

실험 6. On-chip RAM & ROM 사용법

Digital System Design and Experiment

Made by MMS Students

Graduate School of Convergence Science and Technology

Seoul National University



Mobile Multimedia Systems Group

목표

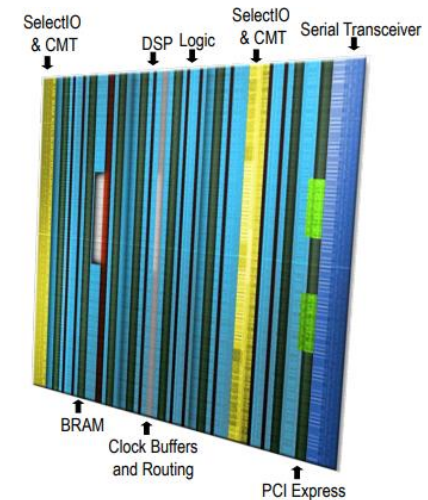
- FPGA 상에서 BRAM을 이용하여 RAM과 ROM을 어떻게 생성하는지 학습한다.
- 생성된 RAM&ROM을 사용하는 법을 익힌다.

1.

Background

FPGA 구조

- Xilinx FPGA는 크게 CLB, BRAM, IOB, DSP, PLL, programmable interconnect 등의 block들로 구성되어 있다.
- CLB : Configurable Logic Block, LUT를 통해 AND, OR 등 Logic을 구현한다.
- BRAM : Block RAM, FPGA 내부의 On-chip memory이다. 앞으로 사용할 보드(Nexys 4 DDR)의 경우 하나의 BRAM은 36Kb이고, 135개의 BRAM이 존재한다. (총 4860Kb)
- IOB : Input/Output Block, I/O 핀과 FPGA 내부 Logic 간의 양방향 인터페이스를 제공한다.
- DSP : Digital Signal Processor, MAC 연산을 고속으로 처리하기 위한 block



FPGA On-chip memory

- Distributed RAM : CLB 내부의 LUT를 Memory로 활용
- Block RAM : CLB 외부의 별도 Memory, Configuration에 따라 다르게 활용 가능
이번 실험에서는 Single-port ROM과 Dual-port RAM을 사용한다.
- Single-port RAM
- Dual-port RAM
- Single-port ROM
- Dual-port ROM

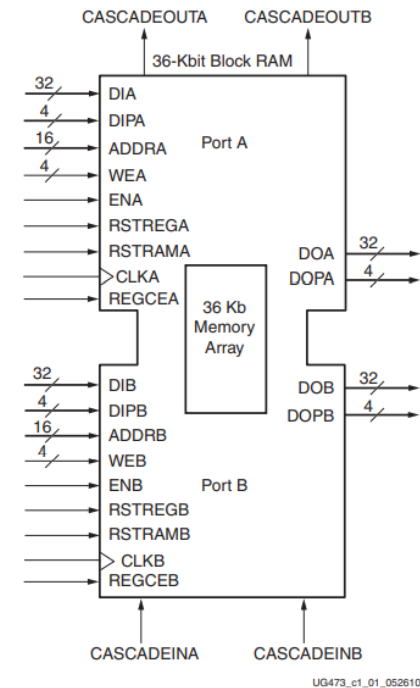


Figure 1-1: True Dual-Port Data Flows for a RAMB36

2.

IP Generator를 이용한 메모리 생성

IP Generator를 이용한 ROM 생성하기

- IP를 이용하여 ROM을 생성할 때, coe file을 이용하여 memory를 초기화 할 수 있다.
- coe 파일의 syntax는 아래의 예제와 같다.
- 이번 lab의 ROM은 data.npy 파일에 저장된 numpy 객체의 값으로 초기화 해야 한다. 이를 위해 data.npy 로부터 numpy 객체를 읽어와 .coe 파일을 생성해야 한다. 이는 ROM_init.py 를 실행하여 수행할 수 있다. 추후 final project에서도 비슷하게 사용되니 시간이 있다면 py 파일도 살펴보자. Python 파일 실행법은 etl에 함께 업로드 되어있는 python_setup.pdf를 참고하자.

```
*****  
***** Example of Dual Port Block Memory .COE file *****  
*****  
; This .COE file specifies the contents for a block memory  
; of depth=4, and width=4. In this case, values are specified  
; in binary format.  
memory_initialization_radix=2;  
memory_initialization_vector=  
1111,  
1111,  
1111,  
0000;
```

IP Generator를 이용한 ROM 생성하기

- IP Catalog 클릭 -> 검색 창에 Block Memory Generator 입력 -> 아래에 나타나는 Block Memory Generator 더블 클릭

The screenshot displays the Vivado IDE interface with the following components:

- Flow Navigator:** A sidebar on the left containing project management tasks such as Settings, Add Sources, Language Templates, IP Catalog (highlighted with a red box), IP INTEGRATOR, SIMULATION, RTL ANALYSIS, SYNTHESIS, and IMPLEMENTATION.
- PROJECT MANAGER - prj_mac:** The central workspace showing the project hierarchy. The 'Sources' pane lists design and simulation sources, while the 'Properties' pane at the bottom indicates 'Select an object to see properties'.
- IP Catalog:** A panel on the right showing search results for 'block' (entered in the search bar, also highlighted with a red box). The results are categorized by interfaces (AXI4).

Cores Interfaces	
Name ^ 1 AXI4	
Vivado Repository	
Alliance Partners	
Xylon	
2D Graphics Accelerator Bit Block Transfer AXI4	
Basic Elements	
Memory Elements	
Block Memory Generator	AXI4
Digital Signal Processing	
Building Blocks	
Complex Multiplier	AXI4-Stream
CORDIC	AXI4-Stream
Memories & Storage Elements	
Details	
Name:	Block Memory Generator
Version:	8.4 (Rev. 1)
Interfaces:	AXI4
Description:	The Xilinx LogiCORE IP Block Memory Generator replaces the Dual Port Block Memory an designs. The core supports RAM and ROM functions over a wide range of widths and dept

IP Generator를 이용한 ROM 생성하기

- Basic 탭에서 Component Name, Interface Type, Memory Type을 설정

Customize IP

Block Memory Generator (8.4)

Documentation IP Location Switch to Defaults

IP Symbol Power Estimation

☒ Show disabled ports

Component Name **blk_mem_gen_0**

Basic Port A Options Other Options Summary

Interface Type **Native** ☐ Generate address interface with 32 bits

Memory Type **Single Port ROM** ☐ Common Clock

ECC Options

ECC Type **No ECC**

☐ Error Injection Pins **Single Bit Error Injection**

Write Enable

☐ Byte Write Enable

Byte Size (bits) **9**

Algorithm Options

Defines the algorithm used to concatenate the block RAM primitives.
Refer datasheet for more information.

Algorithm **Minimum Area**

Primitive **8kx2**

OK Cancel

IP Generator를 이용한 ROM 생성하기

- Port A options 탭에서 Port A Width와 Depth를 설정

The screenshot shows the 'Customize IP' window for the 'Block Memory Generator (8.4)'. The 'Port A Options' tab is selected. The 'Memory Size' section shows 'Port A Width' set to 8 and 'Port A Depth' set to 1024, both highlighted with a red rectangle. The 'Operating Mode' is set to 'Write First' and 'Enable Port Type' is 'Use ENA Pin'. The 'Port A Optional Output Registers' section has 'Primitives Output Register' checked. The 'Port A Output Reset Options' section has 'RSTA Pin (set/reset pin)' checked with an output reset value of 0. The 'READ Address Change A' section has 'Read Address Change A' unchecked.

Customize IP

Block Memory Generator (8.4)

Documentation IP Location Switch to Defaults

IP Symbol Power Estimation

☒ Show disabled ports

Component Name blk_mem_gen_0

Basic Port A Options Other Options Summary

Memory Size

Port A Width 8 Range: 1 to 4608 (bits)

Port A Depth 1024 Range: 2 to 1048576

The Width and Depth values are used for Read Operation in Port A

Operating Mode Write First Enable Port Type Use ENA Pin

Port A Optional Output Registers

☒ Primitives Output Register ☐ Core Output Register

☐ SoftECC Input Register ☐ REGCEA Pin

Port A Output Reset Options

☒ RSTA Pin (set/reset pin) Output Reset Value (Hex) 0

☐ Reset Memory Latch Reset Priority CE (Latch or Register Enable)

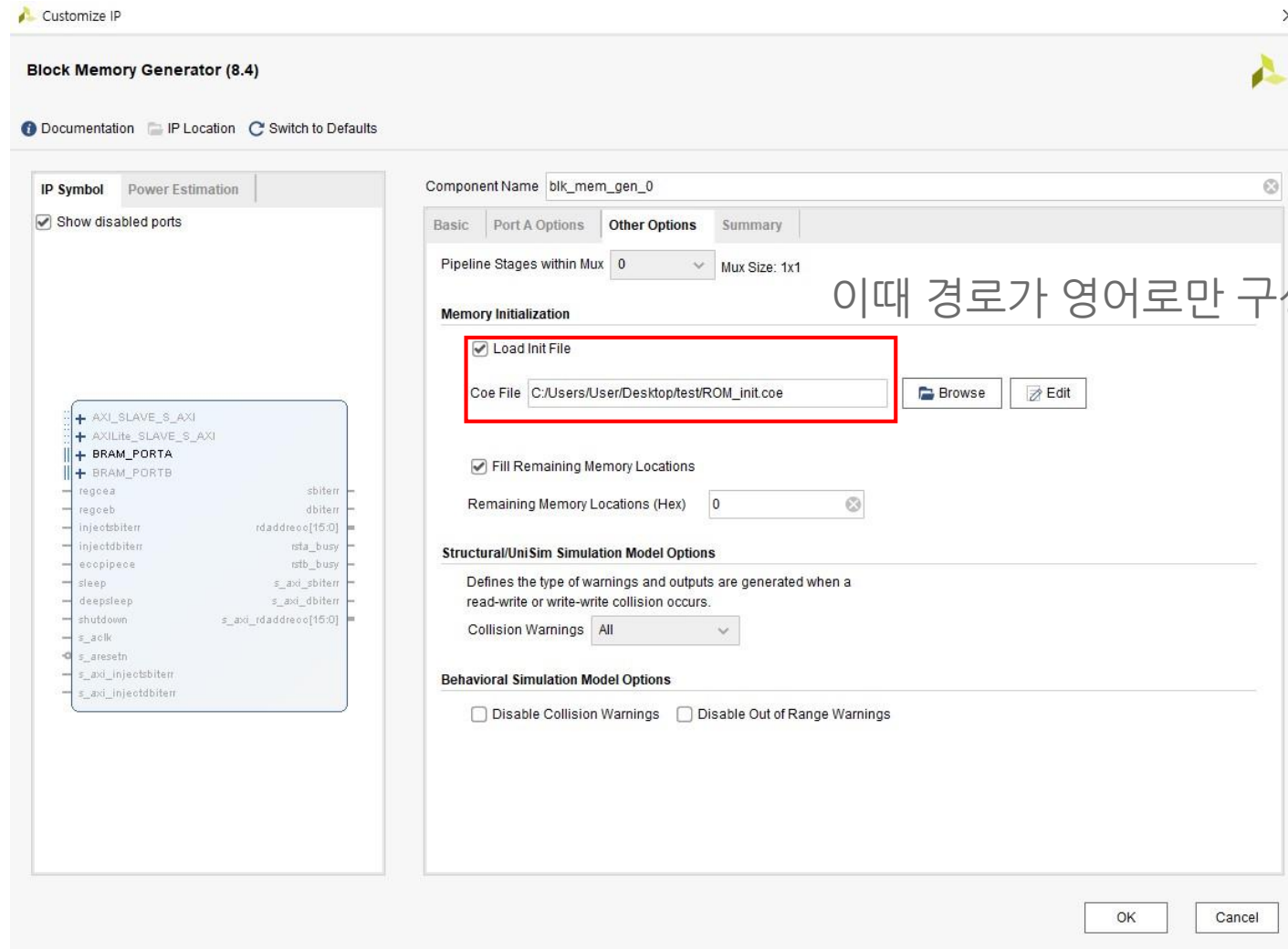
READ Address Change A

☐ Read Address Change A

OK Cancel

IP Generator를 이용한 ROM 생성하기

- Other options 탭에서 먼저 생성해 놓은 coe 파일을 올려준다.

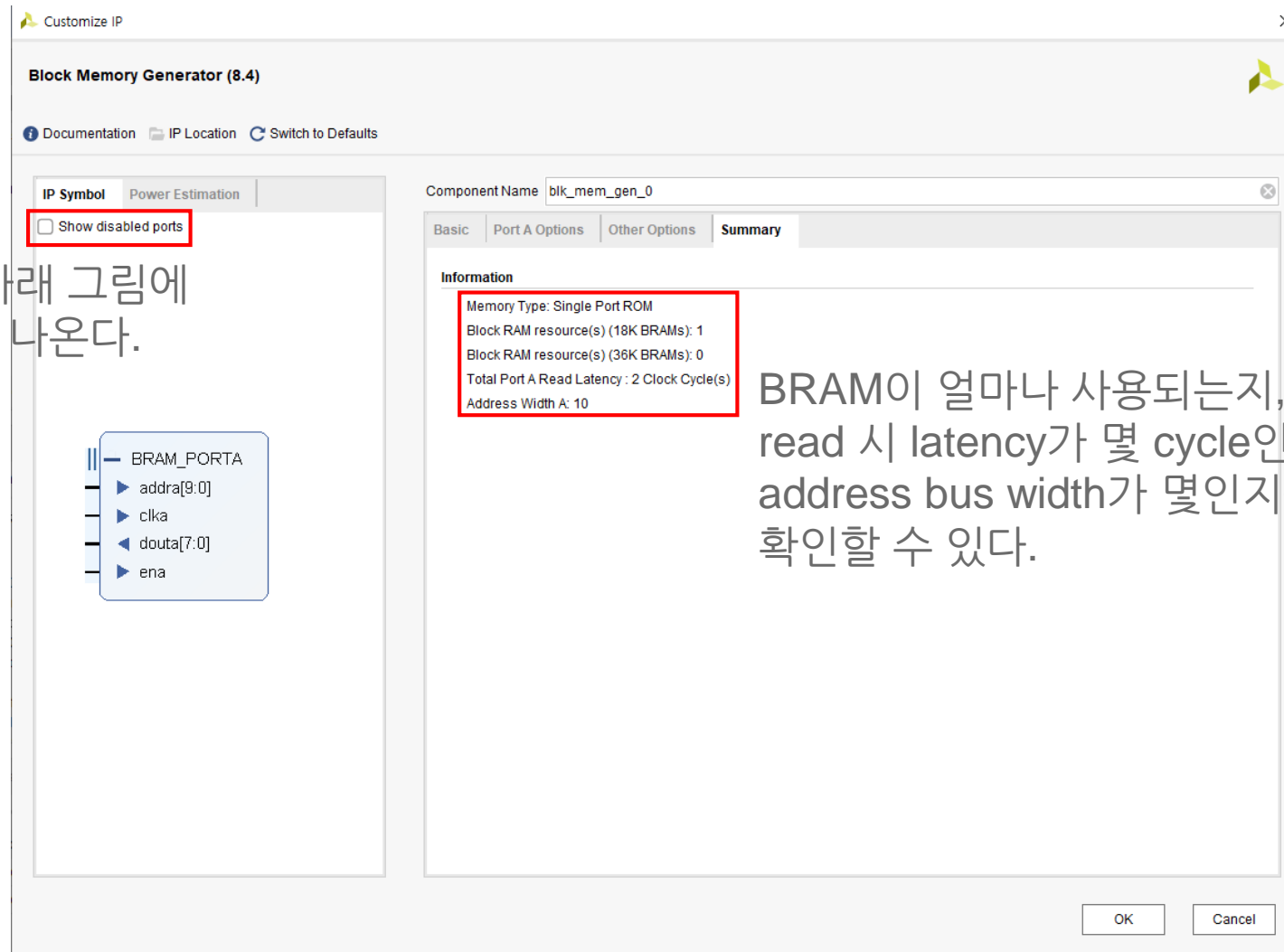


이때 경로가 영어로만 구성되어야 한다

IP Generator를 이용한 ROM 생성하기

■ Summary

체크를 풀면 아래 그림에
필요한 port만 나온다.



BRAM이 얼마나 사용되는지,
read 시 latency가 몇 cycle인지,
address bus width가 몇인지를
확인할 수 있다.

| IP Generator를 이용한 RAM 생성하기

- RAM의 경우 ROM과 마찬가지로 coe 파일을 통해 data를 초기화 할 수 있으나, 이번 lab에서는 생략하도록 한다.
- RAM은 ROM과 달리 data를 쓰는 것이 가능하여 write port가 존재한다.
- Single Port RAM
 - 하나의 port로 read, write 모두 수행
- Simple Dual Port RAM
 - 총 2개의 port로 구성. 하나는 read 전용, 다른 하나는 write 전용
- True Dual Port RAM
 - 2개의 port로, 모두 read, write 지원
- 이번 lab에서 사용할 RAM은 Simple Dual Port RAM이다.

IP Generator를 이용한 RAM 생성하기

- IP Catalog 클릭 -> 검색 창에 Block Memory Generator 입력 -> 아래에 나타나는 Block Memory Generator 더블 클릭

The screenshot displays the Vivado IDE interface with the following components:

- Flow Navigator (Left):** A tree view showing project stages. The 'IP Catalog' option under 'PROJECT MANAGER' is highlighted with a red box.
- Project Manager (Top):** Displays the project name 'prj_mac'.
- Sources (Middle):** A list of design sources including 'main (main.v) (2)', 'tb (tb.v) (1)', and 'u_main : main (main.v) (2)'. The 'Hierarchy' tab is selected.
- Properties (Bottom):** A panel for viewing object properties, currently showing 'Select an object to see properties'.
- IP Catalog (Right):** A search interface with a search bar containing 'block' (highlighted with a red box). Below the search bar, a table lists search results:

Name	AXI4
Vivado Repository	
Alliance Partners	
Xylon	
2D Graphics Accelerator Bit Block Transfer	AXI4
Basic Elements	
Memory Elements	
Block Memory Generator	AXI4
Digital Signal Processing	
Building Blocks	
Complex Multiplier	AXI4-Stream
CORDIC	AXI4-Stream
Memories & Storage Elements	

Below the table, the 'Details' section for the 'Block Memory Generator' is shown:

- Name:** Block Memory Generator
- Version:** 8.4 (Rev. 1)
- Interfaces:** AXI4
- Description:** The Xilinx LogiCORE IP Block Memory Generator replaces the Dual Port Block Memory and designs. The core supports RAM and ROM functions over a wide range of widths and depths.

IP Generator를 이용한 RAM 생성하기

- Basic 탭에서 Component Name, Interface Type, Memory Type을 설정

Customize IP

Block Memory Generator (8.4)

Documentation IP Location Switch to Defaults

IP Symbol Power Estimation

☐ Show disabled ports

Component Name **blk_mem_gen_1**

Basic Port A Options Port B Options Other Options Summary

Interface Type **Native** ☐ Generate address interface with 32 bits

Memory Type **Simple Dual Port RAM** ☐ Common Clock

ECC Options

ECC Type **No ECC**

☐ Error Injection Pins **Single Bit Error Injection**

Write Enable

☐ Byte Write Enable

Byte Size (bits) **9**

Algorithm Options

Defines the algorithm used to concatenate the block RAM primitives. Refer datasheet for more information.

Algorithm **Minimum Area**

Primitive **8x2**

OK Cancel

Byte Write Enables를 체크하면, Byte 단위로 data를 write할 수 있다. Data width가 24bit이고 3byte중 두번째 byte만 수정하고 싶을 때 enable bus에 3'b010을 인가하여 write하면 된다.

IP Generator를 이용한 RAM 생성하기

- Port A options 탭에서 Port A Width와 Depth, enable type을 설정

Customize IP

Block Memory Generator (8.4)

[Documentation](#) [IP Location](#) [Switch to Defaults](#)

IP Symbol | Power Estimation

☐ Show disabled ports

Component Name: blk_mem_gen_1

Basic | **Port A Options** | Port B Options | Other Options | Summary

Memory Size

Port A Width: 8 (Range: 1 to 4608 (bits))

Port A Depth: 1024 (Range: 2 to 1048576)

The Width and Depth values are used for Write Operations in Port A

Operating Mode: Read First | Enable Port Type: Always Enabled

Port A Optional Output Registers

☐ Primitives Output Register ☐ Core Output Register

☐ SoftECC Input Register ☐ REGCEA Pin

Port A Output Reset Options

☐ RSTA Pin (set/reset pin) | Output Reset Value (Hex): 0

☐ Reset Memory Latch | Reset Priority: CE (Latch or Register Enable)

READ Address Change A

☐ Read Address Change A

OK Cancel

IP Generator를 이용한 RAM 생성하기

- Port B options 탭은 확인 후 넘어가자

The screenshot shows the 'Block Memory Generator (8.4)' window with the 'Port B Options' tab selected. The 'Component Name' is 'blk_mem_gen_1'. The 'Memory Size' section shows 'Port B Width' as 8 and 'Port B Depth' as 1024. The 'Operating Mode' is 'Write First' and 'Enable Port Type' is 'Use ENB Pin'. The 'Port B Optional Output Registers' section has 'Primitives Output Register' checked. The 'Port B Output Reset Options' section has 'RSTB Pin (set/reset pin)' checked and 'Reset Memory Latch' unchecked. The 'READ Address Change B' section has 'Read Address Change B' unchecked. The 'IP Symbol' tab on the left shows 'BRAM_PORTA' and 'BRAM_PORTB'.

Customize IP

Block Memory Generator (8.4)

Documentation IP Location Switch to Defaults

IP Symbol Power Estimation

☐ Show disabled ports

Component Name blk_mem_gen_1

Basic Port A Options **Port B Options** Other Options Summary

Memory Size

Port B Width 8

Port B Depth : 1024

The Width and Depth values are used for Read Operation in Port B

Operating Mode Write First Enable Port Type Use ENB Pin

Port B Optional Output Registers

☒ Primitives Output Register ☐ Core Output Register

☐ SoftECC Output Register ☐ REGCEB Pin

Port B Output Reset Options

☒ RSTB Pin (set/reset pin) Output Reset Value (Hex) 0

☐ Reset Memory Latch Reset Priority CE (Latch or Register Enable)

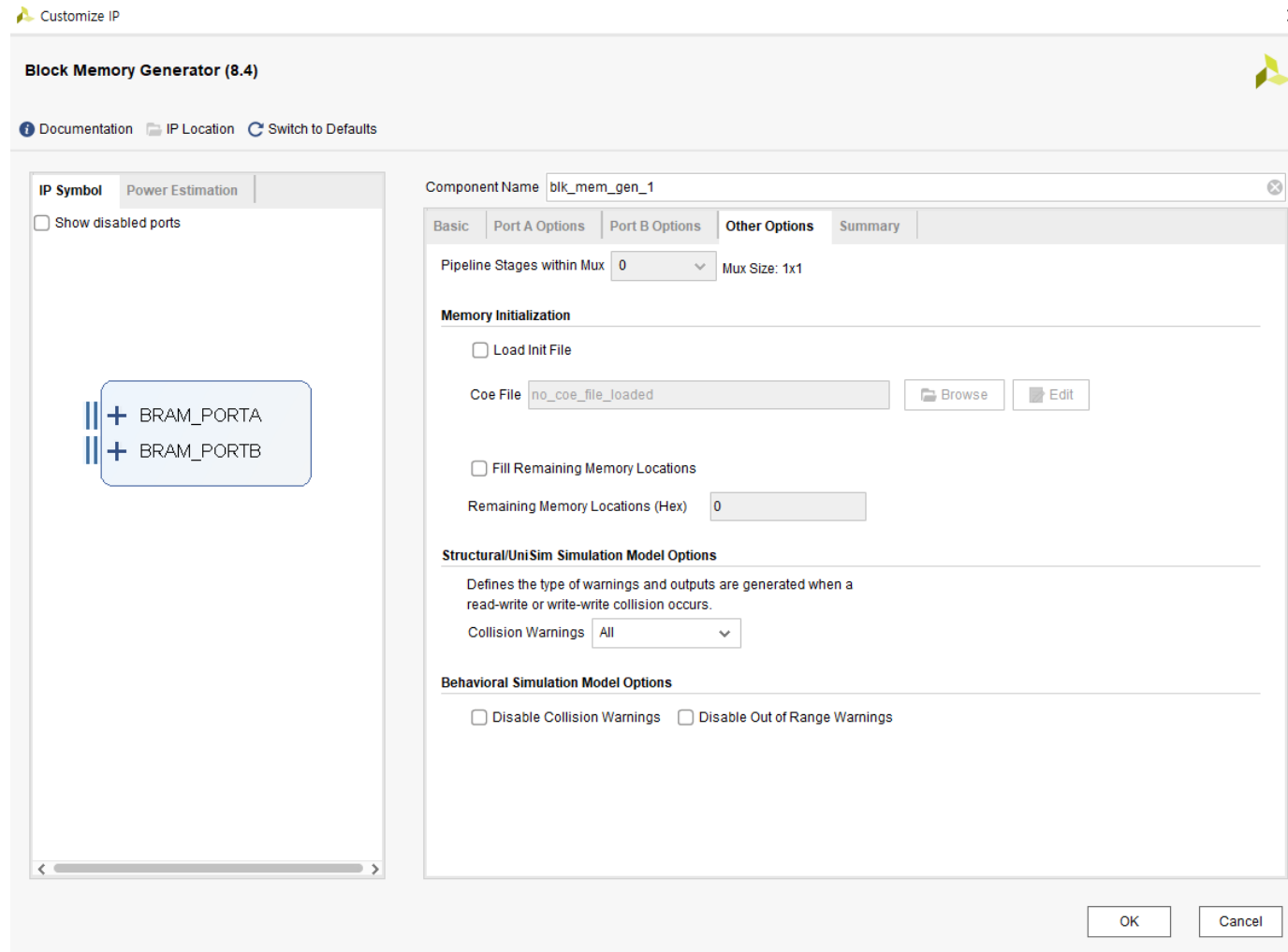
READ Address Change B

☐ Read Address Change B

OK Cancel

IP Generator를 이용한 RAM 생성하기

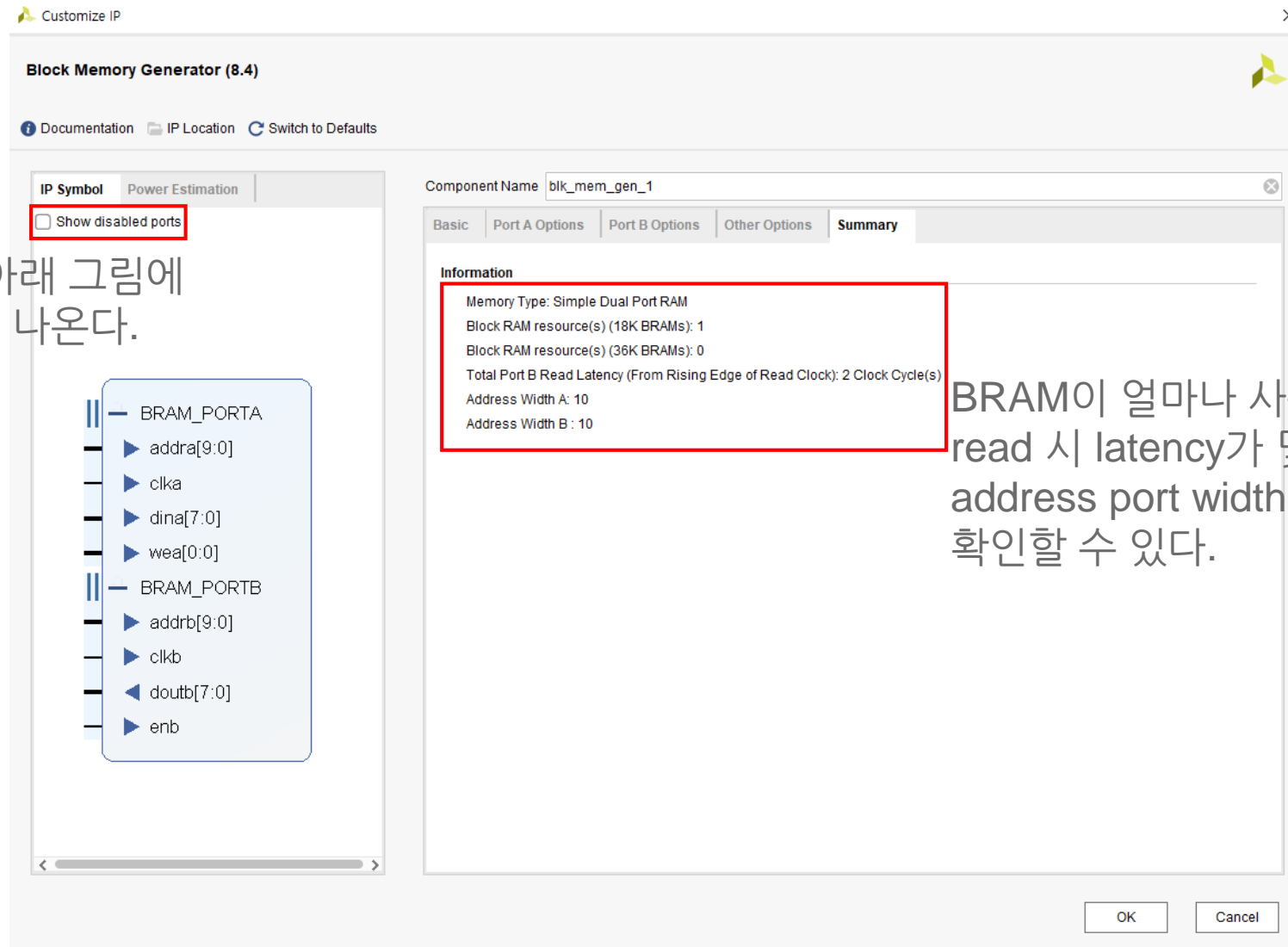
- Other options 탭에서 RAM은 memory 초기화를 하지 않음



IP Generator를 이용한 RAM 생성하기

■ Summary

체크를 풀면 아래 그림에
필요한 port만 나온다.



BRAM이 얼마나 사용되는지,
read 시 latency가 몇 cycle인지,
address port width가 몇인지를
확인할 수 있다.

3.

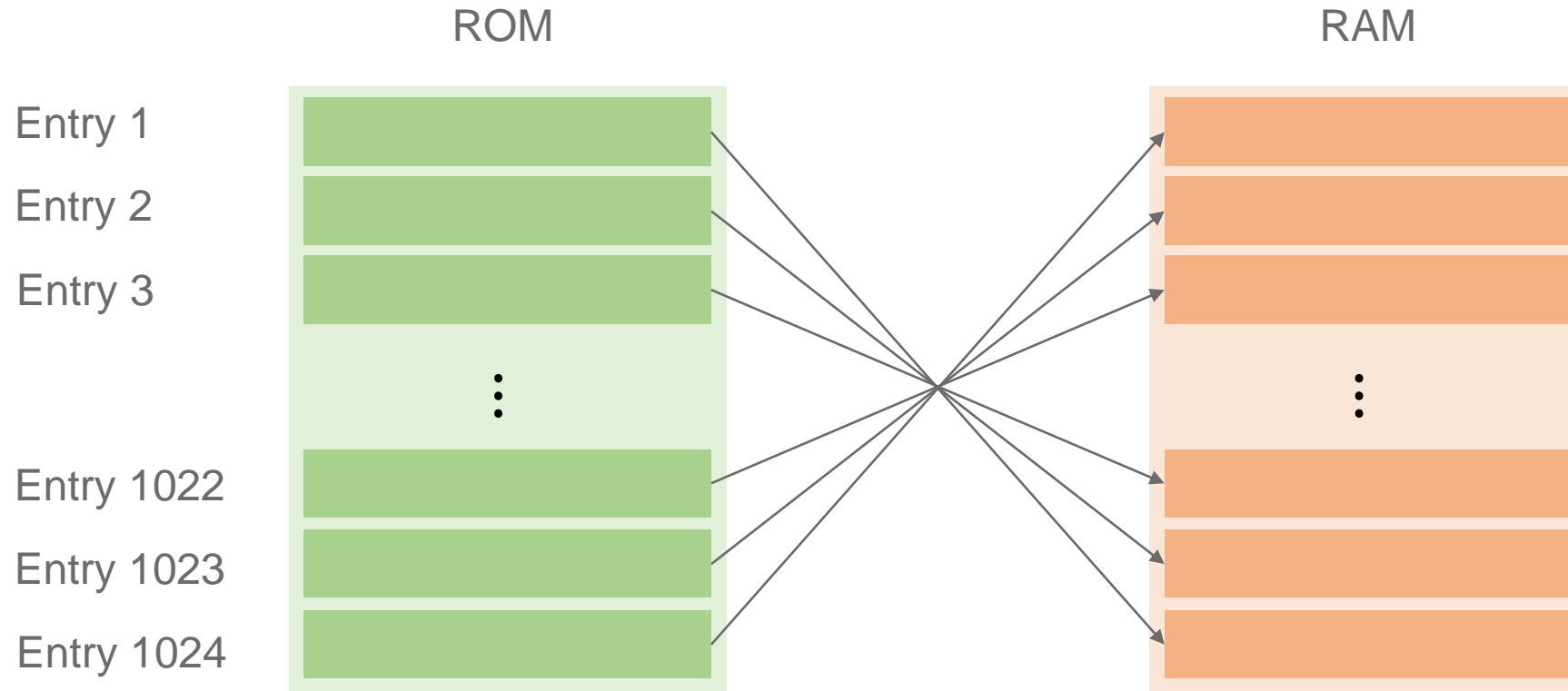
BRAM read / write

BRAM read / write

- RAM과 ROM 모두 동일한 Block RAM을 configure 해서 사용하므로 read 동작이 동일하다.
- 따라서 이번 lab에서는 ROM에서 data를 읽어와 RAM에 저장하고, testbench를 이용하여 제대로 저장되었는지를 확인하여, read와 write를 경험해보고자 한다.
- RAM과 ROM 모두 8-bit data 1024개를 저장할 수 있다. 따라서 address는 10bit이다.
- Read
 - Input clock의 rise edge에서 enable이 high일 때, 인가된 address 위치의 data가 2 clock cycle 후에 dout port로 반환된다.
- Write
 - Input clock의 rise edge에서 write enable이 high일 때, 인가된 din의 data가 인가된 address 위치에 write 된다.

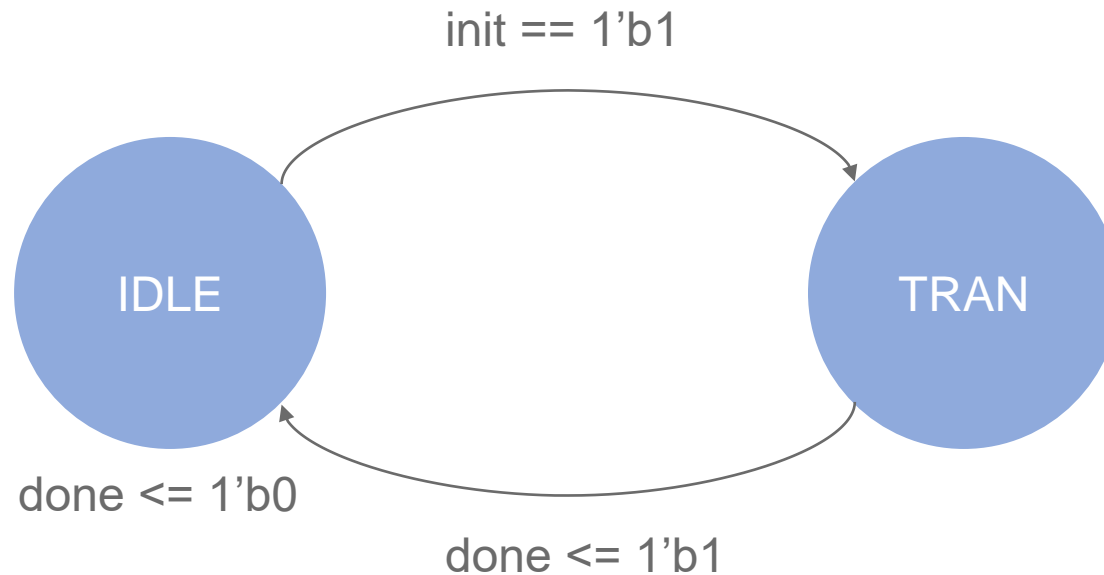
BRAM read / write

- init signal이 high로 인가되면, ROM의 data를 읽어와 모두 RAM에 저장하고, done signal을 high로 인가한다.
- 이 때, 1번째 data는 1024번째에, 2번째 data는 1023번째 순으로 역순으로 저장해야 한다.



FSM of Top

- 외부에서 init port에 1cycle 동안 high를 인가하면, TRAN state로 넘어가 ROM에서 RAM으로 data를 옮긴다.
- 위의 동작이 모두 마무리되면, 1cycle 동안 done을 high로 인가하고, IDLE state로 넘어간다.



4.

Overview of Lab 6

| 실험 제공 파일

- **bram_top.v**
 - 수정해야 할 Verilog 파일. TODO 부분을 수정해야 함
 - blk_mem_gen_0, blk_mem_gen_1 은 직접 생성한 ip로 바뀌어야 함
- **bram_tb.v**
 - data를 제대로 옮겼는지 확인하는 testbench
 - **data.txt의 경로를 본인의 환경에 맞게 변경**
- **Nexys4DDR_Master.xdc**
 - Post-synthesis simulation을 위해 clock period를 지정한 constraint 파일
- **data.txt**
 - RAM에 제대로 data를 옮겼는지 확인하기 위한 비교용 파일.
- **ROM_init.py / data.npy**
 - ROM_init.py 를 실행하여 data.npy 로부터 data를 뽑아내 coe 파일을 생성할 수 있음
 - python code 실행 방법은 python_setup.pdf 참고

Post-synthesis simulation 결과 확인

- TODO 부분의 코드를 작성한 후 synthesis를 수행하고 simulation까지 완료
- Tcl console을 확인하여 테스트를 통과하였는지 확인

```
*****  
  
test passed! 1024 correct!  
  
*****
```

Pass

```
*****  
  
1th data doesn't match with expected value!  
  
*****
```

Fail

결과 제출

- 제출 파일
 - 구현한 bram_top.v 파일
 - Design runs 탭의 synthesis 부분 capture한 이미지 파일
 - Post-synthesis timing simulation 화면과 Tcl console 창의 test passed 부분을 같이 capture한 이미지 파일
- File name: “학번_이름_lab6.zip”
 - ex) 2021-12345_홍길동_lab6.zip
- Deadline: 10월 21일 (금) 23:59 까지