HW4 作業說明

王敬翔 hssiang@g-amil.nsysu.edu.tw

2025 HDL

Outline

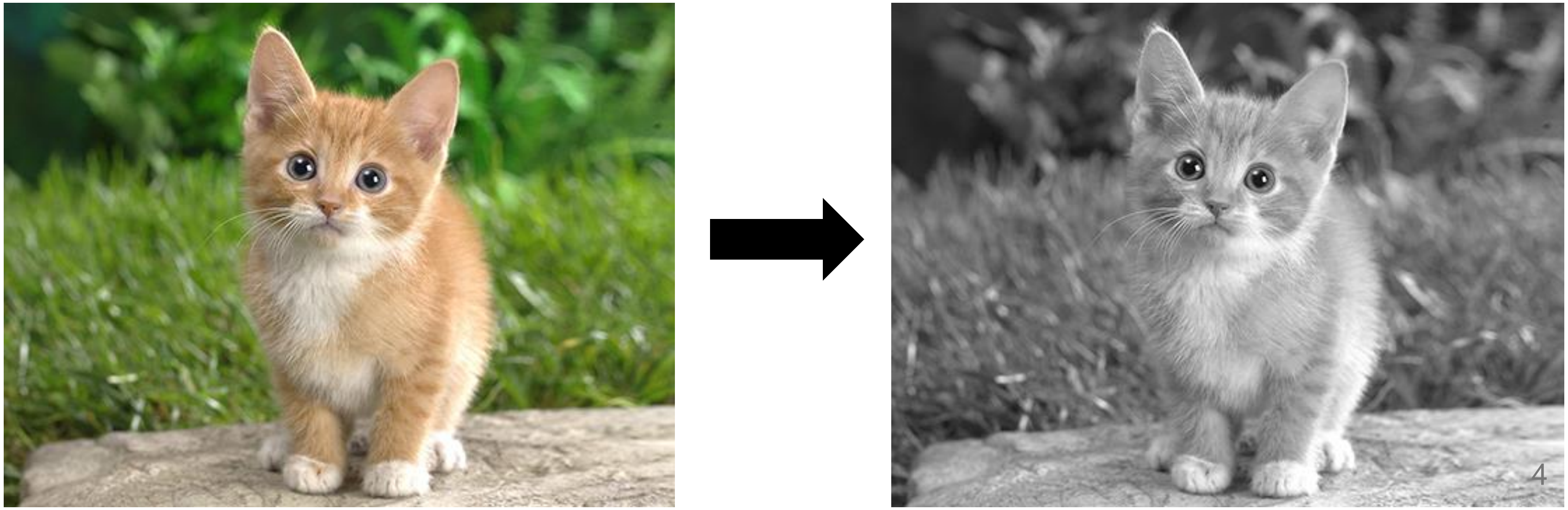
- Sobel Edge Operation
  - Color Space Transformation
  - Convolution with Sobel Filters
- Hardware Implement
  - Read Image(.bmp) File - Testbench
  - Zero Padding - Testbench
  - RGB to YUV - Hardware
  - Line Buffer & Convolution - Hardware
  - Write Image(.bmp) File - Testbench

Sobel Edge Operation

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

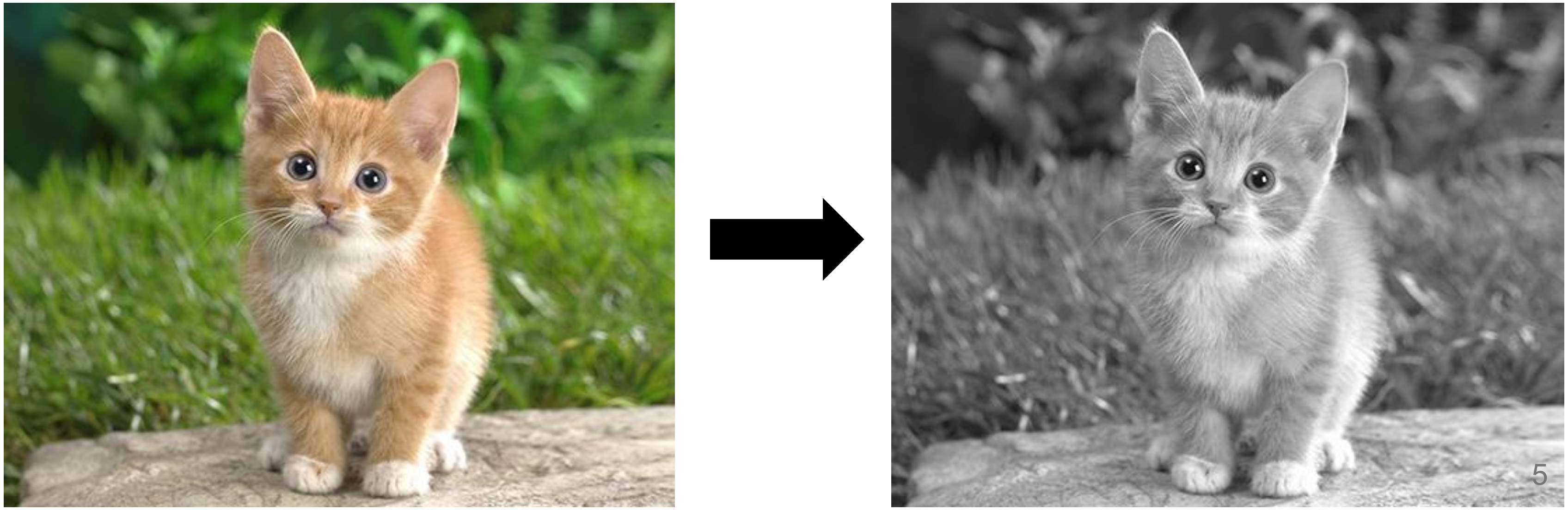
Color Space Transformation

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



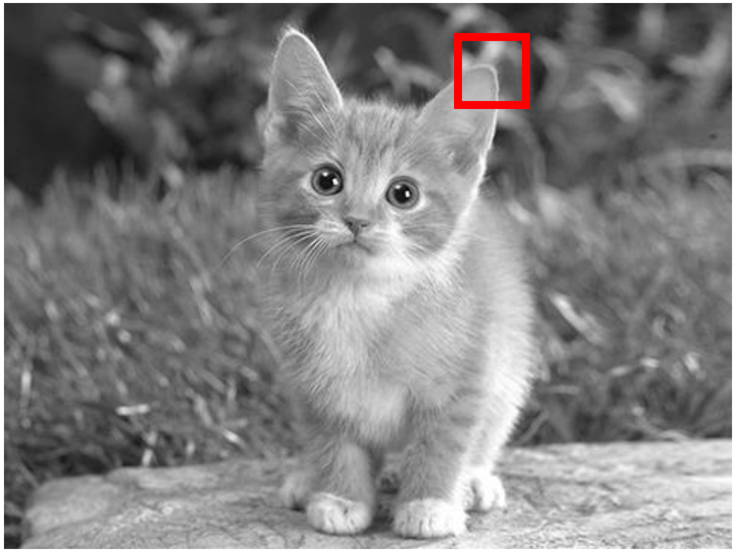
Color Space Transformation

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



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

Stride = 1

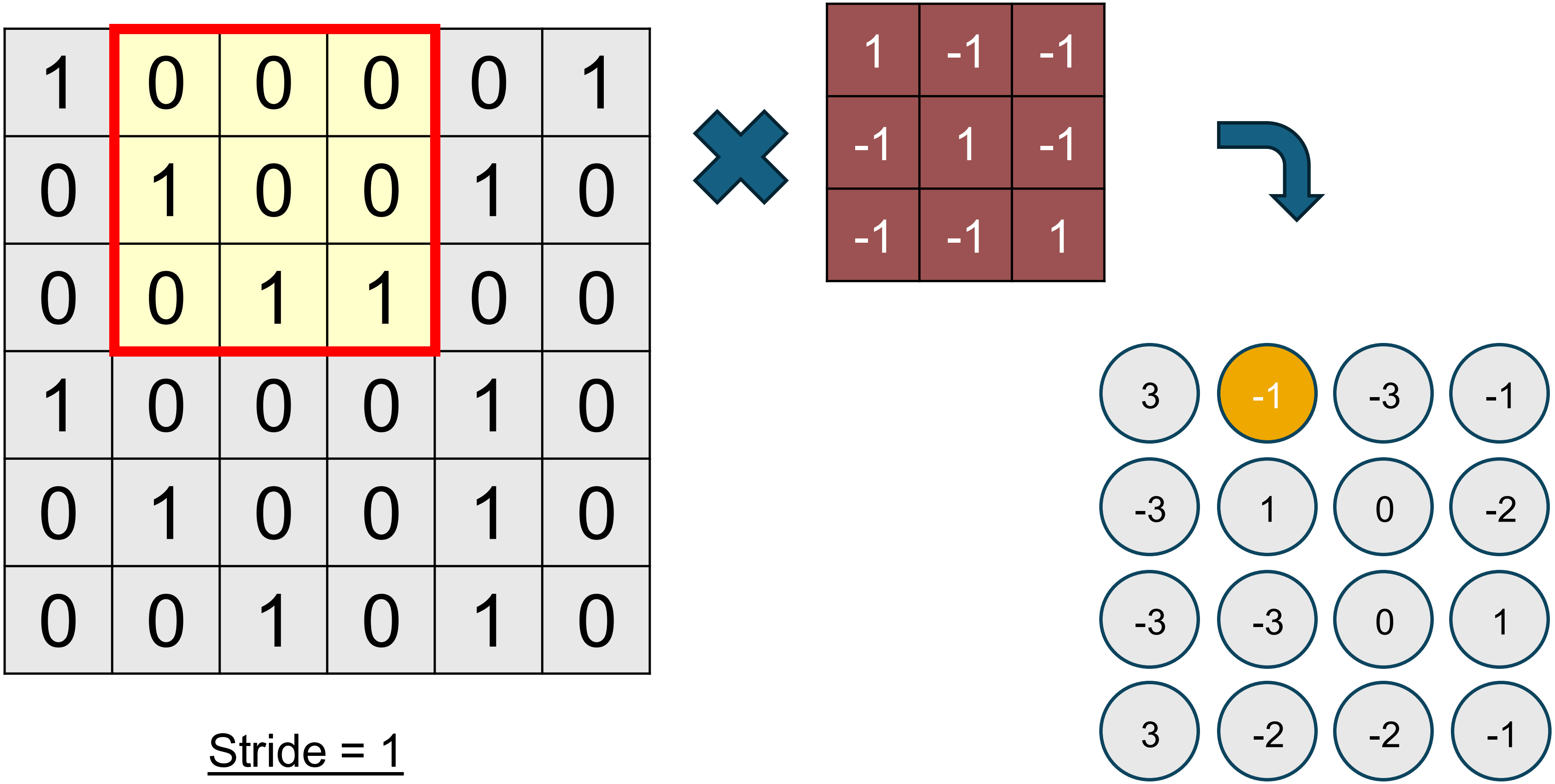
Filter(3 * 3)

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)

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

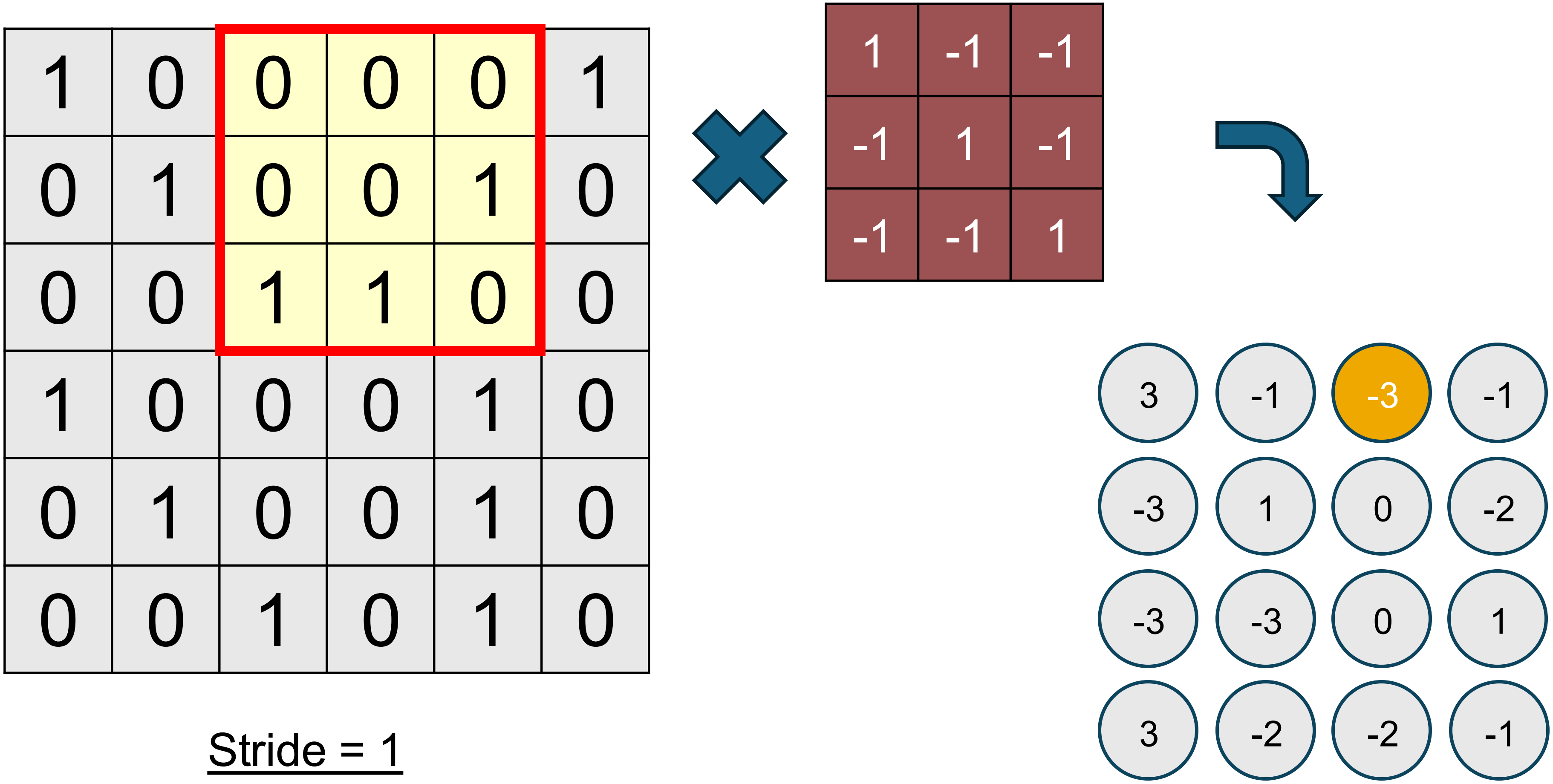
Filter(3 * 3)

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)

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)

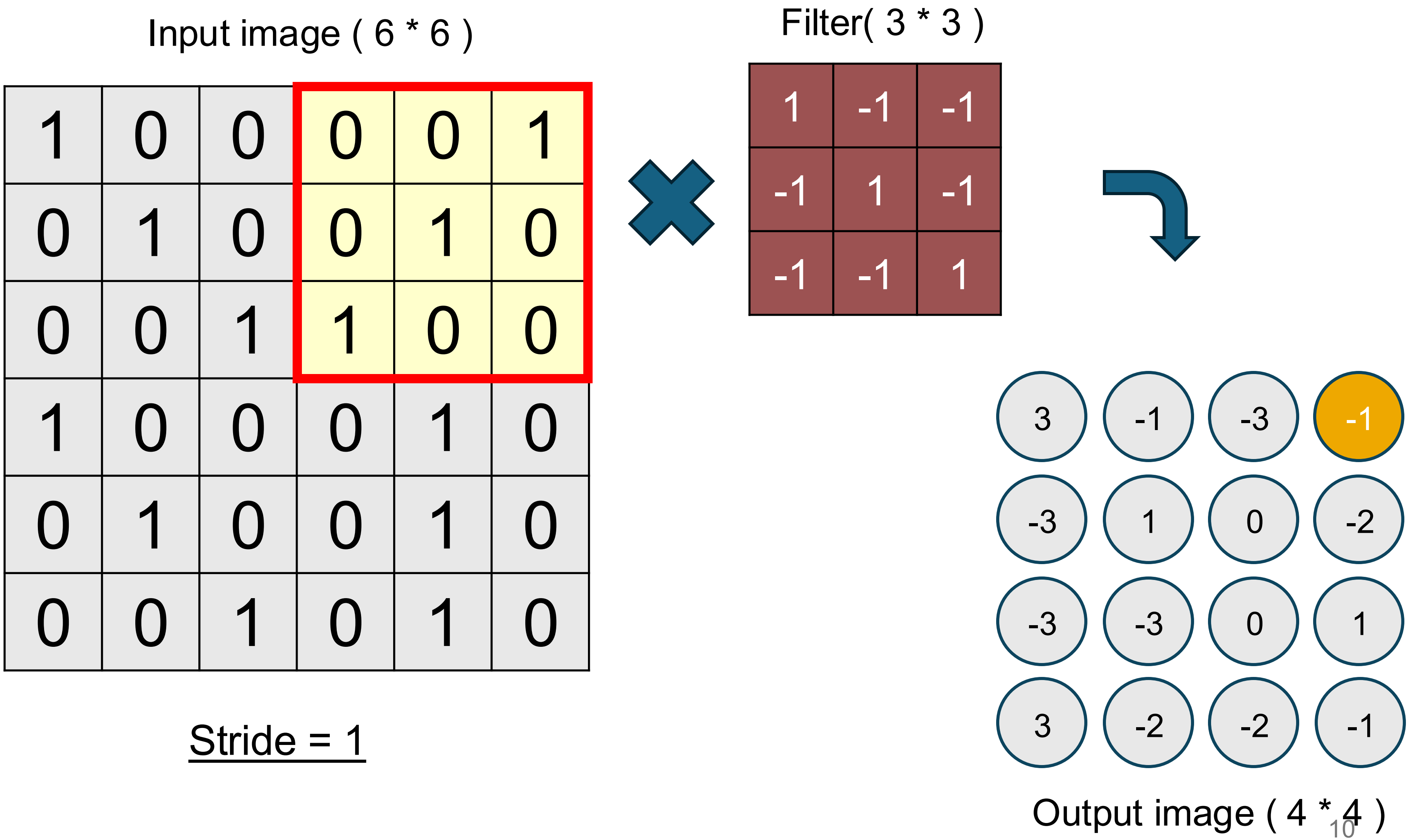
1	-1	-1
-1	1	-1
-1	-1	1

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)

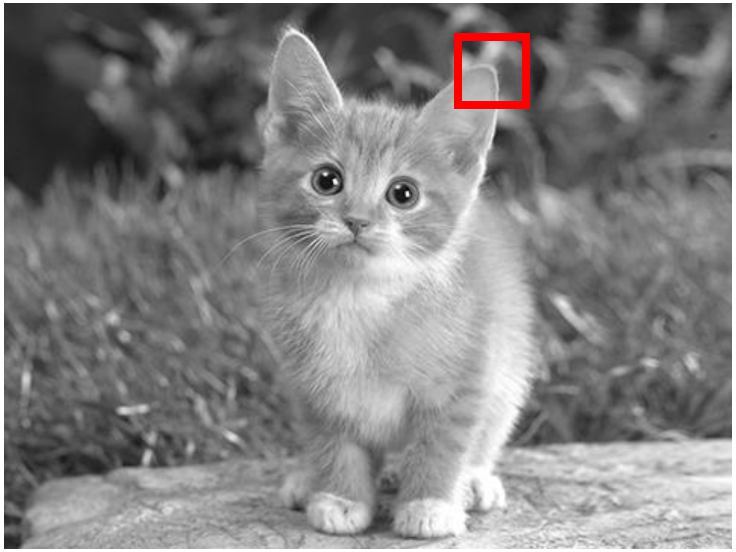
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

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

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

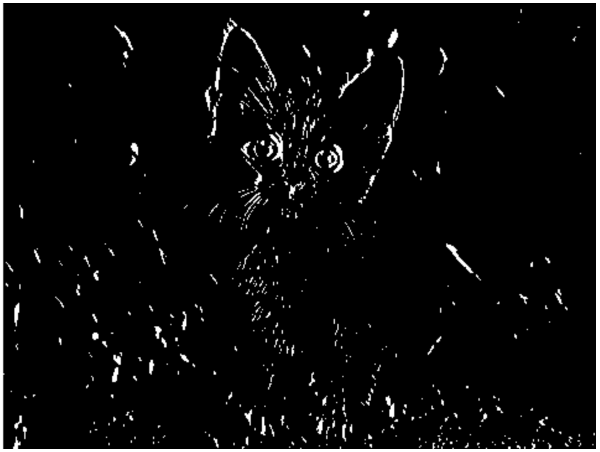
0 ~ 255

convolution

?

-1020
~
+1020

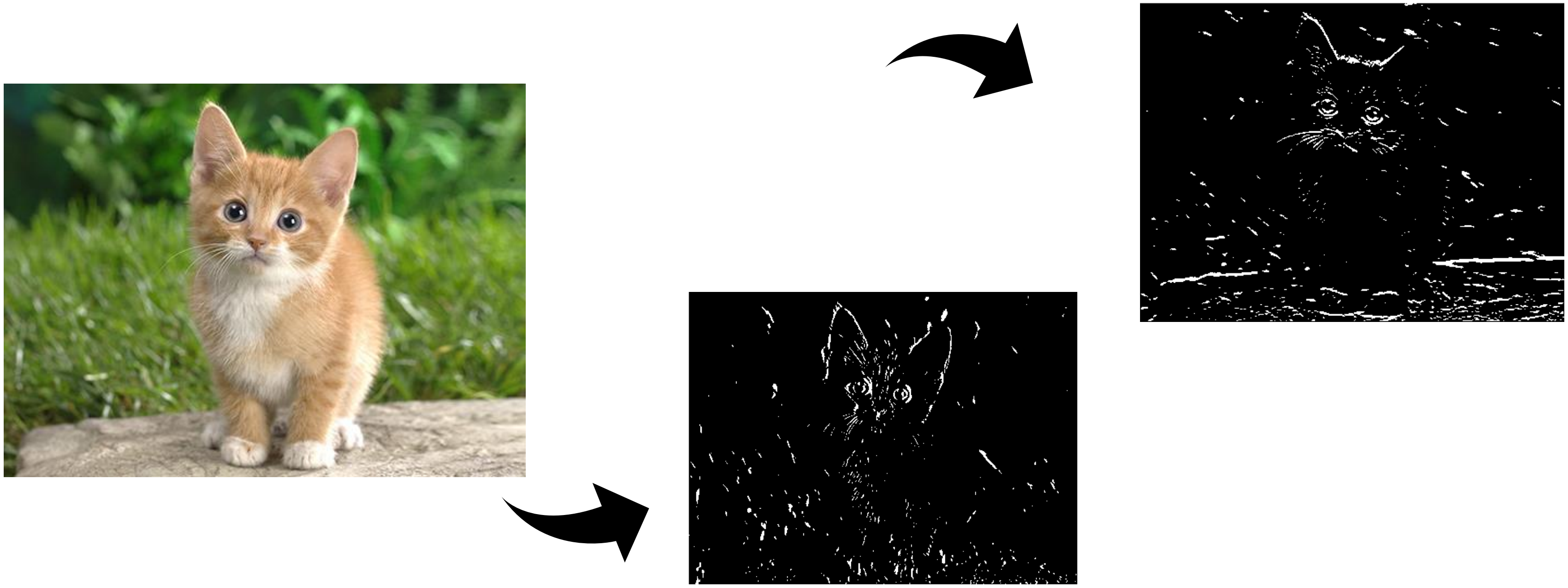
Compare with Threshold
Pixel Value > Threshold → 255 **(White)**
Pixel Value ≤ Threshold → 0 **(Black)**
***** **請自己調整 Threshold**



0 or 255
Only white or black

Sobel Edge Operation

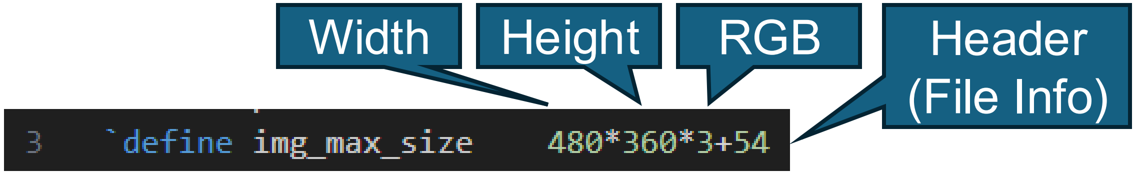
- Color Space Transformation
- Convolution with Sobel Filters
- Pixel to Binary Transformation by Threshold



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



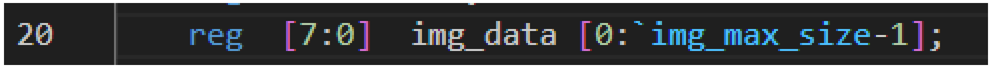
Width

Height

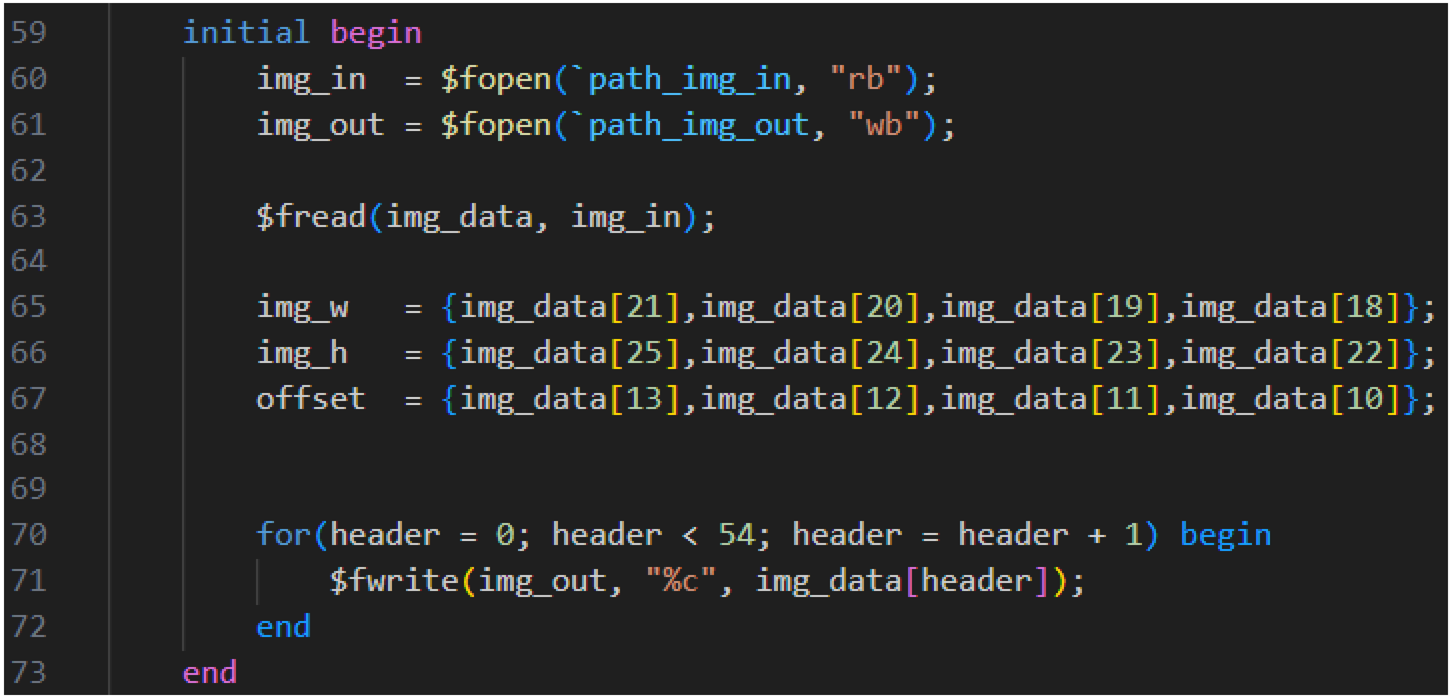
RGB

Header (File Info)

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

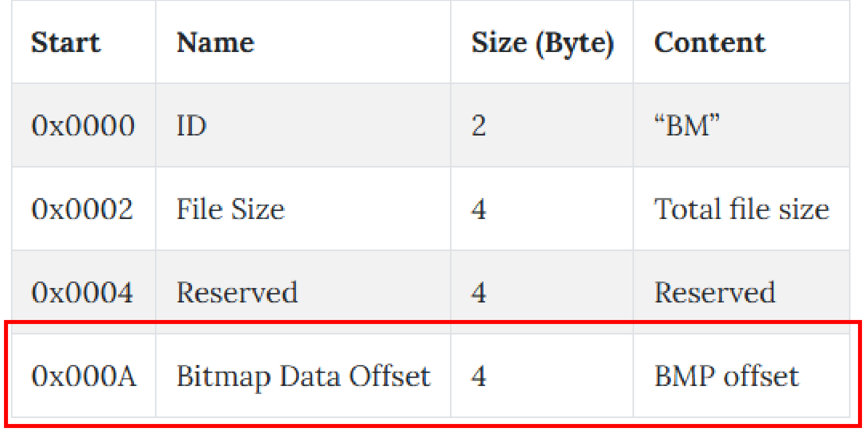


```
    reg  [7:0]  img_data [0:`img_max_size-1];
```

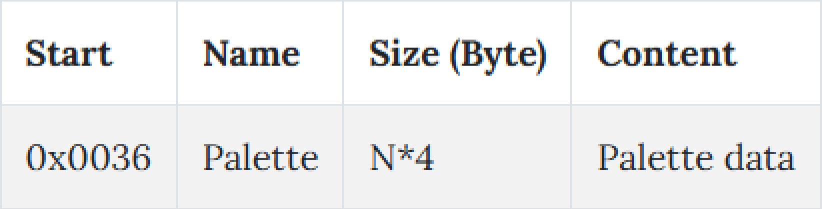


```
    initial begin
        img_in  = $fopen(`path_img_in, "rb");
        img_out = $fopen(`path_img_out, "wb");

        $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
```

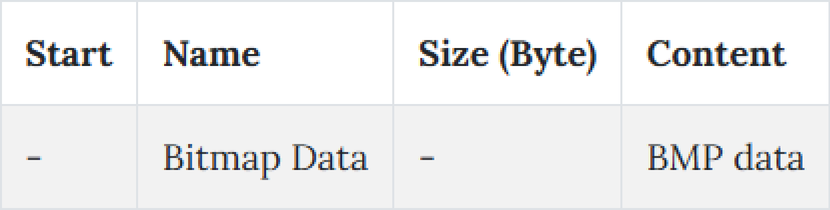
BMP File Format



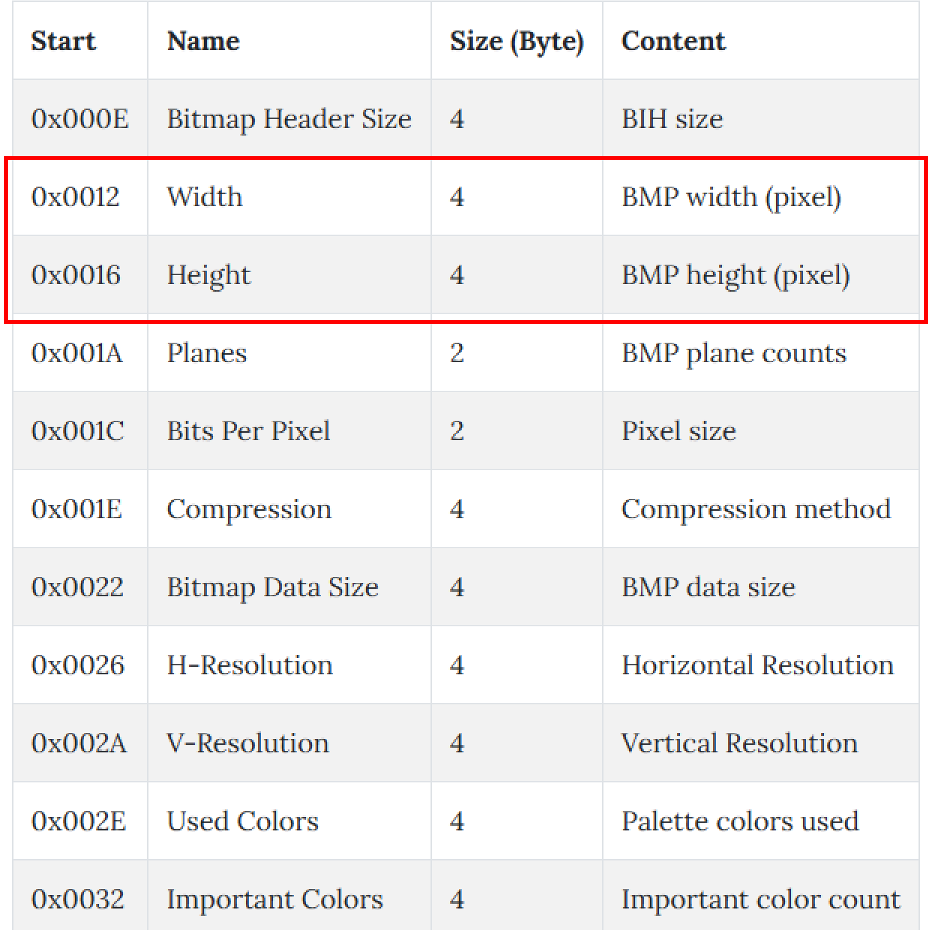
Start	Name	Size (Byte)	Content
0x0000	ID	2	“BM”
0x0002	File Size	4	Total file size
0x0004	Reserved	4	Reserved
0x000A	Bitmap Data Offset	4	BMP offset



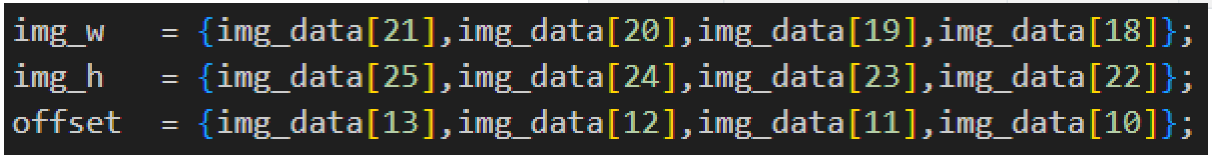
Start	Name	Size (Byte)	Content
0x0036	Palette	N*4	Palette data



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

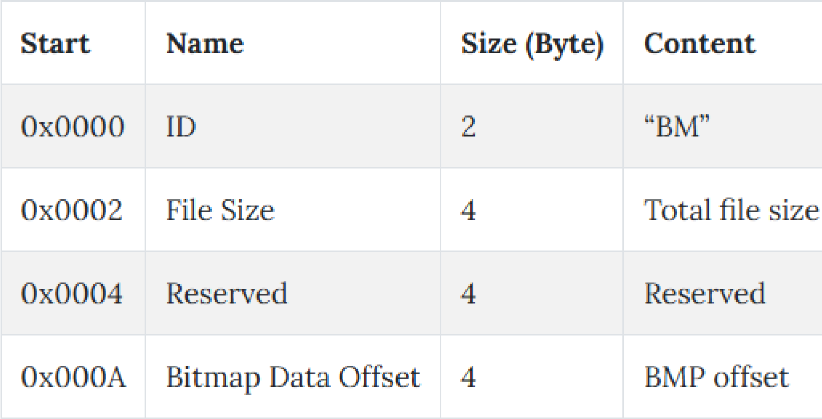


Start	Name	Size (Byte)	Content
0x000E	Bitmap Header Size	4	BIH size
0x0012	Width	4	BMP width (pixel)
0x0016	Height	4	BMP height (pixel)
0x001A	Planes	2	BMP plane counts
0x001C	Bits Per Pixel	2	Pixel size
0x001E	Compression	4	Compression method
0x0022	Bitmap Data Size	4	BMP data size
0x0026	H-Resolution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

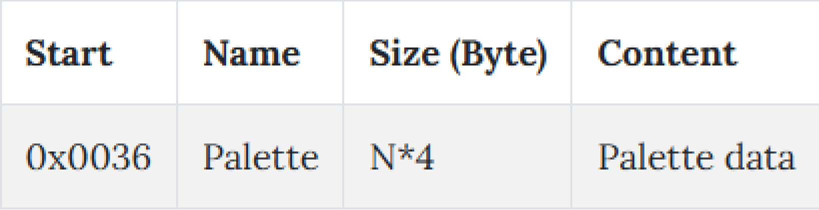


```
img_w    = {img_data[21],img_data[20],img_data[19],img_data[18]};
img_h    = {img_data[25],img_data[24],img_data[23],img_data[22]};
offset   = {img_data[13],img_data[12],img_data[11],img_data[10]};
```

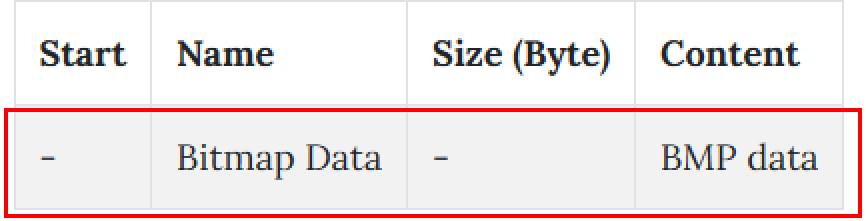
BMP File Format



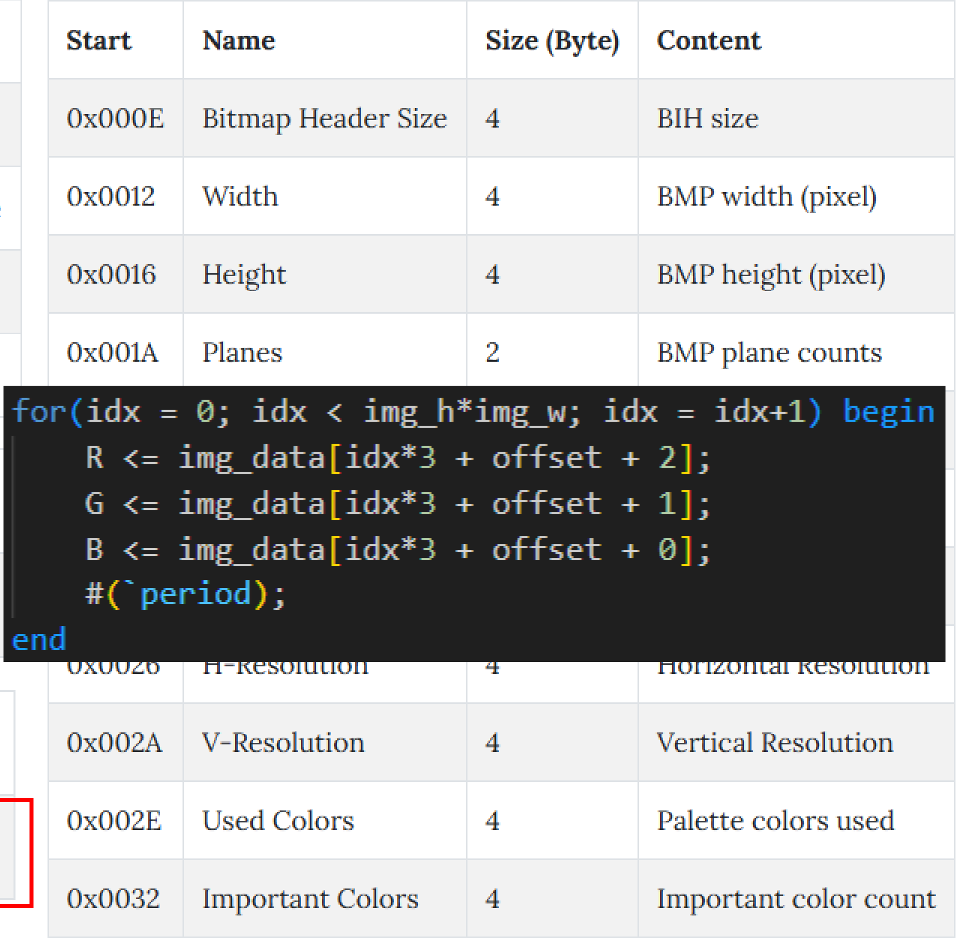
Start	Name	Size (Byte)	Content
0x0000	ID	2	"BM"
0x0002	File Size	4	Total file size
0x0004	Reserved	4	Reserved
0x000A	Bitmap Data Offset	4	BMP offset



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



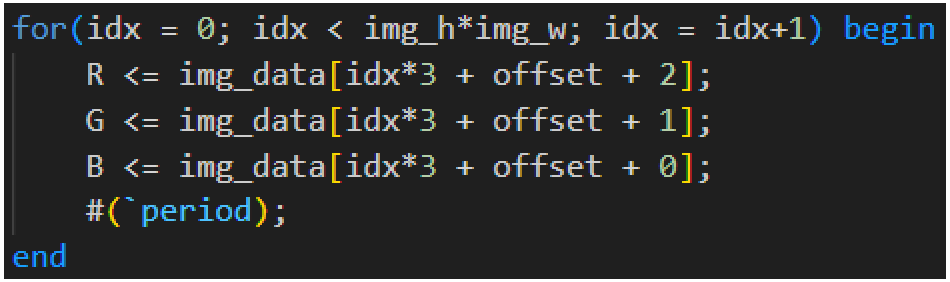
Start	Name	Size (Byte)	Content
-	Bitmap Data	-	BMP data



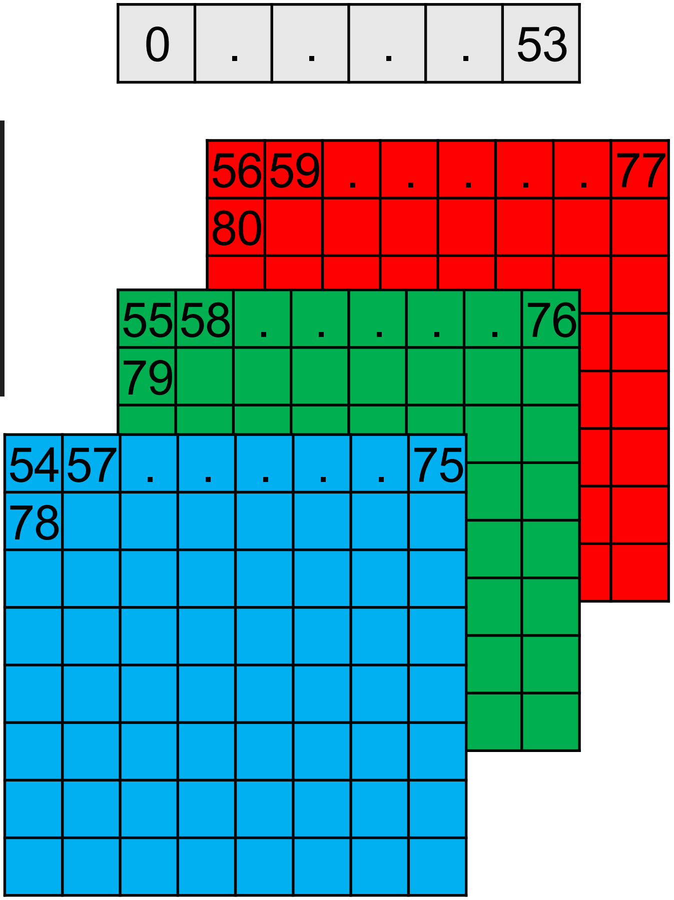
Start	Name	Size (Byte)	Content
0x000E	Bitmap Header Size	4	BIH size
0x0012	Width	4	BMP width (pixel)
0x0016	Height	4	BMP height (pixel)
0x001A	Planes	2	BMP plane counts
0x0026	H-Resolution	4	Horizontal Resolution
0x002A	V-Resolution	4	Vertical Resolution
0x002E	Used Colors	4	Palette colors used
0x0032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



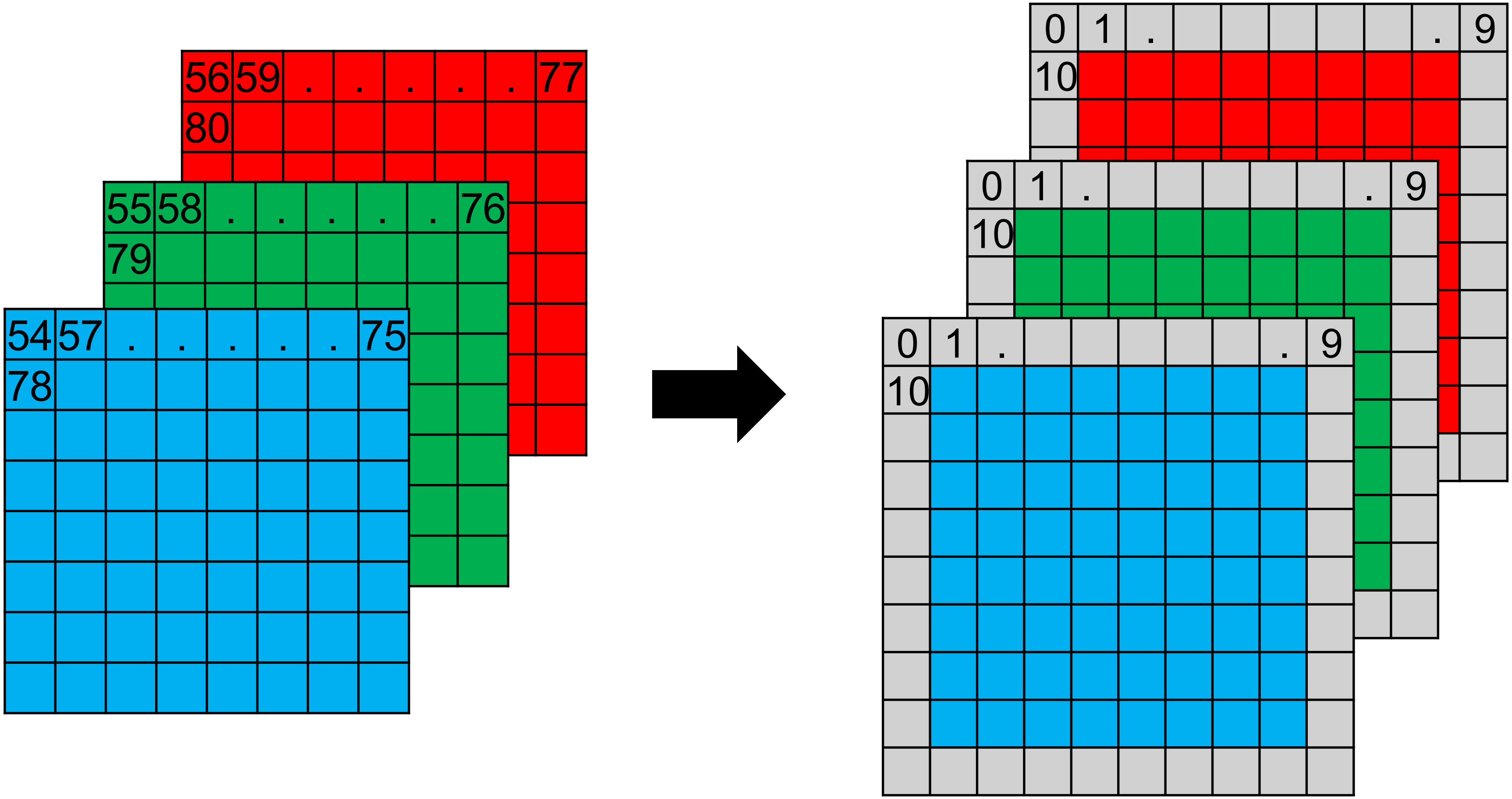
```
for(idx = 0; idx < img_h*img_w; idx = idx+1) begin
    R <= img_data[idx*3 + offset + 2];
    G <= img_data[idx*3 + offset + 1];
    B <= img_data[idx*3 + offset + 0];
    #(`period);
end
```



Outline

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

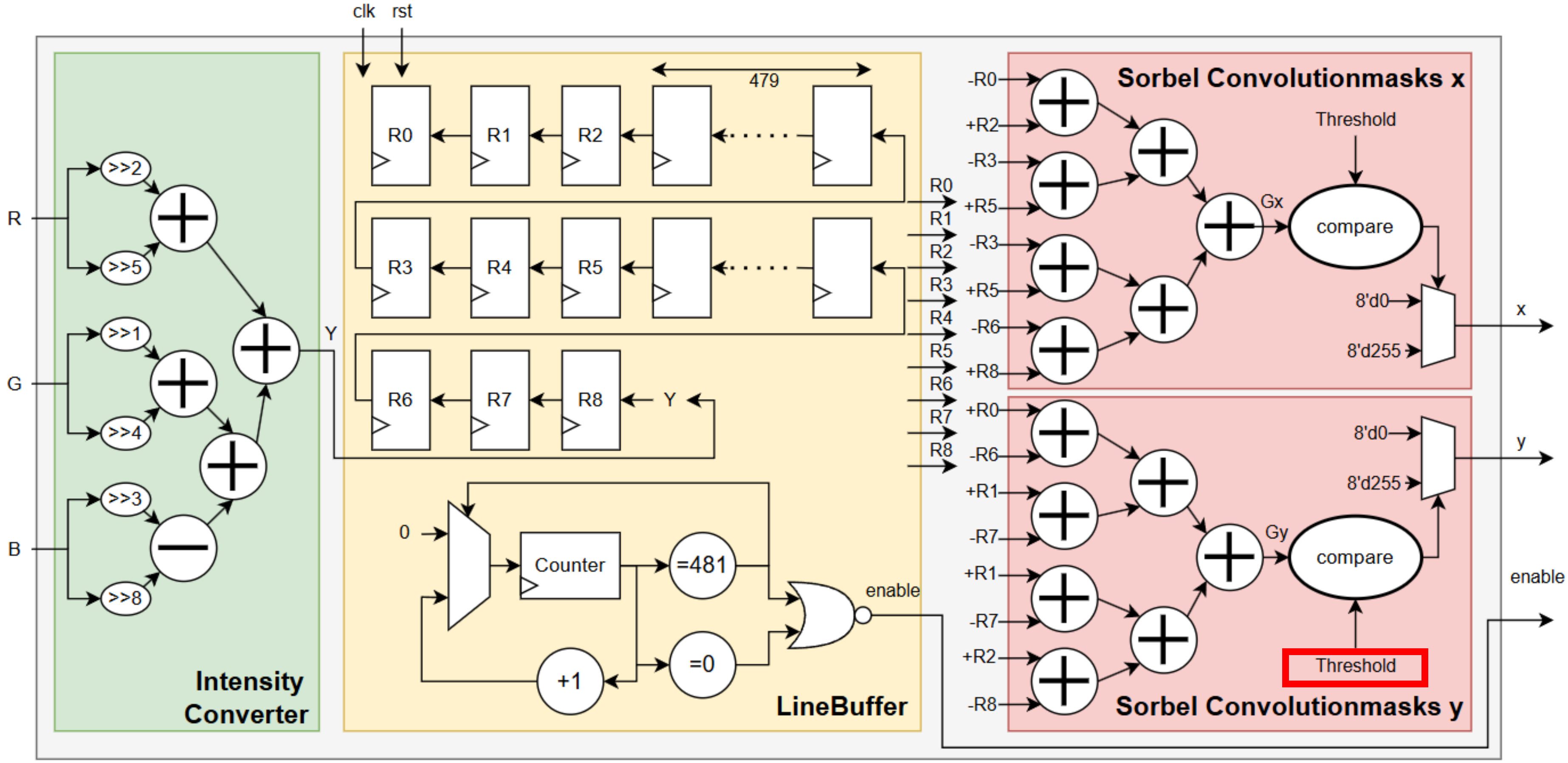
Testbench – Zero Padding



Bmp Address

Input Cycle
(with zero padding)

Hardware – Overall Architecture

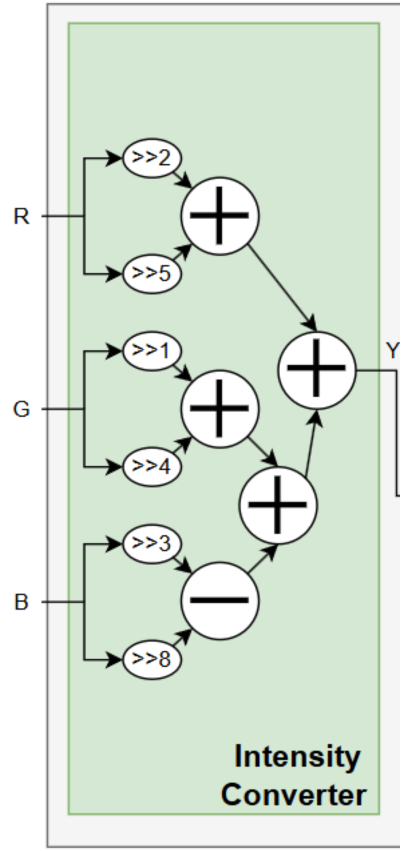


請務必寫上你設定的閾值

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



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

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

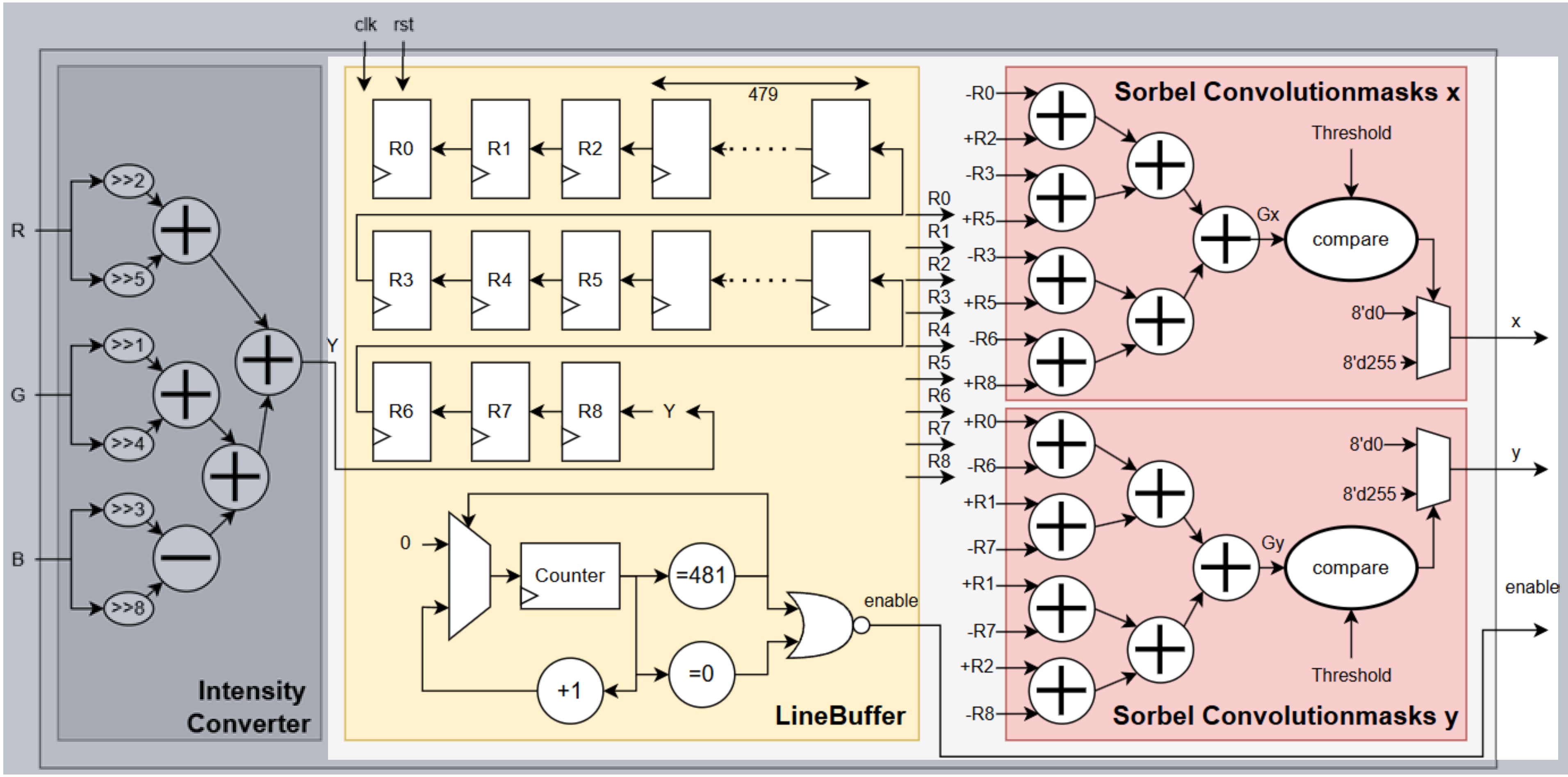
$R * 0.255 = R * 2^{-2} + R * 2^{-5} = R >> 2 + R >> 5$

請自行類推其他兩個

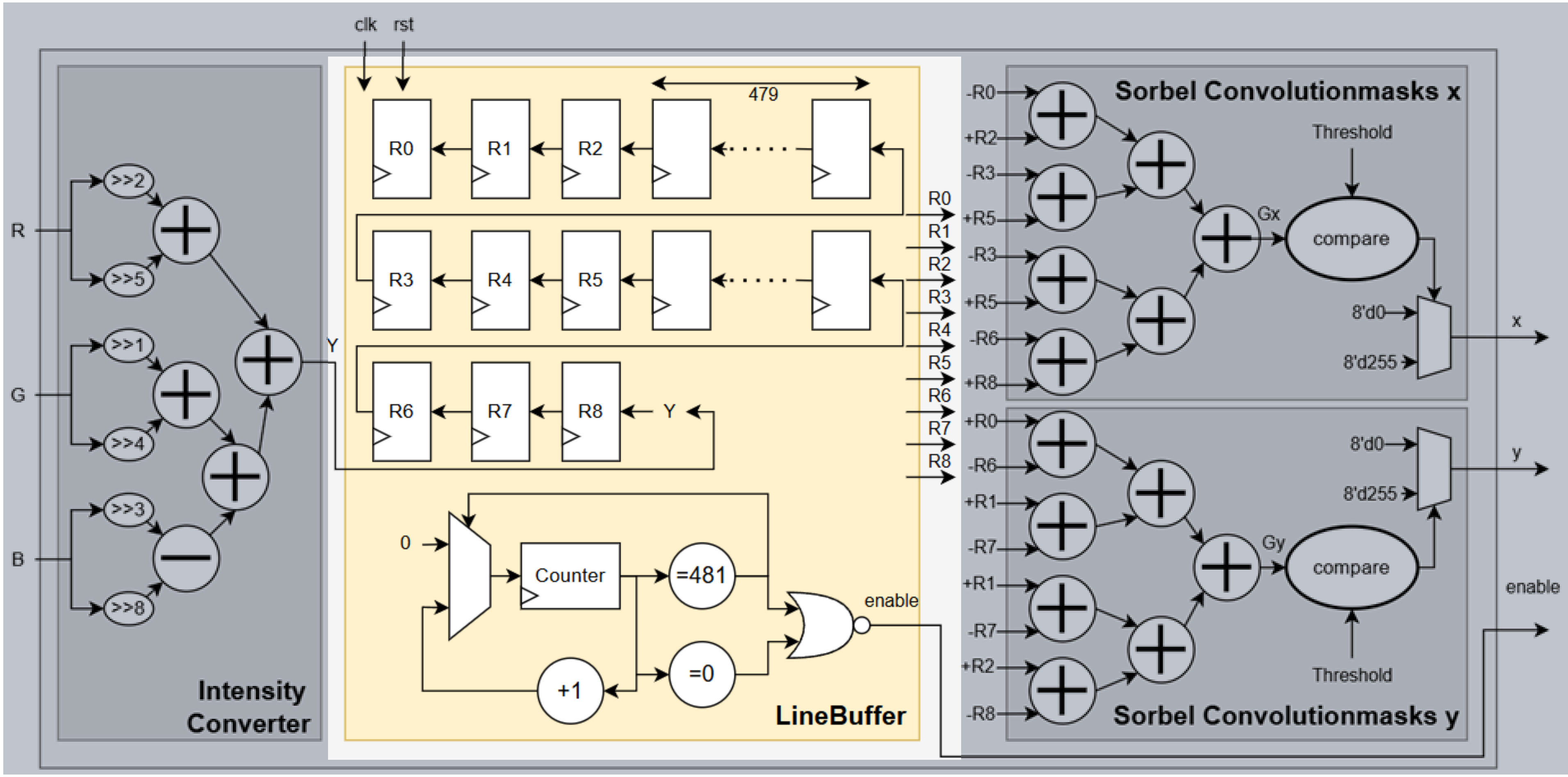
Outline

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

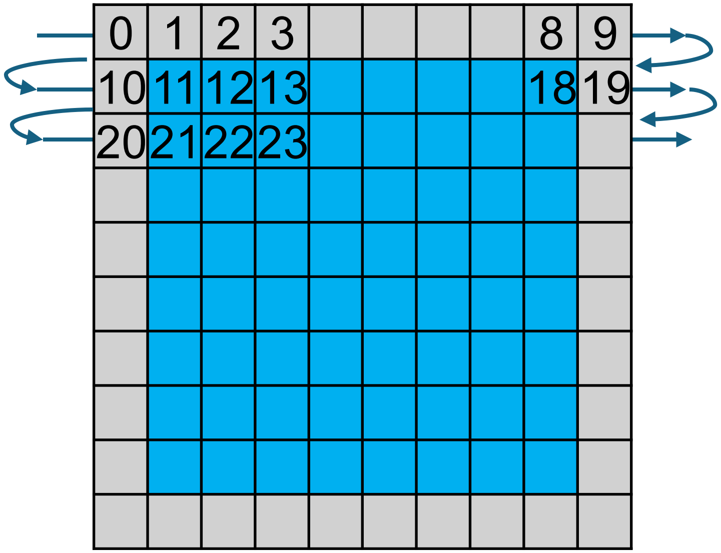
Hardware – Line Buffer & Convolution



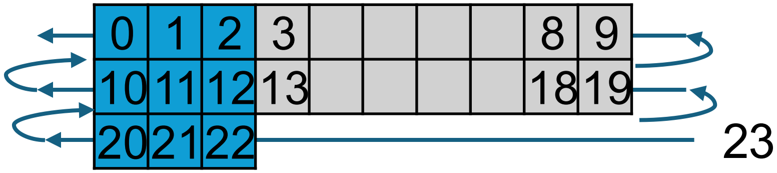
Hardware – Line Buffer



Hardware – Line Buffer

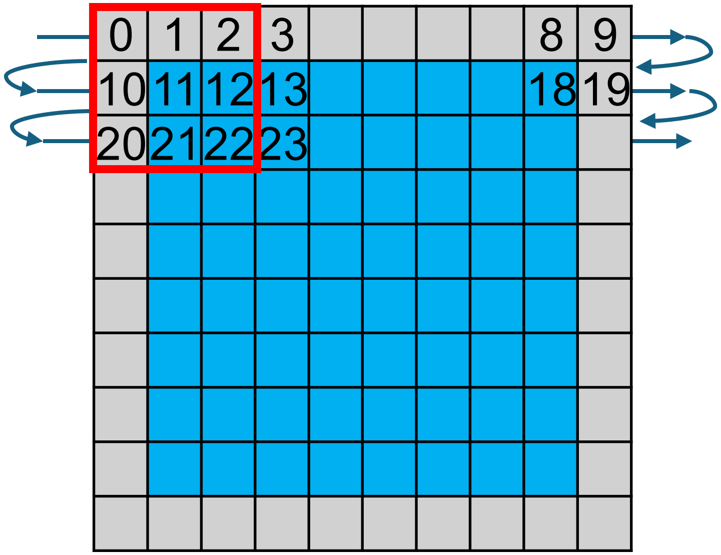


Input Cycle
(with zero padding)

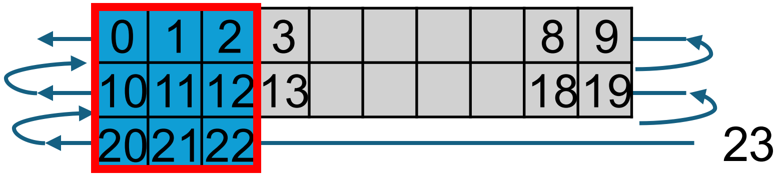


Line Buffer
(Shift Register)

Hardware – Line Buffer



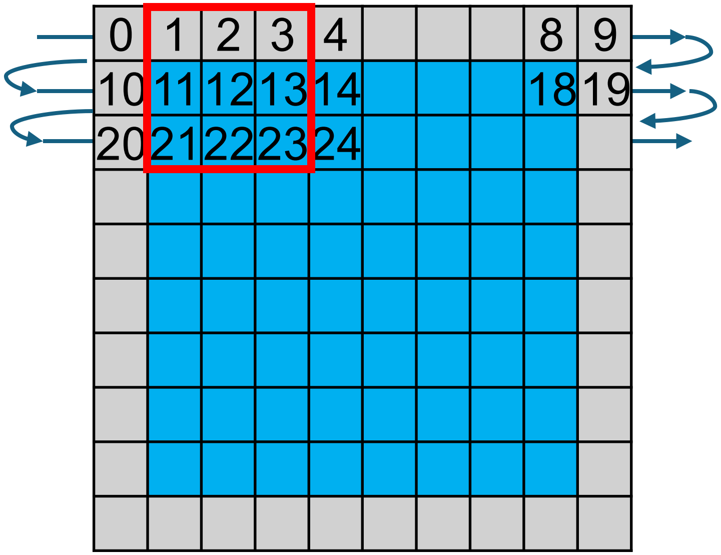
Input Cycle
(with zero padding)



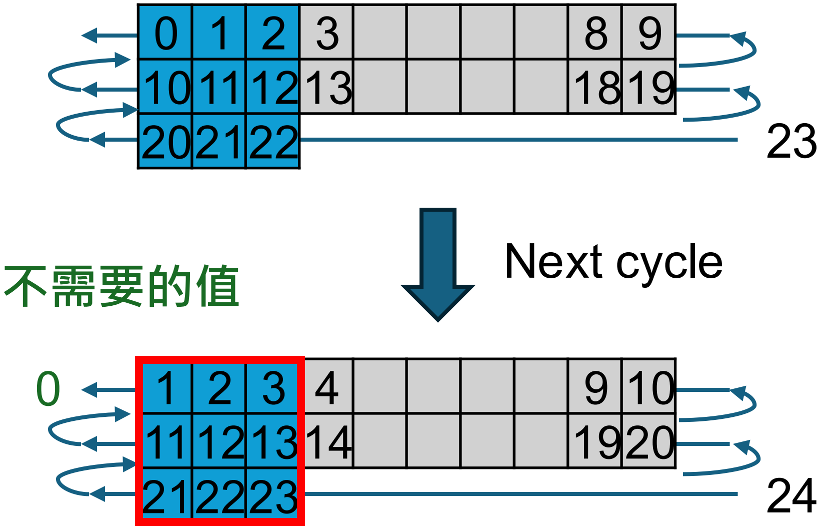
取這裡的值出來

Line Buffer
(Shift Register)

Hardware – Line Buffer



Input Cycle
(with zero padding)

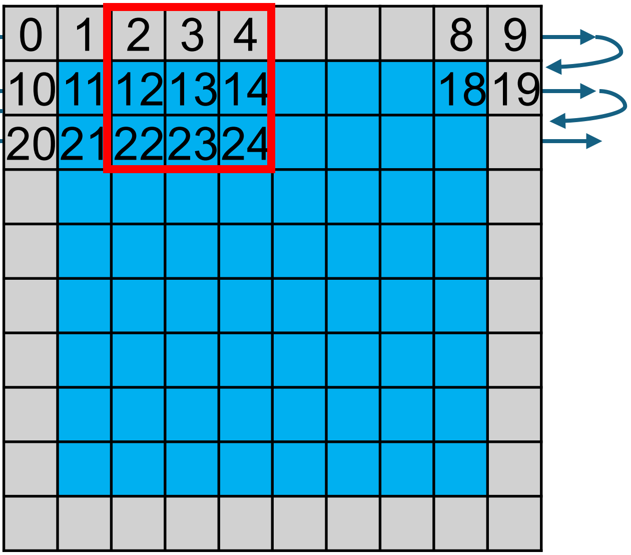


推出不需要的值

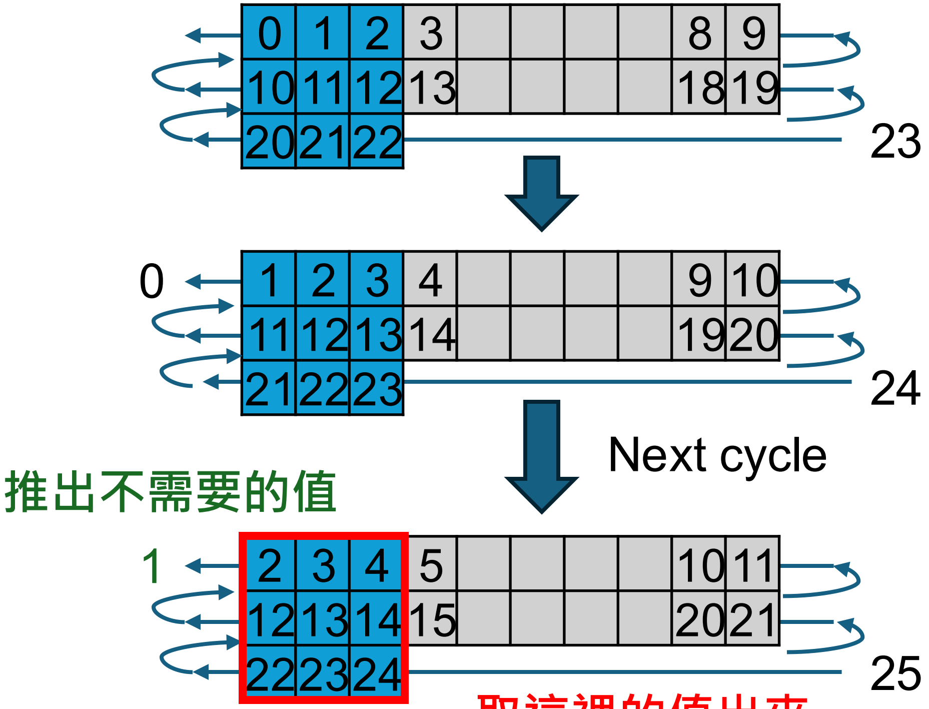
取這裡的值出來

Line Buffer
(Shift Register)

Hardware – Line Buffer



Input Cycle
(with zero padding)

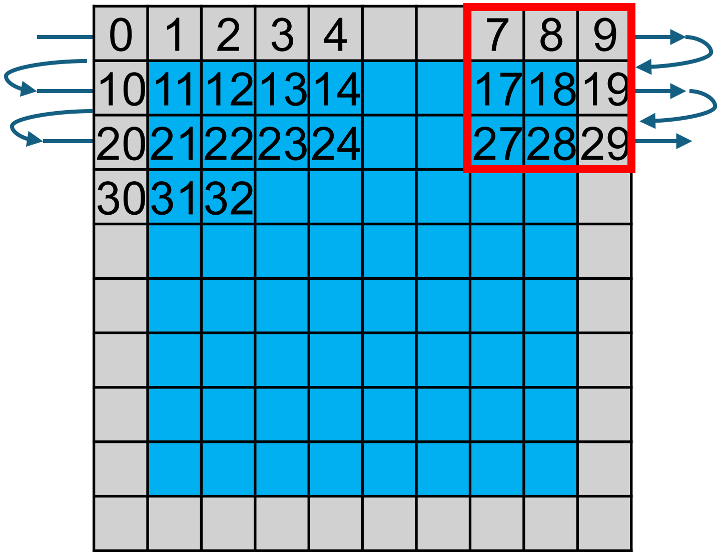


推出不需要的值

取這裡的值出來

Line Buffer
(Shift Register)

Hardware – Line Buffer



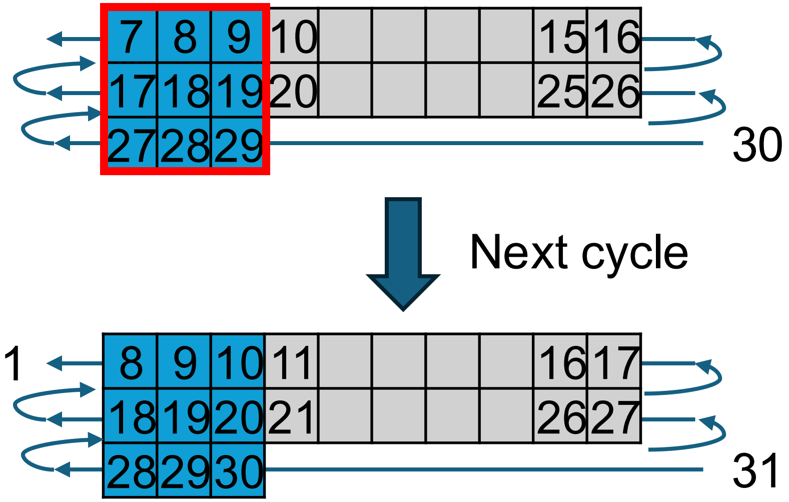
0 1 2 3 4 7 8 9

10 11 12 13 14 17 18 19

20 21 22 23 24 27 28 29

30 31 32

Input Cycle
(with zero padding)



7 8 9 10 15 16

17 18 19 20 25 26

27 28 29

30

Next cycle

1

8 9 10 11 16 17

18 19 20 21 26 27

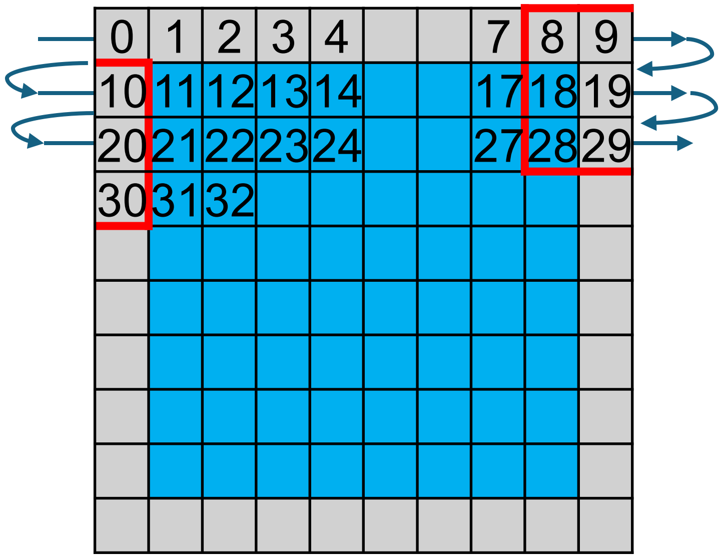
28 29 30

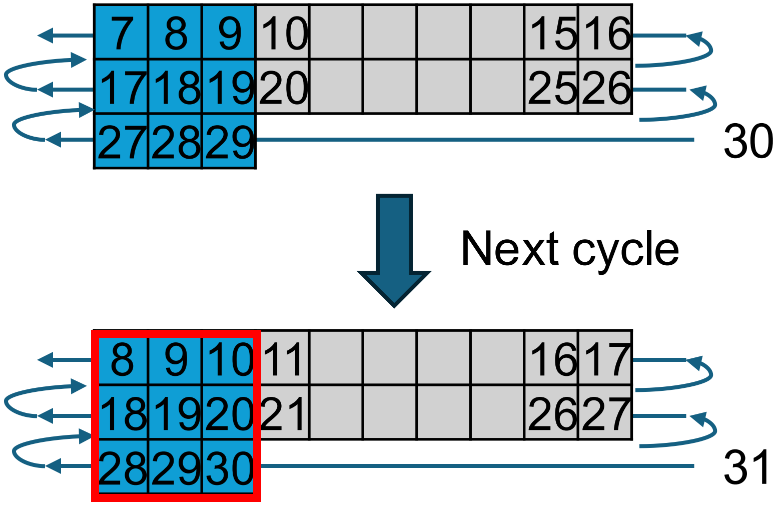
31

Line Buffer
(Shift Register)

Hardware – Line Buffer

Invalid!





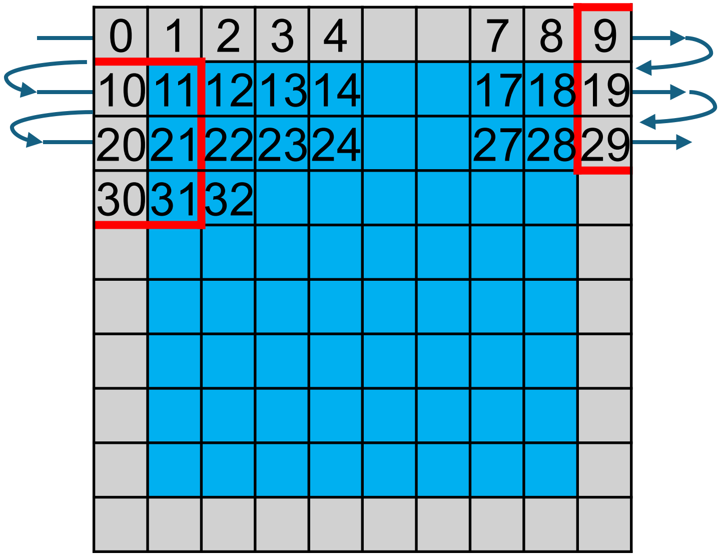
Invalid 需要跳過

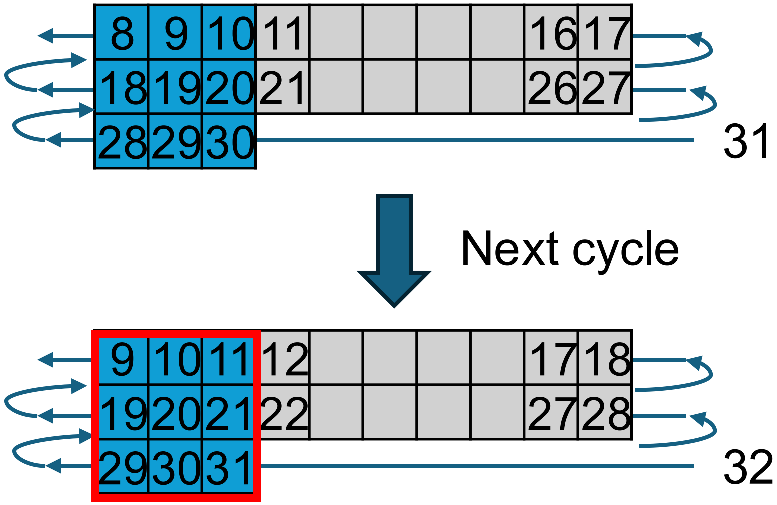
Input Cycle
(with zero padding)

Line Buffer
(Shift Register)

Hardware – Line Buffer

Invalid!



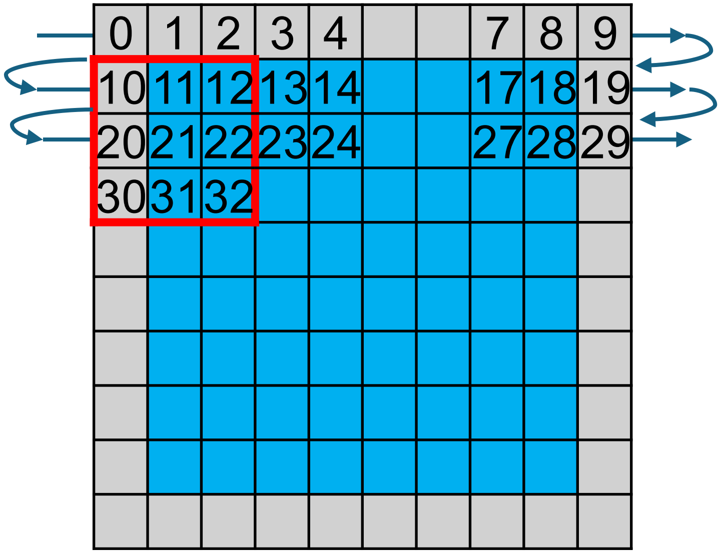


Invalid 需要跳過

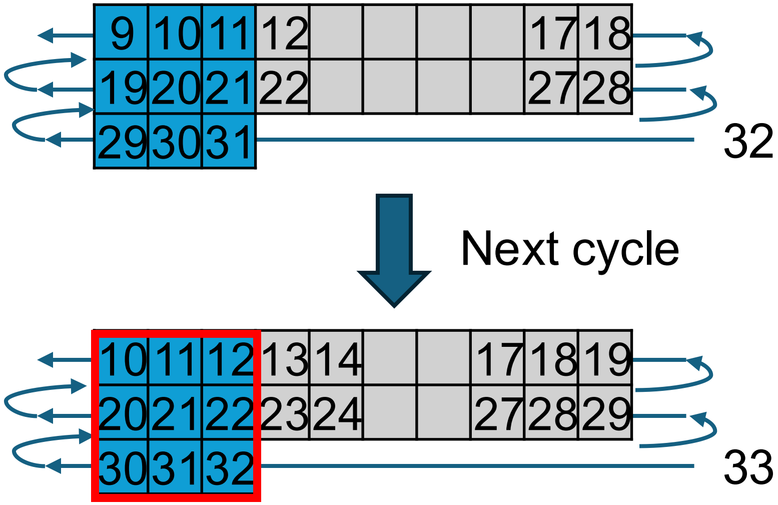
Input Cycle
(with zero padding)

Line Buffer
(Shift Register)

Hardware – Line Buffer

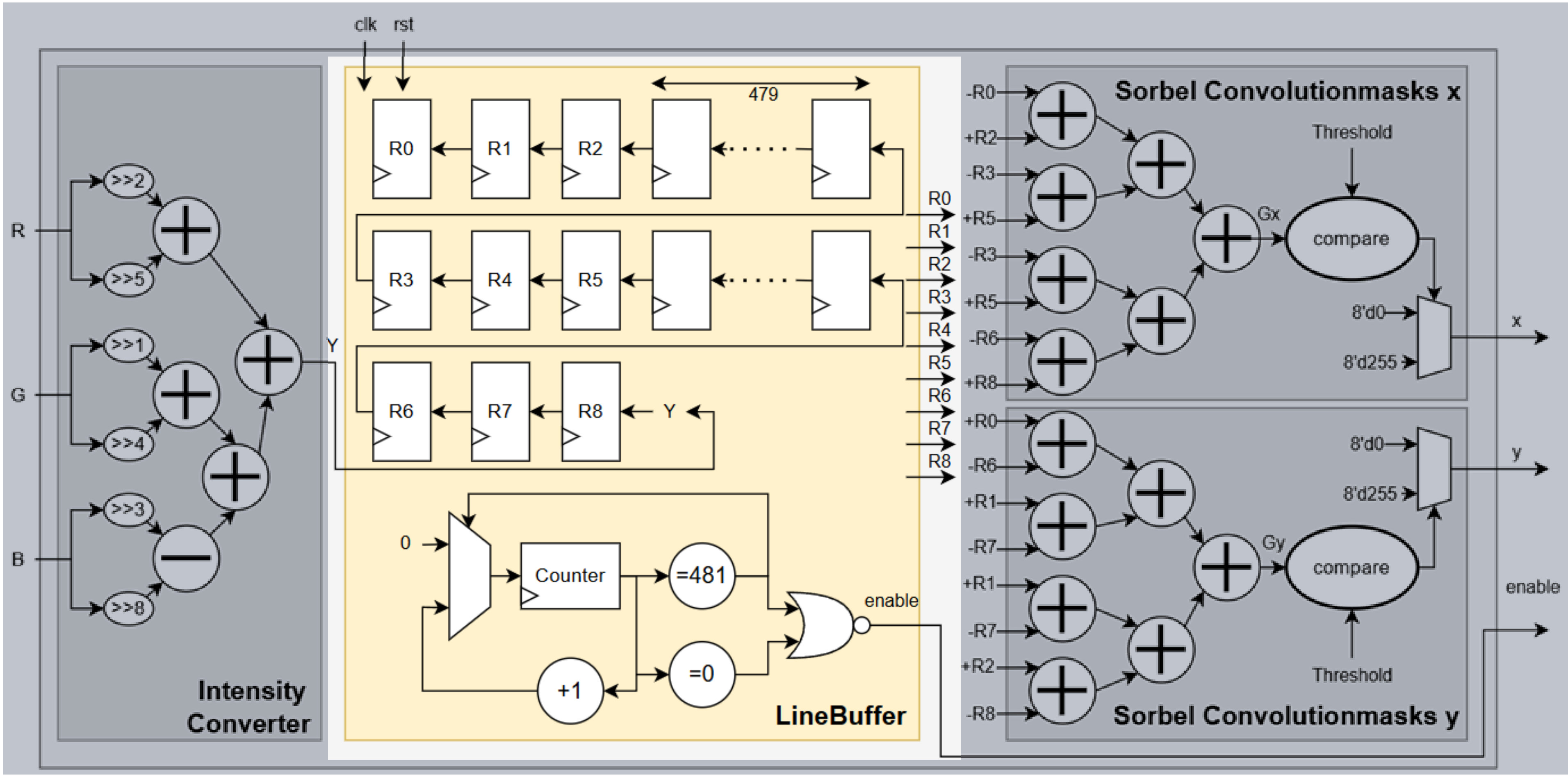


Input Cycle
(with zero padding)

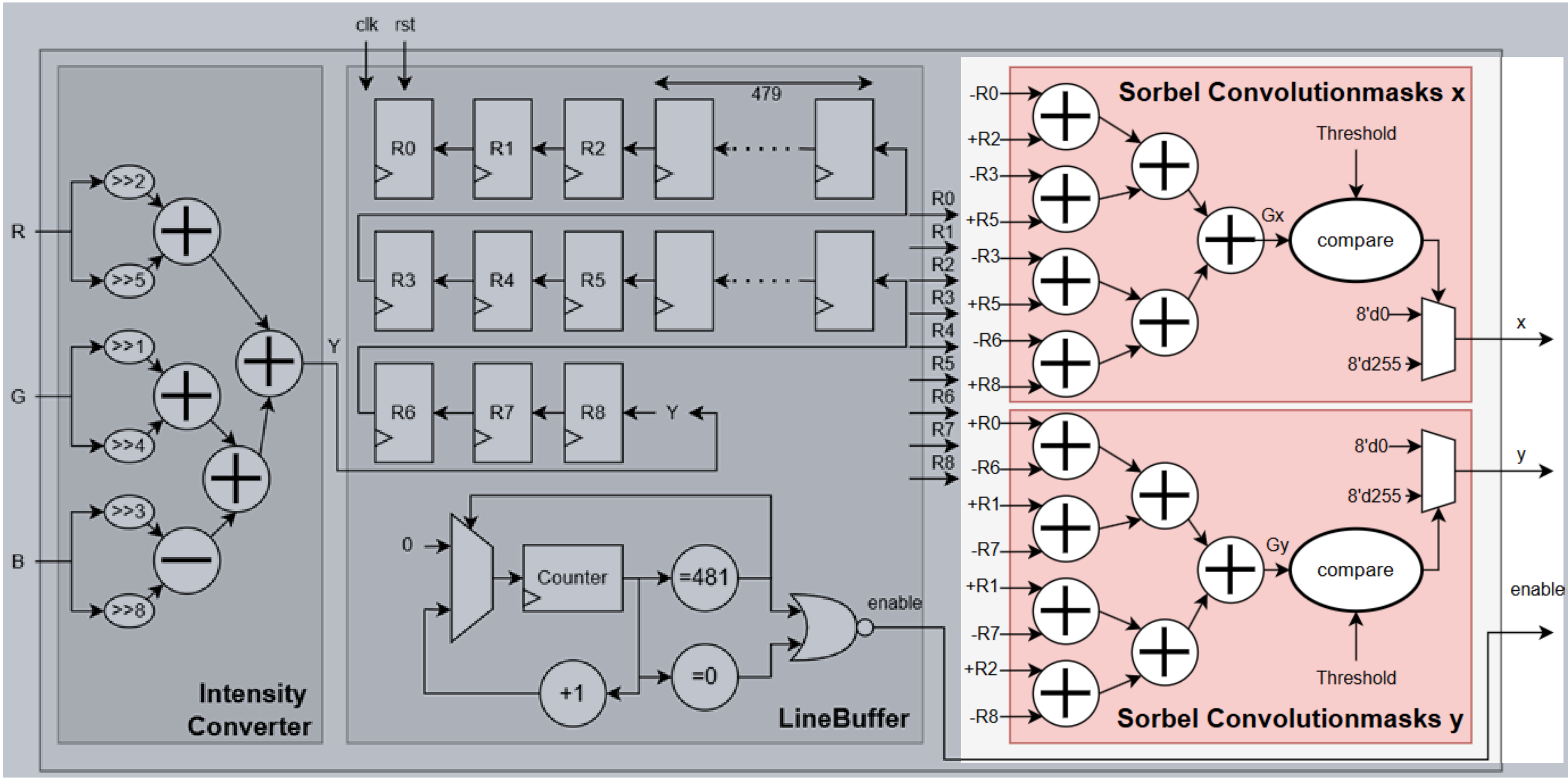


Line Buffer
(Shift Register)

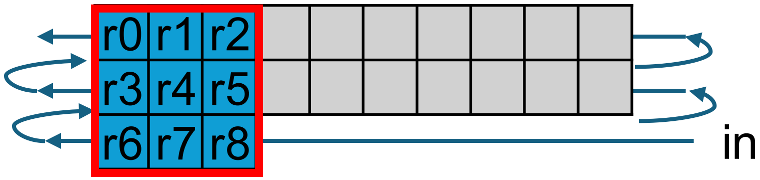
Hardware – Line Buffer



Hardware – Convolution



Hardware – Convolution



×

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

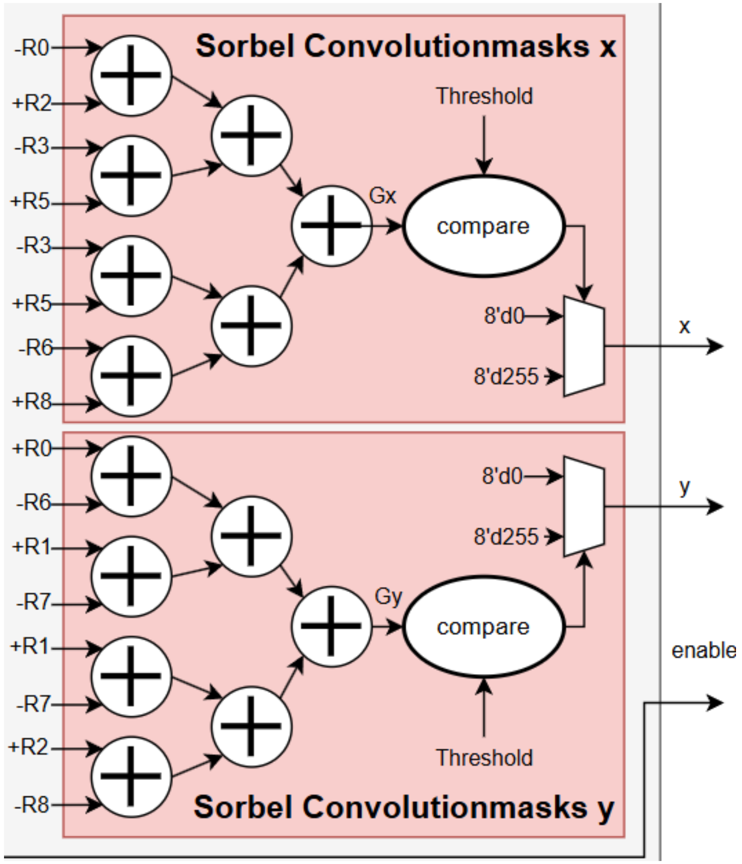
Line Buffer
(Shift Register)

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

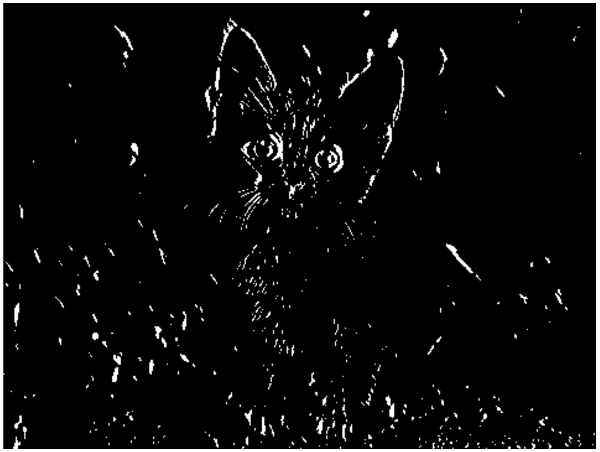
0 ~ 255

convolution

?

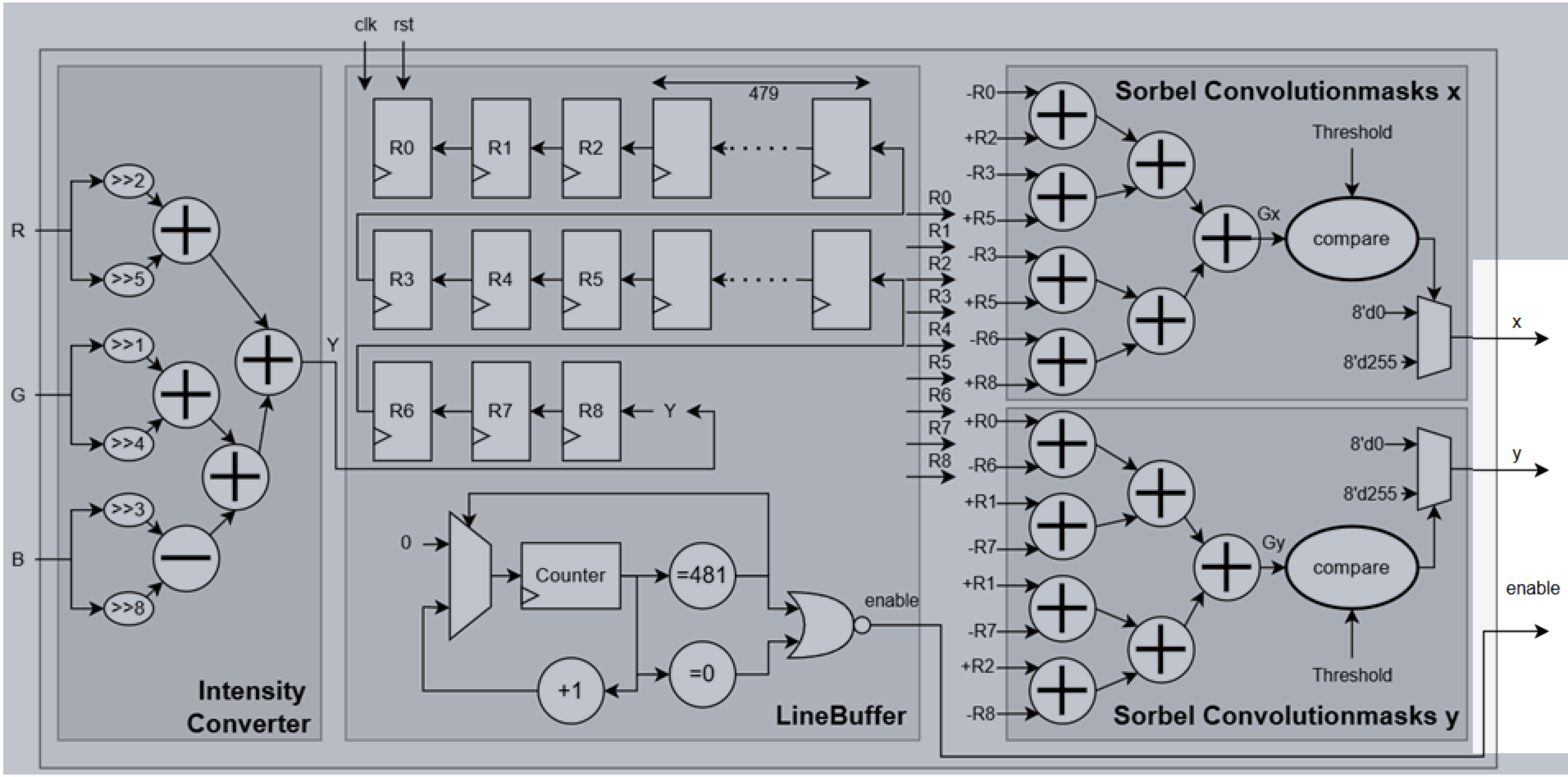
-1020
~
+1020

Compare with Threshold
Pixel Value > Threshold → 255 **(White)**
Pixel Value ≤ Threshold → 0 **(Black)**
*** 請自己調整 Threshold**

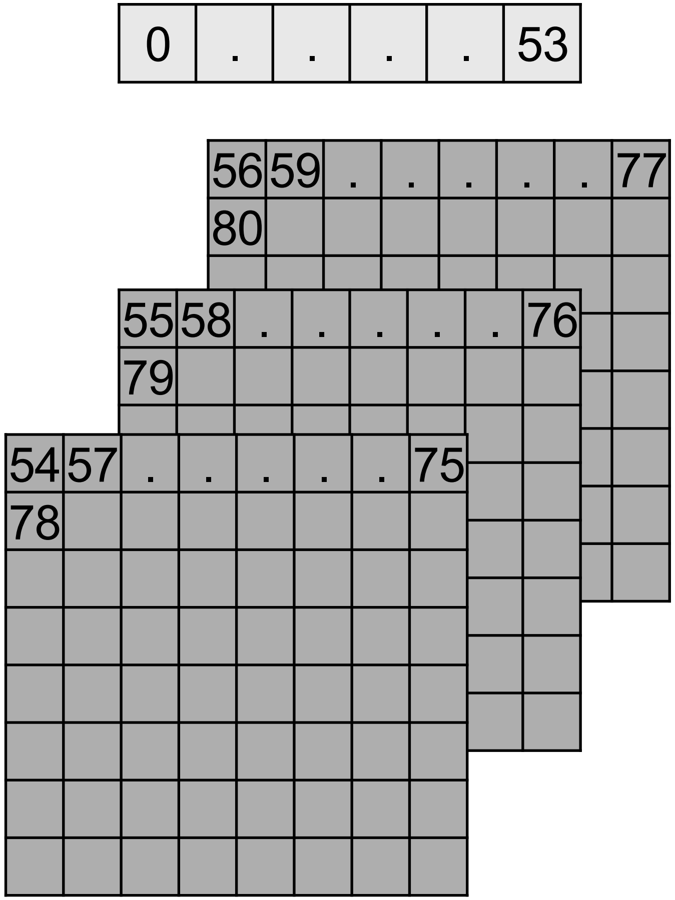


0 or 255
Only white or black

Testbench – Write Image File



Testbench – Write Image File



PDF Report

- Cell Base Part
- Xilinx Vivado Part
- Comments (心得)

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 (Xilinx Vivado)

- 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)

* **Please refer to the HW4 TA's Vivado slides.**

評分檔案(請按照此命名規則)

- HW4
  - Edge.v
  - HW4_TB_fixed_pre.v
  - HW4_TB_fixed_post.v
  - cat.bmp, cat_x.bmp, cat_y.bmp (Picture *3)
  - SYN
    - Timing report, Area report, Power report
    - Edge_syn.v
    - Edge_syn.ddc, Edge_syn.sdc, Edge_syn.sdf
  - Vivado
    - HDL_HW4_學號.xpr.zip
    - Edge.xdc, HDL_HW4_func_impl.wcfg, HDL_HW4_behav.wcfg
  - Report.pdf

繳交檔案

1. Server內繳交

   ① 整個 HW4 資料夾

   ② [指令]

   - cp -r /MasterClass/M143040045_HDL/HW4/ /MasterClass/Homework-Submit/M143040045_HDL/
   - 請替換 M143040045_HDL 成自己的學號

2. 網路大學繳交 Report (只收PDF)

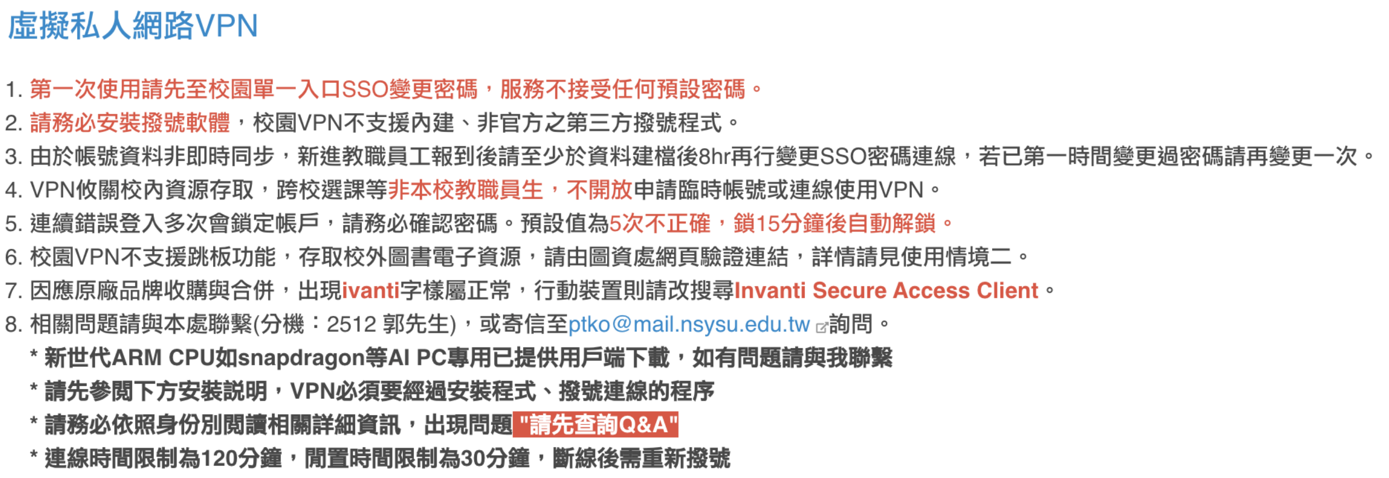
*** Server 與 網路大學 都有繳交才會計分**

評分說明

- **Cell Base (50%)**
  - Verilog RTL code **(30%)**
  - Pre-sim testbench **(10%)**
  - Gate-Level code **(5%)** **（ Delay optimization only ）**
  - Post-sim testbench **(5%)**
- **Xilinx Vivado (15%)**
  - HDL_HW4_MXXXXXXXXX.xpr.zip **(10%)**
  - xdc, wcfg **(5%)**
- **Report (35%)**

注意事項 - 1

- **請勿佔用 Server資源**
  - **開啟過多 Terminal**
  - **使用後不登出**
  - **學校 VPN 連線限制 120 分鐘，請注意連線續使用時間**



虛擬私人網路VPN

1. 第一次使用請先至校園單一入口SSO變更密碼，服務不接受任何預設密碼。
2. 請務必安裝撥號軟體，校園VPN不支援內建、非官方之第三方撥號程式。
3. 由於帳號資料非即時同步，新進教職員工報到後請至少於資料建檔後8hr再行變更SSO密碼連線，若已第一時間變更過密碼請再變更一次。
4. VPN攸關校內資源存取，跨校選課等非本校教職員生，不開放申請臨時帳號或連線使用VPN。
5. 連續錯誤登入多次會鎖定帳戶，請務必確認密碼。預設值為5次不正確，鎖15分鐘後自動解鎖。
6. 校園VPN不支援跳板功能，存取校外圖書電子資源，請由圖資處網頁驗證連結，詳情請見使用情境二。
7. 因應原廠品牌收購與合併，出現**ivanti**字樣屬正常，行動裝置則請改搜尋**Invanti Secure Access Client**。
8. 相關問題請與本處聯繫(分機：2512 郭先生)，或寄信至ptko@mail.nsysu.edu.tw 詢問。

   *** 新世代ARM CPU如snapdragon等AI PC專用已提供用戶端下載，如有問題請與我聯繫**
   *** 請先參閱下方安裝說明，VPN必須要經過安裝程式、撥號連線的程序**
   *** 請務必依照身份別閱讀相關詳細資訊，出現問題 "請先查詢Q&A"**
   *** 連線時間限制為120分鐘，閒置時間限制為30分鐘，斷線後需重新撥號**

注意事項 - 2

- **使用 EDA Tool 前務必先輸入 tcsh**
- **助教會協助 Server 使用上的問題**
    - **請同學自行解決 RTL Code Bug**