Adder/Subtractor V1.0.3

December 17, 1999 Product Specification

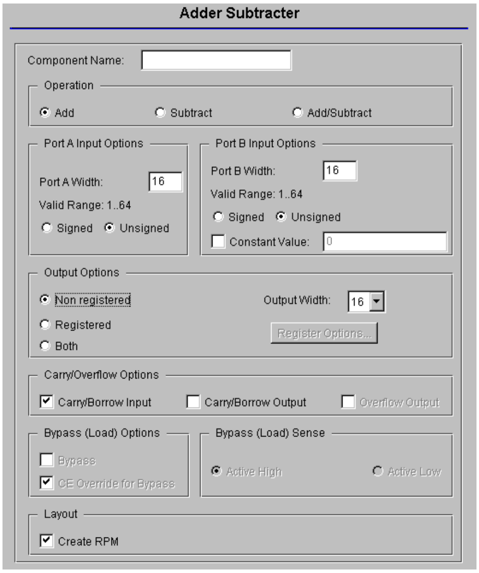
Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com/support/techsup/appinfo
www.xilinx.com/ipcenter

Features

- Drop-in module for Virtex, Virtex™-E and Spartan™–II FPGAs
- Generates Adder, Subtractor and Adder/Subtractor
- Supports signed and unsigned operations
- Supports inputs ranging from 1 to 64 bits wide
- Supports outputs ranging from 1 to 66 bits wide
- Optional registered output with optional clock enable and asynchronous and synchronous controls
- Optional Bypass (Load) capability
- Incorporates Xilinx Smart-IP technology for maximum performance
- To be used with version 2.1i and later of the Xilinx CORE Generator System

Functional Description

The Adder/Subtractor module can create adders, subtractors and adder/subtractors which operate on signed or unsigned data. The data inputs are provided on the A and B input buses, and, optionally, the B value can be set to a constant. The result is available on the output bus. Optional carry input and carry/borrow/overflow outputs are available. Outputs can be registered or non-registered. When a registered output is selected options are also provided for **Clock Enable**, **Asynchronous Set**, **Clear**, and **Init**, and **Synchronous Set**, **Clear** and **Init**. An optional **Bypass** capability is also provided which can load the value on the B 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.



Adder Subtracter

Component Name:

Operation
● Add ○ Subtract ○ Add/Subtract

Port A Input Options
Port A Width: 16
Valid Range: 1..64
○ Signed ● Unsigned

Port B Input Options
Port B Width: 16
Valid Range: 1..64
○ Signed ● Unsigned
□ Constant Value: 0

Output Options
● Non registered
○ Registered
○ Both
Output Width: 16

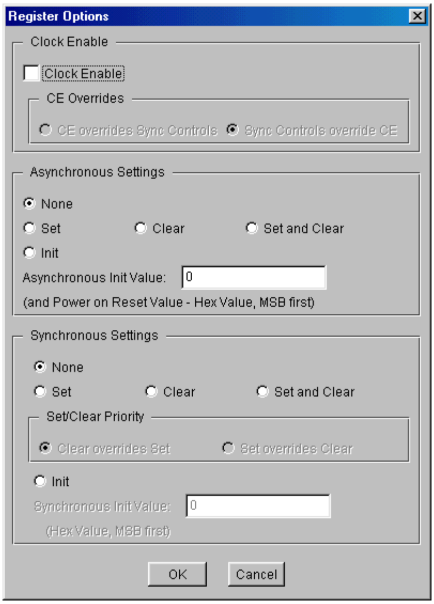
Carry/Overflow Options
☑ Carry/Borrow Input □ Carry/Borrow Output □ Overflow Output

Bypass (Load) Options
□ Bypass
☑ CE Override for Bypass

Bypass (Load) Sense
● Active High ○ Active Low

Layout
☑ Create RPM

Figure 1: Main Adder/Subtractor Parameterization Screen



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: Adder/Subtractor Register Options Parameterization Screen

Table 1: Core Signal Pinout

Signal	Signal Direction	Description
A[N:0]	Input	A Input bus
A_SIGNED	Input	A Input sign control
B_SIGNED	Input	B Input sign control
B[M:0]	Input	B Input bus
ADD	Input	Controls the operation performed by an Adder/Subtractor (High = Addition, Low = Subtraction).
C_IN	Input	Carry Input
OVFL	Output	Overflow Output (signed modules only)
C_OUT	Output	Carry Output (Adder and Adder/Subtractor only)
S[P:0]	Output	Non-registered output
D_OVFL		Internal
D_C_OUT		Internal
D[P:0]		Internal
BYPASS	Input	Bypass Control Signal
CE	Input	Clock Enable
CLK	Input	Clock - rising edge clock signal
ASET	Input	Asynchronous Set: forces 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 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_OVFL	Output	Registered Overflow Output (signed modules only)
Q_C_OUT	Output	Registered Carry Output (Adder and Adder/Subtractor only)
Q[P:0]	Output	Registered output
B_OUT	Output	Borrow Output (Subtractor only - active low)
Q_B_OUT	Output	Registered Borrow Output (Subtractor only)

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.

Pinout

Signal names for the schematic symbol are shown in Figure 3 and described in Table 1. Note that Figure 3 shows the C_OUT and Q_C_OUT pins which appear on adders and adder/subtractors. For a subtractor these pins are named B_OUT and Q_B_OUT, respectively.

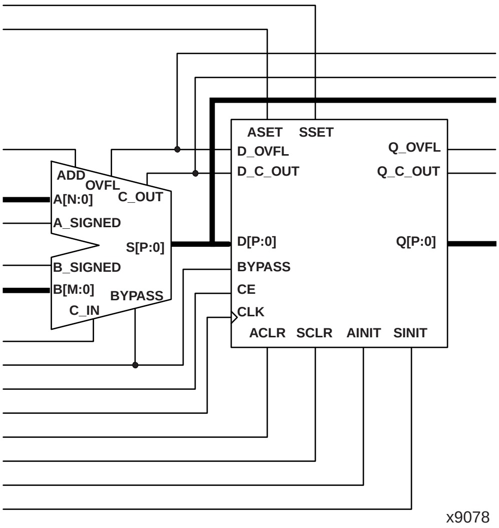


Figure 3: Core Schematic Symbol

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 “_”.
- **Operation**: Select the appropriate radio button for the operation required. The default setting is **Add**.
- **Port A Input Options**:

Figure 1: Availability of Carry/Borrow/Overflow Outputs and Output Data Type/Size Against Input Data Type

A[N:0]	B[M:0]	S[P:0]	Valid Values for P[1]	C_OUT/B_OUT	OVFL
Unsigned	Unsigned	Unsigned	P = Q	Available	Not Available
			P = Q + 1	Not Available	Not Available
Unsigned	Signed	Signed	P = Q + 2	Not Available	Not Available
Signed	Unsigned	Signed	P = Q + 2	Not Available	Not Available
Signed	Signed	Signed	P = Q	Not Available	Available
			P = Q + 1	Not Available	Not Available

Note:

[1] Q represents the larger of N or M.

  - **Port A Width**: Enter the width of the Port A input. The valid range is 1 to 64. The default value is 16.
  - **Port A Sign**: Enter the sign of the Port A input. The default value is **Unsigned**.
- **Port B Input Options**:
  - **Port B Width**: Enter the width of the Port B input. The valid range is 1 to 64. The default value is 16.
  - **Port B Sign**: Enter the sign of the Port B data. The default value is **Unsigned**.
  - **Constant Value**: When this check box is checked Port B is set to the value that is typed into the adjacent text box. The Constant Value must be entered in hex format and must not exceed the specified **Port B Width**. In most cases specifying Port B to be a constant will create a module without Port B. The only exception to this is when bypass functionality is requested, as Port B is needed to provide the bypass data in this case. The default setting is for the Port B value to be provided via Port B.
  - **Output Options**:
  - Select the appropriate radio button for the types of outputs required. The output options settings selected here apply to all outputs. The default setting is registered.
  - **Output Width**: The output width is specified using the pull-down list. The valid range varies depending on the settings of **Port A Width**, **Port A Sign**, and **Port B Width** and **Port B Sign** as shown in Table 2.
- **Register Options**: This button is only enabled when a registered output has been requested via the **Output Options**. Clicking on this button brings up the Register Options parameterization screen (see Figure 2).
- **Carry/Overflow Options**:
  - **Carry/Borrow Input**: The presence of a C_IN or B_IN pin is controlled by the setting of this check box. The pin generated for adders and adder/subtractors is named C_IN. The pin generated for subtractors is named B_IN. The default behavior is to generate a C_IN or B_IN pin.
  - **Carry/Borrow Output**: The presence of a C_OUT or B_OUT pin is controlled by the setting of this check box. This option is only enabled when the module generates an unsigned result (see Table 2). The pin generated for adders and adder/subtractors is named C_OUT. The pin generated for subtractors is named B_OUT. The default behavior is to not generate a C_OUT or B_OUT pin.
  - **Overflow Output**: The presence of an OVFL pin is controlled by the setting of this check box. This option is only enabled when the module generates a signed result (see Table 2). The default behavior is to not generate an OVFL pin.
- **Bypass**: Activating the BYPASS pin allows the value on the Port B to pass through the logic and be loaded into the output register on the next active clock edge. This check box is only available on a registered module. The default is for no BYPASS pin to be generated.
- **CE Override for Bypass**: This parameter controls whether or not the BYPASS input is qualified by CE. When this box is checked the activation of the BYPASS signal will also enable the register. When this box is unchecked the register needs to have CE active in order to load the Port B data. By default this check box is checked.
- **Bypass Sense**: BYPASS 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 saving 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 will be generated with relative location attributes attached. The resulting placement of the module will be 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 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 the synchronous control functionality to be implemented using logic in the Look Up Tables (LUTs) preceding the output register. This results in increased resource utilization even when asynchronous controls are not present.

  When **Sync Controls override CE** is selected an active level on any of the synchronous control inputs will be acted upon irrespective of the state of the CE pin. This setting is more efficient when asynchronous inputs are not present because it allows the dedicated inputs on the flip-flop primitives to be used for the synchronous control functions. It is less efficient when the presence of asynchronous inputs force the synchronous control functionality to be implemented using logic in the LUTs preceding the output register. This is because the CE signal has to be gated with the synchronous control inputs so that they can all generate a CE signal to the flip-flops, slowing down the CE path and resulting in slower overall operation of the module.

  The default setting is **Sync Controls Override CE** so that a 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 input pin is generated.
  - **Clear**: An ACLR input pin is generated.
  - **Set and Clear**: Both ASET and ACLR input pins are generated. ACLR has priority over ASET when both are asserted at the same time.
  - **Init**: An AINIT input 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 width must be less than or equal to the **Input Bus Width**. If a value is entered that is fewer bits than the data width of the output register it is padded with zeros. An invalid value is highlighted in red in the text box.

  The value specified in this text box also functions as the power on reset value for the output register. The default value is 0.
- **Synchronous Settings**: When no asynchronous controls are implemented (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, see the description of 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. In the case when a non-registered output is not present, this logic can (in some cases) be absorbed into the same LUTs used to implement the gate function. In cases where this 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 input pin is generated.
  - **Clear**: An SCLR input pin is generated.
  - **Set and Clear**: Both SSET and SCLR input pins are generated. SCLR/SSET priority is defined by the setting of the Set/Clear Priority parameter.
  - **Init**: An SINIT input pin is generated which, when asserted, will asynchronously 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 priority of synchronous clear to synchronous set can be controlled. This parameter is only enabled when both synchronous set and synchronous clear have been requested.

  It is not possible for **Set** to override **Clear** when the synchronous control functionality is implemented using the dedicated inputs on the flip-flop primitives. This can only be implemented using logic in the LUTs preceding the output register.

  The default setting is **Clear Overrides Set** so that a more efficient implementation can be generated.
- **Synchronous Init Value**: This text box accepts a hex value whose width must be less than or equal to the **Input Bus Width**. If a value is entered that is fewer bits than the data width of the register 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.

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 operation = adder
CSET port_a_width = 16
CSET port_a_sign = unsigned
CSET port_b_width = 16
CSET port_b_sign = unsigned
CSET port_b_constant = FALSE
CSET port_b_constant_value = 0000
CSET output_options = registered
CSET output_width = 16
CSET carry_borrow_input = TRUE
CSET carry_borrow_output = FALSE
CSET overflow_output = FALSE
CSET bypass = FALSE
CSET ce_override_for_bypass = FALSE
CSET bypass_sense = active_high
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
```

Core Resource Utilization

In all but the cases described below, this module uses one Look Up Table (LUT) per output bus bit. When registered outputs are requested one flip-flop is used per output bus bit. The following cases will use two LUTs per output bus bit in addition to the output register:

- A registered adder or subtractor with all of the following control signals:
  - Asynchronous controls
  - Active Low Bypass
  - Synchronous Set
  - Synchronous Clear
- A registered adder/subtractor with asynchronous controls and any two of the following:
  - Active Low Bypass
  - Synchronous Set
  - Synchronous Clear
  - Synchronous Init
- A registered adder/subtractor with Active High Bypass

When the synchronous control functionality cannot be implemented using the dedicated control inputs of the flip-flop primitive (i.e., when asynchronous controls are also requested) and the **CE Overrides** are set to **Sync Controls Override CE**, an additional LUT per module is required.

Ordering Information

This core is downloadable free of charge from the Xilinx IP Center (www.xilinx.com/ipcenter), for use with version 2.1i 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 2: 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
operation	One of the following keywords: adder, subtractor, adder_subtractor	adder
port_a_width	Integer in the range 1 to 64	16
port_a_sign	One of the following keywords: unsigned, signed	unsigned
port_b_width	Integer in the range 1 to 64	16
port_b_sign	One of the following keywords: unsigned, signed	unsigned
port_b_constant	One of the following keywords: true, false	false
port_b_constant_value	Hex value whose value does not exceed $2^{\text{port\_b\_width}} - 1$	0
output_options	One of the following keywords: non_registered, registered, both	registered
output_width	See Table 2	16
carry_borrow_input	One of the following keywords: true, false	true
carry_borrow_output	One of the following keywords: true, false	false
overflow_output	One of the following keywords: true, false	false
bypass	One of the following keywords: true, false	false
ce_override_for_bypass	One of the following keywords: true, false	true
bypass_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 $2^{\text{output\_width}} - 1$	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 $2^{\text{output\_width}} - 1$	0
set_clear_priority	One of the following keywords: clear_overrides_set, set_overrides_clear	clear_overrides_set