Block RAM

2013 – 2014 - 2020

Ando Ki, Ph.D.
(adki@future-ds.com)

Agenda

- On-Chip and On-board Memories
- Memories for FPGA
- Distributed RAM: Xilinx FPGA LUT as a memory
- Distributed RAM using LUT
- BRAM for Xilinx Spartan-3
- Block RAM Waveforms
- Vivado IP manager
- Bram interface types
- Generated files
- Example: memory using BRAM
- How to fill contents
- Xilinx BRAM coefficient file
- Example: memory with initial value
- Projects
- References

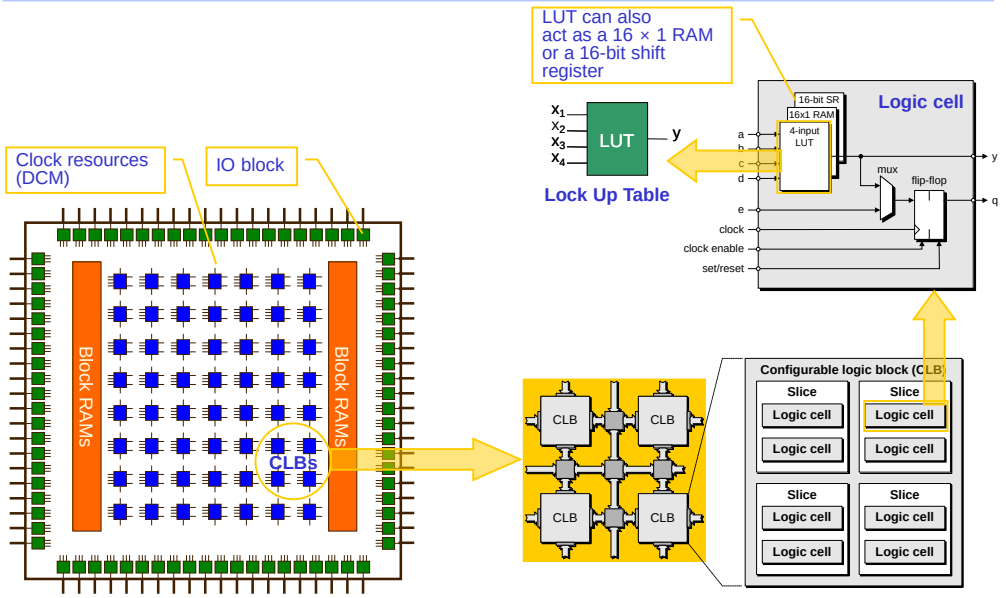
On-Chip and On-board Memories

- On-board memories (off-chip memories)
  - Volatile memory
  - Non-volatile memory
  - Static memory
  - Dynamic memory
  - Synchronous memory
  - Asynchronous memory
  - SDR (single data rate) memory
  - DDR (double data rate) memory
  - DDR2, DDR3
- On-chip memories
  - RAM
  - ROM
  - OTP (One-Time Programmable)

Memories for FPGA

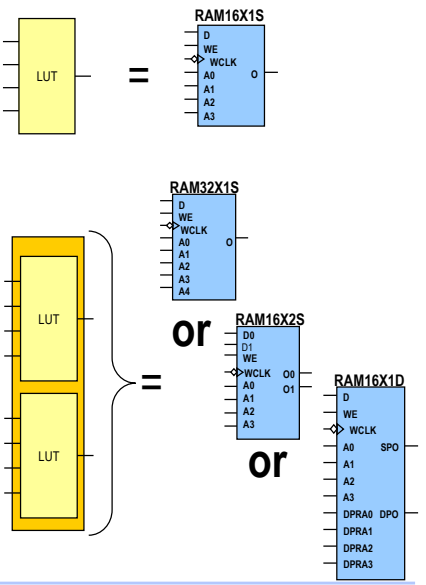
- FPGA normally has two types of memory elements.
  - Distributed RAM takes the form of flip-flops that are created within the general-purpose logic fabric of the chip.
    - This is good for storing small amounts of data, making registers, shift registers, etc. Block RAM is dedicated, configurable memory with address, data, and control ports.
  - BRAM (Block RAM)
- Good things of BRAM
  - Fast,
  - Contents can be initialized at mapping time
- Bad things of BRAM
  - Small of size

Distributed RAM: Xilinx FPGA LUT as a memory



Distributed RAM using LUT

- CLB LUT configurable as Distributed RAM
  - An LUT equals 16x1 RAM
  - Cascade LUTs to increase RAM size
- Synchronous write
- Asynchronous read
  - Can create a synchronous read by using extra flip-flops
  - Naturally, distributed RAM read is asynchronous
- Two LUTs can make
  - 32 x 1 single-port RAM
  - 16 x 2 single-port RAM
  - 16 x 1 dual-port RAM



BRAM for Xilinx Spartan-3

- Xilinx Spartan-3 has multiple block RAM memories, organized in columns.
- Each block RAM contains 18,432 bits (18kbit) of fast static RAM, 16K bits of which is allocated to data storage (i.e., 2K byte) and, in some memory configurations, an additional 2K bits allocated to parity.
  - BRAM can be instantiated using the appropriate "RAMB16" module from the Xilinx design library: RAM16_S9
  - XC3S1000: 24 BRAMs (432Kbits)
  - XC3S2000: 40 BRAMs (720Kbits = 40x18Kbit)
  - XC3S5000: 104 BRAMs (1,827Kbits)

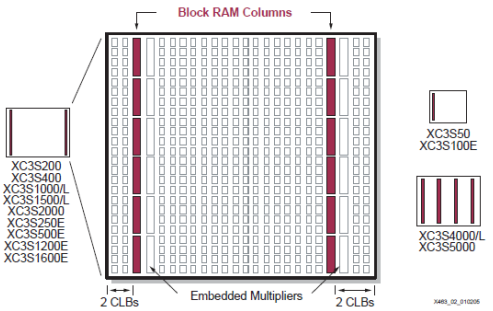


Figure 2: **Block RAMs Arranged in Columns with Detailed Floorplan of XC3S200**

BRAM for Xilinx Spartan-3

- BRAM is true dual-port memory, but can be configured to single-port memory.

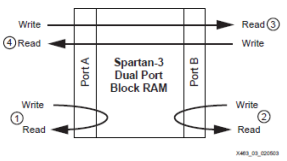
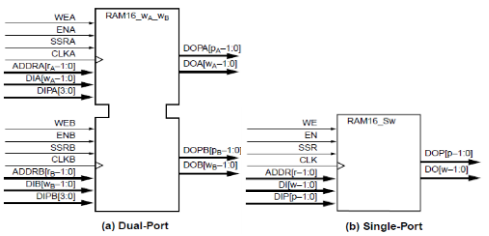


Figure 3: **Block RAM Support Single- and Dual-Port Data Transfers**



BRAM for Xilinx Spartan-3

- Cascade multiple BRAMs to create deeper and wider memory organizations
- A single BRAM can be used different ways as follows.

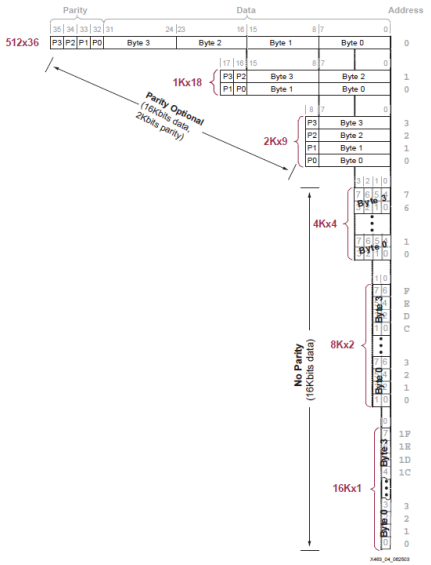
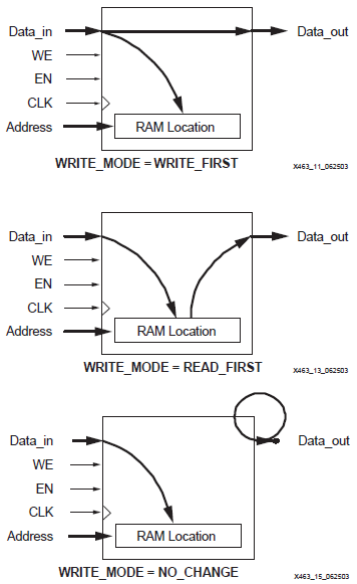


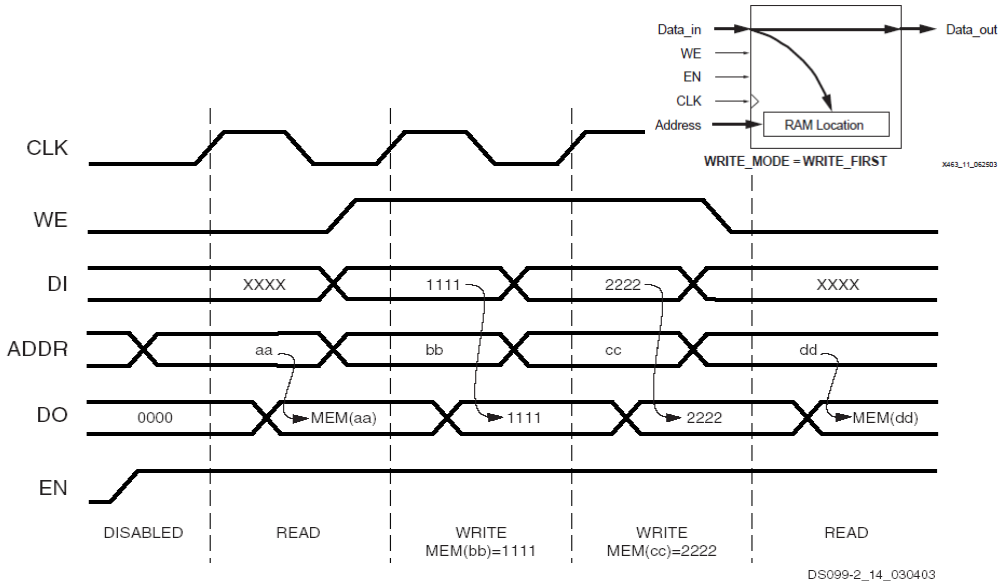
Figure 4: **Data Organization and Mapping Between Modes**

BRAM for Xilinx Spartan-3

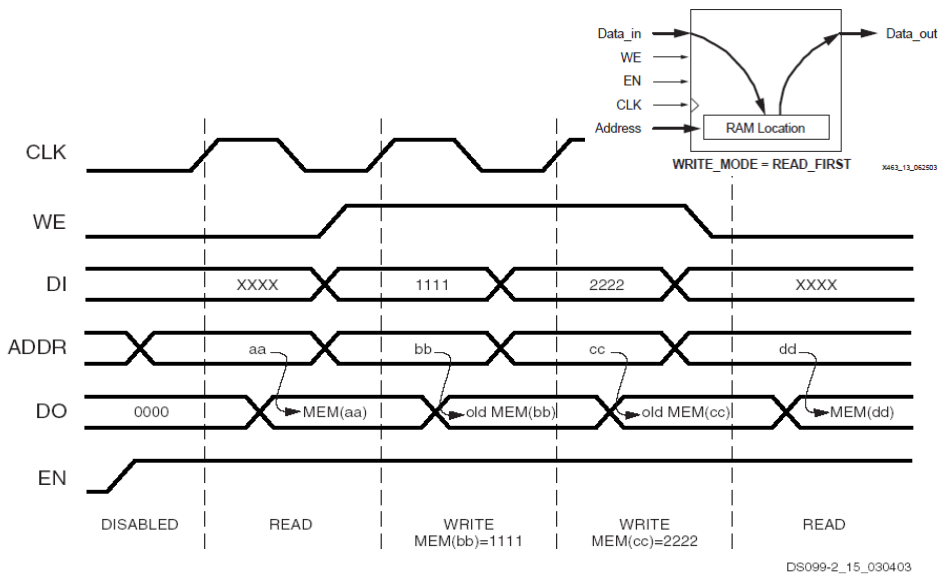
- Write mode
  - WRITE_FIRST: Read After Write (Default)
  - READ_FIRST: Read Before Write
  - NO_CHANGE: No Read on Write



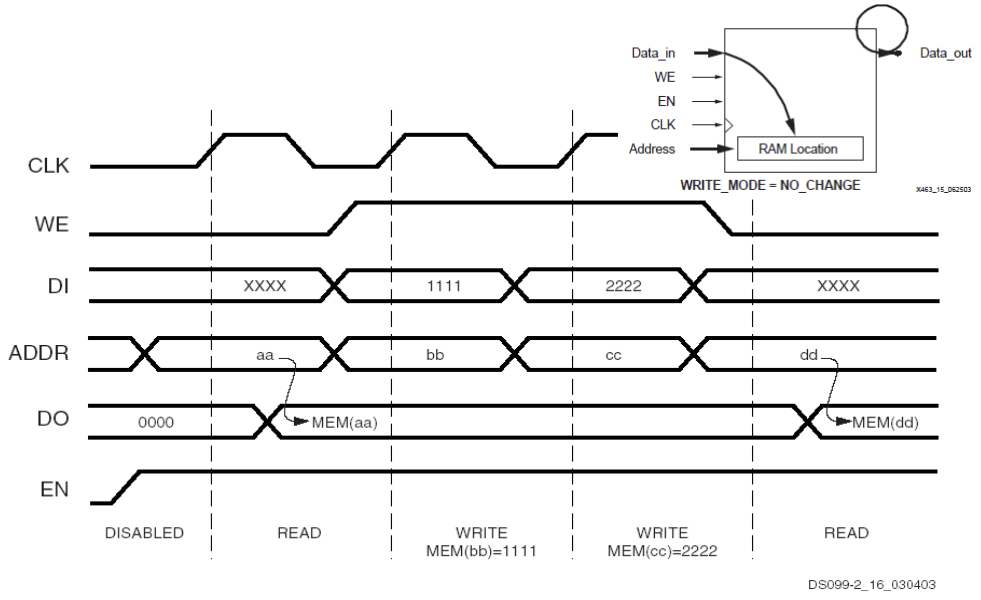
Block RAM Waveforms – WRITE_FIRST mode



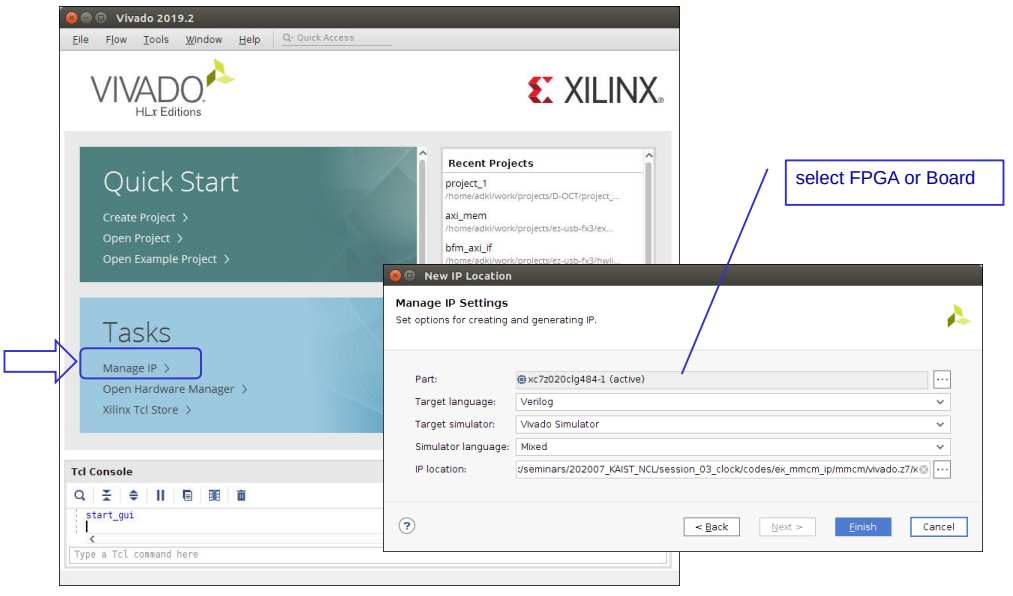
Block RAM Waveforms – READ_FIRST mode



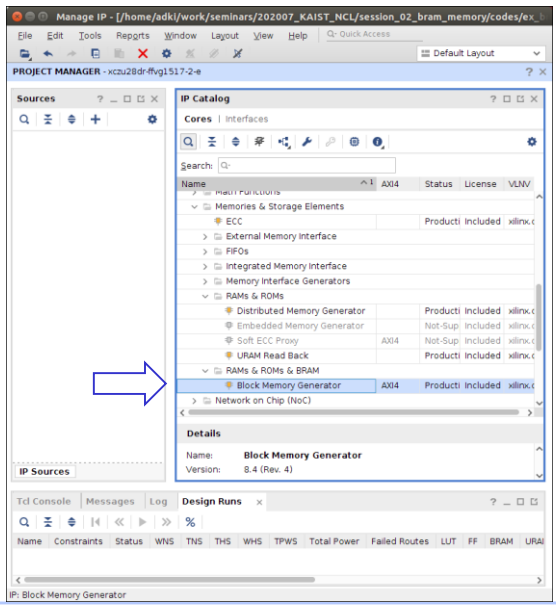
Block RAM Waveforms – NO_CHANGE mode



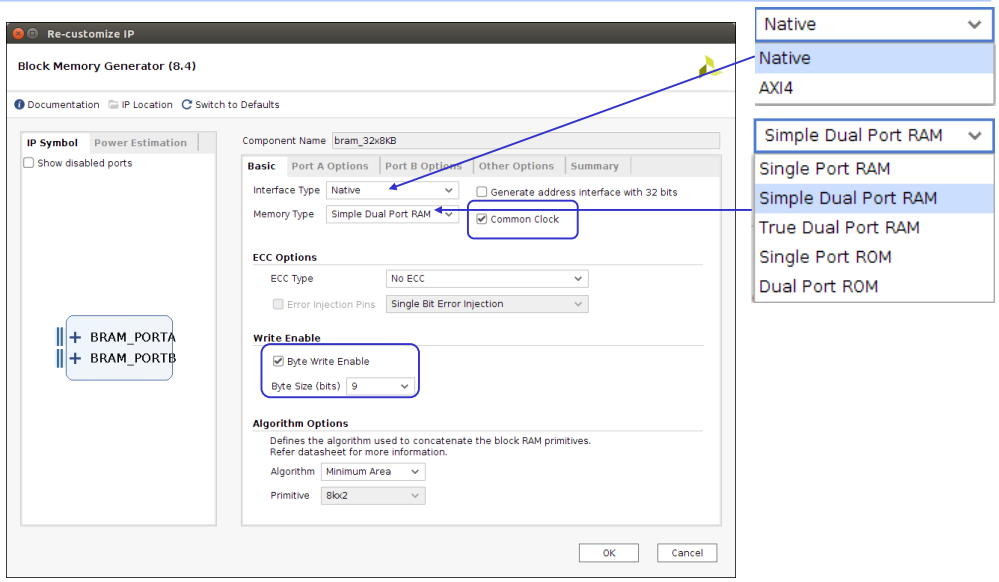
Vivado IP manager



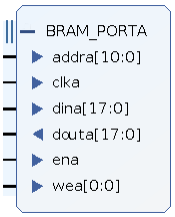
Vivado IP manager



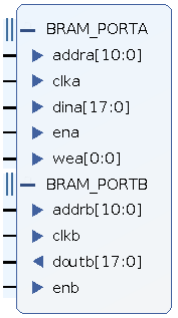
Vivado IP manager



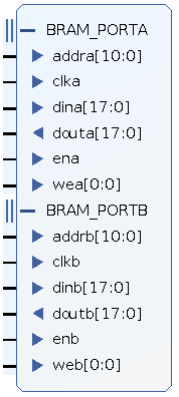
Bram interface types



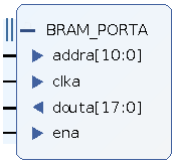
Single port RAM



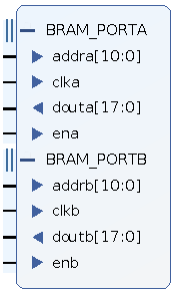
Simple dual port RAM
(one for write; one for read)



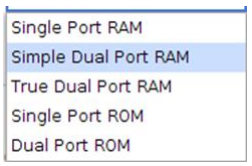
True dual port RAM
(read/write both)



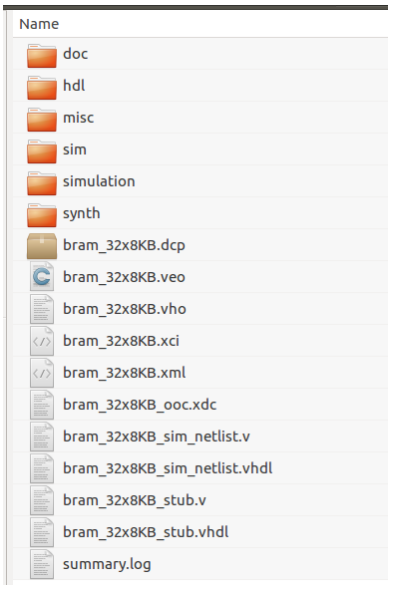
Single port ROM



Dual port ROM



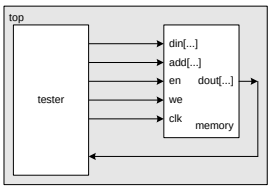
Generated files



- DCP
  - Design check point
  - a sort of archive file
- VEO, VHO
  - Verilog/VHDL template file
- XCI
  - Xilinx Core Instance
  - a kind of project file
- XDC
  - constratint
- sim_netlist.v, sim_netist.vhld
  - Gate-level simulation
- stub.v, stub.vhdl
  - E.g., black box

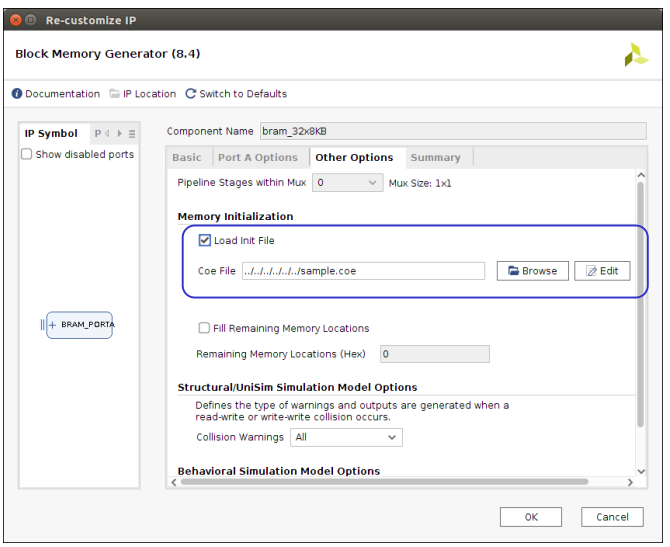
Example: memory using BRAM

- Run an example code as follows
  - Prepare a single port BRAM of 8KByte with 32-bit data width
  - Run simulation and check VCD
    - do not forget to add '-testplusarg VCD=1' option for 'xsim', which enables to generate VCD file




```
[user@host] cd $(PROJECT)/codes/bram/ex_bram/sim/xsim
[user@host] make
[user@host] gtkwave wave.vcd &
```


How to fill contents



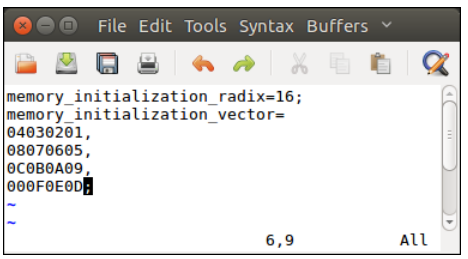
Xilinx BRAM coefficient file

```
*****************************************************************
*********  Example of Dual Port Block Memory .COE file  **********
*****************************************************************
; Sample memory initialization file for Dual Port Block Memory,
; v3.0 or later.
;
; This .COE file specifies the contents for a block memory
; of depth=16, and width=4.  In this case, values are specified
; in hexadecimal format.
memory_initialization_radix=2;
memory_initialization_vector=
1111,
1111,
1111,
1111,
1111,
0000,
0101,
0011,
0000,
1111,
1111,
1111,
1111,
1111,
1111,
1111;
```

```
*****************************************************************
********  Example of Single Port Block Memory .COE file  *********
*****************************************************************
; Sample memory initialization file for Single Port Block Memory,
; v3.0 or later.
;
; This .COE file specifies initialization values for a block
; memory of depth=16, and width=8. In this case, values are
; specified in hexadecimal format.
memory_initialization_radix=16;
memory_initialization_vector=
ff,
ab,
f0,
11,
11,
00,
01,
aa,
bb,
cc,
dd,
ef,
ee,
ff,
00,
ff;
```


Example: memory with initial value

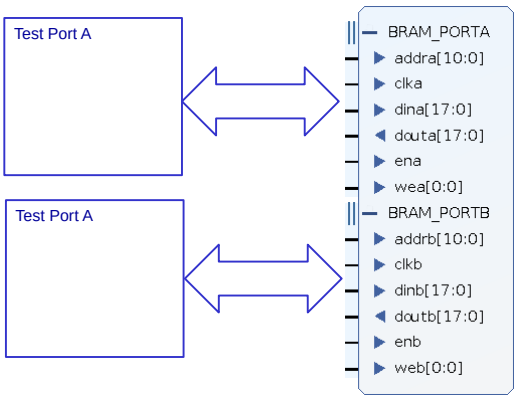
- Run an example code as follows




```
[user@host] cd $(PROJECT)/codes/bram/ex_bram_coe/sim/xsim
[user@host] make
[user@host] gtkwave wave.vcd &
```

Projects

- Make a BRAM of asynchronous 'True Dual Port RAM'
  - Make sure 'clka' and 'clkb' are not the same



References

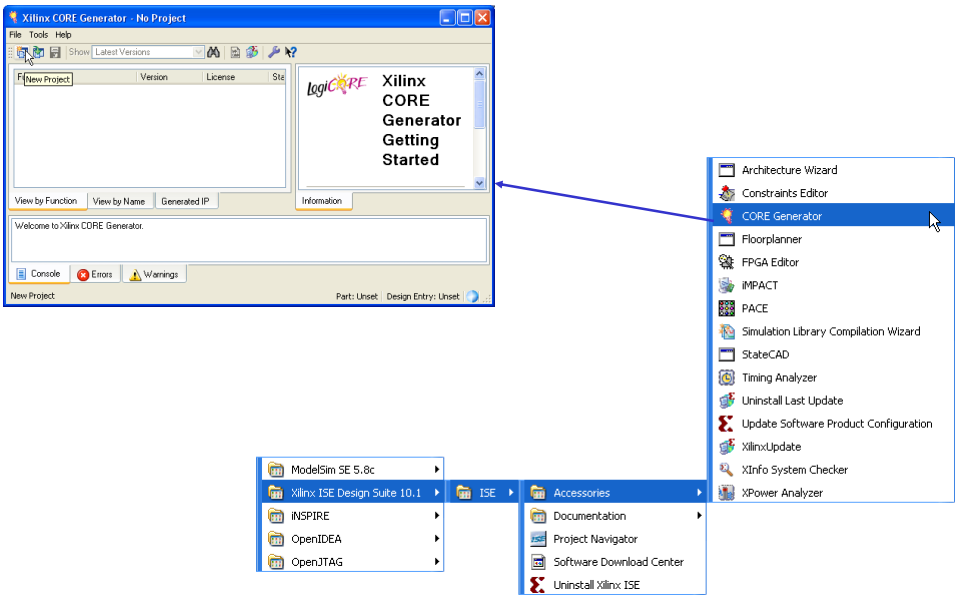
- Using Block RAM in Spartan-3 Generation FPGAs, XAPP463, Xilinx 2005.
- Block Memory Generator LogiCORE IP Product Guide Vivado Design Suite PG058

For the case of ISE

Create BRAM

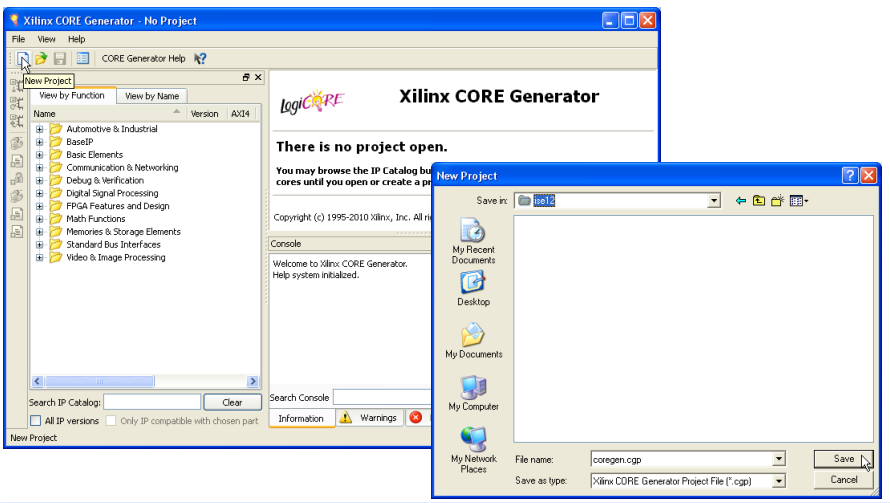
- Use Xilinx CORE Generator

Invoke Xilinx ISE CORE Generator

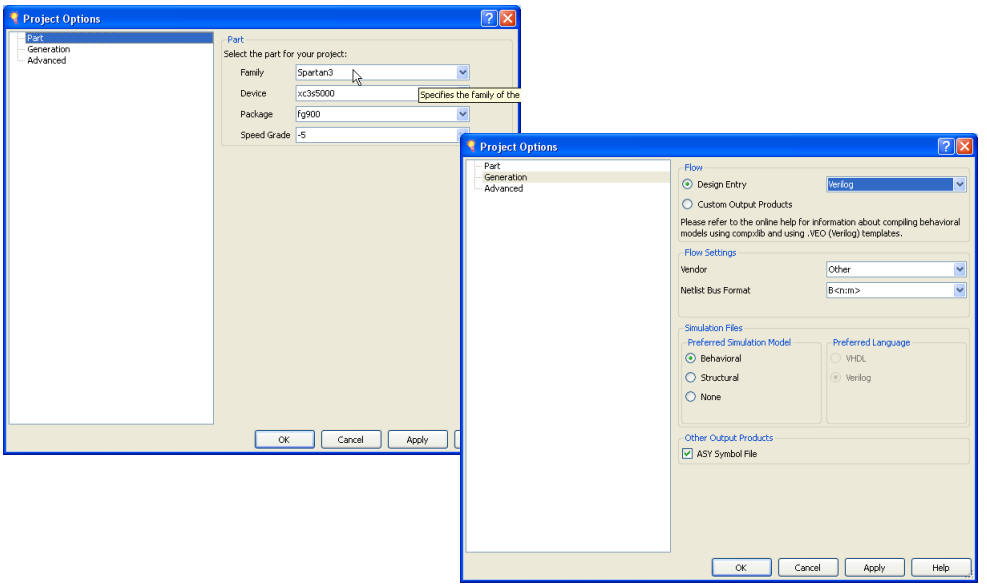


Create BRAM: CORE Generator

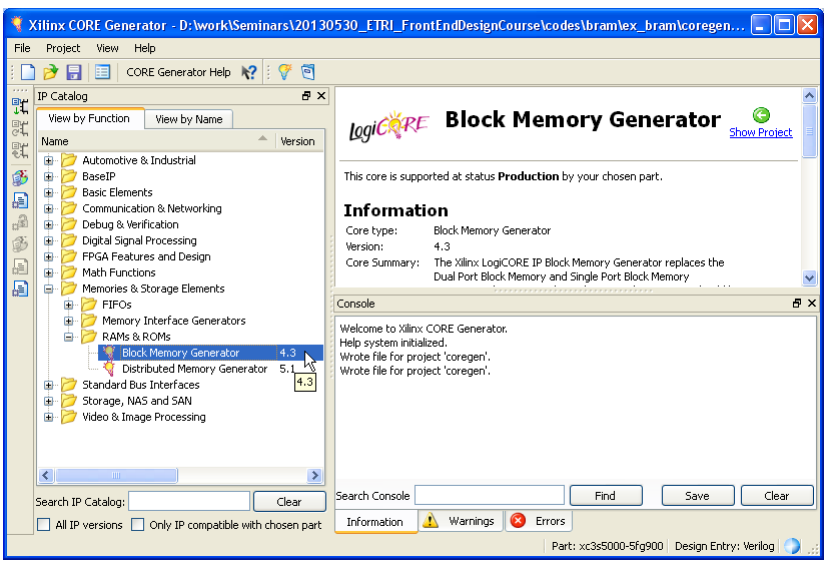
- Create a new project at
  - $(PROJECT)/codes/bram/ex_bram/coregen/ise12



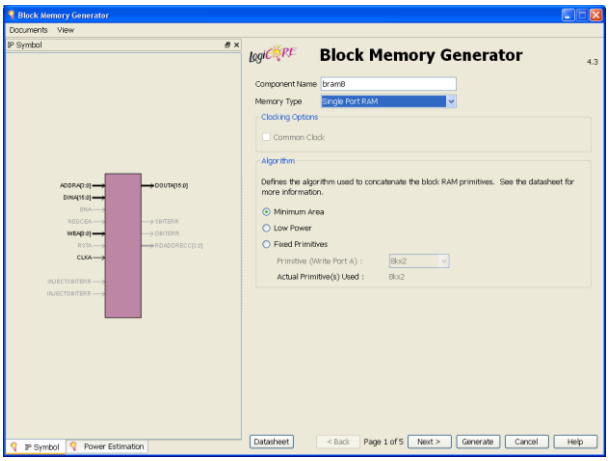
Create BRAM: CORE Generator



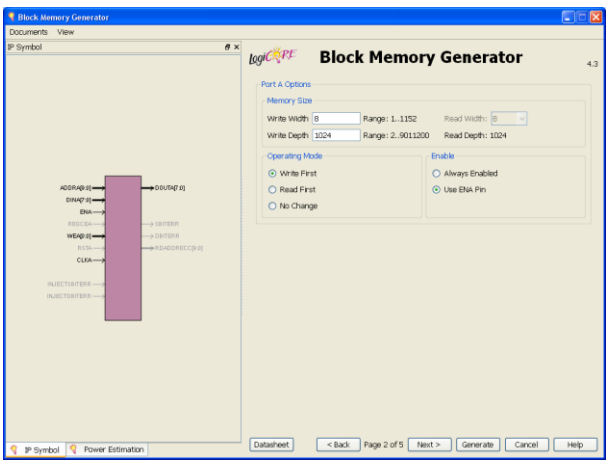
Create BRAM: CORE Generator



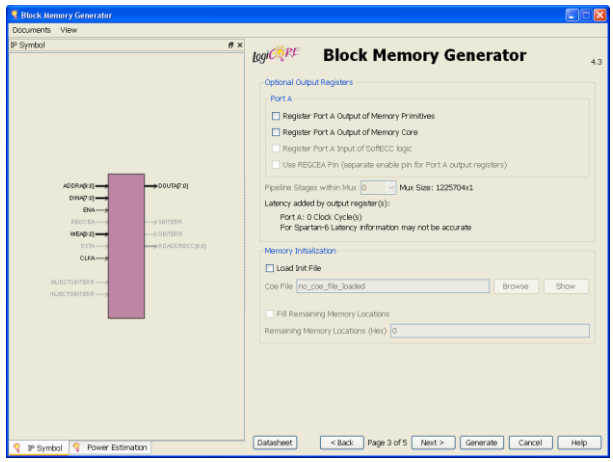
Create BRAM: CORE Generator



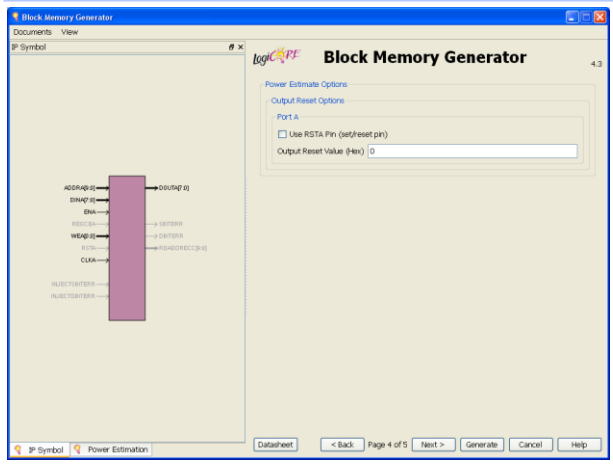
Create BRAM: CORE Generator



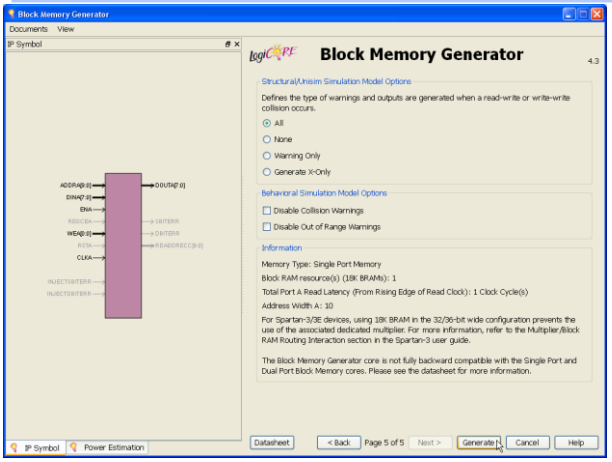
Create BRAM: CORE Generator



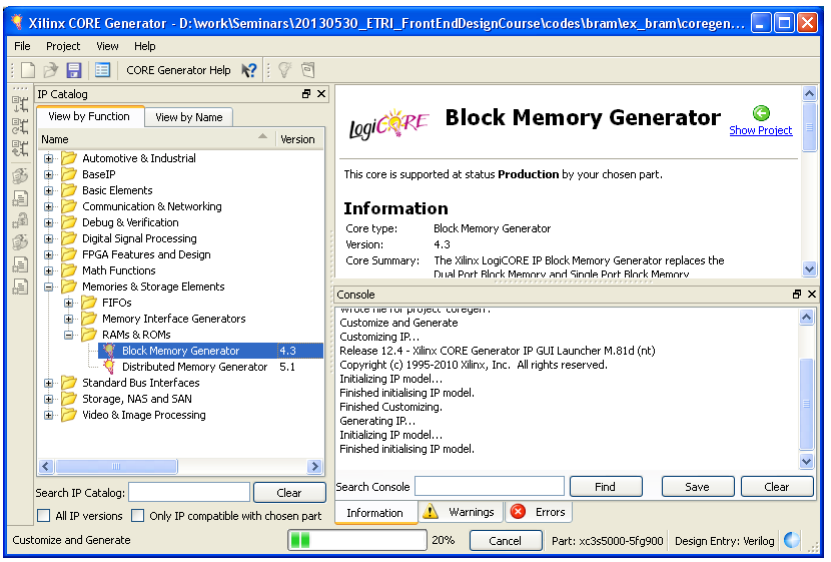
Create BRAM: CORE Generator



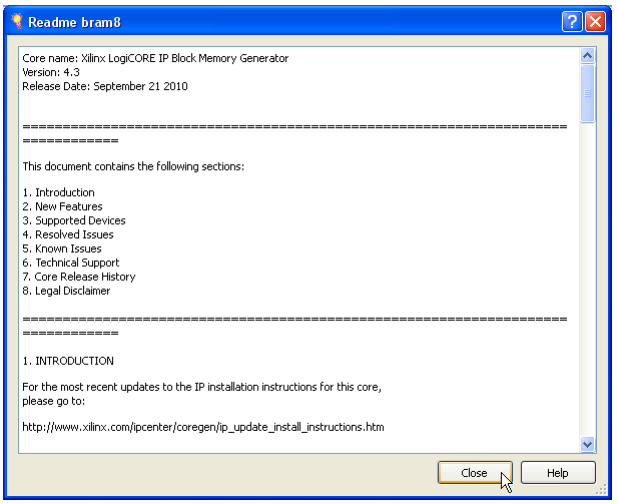
Create BRAM: CORE Generator



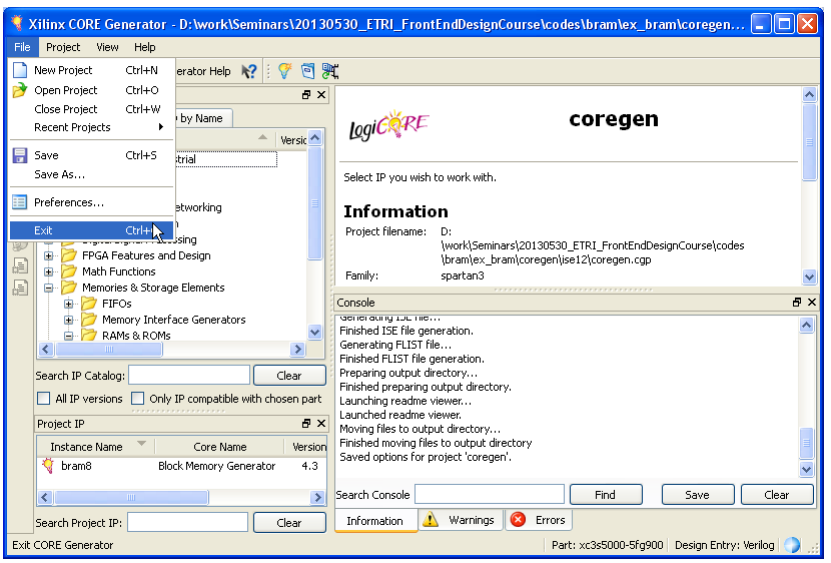
Create BRAM: CORE Generator



Create BRAM: CORE Generator



Create BRAM: CORE Generator



Create BRAM: CORE Generator

Name	Size	Type
_xmsgs		File Folder
tmp		File Folder
xlnx_auto_0_xdb		File Folder
blk_mem_gen_ds512.pdf	3,632 KB	Adobe Acrobat 문서
blk_mem_gen_readme.txt	7 KB	Text Document
bram8.asy	1 KB	ASY File
bram8.gise	2 KB	GISE File
bram8.ngc	14 KB	NGC File
bram8.v	7 KB	.v
bram8.veo	5 KB	VEO File
bram8.xco	3 KB	XCO File
bram8.xise	37 KB	Xilinx ISE Project
bram8_flist.txt	1 KB	Text Document
bram8_xmdf.tcl	3 KB	ActiveTcl Script
coregen.cgc	18 KB	CGC File
coregen.cgp	1 KB	CGP File

```
`timescale 1ns/1ps

module bram8(clka, ena, wea, addra, dina, douta);
input clka;
input ena;
input [0 : 0] wea;
input [9 : 0] addra;
input [7 : 0] dina;
output [7 : 0] douta;

// synthesis translate_off

    BLK_MEM_GEN_V4_3 #(
    ... ...
    .C_XDEVICEFAMILY("spartan3"))
     inst (.CLKA(clka),
           .ENA(ena),
           .WEA(wea),
           .ADDRA(addra),
           .DINA(dina),
           .DOUTA(douta),
           ... ... );

// synthesis translate_on

endmodule
```