Single Port Block RAM

July 17, 1998 Product Specification

Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
E-mail: coregen@xilinx.com
URL: www.xilinx.com

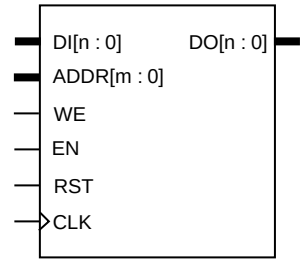

Figure 1: Core Schematic Symbol

Features

- Supports data widths from 1 to 512 bits
- Supports memory depths from 256 to 4096 words
- Uses Virtex™ block memory for performance and efficiency
- Allows power-on memory content to be defined
- Fully synchronous
- Drop-in modules for the Virtex™ family
- Available in Xilinx CORE Generator

Functional Description

This module takes an N-bit data value and an M-bit address. When the RAM is disabled (EN =0) the memory contents and output value remain unaltered. When enabled (EN =1) all memory operations occur on the rising edge of the clock input (CLK). During a write operation (WE = 1) the data value is stored at the location selected by the address. If the memory is not in reset mode (RST = 0) the data value will also appear at the module's output. During a read operation (WE = 0 and RST = 0) the memory contents at the location selected by the address will appear at the module's output. While in reset mode (RST = 1) the module's outputs are all held LOW, although memory write operations may still take place. (Simply asserting RST has no effect on memory contents.)

The initial contents of the memory (i.e. the data stored in the memory immediately after device configuration) may also be specified.

Pinout

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

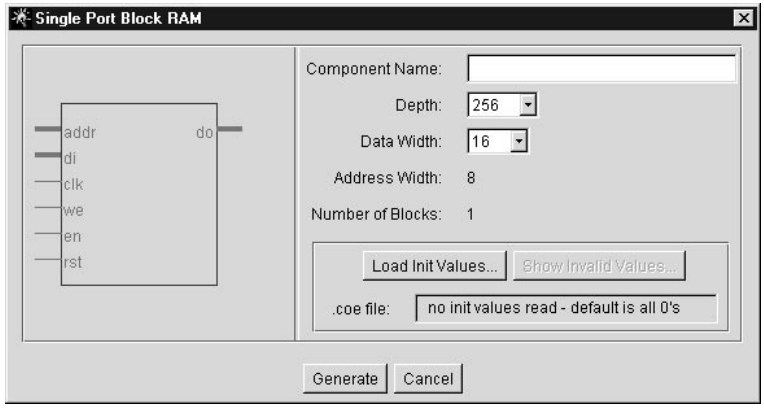

Figure 2: Parameterization Window

CORE Generator Parameters

The CORE Generator parameterization window for this macro is shown in Figure 2. The parameters are as follows:

- **Component Name:** Enter a name for the output files generated for this module.
- **Depth:** Select the number of words in the RAM from the pull down menu. The entries in this field constrain the values that may be entered in the data-width field. Table 2 illustrates the relationship between depth and width.
- **Data Width:** Select the input bit width from the pull down menu. The entries in this field are constrained by the value entered in the depth field. Table 2 illustrates the relationship between depth and width.
- **Address Width:** Shows the number of bits needed to address all of the locations in the RAM. This field is read only.
- **Blocks*:** Shows the number of RAM primitives required to implement the specified Depth and Data width. Table 2 also shows this information. This field is read only.
- **Load Init Values:** Specifies the file that contains the initial values for the RAM.
- **Show Invalid Values:** Display any initial values that are invalid given the chosen parameters, after they have been loaded.
- **coe file:** Displays the name of the coefficient file. This field is read only.

*Note: Ensure that the target device has sufficient memory blocks to accommodate the specified memory, including any memory blocks used elsewhere in your application. (Each RAM primitive is equivalent to 4096 bits of storage.) Table 2 shows how many memory blocks are available in each device.

Table 1: Core Signal Pinout

Signal	Signal Direction	Description
DI[n:0]	Input	DATA INPUT – data to be written into memory.
ADDR[m:0]	Input	RAM ADDRESS – memory location to which data will be written or from which data will be read.
WE	Input	WRITE ENABLE – active high signal used to allow transfer of data into memory.
EN	Input	ENABLE – active high signal used to allow read/write or reset mode operations to take place within memory.
RST	Input	RESET – active high signal used to force the modules outputs LOW. Does not affect RAM contents.
CLK	Input	CLOCK – when RAM is enabled, control and data inputs are registered, and new output data formed on the rising clock edge.
DO[n:0]	Output	DATA OUTPUT – when RAM is enabled (EN=1) this port reflects data stored at the location selected by the address. LOW during reset (RST=1). When RAM is disabled (EN=0), maintains the previous output data value.

Table 2: Block RAM Widths Available By Depth and Number of Blocks

	Memory Depth					Smallest
Blocks	**256**	**512**	**1024**	**2048**	**4096**	**Device**
1	16	8	4	2	1	
2	32	16	8	4	2	
3	48	24	12	6	3	
4	64	32	16	8	4	
5	80	40	20	10	5	
6	96	48	24	12	6	
7	112	56	28	14	7	
8	128	64	32	16	8	XCV50
9	144	72	36	18	9	
10	160	80	40	20	10	XCV100
11	176	88	44	22	11	
12	192	96	48	24	12	XCV150
13	208	104	52	26	13	
14	224	112	56	28	14	XCV200
15	240	120	60	30	15	
16	256	128	64	32	16	XCV300
17	272	136	68	34	17	
18	288	144	72	36	18	
19	304	152	76	38	19	
20	320	160	80	40	20	XCV400
21	336	168	84	42	21	
22	352	176	88	44	22	
23	368	184	92	46	23	
24	384	192	96	48	24	XCV600
25	400	200	100	50	25	
26	416	208	104	52	26	
27	432	216	108	54	27	
28	448	224	112	56	28	XCV800
29	464	232	116	58	29	
30	480	240	120	60	30	
31	496	248	124	62	31	
32	512	256	128	64	32	XCV1000

Specifying Memory Contents

The initial contents of the RAM can be assigned by specifying the desired information in a text file – known as a COE file. In addition to the initial memory contents, all the parameters visible on the parameterization window may be assigned values in the COE file. COE files may take any root file name but must end with the extension ".COE".

To select and load a COE file, press the "Load Init Values…" button on the parameterization window and choose the desired file from the dialog-box. Any field on the parameterization window that is assigned a value in the COE file will lose its previous value when the COE file is loaded. Changing a parameter value that was previously loaded from a COE file causes the COE file's name to be highlighted in red, indicating that the settings have changed since the file was loaded. If any of the initialization values are inconsistent with the other parameters specified, an error is issued. The inconsistent data can then be reviewed by pressing the "Show Invalid Values…" button, which will now be highlighted in red.

For a detailed description of the COE file syntax, please refer to the Xilinx CORE Generator User Guide. The COE keywords supported by the Single Port RAM module, are shown in the Parameter File Information table at the end of this data sheet. An example COE file is shown below in Figure 3.

When specifying the initial contents for a memory in a COE file, the keywords **DEFAULT_DATA**, **MEMORY_INITIALIZATION_VECTOR** and **RADIX** may be used. The **DEFAULT_DATA** keyword allows a value to be assigned to all memory locations with a single statement. If not set, the **DEFAULT_DATA** value is 0. (In general, data values that require fewer than **DATA_WIDTH** bits to express will be padded with "0"s at their most significant end. In the example below the **DEFAULT_DATA** value "F" is assumed to be "000F".) The **DEFAULT_DATA** value is overridden by the **MEMORY_INITIALIZATION_VECTOR** but only for the memory locations covered by the vector. The **MEMORY_INITIALIZATION_VECTOR** takes the form of a sequence of comma separated values, one value per memory location, terminated by a semi-colon. 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 **RADIX** value, which can be "2", "10" or "16" (the default value is 16). The vector values must be consistent with the **RADIX** value and must fall within the range 0 to 2^{DATA_WIDTH}-1. Values may not be negative.

```
Component_Name=ram256x16;
Data_width =16;
Depth =256;
Radix =16;
Default_Data =F;
Memory_Initialization_Vector =123,
456, aaaa;
```

Figure 3: An example COE file for Single Port Block RAM

HDL Simulation

The behavioral models (VHDL and Verilog) for the block memory components initialize their memory contents by reading a memory initialization file. This file, <component_name>.mif, is written into the project directory when the block memory module is generated. It is therefore insufficient to request only a behavioral model for a block memory component, you must also request an EDIF netlist to ensure that the module, and therefore the

.mif file are created. Both the Verilog and the VHDL behavioral models read the same .mif file which must be present in your simulator project directory before simulation begins. If the file is not present the simulator will issue an error and halt.

Core Resource Utilization

The number of RAM primitives required to implement a particular memory is dependent on the values of the depth and data width fields selected in the CORE Generator Parameterization Window, and is equal to:

(depth*data_width)/4096.

Table 2 specifies the number of blocks required for the allowable depth and data width values. The table also details the smallest device, in the Virtex family, that provides a particular number of primitives.

Ordering Information

This macro comes free with the Xilinx CORE Generator. For additional information contact your local Xilinx sales representative, or e-mail requests to: coregen@xilinx.com.

Parameter File Information

Parameter Type	Type	Notes
Component_Name	String	
Depth	Integer	See Table 2
Data_Width	Integer	See Table 2
Memory_Initialization_Vector	Integer List	Comma separated and semi-colon terminated
Default_Data	Integer	
Radix*	Integer	2, 10 or 16

*COE file only