LogiCORE IP RAM-based Shift Register v11.0

DS228 March 1, 2011 **Product Specification**

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

- Drop-in module for Virtex®-7 and Kintex™-7, Virtex-6, Virtex-5, Virtex-4, Spartan®-6, Spartan-3/XA, Spartan-3E/XA, Spartan-3A/3AN/3A DSP/XA FPGAs
- 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
- For use with Xilinx CORE Generator™ and Xilinx System Generator for DSP 13.1

LogiCORE IP Facts Table					
Core Specifics					
Supported Device Family[1]	Virtex-7 and Kintex-7 Virtex-6, Virtex-5, Virtex-4, Spartan-6, Spartan-3/XA, Spartan-3E/XA, Spartan-3A/3AN/3ADSP/XA				
Supported User Interfaces	Not Applicable				
	Resources[2]				**Frequency**
Configuration	**LUTs**	**FFs**	**DSP Slices**	**Block RAMs**	**Max. Freq.**
Virtex-5, variable length, 32-deep, 16-wide shift register	16	16	0	0	452 MHz
Provided with Core					
Documentation	Product Specification				
Design Files	Netlist				
Example Design	Not Provided				
Test Bench	Not Provided				
Constraints File	Not Applicable				
Simulation Model	VHDL behavioral model in the xilinxcorelib library VHDL UniSim structural model Verilog UniSim structural model				
Tested Design Tools					
Design Entry Tools	CORE Generator tool 13.1 System Generator for DSP 13.1				
Simulation	Mentor Graphics ModelSim 6.6d Cadence Incisive Enterprise Simulator (IES) 10.2 Synopsys VCS and VCS MX 2010.06 ISIM 13.1				
Synthesis Tools	N/A				
Support					
Provided by Xilinx, Inc.					

1. For a complete listing of supported devices, see the release notes for this core.
2. For more complete device performance numbers, see Performance and Resource Utilization, page 6.

Pinout

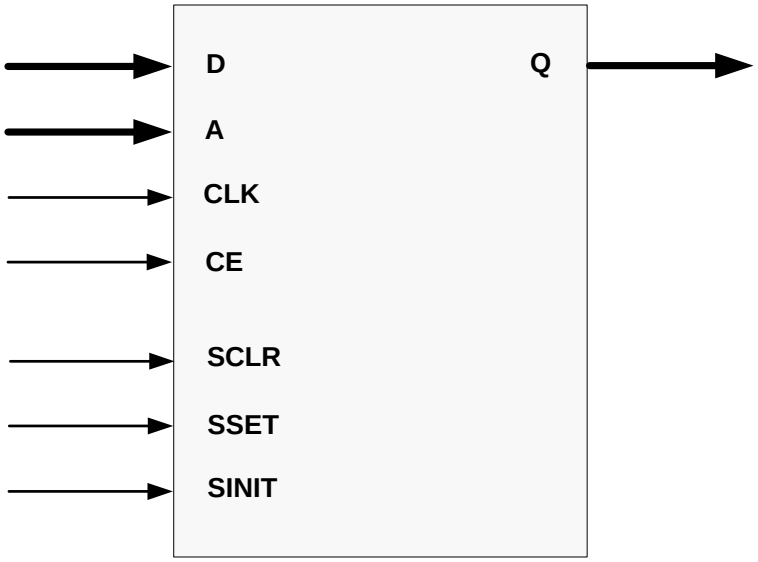

Figure 1: **Core Schematic Symbol**

Figure 1 and Table 1 illustrate and define the schematic symbol signal names. All control inputs are active high. Should an active low input be required for a specific control pin, an inverter must be placed in the path to the pin and will be absorbed appropriately during synthesis and/or mapping.

Table 1: **Core Signal Pinout**

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

CORE Generator Graphical User Interface Parameters

The CORE generator GUI 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 "_".
- **Width**: Specifies the width of the input to the shift register.
- **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) will be created from the MUXF5/F6/F7 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 128 (256 for Virtex-5, Virtex-6, Virtex-7 and Kintex-7), 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 1 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 the Q register will initialize to 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 will be initialized to this value on the next valid clock edge.
- **Clock Enable**: Specifies if the core has a clock enable pin. This control will be applied to all shift register elements - when de-asserted, the entire delay line will be 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 since performance is improved dramatically. For the variable-length shift registers, selecting this option will add 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, since 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, will synchronously set the Q value to the value defined by **Synchronous Init Value**. Note that if SINIT is present, then neither SSET nor SCLR may be present.

Initialization Parameters

- It is possible for the user to define the initial contents of the shift register for Power-on Reset. This can be achieved in a number of 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 will be read into the design at synthesis time and initialize the memory locations. Any bits not specified in the COE file will be 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.

Table 2 is a cross-reference table from the GUI parameters listed above to the XCO parameter names in the XCO file.

Table 2: **CORE Generator GUI and XCO Parameters**

GUI Name	Default Value	Valid Range	XCO Parameter
Component Name	shift_ram		Component_Name
Register Type	Fixed Length	Fixed Length, Variable Length Lossless	ShiftRegType
Optimization	Resources	Resources, Speed	OptGoal
Width	16	1 to 256	Width
Depth	16	1 to 1024 (variable) 1 to 1088 (fixed)	Depth
Register Last Bit	true		RegLastBit
Clock Enable	false		CE
Synchronous Clear	false		SCLR
Synchronous Set	false		SSET
Synchronous Init	false		SINIT
Set and Clear (Reset) Priority	Reset_Overrides_Set	Reset Overrides Set, Set Overrides Reset	SyncCtrlPriority
Synchronous controls (Sync) and Clock Enable (CE) Priority	Sync_Overrides_CE	Sync Overrides CE, CE Overrides Sync	CE_Priority
Synchronous Init Value Radix	2	2,16	SyncInitRadix
Synchronous Init Value	0		SyncInitVal
Default Data Radix	2	2,16	DefaultDataRadix
Default Data	0		DefaultData
Use COE File	false		ReadMifFile
Initialization File		no_coe_file_loaded	MemInitFile
Power-on Reset Init Value Radix	2	2,16	AsyncInitRadix
Power-on Reset Init Value	0		AsyncInitVal

Core Use through CORE Generator

The CORE Generator 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 detailed instructions, see the CORE Generator software documentation.

Simulation Models

The core has a number of options for simulation models:

- VHDL behavioral model in the xilinxcorelib library
- VHDL UniSim structural model
- Verilog UniSim structural model

Xilinx recommends that simulations utilizing UniSim-based structural models are run using a resolution of 1 ps. Some Xilinx library components require a 1 ps resolution to work properly in either functional or timing simulation. The UniSim-based structural models might produce incorrect results if simulation with a resolution other than 1 ps. See the "Register Transfer Level (RTL) Simulation Using Xilinx Libraries" section in *Synthesis and Simulation Design Guide* for more information. This document is part of the ISE® Software Manuals set available at www.xilinx.com/support/software_manuals.htm.

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 may be found in the Xilinx Blockset in the Memory Blocks section. The block is called "Addressable Shift Register". See the System Generator User Manual for more information.

Migrating to RAM-based Shift Register v11.0 from Earlier Versions

Updating from RAM-based Shift Register v9.0 and later

The CORE Generator core update feature may be used to update an existing RAM-based Shift Register XCO file to version 11.0 of the core. The core may then be regenerated to create a new netlist. See the CORE Generator documentation 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. Xilinx recommends that customers use the RAM-based Shift Register v11.0 GUI to customize a new core. Note that some features and configurations may be unavailable in RAM-based Shift Register v11.0. Also, some port names may differ between versions.

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 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 uses two parameters similar to XCO parameters, but the end of each line is defined with the use of 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;
```

Performance and Resource Utilization

Tables 3 and 4 provide RAM-based Shift Register performance and resource usage for a number of different configurations.

The maximum clock frequency results were obtained by double-registering input and output ports to reduce dependence on I/O placement. The inner level of registers used a separate clock signal to measure the path from the input registers to the first output register through the core.

The resource usage results do not include the above "characterization wrapper" registers and represent the true logic used by the core.

The map options used were: "`map -pr b -ol high`."

The par options used were: "`par -ol high`."

Clock frequency does not take clock jitter into account and should be derated by an amount appropriate to the clock source jitter specification.

The maximum achievable clock frequency and the resource counts may also be affected by other tool options, additional logic in the FPGA device, using a different version of Xilinx tools, and other factors. The Xilinx Xplorer™ script can be used to find the optimal settings. Note also that the measurement of the maximum frequency for these XtremeDSP slice families is artificially limited by the maximum speed of the DSP48A or DSP48E slice.

All characterization was done using the default parameter settings unless otherwise noted.

Table 3: **Fabric RAM-based Shift Register: Virtex-5 (Part = XC5VSX50T-1)**

Description	Fixed Length				Variable Length			
Depth	32	64	128	512	32	64	128	512
Width	16	16	16	16	16	16	16	16
Max Clock Frequency (MHz)	452	452	452	452	452	452	452	310
LUT6-FF pairs	16	32	64	256	16	32	64	272
LUTs	16	32	64	256	16	32	64	272
Flip-flops	16	16	16	64	0	0	0	0
DSP48Es	0	0	0	0	0	0	0	0

Table 4: **Fabric RAM-based Shift Register: Spartan-3A DSP (Part = XC3SD3400A-4)**

Description	Fixed Length				Variable Length			
Depth	32	64	128	512	32	64	128	512
Width	16	16	16	16	16	16	16	16
Max Clock Frequency (MHz)	251	251	251	251	251	236	188	125
LUTs	32	64	128	512	32	64	144	596
Flip-flops	16	16	32	128	0	0	0	0
DSP48As	0	0	0	0	0	0	0	0

Support

Xilinx provides technical support for this LogiCORE 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*.

Refer to the IP Release Notes Guide (XTP025) for further information on this core. There is a link to all the DSP IP and then to each core. For each core, there is a master Answer Record that contains the Release Notes and Known Issues list for each core. The following information is listed for each version of the core:

- New Features
- Bug Fixes
- Known Issues

Ordering Information

This LogiCORE IP module is included at no additional cost with the Xilinx ISE Design Suite software and is provided under the terms of the Xilinx End User License Agreement. Use the CORE Generator software included with the ISE Design Suite to generate the core. For more information, please visit the core page.

Please 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.

Revision History

Date	Version	Description of Revisions
4/28/05	8.0	Updated to Xilinx Tools 7.1i
7/13/06	9.0	Added support for Spartan3a
4/24/09	11.0	Updated RAM-based Shift Register to ISE Tools 11.1, added support for Spartan-6 and Virtex-6 devices and removed asynchronous controls.
03/01/11	11.1	Support added for Virtex-7 and Kintex-7. ISE Design Suite 13.1

Notice of Disclaimer