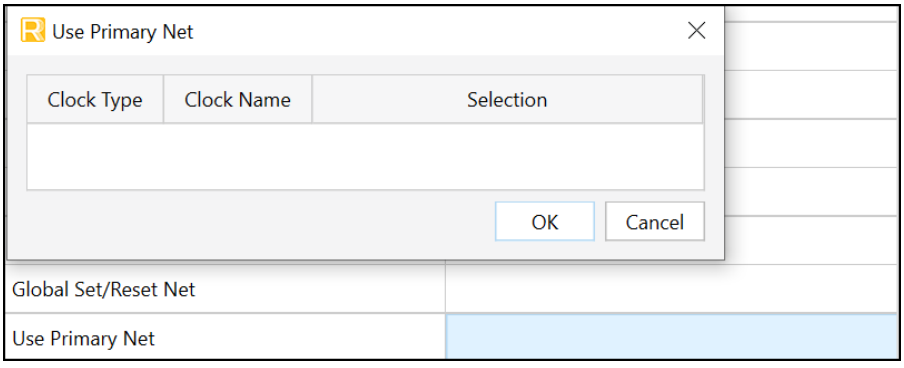
Primary Clock Net Access

In Diamond, primary clocks are specified in a preference file (.lpf) but in the Radiant software, the primary clocks are specified in an .ldc file. In the Radiant software, you can use Device Constraint Editor to set access to the primary clock spine.

To set access to the primary clock spine:

1. Choose **Tools > Device Constraint Editor**.
2. Click the **Global** tab.
3. Double-click on **Use Primary Net**.

   The Use Primary Net dialog box opens.



4. Double-click on the desired clock net.
5. Choose **Primary** or **Prohibit Primary** from the pull-down menu.
6. Click **OK**.

Once the change is saved, the attribute is recorded as shown below:

```
#Set to use Primary clock spine
ldc_set_attribute {USE_PRIMARY=TRUE} [get_nets clk_c]

#Prohibit primary clock spine
ldc_set_attribute {USE_PRIMARY=FALSE} [get_nets clk_c]
```

The MachXO4 device supports new constraints:

1. **Secondary Net**

   MachXO4 supports secondary net, which is not available in other Radiant devices. Two preferences are used in Diamond to use/prohibit secondary net usage for the given net.

   ```
   USE SECONDARY NET "clk_c";
   PROHIBIT SECONDARY NET "clk_c";
   ```

   These preferences are migrated to ldc_set_attribute constraint in Radiant.

   ```
   ldc_set_attribute {USE_SECONDARY=TRUE} [get_nets clk_c]
   ldc_set_attribute {USE_SECONDARY=FALSE} [get_nets clk_c]
   ```

2. **CUSTOM_IDCODE**

   MachXO4 supports CUSTOM_IDCODE, which is not available in other Radiant devices. The CUSTOM_IDCODE preference is used in Diamond to set its format and value.

   ```
   CUSTOM_IDCODE HEX "0000FFFF";
   CUSTOM_IDCODE BIN "00001111000011110000111100001111";
   ```

   These preferences are migrated to ldc_set_attribute constraint in Radiant.

   ```
   ldc_set_attribute {CUSTOM_IDCODE_FORMAT=BIN
   CUSTOM_IDCODE=00001111000011110000111100001111}
   ```

Timing Preferences to Constraints

Table 24 shows the most commonly used Lattice Diamond timing preference keywords and the equivalent Radiant SDC timing constraints.

Table 24: Diamond Timing Preference Keywords Compared to Radiant SDC Timing Constraints

Lattice Diamond Timing Preference	Radiant Software Timing Constraint	Description
BLOCK INTERCLOCK DOMAIN	set_clock_groups	Defines different types of clocking schemes.
BLOCK CLKNET BLOCK PATH CLKSKEWDISABLE	set_false_path	Defines false path cycles.
CLKSKEWDIFF	set_clock_latency	Defines arrival and departure times.
CLOCK_TO_OUT	set_output_delay	Defines output delay relative to a clock.

Table 24: Diamond Timing Preference Keywords Compared to Radiant SDC Timing Constraints (Continued)

Lattice Diamond Timing Preference	Radiant Software Timing Constraint	Description
FREQUENCY/PERIOD	create_clock	Defines the design clocks.
FREQUENCY/PERIOD	create_generated_clock	Defines generated clocks.
INPUT_SETUP	set_input_delay	Defines input delay relative to a clock.
MAX_DELAY	set_max_delay	Defines maximum delay for timing paths.
MAX_DELAY MIN	set_min_delay	Define minimum delay for timing paths.
MULTICYCLE	set_multicycle_path	Defines multicycle clock cycles.
SYSTEM_JITTER CLOCK_JITTER (option)	set_clock_uncertainty	Defines uncertainty delays.

Note

When an INPUT_SETUP or CLOCK_TO_OUT is set in Lattice Diamond without first specifying a clock, Lattice Diamond automatically creates a virtual clock constraint and honors the delay. In the Radiant software, you should first define a clock (create_clock), whether real or virtual, before using the set_input_delay or set_output_delay constraint.

The Radiant software constraints listed in Table 25 require Tcl commands to access object names such as cell, pin, net, port, or clock in a design.

Table 25: Tcl Commands

Object Access Types	Description
all_clocks	Access all clocks in a design.
all_inputs	Access all inputs in a design.
all_outputs	Access all outputs in a design.
get_cells	Access cells in a design.
get_clocks	Access clocks in a design.
get_nets	Access nets in a design.
get_pins	Access pins in a design.
get_ports	Access ports in a design.

The following table lists examples commonly used in Lattice Diamond preferences and the equivalent Radiant software constraints in the SDC format.

Table 26: Examples of Timing Preferences in SDC Format

Lattice Diamond Preference	Radiant Software Constraint
BLOCK PATH FROM PORT “abc” TO CELL “reg1/”;	set_false_path -from [get_ports abc] -to [get_cells reg1]
CLKSKEWDIFF CLKPORT “clk1” CLKPORT “clk2” 2 NS;	set_clock_latency 2 -source [get_clocks clk1]
CLKSKEWDISABLE CLKNET “clk1” CLKNET “clk2”;	set_false_path -from [get_clocks clk1] -to [get_clocks clk2]
CLOCK_TO_OUT PORT “out1” 8 ns CLKPORT=”clk2”;	set_output_delay (x-8) -clock [get_clocks clk2] [get_ports out1] [1, 2]
FREQUENCY (PERIOD) NET “clk1” 100Mhz;	create_clock -period 10 -name clk1 [get_nets clk1]
INPUT_SETUP PORT “in_a” 4 ns CLKNET ”clk1”;	set_input_delay (x-4) -clock [get_clocks clk1] [get_ports_in_a] [1, 2]
MAXDELAY FROM CELL “reg1” TO CELL “reg2” 5 NS	set_max_delay 5 -from [get_cells reg1] -to [get_cells reg2]
MULTICYCLE FROM CLKNET “clk1” TO CLKNET “clk2” 2 X;	set_multicycle_path 2 -from [get_clocks_clk1] -to [get_clocks clk2] [2]
SYSTEM_JITTER 1.0 NS	set_clock_uncertainty 1 [get_clocks *]

1. x = clock period
2. Set create_clock first.

For more information on the details of SDC constraints, see the Radiant Help under **Reference Guides > Constraints Reference Guide > Lattice Synthesis Engine Constraints > Synopsys Design Constraints.**

Attributes Compared

Synthesis attributes are mostly the same in both Lattice Diamond and the Radiant software. The Radiant software does not use some attributes because they are only for Diamond’s preference method. Also, some attributes are for devices not supported in the Radiant software. Any attributes not in Table 27 (below) that were in Lattice Diamond are obsolete and are not used in the Radiant software.

Table 27: Diamond versus Radiant Attributes

Lattice Diamond Software	Radiant Software
BBOX	BBOX
CLAMP	CLAMP
DIFFRESISTOR	DIFFRESISTOR
DRIVE	DRIVE

Table 27: Diamond versus Radiant Attributes (Continued)

Lattice Diamond Software	Radiant Software
GLITCHFILTER	GLITCHFILTER
GSR	GSR
HGROUP	Replaced by GRP
HYSTERSIS	HYSTERSIS
INIT	INIT
IO_TYPE	IO_TYPE
LOC	LOC
NOCLIP	NOCLIP
NOMERGE(SAVE)	NOMERGE
OPENDRAIN	OPENDRAIN
PULLMODE	PULLMODE
RBBOX	RBBOX
REGION	REGION
SLEWRATE	SLEWRATE
TERMINATION	TERMINATION
UGROUP	Replaced by GRP
USERCODE	USERCODE
VREF	VREF

Revision History

The following table gives the revision history for this document.

Date	Version	Description
12/11/2025	1.0	Initial release.