National Cheng Kung University

Department of Electrical Engineering

Introduction to VLSI CAD (Spring 2024)

Lab Session 6

Design of Local Binary Pattern Circuit

Name	Student ID	
劉冠妤	E24116152	
Practical	Points	Marks
Lab 6_1	35	
Lab 6_2	65	
Notes		

Due: 15:00 April 17, 2024 @ moodle

Summary

Hardware		
	RTL(∨/X)	Synthesis(∨/X)
LBP	∨	∨
CLBP	∨	∨

Synthesis result	
Area	Simulation time (ps)
LBP : 346.032010 CLBP : 5074.099316	LBP : 98294030 CLBP : 394559071

Superlint(number of inline messages)			
Total lines	Warning	Error	coverage(%)
LBP : 260 CLBP : 644	3 49	0 0	98.8% 92.4%

Note: You must complete and fill out this form with your design information!!!

Deliverables

1) All Verilog codes including testbenches, .bmp and .hex for each problem should be uploaded.
2) NOTE: Please **DO NOT** include source code in the paper report!
3) NOTE: Please **DO NOT** upload waveforms (.fsdb or .vcd)!
4) If you upload a dead body which we can't even compile, you will get NO credit!
5) All Verilog file should get at least 90% SuperLint Coverage.
6) All homework requirements should be uploaded in this file hierarchy, or you will not get full credit. If you want to use some sub modules in your design but you do not include them in your tar file, you will get 0 point.

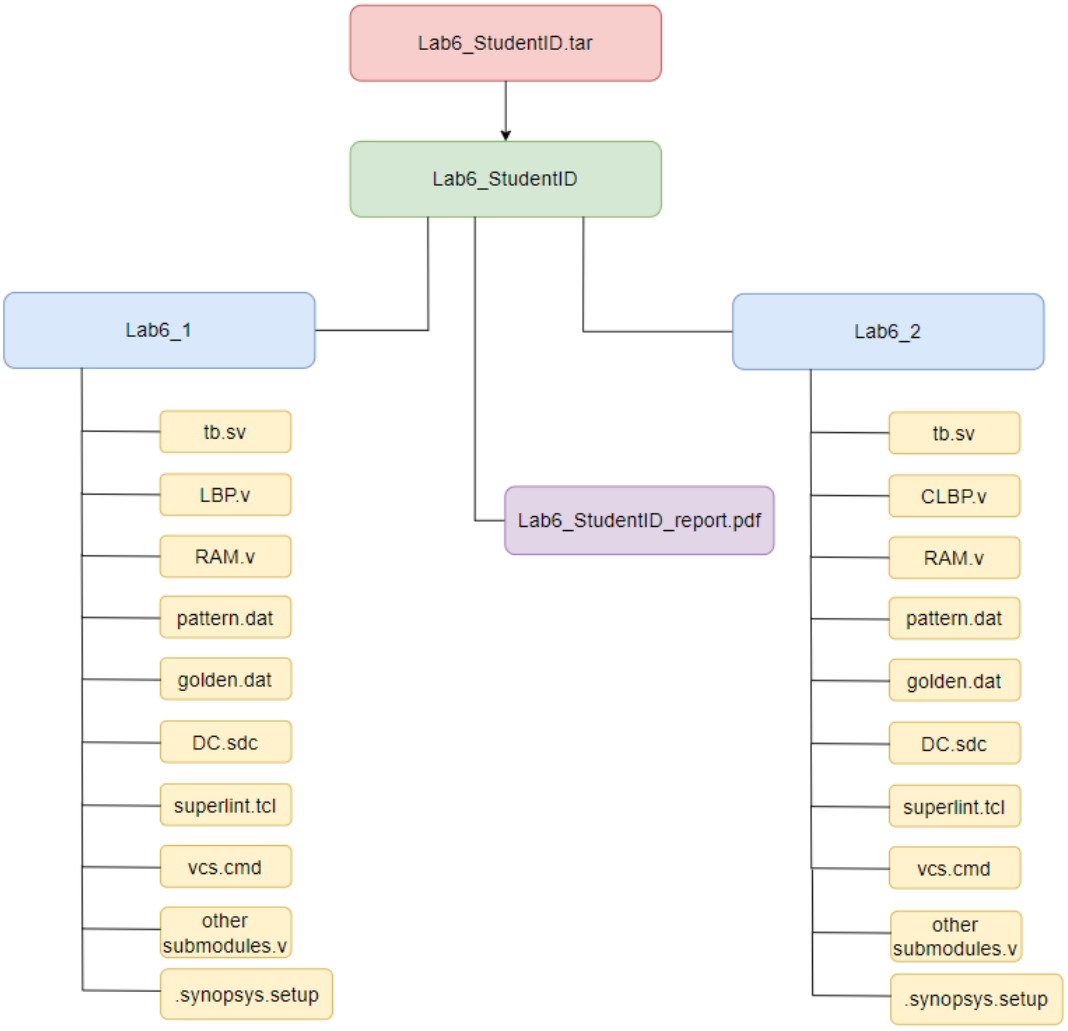


Fig.1 File hierarchy for Homework submission

The design inside the LBP block can be completed by your free will, but do not modify the I/O ports of the LBP block. The block diagram of the testbed-DUT (design under test) system is as shown in **Fig2**.

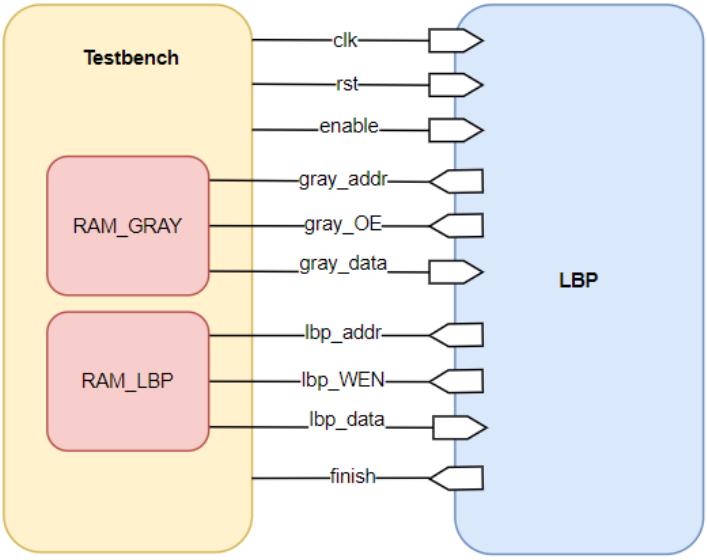


Fig2. The block diagram of local binary pattern circuit

➢ Port list of LBP:

Signal	I/O	Bit-width	Description
clk	I	1	Clock signal
rst	I	1	Reset signal
enable	I	1	Circuit enabling signal
gray_addr	O	12	Address signal connected to RAM_GRAY
gray_OE	O	1	Read enable signal to RAM_GRAY
gray_data	I	8	Read data signal from RAM_GRAY
lbp_addr	O	12	Address signal connected to RAM_LBP
lbp_WEN	O	1	Write enable signal to RAM_LBP
lbp_data	O	8	Write data signal to RAM_LBP
finish	O	1	Indication signal of the circuit is finished

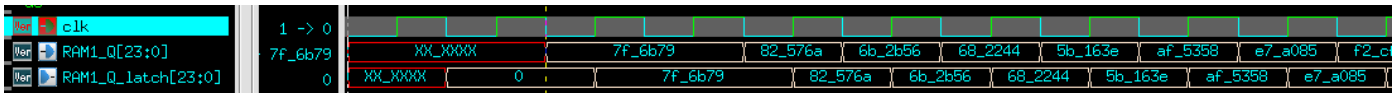


Fig3. example waveform for RAM

- Understanding the function:

  Once system is initialized, it

  a) Choose a pixel in the image and select its neighboring pixels.

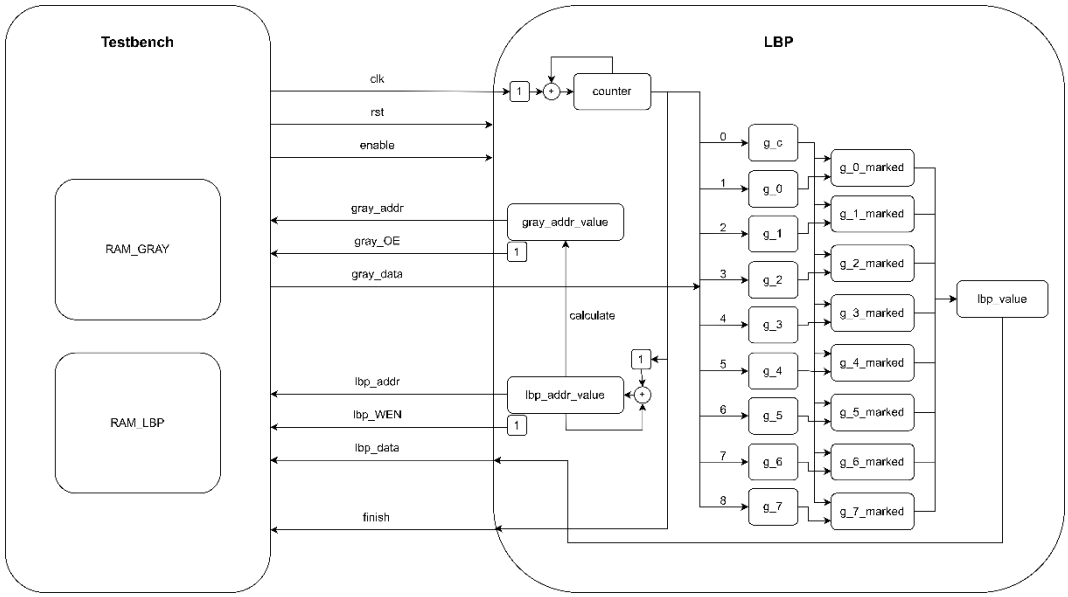
  b) Construct the mask using threshold function.

  c) Combine the binary values for all neighboring pixels to obtain a binary code for the central pixel and convert it to a decimal value.

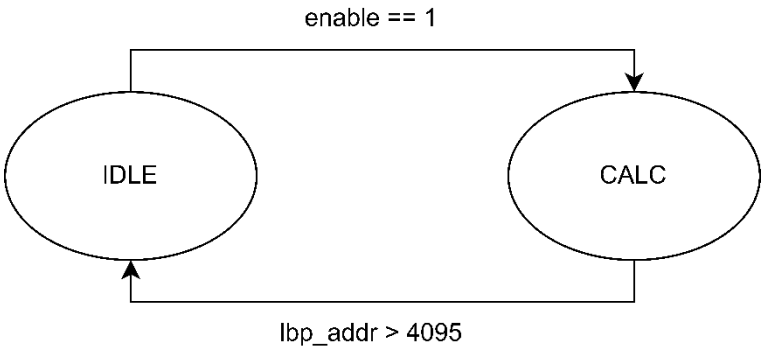
  d) Repeat steps a)–c) for each pixel in the image to obtain a binary code for each pixel.

- Know the basic design rules
  - All operations are activated on the positive edge of the clock.
  - Control signals:
    - *RAM_WE*: To store the data into RAM
    - *RAM_OE*: To read data from RAM
    - *finish*: Stop the process

- Describe your design in detail. You can draw internal architecture or block diagram to help elaborate your design, if don't, plain text description is allowed.



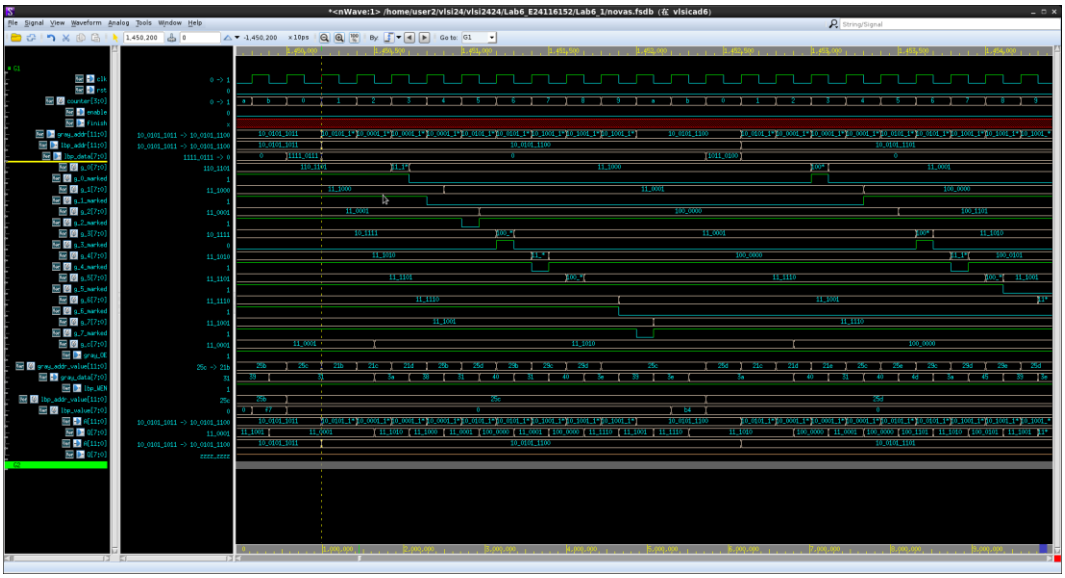
  - Controller
    - Draw your state diagram in controller and explain it.



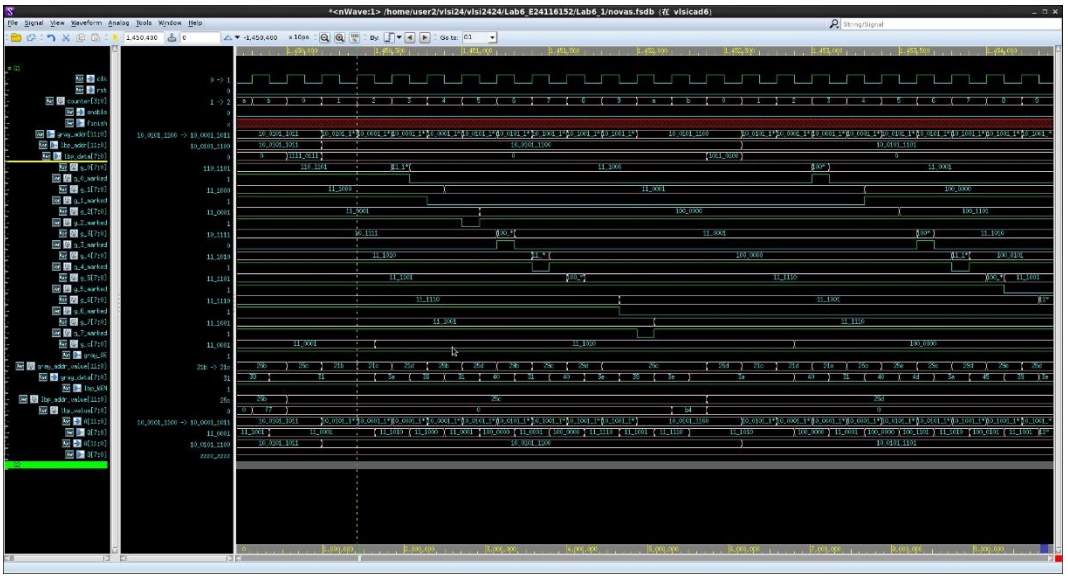
Because this program has more calculation parts, in order to reduce the possibility of errors, I did not divide it into too many states for processing. Instead, I used a CALC state to represent the calculation process.

1. Complete the LBP module, in the system.
2. Compile the verilog code to verify the operations of this module works properly.
3. Synthesize your *LBP.v* with following constraint:
   - Clock period: no more than 2.0 ns.
   - Don't touch network: clk.
   - Wire load model: N16ADFP_StdCellss0p72vm40c.
   - Synthesized verilog file: *LBP_syn.v*.
   - Timing constraint file: *LBP_syn.sdf*.
4. Please **attach your waveforms** and **specify your operations** on the waveforms. Read the value of gray data in sequence.

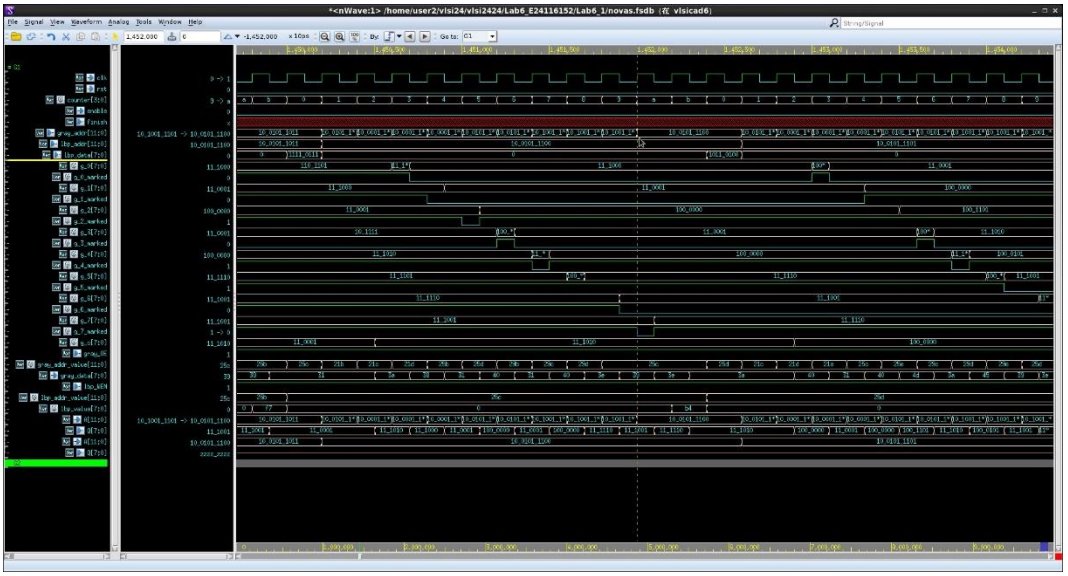
   g_c



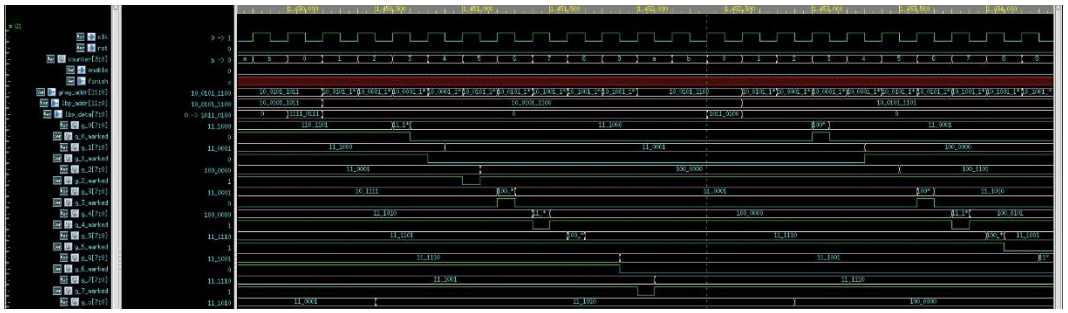
g_0



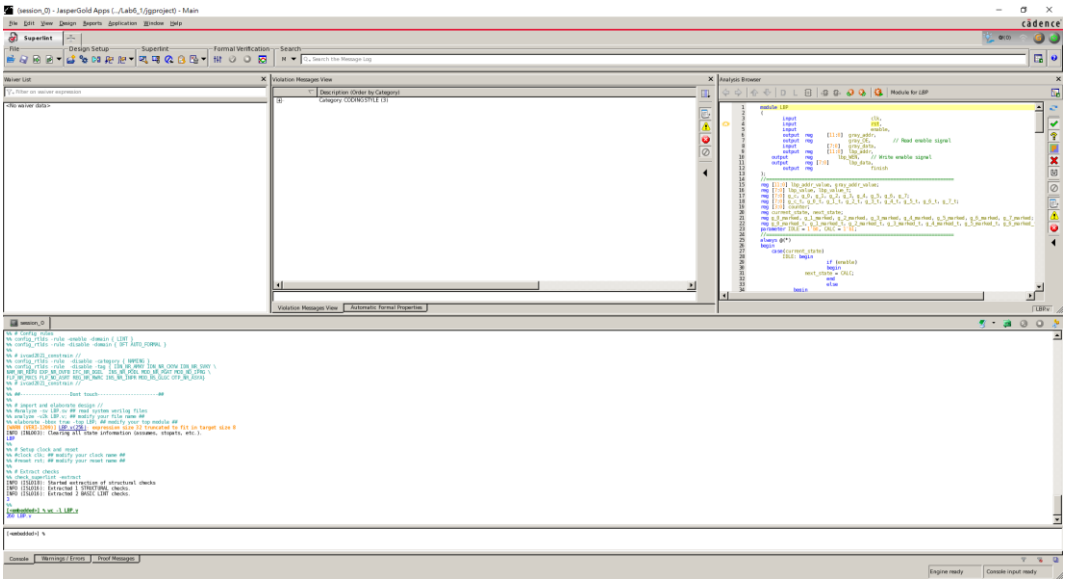
g_1、g_2、g_3、g_4、g_5、g_6, then g_7



Finally, lbp_data is output according to the comparison result.



5. Show SuperLint coverage (including all files)



6. Your clock period, total cell area, post simulation time with screenshot.

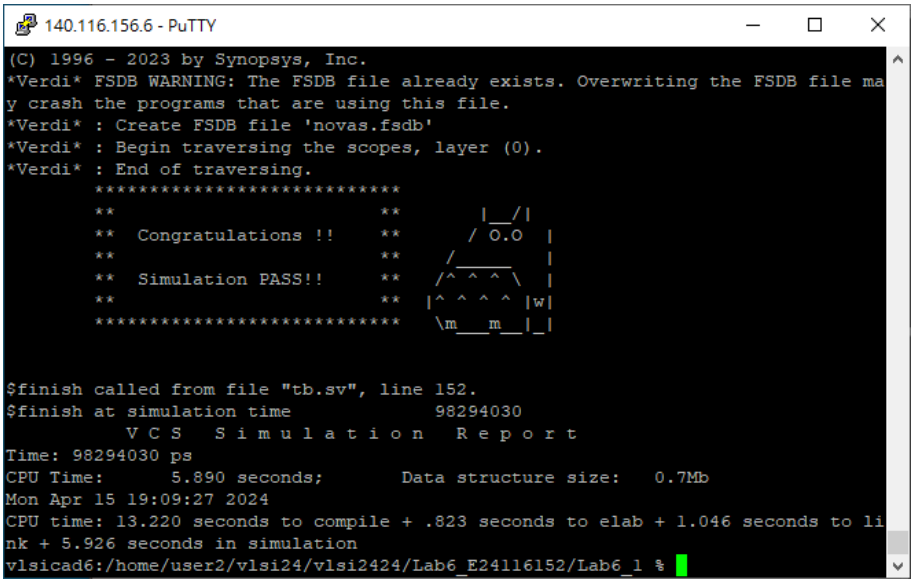
Clock period: 0.57

```
Path Group: clk
Path Type: max

Des/Clust/Port     Wire Load Model       Library
------------------------------------------------
LBP                ZeroWireload          N16ADFP_StdCellss0p72vm40c

Point                                                   Incr       Path
-----------------------------------------------------------------------------
clock clk (rise edge)                                   0.00       0.00
clock network delay (ideal)                             0.20       0.20
input external delay                                    1.20       1.40 f
gray_data[1] (in)                                       0.01       1.41 f
U699/ZN (AOI22D1BWP16P90)                               0.04       1.45 r
U343/ZN (CKND1BWP16P90LVT)                              0.03       1.48 f
U572/ZN (OAI211D1BWP16P90)                              0.02       1.50 r
U571/ZN (OAI21D1BWP16P90)                               0.01       1.51 f
U569/ZN (AOI221D1BWP16P90)                              0.03       1.54 r
U568/ZN (AOI22D1BWP16P90)                               0.02       1.55 f
U564/ZN (OAI221D1BWP16P90)                              0.02       1.57 r
U562/Z (CKMUX2D1BWP16P90)                               0.03       1.60 r
g_3_marked_t_reg/D (DFCNQD2BWP16P90LVT)                 0.00       1.60 r
data arrival time                                                  1.60

clock clk (rise edge)                                   2.00       2.00
clock network delay (ideal)                             0.20       2.20
clock uncertainty                                      -0.02       2.18
g_3_marked_t_reg/CP (DFCNQD2BWP16P90LVT)                0.00       2.18 r
library setup time                                     -0.01       2.17
data required time                                                 2.17
-----------------------------------------------------------------------------
data required time                                                 2.17
data arrival time                                                 -1.60
-----------------------------------------------------------------------------
slack (MET)                                                        0.57


***** End Of Report *****
```

Total cell area: 346.032010

```
****************************************
Report : area
Design : LBP
Version: O-2018.06
Date   : Mon Apr 15 19:04:57 2024
****************************************

Library(s) Used:

    N16ADFP_StdCellss0p72vm40c (File: /usr/cad/CBDK/Executable_Packa

Number of ports:                           94
Number of nets:                           831
Number of cells:                          764
Number of combinational cells:            624
Number of sequential cells:               138
Number of macros/black boxes:               0
Number of buf/inv:                        127
Number of references:                      50

Combinational area:               192.170885
Buf/Inv area:                      21.513601
Noncombinational area:            153.861125
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  346.032010
Total area:                 undefined

***** End Of Report *****
```

Post simulation time: 98294030ps



7. Please describe how you optimize your design when you run into problems in synthesis. .e.g., plug in some registers between two instances to shorten your datapath, resource sharing for some registers to reduce your cell area.

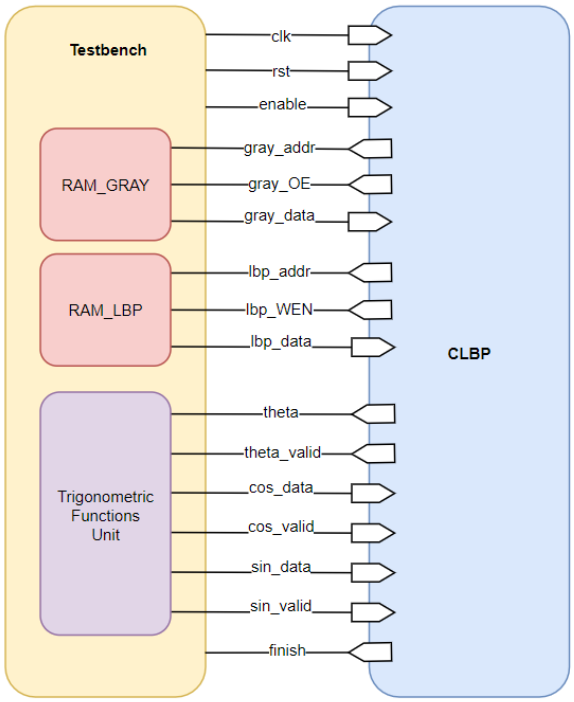
   First, I examine and optimize the structure of the circuit to ensure timing and area issues are minimized. This may include adjusting the layout of logic gates, optimizing data paths, and ensuring proper clock distribution. Second, I use the reporting and analysis capabilities provided by the synthesis tool to check for and resolve any timing constraint violations. This may involve adjusting timing constraints and optimizing clock and signal paths to ensure the circuit operates correctly. In addition, I will optimize the register configuration and resource sharing in the circuit to reduce area and power consumption. This includes fine-grained optimizations to the datapath, such as inserting registers to shorten the datapath, and sharing resources to reduce duplicate elements in the circuit. Finally, I perform post-synthesis simulation and verification to ensure the circuit functions properly on the physical device. If I find any issues, I go back to the design stage to tweak and optimize until all requirements are met.

8. Lessons learned from this lab

   Although this question is relatively easy compared to the second question, it is still relatively difficult compared to previous experiments. At least I encountered fewer problems with this question, and it also helped me solve the subsequent structural problems.

Lab 6_2: Circular Local Binary Pattern

Extend the original LBP algorithm to circular one.



▲The block diagram of circular local binary pattern circuit

➢ Port list of top:

Signal	I/O	Bit-width	Description
clk	I	1	Clock signal
rst	I	1	Reset signal
enable	I	1	Circuit enabling signal
gray_addr	O	12	Address signal connected to RAM_GRAY
gray_OE	O	1	Read enable signal to RAM_GRAY
gray_data	I	8	Read data signal from RAM_GRAY
lbp_addr	O	12	Address signal connected to RAM_LBP

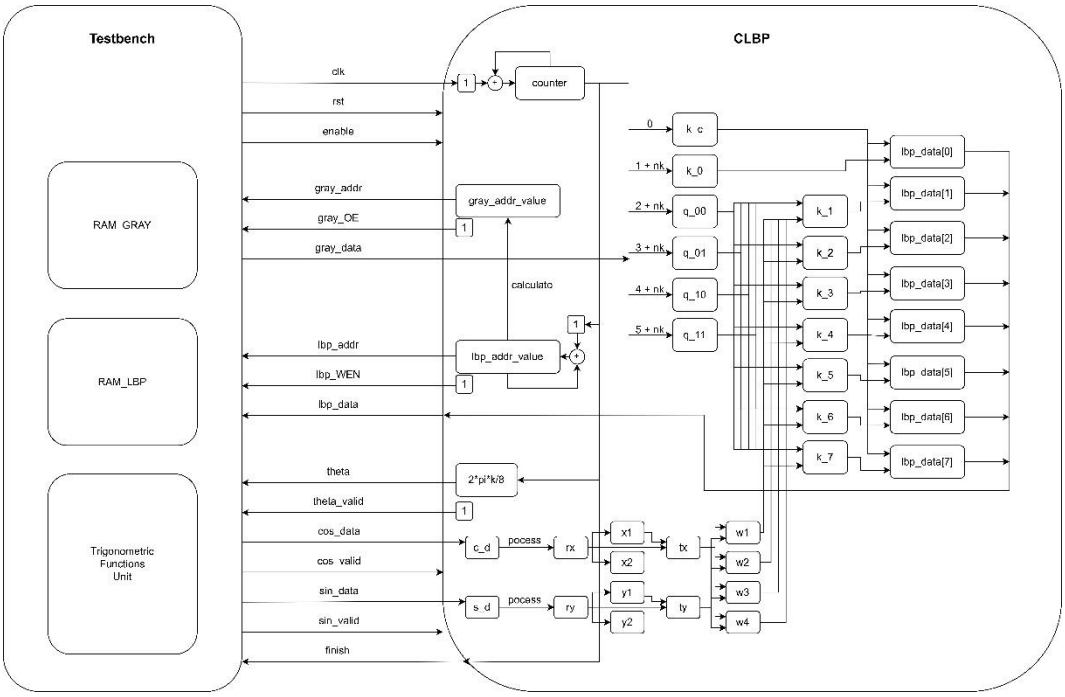
lbp_WEN	**O**	**1**	**Write enable signal to RAM_LBP**
lbp_data	**O**	**8**	**Write data signal to RAM_LBP**
theta	**O**	**25(fixed-point)**	**Current neighbor's angle signal(unit is in radian)**
theta_valid	**O**	**1**	**Indication signal of current neighbor's angle is valid**
cos_data	**I**	**25(fixed-point)**	**Cosine value of the theta(from testbench)**
cos_valid	**I**	**1**	**Indication signal of cosine value is valid**
sin_data	**I**	**25(fixed-point)**	**Sine value of the theta(from testbench)**
sin_valid	**I**	**1**	**Indication signal of sine value is valid**
finish	**O**	**1**	**Indication signal of the circuit is finished**

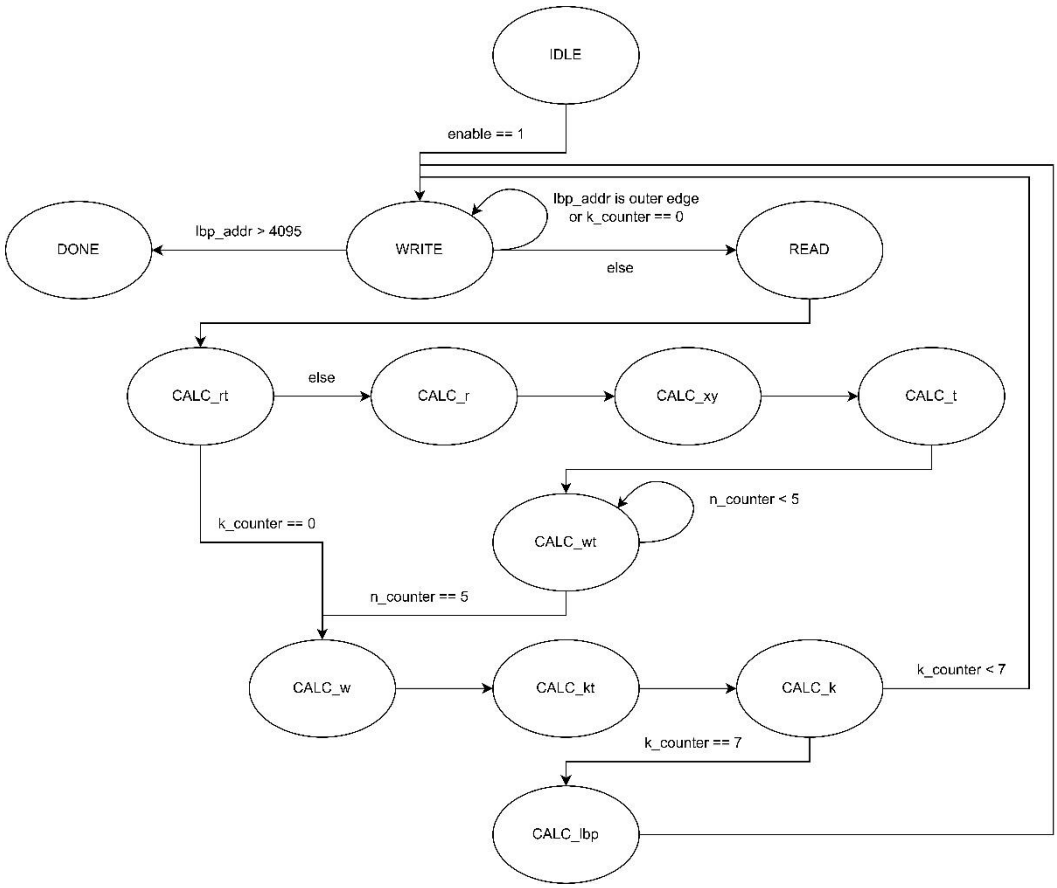
- Understanding the function:

  Once system is initialized, it

  a) Choose a pixel(center) in the image and select its neighboring pixels.
  b) Calculate bilinear interpolation:
     1) Determine r_x & r_y.
     2) Determine x_1, x_2, y_1, y_2.
     3) Determine t_x, t_y.
     4) Determine w1,w2, w3,w4.
     5) Determine $f(0,0)$, $f(0,1)$, $f(1,0)$, $f(1,1)$.
     6) Determine neighbor.
  c) Repeat b) to calculate all neighbors' values.
  d) Construct the mask using threshold function.
  e) Combine the binary values for all neighboring pixels to obtain a binary code for the central pixel and convert it to a decimal value.
  f) Repeat steps a)–e) for each pixel in the image to obtain a binary code for each pixel.

- Draw your state diagram and explain your design. You can draw internal architecture to describe your design.



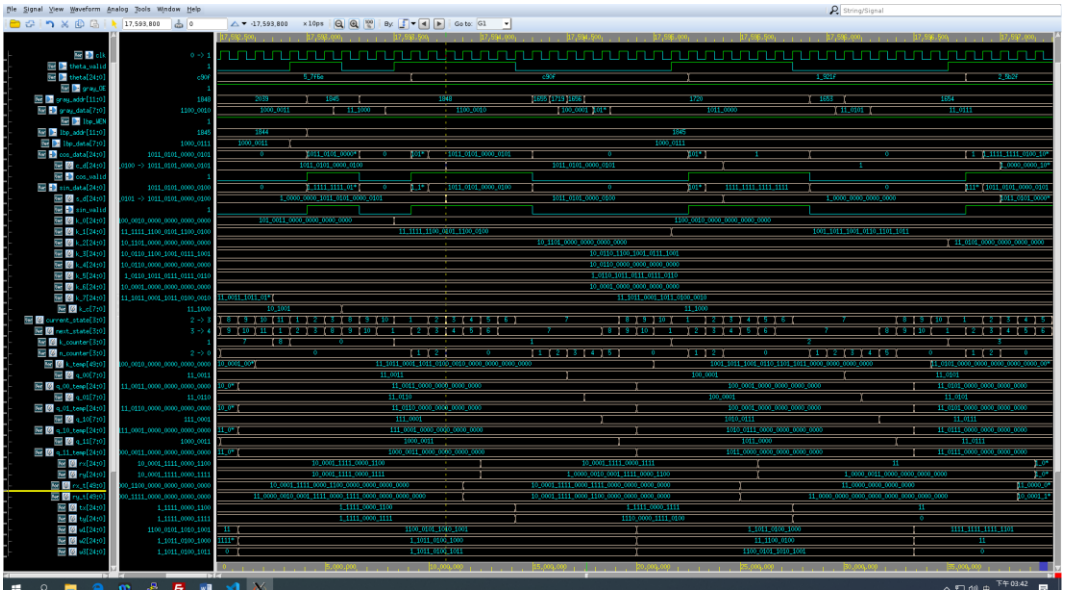


Because this program has more calculation parts, to accurately control the input and output of several jobs, I designed it into many states. The state transition is as shown in the figure, mostly controlled by counter.

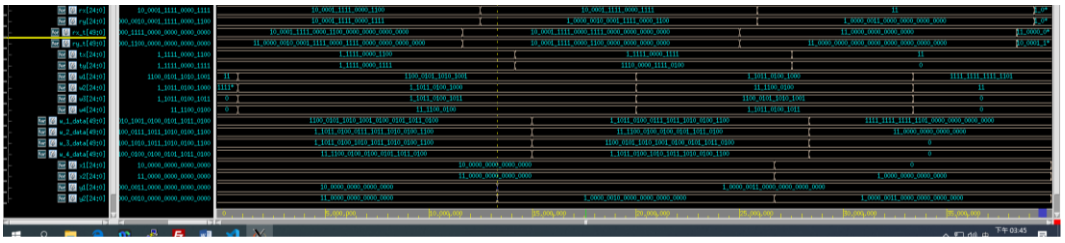
1. Complete the CLBP module.
2. Compile the verilog code to verify the operations of this module works properly.
3. Synthesize your *CLBP.v* with following constraint:
   - Clock period: no more than 2.0 ns.
   - Don't touch network: clk.
   - Wire load model: N16ADFP_StdCellss0p72vm40c.
   - Synthesized verilog file: C*LBP_syn.v*.
   - Timing constraint file: C*LBP_syn.sdf*.
4. Please **attach your waveforms** and **specify your operations** on the waveforms.
   Take 1845 as an example:

Read gray_data to k_c

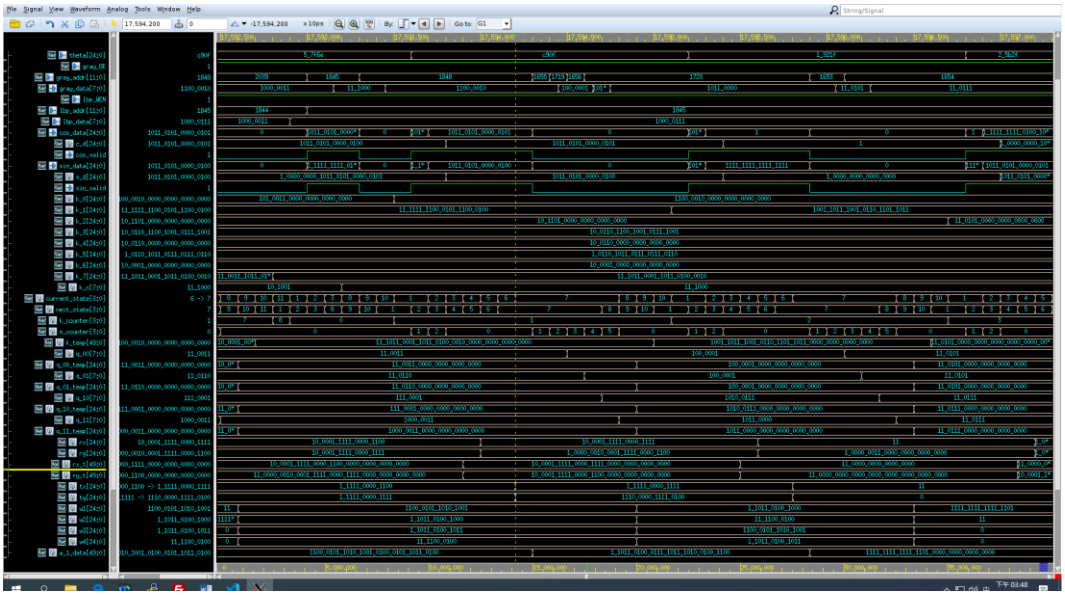
Calculate the theta value and read it into cos_data and sin_data. There is no need to process it here. Read c_d and s_d directly and calculate rx and ry.



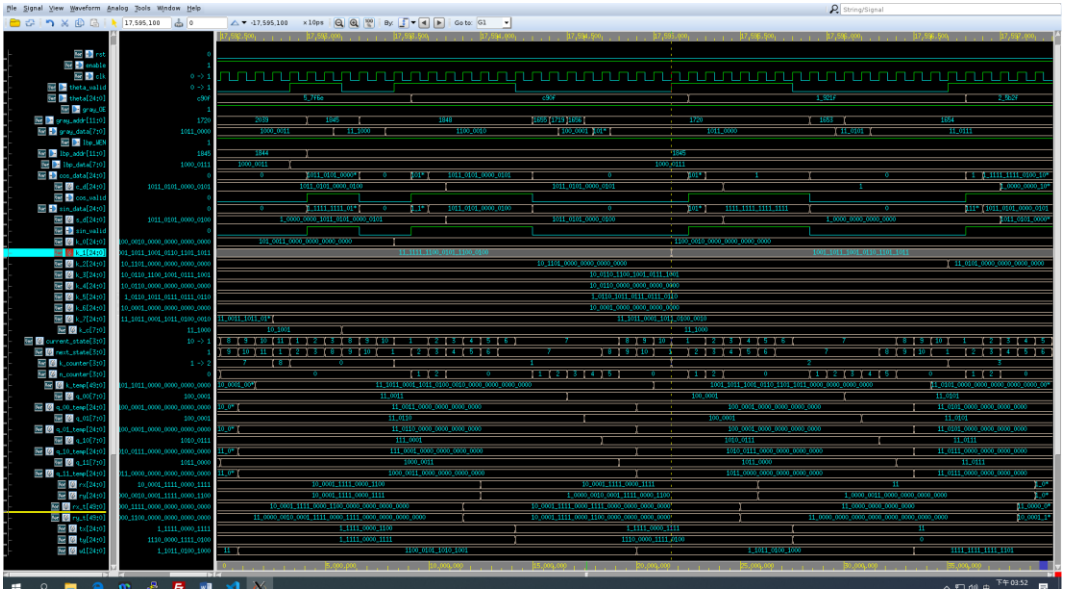
Calculate x1,x2,y1,y2



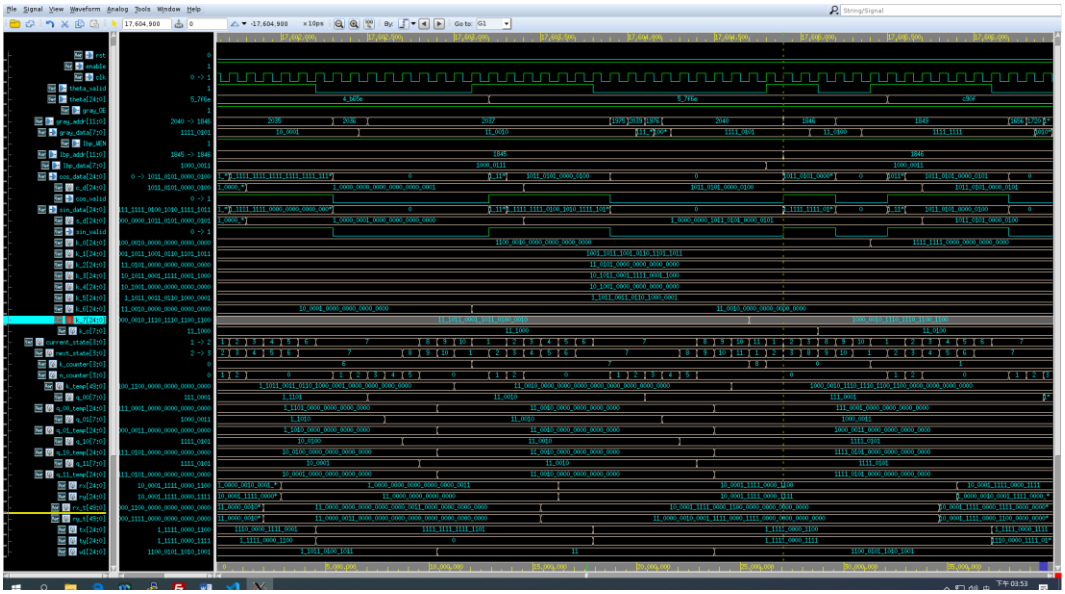
Calculate tx, ty and calculate gray_addr. q_00, q_01, q_10, q_11 receive the transmitted gray_data and calculate w1, w2, w3, w4.



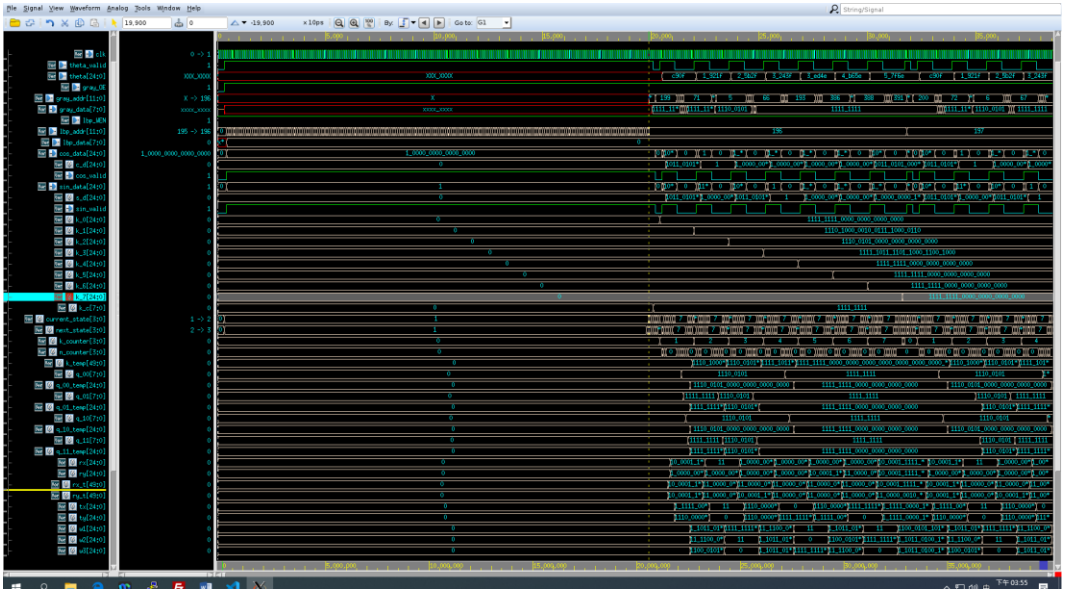
Pass the calculated value into k_1

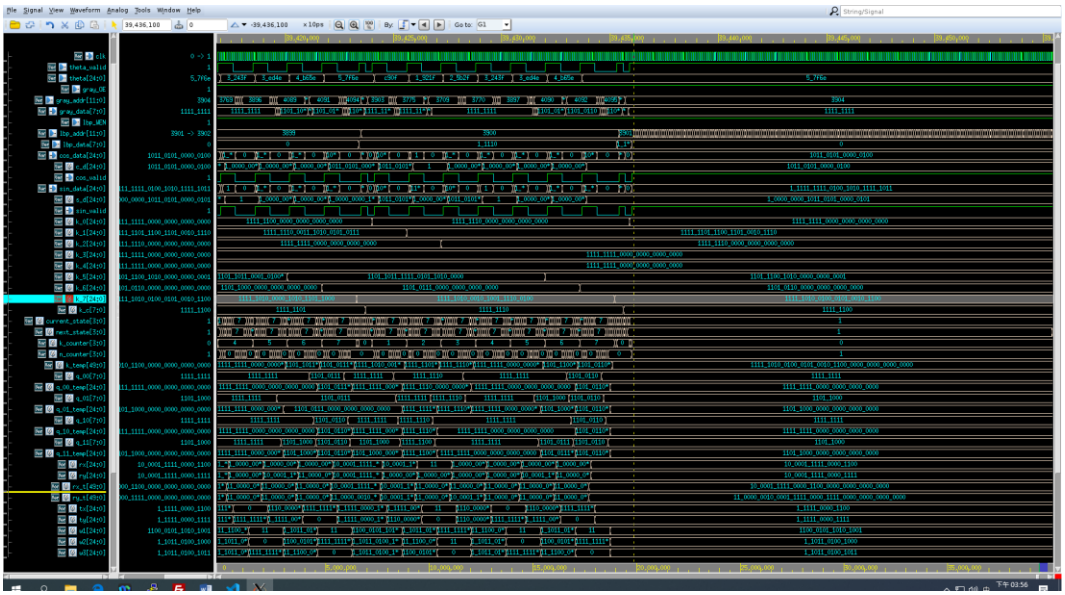


Repeat the above until k_7 is calculated, and calculate the lbp_data value

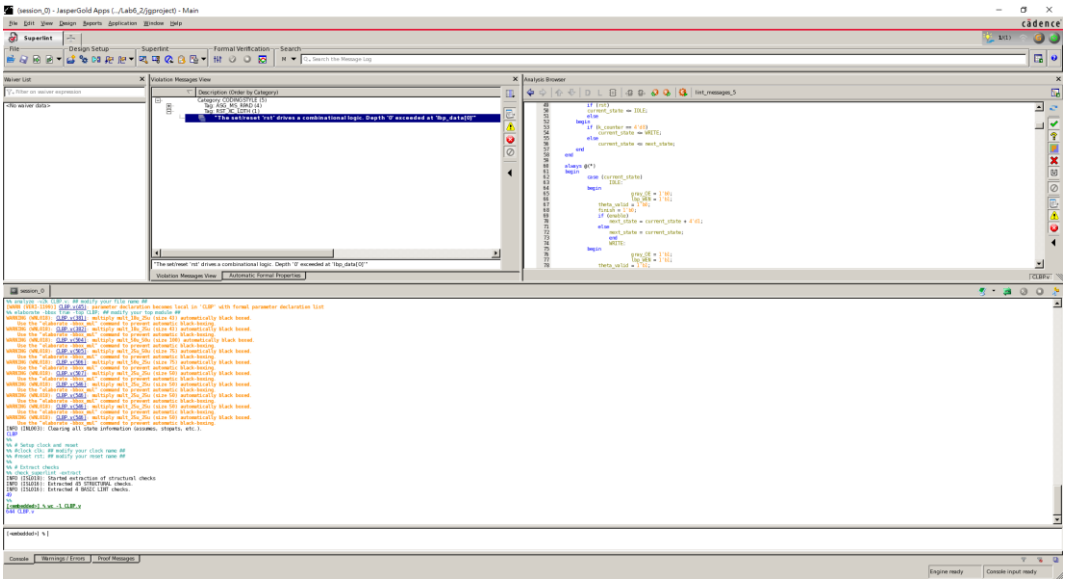


If it is the outer ring range, fill in 0 directly.





5. Show SuperLint coverage (include all files)

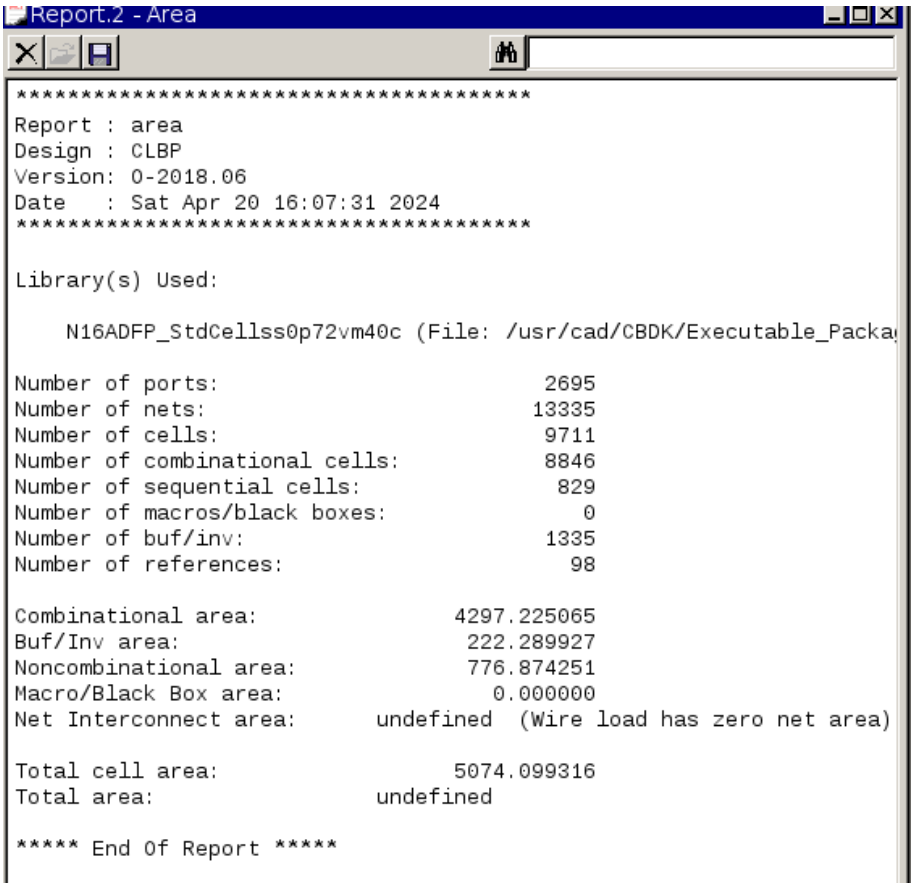


6. Your clock period, total cell area, post simulation time with screenshot

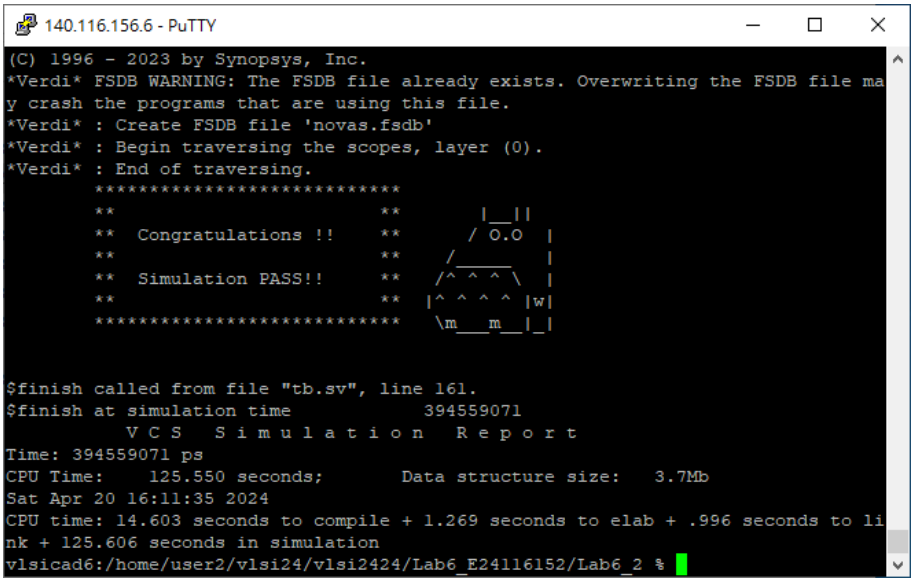
Clock period: 0.39

```
mult_504/U151/CO (FA1D1BWP16P90LVT)                    0.03       1.44 f
mult_504/U150/CO (FA1D1BWP16P90LVT)                    0.03       1.46 f
mult_504/U149/CO (FA1D1BWP16P90LVT)                    0.03       1.49 f
mult_504/U148/CO (FA1D1BWP16P90LVT)                    0.03       1.52 f
mult_504/U147/CO (FA1D1BWP16P90LVT)                    0.03       1.54 f
mult_504/U146/CO (FA1D1BWP16P90LVT)                    0.03       1.57 f
mult_504/U145/CO (FA1D1BWP16P90LVT)                    0.03       1.60 f
mult_504/U144/CO (FA1D1BWP16P90LVT)                    0.03       1.62 f
mult_504/U143/CO (FA1D1BWP16P90LVT)                    0.03       1.65 f
mult_504/U142/CO (FA1D1BWP16P90LVT)                    0.03       1.68 f
mult_504/U141/CO (FA1D1BWP16P90LVT)                    0.02       1.70 f
mult_504/U1341/Z (XOR4D1BWP16P90)                      0.05       1.76 r
mult_504/product[40] (CLBP_DW_mult_uns_10)             0.00       1.76 r
U228/Z (AO22D1BWP16P90LVT)                             0.02       1.77 r
w_1_data_reg[40]/D (DFQD2BWP16P90LVT)                  0.00       1.77 r
data arrival time                                                 1.77

clock clk (rise edge)                                  2.00       2.00
clock network delay (ideal)                            0.20       2.20
clock uncertainty                                     -0.02       2.18
w_1_data_reg[40]/CP (DFQD2BWP16P90LVT)                 0.00       2.18 r
library setup time                                    -0.01       2.17
data required time                                                2.17
-----------------------------------------------------------------------
data required time                                                2.17
data arrival time                                                -1.77
-----------------------------------------------------------------------
slack (MET)                                                       0.39


***** End Of Report *****
```

Total cell area: 5074.099316



Post simulation time: 394559071ps



7. Lessons learned from this lab

   The difficulty of this experiment was very high. I spent a lot of time researching how to complete it. However, after I passed the RTL simulation, problems occurred in the synthesis. After the synthesis was completed, other problems appeared. This experiment can be said to be It's very troublesome, but it also makes me more familiar with the basic concepts and syntax of verilog.

Please compress all the following files into one compressed file (".tar " format) and submit through Moodle website:

※ NOTE:

1. If there are other files used in your design, please attach the files too and make sure they're properly included.
2. Simulation command

Problem	Command
Lab6_1(pre)	vcs -R -full64 -sverilog tb.sv +access+r +vcs+fsdbon
Lab6_1(post)	vcs -R -full64 -sverilog tb.sv +access+r +vcs+fsdbon +define+SDF+SYN
Lab6_2(pre)	vcs -R -full64 -sverilog tb.sv +access+r +vcs+fsdbon
Lab6_2(post)	vcs -R -full64 -sverilog tb.sv +access+r +vcs+fsdbon +define+SDF+SYN