HW4 作業說明

郭昱 ivan010517@gmail.com
黃宥翔 chris900623@gmail.com

2024 HDL

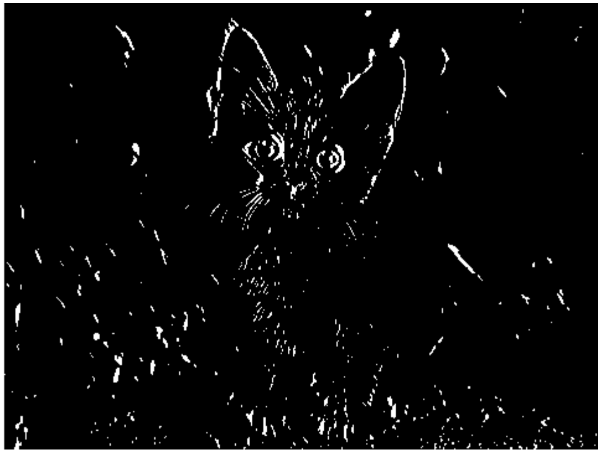
Outline

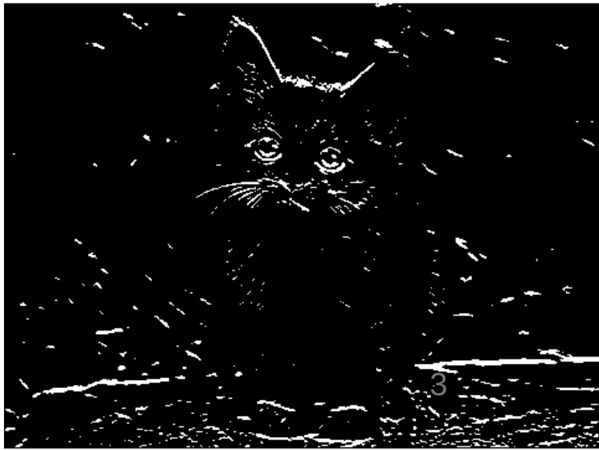
- Sobel Edge Operation
    - Color Space Transformation
    - Convolution with Sobel Filters
- Hardware Implement
    - Read Image(.bmp) File - Testbench
    - Zero Padding - Testbench
    - RGB to YUV - Hardware
    - Line Buffer & Convolution - Hardware
    - Write Image(.bmp) File - Testbench

Sobel Edge Operation

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



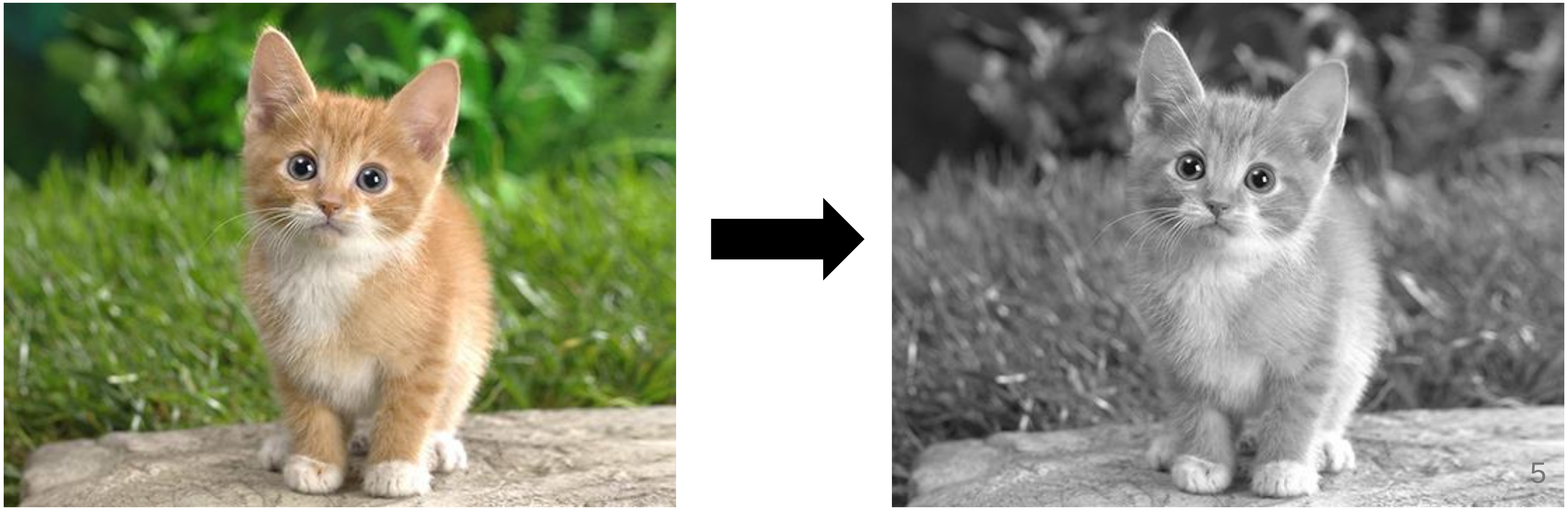


Sobel Edge Operation

- Color Space Transformation
- Convolution with Sobel Filters
- Pixel to Binary Transformation by Threshold

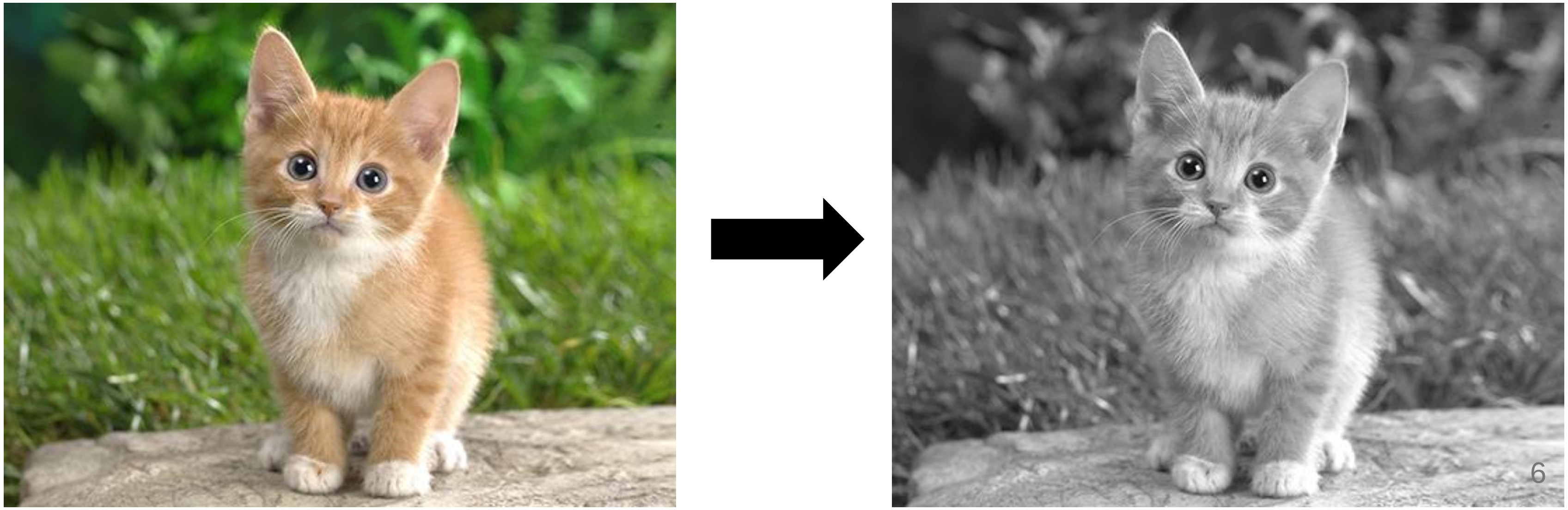
Color Space Transformation

- Color Space Transformation
- Convolution with Sobel Filters
- Pixel to Binary Transformation by Threshold



Color Space Transformation

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



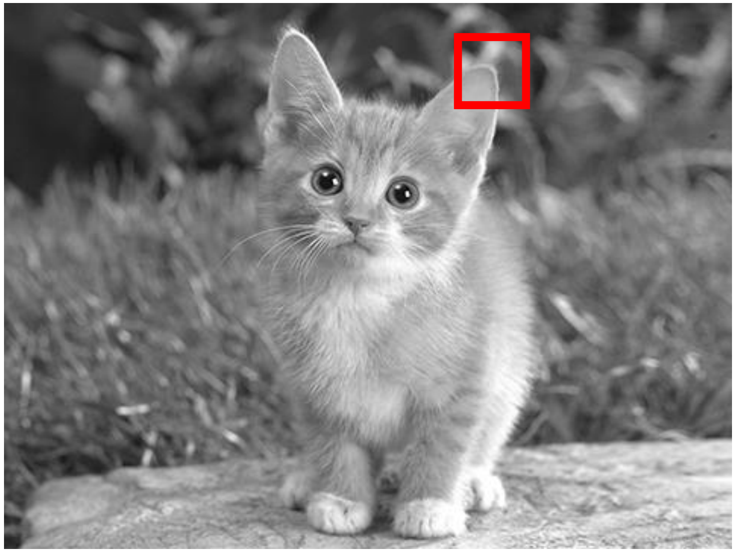
Color Space Transformation

- Color Space Transformation
- Convolution with Sobel Filters
- Pixel to Binary Transformation by Threshold

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Convolution with Sobel Filters

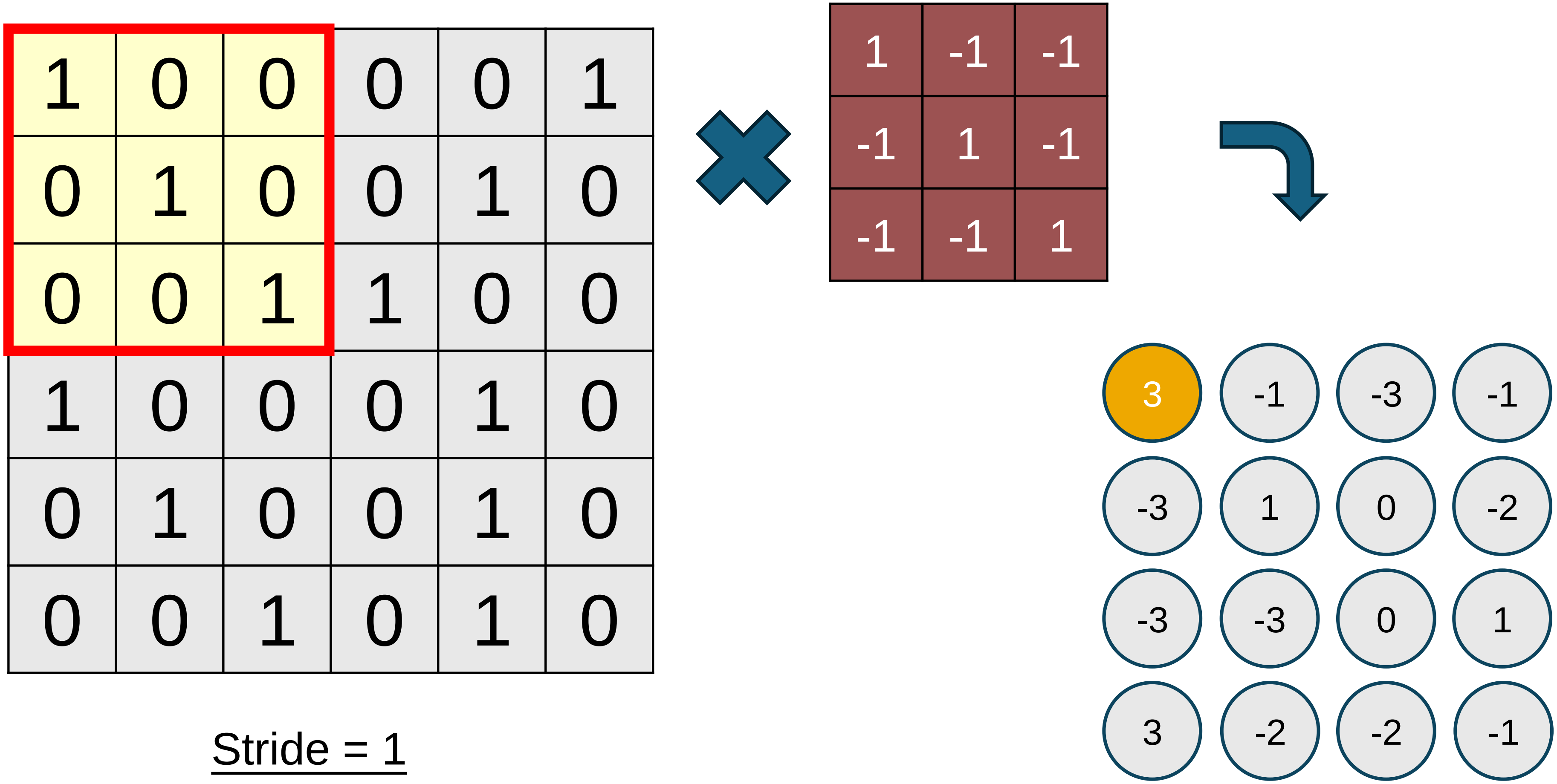
- Color Space Transformation
- Convolution with Sobel Filters
- Pixel to Binary Transformation by Threshold



✖

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix},$$

Convolution with Sobel Filters



Input image (6 * 6)

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Filter(3 * 3)

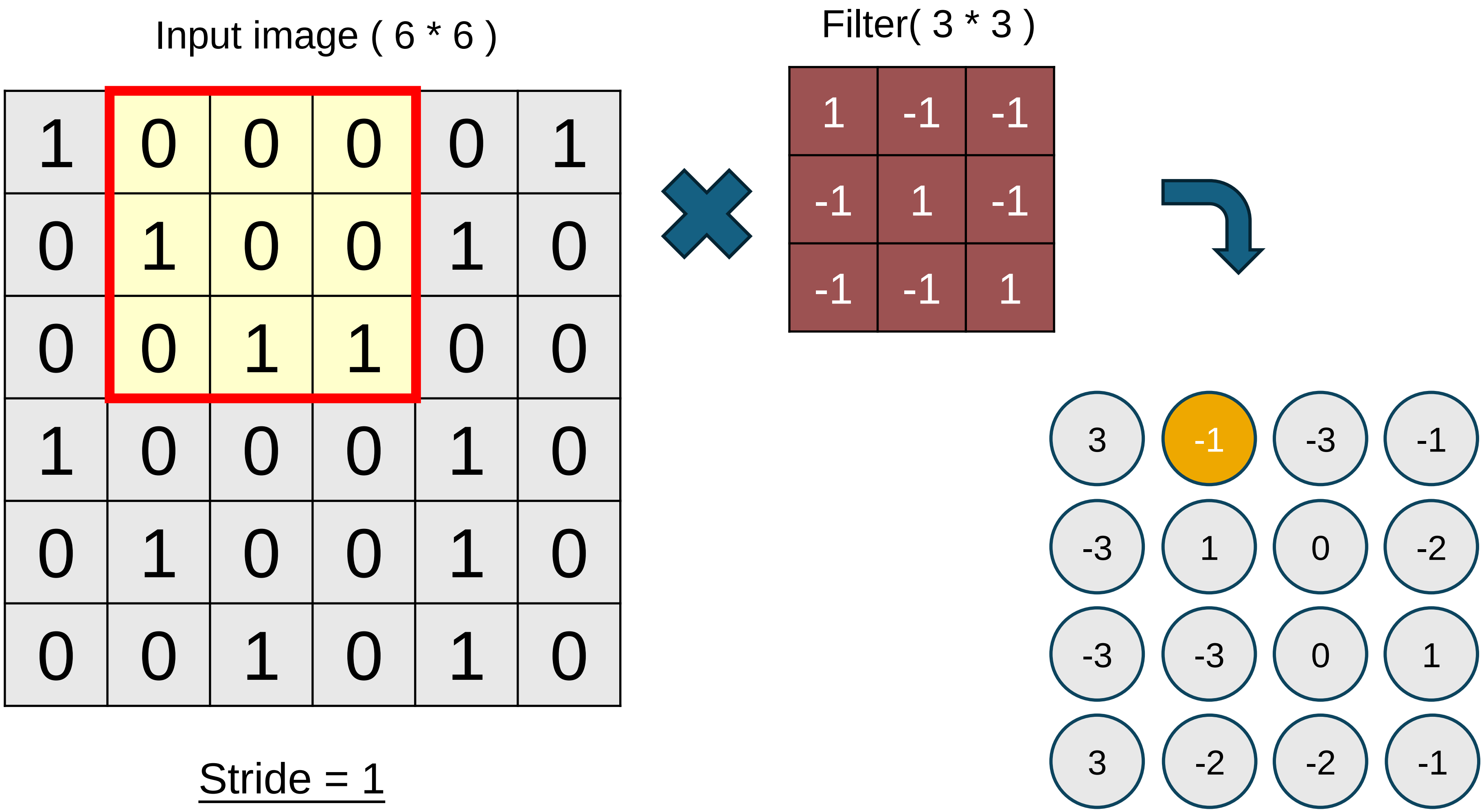
1	-1	-1
-1	1	-1
-1	-1	1

3	-1	-3	-1
-3	1	0	-2
-3	-3	0	1
3	-2	-2	-1

Stride = 1

Output image (4 * 4)

Convolution with Sobel Filters



Input image (6 * 6)

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Filter(3 * 3)

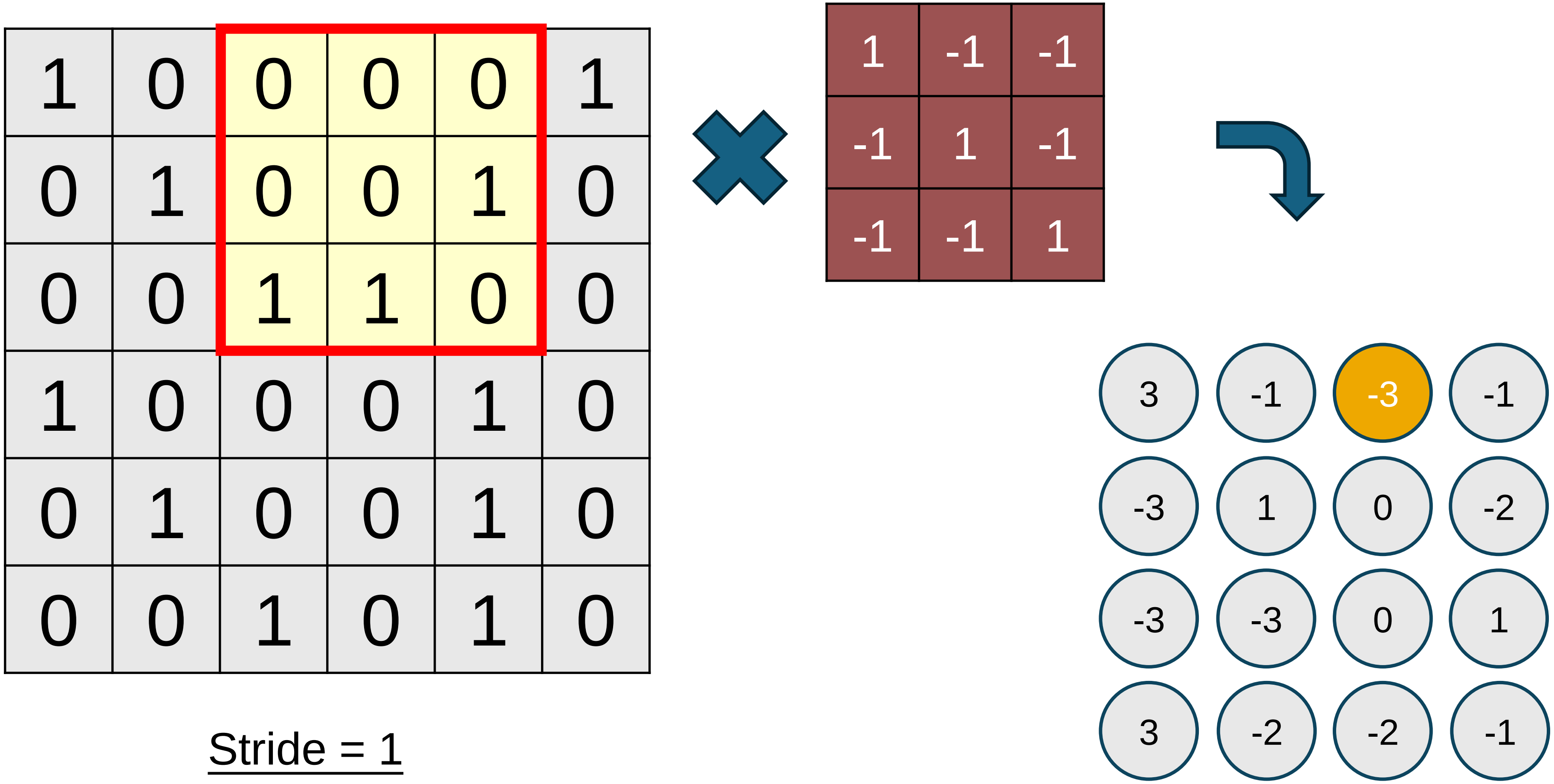
1	-1	-1
-1	1	-1
-1	-1	1

3	-1	-3	-1
-3	1	0	-2
-3	-3	0	1
3	-2	-2	-1

Stride = 1

Output image (4 * 4)

Convolution with Sobel Filters



Input image (6 * 6)

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Filter(3 * 3)

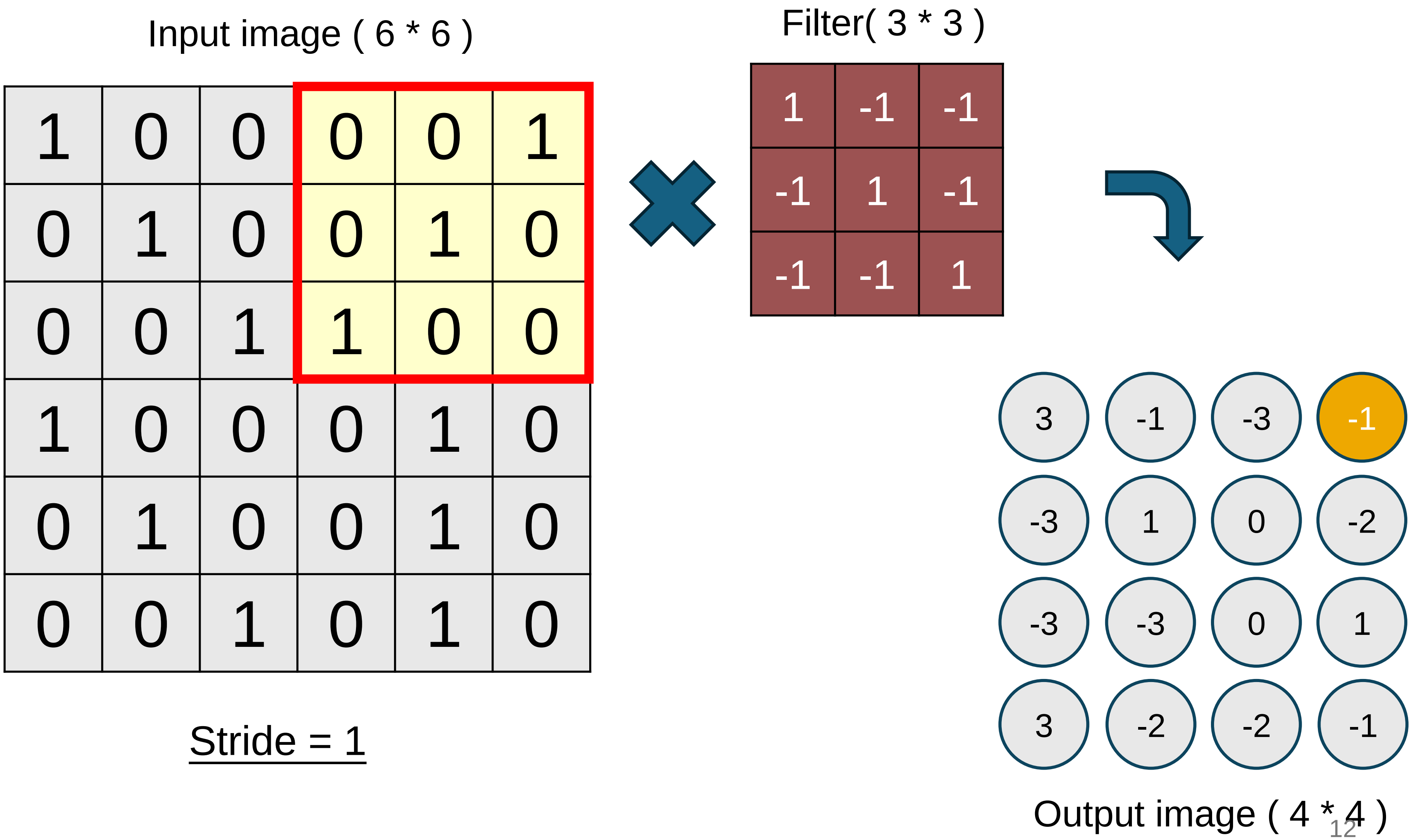
1	-1	-1
-1	1	-1
-1	-1	1

3	-1	-3	-1
-3	1	0	-2
-3	-3	0	1
3	-2	-2	-1

Stride = 1

Output image (4 * 4)

Convolution with Sobel Filters



Input image (6 * 6)

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Stride = 1

Filter(3 * 3)

1	-1	-1
-1	1	-1
-1	-1	1

3	-1	-3	-1
-3	1	0	-2
-3	-3	0	1
3	-2	-2	-1

Output image (4 * 4)

Zero Padding

Input image (6 * 6)

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Stride = 1

Filter(3 * 3)

1	-1	-1
-1	1	-1
-1	-1	1

3	-1	-3	-1
-3	1	0	-2
-3	-3	0	1
3	-2	-2	-1

Output image (4 * 4)

Zero Padding

Input image (6 * 6)

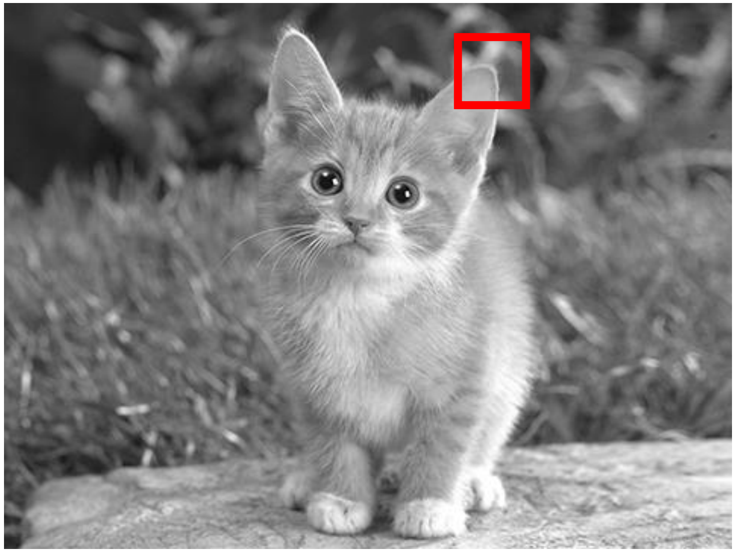
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Input image (8 * 8)

0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0
0	1	0	0	0	1	0	0
0	0	1	0	0	1	0	0
0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0

Convolution with Sobel Filters

- Color Space Transformation
- Convolution with Sobel Filters
- Pixel to Binary Transformation by Threshold



×

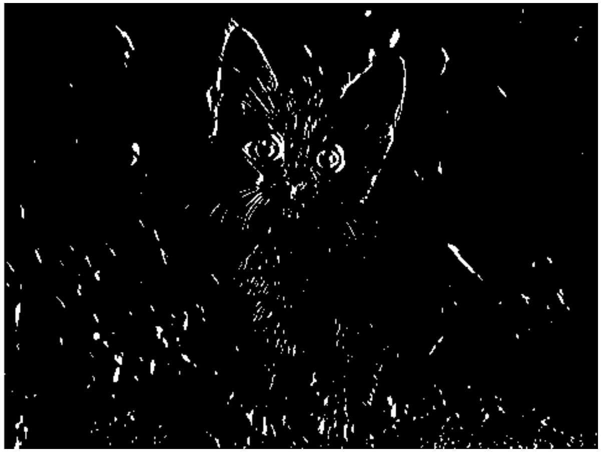
$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix},$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

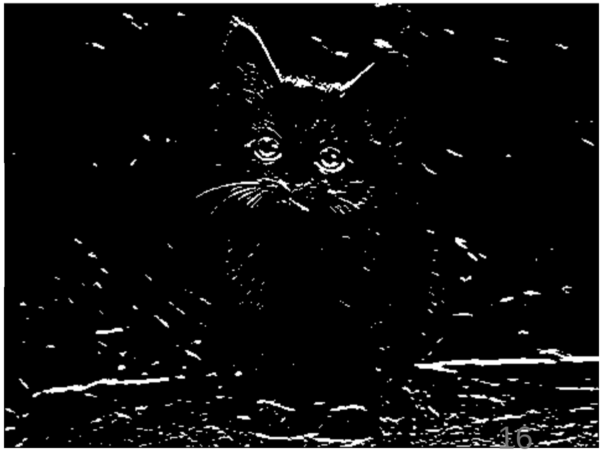
Pixel to Binary Transformation by Threshold

- Color Space Transformation
- Convolution with Sobel Filters
- Pixel to Binary Transformation by Threshold

G_x



G_y



Pixel to Binary Transformation by Threshold

0 ~ 255

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix},$$

-1020
~
+1020

?

Pixel to Binary Transformation by Threshold

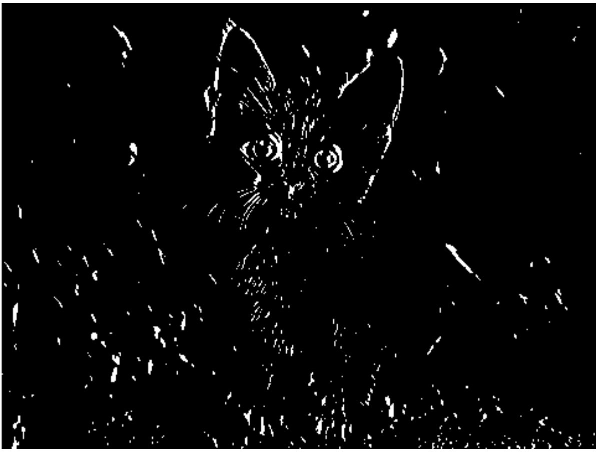
0 ~ 255

convolution

?

-1020
~
+1020

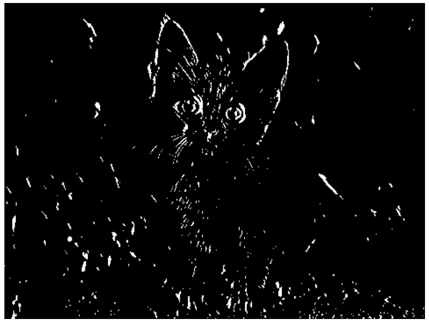
Compare with Threshold
Pixel Value > Threshold → 255
Pixel Value ≤ Threshold → 0

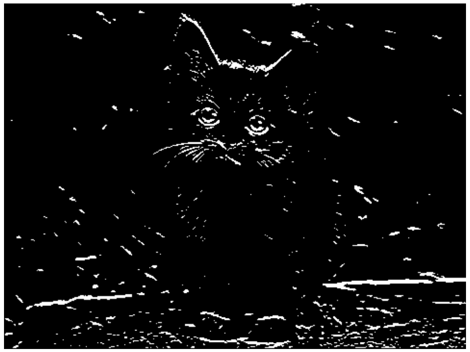


0 or 255
Only white or black

Sobel Edge Operation

- Color Space Transformation
- Convolution with Sobel Filters
- Pixel to Binary Transformation by Threshold

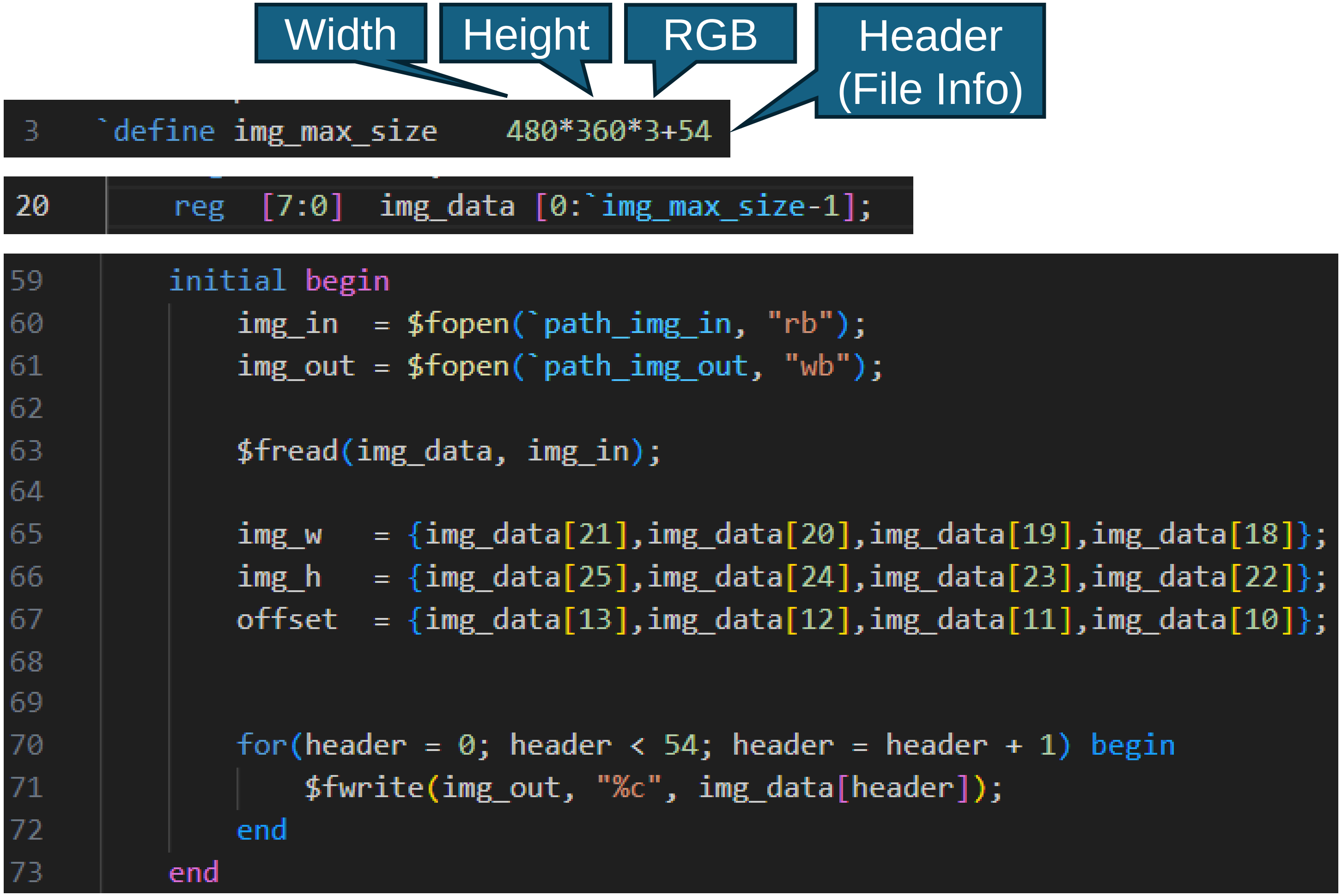




Outline

- Hardware Implement
  - Read Image(.bmp) File - Testbench
  - Zero Padding - Testbench
  - RGB to YUV - Hardware
  - Line Buffer & Convolution - Hardware
  - Write Image(.bmp) File - Testbench

Testbench – Read Image



Width

Height

RGB

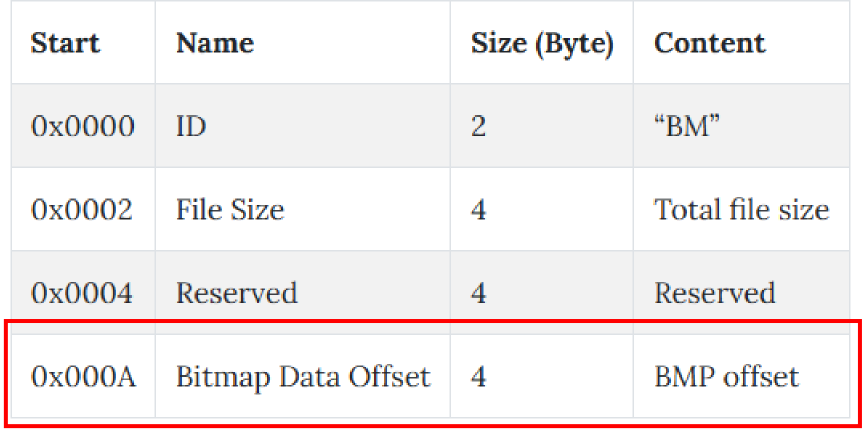
Header
(File Info)

```
`define img_max_size    480*360*3+54
```

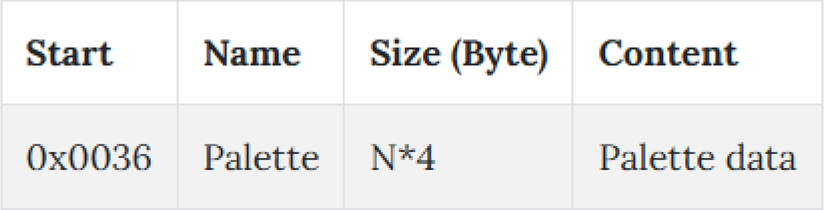
```
    reg  [7:0]  img_data [0:`img_max_size-1];
```

```
    initial begin
        img_in  = $fopen(`path_img_in, "rb");
        img_out = $fopen(`path_img_out, "wb");

        $fread(img_data, img_in);

        img_w   = {img_data[21],img_data[20],img_data[19],img_data[18]};
        img_h   = {img_data[25],img_data[24],img_data[23],img_data[22]};
        offset  = {img_data[13],img_data[12],img_data[11],img_data[10]};


        for(header = 0; header < 54; header = header + 1) begin
            $fwrite(img_out, "%c", img_data[header]);
        end
    end
```

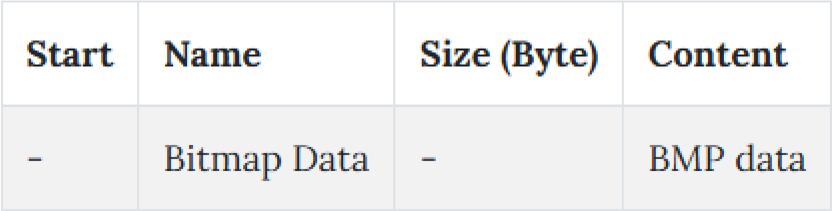
BMP File Format



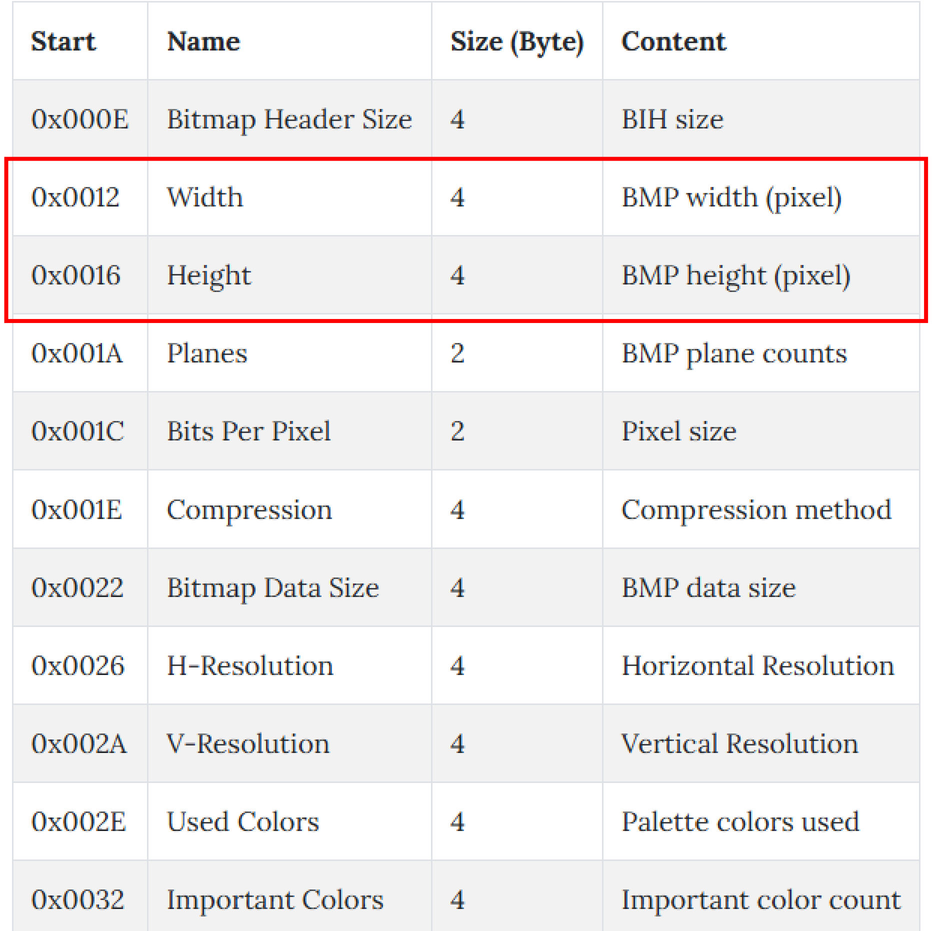
Start	Name	Size (Byte)	Content
0x0000	ID	2	“BM”
0x0002	File Size	4	Total file size
0x0004	Reserved	4	Reserved
0x000A	Bitmap Data Offset	4	BMP offset



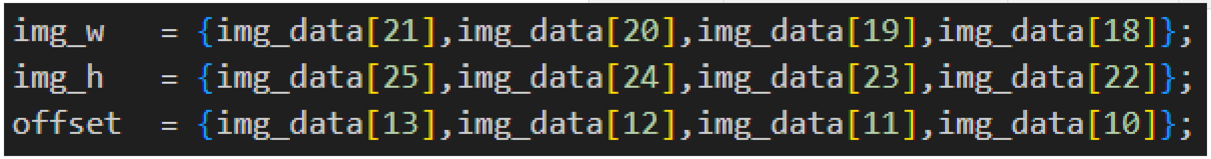
Start	Name	Size (Byte)	Content
0x0036	Palette	N*4	Palette data



Start	Name	Size (Byte)	Content
-	Bitmap Data	-	BMP data

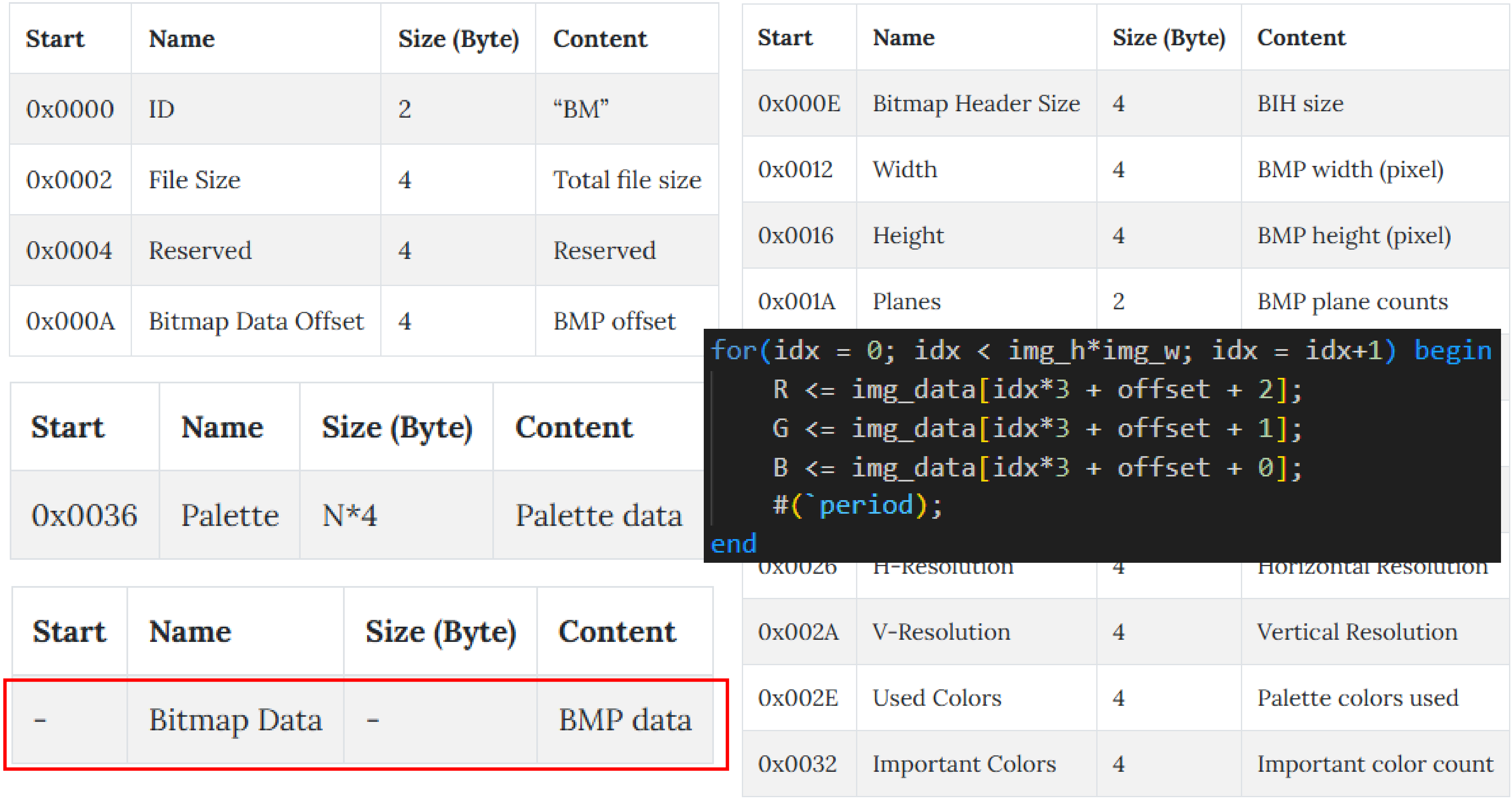


Start	Name	Size (Byte)	Content
0x000E	Bitmap Header Size	4	BIH size
0x0012	Width	4	BMP width (pixel)
0x0016	Height	4	BMP height (pixel)
0x001A	Planes	2	BMP plane counts
0x001C	Bits Per Pixel	2	Pixel size
0x001E	Compression	4	Compression method
0x0022	Bitmap Data Size	4	BMP data size
0x0026	H-Resolution	4	Horizontal Resolution
0x002A	V-Resolution	4	Vertical Resolution
0x002E	Used Colors	4	Palette colors used
0x0032	Important Colors	4	Important color count



```
img_w    = {img_data[21],img_data[20],img_data[19],img_data[18]};
img_h    = {img_data[25],img_data[24],img_data[23],img_data[22]};
offset   = {img_data[13],img_data[12],img_data[11],img_data[10]};
```

BMP File Format



Start	Name	Size (Byte)	Content
0x0000	ID	2	“BM”
0x0002	File Size	4	Total file size
0x0004	Reserved	4	Reserved
0x000A	Bitmap Data Offset	4	BMP offset

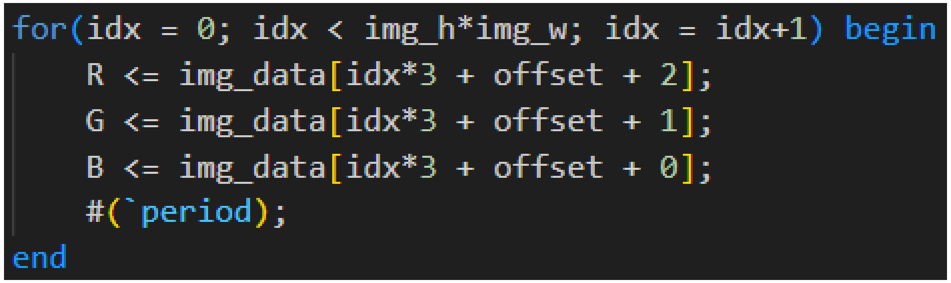
Start	Name	Size (Byte)	Content
0x0036	Palette	N*4	Palette data

Start	Name	Size (Byte)	Content
-	Bitmap Data	-	BMP data

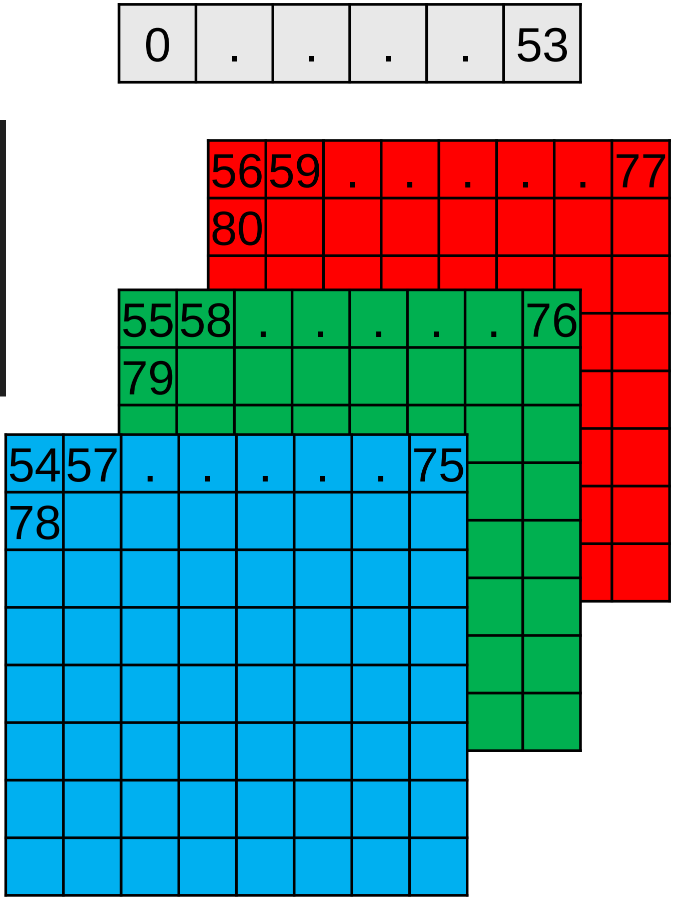
Start	Name	Size (Byte)	Content
0x000E	Bitmap Header Size	4	BIH size
0x0012	Width	4	BMP width (pixel)
0x0016	Height	4	BMP height (pixel)
0x001A	Planes	2	BMP plane counts
0x0026	H-Resolution	4	Horizontal Resolution
0x002A	V-Resolution	4	Vertical Resolution
0x002E	Used Colors	4	Palette colors used
0x0032	Important Colors	4	Important color count

```
for(idx = 0; idx < img_h*img_w; idx = idx+1) begin
    R <= img_data[idx*3 + offset + 2];
    G <= img_data[idx*3 + offset + 1];
    B <= img_data[idx*3 + offset + 0];
    #(`period);
end
```

BMP File Format



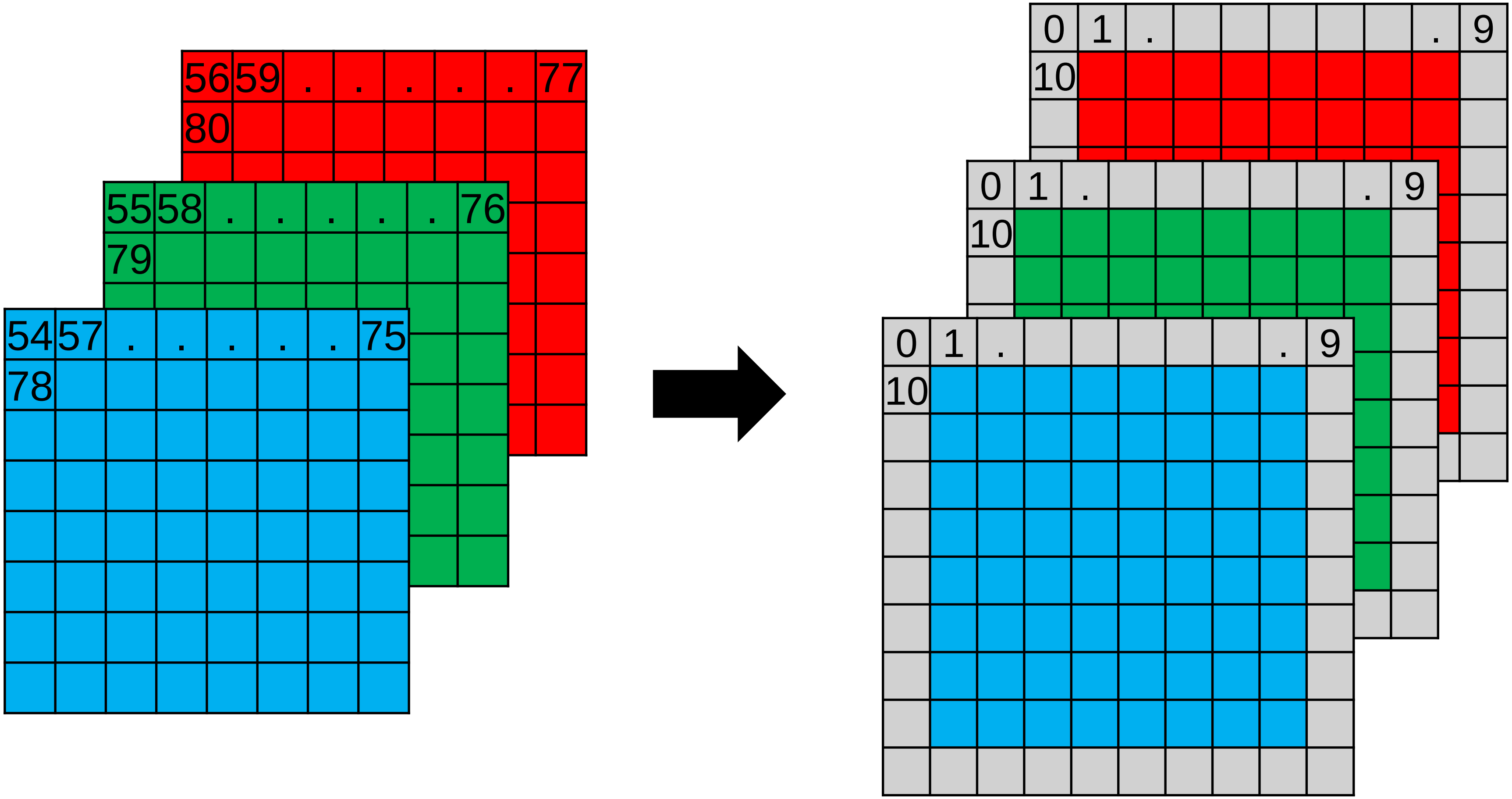
```
for(idx = 0; idx < img_h*img_w; idx = idx+1) begin
    R <= img_data[idx*3 + offset + 2];
    G <= img_data[idx*3 + offset + 1];
    B <= img_data[idx*3 + offset + 0];
    #(`period);
end
```



Outline

- Hardware Implement
  - Read Image(.bmp) File - Testbench
  - Zero Padding - Testbench
  - RGB to YUV - Hardware
  - Line Buffer & Convolution - Hardware
  - Write Image(.bmp) File - Testbench

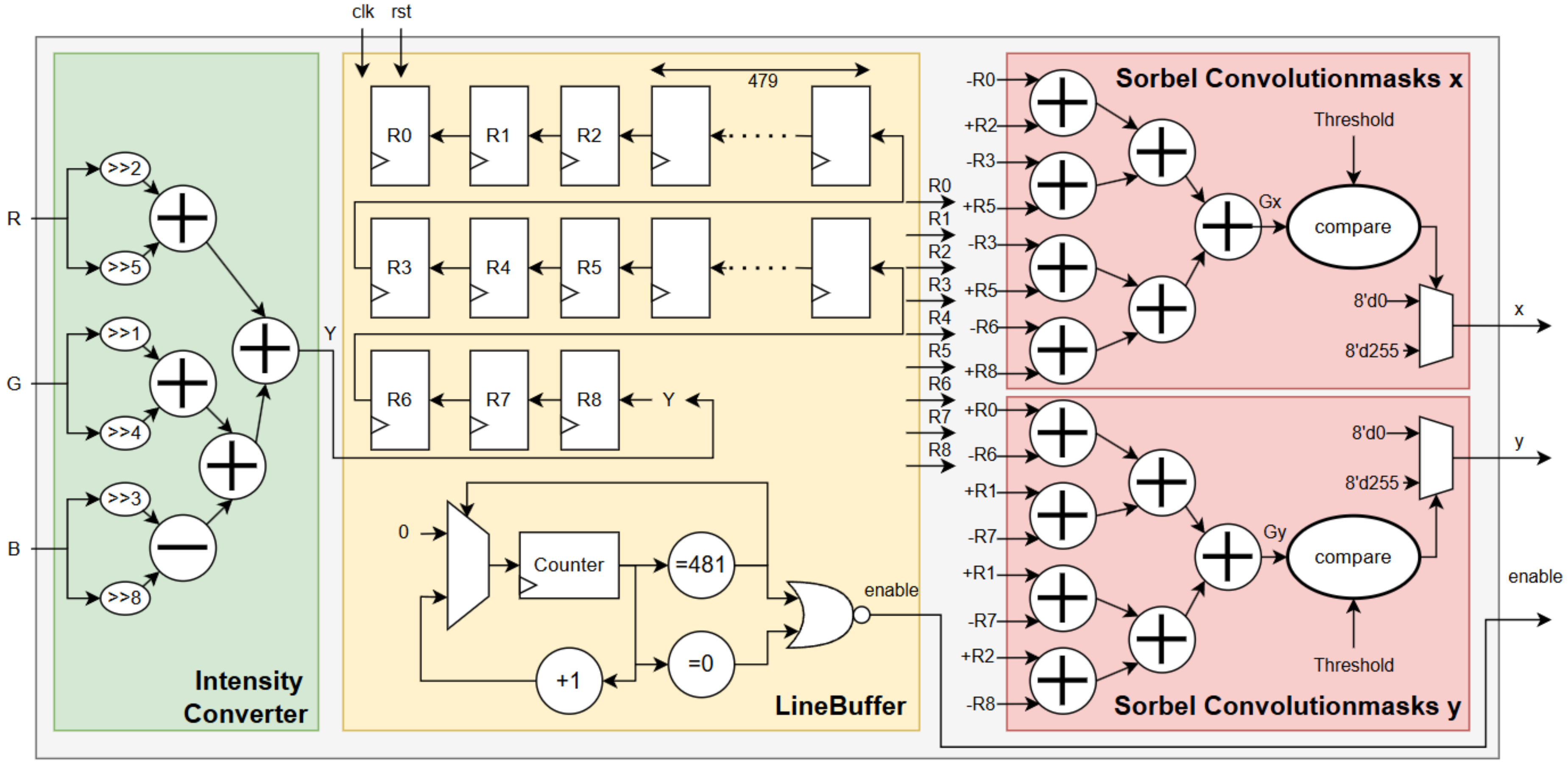
Testbench – Zero Padding



Bmp Address

Input Cycle
(with zero padding)

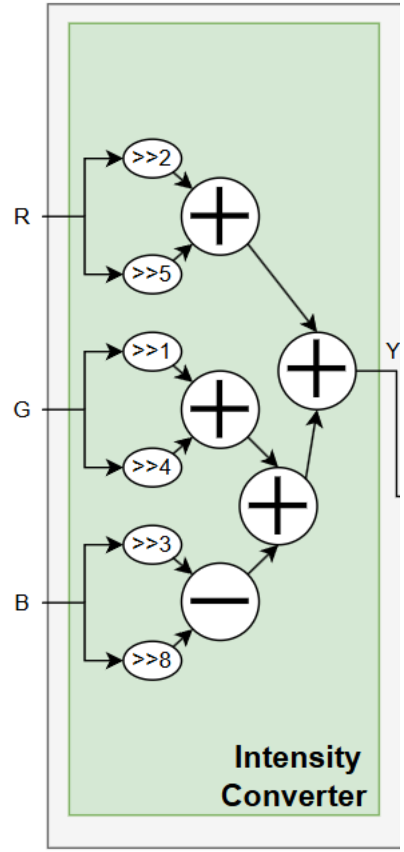
Hardware – Overall Architecture



Outline

- Hardware Implement
  - Read Image(.bmp) File - Testbench
  - Zero Padding - Testbench
  - RGB to YUV - Hardware
  - Line Buffer & Convolution - Hardware
  - Write Image(.bmp) File - Testbench

Hardware – RGB to Gray Level



$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

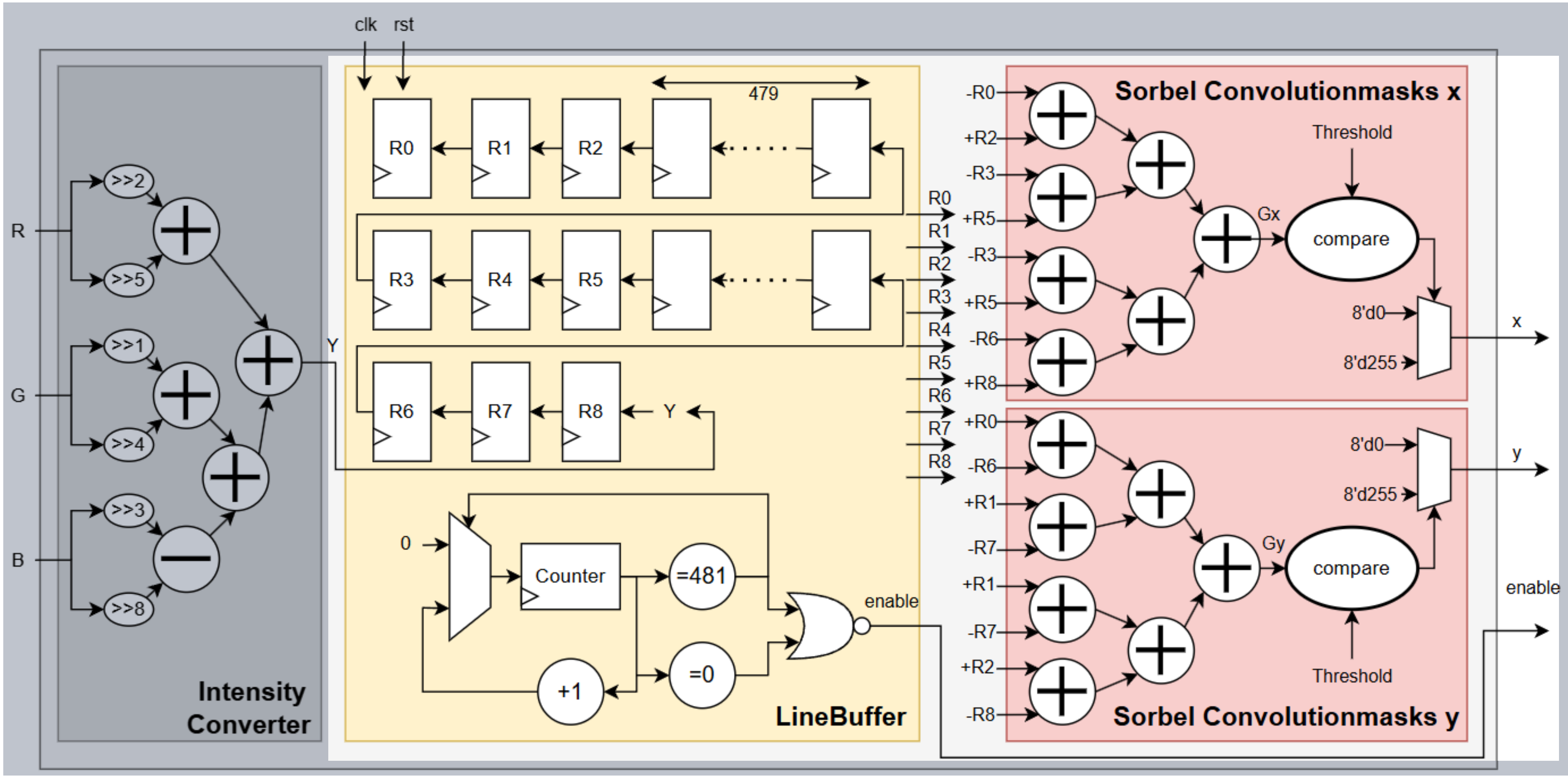
0.255 ~= $2^{-2} + 2^{-5}$; 0.587 ~= $2^{-1} + 2^{-4}$; 0.114 ~= $2^{-3} – 2^{-6}$

R * 0.255
= R * 2^{-2} + R * 2^{-5}
= R > > 2 + R > > 5

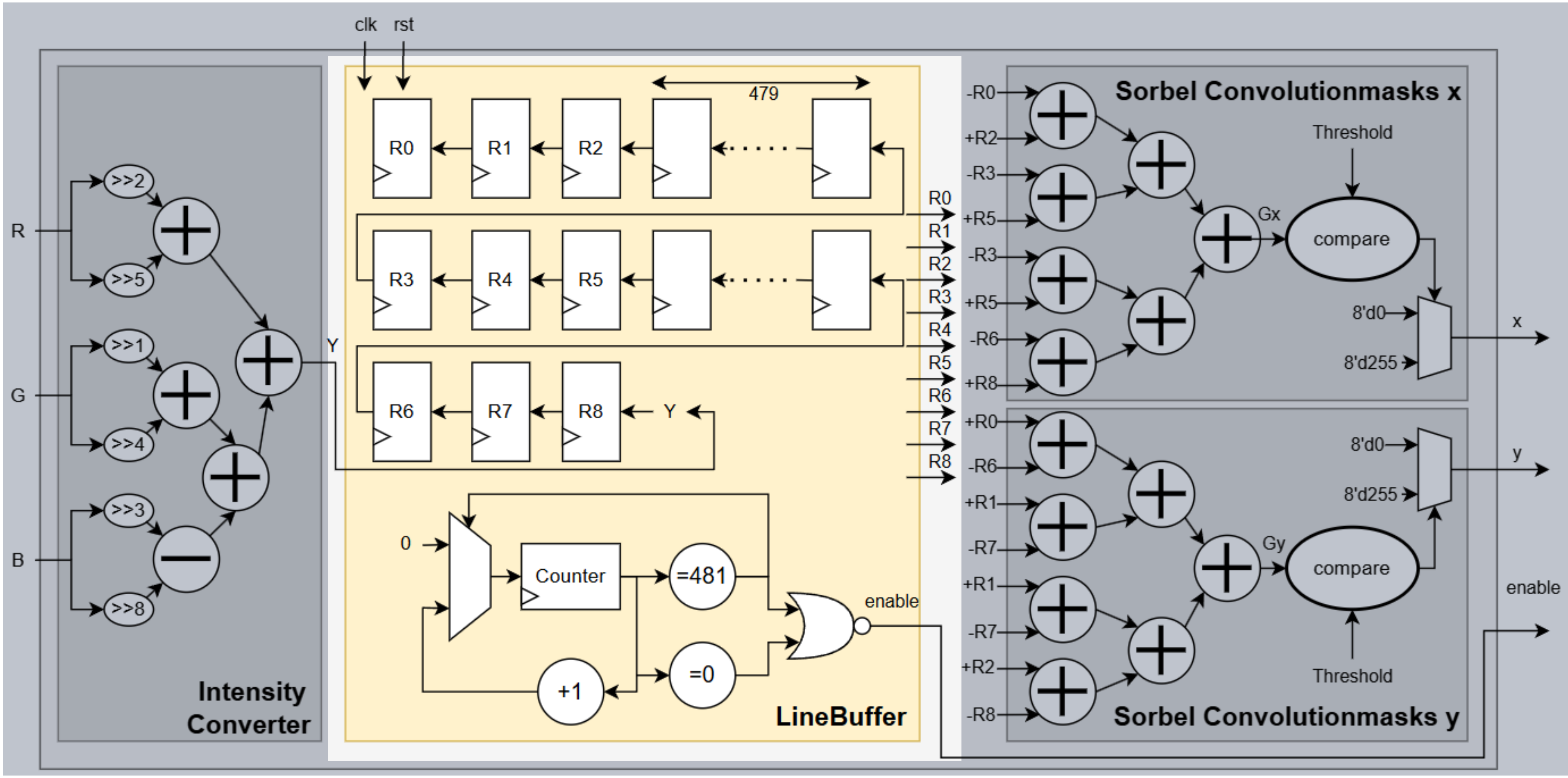
Outline

- Hardware Implement
  - Read Image(.bmp) File - Testbench
  - Zero Padding - Testbench
  - RGB to YUV - Hardware
  - Line Buffer & Convolution - Hardware
  - Write Image(.bmp) File - Testbench

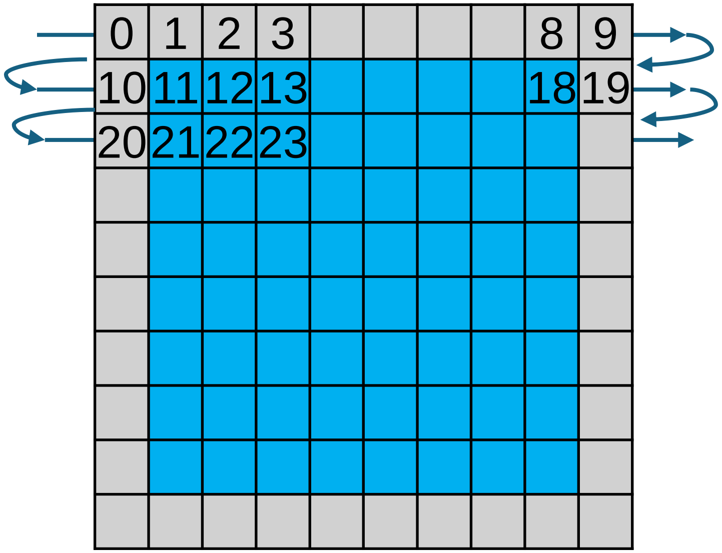
Hardware – Line Buffer & Convolution



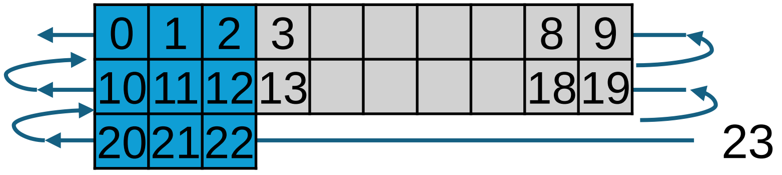
Hardware – Line Buffer



Hardware – Line Buffer

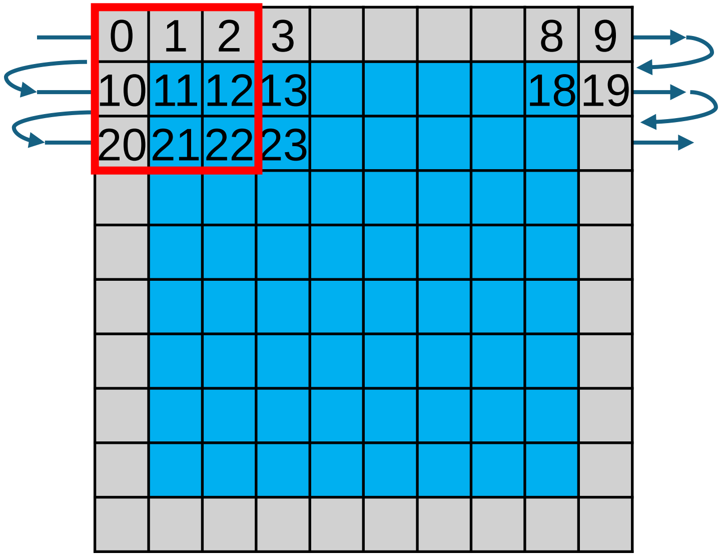


Input Cycle
(with zero padding)

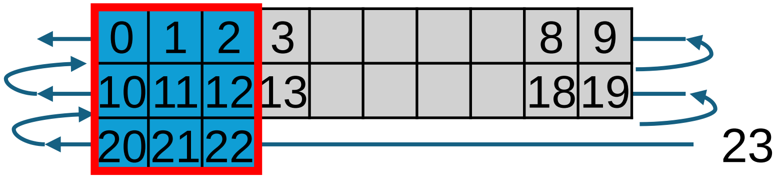


Line Buffer
(Shift Register)

Hardware – Line Buffer

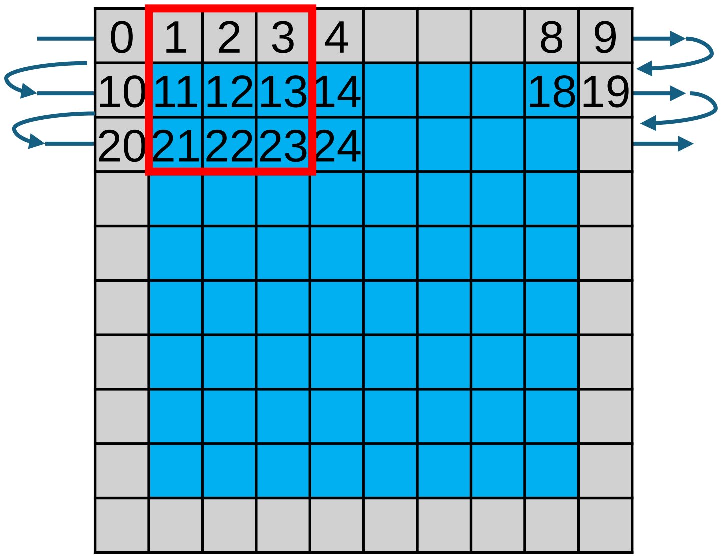


Input Cycle
(with zero padding)

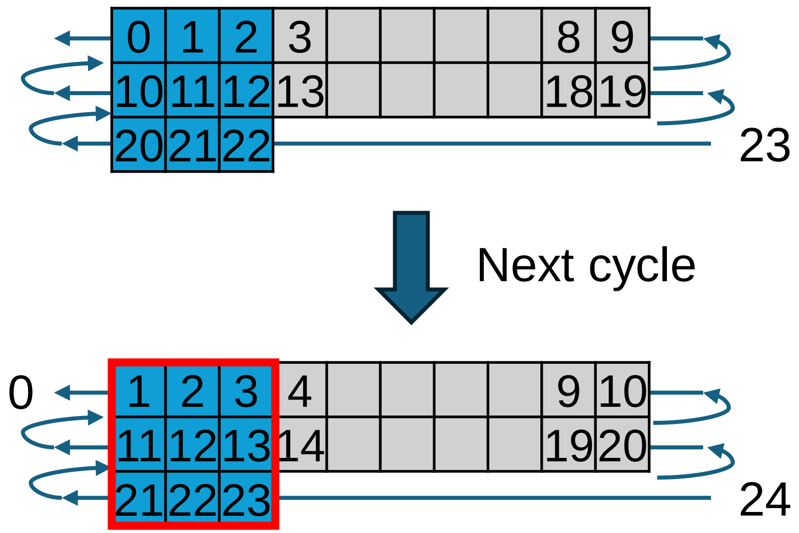


Line Buffer
(Shift Register)

Hardware – Line Buffer

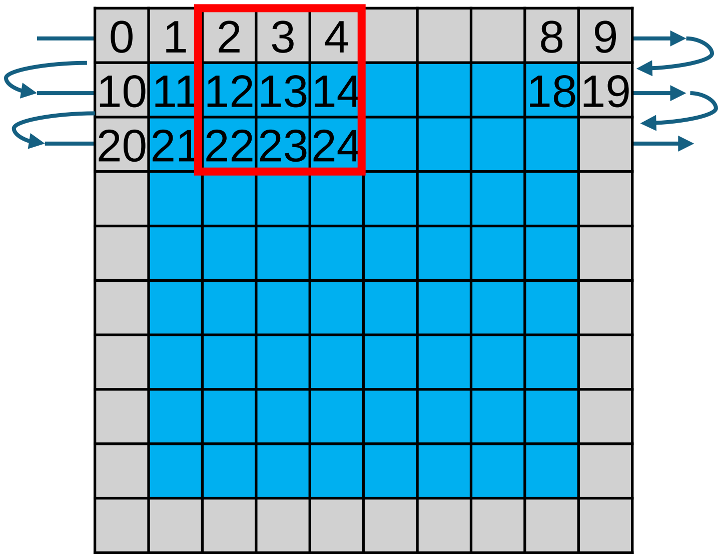


Input Cycle
(with zero padding)

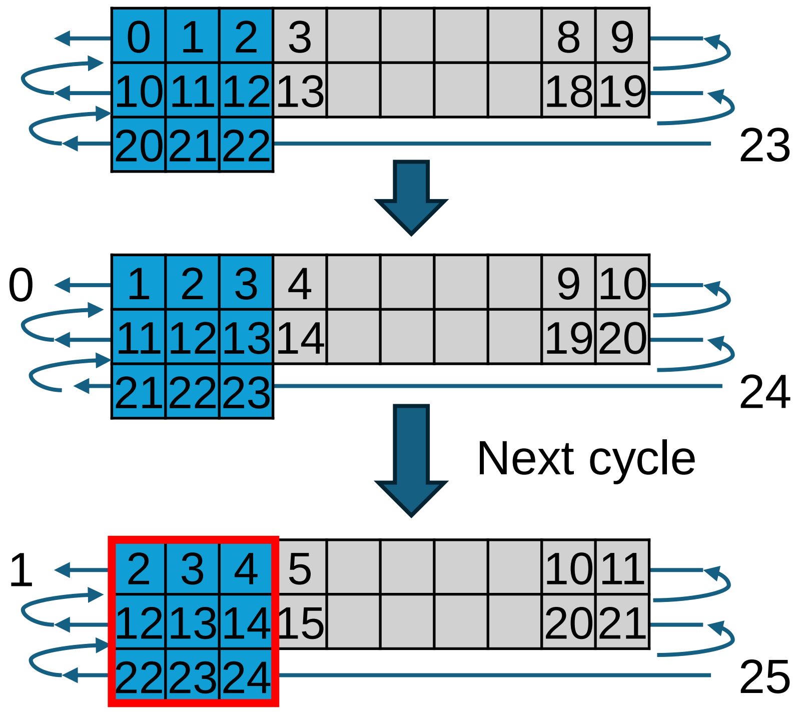


Line Buffer
(Shift Register)

Hardware – Line Buffer

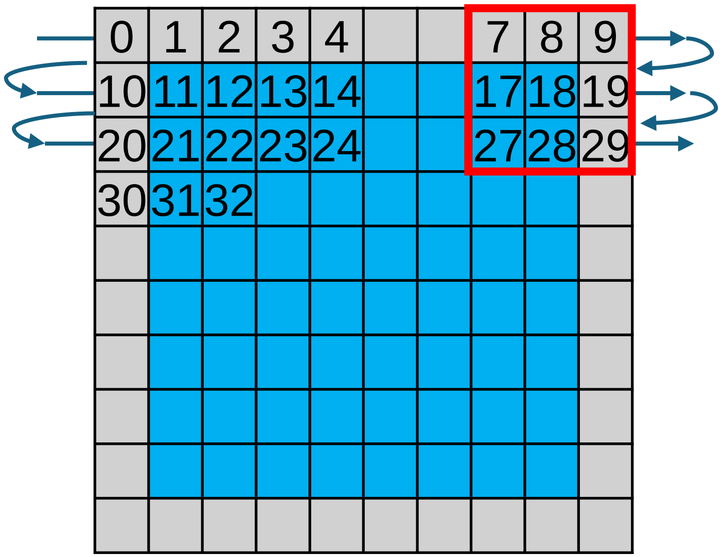


Input Cycle
(with zero padding)

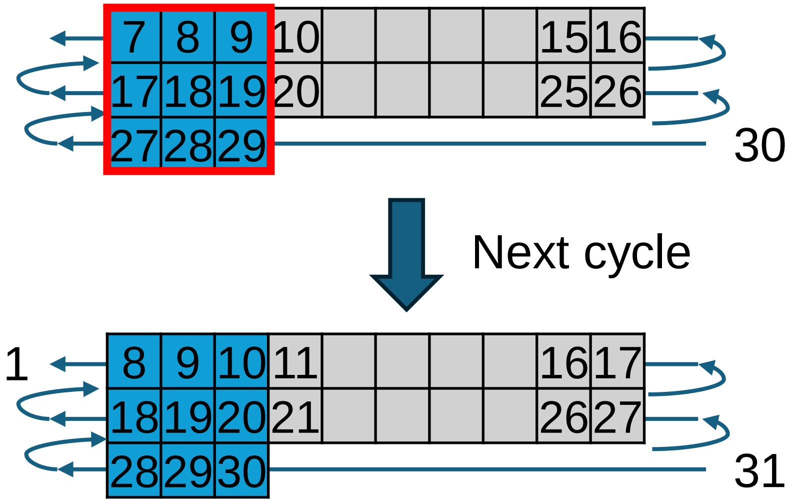


Line Buffer
(Shift Register)

Hardware – Line Buffer

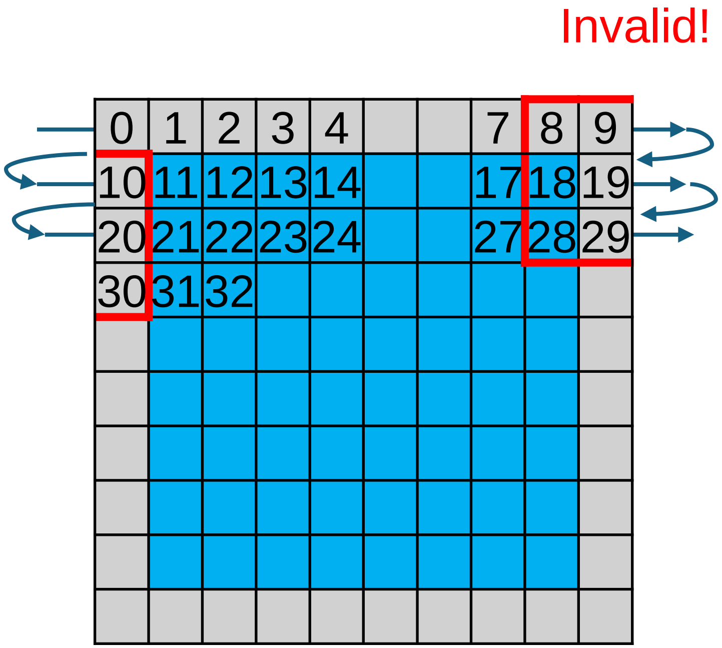


Input Cycle
(with zero padding)

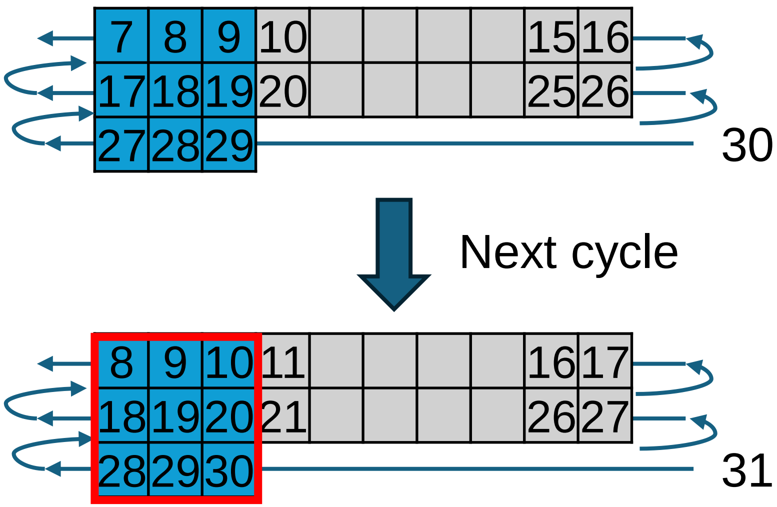


Line Buffer
(Shift Register)

Hardware – Line Buffer



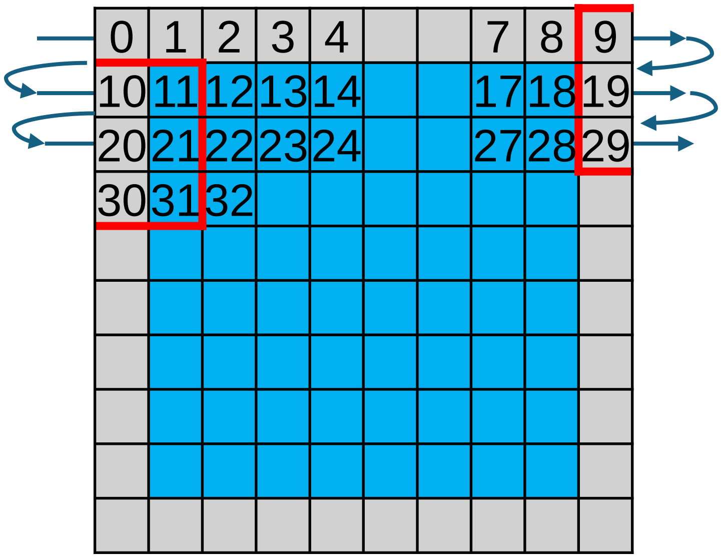
Input Cycle
(with zero padding)



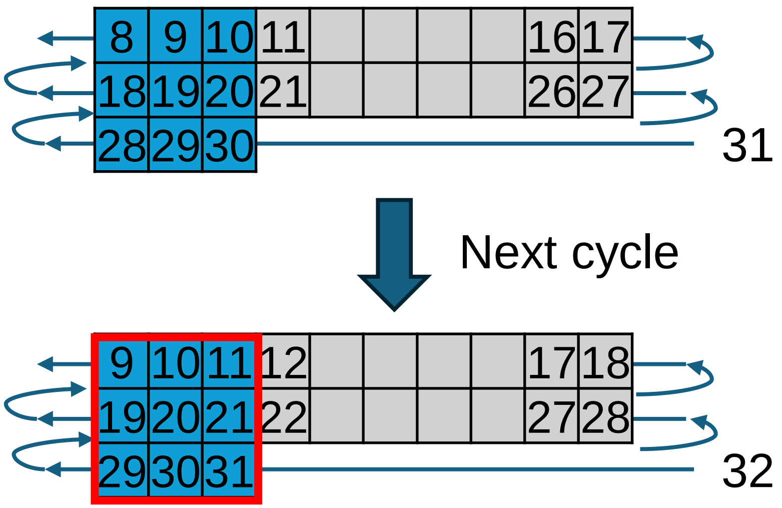
Line Buffer
(Shift Register)

Hardware – Line Buffer

Invalid!

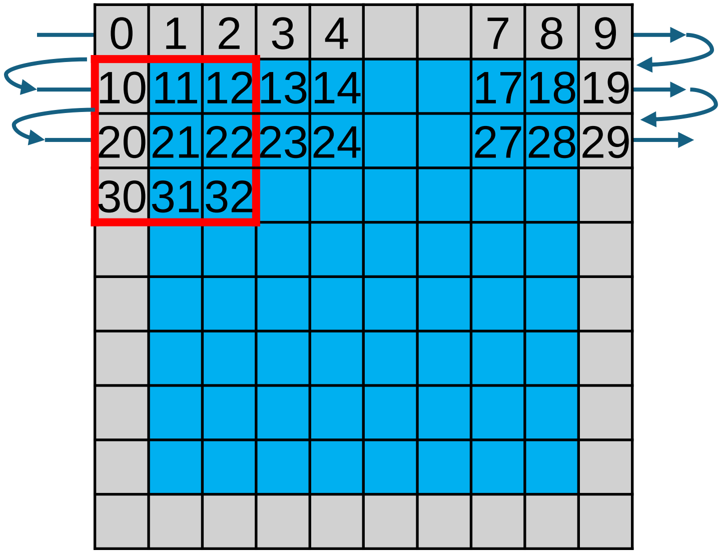


Input Cycle
(with zero padding)

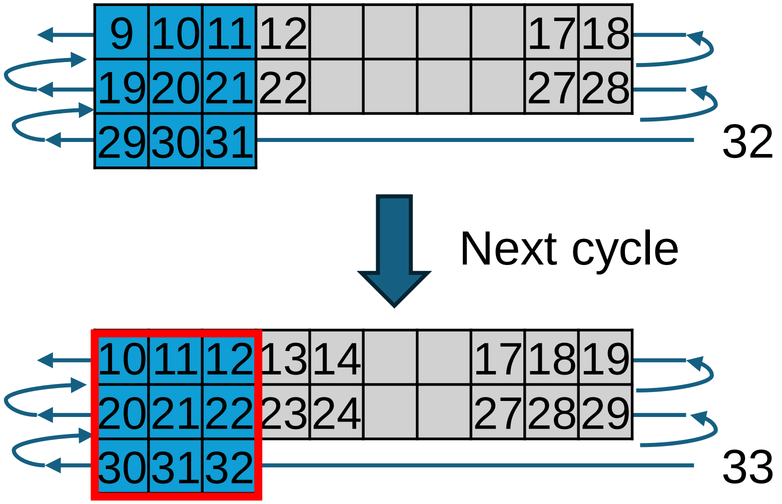


Line Buffer
(Shift Register)

Hardware – Line Buffer

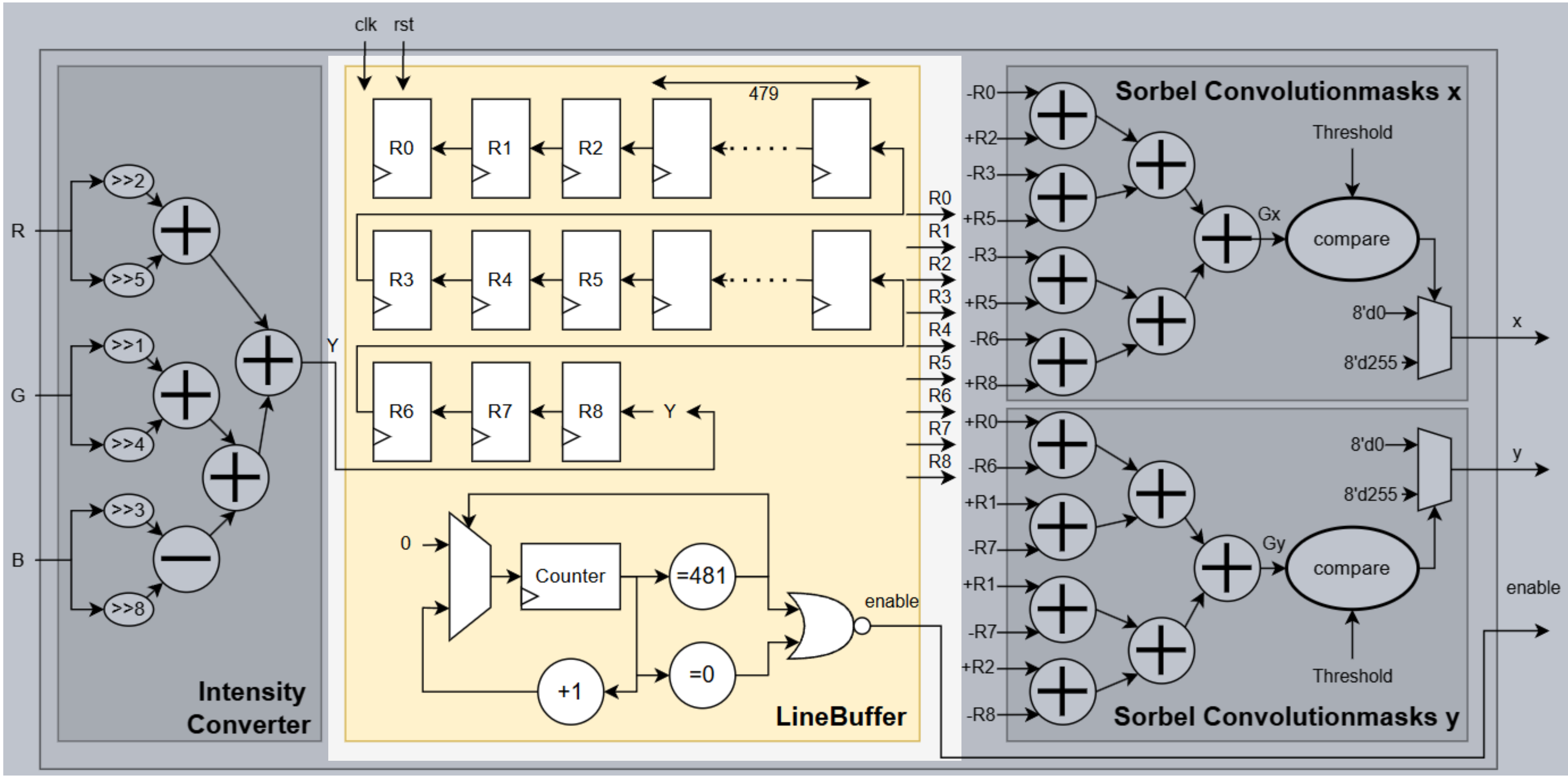


Input Cycle
(with zero padding)

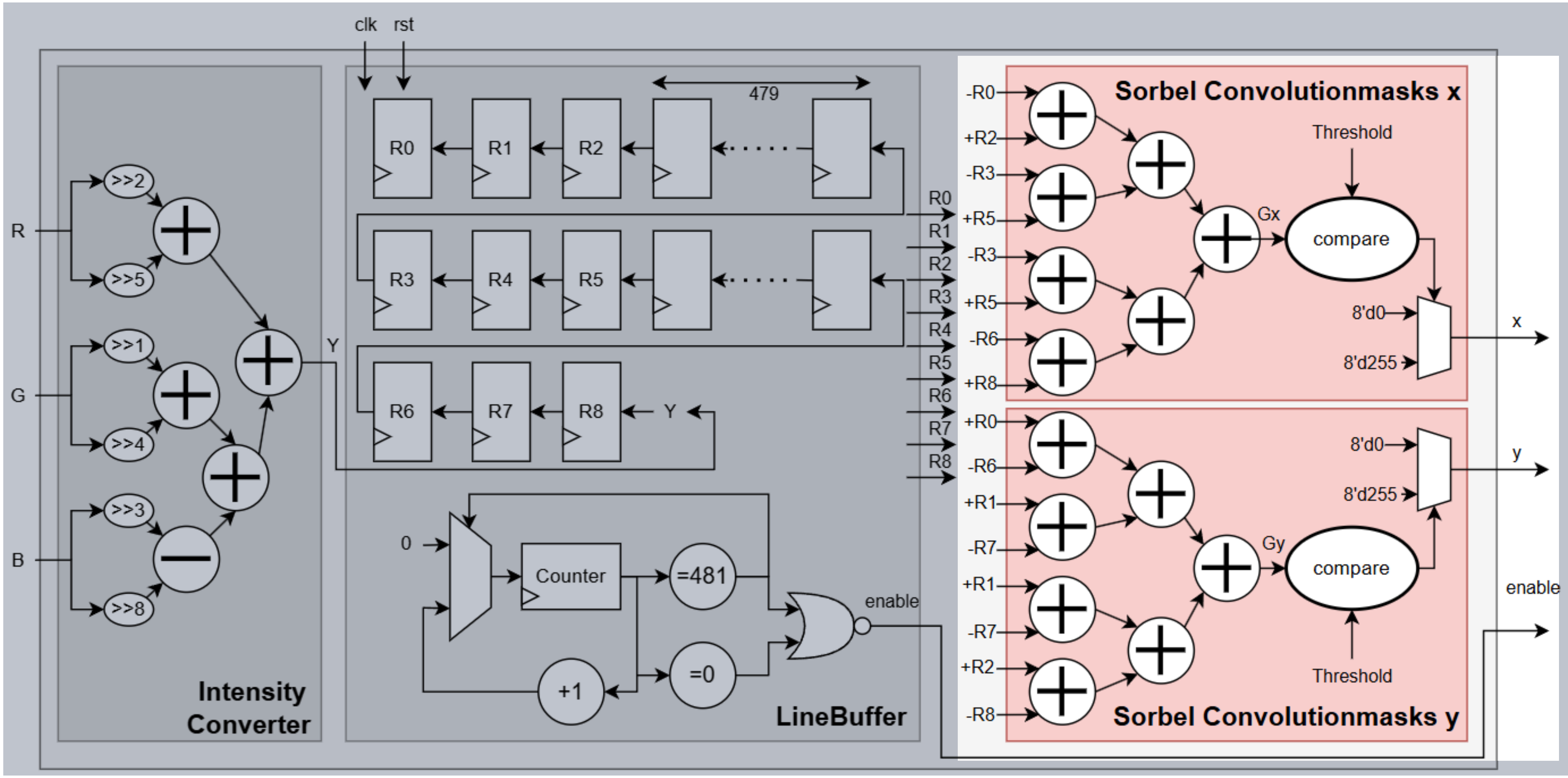


Line Buffer
(Shift Register)

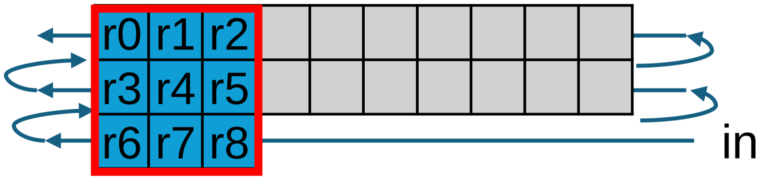
Hardware – Line Buffer



Hardware – Convolution



Hardware – Convolution



×

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix},$$

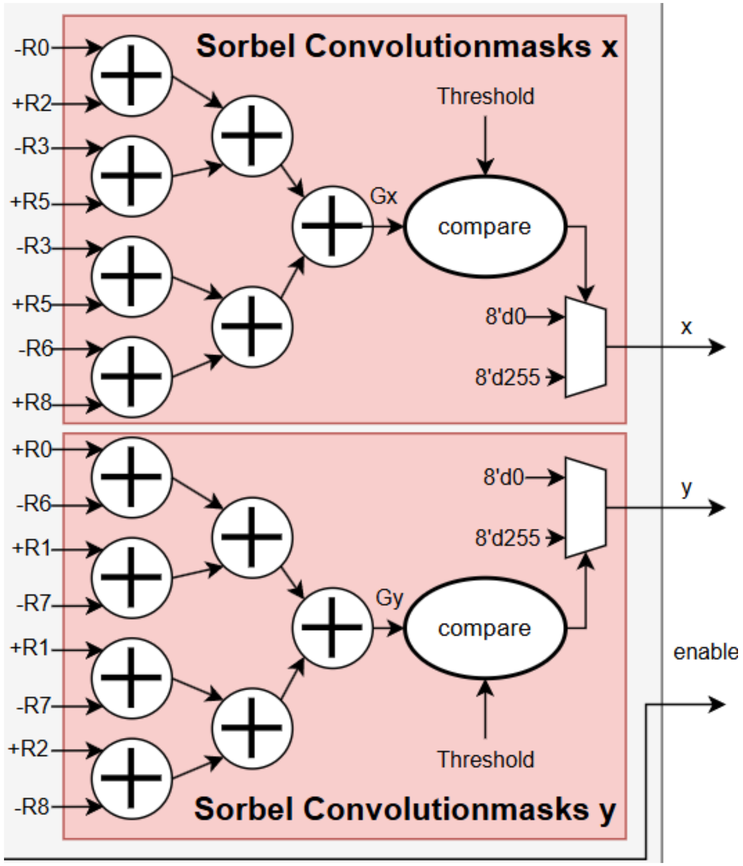
Line Buffer
(Shift Register)

Output
= -r0 + r2 – 2*r3 + 2*r5 – r6 + r8
= -r0 +r2 –r3 –r3 +r5 +r5 –r6 +r8

Hardware – Convolution

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix},$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$



Hardware – Convolution

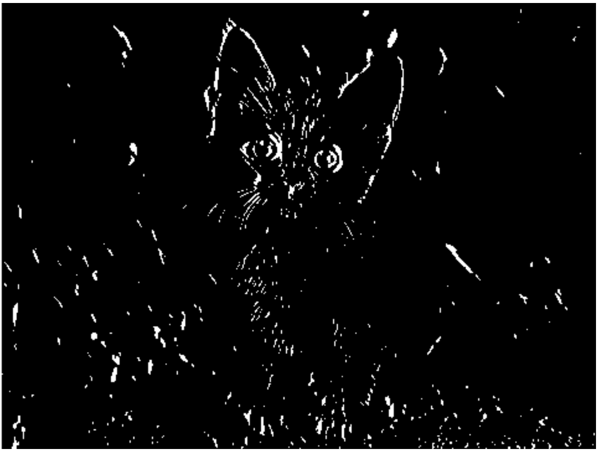
0 ~ 255

convolution

?

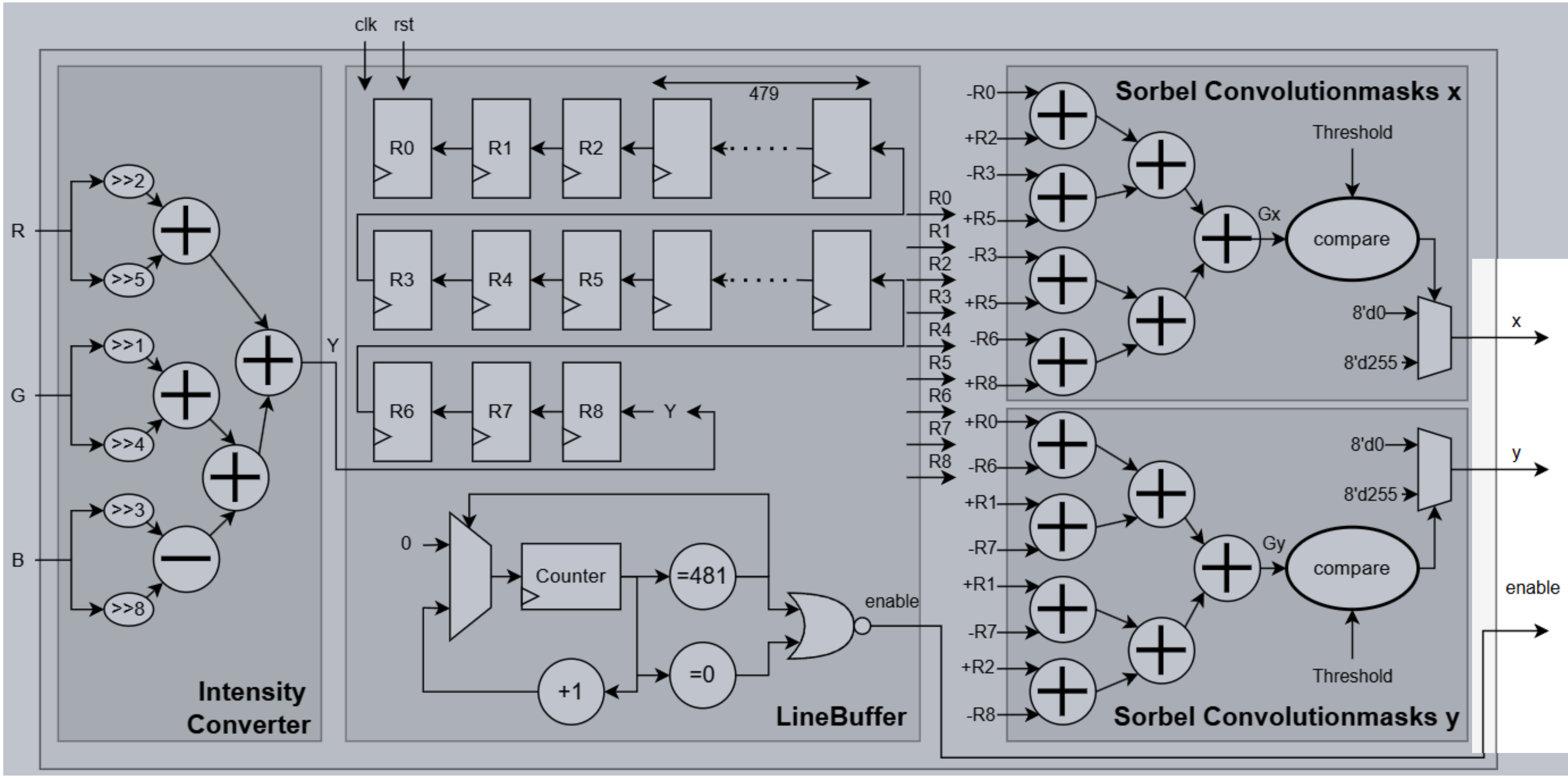
-1020
~
+1020

Compare with Threshold
Pixel Value > Threshold → 255
Pixel Value ≤ Threshold → 0



0 or 255
Only white or black

Testbench – Write Image File



繳交檔案

- Cell Base (Design Complier)
  - RTL code
  - Pre/post-syn simulation Testbench
  - Gate level netlist (including sdf file) -> Area optimize
- FPGA(Xilinx Vivado)
  - .xpr.zip
  - xdc, wcfg
- PDF Report

PDF Report (Cell Base)

- Figure of overall architecture (架構圖)
- Both RTL and gate-level simulation waveforms, including explanations
  (RTL波形 & gate-level波形並解釋)
- area information and critical path delay
  (Area資訊和critical path資訊)
- original input image (of cat),
  image of horizontal edges
  and vertical edges (原貓咪圖、水平邊緣圖片、垂直邊緣圖片)

PDF Report (FPGA)

- Simulation waveforms of both behavior level and post-implementation, including explanations
(Behavior波形 & post-implement波形並解釋)
- Snapshots of project summary-overview
(Project Summary-Overview截圖)
- Comments (心得)