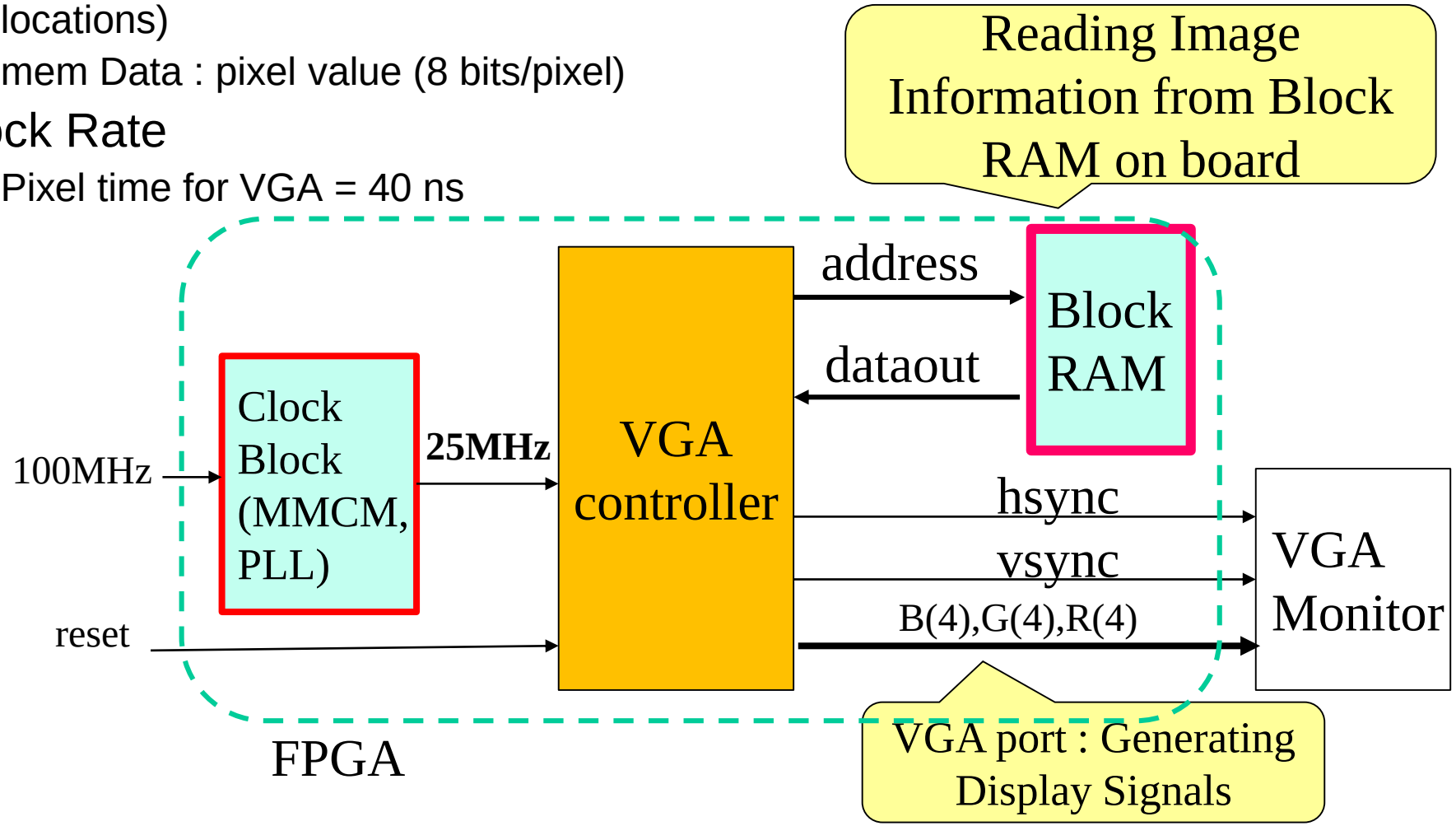
VGA Interface (#2) with Memory

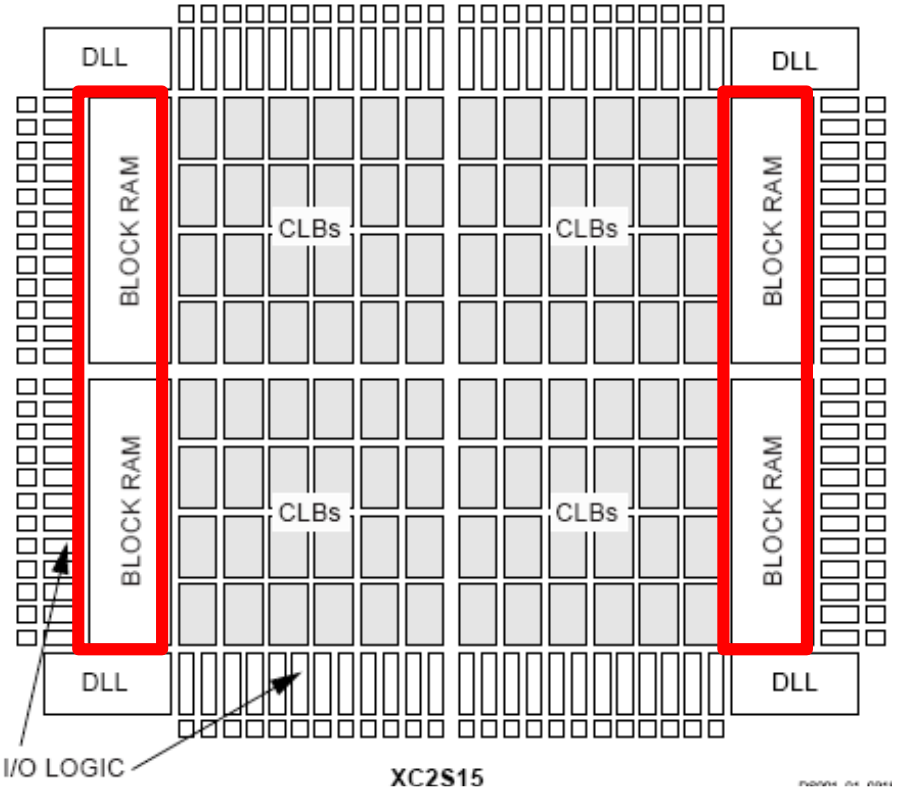
Kang Yi

IP Cores for our VGA controller

- Image is stored in a Memory
  - mem Addr : address output(19bits for 640x480 locations)
  - mem Data : pixel value (8 bits/pixel)
- Clock Rate
  - Pixel time for VGA = 40 ns



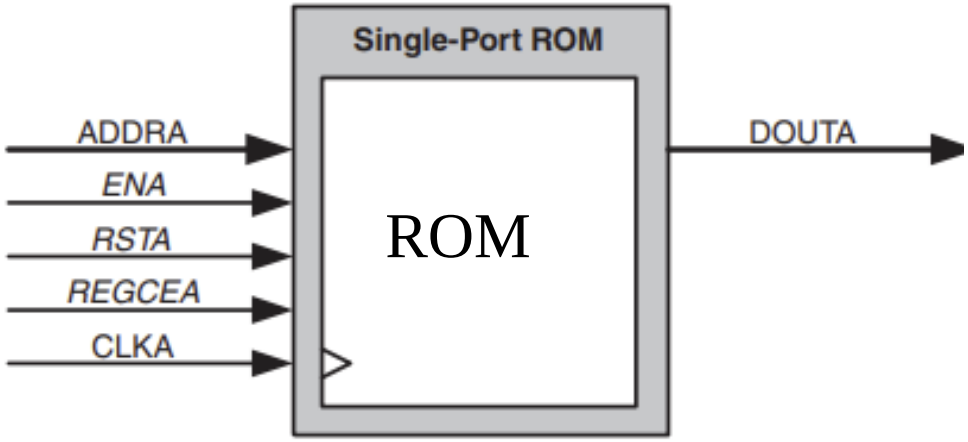
Block RAM Inside FPGA

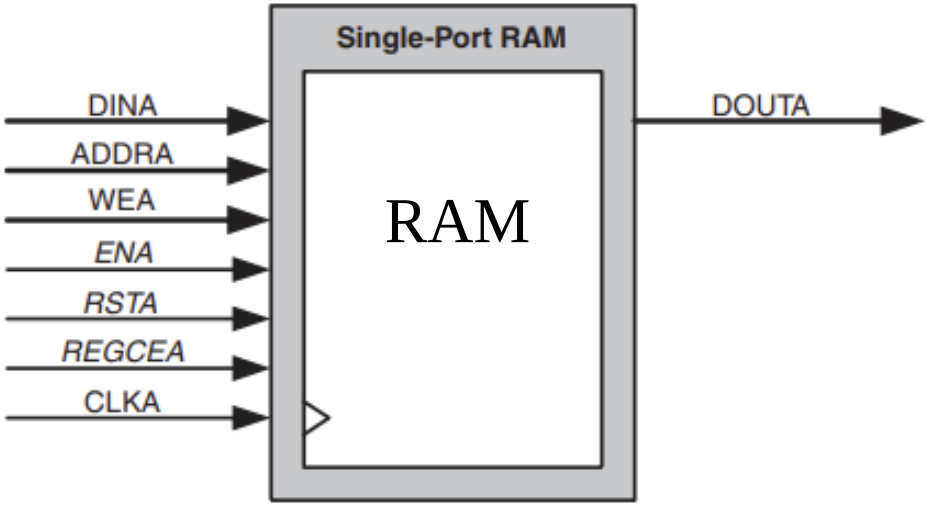


- Block RAM (BRAM)
  - Embedded Memories in FPGA
  - Size: Atrix7 XC7A100T has 4,860Kb = 607.5KB

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices[2]	Block RAM Blocks[3]		
		Slices[1]	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)
XC7A12T	12,800	2,000	171	40	40	20	720
XC7A15T	16,640	2,600	200	45	50	25	900
XC7A25T	23,360	3,650	313	80	90	45	1,620
XC7A35T	33,280	5,200	400	90	100	50	1,800
XC7A50T	52,160	8,150	600	120	150	75	2,700
XC7A75T	75,520	11,800	892	180	210	105	3,780
XC7A100T	101,440	15,850	1,188	240	270	135	4,860
XC7A200T	215,360	33,650	2,888	740	730	365	13,140

Block RAM Types: Single Port



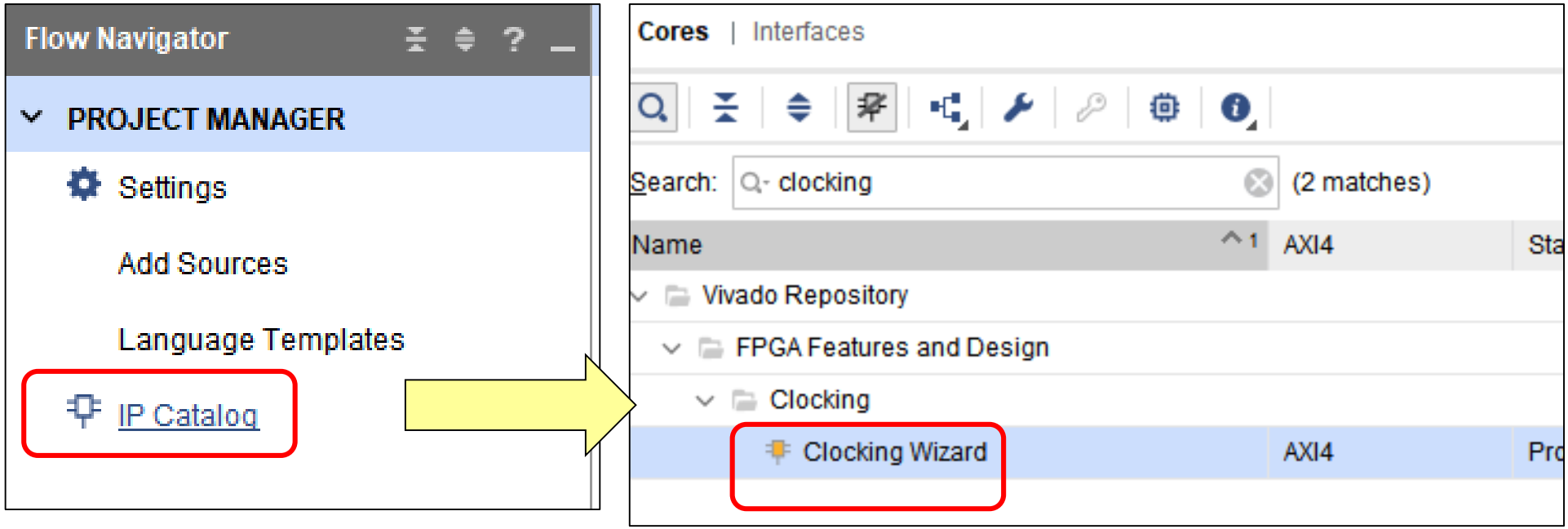


Control Signals Port A

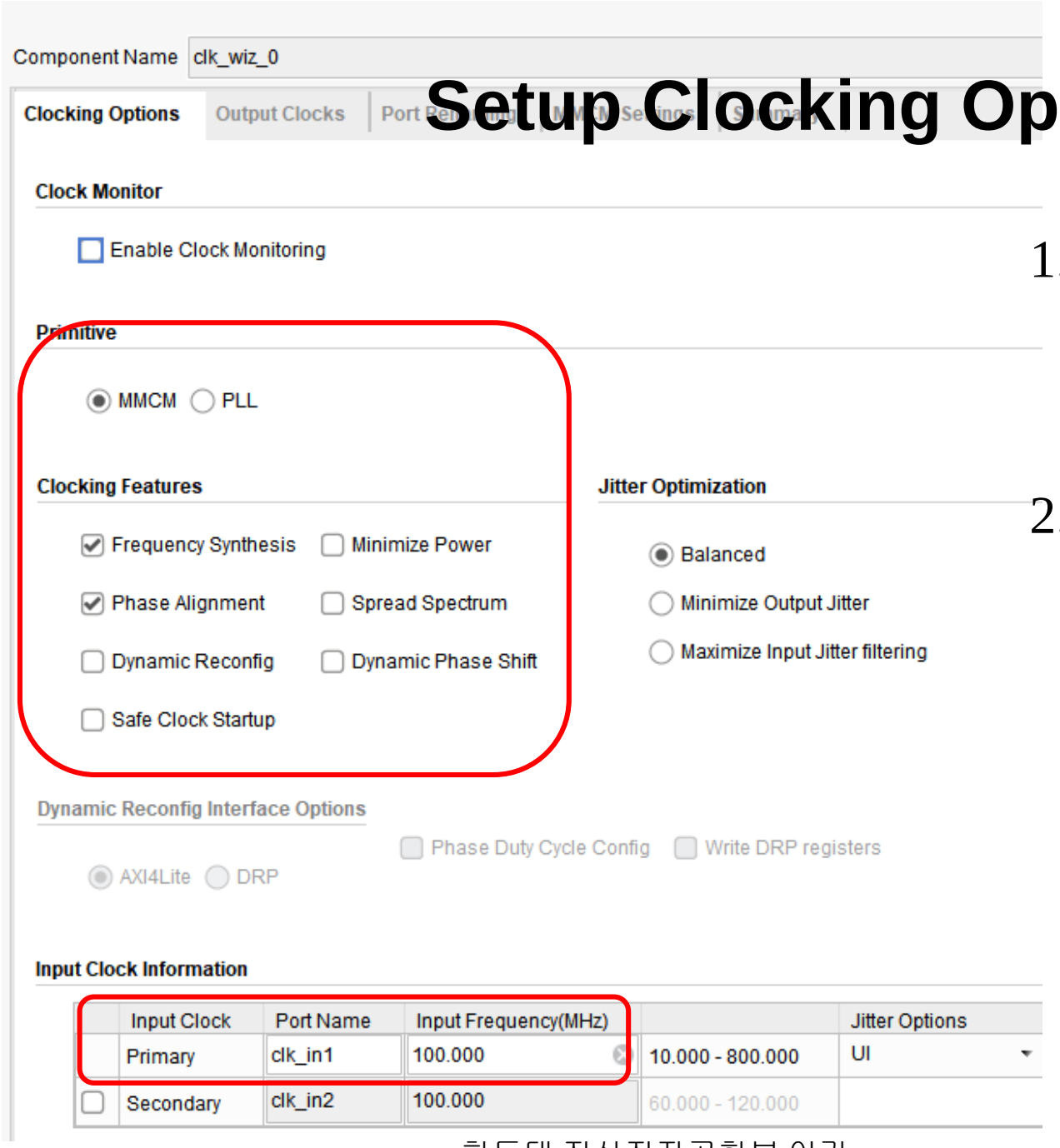
- ADDRA: Address of block memory for read or write operation
- DINA : Data Input to be written
- ENA: Enable write, read, and reset operation
- WEA : Enable write operation
- RSTA: Reset the memory output
- REGCEA: Enables the last output register (if optional output register is used)

Adding Clocking IP (Intellectual Property)

1. Project Manager → IP Catalog
2. Search with keyword “clocking”
3. Choose clocking wizard



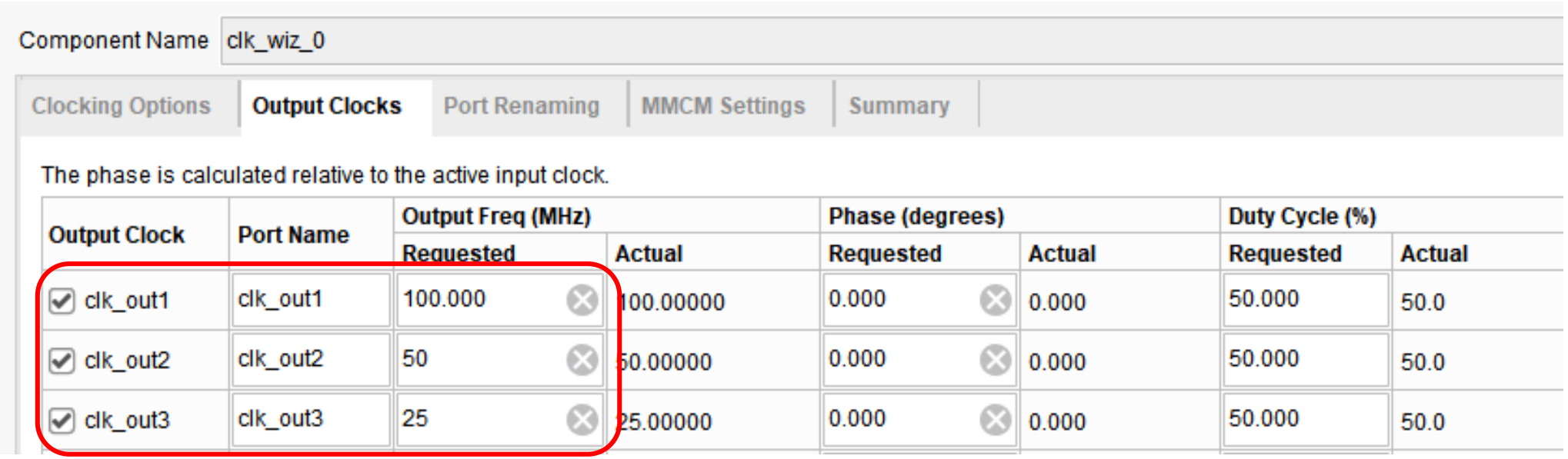
Setup Clocking Options



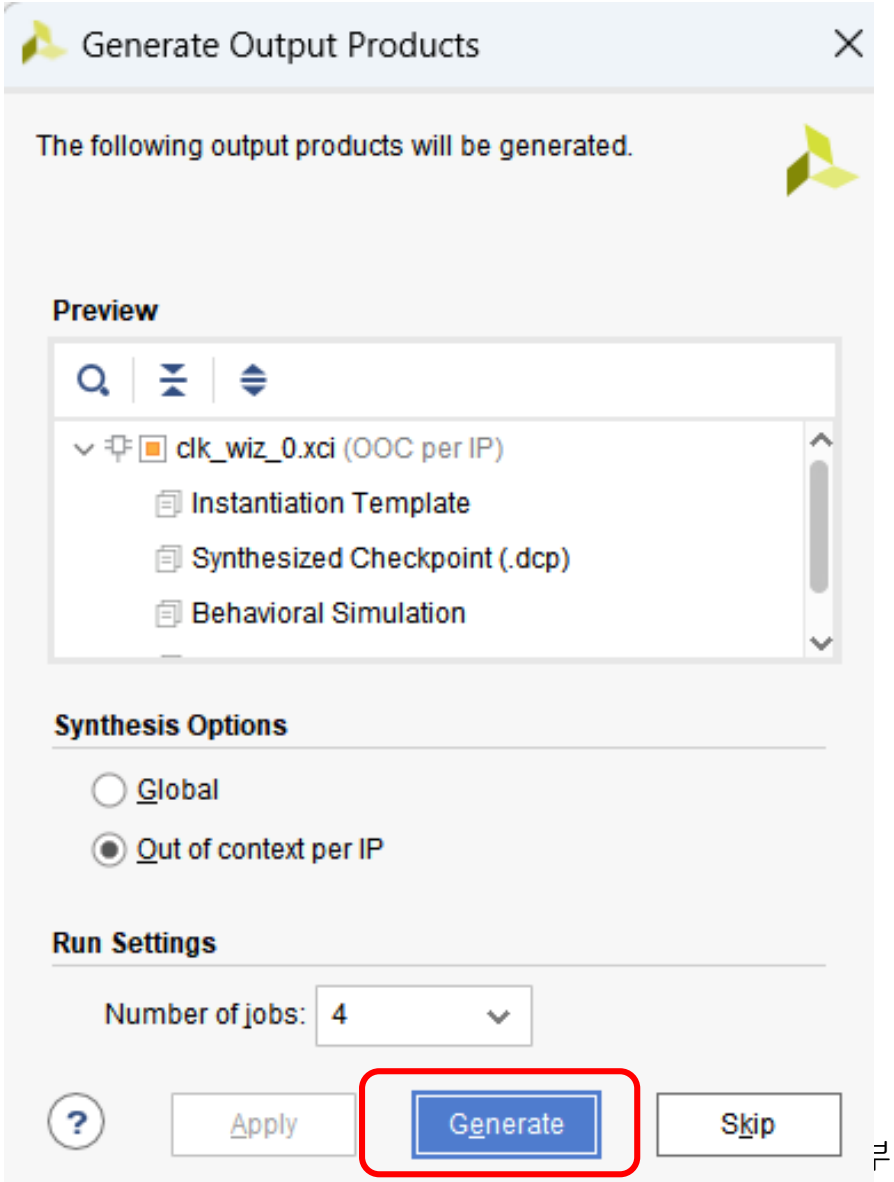
1. Clocking Featuers
   - MMCM
   - Frequency Synthesis
   - Phase Alignment
2. Input Clock: clk_in1 = 100MHz

Setup Output Clocks

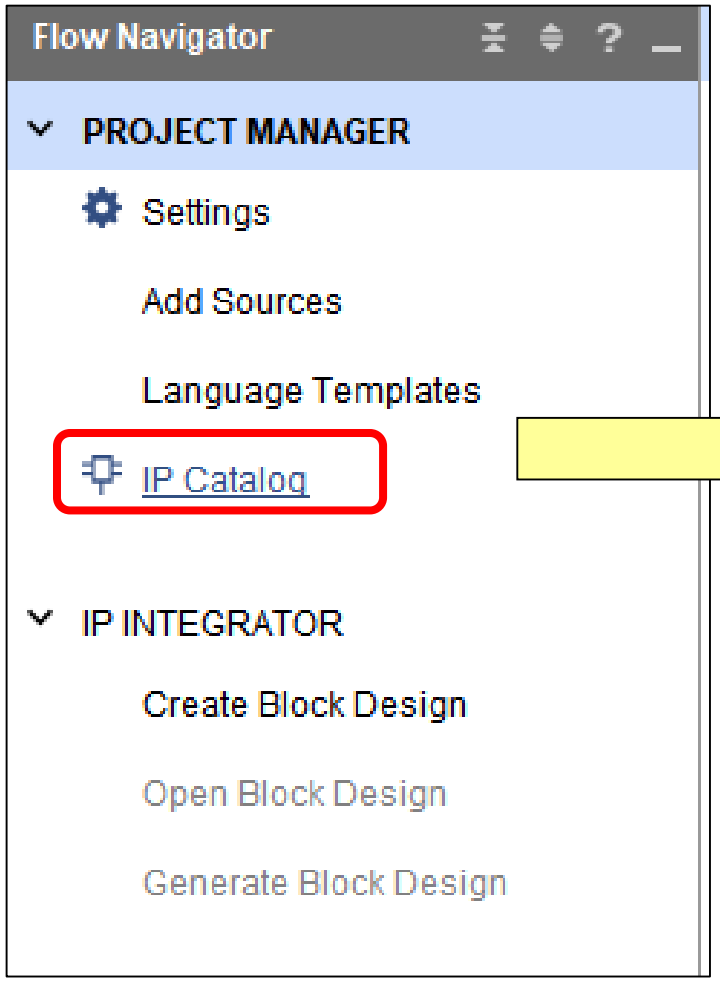
1. clk_out1 = 100MHz
2. clk_out2 = 50MHz
3. clk_out3 = 25MHz



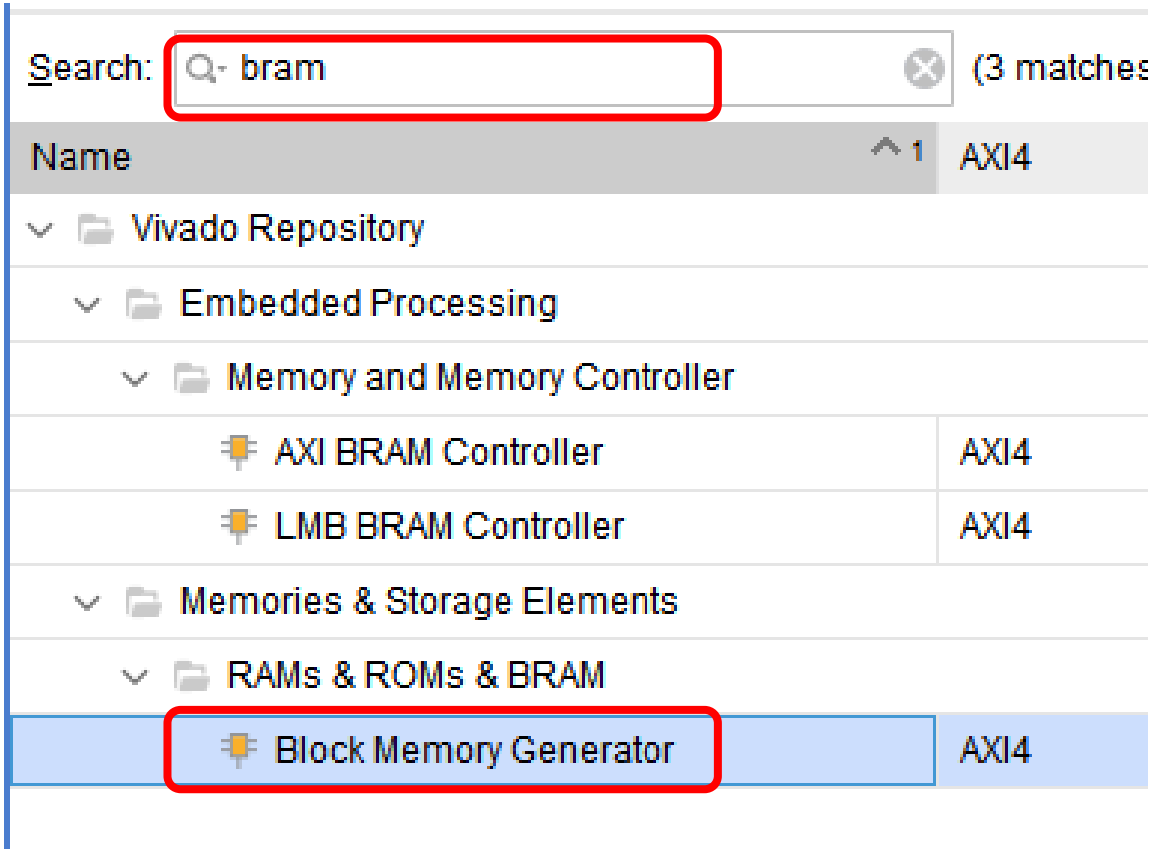
Generating Clocking Management Block



Adding Block RAM IP (Intellectual Property)

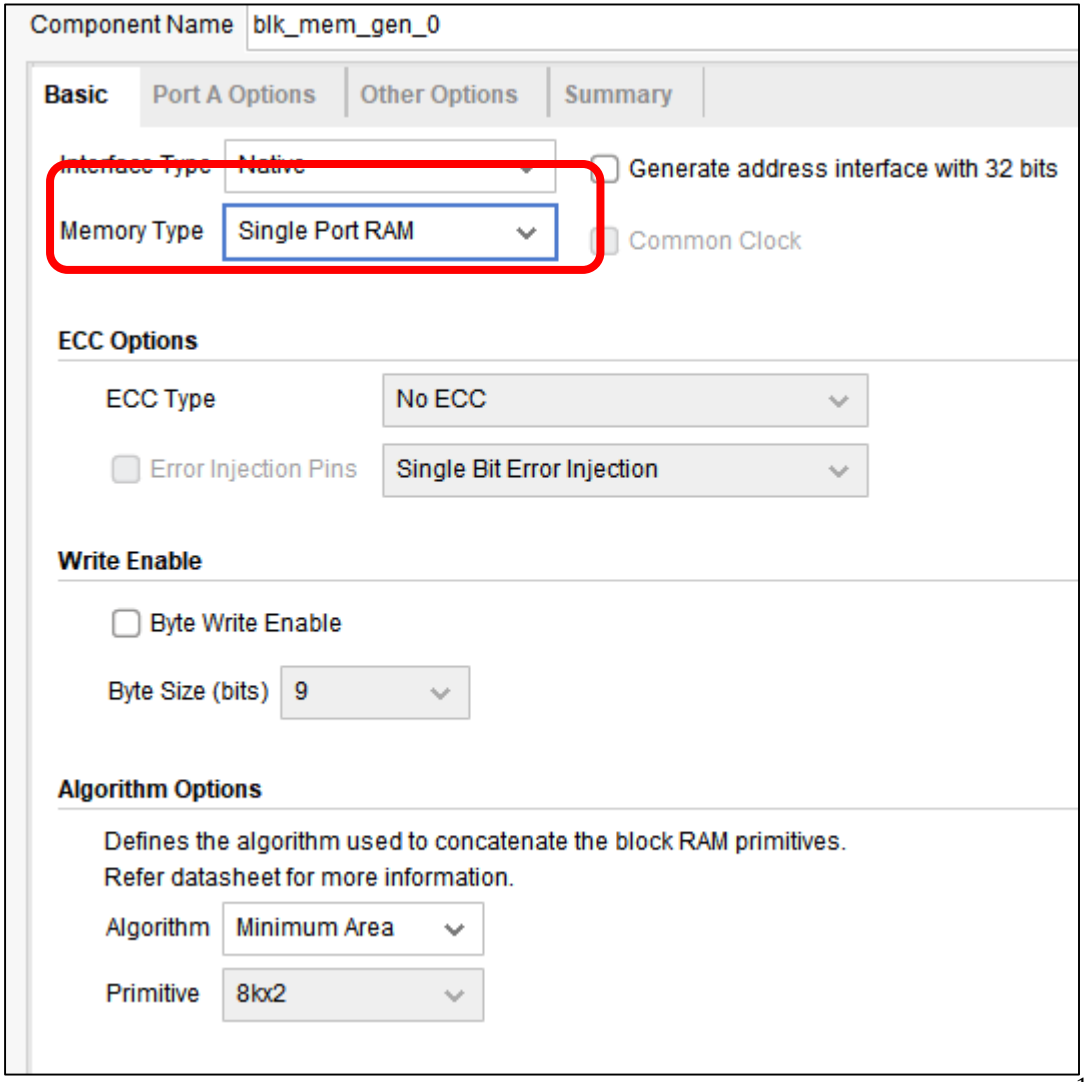


1. Project Manager → IP Catalog
2. Search with keyword “bram”
3. Choose Block Memory Generator



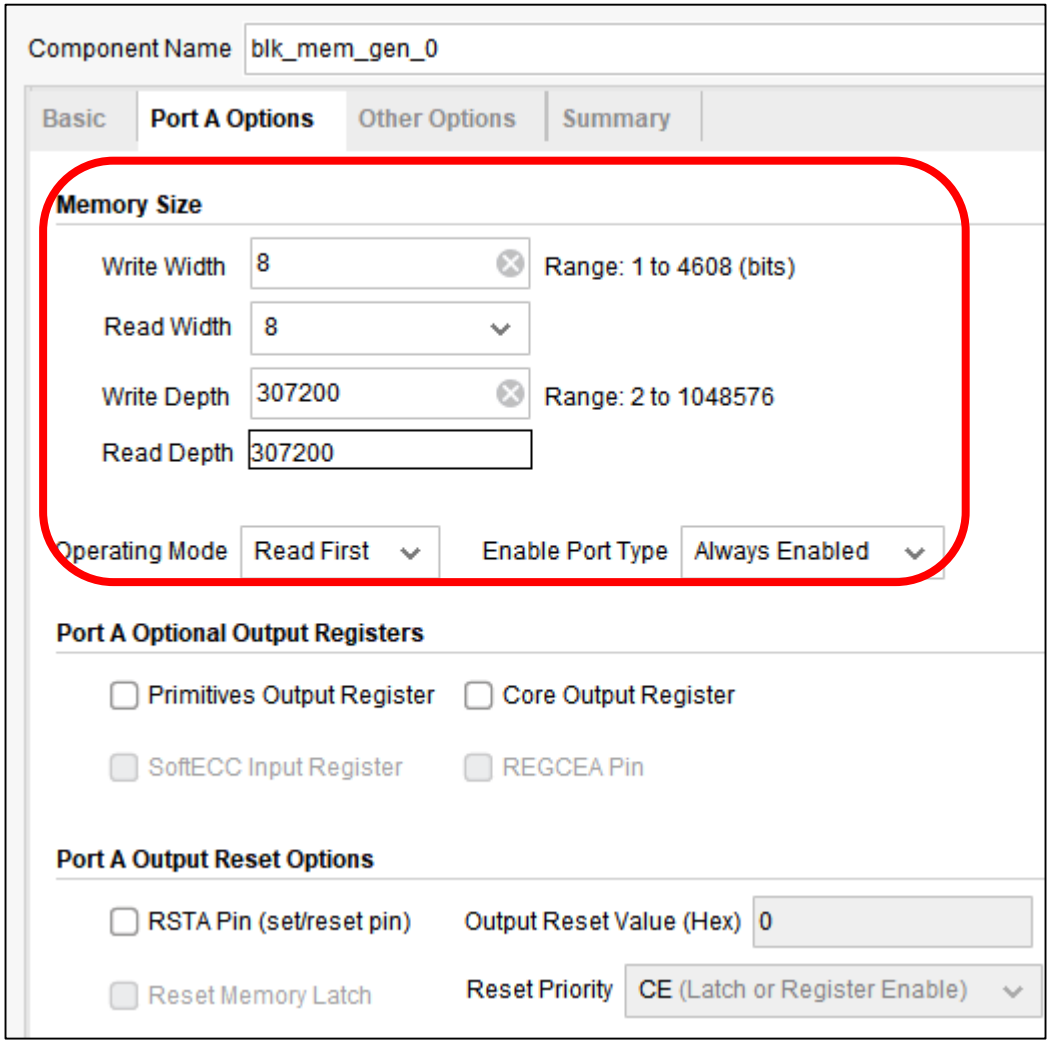
Select Memory Type

Choosing Memory Type **:**
Single Port RAM



Set Up Memory Size

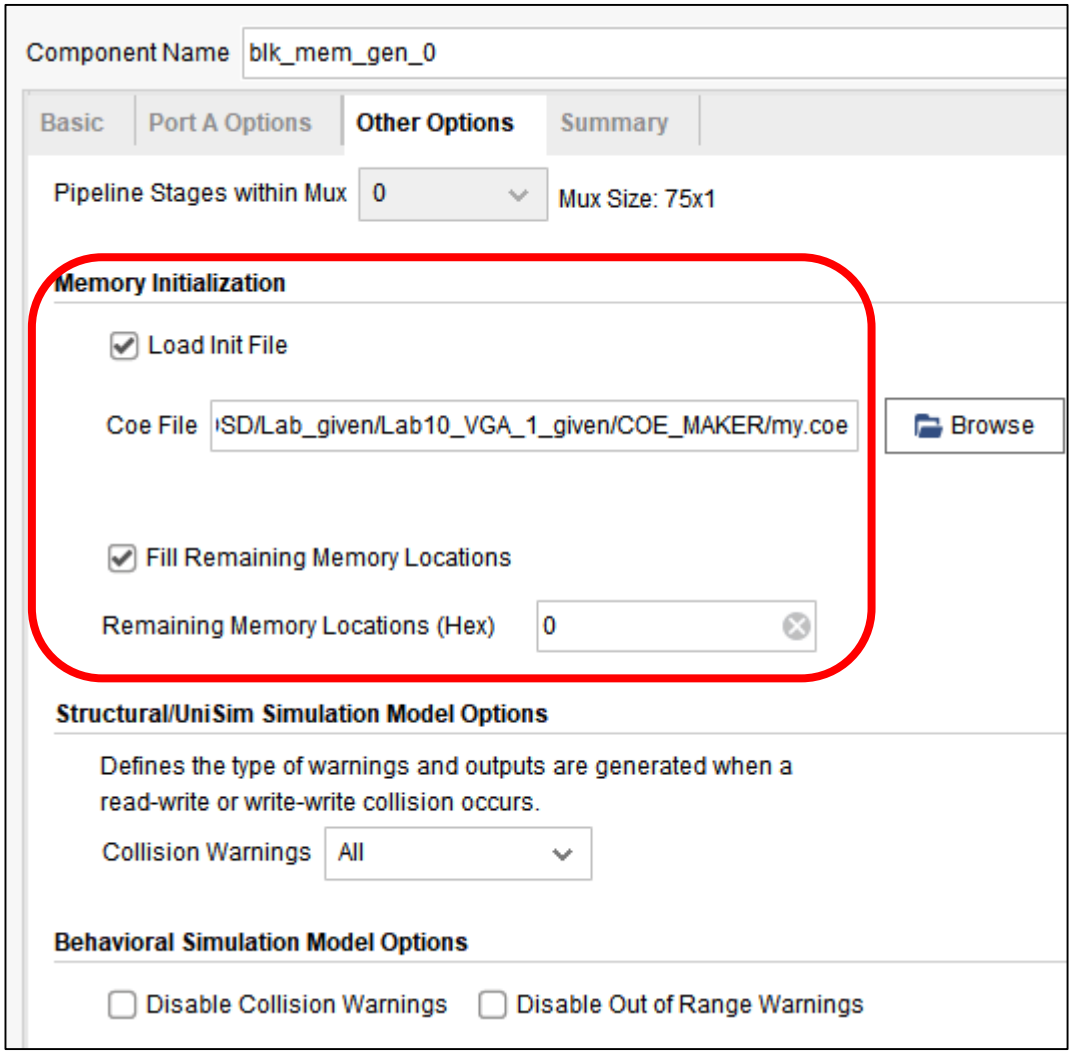
1. Set up Memory Size
   - Width:8
   - Depth: 640*480 (307200)
2. Set Up Operation Mode → Read-First



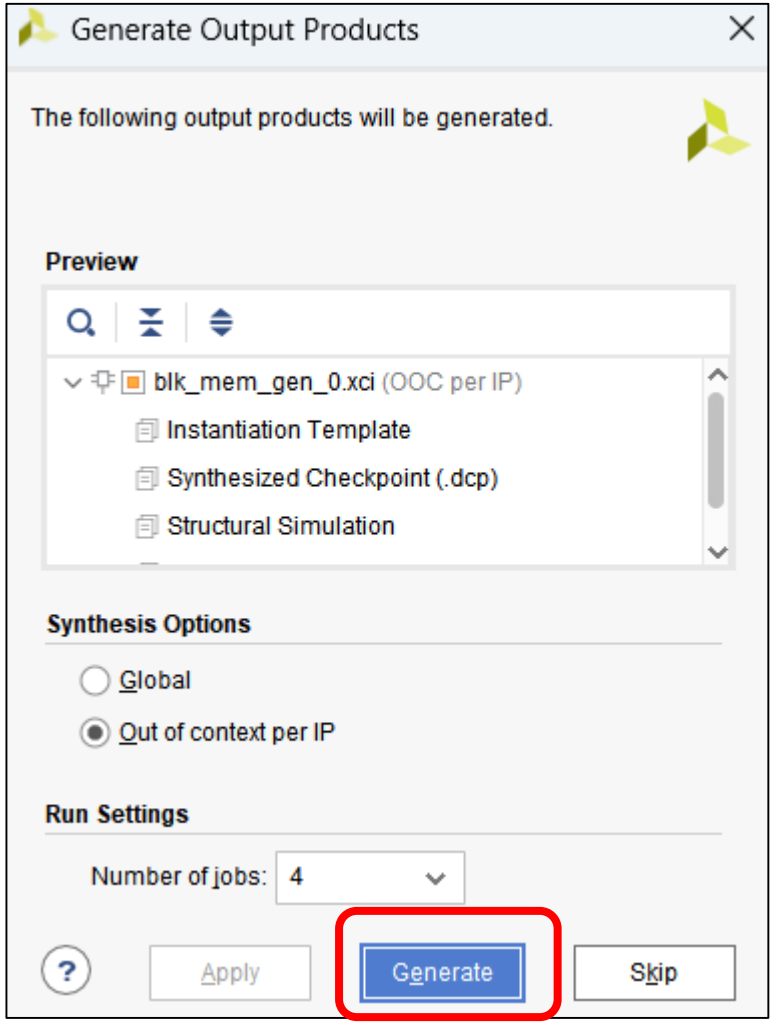
Set Up Memory Initial Contents

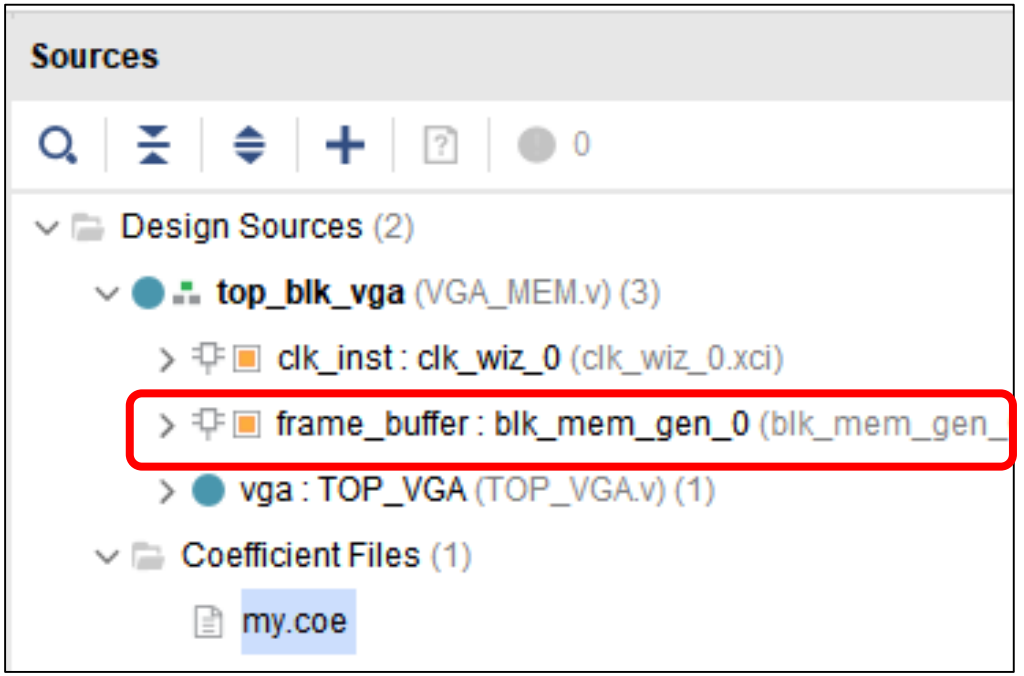
1. Set up Memory initial value using COE file

2. Fill Remaning Location with 0

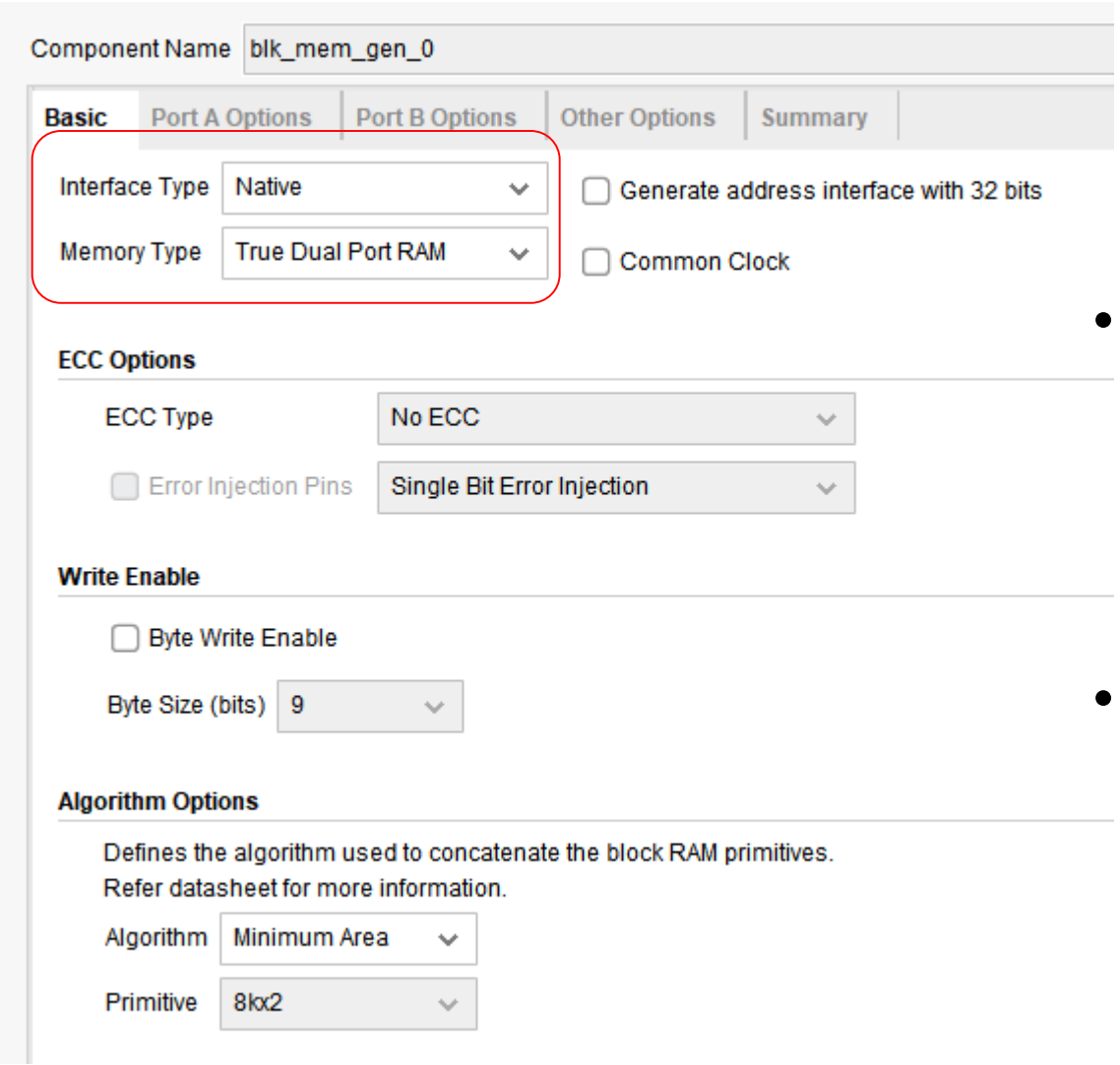


Successfully Added Memory Block IP



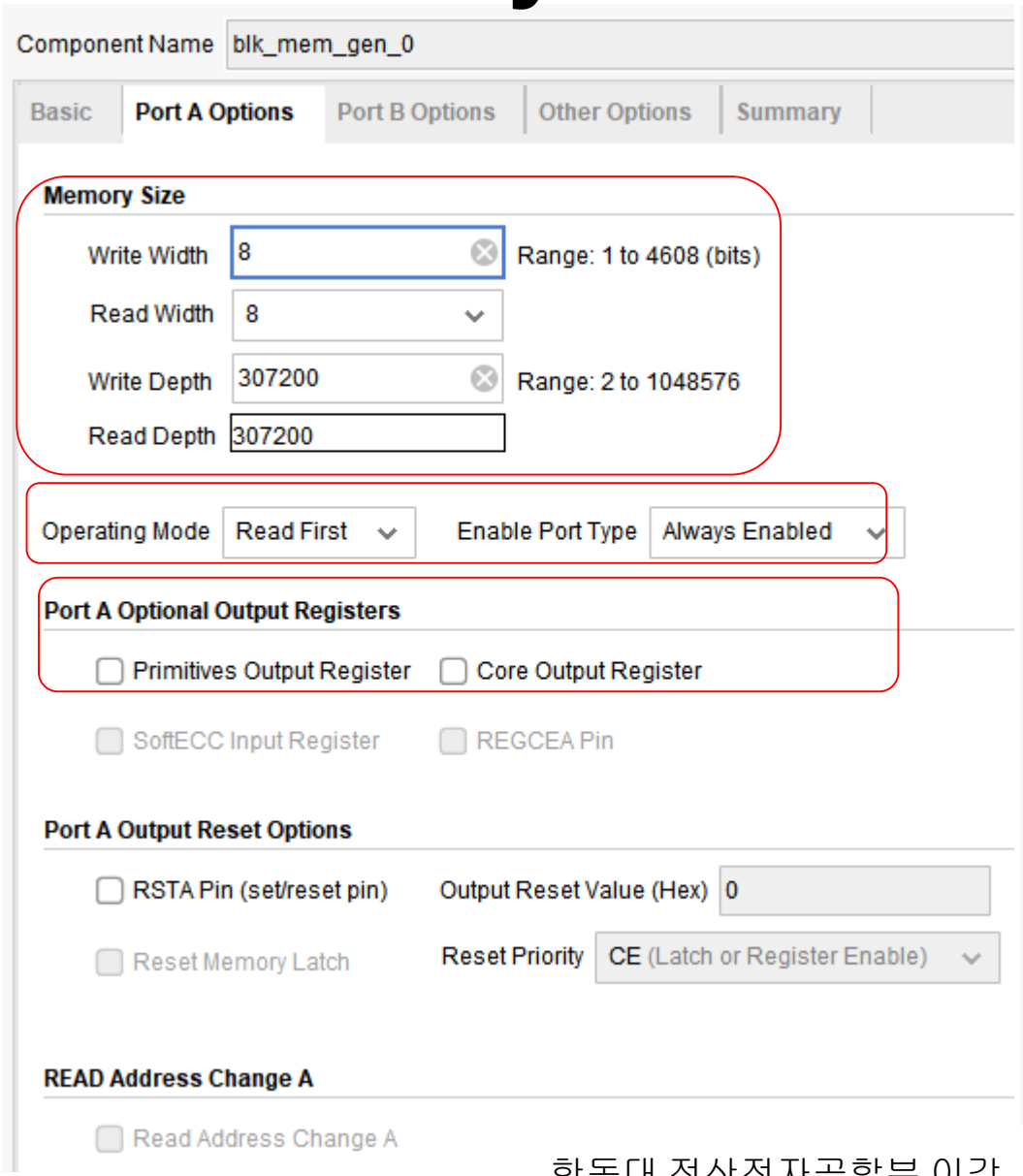


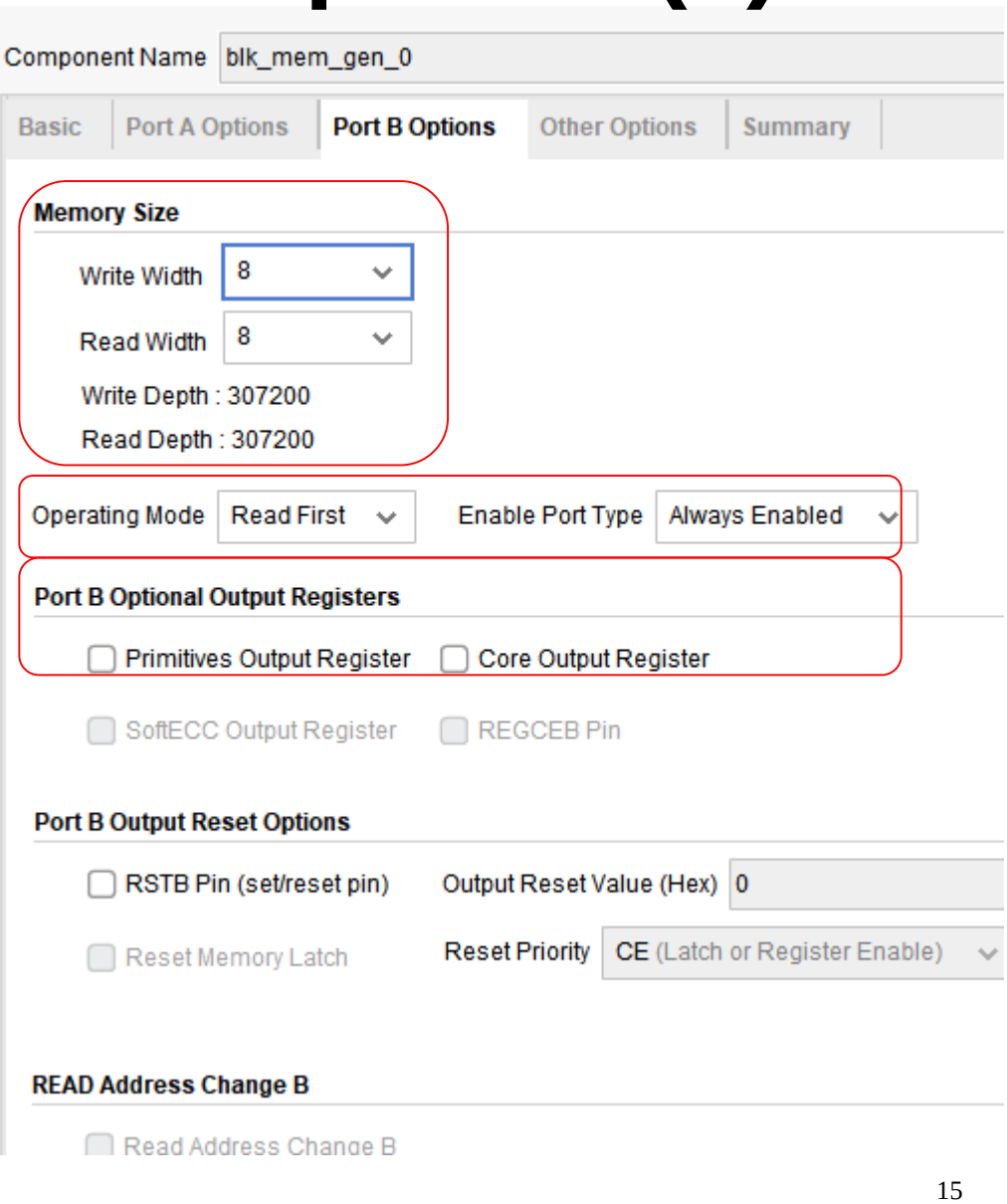
Memory IP Generator Options (1)



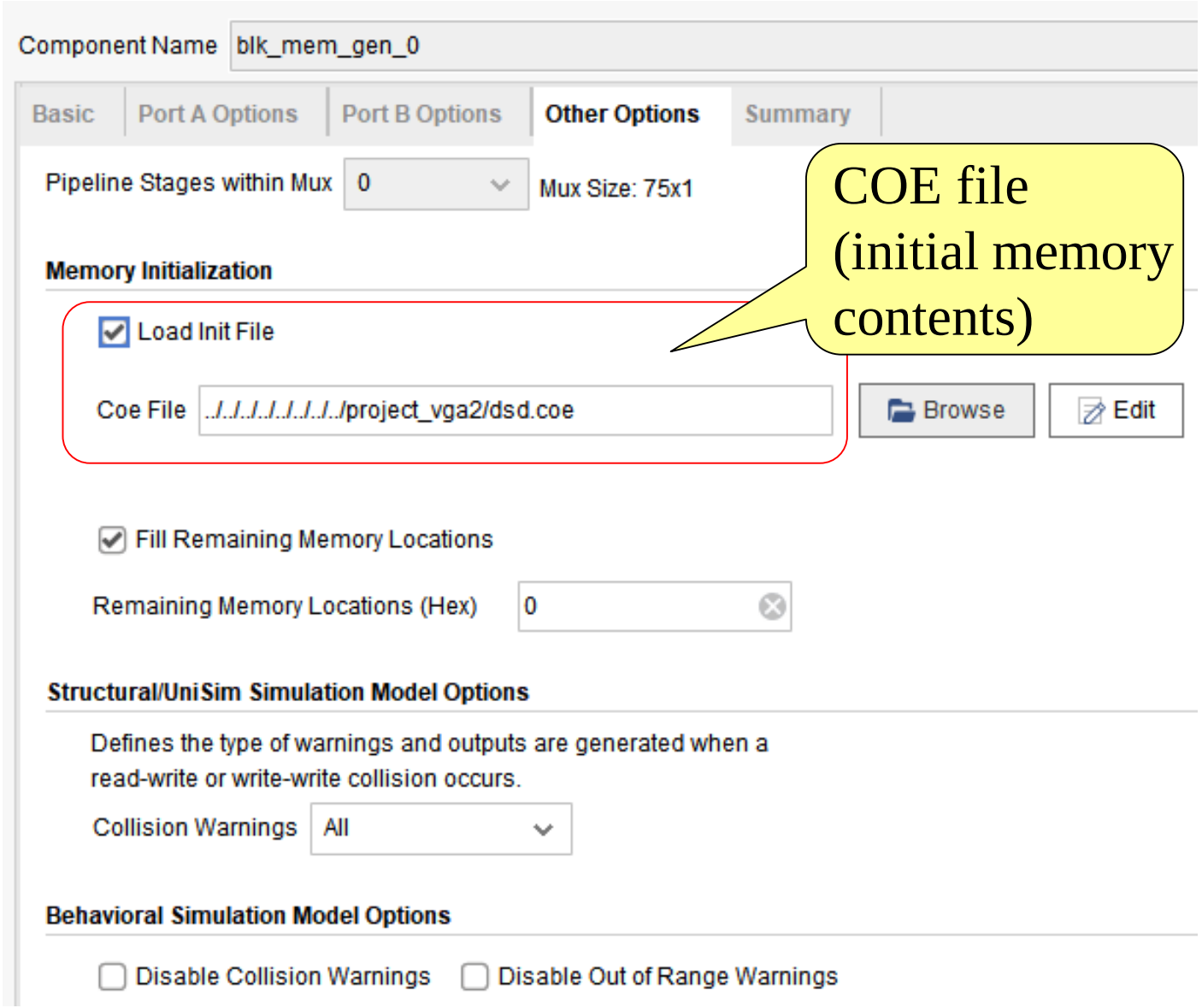
- True Dual Porr RAM
  - Independent clock
  - Simultaneous Access
  - Read-First
  - No Optional Register
- Size
  - Width: 8bit
  - Depth: 640x 480

Memory IP Generator Options (2)



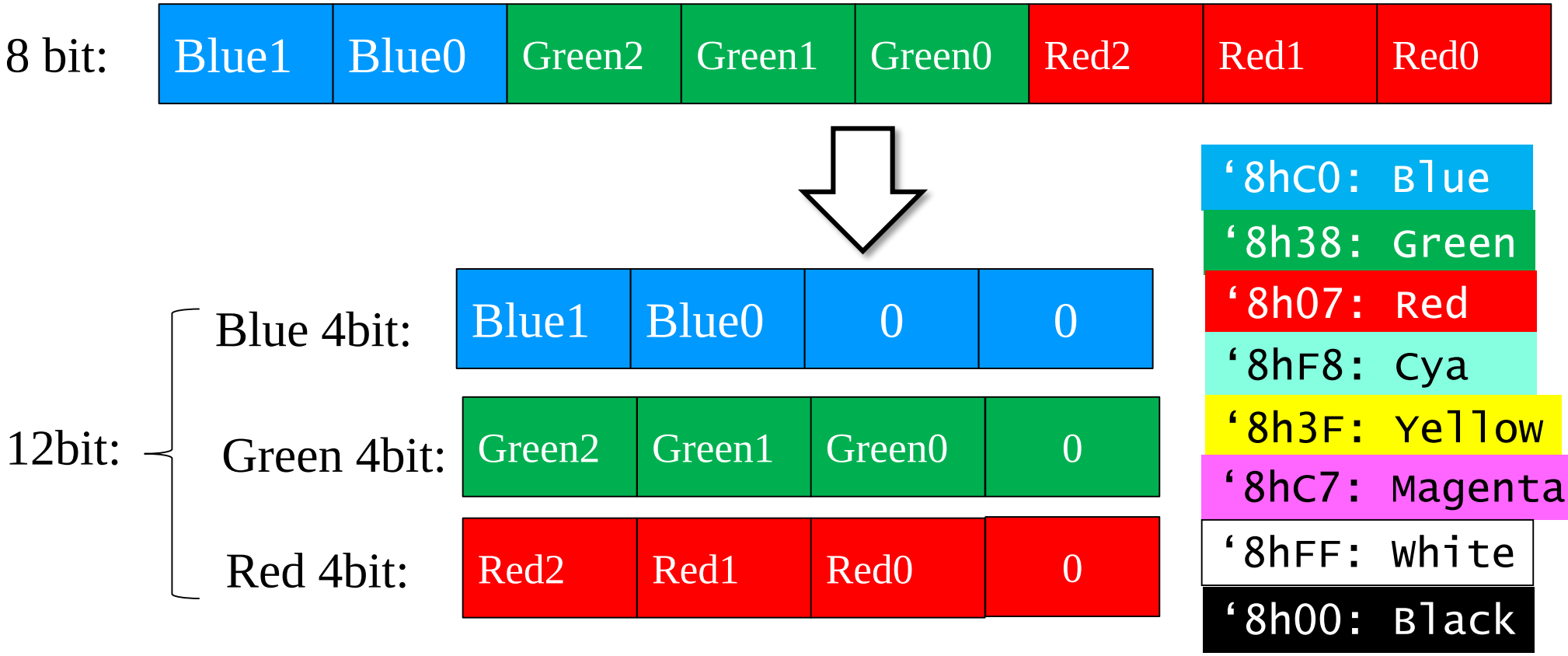


Memory IP Generator Options (3)



Pixel Color ReMapping

- 8 bit (in block memory) → 12 bit (VGA port) Mapping



COE File

- Text-based File to specify the Initialization Values of Memory Elements like Block RAM or LUT of Xilinx FPGAs
- Format:

```
memory_initialization_radix = <radix>;
memory_initialization_vector = <data0>, <data1>, ..., <dataN>;
```

  - radix can be 2, 10, or 16
- Example

```
memory_initialization_radix = 2;
memory_initialization_vector =
0000, 0011, 1001, 0100, 1110, 0110, 0111, 1010;
```

```
memory_initialization_radix = 16;
memory_initialization_vector = 0, 3, 5, 4, E, 6, 7,A;
```

VGA Parameters

assign Xoffset = {XY,6'd00};

assign Yoffset = {XY,5'd00};

localparam XRES = 640;

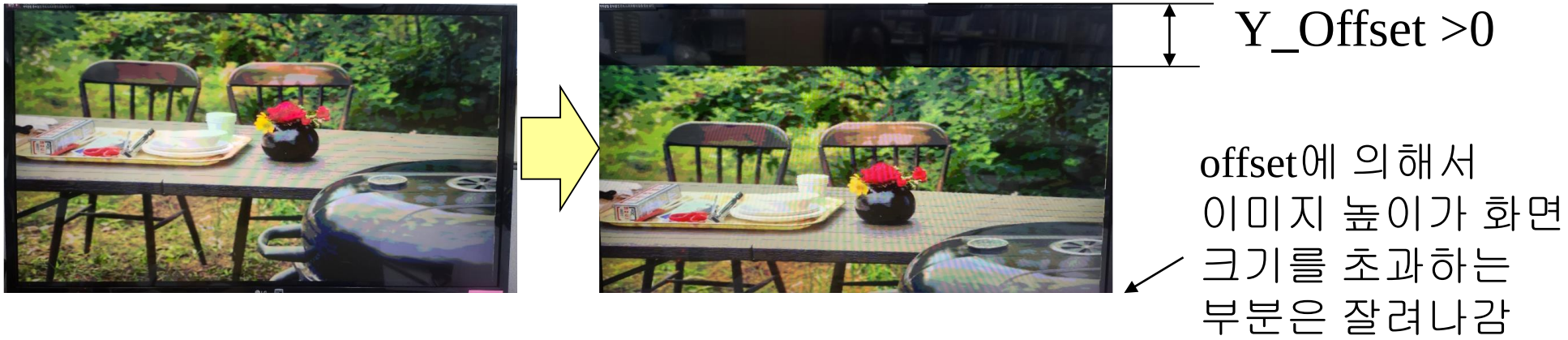
localparam YRES = 480;

(YRES)
Y resolution
= 480

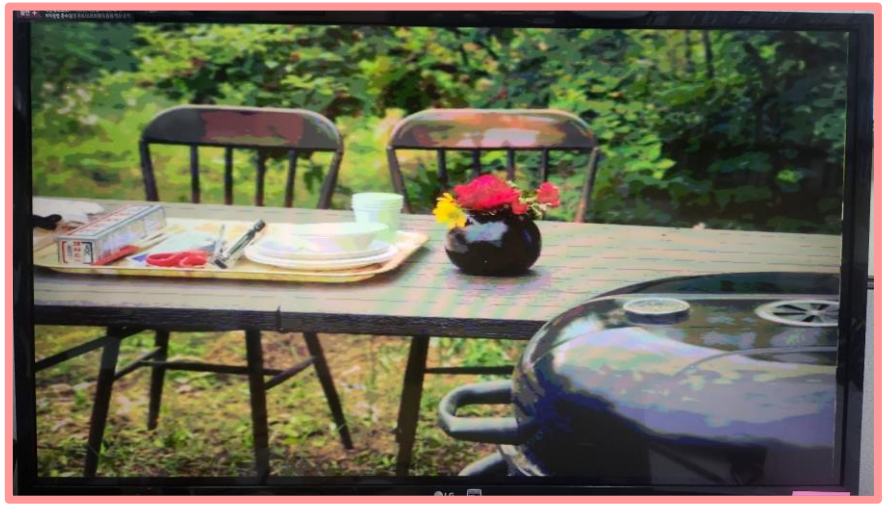
X resolution = 640 (XRES)

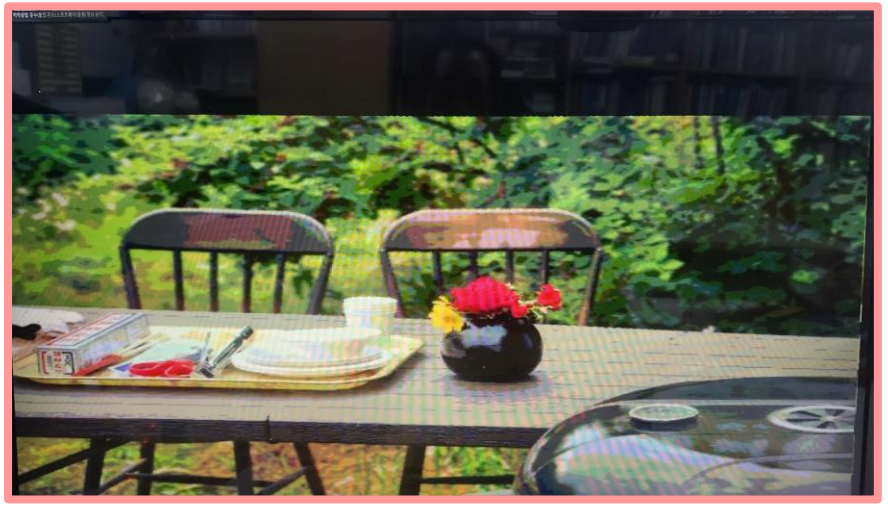
VGA with BRAM #1(1)

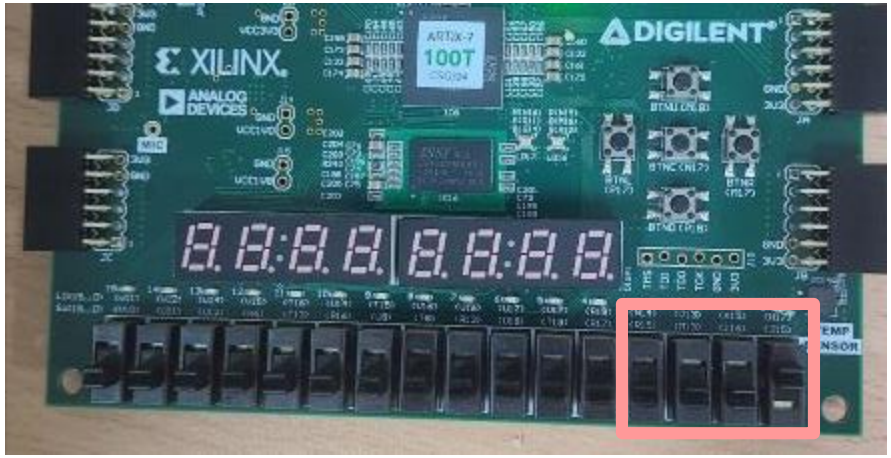
- 640x480 해상도의 이미지를 위한 COE 파일을 만들어 BRAM에 저장하고 BRAM의 값을 읽어 이미지를 출력하도록 설계된 VGA 회로를 FPGA를 타겟으로 합성하라.
- FPGA 보드의 Slide switch 4개를 이용해서 다음 기능을 추가하라.
  - SW[0] = 1 일때만 화면에 출력된다.
  - SW[3:1]의 값에 의해서 화면에 출력되는 이미지의 Y 방향 offset을 정함
    Yoffset = SW[3:1] * 32 pixel



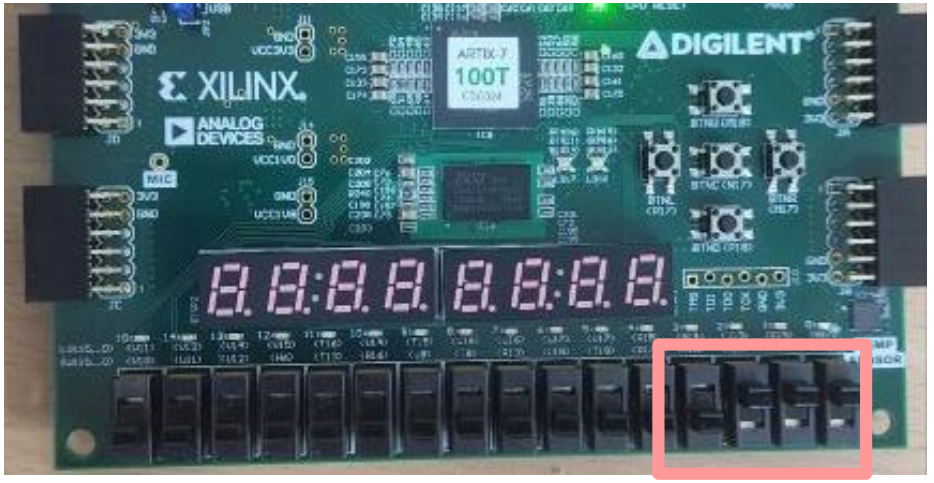
Lab #1 (2)







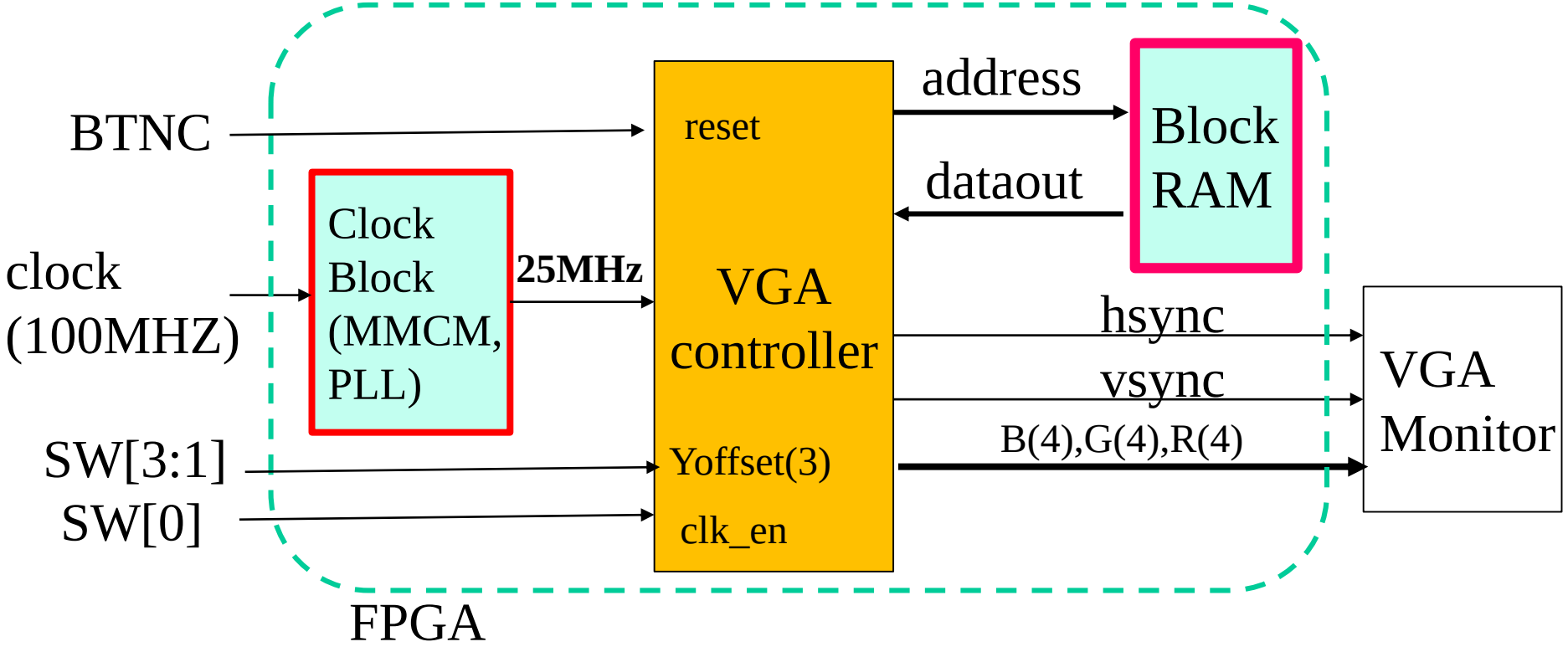
0 0 0 1



0 1 1 1

VGA with BRAM Lab #1 (3)

- VGA Enable : slide switch (SW[0])
- Y_Offset : slide switch (SW[3:1])
- Reset : push button (BTNC)



Lab #1(4): COE file Generation

Binary file

Text file

image file
(.jpg, .bmp)

COE_maker.exe

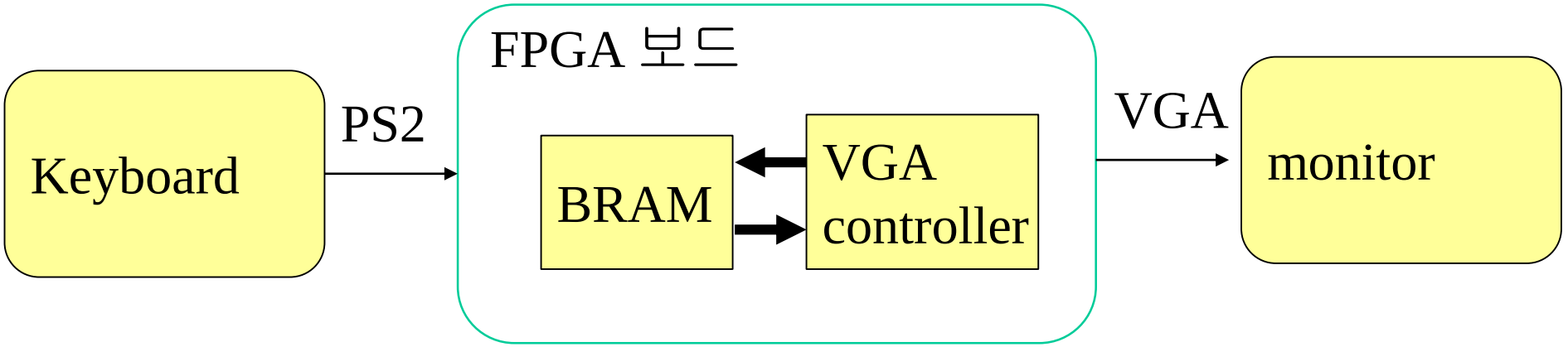
COE file
(.coe)

```
memory_initialization_radix=16;
memory_initialization_vector=
11,
11,
11,
11,
5A,
5A,
5A,
```

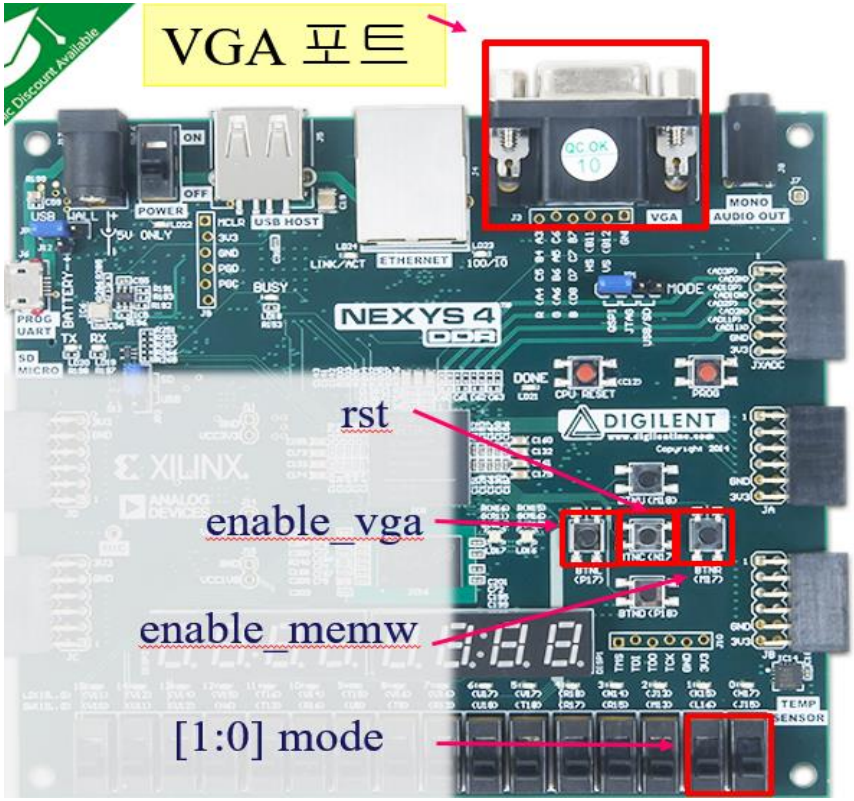
```
Type your file name. It should be the full name
ex) dsd.bmp
The IMAGE file should be in the directory of this EXE file
BMP format is recommanded


File : ./dsd.bmp

width : 640
height : 480

done
```

VGA with BRAM Lab #2

- 보드의 Slide SW대신 PS2 keyboard의 키를 이용해서 이미지를 위 또는 아래로 이동시키도록 하라. 즉,
- 예를 들어 3줄 아래에 출력 후 1줄 아래로 출력 하기 위해서
  - 보드 slide switc를 011로 설정 → 001로 설정하는 대신에,
  - 키보드의 + 키를 세번 입력 후 – 키를 2번 입력하도록
- 화면 해상도가 640 x 480보다 작은 이미지를 COE로 만들어 실험에 사용하라. (Verilog 소스코드의 XRES, YRES 조정 필요)



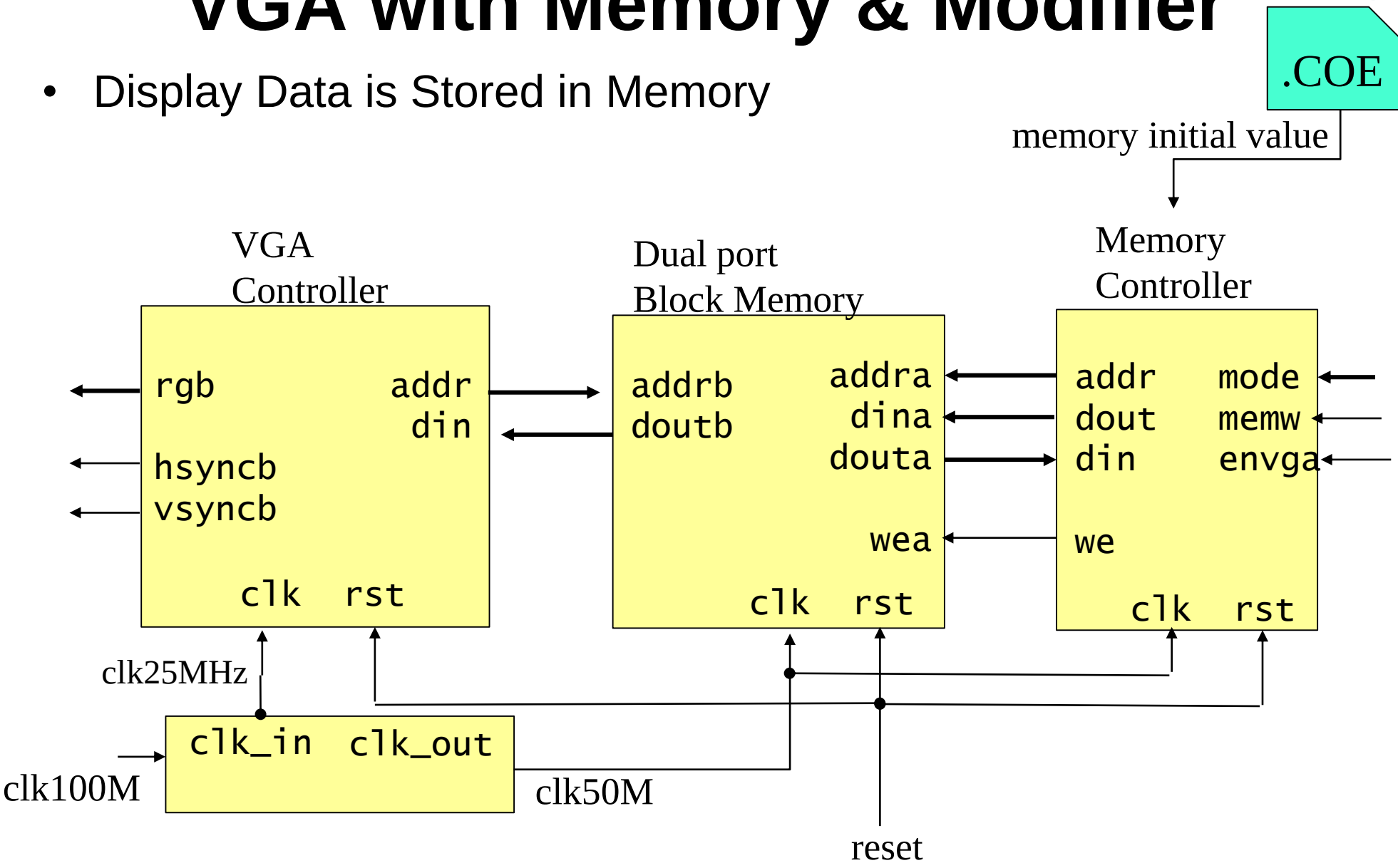
VGA with BRAM Lab #3



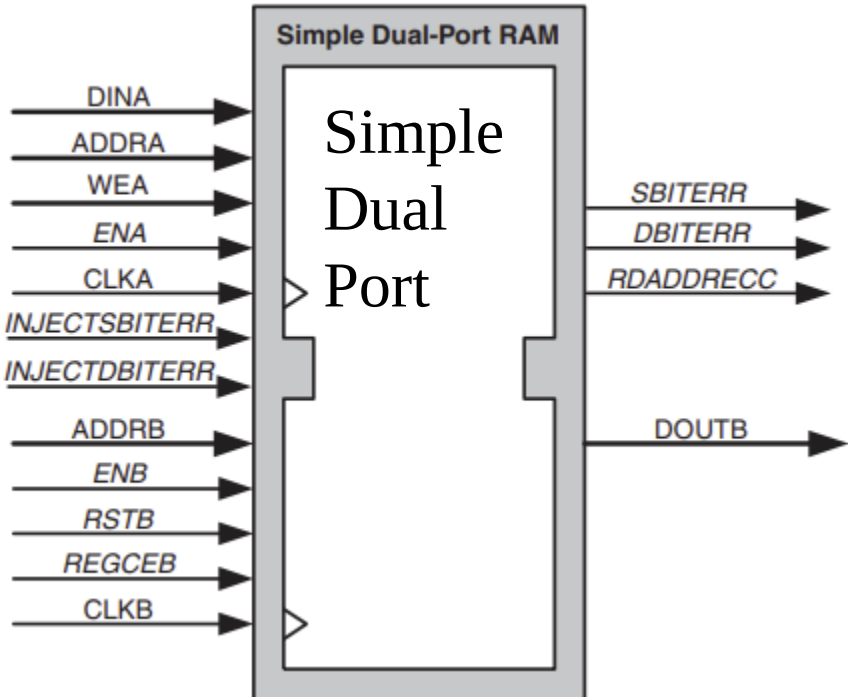
mode	기능	요약
2'b00	CLEAR	메모리를 녹색값(0x38)으로 초기화한다. 출력 화면이 녹색으로 변한다.
2'b01	SHIFTR	메모리를 한 칸 씩 RIght SHIFT. 출력 화면이 오른쪽으로 밀린다.
2'b10	RG_SHIFTL	메모리를 한 칸 씩 Left SHIFT 하면서 R성분과 G성분 화소값을 교환한다.
2'b11	V_COLOR_BAR	메모리에 폭 16픽셀 폭의 수직바 패턴을 넣는다. 화면에 각기 다른 컬러의 수직선 40개 무늬가 출력된다..

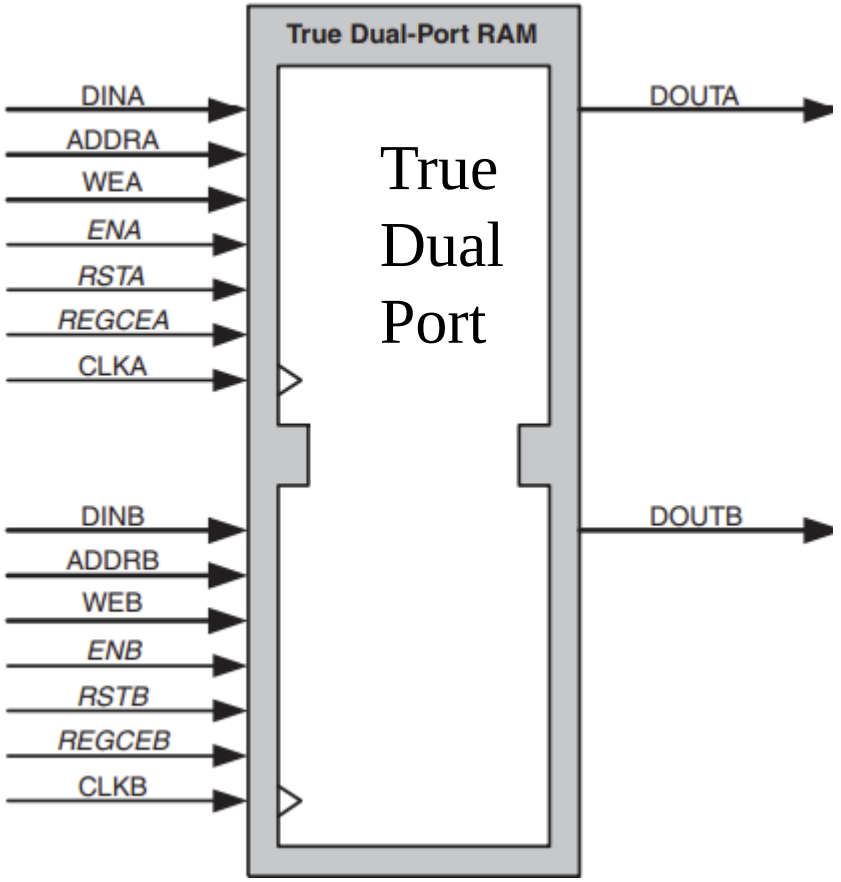
VGA with Memory & Modifier

- Display Data is Stored in Memory



Block RAM Types: Dual Ports





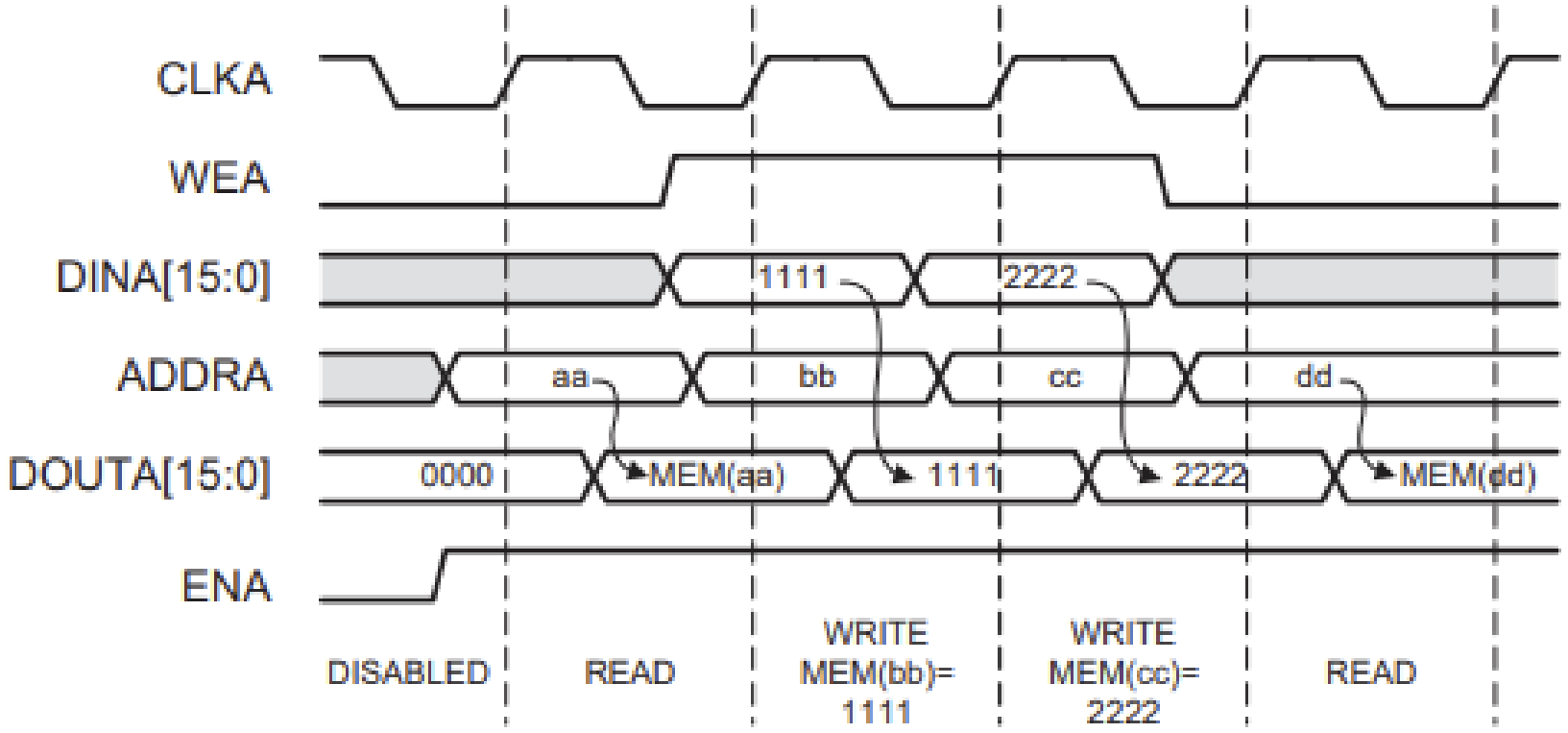
Port A: Write Operation
Port B: Read Operation

Port A: Read and Write Operation
Port B: Read and Write Operation

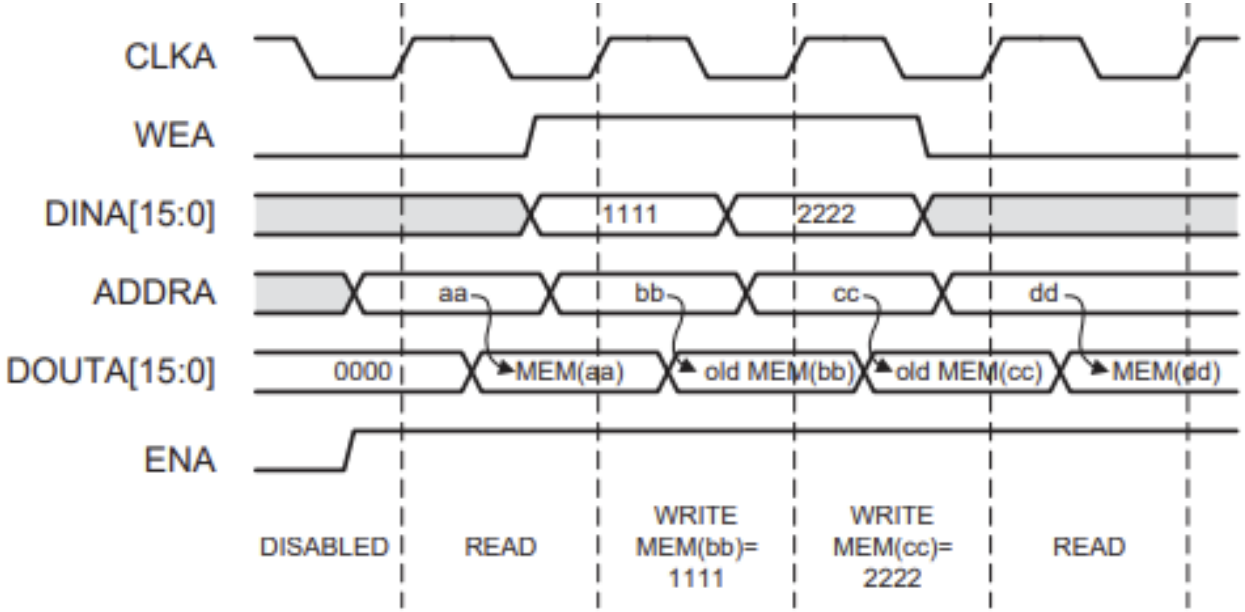
Block RAM Operation Modes

- Port A and Port B can be configured independently in one of three modes
  - Write-First
  - Read-First
  - No-Change
- The Options matters when Address of Port A and Port B Collides.

- Write-First Mode (Transparent mode): the input data is simultaneously written into memory and driven on the data output



- Read-First Mode(Read before write mode): data previously stored at the Write address appears on the data output, while the input data is being stored in memory

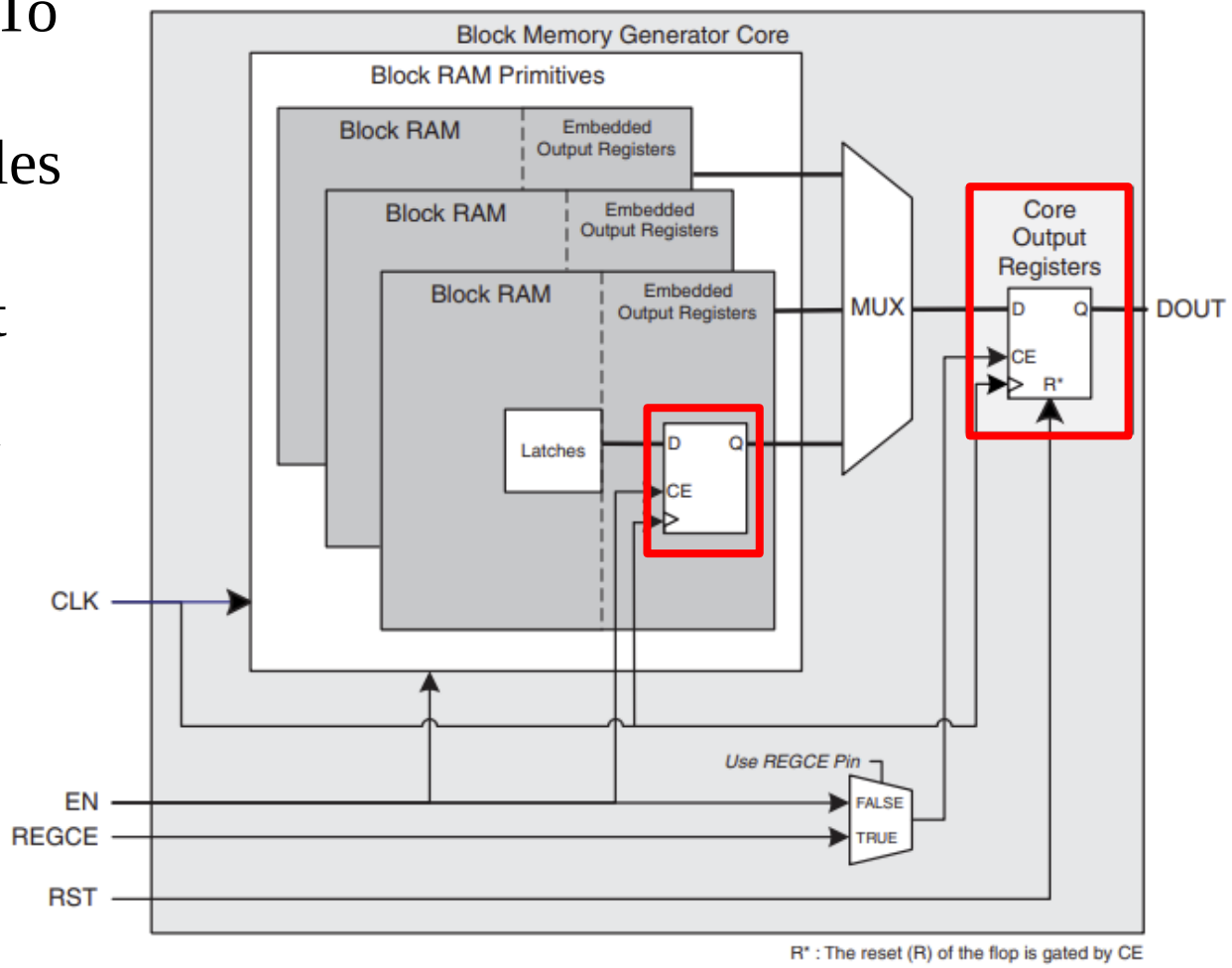


- No-Change: the output latches remain unchanged during a Write operation

Optional Output Registers

- Adding Output Registers: To Improve performance
- But Increases in clock cycles for Read Operation
- Optional Register stages at

1. the **Output of the Block RAM Primitives** ***or/and***
2. the **Output of the Core**



READ and Read Enable Latency

output data

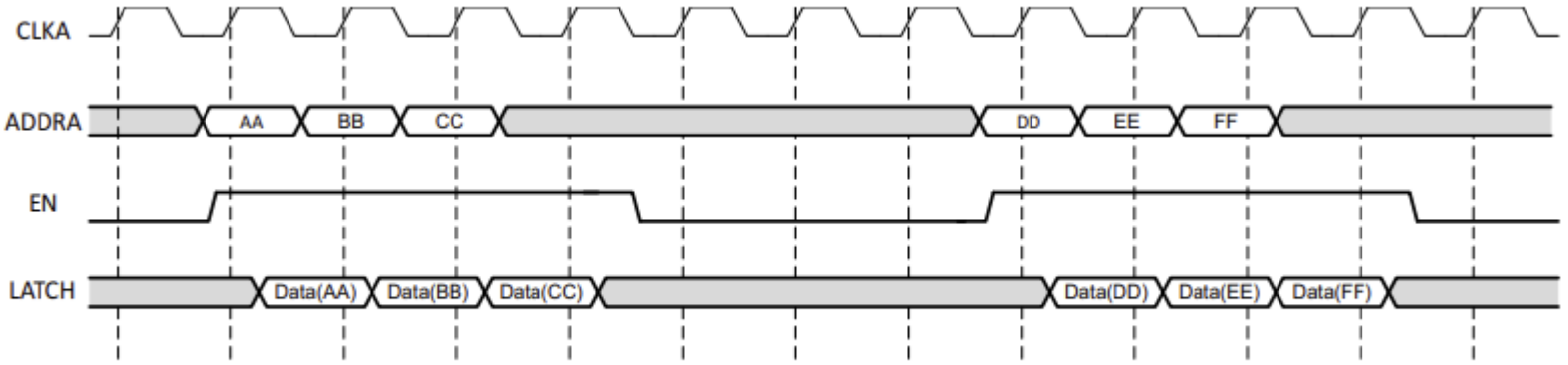


Figure 3-21: **Read Data and Read Enable Latency with no Output Registers**

output data

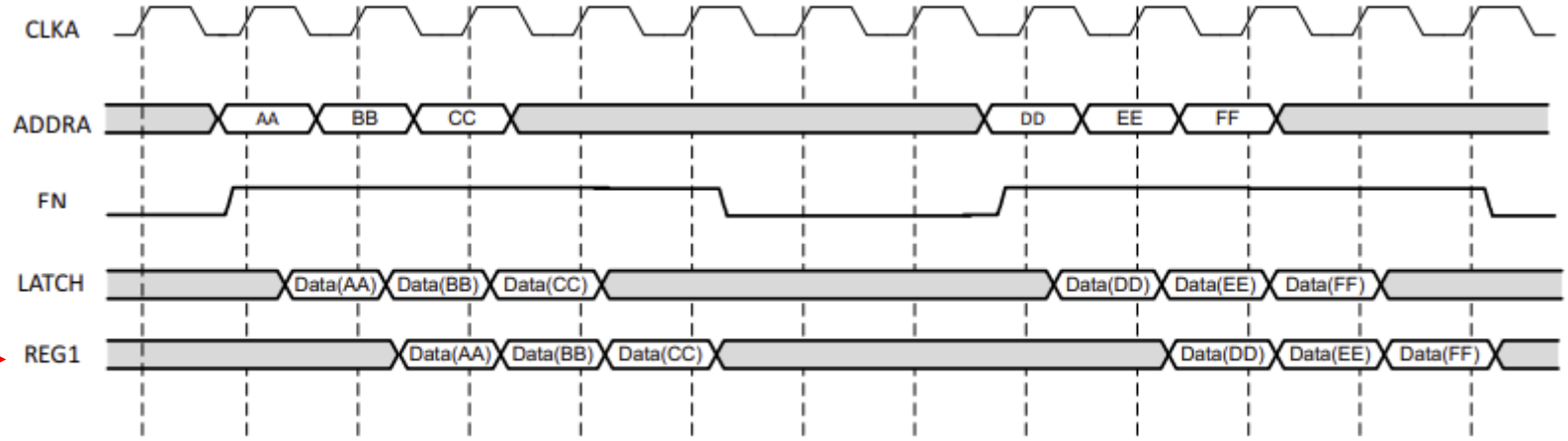
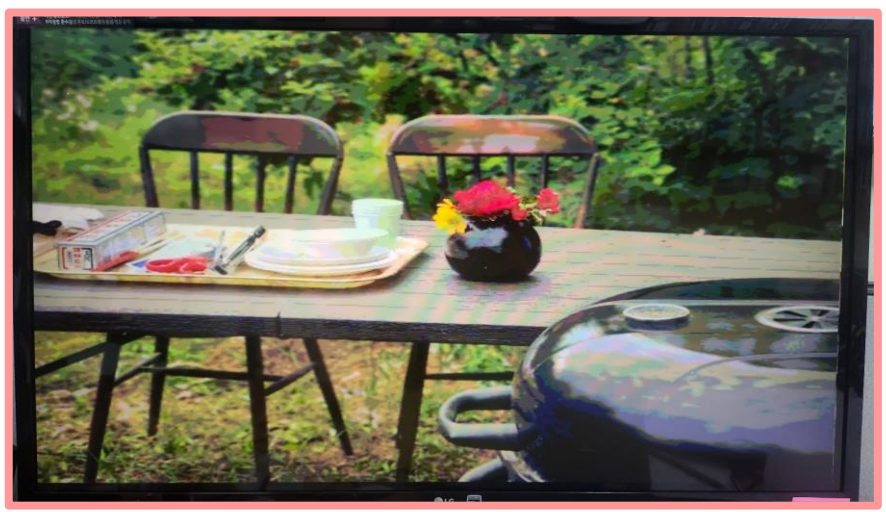


Figure 3-22: **Read Data and Read Enable Latency with Primitive Output Registers**

VGA with BRAM Lab #4

- 640 x 480 해상도의 서로 다른 이미지 2장을 메모리에 저장하고
- 저장된 2장의 이미지가 Slide Swich 조작으로 선택적으로 모니터 화면에 출력되도록 하라.



SW[0] = 1

SW[0] = 0