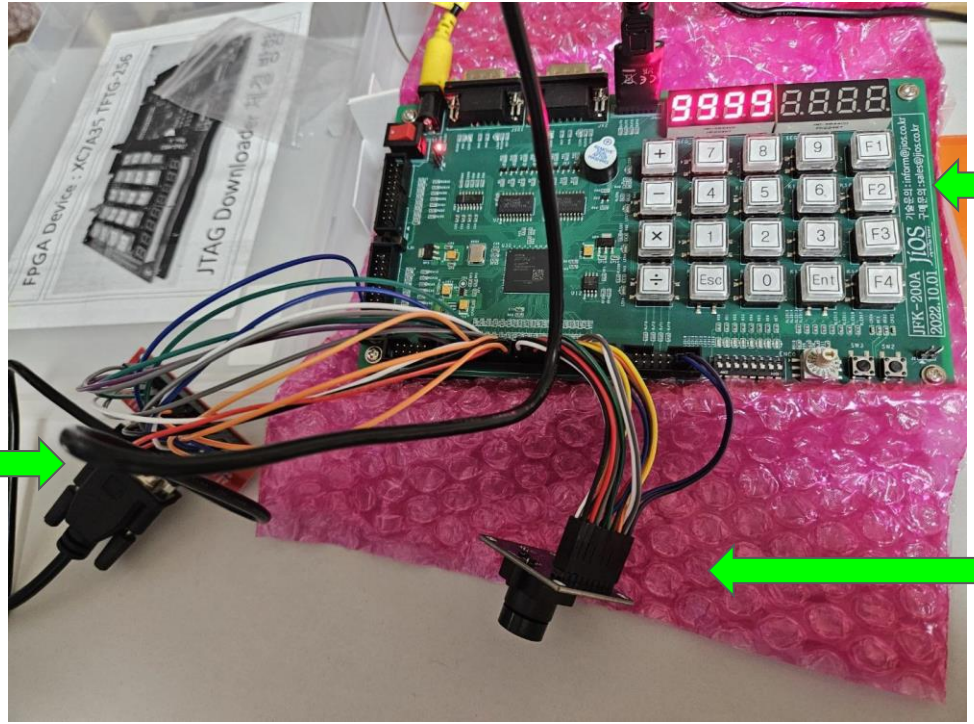
숭실대학교

FPGA를 이용한 실시간 영상 처리

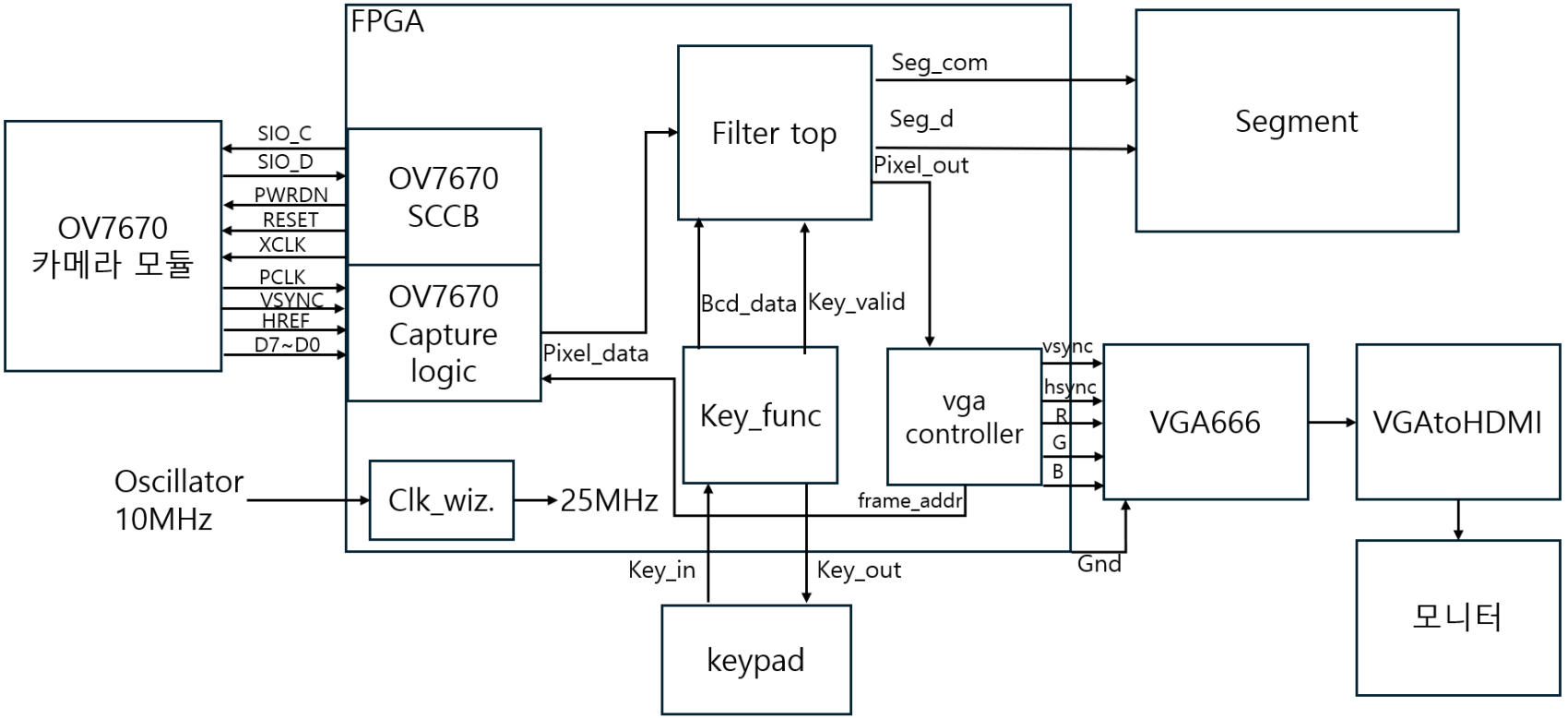
디지털 FPGA 설계

20192513 윤종민 20192537 정원욱

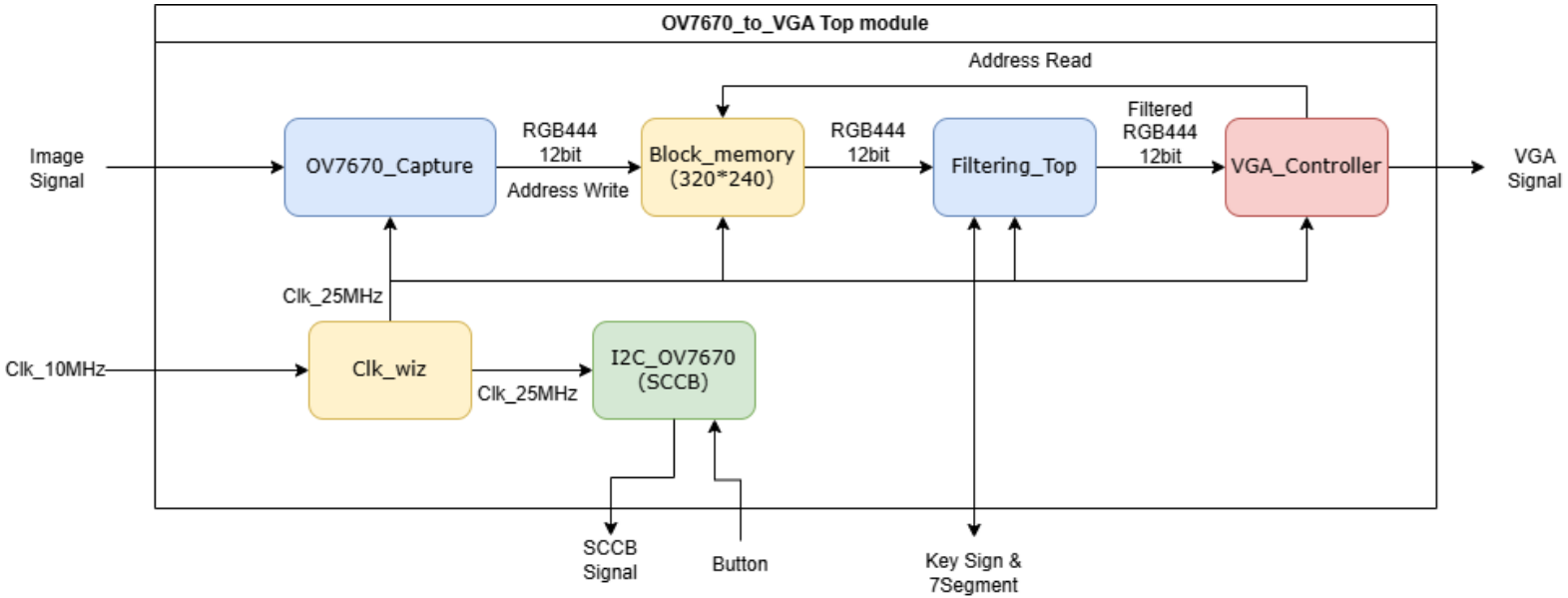
사용한 하드웨어



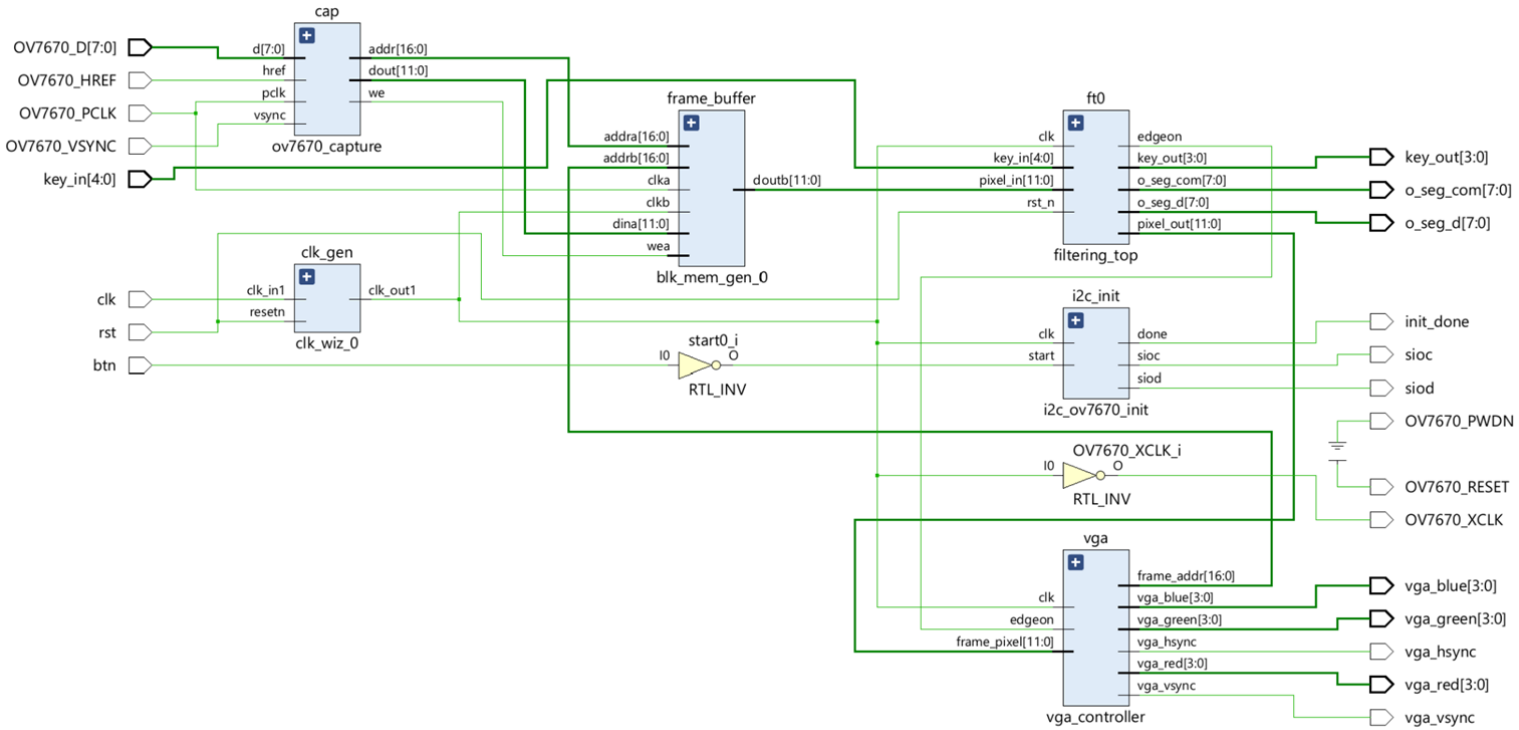
구조도


FPGA
OV7670
카메라 모듈
SIO_C
SIO_D
PWRDN
RESET
XCLK
PCLK
VSYNC
HREF
D7~D0
OV7670
SCCB
OV7670
Capture
logic
Filter top
Seg_com
Seg_d
Pixel_out
Segment
Bcd_data
Key_valid
Pixel_data
Key_func
vga
controller
vsync
hsync
R
G
B
VGA666
VGAtoHDMI
모니터
Oscillator
10MHz
Clk_wiz.
25MHz
frame_addr
Key_in
Key_out
Gnd
keypad

탑 모듈 설명


OV7670_to_VGA Top module
Address Read
Image Signal
OV7670_Capture
RGB444 12bit
Address Write
Block_memory (320*240)
RGB444 12bit
Filtering_Top
Filtered RGB444 12bit
VGA_Controller
VGA Signal
Clk_25MHz
Clk_10MHz
Clk_wiz
Clk_25MHz
I2C_OV7670 (SCCB)
SCCB Signal
Button
Key Sign & 7Segment

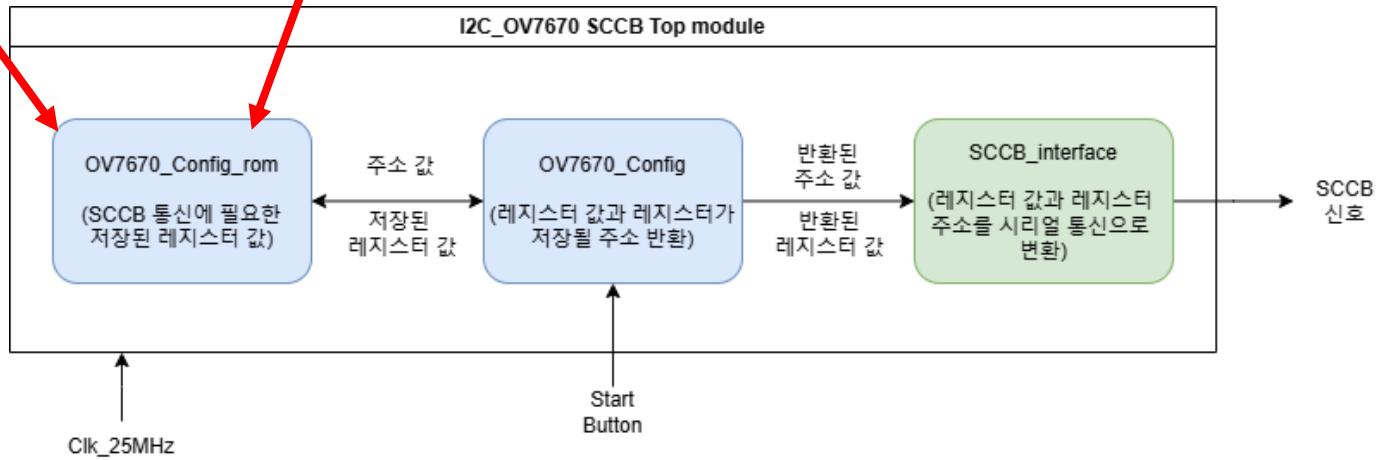
탑 모듈


cap
OV7670_D[7:0]
OV7670_HREF
OV7670_PCLK
OV7670_VSYNC
key_in[4:0]
d[7:0]
href
pclk
vsync
addr[16:0]
dout[11:0]
we
ov7670_capture
frame_buffer
addra[16:0]
addrb[16:0]
clka
clkb
dina[11:0]
wea
doutb[11:0]
blk_mem_gen_0
ft0
clk
key_in[4:0]
pixel_in[11:0]
rst_n
edgeon
key_out[3:0]
o_seg_com[7:0]
o_seg_d[7:0]
pixel_out[11:0]
filtering_top
key_out[3:0]
o_seg_com[7:0]
o_seg_d[7:0]
clk_gen
clk
rst
btn
clk_in1
resetn
clk_out1
clk_wiz_0
start0_i
I0
O
RTL_INV
i2c_init
clk
start
done
sioc
siod
i2c_ov7670_init
init_done
sioc
siod
OV7670_PWDN
OV7670_XCLK_i
I0
O
RTL_INV
OV7670_RESET
OV7670_XCLK
vga
clk
edgeon
frame_pixel[11:0]
frame_addr[16:0]
vga_blue[3:0]
vga_green[3:0]
vga_hsync
vga_red[3:0]
vga_vsync
vga_controller
vga_blue[3:0]
vga_green[3:0]
vga_hsync
vga_red[3:0]
vga_vsync

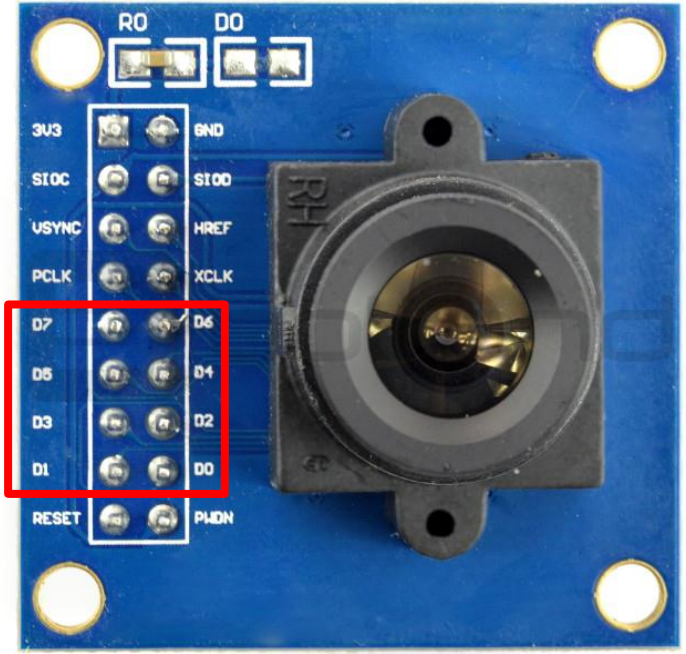
SCCB 인터페이스

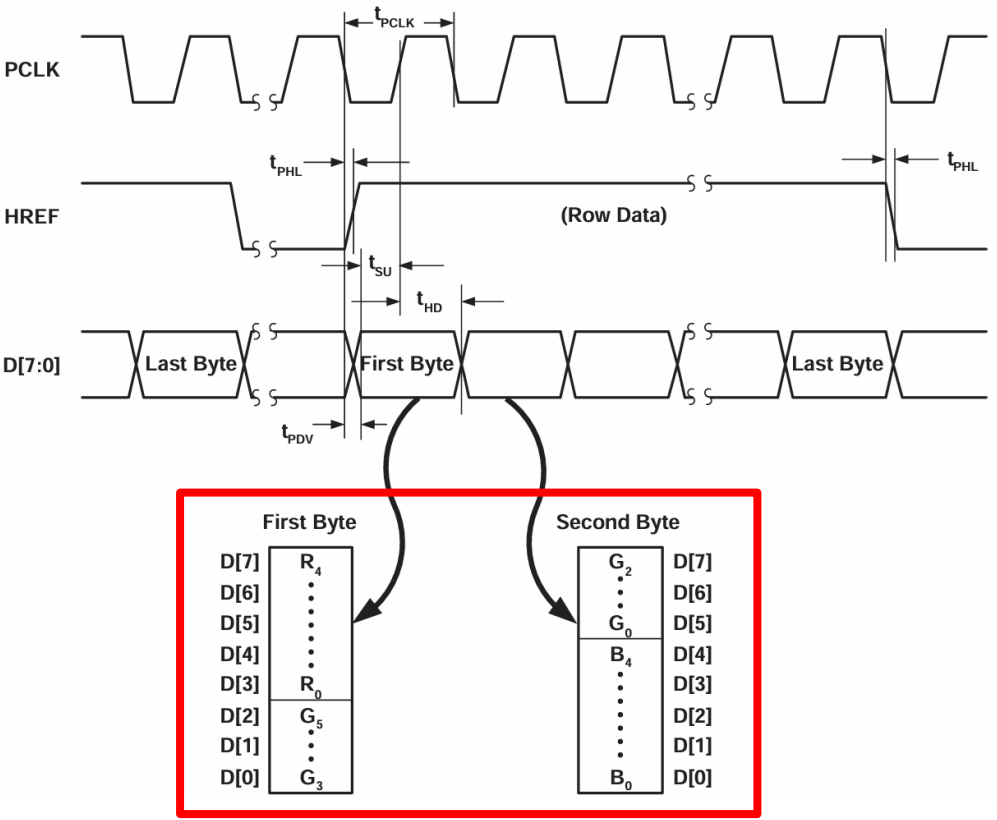
01	BLUE	80	RW	AWB – Blue channel gain setting • Range: [00] to [FF]
02	RED	80	RW	AWB – Red channel gain setting • Range: [00] to [FF]

12	COM7	00	RW	Common Control 7 Bit[7]: SCCB Register Reset 0: No change 1: Resets all registers to default values Bit[6]: Reserved Bit[5]: Output format - CIF selection Bit[4]: Output format - QVGA selection Bit[3]: Output format - QCIF selection Bit[2]: Output format - RGB selection (see below) Bit[1]: Color bar 0: Disable 1: Enable Bit[0]: Output format - Raw RGB (see below) COM7[2] COM7[0] YUV 0 0 RGB 1 0 Bayer RAW 0 1 Processed Bayer RAW 1 1



OV7670 Capture 동작



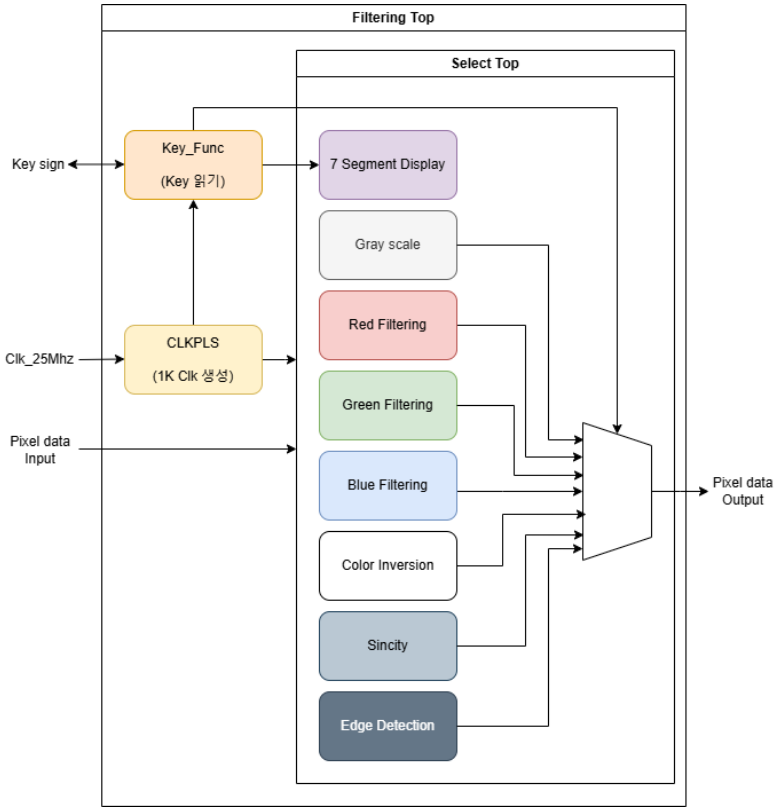


Filtering Top 모듈 구성

각 필터링 모듈
+필터링 모듈끼리 합성한 Data

=총 11개의 Mode

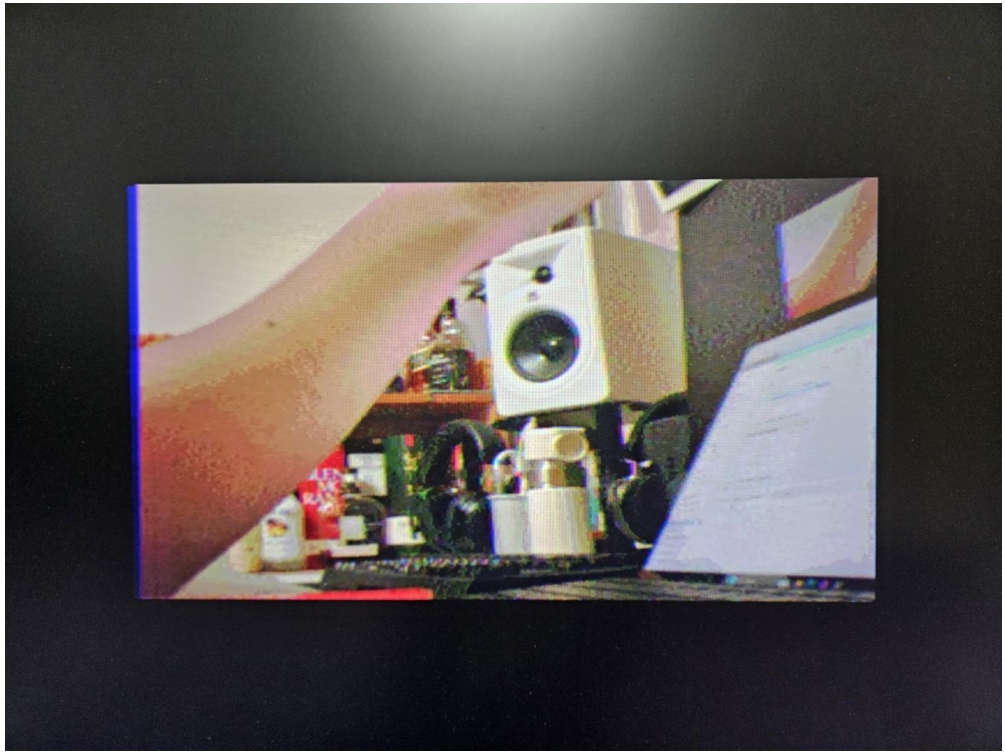
Key pad로 제어 가능



오리지널 영상

320x240 크기의 image
각 픽셀은 12bit로 구성
Red 4bit, Green 4bit, Blue 4bit

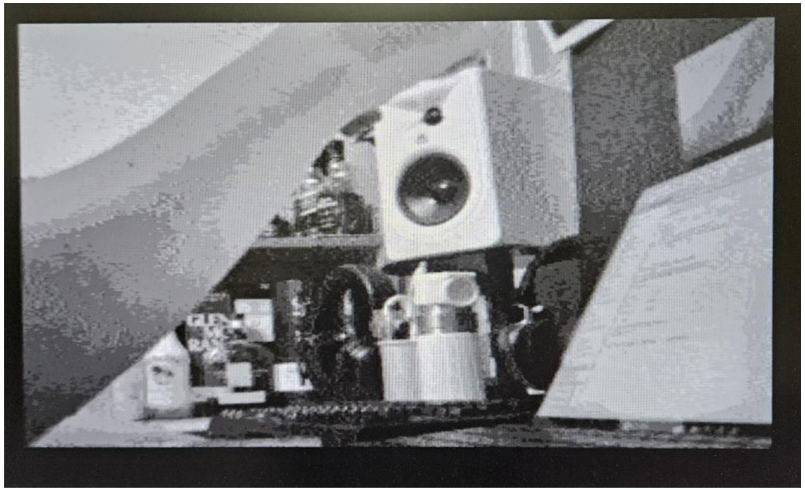
```
{vga_red, vga_green, vga_blue} <= frame_pixel;
```



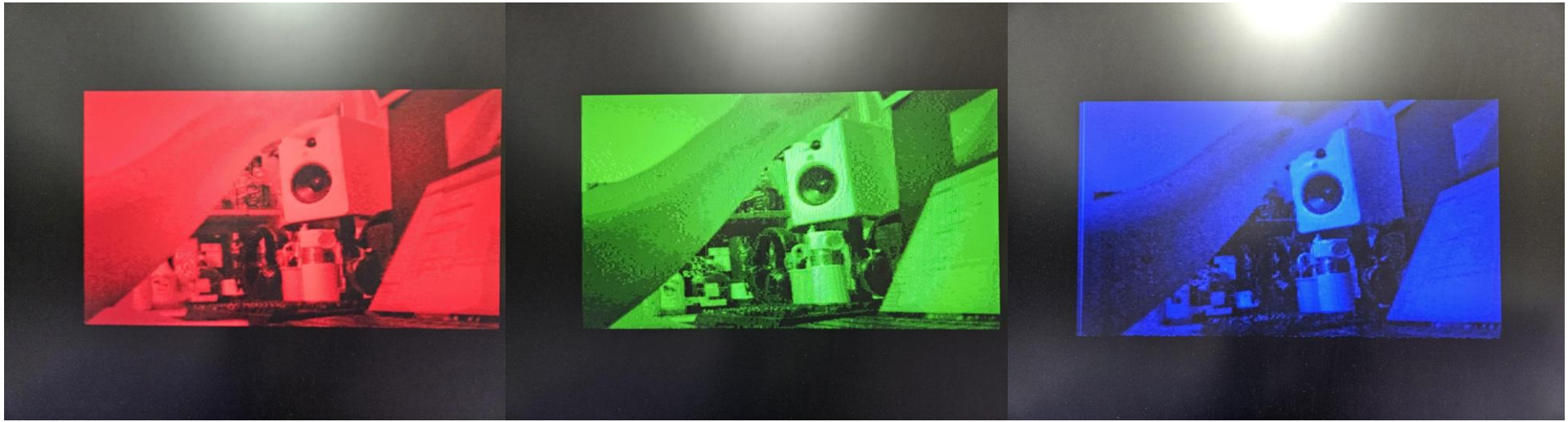
Grayscale

$$Y = 0.299R + 0.587G + 0.114B$$

```
assign grayscale = (299*pixel_in[11:8] + 587*pixel_in[7:4] + 114*pixel_in[3:0])/1000;
```



RGB 필터1

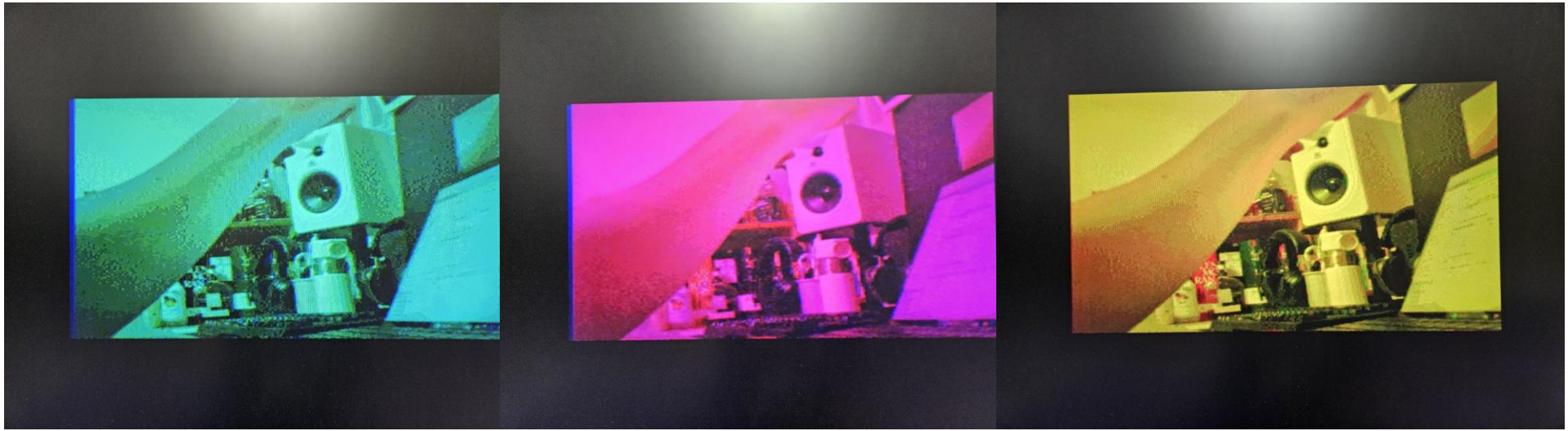


```
assign redfiltered = pixel_in[11:8];//red
```

```
assign greenfiltered = pixel_in[7:4];
```

```
assign bluefiltered = pixel_in[3:0];
```

RGB 필터2



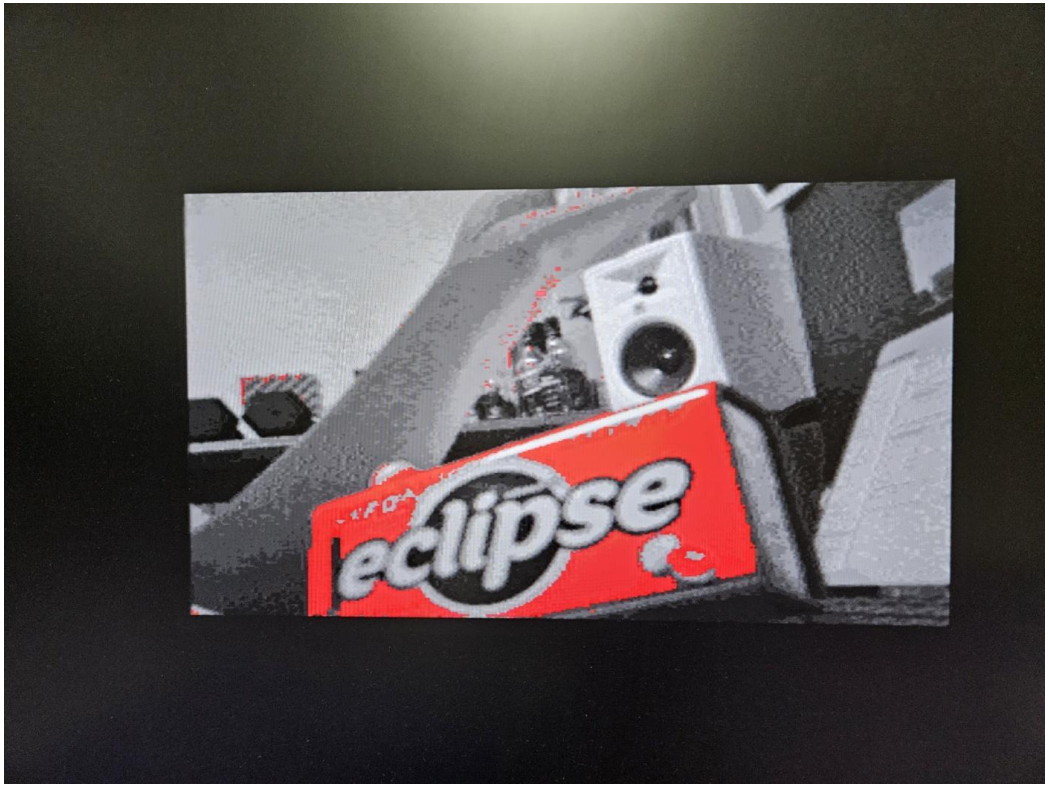
Red filtered (red만 제거)

Green filtered(Green만 제거)

Blue filtered(Blue만 제거)

Grayscale + red

기본적으로 Grayscale에
빨간색이 임계값보다 크게
보이면 Red만 보이게 처리




```
assign pixel_out = ((pixel_in[11:8]> 4'b1010) && (pixel_in[7:4]< 4'b1000)&&(pixel_in[3:0]< 4'b1000)) ?
                {pixel_in[11:8], 4'h0, 4'h0}:
                {pixel_ready, pixel_ready, pixel_ready};
```

색상 반전

```
assign colorinversed = ~pixel_in;
```

Edge detection

이미지의 edge를 감지하여 표시

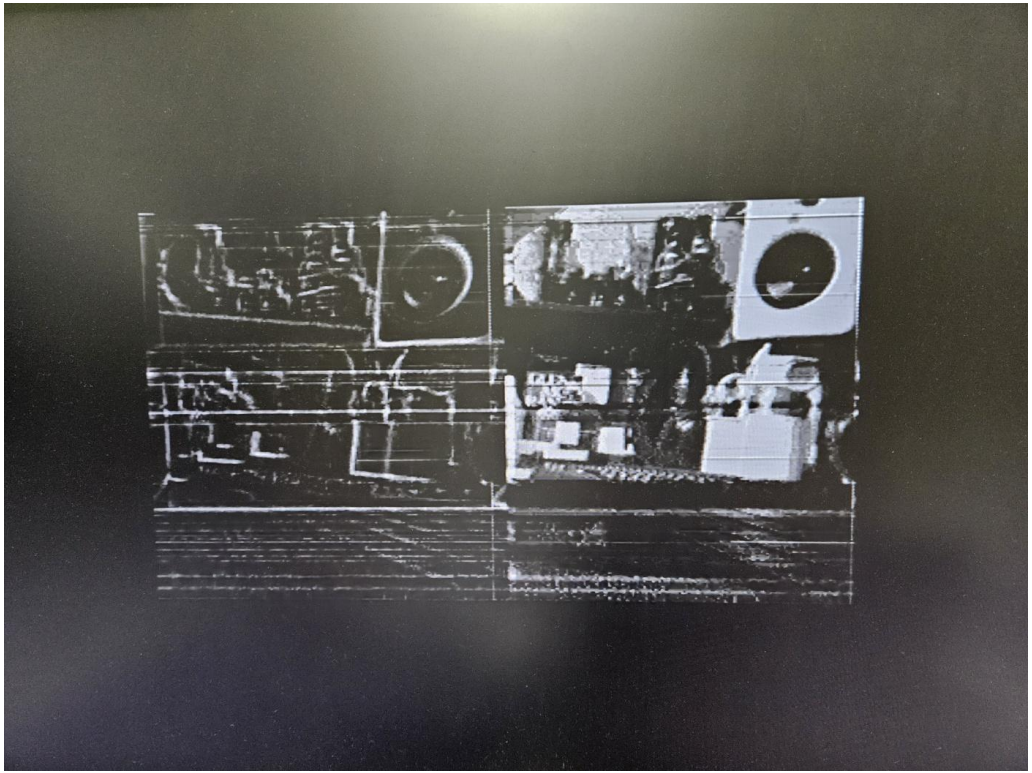
x축 방향의 gradient와 y축 방향의 gradient를 구하여 합침

-1	0	1
-2	0	2
-1	0	1

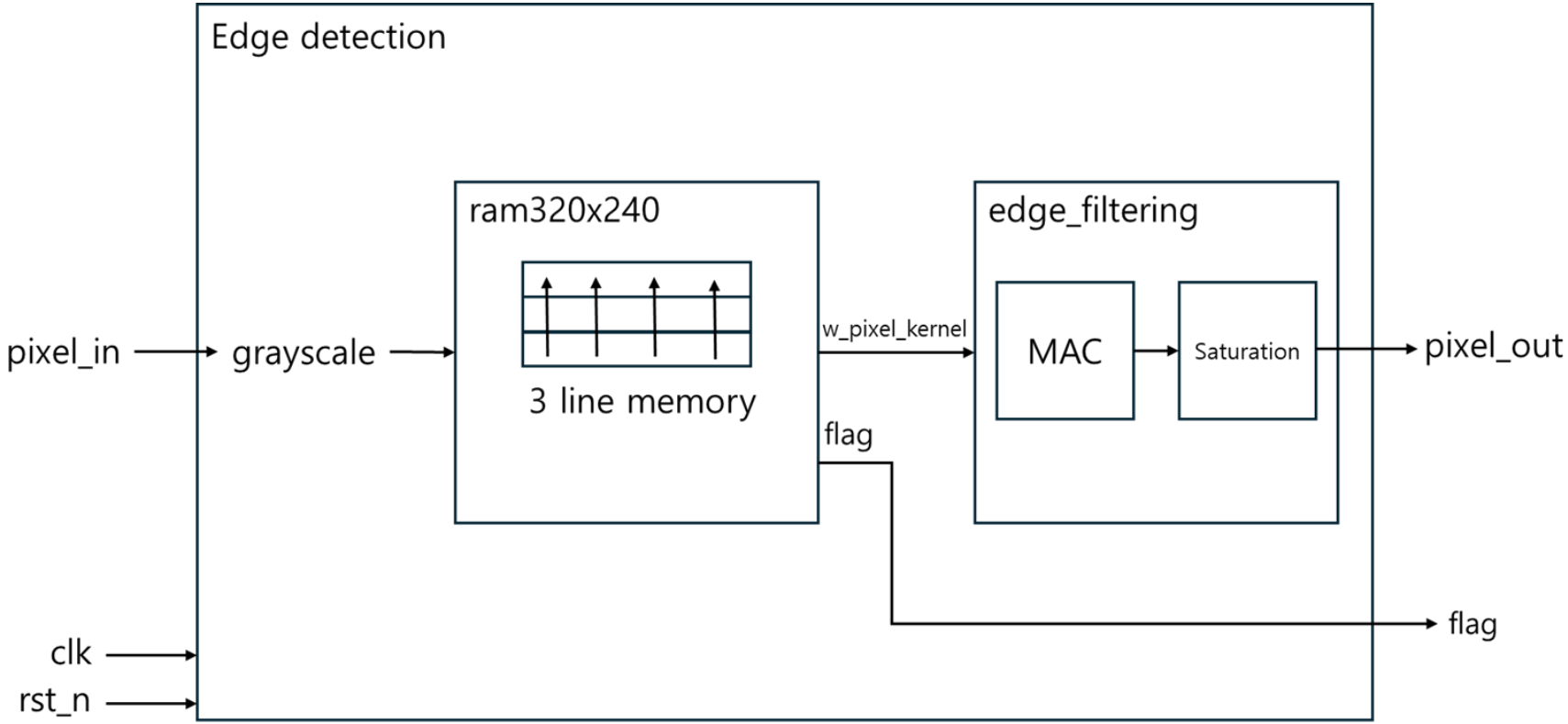
x축 방향

-1	-2	-1
0	0	0
1	2	1

y축 방향

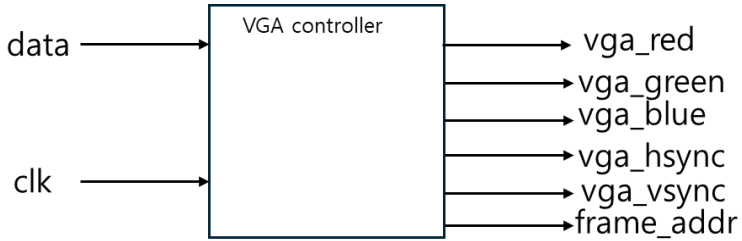


Edge detection



VGA666

VGA666은 라즈베리파이에 VGA출력을 추가하는 모듈이지만, 3.3V로 작동하는 GPIO의 개수만 충분하면, FPGA보드에서도 사용가능하다.



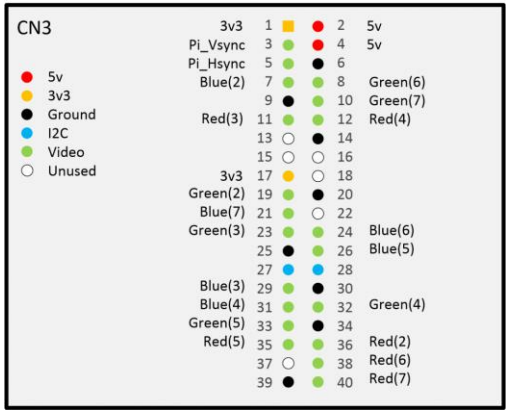
Hsync: 화면의 가로 1줄이 완성되면 1

Vsync: 화면이 전부 완성되면 1

Frame_addr: 메모리에 저장된 픽셀 데이터값 요청

```
always @(posedge clk) begin
    // 수평 카운터 증가
    if (h_count < 799)
        h_count <= h_count + 1;
    else begin
        h_count <= 0;
        // 수직 카운터 증가
        if (v_count < 524)
            v_count <= v_count + 1;
        else
            v_count <= 0;
    end
```

Gert VGA 666



감사합니다