

Accumulator v12.0

LogiCORE IP Product Guide

Vivado Design Suite

PG119 November 18, 2015

Table of Contents

IP Facts

Chapter 1: Overview

Feature Summary	5
Applications	5
Licensing and Ordering Information	5

Chapter 2: Product Specification

Resource Utilization	6
Performance	6
Port Descriptions	6

Chapter 3: Designing with the Core

General Design Guidelines	8
Clocking	8
Resets	9

Chapter 4: Design Flow Steps

Customizing and Generating the Core	10
Constraining the Core	13
Simulation	13
Synthesis and Implementation	14

Chapter 5: Example Design

Chapter 6: Test Bench

Appendix A: Migrating and Upgrading

Migrating to the Vivado Design Suite	17
Upgrading in the Vivado Design Suite	17

Appendix B: Debugging

Finding Help on Xilinx.com	19
----------------------------------	----

Vivado Design Suite Debug Feature 20

Appendix C: Additional Resources and Legal Notices

Xilinx Resources 21
References 21
Revision History 22
Please Read: Important Legal Notices 22

Introduction

The Xilinx LogiCORE™ IP Accumulator core provides LUT and single DSP48 slice accumulation implementations. The Accumulator module can generate adder-based, subtracter-based and adder/subtracter-based accumulators operating on signed or unsigned data. The function can be implemented in a single DSP48 slice or LUTs (but currently not a hybrid of both). Pipelining is available for both implementations.

Features

- Generates add, subtract, and add/subtract-based accumulators
- Supports two's complement signed and unsigned operations
- Supports fabric implementation outputs up to 256 bits wide
- Supports DSP48 slice implementation outputs up to 48 bits wide (max width varies with device family)
- Supports pipelining (automatic and manual)
- User-programmable feedback scaling for fabric implementations
- Optional carry output
- Optional clock enable and sclr
- Optional Bypass (Load) capability

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family ⁽¹⁾	UltraScale+™ Families UltraScale™ Architecture Zynq®-7000 All Programmable SoC 7 Series
Supported User Interfaces	N/A
Resources	Performance and Resource Utilization web page
Provided with Core	
Design Files	Encrypted RTL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	N/A
Simulation Model	Encrypted VHDL
Supported S/W Driver	N/A
Tested Design Flows⁽²⁾	
Design Entry	Vivado® Design Suite System Generator for DSP Vivado
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide .
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx at the Xilinx Support web page	

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](#).

Overview

Feature Summary

The Accumulator core implements area-efficient, high-performance add, subtract and add-subtract accumulators. The core can be customized to use either fabric logic or a DSP48 slice to construct the accumulator.

Applications

The Accumulator core can be used to implement fixed-point accumulators in a wide range of applications, such as phase accumulation for a Numerically Controlled Oscillator (NCO).

Licensing and Ordering Information

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 this and other Xilinx LogiCORE IP modules is available at the [Xilinx Intellectual Property](#) page. For information about 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

Pinout

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

[Table 2-1](#) shows the `SSET` and `SINIT` pins which appear only on fabric implementations. The DSP48 slice implementations do not support `SSET` and `SINIT`.

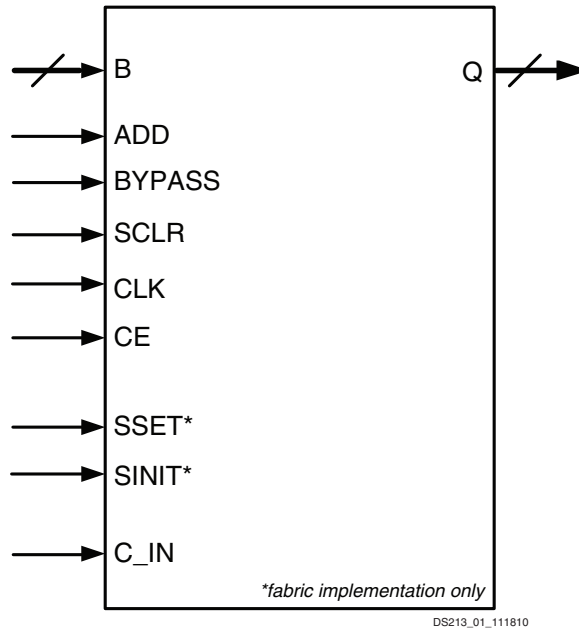


Figure 2-1: Core Symbol

Table 2-1: Core Signal Pinout

Name	Direction	Description
B[M:0]	Input	Input bus
ADD	Input	Controls operation performed by Adder/Subtractor-based accumulator (High = Addition, Low = Subtraction)
Q[P:0]	Output	Output bus
BYPASS	Input	Enables the value on port B to bypass the accumulator logic and appear directly on the output register (optionally active-Low)
CE	Input	Active-High Clock Enable
CLK	Input	Clock signal: rising edge
SCLR	Input	Synchronous Clear: forces the output to a Low state when driven High
SINIT ⁽¹⁾	Input	Synchronous Initialize: forces outputs to user-defined state when driven High
SSET ⁽¹⁾	Input	Synchronous Set: forces the output to a High state when driven High
C_IN	Input	Carry Input

1. Available only for fabric implementations.

Designing with the Core

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

General Design Guidelines

Pipelined Operation

The Accumulator module can be optionally pipelined to improve speed. The pipelined operation is controlled by the latency parameters.

- Set **Latency Configuration** to Automatic to achieve optimal pipelining for maximum speed.
- Set **Latency Configuration** to Manual to allow a valid number of pipeline stages to be entered in the **Latency** parameter.
- For **Latency** = 1, only output registers are present.
- For **Latency** = 2, output and input registers are present.

After power-up or reset, the pipelined module takes many clock cycles, specified by the latency control, for the outputs to become valid.

If **Bypass** is requested on a pipelined module, the `BYPASS` input appears on the output after the number of clock cycles, specified by the latency control.

Clocking

The core requires a single clock, `CLK`, and is active-High triggered.

If selected, the active-High clock enable, `CE`, stalls all core processes when deasserted.

Resets

The core has a single, active-High synchronous reset, `SCLR`. Asserting `SCLR` for a single cycle resets all registers in the core.

The priority of `SCLR` and `CE` pins can be selected when customizing the core.

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) [\[Ref 5\]](#)

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, you can run the **validate_bd_design** command in the Tcl Console.

Vivado Integrated Design Environment

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) [\[Ref 3\]](#)).

Core Parameters

The Accumulator core Vivado IDE provides fields to set the parameter values for the required instantiation. This section provides a description of each field.

- **Implement using:** Sets the implementation type: Fabric or DSP48 Slice.
 - **Input Width:** Sets the width of the input port. In IP integrator, this parameter is auto-updated.
 - **Input Type:** Sets the type of the Port B data: Signed or Unsigned. In IP integrator, this parameter is auto-updated.
 - **Output Width:** Sets the output width.
 - **Accumulation Mode:** Sets the mode of operation of the module. If an adder/subtractor is specified, the `ADD` pin sets the mode of operation.
 - **Carry In:** When set to `TRUE`, this generic creates the `C_IN` port, which is the synchronous carry-in to the accumulator.
 - **Bypass:** When set to `TRUE`, creates a `BYPASS` pin. Activating the `BYPASS` pin sets the output as the value given on Port B. This functionality is used for creating loadable accumulators.
 - **Bypass Sense:** When set to active-Low, the `BYPASS` pin is active-Low. `BYPASS` is the only pin that has a parameter to control its active sense. This is because an historical implementation made significant speed gains with an active-Low `BYPASS`, instead of active-High `BYPASS`. This is no longer necessarily the case, because sometimes active-High is as or more efficient. The details depend on the exact set of parameters.
 - **Accumulator Scaling:** Sets the scaling factor used for the feedback path to Port B on fabric implementations. The value represents the number of low-order bits that are discarded in the feedback process.
 - **Clock Enable:** When set to `TRUE`, the module is generated with a clock enable input.
 - **Power-on Reset Init value:** Specifies in binary the value the output initializes to during power-up reset.
 - **Synchronous Clear:** Specifies if an `SCLR` pin is to be included.
 - **Synchronous Set:** Specifies if an `SSET` pin is to be included. `SSET` pin is not valid in DSP48 implementations. See **Sync Set and Clear (Reset) Priority** for `SCLR/SSET` priorities.
 - **Synchronous Init:** Specifies if an `SINIT` pin is to be included which, when asserted, synchronously sets the output value to the value defined by **Init Value**.
- Note:** If `SINIT` is present, then neither `SSET` nor `SCLR` can be present. The `SINIT` pin is not valid in DSP48 implementations.
- **Init Value:** Specifies, in hex, the value that the output initializes to when `SINIT` is asserted. Ignored if **Synchronous Init** = 0.

- Synchronous Controls and Clock Enable (CE) Priority:** This parameter controls whether or not the SCLR (and if fabric: SSET and SINIT) inputs are qualified by CE. When set to Sync_Overrides_CE, the synchronous controls override the CE signal. When set to CE_Overrides_Sync, the control signals have an effect only when CE is High.

Note: On the fabric primitives, the SCLR and SSET controls override CE, so choosing CE_Overrides_Sync generally results in extra logic.
- Sync Set and Clear (Reset) Priority:** Controls the relative priority of SCLR and SSET. When set to Reset_Overrides_Set, SCLR overrides SSET. The default is Reset_Overrides_Set, as this is the way the primitives are arranged. Making SSET take priority requires extra logic.
- Latency Configuration:** Automatic sets optimal latency for maximum speed; Manual allows you to set Latency to one of the allowed values.
- Latency:** Value used for latency when Latency Configuration is set to Manual. See the section, [Pipelined Operation](#), for more information.

User Parameters

Table 4-1 shows the relationship between the GUI fields in the Vivado IDE (described in [Vivado Integrated Design Environment](#)) and the User Parameters (which can be viewed in the Tcl console).

Table 4-1: GUI Parameter to User Parameter Relationship

GUI Field Label	User Parameter	Default Value
Implement using	implementation	Fabric
Input Type	input_type	Signed
Input Width	input_width	16
Output Width	output_width	16
Accumulation Mode	accum_mode	Add
Latency Configuration	latency_configuration	Manual
Latency	latency	1
Accumulator Scaling	scale	0
Clock Enable(CE)	ce	False
Carry In(C_IN)	c_in	False
Synchronous Clear(SCLR)	sclr	False
Synchronous Set (SSET)	sset	False
Synchronous Init (SINIT)	sinit	False
Init Value (Hex)	sinit_value	False
Bypass	bypass	True
Bypass Sense	bypass_sense	Active_High

Table 4-1: GUI Parameter to User Parameter Relationship (Cont'd)

GUI Field Label	User Parameter	Default Value
Synchronous Set and Clear(Reset) Priority	syncctrlpriority	Reset_Overrides_Set
Synchronous Controls and Clock Enable (CE) Priority	sync_ce_priority	Sync_Overrides_CE

Core Use through the Vivado Design Suite

The Vivado IDE performs error-checking on all input parameters. Resource estimation and latency information is 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 detailed instructions, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Core Use through System Generator

The Accumulator core is available through Xilinx System Generator for DSP, a DSP design tool that enables the use of the model-based design environment Simulink® software for FPGA design. The Accumulator core is one of the DSP building blocks provided in the Xilinx DSP blockset for the Simulink software. The core can be found in the Xilinx Blockset in the Math section. The block is called "Accumulator". See the *System Generator for DSP User Guide* (UG640) [Ref 4] for more information.

Output Generation

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

Constraining the Core

There are no constraints associated with this core.

Simulation

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 5].

Synthesis and Implementation

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

Example Design

No example design is provided for this core.

Test Bench

No demonstration test bench is provided for this core.

Migrating and Upgrading

This appendix contains information about migrating a design from ISE® 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

Updating from Accumulator v9.0 and Later

The Vivado Design Suite IP update feature can be used to update an existing Accumulator to v12.0 of the core. The core can then be regenerated to create a new netlist. See the *ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 6] for more information on this feature.

Updating from Versions Prior to Accumulator v9.0

It is not currently possible to automatically update versions of the Accumulator core prior to v9.0. Some features and configurations might be unavailable in Accumulator v12.0, and some port names might differ between versions.



RECOMMENDED: Use the Accumulator v12.0 Vivado IDE in the Vivado Design Suite to customize a new core.

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.

Parameter Changes

There are no parameter changes in Accumulator v12.0 compared to v9.0 and later.

Port Changes

There are no port changes in Accumulator v12.0 compared to v9.0 and later.

Functionality Changes

There are no changes in functionality in Accumulator v12.0 compared to v9.0 and later.

Simulation

Starting with Accumulator v12.0 (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 5].

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 Accumulator, 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 Accumulator core. 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 messages
- Summary of the issue encountered

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

Master Answer Record for the **Accumulator**

AR: [54492](#)

Technical Support

Xilinx provides technical support at 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 IP core 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 (ILA) and virtual I/O (VIO) 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 represents the functionality in the Vivado Integrated Design Environment (IDE) that is used for logic debugging and validation of a design running in Xilinx devices in hardware.

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

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

See *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [\[Ref 7\]](#).

Additional Resources and Legal Notices

Xilinx Resources

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

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. *System Generator for DSP User Guide* ([UG640](#))
5. *Vivado Design Suite User Guide: Logic Simulation* ([UG900](#))
6. *ISE® to ISE to Vivado Design Suite Migration Guide* ([UG911](#))
7. *Vivado Design Suite User Guide: Programming and Debugging* ([UG908](#))

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
11/18/2015	12.0	Added support for UltraScale+ families.
04/02/2014	12.0	Added link to resource utilization information.
12/18/2013	12.0	Updated IP Facts table to indicate support for UltraScale™ Architecture.
10/02/2013	12.0	Minor updates to IP Facts table and Migrating appendix. Document version number advanced to match the core version number.
03/20/2013	1.0	Initial Xilinx release as a product guide. Replaces <i>LogiCORE IP Accumulator Data Sheet</i> (DS213).

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 <http://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 <http://www.xilinx.com/legal.htm#tos>.

© Copyright 2013-2015 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, 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.