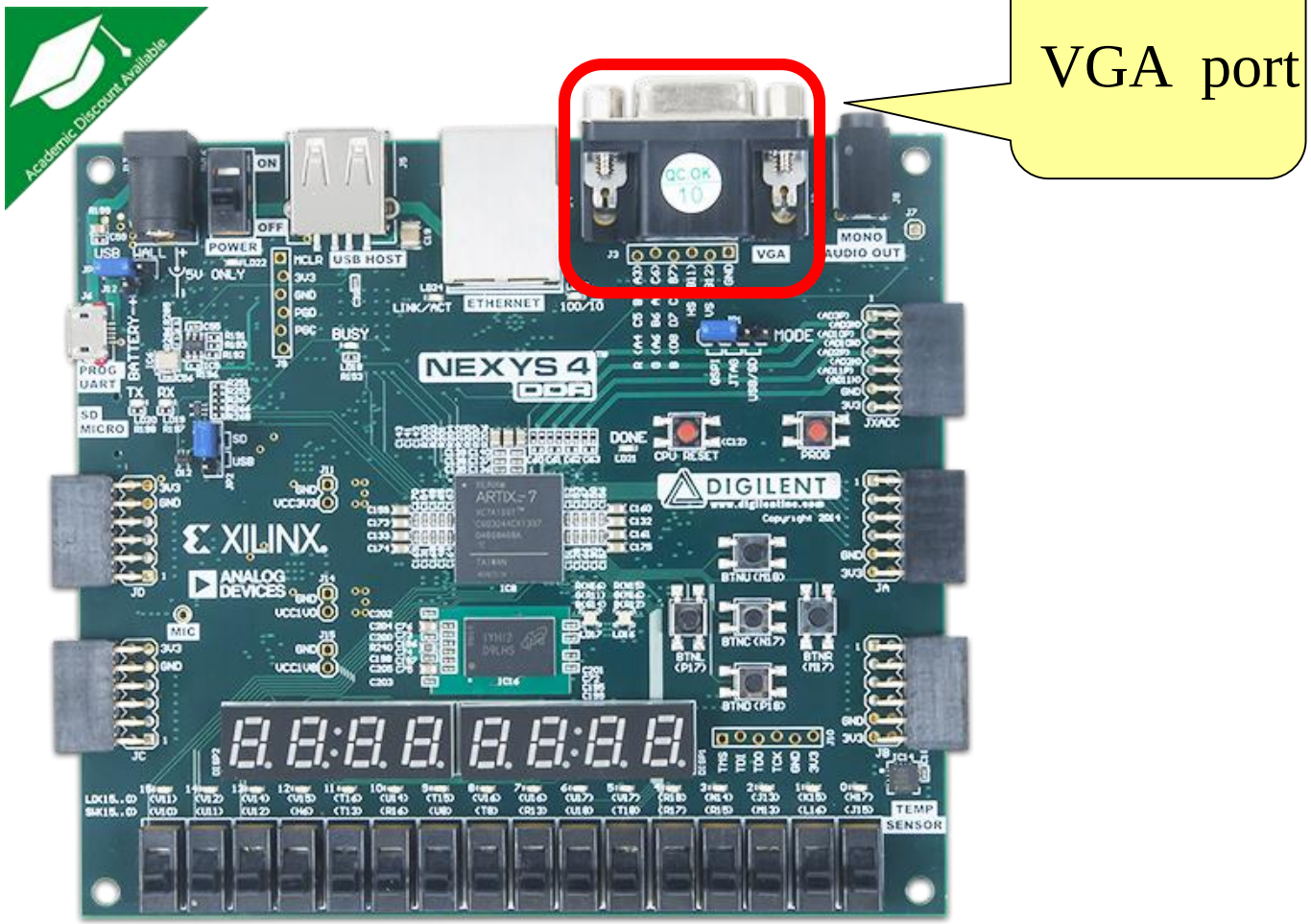
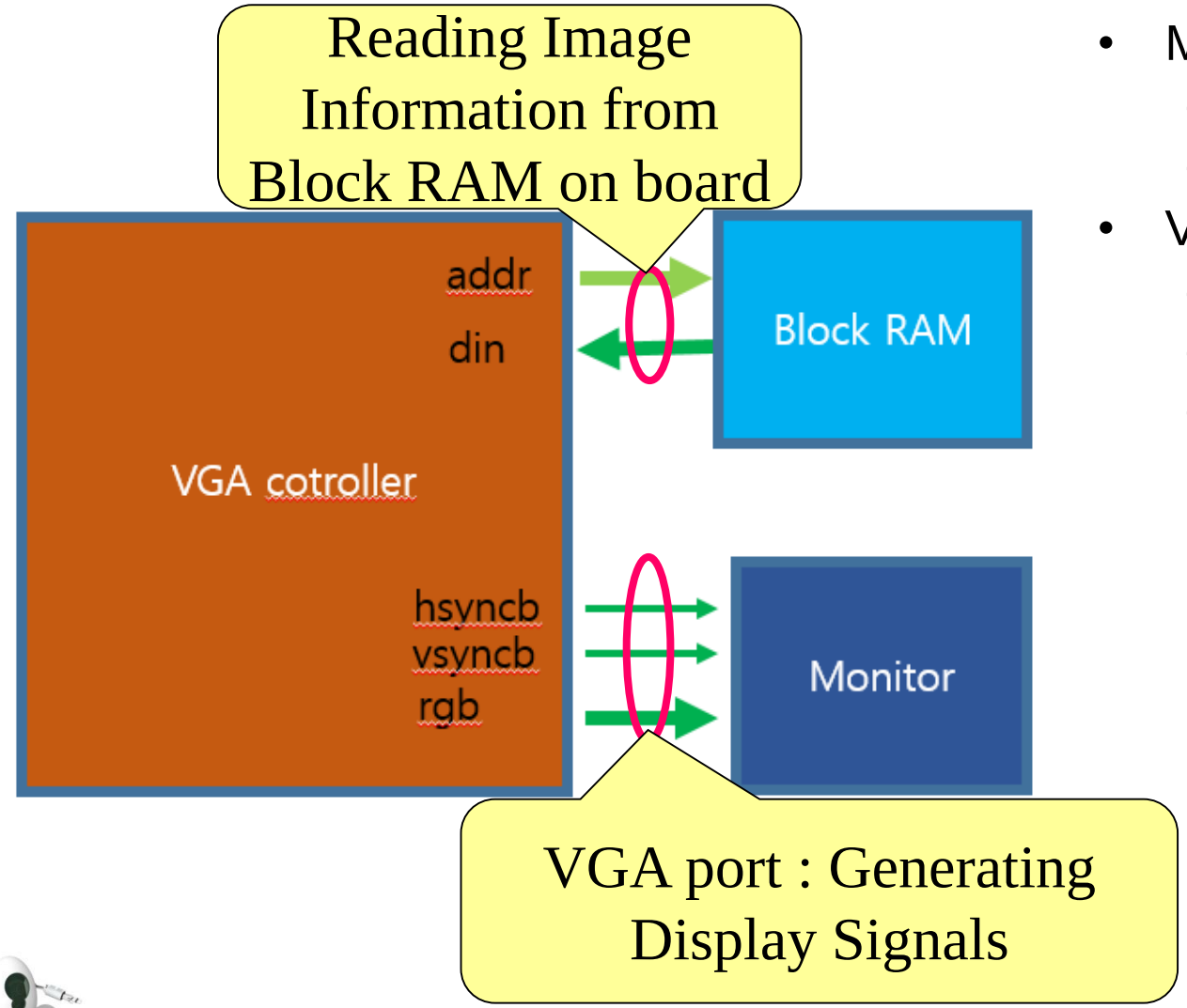
VGA Interface

Kang Yi



한동대 전산전자공학부 이강

VGA IP



- Memory Signals
  - Addr : address output(19bits)
  - Data : data input (16 bits)
- Video Signals
  - Hsync : new line
  - Vsync : new frame
  - RGB : Red, Green and Blue components of pixel color
    - 4:4:4 bits for each component

RGB value

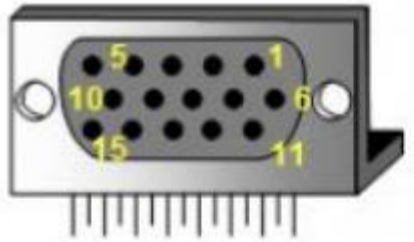
- 444 mode Color (4K colors)
  - Red : 4 bits
  - Green : 4 bits
  - Blue : 4 bits

$$\frac{V_R-R_3}{510}+\frac{V_R-R_2}{1000}+\frac{V_R-R_1}{2000}+\frac{V_R-R_0}{4000}+\frac{V_R}{75}=0$$

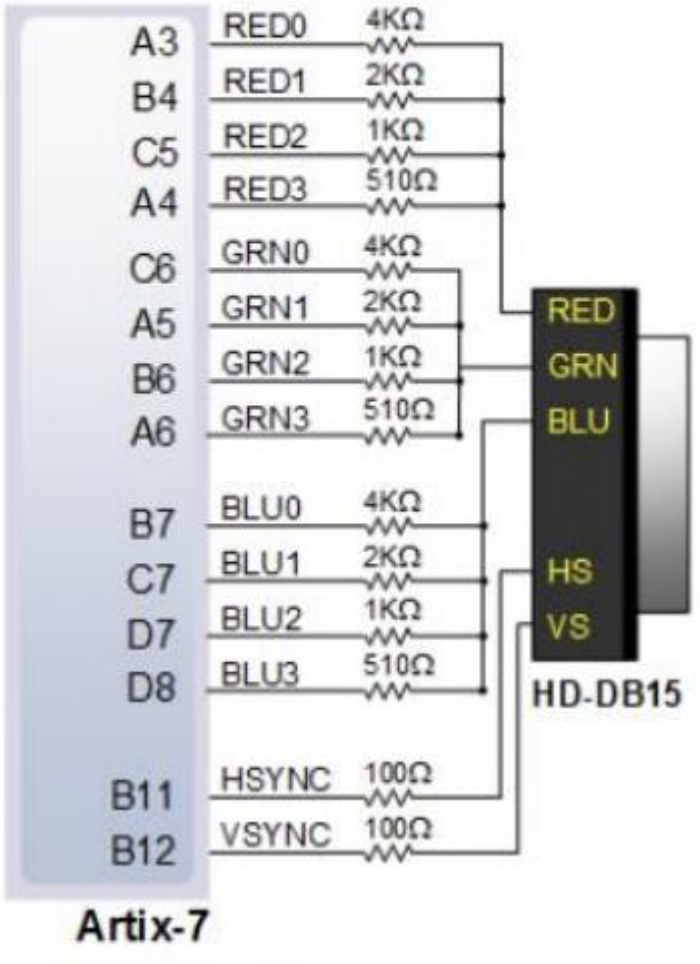
$$V_R = 0.117R_3 + 0.059R_2 + 0.029R_1 + 0.015R_0$$

If $R_2=R_1=R_0=3.3V$, $V_R=0.726V$

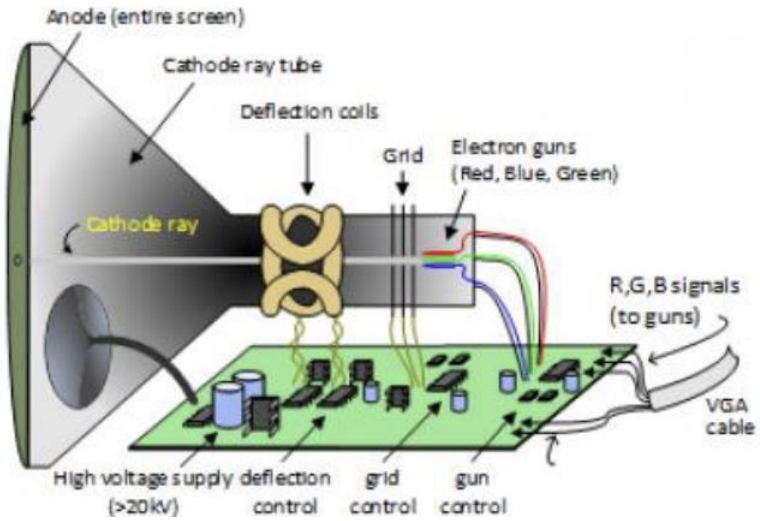
- Digital to Analog Convert by resistor divider circuits
  - 12 bit data → analog
  - 0.72 V (fully on) ~ 0 V (fully off)



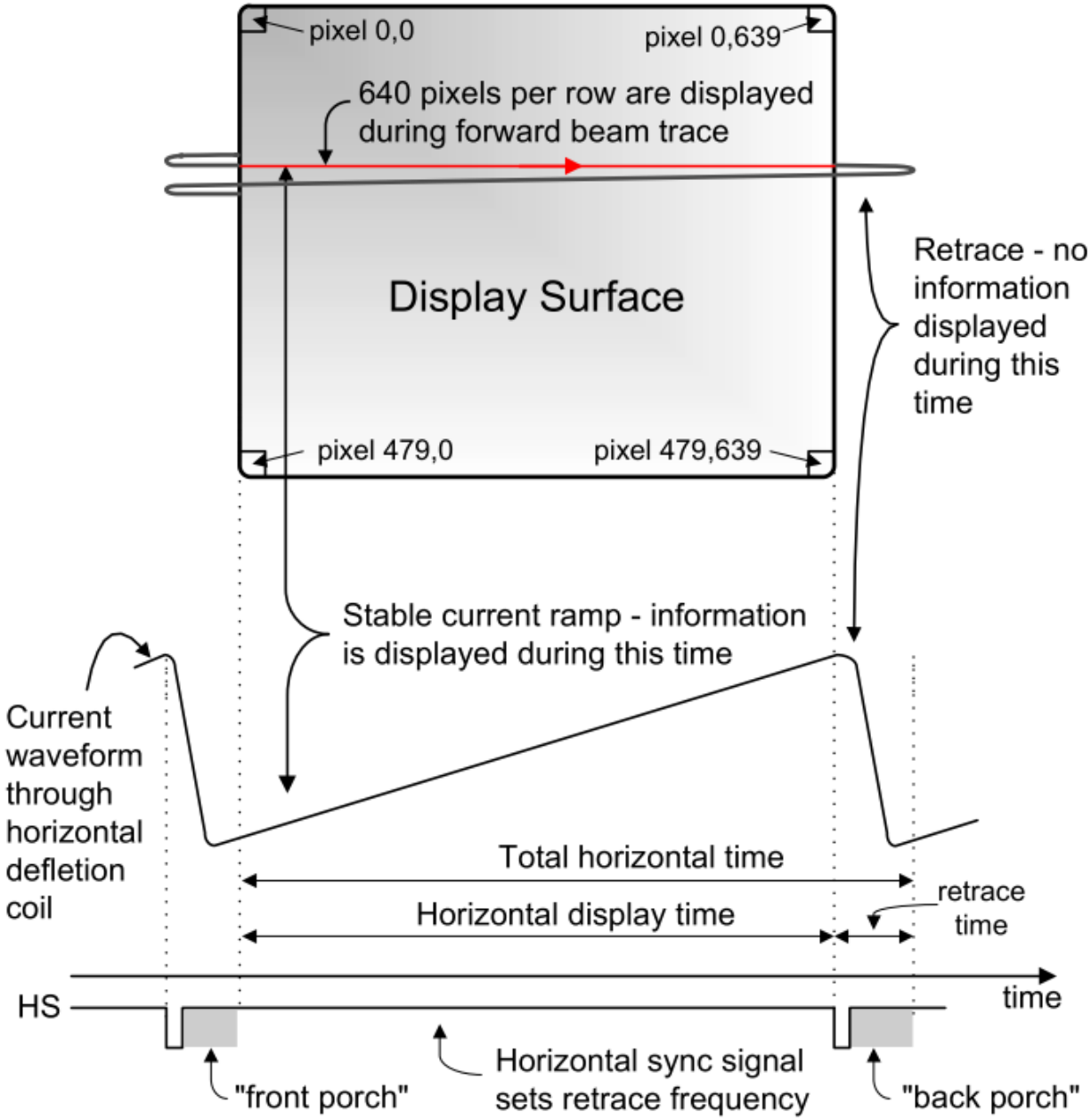
Pin 1: Red
Pin 2: Grn
Pin 3: Blue
Pin 13: HS
Pin 14: VS
Pin 5: GND
Pin 6: Red GND
Pin 7: Grn GND
Pin 8: Blu GND
Pin 10: Sync GND

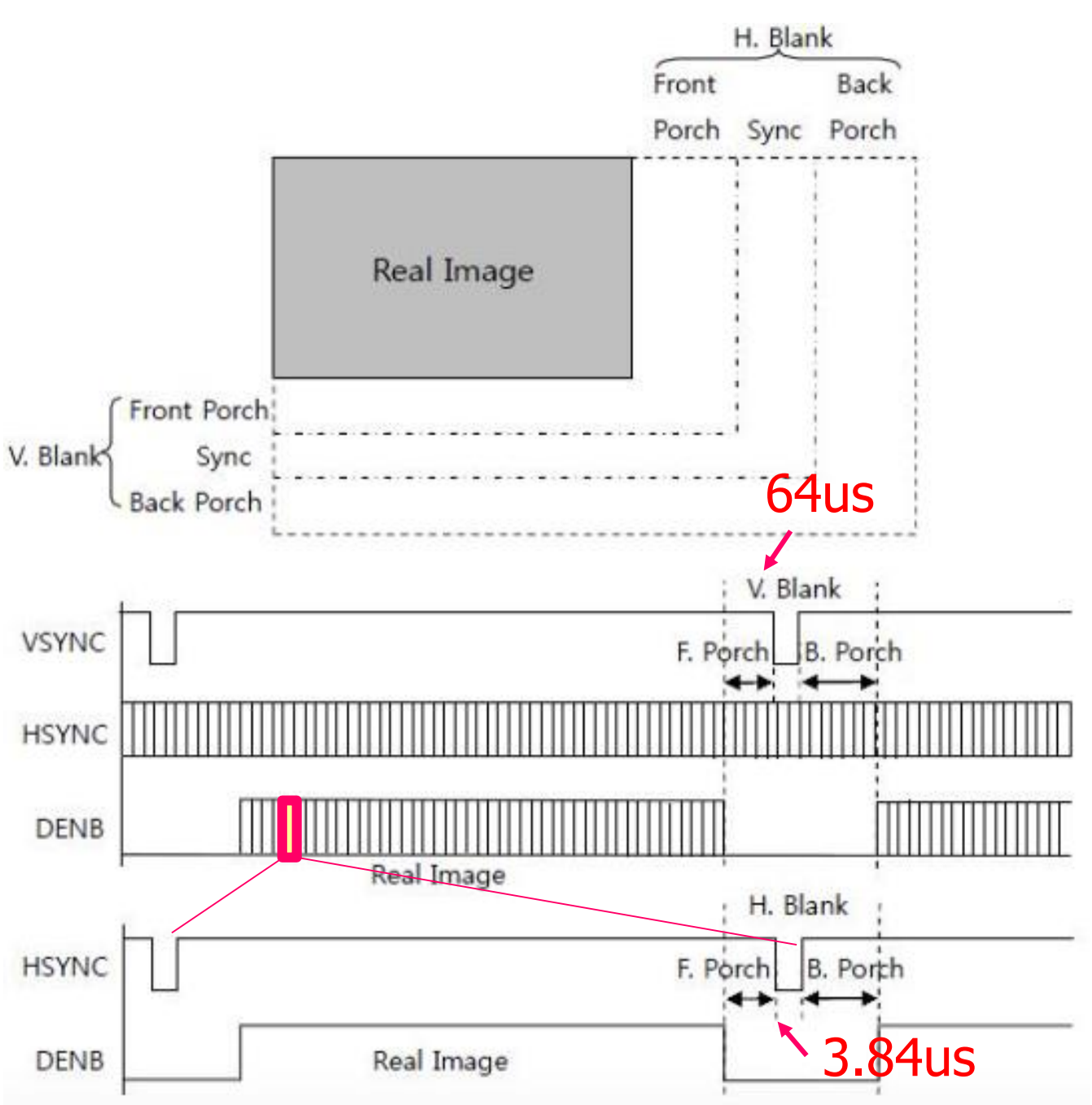


Legacy VGA



- Resolution: 640x480




H. Blank
Front Porch
Back Porch
Sync
Real Image
V. Blank
Front Porch
Sync
Back Porch
64us
V. Blank
VSYNC
F. Porch
B. Porch
HSYNC
DENB
Real Image
H. Blank
HSYNC
F. Porch
B. Porch
DENB
Real Image
3.84us

VGA Timing

- Vsync : frame rate=60Hz
  - ➔ frame time = 1/60 = 16.67 ms
- VGA Resolution :
  - 480 rows by 640 pixels
- Line time =

  (frame time –Vert T_{fp} – T_{bp}-T_{pw})/480

  = (16670–320–928-64)us/480

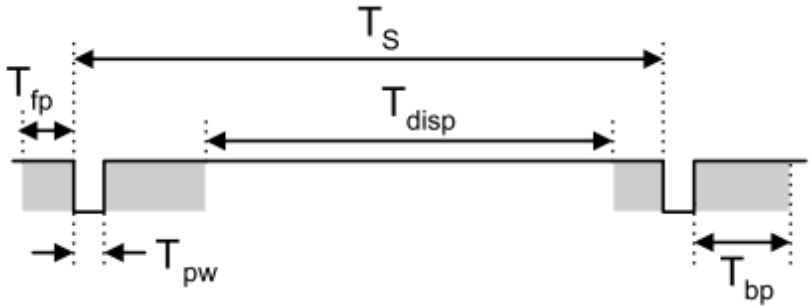
  = 15,358 us/ 480 = 32 us
- Pixel time =

  (Line time–Horiz T_{fp} –T_{bp}-T_{pw})/640

  = (32100–640–1920-3840)ns/640

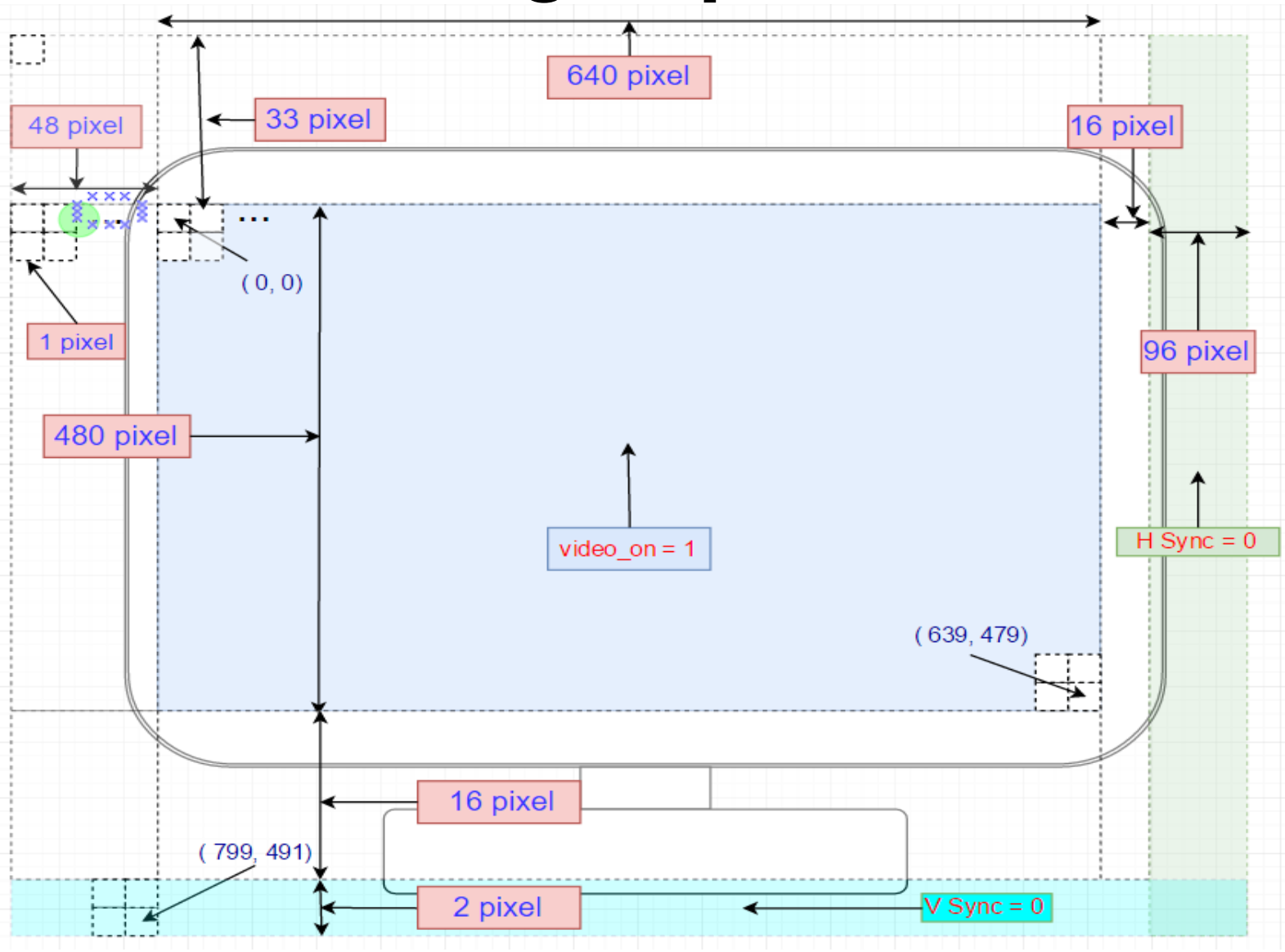
  = 256000/640 = 40 ns

  ➔ 25MHz



Symbol	Parameter	Vertical Sync			Horiz. Sync	
		Time	Clocks	Lines	Time	Clks
T_S	Sync pulse	16.7ms	416,800	521	32 us	800
T_{disp}	Display time	15.36ms	384,000	480	25.6 us	640
T_{pw}	Pulse width	64 us	1,600	2	3.84 us	96
T_{fp}	Front porch	320 us	8,000	10	640 ns	16
T_{bp}	Back porch	928 us	23,200	29	1.92 us	48

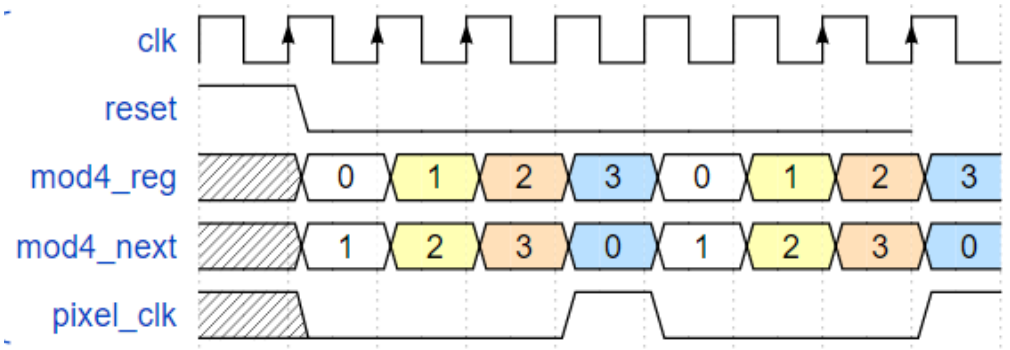
VGA Timing in pixel time

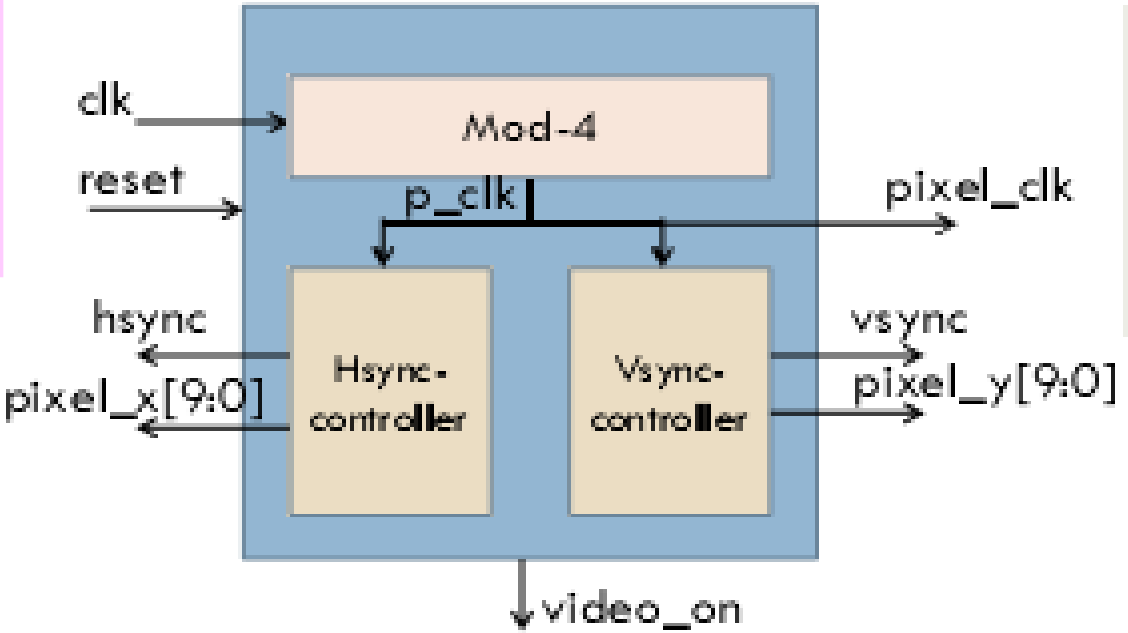


Pixel Clock Enable

- pixel_clk = CLK100MHZ/4

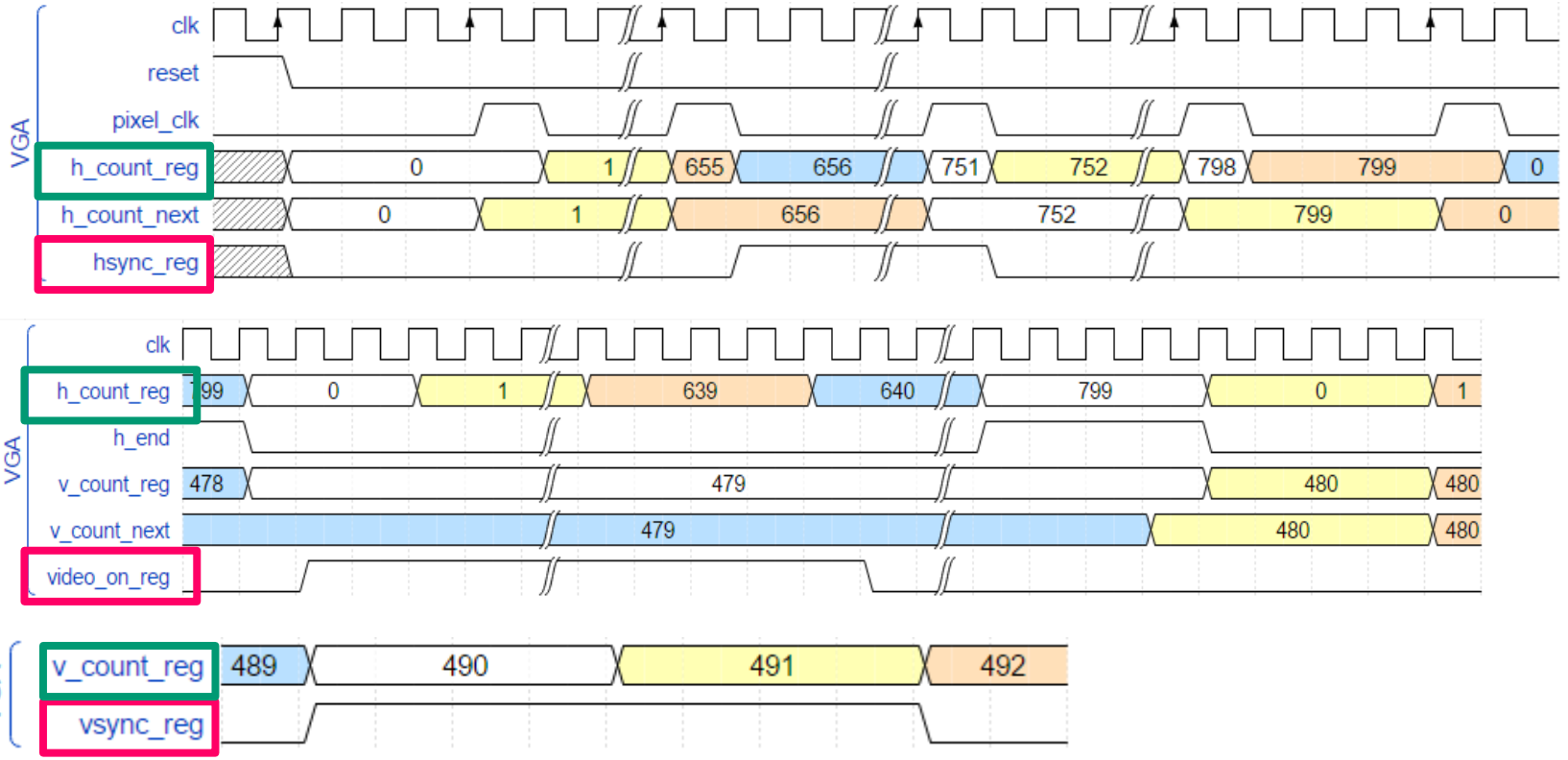
```
// mod-4 counter
wire  [1:0] mod4_next;
reg [1:0] mod4_reg;

always @(posedge  clk)
   if (reset)
      mod4_reg<= 1'b0;
   else
      mod4_reg <= mod4_next;

assign mod4_next = mod4_reg + 1 ;
assign  pixel_clk = (mod4_reg == 2'b11) ? 1 : 0;
```



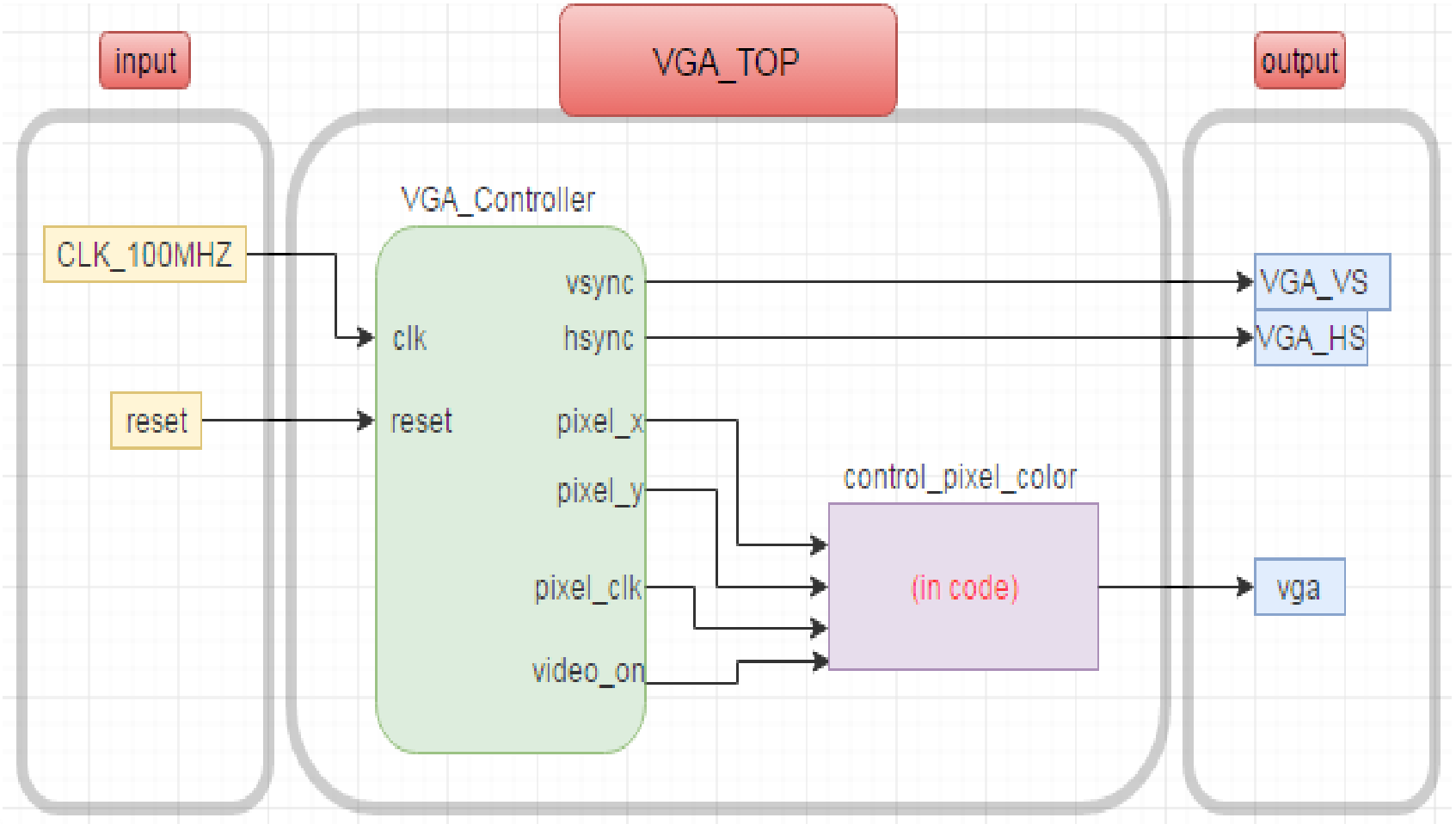


VGA Signal Timing

- Hsync-controller is based on mod-800 counter
- Vsync-controller is based on mod-525 counter



VGA 실습


input
VGA_TOP
output
VGA_Controller
CLK_100MHZ
reset
clk
reset
vsync
hsync
pixel_x
pixel_y
pixel_clk
video_on
VGA_VS
VGA_HS
control_pixel_color
(in code)
vga

실습 code (Top-Level)

Display a red box on a screen

```
    always @*
    begin
        vga_next = vga_reg;
        if(~video_on)
            vga_next = COLOR_BLACK;
        else
        begin
            if(block_on)
                vga_next = COLOR_RED;
            else
                vga_next =  COLOR_WHITE;
        end
    end

    assign block_on = (BLOCK_S_X <= pixel_x ) && (pixel_x < BLOCK_S_X + BLOCK_WIDTH) &&
(BLOCK_S_Y <= pixel_y) && (pixel_y < BLOCK_S_Y + BLOCK_HEIGHT);
    assign vga = vga_reg;
endmodule
```

```
module VGA_top(
    input CLK100MHZ, reset,
    output VGA_HS, VGA_VS,
    output [11:0] vga
    );

    localparam COLOR_WHITE  = 12'b1111_1111_1111;
    localparam COLOR_RED = 12'b0000_0000_1111;
    localparam COLOR_BLACK  = 12'b0000_0000_0000;

    localparam BLOCK_WIDTH  = 100;
    localparam BLOCK_HEIGHT = 80;

    localparam BLOCK_S_X    = 100;
    localparam BLOCK_S_Y    = 200;

    wire video_on,  block_on;
    wire pixel_clk;
    wire [9:0] pixel_x, pixel_y;
    reg [11:0] vga_next, vga_reg;
    VGA_controller VGA_controller_1(
        .clk(CLK100MHZ), .reset(reset),
        .hsync(VGA_HS), .vsync(VGA_VS),
        .video_on(video_on), .pixel_clk(pixel_clk),
        .pixel_x(pixel_x), .pixel_y(pixel_y)
    );
    always @(posedge CLK100MHZ, posedge reset)
    begin
        if(reset)
            vga_reg <= 12'd0;
        else
            if(pixel_clk)
                vga_reg <= vga_next;
    end
```

```
    always @*
    begin
        vga_next = vga_reg;
        if(~video_on)
            vga_next = COLOR_BLACK;
        else
        begin
            if(block_on)
                vga_next = COLOR_RED;
            else
                vga_next =  COLOR_WHITE;
        end
    end

    assign block_on = (BLOCK_S_X <= pixel_x
) && (pixel_x < BLOCK_S_X + BLOCK_WIDTH) &&
(BLOCK_S_Y <= pixel_y) && (pixel_y <
BLOCK_S_Y + BLOCK_HEIGHT);
    assign vga = vga_reg;
endmodule
```

VGA drive PIN _ OUTPUTs

```
154 ##VGA Connector
155
156 set_property -dict { PACKAGE_PIN A3    IOSTANDARD LVCMOS33 } [get_ports { vga[0] }]; #IO_L8N_T1_AD14N_35 Sch=vga_r[0]
157 set_property -dict { PACKAGE_PIN B4    IOSTANDARD LVCMOS33 } [get_ports { vga[1] }]; #IO_L7N_T1_AD6N_35 Sch=vga_r[1]
158 set_property -dict { PACKAGE_PIN C5    IOSTANDARD LVCMOS33 } [get_ports { vga[2] }]; #IO_L1N_T0_AD4N_35 Sch=vga_r[2]
159 set_property -dict { PACKAGE_PIN A4    IOSTANDARD LVCMOS33 } [get_ports { vga[3] }]; #IO_L8P_T1_AD14P_35 Sch=vga_r[3]
160
161 set_property -dict { PACKAGE_PIN C6    IOSTANDARD LVCMOS33 } [get_ports { vga[4] }]; #IO_L1P_T0_AD4P_35 Sch=vga_g[0]
162 set_property -dict { PACKAGE_PIN A5    IOSTANDARD LVCMOS33 } [get_ports { vga[5] }]; #IO_L3N_T0_DQS_AD5N_35 Sch=vga_g[1]
163 set_property -dict { PACKAGE_PIN B6    IOSTANDARD LVCMOS33 } [get_ports { vga[6] }]; #IO_L2N_T0_AD12N_35 Sch=vga_g[2]
164 set_property -dict { PACKAGE_PIN A6    IOSTANDARD LVCMOS33 } [get_ports { vga[7] }]; #IO_L3P_T0_DQS_AD5P_35 Sch=vga_g[3]
165
166 set_property -dict { PACKAGE_PIN B7    IOSTANDARD LVCMOS33 } [get_ports { vga[8] }]; #IO_L2P_T0_AD12P_35 Sch=vga_b[0]
167 set_property -dict { PACKAGE_PIN C7    IOSTANDARD LVCMOS33 } [get_ports { vga[9] }]; #IO_L4N_T0_35 Sch=vga_b[1]
168 set_property -dict { PACKAGE_PIN D7    IOSTANDARD LVCMOS33 } [get_ports { vga[10] }]; #IO_L6N_T0_VREF_35 Sch=vga_b[2]
169 set_property -dict { PACKAGE_PIN D8    IOSTANDARD LVCMOS33 } [get_ports { vga[11] }]; #IO_L4P_T0_35 Sch=vga_b[3]
170
171 set_property -dict { PACKAGE_PIN B11   IOSTANDARD LVCMOS33 } [get_ports { VGA_HS }]; #IO_L4P_T0_15 Sch=vga_hs
172 set_property -dict { PACKAGE_PIN B12   IOSTANDARD LVCMOS33 } [get_ports { VGA_VS }]; #IO_L3N_T0_DQS_AD1N_15 Sch=vga_vs
```

VGA drive PIN _ INPUTs

```
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { CLK100MHZ }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];

set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports { reset }]; #IO_L9P_T1_DQS_14 Sch=btnc
```

실습 과제물

- 출력 사각형의 색깔과 위치를 다음과 같이 변경하라.