

Multiply Adder v3.0

LogiCORE IP Product Guide

Vivado Design Suite

PG192 February 4, 2021



Table of Contents

IP Facts

Chapter 1: Overview

Navigating Content by Design Process	5
Core Overview	5
Feature Summary	6
Applications	6
Unsupported Features	6
Licensing and Ordering	6

Chapter 2: Product Specification

Resource Utilization	7
Performance	7
Port Descriptions	7

Chapter 3: Designing with the Core

General Design Guidelines	9
Clocking	14
Resets	14

Chapter 4: Design Flow Steps

Customizing and Generating the Core	15
Constraining the Core	18
Simulation	19
Synthesis and Implementation	19

Appendix A: Upgrading

Migrating to the Vivado Design Suite	20
Upgrading in the Vivado Design Suite	20

Appendix B: Debugging

Finding Help on Xilinx.com	21
Debug Tools	22

Appendix C: Additional Resources and Legal Notices

Xilinx Resources	23
Documentation Navigator and Design Hubs	23
References	23
Revision History	24
Please Read: Important Legal Notices	24

Introduction

The Xilinx® LogiCORE™ IP Multiply Adder core provides implementations of multiply-add using DSP slices. It performs a multiplication of two operands and adds (or subtracts) the full-precision product to a third operand. The Multiply Adder module operates on signed or unsigned data. The module can be pipelined.

Features

- Supports two's complement-signed and unsigned operations
- Supports multiplier inputs ranging from 1 to 52 bits unsigned or 2 to 53 bits signed and an add or subtract operand input ranging from 1 to 105 bits unsigned or 2 to 106 bits signed
- Optional clock enable and synchronous clear
- Optional pipelined operation

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family ⁽¹⁾	Versal™ ACAP UltraScale+™ Families UltraScale™ Architecture Zynq®-7000 SoC 7 Series
Supported User Interfaces	N/A
Resources	Performance and Resource Utilization web page.
Provided with Core	
Design Files	Netlist
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	Encrypted VHDL
Supported S/W Driver	N/A
Tested Design Flows⁽²⁾	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Not Provided
Support	
Release Notes and Known Issues	Master Answer Record: 54507
All Vivado IP Change Logs	Master Vivado IP Change Logs: 72775
Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see the Vivado IP catalog.
2. For the supported versions of third-party tools, see the [Xilinx Design Tools: Release Notes Guide.](#)

Overview

Navigating Content by Design Process

Xilinx[®] documentation is organized around a set of standard design processes to help you find relevant content for your current development task. This document covers the following design processes:

- **Hardware, IP, and Platform Development:** Creating the PL IP blocks for the hardware platform, creating PL kernels, subsystem functional simulation, and evaluating the Vivado timing, resource and power closure. Also involves developing the hardware platform for system integration. Topics in this document that apply to this design process include:
 - [Port Descriptions](#)
 - [Clocking](#)
 - [Resets](#)
 - [Customizing and Generating the Core](#)
-

Core Overview

The Xilinx[®] LogiCORE™ IP Multiply Adder core provides implementations of multiply-add using DSP slices. It performs a multiplication of two operands and adds (or subtracts) the full-precision product to a third operand.

The core operates on signed or unsigned data and can be pipelined. It performs the operation $P = C +/- A * B$.

Feature Summary

The Multiply Adder core supports the following:

- Twos complement-signed and unsigned operations
- Multiplier inputs ranging from 1 to 52 bits unsigned or 2 to 53 bits signed and an add or subtract operand input ranging from 1 to 105 bits unsigned or 2 to 106 bits signed
- Clock enable and synchronous clear
- Pipelined operation

Applications

The Multiply Adder core is more efficient than separate multiply and add operations in many configurations because it uses the DSP48 primitive. You can use this core in many applications, especially those based on polynomial arithmetic such as FIR and IIR implementations.

Unsupported Features

This core does not support AXI4-Stream interfaces. However, it supports full throughput (new data on each clock cycle).

Licensing and Ordering

This Xilinx LogiCORE™ IP module is provided at no additional cost with the Xilinx® Vivado® Design Suite under the terms of the [Xilinx End User License](#). Information about other Xilinx LogiCORE IP modules is available at the [Xilinx Intellectual Property](#) page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your [local Xilinx sales representative](#).

Product Specification

Resource Utilization

For details about resource utilization, visit the [Performance and Resource Utilization web page](#).

Performance

For details about performance, visit the [Performance and Resource Utilization web page](#).

Port Descriptions

Signal names for the core symbol are shown in [Figure 2-1](#) and described in [Table 2-1](#).

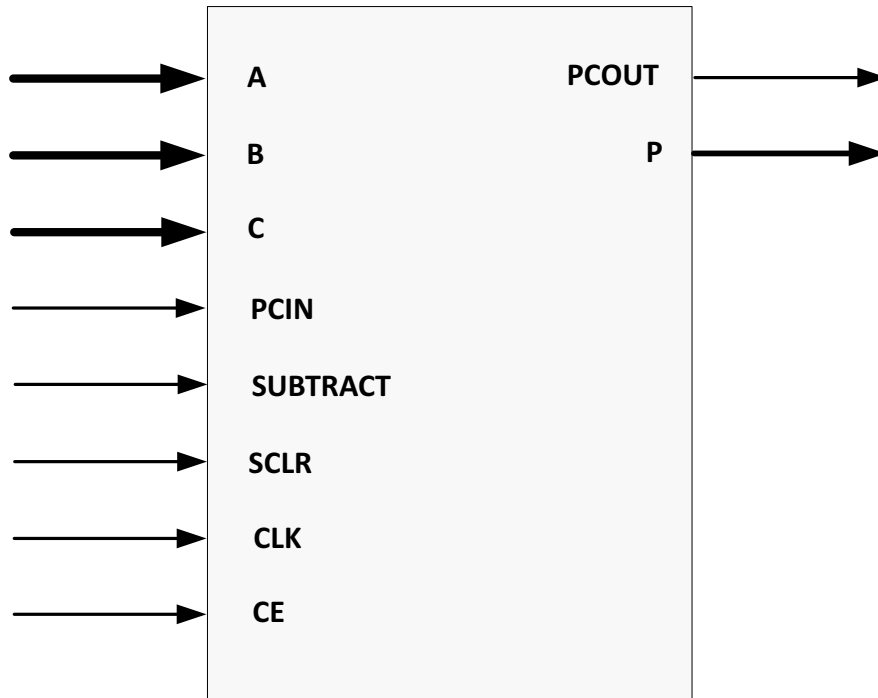


Figure 2-1: Core Symbol

Table 2-1: Core Signal Pinout

Name	I/O	Description
A[N:0]	Input	A Input bus (multiplier operand 1)
B[M:0]	Input	B Input bus (multiplier operand 2)
C[L:0] ⁽¹⁾	Input	C Input bus (operand 1 of add/sub operation)
PCIN ⁽²⁾	Input	Cascade Input
SUBTRACT	Input	Controls Add/Subtract operation (High = subtraction, Low = addition)
CE	Input	Clock Enable (active-High)
CLK	Input	Clock (rising edge)
SCLR	Input	Synchronous Clear (active-High)
PCOUT ⁽²⁾	Output	Cascade Output
P[Q:0]	Output	Output bus

Notes:

1. The multiplier output is added to or subtracted from the C port add/sub operand.
2. Cascade ports are described in [Pipelined Operation](#).

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the Multiply Adder core.

General Design Guidelines

Pipelined Operation

The Multiply Adder core takes into consideration two different latency paths; one from the A and B inputs to the P output and the other from the C/PCIN input to the P output. These latencies are defined as **A:B - P Latency** and **C - P Latency** and shown schematically in [Figure 3-1](#).

These latencies can only take on two values: 0 for no latency or -1 for maximum/optimal latency. If either one of these two latencies are specified as -1, they are both treated as if they are -1; for a completely combinatorial design, both must be set to 0.

The availability and use of the cascaded PCIN port also affect latency. Figure 3-1 shows the placement of registers inside the DSP slice implementation of the Multiply Adder in its pipelined configurations.

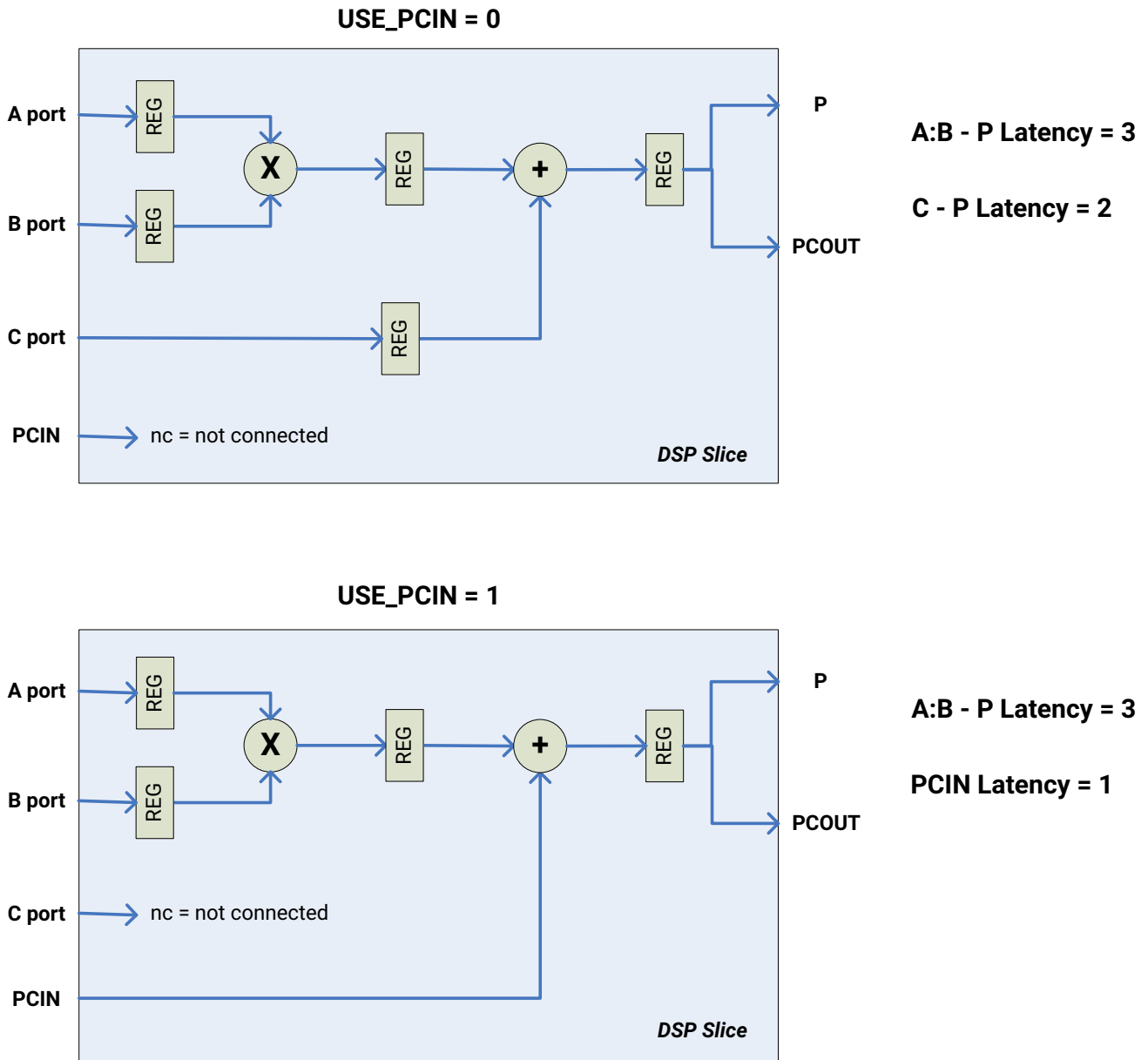


Figure 3-1: Single DSP Slice Implementation

Figure 3-2 shows the Multiply Adder configured using multiple DSP slices. Internally, the Multiplier and the Adder/Subtractor cores are used to create the wide MultAdd function. The latency variations from port-to-port for a multiple DSP slice implementation of a MultAdd are derived from the Multiplier and or the Adder/Subtractor latencies.

When the latency parameters are set to -1, there are functions that return the actual "A:B - P Latency" and "C - P Latency" values based on the setting of all the other parameters.

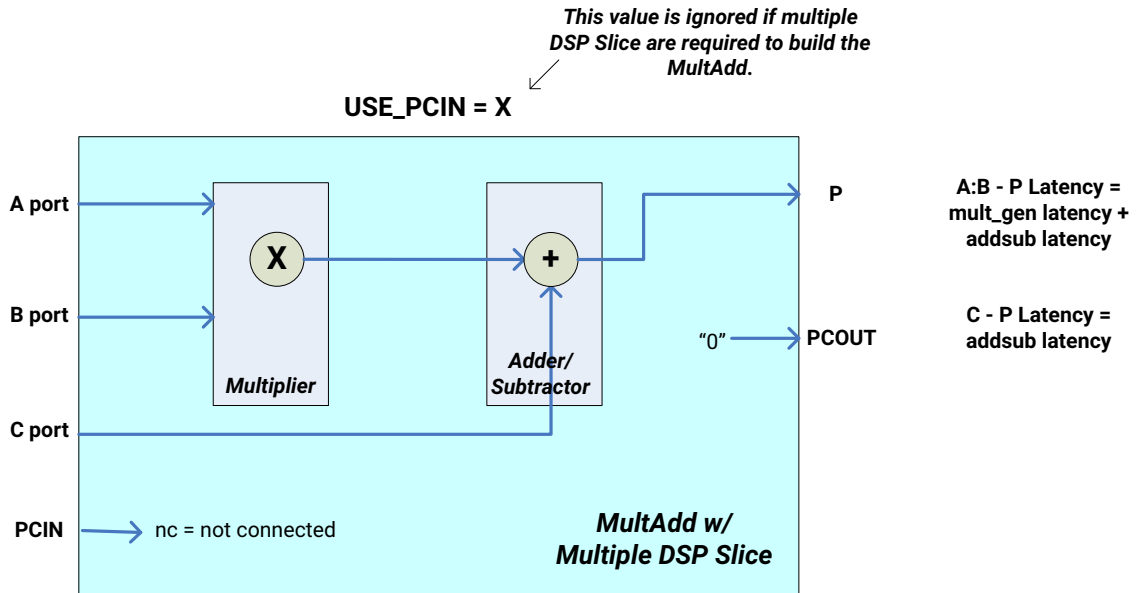


Figure 3-2: Multiple DSP Slice Implementation

Data Alignment

All inputs are right-justified when passed to the operators inside the core. You must set the proper LSB or MSB padding or sign extending of the inputs (relative to the binary point) of the core.

In the Multiply Adder, there is no truncation or rounding of the multiplier output; it is a full precision result. The C input is added to the product LSB-to-LSB. The following example shows how the operations take place. MSB and LSB positions can be chosen to extract the desired "slice" of output data. The slice shown in the example is taken from LSB = 0 to MSB = 11.

$$A*B+C = P$$

where,

- A Width = 6
- B Width = 8
- C Width = 8

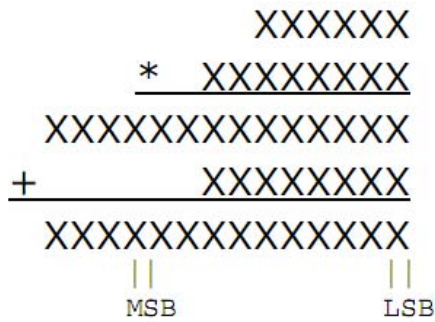


Figure 3-3: Data Alignment Example

Vector Multiply Example

Figure 3-4 (single DSP slice implementation) and Figure 3-5 (multiple DSP slice implementation) show a simple vector multiply and the necessary bit staggering required to line up inputs to the second and third MultAdd cores. If the bit widths of the core require

multiple DSP slice implementations, **Use PCIN** is disallowed and the result is provided through P only.

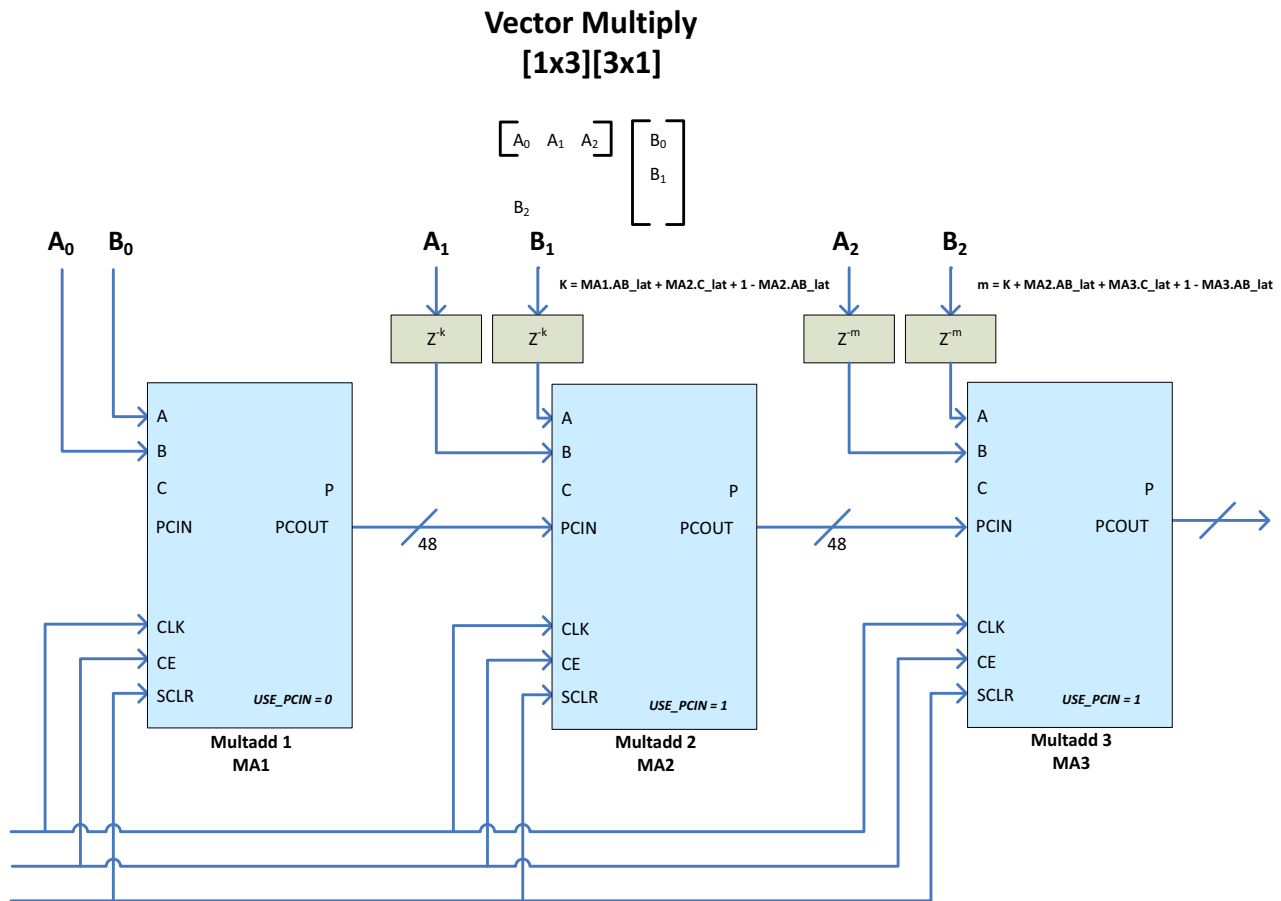


Figure 3-4: Vector Multiply - Single DSP Slice Implementation

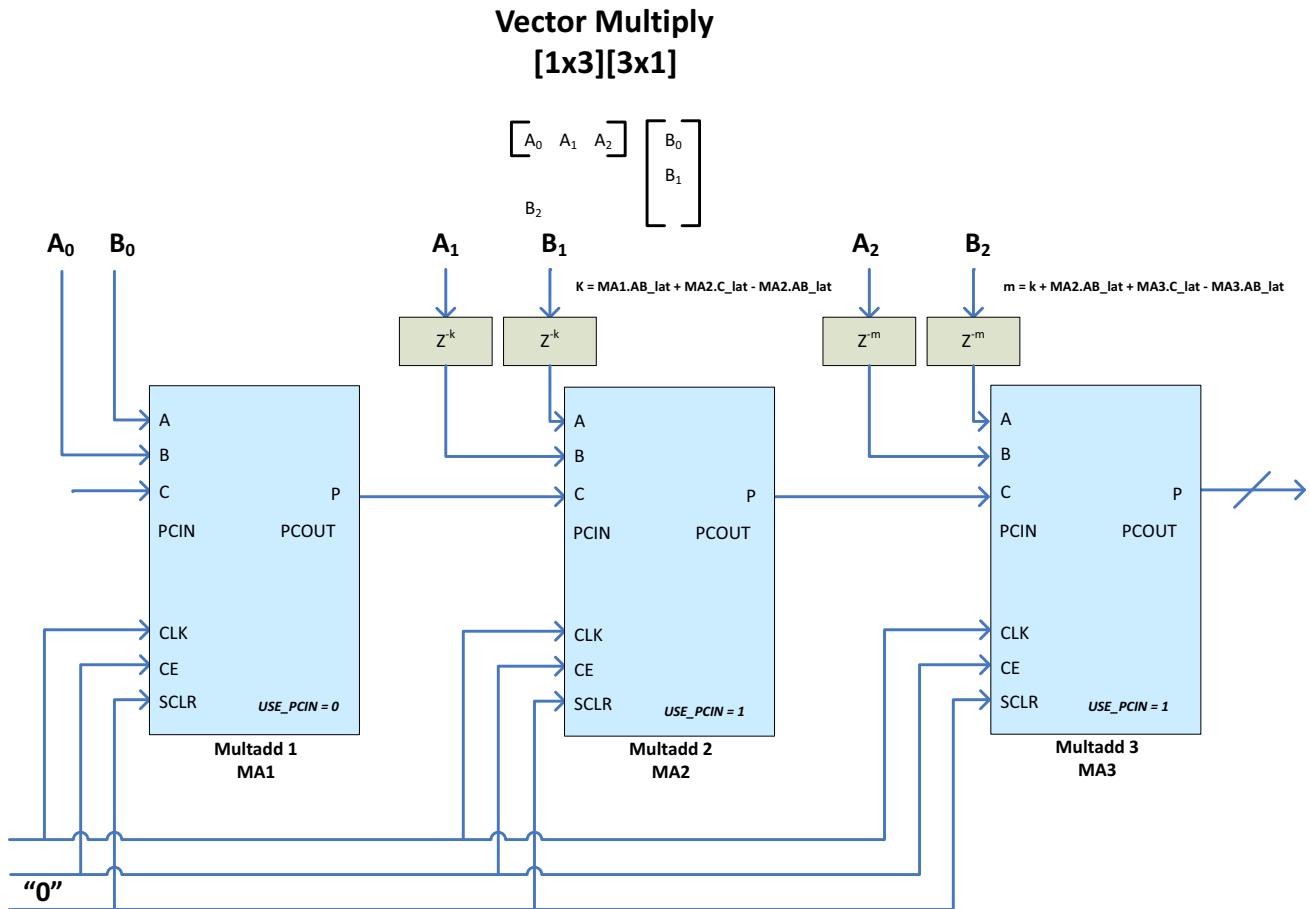


Figure 3-5: Vector Multiply - Multiple DSP Slice Implementation

Clocking

The Multiply Adder core uses a single clock called `clk`. All inputs, outputs, and internal registers are subject to this clock.

Resets

The Multiply Adder core uses a single, optional, active high reset called `sclr`. This global synchronous reset resets all registers in the core. All data in transit through the core is lost when `sclr` is asserted.

Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP integrator* (UG994) [Ref 1]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 3]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 4]

Customizing and Generating the Core

This section includes information about using Xilinx tools to customize and generate the core in the Vivado® Design Suite.

If you are customizing and generating the core in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP integrator* (UG994) [Ref 1] for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl Console.

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 3].

Note: Figures in this chapter are illustrations of the Vivado IDE. The layout depicted here might vary from the current version.

Core Parameters

The core parameters for this module are described below:

- **Component Name:** The name of the core component to be instantiated. The name must begin with a letter and be composed of the following characters: a to z, 0 to 9, and "_".
- **A Input Width:** Sets the width of the Port A (multiplier operand 1) input. The valid range is 1 to 52 unsigned and 2 to 53 signed. The default value is 18. This value is set automatically in IP integrator and can be overridden manually.
- **B Input Width:** Sets the width of the Port B (multiplier operand 2) input. The valid range is 1 to 52 unsigned and 2 to 53 signed. The default value is 18. This value is set automatically in IP integrator and can be overridden manually.
- **C Input Width:** Sets the width of the Port C (add/subtract operand 2) input. The valid range is 1 to 105 unsigned and 2 to 106 signed. The default value is 48. This value is set automatically in IP integrator and can be overridden manually.
- **A Input Type:** Sets the type of the Port A data. 0 = signed, 1 = unsigned. The default value is 0. This value is set automatically in IP integrator and can be overridden manually.
- **B Input Type:** Sets the type of the Port B data. 0 = signed, 1 = unsigned. The default value is 0. This value is set automatically in IP integrator and can be overridden manually.
- **C Input Type:** Sets the type of the Port C data. 0 = signed, 1 = unsigned. The default value is 0. This value is set automatically in IP integrator and can be overridden manually.
- **Output MSB:** Output MSB. The default value is 47. See [Data Alignment](#) for more information.
- **Output LSB:** Output LSB. The default value is 0. See [Data Alignment](#) for more information.
- **Use PCIN:** When this parameter is set to 1, the PCIN port is used. The PCIN port is the cascade input port for an adder/subtractor operand. When set to 0, the PCIN port is ignored. When set to 1, **C Input Width** is limited to 48 bits. The default value is 0.
- **Sync Control CE Priority:** This parameter controls whether or not the SCLR input is qualified by CE. When **Sync Control CE Priority** = 0, SCLR overrides the CE signal. When **Sync Control CE Priority** = 1, SCLR has an effect only when CE is high. The default value is 0.
- **A:B - P Latency:** Latency from the A and B Ports to the output port P. Valid values are: -1, 0; See the section, [Pipelined Operation](#) for more information. The default value is -1.
- **C - P Latency:** Latency from the C or PCIN Port to the output port P. Valid values are: -1, 0; See the section, [Pipelined Operation](#) for more information. The default value is -1.

User Parameters

Table 4-1 shows the relationship between the fields in the Vivado IDE and the User Parameters, which can be viewed in the Tcl Console.

Table 4-1: Vivado IDE Parameter to User Parameter Relationship

Vivado IDE Parameter/Value ⁽¹⁾	User Parameter/Value ⁽¹⁾	Default Value
Input Type (under A)	c_a_type	0
Signed	0	
Unsigned	1	
Input Type (under B)	c_b_type	0
Signed	0	
Unsigned	1	
Input Type (under C)	c_c_type	0
Signed	0	
Unsigned	1	
Use PCIN	c_use_pcin	False
Input Width (under A)	c_a_width	20
Input Width (under B)	c_b_width	20
Input Width (under C)	c_c_width	48
Output MSB	c_out_high	47
Output LSB	c_out_low	0
A:B-P Latency	c_ab_latency	-1
C-P latency	c_c_latency	-1
Synchronous Controls and Clock Enable (CE) Priority	c_ce_overrides_sclr	0

Notes:

- Parameter values are listed when the GUI parameter value differs from the user parameter value. These values are indented below the associated parameter.

Core Use Through Vivado Design Suite

The IP GUI performs error-checking on all input parameters. Resource estimation and latency information are also available.

Several files are produced when a core is generated, and customized instantiation templates for Verilog and VHDL design flows are provided in the .veo and .vho files, respectively. For more information, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Constraining the Core

This section contains information about constraining the core in the Vivado Design Suite.

Required Constraints

This section is not applicable for this IP core.

Device, Package, and Speed Grade Selections

This section is not applicable for this IP core.

Clock Frequencies

This section is not applicable for this IP core.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

Simulation

Starting with Multiply Adder v3.0 (2013.3 version), behavioral simulation models have been replaced with IEEE P1735 Encrypted VHDL. The resulting model is bit and cycle accurate with the final netlist.

For comprehensive information about Vivado simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 4].



IMPORTANT: For cores targeting 7 series or Zynq-7000 devices, UNIFAST libraries are not supported. Xilinx IP is tested and qualified with UNISIM libraries only.

Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Upgrading

This appendix contains information about migrating a design from the ISE® Design Suite to the Vivado® Design Suite, and for upgrading to a more recent version of the IP core. For customers upgrading in the Vivado Design Suite, important details (where applicable) about any port changes and other impact to user logic are included.

Migrating to the Vivado Design Suite

For information about migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* (UG911) [\[Ref 5\]](#).

Multiply Adder core version 3.0 is backwards compatible with earlier versions. There are no changes to parameters, parameter values, ports, or functionality. Performance may vary slightly in different versions of the Vivado Design Suite.

Upgrading in the Vivado Design Suite

This section provides information about any changes to the user logic or port designations that take place when you upgrade to a more current version of this IP core in the Vivado Design Suite.

Multiply Adder core version 3.0 is backwards compatible with earlier versions. There are no changes to parameters, parameter values, ports, or functionality. Performance may vary slightly in different versions of the Vivado Design Suite.

Debugging

This appendix includes details about resources available on the Xilinx[®] Support website and debugging tools.

Finding Help on Xilinx.com

To help in the design and debug process when using the Multiply Adder, the [Xilinx Support web page](#) contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

Documentation

This product guide is the main document associated with the Multiply Adder. This guide, along with documentation related to all products that aid in the design process, can be found on the [Xilinx Support web page](#) or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the [Downloads page](#). For more information about this tool and the features available, open the online help after installation.

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core can be located by using the Search Support box on the main [Xilinx support web page](#). To maximize your search results, use proper keywords such as

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Master Answer Record for the Multiply Adder

AR [54507](#)

Technical Support

Xilinx provides technical support in the [Xilinx Support web page](#) for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support, navigate to the [Xilinx Support web page](#).

Debug Tools

There are many tools available to address Multiply Adder design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx devices.

The Vivado logic analyzer is used with the logic debug IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [\[Ref 6\]](#).

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see [Xilinx Support](#).

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado® IDE, select **Help > Documentation and Tutorials**.
- On Windows, select **Start > All Programs > Xilinx Design Tools > DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the [Design Hubs](#) page.

Note: For more information on Documentation Navigator, see the [Documentation Navigator](#) page on the Xilinx website.

References

These documents provide supplemental material useful with this product guide:

1. *Vivado® Design Suite User Guide: Designing IP Subsystems using IP Integrator* ([UG994](#))
2. *Vivado Design Suite User Guide: Designing with IP* ([UG896](#))

3. *Vivado Design Suite User Guide: Getting Started* ([UG910](#))
4. *Vivado Design Suite User Guide: Logic Simulation* ([UG900](#))
5. *ISE® to Vivado Design Suite Migration Guide* ([UG911](#))
6. *Vivado Design Suite User Guide: Programming and Debugging* ([UG908](#))
7. *Vivado Design Suite User Guide - Implementation* ([UG904](#))
8. Multiply Adder [product page](#)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
02/04/2021	3.0	Added Versal ACAP support.
11/18/2015	3.0	Added support for UltraScale+ families.
06/04/2014	3.0	Initial Xilinx release. Based on DS717.

Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <https://www.xilinx.com/legal.htm#tos>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <https://www.xilinx.com/legal.htm#tos>.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

© Copyright 2015–2021 Xilinx, Inc. Xilinx, the Xilinx logo, Alveo, Artix, Kintex, Spartan, Versal, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.