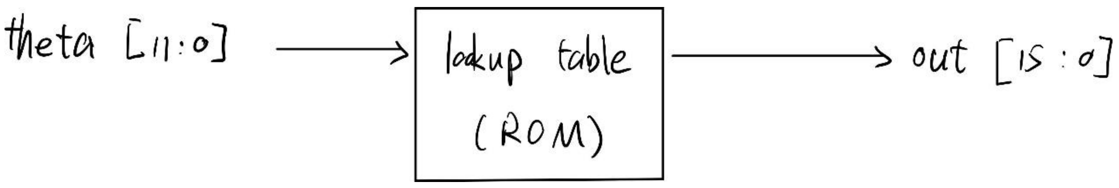
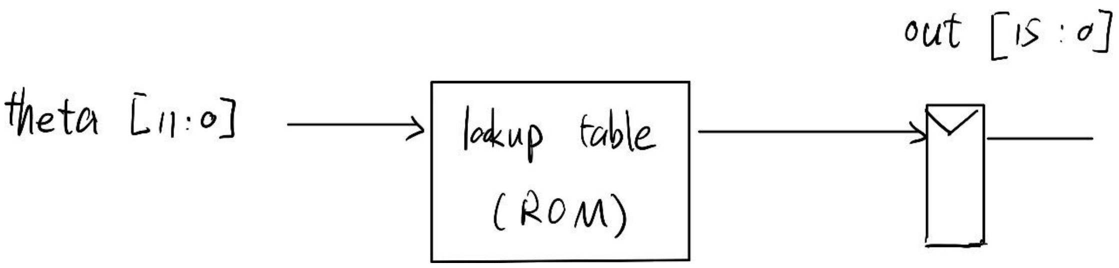
1. This problem requires the design of a block which calculates tan(θ) for a given θ, every clock cycle. Theta ranges from 0 to slightly less than π/4, or 45°. The latency may be as many cycles as needed.

I. Full Lookup Table

a) Detailed block diagram with all functional details.

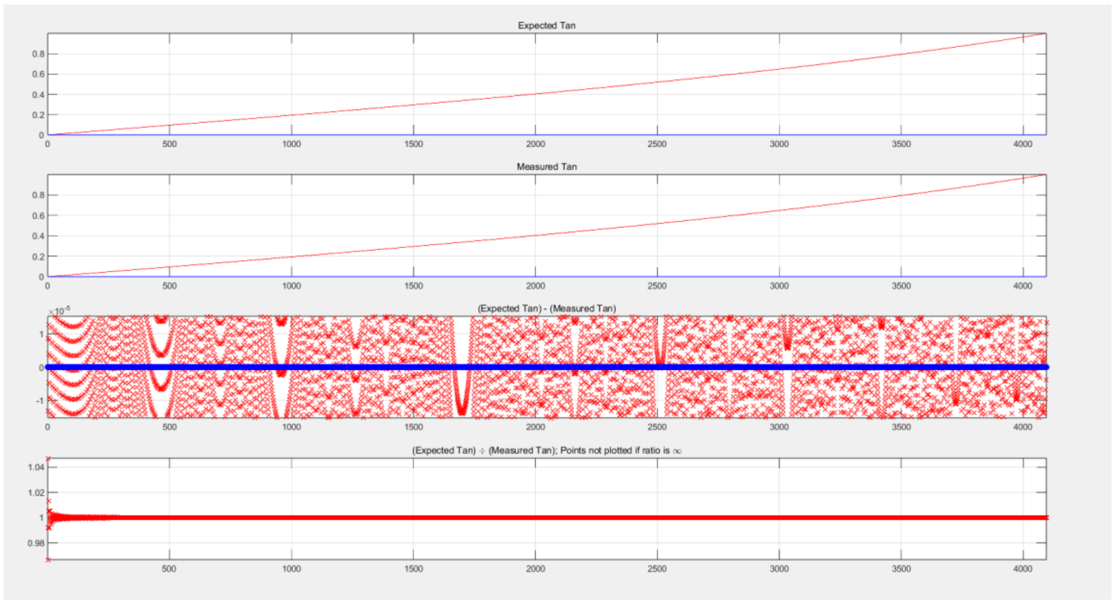


b) Pipelined block diagram.



c) (Accuracy points for smallest error compared to matlab, in comparison to other working designs in the class.)Write the Energy_diff/Energy_data0 value in dB in your report and also submit the four plots produced by difff.m.

Energy_data0/Energy_diff = 3481175748.458281 = 95.417259dB



d) Synthesize your design at the following three cycle time values and report the 1) *achieved* cycle time (and corresponding clock frequency) and 2) area for each:

1. a very long cycle time, e.g., 1 ms (1 KHz), to find the minimum area;

```
****************************************
Report : area
Design : prob1
Version: W-2024.09-SP1
Date   : Tue Mar 18 22:28:51 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db
)

Number of ports:                           29
Number of nets:                          9749
Number of cells:                         9736
Number of combinational cells:           9720
Number of sequential cells:                15
Number of macros/black boxes:               0
Number of buf/inv:                       1660
Number of references:                      25

Combinational area:             9299.626026
Buf/Inv area:                    897.484007
Noncombinational area:            67.829998
Macro/Black Box area:              0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                9367.456023
Total area:                 undefined
1
```

Area = 9367.456023

T_cycle_time = 1ms

2. a very short cycle time, e.g., 0.1 ns (10 GHz), to find the minimum cycle time;

```
****************************************
Report : timing
        -path full
        -delay max
        -nworst 10
        -max_paths 10
Design : prob1
Version: W-2024.09-SP1
Date   : Tue Mar 18 22:34:15 2025
****************************************


Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: theta[1] (input port clocked by clk)
  Endpoint: out_reg[11]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  prob1              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                   Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                   0.00       0.00
  clock network delay (ideal)             0.00       0.00
  input external delay                    0.00       0.00 r
  theta[1] (in)                           0.00       0.00 r
  U12465/ZN (INV_X1)                      0.03       0.04 f
  U11830/ZN (AND2_X2)                     0.05       0.09 f
  U10804/ZN (AND2_X1)                     0.05       0.15 f
  U9907/ZN (INV_X2)                       0.08       0.23 r
  U13077/ZN (NAND2_X1)                    0.06       0.29 f
  U12136/ZN (AND2_X1)                     0.05       0.34 f
  U10978/ZN (AOI221_X4)                   0.13       0.47 r
  U13078/ZN (NAND4_X1)                    0.05       0.52 f
  U13079/ZN (AOI221_X1)                   0.09       0.61 r
```

```
U13120/ZN (NAND4_X1)                   0.05      0.66 f
U11740/ZN (OR2_X1)                     0.07      0.73 f
U12063/ZN (NOR3_X1)                    0.06      0.79 r
U17020/ZN (NAND2_X1)                   0.03      0.82 f
out_reg[11]/D (DFF_X1)                 0.01      0.83 f
data arrival time                                0.83

clock clk (rise edge)                  0.10      0.10
clock network delay (ideal)            0.00      0.10
clock uncertainty                      0.00      0.09
out_reg[11]/CK (DFF_X1)                0.00      0.09 r
library setup time                    -0.04      0.05
data required time                               0.05
-----------------------------------------------------------
data required time                               0.05
data arrival time                               -0.83
-----------------------------------------------------------
slack (VIOLATED)                                -0.78
```

T_minimum_clock = T_data_arrival_time + T_setup = 0.83 + 0.05 = 0.88

T_cycle_time = 1ns

Area = 8010.856049

```
****************************************
Report : area
Design : prob1
Version: W-2024.09-SP1
Date   : Tue Mar 18 23:07:30 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

   NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db
)

Number of ports:                           29
Number of nets:                          7552
Number of cells:                         7539
Number of combinational cells:           7523
Number of sequential cells:                15
Number of macros/black boxes:               0
Number of buf/inv:                       1919
Number of references:                      52

Combinational area:             7943.026051
Buf/Inv area:                   1196.202007
Noncombinational area:            67.829998
Macro/Black Box area:              0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                8010.856049
Total area:                 undefined
1
```

3. a synthesis run with the cycle time set to the result from part (d)(2) multiplied times 1.5

```
****************************************
Report : area
Design : prob1
Version: W-2024.09-SP1
Date   : Tue Mar 18 21:09:37 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db
)

Number of ports:                           29
Number of nets:                          9864
Number of cells:                         9851
Number of combinational cells:           9835
Number of sequential cells:                15
Number of macros/black boxes:               0
Number of buf/inv:                       1678
Number of references:                      34

Combinational area:             9448.586021
Buf/Inv area:                    931.532006
Noncombinational area:            67.829998
Macro/Black Box area:              0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                9516.416018
Total area:                 undefined
1
```

T_cycle_time = 0.88 * 1.5 = 1.32

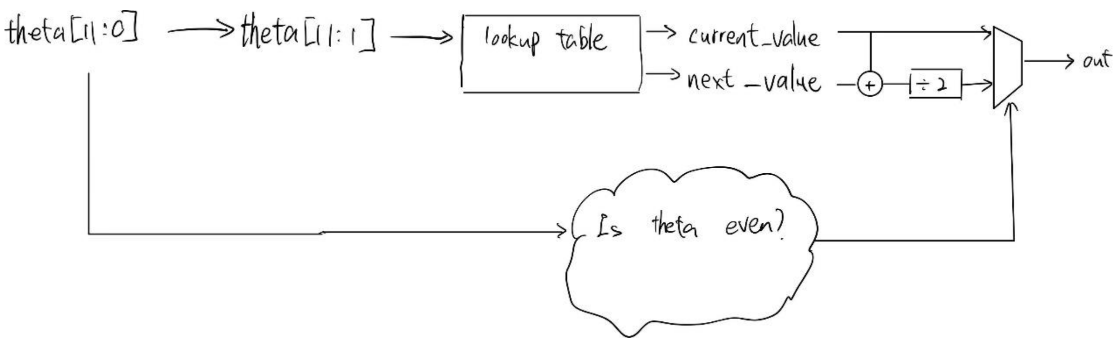
Area = 9516.416018

```
//prob1_1.v
`timescale 10ps/1ps

module prob1 (
    clk,
    theta,
    out
);
    input clk;
    input [11:0] theta;
    output reg [15:0] out;
    always @(posedge clk) begin
        case(theta)
            12'b000000000000: out <= #1 16'b0000000000000000;
            12'b000000000001: out <= #1 16'b0000000000000110;
            12'b000000000010: out <= #1 16'b0000000000001101;
            12'b000000000011: out <= #1 16'b0000000000010011;
            12'b000000000100: out <= #1 16'b0000000000011001;
            12'b000000000101: out <= #1 16'b0000000000011111;
            12'b000000000110: out <= #1 16'b0000000000100110;
            12'b000000000111: out <= #1 16'b0000000000101100;
            12'b000000001000: out <= #1 16'b0000000000110010;
            12'b000000001001: out <= #1 16'b0000000000111001;
            12'b000000001010: out <= #1 16'b0000000000111111;
            12'b000000001011: out <= #1 16'b0000000001000101;
            12'b000000001100: out <= #1 16'b0000000001001011;
-----------------------<Many lines removed>------------------------
            12'b111111101010: out <= #1 16'b0111111011101101;
            12'b111111101011: out <= #1 16'b0111111011111001;
            12'b111111101100: out <= #1 16'b0111111100000110;
            12'b111111101101: out <= #1 16'b0111111100010010;
            12'b111111101110: out <= #1 16'b0111111100011111;
            12'b111111101111: out <= #1 16'b0111111100101011;
            12'b111111110000: out <= #1 16'b0111111100111000;
            12'b111111110001: out <= #1 16'b0111111101000100;
            12'b111111110010: out <= #1 16'b0111111101010001;
            12'b111111110011: out <= #1 16'b0111111101011101;
            12'b111111110100: out <= #1 16'b0111111101101010;
            12'b111111110101: out <= #1 16'b0111111101110110;
            12'b111111110110: out <= #1 16'b0111111110000011;
            12'b111111110111: out <= #1 16'b0111111110001111;
            12'b111111111000: out <= #1 16'b0111111110011100;
            12'b111111111001: out <= #1 16'b0111111110101000;
            12'b111111111010: out <= #1 16'b0111111110110101;
            12'b111111111011: out <= #1 16'b0111111111000001;
            12'b111111111100: out <= #1 16'b0111111111001110;
            12'b111111111101: out <= #1 16'b0111111111011010;
            12'b111111111101: out <= #1 16'b0111111111011010;
            12'b111111111110: out <= #1 16'b0111111111100111;
            12'b111111111111: out <= #1 16'b0111111111110011;
            default: out <= #1 16'b1111111111111111;
        endcase
    end
endmodule
```

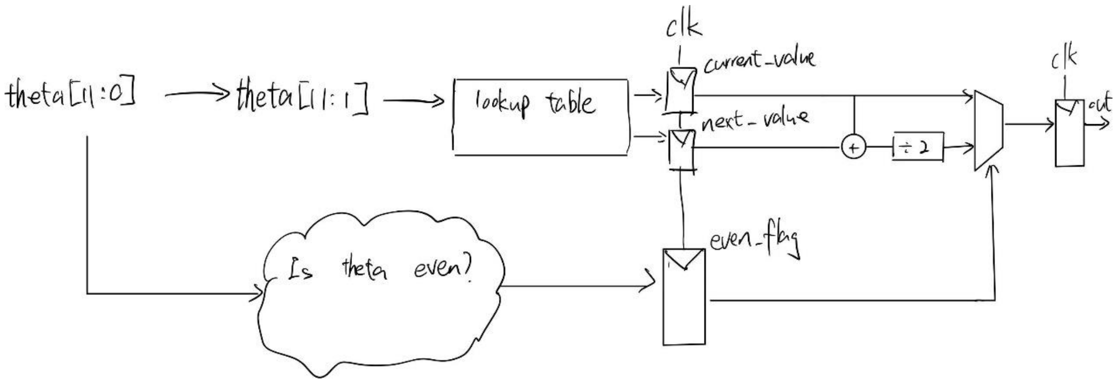
```
//prob1_1_tbench.v
`timescale 10ps/1ps
module prob1_tbench;
    reg clk;
    reg reset;
    reg [11:0] theta;
    wire [15:0] out;
    integer file;
    prob1 _prob1(.clk(clk), .theta(theta), .out(out));
    initial begin
        reset = 1'b1;
        #500;
        reset = 1'b0;
        @(posedge clk); #10

        file = $fopen("output_1.m", "w");
        for (theta = 12'b000000000000; theta < 12'b111111111111; theta = theta +
1) begin
            @(posedge clk); #10
            $fwrite(file,"%d %d\n", theta, out);
        end
        theta = 12'b111111111111;
        @(posedge clk); #10
        $fwrite(file,"%d %d\n", theta, out);
        $fclose(file);
        @(posedge clk); #10
        repeat (50) @ (posedge clk);
        $finish;
    end

    always begin
        if(reset == 1'b1)begin
            clk = 1'b0;
            #1;
        end
        else begin
            #100
            clk = ~clk;
        end
    end

endmodule
```

II. Interpolation

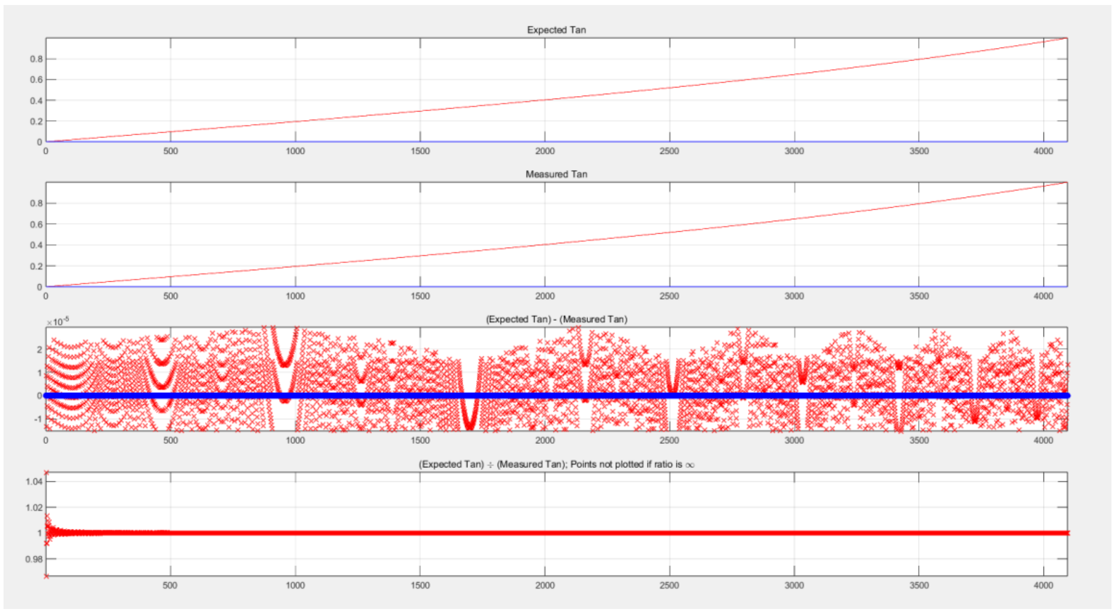
a) Detailed block diagram with all functional details.



b) Pipelined block diagram.



c) (Accuracy points for smallest error compared to matlab, in comparison to other working designs in the class.)Write the Energy_diff/Energy_data0 value in dB in your report and also submit the four plots produced by difff.m.



Energy_data0/Energy_diff = 93.533572dB

d) Synthesize your design at the following three cycle time values and report the 1) *achieved* cycle time (and corresponding clock frequency) and 2) area for each:

1. a very long cycle time, e.g., 1 ms (1 KHz), to find the minimum area;

```
****************************************
Report : area
Design : prob1_2
Version: W-2024.09-SP1
Date   : Tue Mar 18 22:50:25 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    NangateOpenCellLibrary (File: 
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db
)

Number of ports:                           29
Number of nets:                          6021
Number of cells:                         5998
Number of combinational cells:           5946
Number of sequential cells:                48
Number of macros/black boxes:               0
Number of buf/inv:                        902
Number of references:                      21

Combinational area:               5939.248020
Buf/Inv area:                      502.740002
Noncombinational area:             240.995995
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  6180.244015
Total area:                 undefined
1
```

Area = 6180.244015

T_cycle_time = 1ms

2. a very short cycle time, e.g., 0.1 ns (10 GHz), to find the minimum cycle time;

```
****************************************
Report : timing
        -path full
        -delay max
        -nworst 10
        -max_paths 10
Design : prob1_2
Version: W-2024.09-SP1
Date   : Tue Mar 18 22:55:04 2025
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: theta[1] (input port clocked by clk)
  Endpoint: rvalue_r_reg[14]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  prob1_2            5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  input external delay                     0.00       0.00 r
  theta[1] (in)                            0.00       0.00 r
  U8054/ZN (INV_X1)                        0.03       0.04 f
  U6273/ZN (AND3_X2)                       0.05       0.09 f
  U6954/ZN (INV_X2)                        0.05       0.14 r
  U7353/ZN (INV_X1)                        0.04       0.18 f
  U7288/ZN (AND2_X1)                       0.05       0.23 f
  U8070/ZN (AOI211_X1)                     0.09       0.32 r
  U9832/ZN (NAND2_X1)                      0.04       0.36 f
  U8438/ZN (NOR2_X1)                       0.04       0.40 r
  U9835/ZN (NAND3_X1)                      0.03       0.44 f
  U8405/ZN (NOR2_X1)                       0.04       0.48 r
  U9836/ZN (NAND4_X1)                      0.04       0.52 f
```

```
U9887/ZN (NOR4_X1)                    0.06      0.58 r
U7625/ZN (AND2_X1)                    0.06      0.64 r
U7611/ZN (INV_X1)                     0.03      0.68 f
U12807/ZN (NOR4_X1)                   0.08      0.76 r
U12808/ZN (NAND2_X1)                  0.03      0.79 f
rvalue_r_reg[14]/D (DFF_X1)           0.01      0.80 f
data arrival time                               0.80

clock clk (rise edge)                 0.10      0.10
clock network delay (ideal)           0.00      0.10
clock uncertainty                     0.00      0.09
rvalue_r_reg[14]/CK (DFF_X1)          0.00      0.09 r
library setup time                   -0.04      0.05
data required time                              0.05
-----------------------------------------------------------
data required time                              0.05
data arrival time                              -0.80
-----------------------------------------------------------
slack (VIOLATED)                               -0.75
```

T_minimum_clock = T_data_arrival_time + T_setup = 0.80 + 0.05 =0.85

T_cycle_time = 1ns

Area = 7312.074039

```
****************************************
Report : area
Design : prob1_2
Version: W-2024.09-SP1
Date   : Tue Mar 18 22:55:02 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

   NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db
)

Number of ports:                           29
Number of nets:                          7350
Number of cells:                         7341
Number of combinational cells:           7289
Number of sequential cells:                48
Number of macros/black boxes:               0
Number of buf/inv:                       2135
Number of references:                      46

Combinational area:             7071.078044
Buf/Inv area:                   1296.484008
Noncombinational area:           240.995995
Macro/Black Box area:              0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                7312.074039
Total area:                 undefined
1
```

3. a synthesis run with the cycle time set to the result from part (d)(2) multiplied times 1.5

```
****************************************
Report : area
Design : prob1_2
Version: W-2024.09-SP1
Date   : Tue Mar 18 23:00:53 2025
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db
)

Number of ports:                           29
Number of nets:                          6164
Number of cells:                         6151
Number of combinational cells:           6096
Number of sequential cells:                48
Number of macros/black boxes:               0
Number of buf/inv:                        969
Number of references:                      26

Combinational area:               6052.830017
Buf/Inv area:                      556.472001
Noncombinational area:             241.527995
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  6294.358012
Total area:                 undefined
1
```

T_cycle_time = 0.85 * 1.5 = 1.275

Area = 6294.358012

```
\\prob1_2.v
`timescale 10ps/1ps
module prob1_2 (
    clk,
    theta,
    out
);
    input clk;
    input [11:0] theta;
    output reg [15:0] out;
    reg [16:0] out_r;
    reg [16:0] lvalue_r, rvalue_r;
    reg even;
    always @(posedge clk) begin
        case(theta >> 1)
            11'b00000000000: begin lvalue_r <= 16'b0000000000000000; rvalue_r <=
16'b0000000000001101;end
            11'b00000000001: begin lvalue_r <= 16'b0000000000001101; rvalue_r <=
16'b0000000000011001;end
            11'b00000000010: begin lvalue_r <= 16'b0000000000011001; rvalue_r <=
16'b0000000000100110;end
            11'b00000000011: begin lvalue_r <= 16'b0000000000100110; rvalue_r <=
16'b0000000000110010;end
            11'b00000000100: begin lvalue_r <= 16'b0000000000110010; rvalue_r <=
16'b0000000000111111;end
            11'b00000000101: begin lvalue_r <= 16'b0000000000111111; rvalue_r <=
16'b0000000001001011;end
            11'b00000000110: begin lvalue_r <= 16'b0000000001001011; rvalue_r <=
16'b0000000001011000;end
            11'b00000000111: begin lvalue_r <= 16'b0000000001011000; rvalue_r <=
16'b0000000001100101;end
            11'b00000001000: begin lvalue_r <= 16'b0000000001100101; rvalue_r <=
16'b0000000001110001;end
            11'b00000001001: begin lvalue_r <= 16'b0000000001110001; rvalue_r <=
16'b0000000001111110;end
------------------------<Many lines removed>------------------------
11'b11111110100: begin lvalue_r <= 16'b0111111011010100; rvalue_r <=
16'b0111111011101101;end
            11'b11111110101: begin lvalue_r <= 16'b0111111011101101; rvalue_r <=
16'b0111111100000110;end
            11'b11111110110: begin lvalue_r <= 16'b0111111100000110; rvalue_r <=
16'b0111111100011111;end
            11'b11111110111: begin lvalue_r <= 16'b0111111100011111; rvalue_r <=
16'b0111111100111000;end
            11'b11111111000: begin lvalue_r <= 16'b0111111100111000; rvalue_r <=
16'b0111111101010001;end
```

```
            11'b11111111001: begin lvalue_r <= 16'b0111111101010001; rvalue_r <=
16'b0111111101101010;end
            11'b11111111010: begin lvalue_r <= 16'b0111111101101010; rvalue_r <=
16'b0111111110000011;end
            11'b11111111011: begin lvalue_r <= 16'b0111111110000011; rvalue_r <=
16'b0111111110011100;end
            11'b11111111100: begin lvalue_r <= 16'b0111111110011100; rvalue_r <=
16'b0111111110110101;end
            11'b11111111101: begin lvalue_r <= 16'b0111111110110101; rvalue_r <=
16'b0111111111001110;end
            11'b11111111110: begin lvalue_r <= 16'b0111111111001110; rvalue_r <=
16'b0111111111100111;end
            11'b11111111111: begin lvalue_r <= 16'b0111111111100111; rvalue_r <=
16'b1000000000000000;end
            default: begin lvalue_r <= 16'b1111111111111111; rvalue_r <=
16'b1111111111111111;end
        endcase
        if(theta[0] == 1'b0)
            even <= 1'b1;
        else
            even <= 1'b0;
    end
    always @(posedge clk) begin
        if(even == 1'b1)
            out <= #1 lvalue_r;
        else
            out <= #1 ({1'b0,lvalue_r} + {1'b0,rvalue_r}) >> 1;
    end
endmodule
```

```
\\prob1_2_tbench.v
`timescale 10ps/1ps
module prob1_2_tbench;
    reg clk;
    reg reset;
    reg [11:0] theta, theta_old;
    wire [15:0] out;
    integer file;
    prob1_2 _prob1_2(.clk(clk), .theta(theta), .out(out));
    initial begin
        reset = 1'b1;
        #500;
        reset = 1'b0;
        @(posedge clk); #10

        file = $fopen("output_1_2.m", "w");
        for (theta = 12'b000000000000; theta < 12'b111111111111; theta = theta +
1) begin
            @(posedge clk); #10
            if(theta != 12'b000000000000)
                $fwrite(file,"%d %d\n", theta_old, out);
            theta_old = theta;
        end
        theta = 12'b111111111111;
        @(posedge clk); #10
        $fwrite(file,"%d %d\n", theta_old, out);
        @(posedge clk); #10
        $fwrite(file,"%d %d\n", theta, out);
        $fclose(file);
        @(posedge clk); #10
        repeat (50) @ (posedge clk);
        $finish;
    end

    always begin
        if(reset == 1'b1)begin
            clk = 1'b0;
            #1;
        end
        else begin
            #100
            clk = ~clk;
        end
    end

endmodule
```

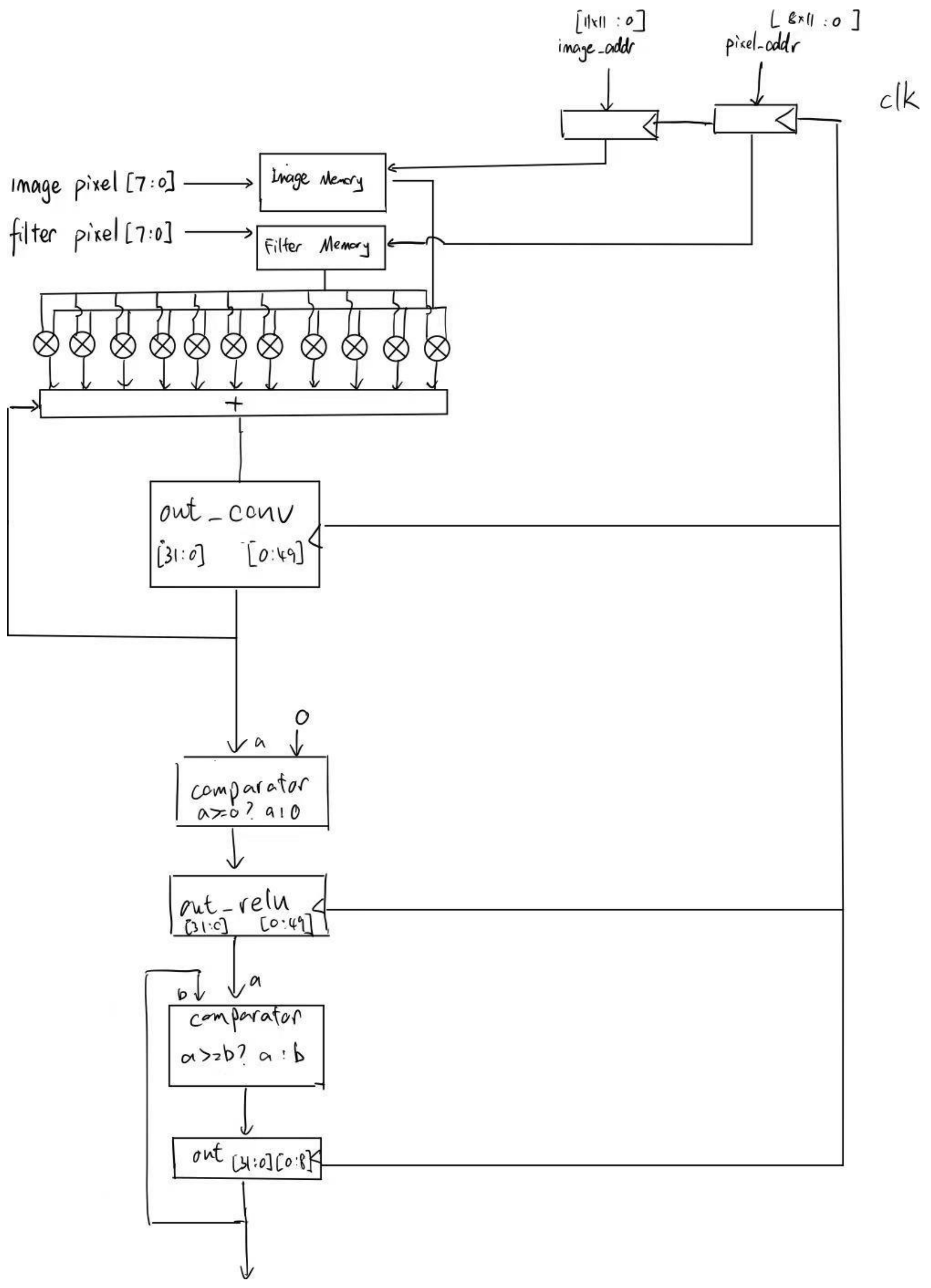
2. The Alexnet convolutional neural net is widely credited with the dramatic rise in popularity of neural nets. Read the 2012 Alexnet paper paying particular attention to Sections 1, 2, 3.4, and 3.5. This project consists of building and synthesizing custom hardware for the first convolutional layer of Alexnet using a reduced image size.

a) [25 pts] A bulleted list and a few sentences describing your design including such features as: the number and size of memories, number of multipliers and adders, exact number of cycles to complete all convolutions, number of pipeline stages, in what order are the pieces and in what order are the convolutions calculated, etc.

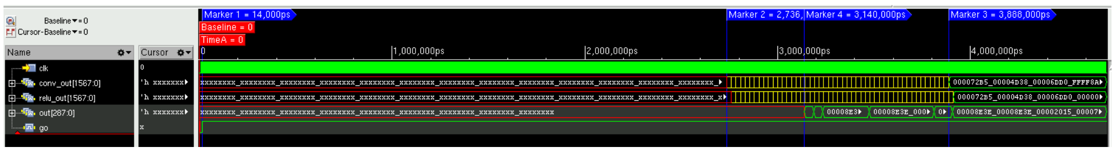
- 1 Image Memory (35 x 35 x 8bits)
- 1 Filter Memory (11 x 11 x 8 bits)
- 1Convolution Result Momory (7 x 7 x 32 bits)
- 1 ReLU Result Memory (7 x 7 x 32 bits)
- 1 Final Result Memory (3 x 3 x 32 bits)
- 11Multipliers
- 11 Adders
- 588 cycles (2 stages, 49 x 12)
- 4 pipeline stages

This design is structured to efficiently perform convolutions using a straightforward pipeline. It utilizes memories for storing input data, filters, and outputs, while parallel multipliers are used to handle element-wise multiplications. The adders accumulate results from each multiplication to compute the final convolution result. By using a pipeline, each stage of the convolution operation is separated, allowing the system to work on multiple pixels simultaneously. The convolution process deals with the data from left to right and from up to down. For each convolution, each time the processor extracts a row of data from image and filter, then accumulates to a register. The next cycle, the processor will extract next row of data and so on.

b) [25 pts] Detailed pipelined block diagram with all functional details.



c) [25 pts] Detailed timing diagram showing the timing of the various processing units



1 clock cycle = 2000 ps. As you can see in the diagram, marker 1 = 14,000 ps(go signal), marker 2 = 2,736,000 ps (first convolution result), marker 3 = 3,888,000 ps (convolution ends). All convolution done in (3888 - 2736) / 2 /48 * 49 ≈ 588 cycles. Marker 4 = 3,140,000(first max pooling as soon as there is enough blocks to do max pooling, first one is 17 blocks, (3140 - 2736) / 2 / 11 = 18.3). Also you can see first relu done immediately after first convolution done.

d) e) [75 pts] Results for all convolutions printed by your verilog testbench. [50 pts] Matlab copy and pasted output showing whether your design matches the matlab model or not. Your hardware must exactly match the output of the alex35.m matlab model.

LoadData==0

```
[will02@coe-ece-2107-32 hw4]$ make run
ncverilog +access+r -l prob2.logv -f prob2.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob2_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
go start

     52631      86274      77123      56607      54545       -234      81273
    -10720      52254      55268      79588     142664     -33182      39128
    129844     111684      63921     195554     104722      95236      23934
    170679     100154     -52600      45082     110094      28123      12692
    106610     -10176      97563      77673      73936     -74634        292
     49520      25797      32954     152831      83581      30723      28378
     36475      -7406      48892      94242      11291      45869      94891

     52631      86274      77123      56607      54545          0      81273
         0      52254      55268      79588     142664          0      39128
    129844     111684      63921     195554     104722      95236      23934
    170679     100154          0      45082     110094      28123      12692
    106610          0      97563      77673      73936          0        292
     49520      25797      32954     152831      83581      30723      28378
     36475          0      48892      94242      11291      45869      94891

    129844     195554     142664
    170679     195554     110094
    106610     152831      94891
Simulation complete via $finish(1) at time 4708 NS + 0
./prob2_tbench.v:174         $finish;
ncsim> exit
```

```
out =


       52631       86274       77123       56607       54545        -234       81273
      -10720       52254       55268       79588      142664      -33182       39128
      129844      111684       63921      195554      104722       95236       23934
      170679      100154      -52600       45082      110094       28123       12692
      106610      -10176       97563       77673       73936      -74634         292
       49520       25797       32954      152831       83581       30723       28378
       36475       -7406       48892       94242       11291       45869       94891



out_relu =


       52631       86274       77123       56607       54545           0       81273
           0       52254       55268       79588      142664           0       39128
      129844      111684       63921      195554      104722       95236       23934
      170679      100154           0       45082      110094       28123       12692
      106610           0       97563       77673       73936           0         292
       49520       25797       32954      152831       83581       30723       28378
       36475           0       48892       94242       11291       45869       94891



out_max_pool =


      129844      195554      142664
      170679      195554      110094
      106610      152831       94891
```

LoadData== 1

```
[will02@coe-ece-2107-32 hw4]$ make run
ncverilog +access+r -l prob2.logv -f prob2.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob2_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
go start

data_type == 1 start

      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420

      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420
      2420       2420       2420       2420       2420       2420       2420

     2420      2420      2420
     2420      2420      2420
     2420      2420      2420
Simulation complete via $finish(1) at time 4708 NS + 0
./prob2_tbench.v:174        $finish;
ncsim> exit
```

```
out =

        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420


out_relu =

        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420
        2420        2420        2420        2420        2420        2420        2420


out_max_pool =

        2420        2420        2420
        2420        2420        2420
        2420        2420        2420
```

LoadData==2

Results are corrupted because there are negative numbers in Image data in this MATLAB- generated data. However, the assignment says each pixel is an unsigned integer. So, there is nothing wrong with the program.

- input image size of 35 pixels x 35 pixels x 1 grayscale "color". Each pixel is an 8-bit unsigned integer.

```
[will02@coe-ece-2107-32 hw4]$ make run
ncverilog +access+r -l prob2.logv -f prob2.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob2_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
go start

data_type == 2 start

         4        924       -200       -268         38       -505        762
        15       -636        773        603         19         66         18
        86         73         48        -11         85         -6         65
       157         68        -13        135        156         40         42
        12         57        -38         99          5         56        -69
       -29         -3         58        -67         -9         22        -49
        86        114          6         66        -36        -42        -26

         4        924          0          0         38          0        762
        15          0        773        603         19         66         18
        86         73         48          0         85          0         65
       157         68          0        135        156         40         42
        12         57          0         99          5         56          0
         0          0         58          0          0         22          0
        86        114          6         66          0          0          0

      924       773       762
      157       156       156
      114        99        56
Simulation complete via $finish(1) at time 4708 NS + 0
./prob2_tbench.v:174        $finish;
ncsim> exit
```

```
out =


    4   156    56   -12    38     7    -6
   15   132     5    91    19    66    18
   86    73    48   -11    85    -6    65
  157    68   -13   135   156    40    42
   12    57   -38    99     5    56   -69
  -29    -3    58   -67    -9    22   -49
   86   114     6    66   -36   -42   -26



out_relu =


    4   156    56     0    38     7     0
   15   132     5    91    19    66    18
   86    73    48     0    85     0    65
  157    68     0   135   156    40    42
   12    57     0    99     5    56     0
    0     0    58     0     0    22     0
   86   114     6    66     0     0     0



out_max_pool =


  156    91    85
  157   156   156
  114    99    56
```

LoadData==3

```
[will02@coe-ece-2107-32 hw4]$ make run
ncverilog +access+r -l prob2.logv -f prob2.vfv
ncverilog(64): 15.20-s031: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.prob2_tbench:v .................... Done
ncsim> source /software/Cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> run
go start

data_type == 3 start

      29365      19768      28112     -29959    -144388     -41970     -30082
    -125837     -35993      36414    -138051       8213     -85921     -82326
     -74384     -16452     -43546     -76804    -123662     -23286     -74481
    -101378      32321    -209081     -99172     -17720    -132176     -22096
    -168578     -43504    -138967     -25643     -44930     -28933    -119788
      62215      40351     -48912    -151844      -9949     -83100       8952
     -99188      40946     -37872    -116204     -84058     -22067      -7495

      29365      19768      28112          0          0          0          0
          0          0      36414          0       8213          0          0
          0          0          0          0          0          0          0
          0      32321          0          0          0          0          0
          0          0          0          0          0          0          0
      62215      40351          0          0          0          0       8952
          0      40946          0          0          0          0          0

     36414     36414      8213
     32321         0         0
     62215         0      8952
Simulation complete via $finish(1) at time 4708 NS + 0
./prob2_tbench.v:174        $finish;
ncsim> exit
```

```
out =


       29365       19768       28112      -29959     -144388      -41970      -30082
     -125837      -35993       36414     -138051        8213      -85921      -82326
      -74384      -16452      -43546      -76804     -123662      -23286      -74481
     -101378       32321     -209081      -99172      -17720     -132176      -22096
     -168578      -43504     -138967      -25643      -44930      -28933     -119788
       62215       40351      -48912     -151844       -9949      -83100        8952
      -99188       40946      -37872     -116204      -84058      -22067       -7495



out_relu =


       29365       19768       28112           0           0           0           0
           0           0       36414           0        8213           0           0
           0           0           0           0           0           0           0
           0       32321           0           0           0           0           0
           0           0           0           0           0           0           0
       62215       40351           0           0           0           0        8952
           0       40946           0           0           0           0           0



out_max_pool =


       36414       36414        8213
       32321           0           0
       62215           0        8952
```

f) [75 pts] Synthesize your design at the following three cycle time values and report the 1) *achieved* cycle time (and corresponding clock frequency) and 2) area for each:

1. a very long cycle time, e.g., 1 ms (1 KHz), to find the minimum area;

T_cycle = 1ms

Area = `54562.717175`

```
Warning: Design 'prob2' has '1' unresolved references. For more detailed
information, use the "link" command. (UID-341)

****************************************
Report : area
Design : prob2
Version: W-2024.09-SP1
Date   : Thu Mar 20 18:03:58 2025
****************************************

Information: Updating design information... (UID-85)
Warning: Design 'prob2' contains 1 high-fanout nets. A fanout number of 1000
will be used for delay calculations involving these nets. (TIM-134)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db
)

Number of ports:                       3442
Number of nets:                       26696
Number of cells:                      21917
Number of combinational cells:        14624
Number of sequential cells:            7276
Number of macros/black boxes:             0
Number of buf/inv:                     1486
Number of references:                    36

Combinational area:             21673.414367
Buf/Inv area:                     958.397991
Noncombinational area:          32889.302807
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                54562.717175
Total area:                 undefined

Information: This design contains black box (unknown) components. (RPT-8)
1
```

2. a very short cycle time, e.g., 0.1 ns (10 GHz), to find the minimum cycle time;

T_cycle = 0.1ns

T_minimum_cycle = 1.55ns + 0.06ns = 1.61ns

Area = 59950.280826

```
Warning: Design 'prob2' has '1' unresolved references. For more detailed
information, use the "link" command. (UID-341)

****************************************
Report : timing
        -path full
        -delay max
        -nworst 10
        -max_paths 10
Design : prob2
Version: W-2024.09-SP1
Date   : Thu Mar 20 18:09:09 2025
****************************************

 # A fanout number of 1000 was used for high fanout net computations.

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: conv_cnt_reg[2]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: out_conv_reg[22><31]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  prob2              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  conv_cnt_reg[2]/CK (DFF_X1)              0.00 #     0.00 r
  conv_cnt_reg[2]/Q (DFF_X1)               0.09       0.09 r
  U15604/ZN (INV_X1)                       0.03       0.12 f
  U15508/ZN (AND3_X2)                      0.05       0.18 f
  U20891/ZN (NAND2_X1)                     0.05       0.23 r
  U15155/Z (CLKBUF_X3)                     0.07       0.29 r
  U22884/ZN (OAI222_X1)                    0.06       0.35 f
  U22887/ZN (NOR4_X1)                      0.10       0.45 r
  U22899/ZN (NAND4_X1)                     0.07       0.52 f
  U667/ZN (NOR2_X1)                        0.06       0.59 r
  U696/ZN (OAI21_X1)                       0.04       0.62 f
  U669/ZN (AOI21_X1)                       0.07       0.69 r
```

```
U3801/ZN (OAI21_X1)                      0.04      0.74 f
U3762/ZN (AOI21_X1)                      0.06      0.79 r
U3852/Z (BUF_X1)                         0.06      0.85 r
U3882/ZN (OAI21_X1)                      0.04      0.89 f
U3872/ZN (XNOR2_X1)                      0.08      0.97 r
U360/ZN (NOR2_X1)                        0.03      1.01 f
U3551/ZN (NOR2_X1)                       0.05      1.06 r
U3541/ZN (NAND2_X1)                      0.04      1.10 f
U3571/ZN (NOR2_X1)                       0.06      1.16 r
U3621/ZN (AND2_X1)                       0.05      1.21 r
U3641/ZN (NOR2_X1)                       0.03      1.25 f
U3691/ZN (OAI21_X1)                      0.05      1.30 r
U3681/ZN (XNOR2_X1)                      0.07      1.37 r
U15529/ZN (AND2_X1)                      0.06      1.43 r
U15393/Z (BUF_X1)                        0.06      1.48 r
U20838/Z (MUX2_X1)                       0.05      1.54 r
out_conv_reg[22><31]/D (DFF_X1)           0.01      1.55 r
data arrival time                                  1.55

clock clk (rise edge)                    0.10      0.10
clock network delay (ideal)               0.00      0.10
clock uncertainty                        0.00      0.09
out_conv_reg[22><31]/CK (DFF_X1)          0.00      0.09 r
library setup time                      -0.03      0.06
data required time                                 0.06
-----------------------------------------------------------
data required time                                 0.06
data arrival time                                 -1.55
-----------------------------------------------------------
slack (VIOLATED)                                  -1.49
```

```
Warning: Design 'prob2' has '1' unresolved references. For more detailed
information, use the "link" command. (UID-341)

****************************************
Report : area
Design : prob2
Version: W-2024.09-SP1
Date   : Thu Mar 20 18:08:59 2025
****************************************

Information: Updating design information... (UID-85)
Warning: Design 'prob2' contains 1 high-fanout nets. A fanout number of 1000
will be used for delay calculations involving these nets. (TIM-134)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db
)

Number of ports:                         3442
Number of nets:                         30096
Number of cells:                        24802
Number of combinational cells:          17500
Number of sequential cells:              7276
Number of macros/black boxes:               0
Number of buf/inv:                       3674
Number of references:                      50

Combinational area:             27060.446019
Buf/Inv area:                    2479.386000
Noncombinational area:          32889.834807
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                59950.280826
Total area:                 undefined

Information: This design contains black box (unknown) components. (RPT-8)
1
```

3. a synthesis run with the cycle time set to the result from part (f)(2) multiplied times 1.5

T_cycle = T_minimum_cycle * 1.5 = 2.415ns

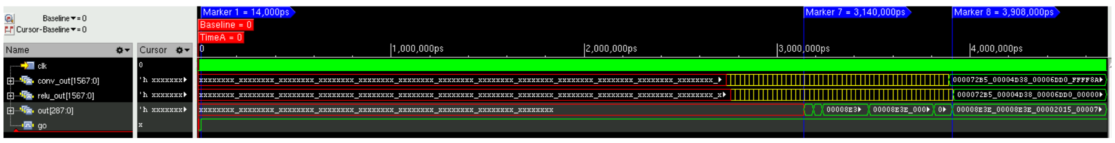
Area = 58259.318910

```
Warning: Design 'prob2' has '1' unresolved references. For more detailed
information, use the "link" command. (UID-341)

****************************************
Report : timing
        -path full
        -delay max
        -nworst 10
        -max_paths 10
Design : prob2
Version: W-2024.09-SP1
Date   : Thu Mar 20 18:30:49 2025
****************************************

 # A fanout number of 1000 was used for high fanout net computations.

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: pos_cnt_reg[0]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: out_max_pool_reg[0]<5]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  prob2              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                       Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                       0.00       0.00
  clock network delay (ideal)                 0.00       0.00
  pos_cnt_reg[0]/CK (DFF_X1)                  0.00 #     0.00 r
  pos_cnt_reg[0]/Q (DFF_X1)                   0.13       0.13 r
  U14767/ZN (NAND3_X1)                        0.06       0.19 f
  U20922/ZN (OAI21_X1)                        0.04       0.23 r
  U20916/ZN (INV_X1)                          0.03       0.26 f
  U20968/ZN (OAI21_X1)                        0.04       0.30 r
  U71/ZN (INV_X1)                             0.03       0.34 f
  U2_11/CO (FA_X1)                            0.11       0.44 f
  U2_21/CO (FA_X1)                            0.09       0.54 f
  U2_31/CO (FA_X1)                            0.09       0.63 f
  U2_41/CO (FA_X1)                            0.09       0.72 f
  U12/ZN (XNOR2_X1)                           0.09       0.81 r
  U15866/ZN (AND2_X1)                         0.07       0.89 r
  U20492/ZN (NAND2_X1)                        0.12       1.01 f
```

```
U20491/Z (BUF_X1)                      0.13       1.13 f
U20469/ZN (OAI221_X1)                   0.08       1.22 r
U20528/ZN (NOR4_X1)                    0.03       1.25 f
U20952/ZN (NAND4_X1)                   0.05       1.29 r
U703/ZN (INV_X1)                       0.03       1.32 f
U757/ZN (AOI21_X1)                     0.04       1.36 r
U756/ZN (AOI22_X1)                     0.04       1.40 f
U704/ZN (INV_X1)                       0.03       1.43 r
U754/ZN (OAI221_X1)                    0.04       1.47 f
U753/ZN (AOI211_X1)                    0.08       1.55 r
U685/ZN (OR3_X1)                       0.04       1.59 r
U746/ZN (OAI211_X1)                    0.04       1.63 f
U706/ZN (OAI21_X1)                     0.08       1.72 r
U20591/Z (BUF_X2)                      0.12       1.84 r
U20603/ZN (OAI221_X1)                   0.08       1.93 f
U20604/ZN (NOR4_X1)                    0.10       2.03 r
U20169/ZN (NAND4_X1)                   0.05       2.08 f
U20168/ZN (OAI21_X1)                   0.04       2.12 r
U20163/Z (BUF_X1)                      0.06       2.19 r
U20843/ZN (OAI22_X1)                   0.05       2.23 f
out_max_pool_reg[0><5]/D (DFF_X1)        0.01       2.24 f
data arrival time                                 2.24

clock clk (rise edge)                   2.41       2.41
clock network delay (ideal)              0.00       2.41
clock uncertainty                     -0.12       2.29
out_max_pool_reg[0><5]/CK (DFF_X1)       0.00       2.29 r
library setup time                    -0.05       2.24
data required time                                2.24
-----------------------------------------------------------
data required time                                2.24
data arrival time                                -2.24
-----------------------------------------------------------
slack (MET)                                      0.00
```

```
Warning: Design 'prob2' has '1' unresolved references. For more detailed
information, use the "link" command. (UID-341)

****************************************
Report : area
Design : prob2
Version: W-2024.09-SP1
Date   : Thu Mar 20 18:30:38 2025
****************************************

Information: Updating design information... (UID-85)
Warning: Design 'prob2' contains 1 high-fanout nets. A fanout number of 1000
will be used for delay calculations involving these nets. (TIM-134)
Library(s) Used:

    NangateOpenCellLibrary (File:
/software/Synopsys/DesignCompiler/EEC281/lib/nangate45/NangateOpenCellLibrary.db
)

Number of ports:                         3442
Number of nets:                         31531
Number of cells:                        24061
Number of combinational cells:          16768
Number of sequential cells:              7276
Number of macros/black boxes:               0
Number of buf/inv:                       3233
Number of references:                      39

Combinational area:             25370.016102
Buf/Inv area:                    1874.502002
Noncombinational area:          32889.302807
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                58259.318910
Total area:                 undefined

Information: This design contains black box (unknown) components. (RPT-8)
1
```

g) [10 pts] Report the number of clock cycles between the *go* signal and 1) the first output, and 2) the final output.



T(first output - go) = (3140 – 14) / 2 = 1563

T(final output - go) = 1947

```
`timescale 10ps/1ps
module prob2 (
    clk,
    image_pixel,
    filter_pixel,
    go,
    conv_out,
    relu_out,
    out
);
    input clk, go;
    input [7:0] image_pixel;
    input [7:0] filter_pixel;
    output reg [32 * 9 - 1:0] out;
    output reg [32 * 49 - 1:0] conv_out;
    output reg [32 * 49 - 1:0] relu_out;
    reg [8 * 11 - 1:0] image_din;
    reg [8 * 11 - 1:0] filter_din;
    wire [8 * 11 - 1:0] image_dout;
    wire [8 * 11 - 1:0] filter_dout;
    reg [11 * 11 - 1:0] image_addr;
    reg [7 * 11 - 1:0] filter_addr;

    reg image_we, filter_we;
    memory #(
        .DATA_WIDTH(8),
        .ADDR_WIDTH(11)
    ) image (
        .clk(clk),
        .we(image_we),
        .addr(image_addr),
        .din(image_din),
        .dout(image_dout)
    );
```

```verilog
memory #(
    .DATA_WIDTH(8),
    .ADDR_WIDTH(7)
) filter (
    .clk(clk),
    .we(filter_we),
    .addr(filter_addr),
    .din(filter_din),
    .dout(filter_dout)
);

reg signed [31:0] out_conv [0:48];
reg [31:0] out_relu [0:48];
reg [31:0] out_max_pool [0:8];


localparam IDLE = 3'b000;
localparam READ_IMAGE = 3'b001;
localparam READ_FILTER = 3'b010;
localparam CONV = 3'b011;
localparam PREPARE = 3'b100;
localparam STOP = 3'b110;

reg [2:0] state, state_c;
reg [10:0] cnt, cnt_c;
always @(state or go or cnt) begin
    state_c = state;
    cnt_c = cnt - 11'b00000000001;
    case (state)
        IDLE:begin
            if(go == 1'b1)begin
                $display("go start\n");
                state_c = PREPARE;
                cnt_c = 11'b00000000000;
            end
            else
                cnt_c = 11'b00000000000;
        end
        PREPARE:begin
            if(cnt == 11'b00000000000)begin
                // $display("PREPARE finished\n");
                state_c = READ_IMAGE;
                cnt_c = 11'b10011001000;
                image_we = 1'b1;
            end
        end
        READ_IMAGE:begin
            if(cnt == 11'b00000000000) begin
                // $display("READ_IMAGE finished\n");
                state_c = READ_FILTER;
                cnt_c = 11'b00001111001;

                filter_we = 1'b1;
            end
        end
        READ_FILTER:begin
            if(cnt == 11'b00000000000) begin
                // $display("READ_FILTER finished\n");
                state_c = CONV;
                //cnt_c = 11'b00100001100;
                cnt_c = 11'b00000000001;
            end
        end
        CONV:begin
            if(cnt == 11'b00000000000) begin
                image_we = 1'b0;
                filter_we = 1'b0;
            end

        end
        STOP:begin end
        default: state_c = IDLE;
    endcase
end
always @(posedge clk) begin
    state <= state_c;
    cnt <= cnt_c;
    // $display("state_c: %d, cnt_c: %d\n",state_c, cnt);
end

reg [5:0] conv_cnt;
reg [3:0] row;
reg signed [31:0] sum;
reg [10:0] image_base_addr;
reg [6:0] filter_base_addr;
reg [3:0] idx;
reg [3:0] pos_cnt;
reg [5:0] max_pool_cnt[0:8];
reg [4:0] pos[0:8];
integer i;
always @(posedge clk) begin
    if (state == PREPARE) begin
        conv_cnt <= 0;
        row <= 0;
        sum <= 0;
        image_base_addr <= 0;
        filter_base_addr <= 0;
        idx <= 0;
        pos_cnt <= 0;
        max_pool_cnt[0] <= 16;
        max_pool_cnt[1] <= 18;
        max_pool_cnt[2] <= 20;
        max_pool_cnt[3] <= 30;
        max_pool_cnt[4] <= 32;
        max_pool_cnt[5] <= 34;
        max_pool_cnt[6] <= 44;
        max_pool_cnt[7] <= 46;
        max_pool_cnt[8] <= 48;
        pos[0] <= 16;
        pos[1] <= 15;
        pos[2] <= 14;
        pos[3] <= 9;
        pos[4] <= 8;
        pos[5] <= 7;
        pos[6] <= 2;
        pos[7] <= 1;
        pos[8] <= 0;
        out_max_pool[0] <= 0;
        out_max_pool[1] <= 0;
        out_max_pool[2] <= 0;
        out_max_pool[3] <= 0;
        out_max_pool[4] <= 0;
        out_max_pool[5] <= 0;
        out_max_pool[6] <= 0;
        out_max_pool[7] <= 0;
        out_max_pool[8] <= 0;


    end
    if (state == READ_IMAGE) begin
        // if(11'b10011001000 - cnt == 0)
            // $display("cnt:%d, image_addr:%d, image_pixel: %d\n",cnt,
11'b10011001000 - cnt,image_pixel);
        image_din <= {{80{1'b1}},image_pixel};
        image_addr <= {{110{1'b1}},11'b10011001000 - cnt};
    end
    else if (state == READ_FILTER) begin
        // $display("cnt:%d, filter_addr:%b, filter_pixel: %d\n",cnt,
{7'b1111000 - cnt[6:0]}, $signed(filter_pixel));
        filter_din <= {{80{1'b1}},filter_pixel};
        filter_addr <= {{70{1'b1}},7'b1111001 - cnt[6:0]};
    end
    else if(state == CONV) begin
        if(conv_cnt <= 49) begin
            // stage 1: read data

            if(row == 11) begin
                image_addr <= image_addr;
                filter_addr <= filter_addr;
                row <= 0;
                conv_cnt <= conv_cnt + 1;
                image_base_addr <= 35 * ((conv_cnt + 1) / 7 * 4) + (conv_cnt +
1) % 7 * 4;
                filter_base_addr <= 0;

            end
            else begin
                // $display("row:%d, filter_addr_1 = %d, filter_addr_2 = %d\n",
row,filter_addr[7 * 2 - 1 : 7 * 1], filter_addr[7 * 1 - 1 : 7 * 0]);
                image_addr[11 * 11 - 1 : 11 * 10] <= image_base_addr + 0;
                image_addr[11 * 10 - 1 : 11 * 9] <= image_base_addr + 1;
                image_addr[11 * 9 - 1 : 11 * 8] <= image_base_addr + 2;
                image_addr[11 * 8 - 1 : 11 * 7] <= image_base_addr + 3;
                image_addr[11 * 7 - 1 : 11 * 6] <= image_base_addr + 4;
                image_addr[11 * 6 - 1 : 11 * 5] <= image_base_addr + 5;
                image_addr[11 * 5 - 1 : 11 * 4] <= image_base_addr + 6;
                image_addr[11 * 4 - 1 : 11 * 3] <= image_base_addr + 7;
                image_addr[11 * 3 - 1 : 11 * 2] <= image_base_addr + 8;
                image_addr[11 * 2 - 1 : 11 * 1] <= image_base_addr + 9;
                image_addr[11 * 1 - 1 : 11 * 0] <= image_base_addr + 10;

                filter_addr[7 * 11 - 1 : 7 * 10] <= filter_base_addr + 0;
                filter_addr[7 * 10 - 1 : 7 * 9] <= filter_base_addr + 1;
                filter_addr[7 * 9 - 1 : 7 * 8] <= filter_base_addr + 2;
                filter_addr[7 * 8 - 1 : 7 * 7] <= filter_base_addr + 3;
                filter_addr[7 * 7 - 1 : 7 * 6] <= filter_base_addr + 4;
                filter_addr[7 * 6 - 1 : 7 * 5] <= filter_base_addr + 5;
                filter_addr[7 * 5 - 1 : 7 * 4] <= filter_base_addr + 6;
                filter_addr[7 * 4 - 1 : 7 * 3] <= filter_base_addr + 7;
                filter_addr[7 * 3 - 1 : 7 * 2] <= filter_base_addr + 8;
                filter_addr[7 * 2 - 1 : 7 * 1] <= filter_base_addr + 9;
                filter_addr[7 * 1 - 1 : 7 * 0] <= filter_base_addr + 10;
                image_base_addr <= image_base_addr + 35;
                filter_base_addr <= filter_base_addr + 11;
                row <= row + 1;
            end

            //stage 2: multiply and add up
            if(row == 0) begin
                // out_conv[conv_cnt - 1] <= sum;
                // $display("out_conv[%d]: %d",conv_cnt - 1, sum);
                // sum <= 0;
                out_conv[conv_cnt] <= 0;
                // $display("conv_out: %b", conv_out);
                conv_out <= {out_conv[0], out_conv[1], out_conv[2], out_conv[3],
out_conv[4], out_conv[5], out_conv[6],
                          out_conv[7], out_conv[8], out_conv[9], out_conv[10],
out_conv[11], out_conv[12], out_conv[13],
                          out_conv[14], out_conv[15], out_conv[16],
out_conv[17], out_conv[18], out_conv[19], out_conv[20],
                          out_conv[21], out_conv[22], out_conv[23],
out_conv[24], out_conv[25], out_conv[26], out_conv[27],
                          out_conv[28], out_conv[29], out_conv[30],
out_conv[31], out_conv[32], out_conv[33], out_conv[34],
                          out_conv[35], out_conv[36], out_conv[37],
out_conv[38], out_conv[39], out_conv[40], out_conv[41],
                          out_conv[42], out_conv[43], out_conv[44],
out_conv[45], out_conv[46], out_conv[47], out_conv[48]};
            end
            else begin
                out_conv[conv_cnt] <= $signed({1'b0,image_dout[8 * 11 - 1 : 8 *
10]}) * $signed(filter_dout[8 * 11 - 1 : 8 * 10])
                + $signed({18'b0,image_dout[8 * 10 - 1 : 8 * 9]}) *
$signed(filter_dout[8 * 10 - 1 : 8 * 9])
                + $signed({18'b0,image_dout[8 * 9 - 1 : 8 * 8]}) *
$signed(filter_dout[8 * 9 - 1 : 8 * 8])
                + $signed({18'b0,image_dout[8 * 8 - 1 : 8 * 7]}) *
$signed(filter_dout[8 * 8 - 1 : 8 * 7])
                + $signed({18'b0,image_dout[8 * 7 - 1 : 8 * 6]}) *
$signed(filter_dout[8 * 7 - 1 : 8 * 6])
                + $signed({18'b0,image_dout[8 * 6 - 1 : 8 * 5]}) *
$signed(filter_dout[8 * 6 - 1 : 8 * 5])
```

```
                    + $signed({18'b0,image_dout[8 * 5 - 1 : 8 * 4]}) *
$signed(filter_dout[8 * 5 - 1 : 8 * 4])
                    + $signed({18'b0,image_dout[8 * 4 - 1 : 8 * 3]}) *
$signed(filter_dout[8 * 4 - 1 : 8 * 3])
                    + $signed({18'b0,image_dout[8 * 3 - 1 : 8 * 2]}) *
$signed(filter_dout[8 * 3 - 1 : 8 * 2])
                    + $signed({18'b0,image_dout[8 * 2 - 1 : 8 * 1]}) *
$signed(filter_dout[8 * 2 - 1 : 8 * 1])
                    + $signed({18'b0,image_dout[8 * 1 - 1 : 8 * 0]}) *
$signed(filter_dout[8 * 1 - 1 : 8 * 0])
                    + out_conv[conv_cnt];
                end

            end
            if(row == 0 && conv_cnt != 0) begin // 1 conv complete -> ReLU
                out_relu[conv_cnt - 1] <= (out_conv[conv_cnt - 1][31] == 1'b1) ? 0 :
out_conv[conv_cnt - 1];
                // $display("out_relu[%d]: %d\n",conv_cnt - 1,  (out_conv[conv_cnt -
1][31] == 1'b1) ? 0 : out_conv[conv_cnt - 1]);
                relu_out <= {out_relu[0], out_relu[1], out_relu[2], out_relu[3],
out_relu[4], out_relu[5], out_relu[6],
                            out_relu[7], out_relu[8], out_relu[9], out_relu[10],
out_relu[11], out_relu[12], out_relu[13],
                            out_relu[14], out_relu[15], out_relu[16], out_relu[17],
out_relu[18], out_relu[19], out_relu[20],
                            out_relu[21], out_relu[22], out_relu[23], out_relu[24],
out_relu[25], out_relu[26], out_relu[27],
                            out_relu[28], out_relu[29], out_relu[30], out_relu[31],
out_relu[32], out_relu[33], out_relu[34],
                            out_relu[35], out_relu[36], out_relu[37], out_relu[38],
out_relu[39], out_relu[40], out_relu[41],
                            out_relu[42], out_relu[43], out_relu[44], out_relu[45],
out_relu[46], out_relu[47], out_relu[48]};
            end
            if(row >= 1 && conv_cnt - 1 == max_pool_cnt[idx]) begin
                // $display("conv_cnt: %d, max_pool_cnt[%d]: %d,
pos_cnt: %d\n",conv_cnt - 1, idx, max_pool_cnt[idx], pos_cnt);
                if(pos_cnt < 9) begin
                    if(out_max_pool[idx] < out_relu[conv_cnt - 1 - pos[pos_cnt]])
                        out_max_pool[idx] <= out_relu[conv_cnt - 1 - pos[pos_cnt]];
                    else
                        out_max_pool[idx] <= out_max_pool[idx];
                    pos_cnt <= pos_cnt + 1;
                    // $display("out_max_pool[%d]: %d, out_relu[%d - 1 -
pos[%d]]: %d\n",idx, out_max_pool[idx],conv_cnt, pos_cnt, out_relu[conv_cnt - 1 -
pos[pos_cnt]]);
                end
                else begin
                    out <= {out_max_pool[0],out_max_pool[1], out_max_pool[2],
out_max_pool[3],
                    out_max_pool[4], out_max_pool[5], out_max_pool[6],
                    out_max_pool[7], out_max_pool[8]};
                    pos_cnt <= 0;
                    idx <= idx + 1;
                end
            end
        end
    end

endmodule
```

```
`timescale 10ps/1ps
module prob2_tbench;

    reg clk, reset;
    reg [7:0] image_pixel;
    reg [7:0] filter_pixel;
    reg go;
    wire [32 * 49 - 1:0] conv_out;
    wire [32 * 49 - 1:0] relu_out;
    wire [32 * 9 - 1:0] out;

    prob2 uut (
        .clk(clk),
        .image_pixel(image_pixel),
        .filter_pixel(filter_pixel),
        .go(go),
        .conv_out(conv_out),
        .relu_out(relu_out),
        .out(out)
    );

    integer i, data_type;
    integer scan_image, scan_filter;
    integer image_file, filter_file;
    integer r;
    initial begin
        $recordfile("prob2_tbench");
        $recordvars(prob2_tbench);
        reset = 1'b1;
        #500;
        reset = 1'b0;

        go = 0;
        data_type = 3;
        image_pixel = 0;
        filter_pixel = 0;
        repeat (5) @ (posedge clk);
        go = 1;
        if(data_type == 0) begin
            image_file = $fopen("image_0.txt", "w");
            filter_file = $fopen("filter_0.txt", "w");
            r = 123;
            // $fwrite(image_file, "aa = [");
            @ (posedge clk);
            for (i = 0; i < 1225; i = i + 1) begin
                @ (posedge clk);
                r = $random(r);
                image_pixel = (r & 8'hFF);
                $fwrite(image_file, "%d ", image_pixel);
                if (i % 35 == 34 && i != 1224)
                    $fwrite(image_file, "\n");
            end
            // $fwrite(image_file, "]\n");
            // $fwrite(image_file, "f = [");
            for (i = 0; i < 121; i = i + 1) begin
                @ (posedge clk);
                r = $random(r);
                filter_pixel = $signed((r & 8'hFF) - 128);
                $fwrite(filter_file, "%d ", $signed(filter_pixel));
                if (i % 11 == 10 && i != 120)
                    $fwrite(filter_file, "\n");
            end
            // $fwrite(image_file, "]\n");
        end
        else if (data_type == 1) begin
            // @(posedge clk);
            @(posedge clk);
            $display("data_type == 1 start\n");
            image_file = $fopen("image_1.txt", "r");
            filter_file = $fopen("filter_1.txt", "r");
            for (i = 0; i < 1225; i = i + 1) begin
                @(posedge clk);
                scan_image = $fscanf(image_file, "%d", image_pixel);
                if (scan_image != 1) begin
                    $display("Error: Failed to read image_data.txt!");
                    $finish;
                end
            end
            // $display("read image finished\n");
            // @(posedge clk);
            // @(posedge clk);
            for (i = 0; i < 121; i = i + 1) begin
                @(posedge clk);
                scan_filter = $fscanf(filter_file, "%d", filter_pixel);
                if (scan_filter != 1) begin
                    $display("Error: Failed to read filter_data.txt!");
                    $finish;
                end
            end
            // $display("read filter finished\n");
        end
        else if (data_type == 2) begin
            // @(posedge clk);
            @(posedge clk);
            $display("data_type == 2 start\n");
            image_file = $fopen("image_2.txt", "r");
            filter_file = $fopen("filter_2.txt", "r");
            for (i = 0; i < 1225; i = i + 1) begin
                @(posedge clk);
                scan_image = $fscanf(image_file, "%d", image_pixel);
                if (scan_image != 1) begin
                    $display("Error: Failed to read image_data.txt!");
                    $finish;
                end
            end
            // $display("read image finished\n");
            // @(posedge clk);
            // @(posedge clk);
            for (i = 0; i < 121; i = i + 1) begin
                @(posedge clk);
                scan_filter = $fscanf(filter_file, "%d", filter_pixel);
                if (scan_filter != 1) begin
                    $display("Error: Failed to read filter_data.txt!");
                    $finish;
                end
            end
            // $display("read filter finished\n");
        end
        else if (data_type == 3) begin
            // @(posedge clk);
            @(posedge clk);
            $display("data_type == 3 start\n");
            image_file = $fopen("image_3.txt", "r");
            filter_file = $fopen("filter_3.txt", "r");
            for (i = 0; i < 1225; i = i + 1) begin
                @(posedge clk);
                scan_image = $fscanf(image_file, "%d", image_pixel);
                if (scan_image != 1) begin
                    $display("Error: Failed to read image_data.txt!");
                    $finish;
                end
            end
            // $display("read image finished\n");
            // @(posedge clk);
            // @(posedge clk);
            for (i = 0; i < 121; i = i + 1) begin
                @(posedge clk);
                scan_filter = $fscanf(filter_file, "%d", filter_pixel);
                if (scan_filter != 1) begin
                    $display("Error: Failed to read filter_data.txt!");
                    $finish;
                end
            end
            // $display("read filter finished\n");
        end
        repeat (1000) @ (posedge clk);
        $display("%d %d %d %d %d %d %d\n%d %d %d %d %d %d %d\n%d %d %d %d %d %d %d\n
    %d %d %d %d %d %d %d\n%d %d %d %d %d %d %d\n%d %d %d %d %d %d %d\n%d %d %d %d %d
    %d\n",
            $signed(conv_out[32 * 49 - 1 : 32 * 48]), $signed(conv_out[32 * 48 - 1 :
    32 * 47]), $signed(conv_out[32 * 47 - 1 : 32 * 46]), $signed(conv_out[32 * 46 - 1 :
    32 * 45]), $signed(conv_out[32 * 45 - 1 : 32 * 44]), $signed(conv_out[32 * 44 - 1 :
    32 * 43]), $signed(conv_out[32 * 43 - 1 : 32 * 42]),
            $signed(conv_out[32 * 42 - 1 : 32 * 41]), $signed(conv_out[32 * 41 - 1 :
    32 * 40]), $signed(conv_out[32 * 40 - 1 : 32 * 39]), $signed(conv_out[32 * 39 - 1 :
```

```
32 * 38]), $signed(conv_out[32 * 38 - 1 : 32 * 37]), $signed(conv_out[32 * 37 - 1 :
32 * 36]), $signed(conv_out[32 * 36 - 1 : 32 * 35]),
            $signed(conv_out[32 * 35 - 1 : 32 * 34]), $signed(conv_out[32 * 34 - 1 :
32 * 33]), $signed(conv_out[32 * 33 - 1 : 32 * 32]), $signed(conv_out[32 * 32 - 1 :
32 * 31]), $signed(conv_out[32 * 31 - 1 : 32 * 30]), $signed(conv_out[32 * 30 - 1 :
32 * 29]), $signed(conv_out[32 * 29 - 1 : 32 * 28]),
            $signed(conv_out[32 * 28 - 1 : 32 * 27]), $signed(conv_out[32 * 27 - 1 :
32 * 26]), $signed(conv_out[32 * 26 - 1 : 32 * 25]), $signed(conv_out[32 * 25 - 1 :
32 * 24]), $signed(conv_out[32 * 24 - 1 : 32 * 23]), $signed(conv_out[32 * 23 - 1 :
32 * 22]), $signed(conv_out[32 * 22 - 1 : 32 * 21]),
            $signed(conv_out[32 * 21 - 1 : 32 * 20]), $signed(conv_out[32 * 20 - 1 :
32 * 19]), $signed(conv_out[32 * 19 - 1 : 32 * 18]), $signed(conv_out[32 * 18 - 1 :
32 * 17]), $signed(conv_out[32 * 17 - 1 : 32 * 16]), $signed(conv_out[32 * 16 - 1 :
32 * 15]), $signed(conv_out[32 * 15 - 1 : 32 * 14]),
            $signed(conv_out[32 * 14 - 1 : 32 * 13]), $signed(conv_out[32 * 13 - 1 :
32 * 12]), $signed(conv_out[32 * 12 - 1 : 32 * 11]), $signed(conv_out[32 * 11 - 1 :
32 * 10]), $signed(conv_out[32 * 10 - 1 : 32 * 9]), $signed(conv_out[32 * 9 - 1 :
32 * 8]), $signed(conv_out[32 * 8 - 1 : 32 * 7]),
            $signed(conv_out[32 * 7 - 1 : 32 * 6]), $signed(conv_out[32 * 6 - 1 : 32
* 5]), $signed(conv_out[32 * 5 - 1 : 32 * 4]), $signed(conv_out[32 * 4 - 1 : 32 *
3]), $signed(conv_out[32 * 3 - 1 : 32 * 2]), $signed(conv_out[32 * 2 - 1 : 32 *
1]), $signed(conv_out[32 * 1 - 1 : 32 * 0]));


        $display("%d %d %d %d %d %d %d\n%d %d %d %d %d %d %d\n%d %d %d %d %d %d %d\n
%d %d %d %d %d %d %d\n%d %d %d %d %d %d %d\n%d %d %d %d %d %d %d\n%d %d %d %d %d
%d\n",
            $signed(relu_out[32 * 49 - 1 : 32 * 48]), $signed(relu_out[32 * 48 - 1 :
32 * 47]), $signed(relu_out[32 * 47 - 1 : 32 * 46]), $signed(relu_out[32 * 46 - 1 :
32 * 45]), $signed(relu_out[32 * 45 - 1 : 32 * 44]), $signed(relu_out[32 * 44 - 1 :
32 * 43]), $signed(relu_out[32 * 43 - 1 : 32 * 42]),
            $signed(relu_out[32 * 42 - 1 : 32 * 41]), $signed(relu_out[32 * 41 - 1 :
32 * 40]), $signed(relu_out[32 * 40 - 1 : 32 * 39]), $signed(relu_out[32 * 39 - 1 :
32 * 38]), $signed(relu_out[32 * 38 - 1 : 32 * 37]), $signed(relu_out[32 * 37 - 1 :
32 * 36]), $signed(relu_out[32 * 36 - 1 : 32 * 35]),
            $signed(relu_out[32 * 35 - 1 : 32 * 34]), $signed(relu_out[32 * 34 - 1 :
32 * 33]), $signed(relu_out[32 * 33 - 1 : 32 * 32]), $signed(relu_out[32 * 32 - 1 :
32 * 31]), $signed(relu_out[32 * 31 - 1 : 32 * 30]), $signed(relu_out[32 * 30 - 1 :
32 * 29]), $signed(relu_out[32 * 29 - 1 : 32 * 28]),
            $signed(relu_out[32 * 28 - 1 : 32 * 27]), $signed(relu_out[32 * 27 - 1 :
32 * 26]), $signed(relu_out[32 * 26 - 1 : 32 * 25]), $signed(relu_out[32 * 25 - 1 :
32 * 24]), $signed(relu_out[32 * 24 - 1 : 32 * 23]), $signed(relu_out[32 * 23 - 1 :
32 * 22]), $signed(relu_out[32 * 22 - 1 : 32 * 21]),
            $signed(relu_out[32 * 21 - 1 : 32 * 20]), $signed(relu_out[32 * 20 - 1 :
32 * 19]), $signed(relu_out[32 * 19 - 1 : 32 * 18]), $signed(relu_out[32 * 18 - 1 :
32 * 17]), $signed(relu_out[32 * 17 - 1 : 32 * 16]), $signed(relu_out[32 * 16 - 1 :
32 * 15]), $signed(relu_out[32 * 15 - 1 : 32 * 14]),
            $signed(relu_out[32 * 14 - 1 : 32 * 13]), $signed(relu_out[32 * 13 - 1 :
32 * 12]), $signed(relu_out[32 * 12 - 1 : 32 * 11]), $signed(relu_out[32 * 11 - 1 :
32 * 10]), $signed(relu_out[32 * 10 - 1 : 32 * 9]), $signed(relu_out[32 * 9 - 1 :
32 * 8]), $signed(relu_out[32 * 8 - 1 : 32 * 7]),
            $signed(relu_out[32 * 7 - 1 : 32 * 6]), $signed(relu_out[32 * 6 - 1 : 32
* 5]), $signed(relu_out[32 * 5 - 1 : 32 * 4]), $signed(relu_out[32 * 4 - 1 : 32 *
3]), $signed(relu_out[32 * 3 - 1 : 32 * 2]), $signed(relu_out[32 * 2 - 1 : 32 *
1]), $signed(relu_out[32 * 1 - 1 : 32 * 0]));

        $display("%d %d %d\n%d %d %d\n%d %d %d",
                    out[32 * 9 - 1 : 32 * 8],
                    out[32 * 8 - 1 : 32 * 7],
                    out[32 * 7 - 1 : 32 * 6],
                    out[32 * 6 - 1 : 32 * 5],
                    out[32 * 5 - 1 : 32 * 4],
                    out[32 * 4 - 1 : 32 * 3],
                    out[32 * 3 - 1 : 32 * 2],
                    out[32 * 2 - 1 : 32 * 1],
                    out[32 * 1 - 1 : 32 * 0]);
        $finish;
    end
    always begin
        if(reset == 1'b1)begin
            clk = 1'b0;
            #1;
        end
        else begin
            #100
            clk = ~clk;
        end
    end

endmodule
```