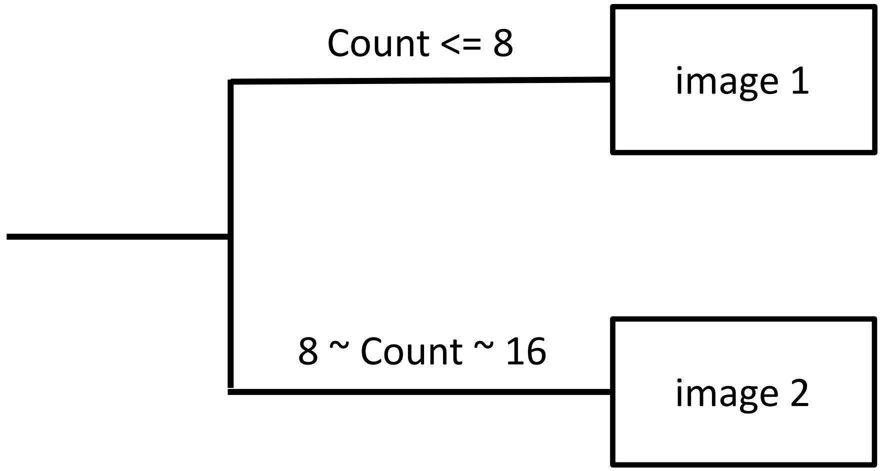
Lab04 Code Review

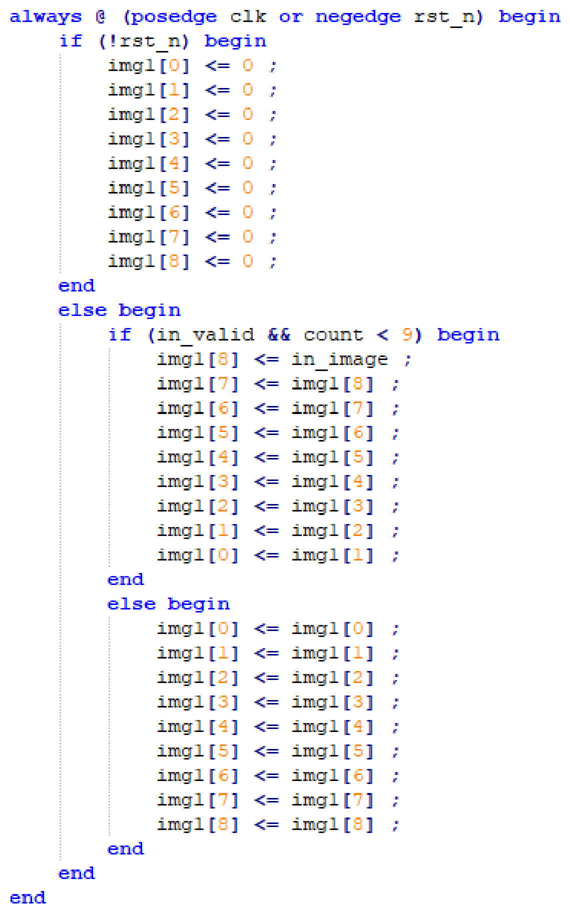
Review

$$\underset{\text{img1}}{\begin{bmatrix} 15 & 10 & 11 \\ 12 & 12 & 14 \\ 13 & 13 & 13 \end{bmatrix}} - \underset{\text{img2}}{\begin{bmatrix} 9 & 5 & 1 \\ 6 & 4 & 3 \\ 5 & 9 & 1 \end{bmatrix}} = \begin{bmatrix} 6 & 5 & 10 \\ 6 & 8 & 11 \\ 8 & 4 & 12 \end{bmatrix}$$

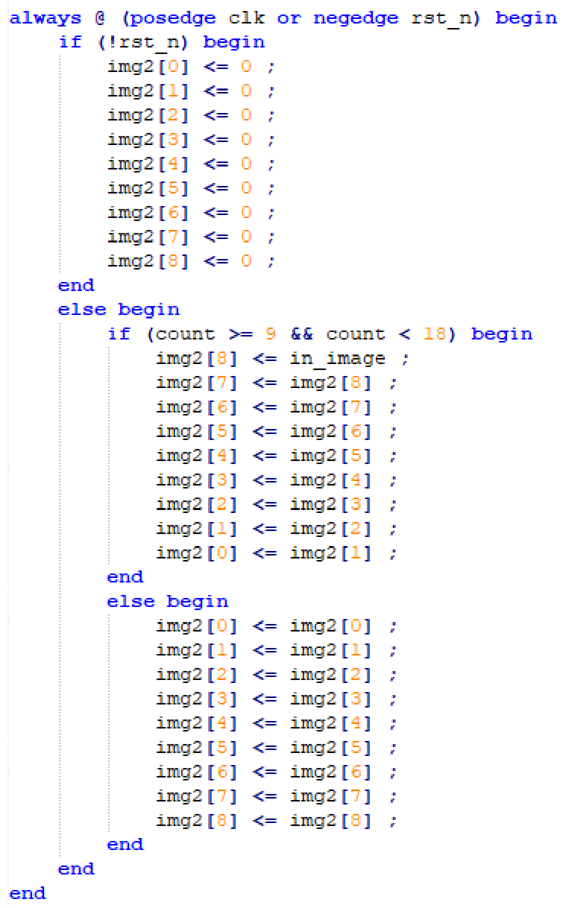
Block Diagram



Reference Code (Store Data)

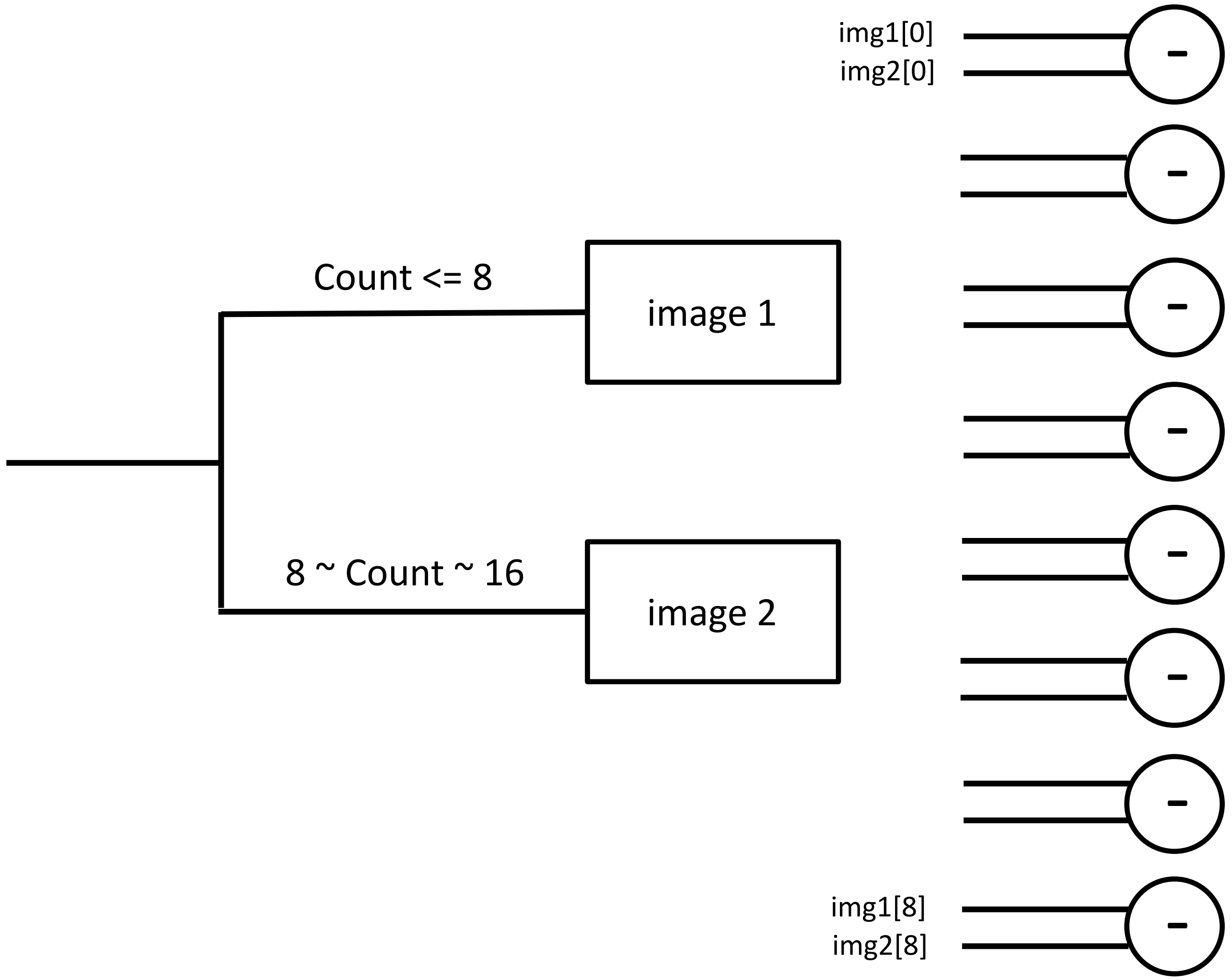


```verilog
always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        img1[0] <= 0 ;
        img1[1] <= 0 ;
        img1[2] <= 0 ;
        img1[3] <= 0 ;
        img1[4] <= 0 ;
        img1[5] <= 0 ;
        img1[6] <= 0 ;
        img1[7] <= 0 ;
        img1[8] <= 0 ;
    end
    else begin
        if (in_valid && count < 9) begin
            img1[8] <= in_image ;
            img1[7] <= img1[8] ;
            img1[6] <= img1[7] ;
            img1[5] <= img1[6] ;
            img1[4] <= img1[5] ;
            img1[3] <= img1[4] ;
            img1[2] <= img1[3] ;
            img1[1] <= img1[2] ;
            img1[0] <= img1[1] ;
        end
        else begin
            img1[0] <= img1[0] ;
            img1[1] <= img1[1] ;
            img1[2] <= img1[2] ;
            img1[3] <= img1[3] ;
            img1[4] <= img1[4] ;
            img1[5] <= img1[5] ;
            img1[6] <= img1[6] ;
            img1[7] <= img1[7] ;
            img1[8] <= img1[8] ;
        end
    end
end
```

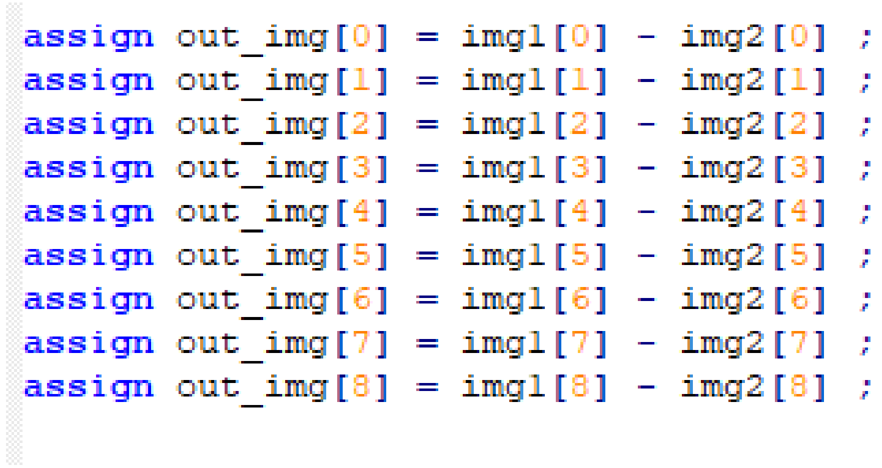


```verilog
always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        img2[0] <= 0 ;
        img2[1] <= 0 ;
        img2[2] <= 0 ;
        img2[3] <= 0 ;
        img2[4] <= 0 ;
        img2[5] <= 0 ;
        img2[6] <= 0 ;
        img2[7] <= 0 ;
        img2[8] <= 0 ;
    end
    else begin
        if (count >= 9 && count < 18) begin
            img2[8] <= in_image ;
            img2[7] <= img2[8] ;
            img2[6] <= img2[7] ;
            img2[5] <= img2[6] ;
            img2[4] <= img2[5] ;
            img2[3] <= img2[4] ;
            img2[2] <= img2[3] ;
            img2[1] <= img2[2] ;
            img2[0] <= img2[1] ;
        end
        else begin
            img2[0] <= img2[0] ;
            img2[1] <= img2[1] ;
            img2[2] <= img2[2] ;
            img2[3] <= img2[3] ;
            img2[4] <= img2[4] ;
            img2[5] <= img2[5] ;
            img2[6] <= img2[6] ;
            img2[7] <= img2[7] ;
            img2[8] <= img2[8] ;
        end
    end
end
```

Block Diagram

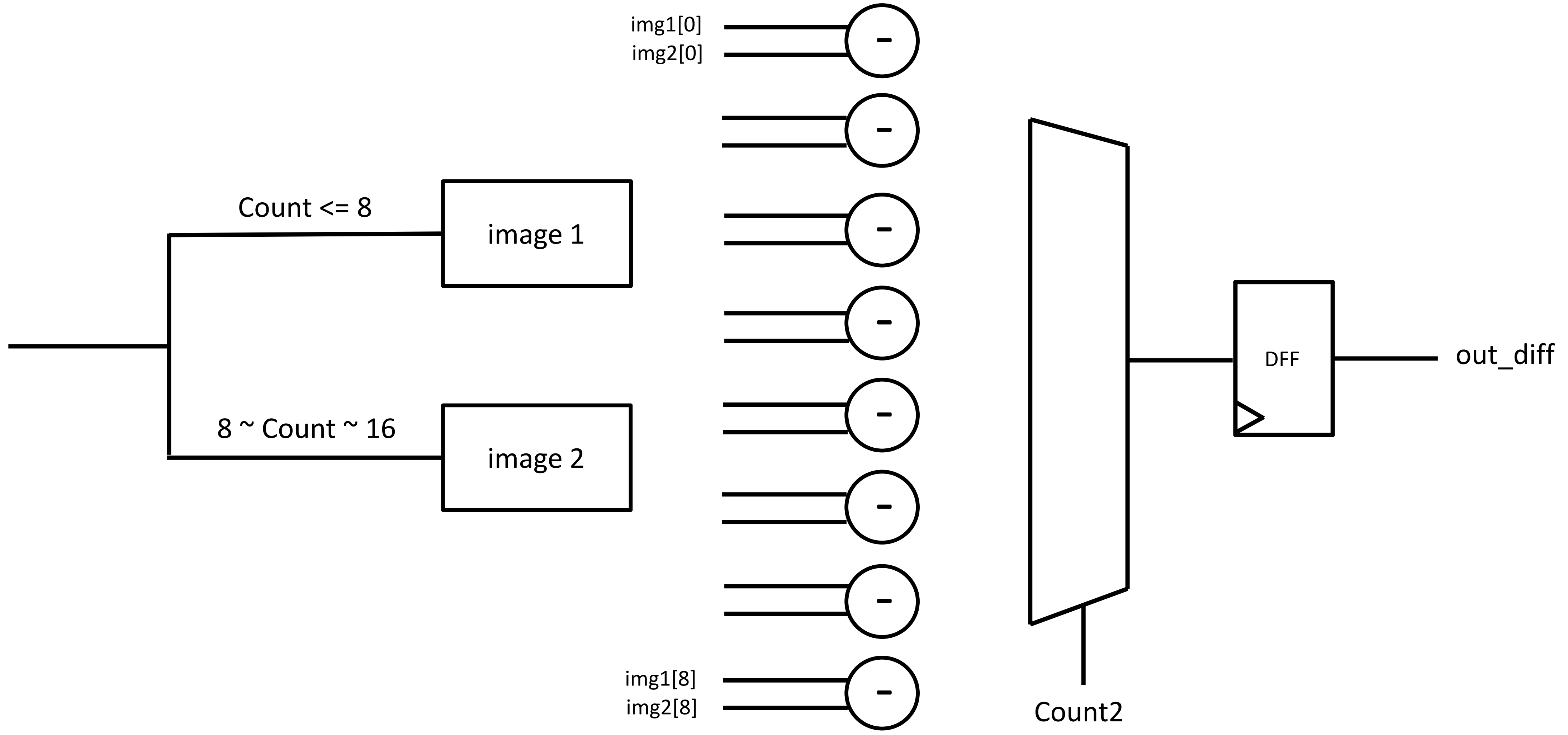


Reference Code (Implement sub)

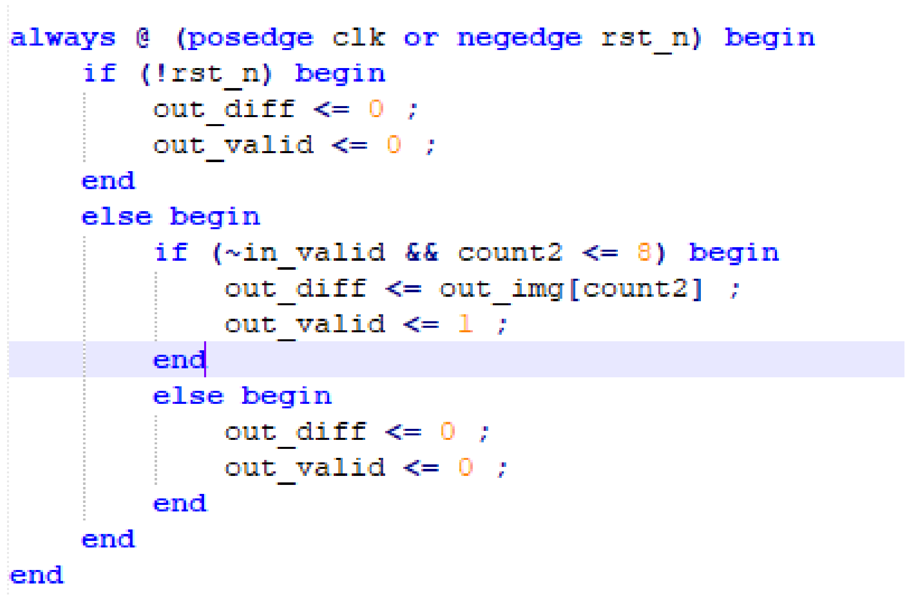


```
assign out_img[0] = img1[0] - img2[0] ;
assign out_img[1] = img1[1] - img2[1] ;
assign out_img[2] = img1[2] - img2[2] ;
assign out_img[3] = img1[3] - img2[3] ;
assign out_img[4] = img1[4] - img2[4] ;
assign out_img[5] = img1[5] - img2[5] ;
assign out_img[6] = img1[6] - img2[6] ;
assign out_img[7] = img1[7] - img2[7] ;
assign out_img[8] = img1[8] - img2[8] ;
```

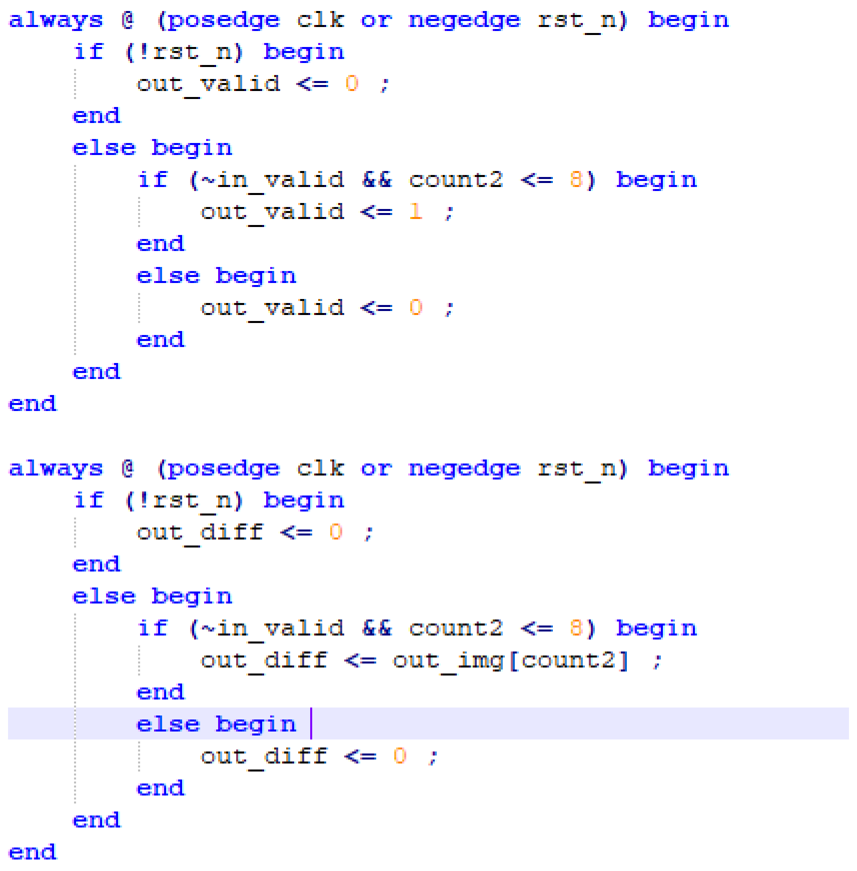
Block Diagram



Reference Code (output)



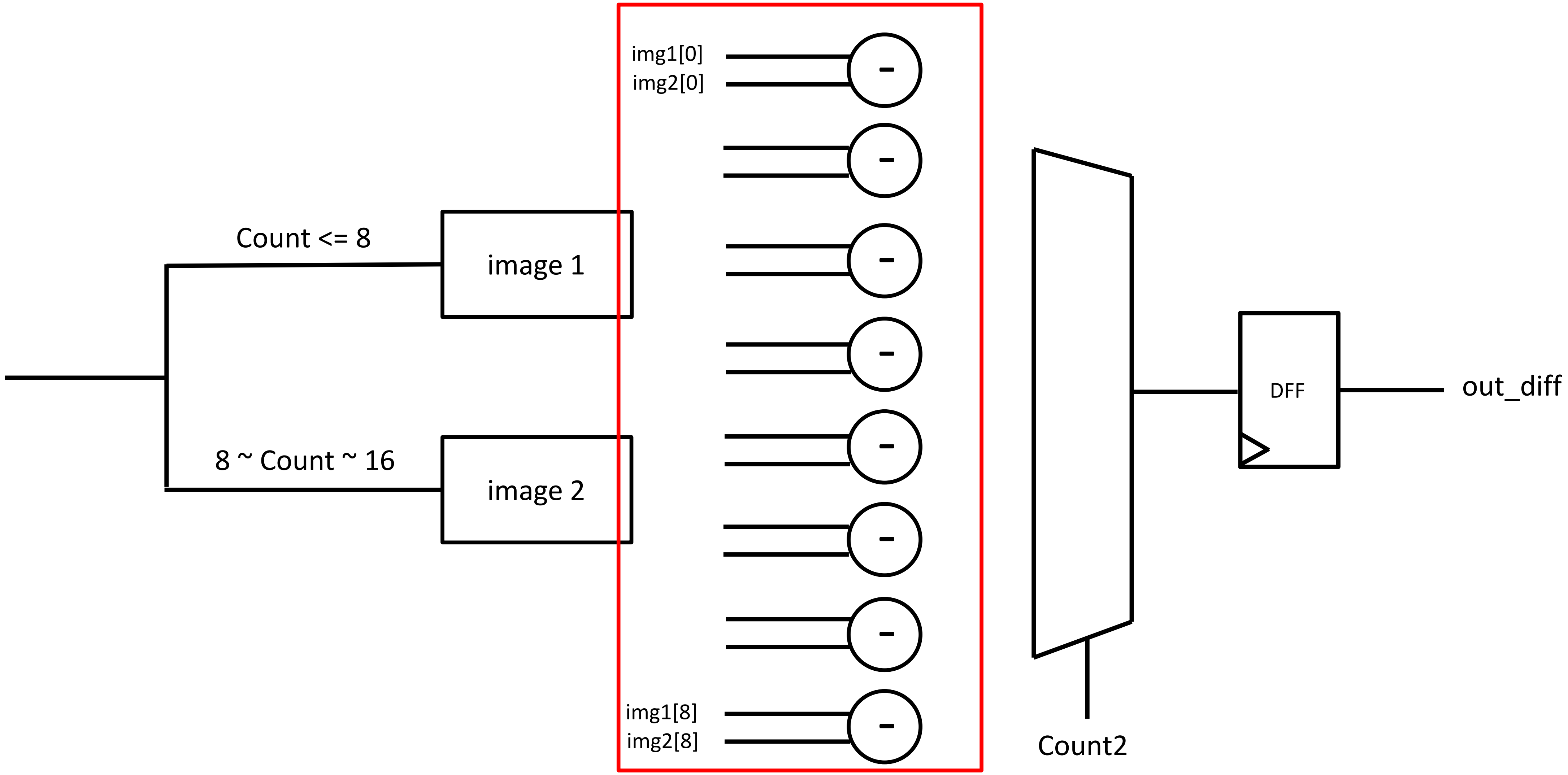
```
always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        out_diff <= 0 ;
        out_valid <= 0 ;
    end
    else begin
        if (~in_valid && count2 <= 8) begin
            out_diff <= out_img[count2] ;
            out_valid <= 1 ;
        end
        else begin
            out_diff <= 0 ;
            out_valid <= 0 ;
        end
    end
end
```



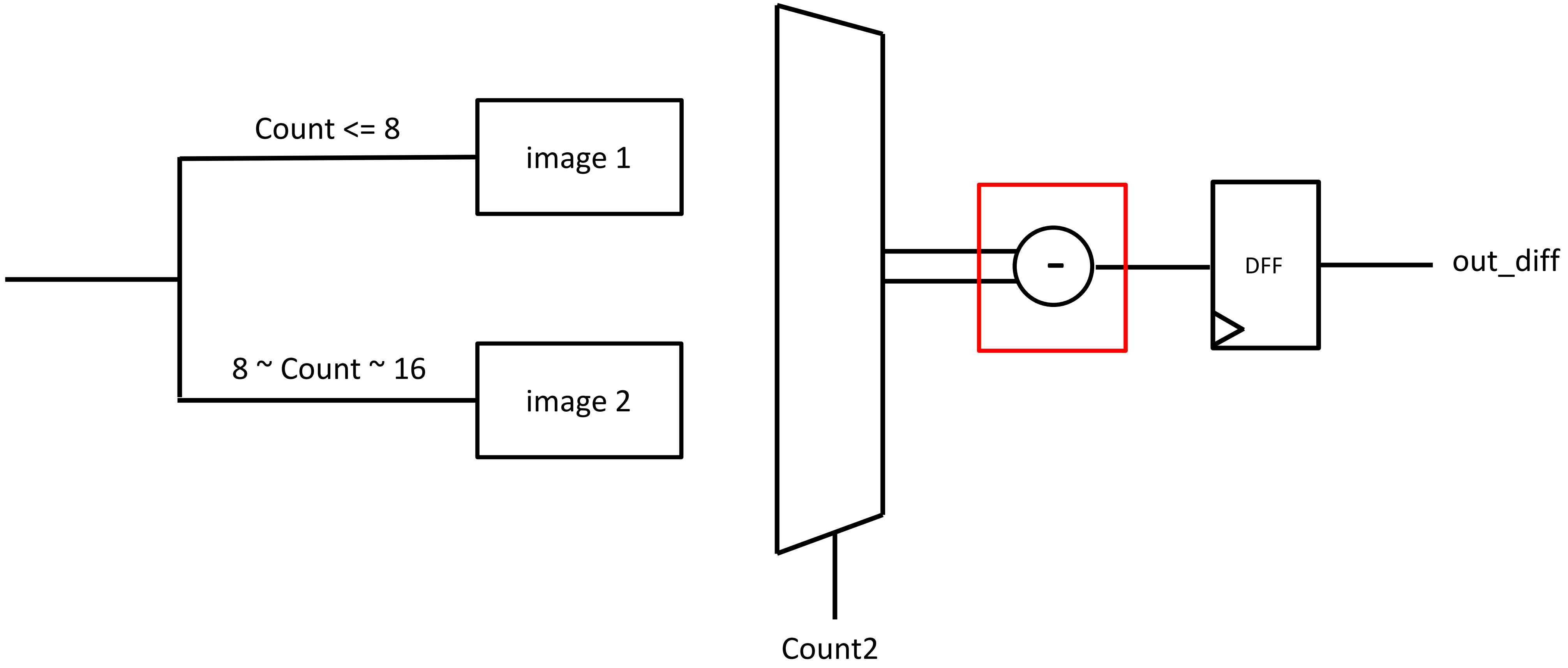
```
always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        out_valid <= 0 ;
    end
    else begin
        if (~in_valid && count2 <= 8) begin
            out_valid <= 1 ;
        end
        else begin
            out_valid <= 0 ;
        end
    end
end

always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        out_diff <= 0 ;
    end
    else begin
        if (~in_valid && count2 <= 8) begin
            out_diff <= out_img[count2] ;
        end
        else begin
            out_diff <= 0 ;
        end
    end
end
```

recommend

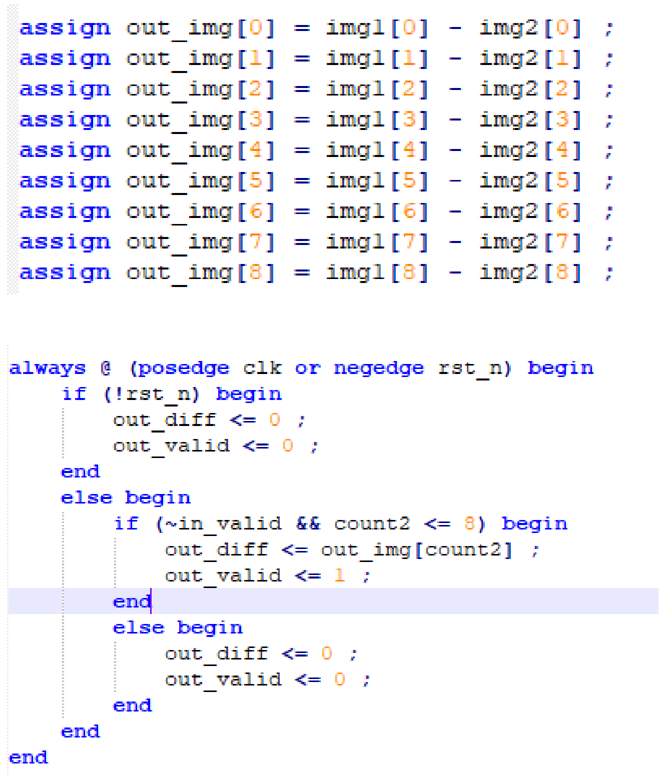
Optimization (Hardware Sharing)



Optimization (Hardware Sharing)

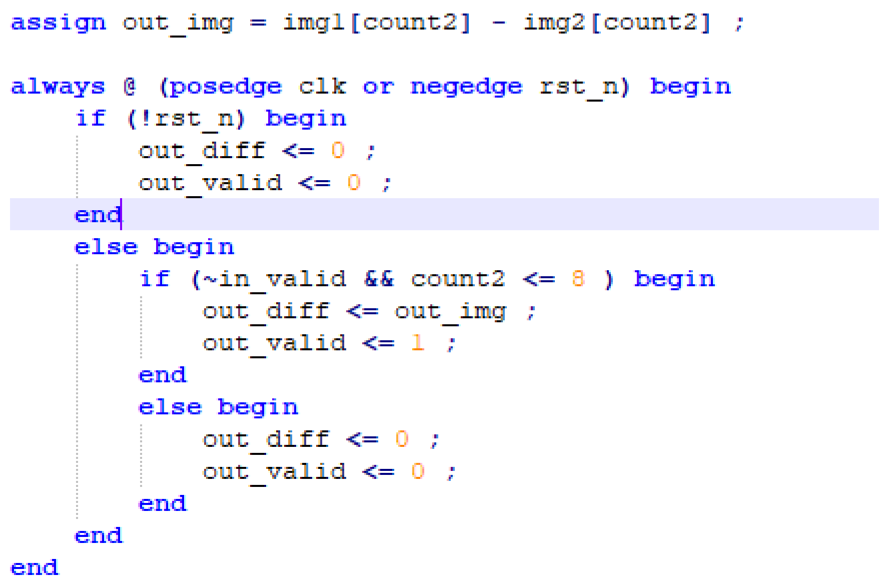


Reference Code



```
assign out_img[0] = img1[0] - img2[0] ;
assign out_img[1] = img1[1] - img2[1] ;
assign out_img[2] = img1[2] - img2[2] ;
assign out_img[3] = img1[3] - img2[3] ;
assign out_img[4] = img1[4] - img2[4] ;
assign out_img[5] = img1[5] - img2[5] ;
assign out_img[6] = img1[6] - img2[6] ;
assign out_img[7] = img1[7] - img2[7] ;
assign out_img[8] = img1[8] - img2[8] ;

always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        out_diff <= 0 ;
        out_valid <= 0 ;
    end
    else begin
        if (~in_valid && count2 <= 8) begin
            out_diff <= out_img[count2] ;
            out_valid <= 1 ;
        end
        else begin
            out_diff <= 0 ;
            out_valid <= 0 ;
        end
    end
end
```

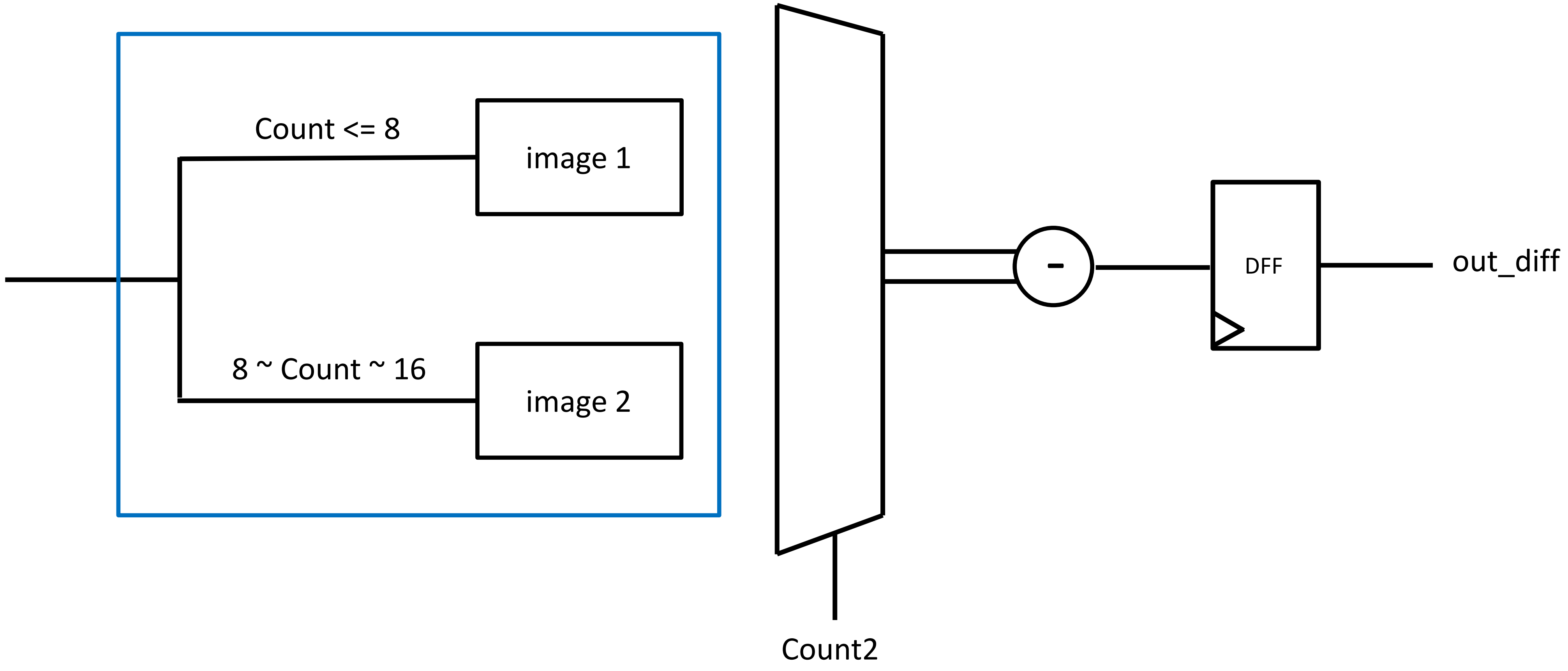
Before sharing



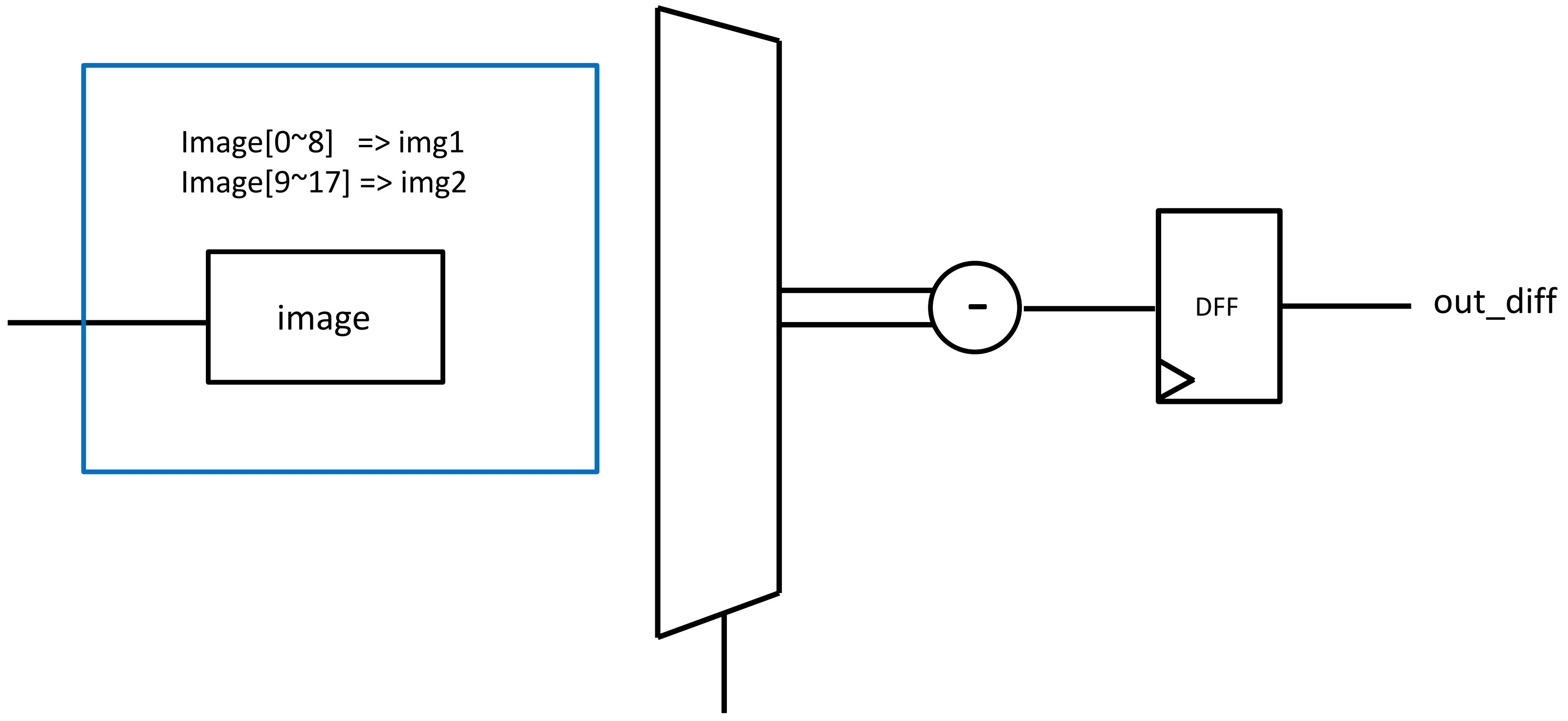
```
assign out_img = img1[count2] - img2[count2] ;

always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        out_diff <= 0 ;
        out_valid <= 0 ;
    end
    else begin
        if (~in_valid && count2 <= 8 ) begin
            out_diff <= out_img ;
            out_valid <= 1 ;
        end
        else begin
            out_diff <= 0 ;
            out_valid <= 0 ;
        end
    end
end
```

After Sharing

Optimization (Shift Register)

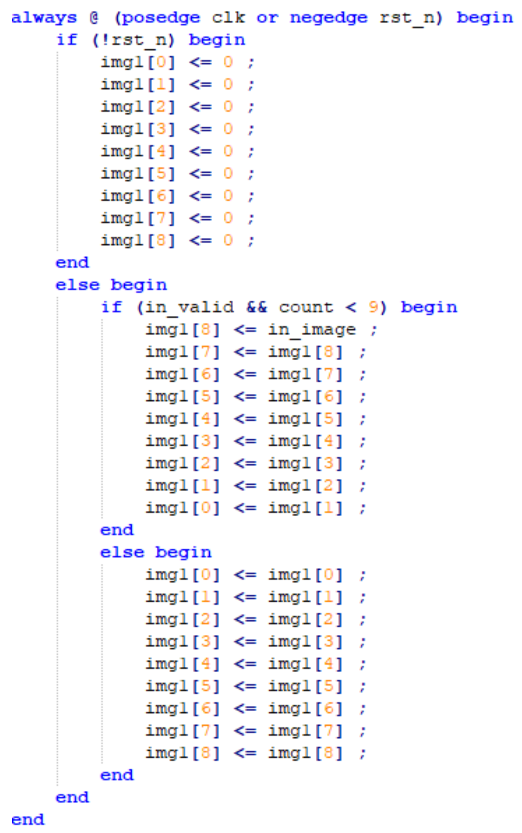


Optimization (Shift Register)

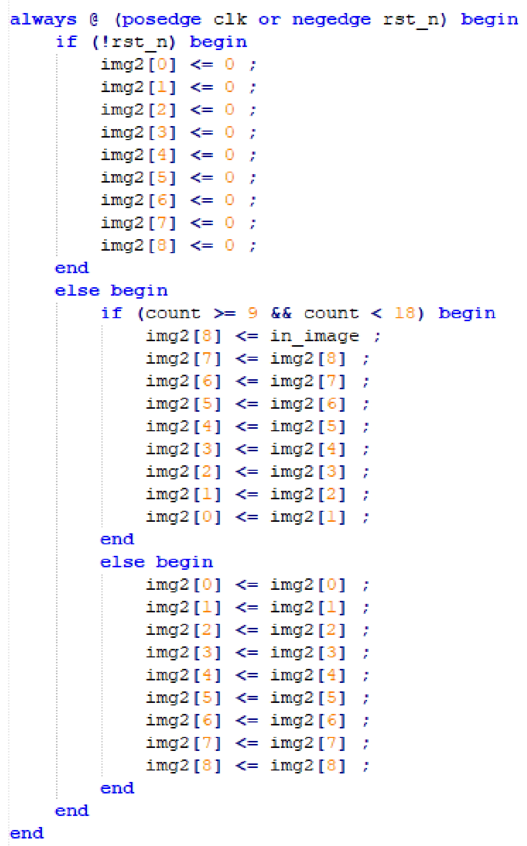


Count2 (plus a adder……..)

Reference Code

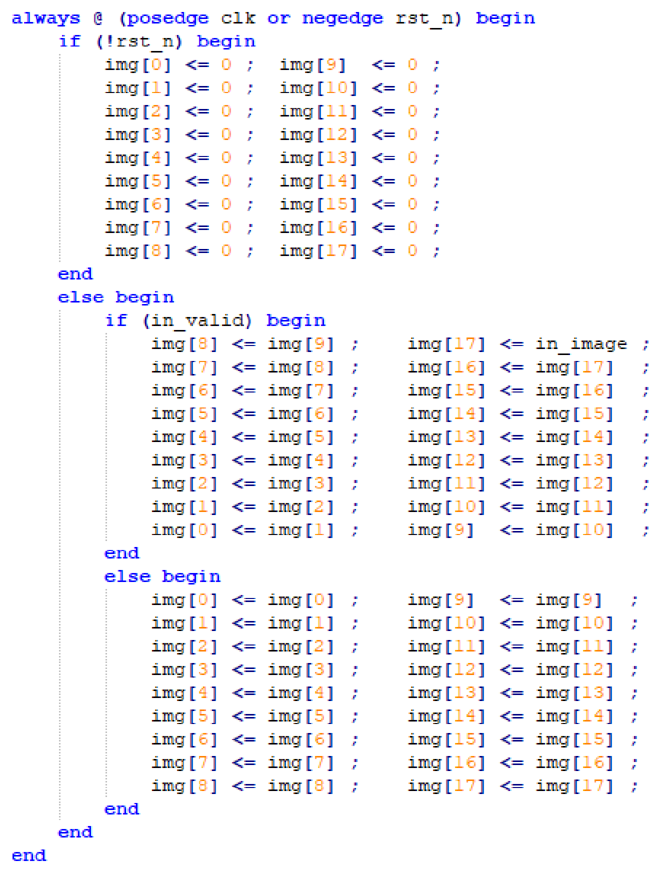


```
always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        img1[0] <= 0 ;
        img1[1] <= 0 ;
        img1[2] <= 0 ;
        img1[3] <= 0 ;
        img1[4] <= 0 ;
        img1[5] <= 0 ;
        img1[6] <= 0 ;
        img1[7] <= 0 ;
        img1[8] <= 0 ;
    end
    else begin
        if (in_valid && count < 9) begin
            img1[8] <= in_image ;
            img1[7] <= img1[8]  ;
            img1[6] <= img1[7]  ;
            img1[5] <= img1[6]  ;
            img1[4] <= img1[5]  ;
            img1[3] <= img1[4]  ;
            img1[2] <= img1[3]  ;
            img1[1] <= img1[2]  ;
            img1[0] <= img1[1]  ;
        end
        else begin
            img1[0] <= img1[0]  ;
            img1[1] <= img1[1]  ;
            img1[2] <= img1[2]  ;
            img1[3] <= img1[3]  ;
            img1[4] <= img1[4]  ;
            img1[5] <= img1[5]  ;
            img1[6] <= img1[6]  ;
            img1[7] <= img1[7]  ;
            img1[8] <= img1[8]  ;
        end
    end
end
```



```
always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        img2[0] <= 0 ;
        img2[1] <= 0 ;
        img2[2] <= 0 ;
        img2[3] <= 0 ;
        img2[4] <= 0 ;
        img2[5] <= 0 ;
        img2[6] <= 0 ;
        img2[7] <= 0 ;
        img2[8] <= 0 ;
    end
    else begin
        if (count >= 9 && count < 18) begin
            img2[8] <= in_image ;
            img2[7] <= img2[8]  ;
            img2[6] <= img2[7]  ;
            img2[5] <= img2[6]  ;
            img2[4] <= img2[5]  ;
            img2[3] <= img2[4]  ;
            img2[2] <= img2[3]  ;
            img2[1] <= img2[2]  ;
            img2[0] <= img2[1]  ;
        end
        else begin
            img2[0] <= img2[0]  ;
            img2[1] <= img2[1]  ;
            img2[2] <= img2[2]  ;
            img2[3] <= img2[3]  ;
            img2[4] <= img2[4]  ;
            img2[5] <= img2[5]  ;
            img2[6] <= img2[6]  ;
            img2[7] <= img2[7]  ;
            img2[8] <= img2[8]  ;
        end
    end
end
```

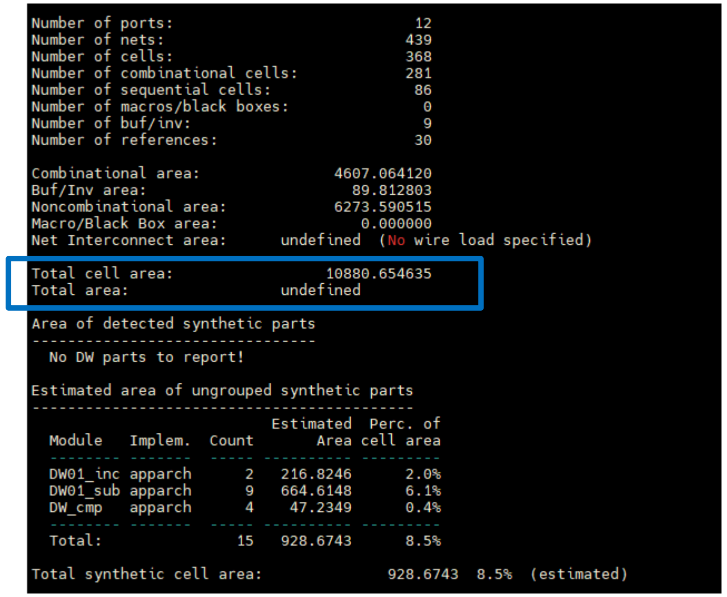
Without Shift Register



```
always @ (posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        img[0]  <= 0 ;  img[9]   <= 0 ;
        img[1]  <= 0 ;  img[10]  <= 0 ;
        img[2]  <= 0 ;  img[11]  <= 0 ;
        img[3]  <= 0 ;  img[12]  <= 0 ;
        img[4]  <= 0 ;  img[13]  <= 0 ;
        img[5]  <= 0 ;  img[14]  <= 0 ;
        img[6]  <= 0 ;  img[15]  <= 0 ;
        img[7]  <= 0 ;  img[16]  <= 0 ;
        img[8]  <= 0 ;  img[17]  <= 0 ;
    end
    else begin
        if (in_valid) begin
            img[8] <= img[9] ;    img[17] <= in_image ;
            img[7] <= img[8] ;    img[16] <= img[17]  ;
            img[6] <= img[7] ;    img[15] <= img[16]  ;
            img[5] <= img[6] ;    img[14] <= img[15]  ;
            img[4] <= img[5] ;    img[13] <= img[14]  ;
            img[3] <= img[4] ;    img[12] <= img[13]  ;
            img[2] <= img[3] ;    img[11] <= img[12]  ;
            img[1] <= img[2] ;    img[10] <= img[11]  ;
            img[0] <= img[1] ;    img[9]  <= img[10]  ;
        end
        else begin
            img[0] <= img[0] ;    img[9]  <= img[9]  ;
            img[1] <= img[1] ;    img[10] <= img[10] ;
            img[2] <= img[2] ;    img[11] <= img[11] ;
            img[3] <= img[3] ;    img[12] <= img[12] ;
            img[4] <= img[4] ;    img[13] <= img[13] ;
            img[5] <= img[5] ;    img[14] <= img[14] ;
            img[6] <= img[6] ;    img[15] <= img[15] ;
            img[7] <= img[7] ;    img[16] <= img[16] ;
            img[8] <= img[8] ;    img[17] <= img[17] ;
        end
    end
end
```

With Shift Register

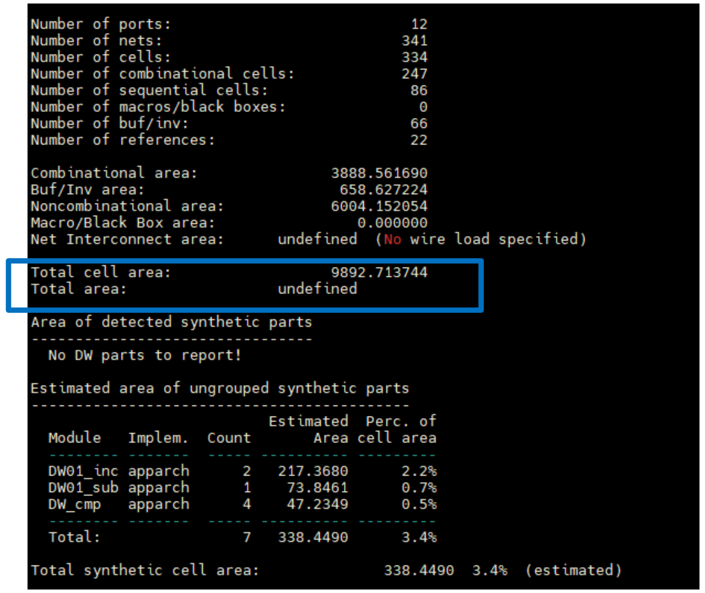
Result



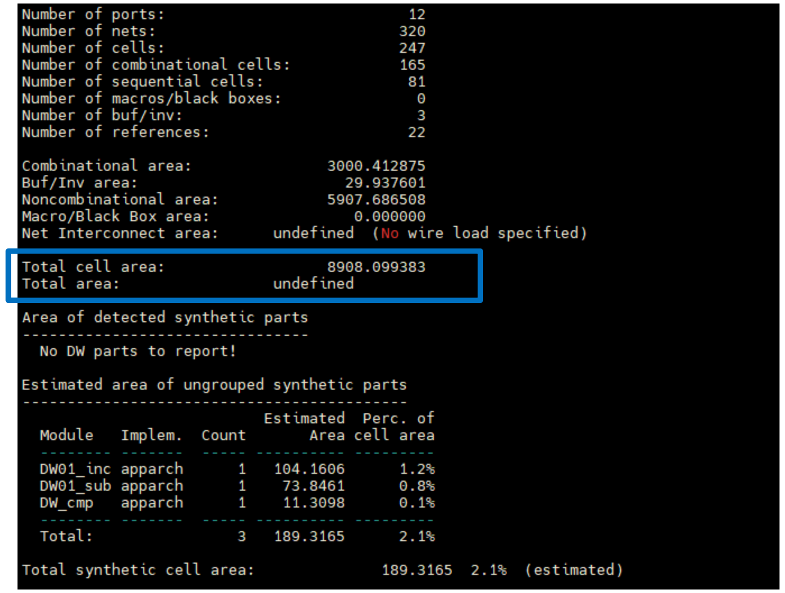
```
Number of ports:                           12
Number of nets:                           439
Number of cells:                          368
Number of combinational cells:            281
Number of sequential cells:                86
Number of macros/black boxes:               0
Number of buf/inv:                          9
Number of references:                      30

Combinational area:               4607.064120
Buf/Inv area:                       89.812803
Noncombinational area:            6273.590515
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                 10880.654635
Total area:                 undefined

Area of detected synthetic parts
--------------------------------
  No DW parts to report!

Estimated area of ungrouped synthetic parts
-------------------------------------------
                           Estimated  Perc. of
  Module   Implem.  Count       Area cell area
  -------- -------  -----  ---------  ---------
  DW01_inc apparch      2   216.8246       2.0%
  DW01_sub apparch      9   664.6148       6.1%
  DW_cmp   apparch      4    47.2349       0.4%
  -------- -------  -----  ---------  ---------
  Total:               15   928.6743       8.5%

Total synthetic cell area:          928.6743  8.5%  (estimated)
```

Original

Hardware Sharing



```
Number of ports:                           12
Number of nets:                           341
Number of cells:                          334
Number of combinational cells:            247
Number of sequential cells:                86
Number of macros/black boxes:               0
Number of buf/inv:                         66
Number of references:                      22

Combinational area:               3888.561690
Buf/Inv area:                      658.627224
Noncombinational area:            6004.152054
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                  9892.713744
Total area:                 undefined

Area of detected synthetic parts
-----------------------------------
  No DW parts to report!

Estimated area of ungrouped synthetic parts
-------------------------------------------
                           Estimated  Perc. of
  Module   Implem.  Count       Area cell area
  -------- -------  -----  ---------  ---------
  DW01_inc apparch      2   217.3680       2.2%
  DW01_sub apparch      1    73.8461       0.7%
  DW_cmp   apparch      4    47.2349       0.5%
  -------- -------  -----  ---------  ---------
  Total:                7   338.4490       3.4%

Total synthetic cell area:          338.4490  3.4%  (estimated)
```



```
Number of ports:                           12
Number of nets:                           320
Number of cells:                          247
Number of combinational cells:            165
Number of sequential cells:                81
Number of macros/black boxes:               0
Number of buf/inv:                          3
Number of references:                      22

Combinational area:               3000.412875
Buf/Inv area:                       29.937601
Noncombinational area:            5907.686508
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                  8908.099383
Total area:                 undefined

Area of detected synthetic parts
--------------------------------
  No DW parts to report!

Estimated area of ungrouped synthetic parts
-------------------------------------------
                           Estimated  Perc. of
  Module   Implem.  Count       Area cell area
  -------- -------  -----  ---------  ---------
  DW01_inc apparch      1   104.1606       1.2%
  DW01_sub apparch      1    73.8461       0.8%
  DW_cmp   apparch      1    11.3098       0.1%
  -------- -------  -----  ---------  ---------
  Total:                3   189.3165       2.1%

Total synthetic cell area:          189.3165  2.1%  (estimated)
```

Hardware Sharing

+

Shift Register

Conclusion (For Lab)

1. After reviewing spec, organize your block diagram first

2. According to your block diagram, try to optimize your design at system level

3. Write your RTL code with clear coding style

4. Plot your block diagram, try to find somewhere can be optimized

5. Rewrite your code, try to get better performance

Conclusion (For HW)

1. After reviewing spec, organize your block diagram first

2. According to your block diagram, try to optimize your design at system level

3. Write your RTL code with clear coding style

4. Plot your block diagram, try to find somewhere can be optimized

5. Rewrite your code, try to get better performance