

硬體描述語言設計與模擬

Homework #2

Title :7seg display

班級：電子二 B

學號：B10713138

姓名：蔡尚哲

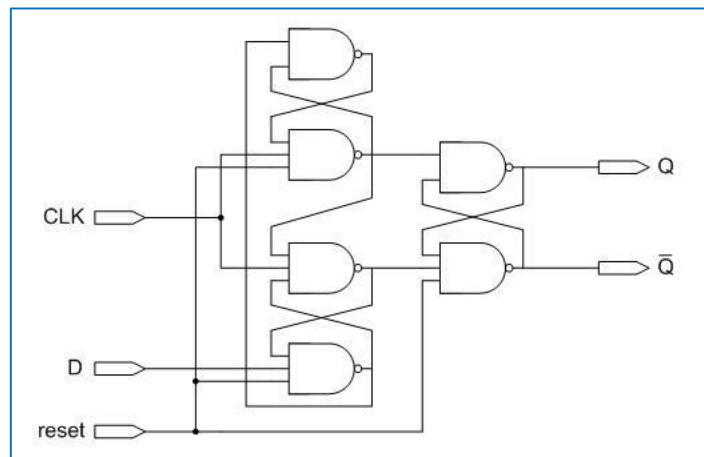
教師：許明華老師

繳交日期：2019/10/14

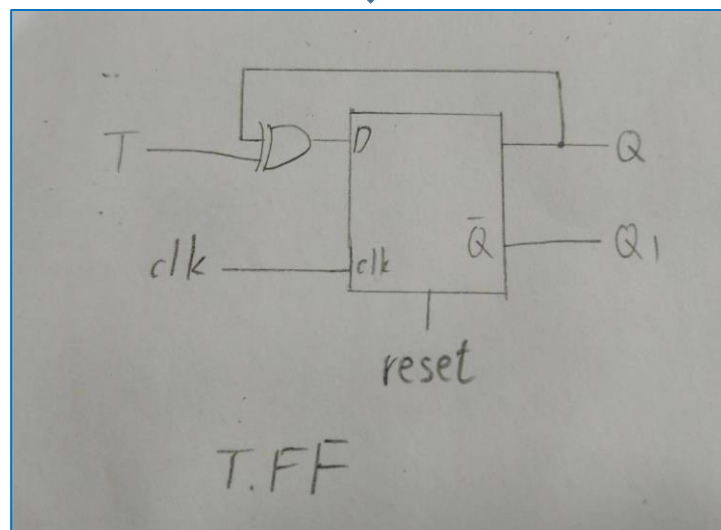
電路原理

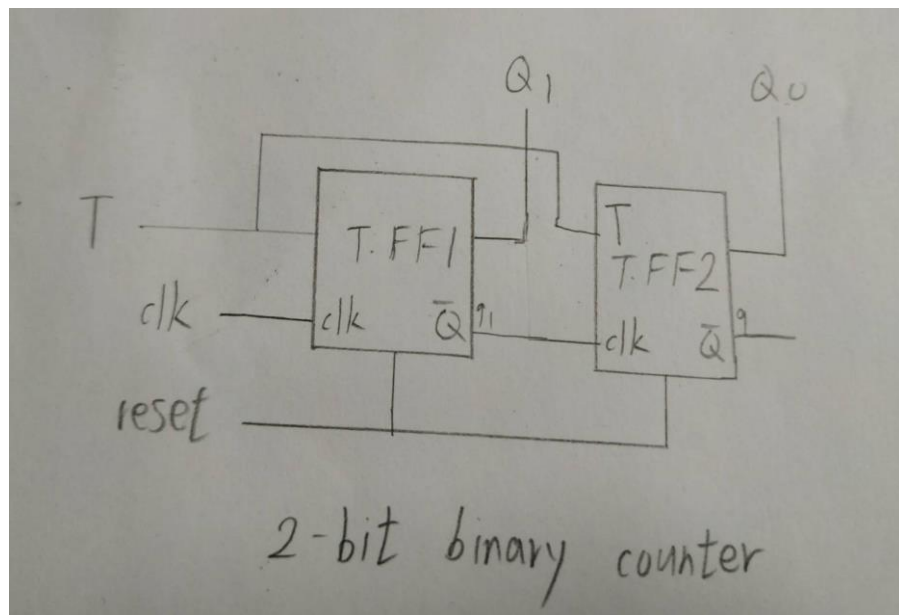
藉由 2bit binary counter 與 2-to-4 decoder 來輪流掃描，一定的頻率下，因視覺暫留的影響，會造成七段顯示器視覺上常亮的效果，而 4bits 4-to-1 MUX 用來選擇輸入的數字，再利用 BCD-to-segment decoder 把 BCD 碼轉成七段顯示器的字型碼，組合成一個 4 位元七段顯示器電路。

電路架構

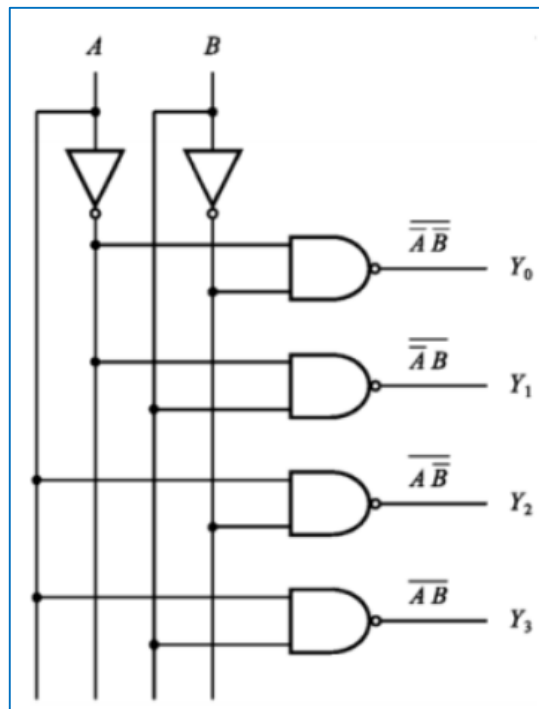


D.FF

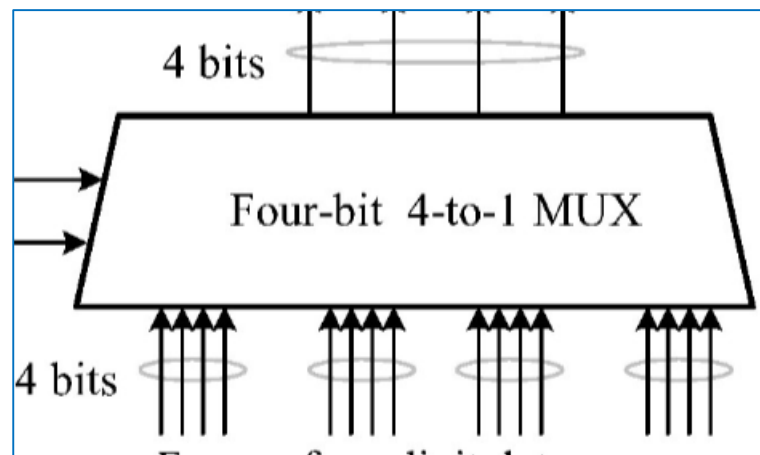




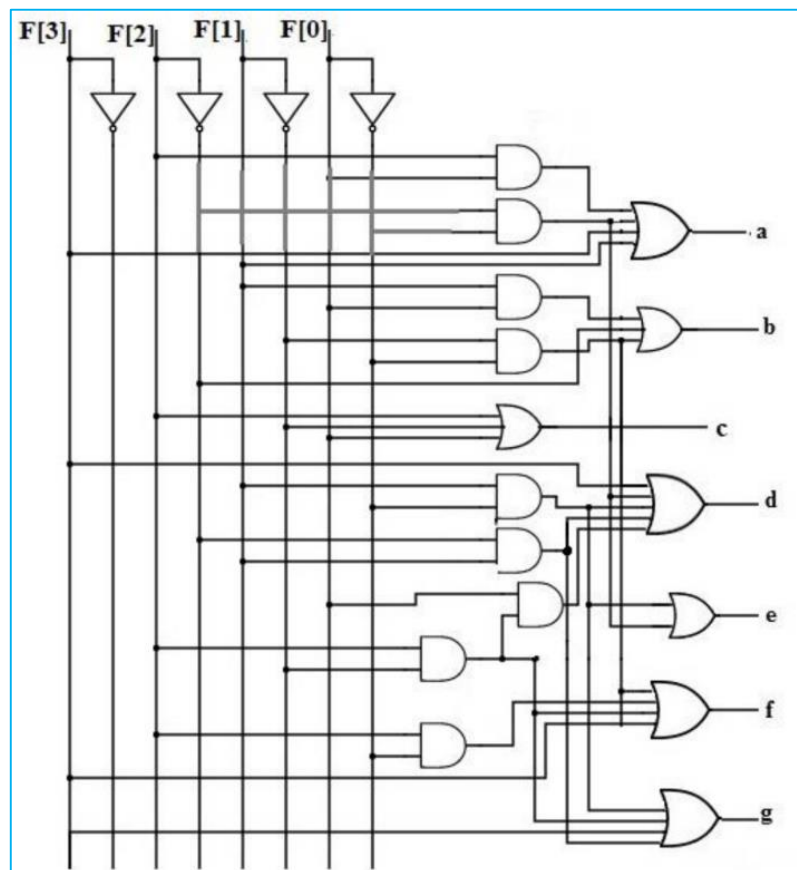
2-bit binary counter



2-to-4 decoder

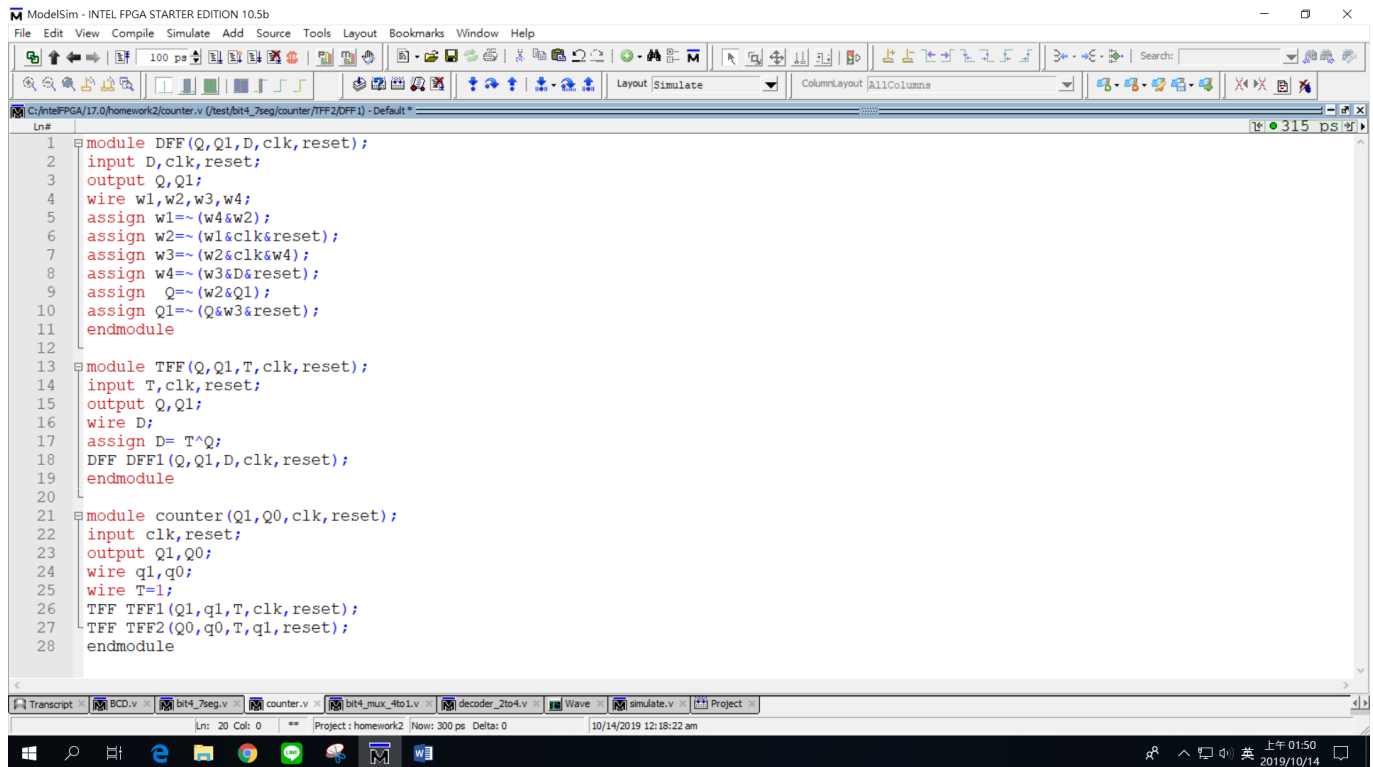


4bits 4-to-1 MUX



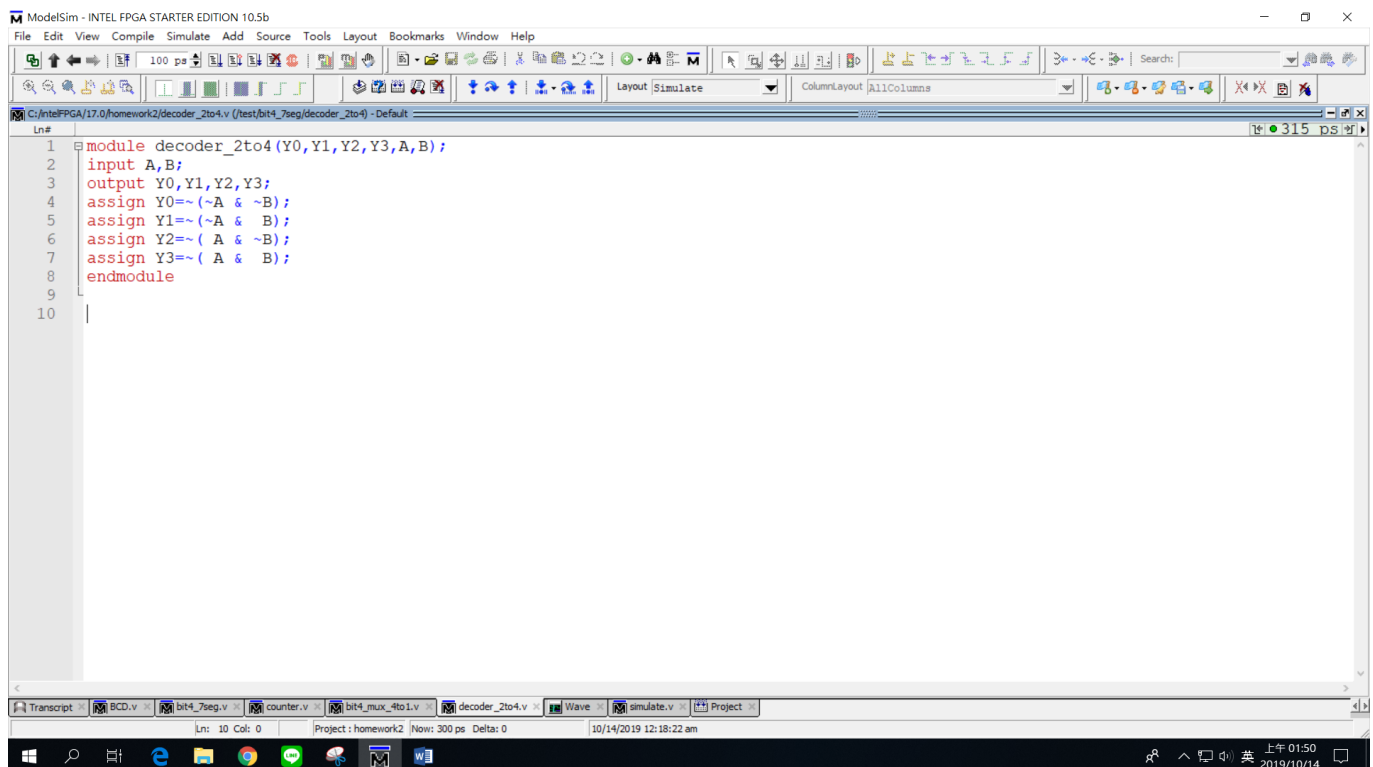
BCD-to-segment decoder

程式擷取畫面



```
1 module DFF(Q,Q1,D,clk,reset);
2   input D,clk,reset;
3   output Q,Q1;
4   wire w1,w2,w3,w4;
5   assign w1=~(w4&w2);
6   assign w2=~(w1&clk&reset);
7   assign w3=~(w2&clk&w4);
8   assign w4=~(w3&D&reset);
9   assign Q=~(w2&Q1);
10  assign Q1=~(Q&w3&reset);
11 endmodule
12
13 module TFF(Q,Q1,T,clk,reset);
14   input T,clk,reset;
15   output Q,Q1;
16   wire D;
17   assign D= T^Q;
18   DFF DFF1(Q,Q1,D,clk,reset);
19 endmodule
20
21 module counter(Q1,Q0,clk,reset);
22   input clk,reset;
23   output Q1,Q0;
24   wire q1,q0;
25   wire T=1;
26   TFF TFF1(Q1,q1,T,clk,reset);
27   TFF TFF2(Q0,q0,T,q1,reset);
28 endmodule
```

2bit binary counter



```
1 module decoder_2to4(Y0,Y1,Y2,Y3,A,B);
2   input A,B;
3   output Y0,Y1,Y2,Y3;
4   assign Y0=~(A & ~B);
5   assign Y1=~(A & B);
6   assign Y2=~(A & ~B);
7   assign Y3=~(A & B);
8 endmodule
9
10
```

2-to-4 decoder

The screenshot shows the ModelSim-Intel FPGA Starter Edition 10.5b interface. The main window displays a Verilog module named `bit4_mux_4to1`. The code defines a 4-bit 4-to-1 multiplexer with inputs `I0, I1, I2, I3` and select inputs `S0, S1`. The outputs are `O[3], O[2], O[1], O[0]`. The logic is implemented using assign statements with bitwise operations.

```
1 module bit4_mux_4to1(O[3],O[2],O[1],O[0],I0,I1,I2,I3,S0,S1);
2   input [3:0]I0,I1,I2,I3;
3   input S0,S1;
4   output O[3],O[2],O[1],O[0];
5   assign O[0]=(~S1 & ~S0 & I0[0]) | (~S1 & S0 & I1[0]) | (S1 & ~S0 & I2[0]) | (S1 & S0 & I3[0]);
6   assign O[1]=(~S1 & ~S0 & I0[1]) | (~S1 & S0 & I1[1]) | (S1 & ~S0 & I2[1]) | (S1 & S0 & I3[1]);
7   assign O[2]=(~S1 & ~S0 & I0[2]) | (~S1 & S0 & I1[2]) | (S1 & ~S0 & I2[2]) | (S1 & S0 & I3[2]);
8   assign O[3]=(~S1 & ~S0 & I0[3]) | (~S1 & S0 & I1[3]) | (S1 & ~S0 & I2[3]) | (S1 & S0 & I3[3]);
9
10  endmodule
11
```

The bottom status bar shows the project name "homework2", the current file "bit4_mux_4to1.v", and the simulation time "10/14/2019 12:18:22 am".

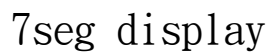
4bits 4-to-1 MUX

The screenshot shows the ModelSim-Intel FPGA Starter Edition 10.5b interface. The main window displays a Verilog module named `BCD_to_seven`. The code defines a BCD-to-segment decoder with inputs `B0, B1, B2, B3` and outputs `a, b, c, d, e, f, g`. The logic is implemented using assign statements with bitwise operations.

```
1 module BCD_to_seven(a,b,c,d,e,f,g,B0,B1,B2,B3);
2   input B0,B1,B2,B3;
3   output a,b,c,d,e,f,g;
4
5   assign a=B1|B3|(~B0&~B2)|(B0&B2);
6   assign b=~B2|(~B0&~B1)|(B0&B1);
7   assign c=B0|~B1|B2;
8   assign d=B3|(~B0&~B2)|(~B0&B1)|(B0&~B1&B2)|(B1&~B2);
9   assign e=~B0&~B2|(~B0&~B1);
10  assign f=B3|(~B0&~B1)|(~B1&B2)|(~B0&B2);
11  assign g=B3|(~B1&B2)|(B1&~B2)|(~B0&B1);
12  endmodule
13
14
```

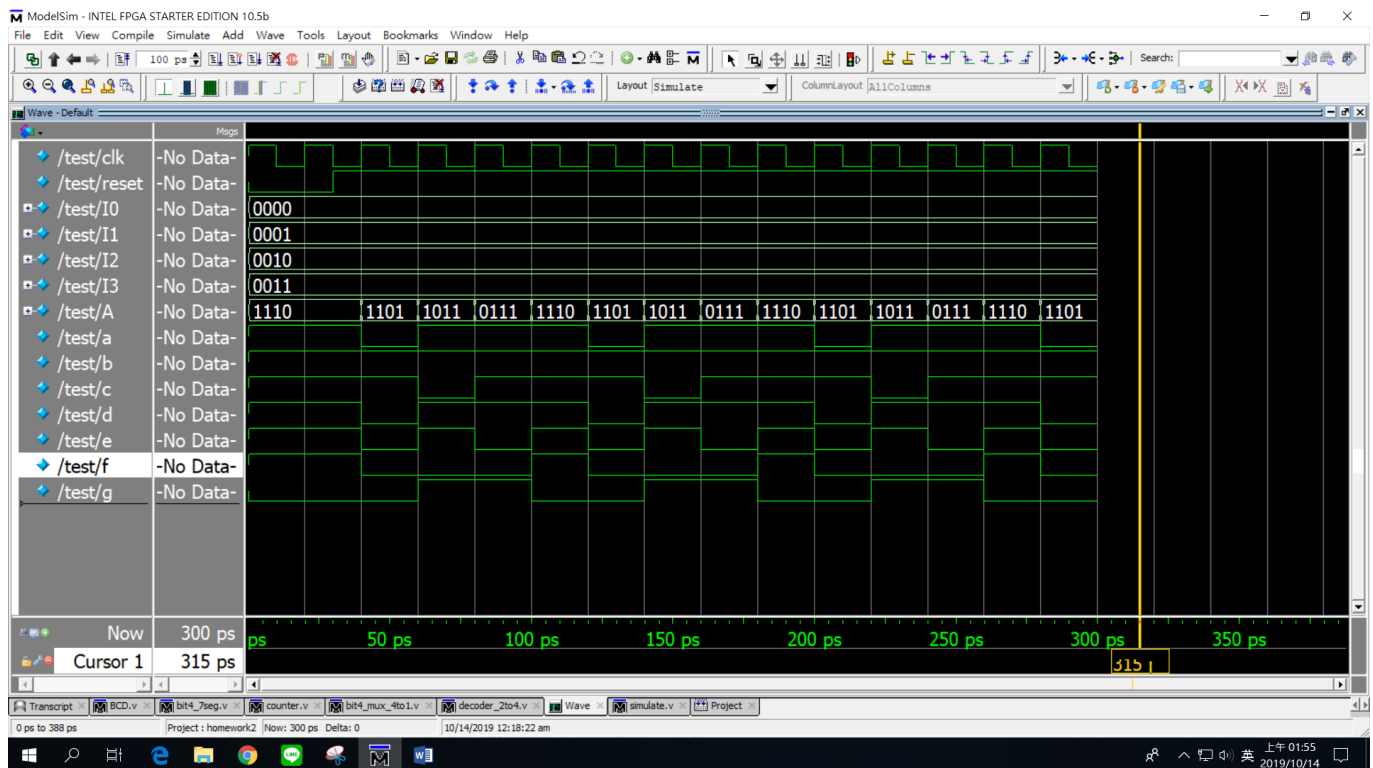
The bottom status bar shows the project name "homework2", the current file "BCD_to_seven.v", and the simulation time "10/14/2019 12:18:22 am".

BCD-to-segment decoder



執行結果

```
ModelSim - INTEL FPGA STARTER EDITION 10.5b
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
100 ps
Transcript
# Loading work1.tff
# Loading work.DFF
# Loading work.BCD_to_seven
# Loading work.bit4_mux_4to1
# Load canceled
add wave -position insertpoint sim:/test/*
add wave -position insertpoint sim:/test/*
V$M 30> run -all
#
# 0reset=0,A=1110,abcdefg=1111110,
# 30reset=1,A=1110,abcdefg=1111110,
# 40reset=1,A=1101,abcdefg=0110000,
# 60reset=1,A=1011,abcdefg=1101101,
# 80reset=1,A=0111,abcdefg=1111001,
# 100reset=1,A=1110,abcdefg=1111110,
# 120reset=1,A=1101,abcdefg=0110000,
# 140reset=1,A=1011,abcdefg=1101101,
# 160reset=1,A=0111,abcdefg=1111001,
# 180reset=1,A=1110,abcdefg=1111110,
# 200reset=1,A=1101,abcdefg=0110000,
# 220reset=1,A=1011,abcdefg=1101101,
# 240reset=1,A=0111,abcdefg=1111001,
# 260reset=1,A=1110,abcdefg=1111110,
# 280reset=1,A=1101,abcdefg=0110000,
# ** Note: $finish : C:/intelFPGA/17.0/homework2/simulate.v(27)
# Time: 300 ps Iteration: 0 Instance: /test
# 1
# Break in Module test at C:/intelFPGA/17.0/homework2/simulate.v line 27
add wave -position insertpoint sim:/test/*
add wave -position insertpoint sim:/test/*
V$M 32>
Transcript BCD.v bit4_7seg.v counter.v bit4_mux_4to1.v decoder_2to4.v Wave simulate.v Project
Project: homework2 Now: 300 ps Delta: 0 10/14/2019 12:18:22 am
上午 01:54
2019/10/14
```



A=1110 顯示 1 A=1101 顯示 2

A=1011 顯示 3 A=0111 顯示 4

心得

這次的作業花了不少時間，沒有做好規劃與整理，整個處理起來很沒有效率，需要除錯時，因為程式中的許多自己沒整理清楚的名稱，導致眼花撩亂，不知道該從哪方面下手，常常取一些無意義的名稱，是我的一個很嚴重的壞習慣，讓我這次吃足了苦頭，相信有了這一次的經驗，一定能夠讓我在之後能夠警惕自己，別再犯相同的錯誤。