第一次平時作業

班級：資工四甲

學號：4A1G0056

姓名：簡明旋

指導老師：陳福坤

程式碼：

FULLADD.V

```
module   FullAdd(a, b, carryin, sum, carryout);
input      a, b, carryin;
output    sum, carryout;
wire       sum, carryout;

  assign   {carryout, sum} = a + b + carryin;
endmodule
```

FULLADD4.V

```
module       FullAdd4(a, b, carry_in, sum, carry_out);
input          [3:0] a, b;
input                carry_in;
output         [3:0] sum;
output               carry_out;
wire           [3:0] sum;
wire                 carry_out;

  FullAdd   fa0(a[0], b[0], carry_in,   sum[0], carry_out1);
  FullAdd   fa1(a[1], b[1], carry_out1, sum[1], carry_out2);
  FullAdd   fa2(a[2], b[2], carry_out2, sum[2], carry_out3);
  FullAdd   fa3(a[3], b[3], carry_out3, sum[3], carry_out);
endmodule
```

FULLADD8.V

```
module       FullAdd8(a, b, carry_in, sum, carry_out);
input          [7:0] a, b;
input                carry_in;
output         [7:0] sum;
output               carry_out;
wire           [7:0] sum;
wire                 carry_out;

  FullAdd4   fa0(a[3:0], b[3:0], carry_in,   sum[3:0], carry_out1);
  FullAdd4   fa1(a[7:4], b[7:4], carry_out1, sum[7:4], carry_out);
endmodule
```

FULLADD8.TF

```
//`timescale 1ns/1ns

module t;

reg [7:0] a, b;
reg carry_in;
wire [7:0] sum;
wire carry_out;

     FullAdd8 m (.a(a), .b(b), .carry_in(carry_in), .sum(sum), .carry_out(carry_out));

     // Enter fixture code here
     initial
        begin
```

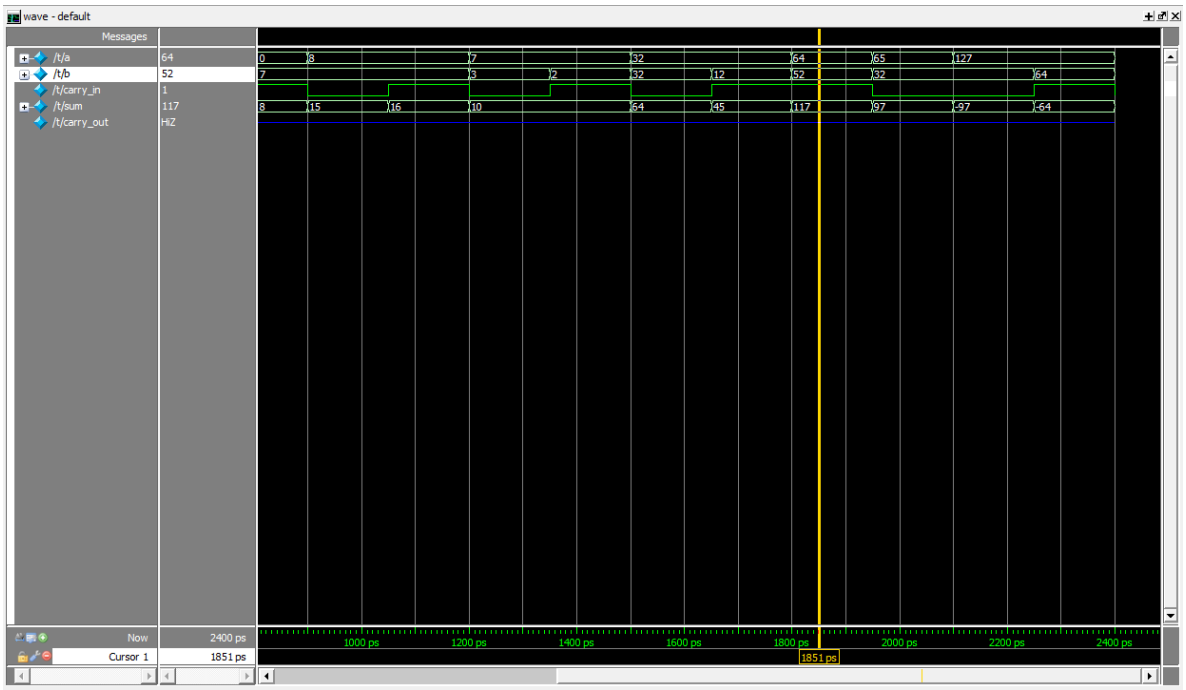
```
#0
        a        = 0;
    b        = 0;
    carry_in = 0;

#150
        a        = 0;
    b        = 0;
    carry_in = 1;

// -------------
#150
        a        = 0;
    b        = 15;
    carry_in = 0;

#150
        a        = 0;
    b        = 15;
    carry_in = 1;

// -------------
#150
        a        = 0;
    b        = 7;
    carry_in = 0;

#150
        a        = 0;
    b        = 7;
    carry_in = 1;

// -------------
#150
        a        = 8;
    b        = 7;
    carry_in = 0;

#150
        a        = 8;
    b        = 7;
    carry_in = 1;

// -------------
#150
        a        = 7;
    b        = 3;
    carry_in = 0;

#150
        a        = 7;
    b        = 2;
    carry_in = 1;

#150
        a        = 32;
    b        = 32;
    carry_in = 0;
```

```
    #150
        a        = 32;
      b        = 12;
      carry_in = 1;

    #150
        a        = 64;
      b        = 52;
      carry_in = 1;

    #150
        a        = 65;
      b        = 32;
      carry_in = 0;

    #150
        a        = 127;
      b        = 32;
      carry_in = 0;

    #150
        a        = 127;
      b        = 64;
      carry_in = 1;

    #150
        a        = 0;
      b        = 0;
      carry_in = 0;
  end

endmodule // t
```

run.do

```
quit -sim
--cd /FULLADD8
--do run.do

vlib work
vmap work work

vlog FULLADD.v
vlog FULLADD4.v
vlog FULLADD8.v

vlog FULLADD8.tf
vsim t

add wave *
--do wave_FULLADD8.do

run -all
```

執行結果：


wave - default
Messages
/t/a
/t/b
/t/carry_in
/t/sum
/t/carry_out
64
52
1
117
HiZ
0
8
7
32
64
65
127
7
3
2
32
12
52
32
64
8
15
16
10
64
45
117
97
97
64
Now
2400 ps
Cursor 1
1851 ps
1000 ps
1200 ps
1400 ps
1600 ps
1800 ps
2000 ps
2200 ps
2400 ps
1851 ps

心得：

第一次的課堂作業，使用課堂上所教授的 4 位元全加器，延伸製作出 8 位元的全加器，只需要使用兩顆 4 位元全加器並把電路做修改即可完成，最後修改模擬的數字及腳本，執行後就可以發現電路正確執行成功，並完成 8 位元的全加器。