

## Lab15

實驗主題:Memory

實驗日期:2022/12/12

學號姓名:B103040009 尹信淳

實驗內容: Design and Verify the following circuits using Verilog HDL:

4 x 4 RAM (Structural modeling)

8 x 8 Register File (Behavioral modeling)

### Exercise1: 4x4 RAM

Design and verify the 4 x 4 RAM using Verilog HDL Structural modeling

Code:

```
module decoder(decoder_out, binary_in, enable);
    input[1:0] binary_in;
    input enable;
    output[3:0] decoder_out;
    wire[3:0] decoder_out;
    assign decoder_out = (enable) ? (1 << binary_in) : 4'b0;
endmodule

module SR(Q,Q_b,S,R);
    output Q,Q_b;
    input S,R;
    reg Q,Q_b;
    wire S,R;
    always@(S|R)
        begin
            Q = ~(R|Q_b);
            Q_b = ~(S|Q);
        end
endmodule

module Cell(out,in,select,RW);
    output out;
    input in,select,RW;
    wire n1,n2,Q,Q_b;
    and (n1, select, ~RW, in);
    and (n2, select, ~RW, ~in);
    SR S1(Q,Q_b,n1,n2);
    and (out, select, Q, RW);
endmodule

module RAM(data_out, data_in, binary_in, RW, enable);
    output[3:0] data_out;
    input[3:0] data_in;
    input[1:0] binary_in;
    input RW,enable;
    wire[3:0] word,out1,out2,out3,out4;
    decoder D1(word, binary_in, enable);
    Cell C1(out1[0],data_in[0],word[0],RW);
    Cell C2(out1[1],data_in[1],word[0],RW);
    Cell C3(out1[2],data_in[2],word[0],RW);
    Cell C4(out1[3],data_in[3],word[0],RW);
    Cell C5(out2[0],data_in[0],word[1],RW);
    Cell C6(out2[1],data_in[1],word[1],RW);
    Cell C7(out2[2],data_in[2],word[1],RW);
    Cell C8(out2[3],data_in[3],word[1],RW);
    Cell C9(out3[0],data_in[0],word[2],RW);
    Cell C10(out3[1],data_in[1],word[2],RW);
    Cell C11(out3[2],data_in[2],word[2],RW);
    Cell C12(out3[3],data_in[3],word[2],RW);
    Cell C13(out4[0],data_in[0],word[3],RW);
    Cell C14(out4[1],data_in[1],word[3],RW);
    Cell C15(out4[2],data_in[2],word[3],RW);
    Cell C16(out4[3],data_in[3],word[3],RW);
    or(data_out[0],out1[0],out2[0],out3[0],out4[0]);
    or(data_out[1],out1[1],out2[1],out3[1],out4[1]);
    or(data_out[2],out1[2],out2[2],out3[2],out4[2]);
    or(data_out[3],out1[3],out2[3],out3[3],out4[3]);
endmodule
```

Testbench:

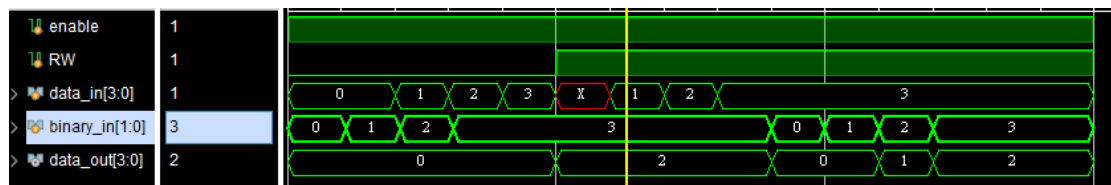
```
module tb;
  reg enable, RW;
  reg [3:0] data_in;
  reg [1:0] binary_in;
  wire [3:0] data_out;
  RAM UUT(.data_out(data_out),.data_in(data_in),.binary_in(binary_in),.RW(RW),.enable(enable));
  initial #150 $finish;
  initial begin

    enable = 1;
    RW = 0;
    binary_in = 2'b00;
    data_in = 4'b0000;

    #10 RW=1'b0; //Write
    #1 binary_in =2'b01; #9data_in =4'b0001;
    #1 binary_in =2'b10; #9data_in =4'b0010;
    #1 binary_in =2'b11; #9data_in =4'b0011;
    #10 RW=1'b1; data_in =4'bxxxx; //Read
    #10 data_in =4'b0001;
    #10 data_in =4'b0010;
    #10 data_in =4'b0011;
    #10 binary_in =2'b00;
    #10 binary_in =2'b01;
    #10 binary_in =2'b10;
    #10 binary_in =2'b11;

  end
endmodule
```

波形圖:



Exercise1 實驗過程描述:

這題首先要寫一個 2x4 的 decoder，我選擇用 dataflow modeling 的方式寫，因為以前練習過，所以這部分還算順利；接著要寫 RAM 的記憶元件，我最早用了 structural modeling 來寫，包括當中的 SR latch 也是用 structural。但等到我把寫好的 decoder 跟 16 個記憶元件接成 RAM，然後用 testbench 測試時就遇到了問題:這個 RAM 在做 read 的時候，讀到的永遠都是 4b'0011。一開始我認為是我的 testbench 寫錯，但因為這題的 testbench 是改寫老師提供的 tb 所以沒找到我認為是錯的地方(至少我認為該改寫的地方都改寫了)，接著我就回到 code 那邊看是不是我 structure 接錯了，但整個 code 檢查了至少四次之後還是找不到有哪邊接錯，最後在助教的幫忙下，把 SR latch 的程式碼改成 behavioral modeling 以及對 testbench 做了點調整之後，出來的波形才有正常些。這也許是老師允許我們用 behavioral modeling 寫記憶元件的原因?但老師也只是說'可以用行為描述，所以我還是蠻不解的，很失望。

---

## Exercise2: 8 x 8 Register File

Design and verify the 8 x 8 register file with 2 read ports and 1 write port using Verilog HDL Behavioral modeling

### Exercise2 實驗過程描述:

因為在第一題花了太多時間，因此換到這題的時候已經 11:50，到前面登記之後就下課了。

---

### 實驗心得

這次花了太多太多時間在第一題上，且到最後也不太清楚是卡住的原因，還蠻懊惱的，但有始有終、下禮拜的實驗再接再厲就好。我相信在我卡住，不斷來回檢查程式碼的時候，我對 RAM 的架構有變得更熟悉。即使卡住很久，但在過程中也是有所收穫的。而第二題 **register file** 的部分我可能要再自己花時間去了解他的架構以及運作的原理了。