2024 IC Design Contest

Cell-Based IC Design Category for Graduate Level

Bicubic Resize Engine

1.Problem description

Please complete the circuit design of ***the Bicubic Resize Engine (hereinafter referred to as* Bicubic)** function. This circuit can convert the selected 2 The dimensional matrix data is enlarged to the required size, which is often used to enlarge the selected area of the image.

Detailed specifications of Bicubic will be described later. Table 1 shows the function description of each input and output signal of this circuit. Each participant Teams must complete design verification in accordance with the design specifications specified in the next section and the test samples in Appendix C.

The competition time of this IC design competition is from 08:30 am to 20:30 pm. When the IC design competition ends, this question will be scored according to the scoring criteria in Section 3. For the convenience of grading assignments, each participating team should refer to Appendix C and Appendix D Requirements listed in , attach the files needed for scoring.

2.Design specifications

2.1 System block diagram

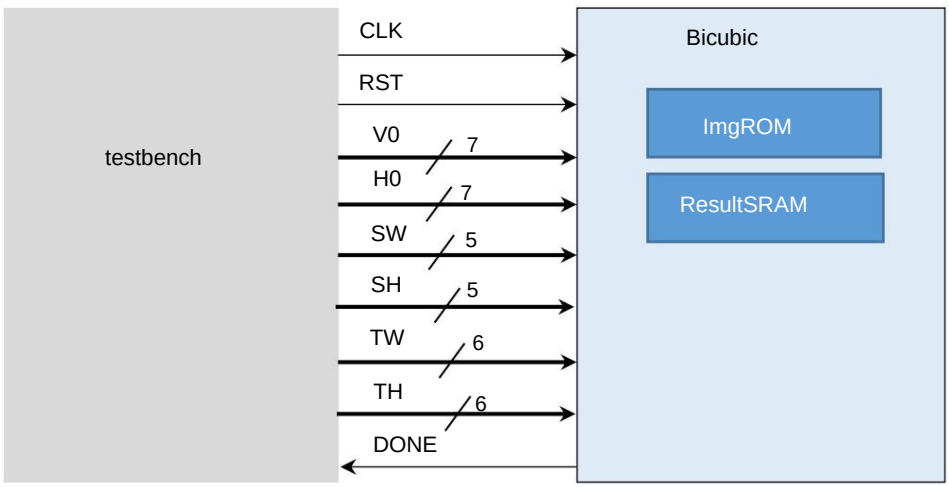


Figure 1. System block diagram

2.2 Input/output interface

Table 1 - Input/Output Signals

Signal Name	I/O	Width	Simple Description
CLK	I	1	System working clock, this system is designed to be synchronized with the positive edge of the clock.
RST	I	1	System reset signal reset signal (active high). Provided by testbench, Pull high for 2 cycles and then return to low.
H0	I	7	The H coordinate value of the upper left corner coordinate of the area to be processed by the Bicubic circuit is introduced. From 0 to 99.
V0	I	7	The V coordinate value of the upper left corner coordinate of the area to be processed by the Bicubic circuit is introduced. From 0 to 99.
SW	I	5	The horizontal width of the area to be processed by the Bicubic circuit. This value is added to H0 It will not exceed the scope of the original image. H0+SW< 99
SH	I	5	The vertical height of the area to be processed by the Bicubic circuit. This value plus V0 It will not exceed the scope of the original image. V0+SH < 99
TW	I	6	Bicubic circuit processing area amplified horizontal width, TW will not More than 2 times SW size. SW< TW < 2*SW
TH	I	6	Bicubic circuit processing area amplified vertical height, TH will not More than 2 times SH size. SH< TH < 2*SH
DONE	O	1	When the Bicubic circuit completes the required work, the signal is pulled high for 1 cycle. Please testbench to start scoring.

2.3 System description

The purpose of this Bicubic circuit design is to place the specified area from the original image (upper left corner (H0, V0), size SW x SH).

Dacheng TW x TH size image.

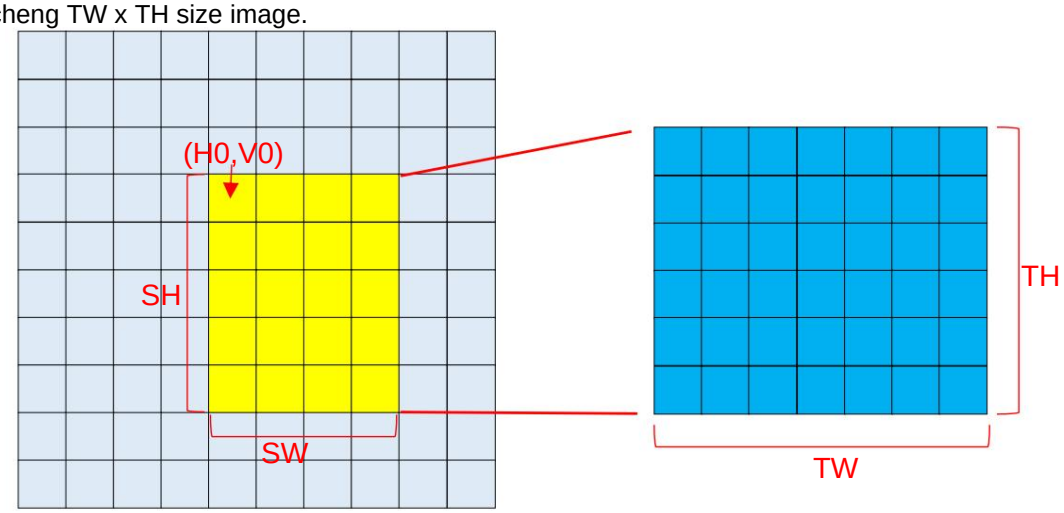


Figure 2. Schematic diagram of the working principle of Bicubic Resize Engine

The original image data is an image of fixed size 100x100, and the image data is stored in ImgROM. Since the system reset,

Obtain information such as the coordinates (H0, V0) of the upper left corner of the area to be enlarged and the size SW x SH from the input port. Bicubic Electronics

The method enlarges the range image to the required TW x TH size and writes the result content into ResultSRAM. Writing completed

Finally, pull the DONE signal to High for one cycle, and the testbench platform will directly read the ResultSRAM content for scoring.

2.3.1 Obtaining image data

The original data to be processed by the Bicubic circuit is an image of 100x100 size, and each pixel data is 8bits.

numbers and stored in ImgROM in one dimension . The storage method is stored sequentially from the first horizontal column (corresponding to the V coordinate of 0).

Save to the 100th horizontal column (corresponding to V coordinate 99).

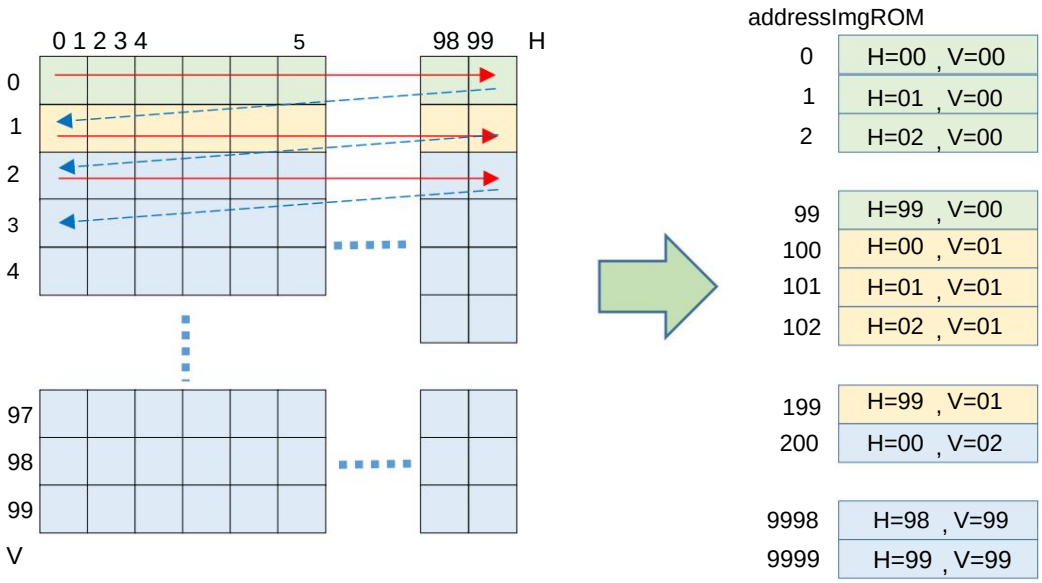


Figure 3. How 2D original data is stored in ImgROM

2.3.2 Determining interpolation endpoints

Consider the case of enlarging a 1x7 image P to a 1x10 image Q in 1D (one-dimensional space). The goal is to find the Q map

All values. Imagine that when the Q picture is compressed to the same length as the P picture, the front end points of the two pictures overlap and the last end points also overlap.

Therefore Q(0)=P(0), Q(9)=P(9).

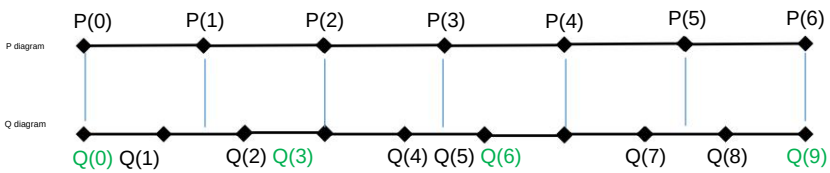


Figure 4. Finding interpolation endpoints

Some intermediate points may coincide with each other, and the values can be directly applied. In this case, Q(3)=P(2), Q(6)=P(4).

The remaining non-overlapping points cannot directly obtain the value. The closest endpoint must be found from the P chart, and then the value of the point can be found by interpolation.

In this case, the interpolation endpoints of Q(1) are P(0) and P(1), the interpolation endpoints of Q(2) are P(1) and P(2), and the interpolation endpoints of Q(4)

are P(2) and P(3), and so on.

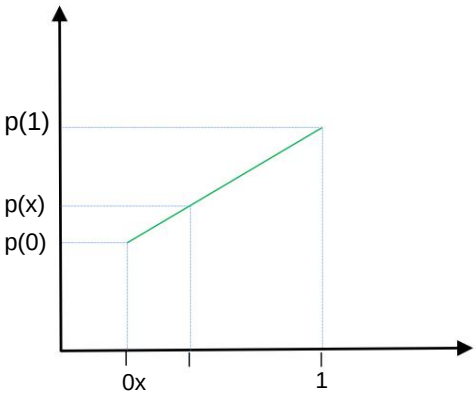
2.3.3 One-dimensional interpolation

First, we will introduce linear interpolation and cubic interpolation in one dimension to facilitate subsequent extension to two-dimensional Bicubic.

Description of interpolation.

2.3.3.1 **linear interpolation**

Given the two values of p(0) and p(1), find the point p(x) between the two points, 0ÿ x ÿ1, the linear interpolation formula of p(x) is:



$$p(x) = p(0) + \frac{x\ddot{y}0}{1\text{-}0}(p(1)\ddot{y}\,p(0))$$

$$= p(0) + x(p(1)\ddot{y}\,p(0))$$ Since the horizontal direction difference between p(0) and p(1) is 1, x just represents the position ratio between 0 and 1

Figure 5. Linear interpolation

Taking Figure 4 as an example, Q(2) falls 1/3 between P(1) and P(2), we can get $Q(2) = P(1) + \frac{1}{3}(\ (2)\ddot{y}\ (1))$

2.3.3.2 **cubic interpolation** (cubic interpolation)

Because linear interpolation only considers the linearity between two points, using linear interpolation to enlarge the curve is prone to non-smoothness (green line in Figure 6).

This problem can be improved by cubic interpolation (red line in Figure 6).

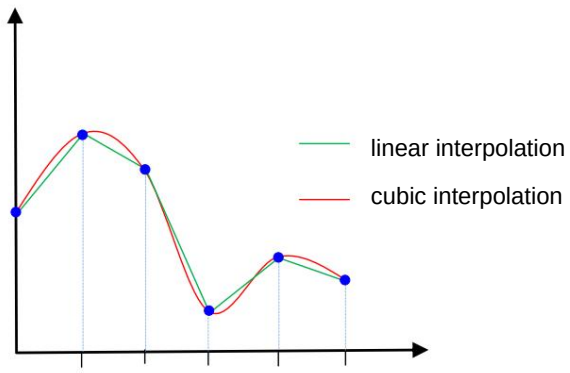
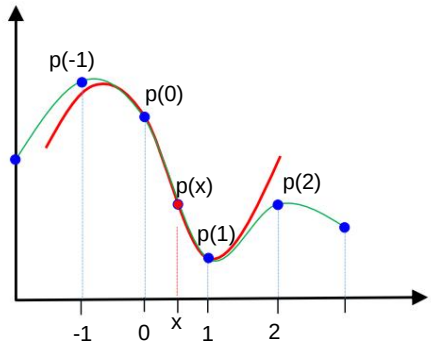


Figure 6. Linear interpolation versus cubic interpolation

See Figure 7. Assume that the values of p (0) and p(1) are known, as well as the values of p(-1) and p(2). p(-1) refers to the point before p(0), p(2)

Refers to the point after p(1), find the value of point p(x) between p(0) and p(1). The method

of cubic interpolation is to assume that the curve between the two points p(0) to p(1) is a third-order polynomial, as shown in the red curve in Figure 7



Assume that p(x) = ax3+bx2+cx+d , 0ÿxÿ1

p(0) and p(1) are known values,

Equation therefore----

1 p(0) = dp(1) = a+b+c+ d ----Formula 2

Figure 7. Cubic interpolation

This question defines the slope of the p(0) position as $\frac{P(1)-P(-1)}{2}$, and the slope of the p(1) position as $\frac{P(2)-P(0)}{2}$

p'(x) = 3ax2+2bx+c, therefore

$$p'(0) = c = \frac{p(1)-p(-1)}{2} \quad \text{----Formula 3}$$

$$p'(1) = 3a+2b+c = \frac{p(2)-p(0)}{2} \quad \text{----Formula 4}$$

Solving the simultaneous equations from Equation 1 to Equation 4, we can get

$$a = -\frac{1}{2} p(-1) + \frac{3}{2} p(0) - \frac{3}{2} p(1) + \frac{1}{2} p(2) \quad (2)$$

$$5\ b = p(-1) - \frac{5}{2} p(0) + 2\, p(1) - \frac{1}{2} p(2) \quad (2)$$

$$= -\frac{1}{2} p(-1) + \frac{1}{2} p(1) \quad (1)$$

$$= p(0) \quad (0)$$

Substitute a, b, c, and d back into the original equation to obtain the value of p(x).

2.3.4 Two-dimensional **Bicubic interpolation** (bicubic interpolation)

The one-dimensional cubic interpolation uses the values of 4 points to obtain the interpolated value, while the two-dimensional cubic interpolation uses 16 points to obtain the image interpolated value. For example, in the figure below, to obtain the interpolation value between the four points of abcd, a total of 16 surrounding points are needed; the method is to calculate the one-dimensional cubic interpolation results of each of the 4 horizontal rows (red lines), and then use these 4 results to calculate the straight row (One-dimensional cubic interpolation of green line).

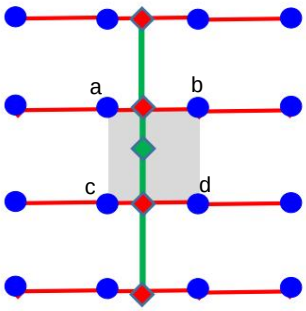
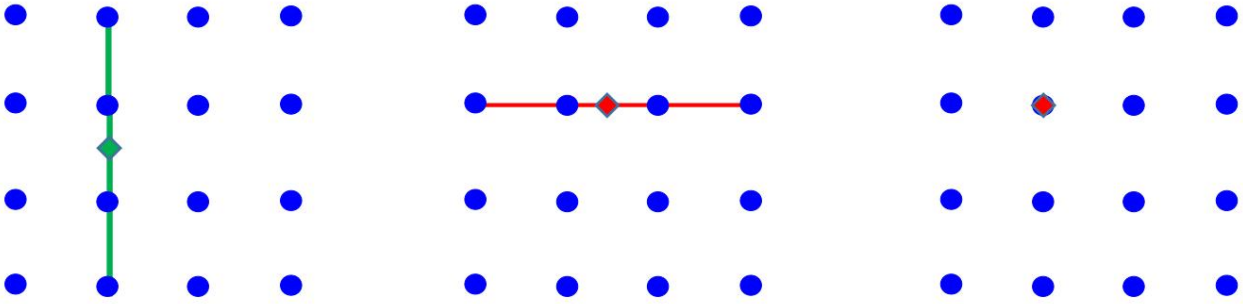


Figure 8. Two-dimensional bicubic interpolation

If the vertical or horizontal arrangement coincides with the original image, the original image value will be used directly.



The vertical V direction coincides with the original image, and the horizontal H direction coincides with the original image.

Both horizontal and vertical directions coincide

Figure 9, horizontal or vertical overlap

ÿNotesÿ

1. The enlarged area specified in this question will not be at the extreme boundary of the original image, so there is no need to deal with boundary issues.

2. Please round off the result of each Cubic operation to an integer (including 4 horizontal interpolations and 1 vertical interpolation).

3. The Cubic interpolation result may exceed the 8-bit value range. If the interpolation result is greater than 255, please set the result to 255;
   If less than 0, set the result to 0 (including 4 horizontal interpolations and 1 vertical interpolation).

4. If you do vertical interpolation first and then horizontal interpolation, you can also get the result, but the result may be slightly different due to the approximate number.
   testbench has been designed to accept both sequential approaches.

2.3.5 Results are stored in **ResultSRAM**

The result of Bicubic processing is an image of TW x TH size. Each pixel data is an 8bits positive integer. The result Need to be stored in one-dimensional ResultSRAM. Starting from the first horizontal column (corresponding to V coordinate 0), store them in sequence ResultSRAM, ResultSRAM size is 16384x8, testbench will only judge the correctness of data from addresses 0 to (TW x TH -1) in ResultSRAM, so the unused space in ResultSRAM does not need to be processed.

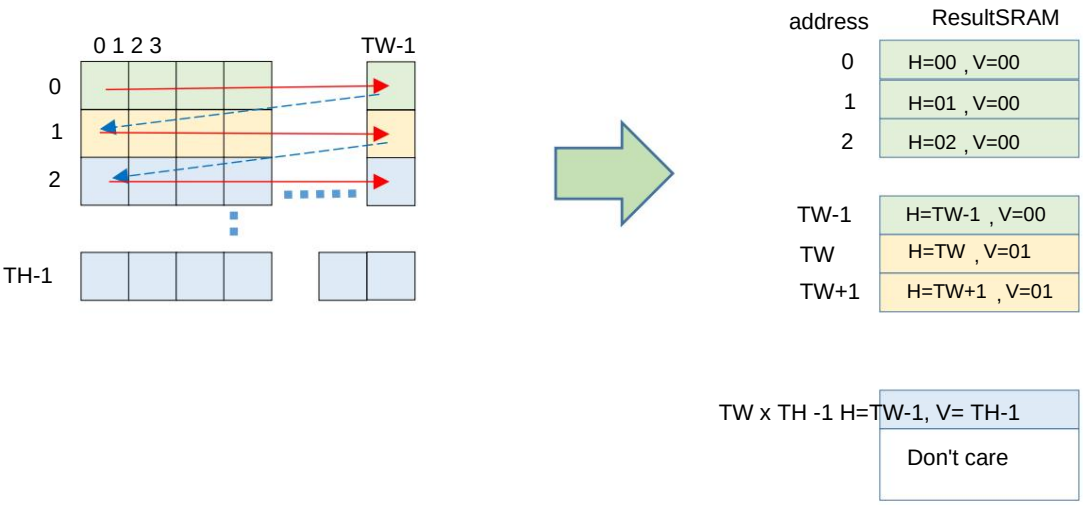


Figure 10. How images are stored in ResultSRAM

2.4 Bicubic circuit timing specifications

2.4.1 Timing diagram of **Bicubic** circuit

The corresponding timing waveform diagram of the output is as shown in the figure, and please read it according to the instructions below.

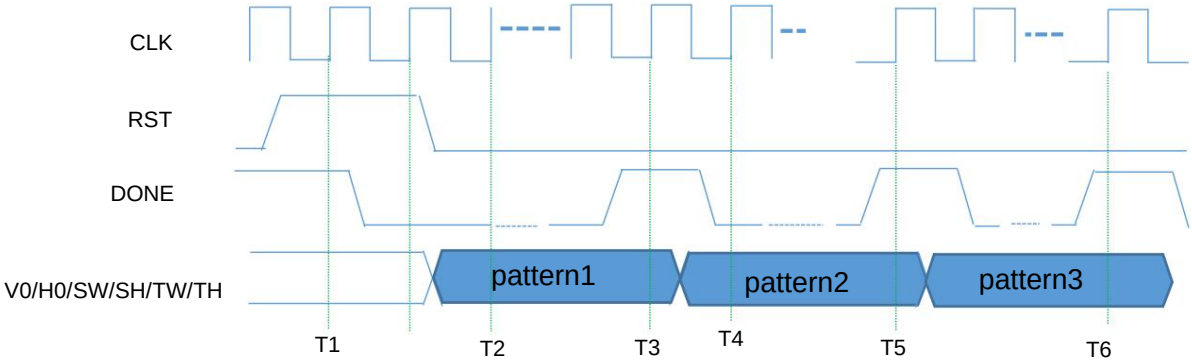


Figure 11. Bicubic circuit timing diagram

1. At T1 time point, starting from 2 cycles of RST, Bicubic must pull DONE to low.
2. At T2 time point, start inputting V0, H0, SW, SH, TW, TH and other data, and continue until the DONE signal is pulled high.
3. At time T3, Bicubic completes the operation, stores the result in ResultSRAM, and pulls the DONE signal high. At this time testbench starts to compare the correct data in ResultSRAM and sends the second pattern.
4. At T4 time point, Bicubic must pull DONE back to low.
5. At time T5, Bicubic completes the second pattern, pulls the DONE signal high and pulls it low again in the next cycle.
6. At time T6, Bicubic completes the last pattern operation, the result is also stored in ResultSRAM, and DONE The signal is pulled high.
7. When testbench receives the DONE signal of the last pattern, the simulation ends

testbench presets that a single pattern simulation will take up to 50,000 cycles, and $finish will be issued at 50,000 cycles to force the simulation to stop. If you need more cycles for simulation, please modify the MAX_CYCLE_PER_PATTERN value of tb.v yourself, and fill in the used MAX_CYCLE_PER_PATTERN value in the scoring table (Appendix C, report.txt).

2.4.2 Timing diagram of **ROM and SRAM**

ImgROM (Address 14bit body , Data 8bits), ResultSRAM (Address 14bit , Data 8bits), memory provided below read and write timing diagram, please refer to the memory files ImgROM.pdf and ResultSRAM.pdf for details.

Since the input signal of the memory may not have been initialized during the Reset period, timing violation may occur during simulation. This timing violation that occurs during Reset can be ignored.

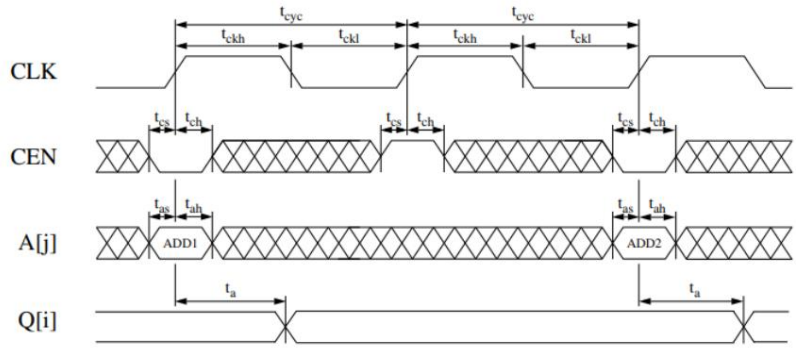


Figure 12. Synchronous ROM single read-cycle timing

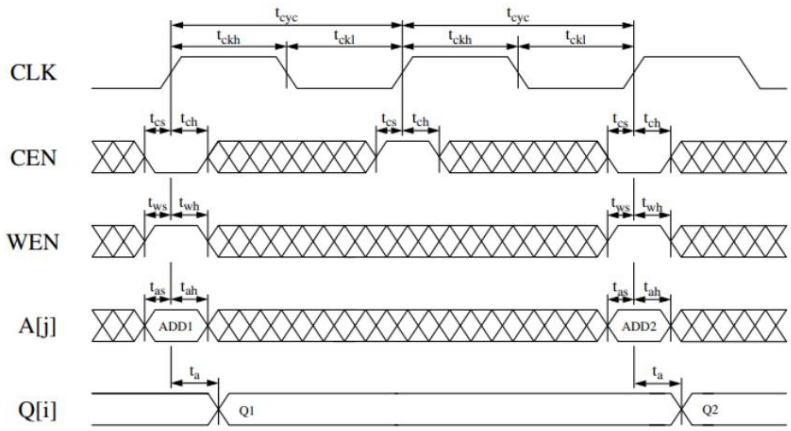


Figure 13. Synchronous single-port SRAM read-cycle timing

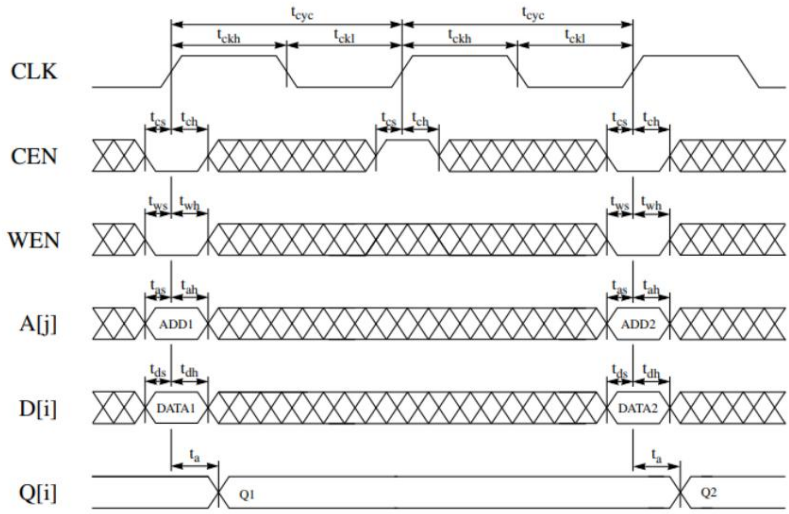


Figure 14. Synchronous single-port SRAM write-cycle timing

3. Scoring criteria

This question uses the same original image to conduct three sets of tests with different parameters (referring to the processing area and target size). Participants must Bicubic circuit functions must be completed and the results are completely correct.

The scoring method for this question is to multiply the total power consumption by the area to score. You need to use PrimeTimePX to obtain the simulated average power. The total simulation time is added to obtain the simulated power consumption, and then multiplied by the area as the basis for scoring.

The CBDK_IC_Contest_v2.5 version and slow case library are used for scoring this question . Please do not use the wrong cell library. Or a corner to avoid being unable to score.

Participants can adjust the CYCLE_TIME value in tb.sv and provide it in the Report File (see Appendix C) for correct simulation. The cycle time (CYCLE TIME) is given to the scorer to verify the correctness of the circuit.

```
`timescale 1ns/10ps
`define CYCLE_TIME      8.0
`define SDFFILE     "./Bicubic_syn.sdf"
```

ImgROM and ResultSRAM are part of the Bicubic circuit, and the power consumption of these two IPs will also be included in the power consumption calculation.

ÿ Scoring item one: Simulation time (Time) Use

testbench to simulate. The simulation time will be displayed after the simulation is completed.

```
*********************************
**   Finish Simulation
**   Error pixels:     0
**   Simulation time:  333333.00 ns
*********************************
```

ÿ Scoring item 2: Area (Area)

Take Design compile report area example: dc_shell> report_area

```
 Net Interconnect area:          411111.111111

 Total cell area:                 33333.333333
 Total area:                     444444.444444
```

ÿ Scoring item three: Power consumption (Power)

The PrimeTimePX automated execution file has been provided for this question. Participants only need to prepare the relevant files (i.e. Bicubic_syn.v, Bicubic.sdc, Bicubic.vcd, .synopsys_pt.setup) used by the pt_script.tcl file and execute the following instructions. Power analysis.

unix% pt_shell -f ./pt_script.tcl

```
Total Power             = 4.421e-03  (100.00%)

X Transition Power      = 3.207e-08
Glitching Power         = 2.946e-05

Peak Power              =    0.0268
Peak Time               = 42124.986
```

The PrimeTimePX version used when scoring is 2022.12. If the contestant uses a different version, it may not be executed smoothly. Primetime power analysis and scoring will use this version to obtain the Power value.

The scoring method is divided into four levels: A, B, C, and D according to the degree of design completion. The ranking order is A>B>C>D.

There are four levels of design completion. In each level, the smaller the Score , the better the ranking. ÿLevel A:

- Under the user-specified cycle time (CYCLE TIME), the Gate--Level and RTL simulation results are correct. Scoring method for Level A: Score = Power x Time x Area

ÿLevel B:

- RTL simulation results are correct.
- Synthesis completed, Gate-Level can be simulated but has some errors.

Scoring method for level B: Score = Gate-Level simulation error points

ÿLevel C:

- RTL simulation results are correct. -

The synthesis has not been completed or the Gate-Level cannot simulate Level C. The scoring method: Score = RTL simulation cycle

ÿLevel D:

- RTL simulation has some errors.

Scoring method of level D: Score = RTL simulation error points

appendix

Appendix A is a description of the design files of each contestant provided by the organizer.

Appendix B is a description of the test samples provided by the organizer. Appendix C is the scoring file, that is, the file information that the contestants must submit.

Appendix D is the steps for compressing and organizing the design files. illustrate

Appendix A. Design files

1. The following table is the design file of each contestant provided by the

organizer 2. Design file description

File name	illustrate
Bicubic.v	The design file template for this question already contains the declaration of the system Input/Output Port. Please use this file as the Bicubic circuit design.
tb.sv	testbench file, contestants can adjust the CYCLE_TIME value themselves
Bicubic.sdc	In the synthesized constraint file, contestants can adjust the cycle_time value by themselves.
ImgROM.rcf	Image file data
pattern1 pattern2 pattern3	The output golden pattern during Bicubic circuit simulation.
dc_syn.tcl	dc synthesis reference command
.synopsys_dc.setup .synopsys_pt.setup synopsys_dc.setup	Design Compiler/ Primetime setup initialization sample file. Contestants please follow The actual placement of the Library, modify the Search Path settings yourself. Note: Please use the worst case library for synthesis.
pt_script.tcl	primetime analysis power script
xrun.cmd	xrun simulation reference instructions
vcs.cmd	vcs simulation reference instructions
vsim.cmd	vsim simulation reference instructions

2. Please use Bicubic.v to design the circuit for this question. Its Verilog module name and output/input port declaration are as follows:

If necessary, participants can remove the reg declaration of the output by themselves.

ImgROM and ResultSRAM have been added to Bicubic.v. Participants must connect the signals by themselves, but please do not change their component names (u_ImgROM, u_ResultSRAM) and their levels to avoid testbench being unable to read the results.

```
module Bicubic ( input
CLK, input RST,
input [6:0] V0,
input [6:0] H0, input [4:0]
SW, input [4:0] SH, input
[5:0] TW, input [5:0] TH,
output reg DONE);



ImgROM u_ImgROM (.Q( ), .CLK( ), .CEN( ), .A( )); ResultSRAM u_ResultSRAM
(.Q( ), .CLK( ), .CEN( ), .WEN( ), .A( ), .D( ));


endmodule
```

3. The testbench file provided in this question has several lines of define added as follows:

```
`define SDFFILE "./Bicubic_syn.sdf"
`ifdef SDF
    initial $sdf_annotate(`SDFFILE , u_Bicubic);
`endif
```

3.1 Please modify the SDF file name according to the actual SDF file name and path before simulating. 3.2 When performing gate-level simulation, contestants need to add an additional +define+SDF parameter to the simulation command to simulate smoothly. The example is as follows

```
xrun tb.sv Bicubic_syn.v –v ImgROM.v –v ResultSRAM.v -v tsmc13_neg.v
+define+P1 +define+SDF
```

4. The organizer provides three sets of test samples for participants to verify the correctness of the design. Please use the +define+P1, +define+P2, +define+P3 parameters to switch by yourself. If definitions such as +define+P1 are not used, all test samples will be simulated.

5. After the testbench simulation is completed, the source image and result image content will be displayed (such as Appendix B). The blue numbers are the relative coordinates within the block, and the points with wrong results will be marked in red.

```
---- orgin ( 10 ,  68), size  13 x  13
    0  1  2  3  4  5  6  7  8  9 10 11 12
 0 3f 4c 4c 4f 4f 4f 3a 36 4f 4e 4c 4b 4a
 1 4c 0f 1c 4e 4f 4f 38 35 4e 4d 4d 4b 4a
 2 4a 3a 49 4a 4e 4e 38 35 4d 4c 4c 4b 4a
 3 49 4a 0c 02 45 46 32 2f 45 44 43 43 42
```

If your simulation environment cannot display colors, causing layout confusion, please remove the USECOLOR definition in tb.sv. as follows

```
`define MAX_CYCLE_PER_PATTERN 50000
`define PLOT_IMG
//`define USECOLOR
//`define P1
```

6. The SRAM and ROM verilog models have timing check. You can add the notimingcheck parameter during RTL simulation to ignore the setup and hold time checks. Please remember to correct the hold time during synthesis to prevent gate-level from being unable to simulate correctly . 7. Do not target these three sets of tests. Design the content of the sample, such as judging the nth pattern and directly setting certain results, etc.

If found, no points will be awarded. Other test samples will be added when scoring to filter out tricky designs.

8. Please do not plagiarize or provide design results from people who are not in the same group. The scorers will cross-compare all design results that belong to the same question. If If you have the same design, you will definitely find it.

9. The instructions for RTL and Gate-level simulation are as follows

Instructions for using RTL simulation of P1 test sample: ÿ Use xrun simulation command (xrun.cmd)

```
xrun tb.sv Bicubic.v -v ImgROM.v -v ResultSRAM.v +define+P1 +access+r -
clean
```

ÿ Use vcs simulation command (vcs.cmd)

```
vcs -R -sverilog tb.sv Bicubic.v -v ImgROM.v -v ResultSRAM.v +define+P1
+access+r +vcs+fsdbon +fsdb+mda +fsdbfile+Bicubic.fsdb
```

ÿ To use modelsim simulation, please refer to the instructions in vsim.cmd :

```
vsim -c -do vsim.cmd
```

vsim.cmd content:

```
vlib work
vlog tb.sv Bicubic.v -v ImgROM.v -v ResultSRAM.v +define+P1
vsimwork.tb
run-All
```

Instructions for Gate-level simulation using P1 test sample: ÿ Use xrun simulation command (xrun.cmd)

```
xrun -sv tb.sv Bicubic_syn.v -v ImgROM.v -v ResultSRAM.v -v tsmc13_neg.v
+define+P1+SDF +access+r -clean -maxdelays
```

ÿ Use vcs simulation command (vcs.cmd)

```
vcs -R -sverilog tb.sv Bicubic_syn.v -v ImgROM.v -v ResultSRAM.v -v
tsmc13_neg.v +define+P1+SDF +access+r +vcs+fsdbon +fsdb+mda
+fsdbfile+Bicubic.fsdb +maxdelays +neg_tchk
```

ÿ To use modelsim simulation, please refer to the commands in vsim.cmd :

(Please annotate the RTL sim command and uncomment the Gate-level sim command in vsim.cmd)

```
vsim -c -do vsim.cmd
```

vsim.cmd content:

```
vlib work
vlog tb.sv Bicubic_syn.v -v ImgROM.v -v ResultSRAM.v -v tsmc13_neg.v
     +define+SDF
vsimwork.tb
run-All
```

10. Use dc_shell to perform synthesis. The instruction instructions are as follows:

```
dc_shell –f dc_syn.tcl
```

When the synthesis is completed, the area report (area.log) and load information (Bicubic_syn.spef) will be written out. 11. Use pt_shell to perform power analysis instructions as follows: pt_shell -f ./pt_script.tcl

Appendix B. Test sample

The following provides 3 sets of test sample content. The source data refers to the range selected from the original image, and the gray part indicates the selection.

For points in the outer circle of the range, the pixel value is expressed in hexadecimal. The target data represents the amplified result.

pattern1: (H0 V0) = (10 68), (SW SH) = (13 13), (TW TH) = (19 19)

source

4d 6 17 4f 50 51 50 3a 37 4f 4e 4c 4b 49 48
4c 3f 4c 4c 4f 4f 4f 3a 36 4f 4e 4c 4b 4a 48
4b 4c f 1c 4e 4f 4f 38 35 4e 4d 4d 4b 4a 48 49 4a 3a 49
4a 4e 4e 38 35 4d 4c 4c 4b 4a 49
48 49 4a c 2 45 46 32 2f 45 44 43 43 42 41
3e 48 4a 3b 76 6 22 19 19 22 20 20 20 1f 1c
12 46 49 49 1c 22 4b 36 34 4c 4c 4b 4b 4a 49 ad 2a 46
49 4a 26 16 35 32 4c 4c 4b 4b 4a 49 bb 46 44 46 49 40
50 16 32 4b 4b 4b 4b 4a 49
90bb 6 46 46 48 2c 5c 27 4b 4a 4a 4b 4a 4a
40 ab 99 2a 46 46 48 13 3e 48 4a 4a 4a 4a 4a
24 51 bc 5e 3c 46 46 46 1a 13 4a 4a 4a 4a 4a 1e 1a 7c
bd 27 45 46 46 46 1c 1 4a 4a 49 49 1c 1c 30 b0 bc 6 46
46 46 44 32 6 4a 49 49
1f 2d 2e 34 a6 bb 6 46 46 46 43 25 10 46 49

target

3f 48 4d 4c 4e 4f 4f 51 49 3a 35 3d 4f 4f 4d 4c 4b 4b 4a
4a 2c 1d 27 43 52 4f 51 48 39 34 3c 4e 4e 4d 4d 4c 4b 4a 4c 25 16 28
45 52 4f 51 48 38 33 3c 4e 4e 4d 4d 4c 4b 4a 4a 3e 3d 49 4a 4b 4e 50
48 38 33 3c 4d 4d 4c 4c 4b 4b 4a
49 4b 3a 1c e 20 4c 50 45 36 31 39 4a 4a 49 48 48 48 47
49 4e 3b 15 1d 27 31 39 35 29 26 2c 38 38 37 36 36 36 35
48 4b 44 3b 6c 58 6 16 21 19 18 1c 22 21 20 20 20 20 1f 49 4b 4a 47
3e 2b 17 35 3e 2d 29 31 3e 3f 3e 3d 3d 3d 3c 3d 45 4b 4a 2f 20 24 36
3d 39 34 3d 4f 50 4f 4e 4e 4e 4d
2a 39 49 49 4c 40 26 17 1e 35 32 39 4c 4d 4c 4b 4b 4b 4a
34 3e 4a 47 4b 46 38 3f 34 1a 28 3b 4b 4c 4b 4b 4b 4b 4a
6b 3a 2f 47 49 47 44 49 40 2b 2a 36 4b 4c 4b 4b 4b 4b 4a bb 38 c 46
48 48 48 31 3a 5c 36 2e 4b 4c 4a 4a 4b 4b 4a bb 89 51 2e 3e 49 47 43
36 27 32 41 4d 4c 4a 4a 4a 4a 4a
91 b5 8d 34 3b 45 46 4b 3c 1d 2b 36 38 47 4d 4a 4a 4a 4a
51 a5 a9 5e 42 3c 46 46 48 46 28 13 13 39 4e 4a 4a 4a 4a
27 74 a7 a4 46 24 4a 49 47 48 3f 2d 14 d 25 4f 4e 4a 49 16 43 88 c5 7b
3e 2c 3f 4a 46 4a 41 28 e 13 35 47 4d 49 1c 21 57 b0 cb 8 5 6 2c 4b 46
46 46 44 3b 21 6 34 4f 49

pattern2: (H0 V0) = (81 18), (SW SH) = (17 15), (TW TH) = (22 28)

source

39 38 37 25 78 cb 96 6a 6d 72 73 71 73 76 79 7a 7c 83 8a
29 3a 39 38 1f 8d ca 8f 6a 6c 6e 6e 71 75 77 78 7f 80 81 61 27 3a
39 38 16 a4 bc 7c 69 6b 6d 6e 72 76 7c 7b 7a 7c 1e 5c 2d 3a 3a
39 6 c8 b6 6e 69 6b 6d 72 7a 78 76 78 7a
3e 17 55 2b 3b 3a 39 22 c7 ac 68 69 6e 79 75 72 73 75 77
40 3f 6 71 3a 3b 3a 34 57 c7 8f 6b 77 72 6e 6e 71 73 7b
40 40 3f 57 12 3d 3c 3b 26 95 bc 7d 6e 6a 6b 6d 70 7b 81 41 41
40 6 44 16 3d 3d 3b 6 d1 af 66 67 69 6b 78 7e 78 41 41 41 e 2e 60
2f 3e 3d 38 58 c6 82 66 67 6e 74 71 70
41 41 41 e 41 11 30 3f 3e 3d 1c c3 b1 67 66 6b 6b 6c 6e
25 25 25 6 25 25 53 1 3f 3e 3d 24 c4 8e 64 67 68 6a 6c
2e 2f 2e c 2e 2e 14 65 3c 3f 3e 2c a9 b1 65 64 66 68 6e 42 43 43
10 43 43 43 6 5 41 40 3f 22 c2 89 63 65 6b 7c 42 43 43 11 43 43
43 3c 5e 3a 41 40 2c ac b7 62 68 78 73
42 43 43 11 44 45 44 43 6 1 41 41 3f 2a c1 85 75 72 69
42 43 43 11 44 45 45 45 3b 5d 3c 41 40 1c c0 af 6b 64 64
42 42 43 11 44 45 45 45 45 cd 43 42 40 63 b5 67 60 62

target

3a 3a 3a 2f 21 79 bb be 8a 6d 69 6d 6e 6e 70 73 76 77 77 7b 80 80 2d 37 3c 34 27
3c 95 c6 a3 75 69 6a 6c 6d 6e 71 74 76 79 7c 7 e 7d
28 35 3b 3a 37 11 63 b5 ba 86 6e 68 6b 6d 6d 70 73 76 7b 7c 7b 7a
49 34 34 3c 3a 24 2c 72 d3 a6 78 64 6a 6c 6c 6f 72 79 7a 7a 78 79
5a 37 31 3a 3a 3d c 37 c8 bf 8b 69 69 6b 6c 6f 73 7a 79 77 76 78 31 48 3a 2d 3b
3d 25 2b 6b c4 af 7e 63 69 6c 71 77 78 76 74 74 76 17 47 41 30 3b 3a 3d 2c 25 b0
c3 97 66 68 6c 73 79 74 72 72 73 75
2e 1c 3f 58 3d 3a 3d 32 2a 75 b6 b1 78 69 70 76 74 70 70 70 71 73
41 b 41 6b 34 3a 3c 38 36 43 9e bf 8f 6e 72 75 70 6d 6e 6f 71 74
41 2b 4f 58 1a 39 3f 3b 38 29 82 bb a9 78 70 6f 6c 6c 6d 6e 71 78 40 42 4a 3e 17
31 3c 3d 3a 23 59 a0 c1 8a 71 69 69 6b 6c 6e 73 7c 41 44 2a 1e 38 1d 2e 40 3c 36
1d 65 db a8 75 62 68 6a 6a 70 78 7e
41 45 1b e 44 26 2f 3d 3e 40 b 42 c8 bb 80 63 66 69 6a 71 7a 7c
41 45 21 f 34 52 42 33 3f 3e 2a 40 83 c1 92 6c 65 67 6c 72 76 74
42 46 25 13 33 52 3e 30 41 3d 3c 38 50 c5 a8 79 62 67 6c 70 72 6f 43 47 24 18 3f
23 24 35 44 3e 3f 2c 32 c5 bf 89 60 66 6b 6c 6d 6d 3a 3d 1f 15 3b 15 28 39 2d 3c
41 2e 2a 91 bb 9e 6a 66 69 6a 6a 6b
28 2b 15 c 29 20 43 3c 4 38 41 3c 36 34 a0 b6 7e 65 66 69 68 6a
25 27 13 c 26 27 3a 37 1f 3d 40 41 3a 15 92 c1 91 64 63 67 67 69
2d 2e 19 11 2d 2e 1c 2d 60 43 3d 40 3b 24 8a be a2 65 62 65 66 68 37 39 1e 15
37 38 26 2c 49 28 35 42 3d 31 5f 9d b4 72 64 64 66 68 42 45 24 19 43 43 44 30 8
2 2c 45 40 3e 26 6b c3 87 69 61 66 6a
44 48 26 19 45 45 49 37 13 23 39 43 41 42 1a 5f c7 a0 6d 5e 68 71
43 46 25 19 44 44 43 3f 3b 5a 49 3c 41 41 27 61 bb b7 74 5d 6b 78
43 46 25 19 44 45 44 44 42 3d 26 27 43 41 33 4a 83 be 84 69 71 77 43 46 25 1a
45 46 45 46 3f d 0 17 44 41 40 2f 42 be 98 79 74 72 43 46 25 1a 45 46 45 46 41 1c
26 37 42 41 44 27 2f c3 ae 85 6e 6b
43 46 25 1a 45 46 45 45 44 3a 53 54 3b 41 43 29 31 bc bd 8e 65 64

pattern3: (H0 V0) = (45 45), (SW SH) = (16 29), (TW TH) = (26 37)

source

45 46 49 4b 4c 45 25 4e 4c 4b 49 46 4e bc c0 c3 c7 76
2e 4b 4d 4f 50 48 27 51 51 50 4e 4b 4a b9 be c0 c4 74 9f 13
39 52 53 4d 2c 54 54 52 50 4e 48 b5 ba be c2 72 b1 ae ac 1f
39 4f 2f 58 57 55 54 51 45 b3 b7 bb bf 73
ae ac a9 a7 a6 25 13 5a 5a 57 56 53 42 b0 b4 b9 bd c2
ad aa a8 a6 a4 a3 8c 15 4c 4b 48 45 4f af b2 b7 bb c0
ac a9 a7 a4 a3 a0 a0 93 c 3e 3c 3a 5a ad b2 b5 ba c0 4c a9
a6 a4 a1 a0 a0 9f 52 57 5a 57 3e ac b1 b4 b9 bf 50 1c 5c a4
a1 a0 9f 9e 9f 2b 5b 58 3e ac b1 b5 b9 75
52 55 45 45 a2 a1 a0 9f a0 4 5a 58 3f ad b1 b5 b9 6b
50 54 56 30 a4 a2 a1 a1 a2 10 59 57 40 ae b1 b6 ba 6c
4f 52 54 38 a5 a4 a3 a3 a3 4 57 55 42 af b3 b7 bc 6d 4d 4f 50
17 a7 a7 a6 a6 a6 f 54 52 44 b2 b5 ba be 6e 1c 6 49 ad ab aa
a9 a9 a9 3b 51 4f 46 b5 b8 bc c1 71
b7 b4 b1 b0 ae ad ac ad 1c 4f 4e 4c 4a b8 bb bf c4 74
bb b8 b5 b3 b2 b1 b0 43 44 4c 4b 48 4e bc bf c3 c7 78
c0 bc ba b8 b5 a9 4e 32 3d 3b 3a 38 5d c0 c4 c7 cb 81 c3 c1
8b 63 32 1 c 20 1f 1e 1c 1a 45 59 5c 5f 5f 64 2c 32 3b 3d 3e
36 17 3f 3f 3d 3b 3a 37 35 32 2f 29 4
32 34 35 37 38 31 12 3a 39 38 36 35 32 30 2d 2c 26 6
2f 30 32 32 32 2b c 34 34 32 31 2f 2e 2c 2a 27 22 6
2c 2d 2d 2e 30 27 6 2e 2e 2d 2d 2b 2a 29 27 25 20 c 29 2a 2a
2b 2b 22 2 2b 2a 29 29 29 27 25 24 22 1f e 27 27 27 27 27 1f
1 27 27 25 26 25 22 22 20 20 1c f
26 26 25 25 25 1c 4 24 24 22 22 20 20 20 20 20 1c f
24 24 24 24 22 1a 5 22 22 20 20 20 1f 1f 1f 1e 1a f
22 22 22 22 22 1a 6 20 20 20 20 1f 1f 1f 1f 1f 1a 10 1c 1a 1a
17 18 11 c 18 18 18 17 16 16 16 16 16 11 15 6 6 6 cf 18 25 1f
22 27 2c 30 30 32 34 37 3a 3a
24 22 22 22 20 19 6 20 20 1f 20 1f 1a 19 16 16 15 11
25 25 24 22 22 1a 6 20 20 20 20 20 20 20 20 20 20 c

target

4b 4d 4d 4f 50 50 4d 42 29 34 51 53 51 50 4f 4e 4c 49 43 73 b9 c1 bf c0 c5 c4
12 20 3c 51 57 54 52 46 2d 38 53 55 53 52 50 4f 4e 4b 42 71 b6 be bc be c4 c2
68 75 6e 37 2c 3d 4d 4d 33 3c 56 58 56 54 53 52 51 4c 41 6e b4 bb b9 bb c2 c0 b9 be a2
53 45 57 4a 37 23 36 5b 5c 59 57 56 55 54 4e 3f 6c b2 b9 b7 b9 be be
ac a9 aa ac b3 ab 5d 21 17 2e 54 5b 5a 57 56 55 54 4d 3e 6b b0 b7 b5 b8 bb bd
aa a9 a8 a7 a6 a5 a0 97 8c 54 14 35 53 4e 4b 4a 47 46 46 73 af b5 b3 b6 b9 bb
a9 a8 a7 a5 a4 a3 a5 a5 a3 8f 6a 33 1a 38 41 3d 39 3e 4f 7b ae b4 b3 b5 b8 ba b1 b0 a9
a4 a3 a2 a0 a0 a1 a8 a1 4f 20 42 4d 48 46 45 48 72 ac b4 b3 b3 b6 ba 8f 98 9d a4 a5 a1
a0 a0 a0 a2 a0 7c 5e 53 55 5c 5d 53 3a 64 ac b4 b2 b3 b7 b9
1c 3c 6b 9a a8 a1 a0 a0 9f 9f 9e a7 8d 3a 35 5b 5d 52 3b 65 ac b4 b2 b4 ba b9
48 45 47 50 74 a2 a5 a1 a0 a0 9f ab 89 1b 1d 5a 5f 52 3c 66 ad b4 b2 b4 bb b9
58 53 47 30 5c a3 a7 a1 a0 a0 a0 ac 89 1c 1d 5a 5e 52 3c 67 ae b5 b2 b5 bc b9 53 58 4f
30 5a a4 a9 a3 a2 a2 a2 ad 8b 1f 1f 58 5c 51 3d 68 ae b5 b3 b5 bc bb 52 55 4d 32 5d a5
aa a4 a3 a3 a3 ae 8a 17 18 57 5b 50 3e 69 af b6 b4 b6 bd bc
52 56 45 12 46 a7 ae a7 a6 a6 a6 b1 8d 1f 1c 54 58 4e 3f 6c b2 b9 b6 b9 c0 be
f 2b 4f 74 93 aa ab a9 a8 a7 a8 b8 9f 3f 31 52 53 4d 40 6e b4 bb b8 ba c1 c0
4c 60 83 b0 b8 ac ab ab aa b0 b1 88 60 4a 47 50 4f 4b 41 71 b6 bd ba bc c3 c2
c0 bc b7 b2 af af ae af b0 ad 9b 46 19 46 53 4e 4c 4a 44 74 b9 c0 bd bf c6 c5 b8 b6 b5 b3
b3 b2 b1 b3 b7 87 43 3e 45 4b 4c 4b 49 47 46 78 bc c3 c0 c2 c9c7
bb bc bd bc bd bd bf aa 73 4b 34 3a 41 41 40 3f 3b 40 50 86 c6 ce cb cc d3 d2
c9 b4 9f 90 7e 6a 54 3e 27 20 25 28 29 28 27 26 22 2a 47 69 88 8d 8d 8f 91 91
97 81 69 58 43 2c 11 3 7 16 28 28 26 25 24 23 20 26 3a 44 47 48 48 4a 49 47 2c 33 38 3b
3d 3e 3d 32 19 24 40 42 40 3e 3d 3 c 3b 3b 37 35 33 31 2f 2e 2c 26 33 35 35 37 39 39 37
2c 15 20 3b 3c 3a 39 38 37 36 36 34 31 30 2e 2d 2c 2b 26
31 32 33 34 34 34 32 27 10 1b 36 38 36 34 34 33 32 31 30 2e 2d 2c 2a 29 28 23
2f 2f 30 30 30 31 2f 23 b 16 31 33 31 30 30 2f 2e 2d 2c 2c 2b 2a 28 27 25 21
2c 2c 2c 2d 2e 2f 2c 20 7 12 2d 2f 2d 2c 2c 2c 2b 2a 29 29 28 27 26 25 24 20 2a 2a 2a 2b
2c 2b 28 1c 4 f 2b 2c 2a 29 29 29 29 29 27 26 25 24 24 22 22 1f 28 28 28 28 28 28 25 1a 3
d 28 2a 28 26 26 27 27 25 23 23 23 22 21 20 1f 1d
26 26 26 26 26 26 22 18 4 d 25 27 25 23 23 24 23 21 21 21 21 20 20 20 1f 1c
25 25 25 25 25 24 20 17 5 e 23 25 23 21 21 21 20 20 20 20 20 20 20 1f 1e 1b
24 24 24 24 24 22 1e 16 6 e 22 24 22 20 20 20 20 20 1f 1f 1f 1f 1f 1e 1d 1a 22 22 22 23 23
22 1e 16 7 e 20 22 20 20 20 20 1f 1f 1f 1f 1f 1f 1f 1f 1e 1a 1e 1e 1e 1b 1b 1c 17 11 9 e 1a
1b 19 19 19 18 17 17 17 17 17 17 16 16 14 11
ff 10 10 11 13 13 15 18 1a 1a 1b 1c 1e 1f 20 21 22 22 22 23 23 24 25 25 24
9 9 ae 10 11 16 1b 21 22 20 21 23 26 28 2b 2e 2f 2e 2e 2f 30 31 32 34 36
22 22 22 22 22 20 1d 15 7 e 20 21 20 1f 1f 20 20 1e 1b 19 19 17 16 16 16 15

Appendix C. Grading files

The files required for scoring can be divided into several parts:

(1) RTL design, that is, the RTL code designed by each participating team for this competition . If the design is modular and has multiple designs,

Please be sure to submit the calculation files together to avoid being unable to simulate when the judges are making scores.

If designwell is used in the design, please also submit the DW function model (DW*.v) used to avoid evaluation.

Simulation cannot be performed while the reviewer is grading.

(2) Gate-Level design, which is the gate-level netlist generated by the synthesis software and the corresponding SDF file.

(3) report file. The participating teams must write a report.txt file according to their own design content to facilitate the organizer's preparation.

Row rating.

Table 3. Submission of files

RTL category		
Design stage	*File*	*Description*
N/A	report.txt Design Report Form	
RTL Simulation	*.v or *.vhd Verilog (or VHDL)	synthesizable RTL code
Gate-Level category		
Design stage	*File*	*Description*
Pre-layout Gate-level Simulation	*_syn.v Verilog gate-level netlist	
	*_syn.sdf Pre-layout	gate-level sdf
	*_syn.spef Pre-layout	gate-level parasitic file

report.txt

```
FTP account:                    B24xxx, FTP account
Level:                          A/B/C/D Design Completion Level
CYCLE_TIME:                     8. During simulation, the CYCLE_TIME value used by testbench
Time:                           3 patterns total simulation time
Power:                          PrimeTime Analysis Power
Area:                           synthetic area
MAX_CYCLE_PER_PATTERN: 50000, the upper limit of the number of cycles of a single pattern defined in tb.sv


---RTL category---
HDL simulator:                  xrun/vcs/vsim, the name of the HDL simulator used
RTL filename:                   Bicubic.v, RTL files, submodule files, DW simulation files…
--- Pre-layout gate-level ---
gate_level filename:            Bicubic_syn.v, gate-level file name
gate-level sdf filename:        Bicubic_syn.sdf, sdf file name
gate-level spef filename:       Bicubic_syn.spef, spef file name
-----(annotation)---------------------------------------- ----------
```

(The rest of the notes are optional and should be filled in according to the needs of each participating team)

Appendix D. File compression and organization steps

When all documents are prepared as listed in Table 3, they need to be submitted to TSRI. Please follow the steps below to submit relevant design files. Copy all files to the same folder and compress them. The steps are as follows:

1. Create a result_xxx folder. Among them, "xxx" means the submitted version. For example, "000" represents the first upload; "001" represents the second upload; 002 represents the third upload, and so on...

   > ***mkdir result_000***

2. Copy the files required in Appendix C to the result_xxx directory. For example:

   > ***cp Bicubic.v*** > ***cp result_000***

   Bicubic_syn.v result_000 > ***cp Bicubic_syn.sdf***

   result_000 > ***cp report.000 result_000*** Please put all

   files in the result_000 directory and do not add

   other subdirectories.

3. Execute the tar command to package the result_xxx folder. The tar command example is as follows: >

   tar cvf result_000.tar After result_000 is executed, you

   should get the result_000.tar file. If you use the Windows system,

   you can compress it in zip format, except for tar and zip formats . In addition, please do not use other compression formats.

4. Use ftp to upload result_xxx.tar to the ftp server provided by TSRI. The review will be based on the last uploaded design file number.

   Do graded assignments.

   The FTP upload needs to be switched to binary mode, and the transmission port is set to 21 (port:21).

   The ftp account number and password have been emailed to each participant before the game. If you have any questions, please contact the organizer

   Hsinchu Semiconductor Center: iccftp.tsri.org.tw (140.126.24.18)

   EDA Cloud: Please see the terminal announcement in EDA Cloud

5. If you need to submit an updated version, please repeat the above steps and remember to modify the version number of the tar/zip file because you cannot modify it.

   Change, delete or overwrite previously uploaded information