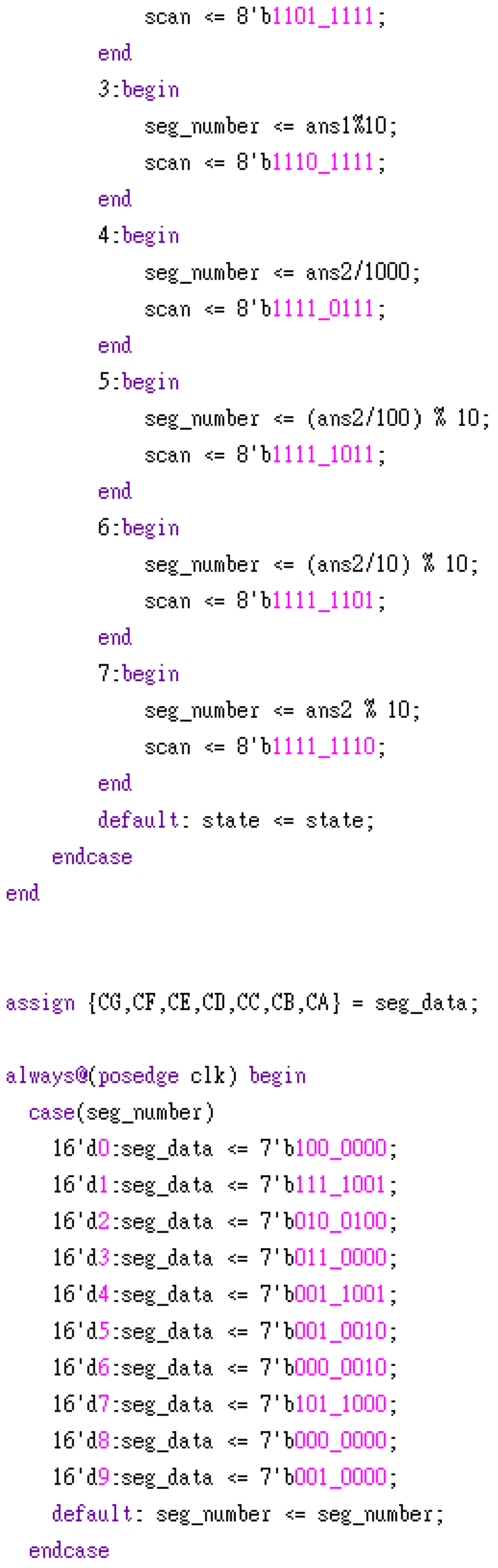
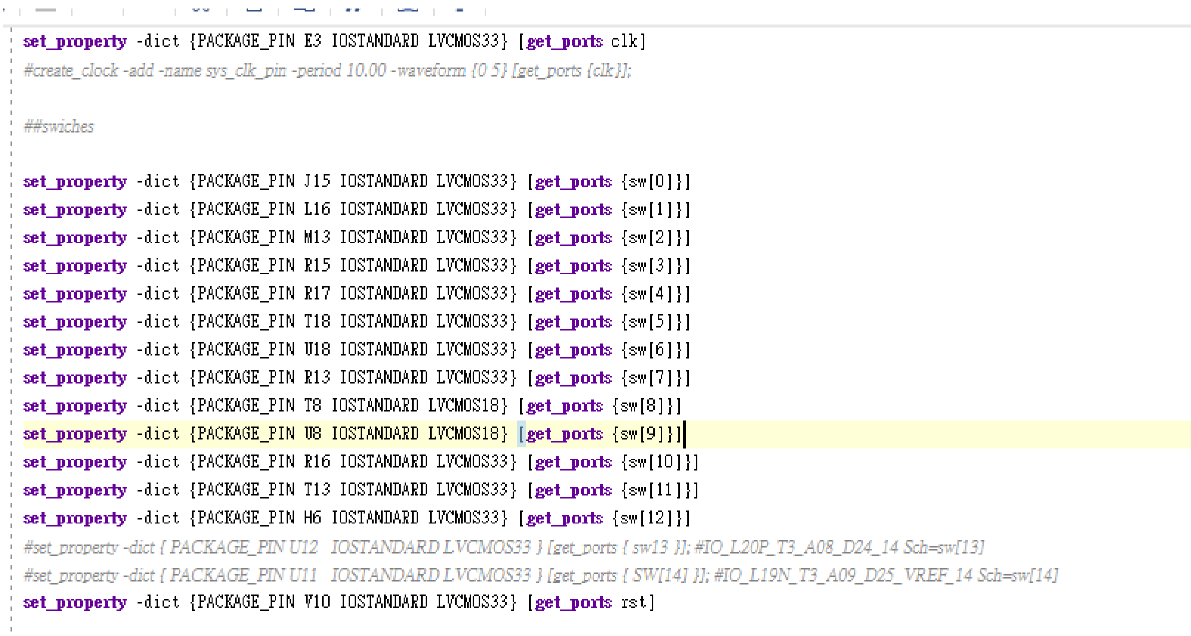
計算機組織

使用七段顯示器進行顯示



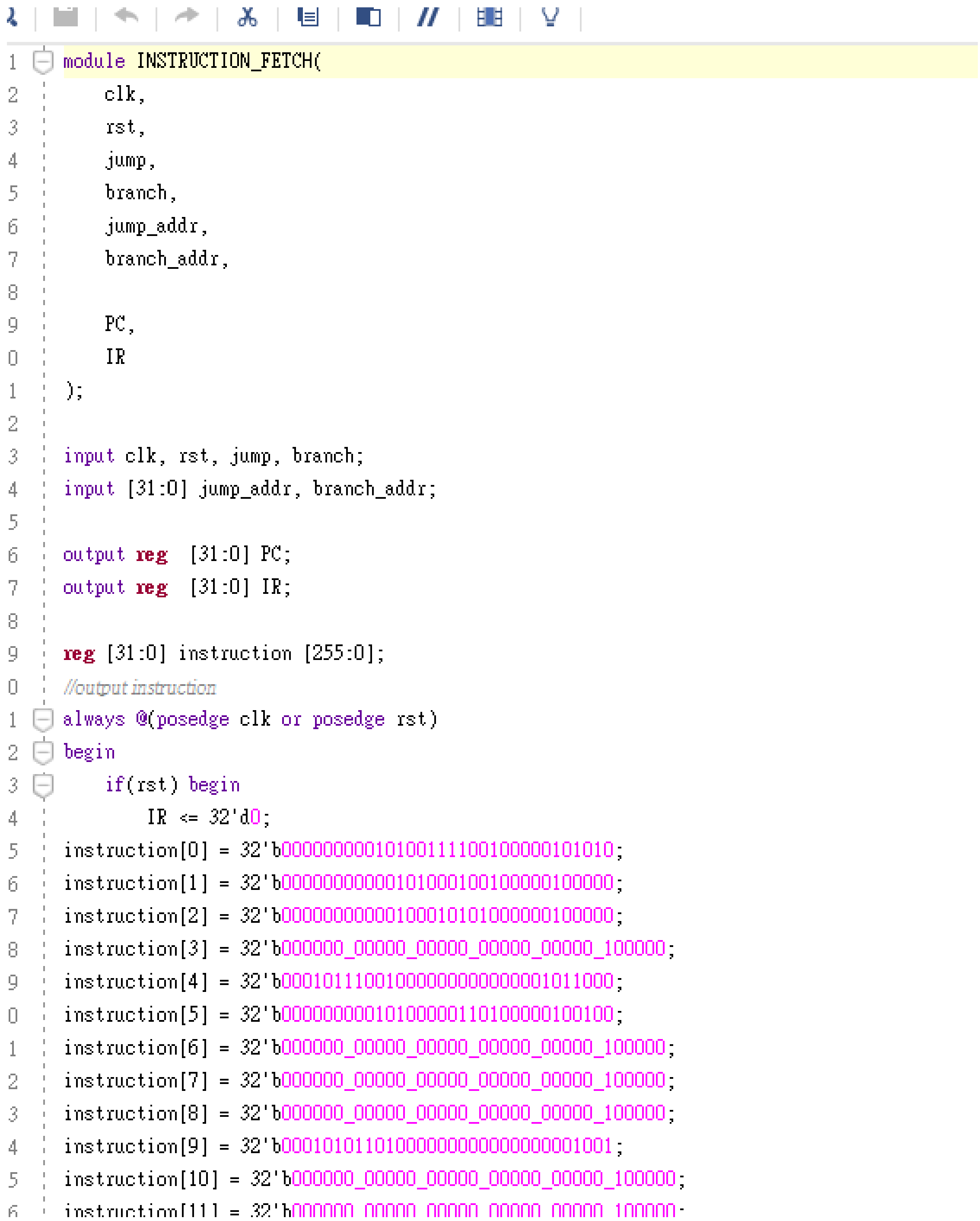
```
            scan <= 8'b1101_1111;
        end
        3:begin
            seg_number <= ans1%10;
            scan <= 8'b1110_1111;
        end
        4:begin
            seg_number <= ans2/1000;
            scan <= 8'b1111_0111;
        end
        5:begin
            seg_number <= (ans2/100) % 10;
            scan <= 8'b1111_1011;
        end
        6:begin
            seg_number <= (ans2/10) % 10;
            scan <= 8'b1111_1101;
        end
        7:begin
            seg_number <= ans2 % 10;
            scan <= 8'b1111_1110;
        end
        default: state <= state;
    endcase
end


assign {CG,CF,CE,CD,CC,CB,CA} = seg_data;

always@(posedge clk) begin
  case(seg_number)
    16'd0:seg_data <= 7'b100_0000;
    16'd1:seg_data <= 7'b111_1001;
    16'd2:seg_data <= 7'b010_0100;
    16'd3:seg_data <= 7'b011_0000;
    16'd4:seg_data <= 7'b001_1001;
    16'd5:seg_data <= 7'b001_0010;
    16'd6:seg_data <= 7'b000_0010;
    16'd7:seg_data <= 7'b101_1000;
    16'd8:seg_data <= 7'b000_0000;
    16'd9:seg_data <= 7'b001_0000;
    default: seg_number <= seg_number;
  endcase
```

sw15作為rst用



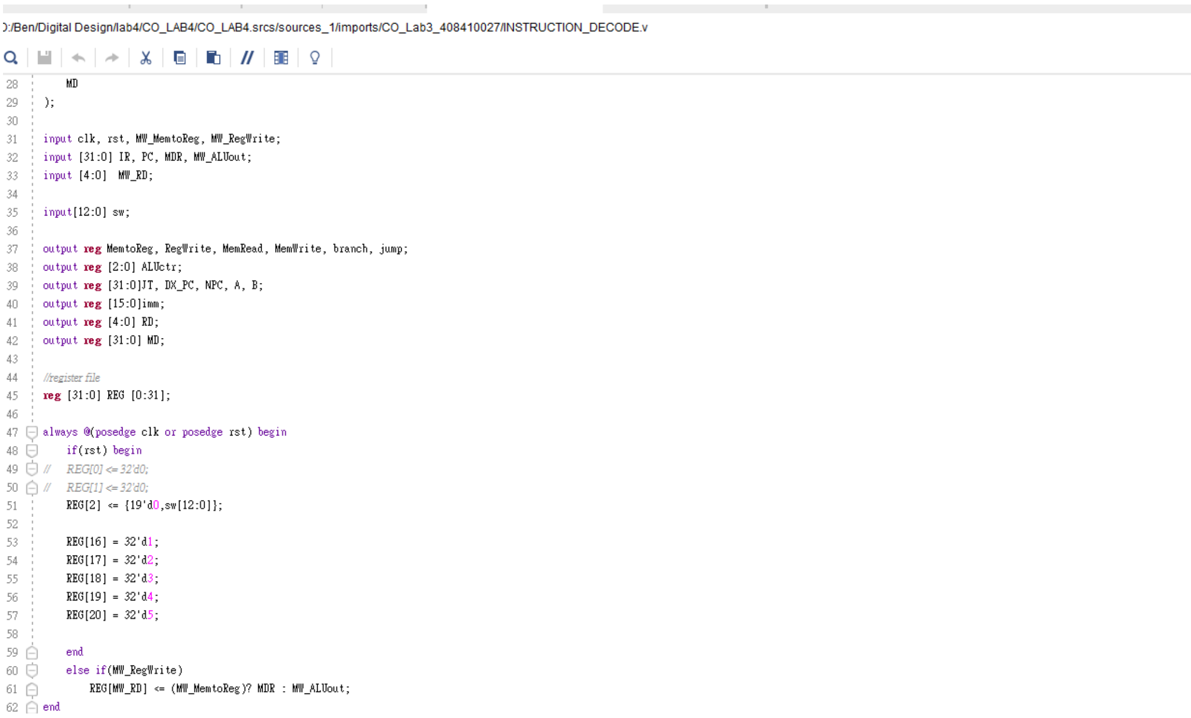
```
set_property -dict {PACKAGE_PIN E3 IOSTANDARD LVCMOS33} [get_ports clk]
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {clk}];

##swiches

set_property -dict {PACKAGE_PIN J15 IOSTANDARD LVCMOS33} [get_ports {sw[0]}]
set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33} [get_ports {sw[1]}]
set_property -dict {PACKAGE_PIN M13 IOSTANDARD LVCMOS33} [get_ports {sw[2]}]
set_property -dict {PACKAGE_PIN R15 IOSTANDARD LVCMOS33} [get_ports {sw[3]}]
set_property -dict {PACKAGE_PIN R17 IOSTANDARD LVCMOS33} [get_ports {sw[4]}]
set_property -dict {PACKAGE_PIN T18 IOSTANDARD LVCMOS33} [get_ports {sw[5]}]
set_property -dict {PACKAGE_PIN U18 IOSTANDARD LVCMOS33} [get_ports {sw[6]}]
set_property -dict {PACKAGE_PIN R13 IOSTANDARD LVCMOS33} [get_ports {sw[7]}]
set_property -dict {PACKAGE_PIN T8 IOSTANDARD LVCMOS18} [get_ports {sw[8]}]
set_property -dict {PACKAGE_PIN U8 IOSTANDARD LVCMOS18} [get_ports {sw[9]}]
set_property -dict {PACKAGE_PIN R16 IOSTANDARD LVCMOS33} [get_ports {sw[10]}]
set_property -dict {PACKAGE_PIN T13 IOSTANDARD LVCMOS33} [get_ports {sw[11]}]
set_property -dict {PACKAGE_PIN H6 IOSTANDARD LVCMOS33} [get_ports {sw[12]}]
#set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 } [get_ports { sw13 }]; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
#set_property -dict { PACKAGE_PIN U11   IOSTANDARD LVCMOS33 } [get_ports { SW[14] }]; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
set_property -dict {PACKAGE_PIN V10 IOSTANDARD LVCMOS33} [get_ports rst]
```

instruction存於IF



```
module INSTRUCTION_FETCH(
    clk,
    rst,
    jump,
    branch,
    jump_addr,
    branch_addr,

    PC,
    IR
);

input clk, rst, jump, branch;
input [31:0] jump_addr, branch_addr;

output reg  [31:0] PC;
output reg  [31:0] IR;

reg [31:0] instruction [255:0];
//output instruction
always @(posedge clk or posedge rst)
begin
    if(rst) begin
        IR <= 32'd0;
instruction[0] = 32'b00000000010100111100100000101010;
instruction[1] = 32'b00000000000101000100100000100000;
instruction[2] = 32'b00000000000100010101000000100000;
instruction[3] = 32'b000000_00000_00000_00000_00000_100000;
instruction[4] = 32'b00010111001000000000000001011000;
instruction[5] = 32'b00000000010100000110100000100100;
instruction[6] = 32'b000000_00000_00000_00000_00000_100000;
instruction[7] = 32'b000000_00000_00000_00000_00000_100000;
instruction[8] = 32'b000000_00000_00000_00000_00000_100000;
instruction[9] = 32'b00010101101000000000000000001001;
instruction[10] = 32'b000000_00000_00000_00000_00000_100000;
instruction[11] = 32'b000000_00000_00000_00000_00000_100000;
```

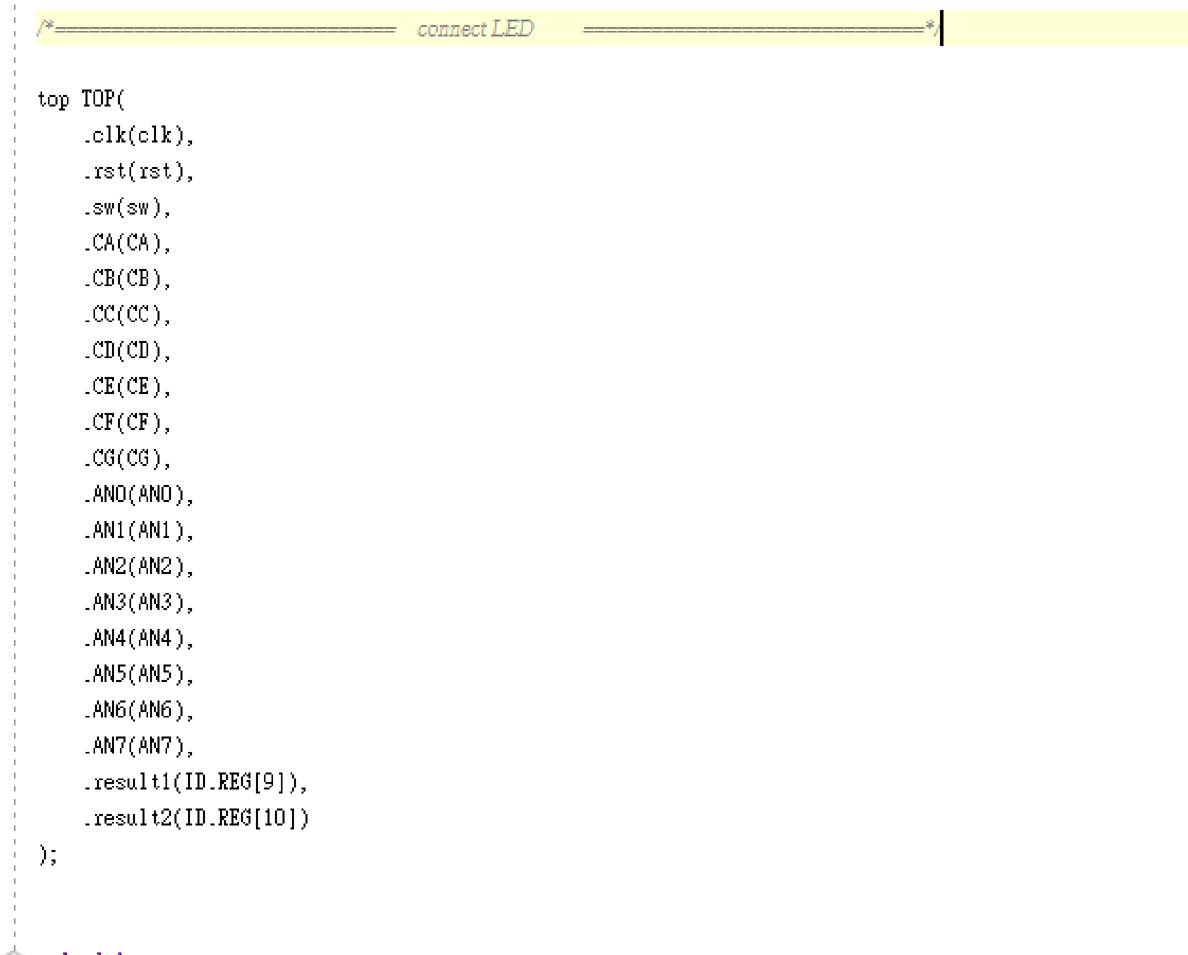
ID內 在rst時會配置好register的初始值，包含按鈕的值



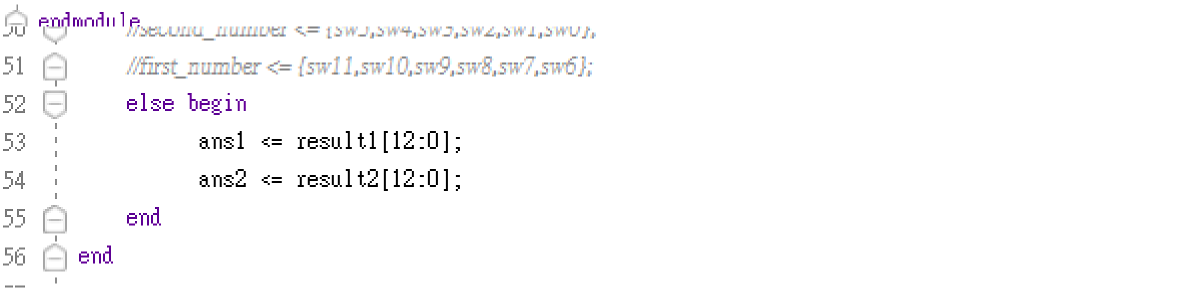
):/Ben/Digital Design/lab4/CO_LAB4/CO_LAB4.srcs/sources_1/imports/CO_Lab3_408410027/INSTRUCTION_DECODE.v

```verilog
    MD
);

input clk, rst, MW_MemtoReg, MW_RegWrite;
input [31:0] IR, PC, MDR, MW_ALUout;
input [4:0]  MW_RD;

input[12:0] sw;

output reg MemtoReg, RegWrite, MemRead, MemWrite, branch, jump;
output reg [2:0] ALUctr;
output reg [31:0]JT, DX_PC, NPC, A, B;
output reg [15:0]imm;
output reg [4:0] RD;
output reg [31:0] MD;

//register file
reg [31:0] REG [0:31];

always @(posedge clk or posedge rst) begin
    if(rst) begin
//    REG[0] <= 32'd0;
//    REG[1] <= 32'd0;
    REG[2] <= {19'd0,sw[12:0]};

    REG[16] = 32'd1;
    REG[17] = 32'd2;
    REG[18] = 32'd3;
    REG[19] = 32'd4;
    REG[20] = 32'd5;

    end
    else if(MW_RegWrite)
        REG[MW_RD] <= (MW_MemtoReg)? MDR : MW_ALUout;
end
```

output的LED，使用wire直接跟$9 $10接入 並且傳給七段顯示器



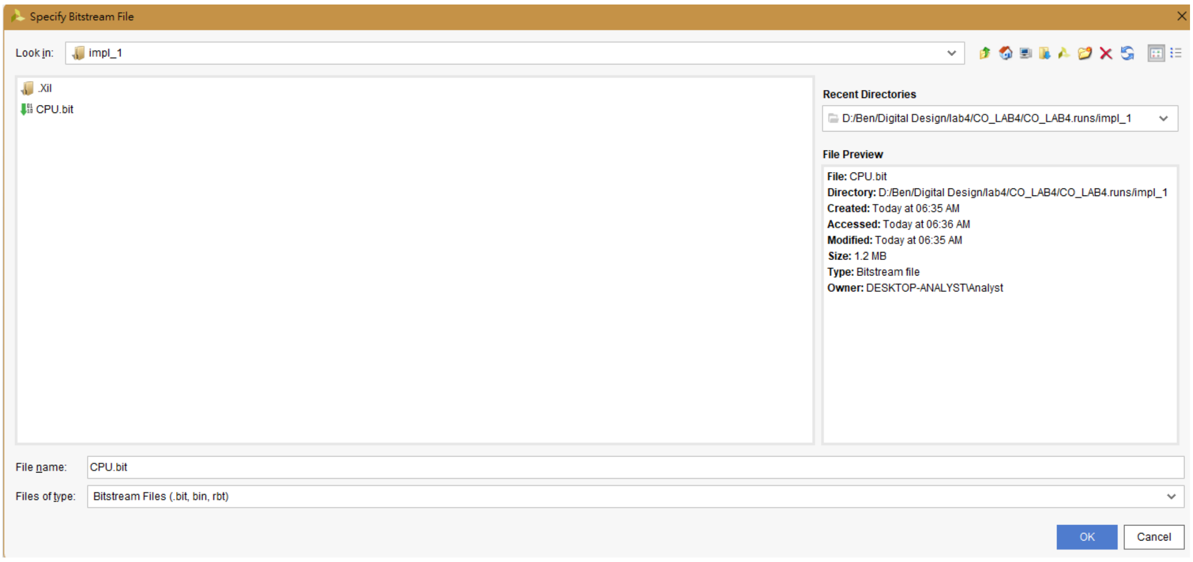
```
/*===============================   connect LED   ===============================*/

top TOP(
    .clk(clk),
    .rst(rst),
    .sw(sw),
    .CA(CA),
    .CB(CB),
    .CC(CC),
    .CD(CD),
    .CE(CE),
    .CF(CF),
    .CG(CG),
    .AN0(AN0),
    .AN1(AN1),
    .AN2(AN2),
    .AN3(AN3),
    .AN4(AN4),
    .AN5(AN5),
    .AN6(AN6),
    .AN7(AN7),
    .result1(ID.REG[9]),
    .result2(ID.REG[10])
);


endmodule
```



```
        //second_number <= {sw5,sw4,sw3,sw2,sw1,sw0};
        //first_number <= {sw11,sw10,sw9,sw8,sw7,sw6};
        else begin
                ans1 <= result1[12:0];
                ans2 <= result2[12:0];
        end
    end
```

CPU設計在CO_LAB4\CO_LAB4.srcs\sources_1\imports\CO_Lab3_408410027

bitstream檔案 1.2MB

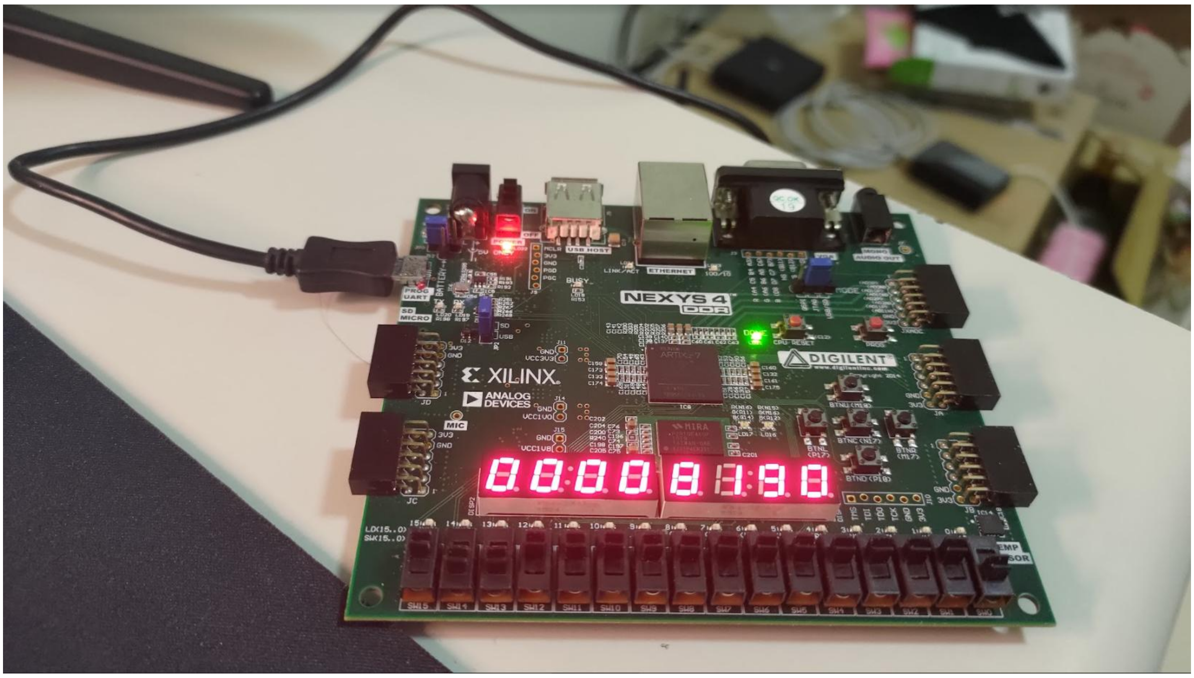


Recent Directories: D:/Ben/Digital Design/lab4/CO_LAB4/CO_LAB4.runs/impl_1

File: CPU.bit
Directory: D:/Ben/Digital Design/lab4/CO_LAB4/CO_LAB4.runs/impl_1
Created: Today at 06:35 AM
Accessed: Today at 06:36 AM
Modified: Today at 06:35 AM
Size: 1.2 MB
Type: Bitstream file
Owner: DESKTOP-ANALYST\Analyst

File name: CPU.bit
Files of type: Bitstream Files (.bit, bin, rbt)

按下rst設置input



放開rst 計算並顯示 左為大於 右為小於