**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(ˇ/X) | | Synthesis(ˇ/X) |
| LBP | | | | ˇ | | ˇ |
| CLBP | | | | ˇ | | ˇ |
| 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.

一張含有 文字, 螢幕擷取畫面, 圖表, 數字 的圖片

自動產生的描述

Fig.1 File hierarchy for Homework submission

Lab 6\_1: Local Binary Pattern

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**.

一張含有 文字, 螢幕擷取畫面, 圖表, 數字 的圖片

自動產生的描述

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

* 1. Choose a pixel in the image and select its neighboring pixels.
  2. Construct the mask using threshold function.
  3. Combine the binary values for all neighboring pixels to obtain a binary code for the central pixel and convert it to a decimal value.
  4. 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.

一張含有 文字, 寫生, 圖表, 圖畫 的圖片

自動產生的描述

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

一張含有 圓形, 螢幕擷取畫面 的圖片

自動產生的描述

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*.

1. Please **attach your waveforms** and **specify your operations** on the waveforms.

Read the value of gray data in sequence.

g\_c

一張含有 文字, 螢幕擷取畫面, 陳列, 電腦 的圖片

自動產生的描述

g\_0

一張含有 文字, 螢幕擷取畫面, 電子產品, 軟體 的圖片

自動產生的描述

g\_1、g\_2、g\_3、g\_4、g\_5、g\_6, then g\_7

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦 的圖片

自動產生的描述

Finally, lbp\_data is output according to the comparison result.

一張含有 文字, 螢幕擷取畫面, 電子產品, 軟體 的圖片

自動產生的描述

1. Show SuperLint coverage (including all files)

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

自動產生的描述

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

Clock period: 0.57

一張含有 文字, 軟體, 陳列, 螢幕擷取畫面 的圖片

自動產生的描述

Total cell area: 346.032010

一張含有 文字, 軟體, 陳列, 螢幕擷取畫面 的圖片

自動產生的描述

Post simulation time: 98294030ps

一張含有 文字, 電子產品, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

1. 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.

1. 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.

一張含有 文字, 螢幕擷取畫面, 圖表, 數字 的圖片

自動產生的描述

▲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

* 1. Choose a pixel(center) in the image and select its neighboring pixels.
  2. Calculate bilinear interpolation:
     1. Determine & .
     2. Determine , , , .
     3. Determine , .
     4. Determine ,,,.
     5. Determine , , , .
     6. Determine neighbor.
  3. Repeat b) to calculate all neighbors’ values.
  4. Construct the mask using threshold function.
  5. Combine the binary values for all neighboring pixels to obtain a binary code for the central pixel and convert it to a decimal value.
  6. 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.

一張含有 文字, 圖表, 方案, 地圖 的圖片

自動產生的描述

一張含有 圖表, 寫生, 圖畫 的圖片

自動產生的描述

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*.

1. Please **attach your waveforms** and **specify your operations** on the waveforms.

Take 1845 as an example:一張含有 電子產品, 文字, 螢幕擷取畫面, 電腦 的圖片

自動產生的描述

Read gray\_data to k\_c一張含有 文字, 螢幕擷取畫面, 電子產品, 軟體 的圖片

自動產生的描述

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.一張含有 文字, 電子產品, 螢幕擷取畫面, 電腦 的圖片

自動產生的描述

Calculate x1,x2,y1,y2一張含有 文字, 螢幕擷取畫面, 電子產品, 電腦 的圖片

自動產生的描述

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, w4.一張含有 文字, 螢幕擷取畫面, 電子產品, 電腦 的圖片

自動產生的描述

Pass the calculated value into k\_1一張含有 文字, 螢幕擷取畫面, 電子產品, 軟體 的圖片

自動產生的描述

Repeat the above until k\_7 is calculated, and calculate the lbp\_data value一張含有 文字, 螢幕擷取畫面, 電子產品, 軟體 的圖片

自動產生的描述

If it is the outer ring range, fill in 0 directly.一張含有 文字, 螢幕擷取畫面, 電子產品, 電腦 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 電腦, 軟體 的圖片

自動產生的描述

1. Show SuperLint coverage (include all files)

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

自動產生的描述

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

Clock period: 0.39

一張含有 文字, 軟體, 螢幕擷取畫面, 陳列 的圖片

自動產生的描述

Total cell area: 5074.099316

一張含有 文字, 軟體, 螢幕擷取畫面, 陳列 的圖片

自動產生的描述

Post simulation time: 394559071ps

一張含有 文字, 電子產品, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

1. 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 |