Single-Port Block Memory Core v6.2

DS234 April 28, 2005 — Product Specification



## Features

- Fully synchronous drop-in module for Virtex™, Virtex-II, Virtex-II Pro, Virtex-4, Spartan™-II, Spartan-IIE, Spartan-3, and Spartan-3E FPGAs
- Supports all three Virtex-II write mode options: Read-After-Write, Read-Before-Write; No-Read-On-Write (Available only for Virtex-II, Virtex-II Pro, Virtex-4, Spartan-3, and Spartan-3E implementation)
- Supports ROM and RAM functions
- Supports data widths from 1 to 256 bits
- Supports memory depths 2 to 1M words depending on architecture selected
- Incorporates Xilinx Smart-IP™ technology for utmost parameterization and optimum implementation
- Supports cores designed for area optimization or using a single SelectRAM+™ or SelectRAM-II primitive
- Supports different pin polarities for control signals: clock, enable, write enable and output initialization pins
- Available in the Xilinx CORE Generator™ system v7.1i SP1 and later

## Functional Description

The Single-Port Block Memory module is generated based on the user-specified width and depth. This module for Spartan-II and Virtex is composed of single or multiple 4 Kb blocks called SelectRAM+. The Virtex-II, Virtex-II Pro, Virtex-4, and Spartan-3 Single-Port Block Memory modules, on the other hand, are composed of single or multiple 18 Kb blocks called SelectRAM-II. Since Spartan-II and Virtex both use the 4 Kb SelectRAM+ blocks, any particular reference to a Virtex implementation also applies to a Spartan-II, Virtex-E, Virtex-II Pro, or Spartan-IIE implementation.

Similarly, because Virtex-II, Virtex-II Pro, Virtex-4, and Spartan-3 all use 18 Kb SelectRAM-II blocks, any specific reference to a Virtex-II implementation also applies to a Virtex-II Pro, Virtex-4, or Spartan-3 implementation.

When Block Memory is enabled, all memory operations occur on the active edge of the clock input (CLK). The Block Memory can be configured to be active on the rising edge and the falling edge. When the block memory is disabled (enable inactive), the memory configuration and output value remain unaltered.

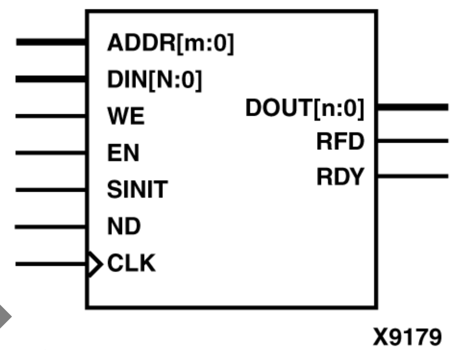


*Figure 1:* **Core Schematic Symbol**

During a write operation (WE asserted), the data presented at the port's data input is stored in memory at the location selected by the port's address input. During this operation, the data output port behaves differently for the Virtex and Virtex-II architectures.

The Virtex implementation supports a single write mode option, Read-After-Write. This write mode causes the data being written to the addressed memory location to be transferred to the data output port when a write operation occurs.

The Virtex-II implementation supports three write mode options to determine the behavior of the data output port (read port) during a write operation:

- Read-After-Write (Write First)
- Read-Before-Write (Read First)
- No-Read-On-Write (No Change)

During a read operation, the memory contents at the location selected by the address will appear at the module's output. When Synchronous Initialization (SINIT) is active, the module's registered outputs are synchronously reset to zero for Virtex and to a user-defined value for Virtex-II. The Synchronous Initialization command has no effect on the contents of the memory or write operations.

The initial contents of the memory (that is, the data stored in the memory immediately after device configuration) can also be specified.

The enable, write enable, and synchronous initialization control signals can also be specified as active high or active low.

For additional information on the BlockRAM implementations, see databook for the selected architecture available at http://www.xilinx.com/partinfo/databook.htm.

*Table 1:* **Core Signal Pinout**

| Signal | Direction | Description |
|---|---|---|
| DIN[n:0] (optional) | Input | **Data Input**: Data written into memory. |
| ADDR [m:0] | Input | **Address**: Memory location for data written to/read from. |
| WE (Optional) | Input | **Write Enable**: Allows data transfer into memory. |
| EN (Optional) | Input | **Enable**: Enables access to memory via read and write operations. |
| SINIT (Optional) | Input | **Synchronous Initialization**: Forces module outputs to a predefined state. |
| CLK | Input | **Clock**: All memory operations synchronous with rising or falling edge of clock input, depending on user configuration of the clock pin polarity. When memory is enabled, all control signals, input/output data are registered on the rising or falling edge of clock. |
| ND (Optional) | Input | **New Data**: Indicates new and valid address on ADDR (Active high). |
| DOUT[n:0] | Output | **Data Output**: Synchronous output of the memory. |
| RFD (Optional) | Output | **Ready for Data**: Indicates that memory is ready for new address. |
| RDY (Optional) | Output | **Output Ready**: Indicates valid data on DOUT port (Active High). |

## Pinout

Port names for the core module are shown in Figure 1 and described in Table 1. The inclusion of some ports on the module is optional; excluding these ports will alter the function of the module. The optional ports are designated in Table 1.

### Clock - CLK

Block Memory is fully synchronous with the clock input. All input pins have setup time referenced to the port CLK pin. The DOUT port has a clock-to-out time referenced to the CLK pin.

By default all memory operations are performed on the rising edge of the clock. Users, however, have the option to perform all memory operations on the rising or the falling edge of the clock. Performing the memory operation on the falling edge of the clock will not use any extra resources.

### Enable - EN

The enable pin affects the read, write, and SINIT functionality of the port. When the Block Memory has an inactive enable pin, the output pins are held in the previous state and writing to the memory is disabled.

By default the enable pin is active high. Users, however, have the option to configure the enable pin active high or active low. Configuring the enable pin active low will not use extra resources.

### Write Enable - WE

Activating the write enable pin enables writing to the memory locations. When active, the contents of the DIN bus is written to memory at the address pointed to by the ADDR bus. The output latches are loaded or not loaded according to the write configuration (Write First, Read First, No Change). When WE is inactive, a read operation occurs, and the contents of the memory addressed by the ADDR bus are driven on the DOUT bus. In the Read Only port configuration (ROM configuration), the WE pin is not available.

By default the write enable pin is active high. Users, however, have the option to configure the write enable pin active high or active low. Configuring the write enable pin active low will not use extra resources.

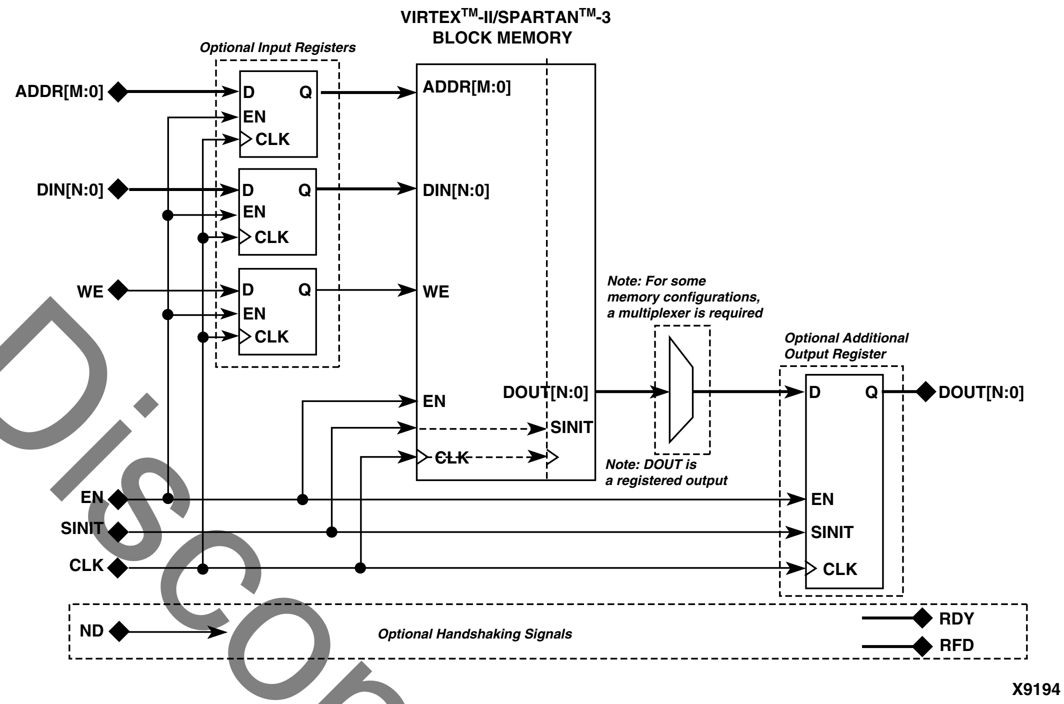


*Figure 2:* **Single-Port Memory Block Diagram**



## Synchronous Initialization - SINIT

When enabled, the SINIT pin forces the data output latches to synchronously load the predefined SINIT value. For the Virtex implementation, the SINIT value is zero. Therefore, asserting the SINIT pin causes the output latches to reset. For the Virtex-II implementation, the SINIT value is defined by the user. Consequently, asserting the SINIT pin causes the output latches to contain the user-defined SINIT value. This operation does not affect memory locations and does not disturb write operations. Like the read and write operation, the SINIT function is active only when the enable pin of the port is active.

By default, the SINIT pin is active high. Users, however, have the option to configure the SINIT pin active high or active low. Configuring the write enable pin active low will not use extra resources.

## Address Bus - ADDR[m:0]

The address bus selects the memory location for read or write access.

## Data-In Bus - DIN[n:0]

The DIN bus provides the data value to be written into the memory. Data input and output signals are always buses; that is, in a 1-bit width configuration, the data input signal is DIN[0] and the data output signal is DOUT[0]. In the Read Only port configuration (ROM configuration), the DIN bus is not available.

## Data-Out Bus - DOUT[n:0]

The DOUT bus reflects the contents of memory locations referenced by the address bus during a read operation.

During a write operation of a Virtex memory (Write First configuration), the DOUT bus reflects the data being written on the DIN bus.

During a write operation of a Virtex-II or Spartan-3 memory (Write First or Read First Configuration), the data-out bus reflects either the DIN bus (Write First) or the current memory contents, previously stored value (Read First). During a write operation in No Change mode, the data-out bus is not affected.

### New Data - ND

Indicates that there is a new and valid address on ADDR Port.

### Ready for Data - RFD

Indicates that the memory is ready to accept a new address. RFD is always true, except when EN is inactive.

### Output Ready (Valid) - RDY

Indicates valid output on the DOUT port. RDY will lag ND by the latency of the block memory.

Figure 3 shows the operation of the ND, RFD, and RDY handshaking signals.

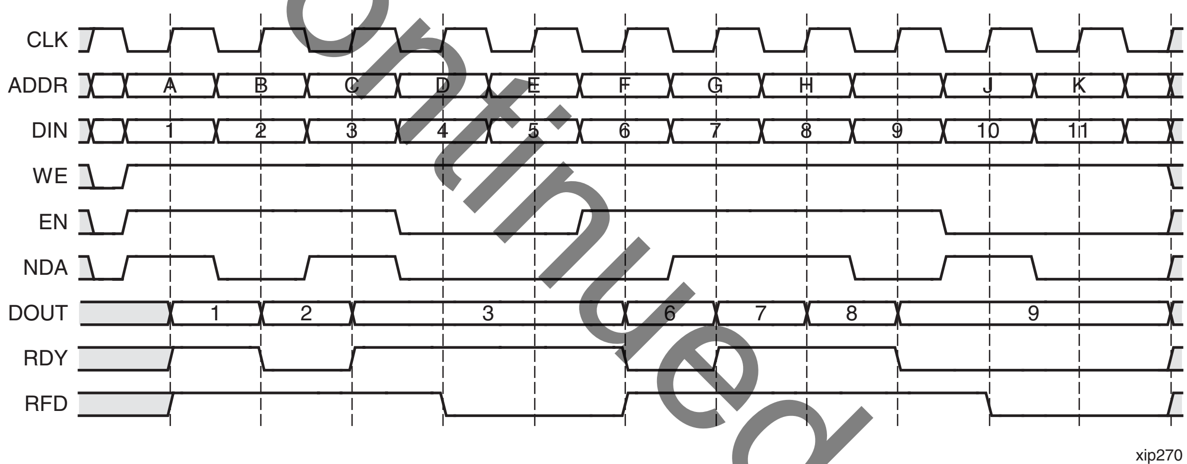


*Figure 3:* **Handshaking Signals Operation (Write-First Mode)**

## CORE Generator Parameters

The main screen of the CORE Generator includes the following parameter selections:

- **Component Name**: Enter a name for the output files generated for this module (up to 256 characters).
- **Port Configuration**: Select one; the default is Read and Write.
  - **Read and Write**: Configured as Random Access Memory (RAM).
  - **Read Only**: Configured as a Read Only Memory (ROM).
- **Port Options**
  - **Width**: Select the data bit width. The width values can be between 1 and 256. Cores should not exceed the number of Block RAM primitives available in the targeted device.

- **Depth**: Enter the number of words in the memory. The range of values is 2 to 1,048,576 (1M) depending on architecture selected. The absolute maximum number of words is 1M for Virtex-II. Blocks used should not exceed the number of Block RAM primitives available in the targeted device.

- **Write Mode**: Select one for Virtex-II architecture. The default is Read-After-Write. The Virtex architecture supports only Read-After-Write.
  - **Read-after-Write** (Virtex-II, Virtex)
    - (1) No Inputs or Outputs Registered: The input data is transferred onto the DOUT port on the active clock edge immediately following the assertion of the WE input.
    - (2) With Inputs Registered Only: The input data is transferred onto the DOUT port on the second active clock edge immediately following the assertion of the WE input.
    - (3) With Outputs Registered Only: The input data is transferred onto the DOUT port on the second active clock edge immediately following the assertion of the WE input.
    - (4) With Inputs and Outputs Registered: The input data is transferred onto the DOUT port on the third active clock edge immediately following the assertion of the WE input.
  - **Read-before-Write** (Virtex-II only)
    - (1) No Inputs or Outputs Registered: The current data in the addressed memory location is transferred onto the DOUT port on the active clock edge immediately following the assertion of the WE input.
    - (2) With Inputs Registered Only: The current data in the addressed memory location is transferred onto the DOUT port on the second active clock edge immediately following the assertion of the WE input.
    - (3) With Outputs Registered Only: The current data in the addressed memory location is transferred onto the DOUT port on the second active clock edge immediately following the assertion of the WE input.
    - (4) With Inputs and Outputs Registered: The current data in the addressed memory location is transferred onto the DOUT port on the third active clock edge immediately following the assertion of the WE input.
  - **No-Read-on-Write** (Virtex-II only)
- (1) A write operation has no effect on the content of the DOUT port. The DOUT port is updated when WE is inactive.

The second screen of the CORE Generator includes the following parameter selections:

- **Implementation Options**
  - **Limit Data Pitch**: Select to limit the data pitch (or data width) of each of the RAM Blocks used to implement the memory. This optimizes the routing implementation of the block memory. Data pitch can be set to 8 or 16 for Virtex and 18 or 36 for Virtex-II.
- **Optional Pins**
  - **Enable Pin**: Check the box to include the enable port on the module; uncheck the box to remove it. This port provides an enable for all memory read and write operations. When it is inactive, the memory is disabled.
  - **Handshaking Pins**: Select the option to include the following ports; deselect to remove them.

- **ND** [New Data]: Signals a new and valid memory address whenever active. This port has no effect on the memory read and write operations. ND is valid only when RFD is active.
- **RFD** [Ready For Data]: Indicates that the memory can accept new addresses. Always active when the memory is enabled.
- **RDY** [Output is Ready]: Indicates to the user that the data on the output is valid. RDY will lag ND by the latency of the module.

- **Register Options**
  - **Register Inputs**: Check this box to register ports DIN, ADDR, and WE prior to accessing block memory. See Figure 2.
- **Output Register Options**
  - **Additional Output Pipe Stages**: Select "1" to enable an additional register on the output of the memory; select "0" to disable an additional register on the output of the memory. See Figure 2.
  - **SINIT PIN**: Check box to add the synchronous port SINIT to the memory. When this signal is active, the output of the memory is set to a predefined value. Enabling this port has no effect on the contents of the memory.
    - **SINIT Value** (HEX value): Enter the HEX value that the output port will get set to when the SINIT port is true. For Virtex implementation, this value is fixed to zero and cannot be altered.

      The Virtex-4 architecture RAMB16 primitive contains embedded output registers. These registers can improve timing for high speed designs. In this core, the embedded registers will only be used when the memory is 1 primitive deep and the SINIT pin is not used. If either the SINIT pin is used or the memory is more than one primitive deep, these embedded registers will not be used. Instead, registers in the slice fabric will be used.

The third screen of the CORE Generator includes the following parameter selections:

- **Primitive selection:** Choose whether the core is optimized for area or created using a single 4kb SelectRAM+ or 16kb SelectRAM-II block or primitive.
  - **Select primitive**: Choose the block or primitive used to create the core.
    - The primitives for the Virtex architectures are: 4kx1, 2kx2, 1kx4, 512x8 and 256x16.
    - The primitives for the Virtex-II architecture are: 16kx1, 8kx2, 4kx4, 2kx9, 1kx18, 512x36.
    - The primitives for the Virtex-4 architecture are: 32kx1, 16kx1, 8kx2, 4kx4, 2kx9, 1kx18, and 512x36.
- **Pin Polarity**: Lets the user configure the polarities of the control signals if the signals exist.
  - **Active Clock Edge**: Select whether the memory operation occurs on the rising edge or falling edge of the clock.
  - **Enable Pin**: Select whether the enable pin is active high or active low.
  - **Write Enable Pin**: Select whether the write enable pin in active high or active low.
  - **Initialization Pin**: Select whether the initialization pin is active high or active low.

The final screen of the CORE Generator GUI includes the following parameter selections:

- **Simulation Model Options**: Deselect to enable warning messages in the simulation model. The default value is to disable warning messages in the simulation model.

- **Initial Contents**: Enter the parameter fields related to the data stored in the memory directly after device configuration.
  - **Global Init Value**: Enter the value to be stored in any memory location not specified by any other means. When no values are entered, this field defaults to 0. Value must be in Hex.
  - **Load Init File**: Selects the initial contents of the memory to be read from a *coe* file.
  - **Load File**: Click to activate a browser window that lets the user select a coefficient or *coe* file containing the initial contents of the memory. (This is an ASCII file with a ".coe" extension.) For more information about the memory's initial contents, see Specifying Memory Contents.
- **Information Panel**: Window provides feedback about memory based on the selected values.
  - **Address Width**: Shows the number of bits needed to address all of the words in the memory.
  - **Blocks Used**: Shows the number of Block RAM primitives required to implement the specified Memory Depth and Width.
  - **Read Pipeline Latency**: Calculates the latency from the address port (ADDR) to the data output port (DOUT).

## Operating Modes

To maximize utilization of the memory at each clock edge, the Virtex-II block SelectRAM-II memory supports three different write modes. The Read-Before-Write mode offers the flexibility of using the data output bus during a write operation on the same port. Output latch values are determined by the configuration. This choice increases the effective bandwidth of the Block Memory.

Note that the Virtex SelectRAM+ supports only the Write First mode.

### Read Operation

Read operations are synchronous to the active edge of the clock. The data in the memory location selected by the address appears on the DOUT port after the active edge of the clock.

### Write Operation

Write operations are synchronous to the active edge of the clock. The data on the DIN port is written into the memory location selected by the address on the active edge of the clock when WE is active. The user can configure the memory in one of the following three ways to determine the behavior of the DOUT port during a write cycle. Note that the timing diagrams and descriptions of the write modes assume that the memory has been configured without input registering and additional output registers.

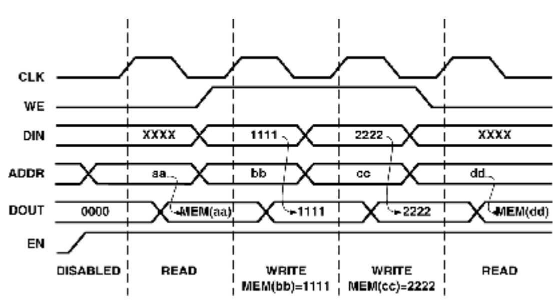


*Figure 4:* **Write First Mode Waveform**

## Write First or Read-After-Write (or Transparent) Mode

In Write First mode, data input is loaded simultaneously with a write operation on the DOUT port. As shown in Figure 4, the data input is stored in memory and mirrored on the output.

## Read First or Read-Before-Write Mode

In this mode, data previously stored at the write address appears on the output latches. Data input is stored in memory and the prior content of that location driven on the output, during the same clock cycle (shown in Figure 5).

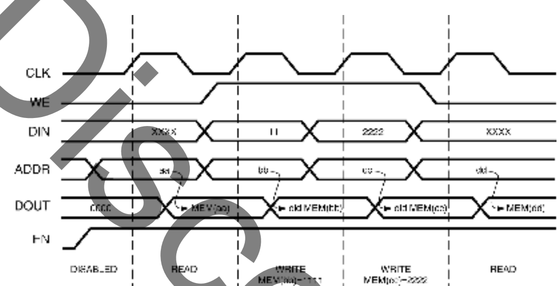


*Figure 5:* **Read First Mode Waveform**

## No Change or No-Read-On-Write Mode

In No Change mode, the DOUT port remains unchanged during a write operation. As shown in Figure 6, data output is still the last read data and is unaffected by a write operation on the same port.

Mode configuration is static. One of these three modes is set individually for each port by an attribute. The default mode is write first.

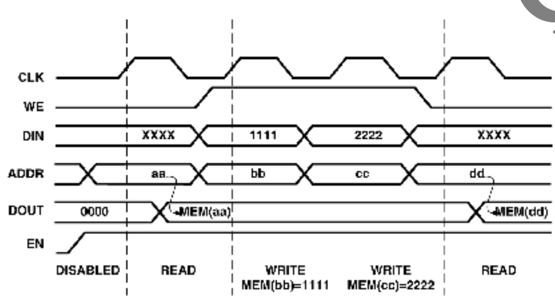


*Figure 6:* **No Change on Write Mode Waveform**

## Specifying Memory Contents

The initial contents of the memory can be assigned by specifying the desired information in a separate text file called a *coe* file. To select and load a *coe* file, click Load Init Values on the parameterization screen; then choose the desired file from the from dialog box. An example of a *coe* file for a 3 by 16 RAM is shown in Figure 7.

MEMORY_INITIALIZATION_RADIX=16;

MEMORY_INITIALIZATION_VECTOR=1

*Figure 7:* **An example of a *coe* file for a Virtex-II Single-Port Block RAM**

When specifying the initial contents for a memory in a *coe* file, the keywords MEMORY_INITIALIZATION_RADIX and MEMORY_INITIALIZATION_VECTOR can be used. The MEMORY_INITIALIZATION_VECTOR takes the form of a sequence of comma-separated values, one value per memory location, terminated by a semicolon. Any amount of white space, including new lines, can be included in the vector to enhance readability. The format of an individual value in the vector will depend on the MEMORY_INITIALIZATION_RADIX value, which can be 2, 10, or 16 (the default value is 10). The vector must be consistent with the MEMORY_INITIALIZATION_RADIX value and must fall within the range of 0 to $2^{DATA\_WIDTH}$-1. Values must not be negative. Note that the first entry in the coe file corresponds to the lowest block memory address.

If the initial contents for a memory is specified by a coe file, the initial values will be embedded in the EDIF netlist that is needed for implementation. To support HDL simulations, MIF files containing the initialization values are generated. These files must be copied to the active simulation directory for a successful simulation of a single port block memory core.

## Parameter Values in the XCO File

Names of the XCO 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 and Table 3 show 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 width = 16
CSET depth = 256
CSET port_configuration = read_and_write
CSET write_mode = read_before_write
CSET global_init_value = 456a
CSET load_init_file = true
CSET coefficient_file = example.coe
CSET enable_pin = false
CSET handshaking_pins = true
CSET register_inputs = true
CSET additional_output_pipe_stages = 0
CSET init_pin = false
CSET init_value = 1234
CSET has_limit_data_pitch=false
CSET limit_data_pitch=18
CSET primitive_selection = optimize_for_area
CSET select_primitive = 4kx1
CSET enable_pin_polarity = active_high
CSET initialization_pin_polarity = active_high
CSET write_enable_pin_polarity = active_high
CSET active_clock_edge = rising_edge_triggered
```

*Table 2:* **Parameter File Information for Virtex-II**

| Parameter Name | XCO Filename Values | Default GUI Strings |
|---|---|---|
| component_name | ASCII text starting with a letter and based upon the following character set: a..z, 0..9 and _. | blank |
| width | Integer in the range of 2 to 256 | 16 |
| depth | Integer in the range of 2 to 1 M (256K for Spartan-3) | 256 |
| port_configuration | One of the following keywords: read_and_write, read_only | read_and_write |
| write_mode | One of the following keywords: read_before_write, read_after_write and no_read_on_write | read_after_write |
| global_init_value | A hex value in the range of 0 to $2^{width} - 1$ | 0 |
| load_init_file | One of the following keywords: true, false | false |
| coefficient_file | The name of the coe file in ASCII text starting with a letter and based upon the following character set: a..z, 0..9 and _. | blank |
| enable_pin | One of the following keywords: true, false | false |
| handshaking_pins | One of the following keywords: true, false | false |
| register_inputs | One of the following keywords: true, false | false |
| additional _output_pipe_stages | Integer in the range of 0 to 1 | 0 |
| init_pin | One of the following keywords: true, false | false |
| init_value | A hex value in the range of 0 to $2^{width} - 1$ | 0 |
| has_limit_data_pitch | One of the following keywords: true, false | false |
| limit_data_pitch | One of the two values: 18, 36 | 18 |
| primitive_selection | One of two values: optimize_for_area, select_primitive | optimize_for_area |
| select_primitive | 16kx1, 8kx2, 4kx4, 2kx9, 1kx18, 512x36 | 16kx1 |
| enable_pin_polarity | One of two values: active_high, active_low | active_high |
| initialization_pin_polarity | One of two values: active_high, active_low | active_high |
| active_clock_edge | One of two values: rising_edge_triggered, falling_edge_triggered | rising_edge_triggered |
| write_enable_pin_polarity | One of two values: active_high, active_low | active_high |
| disable_working_messages | One of the following keywords: true, false | true |

*Table 3:* **Parameter File Information for Virtex-4**

| Parameter Name | XCO Filename Values | Default GUI Strings |
|---|---|---|
| component_name | ASCII text starting with a letter and based upon the following character set: a..z, 0..9 and _. | blank |
| width | Integer in the range of 2 to 256 | 16 |
| depth | Integer in the range of 2 to 256K | 256 |
| port_configuration | One of the following keywords: read_and_write, read_only, write_only | read_and_write |
| write_mode | There is only one options for Spartan-II/Virtex architecture: read_after_write | read_after_write |
| global_init_value | A hex value in the range of 0 to $2^{width} - 1$ | 0 |
| load_init_file | One of the following keywords: true, false | false |
| coefficient_file | The name of the coe file in ASCII text starting with a letter and based upon the following character set: a..z, 0..9 and _. | blank |
| enable_pin | One of the following keywords: true, false | false |
| handshaking_pins | One of the following keywords: true, false | false |
| register_inputs | One of the following keywords: true, false | false |
| additional _output_pipe_stages | Integer in the range of 0 to 1 | 0 |
| init_pin | One of the following keywords: true, false | false |
| init_value | A hex value in the range of 0 to $2^{width} - 1$ | 0 |
| has_limit_data_pitch | One of the following keywords: true, false | false |
| limit_data_pitch | One of the two values: 8, 16 | 8 |
| primitive_selection | One of two values: optimize_for_area, select_primitive | optimize_for_area |
| select_primitive | 32kx1, 16kx1, 8kx2, 4kx4, 2kx9, 1kx18, 512x36 | 32kx1 |
| enable_pin_polarity | One of two values: active_high, active_low | active_high |
| initialization_pin_polarity | One of two values: active_high, active_low | active_high |
| active_clock_edge | One of two values: rising_edge_triggered, falling_edge_triggered | rising_edge_triggered |
| write_enable_pin_polarity | One of two values: active_high, active_low | active_high |
| disable_working_messages | One of the following keywords: true, false | true |

*Table 4:* **Parameter File Information for Virtex**

| Parameter Name | XCO Filename Values | Default GUI Strings |
|---|---|---|
| component_name | ASCII text starting with a letter and based upon the following character set: a..z, 0..9 and _. | blank |
| width | Integer in the range of 2 to 256 | 16 |
| depth | Integer in the range of 2 to 1 M (256K for Spartan-3) | 256 |
| port_configuration | One of the following keywords: read_and_write, read_only | read_and_write |
| write_mode | One of the following keywords: read_before_write, read_after_write and no_read_on_write | read_after_write |
| global_init_value | A hex value in the range of 0 to $2^{width} - 1$ | 0 |
| load_init_file | One of the following keywords: true, false | false |
| coefficient_file | The name of the coe file in ASCII text starting with a letter and based upon the following character set: a..z, 0..9 and _. | blank |
| enable_pin | One of the following keywords: true, false | false |
| handshaking_pins | One of the following keywords: true, false | false |
| register_inputs | One of the following keywords: true, false | false |
| additional _output_pipe_stages | Integer in the range of 0 to 1 | 0 |
| init_pin | One of the following keywords: true, false | false |
| init_value | A hex value in the range of 0 to $2^{width} - 1$ | 0 |
| has_limit_data_pitch | One of the following keywords: true, false | false |
| limit_data_pitch | One of the two values: 18, 36 | 18 |
| primitive_selection | One of two values: optimize_for_area, select_primitive | optimize_for_area |
| select_primitive | 4kx1, 2kx2, 512x8, 512x16 | 4kx1 |
| enable_pin_polarity | One of two values: active_high, active_low | active_high |
| initialization_pin_polarity | One of two values: active_high, active_low | active_high |
| active_clock_edge | One of two values: rising_edge_triggered, falling_edge_triggered | rising_edge_triggere d |
| write_enable_pin_polarity | One of two values: active_high, active_low | active_high |
| disable_working_messages | One of the following keywords: true, false | true |

## Core Resource Utilization

The number of Block RAM primitives required depends on the values of data depth and width selected in the CORE Generator parameterization window.

For Virtex implementation, this value must be at least (width x depth)/4096; while for Virtex-II implementation, this value must be at least (depth x width)/18432. Note that for many configurations, the number of Block RAMs will exceed this estimated value.

For some memory depths, extra logic is required to decode the address and multiplex the outputs from various primitives. Virtex or Virtex-II CLB slices are used to provide this functionality. The number of slices required depends on the way that the depth is constructed from the primitives, the data width, and the implementation of any decoding or multiplexing.

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

For more information about the number of block RAMs in each device, see Tables 5 through 11.

*Table 5:* **Spartan-II Device Block RAM Counts**

| Devices | # Blocks | Total Block (bits) |
|---|---|---|
| XC2S15 | 4 | 16,384 |
| XC2S30 | 6 | 24,576 |
| XC2S50 | 8 | 32,768 |
| XC2S100 | 10 | 40,960 |
| XC2S150 | 12 | 49,152 |
| XC2S200 | 14 | 57,344 |

*Table 6:* **Virtex Device Block RAM Counts**

| Devices | # Blocks | Total Block (bits) |
|---|---|---|
| XCV50 | 8 | 32,768 |
| XCV100 | 10 | 40,910 |
| XCV150 | 12 | 49,152 |
| XCV200 | 14 | 57,344 |
| XCV300 | 16 | 65,536 |
| XCV400 | 20 | 81,920 |
| XCV600 | 24 | 98,304 |
| XCV800 | 28 | 114,688 |
| XCV1000 | 32 | 131,072 |

*Table 7:* **Virtex-E Device Block RAM Counts**

| Devices | # Blocks | Total Block (bits) |
|---|---|---|
| XCV50E | 16 | 65,536 |
| XCV100E | 20 | 81,920 |
| XCV200E | 28 | 114,688 |
| XCV300E | 32 | 131,072 |
| XCV400E | 40 | 163,840 |
| XCV600E | 72 | 294,912 |
| XCV1000E | 96 | 393,216 |
| XCV1600E | 144 | 589,824 |
| XCV2000E | 160 | 655,360 |
| XCV2600E | 184 | 753,664 |
| XCV3200E | 208 | 851,968 |
| XCV405E | 140 | 573,440 |
| XCV812E | 280 | 1,146,880 |

*Table 8:* **Virtex-II Device Block RAM Counts**

| Devices | # Blocks | Total Block (Kb) |
|---|---|---|
| XC2V40 | 4 | 73,728 |
| XC2V80 | 8 | 147,456 |
| XC2V250 | 24 | 442,368 |
| XC2V500 | 32 | 589,824 |
| XC2V1000 | 40 | 737,280 |
| XC2V1500 | 48 | 884,736 |
| XC2V2000 | 56 | 1,032,192 |
| XC2V3000 | 96 | 1,769,472 |
| XC2V4000 | 120 | 2,211,840 |
| XC2V6000 | 144 | 2,654,208 |
| XC2V8000 | 168 | 3,096,576 |

*Table 9:* **Virtex-II Pro Device Block RAM Counts**

| Devices | # Blocks | Total Block (Kb) |
|---|---|---|
| XC2VP2 | 12 | 221,184 |
| XC2VP4 | 28 | 516,096 |
| XC2VP7 | 44 | 811,008 |
| XC2VP20 | 88 | 1,622,016 |

*Table 9:* **Virtex-II Pro Device Block RAM Counts** ***(Continued)***

| | | |
|---|---|---|
| XC2VPX20 | 88 | 1,622,016 |
| XC2VP30 | 136 | 2,506,752 |
| XC2VP40 | 192 | 3,538,944 |
| XC2VP50 | 232 | 4,276,224 |
| XC2VP70 | 328 | 6,045,696 |
| XC2VPX70 | 308 | 5,677,056 |
| XC2VP100 | 444 | 8,183,808 |
| XC2VP125 | 556 | 10,248,192 |

*Table 10:* **Spartan-II E Device Block RAM Counts**

| Devices | # Blocks | Total Block (Kb) |
|---|---|---|
| XC2S50E | 8 | 32,768 |
| XC2S100E | 10 | 40,960 |
| XC2S150E | 12 | 49,152 |
| XC2S200E | 14 | 57,344 |
| XC2S300E | 16 | 65,636 |
| XC2S400E | 40 | 163,840 |
| XC2S600E | 72 | 294,912 |

*Table 11:* **Spartan-3 Device Block RAM Counts**

| Devices | # Blocks | Total Block (Kb) |
|---|---|---|
| XC3S50 | 4 | 73,728 |
| XC3S200 | 12 | 221,184 |
| XC3S400 | 16 | 294,912 |
| XC3S1000 | 24 | 442,368 |
| XC3S1500 | 32 | 589,824 |
| XC3S2000 | 40 | 737,280 |
| XC3S4000 | 96 | 1,769,472 |
| XC3S5000 | 104 | 1,916,928 |

## 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 Xilinx CORE Generator system is bundled with the ISE Foundation software 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.

## Revision History

| Date | Version | Revision |
|---|---|---|
| 5/21/04 | 1.0 | Revision History added to document. |
| 5/21/04 | 1.1 | Added support for Virtex-4 and v6.2i of Xilinx Core Generator system. |
| 04/28/05 | 1.2 | Added support for Spartan-3E and Xilinx software v7.1i. |