National Cheng Kung University

Department of Electrical Engineering

Introduction to VLSI CAD (Spring 2022)

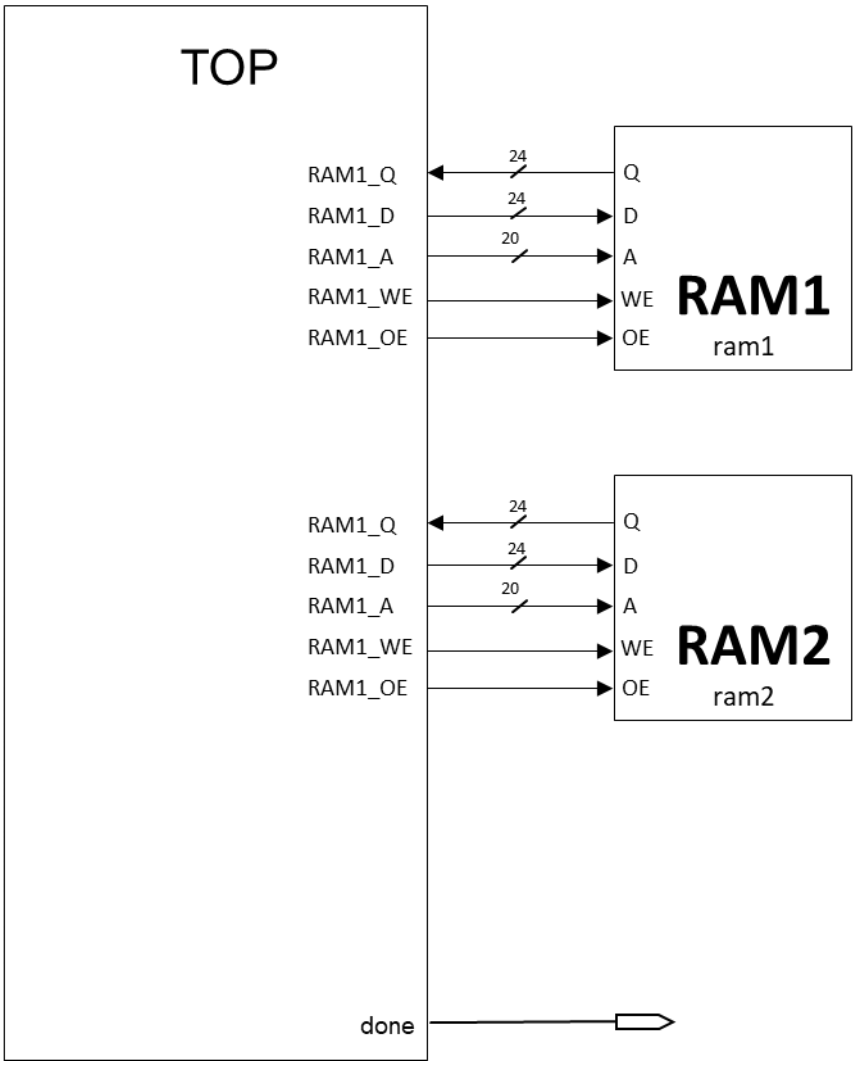
Lab Session 6

Design of Compressing system and Decompressing system

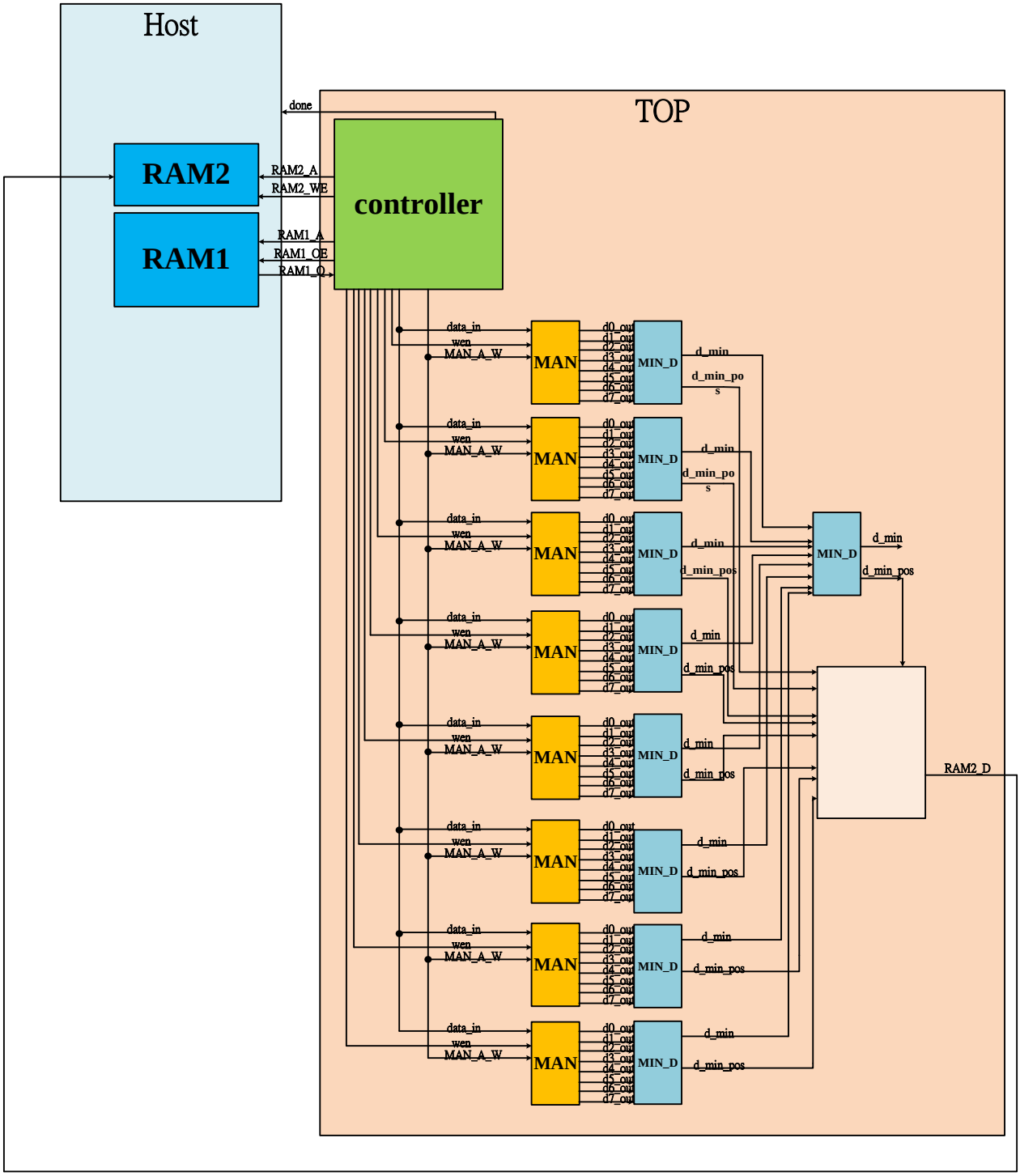
Yu-Chi Chu

Lab 6_1: compress the image with a codebook

You are about to integrate all components (MAN, MIN_D, 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**. (Clock pin and reset pin is ignored in the graph, but you should implement it)



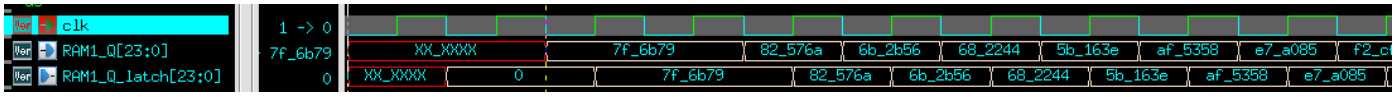
▲Fig2. The block diagram of system (external)



▲Fig3. The block diagram of system (internal)

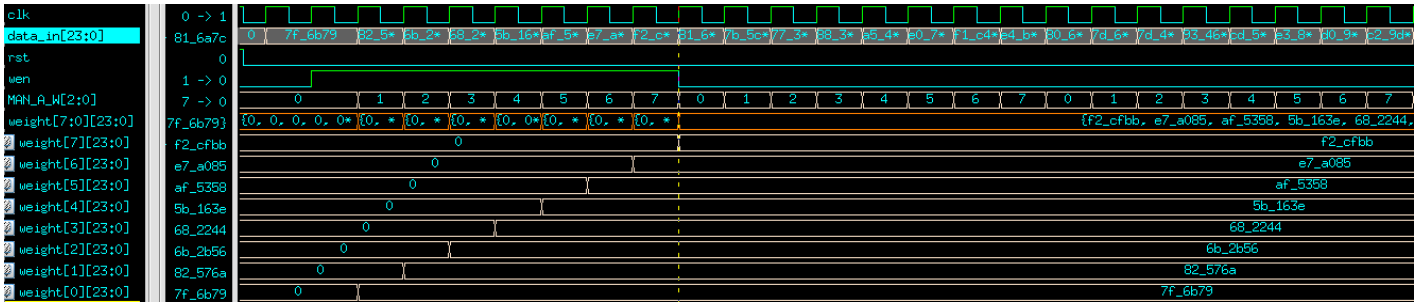
- **Port list of each module:**
- controller

signal	I/O	#bit	Description
clk	input	1	clock
rst	input	1	reset, active high, asynchronous
RAM1_Q	input	24	data output from RAM1
done	output	1	When all data tags are written into RAM2, done is pull to 1, or done is push to 0 at positive edge clk
RAM1_OE	output	1	RAM1 read enable signal
RAM1_A	output	20	RAM1 address signal
RAM2_WE	output	1	RAM2 write enable
RAM2_A	output	20	RAM2 address signal
RAM1_Q_latch	output	24	Store data output from RAM1 for MAN
wen0~wen7	output	1	Write enable signal for MAN
MAN_A_WEIGHT	output	3	Write address for MAN



- **MAN**

signal	I/O	#bit	Description
clk	input	1	clock
rst	input	1	reset, active high, asynchronous
data_in	input	24	data from RAM1
wen	input	1	Write enable signal，at positive edge clk，if wen is 1，write data_in into register file according to MAN_A_W
MAN_A_W	input	3	Write address for MAN
d0_out~d7_out	output	11	Manhattan distances between a pixel and 8 weights in a codebook
weight	reg	8x24	Store 8 weights in a codebook



- **MIN_D**

signal	I/O	#bit	Description
clk	input	1	clock
rst	input	1	reset, active high, asynchronous
d0~d7	input	11	8 Manhattan distances computed from MAN
d_min	output	11	The shortest distance among those 8 Manhattan distances
d_min_pos	output	3	Range from 0~7,for example, if id is 0, then it represent d0 is the smallest

	distance; If id is 1, then it represent d1 is the smallest distance, and so on.

- top

signal	I/O	#bit	Description
clk	input	1	clock
rst	input	1	reset, active high, asynchronous
RAM1_Q	input	24	data output from RAM1
RAM1_D	output	24	Data written into RAM1
RAM1_A	output	20	RAM1 address signal
RAM1_WE	output	1	RAM1 write enable
RAM1_OE	output	1	RAM1 read enable signal
done	output	1	When all data tags are written into RAM2, done is pull to 1, or done is push to 0 at positive edge clk
RAM2_D	output	24	Data written into RAM2
RAM2_A	output	20	RAM2 address signal
RAM2_WE	output	1	RAM2 write enable
RAM2_OE	output	1	RAM2 read enable signal

- Understanding the function:

  Once system is initialized, it

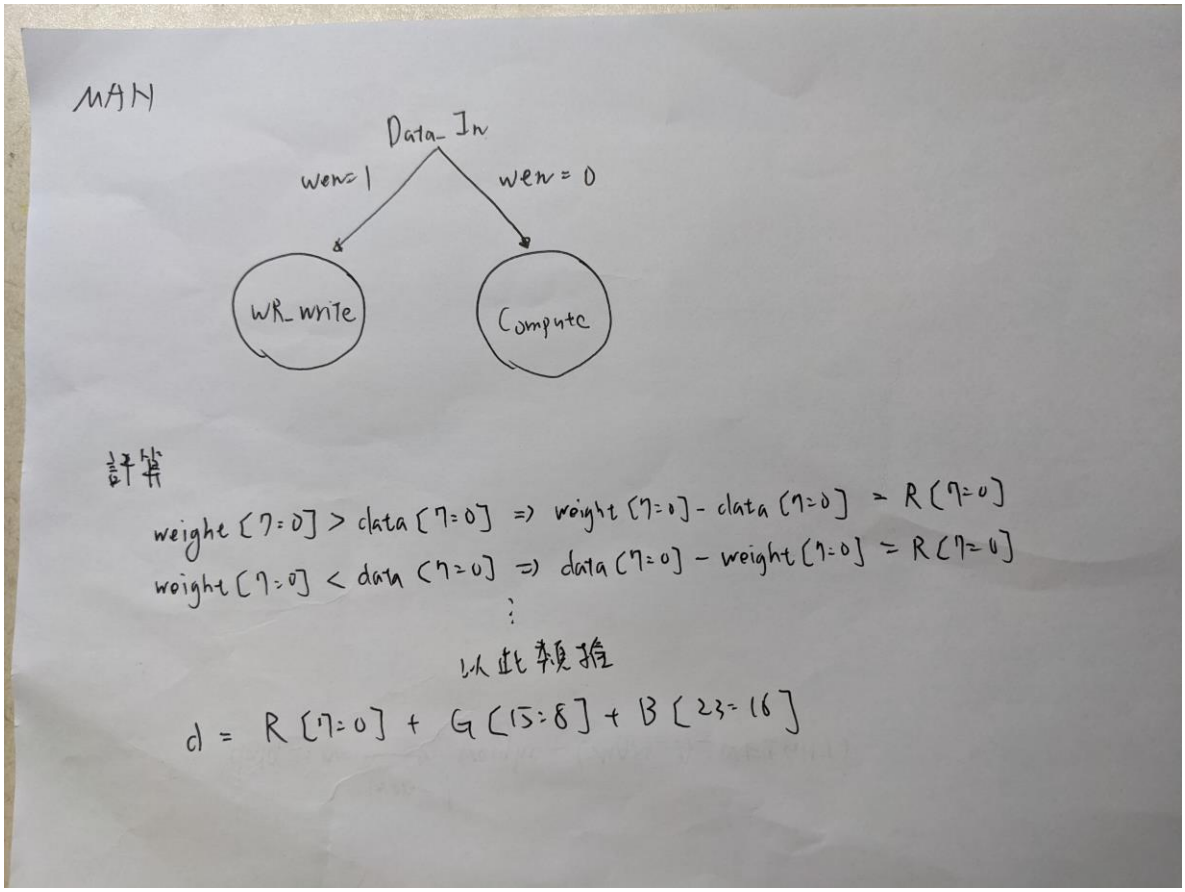
  a) read codebook from the RAM1 to 8 MAN instances
  b) read a pixel from the RAM1 at a time, and compute the Manhattan distances among that pixel and 64 weights in a codebook, after that, choose the id which represent the data tag for the weight having the shortest distance among other weight
  c) writes the data tag back to the RAM2;
  d) repeats the process step (b)-(c) until the last pixel of RAM1 is updated;
  e) 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

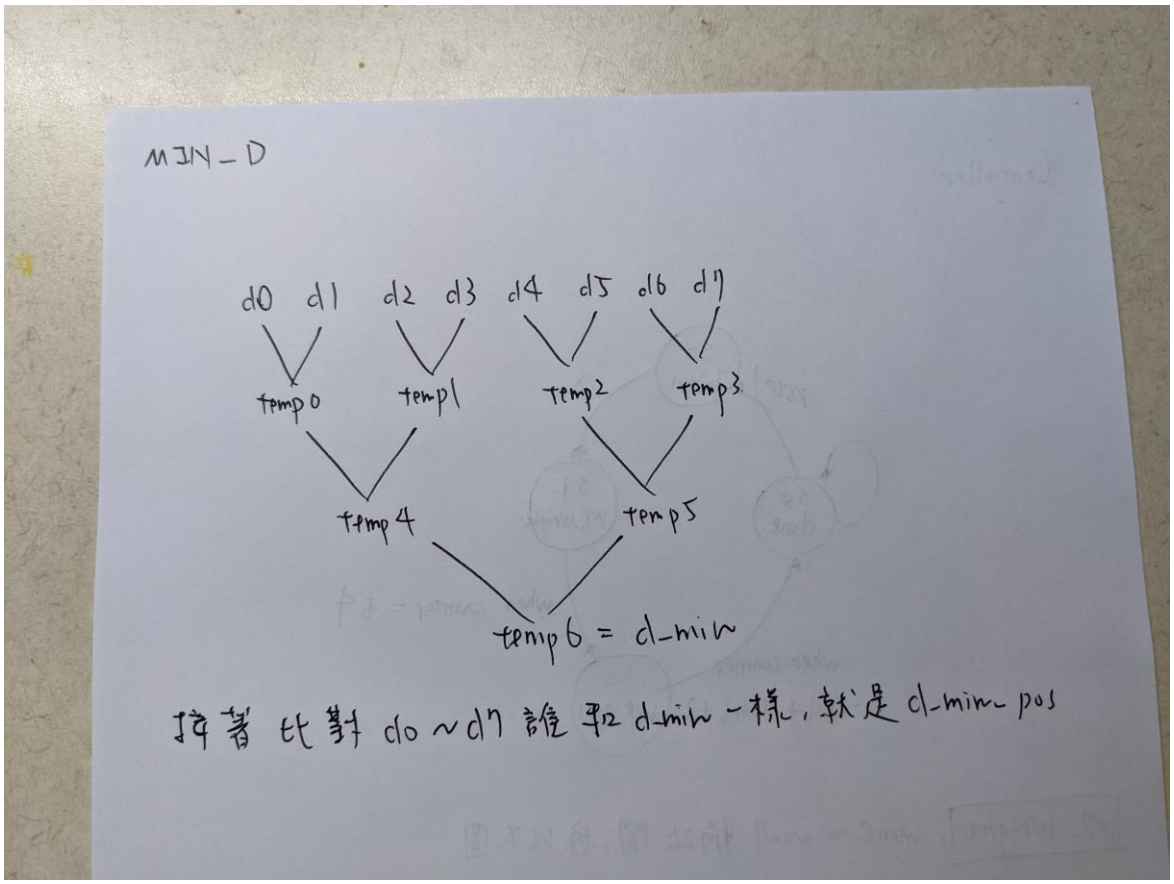
- *done*: Stop the process

➢ Describe your design in detail. You can draw internal architecture or block diagram to describe your dsign.

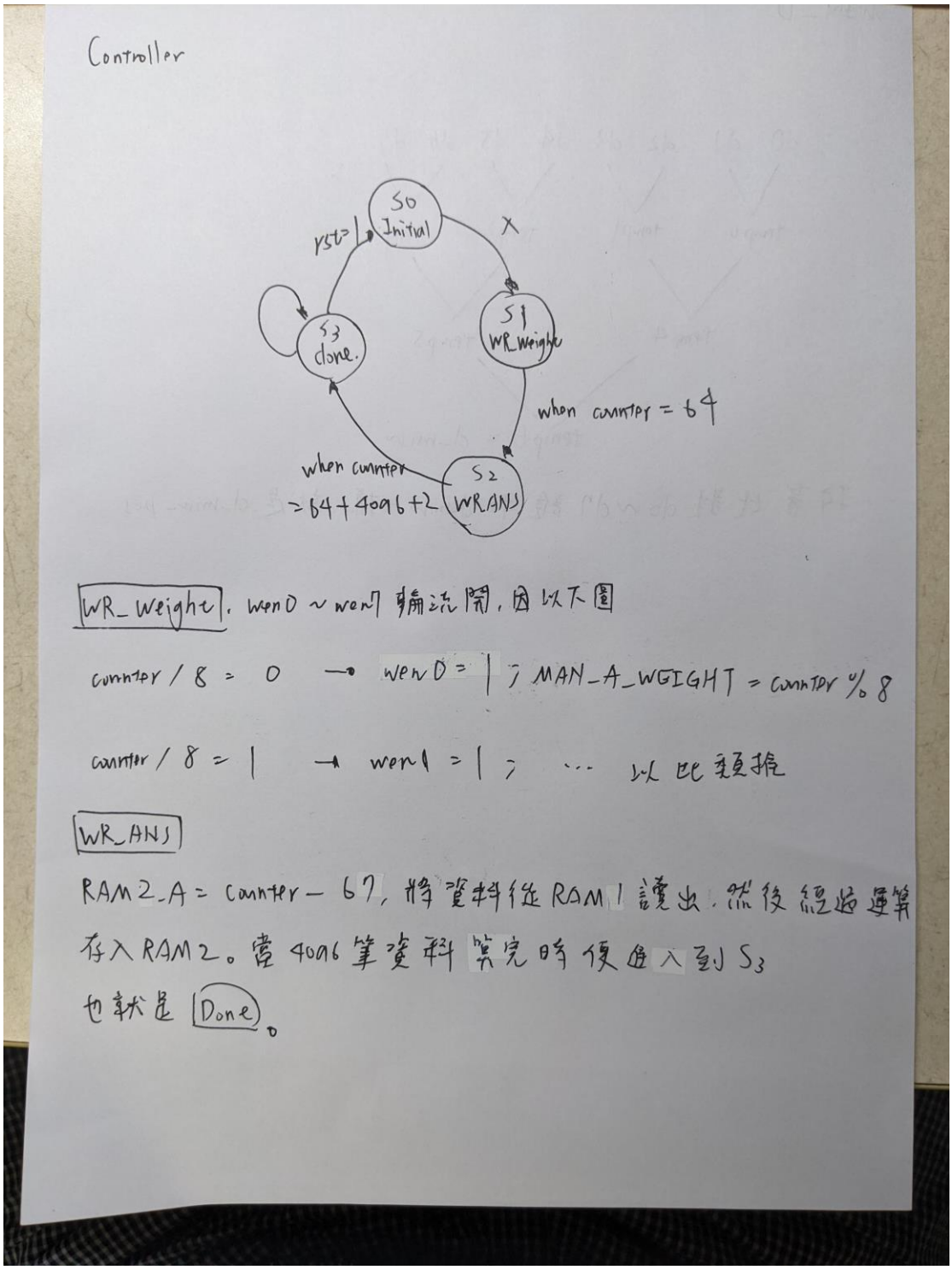
■ MAN



■ MIN_D



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



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.

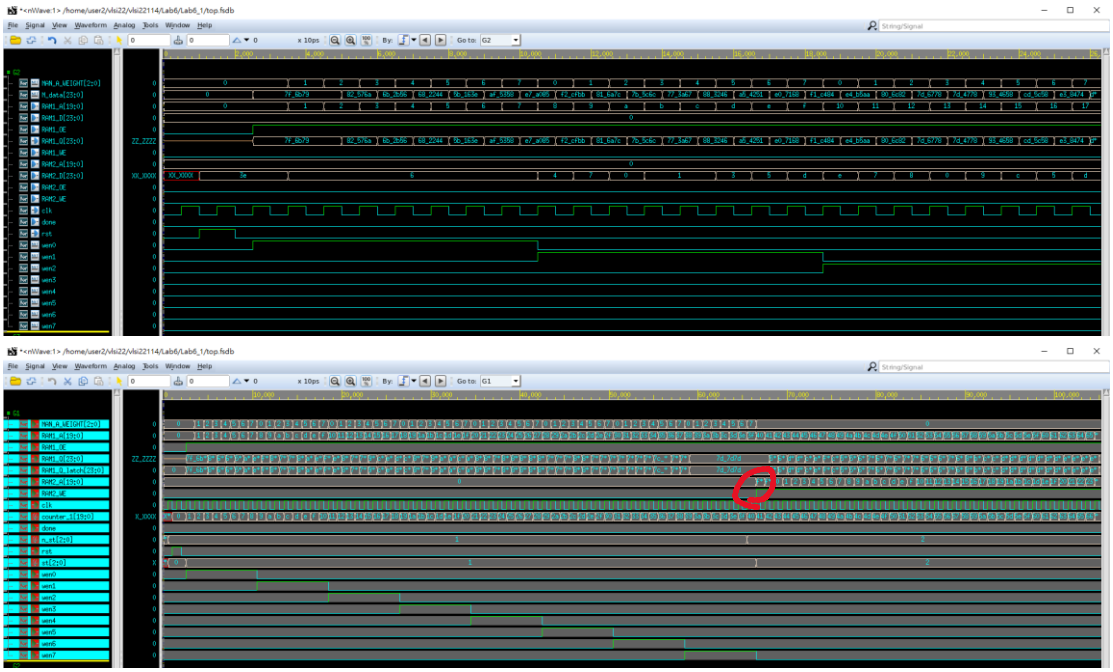
3) Synthesize your *top.v* with following constraint:

- 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*.

4) Please **attach your waveforms** and **specify your operations** on the waveforms.

(Before Synthesis)



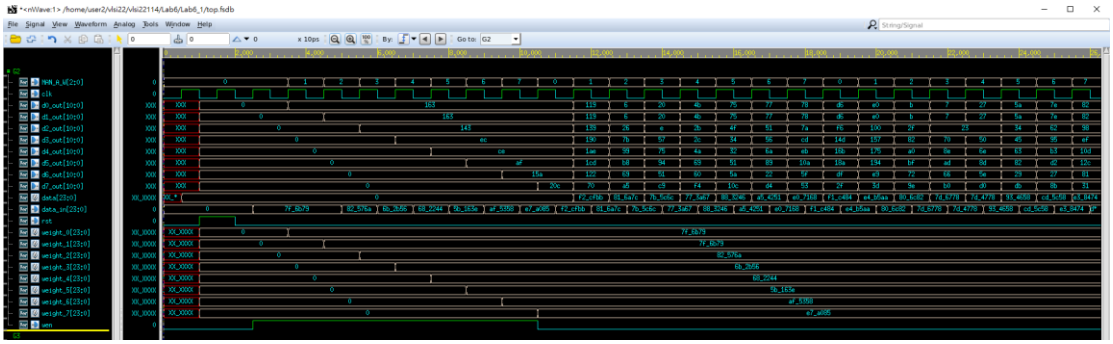
Reset (rst = 1): Initially, all registers are reset.

Reading from RAM1 (RAM1_OE = 1): The RAM1 output enable (RAM1_OE) is activated, allowing data (Codebook values) to be read from RAM1.

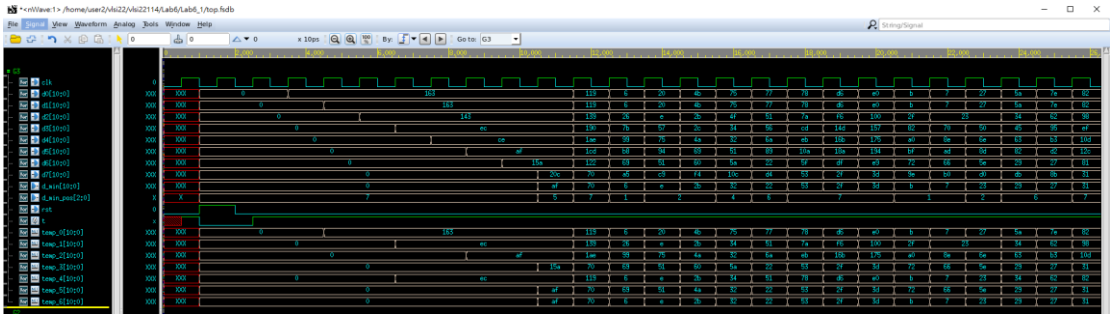
Writing to MAN (wen0, wen1... sequentially activated): The write enable signals for MAN (wen0, wen1, etc.) are activated sequentially over 8 clock cycles. This means 64 codebook values (8 sets of 8 values) are read from RAM1 and stored in MAN.

MAN Address Tracking (MAN_A_WEIGHT): The MAN_A_WEIGHT signal tracks which set of 8 weights is currently being written into MAN. It cycles through 8 values, corresponding to the 8 MAN units.

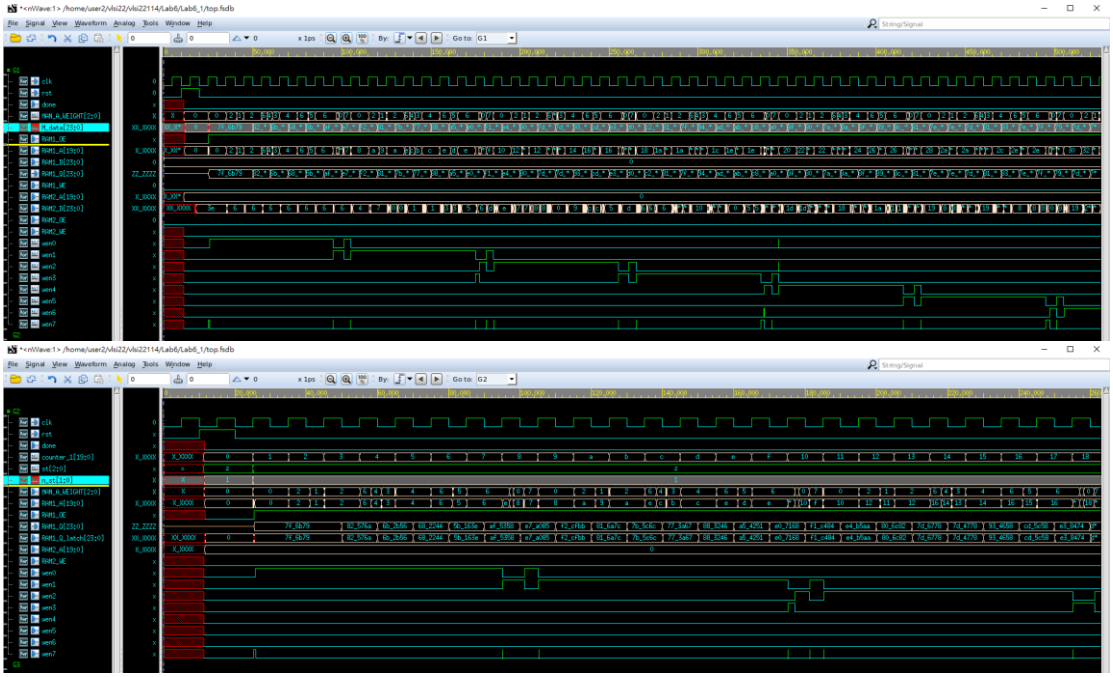
Writing to RAM2 (RAM2_WE pulled high): After the 8 MAN units are filled, the RAM2 write enable (RAM2_WE) is activated. This indicates that subsequent data read from RAM1 and processed by MAN and MIN will be written into RAM2.

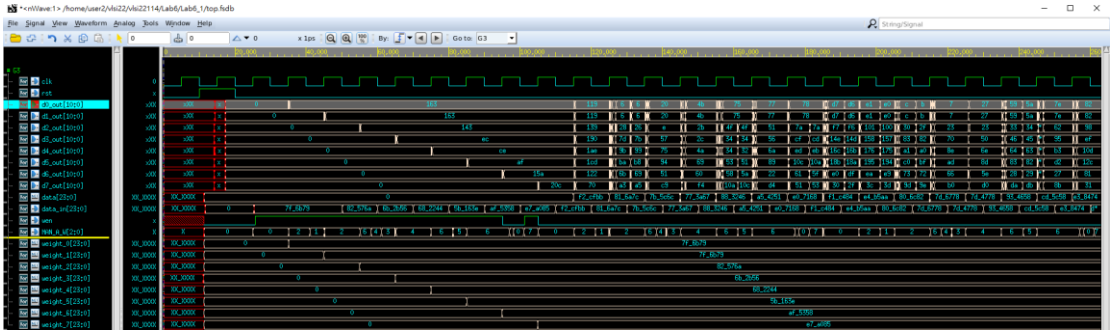


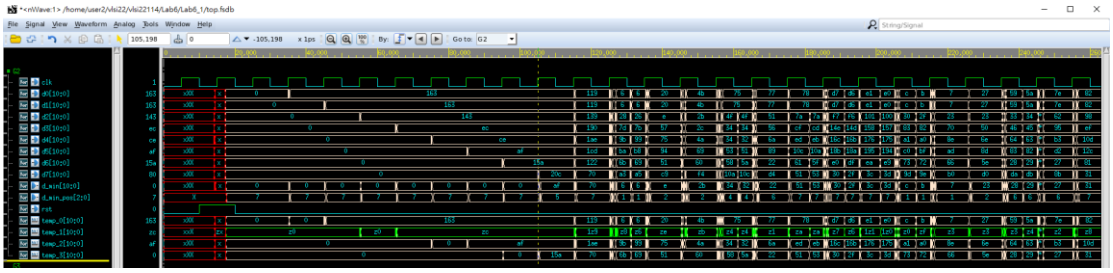
Weight Read (wen = 1): While the write enable signal (wen) is high, weights weight0 through weight7 are sequentially read and stored in MAN.



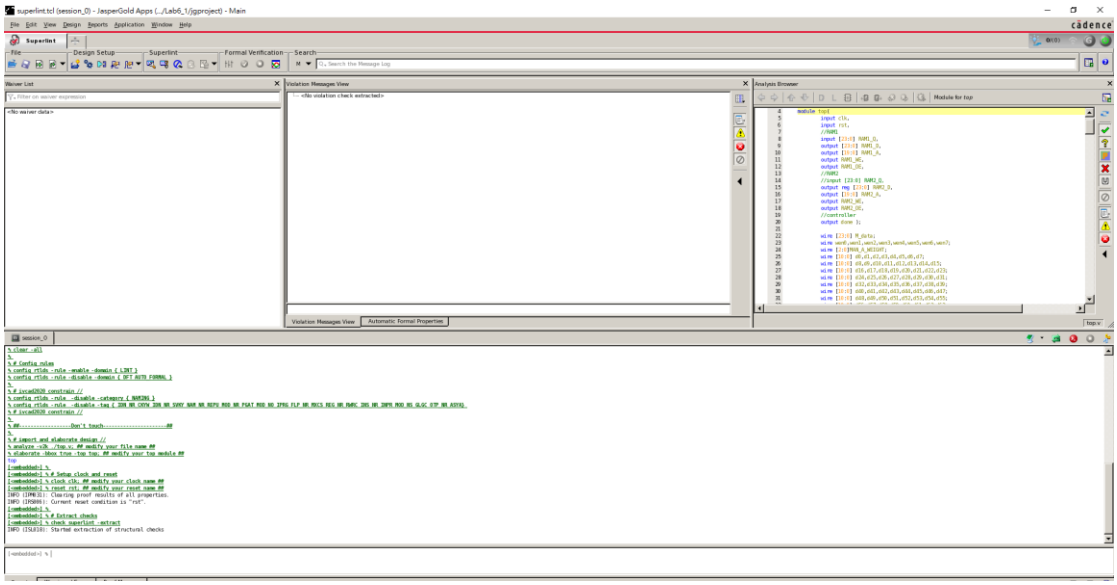
Input data (d0 to d7) are processed along with the weights stored in MAN. The results of the computations are stored in d0_out through d7_out.
The values d0_out through d7_out are fed into a circuit that calculates the minimum value and its position. The final result (the minimum value) is output to d_min, and its position (index) is output to d_min_pos.
(After synthesis)







5) Show SuperLint coverage (top.v)



Coverage = 1

6) Your clock period, total cell area, post simulation time (top.v)

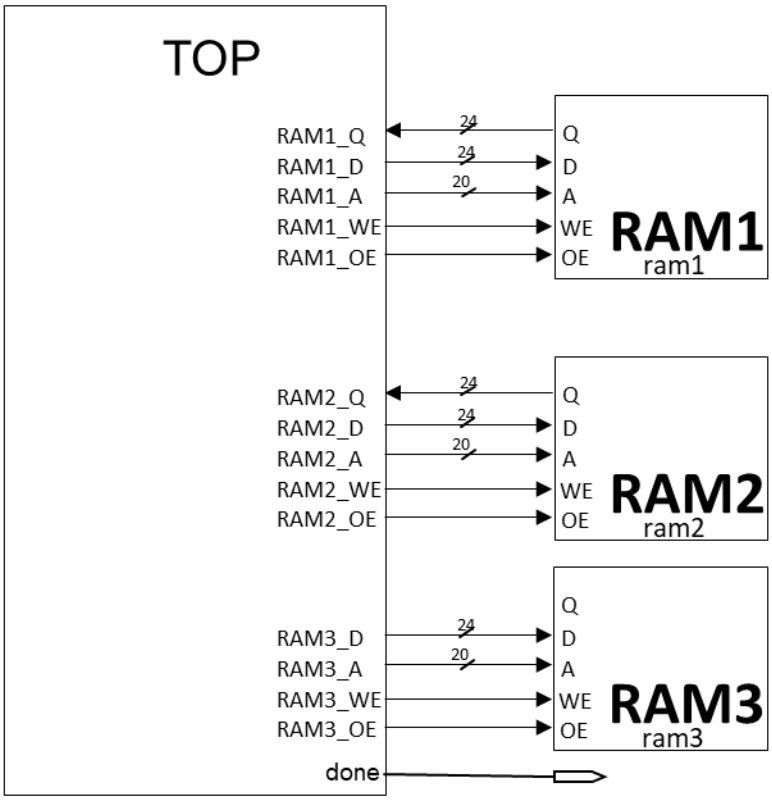
Clock period:10

Total cell area:11551.192688

Post simulation time:41645

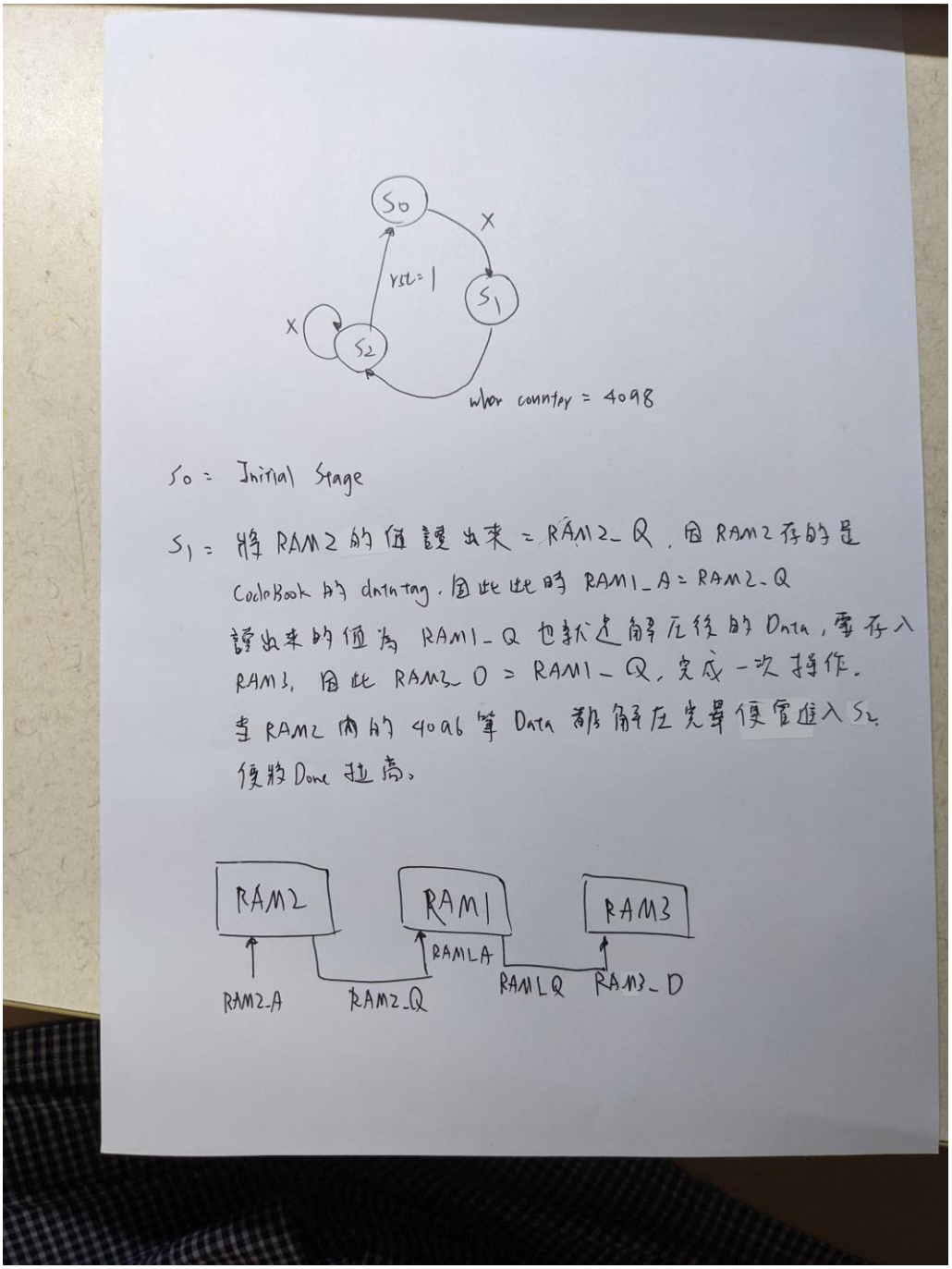
Lab 6_2: decompress the image

We can use code book and data tags to decompress the image



▲Block diagram for architecture(external)

- 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

➢ 

Complete the top module, in the system.

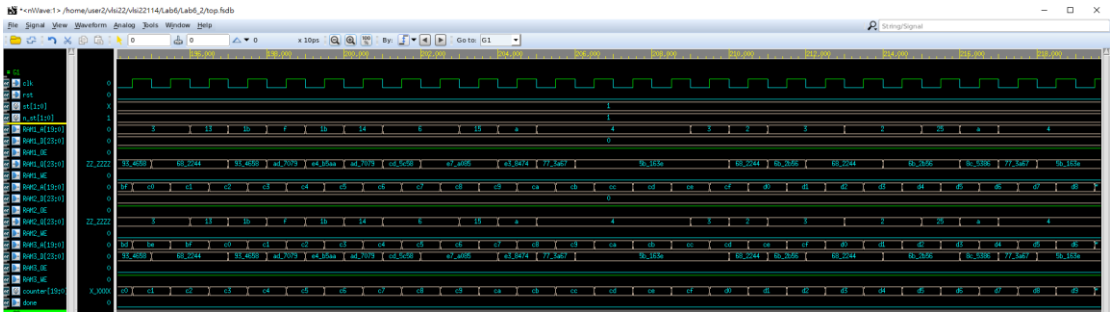
7) Compile the verilog code to verify the operations of this module works properly.

8) 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*.

9) Please **attach your waveforms** and **specify your operations** on the waveforms.

(Before synthesis)



Address Calculation (Counter): A counter is used to generate addresses for RAM2 (RAM2_A) and RAM1 (RAM1_A).

Enabling RAMs (RAM2_OE, RAM1_OE, RAM3_WE high): The output enables for RAM2 and RAM1 (RAM2_OE, RAM1_OE) and the write enable for RAM3 (RAM3_WE) are activated.

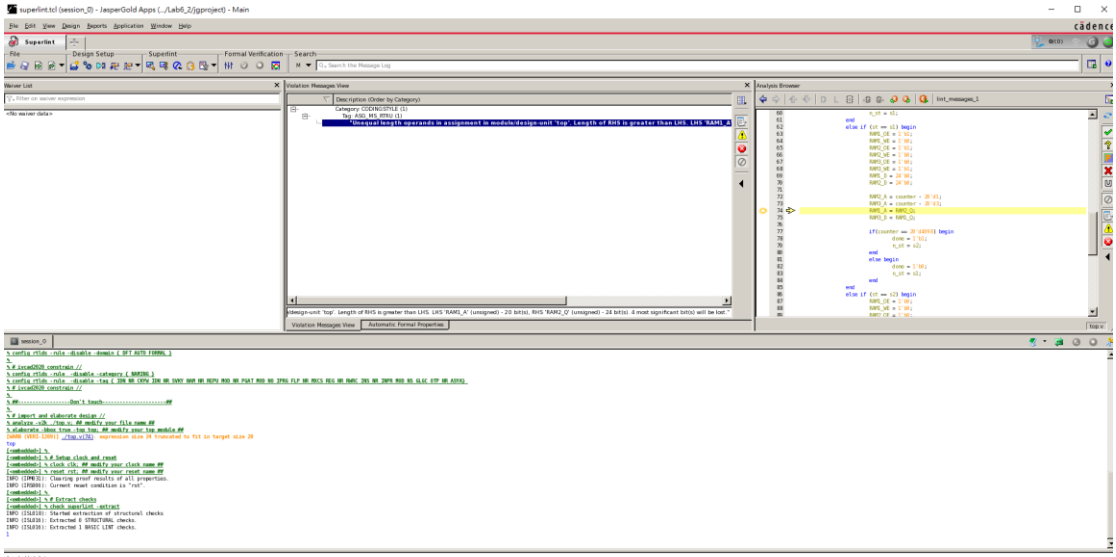
Reading from RAM2 and RAM1: The value from RAM2 is read (e.g., RAM2_Q = 13). This value is then used as the address for RAM1 (RAM1_A = 13). The value at this address in RAM1 is read (RAM1_Q, e.g., RAM1_Q = 68_2244).

Writing to RAM3: The value read from RAM1 (RAM1_Q) is written into RAM3 (RAM3_D = 68_2244).

Completion (done high): This process repeats until all values in RAM2 have been processed. Once complete, the done signal is activated.

(After synthesis)

10) Show SuperLint coverage (top.v)



Coverage = 106/107 = 0.99

11) Your clock period, total cell area, post simulation time (top.v)

clock period: 10ns

total cell area: 98.2128

post simulation time:41005

➢ Lessons learned from this lab

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_tb2.v +access+r +define+FSDB
Lab6_1(post)	ncverilog top_tb2.v +access+r +define+FSDB+syn
Lab6_2(pre)	ncverilog top_tb3.v +access+r +define+FSDB
Lab6_2(post)	ncverilog top_tb3.v +access+r +define+FSDB+syn