NMJ31804 – Principles of Computer Architecture

SEMESTER 2, 2021/22

LAB 3) RAM and ROM

Name: TAN YI JIE

Matric Number: 191020976

Program code: RK20 - Computer Engineering

1. Design of ROM

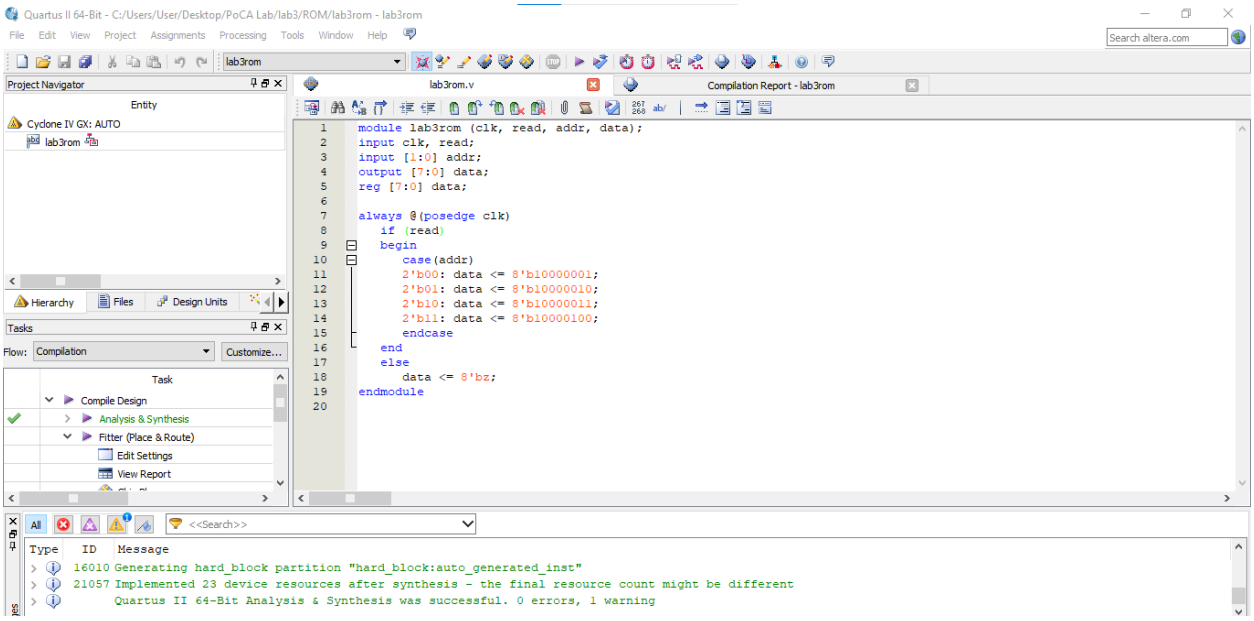

Figure 1: Verilog Code for ROM

Figure 1 shows that the compilation of the ROM module is successful.

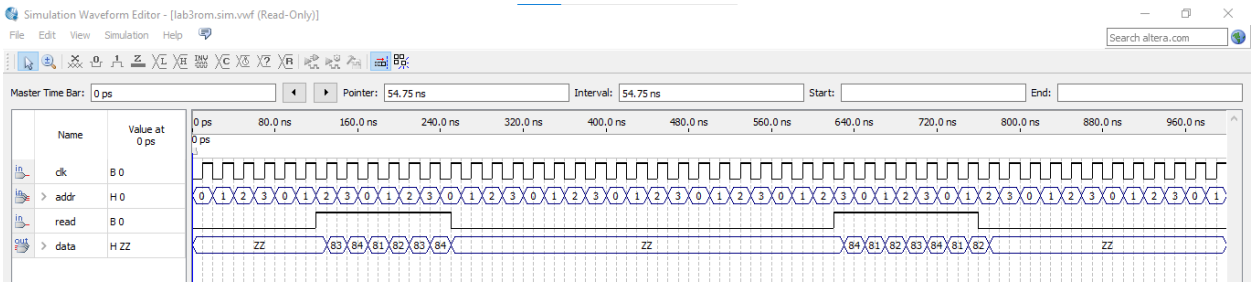

Figure 2: Simulation Result for ROM

Figure 2 shows the simulation result for the ROM module, the output data will have specific result in hexadecimal radix when the input of read is HIGH and will reflect it during each of the positive edge of the clock. Otherwise, the output data will show “ZZ”.

2. Design of RAM

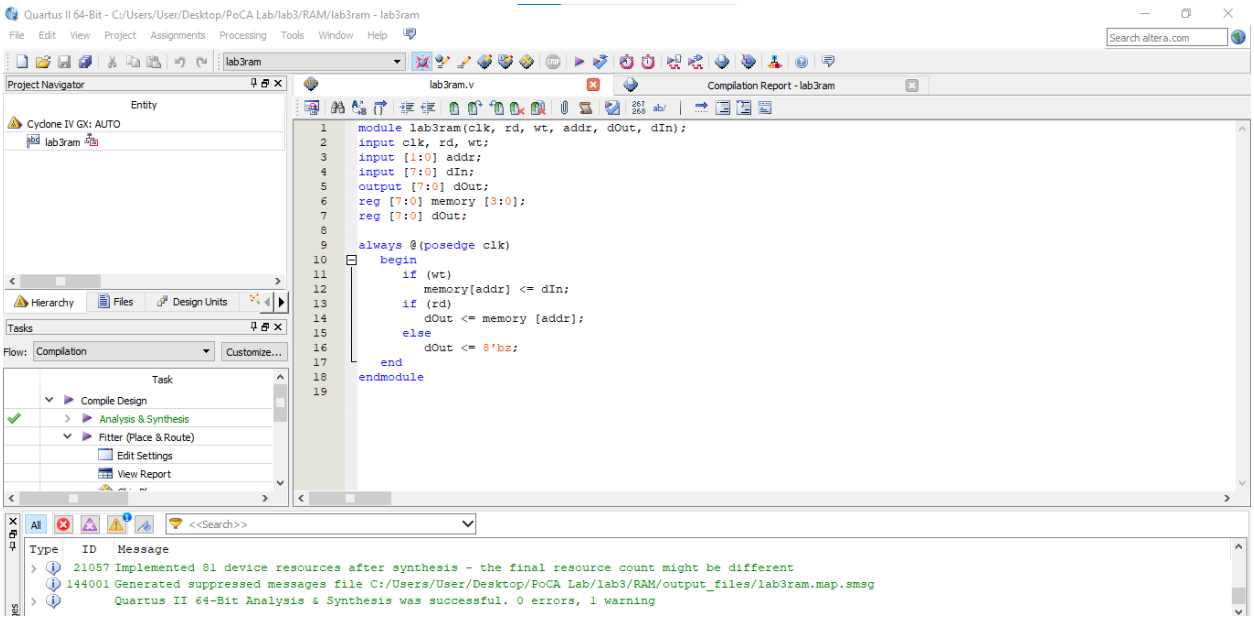

Figure 3: Verilog Code for RAM

Figure 3 shows that the compilation of the RAM module is successful.

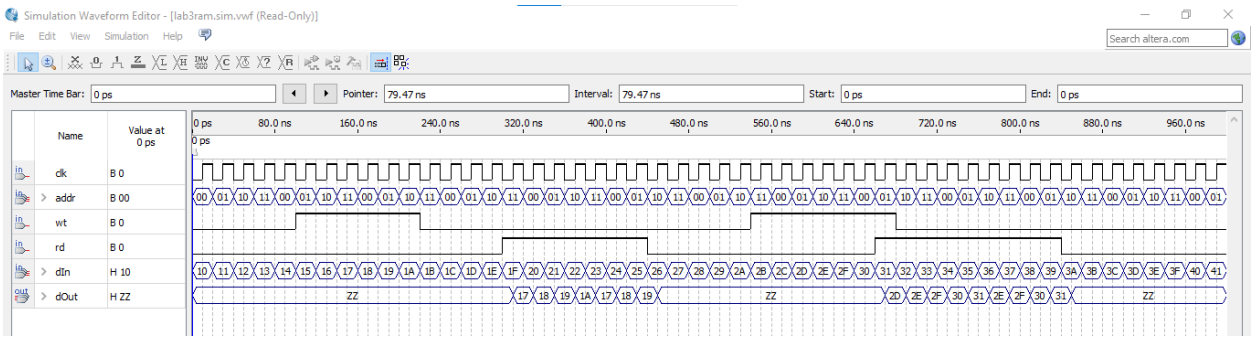

Figure 4: Simulation Result for RAM

Figure 4 shows the simulation result for the RAM module. For each cycle of positive edge of the clock (clk), the dOut will show the value that been write, wt and read, rd from dIn for the range of 2-bits only since the address input been assigned for 2-bits only. For example, the first dOut been write and read for the last 4 values which are 17, 18, 19, and 1A from the Din. Then the dOut will repeat the cycle of the 4 values until read (rd) been reached LOW. Otherwise, dOut will show “ZZ”.

3. Design of control unit to transfer data from ROM to RAM

lab3fsm.v | Compilation Report - lab3fsm

```
1  module lab3fsm(clk, rst, addr, rom, ram, rd);
2  input clk, rst;
3  output rom, ram, rd;
4  output [1:0] addr;
5  parameter [2:0] s1 = 3'b000, s2 = 3'b001, s3 = 3'b010, s4 = 3'b011, s5 = 3'b100, s6 = 3'b101, s7 = 3'b110, s8 = 3'b111;
6  reg [2:0] pre, next;
7  reg [1:0] addr;
8  reg ram, rom, rd;
9
10 always @(pre)
11   begin
12   case (pre)
13      s1: begin
14          next <= s2;
15          addr <= 2'b00;
16          rom <= 1'b1;
17          ram <= 1'b1;
18          rd <= 1'b0;
19          end
20
21      s2: begin
22          next <= s3;
23          addr <= 2'b01;
24          rom <= 1'b1;
25          ram <= 1'b1;
26          rd <= 1'b0;
27          end
28
29      s3: begin
30          next <= s4;
31          addr <= 2'b10;
32          rom <= 1'b1;
33          ram <= 1'b1;
34          rd <= 1'b0;
35          end
36
37      s4: begin
38          next <= s5;
39          addr <= 2'b11;
40          rom <= 1'b1;
41          ram <= 1'b1;
42          rd <= 1'b0;
43          end
44
45      s5: begin
46          next <= s6;
47          addr <= 2'b00;
48          rom <= 1'b0;
49          ram <= 1'b0;
50          rd <= 1'b1;
51          end
52
53      s6: begin
54          next <= s7;
55          addr <= 2'b01;
56          rom <= 1'b0;
57          ram <= 1'b0;
58          rd <= 1'b1;
59          end
60
61      s7: begin
62          next <= s8;
63          addr <= 2'b01;
64          rom <= 1'b0;
65          ram <= 1'b0;
66          rd <= 1'b1;
67          end
68
69      s8: begin
70          next <= s1;
71          addr <= 2'b01;
72          rom <= 1'b0;
73          ram <= 1'b0;
74          rd <= 1'b1;
75          end
76   endcase
77   end
78
79 always @(posedge clk)
80   begin
81   if (rst == 0)
```

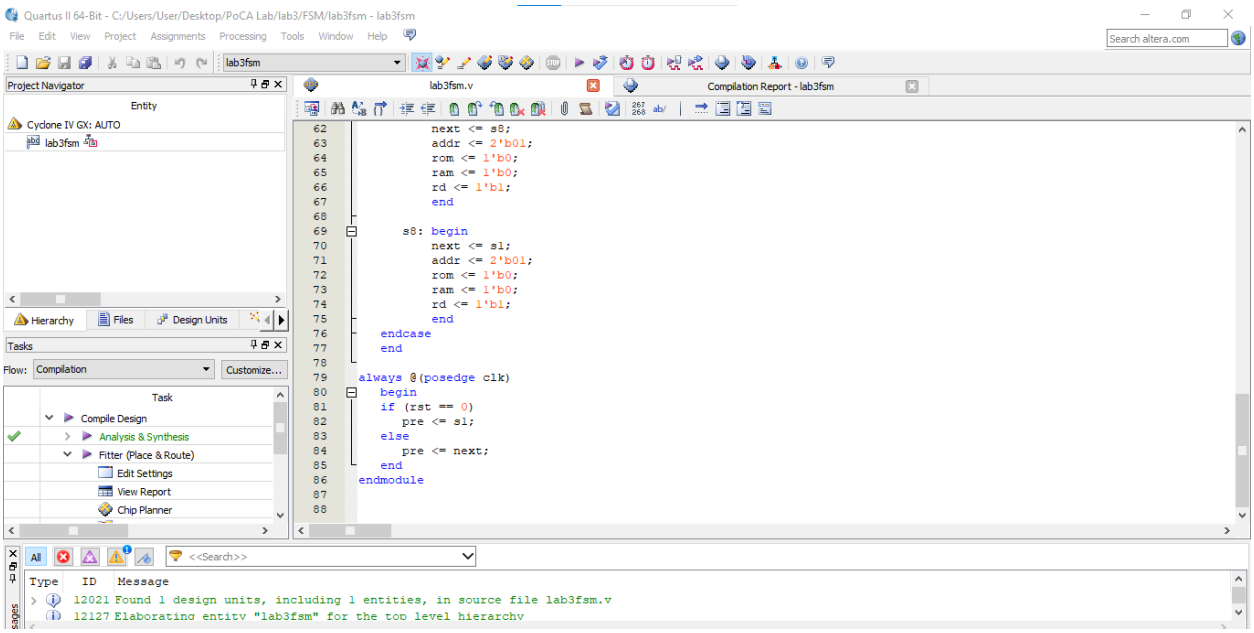

Figure 5: Verilog Code for lab3fsm Module

Figure 5 shows that the compilation of the lab3fsm module is successful.

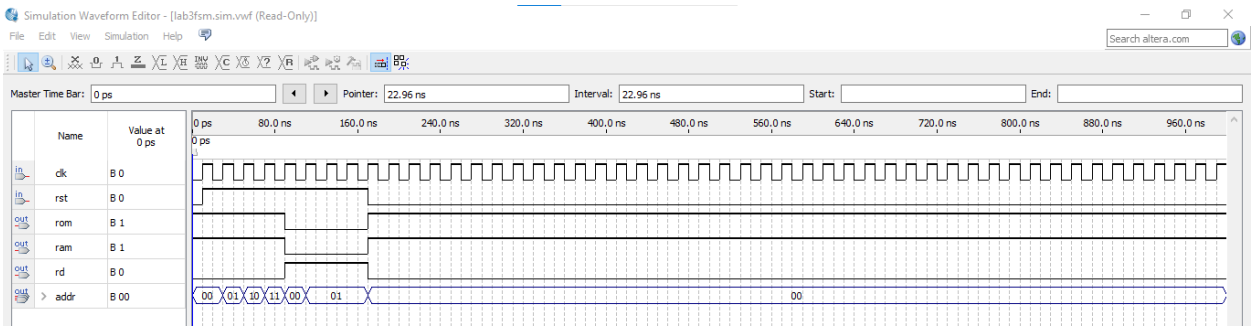

Figure 6: Simulation Result for lab3fsm Module

Figure 6 shows that the simulation result for the lab3fsm Module. For each cycle of positive edge of the clock (clk), when the reset (rst) is HIGH, the rom and ram will have the value of HIGH for 4 cycles of clock then invert to LOW, whereas the result read (rd) is invert of it. During that period, the address (addr) will show the result of state 1 until state 4 which are 00, 01, 10, and 11, then it continues with the result of state 5 until state 8. If reset is LOW, then all output will have the result of state 1.