RAM-Based Shift Register v12.0

LogiCORE IP Product Guide

Vivado Design Suite

PG122 February 4, 2021

XILINX

Table of Contents

Appendix B: Debugging

Appendix C: Additional Resources and Legal Notices

Introduction

The Xilinx® LogiCORE™ IP RAM-based Shift Register core provides a very efficient multi-bit wide shift register for use in FIFO-like applications or as a delay line. Fixed-length shift registers and variable-length shift registers can be created.

Features

- Generates fast, compact, FIFO-style shift registers or delay lines using the SRL16/SRL32 mode of the slice LUTs
- User options to create fixed-length or variable-length shift registers
- Speed or resource optimization for variable length shift registers
- Optional output register with clock enable and synchronous controls for variable length shift registers

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	UltraScale+™ Families UltraScale™ Architecture Versal™ ACAP Zynq®-7000 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
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Vivado Synthesis
Support	
Release Notes and Known Issues	Master Answer Record: 54508
All Vivado IP Change Logs	Master Vivado IP Change Logs: 72775
Xilinx Support web page	

Notes:

1. For a complete listing 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

Feature Summary

The RAM-based Shift Register core implements area-efficient, high-performance first-in-first-out (FIFO)-style buffers and delay lines using the SRL16 and SRL32 features of the FPGA fabric.

Applications

The buffers created by the core can be used in a wide variety of applications, such as:

- General-purpose pipeline balancing delays
- Temporary buffers in a data pipeline
- A building block for custom FPGA fabric-based FIFOs

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

Performance

For details about performance, visit the Performance and Resource Utilization web page.

Resource Utilization

For details about resource utilization, visit the Performance and Resource Utilization web page.

Port Descriptions

Figure 2-1 and Table 2-1 illustrate and define the schematic symbol signal names.

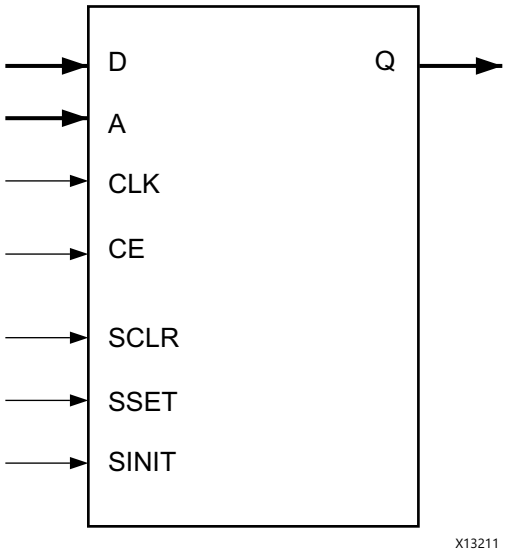

Figure 2-1: **Core Schematic Symbol**

Table 2-1: **Core Signal Pinout**

Signal	I/O	Description
D[N:0]	I	Parallel Data Input
A[M:0]	I	Address Input (required on variable-length modules only)
CLK	I	Rising-edge Clock Input
CE	I	Optional active-High Clock Enable
SCLR	I	Optional Synchronous Clear. Forces outputs to a low state when driven high
SSET	I	Optional Synchronous Set. Forces outputs to a high state when driven high
SINIT	I	Synchronous Initialize. Forces outputs to a user-defined state when driven high
Q[N:0]	O	Parallel Data Output

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the RAM-based Shift Register core.

General Design Guidelines

This section describes the general design guidelines for the RAM-based Shift Register core.

Power-On Conditions

If **Register Last Bit** has been selected, the final bit of the shift register powers up according to the **Power-on Reset Init Value** value or the register controls that have been selected. The Select RAM elements power up according to the initialization method used; see Initialization Parameters in Chapter 4 for detailed information.

Specifying Memory Contents Using a COE File

The initial contents of the memory can be defined using a text file known as a Coefficient (COE) file. COE files must have a `.coe` extension. A COE file specifies two parameters with the end of each line defined by a semicolon. The two parameters are:

- **memory_initialization_vector**: Each row of memory elements is defined with a binary or hexadecimal number, the equivalent binary value of which represents whether an individual memory element along the width of the row is set to a 1 or a 0. Each row of memory initialization is separated by a comma or white space, up to the depth of the memory. Negative values are not allowed, but the example below shows the robustness of the reader for accepting values with varying formats.
- **memory_initialization_radix**: The radix of the initialization value is specified here, the choice being 2 or 16.

  An example of a COE file is:

```
; Sample Initialization file for a 16x32 RAM-based Shift Register
memory_initialization_radix = 16;
memory_initialization_vector =
23f4 0721 11ff ABe1 0001 1 0A 0
23f4 0721 11ff ABe1 0001 1 0A 0
```

```
23f4 721 11ff ABe1 0001 1 A 0
23f4 721 11ff ABe1 0001 1 A 0;
```

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 the output registers (if present) only. The SRL16/SRL32 primitives cannot be reset.

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 6]

Customizing and Generating the Core

This section includes information about using the Vivado Design Suite and the System Generator for DSP to customize and generate the RAM-based Shift Register core.

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.

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 RAM-based Shift Register core graphical user interface (GUI) provides fields to set the parameter values for the required instantiation. This section provides a description of each GUI field.

- **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 "_".
- **Width**: Specifies the width of the input to the shift register. In IP Integrator, this parameter is auto-updated.
- **Depth**: Specifies the depth of the shift register in bits.
- **Register Type**: Specifies the shift register implementation.
  - Fixed-length: Parallel data is clocked into the shift register and appears at the output **Depth** clock cycles later.
  - Variable-length Lossless: The delay (in number of clocks) that it takes for data to be cycled through from input bus to output bus is defined by the value on the `A[M:0]` (Address) input bus. This module is referred to as lossless because, when the address is changed, the output is always valid.
- **Optimization**: Specifies if the variable-length shift register should be optimized for speed or resources.
  - Resources: the output multiplexer (if required) is created from the MUXF7/F8 dedicated multiplexers where possible. This creates a very compact core with minimal LUT resource usage running at reasonably high clock frequencies. The number of flip-flops used in the design is minimized.
  - Speed: extra flip-flops are used. In the variable-length lossless mode, the outputs of the Select RAM are registered and, when the **Depth** parameter is greater than 256 (Virtex®-7 and Kintex®-7 devices), the multiplexer uses an additional internal pipeline stage. The read latency on the output increases by 1 or 2 cycles when the speed option is used; this value is displayed on the GUI latency information panel. The pipeline registers do not take any synchronous controls (this is left for the final output register, if selected), but can use a clock enable.
  - The Fixed-length shift register is optimized for resources when less than one CLB in length, otherwise it is automatically optimized for speed by inserting additional flip-flops.
- **Power-on Reset Init Value**: Specifies in binary or hex (determined by **Power-on Reset Init Value Radix**) the value to which the `Q` register initializes during power-up reset. The width of this parameter is specified by **Width**. The default value is sixteen zeros.
- **Synchronous Init Value**: If an `SINIT` pin is included and asserted, the **Synchronous Init Value** is applied to the `Q` port. The width of this parameter is specified by **Width**. The value can be input as hex or binary depending on **Synchronous Init Value Radix**.

The default value is sixteen zeros. When the core has Init asserted, the output register is initialized to this value on the next valid clock edge.

- **Clock Enable**: Specifies if the core has a clock enable pin. This control is applied to all shift register elements — when deasserted, the entire delay line is stalled.
- **Register Last Bit**: Specifies if the final bit in the module is to be registered with flip-flops. This improves the clock-to-output of the SRL16/SRL32 elements. For fixed-length modules, this register is accounted for in the depth selection and is always used when configuring through the GUI because performance is improved dramatically. For the variable-length shift registers, selecting this option adds one cycle of latency to the output. Other register control options are not used unless this option is enabled.
- **Set and Clear (Reset) Priority**: Controls the relative priority of SCLR and SSET. The default is Reset_Overrides_Set, because this is also the way the primitives behave, resulting in no extra logic being required.
- **Synchronous Controls (Sync) and Clock Enable (CE) Priority**: This parameter controls whether or not the SSET, SCLR, 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. The parameters are ignored unless **Clock Enable** is true. Note that on the fabric primitives, the SCLR and SSET controls override CE, so choosing CE_Overrides_Sync generally results in extra logic.
- **Synchronous Settings**:
  - **Synchronous Clear**: Specifies if an SCLR pin is to be included.
  - **Synchronous Set**: Specifies if an SSET pin is to be included. See **Sync Control Priority** for SCLR/SSET priorities.
  - **Synchronous Init**: Specifies if an SINIT pin is to be included which, when asserted, synchronously set the Q value to the value defined by **Synchronous Init Value**. Note that if SINIT is present, then neither SSET nor SCLR are present.

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	User Parameter	Default Value
Shift Register Type	shiftregtype	Fixed_Length
Optimization	optgoal	Resources
Register Last Bit	reglastbit	True
Clock Enable(CE)	ce	False
Width	width	16

Table 4-1: **Vivado IDE Parameter to User Parameter Relationship *(Cont'd)***

Vivado IDE Parameter	User Parameter	Default Value
Depth	depth	16
Initialization Options: Radix	defaultdataradix	2
Initialization Options: Default Data	defaultdata	0000000000000000
Use COE File	readmiffile	False
Initialization File	meminitfile	No_coe_file_loaded
Power-On Reset Settings: Radix	asyncinitradix	2
Power-On Reset Settings: Init Value	asyncinitval	0000000000000000
Set(SSET)	sset	False
Clear(SCLR)	sclr	False
Init(SINIT)	sinit	False
Synchronous Settings: Radix	syncinitradix	2
Synchronous Settings: Init Value	syncinitval	0000000000000000
Set and Clear(Reset) Priority	syncctrlpriority	Reset_overrides_set
Synchronous Controls(Sync) and Clock Enable(CE) Priority	cepriority	Sync_overrides_ce

Initialization Parameters

You can define the initial contents of the shift register for Power-on Reset in several ways. Trivial initialization can be achieved by using the **Default Data** parameter, which allows all memory locations in each row (the number of rows being specified by **Width**) to be initialized to the same binary value. Arbitrary initialization can be achieved by reading initialization data from a file. By specifying the relative path to a pre-defined COE file and setting **Use COE File** to true, the COE file contents are read into the design at synthesis time and initialize the memory locations. Any bits not specified in the COE file are set to the **Default Data** value for that memory location.

- **Use COE File**: Specifies if the module should read in a MIF file for initialization purposes.
- **Initialization File**: Specifies the name of the COE file. This parameter is only used if **Use COE File** is set to true.
- **Default Data**: Specifies the default values for initializing each bit of width of the module. This value can be input as hex or binary depending on **Default Data Radix**. The default value is sixteen zeros.

Core Use through Vivado Design Suite

The Vivado IP catalog 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 detailed instructions, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Core Use through System Generator

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

Output Generation

For details, see 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 6].

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].

Example Design

No example design is provided for this core.

Test Bench

No test bench is available for this core.

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 RAM-based Shift Register v9.0 and later

The Vivado Design Suite IP update feature can be used to update an existing RAM-based Shift Register to version 12.0 of the core. The core can then be regenerated to create a new netlist. See the *Vivado Design Suite User Guide: Designing with IP (UG896)* [Ref 2] for more information on this feature.

Updating from versions prior to RAM-based Shift Register v9.0

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

RECOMMENDED: *Use the RAM-based Shift Register v12.0 GUI 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 RAM-based Shift Register v12.0 compared to v9.0 and later.

Port Changes

There are no port changes in RAM-based Shift Register v12.0 compared to v9.0 and later.

Functionality Changes

There are no changes in functionality in RAM-based Shift Register v12.0 compared to v9.0 and later.

Simulation

Starting with RAM Based Shift Register 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 6].

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 RAM-based Shift Register, 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 RAM-based Shift Register. 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 are listed below, and 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 **RAM-based Shift Register**

AR 54508

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 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 allow 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 to interact with the logic debug LogiCORE™ IP cores, including:

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

Appendix C

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. *System Generator for DSP User Guide* (UG640)
5. *ISE to Vivado Design Suite Migration Methodology Guide* (UG911)
6. *Vivado Design Suite User Guide: Logic Simulation* (UG900)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
02/04/2021	12.0	Added Versal ACAP support.
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	• Added UltraScale™ architecture support information. • Added Simulation, Synthesis, Example Design and Test Bench chapters. • Updated Migrating appendix.
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 release as a product guide. This document replaces *LogiCORE IP RAM-based Shift Register (DS228)*.

Please Read: Important Legal Notices

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.