第一次實驗報告

題目：

Add two 1-digit BCD numbers

姓名：黃子軒

學號：**310581003**

繳交日期：**111** 年 **10** 月 **15** 日

1. 實驗目的

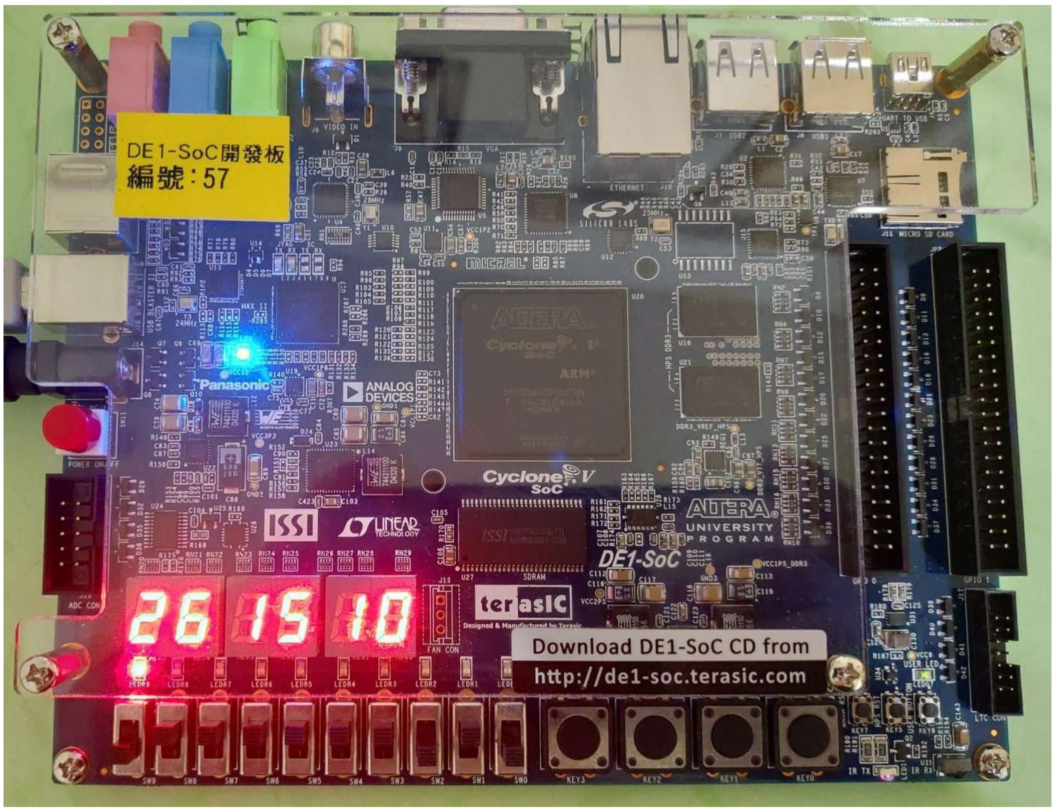
透過本次實驗，除了熟悉如何撰寫 verilog 並實現於 FPGA 板，更學習如何使用 Structure Description 和 Data Description 以實現電路設計。而本實驗內容，即嘗試以 switch、7-segment decoders、mux、4-bit ripple-carry adder、LED 設計並實現一簡易的加法計算器，以了解硬體電路設計流程。

2. 實驗程式碼

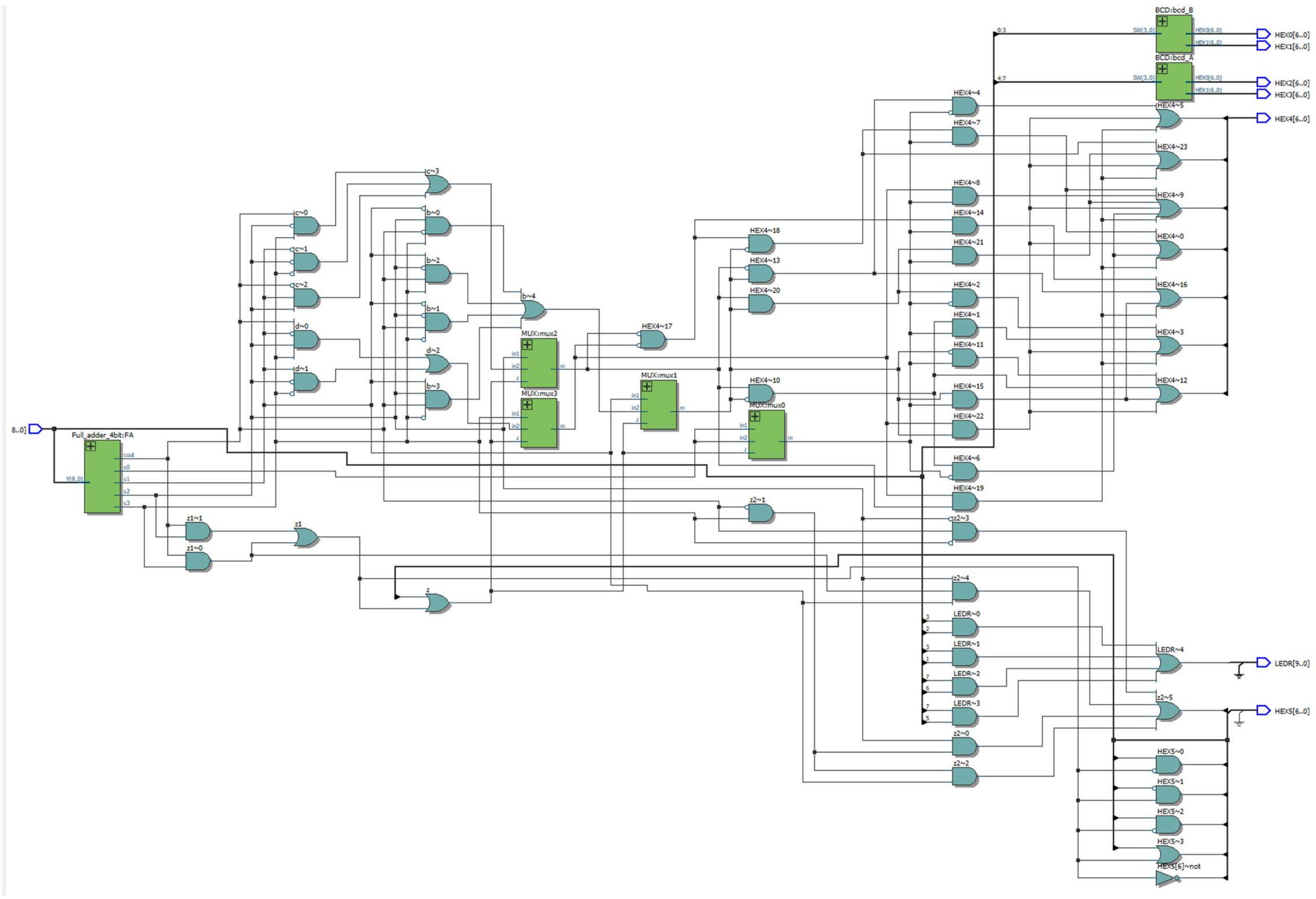
```
module Lab1_part7(SW, HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, LEDR);

input [8:0] SW;

output [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;

output [9:0] LEDR;

wire s0, s1, s2, s3, cout, z1, z2;


BCD bcd_A(SW[7:4], HEX2, HEX3);

BCD bcd_B(SW[3:0], HEX0, HEX1);


Full_adder_4bit FA(SW, s0, s1, s2, s3, cout);

assign LEDR[9]= (SW[3]&SW[2]|SW[3]&SW[1])|(SW[7]&SW[6]|SW[7]&SW[5]);


assign z1 = (cout&s3)|(cout&s2);

assign z2 = (~cout&s3&s2)|(~cout&s3&s1)|(cout&~s3&~s2)|(cout&s3&s2&s1);

assign z = z1|z2;


assign HEX5[0]=~z1&z2;

assign HEX5[1]=0;

assign HEX5[2]=z1&~z2;

assign HEX5[3]=~z1&z2;

assign HEX5[4]=z2;

assign HEX5[5]=z1|z2;

assign HEX5[6]=~z1;


assign a = s0;

assign b = (~cout&s3&s2&~s1)|(cout&~s3&~s2&~s1)|(cout&s3&~s2&s1)|(cout&~s3&s2&s1);

assign c = (cout&s3&~s2)|(~s3&~s2&~s1)|(~cout&s2&s1);
assign d = (cout&s3&s2&~s1)|(~s3&~s2&s1);


MUX mux3(m3, s3, d, z);

MUX mux2(m2, s2, c, z);

MUX mux1(m1, s1, b, z);

MUX mux0(m0, s0, a, z);


assign HEX4[0] = (~m3&~m2&~m1&m0)|(m2&~m1&~m0)|(m3&m2)|(m3&m1);

assign HEX4[1] = (m2&~m1&m0)|(m2&m1&~m0)|(m3&m2)|(m3&m1);

assign HEX4[2] = (m3&m2)|(m3&m1)|(~m2&m1&~m0);

assign HEX4[3] =
(m2&~m1&~m0)|(m2&m1&m0)|(m3&m1)|(~m3&~m2&~m1&m0)|(m3&m2)|(m3&m0);

assign HEX4[4] = (m2&~m1)|(m3&m1)|(~m1&m0)|(m1&m0);

assign HEX4[5] = (m3&m2)|(~m2&m1)|(~m3&~m2&m0)|(m1&m0);

assign HEX4[6] = (~m3&~m2&~m1)|(m3&m2)|(m2&m1&m0)|(m3&m1);

endmodule


module BCD(SW, HEX0, HEX1);

output [6:0]HEX0;

output [6:0]HEX1;

input [3:0]SW;

wire z, m0, m1, m2, m3, a, b, c;


assign v0 = SW[0];

assign v1 = SW[1];

assign v2 = SW[2];

assign v3 = SW[3];


Comparator comparator(z, v3, v2, v1, v0);

CircuitA circuitA(a, b, c, v2, v1, v0);


assign HEX1[0] = 1;
assign HEX1[1] = ~z;

assign HEX1[2] = ~z;

assign HEX1[3] = 1;

assign HEX1[4] = 1;

assign HEX1[5] = 1;

assign HEX1[6] = 1;


MUX mux3(m3, v3, 0, z);

MUX mux2(m2, v2, c, z);

MUX mux1(m1, v1, b, z);

MUX mux0(m0, v0, a, z);


assign HEX0[0] = (~m3&~m2&~m1&m0)|(m2&~m1&~m0)|(m3&m2)|(m3&m1);

assign HEX0[1] = (m2&~m1&m0)|(m2&m1&~m0)|(m3&m2)|(m3&m1);

assign HEX0[2] = (m3&m2)|(m3&m1)|(~m2&m1&~m0);

assign HEX0[3] =
(m2&~m1&~m0)|(m2&m1&m0)|(m3&m1)|(~m3&~m2&~m1&m0)|(m3&m2)|(m3&m0);

assign HEX0[4] = (m2&~m1)|(m3&m1)|(~m1&m0)|(m1&m0);

assign HEX0[5] = (m3&m2)|(~m2&m1)|(~m3&~m2&m0)|(m1&m0);

assign HEX0[6] = (~m3&~m2&~m1)|(m3&m2)|(m2&m1&m0)|(m3&m1);

endmodule


module Full_adder_4bit(SW, s0, s1, s2, s3, cout);

input [8:0] SW;

output s0, s1, s2, s3;

output cout;

wire c1, c2, c3;


assign cin = SW[8];

assign a0 = SW[0];

assign a1 = SW[1];

assign a2 = SW[2];
assign a3 = SW[3];

assign b0 = SW[4];
assign b1 = SW[5];
assign b2 = SW[6];
assign b3 = SW[7];

Full_adder FA1(c1, s0, a0, b0, cin);
Full_adder FA2(c2, s1, a1, b1, c1);
Full_adder FA3(c3, s2, a2, b2, c2);
Full_adder FA4(cout, s3, a3, b3, c3);
endmodule

module MUX(m, in1, in2, z);
input in1, in2, z;
output m;
assign m=(~z&in1)|(z&in2);
endmodule

module Comparator(z, v3, v2, v1, v0);
input v3, v2, v1, v0;
output z;
assign z=(v2&v3)|(v3&v1);
endmodule

module CircuitA(a, b, c, v2, v1, v0);
input v2, v1, v0;
output a, b ,c;
assign a= v0;
assign b= ~v1;
assign c= v2&v1;
endmodule
```

```
module Full_adder(co, s, a, b, ci);
input a, b, ci;
output co, s;
wire d;
assign d=a^b;
assign s=ci^d;
assign co=(~d&b)|(d&ci);
endmodule
```

3. 實驗結果照片(optional)



4. RTL 佈局(optional)



5.問題與討論

本實驗除了原本實驗要求，更嘗試將計算機做得更完整，即讓計算結果能顯示“20”以上的數字，架構圖可以參考以上 RTL，其架構與原先實驗差不多，特別的是在 Comparator 輸出有兩個，分別為 z1 和 z2，並以這兩個 Bit 關係去操控計算結果的十位數，如下表。

十位數	Z1	Z2
0	0	0
1	0	1
2	1	0
3	1	1