

Features

- Drop-in module for Virtex™, Virtex-E, Virtex-II, Virtex-II Pro, Spartan™-II, Spartan-IIE, Spartan-3, and Spartan-3E FPGAs
- Supports data from 1 to 256 bits wide
- Optional Clock Enable input
- Optional Asynchronous Set, Clear and Init
- Optional Synchronous Set, Clear and Init
- Uses relationally placed macro (RPM) mapping and placement technology, for maximum and predictable performance
- Incorporates Xilinx Smart-IP™ technology for utmost parameterization and optimum implementation
- For use with v7.1i and later of the Xilinx CORE Generator™ system

Functional Description

The flip-flop based data register is a member of the BaseBLOX series of building blocks for the Virtex architecture. Options are provided for Clock Enable, Asynchronous Set, Clear and Init., and Synchronous Set, Clear and Init. The module can optionally be generated as a Relationally Placed Macro (RPM) or as unplaced logic. When an RPM is generated the logic is placed in a column.

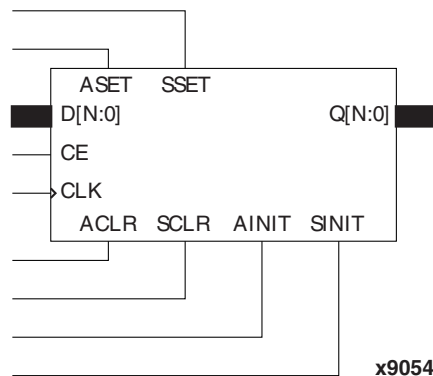


Figure 1: Core Schematic Symbol

Pinout

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

Table 1: Core Signal Pinout

Name	Direction	Description
D[N:0]	Input	Data input
CE	Input	Clock Enable
CLK	Input	Clock: rising edge
ASET	Input	Asynchronous Set – forces all outputs to a High state when driven
ACLR	Input	Asynchronous Clear: forces all outputs to a Low state when driven.
SSET	Input	Synchronous Set: forces all outputs to a High state on next concurrent clock edge
SCLR	Input	Synchronous Clear: forces all outputs to a Low state on next concurrent clock edge
AINIT	Input	Asynchronous Initialize: forces the outputs to a user defined state when driven

© 2005 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Table 1: Core Signal Pinout (Continued)

Name	Direction	Description
SINIT	Input	Synchronous Initialize - forces the outputs to a user defined state on next concurrent clock edge
Q[N:0]	Output	Data Output

Notes:

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

CORE Generator Parameters

The main CORE Generator parameters are as follows:

- **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 data register width. The valid range is 1 to 256. The default value is **16**.
- **Register Options:** Clicking on this button brings up the Register Options parameterization screen.
- **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.

The Register Options 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 results in increased resource utilization.

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 results in a performance degradation for the module due to the additional gating in the CE path.

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 at the same time.
 - **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 whose equivalent bit width 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. There are exceptions to this which 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. With modules where a non-registered output is not required there are combinations of parameters that allow this logic to be absorbed into the

same LUTs used to implement 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:** By selecting the appropriate radio button, the relative priority of SCLR and SSET can be controlled. 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 be implemented using logic in the LUTs only preceding the output register.

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

- **Synchronous Init Value:** This text box accepts a hex value whose equivalent bit width 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. This parameter is enabled only when the **Synchronous Settings** parameter is set to **Init**. The default value is **0**.

Power-On Conditions

Table 2 defines the power on reset (POR) behavior of the register when control functions are implemented using dedicated inputs on the register primitive.

In a case where the synchronous function is implemented in logic, the POR action follows the settings for the asynchronous control function. When no asynchronous controls are present and synchronous controls are implemented in logic, the POR action will be to go to the

Table 2: Power-On Reset Behavior

Register Primitive Dedicated Inputs in Use						
ACLR	ASET	AINIT	SCLR	SSET	SINIT	POR Action
No	No	No	No	No	No	AINIT Value
Yes	X	X	X	X	X	All 0's
No	Yes	X	X	X	X	All 1's
No	No	Yes	X	X	X	AINIT Value
No	No	No	Yes	X	X	All 0's
No	No	No	No	Yes	X	All 1's
No	No	No	No	No	Yes	SINIT Value

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 3 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 create_rpm = TRUE
CSET clock_enable = FALSE
```

```
CSET ce_overrides = sync_controls_override_ce
CSET asynchronous_settings = none
CSET async_init_value = 0000
CSET synchronous_settings = none
CSET sync_init_value = 0000
CSET set_clear_priority = clear_overrides_set
```

Table 3: XCO File Values and Default Values

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 2 to 256	16
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
async_init_value	Hex value whose value does not exceed, $2^{\text{data_width}} - 1$	0
synchronous_settings	One of the following keywords: none, set, clear, set_and_clear, init	none
sync_init_value	Hex value whose value does not exceed, $2^{\text{data_width}} - 1$	0
set_clear_priority	One of the following keywords: clear_overrides_set or set_overrides_clear	clear_overrides_set

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 the CORE Generator.

Ordering Information

This core may be downloaded from the Xilinx [IP Center](#) for use with the Xilinx CORE Generator System v7.1i and later. The CORE Generator system is bundled with all Foundation Series Software packages, at no additional charge.

To order Xilinx software, please visit the Xilinx [Silicon Xpresso Cafe](#) or contact your local Xilinx [sales representative](#).

Information on additional Xilinx LogiCORE modules is available on the Xilinx [IP Center](#).