Verilog-HDL講習会
追加資料(9)
実用編(1)
～Xilinx CORE Generatorの使い方～

7, August, 2012

ブロックRAM (BRAM)を使おう

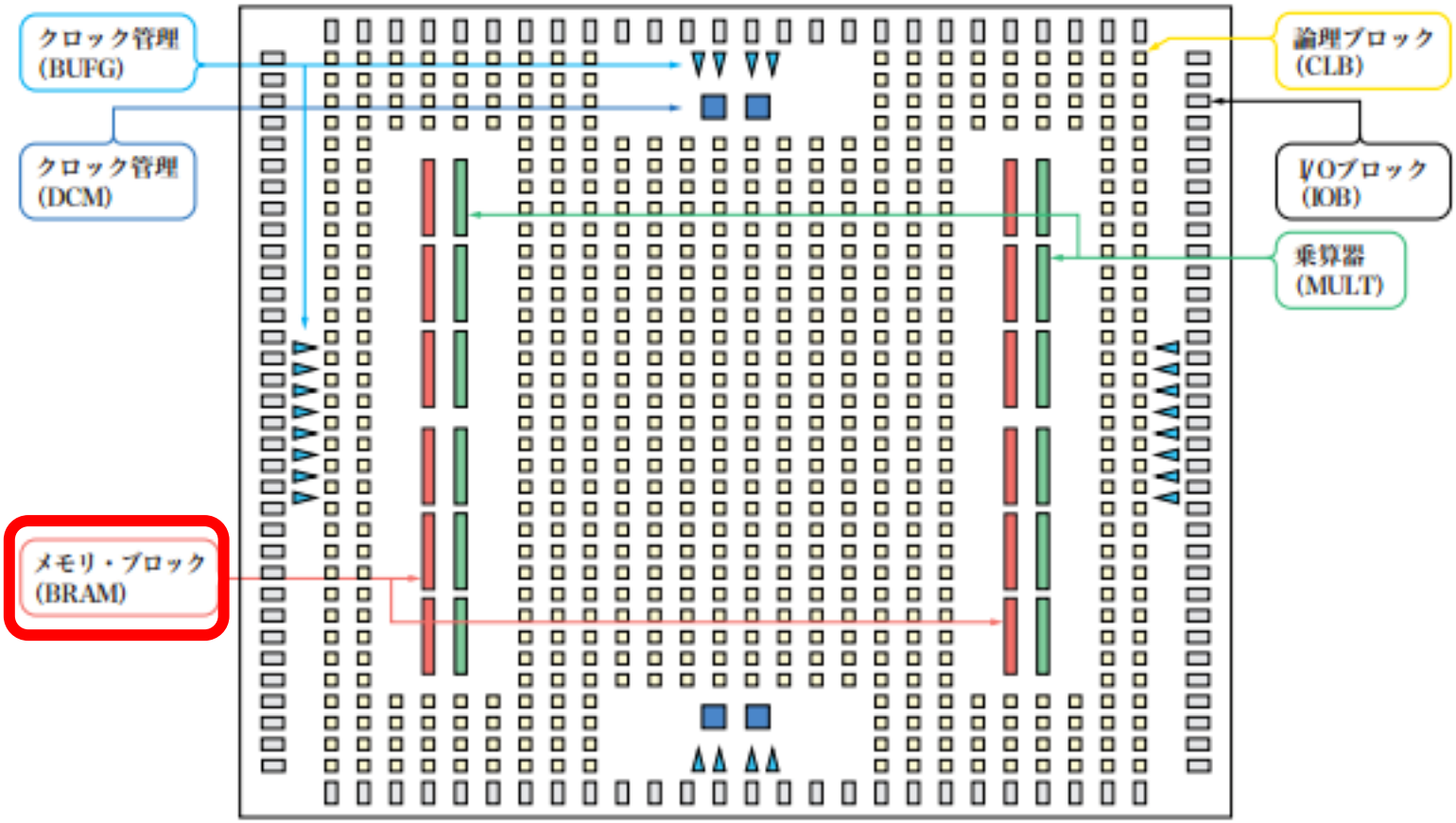


図2 XC3S250Eの構造

論理ブロック(CLB：configurable logic block)，I/Oブロック(IOB)，クロック管理ブロック(DCM，BUFG)，メモリ・ブロック(BRAM)，乗算器ブロック(MULT)，コンフィグレーション回路から構成されている．

BRAMとは

- FPGAに内蔵されているメモリ
- デュアルポート可能
- BRAM1個で18Kビット
- メモリ量はあまりない（せいぜい数百Kビット）

Table 2: **Block RAM Available in Spartan-3E Devices**

Device	RAM Columns	RAM Blocks per Column	Total RAM Blocks	Total RAM Bits	Total RAM Kbits
XC3S100E	1	4	4	73728	72
XC3S250E	2	6	12	221184	216
XC3S500E	2	10	20	368640	360
XC3S1200E	2	14	28	516096	504
XC3S1600E	2	18	36	663552	648

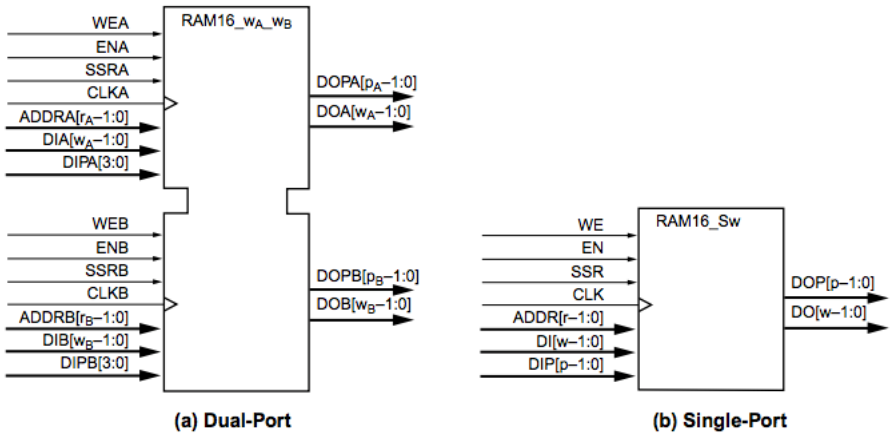
BRAMの仕様

Table 3: **SelectRAM 18K Block Memory Features and Applications**

Total RAM bits, including parity	18,432 (16K data + 2K parity)
Memory Organizations	16Kx1 8Kx2 4Kx4 2Kx8 (no parity) 2Kx9 (x8 + parity) 1Kx16 (no parity) 1Kx18 (x16 + 2 parity) 512x32 (no parity) 512x36 (x32 + 4 parity) 256x72 (single-port only)
Parity	Available and optional only for organizations greater than byte-wide. Parity bits optionally available as extra data bits.
Performance	200 MHz (estimated)
Timing Interface	Simple synchronous interface. Similar to reading and writing from a register with a setup time for write operations and clock-to-output delay for read operations.
Single-Port	Yes
True Dual-Port	Yes
ROM, Initial RAM Contents	Yes
Mixed Data Port Widths	Yes
Power-Up Condition	User-defined data, defaults to zero
Potential Applications	Local data storage, FIFOs, elastic stores, register files, buffers, stacks, circular buffers, shift registers, delay lines, waveform storage and generation, direct digital synthesis, CAMs, associative memories, function tables, function generators, wide logic functions, code converters, encoders, decoders, counters, state machines, microsequencers, program storage for embedded processor(s)

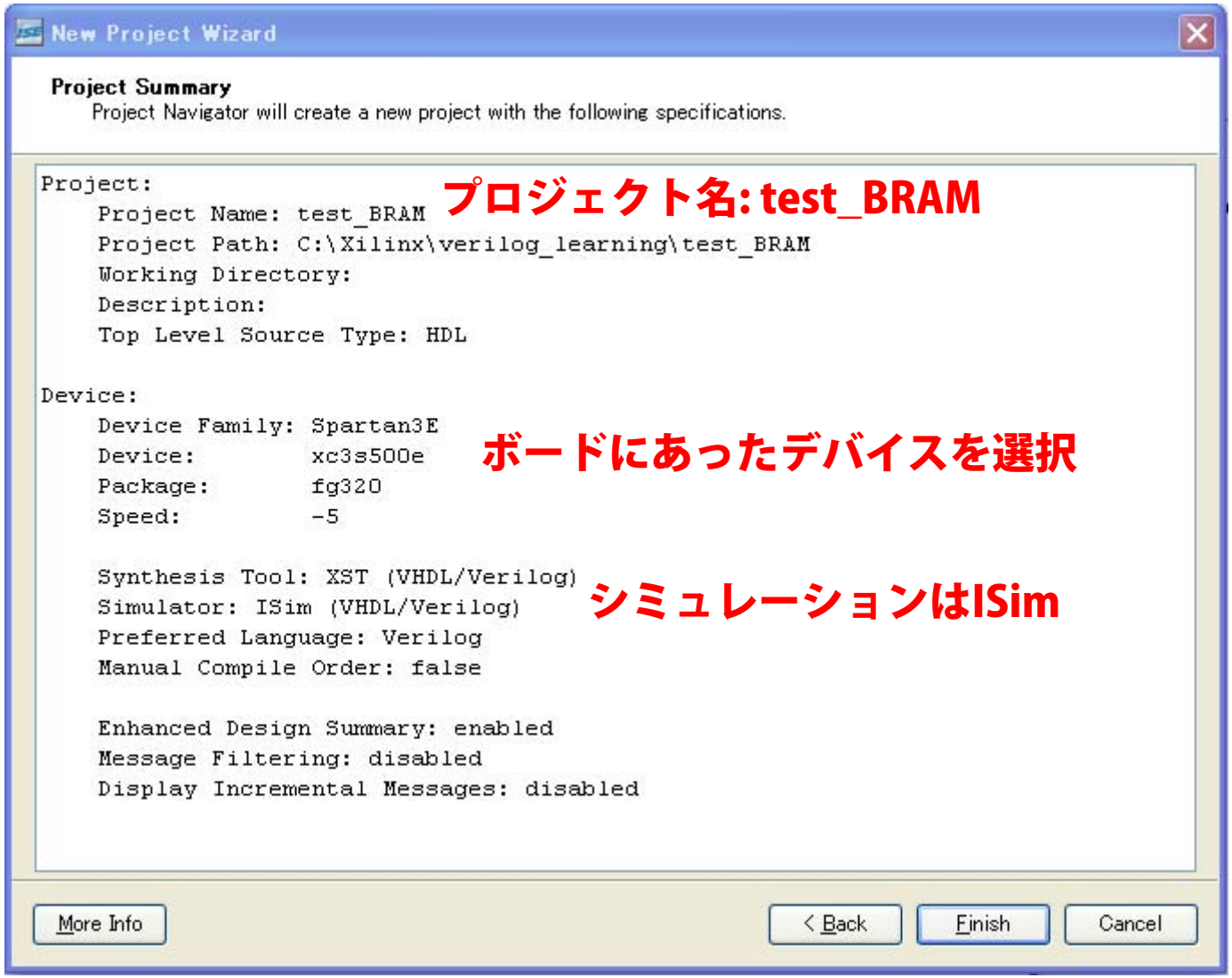
容量は18Kビット

ワード数とワード長を構成可能
(足りないときは複数のBRAMを使う)

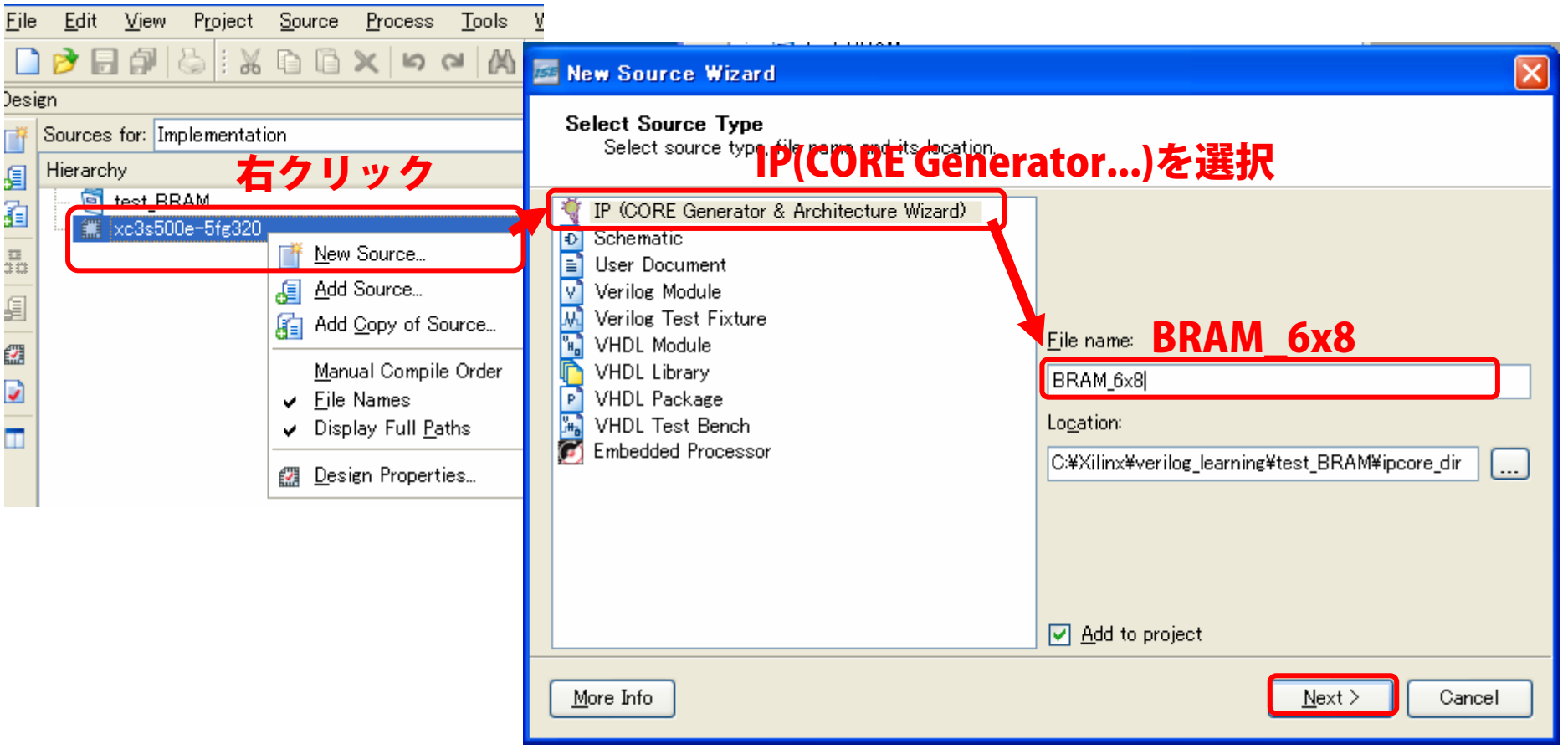


デュアルポートも構成できる

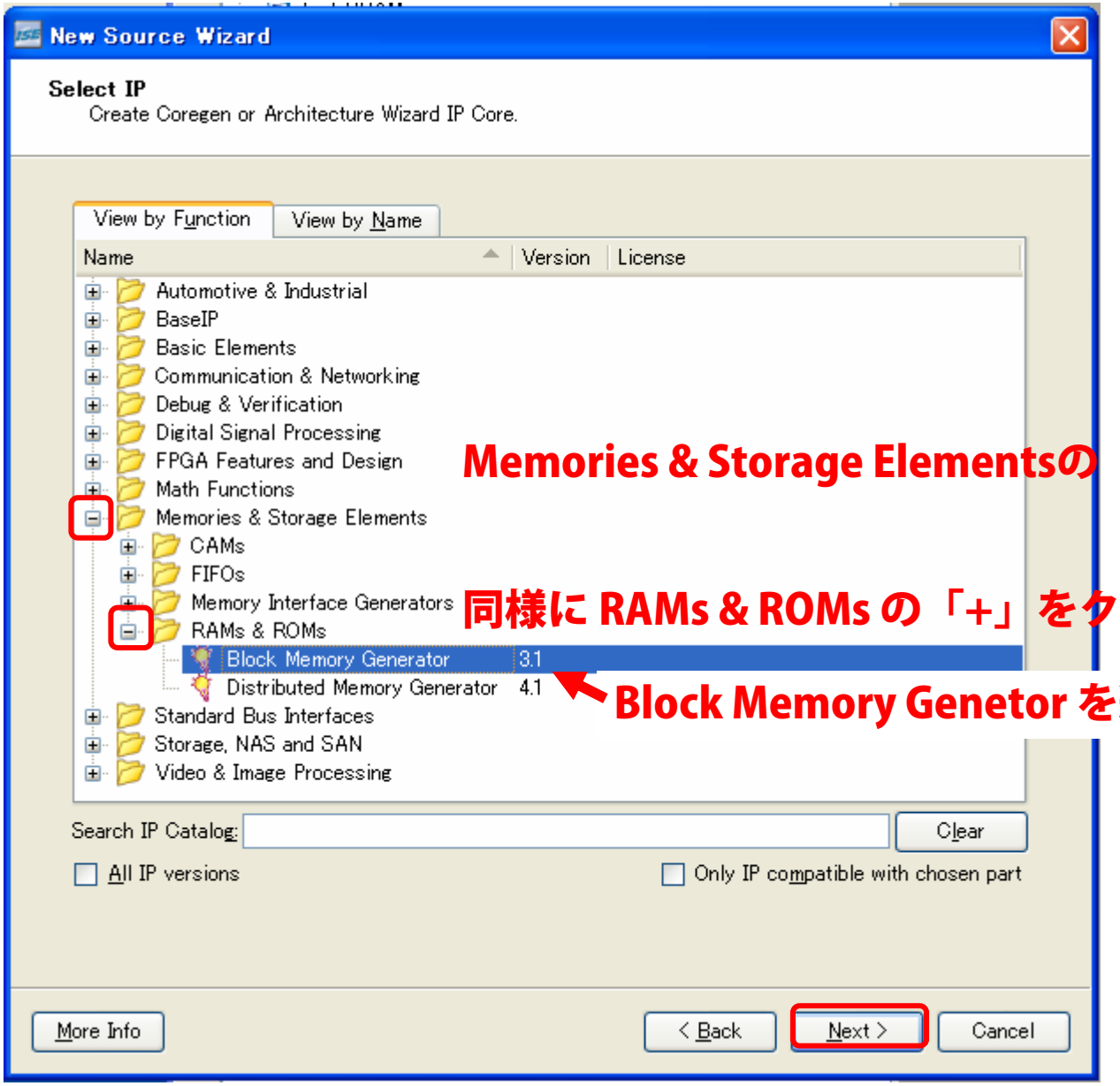
プロジェクトの作成


New Project Wizard
Project Summary
Project Navigator will create a new project with the following specifications.
Project:
Project Name: test_BRAM
プロジェクト名: test_BRAM
Project Path: C:\Xilinx\verilog_learning\test_BRAM
Working Directory:
Description:
Top Level Source Type: HDL
Device:
Device Family: Spartan3E
Device: xc3s500e
ボードにあったデバイスを選択
Package: fg320
Speed: -5
Synthesis Tool: XST (VHDL/Verilog)
Simulator: ISim (VHDL/Verilog)
シミュレーションはISim
Preferred Language: Verilog
Manual Compile Order: false
Enhanced Design Summary: enabled
Message Filtering: disabled
Display Incremental Messages: disabled
More Info
< Back
Finish
Cancel

CoreGeneratorの起動



生成するIPを選択

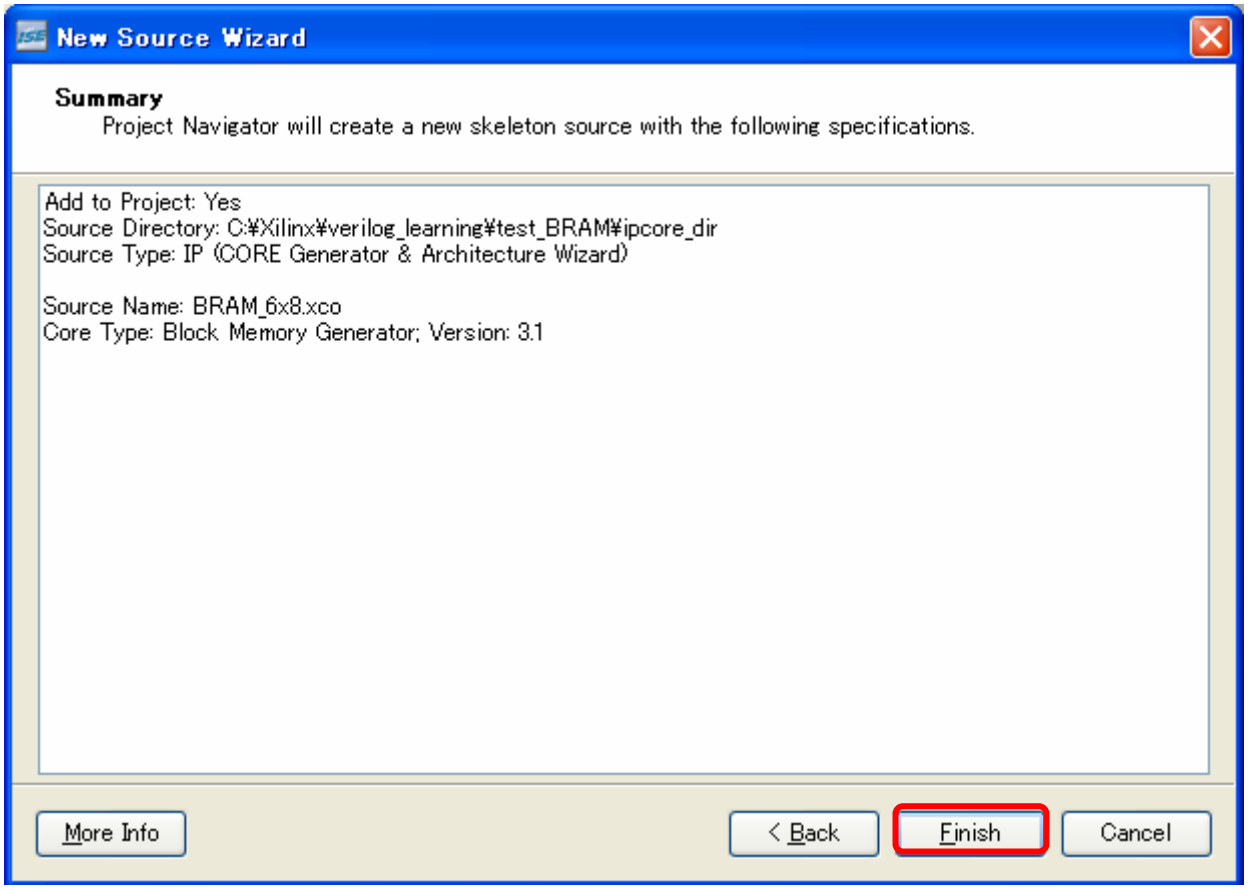


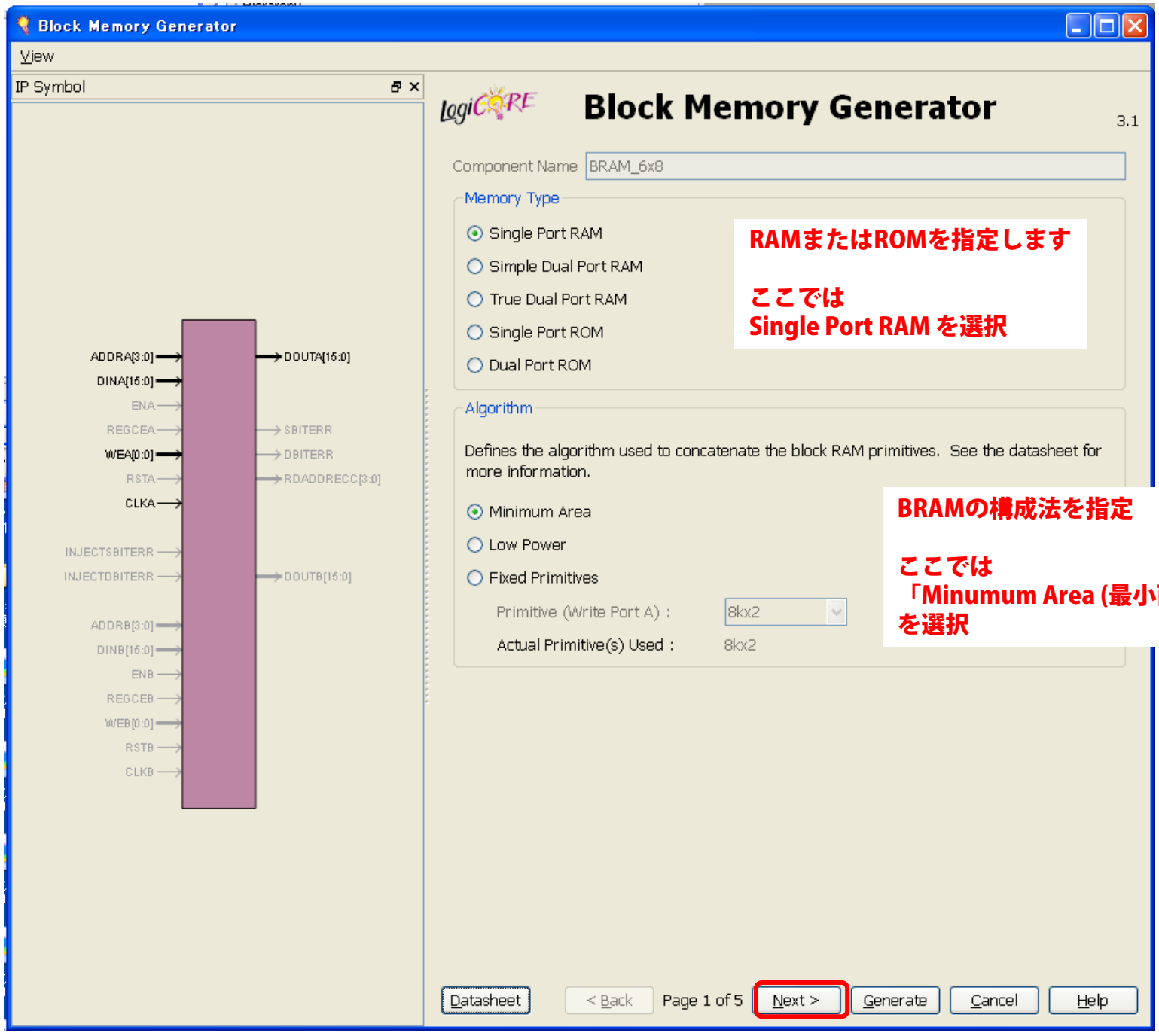
Memories & Storage Elementsの「+」をクリック

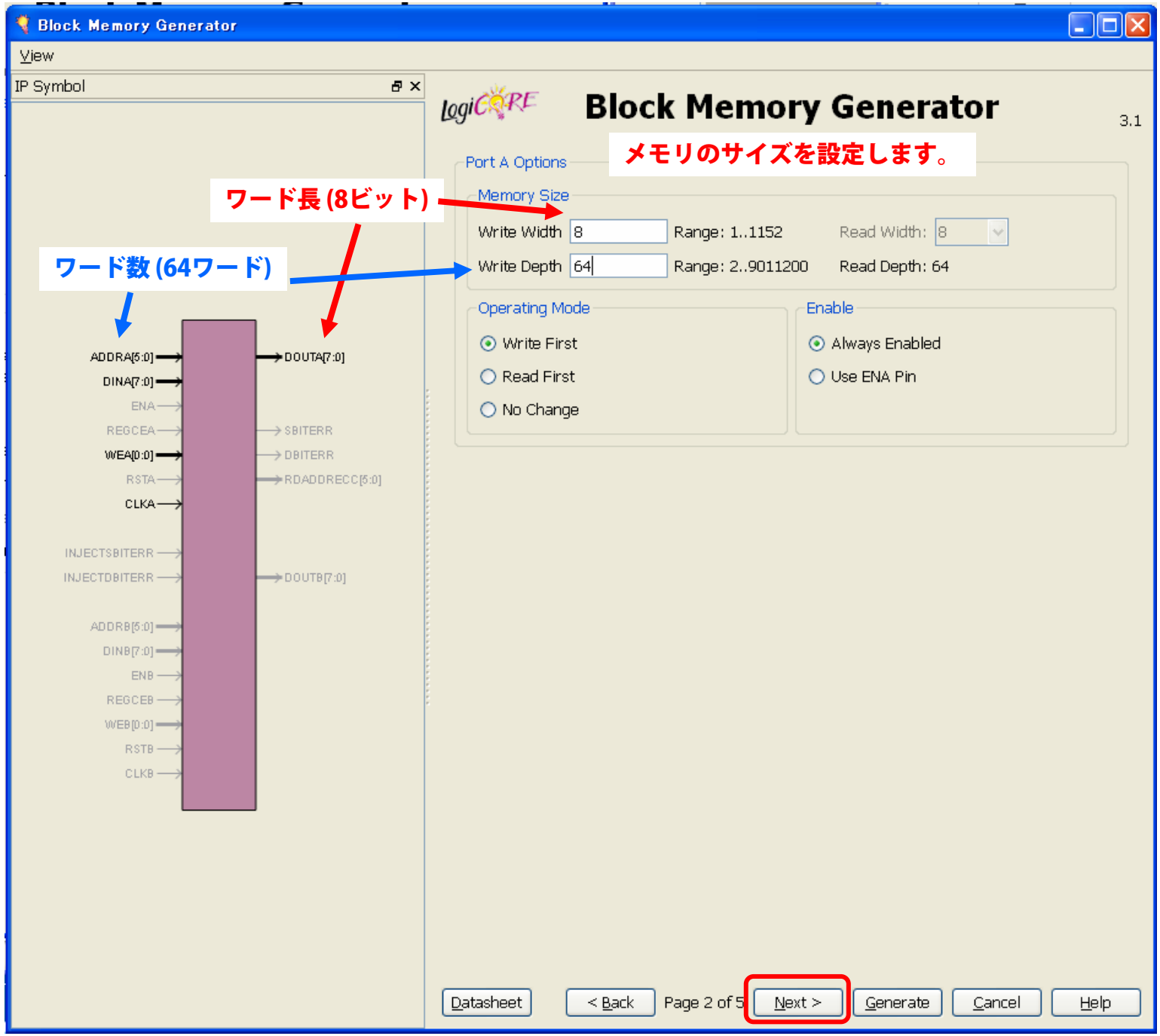
同様に RAMs & ROMs の「+」をクリック

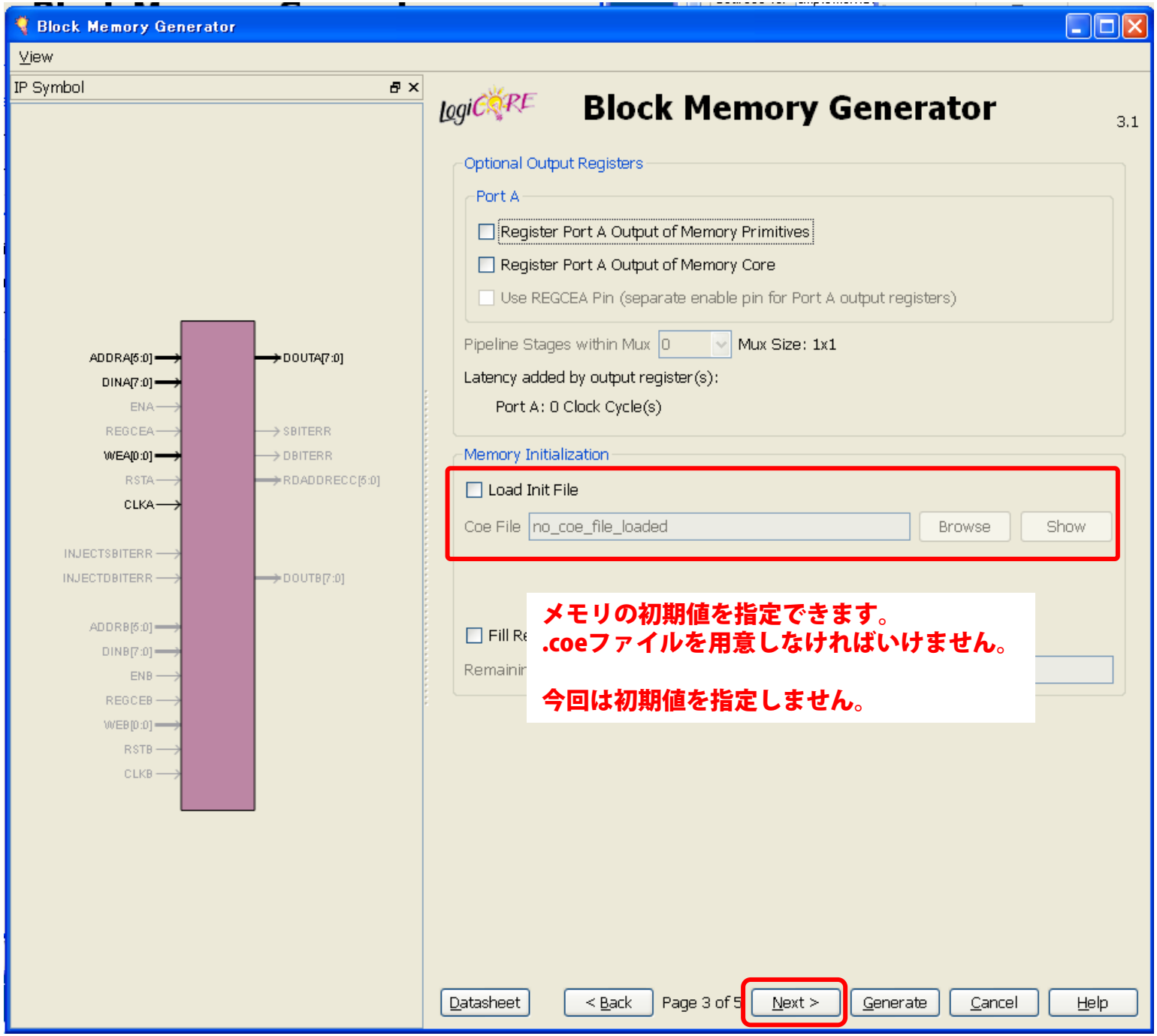
Block Memory Genetor を選択

生成するIPコアの確認

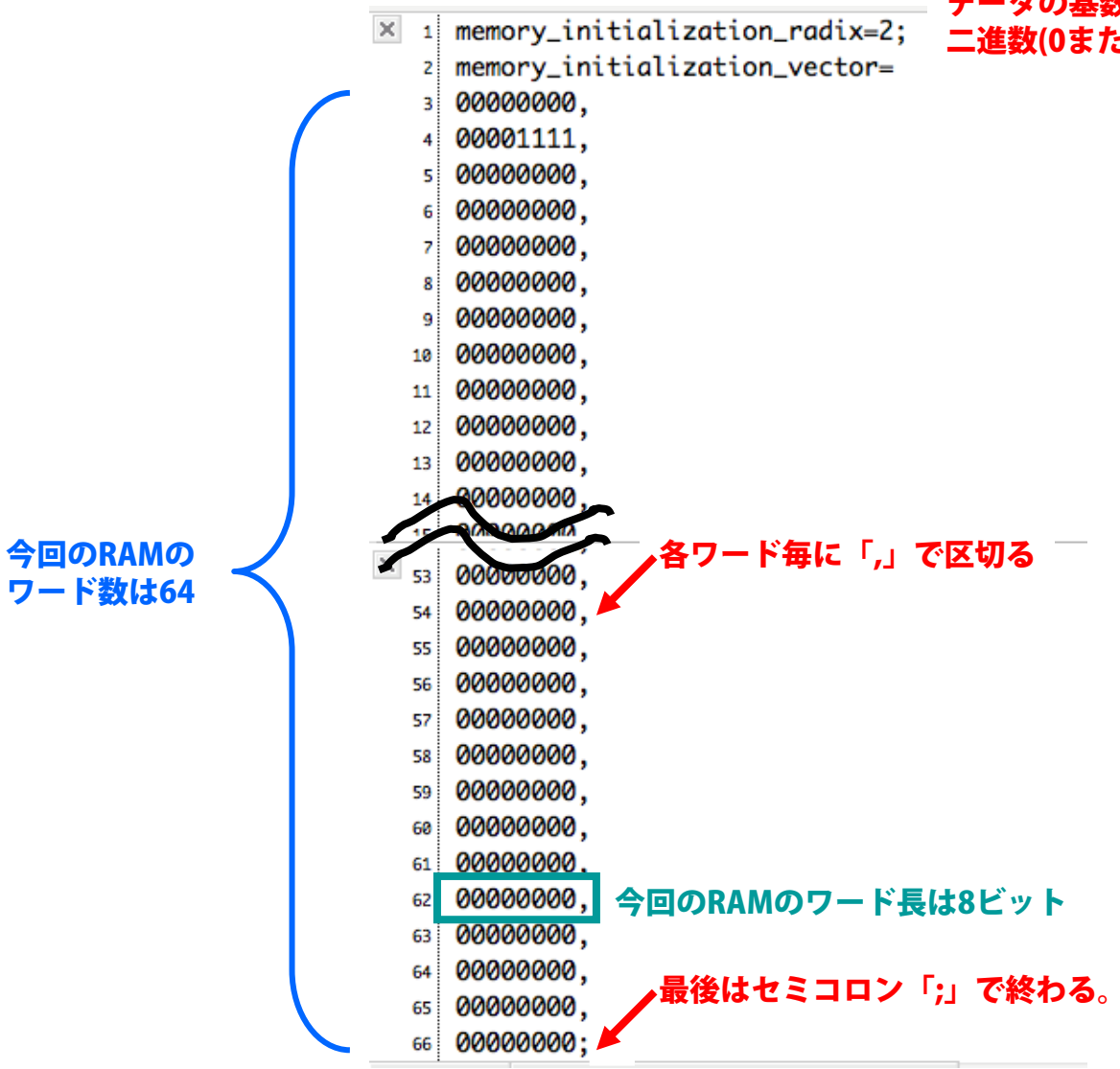


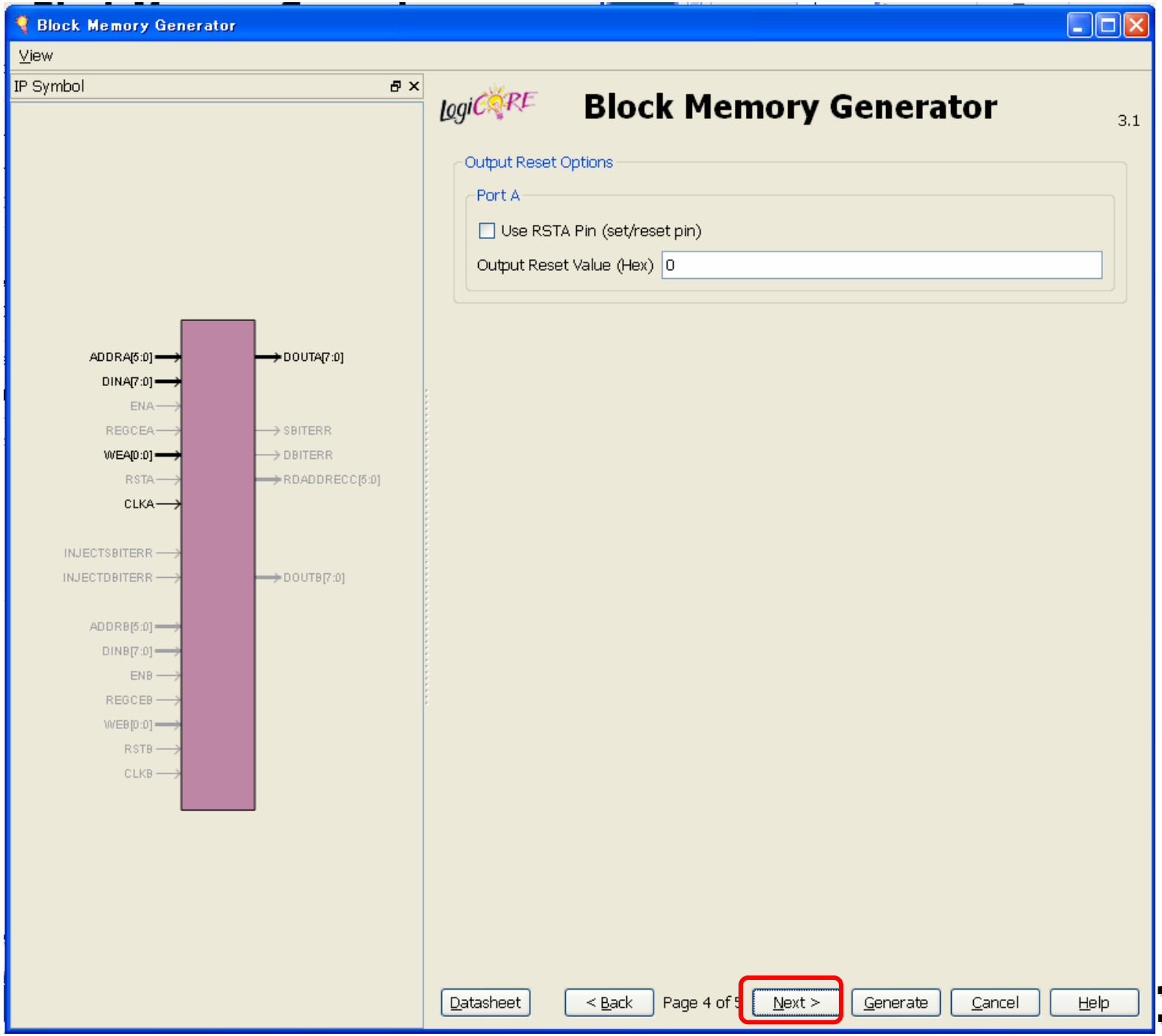

Block Memory Generator
View
IP Symbol
ADDRA[3:0]
DINA[15:0]
ENA
REGCEA
WEA[0:0]
RSTA
CLKA
INJECTSBITERR
INJECTDBITERR
ADDRB[3:0]
DINB[15:0]
ENB
REGCEB
WEB[0:0]
RSTB
CLKB
DOUTA[15:0]
SBITERR
DBITERR
RDADDRECC[3:0]
DOUTB[15:0]
Block Memory Generator
3.1
Component Name
BRAM_6x8
Memory Type
Single Port RAM
Simple Dual Port RAM
True Dual Port RAM
Single Port ROM
Dual Port ROM
RAMまたはROMを指定します
ここでは
Single Port RAM を選択
Algorithm
Defines the algorithm used to concatenate the block RAM primitives. See the datasheet for more information.
Minimum Area
Low Power
Fixed Primitives
Primitive (Write Port A) :
8kx2
Actual Primitive(s) Used :
8kx2
BRAMの構成法を指定
ここでは
「Minumum Area (最小面積)」
を選択
Datasheet
< Back
Page 1 of 5
Next >
Generate
Cancel
Help

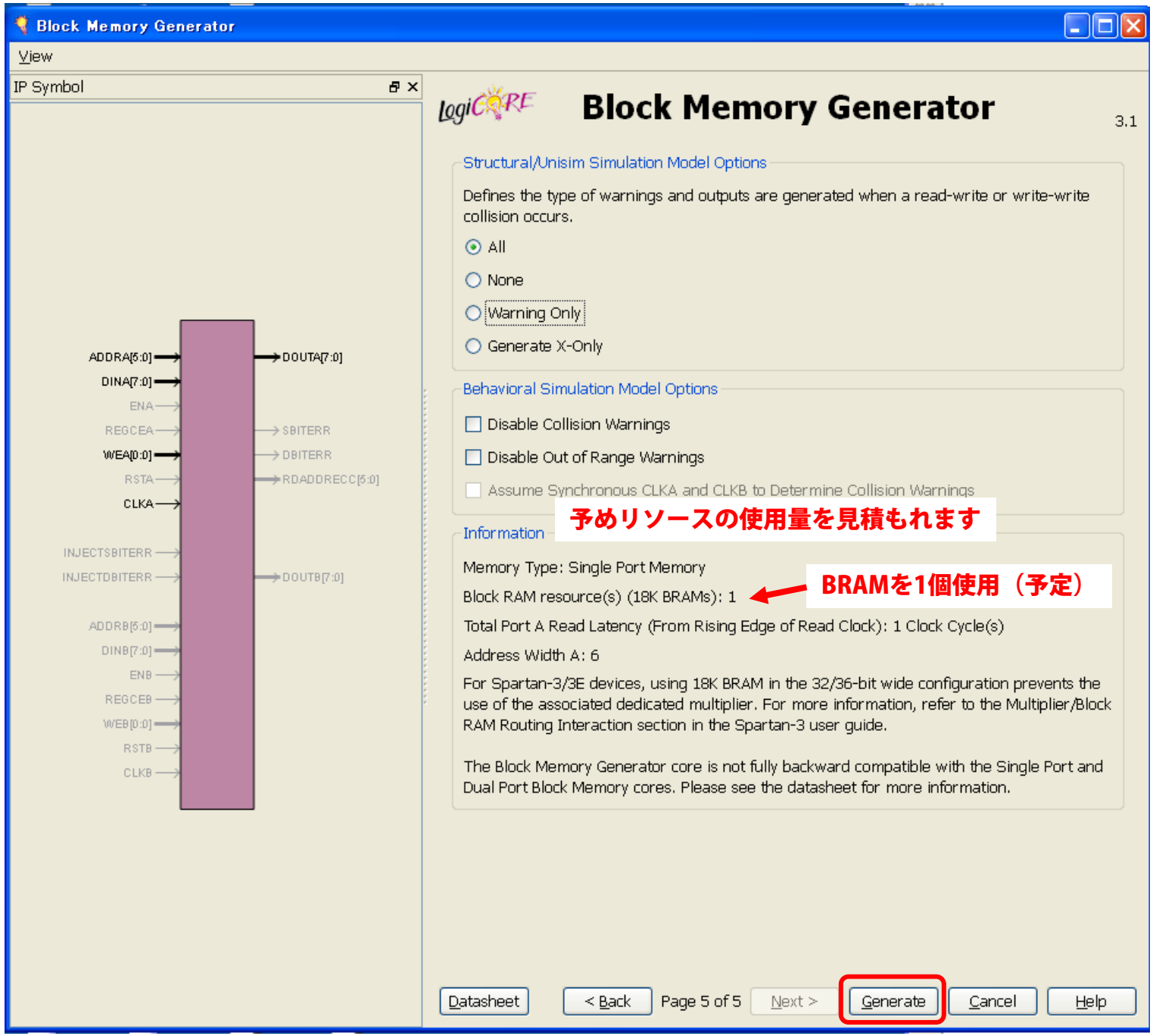

Block Memory Generator
View
IP Symbol
LogiCORE
Block Memory Generator
3.1
メモリのサイズを設定します。
ワード長 (8ビット)
ワード数 (64ワード)
Port A Options
Memory Size
Write Width 8 Range: 1..1152 Read Width: 8
Write Depth 64 Range: 2..9011200 Read Depth: 64
Operating Mode
Write First
Read First
No Change
Enable
Always Enabled
Use ENA Pin
ADDRA[5:0]
DINA[7:0]
ENA
REGCEA
WEA[0:0]
RSTA
CLKA
INJECTSBITERR
INJECTDBITERR
ADDRB[5:0]
DINB[7:0]
ENB
REGCEB
WEB[0:0]
RSTB
CLKB
DOUTA[7:0]
SBITERR
DBITERR
RDADDRECC[5:0]
DOUTB[7:0]
Datasheet
< Back
Page 2 of 5
Next >
Generate
Cancel
Help


Block Memory Generator
View
IP Symbol
ADDRA[5:0]
DINA[7:0]
ENA
REGCEA
WEA[0:0]
RSTA
CLKA
INJECTSBITERR
INJECTDBITERR
ADDRB[5:0]
DINB[7:0]
ENB
REGCEB
WEB[0:0]
RSTB
CLKB
DOUTA[7:0]
SBITERR
DBITERR
RDADDRECC[5:0]
DOUTB[7:0]
Block Memory Generator
3.1
Optional Output Registers
Port A
Register Port A Output of Memory Primitives
Register Port A Output of Memory Core
Use REGCEA Pin (separate enable pin for Port A output registers)
Pipeline Stages within Mux 0 Mux Size: 1x1
Latency added by output register(s):
Port A: 0 Clock Cycle(s)
Memory Initialization
Load Init File
Coe File no_coe_file_loaded Browse Show
メモリの初期値を指定できます。
.coeファイルを用意しなければいけません。
今回は初期値を指定しません。
Datasheet
< Back
Page 3 of 5
Next >
Generate
Cancel
Help

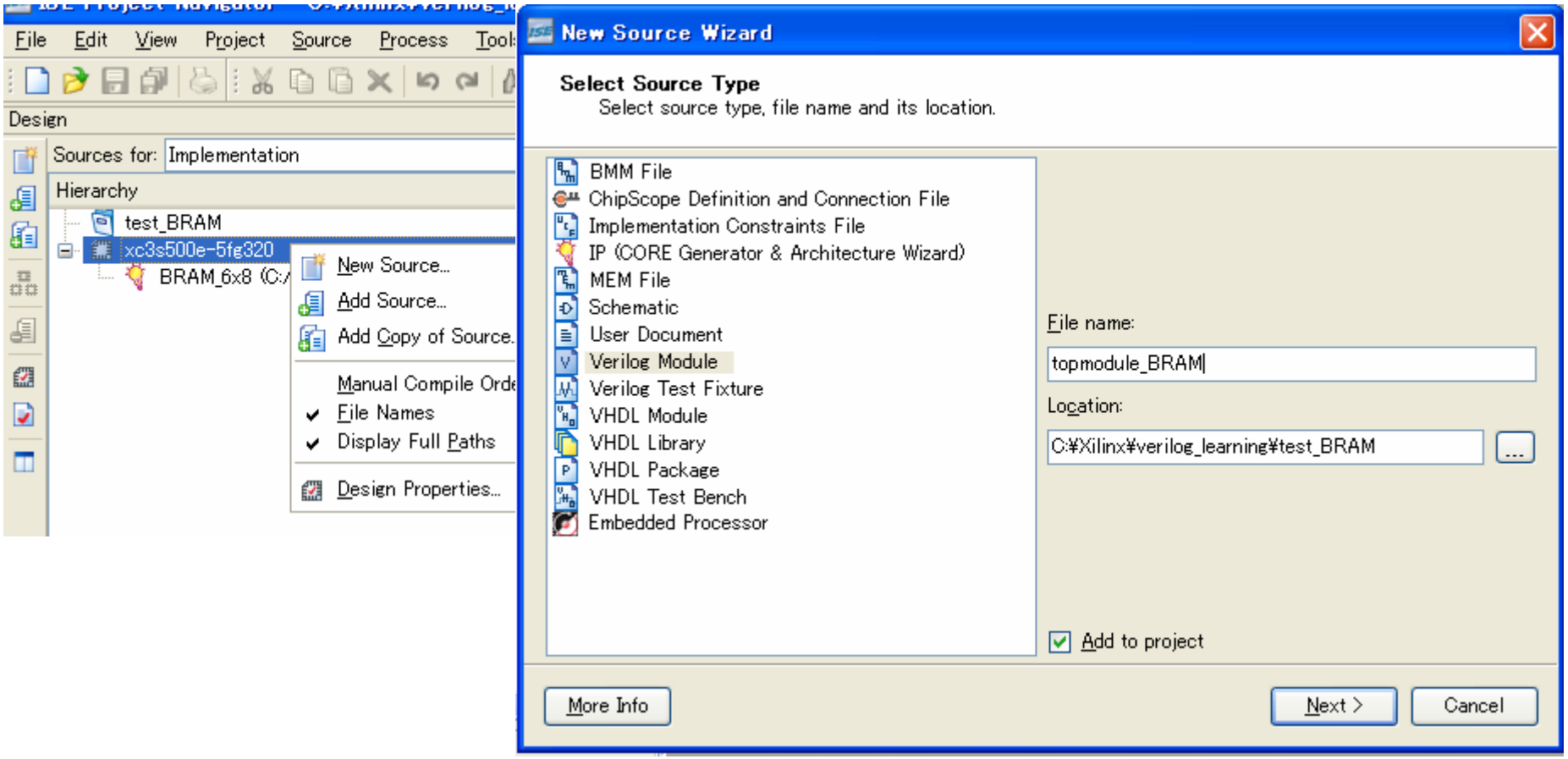
.coeファイルのフォーマット



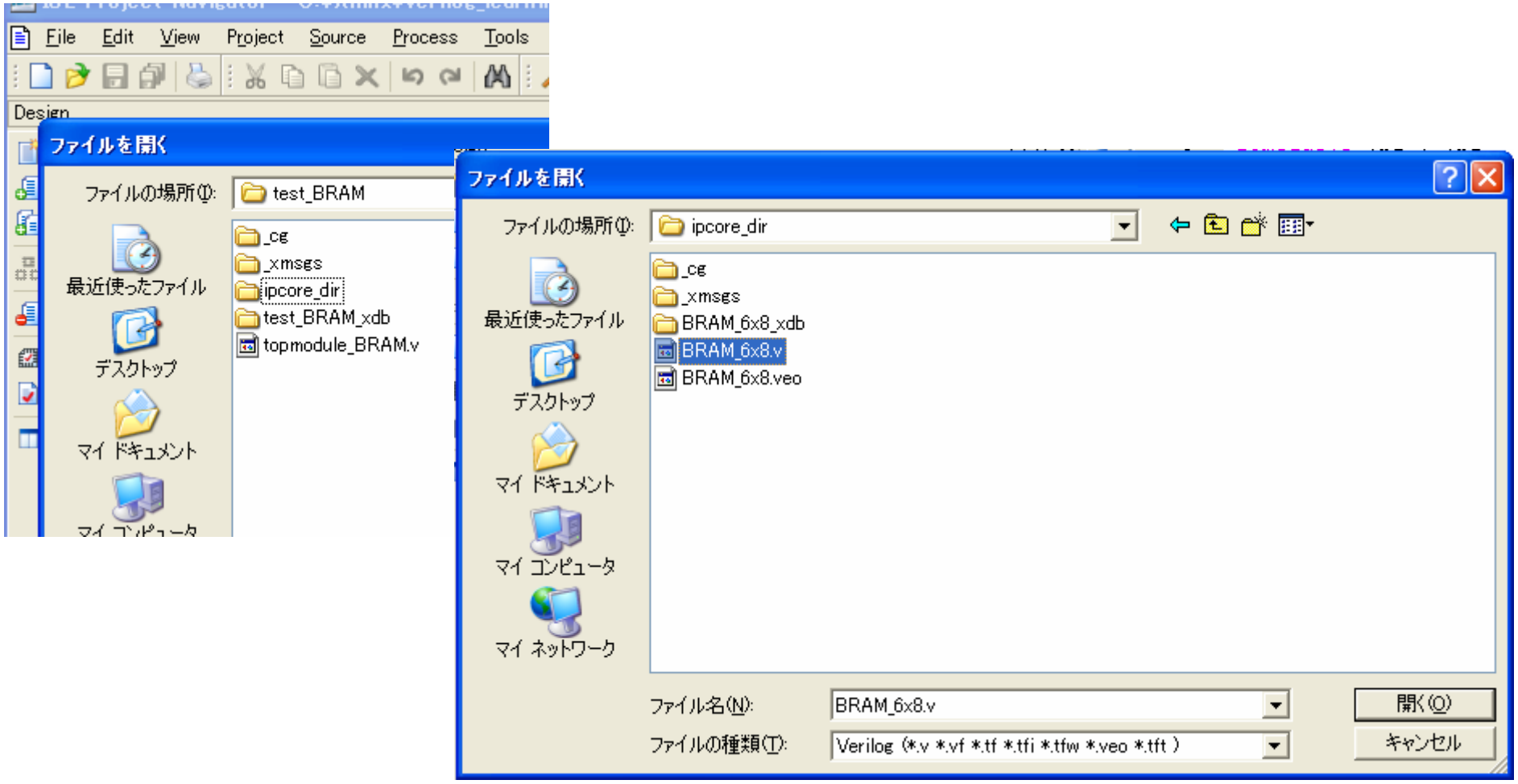

Block Memory Generator
View
IP Symbol
ADDRA[5:0]
DINA[7:0]
ENA
REGCEA
WEA[0:0]
RSTA
CLKA
INJECTSBITERR
INJECTDBITERR
ADDRB[5:0]
DINB[7:0]
ENB
REGCEB
WEB[0:0]
RSTB
CLKB
DOUTA[7:0]
SBITERR
DBITERR
RDADDRECC[5:0]
DOUTB[7:0]
LogiCORE
Block Memory Generator
3.1
Output Reset Options
Port A
Use RSTA Pin (set/reset pin)
Output Reset Value (Hex)
0
Datasheet
< Back
Page 4 of 5
Next >
Generate
Cancel
Help


Block Memory Generator
View
IP Symbol
ADDRA[5:0]
DINA[7:0]
ENA
REGCEA
WEA[0:0]
RSTA
CLKA
INJECTSBITERR
INJECTDBITERR
ADDRB[5:0]
DINB[7:0]
ENB
REGCEB
WEB[0:0]
RSTB
CLKB
DOUTA[7:0]
SBITERR
DBITERR
RDADDRECC[5:0]
DOUTB[7:0]
LogiCORE
Block Memory Generator
3.1
Structural/Unisim Simulation Model Options
Defines the type of warnings and outputs are generated when a read-write or write-write collision occurs.
All
None
Warning Only
Generate X-Only
Behavioral Simulation Model Options
Disable Collision Warnings
Disable Out of Range Warnings
Assume Synchronous CLKA and CLKB to Determine Collision Warnings
予めリソースの使用量を見積もれます
Information
Memory Type: Single Port Memory
Block RAM resource(s) (18K BRAMs): 1
BRAMを1個使用（予定）
Total Port A Read Latency (From Rising Edge of Read Clock): 1 Clock Cycle(s)
Address Width A: 6
For Spartan-3/3E devices, using 18K BRAM in the 32/36-bit wide configuration prevents the use of the associated dedicated multiplier. For more information, refer to the Multiplier/Block RAM Routing Interaction section in the Spartan-3 user guide.
The Block Memory Generator core is not fully backward compatible with the Single Port and Dual Port Block Memory cores. Please see the datasheet for more information.
Datasheet
< Back
Page 5 of 5
Next >
Generate
Cancel
Help

トップモジュールの追加



IPコアのテンプレート (.veoファイル)を開く



.veoファイルの中身

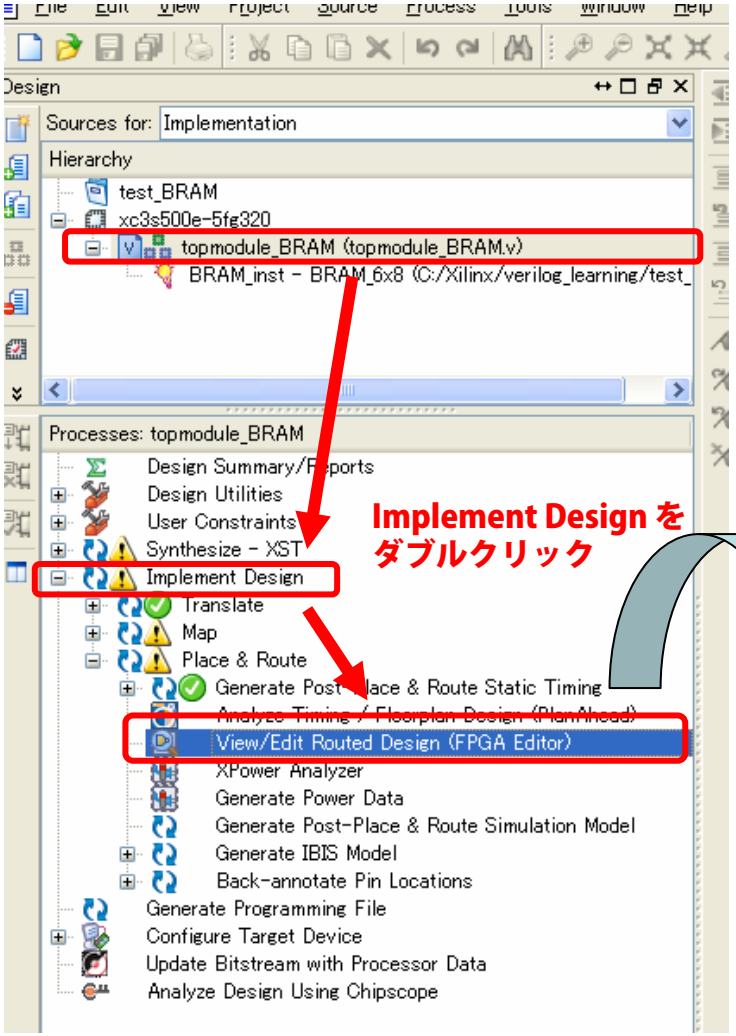
- コメントにバス幅が書かれている
- 「Your InstanceName」を自分のインスタンス名に書き直せばOK

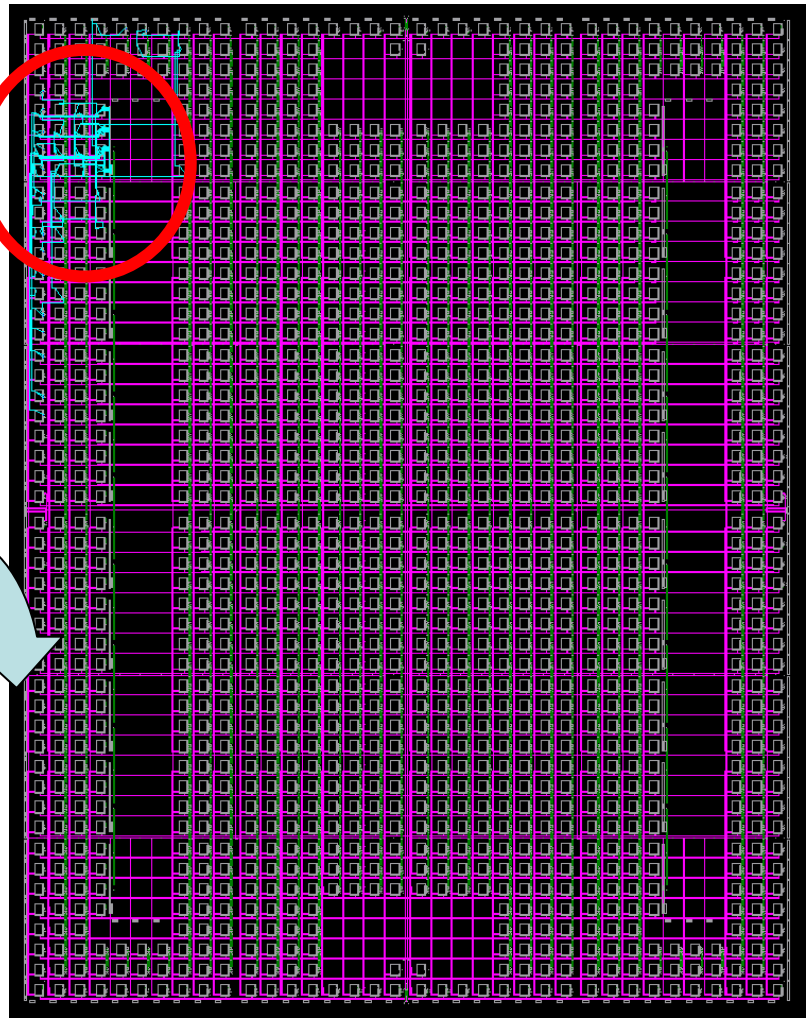
```
33  //------------ Begin Cut here for INSTANTIATION Template ---// INST_TAG
34  BRAM_6x8 YourInstanceName (
35     .clka(clka),
36     .wea(wea), // Bus [0 : 0]
37     .addra(addra), // Bus [5 : 0]
38     .dina(dina), // Bus [7 : 0]
39     .douta(douta)); // Bus [7 : 0]
40
41  // INST_TAG_END ------ End INSTANTIATION Template ---------
```

今回記述するVerilog-HDL

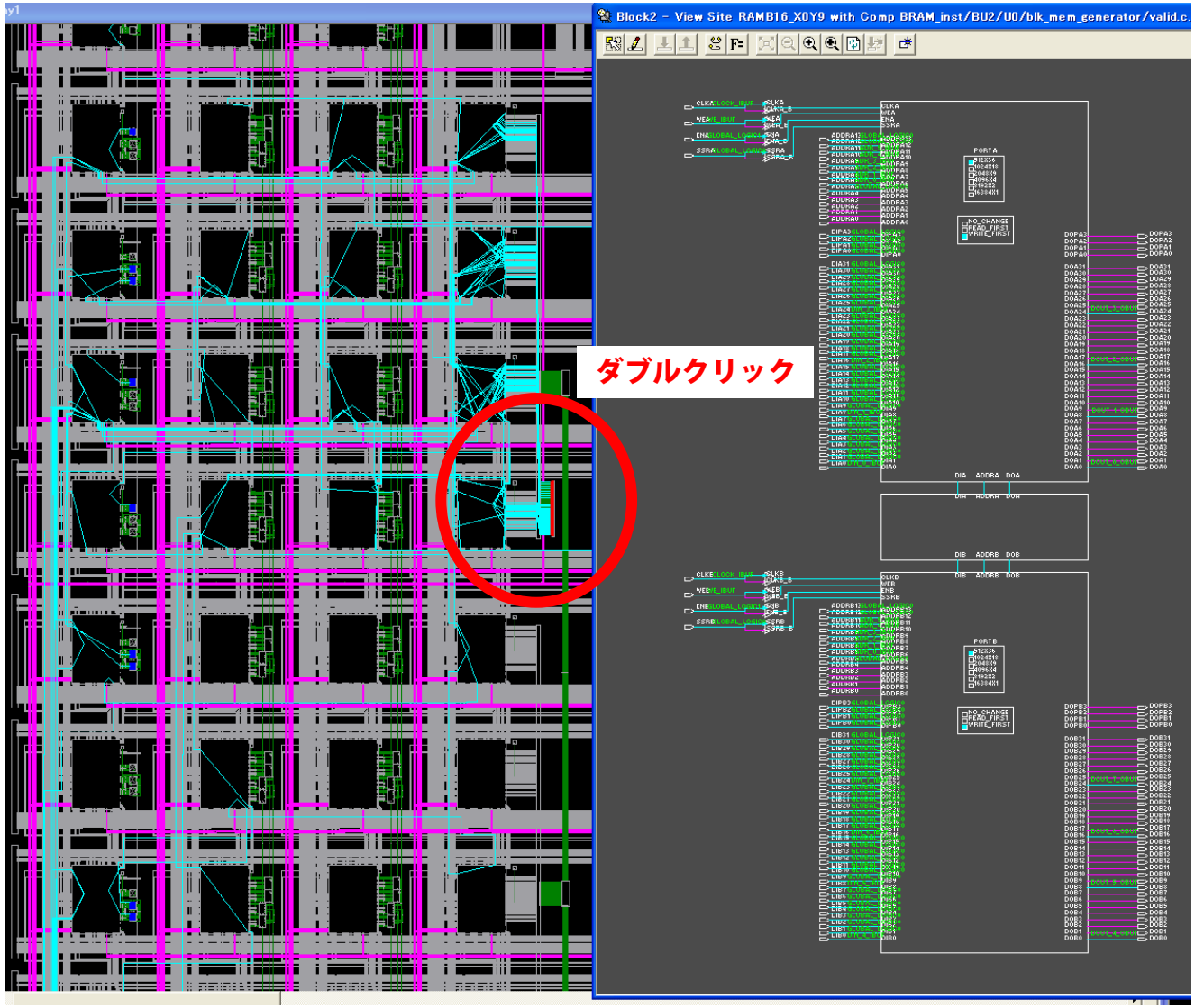
```
21  module topmodule_BRAM(
22      CLOCK, WE, ADR, DIN, DOUT
23       );
24
25  input CLOCK;
26  input WE;
27  input [5:0]ADR;
28  input [7:0]DIN;
29  output [7:0]DOUT;
30
31  BRAM_6x8 BRAM_inst (
32      .clka( CLOCK),
33      .wea( WE), // Bus [0 : 0]
34      .addra( ADR), // Bus [5 : 0]
35      .dina( DIN), // Bus [7 : 0]
36      .douta( DOUT)); // Bus [7 : 0]
37
38  endmodule
```

合成してFPGAエディタで確認

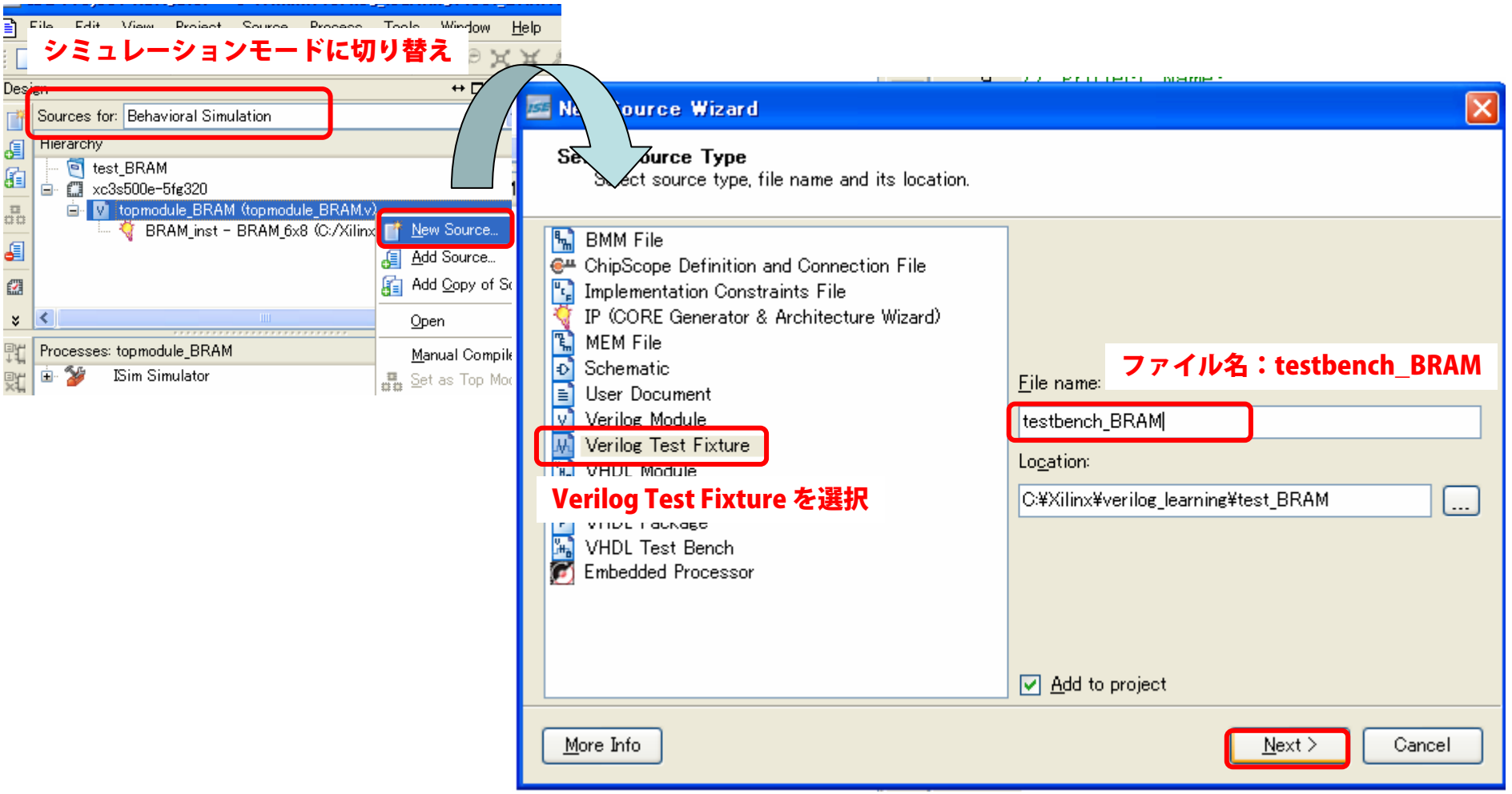




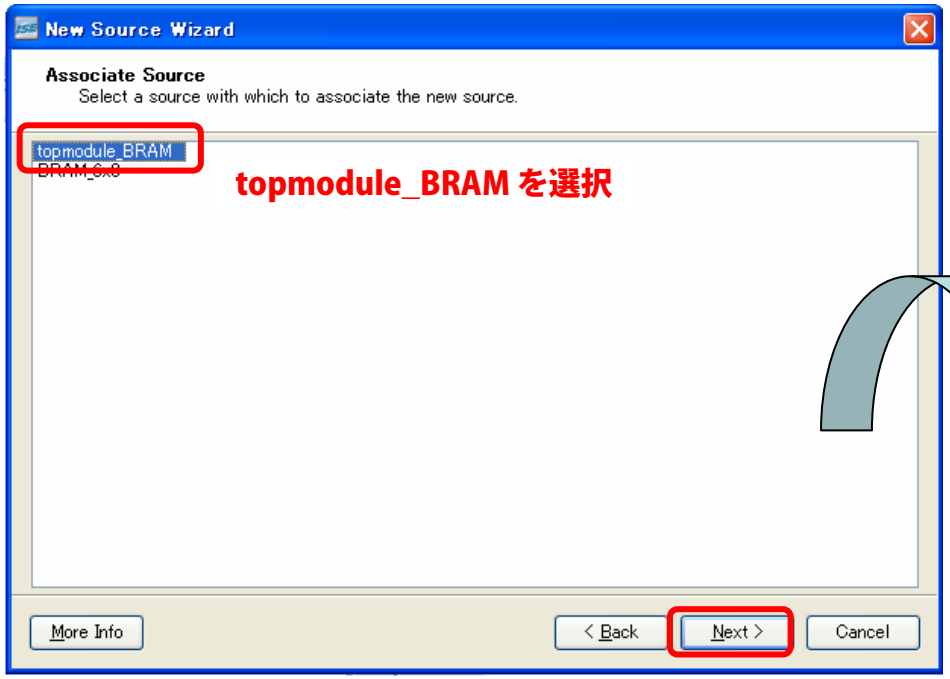
BRAMを拡大する

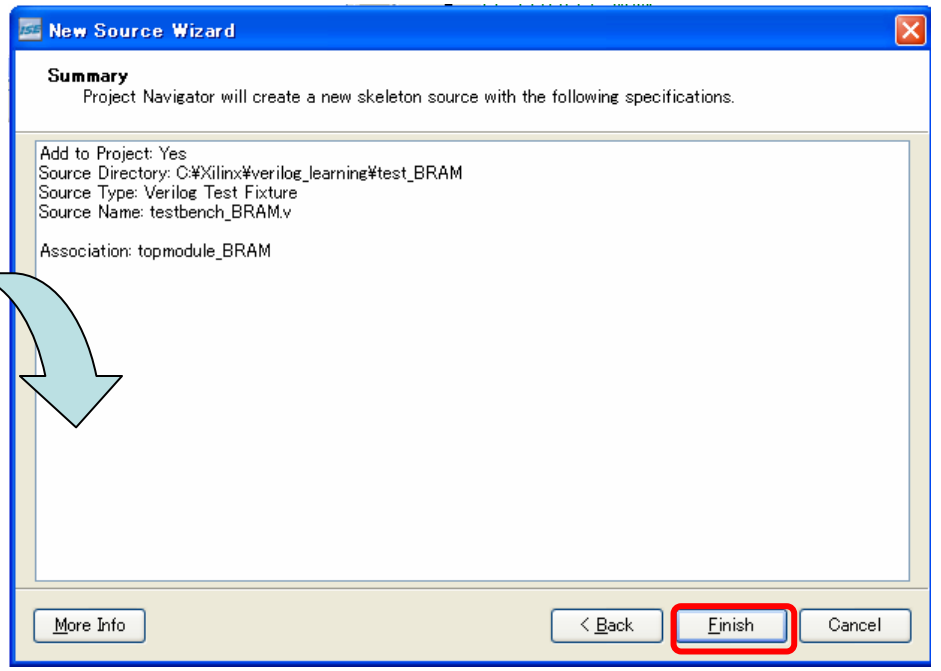

Block2 - View Site RAMB16_X0Y9 with Comp BRAM_inst/BU2/U0/blk_mem_generator/valid.c
ダブルクリック

テストベンチを追加


シミュレーションモードに切り替え
Sources for: Behavioral Simulation
Hierarchy
test_BRAM
xc3s500e-5fg320
topmodule_BRAM (topmodule_BRAM.v)
BRAM_inst - BRAM_6x8 (C:/Xilinx
New Source...
Add Source...
Add Copy of So
Open
Manual Compile
Set as Top Mod
Processes: topmodule_BRAM
ISim Simulator
Source Wizard
Select Source Type
Select source type, file name and its location.
BMM File
ChipScope Definition and Connection File
Implementation Constraints File
IP (CORE Generator & Architecture Wizard)
MEM File
Schematic
User Document
Verilog Module
Verilog Test Fixture
VHDL Module
VHDL Test Bench
Embedded Processor
Verilog Test Fixture を選択
File name:
testbench_BRAM
ファイル名：testbench_BRAM
Location:
C:¥Xilinx¥verilog_learning¥test_BRAM
Add to project
More Info
Next >
Cancel

シミュレーションの設定





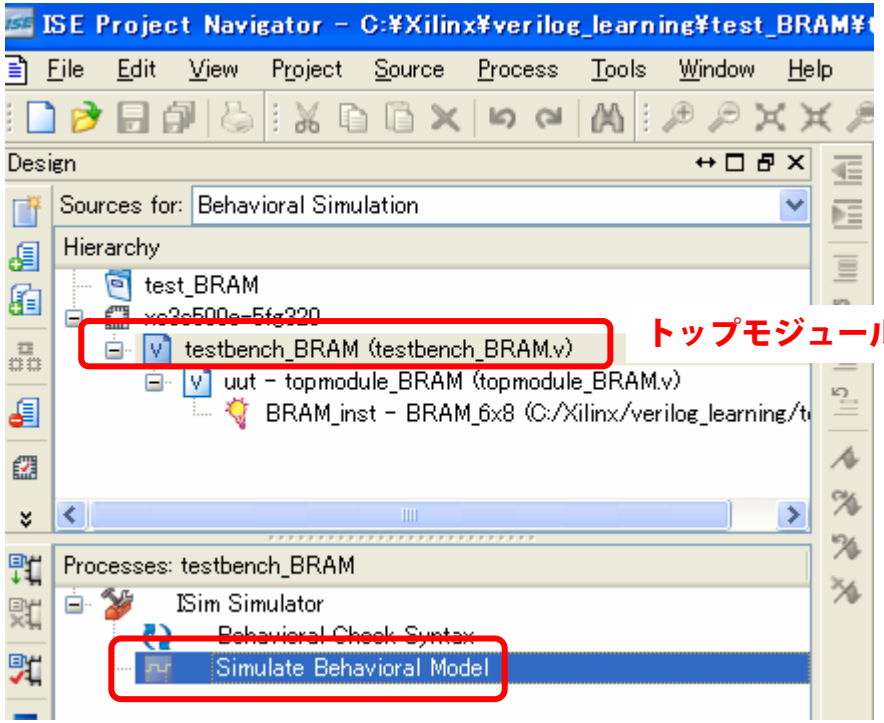
テストベンチの記述

```
45    initial begin
46       // Initialize Inputs
47       CLOCK = 0;
48       WE = 0;
49       ADR = 0;
50       DIN = 0;
51
52       // Wait 100 ns for global reset to finish
53       #100;
54
55       // Add stimulus here
56       ADR = 6'b000100;
57       DIN = 8'b10101010;
58       #100;
59       WE = 1'b1;
60       #100;
61       WE = 1'b0;
62       #100;
63       ADR = 6'b000101;
64       DIN = 8'b11001100;
65       #100;
66       WE = 1'b1;
67       #100;
68       WE = 1'b0;
69       #100;
70       ADR = 6'b000100;
71
72    end
73
74    initial begin
75       forever begin
76          #20 CLOCK = ~CLOCK;
77       end
78    end
```

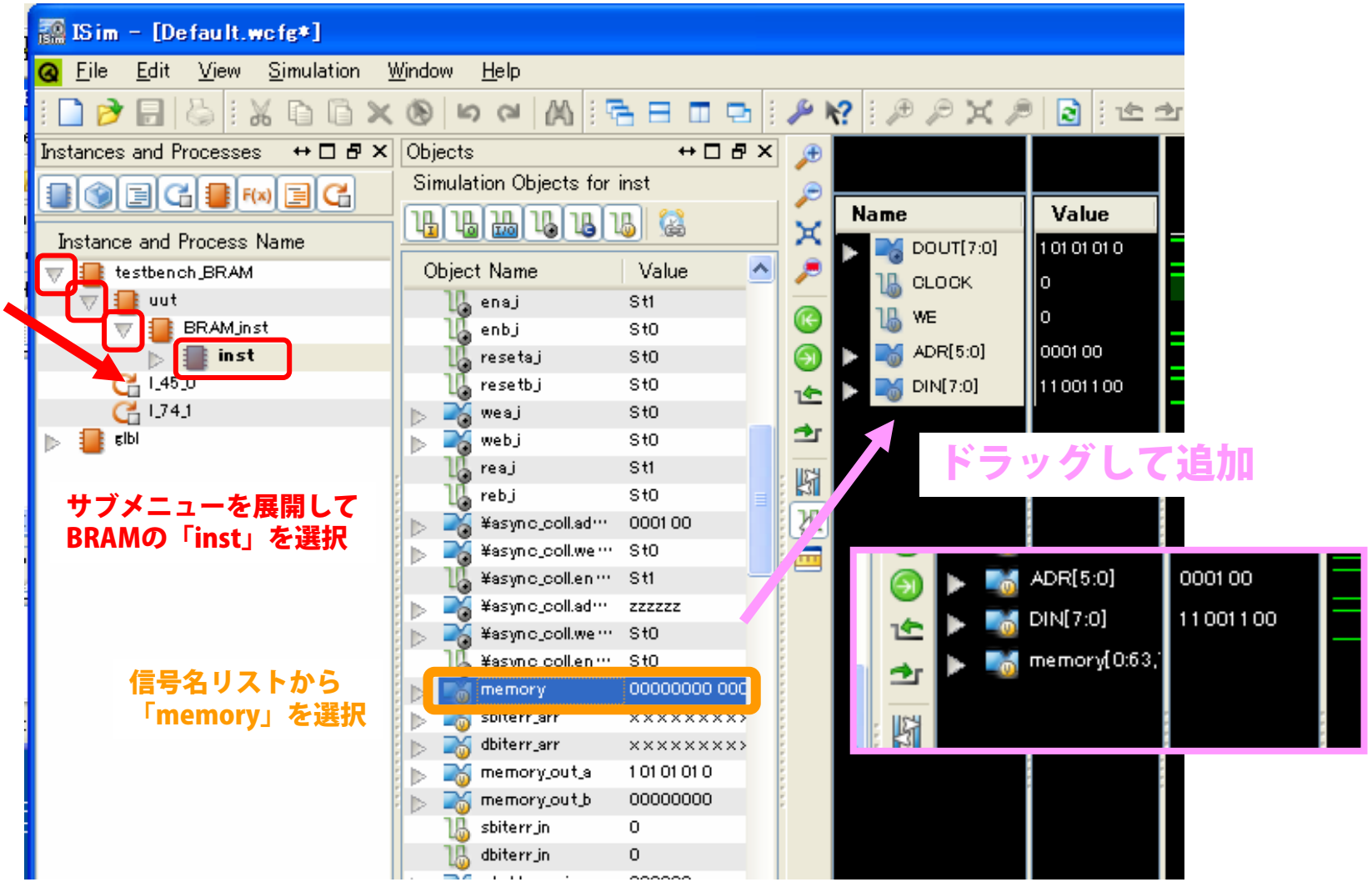
BRAMへの書き込み、
読み出しを記述

クロックを記述

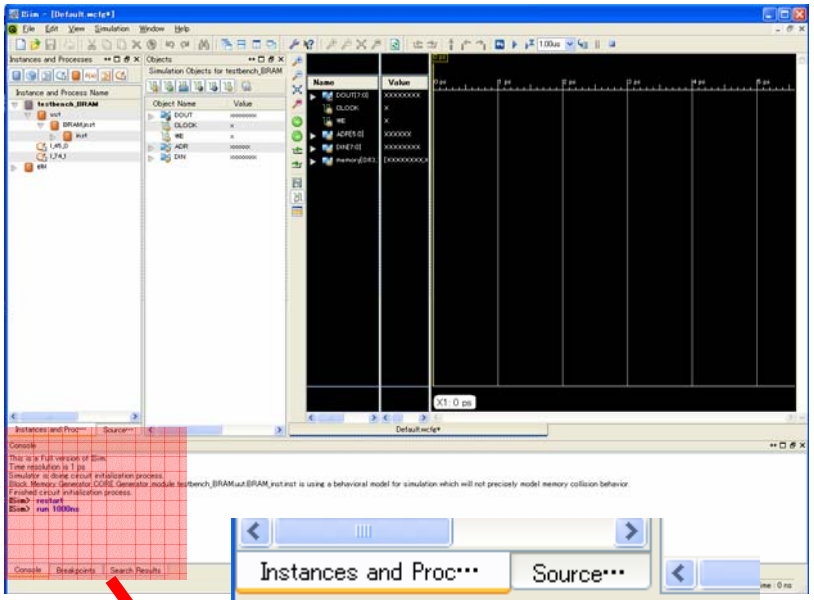
ISimの実行

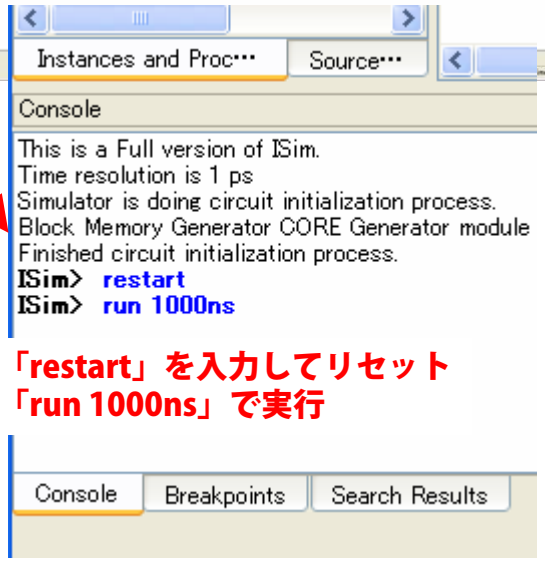

ISE Project Navigator - C:¥Xilinx¥verilog_learning¥test_BRAM¥
File Edit View Project Source Process Tools Window Help
Design
Sources for: Behavioral Simulation
Hierarchy
test_BRAM
testbench_BRAM (testbench_BRAM.v)
uut - topmodule_BRAM (topmodule_BRAM.v)
BRAM_inst - BRAM_6x8 (C:/Xilinx/verilog_learning/t
Processes: testbench_BRAM
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
トップモジュールを選択

メモリを信号リストに追加

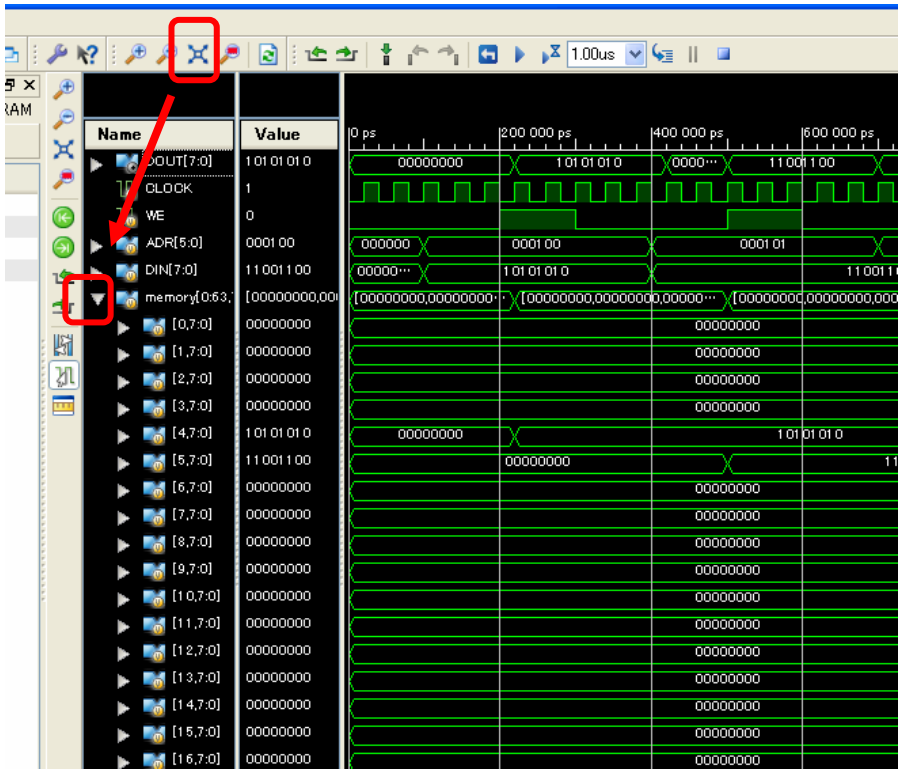


BRAMの中身を見る





「restart」を入力してリセット
「run 1000ns」で実行



BRAMのデータが書き換わっているのが
確認できます

まとめ

- XilinxのCORE Generatorを使ってBRAMを実装する方法を学習
- シミュレーションでBRAMの中身を確認する方法を学習

課題

- デュアルポートBRAMを実現し
  - 書き込み専用ポート
  - 読み出し専用ポート

  を実装してシミュレーションで動作を確認せよ
- .coeファイルにデータを書き込んだBRAMから1秒毎に読み出したデータをLEDに表示させよ
  - ただし、アドレスも1つずつインクリメントさせる