

111-1 535313

進階可程式邏輯系統設計與應用

Advanced Programmable Logic System
Design and Application

實驗編號：LAB 05

Memory Blocks

實驗名稱：Substitute dual-port
memory by single-port memory

結報完成日期：2022.11.11

姓名：王語

系級：機械四

學號：0811127

一、實驗目的

了解FPGA中Memory blocks的使用以及功能差異，藉由加入Quartus II 現有RAM IP 練習相關操作，熟悉運作原理。

二、Verilog 程式碼

```
1
2 module Lab05_0811127 ( input CLOCK_50 ,
3                         input [9:0] SW ,
4                         output [0:0] LEDR ,
5                         output [6:0] HEX0 , HEX2 , HEX3 );
6 wire [3:0] _HEX0 , _HEX1 , _HEX2 , _HEX3 ;
7
8 // 32 X 8 RAM
9 one_port_RAM ( .address( SW [4:0] ),
10               .clock( CLOCK_50 ),
11               .data( SW [8:5] ),
12               .wren( SW [9] ),
13               .q( _HEX0 ) );
14 // input [4:0] address;
15 // input clock;
16 // input [7:0] data;
17 // input wren;
18 // output [7:0] q;
19 ssd ssd_0 ( .Din( _HEX0 ),
20             .Dout( HEX0 ) );
21 // ssd ssd_1 ( .Din( _HEX1 ),
22 //             .Dout( HEX1 ) );
23 ssd ssd_2 ( .Din( _HEX2 ),
24             .Dout( HEX2 ) );
25 ssd ssd_3 ( .Din( _HEX3 ),
26             .Dout( HEX3 ) );
27
28 assign _HEX2 = SW [3:0] ;
29 assign _HEX3 = {3'b000,SW[4]} ;
30 assign LEDR [0] = SW[9];
31
32 endmodule
```

```

35 // sub module
36 // seven segment display
37 module ssd (    input [3:0] Din,
38                output [6:0] Dout );
39 assign Dout[0] = ((!Din[3])&(!Din[2])&(!Din[1])& (Din[0]))|
40                ((!Din[3])& (Din[2])&(!Din[1])&(!Din[0]))|
41                ((Din[3])& (Din[2])&(!Din[1])& (Din[0]))|
42                ((Din[3])&(!Din[2])& (Din[1])& (Din[0]));
43
44 assign Dout[1] = ((!Din[3])& (Din[2])&(!Din[1])& (Din[0]))|
45                ( (Din[2])& (Din[1])&(!Din[0]))|
46                ((Din[3])& (Din[1])& (Din[0]))|
47                ((Din[3])& (Din[2])& (!Din[0]));
48
49 assign Dout[2] = ((!Din[3])&(!Din[2])& (Din[1])&(!Din[0]))|
50                ((Din[3])& (Din[2])& (Din[1]) )|
51                ((Din[3])& (Din[2])& (!Din[0]));
52
53
54 assign Dout[3] = ((!Din[3])&(!Din[2])&(!Din[1])& (Din[0]))|
55                ((!Din[3])& (Din[2])&(!Din[1])&(!Din[0]))|
56                ((Din[3])&(!Din[2])& (Din[1])&(!Din[0]))|
57                ( (Din[2])& (Din[1])& (Din[0]));
58
59 assign Dout[4] = ((!Din[3])& (!Din[0]))|
60                ((!Din[2])&(!Din[1])& (Din[0]))|
61                ((!Din[3])& (Din[2])&(!Din[1]) );
62
63
64 assign Dout[5] = ((Din[3])& (Din[2])&(!Din[1])& (Din[0]))|
65                ((!Din[3])&(!Din[2])& (Din[0]))|
66                ((!Din[3])&(!Din[2])& (Din[1]) )|
67                ((!Din[3])& (Din[1])& (Din[0]));
68
69 assign Dout[6] = ((!Din[3])& (Din[2])& (Din[1])& (Din[0]))|
70                ((Din[3])& (Din[2])&(!Din[1])&(!Din[0]))|
71                ((!Din[3])&(!Din[2])&(!Din[1]) );
72
73 endmodule

```

三、 實驗結果



燒錄中

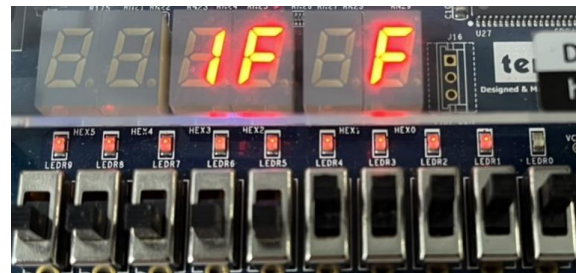


燒錄完成

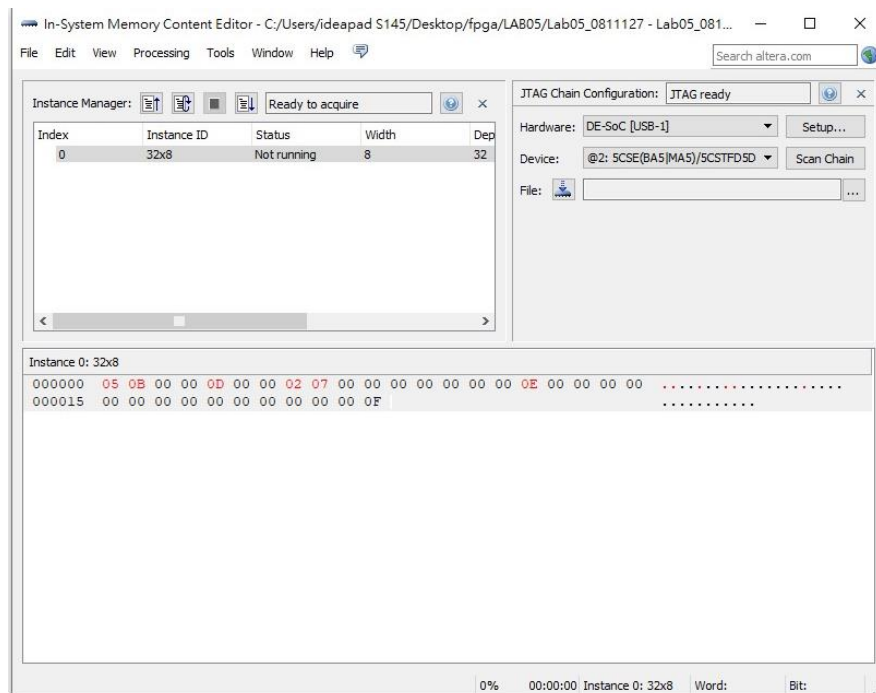
SW[9] = 1 (寫入數值)



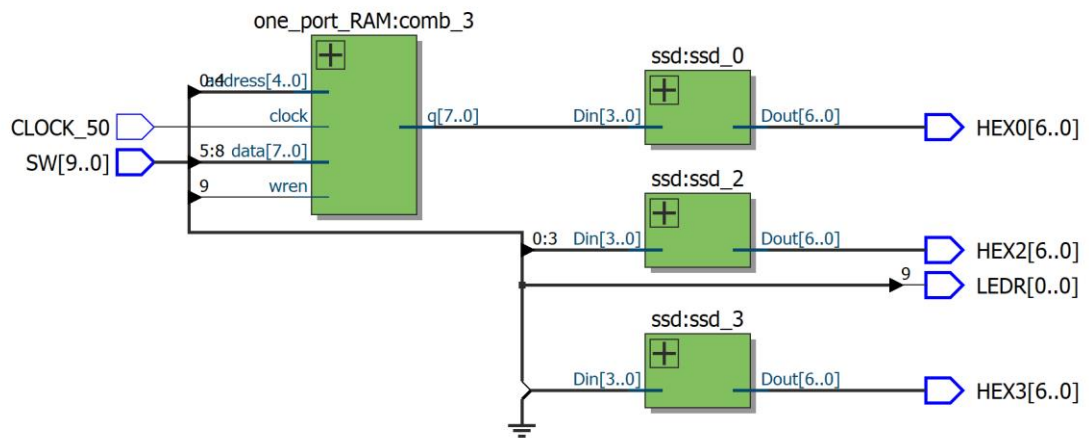
SW[9] = 0 (不寫入數值，查看該地址存取數值)



於 Quartus 上展示 RAM 內之值

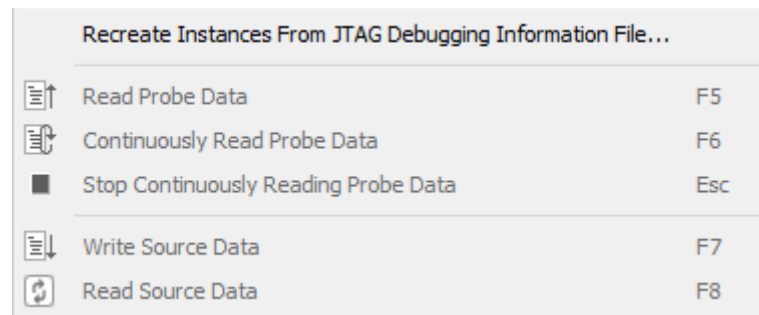


四、RTL



五、問題與討論

這次實驗卡關最久的地方在設定，由於沒有設定好，在 In-System Memory Content Editor 視窗一值無法讀取 RAM 內的數值。



首先是 Allow In-System Memory Content Editor 找不到 CYCLONE_SAFE_WRITE，看了半天才知道那個設定原來要自己輸入。

其次，在 In-System Memory Content Editor 視窗內也找不到 USB-Blaster 的裝置，但經過裝置管理員一連串自動偵測與設置後，成功以 USB 連接 DE1-SOC。

而程式部分需要多注意 IP 的導入，要記得加入在專案內，這樣 module 才能順利被使用。

六、心得

這次實驗很簡單，因為先前已經熟悉 IP 的導入，因此在程式中加入現有 single-port RAM 易如反掌，而這次實驗也就這樣順利完成了。