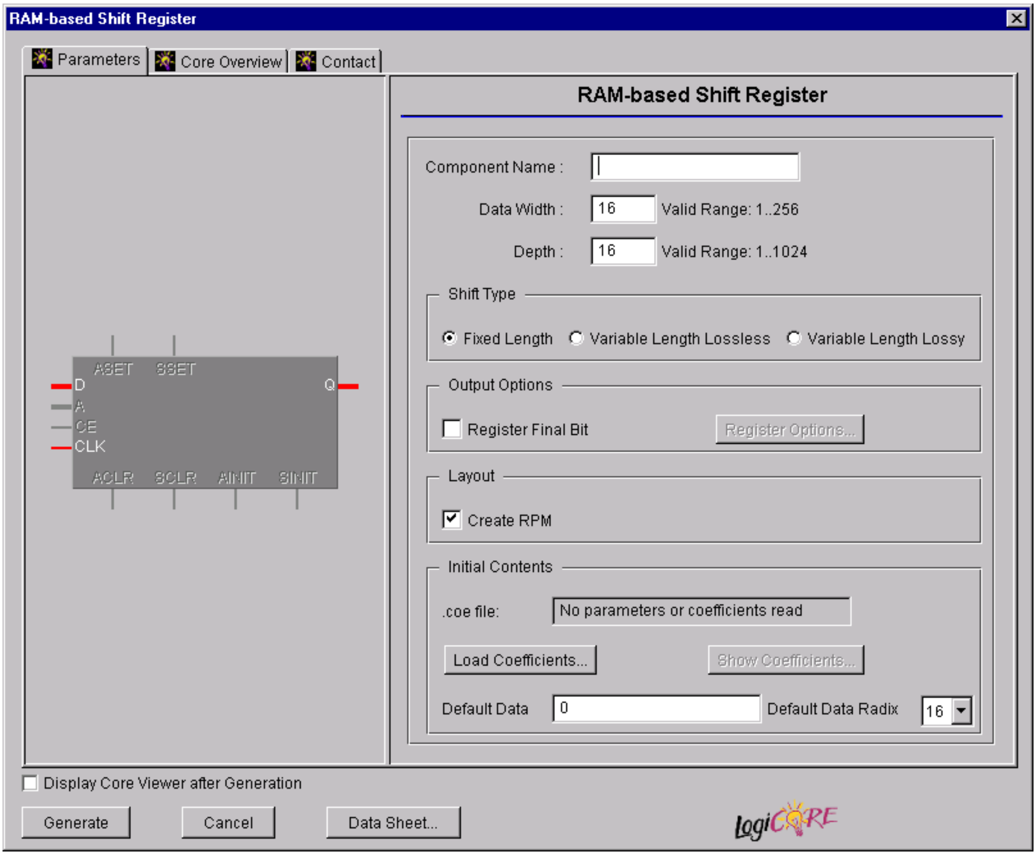
RAM-based Shift Register V3.0

November 3, 2000 Product Specification

Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com/ipcenter
Support: support.xilinx.com

Features

- Drop-in module for Virtex™-II, Virtex, Virtex™-E, and Spartan™-II FPGAs
- Generates fast, compact FIFO-style shift registers, delay lines or time-skew buffers using the SRL16 mode of the Select RAM
- User options to create Fixed-length or Variable-length shift registers
- Optional output register capability with clock enable and asynchronous and synchronous controls
- Memory initialization options
- Incorporates Xilinx Smart-IP technology for maximum performance
- To be used with version 3.1i or later of the Xilinx CORE Generator System



RAM-based Shift Register

Component Name :

Data Width : 16 Valid Range: 1..256

Depth : 16 Valid Range: 1..1024

Shift Type

● Fixed Length ○ Variable Length Lossless ○ Variable Length Lossy

Output Options

□ Register Final Bit

Layout

☑ Create RPM

Initial Contents

.coe file: No parameters or coefficients read

Default Data 0 Default Data Radix 16

□ Display Core Viewer after Generation

Figure 1: Main RAM-based Shift Register Parameterization Screen

Functional Description

The RAM-based Shift Register module provides a very efficient multibit wide shift register for use in FIFO-like applications or as a delay line or time skew buffer. Fixed-length shift registers and variable-length shift registers can be created. An option is also provided to register the outputs of the module. If output registering is selected, there are additional options for **Clock Enable**, **Asynchronous Set**, **Clear**, and **Init**, and **Synchronous Set**, **Clear** and **Init** of the output register. The module can optionally be generated as a relationally placed macro (RPM) or as unplaced logic.

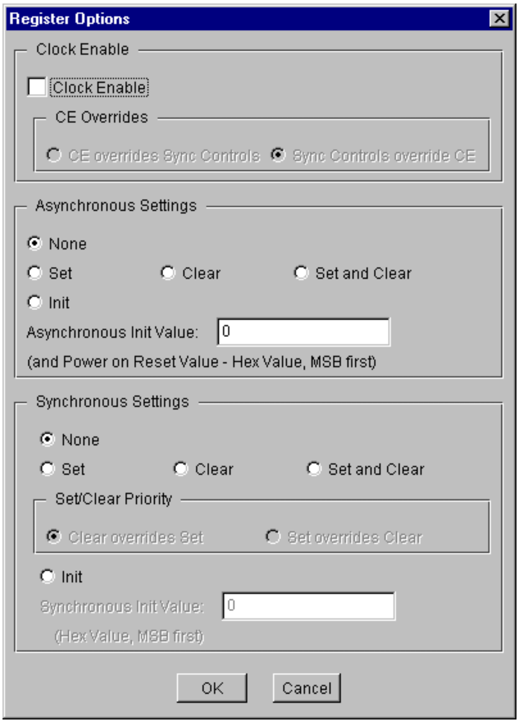
Pinout

Signal names for the schematic symbol are shown in Figure 3 and described in Table 1.

CORE Generator Parameters

The main CORE Generator parameterization screen for this module is shown in Figure 1. The parameters are as follow:

- **Component Name**: The component name is used as the base name of the output files generated for this module. Names must begin with a letter and must be composed from the following characters: a to z, 0 to 9 and “_”.
- **Data Width**: Enter the width of the input to the shift register. The valid range is 1 to 256. The default value is 16.
- **Depth**: Enter the depth of the shift register. The valid range is 1 to 1024. The default value is 16. Note that for variable-length modules, the minimum is 2, but when the output is registered, the minimum is 3.
- **Shift Type**: Select the appropriate radio button to specify the required type of shift register. The default selection is **Fixed Length**.
  - **Fixed Length**: Parallel data is clocked into the shift register and appears at the output bus **Depth** clock cycles later.



Register Options

Clock Enable

□ Clock Enable

CE Overrides

○ CE overrides Sync Controls ● Sync Controls override CE

Asynchronous Settings

● None

○ Set ○ Clear ○ Set and Clear

○ Init

Asynchronous Init Value: 0

(and Power on Reset Value - Hex Value, MSB first)

Synchronous Settings

● None

○ Set ○ Clear ○ Set and Clear

Set/Clear Priority

● Clear overrides Set ○ Set overrides Clear

○ Init

Synchronous Init Value: 0

(Hex Value, MSB first)

Figure 2: RAM-based Shift Register Options Parameterization Screen

Table 1: Core Signal Pinout

Signal	Signal Direction	Description
D[N:0]	Input	Parallel Data Input
A[M:0]	Input	Address Input (present on variable-length modules only)
CE	Input	Clock Enable
CLK	Input	Clock: rising edge clock signal
ASET	Input	Asynchronous Set: forces registered outputs to a high state when driven
ACLR	Input	Asynchronous Clear: forces registered outputs to a low state when driven
SSET	Input	Synchronous Set: forces registered output to a high state on next concurrent clock edge
SCLR	Input	Synchronous Clear: forces registered output to a low state on next concurrent clock edge
AINIT	Input	Asynchronous Initialize: forces registered outputs to user-defined state when driven
SINIT	Input	Synchronous Initialize: forces registered outputs to user defined state on next concurrent clock edge
Q[N:0]	Output	Parallel Data Output

Note:
All control inputs are Active High. Should an Active Low input be required for a particular control pin, an inverter must be placed in the path to the pin. The inverter will be absorbed appropriately during mapping.

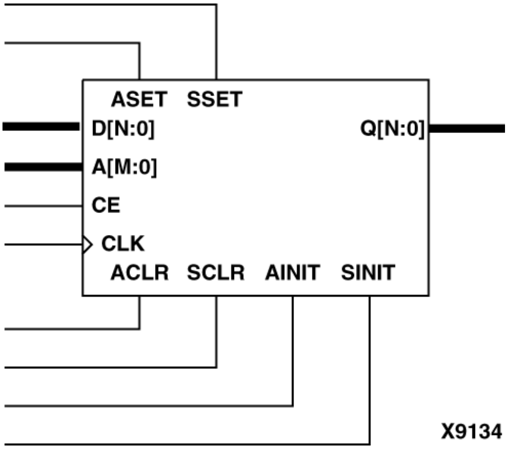


Figure 3: Core Schematic Symbol

- **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.
- **Variable-Length Lossy**: 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 “lossy” because, when the address is changed, the output cannot be guaranteed to be correct for C clock cycles where C is the new value for the address.

- **Output Options**:
  - **Register Final Bit**: This checkbox defines the presence of the FD-based registers on the output of the module. This effectively adds one extra clock to the shift delay through the module, but this is accounted for in the depth selection. Register Options are not enabled unless this option is checked. The default setting is unchecked.
  - **Register Options**: Clicking on this button brings up the Register Options parameterization screen (see Figure 2).

The Register Options parameterization screen for this module is shown in Figure 2. The parameters are as follows:

- **Clock Enable**: When this box is checked, the module is generated with a clock enable input. The default setting is unchecked.
- **CE Overrides**: This parameter controls whether or not the SSET, SCLR, and SINIT inputs are qualified by CE. This parameter is enabled only when a **Clock Enable** input has been requested.

  When **CE Overrides Sync Controls** is selected, an active level on any of the synchronous control inputs will be acted upon only when the CE pin is also Active. Note that this is not the way that the dedicated inputs on the flip-flop primitives work, and so setting the **CE Overrides** parameter to **CE Overrides Sync Controls** will force any synchronous control functionality to be implemented using logic in the look-up-tables (LUTs) preceding the output register. This increases utilization of resources.

  When **Sync Controls Override CE** is selected, an active level on any of the synchronous control inputs is acted upon irrespective of the state of the CE pin. This setting allows the dedicated inputs on the flip-flop primitives to be used for the synchronous control functions provided that asynchronous controls are not requested. If both asynchronous and synchronous controls are requested, the synchronous control functionality must be implemented using logic in the LUTs preceding the output register. In this case, the CE input has to be gated with the synchronous control inputs so that each synchronous control input and the CE input can generate a CE signal to the flip-flops. This additional gating in the CE path causes a performance degradation for the module.

  The default setting is **Sync Controls Override CE** so that the more efficient implementation can be generated.
- **Asynchronous Settings**: All asynchronous controls are implemented using the dedicated inputs on the flip-flop primitives. The module can be generated with the following asynchronous control inputs by clicking on the appropriate button:
  - **None**: No asynchronous control inputs. This is the default setting.
  - **Set**: An ASET control pin is generated.
  - **Clear**: An ACLR control pin is generated.
  - **Set and Clear**: Both ASET and ACLR control pins are generated. ACLR has priority over ASET when both are asserted simultaneously.
  - **Init**: An AINIT control pin is generated which, when asserted, will asynchronously set the output register to the value defined in the **Asynchronous Init Value** text box.
- **Asynchronous Init Value**: This text box accepts a hex value with an equivalent bit width that must be less than or equal to the **Data Width**. If a value is entered that has fewer bits than the **Data Width,** it is padded with zeros. An invalid value is highlighted in red in the text box.
  The default value is 0.
- **Synchronous Settings**: When no asynchronous controls are requested (i.e., the **Asynchronous Setting** is **None**) the synchronous controls can be implemented using the dedicated inputs on the flip-flop primitives. Exceptions to this are described in the

sections for the **Set/Clear Priority** and **CE Overrides** parameters.

When asynchronous controls are present, any synchronous control functionality must be implemented using logic in the look-up tables (LUTs) preceding the output register. Modules requiring no nonregistered output use combinations of parameters that allow this logic to be absorbed into the same LUTs implementing the function. In cases where this absorption is not possible, the synchronous control logic will require an additional LUT per output bit.

The module can be generated with the following synchronous control inputs by clicking on the appropriate button:

- **None**: No synchronous control inputs. This is the default setting.
- **Set**: An SSET control pin is generated.
- **Clear**: An SCLR control pin is generated.
- **Set and Clear**: Both SSET and SCLR control pins are generated. SCLR/SSET priority is defined by the setting of the **Set/Clear Priority** parameter.
- **Init**: An SINIT control pin is generated which, when asserted, will synchronously set the output register to the value defined in the **Synchronous Init Value** text box.

- **Set/Clear Priority**: Selecting the appropriate radio button controls the relative priority of SCLR and SSET. This parameter is enabled only when **Set and Clear** is selected for **Synchronous Settings**.

A setting of **Clear Overrides Set** corresponds to the native operation of the flip-flop primitive. This setting will result in a more efficient implementation when asynchronous controls are not requested. A setting of **Set Overrides Clear** can only be implemented using logic in the LUTs preceding the output register.

The default setting **Clear Overrides Set** assures that the dedicated inputs on the flip-flops can be used if available.

- **Synchronous Init Value**: This text box accepts a hex value requiring an equivalent bit width less than or equal to the **Data Width**. If a value is entered that has fewer bits than the **Data Width,** it is padded with zeros. An invalid value is highlighted in red in the text box. This parameter is enabled only when the **Synchronous Settings** parameter is set to **Init**. The default value is 0.

- **Initial Contents...**: The initial values of the memory elements can be set with the use of a Coefficients file (COE), by loading the ".coe" file using the **Load Coefficients...** button. The initial contents can be viewed by selecting the **Show Coefficients...** button. For a description of the COE file, refer to the section titled, "Specifying Memory Contents using a COE file." The contents of the COE file are converted to a Memory Initialization File (MIF) having the values in a binary format. This file describes the true memory contents used by the core and the simulation models. For a description of the memory initialization file, refer to the section, "Specifying Memory Contents using a MIF File."
  - **Default Data**: Enter the initial value to be stored in any memory location not specified by another means. When no value is entered, this field defaults to 0. Values can be entered in binary or hex format, as defined by the **Radix** entry.
  - **Default_Data_Radix**: Choose the radix of the **Default Data** value. Valid entries are 2, 10, and 16.
- **Create RPM**: When this box is checked, the module is generated with relative location attributes attached. The resulting placement of the module is in a column with two bits per slice. The default operation is to create an RPM.

  Note that when a module is created as an RPM it is possible that one or more of the module dimensions may exceed those of the device being targeted. If this is the case, mapping errors will occur and the compilation process will fail. In this case, the module should be re-generated with the **Create RPM** checkbox unchecked. This will reduce the performance of the module as the placement will no longer be controlled.

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.

The COE file consists of two parameters similar to an XCO file, but the end of each line is determined with the use of a semicolon. The two parameters are:

- **memory_initialization_vector**: Each row of memory elements are defined with a binary, decimal 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 whitespace, up to the depth of the memory.
- **memory_initialization_radix:** The radix of the initialization value is specified here, the choice being 2, 10, 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;
```

Specifying Memory Contents using a MIF File

The COE file provides a wrapper to the user to allows initialization of memory contents. However, the MIF file holds the actual binary data used to initialize the memory in the core and simulation models. The MIF file consists of one line of text per memory location, the first line in the file corresponding to address 0, the second line corresponding to address 1, and so on. The text on each line must be the initialization value (MSB first) for the corresponding memory address in binary format, with exactly one binary digit per bit of the memory's width.

Power On Conditions

See the **FD-based Register** datasheet for information on the power up values for registered RAM-based shift registers.

Parameter Values in the XCO File

Names of XCO file parameters and their parameter values are identical to the names and values shown in the GUI, except that underscore characters (_) are used instead of spaces. The text in an XCO file is case insensitive.

Table 2 shows the XCO file parameters and values, and summarizes the GUI defaults. The following is an example of the CSET parameters in an XCO file:

```
CSET component_name = abc123
CSET data_width = 16
CSET depth = 16
CSET shift_type = Fixed_Length
CSET register_final_bit = FALSE
CSET create_rpm = TRUE
CSET clock_enable = FALSE
CSET ce_overrides = sync_controls_override_ce
CSET asynchronous_settings = none
CSET asynchronous_init_value = 0000
CSET synchronous_settings = none
CSET synchronous_init_value = 0000
CSET set_clear_priority = clear_overrides_set
CSET coefficient_file = abc123.coe
CSET default_data = 0
CSET default_data_radix = 16
```

Core Resource Utilization

For an accurate measure of the usage of primitives, slices, and CLBs for a particular point solution, check the **Display Core Viewer after Generation** checkbox, in CoreGen.

Ordering Information

This core is downloadable free of charge from the Xilinx IP Center (www.xilinx.com/ipcenter), for use with version 3.1i or later of the Xilinx CORE Generator System. The CORE Generator System is bundled with the Alliance and Foundation implementation tools.

To order Xilinx software, contact your local Xilinx sales representative. For information on the Xilinx sales office nearest you, please refer to:

Table 2: XCO File Values and Default Settings

Parameter	XCO File values	Default GUI Setting
component_name	ASCII text starting with a letter and based upon the following character set: a..z, 0..9 and _	blank
data_width	Integer in the range 1 to 256	16
depth	Integer in the range 1 to 1024	16
shift_type	One of the following keywords: fixed_length, variable_length_lossy, variable_length_lossless	fixed_length
register_final_bit	One of the following keywords: true, false	false
create_rpm	One of the following keywords: true, false	true
clock_enable	One of the following keywords: true, false	false
ce_overrides	One of the following keywords: sync_controls_override_ce, ce_overrides_sync_controls	sync_controls_override_ce
asynchronous_settings	One of the following keywords: none, set, clear, set_and_clear, init	none
asynchronous_init_value	Hex value whose value does not exceed $2^{data\_width} - 1$	0
synchronous_settings	One of the following keywords: none, set, clear, set_and_clear, init	none
synchronous_init_value	Hex value whose value does not exceed $2^{data\_width} - 1$	0
set_clear_priority	One of the following keywords: clear_overrides_set or set_overrides_clear	clear_overrides_set
coefficient_file	ASCII text starting with a letter and based upon the following character set: a .. z, 0..9 and _; filename must end with a ".coe" extension	blank
default_data	Numeric value in the Radix specified by the default_data_radix keyword whose value does not exceed $2^{DATA \backslash WIDTH - 1}$	0
default_data_radix	One of the following keywords: 2, 10, 16	16