Avant DSP Arithmetic Modules - Lattice Radiant Software

User Guide

FPGA-IPUG-02183-1.1

December 2023

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
FPGA	Field-Programmable Gate Array
IP	Intellectual Property
RTL	Register Transfer Level

1. Introduction

The IP Catalog provides a variety of modules. These modules cover a variety of common functions and can be customized. They are optimized for Lattice FPGA devices built on the Lattice Avant™ platform. Use these modules to speed up your design work and to produce the most effective results.

This guide describes the DSP Arithmetic modules that comes with the IP Catalog. These modules serve as the building block to realize a more complex or user-defined function. The descriptions mainly cover the ports, attributes and sample functional timing diagrams of the modules.

2. DSP Arithmetic Modules

2.1. Multiplier

A two-input multiplier performs signed/unsigned multiplication of the data from inputs data_a_i and data_b_i. The output result_o carries the Product of the multiplication operation.

The shaded portions of the Multiplier Schematic Diagram, as shown in Figure 2.1, are optional and are removed/ignored in certain configurations.

Function formula:

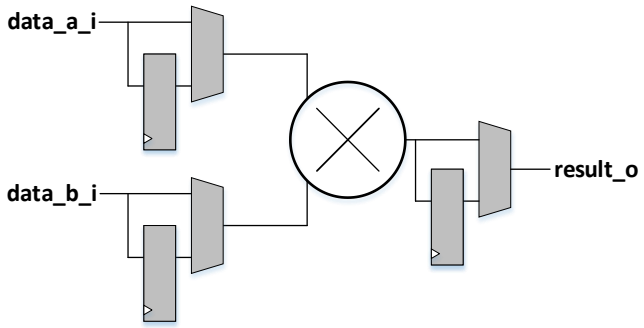

Figure 2.1. Multiplier Schematic Diagram

2.1.1. Multiplier Operation

Table 2.1. Multiplier Operation

Configuration	Operation
Default	result_o = data_a_i*data_b_i

2.1.2. Ports

Table 2.2. Multiplier Ports

Signal Name	Direction	Width (bits)	Description
clk_i	IN	1	This signal connects to the clock pin of the input, and output Unused when neither Input A, Input B or Output is set to Registered
ce_a_i	IN	1	This signal is an active HIGH synchronous clock enable for input A 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Unused when Input A is not set to Registered.
ce_b_i	IN	1	This signal is an active HIGH synchronous clock enable for input B 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Unused when Input B is not set to Registered
ce_o_i	IN	1	This signal is an active HIGH synchronous clock enable for Output 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Unused when Output is not set to Registered
rst_i	IN	1	It is an active HIGH reset. Unused when neigther REGINPUTA, REGINPUTB, or REGOUTPUT is not set to Registered. Reset mode depends on RST_MODE.
data_a_i	IN	A_WIDTH	This signal contains an input for multiplication operation.

Signal Name	Direction	Width (bits)	Description
data_b_i	IN	B_WIDTH	This signal contains an input for multiplication operation.
result_o	OUT	OUT_WIDTH	This signal carries the product value of the multiplication. Core functionality is represented below. data_a_i *data_b_i

2.1.3. Attributes

Table 2.3. Multiplier Attributes

Configuration	Range	Default Value	Description
A_WIDTH	2–72	9	This configuration controls the width of the input data_a.
B_WIDTH	2–72	9	This configuration controls the width of the input data_b.
OUT_WIDTH	4-144	18	This parameter is uneditable and only gets the sum of A_WIDTH and B_WIDTH. InputA Width + InputB Width = Result Width
RST_MODE	SYNC or ASYNC	SYNC	This configuration controls the reset mode, either asynchronous reset or asynchronous reset. When set to synchronous reset, the reset value occurs at the preceding clock edge.
A_SIGNED	Signed, Unsigned	Signed	This configuration controls the sign interpretation of the input data_a_i, either signed or unsigned.
B_SIGNED	Signed, Unsigned	Signed	This configuration controls the sign interpretation of the input data_b_i, either signed or unsigned.
REGINPUTA	on, off	off	This configuration controls the registering of input data_a_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.
REGINPUTB	on, off	off	This configuration controls the registering of the inputs, data_b_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.
REGOUTPUT	on, off	off	This configuration controls the registering of the multiplier result onto the output, result_o. off – Unregistered. Result is directly routed to the output. on – Registered. Result is registered at the output.
Internal Parameters			
A_WDT	—	—	A_WDT = A_WIDTH
B_WDT	—	—	B_WDT = B_WIDTH

2.1.4. Timing Diagram

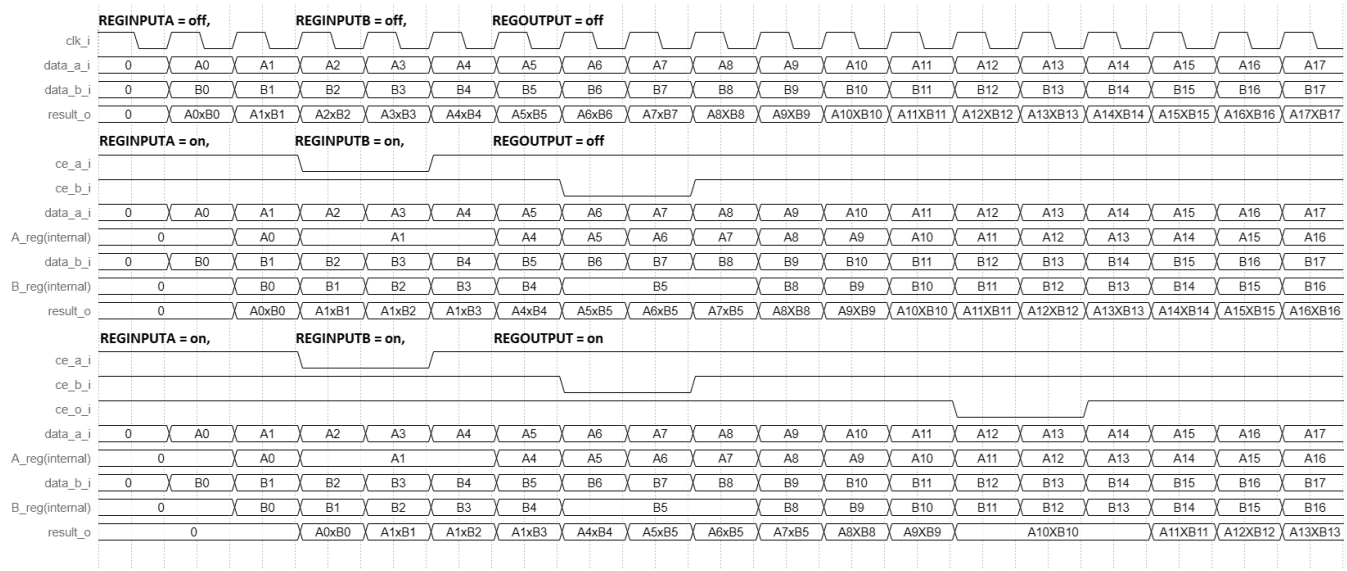

Figure 2.2. Multiplier Timing Diagram

Table 2.4. Multiplier Timing Table

Attribute		Latency from Input to Output
REGINPUT*	REGOUTPUT	
0	0	0
0	1	1
1	0	1
1	1	2

2.2. Multiply-Accumulate

A two-input multiplier with accumulate performs signed/unsigned multiplication of the data from inputs data_a_i and data_b_i and accumulates the result. The output result_o carries the Accumulation of Products.

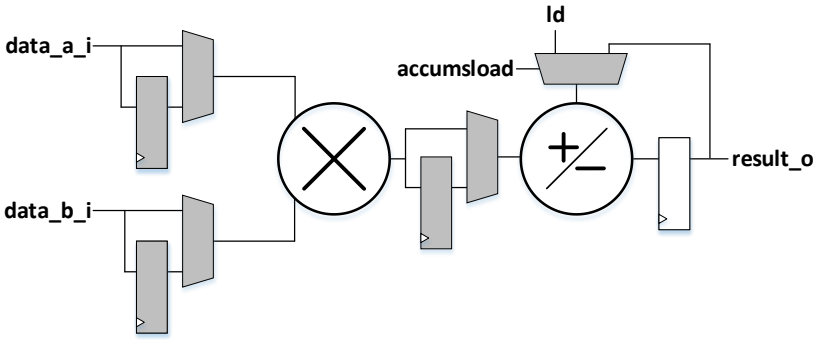

Figure 2.3. Multiply-Accumulate Schematic

2.2.1. Multiply-Accumulate Operation

Table 2.5. Multiply-Accumulate Operations

Operation Dependency	Operation
addnsub_i = 0, accumsload = 0	result_o = result_prev + (data_a_i*data_b_i)
addnsub_i = 0, accumsload = 1	result_o = ld + (data_a_i*data_b_i)
addnsub_i = 1, accumsload = 0	result_o = result_prev - (data_a_i*data_b_i)
addnsub_i = 1, accumsload = 1	result_o = ld - (data_a_i*data_b_i)

2.2.2. Ports

Table 2.6. Multiply-Accumulate Ports

Signal Name	Direction	Width (bits)	Description
clk_i	IN	1	This signal connects to the clock pin of the input, and output.
ce_a_i	IN	1	This signal is an active HIGH synchronous clock enable for input A 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Unused when Input A is not set to Registered.
ce_b_i	IN	1	This signal is an active HIGH synchronous clock enable for input B 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Unused when Input B is not set to Registered
ce_i	IN	1	This signal is an active HIGH synchronous clock enable for Pipeline and Output 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic
rst_i	IN	1	It is an active HIGH reset. Reset mode depends on RST_MODE.
data_a_i	IN	A_WIDTH	This signal contains the multiplicand value of the multiplication.
data_b_i	IN	B_WIDTH	This signal contains the multiplier value of the multiplication.
accumsload	IN	1	This signal destates if the ld input or the previews result_o is added/subtracted from the product of the two multipliers.
ld	IN	ACC_WIDTH	This signal is added or subtracted from the product of the two multiplier when accumsload is high.

Signal Name	Direction	Width (bits)	Description
addnsub_i	IN	1	This signal dictates the operation to be used by the accumulator – *1'b0* for addition and *1'b1* for subtraction.
result_o	OUT	ACC_WIDTH	This signal carries the accumulated value of the product of inputs data_a_i and data_b_i. Core functionality is represented below. $Result = LD + (DataA_i * DataB_i)$; accumsload = 1, addnsub_i = 0 $Result = LD - (DataA_i * DataB_i)$; accumsload = 1, addnsub_i = 1 $Result = Result_{prev} + (DataA_i * DataB_i)$; accumsload = 1, addnsub_i = 0 $Result = Result_{prev} - (DataA_i * DataB_i)$; accumsload = 1, addnsub_i = 1

2.2.3. Attributes

Table 2.7. Multiply-Accumulate Attributes

Configuration	Range	Default Value	Description
A_WIDTH	2–72	9	This configuration controls the width of the input data_a_i.
B_WIDTH	2–72	9	This configuration controls the width of the input data_b_i.
OUT_WIDTH	5-145	19	This configuration controls the accumulator width.
A_SIGNED	Signed, Unsigned	Signed	This configuration controls the sign interpretation of the input data_a_i, either signed or unsigned.
B_SIGNED	Signed, Unsigned	Signed	This configuration controls the sign interpretation of the input data_b_i, either signed or unsigned.
RST_MODE	SYNC or ASYNC	ASYNC	This configuration controls the reset mode, either asynchronous reset or synchronous reset. When set to asynchronous reset, the reset value occurs at the preceding clock edge.
REGINPUTA	on, off	off	This configuration controls the registering of the inputs, data_a_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.
REGINPUTB	on, off	off	This configuration controls the registering of the inputs, data_b_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.
REGPIPE	on, off	off	This configuration controls the insertion of pipeline into the multiplier operation.
REGOUTPUT	on	on	This configuration controls the registering of the result onto the output, result_o. This is not editable. Result is registered at the output.

2.2.4. Timing Diagram

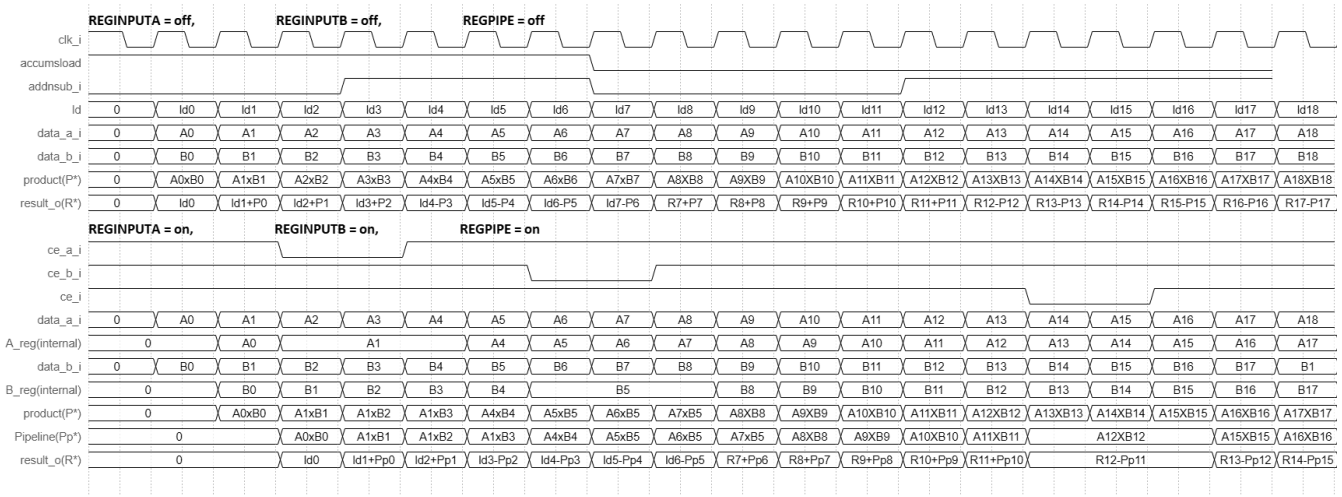

Figure 2.4. Multiply-Accumulate Timing Waveforms

Table 2.8. Multiply-Accumulate Timing Table

Attribute			Latency from Input to Output
REGINPUT*	REGPIPE	REGOUTPUT	
0	0	1	1
0	1	1	2
1	0	1	2
1	1	1	3

2.3. Multiply-Add-Subtract

A pair of two-input multipliers that performs signed/unsigned multiplication of the data from inputs data_a and data_b with addition/subtraction on the product pair. The output result_o carries the Sum/Difference of Products.

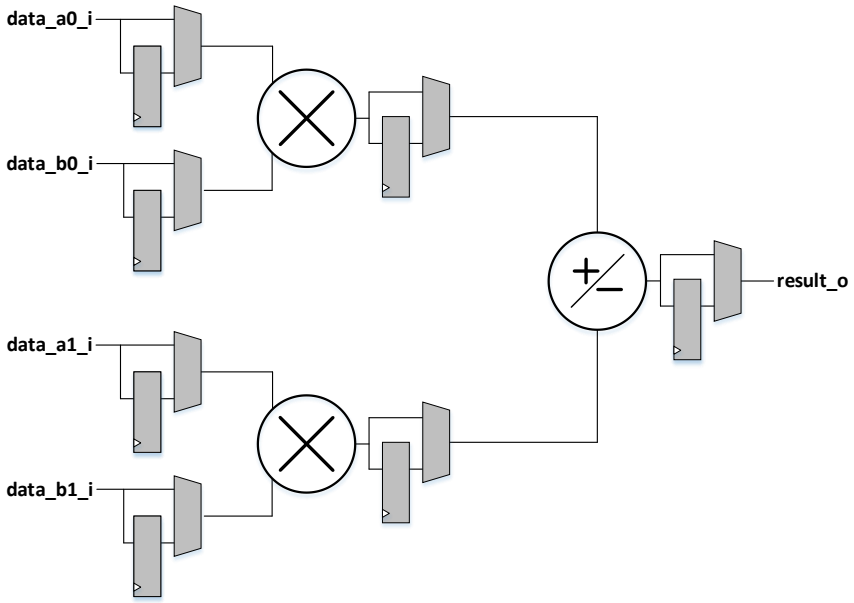

Figure 2.5. Multiply-Add-Subtract Schematic

2.3.1. Multiply-Add-Subtract Operation

Table 2.9. Multiply-Add-Subtract Operations

Operation Dependency	Operation
addnsub_i = 0	result_o = (data_a0_i*data_b0_i) + (data_a1_i *data_b1_i)
addnsub_i = 1	result_o = (data_a0_i*data_b0_i) - (data_a1_i *data_b1_i)

2.3.2. Ports

Table 2.10. Multiply-Add-Subtract Ports

Signal Name	Direction	Width (bits)	Description
clk_i	IN	1	This signal connects to the clock pin of the input, and output Unused when neither Input A, Input B or Output is set to Registered
ce_a_i	IN	1	This signal is an active HIGH synchronous clock enable for input A 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Unused when Input A is not set to Registered.
ce_b_i	IN	1	This signal is an active HIGH synchronous clock enable for input B 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Unused when Input B is not set to Registered
ce_i	IN	1	This signal is an active HIGH synchronous clock enable for Pipeline and Output 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Unused when Output is not set to Registered

Signal Name	Direction	Width (bits)	Description
rst_i	IN	1	It is an active HIGH . Reset mode depends on RST_MODE.
data_a0_i	IN	A_WIDTH	This signal contains the multiplicand value of the multiplication pair data_a0_i and data_b0_i.
data_a1_i	IN	A_WIDTH	This signal contains the multiplicand value of the multiplication pair data_a1_i and data_b1_i.
data_b0_i	IN	B_WIDTH	This signal contains the multiplier value of the multiplication pair data_a0_i and data_b0_i.
data_b1_i	IN	B_WIDTH	This signal contains the multiplier value of the multiplication pair data_b0_i and data_b1_i.
addnsub_i	IN	1	This signal dictates the operation to be used by the accumulator – *1'b0* for addition and *1'b1* for subtraction.
result_o	OUT	ACC_WIDTH	This signal carries the result_o of the addition/subtraction of the multiplication products. Core functionality is represented below. M0 = (data_a0_i * data_b0_i) M1 = (data_a1_i * data_b1_i) addnsub_i == 0: {M0 + M1} addnsub_i == 1: {M0 – M1}

2.3.3. Attributes

Table 2.11. Multiply-Add-Subtract Attributes

Configuration	Range	Default Value	Description
A_WIDTH	2–72	9	This configuration controls the width of the inputs data_a0_i and data_a1_i.
B_WIDTH	2–72	9	This configuration controls the width of the inputs data_b0_i and data_b1_i.
OUT_WIDTH	5-145	19	This parameter is uneditable and only gets the sum of A_WIDTH, B_WIDTH + 1. Input A Width + Input B Width + 1 = Result Width
A_SIGNED	Signed, Unsigned	Unsigned	This configuration controls the sign interpretation of the input data_a0_i and data_a1_i, either signed or unsigned.
B_SIGNED	Signed, Unsigned	Unsigned	This configuration controls the sign interpretation of the input data_b0_i and data_b1_i, either signed or unsigned.
RST_MODE	SYNC or ASYNC	SYNC	This configuration controls the reset mode, either asynchronous reset or synchronous reset. When set to asynchronous reset, the reset value occurs at the preceding clock edge.
REGINPUTAB0	on, off	on	This configuration controls the registering of the inputs, data_a0_i and data_b0_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.
REGINPUTAB1	on, off	on	This configuration controls the registering of the inputs, data_a1_i and data_b1_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.

Configuration	Range	Default Value	Description
REGOUTPUT	on, off	on	This configuration controls the registering of the output, result_o. off – Unregistered. Result is directly routed to the output. on – Registered. Result is registered at the output.
REGPIPE	on, off	on	This configuration controls the insertion of pipeline into the multiplier operation.

2.3.4. Timing Diagram

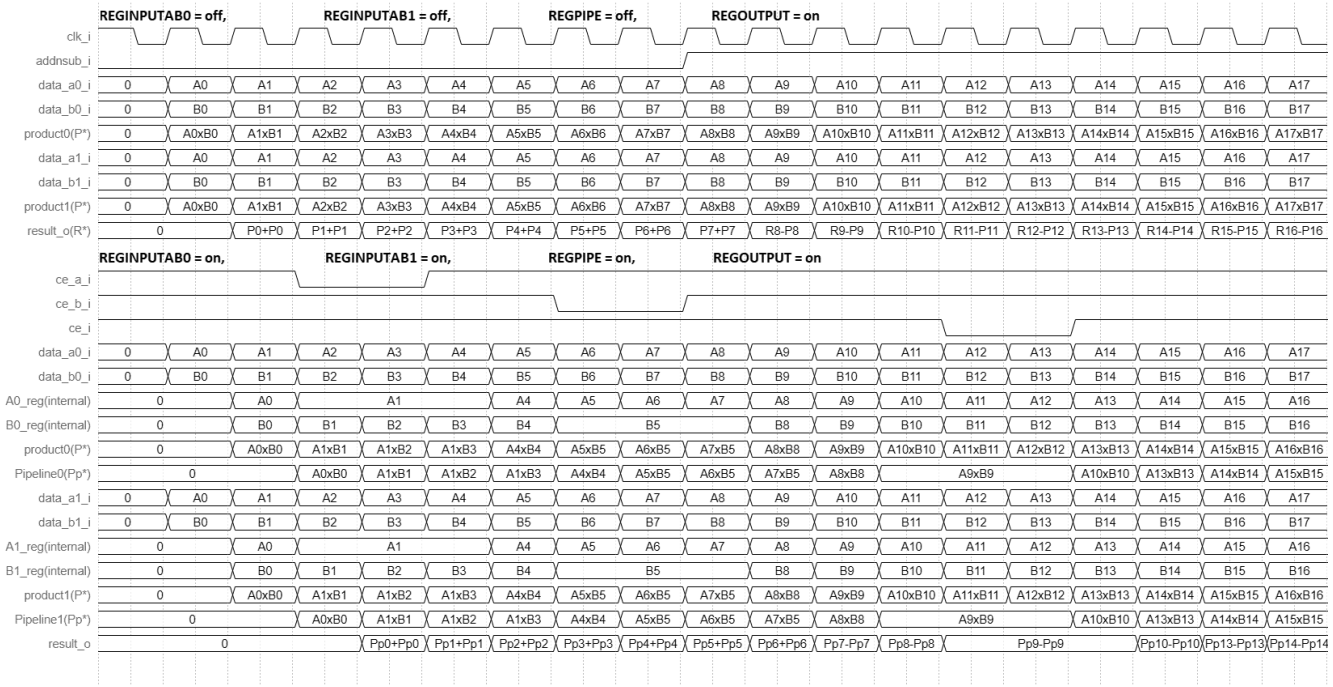

Figure 2.6. Multiply-Add-Subtract Timing Diagram

Table 2.12. Multiply-Add-Subtract Timing Table

Attribute			Latency from Input to Output
REGINPUT *	REGPIPE	REGOUPUT	
0	0	0	0
1	0	0	1
0	1	0	1
0	0	1	1
0	1	1	2
1	0	1	2
1	1	0	2
1	1	1	3

2.4. Multiply-Add-Subtract-Sum

Two pair of two-input multipliers that performs signed/unsigned multiplication of the data from inputs data_a and data_b with addition/subtraction on the product pair. The output result_o carries the Sumation of the results from the result of the added/subtracted procut pair.

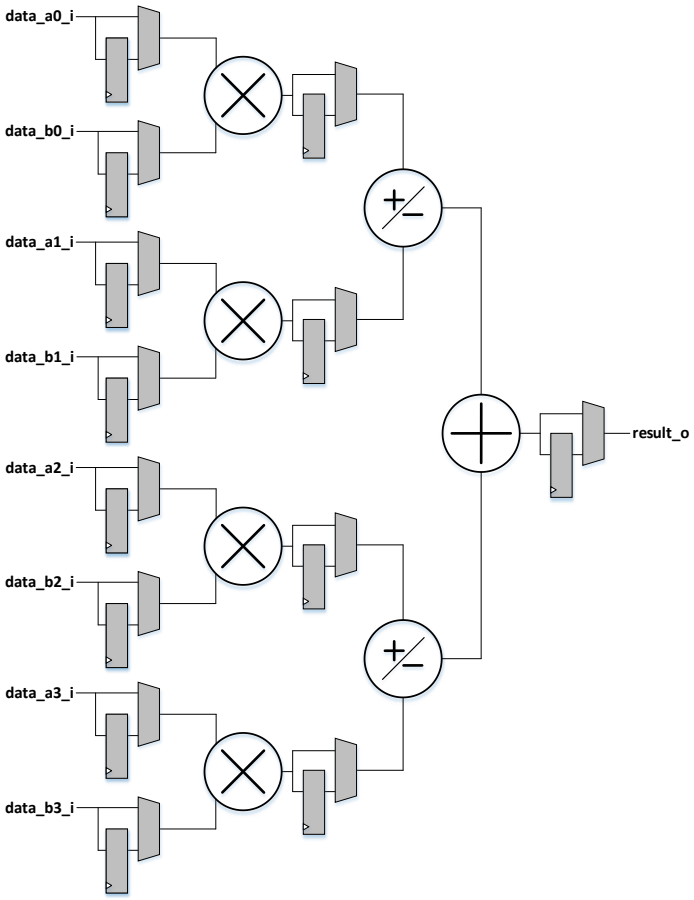

Figure 2.7. Multiply-Add-Subtract-Sum Schematic

2.4.1. Multiply-Add-Subtract-Sum Operation

Table 2.13. Multiply-Add-Subtract-Sum Operation

Operation Dependency	Operation
addnsub0_i = 0 addnsub1_i = 0	result_o = {(data_a0_i*data_b0_i) + (data_a1_i *data_b1_i) } + {(data_a2_i*data_b2_i) + (data_a3_i *data_b3_i)}
addnsub0_i = 1 addnsub1_i = 1	result_o = {(data_a0_i*data_b0_i) - (data_a1_i *data_b1_i)} + {(data_a2_i*data_b2_i) - (data_a3_i *data_b3_i)} }

2.4.2. Ports

Table 2.14. Multiply-Add-Subtract-Sum Ports

Signal Name	Direction	Width (bits)	Description
clk_i	IN	1	This signal connects to the clock pin of the input, and output Unused when neither Input A, Input B or Output is set to Registered
ce_a_i	IN	1	This signal is an active HIGH synchronous clock enable for input A 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Un-used when Input A is not set to Registered.
ce_b_i	IN	1	This signal is an active HIGH synchronous clock enable for input B 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Un-used when Input B is not set to Registered
ce_i	IN	1	This signal is an active HIGH synchronous clock enable for Pipeline and Output 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic Unused when Output is not set to Registered
rst_i	IN	1	It is an active HIGH . Reset mode depends on RST_MODE.
data_a0_i	IN	A_WIDTH	This signal contains the multiplicand value of the multiplication pair data_a0_i and data_b0_i.
data_a1_i	IN	A_WIDTH	This signal contains the multiplicand value of the multiplication pair data_a1_i and data_b1_i.
data_a2_i	IN	A_WIDTH	This signal contains the multiplicand value of the multiplication pair data_a2_i and data_b2_i.
data_a3_i	IN	A_WIDTH	This signal contains the multiplicand value of the multiplication pair data_a3_i and data_b3_i.
data_b0_i	IN	B_WIDTH	This signal contains the multiplier value of the multiplication pair data_a0_i and data_b0_i.
data_b1_i	IN	B_WIDTH	This signal contains the multiplier value of the multiplication pair data_a1_i and data_b1_i.
data_b2_i	IN	B_WIDTH	This signal contains the multiplier value of the multiplication pair data_a2_i and data_b2_i.
data_b3_i	IN	B_WIDTH	This signal contains the multiplier value of the multiplication pair data_a3_i and data_b3_i.
addnsub_i	IN	1	This signal dictates the operation to be used by the accumulator – *1'b0* for addition and *1'b1* for subtraction.
result_o	OUT	ACC_WIDTH	This signal carries the result_o of the addition/subtraction of the multiplication products. Core functionality is represented below. M0 = (data_a0_i * data_b0_i), M1 = (data_a1_i * data_b1_i), M2 = (data_a2_i * data_b2_i), M3 = (data_a3_i * data_b3_i) addnsub0_i == 0: {M0 + M1}, addnsub0_i == 1: {M0 – M1}, addnsub1_i == 0: {M2 + M3}, addnsub1_i == 1: {M2 – M3}

2.4.3. Attributes

Table 2.15. Multiply-Add-Subtract-Sum Attributes

Configuration	Range	Default Value	Description
A_WIDTH	2–72	9	This configuration controls the width of the inputs data_a0_i, data_a1_i, data_a2_i, and data_a3_i.
B_WIDTH	2–72	9	This configuration controls the width of the inputs data_b0_i, data_b1_i, data_b2_i, and data_b3_i.
OUT_WIDTH	6–146	20	This parameter is uneditable, it only gets the sum of A_WIDTH, B_WIDTH + 3. Input A Width + Input B Width + 3 = Result Width
A_SIGNED	Signed, Unsigned	Unsigned	This configuration controls the sign interpretation of the input data_a0_i, data_a1_i, data_a2_i, and data_a3_i; either signed or unsigned.
B_SIGNED	Signed, Unsigned	Unsigned	This configuration controls the sign interpretation of the input data_b0_i, data_b1_i, data_b2_i, and data_b3_i; either signed or unsigned.
RST_MODE	SYNC or ASYNC	SYNC	This configuration controls the reset mode, either asynchronous reset or synchronous reset. When set to asynchronous reset, the reset value occurs at the preceding clock edge.
REGINPUTA	on, off	on	This configuration controls the registering of the inputs, data_a0_i, data_a1_i, data_a2_i, and data_a3_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.
REGINPUTB	on, off	on	This configuration controls the registering of the inputs, data_b0_i, data_b1_i, data_b2_i, and data_b3_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.
REGPIPE	on, off	on	This configuration controls the insertion of pipeline into the multiplier operation.
REGOUTPUT	on, off	on	This configuration controls the registering of the output, result_o. off – Unregistered. Result is directly routed to the output. on – Registered. Result is registered at the output.

2.4.4. Timing Diagram

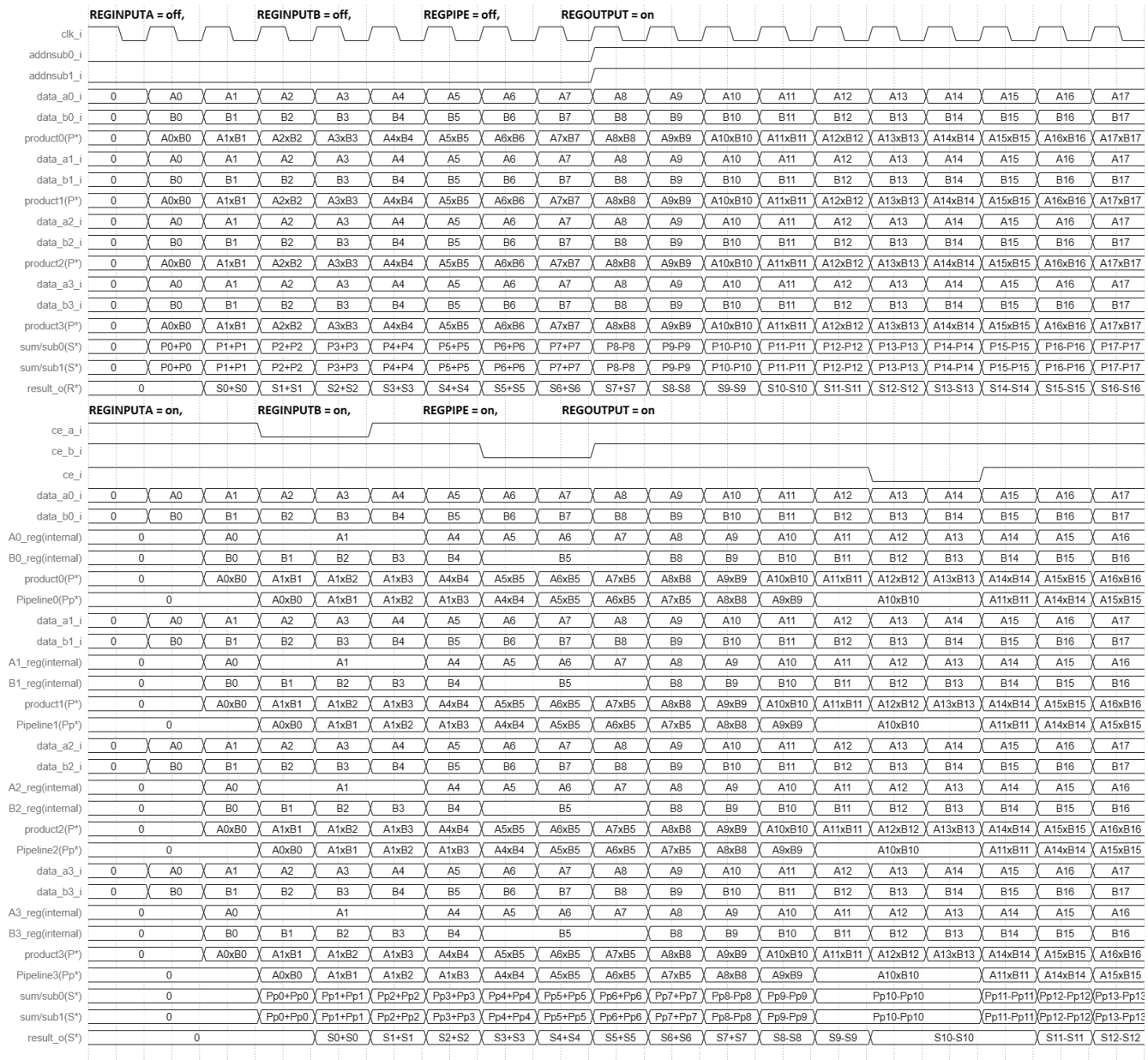

Figure 2.8. Multiply-Add-Subtract-Sum Timing Diagram

Table 2.16. Multiply-Add-Subtract-Sum Timing Table

Attribute			Latency from Input to Output
REGINPUT *	REGPIPE	REGOUTPUT	
0	0	0	0
1	0	0	1
0	1	0	1
0	0	1	1
0	1	1	2
1	0	1	2
1	1	0	2
1	1	1	3

2.5. Multiply-Multiply-Accumulate

A pair of two-input multipliers that performs signed/unsigned multiplication of the data from inputs data_a* and data_b* with addition/subtraction of the product pair and an accumulator at the end. The output result_o carries the Sum/Difference of Products added to the previews result_o or ld input depending on the accumsload value.

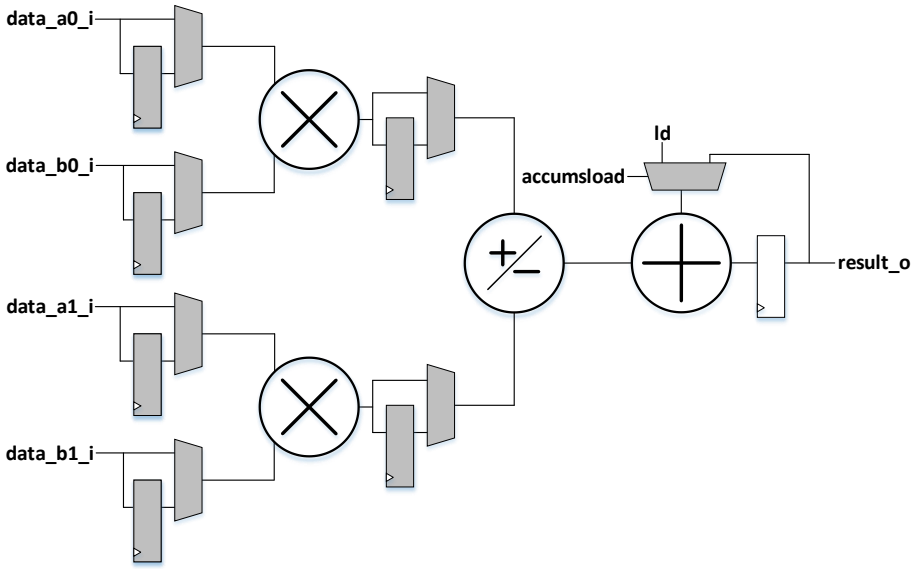

Figure 2.9. Multiply-Multiply-Accumulate Schematic

2.5.1. Multiply-Multiply-Accumulate Operation

Table 2.17. Multiply-Multiply-Accumulate Operations

Operation Dependency	Operation
addnsub_i = 0, accumsload = 0	result_o = result_nxt + (data_a0_i*data_b0_i) + (data_a1_i *data_b1_i)
addnsub_i = 0, accumsload = 1	result_o = ld + (data_a0_i*data_b0_i) + (data_a1_i *data_b1_i)
addnsub_i = 1, accumsload = 0	result_o = result_nxt + (data_a0_i*data_b0_i) - (data_a1_i *data_b1_i)
addnsub_i = 1, accumsload = 1	result_o = ld + (data_a0_i*data_b0_i) - (data_a1_i *data_b1_i)

2.5.2. Ports

Table 2.18. Multiply-Multiply-Accumulate Ports

Signal Name	Direction	Width (bits)	Description
clk_i	IN	1	This signal connects to the clock pin of the input, and output Un-used when neither Input A, Input B or Output is set to Registered
ce_a_i	IN	1	This signal is an active HIGH synchronous clock enable for input A 1'b0 - retain the previous state 1'b1 - toggle on the rising edge of the clock as per the logic Un-used when Input A is not set to Registered.

Signal Name	Direction	Width (bits)	Description
ce_b_i	IN	1	This signal is an active HIGH synchronous clock enable for input B 1'b0 - retain the previous state 1'b1 - toggle on the rising edge of the clock as per the logic Un-used when Input B is not set to Registered
ce_i	IN	1	This signal is an active HIGH synchronous clock enable for Pipeline and Output 1'b0 – retain the previous state 1'b1 – toggle on the rising edge of the clock as per the logic
rst_i	IN	1	It is an active HIGH reset. Reset mode depends on RST_MODE.
data_a0_i	IN	A_WIDTH	This signal contains the multiplicand value of the multiplication pair data_a0_i and data_b0_i.
data_a1_i	IN	A_WIDTH	This signal contains the multiplicand value of the multiplication pair data_a1_i and data_b1_i.
data_b0_i	IN	B_WIDTH	This signal contains the multiplier value of the multiplication pair data_a0_i and data_b0_i.
data_b1_i	IN	B_WIDTH	This signal contains the multiplier value of the multiplication pair data_a1_i and data_b1_i.
accumsload	IN	1	This signal destates if the ld input or the previews result_o is added from the product of the two multipliers.
ld	IN	ACC_WIDTH	This signal is added or subtracted from the product of the two multiplier when accumsload is high.
addnsub_i	IN	1	This signal dictates the operation to be used on the product pair. ‘0’ for addition and ‘1’ for subtraction.
result_o	OUT	ACC_WIDTH	This signal carries the result_o of the addition/subtraction of the multiplication products added to the previews result_o. Core functionality is represented below. M0 = (data_a0_i * data_b0_i) M1 = (data_a1_i * data_b1_i) result_nxt = previews result_o addnsub_i == 0: {M0 + M1} addnsub_i == 1: {M0 – M1}

2.5.3. Attributes

Table 2.19. Multiply-Multiply-Accumulate Attributes

Configuration	Range	Default Value	Description
A_WIDTH	2–72	9	This configuration controls the width of the inputs data_a0_i and data_a1_i.
B_WIDTH	2–72	9	This configuration controls the width of the inputs data_b0_i and data_b1_i.
ACC_WIDTH	7-147	21	This parameter is uneditable, it only gets the sum A_WIDTH, B_WIDTH + 3. Input A Width + Input B Width + 3 = Result Width
A_SIGNED	Signed, Unsigned	Unsigned	This configuration controls the sign interpretation of the input data_a0_i and data_a1_i, either signed or unsigned.

Configuration	Range	Default Value	Description
B_SIGNED	Signed, Unsigned	Unsigned	This configuration controls the sign interpretation of the input data_b0_i and data_b1_i, either signed or unsigned.
RST_MODE	SYNC or ASYNC	SYNC	This configuration controls the reset mode, either asynchronous reset or synchronous reset.
REGINPUTAB0	on, off	on	This configuration controls the registering of the inputs, data_a0_i and data_b0_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.
REGINPUTAB1	on, off	on	This configuration controls the registering of the inputs, data_a1_i and data_b1_i before routing them to the multiplication operation. off – Unregistered. Inputs are directly routed. on – Registered. Inputs are registered before routing.
REGPIPE	on, off	off	This configuration controls the insertion of pipeline into the multiplier operation.
REGOUTPUT	on	on	Not Configurable

2.5.4. Timing Diagram

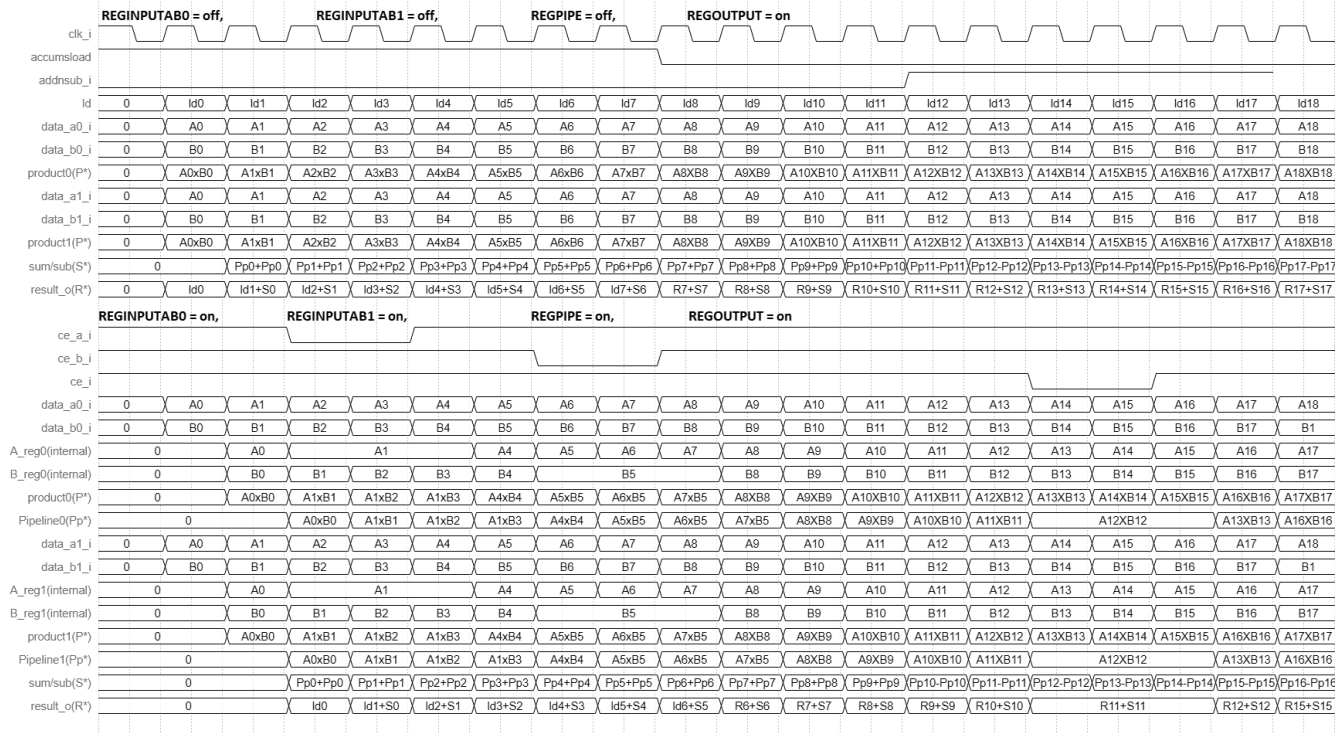

Figure 2.10. Multiply-Multiply-Accumulate Timing Diagram

Table 2.20. Multiply-Multiply-Accumulate Timing Table

Attribute			Latency from Input to Output
REGINPUT *	REGPIPE	REGOUTPUT	
0	0	0	0
1	0	0	1
0	1	0	1
0	0	1	1
0	1	1	2
1	0	1	2
1	1	0	2
1	1	1	3

3. IP Generation and Simulation

This section provides information on how to generate the IP Core using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant software, refer to the Lattice Radiant software user guide.

3.1. Generating the IP

The following section shows an example of generating a 32-bit Multiplier module. This process is similar for all DSP Arithmetic Modules.

To generate a 32-bit Multiplier module:

1. Double-click **DSP Multiplier** under **DSP_Arithmetic_Modules**. This opens the **Module/IP Block Wizard**.
2. Fill out the following information then click **Next**. An example is shown in Figure 3.1.
   - Language
   - Instance name
   - Create in

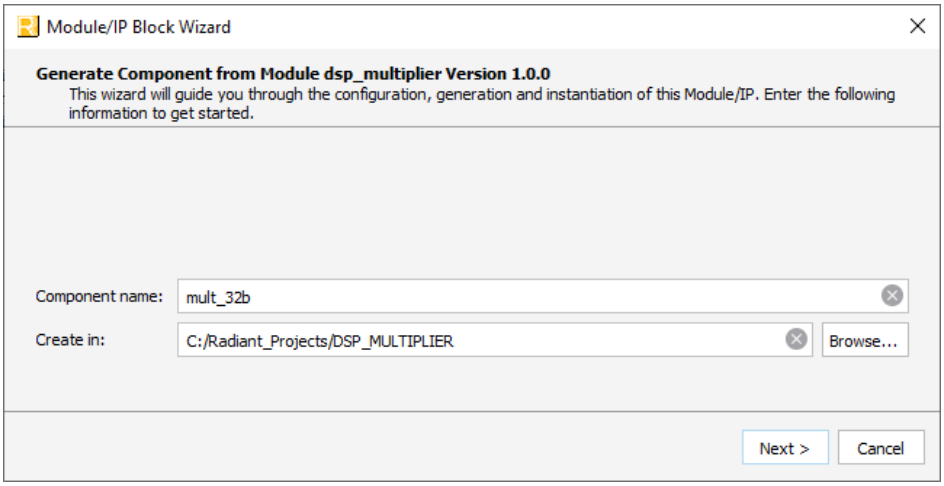

Figure 3.1. Example: Generating 32-bit Multiplier Using Module/IP Block Wizard

3. In the **IP Configuration** page, customize the Multiplier by selecting options as shown in Figure 3.2.

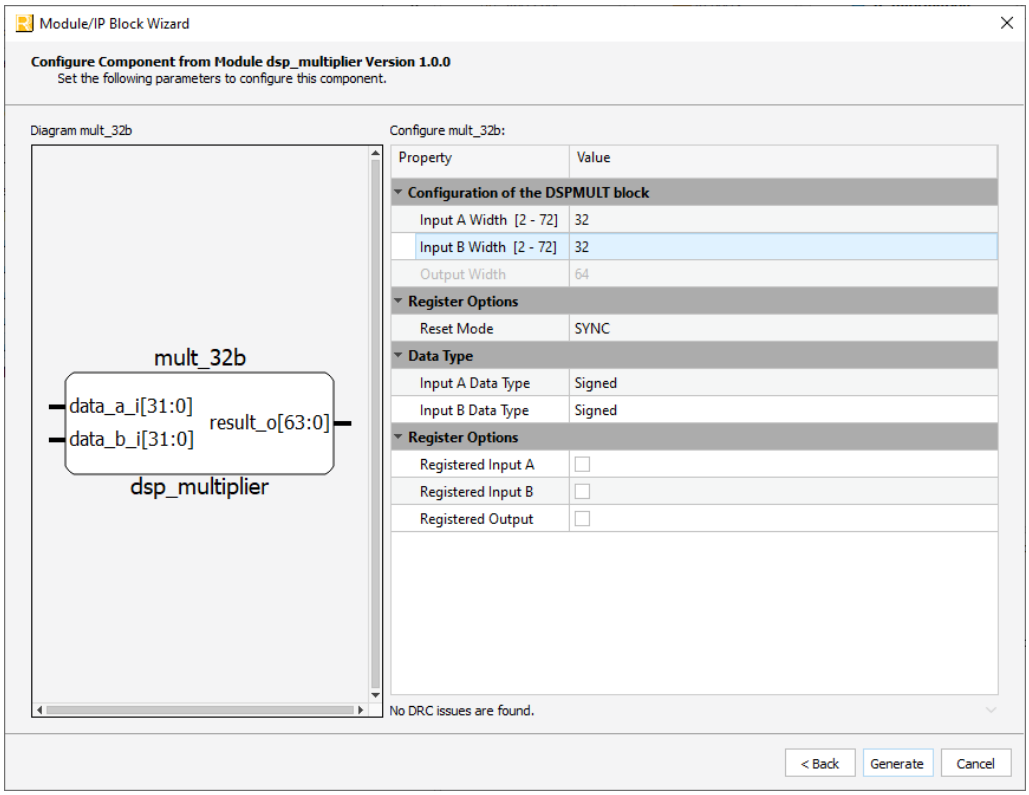

Figure 3.2. Example: Generating 32-bit Multiplier in IP Configuration

4. When all the options are set, click **Generate**.
5. Click **Finish**.
6. Once this module is in the Lattice Radiant Software project, it can be instantiated in other modules within the project. You can view the files and instance/s added in the **File List** tab.

3.2. Running Functional Simulation

Lattice Radiant Software also allows you to simulate configured arithmetic modules, so you can observe the conditions of the output with a given stimuli. This section details the individual steps needed to run simulation through Simulation Wizard. In the sample procedure, a Multiplier on default settings is generated, including an instance name for the multiplier.

Table 3.1 provides a summary description of the files used in the project.

Table 3.1. File List

File	Sim	Synthesis	Description
rtl/<instance_name>.v	Yes	Yes	Top Level RTL file with the selected configuration. The main IP file
testbench/dut_params	Yes	—	Top level parameters of the generated RTL file
testbench/dut_inst	Yes	—	Instantiated version of the <IP_name>.v file for simulation use
tb_top.v	Yes	—	Test bench template, this can be edited to match your specific needs.
<instance name>.cfg	—	—	This file contains the configuration options used to recreate or modify the core in the IP Platform.

File	Sim	Synthesis	Description
<instance name>.ipx	—	—	The IPX file holds references to all of the elements of an IP or Module after it is generated from the IP Platform GUI. The file is used to bring in the appropriate files during the design implementation and analysis. It is also used to re-load parameter settings into the IP Platform when the IP/Module is being regenerated.
component.xml	—	—	Contains the ipxact:component information of the IP.
design.xml	—	—	Documents the configuration parameters of the IP in IP-XACT 2014 format.
rlt/<instance name>_bb.v	—	—	This file provides the synthesis black box.
misc/<Instance Name>_tmpl.v misc /<instance name>_tmpl.vhd	—	—	These files provide instance templates for the IP core.
eval/constraint.pdc	—	—	This file contains the PDC constraints for this IP. Refer to section 3.3 on how to use this file.

To run functional simulation:

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in Figure 3.3. The **Simulation Wizard** window appears. Click **Next**. You are prompted to select the working folder and the test bench name. Enter project name for simulation in this case **rtl_sim**.

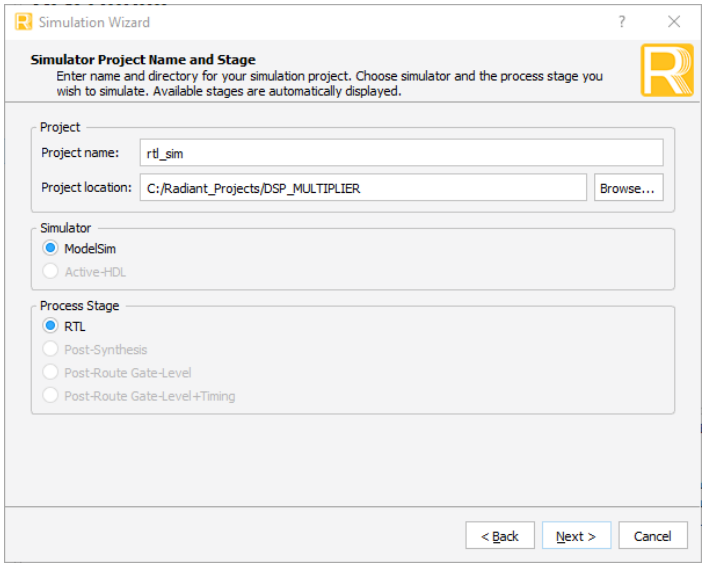

Figure 3.3. Simulation Wizard

2. Click **Next**.
3. In the prompt for folder creation, click **Yes**.
4. Moving to the processing stage, select **RTL**. Click **Next**.
5. The **Source Files** list appears. This shows the list of files included for simulation. Here, you can add or subtract any other files that are missed. Leave the file order for now. Click **Next**.

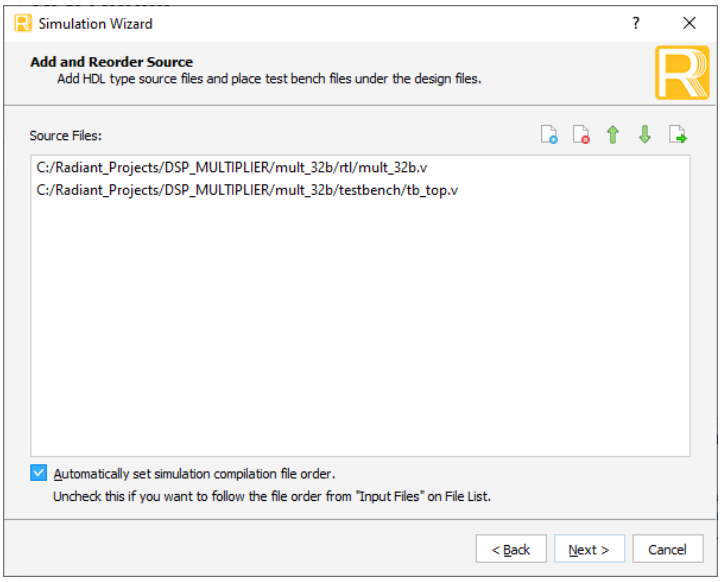

Figure 3.4. Final Simulation Files

6. In the next section, the files as well as the other dependencies are parsed to check for final errors. We can also select the top simulation module, before passing the files to a simulator for execution. Click **Next** for a summary view.

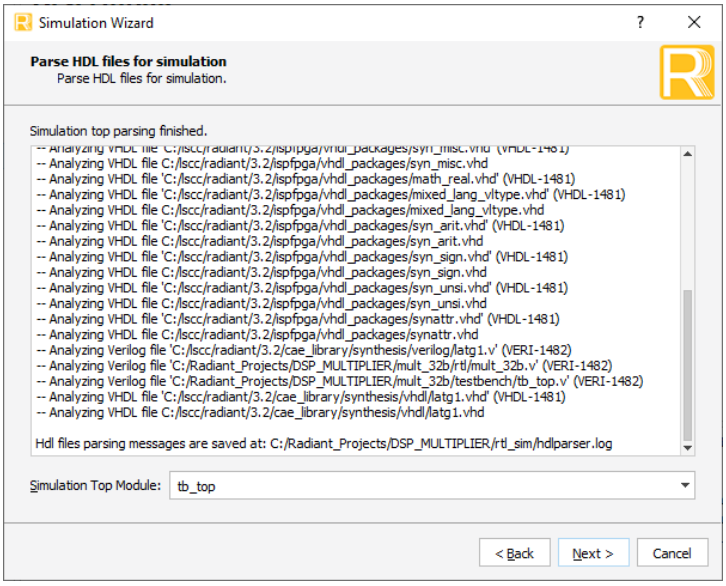

Figure 3.5. File Parsing and Top Module Selection

7. Click **Finish**. The ModelSim Lattice Edition software opens, which automatically compiles and runs the RTL simulation.

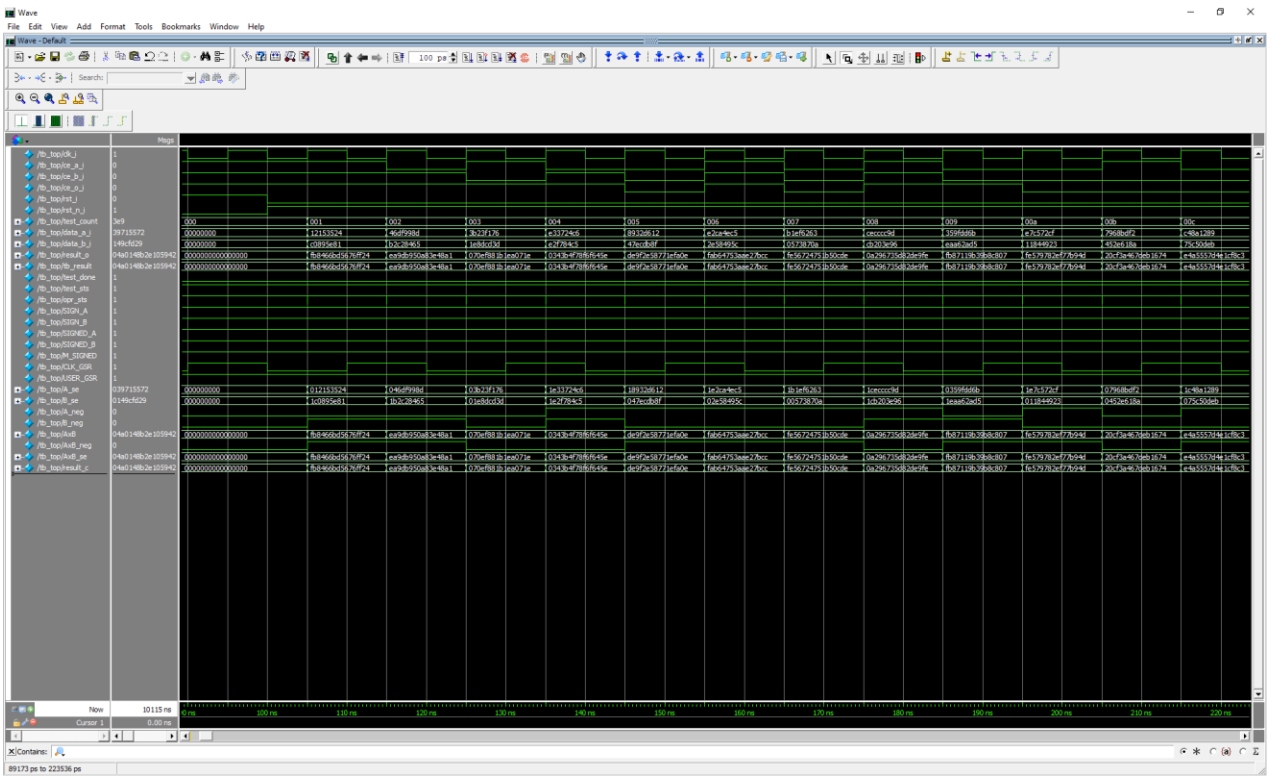

Figure 3.6. Simulation Waveforms

When Modelsim opens, compiles the included RTL files and runs the simulation up to the first 100 ns. To view the rest of the simulation, click the **Simulate>Run>Run -All** button on top and zoom out for the overview of the signals.

3.3. Constraining the IP

You need to provide proper timing and physical design constraints to ensure that your design meets the desired performance goals on the FPGA. Add the content of the following IP constraint file to your design constraints:

<IP Instance_Path>/<IP Instance_Name>/eval/constraint.pdc

The constraint file has been verified during IP evaluation with the IP instantiated directly in the top-level module. You can modify the constraints in this file with thorough understanding of the effect of each constraint.

Refer to Lattice Radiant Timing Constraints Methodology for details on how to constrain your design.

Appendix A. Resource Utilization

This appendix provides resource utilization information for Lattice FPGAs using the Arithmetic Modules. The IP configurations shown in this chapter are generated using the Lattice Radiant tool.

A.1. Multiplier

Table A.1. Performance and Resource Utilization

Configuration	Clk Fmax (MHz)	LUTs	Registers	DSP	EBRs
2-bit Signed Multiplier with its input and output registered	250.000	0	0	1	0
18-bit Signed Multiplier with its input and output registered	250.000	0	0	1	0
72-bit Signed Multiplier with its input and output registered	71.767	0	0	16	0

Note: Performance and utilization characteristics are generated targeting the LAV-AT-E70-3LFG1156C device using the Lattice Radiant 2023.1 Software and Synplify Pro Synthesis Tool. Performance may vary when using this IP in a different density, speed, or grade within the LAV-AT family or in a different software version. Fmax is generated when the FPGA design contains only the DSP Multiplier Module and the target frequency is 250 MHz.

A.2. Multiply-Accumulate

Table A.2. Performance and Resource Utilization

Configuration	Clk Fmax (MHz)	LUTs	Registers	DSP	EBRs
2-bit Signed Mult-Acc with input, output, and pipeline register	250.000	48	6	1	0
18-bit Signed Mult-Acc with input, output, and pipeline register	190.078	48	38	1	0
72-bit Signed Mult-Acc with input, output and pipeline register	55.782	145	146	16	0

Note: Performance and utilization characteristics are generated targeting the LAV-AT-E70-3LFG1156C device using the Lattice Radiant 2023.1 Software and Synplify Pro Synthesis Tool. Performance may vary when using this IP in a different density, speed, or grade within the LAV-AT family or in a different software version. Fmax is generated when the FPGA design contains only the DSP Multiply-Accumulate Module and the target frequency is 250 MHz.

A.3. Mult-Add-Subtract

Table A.3. Performance and Resource Utilization

Configuration	Clk Fmax (MHz)	LUTs	Registers	DSP	EBRs
2-bit Signed Mult-Add-Sub with input, output, and pipeline register	250.000	0	0	2	0
18-bit Signed Mult-Add-Sub with input, output, and pipeline register	250.000	0	0	2	0
72-bit Signed Mult-Add-Sub with input, output, and pipeline register	62.590	0	0	32	0

Note: Performance and utilization characteristics are generated targeting the LAV-AT-E70-3LFG1156C device using the Lattice Radiant 2023.1 Software and Synplify Pro Synthesis Tool. Performance may vary when using this IP in a different density, speed, or grade within the LAV-AT family or in a different software version. Fmax is generated when FPGA design contains only the DSP Mult-Add-Subtract Module and the target frequency is 250 MHz.

A.4. Mult-Add-Subtract-Sum

Table A.4. Performance and Resource Utilization

Configuration	Clk Fmax (MHz)	LUTs	Registers	DSP	EBRs
2-bit Signed Mult-Add-Sub-Sum with input, output, and pipeline register	221.484	0	0	5	0
18-bit Signed Mult-Add-Sub-Sum with input, output, and pipeline register	155.280	0	0	7	0
36-bit Signed Mult-Add-Sub-Sum with input, output, and pipeline register	86.550	0	0	21	0

Note: Performance and utilization characteristics are generated targeting the LAV-AT-E70-3LFG1156C device using the Lattice Radiant 2023.1 Software and Synplify Pro Synthesis Tool. Performance may vary when using this IP in a different density, speed, or grade within the LAV-AT family or in a different software version. Fmax is generated when FPGA design contains only the DSP Mult-Add-Subtract-Sum Module and the target frequency is 250 MHz.

A.5. Mult-Mult-Accumulate

Table A.5. Performance and Resource Utilization

Configuration	Clk Fmax (MHz)	LUTs	Registers	DSP	EBRs
9-bit Signed Mult-Mult-Accumulate with input, output, and pipeline register	156.397	21	39	4	0
18-bit Signed Mult-Mult-Accumulate with input, output, and pipeline register	177.117	39	39	4	0
36-bit Signed Mult-Mult-Accumulate with input, output, and pipeline register	86.319	77	75	13	0

Note: Performance and utilization characteristics are generated targeting the LAV-AT-E70-3LFG1156C device using the Lattice Radiant 2023.1 Software and Synplify Pro Synthesis Tool. Performance may vary when using this IP in a different density, speed, or grade within the LAV-AT family or in a different software version. Fmax is generated when FPGA design contains only the DSP Mult-Mult-Accumulate Module and the target frequency is 250 MHz.

Appendix B. Nexus to Avant Migration Guide

This section provides details about migrating current Nexus designs using the Lattice Radiant software generated DSP Arithmetic Modules to Avant devices.

B.1. Multiplier

Table B.1. DSP_Multiplier PORT COMPATIBILITY

Nexus Radiant IP Catalog signal name	Avant Radiant IP Catalog signal name
clk_i	clk_i
ce_a_i	ce_a_i
ce_b_i	ce_b_i
ce_o_i	ce_o_i
rst_a_i	*< no equivalent port >*
rst_b_i	*< no equivalent port >*
rst_o_i	*< no equivalent port >*
< no equivalent port >	rst_i
data_a_i	data_a_i
data_b_i	data_b_i
result_o	result_o

B.2. Multiply-Accumulate

Table B.2. DSP_Mult_Accumulate PORT COMPATIBILITY

Nexus Radiant IP Catalog signal name	Avant Radiant IP Catalog signal name
clk_i	clk_i
ce_a_i	ce_a_i
ce_b_i	ce_b_i
ce_p_i	*< no equivalent port >*
ce_o_i	ce_i
rst_a_i	*< no equivalent port >*
rst_b_i	*< no equivalent port >*
rst_p_i	*< no equivalent port >*
rst_o_i	*< no equivalent port >*
< no equivalent port >	rst_i
accumsload	accumsload
addnsub_i	addnsub_i
data_a_i	data_a_i

Nexus Radiant IP Catalog signal name	Avant Radiant IP Catalog signal name
data_b_i	data_b_i
ld	ld
result_o	result_o

B.3. Mult-Add-Subtract

Table B.3. DSP_Mult_Add_Sub PORT COMPATIBILITY

Nexust Radiant IP Catalog signal name	Avant Radiant IP Catalog signal name
clk_i	clk_i
ce_a_i	ce_a_i
ce_b_i	ce_b_i
ce_p_i	*< no equivalent port >*
ce_o_i	ce_i
rst_a_i	*< no equivalent port >*
rst_b_i	*< no equivalent port >*
rst_p_i	*< no equivalent port >*
rst_o_i	*< no equivalent port >*
< no equivalent port >	rst_i
addnsub_i	addnsub_i
data_a0_i	data_a0_i
data_a1_i	data_a1_i
data_b0_i	data_b0_i
data_b1_i	data_b1_i
result_o	result_o

B.4. Mult-Add-Subtract-Sum

Table B.4. DSP_Mult_Add_Sub_Sum PORT COMPATIBILITY

Nexus Radiant IP Catalog signal name	Avant Radiant IP Catalog signal name
clk_i	clk_i
ce_a_i	ce_a_i
ce_b_i	ce_b_i
ce_p_i	*< no equivalent port >*
ce_o_i	ce_i
rst_a_i	*< no equivalent port >*

Nexus Radiant IP Catalog signal name	Avant Radiant IP Catalog signal name
rst_b_i	*< no equivalent port >*
rst_p_i	*< no equivalent port >*
rst_o_i	*< no equivalent port >*
< no equivalent port >	rst_i
addnsub0_i	addnsub0_i
addnsub1_i	addnsub1_i
data_a0_i	data_a0_i
data_a1_i	data_a1_i
data_a2_i	data_a2_i
data_a3_i	data_a3_i
data_b0_i	data_b0_i
data_b1_i	data_b1_i
data_b2_i	data_b2_i
data_b3_i	data_b3_i
result_o	result_o

B.5. Mult-Mult-Accumulate

Table B.5. DSP_Mult_Mult_Accumulate PORT COMPATIBILITY

Nexus Radiant IP Catalog signal name	Avant Radiant IP Catalog signal name
clk_i	clk_i
ce_a_i	ce_a_i
ce_b_i	ce_b_i
ce_p_i	*< no equivalent port >*
ce_o_i	ce_i
rst_a_i	*< no equivalent port >*
rst_b_i	*< no equivalent port >*
rst_p_i	*< no equivalent port >*
rst_o_i	*< no equivalent port >*
< no equivalent port >	rst_i
accumsload	accumsload
addnsub_i	addnsub_i
data_a0_i	data_a0_i
data_a1_i	data_a1_i

Nexus Radiant IP Catalog signal name	Avant Radiant IP Catalog signal name
data_b0_i	data_b0_i
data_b1_i	data_b1_i
ld	ld
result_o	result_o

References

For more information refer to:

- Lattice Insights for Lattice Semiconductor training courses and learning plans
- Lattice Radiant Timing Constraints Methodology
- Lattice Avant Platform web page.
- Lattice Radiant Software web page.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.1, December 2023

Section	Change Summary
All	Performed minor editorial changes.
Disclaimers	Updated this section.
IP Generation and Simulation	• Edited step 1 in section 3.1. • Replaced instances of *IP name* with *instance name*. • Updated Table 3.1. File List. • Added section 3.3 Constraining the IP.
Resource Utilization	• Added a column for Clk Fmax (MHz) in Table A.1, Table A.2, Table A.3, Table A.4, and Table A.5. • Updated notes for Table A.1, Table A.2, Table A.3, Table A.4, and Table A.5.
References	Added this section.
Technical Support Assistance	Added link to the Lattice Answer Database.

Revision 1.0, November 2022

Section	Change Summary
All	Initial Release.

Revision 0.8, May 2022

Section	Change Summary
All	Preliminary release.

www.latticesemi.com