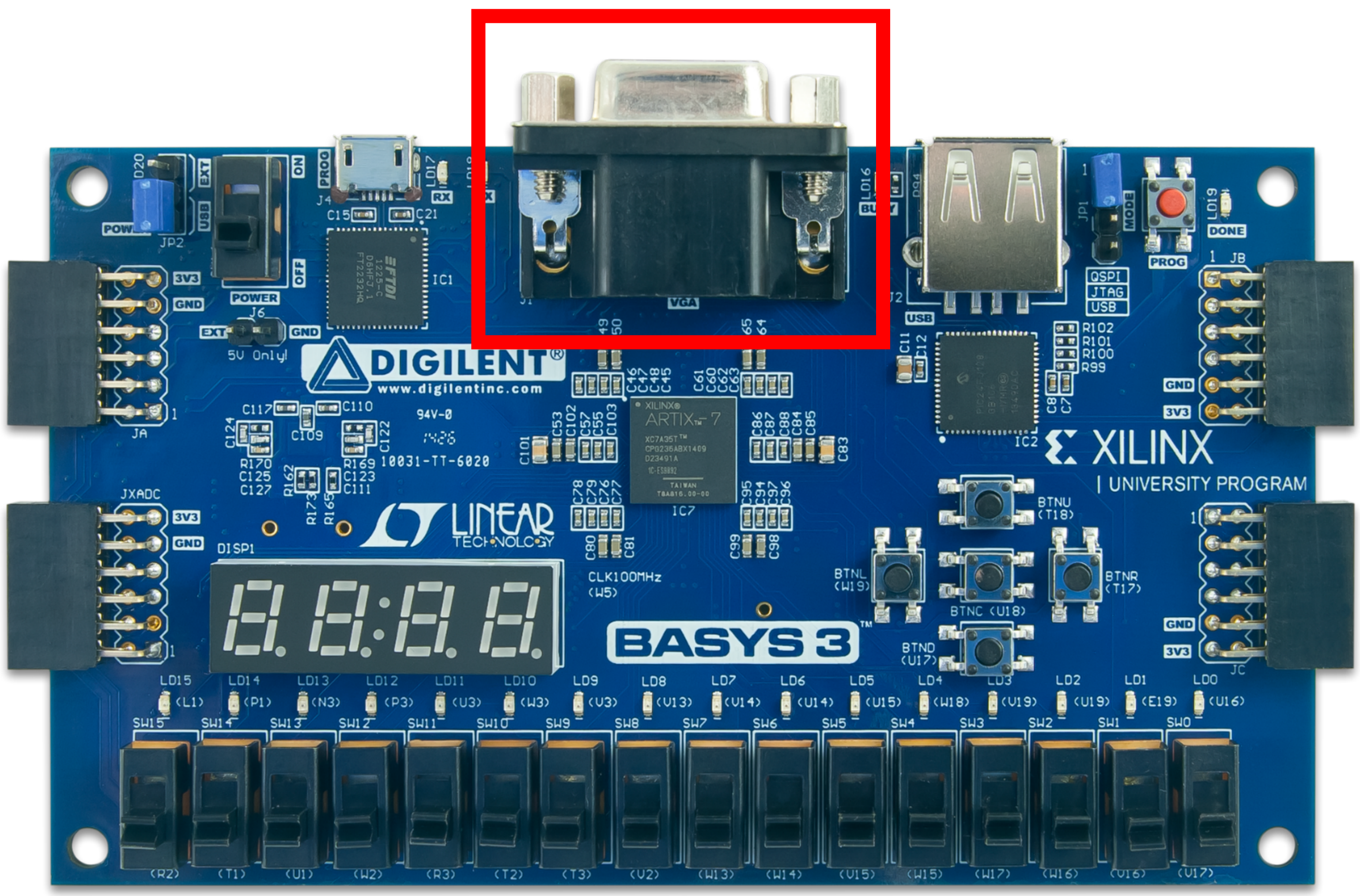
VGA Display

Hsi-Pin Ma

https://eeclass.nthu.edu.tw/course/18498
Department of Electrical Engineering
National Tsing Hua University

VGA Port

VGA


DIGILENT
www.digilentinc.com
XILINX
UNIVERSITY PROGRAM
BASYS 3

VGA

- VGA = Video Graphics Array
- Introduced by IBM in 1987, and still used today
- Transmitting analog signal

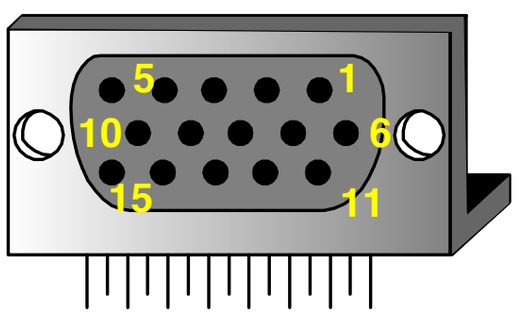
Cathode-Ray Tube Monitor

Video Graphics Array
DE-15 female and male connector

LCD Monitor

VGA Video Signal

- A VGA video signal contains 5 active signals (RGBHV)
  - horizontal sync (HS): used for video synchronization in the horizontal direction
  - vertical sync (VS): used for video synchronization in the vertical direction
  - red (R): used to control the red color, 0v (fully off) ~ 0.7v (fully on)
  - green (G): used to control the green color, 0v (fully off) ~ 0.7v (fully on)
  - blue (B): used to control the blue color, 0v (fully off) ~ 0.7v (fully on)

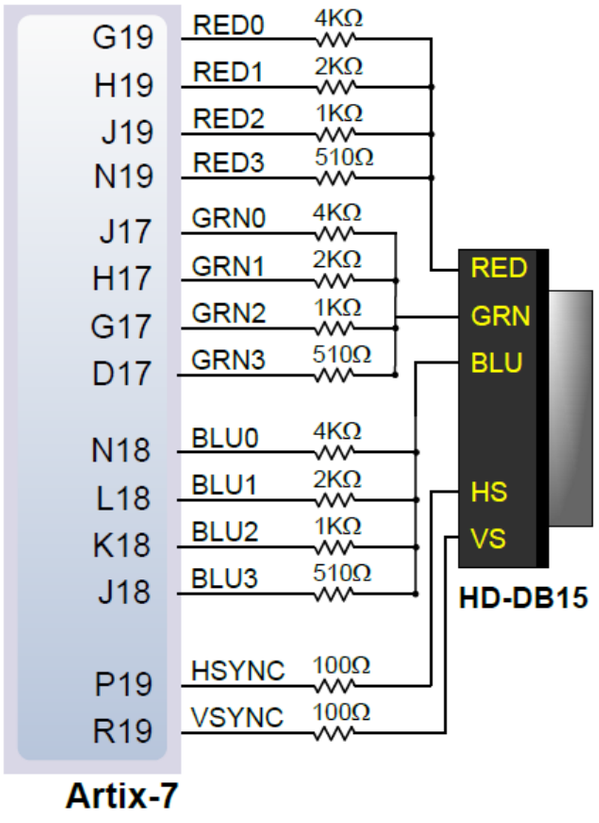


Pin 1: Red	Pin 5: GND
Pin 2: Grn	Pin 6: Red GND
Pin 3: Blue	Pin 7: Grn GND
Pin 13: HS	Pin 8: Blu GND
Pin 14: VS	Pin 10: Sync GND

Basys 3 Control Signals for VGA

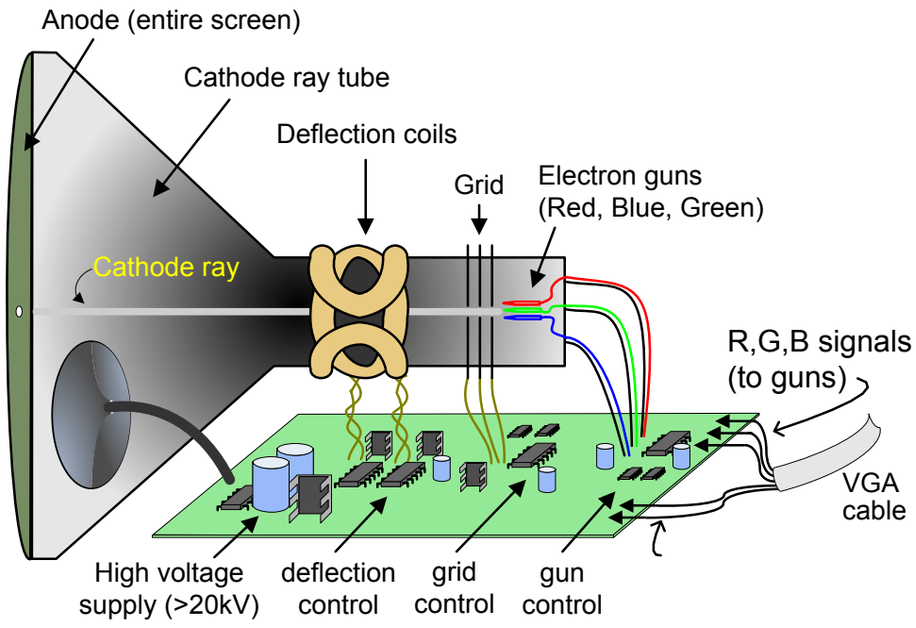
- 14 FPGA pins
  - 4-bits per color (R, G, B)
  - 2 standard sync signals (HS, VS)

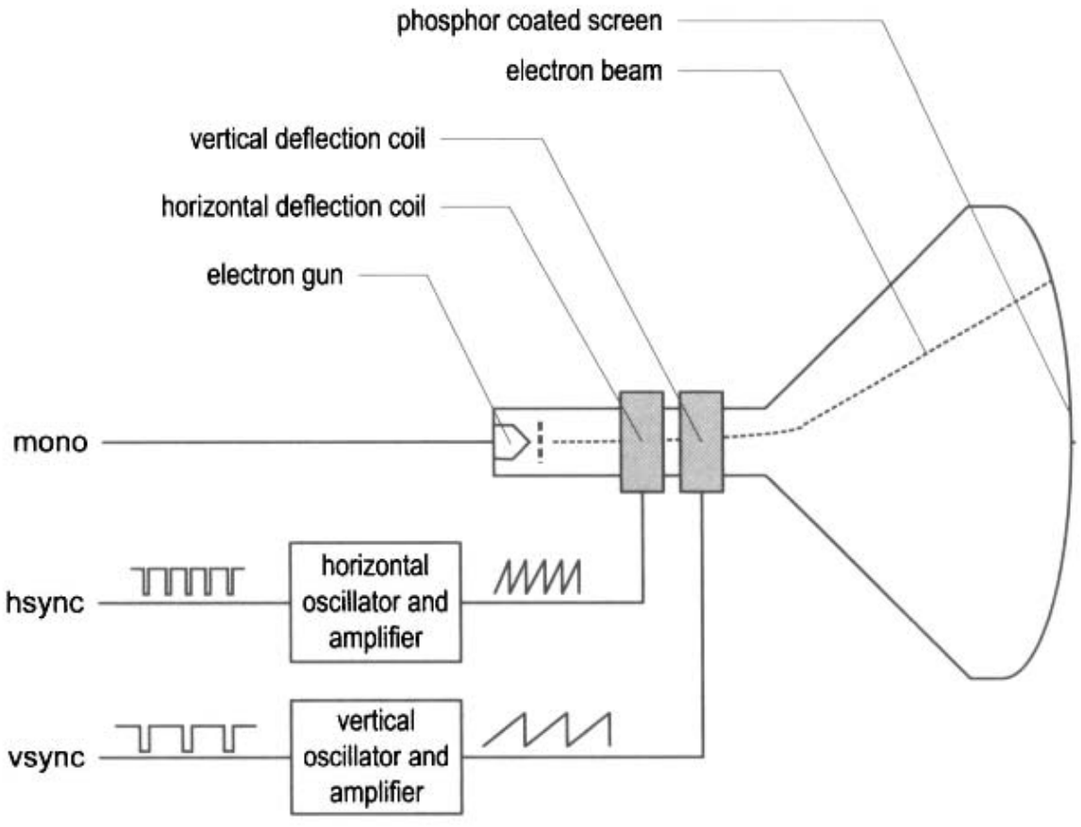
```
##VGA Connector
set_property PACKAGE_PIN G19 [get_ports {vgaRed[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[0]}]
set_property PACKAGE_PIN H19 [get_ports {vgaRed[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[1]}]
set_property PACKAGE_PIN J19 [get_ports {vgaRed[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[2]}]
set_property PACKAGE_PIN N19 [get_ports {vgaRed[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[3]}]
set_property PACKAGE_PIN N18 [get_ports {vgaBlue[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[0]}]
set_property PACKAGE_PIN L18 [get_ports {vgaBlue[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[1]}]
set_property PACKAGE_PIN K18 [get_ports {vgaBlue[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[2]}]
set_property PACKAGE_PIN J18 [get_ports {vgaBlue[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[3]}]
set_property PACKAGE_PIN J17 [get_ports {vgaGreen[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[0]}]
set_property PACKAGE_PIN H17 [get_ports {vgaGreen[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[1]}]
set_property PACKAGE_PIN G17 [get_ports {vgaGreen[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[2]}]
set_property PACKAGE_PIN D17 [get_ports {vgaGreen[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[3]}]
set_property PACKAGE_PIN P19 [get_ports hsync]
set_property IOSTANDARD LVCMOS33 [get_ports hsync]
set_property PACKAGE_PIN R19 [get_ports vsync]
set_property IOSTANDARD LVCMOS33 [get_ports vsync]
```

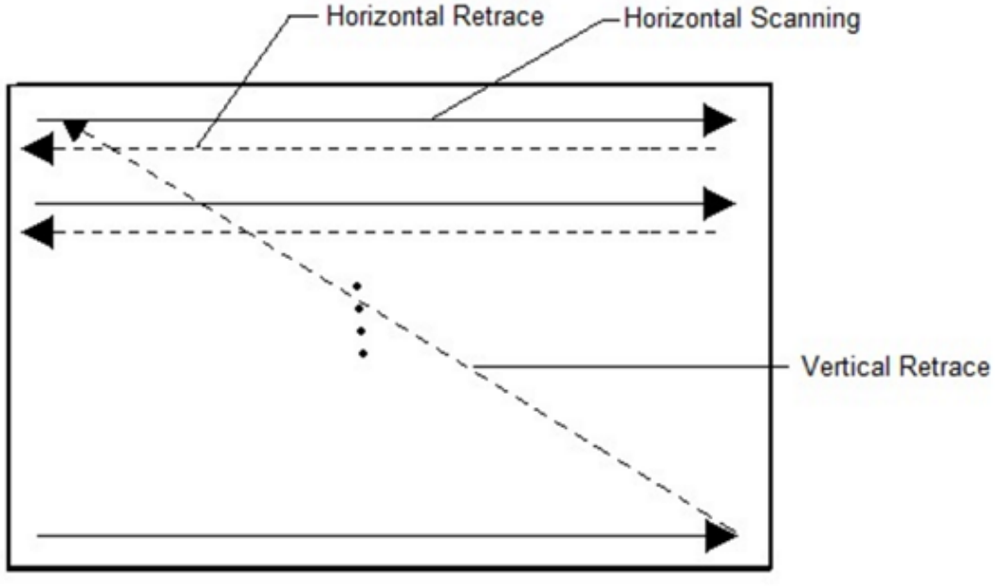


Basic Operation of CRT

- Cathode Ray Tube (CRT) is a vacuum tube containing one or more electron guns, and a phosphorescent screen is used to view images.







RGB Bitmap

- A digital color image is composed by a lot of pixels.
- Each pixel contains three R, G, B values to represent the intensity of these three primary colors.

RGB Color Mode

- Three primary colors
  - **Red**
  - **Green**
  - **Blue**
- A video signal of an N-bit word, can be converted to 2^N analog levels. The video signals can generate different 2^{3N} colors. (N=4 in Basys 3 board)

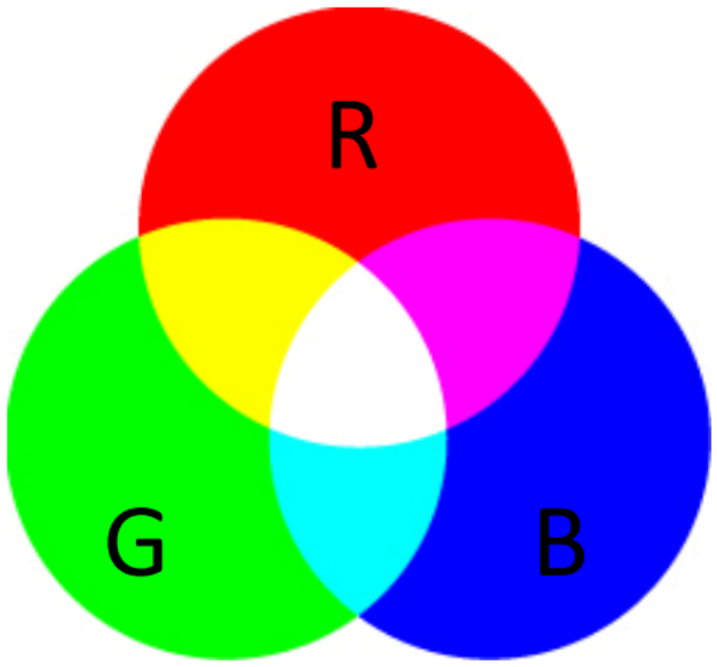
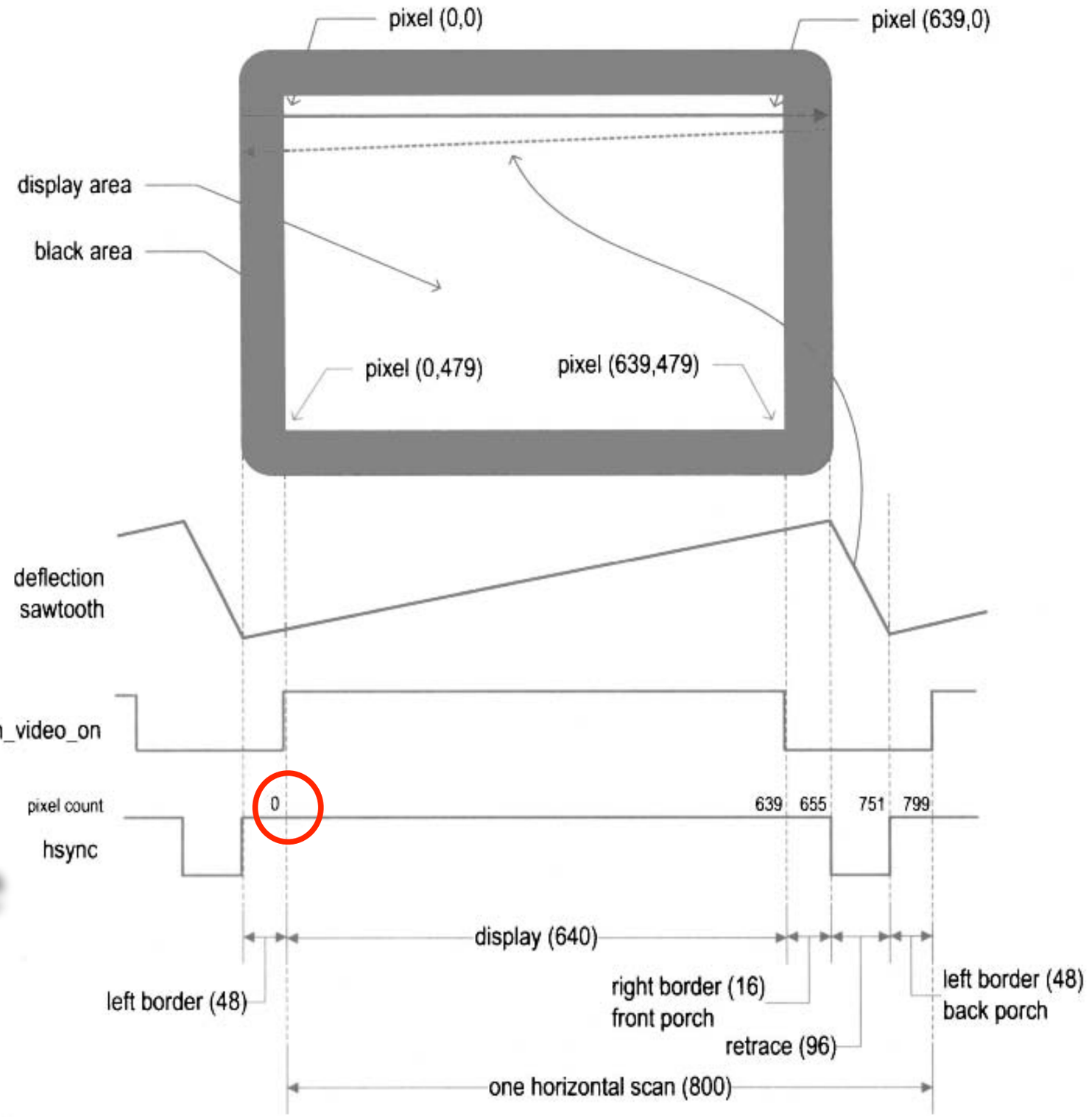


Table 13.1 Three-bit VGA color combinations N=1

Red (R)	Green (G)	Blue (B)	Resulting color
0	0	0	black
0	0	1	blue
0	1	0	green
0	1	1	cyan
1	0	0	red
1	0	1	magenta
1	1	0	yellow
1	1	1	white

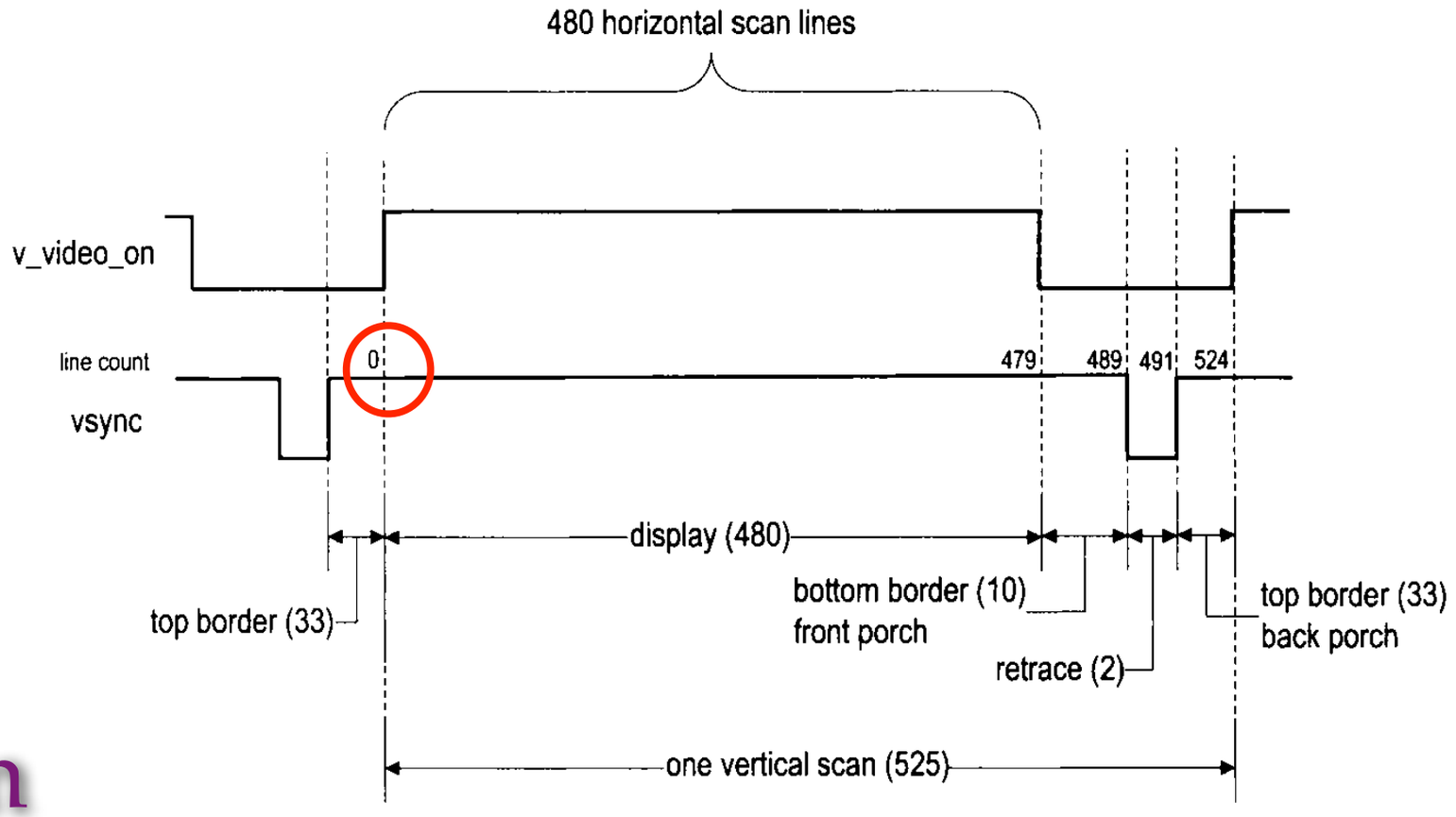
VGA Synchronization: Horizontal

- Four regions
  - Display (visible area)
  - Retrace (sync pulse)
  - Right boarder (front porch)
  - Left border (back porch)
- Start counting from the beginning of the display region



VGA Synchronization: Vertical

- Four regions
  - Display
  - Retrace (sync pulse)
  - Bottom border (front porch)
  - Top border (back porch)
- Start counting from the beginning of the display region



Pixel Clock Calculation

- For VGA 640x480 resolution and 60-Hz frame refresh rate

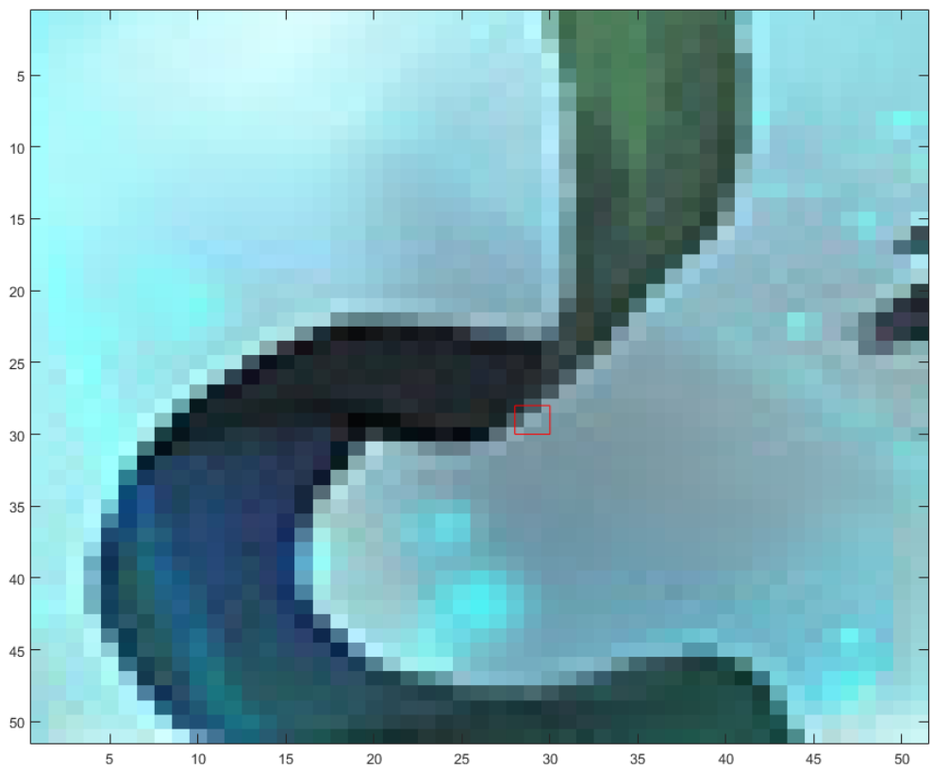
$$800\frac{\text{pixels}}{\text{line}} \cdot 525\frac{\text{lines}}{\text{frame}} \cdot 60\frac{\text{frames}}{\text{sec}} \approx 25M\frac{\text{pixels}}{\text{sec}}$$

Parameter	**Ver. Sync**		**Hor. Sync**	
	Lines	Time(ms)	Pixels	Time(μs)
Visible area	480	15.3	640	25
Front porch	10	0.3	16	0.64
Sync pulse	2	0.064	96	3.8
Back porch	33	1.05	48	1.9
Whole line	525	16.7	800	32

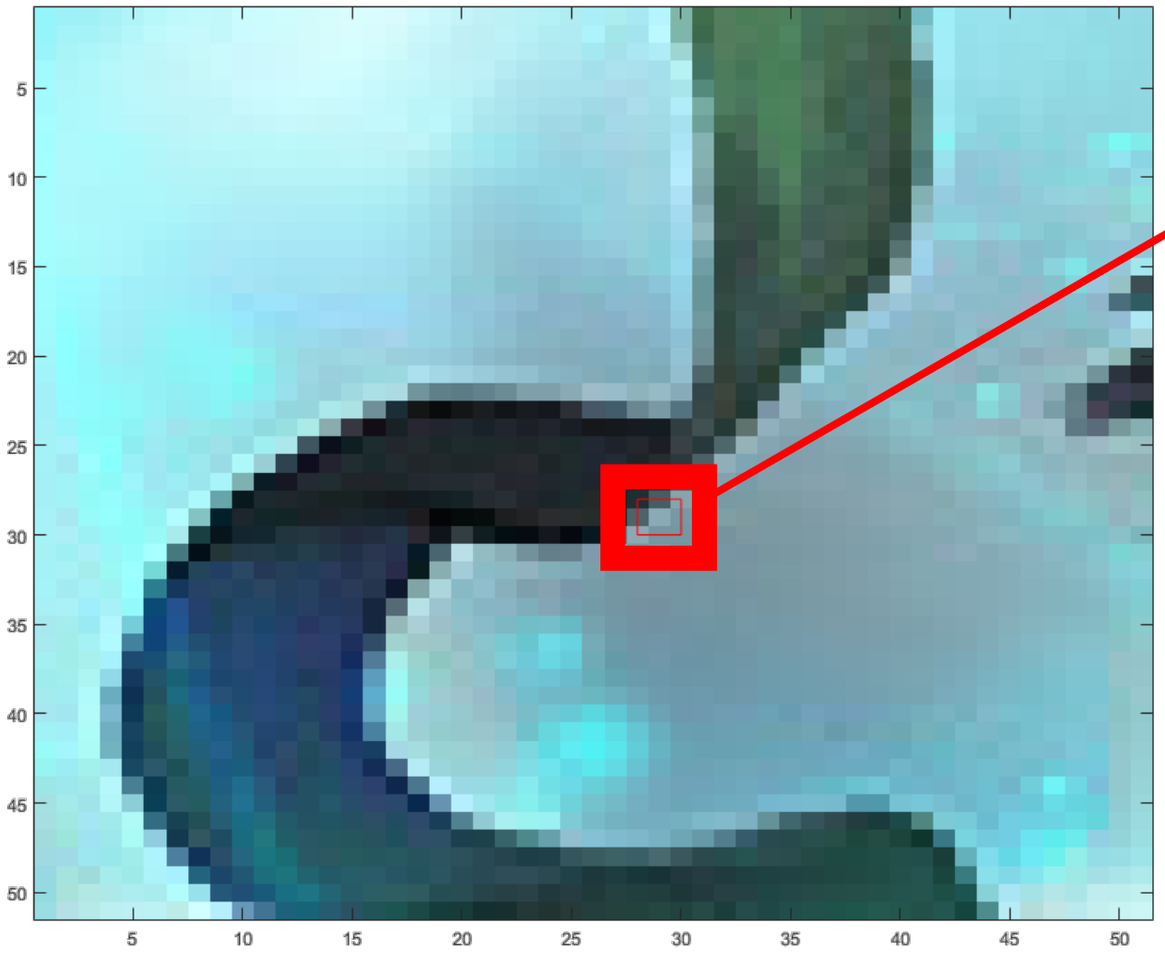
RGB Bitmap Example (1/2)

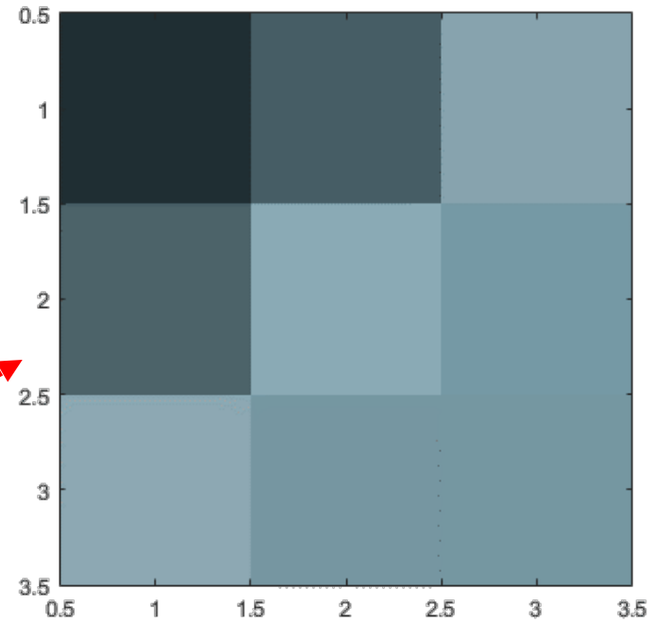
Size: 1215*717

Size: 51*51



RGB Bitmap Example (2/2)



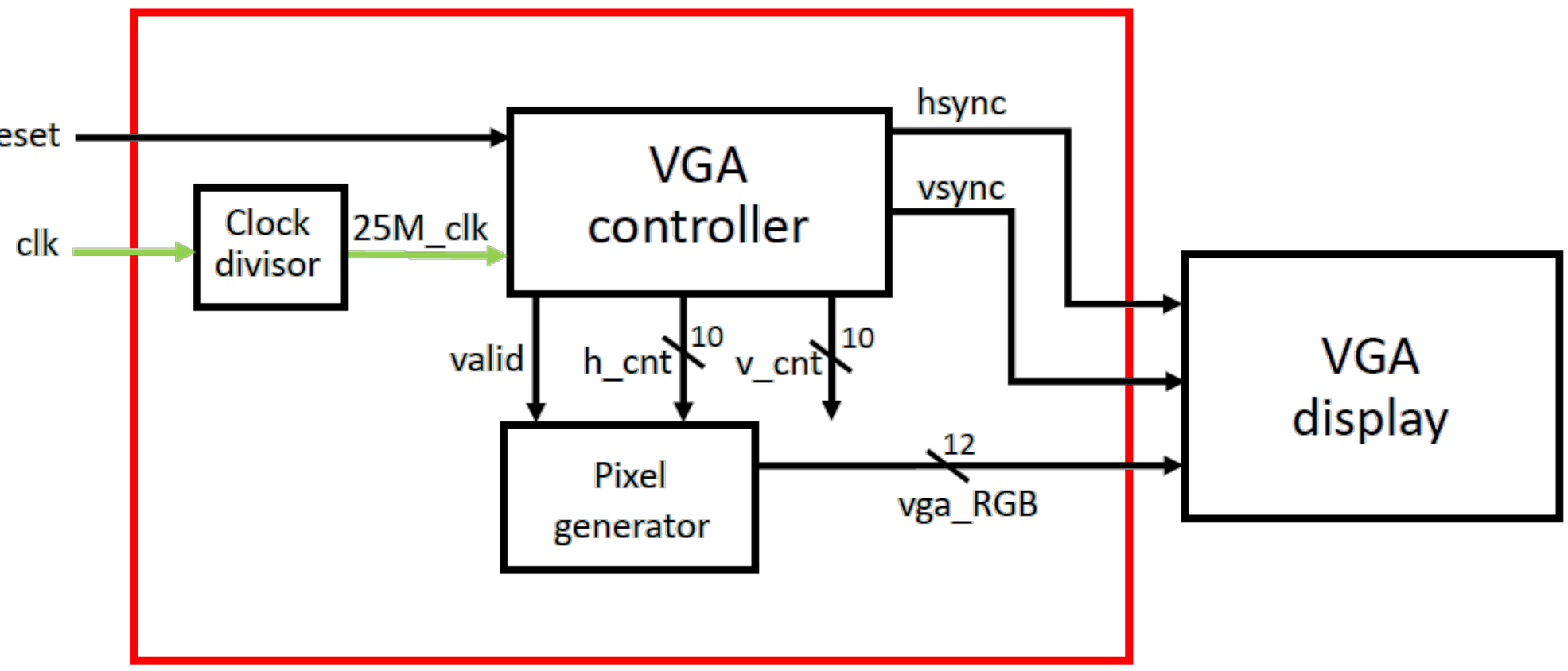


(R,G,B) : range from 0 to 255 N=8

(31,46,51)	(70,93,101)	(135,163,174)
(76,99,105)	(138,170,181)	(117,153,165)
(141,168,179)	(118,150,161)	(117,151,161)

Demo 1: Block Diagram

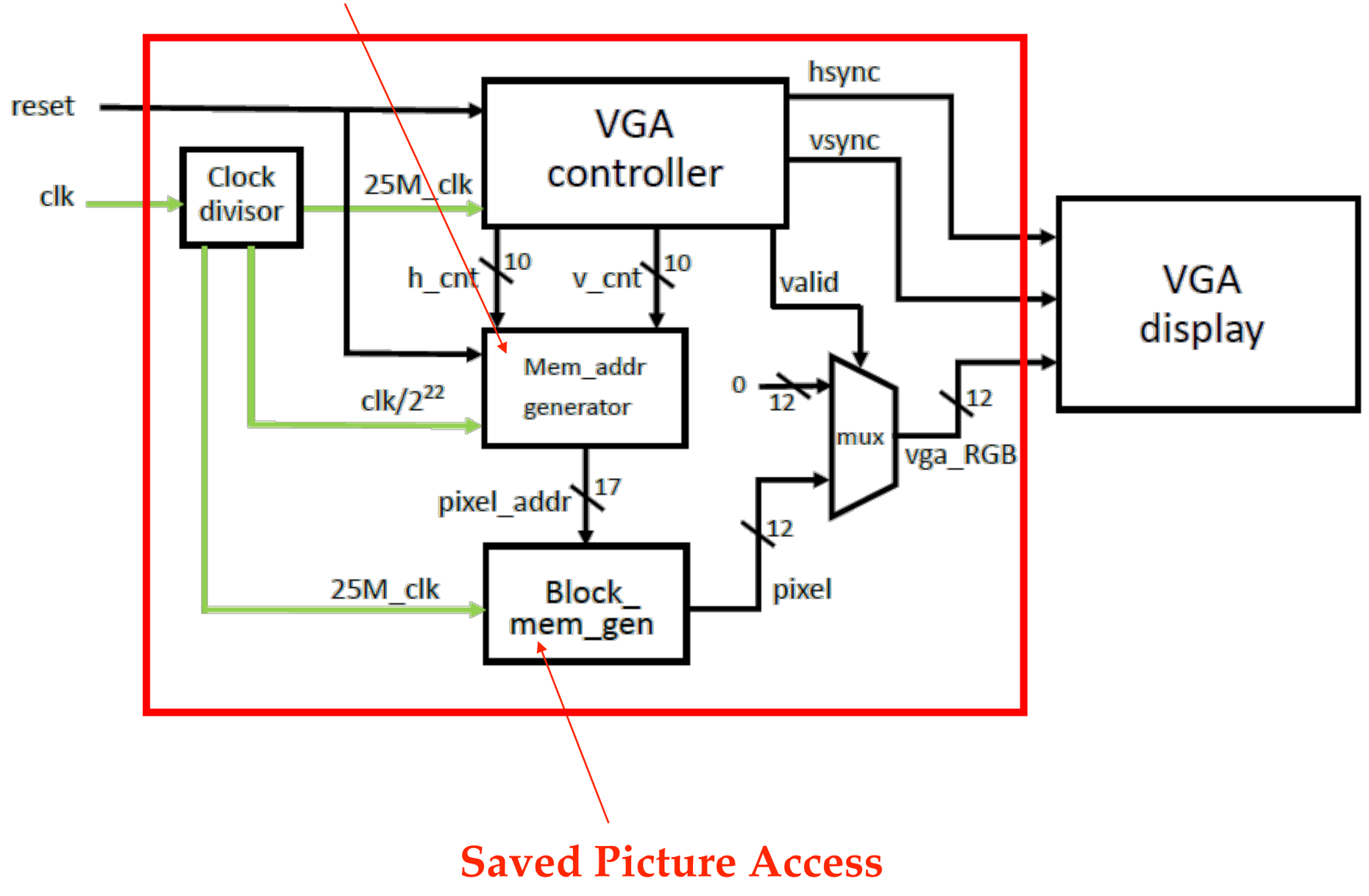
Top



High active reset

Demo 2: Block Diagram

reduce the resolution from 640x480 to 320x240



Saved Picture Access

FPGA Feature Summary

Artix-7 FPGA Feature Summary

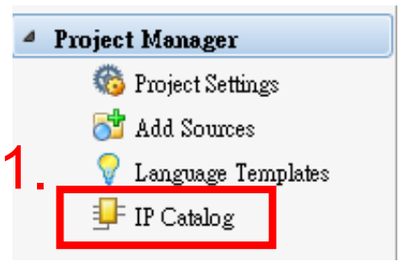
Table 4: **Artix-7 FPGA Feature Summary by Device**

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices[2]	Block RAM Blocks[3]			CMTs[4]	PCIe[5]	GTPs	XADC Blocks	Total I/O Banks[6]	Max User I/O[7]
		Slices[1]	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)						
XC7A12T	12,800	2,000	171	40	40	20	720	3	1	2	1	3	150
XC7A15T	16,640	2,600	200	45	50	25	900	5	1	4	1	5	250
XC7A25T	23,360	3,650	313	80	90	45	1,620	3	1	4	1	3	150
XC7A35T	33,280	5,200	400	90	100	50	1,800	5	1	4	1	5	250
XC7A50T	52,160	8,150	600	120	150	75	2,700	5	1	4	1	5	250
XC7A75T	75,520	11,800	892	180	210	105	3,780	6	1	8	1	6	300
XC7A100T	101,440	15,850	1,188	240	270	135	4,860	6	1	8	1	6	300
XC7A200T	215,360	33,650	2,888	740	730	365	13,140	10	1	16	1	10	500

Notes:
1. Each 7 series FPGA slice contains four LUTs and eight flip-flops; only some slices can use their LUTs as distributed RAM or SRLs.
2. Each DSP slice contains a pre-adder, a 25 x 18 multiplier, an adder, and an accumulator.
3. Block RAMs are fundamentally 36 Kb in size; each block can also be used as two independent 18 Kb blocks.
4. Each CMT contains one MMCM and one PLL.
5. Artix-7 FPGA Interface Blocks for PCI Express support up to x4 Gen 2.
6. Does not include configuration Bank 0.
7. This number does not include GTP transceivers.

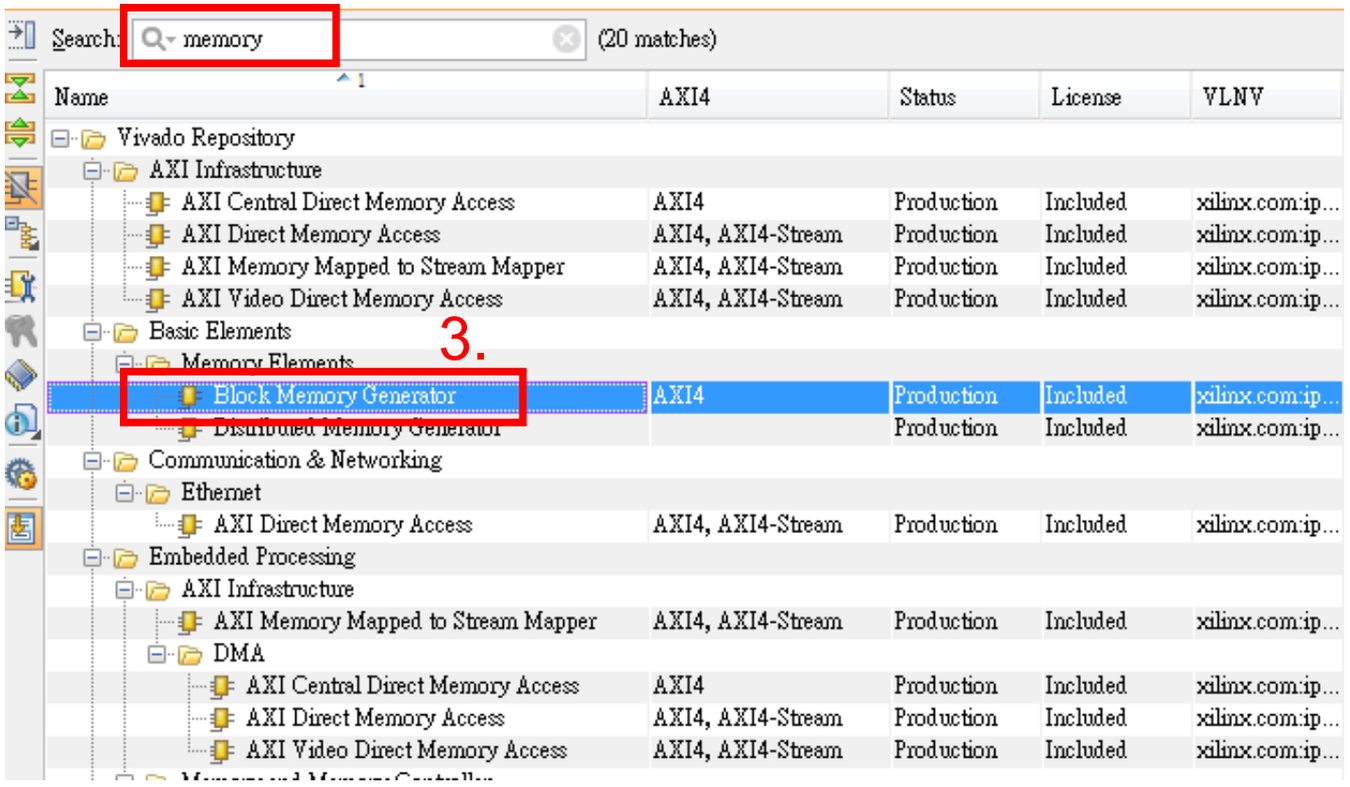
Memory IP (1/5)

1.

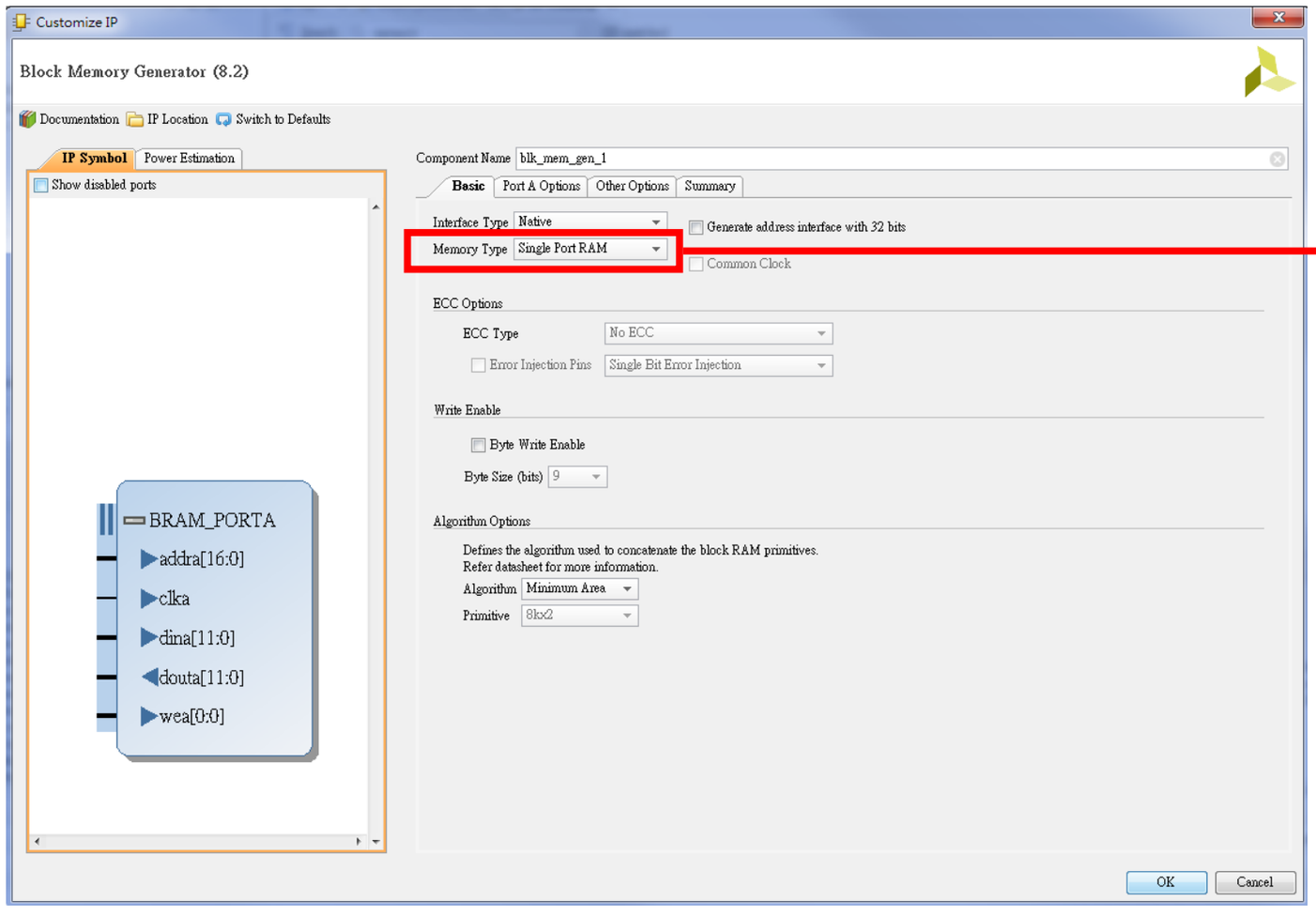


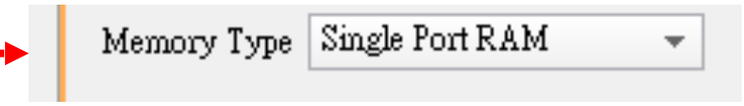
2.

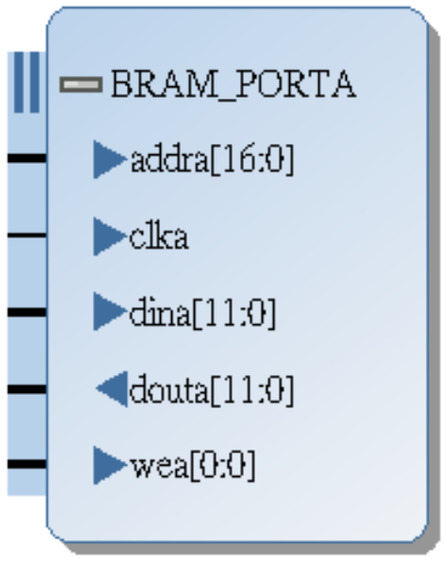
3.



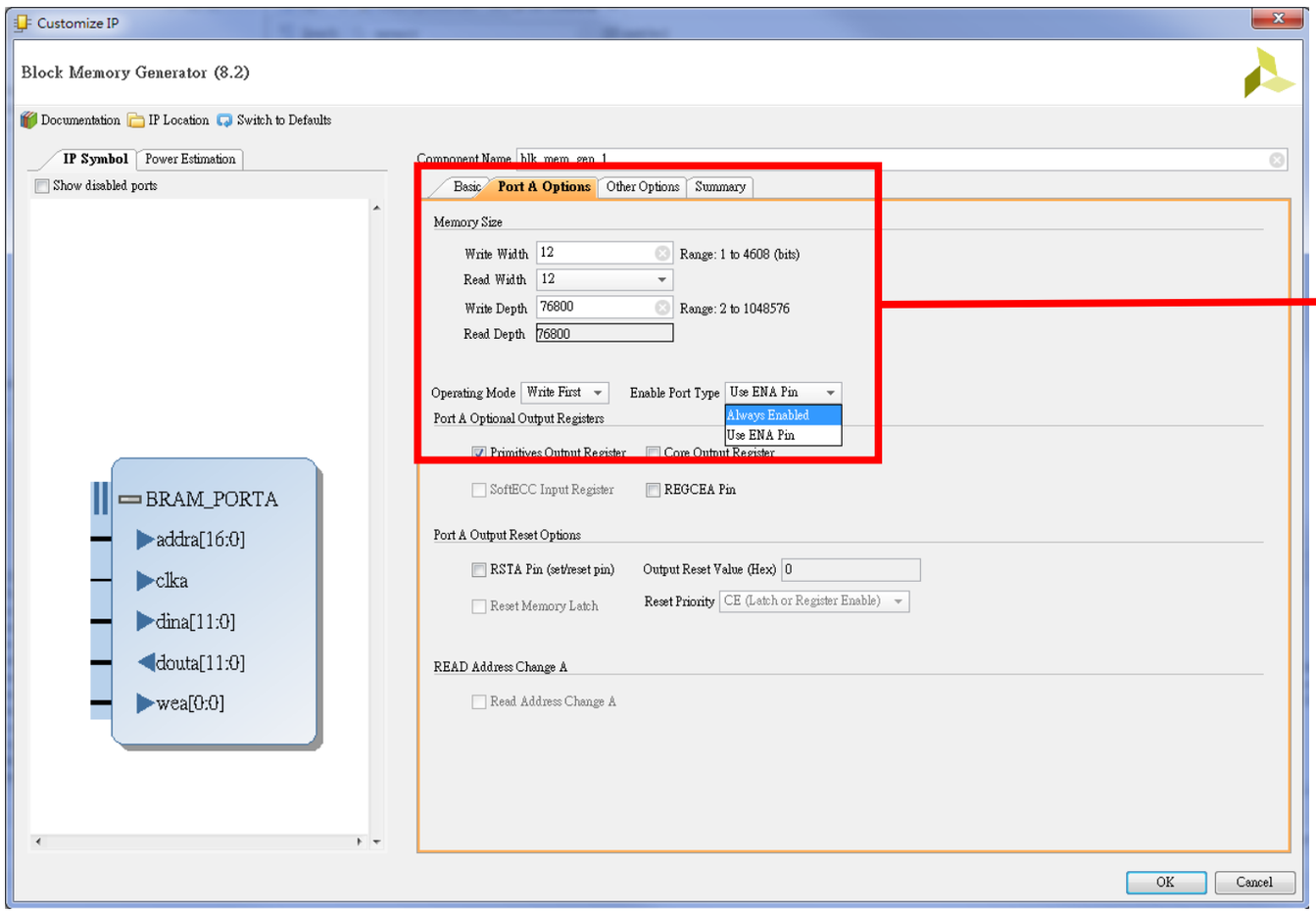
Memory IP (2/5)

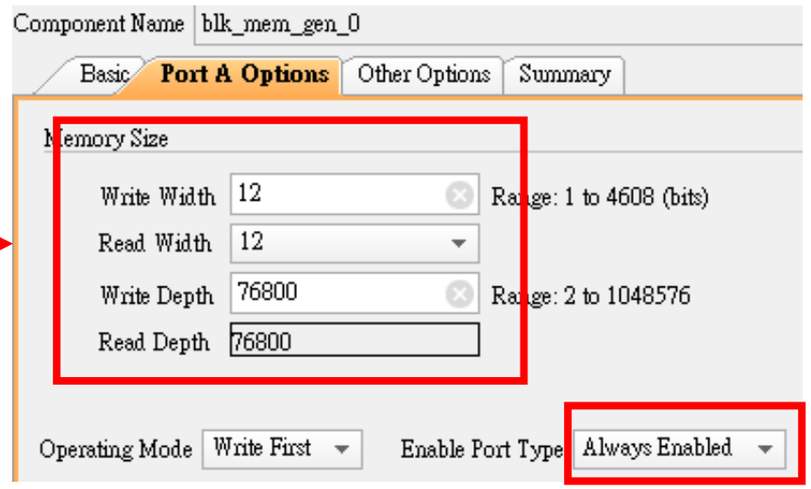

Customize IP
Block Memory Generator (8.2)
Documentation
IP Location
Switch to Defaults
IP Symbol
Power Estimation
Show disabled ports
Component Name
blk_mem_gen_1
Basic
Port A Options
Other Options
Summary
Interface Type
Native
Generate address interface with 32 bits
Memory Type
Single Port RAM
Common 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
8kx2
OK
Cancel
BRAM_PORTA
addra[16:0]
clka
dina[11:0]
douta[11:0]
wea[0:0]


Memory Type
Single Port RAM


BRAM_PORTA
addra[16:0]
clka
dina[11:0]
douta[11:0]
wea[0:0]

Memory IP (3/5)

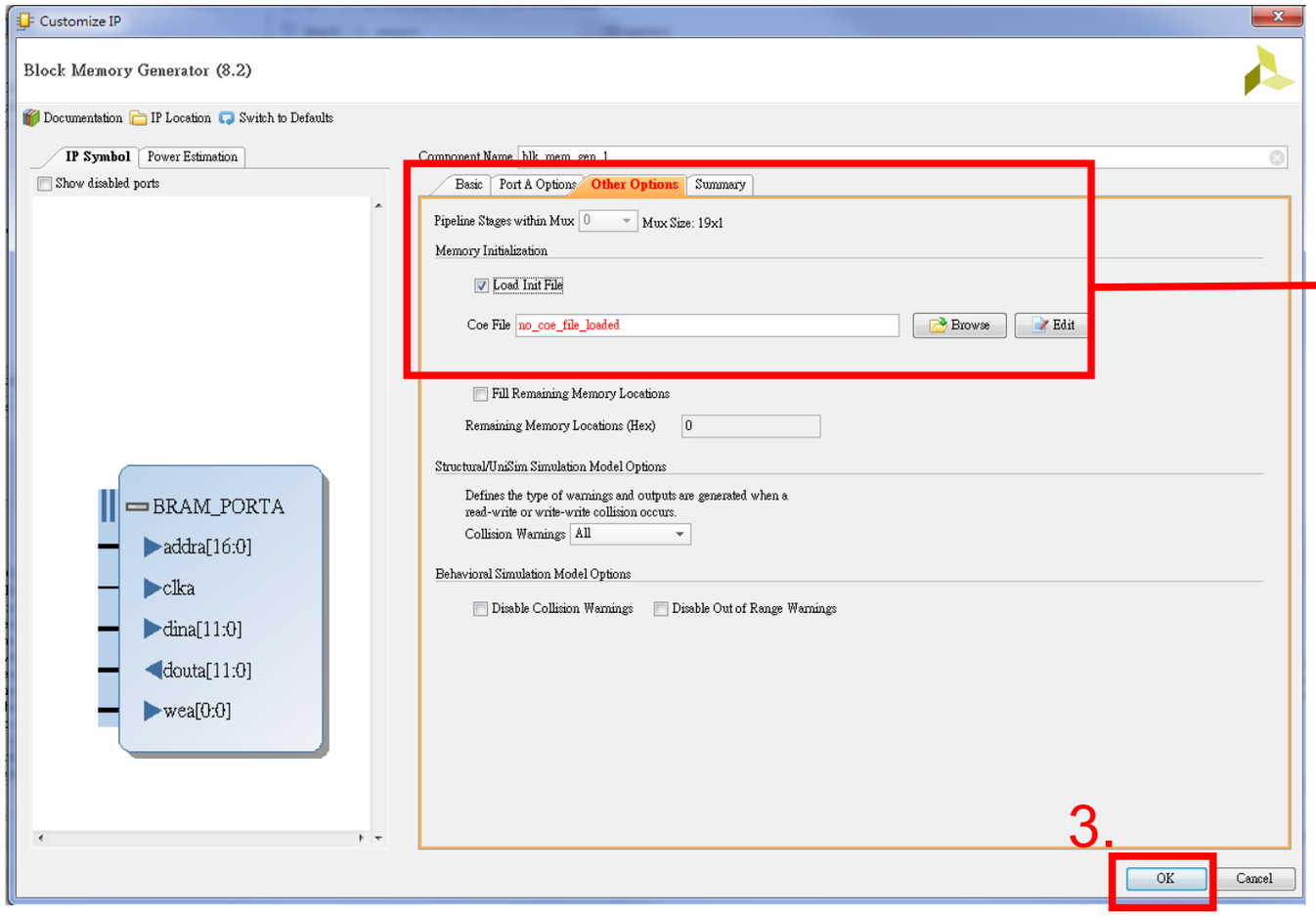


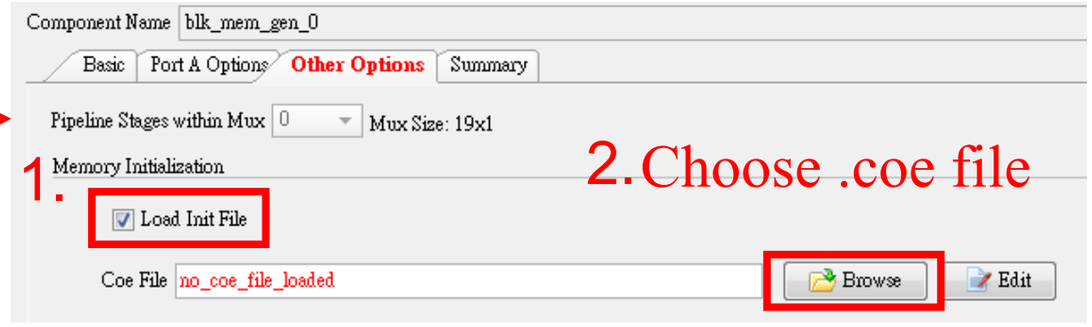


RGB : 12 bits
320x240 : 76800
Total bits : 921600 bits

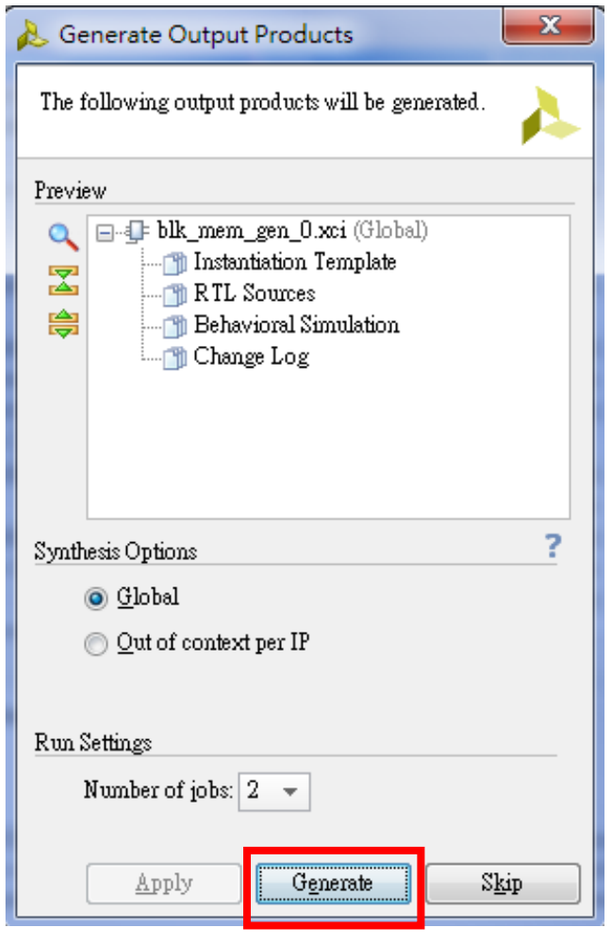
1800kbits of fast block RAM in FPGA

Memory IP (4/5)


Customize IP
Block Memory Generator (8.2)
Documentation
IP Location
Switch to Defaults
IP Symbol
Power Estimation
Show disabled ports
Component Name blk_mem_gen_1
Basic
Port A Options
Other Options
Summary
Pipeline Stages within Mux 0
Mux Size: 19x1
Memory Initialization
Load Init File
Coe File no_coe_file_loaded
Browse
Edit
Fill Remaining Memory Locations
Remaining Memory Locations (Hex) 0
Structural/UniSim Simulation Model Options
Defines the type of warnings and outputs are generated when a read-write or write-write collision occurs.
Collision Warnings All
Behavioral Simulation Model Options
Disable Collision Warnings
Disable Out of Range Warnings
BRAM_PORTA
addra[16:0]
clka
dina[11:0]
douta[11:0]
wea[0:0]
3.
OK
Cancel


Component Name blk_mem_gen_0
Basic
Port A Options
Other Options
Summary
Pipeline Stages within Mux 0
Mux Size: 19x1
1.
Memory Initialization
Load Init File
2. Choose .coe file
Coe File no_coe_file_loaded
Browse
Edit

Memory IP (5/5)


Generate Output Products
The following output products will be generated.
Preview
blk_mem_gen_0.xci (Global)
Instantiation Template
RTL Sources
Behavioral Simulation
Change Log
Synthesis Options
Global
Out of context per IP
Run Settings
Number of jobs: 2
Apply
Generate
Skip

Picture Format Translation (1/2)

amumu.jpg

PicTrans.exe

out.coe

out.coe

```
1  memory_initialization_radix=16;
2  memory_initialization_vector=
3  115,
4  115,
5  115,
6  115,
7  115,
8  115,
9  115,
10 115,
11 115,
12 115,
   ⋮   ⋮
76797 743,
76798 743,
76799 743,
76800 743,
76801 743,
76802 743;
76803
```

Picture Format Translation (2/2)

- PicTrans.exe:
  - Convert a *.jpg file to a bit map file
- Input:
  - image (*.jpg)
  - the width of the output file
  - the height of the output file
- Output:
  - out.coe

Lab 9: VGA Display

Action Item (1/2)

Modify the Verilog code introduced in class to design a circuit for controlling the VGA display.

- input ports:

```
input  clk;
input  reset;
input  en;
```

- output ports:

```
output  [3:0]vgaRed;
output  [3:0]vgaGreen;
output  [3:0]vgaBlue;
output  hsync;
output  vsync;
```

Action Item (2/2)

- At the beginning or when pressing the **reset** button, the VGA display will show the image (e.g., amumu.jpg) at the origin position. It will stay still until the **en** button is pressed.
- The image will start/resume scrolling down row by row under the frequency of clk divided by 2^{22} (i.e., clk/2^{22}), or pause, depending on whether the number of the **en** button pressed is odd or even.

Example (1/6)

at the beginning or pressing **reset**

(0 row scrolled down)

Example (2/6)

press **en** ➔ start to scroll down

(100 rows scrolled down)

Example (3/6)

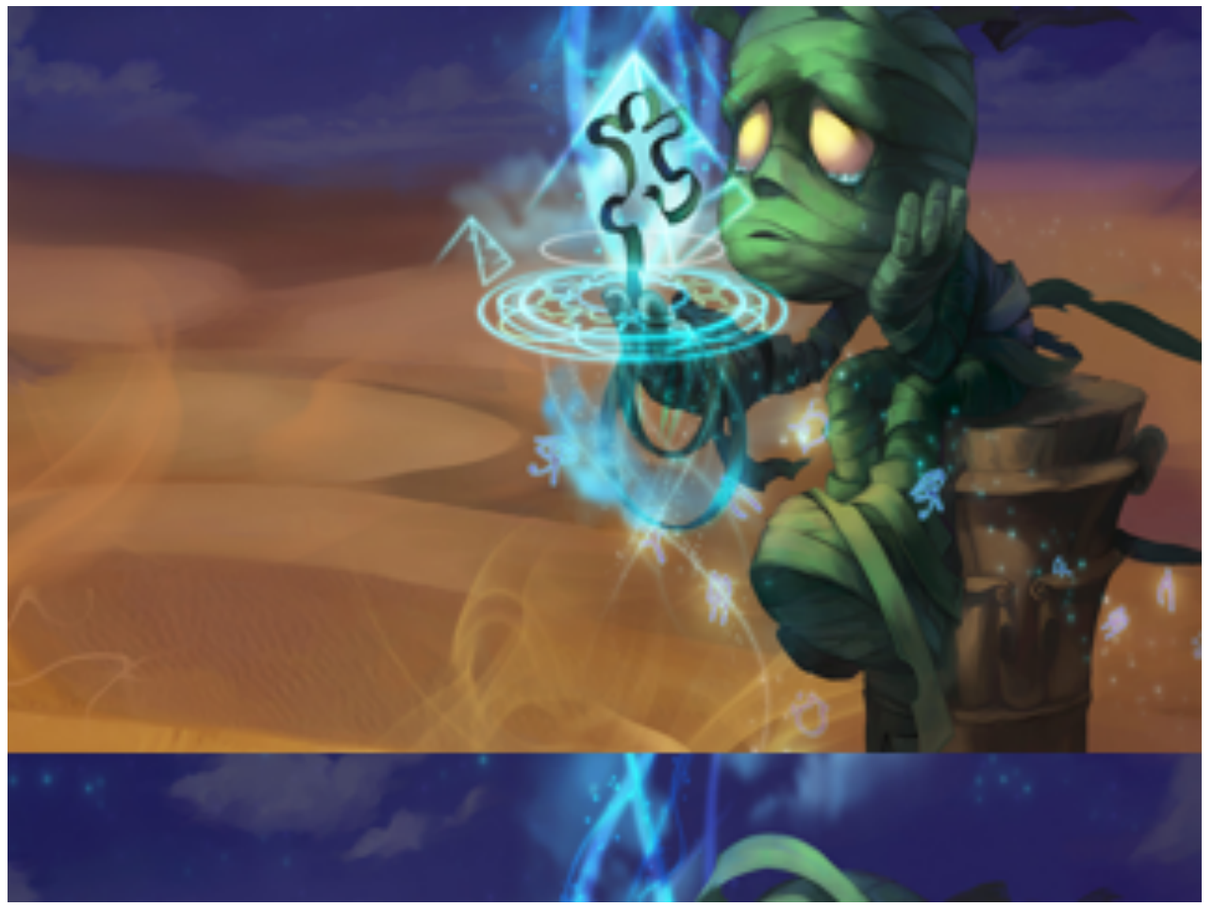
(200 rows scrolled down)

Example (4/6)

(300 rows scrolled down)

Example (5/6)

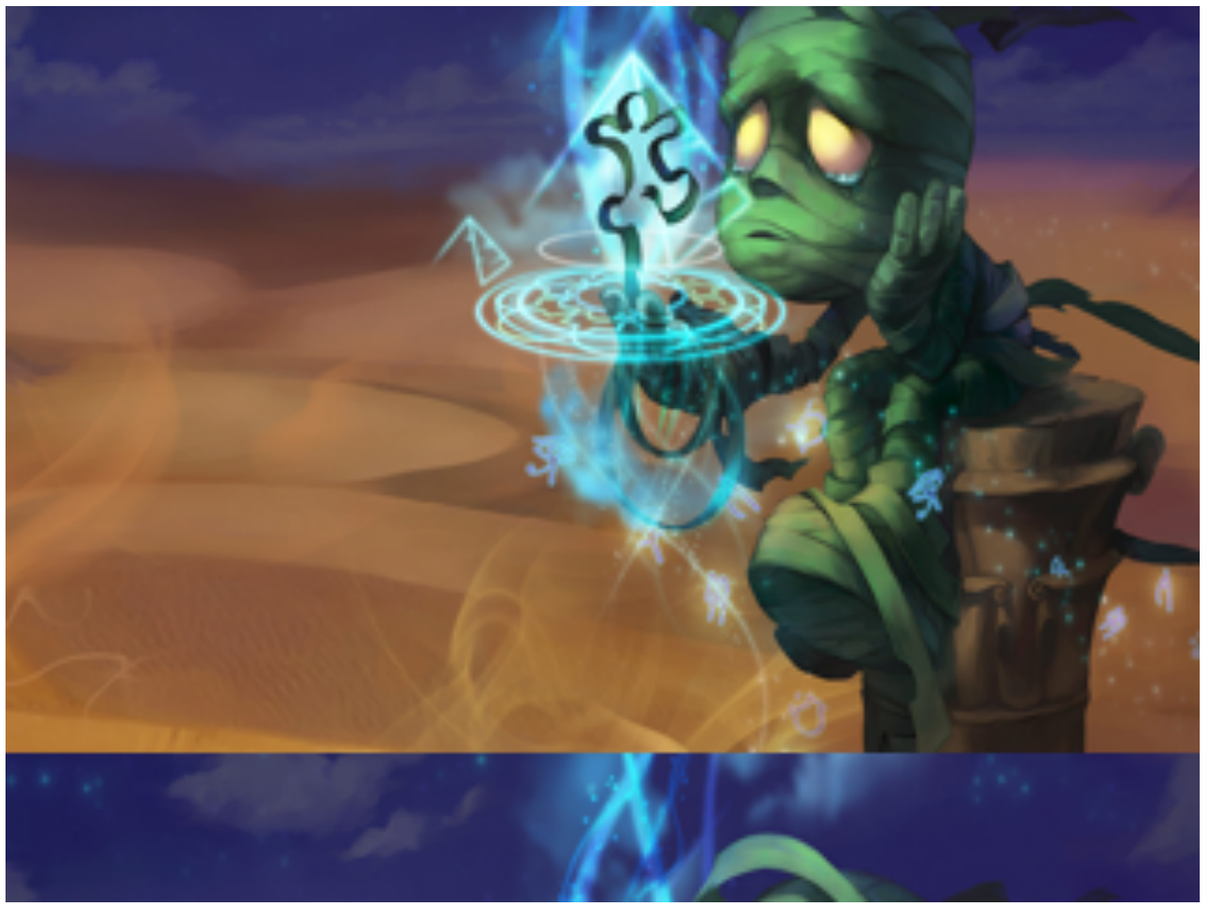
(400 rows scrolled down)



Example (6/6)

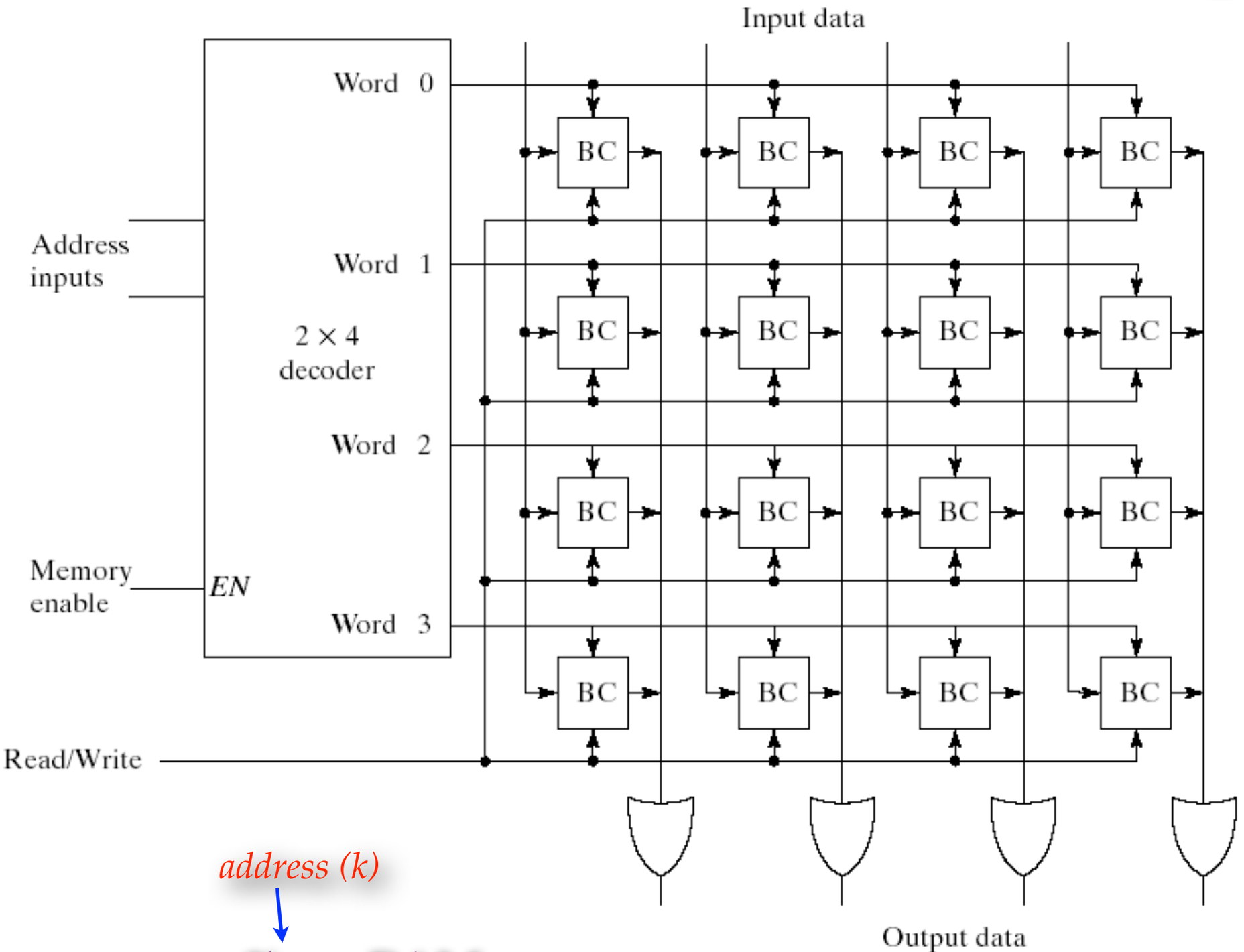
press **en** → pause

(400 rows scrolled down)



RAM Revisit

2^2 x 4 RAM



address (k)

2^k x n RAM

# of output bits (n)

COE Format

- COE: memory coefficient file
- Two parameter:
  - **memory_initialization_radix**
    - Radix of the values in the *memory_initialization_vector*
    - Ex: 2, 10, or 16
  - **memory_initialization_vector:**
    - Memory content
    - Memory words are separated by whitespace
    - You can use comma (,) to help identify the boundary
    - Vector (entire memory) ended by semicolon

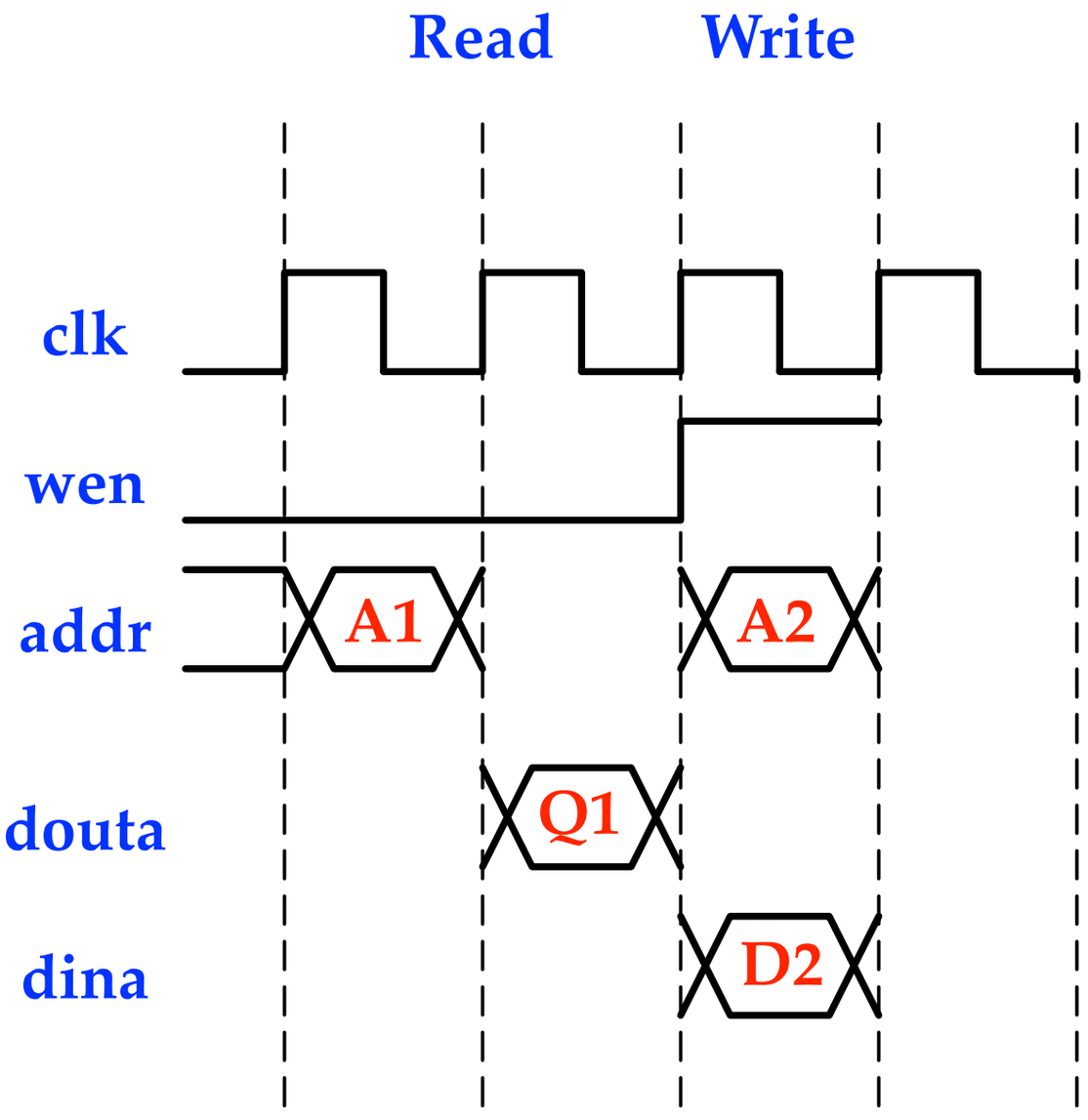
How to Use RAM Module

- Memory Read and Write is controlled by
  - wen pin: wen=0 for "read" and wen=1 for "write"
  - Separate read-address generator and write-address generator
- To refresh the image, you can
  - Always read out image from memory (wen=0) and display image in VGA synchronized with hsync and vsync. Write in image changes (wen=1) when necessary

How to Use RAM Module

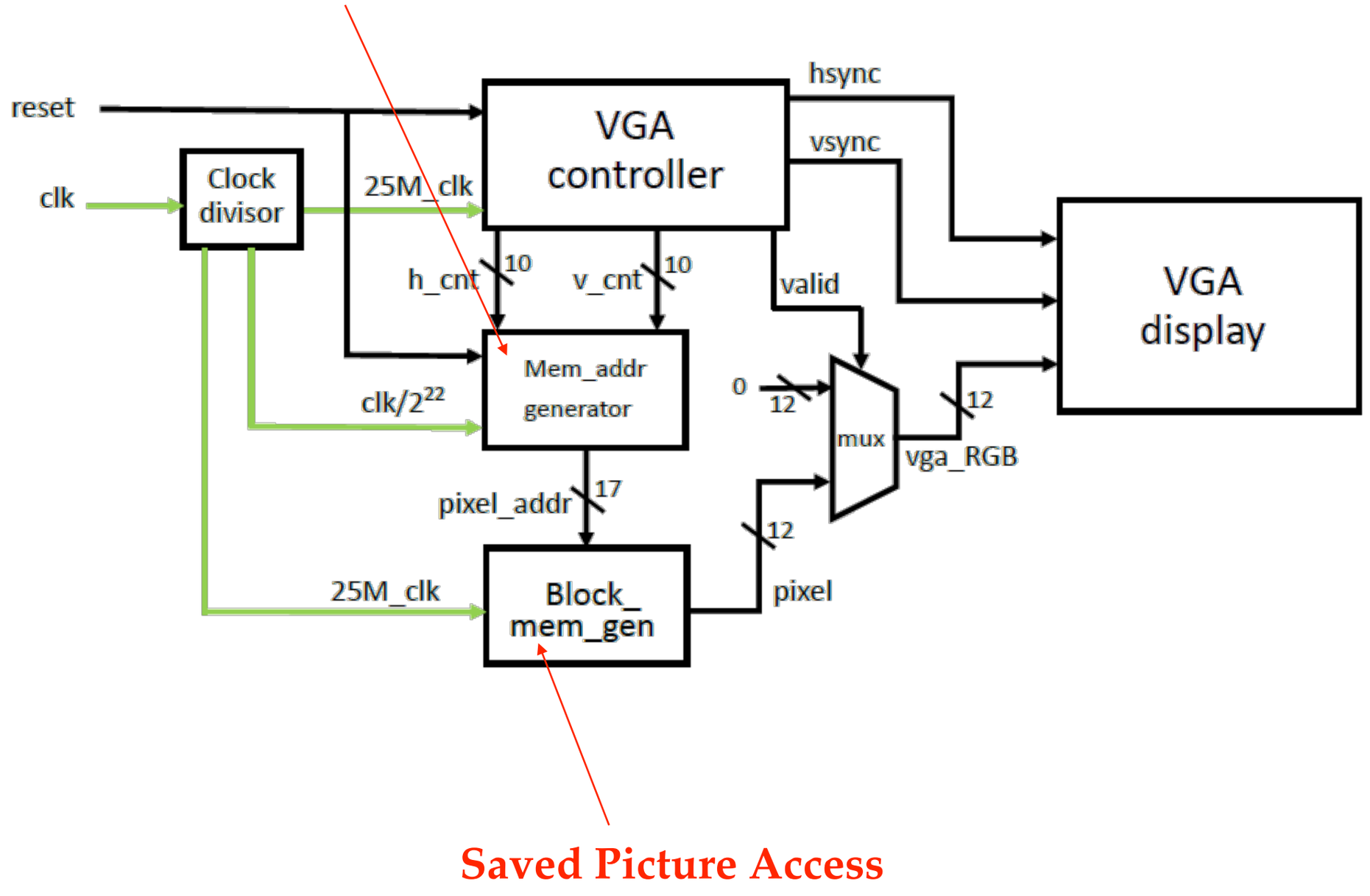
```
wire clk;
wire wen;
wire [63:0] data_in;
wire [63:0] out_64;
wire [5:0] addr;


RAM R1(
  .clka(clk),
  .wea(wen),
  .addra(addr),
  .dina(data_in),
  .douta(out_64)
);
```



Demo 2: Block Diagram

reduce the resolution from 640x480 to 320x240



Saved Picture Access

Memory address generator

```
module mem_addr_gen(
    input clk,
    input rst,
    input [9:0] h_cnt,
    input [9:0] v_cnt,
    output [16:0] pixel_addr
    );

    reg [7:0] position;

    assign pixel_addr = ((h_cnt>>1)+320*(v_cnt>>1)+ position*320 )% 76800;  //640*480 --> 320*240

    always @ (posedge clk or posedge rst) begin
        if(rst)
            position <= 0;
        else if(position < 239)
            position <= position + 1;
        else
            position <= 0;
    end

endmodule
```