LogiCORE IP Multiply Adder v3.0

DS717 April 02, 2014 **Product Specification**

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 twos 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)	Zynq®-7000, 7 Series, UltraScale™ Architecture
Supported User Interfaces	N/A
Provided with Core	
Documentation	Product Specification
Design Files	Netlist
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	Encrypted VHDL
Tested Design Tools(2)	
Design Entry Tools	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide
Synthesis Tools	Vivado Synthesis
Support	
Provided by Xilinx, Inc.	

Notes:

1. For a complete listing of supported devices, see the Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Pinout

Signal names for the core symbol are shown in Figure 1 and described in Table 1.

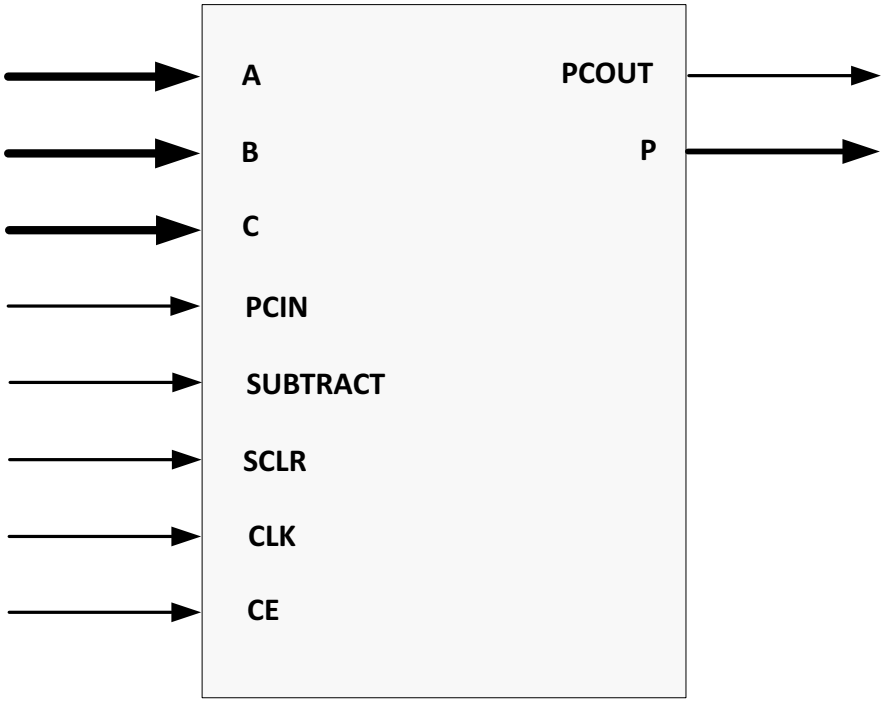

Figure 1: **Core Symbol**

Table 1: **Core Signal Pinout**

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

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

GUI 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.
- **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.
- **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.
- **A Input Type**: Sets the type of the Port A data. 0 = signed, 1 = unsigned. The default value is 0.
- **B Input Type**: Sets the type of the Port B data. 0 = signed, 1 = unsigned. The default value is 0.
- **C Input Type**: Sets the type of the Port C data. 0 = signed, 1 = unsigned. The default value is 0.
- **Output MSB**: Output MSB. The default value is 47. See the section, Data Alignment for more information.
- **Output LSB**: Output LSB. The default value is 0. See the section, 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/subtracter 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 2 shows the relationship between the GUI fields in the Vivado IDE and the User Parameters, which can be viewed in the Tcl console.

Table 2: **GUI Parameter to User Parameter Relationship**

GUI Parameter/Value[1]	User Parameter/Value[1]	Default Value
Input Type (under A)	c_a_type	
Signed	0	0
Unsigned	1	
Input Type (under B)	c_b_type	
Signed	0	0
Unsigned	1	
Input Type (under C)	c_c_type	
Signed	0	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

Table 2: **GUI Parameter to User Parameter Relationship *(Cont'd)***

GUI Parameter/Value[1]	User Parameter/Value[1]	Default Value
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

1. 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 documentation [Ref 1].

Simulation Models

Starting with Multipy 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 more information on simulation, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 1].

Pipelined Operation

The Multiply Adder 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 2.

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.

One other factor that effects the latency is the availability and use of the cascaded PCIN port. Figure 2 shows the placement of registers inside the DSP slice implementation of the Multiply Adder in its pipelined configurations.

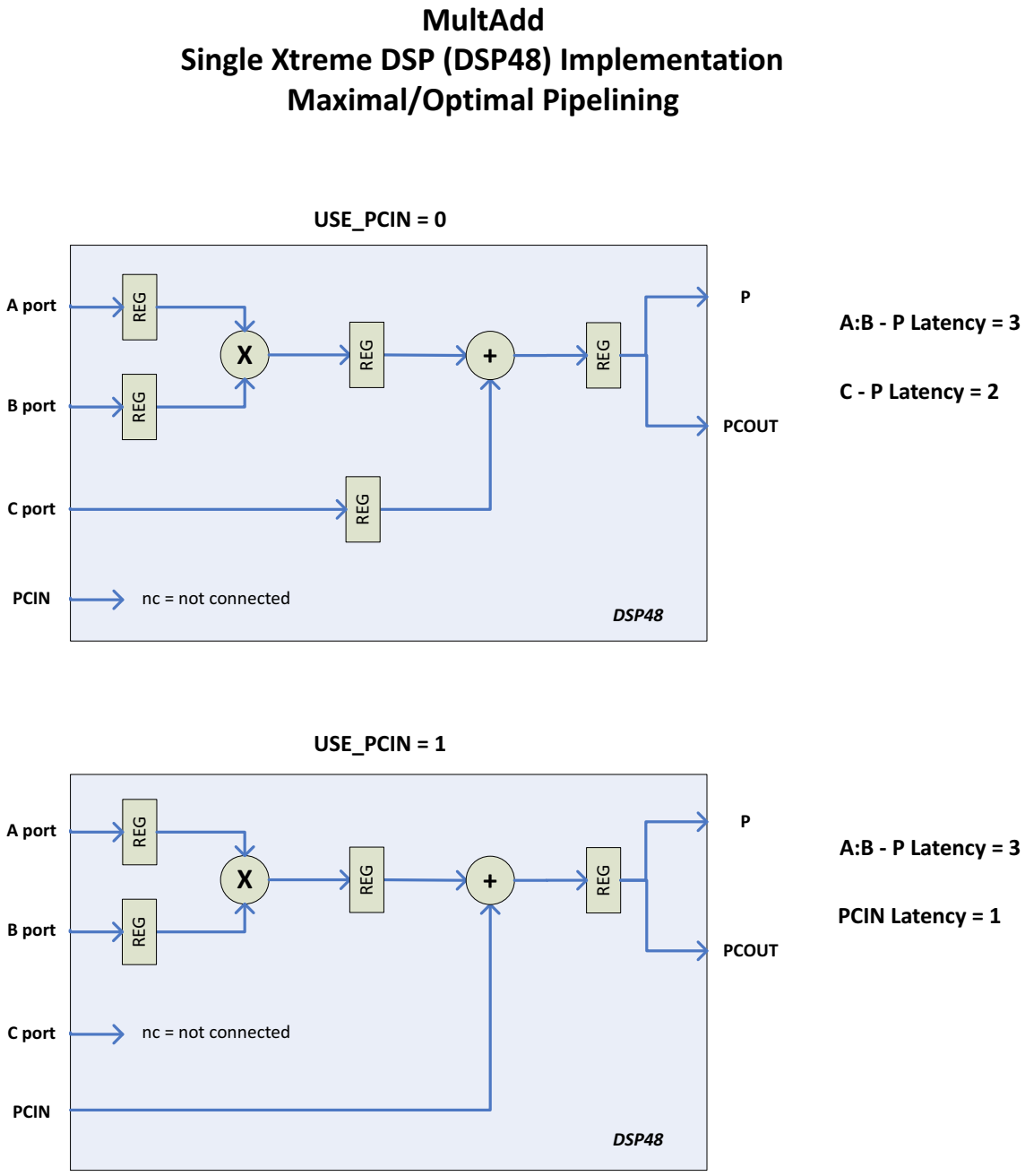

Figure 2: **Single DSP Slice Implementation**

Figure 3 shows the Multiply Adder configured using multiple DSP slices. Internally, the Multiplier and the Adder/Subtracter 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/Subtracter 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.

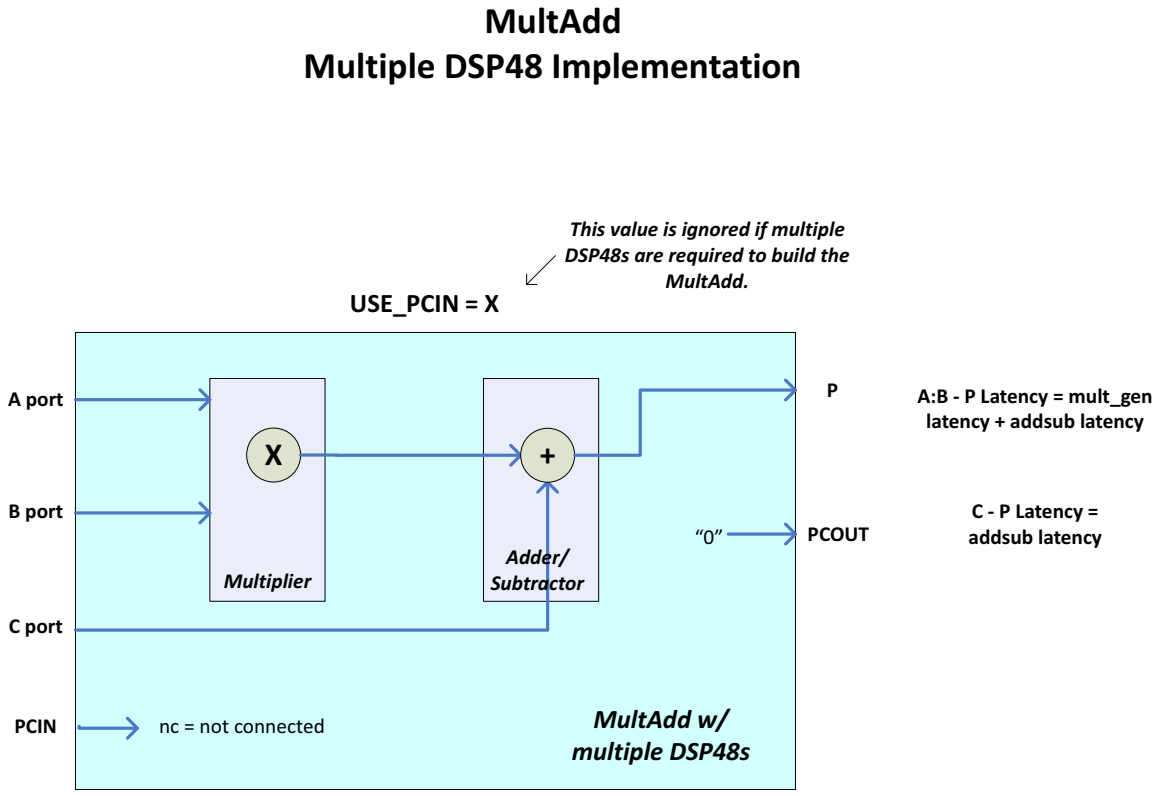

Figure 3: **Multiple DSP Slice Implementation**

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

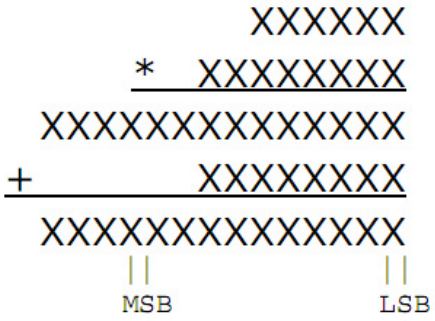

Figure 4: **Data Alignment Example**

Vector Multiply Example

Figure 5 shows a simple vector multiply and the necessary bit staggering required to line up inputs to the second and third MultAdd cores. The following example works for all supported bitwidths, that is, single or multiple DSP slice implementations of the Multiply Adder. If the bitwidths of the core requires multiple DSP slice implementations, then **Use PCIN** is disallowed and the result is provided through P only.

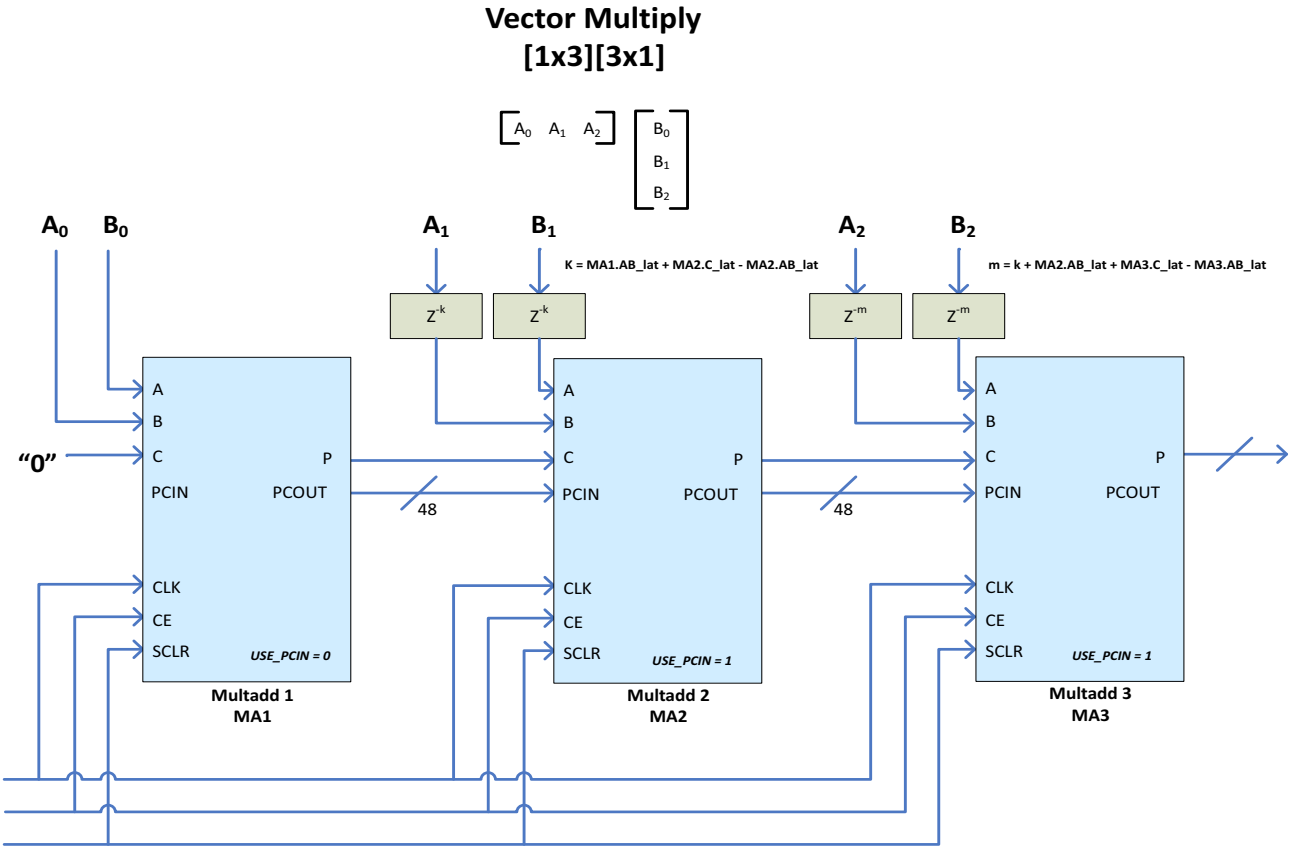

Figure 5: **Vector Multiply Implementation**

Resource Utilization

See the Multiply Adder core product web page documentation links to view the resource utilization numbers for the core.

Performance

See the Multiply Adder core product web page to find performance information for the core.

Support

Xilinx provides technical support at www.xilinx.com/support for this LogiCORE IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Known Issues

The master answer record for the Multiply Adder is AR 54507.

Ordering Information

This LogiCORE IP module is included at no additional cost with the Vivado Design Suite and is provided under the terms of the Xilinx End User License Agreement. To generate the core, use the Vivado IP catalog, which is part of the Vivado Design Suite.

Contact your local Xilinx sales representative for pricing and availability of additional Xilinx LogiCORE modules and software. Information about additional Xilinx LogiCORE modules is available on the Xilinx IP Center.

References

1. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
2. *Vivado Design Suite User Guide: Designing with IP* (UG896)
3. Multiply Adder product page

Revision History

Date	Version	Revision
04/04/2014	3.0	Updated Resource Utilization and Performance Information.
12/18/2013	3.0	Updated to indicate UltraScale™ architecture support.
10/02/2013	3.0	Updated IP Facts table and Simulation Models information.
03/20/2013	3.0	Updated for core v3.0, and Vivado Design Suite-only support.
03/01/2011	2.1	Added support for Virtex-7 and Kintex-7.
04/24/2009	2.0	First customer release of Multiply Adder core.

Notice of Disclaimer