Binary Counter 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 Up, Down and Up/Down Counters
- Supports counts ranging from 2 to 64 bits wide
- Optional load capability
- Optional user programmable threshold outputs
- Optional clock enable and asynchronous and synchronous controls
- Counter increment value can be user defined or supplied externally.
- User-programmable Count limit
- Incorporates Xilinx Smart-IP technology for maximum performance
- To be used with version 3.1i and later of the Xilinx CORE Generator System

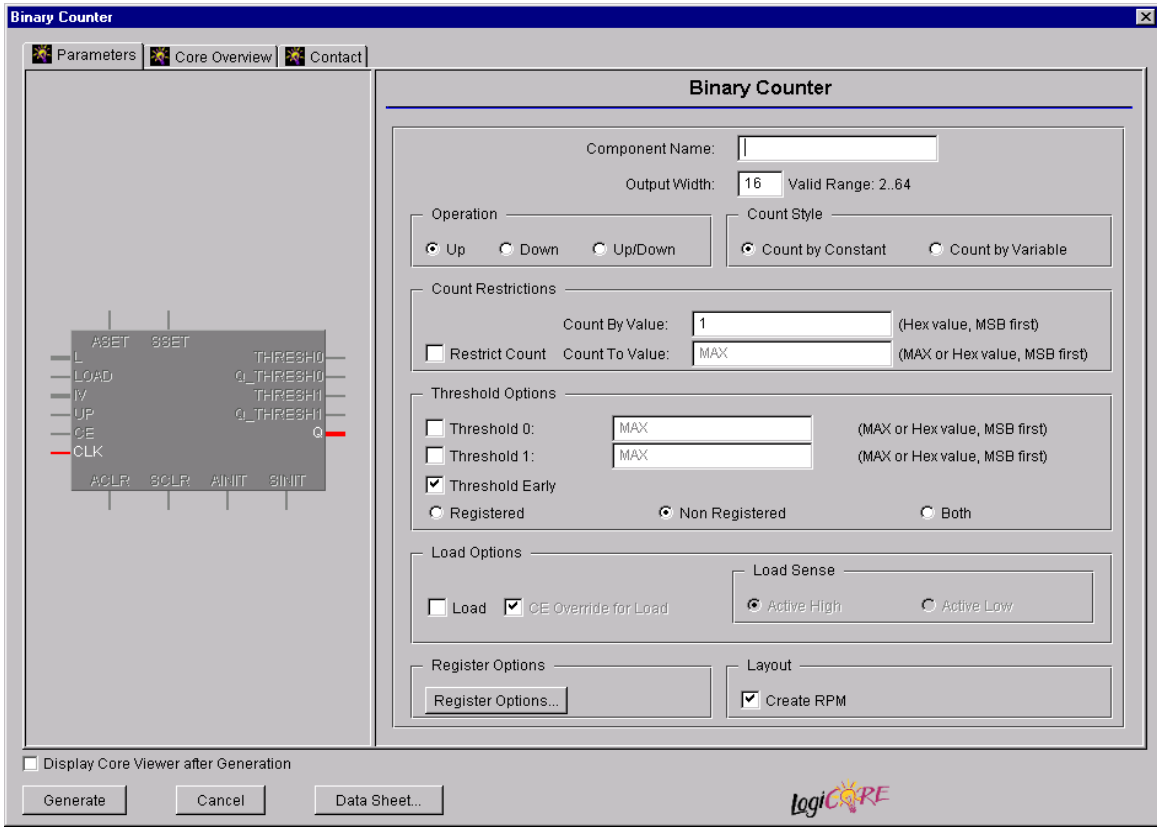

Figure 1: Main Binary Counter Parameterization Screen

Functional Description

The binary counter is used to create up counters, down counters and up/down counters with outputs of up to 64 bits wide. Support is provided for two threshold signals that can be programmed to become active when the counter reaches a user defined count. The upper limit of the count is user programmable, and the counter's increment value can be user defined or provided via an external port. Options are provided for **Clock Enable**, **Asynchronous Set**, **Clear**, and **Init**, and **Synchronous Set**, **Clear** and **Init**. An optional **Load** capability is also provided which can load the value on the Load port directly into the output register. 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. When the counter reaches terminal count or the "count to value" the next count will be zero.

Pinout

Signal names for the schematic symbol are shown in Figure 3 and described in Table 2.

CORE Generator Parameters

The main CORE Generator parameterization screen for this module is shown in Figure 1. The 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 "_".
- **Output Width**: Enter the width of the counter. The valid range is 2 to 64. The default value is 16.
- **Operation**: Select the appropriate radio button for the operation required. The default setting is **Up**.
- **Count Style**: The module can be incremented by a constant value, or it can have its count value supplied via the IV[N:0] port. The default setting is **Count by Constant**.
- **Count Restrictions**:
  - **Count By Value**: Enter the count increment as a hex value. This text box is only enabled for the **Count Style** of **Count by Constant**. When the **Restrict Count** check box is unchecked (or the **Count To Value** is set to **MAX**) the valid range is 1 to $2^{\text{Output Width}} - 1$. When the **Restrict Count** check box is checked the valid settings for the **Count By Value** are governed by the equation:

    (**Count To Value** + 1) / **Count By Value** = Integer

    The default value is 1.
  - **Restrict Count**: When this check box is checked the counter will only count up to the value specified in the **Count To Value** box. When unchecked the counter will count up to the maximum value that can be represented using the specified output width.The default is for no count restriction.
  - **Count To Value**: Enter the count limit as a hex value or the keyword **MAX** (which corresponds to 1's for each of the counter output bits). The valid range for hex values is 1 to $2^{\text{Output Width}} - 1$.This text box is only enabled when the **Restrict Count** check box is checked. The default value is MAX.

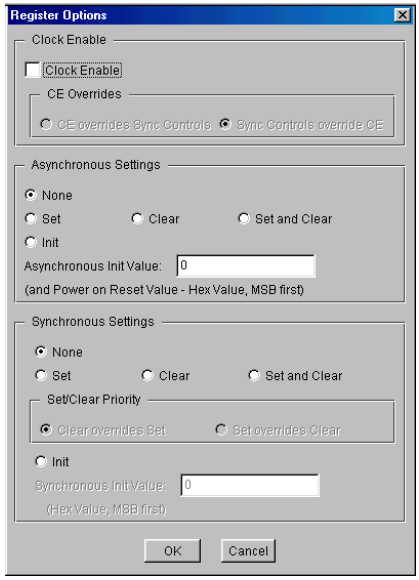

Figure 2: Binary Counter Register Options Parameterization Screen

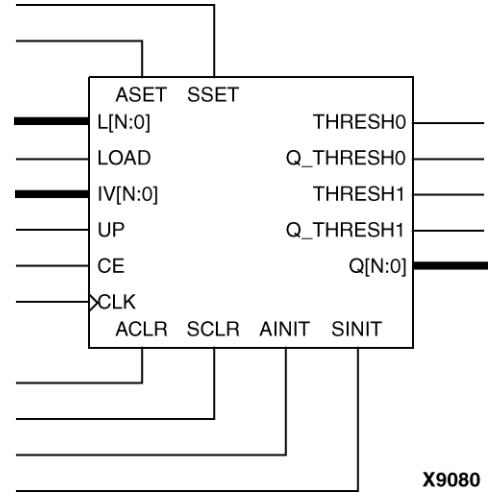

Figure 3: Core Schematic Symbol

- **Output Options**: The behavior of the threshold outputs depends on the state of the **Threshold Early** checkbox. If **Threshold Early** is checked, a registered threshold output will become active on the same clock edge that sets the output of the counter to the **Threshold Value**. A nonregistered threshold output will become active one clock cycle before the threshold value is reached on the counter outputs. This allows more setup time for ripple operations that utilize the nonregistered threshold output prior to the clock edge that sets the counter outputs to the threshold value.
  If **Threshold Early** is unchecked, a nonregistered output will become active when the threshold value is reached on the counter outputs and a registered output will become active on the next clock edge. This allows implementation of a combinatorial terminal count signal.
  - **Threshold 0**: When this check box is checked the Threshold 0 output(s) (registered, non-registered, or both) will be generated. The default is to not generate a Threshold 0 output.
  - **Threshold 0 Value**: Enter the value at which the THRESH_0 value will be activated as a hex value or the keyword **MAX** (which corresponds to 1's for each of the counter output bits). The valid range for hex values is 0 to **Count To Value**. This text box is only enabled when the **Threshold 0** check box is checked. The default value is MAX.

    Note that when the **Restrict Count** check box is checked the **Threshold 0 Value** setting must be less than or equal to the **Count To Value** setting.

  - **Threshold 1**: When this check box is checked the Threshold 1 output(s) (registered, non-registered, or both) will be generated. The default is to not generate a Threshold 1 output.
  - **Threshold 1 Value**: Enter the value at which the THRESH_1 value will be activated as a hex value or the keyword **MAX** (which corresponds to 1's for each of the counter output bits). The valid range for hex values is 0 to **Count To Value**. This text box is only enabled when the **Threshold 1** check box is checked. The default value is MAX.

    Note that when the **Restrict Count** check box is checked the **Threshold 1 Value** setting must be less than or equal to the Count to Value settings.

  - **Output Options**: Select the appropriate radio button for the types of outputs required for the Threshold signals. The default setting is **Non Registered**.
  - **Register Options**: Clicking on this button brings up the Register Options parameterization screen (see figure 2).
- **Load**: Activating the LOAD pin allows the value on the L[N:0] input port to pass through the logic and be loaded into the output register on the next active clock edge. The default is for no LOAD pin to be generated. If any synchronous options are selected through the Register Options parameterization screen, the precedence of the LOAD port depends on whether the **Restrict Count** option is checked. The details are in Table 1.

Table 1: Precedence of LOAD over Synchronous Signals

LOAD port and:	Restrict Count	
	No	Yes
SCLR	LOAD	LOAD
SSET	SSET	LOAD
SINIT	SINIT	LOAD

The change in precedence is is a result of the internal implementation of restricted count functionality. If a different precedence is required by the user, it must be implemented in external logic.

- **CE Override for Load**: This parameter controls whether or not the LOAD input is qualified by CE. When this box is checked the activation of the LOAD signal will also enable the output register. When this box is unchecked the register needs to have CE active in order to load the B port data. By default this check box is checked.

Table 2: Core Signal Pinout

Signal	Signal Direction	Description
L[N:0]	Input	Load data port
LOAD	Input	Load Control signal
IV[N:0]	Input	Count Increment Value
UP	Input	Controls the count direction on an up/down counter. Counts Up when High, Down when Low
CE	Input	Clock Enable
CLK	Input	Clock - rising edge clock signal
ASET	Input	Asynchronous Set - forces the registered output to a High state when driven
ACLR	Input	Asynchronous Clear - forces outputs to a Low state when driven
SSET	Input	Synchronous Set - forces the registered output to a High state on next concurrent clock edge

Table 2: Core Signal Pinout

Signal	Signal Direction	Description
SCLR	Input	Synchronous Clear - forces the registered output to a Low state on next concurrent clock edge
AINIT	Input	Asynchronous Initialize - forces the registered outputs to a user defined state when driven
SINIT	Input	Synchronous Initialize - forces the registered outputs to a user defined state on next concurrent clock edge
THRESH0	Output	User programmable threshold signal
Q_THRESH0	Output	Registered user programmable threshold signal
THRESH1	Output	User programmable threshold signal
Q_THRESH1	Output	Registered user programmable threshold signal
Q[N:0]	Output	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.

- **Load Sense:** LOAD is the only pin that has a parameter to control its active sense. This is because selection of an Active Low bypass results in a significant area savings for the module. By default this parameter is set to Active High so that it conforms with the active sense of all other control signals.
- **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 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 only enabled 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 only be acted upon 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 **Output Width**. If a value is entered that has fewer bits than the **Output 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 only enabled 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 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 **Output Width**. If a value is entered that has fewer bits than the **Output Width** it is padded with zeros. An invalid value is highlighted in red in the text box. This parameter is only enabled when the **Synchronous Settings** parameter is set to **Init**. The default value is 0.

Power On Conditions

See the **FD-based Register** datasheet for information on the power up values for the counter. Note that if the user requests **Restrict Count** functionality, the final register has an internally used synchronous clear which may affect the power up 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 output_width = 16
CSET operation = up
CSET count_style = count_by_constant
CSET count_by_value = 1
CSET restrict_count = FALSE
CSET count_to_value = MAX
CSET threshold_0 = FALSE
CSET threshold_0_value = MAX
CSET threshold_1 = FALSE
CSET threshold_1_value = MAX
CSET threshold_options = non_registered
CSET threshold_early = TRUE
CSET load = FALSE
CSET ce_override_for_load = FALSE
CSET load_sense = active_high
CSET create_rpm = TRUE
CSET clock_enable = FALSE
CSET ce_overrides = ce_overrides_sync_controls
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
```

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 and later versions 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:

http://www.xilinx.com/company/sales.htm.

Table 3: Default Values and XCO File 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
output_width	Integer in the range 2 to 64	16
operation	One of the following keywords: up, down, up_down	up
count_style	One of the following keywords: count_by_constant, count_by_variable	count_by_constant
count_by_value	Hex value in the range 1 to count_to_value that meets the following restriction: (count_to_value + 1) / count_by_value = Integer	1
restrict_count	One of the following keywords: true, false	false
count_to_value	The keyword MAX (which corresponds to a hex setting of $2^{output\_width} - 1$) or a hex value in the range 1 to $2^{output\_width} - 1$	MAX
threshold_0	One of the following keywords: true, false	false
threshold_0_value	The keyword “MAX” or a hex value in the range 0 to $2^{count\_to\_value} - 1$	MAX
threshold_1	One of the following keywords: true, false	false
threshold_1_value	The keyword “MAX” or a hex value in the range 0 to $2^{count\_to\_value} - 1$	MAX
threshold_options	One of the following keywords: non_registered, registered, both	non_registered
threshold_early	One of the following keywords: true, false	true
load	One of the following keywords: true, false	false
ce_override_for_load	One of the following keywords: true, false	true
load_sense	One of the following keywords: active_high, active_low	active_high
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 $2^{output\_width} - 1$	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 $2^{output\_width} - 1$	0
set_clear_priority	One of the following keywords: clear_overrides_set, set_overrides_clear	clear_overrides_set