**National Cheng Kung University**

**Department of Electrical Engineering**

***Introduction to VLSI CAD (Spring 2023)***

**Lab Session 6**

**Design of Image Compressor and Decompressor**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Student ID1 | | |
| 陳慕丞 | E94096097 | | |
| Name | Student ID2 | | |
|  |  | | |
| Practical | | Points | Marks |
| Lab 6\_1 | | 65 |  |
| Lab 6\_2 | | 35 |  |
| Notes | |  |  |

**Due: 15:00 April 12, 2023 @ moodle**

**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

Lab 6\_1: compress the image with a codebook

You are about to integrate all components (MAN, MIN, controller…) to form a compressing system (*system*). The system will be the framework for your final design lab. The block diagram of system is as shown in **Fig2** and **Fig3**. You can modify the design but not the I/O of the top.

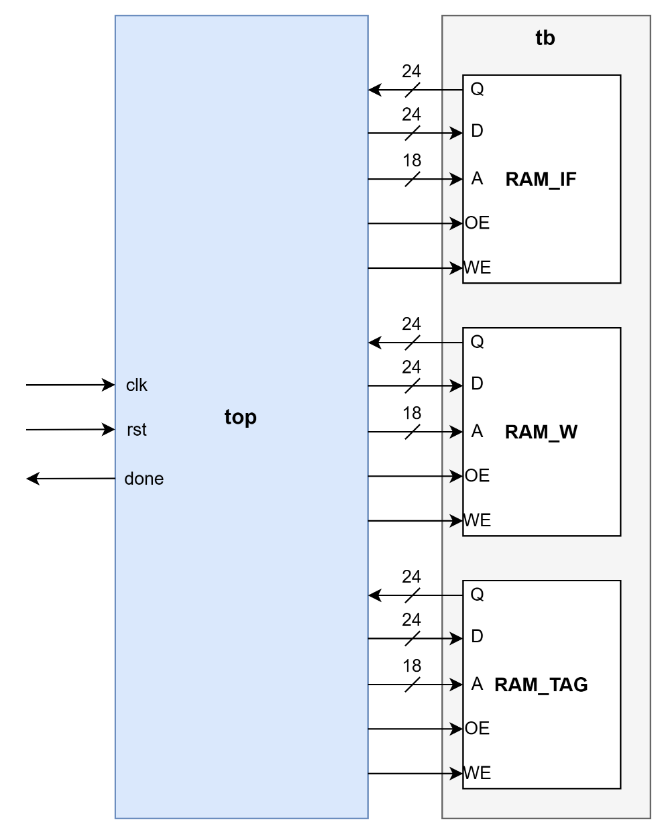


Fig2. The block diagram of system (external)

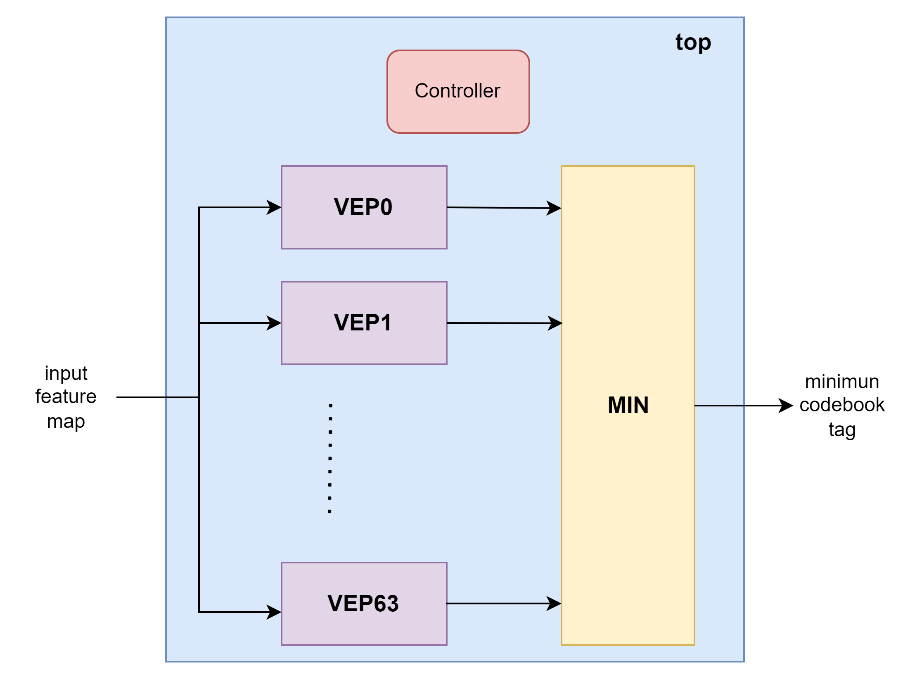
****

Fig3. The block diagram of system (internal)

* **Port list of top:**

|  |  |  |  |
| --- | --- | --- | --- |
| **signal** | **I/O** | **#bit** | **Description** |
| **clk** | input | 1 | clock |
| **rst** | input | 1 | reset, active high, asynchronous |
| **done** | output | 1 | When all data tags are written into RAM2, done is pull to 1.  Otherwise, done is push to 0. |
| **RAM\_IF\_Q** | input | 24 | data output from RAM\_IF |
| **RAM\_IF\_D** | output | 24 | data input from RAM\_IF |
| **RAM\_IF \_A** | output | 18 | RAM\_IF address signal |
| **RAM\_IF \_OE** | output | 1 | RAM\_IF read enable signal |
| **RAM\_IF\_WE** | output | 1 | RAM\_IF write enable signal |
| **RAM\_W\_Q** | input | 24 | data output from RAM\_W |
| **RAM\_W \_D** | output | 24 | data input from RAM\_W |
| **RAM\_W \_A** | output | 18 | RAM\_W address signal |
| **RAM\_W \_OE** | output | 1 | RAM\_W read enable signal |
| **RAM\_W \_WE** | output | 1 | RAM\_W write enable signal |
| **RAM\_TAG\_Q** | input | 24 | data output from RAM\_TAG |
| **RAM\_ TAG \_D** | output | 24 | data input from RAM\_TAG |
| **RAM\_ TAG \_A** | output | 18 | RAM\_TAG address signal |
| **RAM\_ TAG \_OE** | output | 1 | RAM\_TAG read enable signal |
| **RAM\_ TAG \_WE** | output | 1 | RAM\_TAG write enable signal |

****

Fig4. example waveform for RAM

* Understanding the function:

Once system is initialized, it

* 1. read codebook from the RAM\_W to 64 VEPs instances
  2. read a pixel from the RAM\_IF at a time, and compute the Manhattan distances among that pixel and 64 weights of codebook, after that, choose the id which represents the data tag for the weight having the shortest distance among other weights.
  3. writes the data tag back to the RAM\_TAG.
  4. repeats the process step (b)-(c) until the last pixel of RAM\_IF is updated.
  5. flags “done” when step (d) is completed.
* Know the basic design rules
  + All operations initiated on the positive edge trigger of the clock
  + Control signals:
    - *RAM\_WE*: To store the data to RAM
    - *RAM\_OE*: To read data from RAM
    - *done*: Stop the process
* Describe your design in detail. You can draw internal architecture or block diagram to describe your dsign.
  + MAN

一張含有 圖表 的圖片

自動產生的描述

* + MIN

一張含有 圖表, 圖解 的圖片

自動產生的描述

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

一張含有 圖表 的圖片

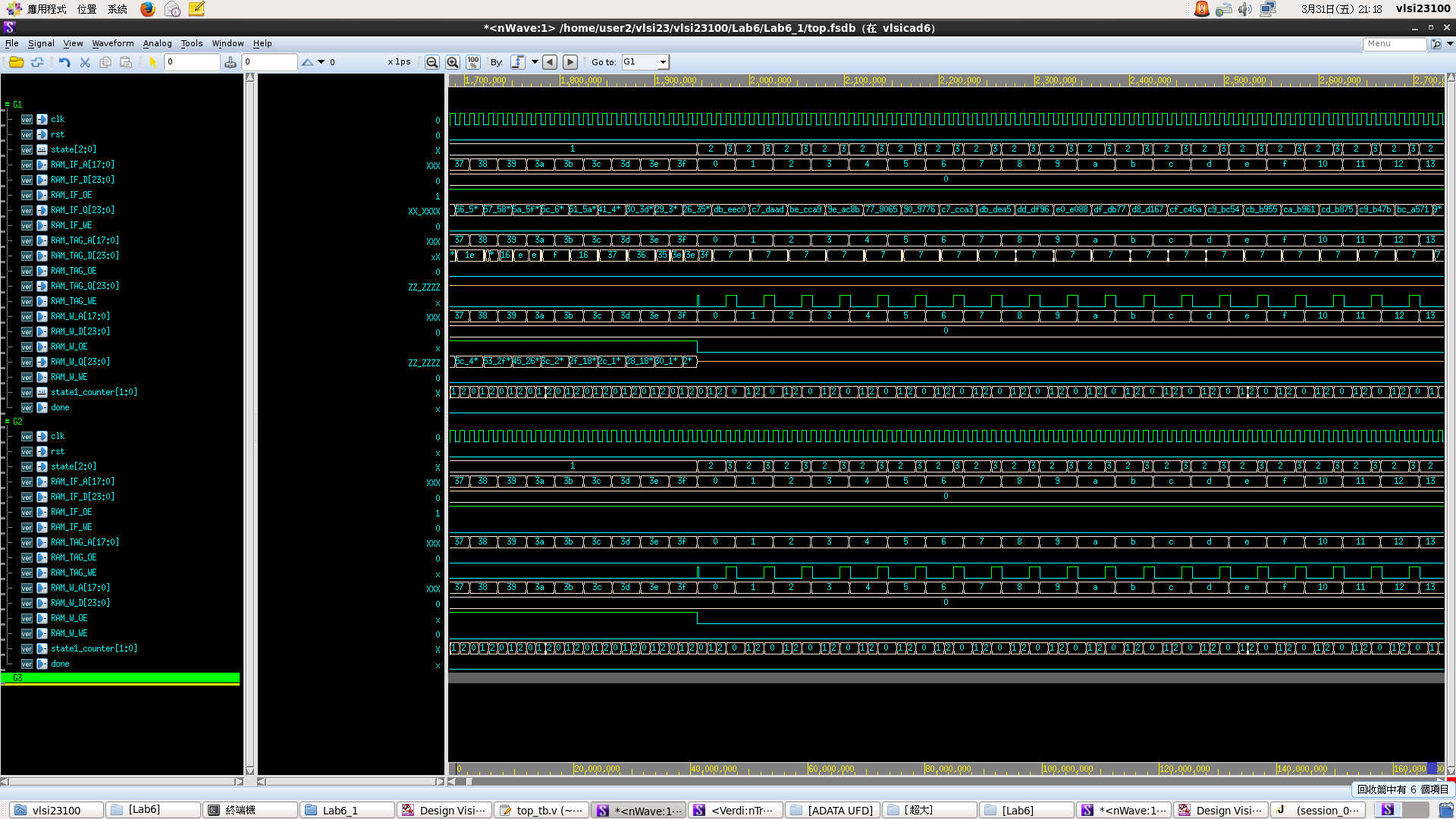
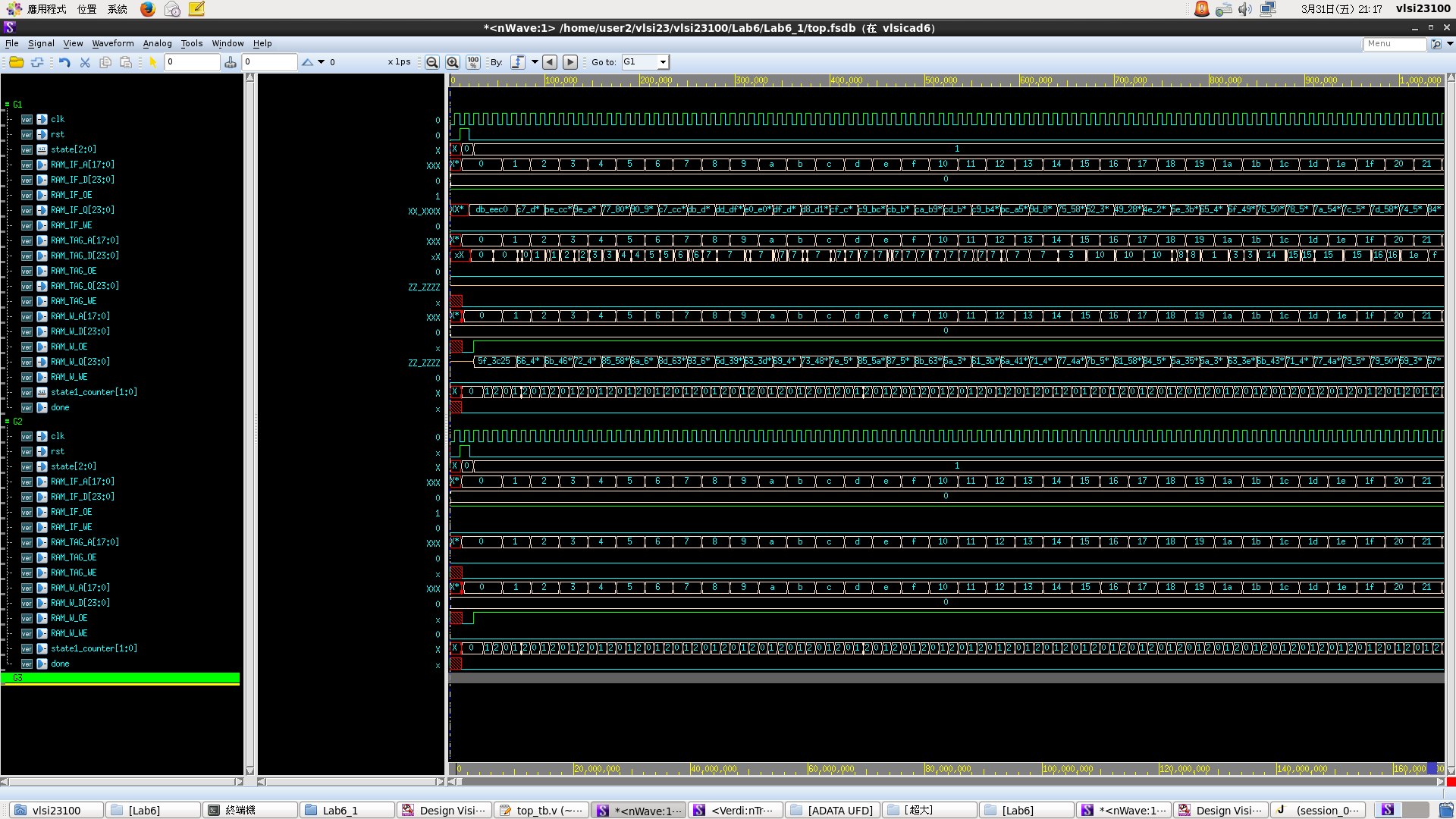
自動產生的描述

IDLE為初始state，接著進到LOAD\_WEIGHT，在LOAD\_WEIGHT中將所有RAM\_W中的weight load到codebook中，當所有的weight都load完之後進到FIND\_MIN，在FIND\_MIN中藉由WSC找出最小的manhattan distance，再來進入WRITE\_TAG將找出的tag寫進RAM\_TAG，若是所有pixel都算完了就拉高done並進入FINISH，否則回到FIND\_MIN繼續下一個pixel的計算。

1. Complete the controller ,MAN, MIN\_D, and top module, in the system.
2. Compile the verilog code to verify the operations of this module works properly.
3. Synthesize your *top.v* with following constraint:

* Clock period: no more than 20 ns.
* Don’t touch network: clk.
* Wire load model: saed14rvt\_ss0p72v125c.
* Synthesized verilog file: *top\_syn.v*.
* Timing constraint file: *top\_syn.sdf*.

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



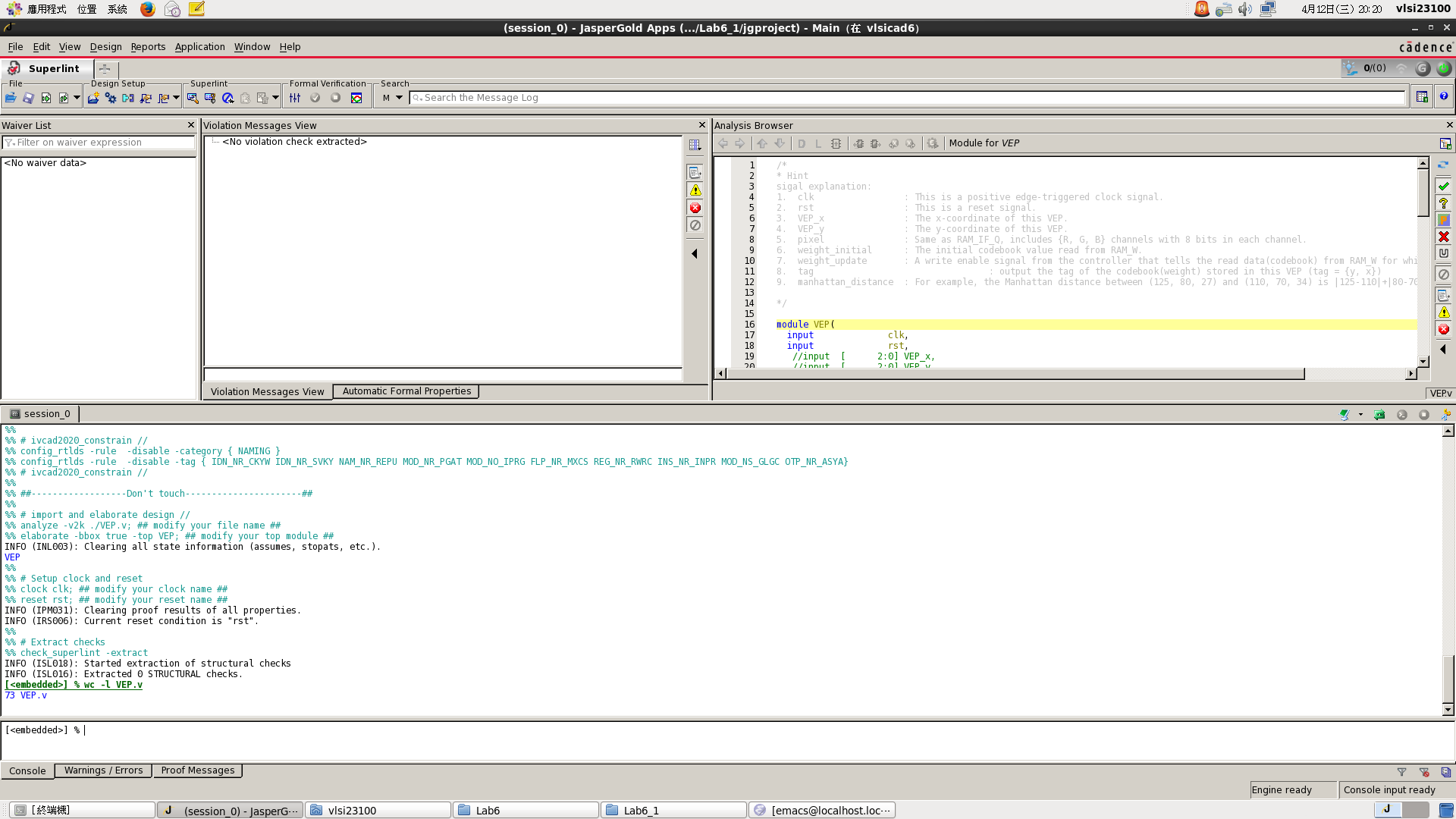
一張含有 文字, 螢幕, 螢幕擷取畫面 的圖片

自動產生的描述

第一張圖為正在存weight到codebook，由於從RAM\_W中提取weight需要1個半cycle的時間，加上存進codebook的時間，因此將每個weight的存取時間設為3個cycle，以state1\_counter計數。第二張圖為進入FIND\_MIN和WRITE\_TAG不斷循環的狀況，FIND\_MIN中一樣給讀取RAM\_IF中的資料3個cycle的時間再進入WRITE\_TAG，將結果存入RAM\_TAG並將RAM\_TAG\_WE拉起。第三張圖可以看到當4096個pixel都運算完後done拉起並將state停在FINISH。

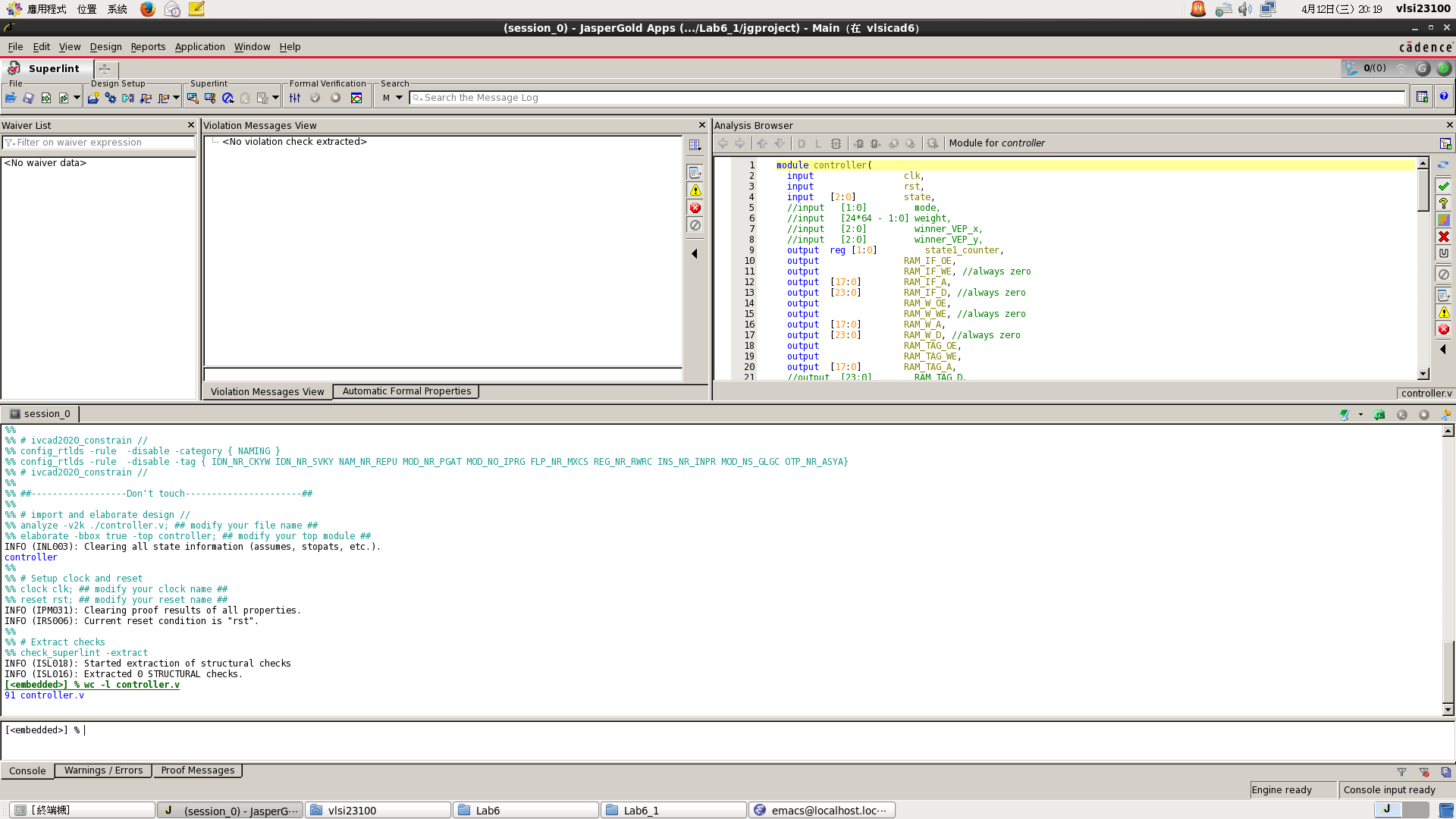
1. Show SuperLint coverage (including all files)

VEP.v:



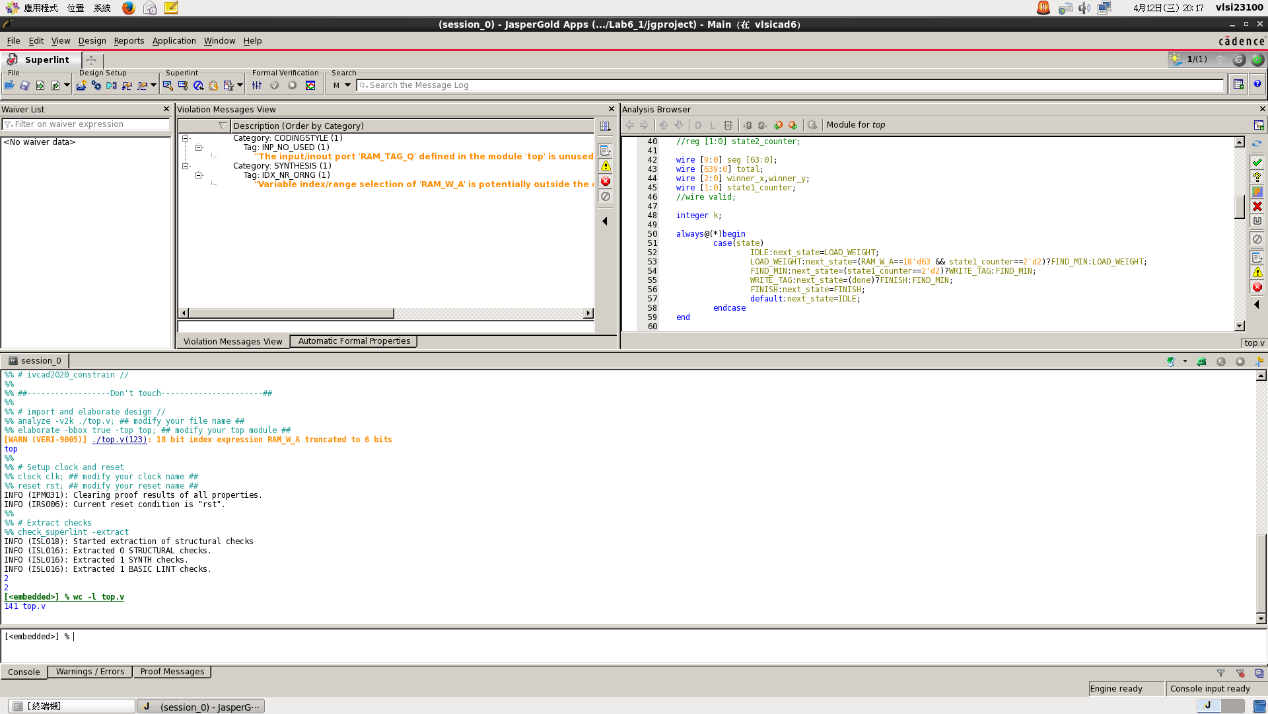
Coverage:100%

controller.v:



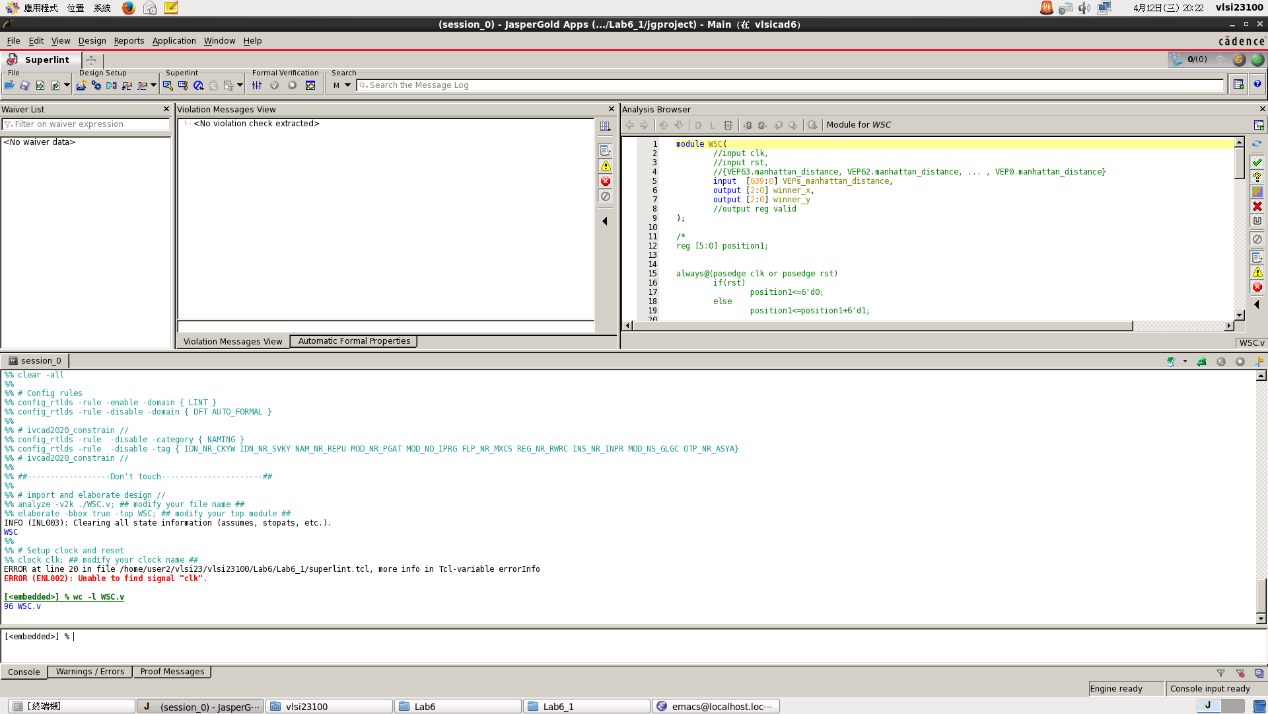
Coverage:100%

top.v:



Coverage:98.58%

WSC.v:



Coverage:100%

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

Clock period:10

一張含有 文字 的圖片

自動產生的描述

Total cell area:17782.421905

一張含有 文字, 信 的圖片

自動產生的描述

Post simulation time:165785NS

一張含有 資料表 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.

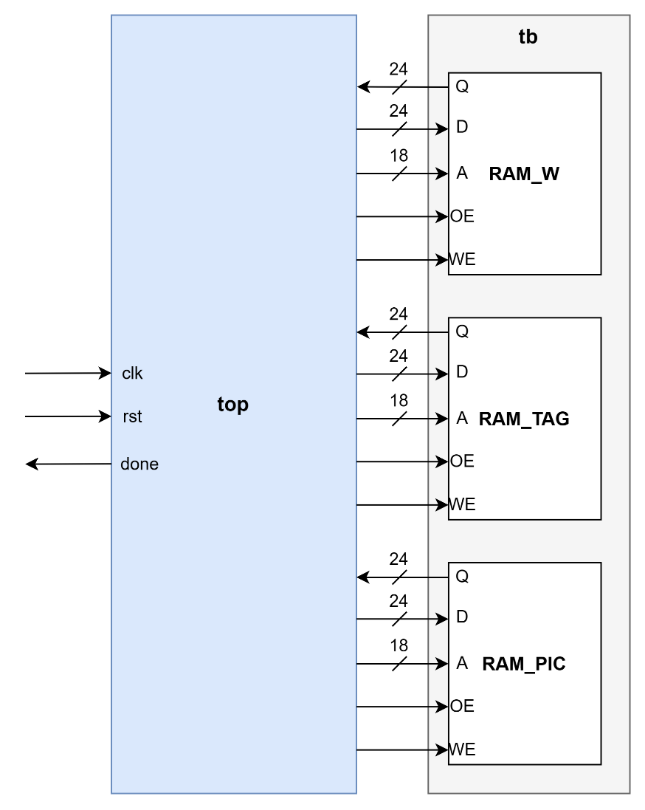
最初在設計WSC時是1個cycle比較2個值而已，但時間耗費太久跑不完，因此改成全部都用combinational電路比較，在1 cycle內就比完，大幅降低耗費的時間，也沒有Time violation的狀況發生。

1. Lessons learned from this lab

在這個lab學到如何利用拼湊多個module並相互傳遞不同資料已達成一件看似困難的任務，也學到如何運用generate語法快速複製多個module或串接wire，並練習運用一個FSM控制多個module。

Lab 6\_2: decompress the image

Use codebook and data tags to decompress the image



▲Block diagram for architecture(external)

* **Port list of top:**

|  |  |  |  |
| --- | --- | --- | --- |
| **signal** | **I/O** | **#bit** | **Description** |
| **clk** | input | 1 | clock |
| **rst** | input | 1 | reset, active high, asynchronous |
| **done** | output | 1 | When all data tags are written into RAM\_PIC, done is pull to 1. Otherwise, done is push to 0. |
| **RAM\_IF\_Q** | input | 24 | data output from RAM\_IF |
| **RAM\_IF\_D** | output | 24 | data input from RAM\_IF |
| **RAM\_IF \_A** | output | 18 | RAM\_IF address signal |
| **RAM\_IF \_OE** | output | 1 | RAM\_IF read enable signal |
| **RAM\_IF\_WE** | output | 1 | RAM\_IF write enable signal |
| **RAM\_PIC\_Q** | input | 24 | data output from RAM\_PIC |
| **RAM\_ PIC\_D** | output | 24 | data input from RAM\_ PIC |
| **RAM\_ PIC\_A** | output | 18 | RAM\_ PIC address signal |
| **RAM\_ PIC \_OE** | output | 1 | RAM\_ PIC read enable signal |
| **RAM\_ PIC \_WE** | output | 1 | RAM\_ PIC write enable signal |
| **RAM\_TAG\_Q** | input | 24 | data output from RAM\_TAG |
| **RAM\_ TAG \_D** | output | 24 | data input from RAM\_TAG |
| **RAM\_ TAG \_A** | output | 18 | RAM\_TAG address signal |
| **RAM\_ TAG \_OE** | output | 1 | RAM\_TAG read enable signal |
| **RAM\_ TAG \_WE** | output | 1 | RAM\_TAG write enable signal |

* Control signal
  + done: Stop the process if you decompress all pixels in a figure.
* Draw your state diagram and explain your design. You can draw internal architecture to describe your design

一張含有 圖表 的圖片

自動產生的描述

RAM\_W\_WE、RAM\_W\_D、RAM\_TAG\_WE、RAM\_TAG\_D、RAM\_PIC\_OE都為0，RAM\_W\_OE、RAM\_TAG\_OE都為1，RAM\_W\_A為RAM\_TAG\_Q的值，RAM\_PIC\_D為RAM\_W\_Q的值，wait\_counter、RAM\_PIC\_WE、pixel\_counter以wait\_counter是否數到2為判別依據，若數到2 wait\_counter、pixel\_counter加1，RAM\_PIC\_WE為1，而RAM\_TAG\_A和RAM\_PIC\_A為pixel\_counter的值，done在所有pixel都運算完的下一個cycle拉起。

1. Complete the top module, in the system.
2. Compile the verilog code to verify the operations of this module works properly.
3. Synthesize your *top.v* with following constraint:

* Clock period: no more than 10 ns.
* Don’t touch network: clk.
* Wire load model: saed14rvt\_ss0p72v125c.
* Synthesized verilog file: *top\_syn.v*.
* Timing constraint file: *top\_syn.sdf*.

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

一張含有 文字, 螢幕, 電子產品, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 文字, 螢幕, 電子產品, 螢幕擷取畫面 的圖片

自動產生的描述

RAM\_W\_WE、RAM\_W\_D、RAM\_TAG\_WE、RAM\_TAG\_D、RAM\_PIC\_OE都為0，RAM\_W\_OE、RAM\_TAG\_OE都為1，RAM\_W\_A為RAM\_TAG\_Q的值，RAM\_PIC\_D為RAM\_W\_Q的值，RAM\_TAG\_A和RAM\_PIC\_A為pixel\_counter的值，當wait\_counter數到2時RAM\_PIC\_WE為1，pixel\_counter加1，當所有pixel都做完時done拉起。

1. Show SuperLint coverage (include all files)

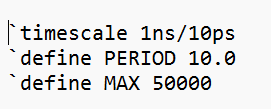
一張含有 文字 的圖片

自動產生的描述

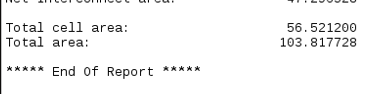
Coverage:96.61%

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

Clock period:10



Total cell area:56.521200



Post simulation time: 122895NS

一張含有 資料表 的圖片

自動產生的描述

1. Lessons learned from this lab

在此lab我學到解壓縮圖片的原理，並理解到壓縮圖片比解壓縮圖片困難與繁瑣，也學到如何控制RAM的讀取與寫入。

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) | ncverilog top\_tb.v +access+r +define+FSDB |
| Lab6\_1(post) | ncverilog top\_tb.v +access+r +define+FSDB+syn |
| Lab6\_2(pre) | ncverilog top\_tb.v +access+r +define+FSDB |
| Lab6\_2(post) | ncverilog top\_tb.v +access+r +define+FSDB+syn |