

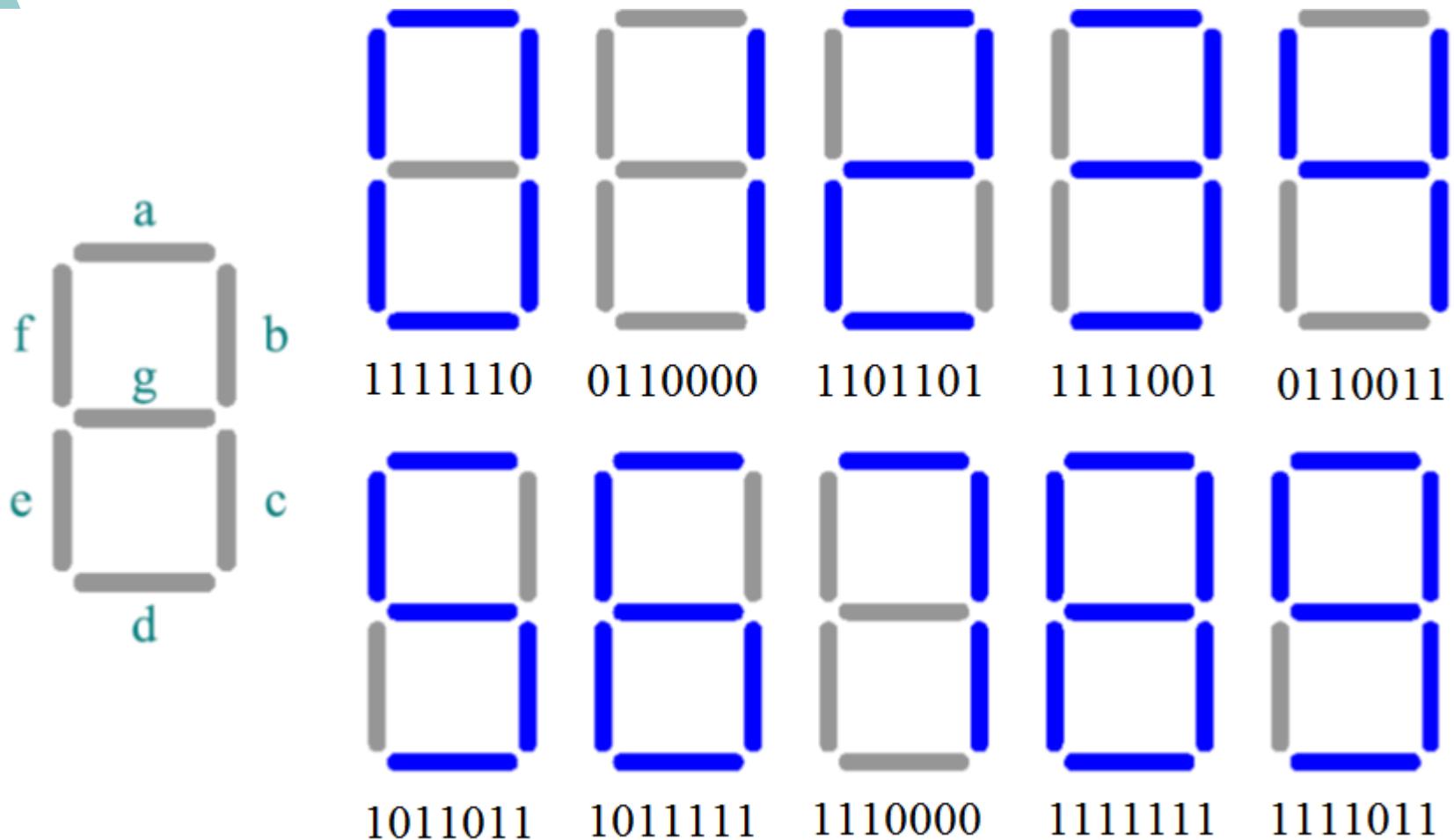
FPGA Lab-02

Application of seven-segment displays(1)

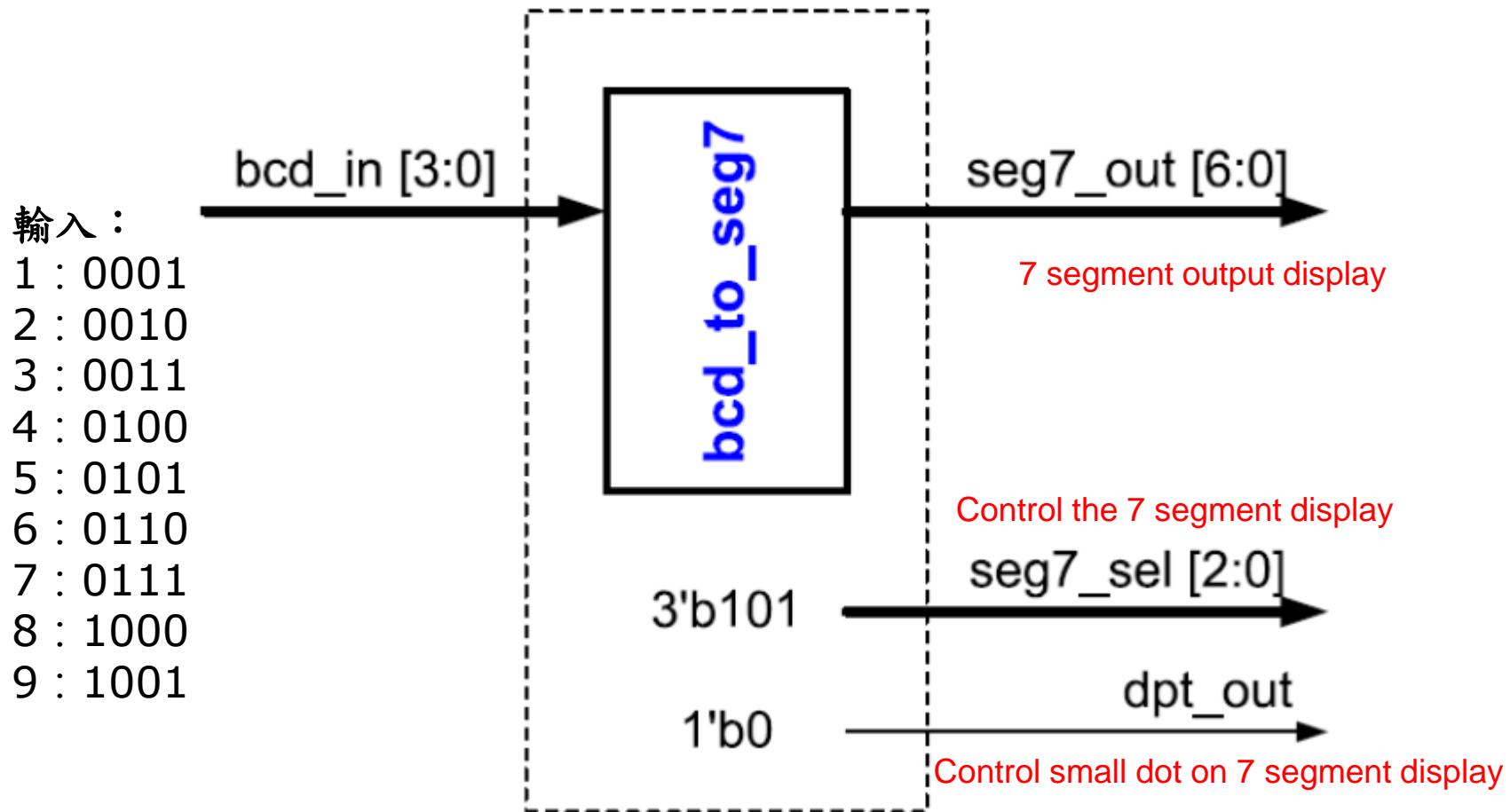
Lab

- Experiment 1: Understanding 7-segment Display Decoder - Decoding Principle
- Experiment 2: Displaying 0-to-9 on a seven-segment display with an up-counter
- Experiment 3: Displaying 00-to-99 on a seven-segment display with an upward counting counter

Experiment 1: Seven Segment Display Decoder - Decoding Principle

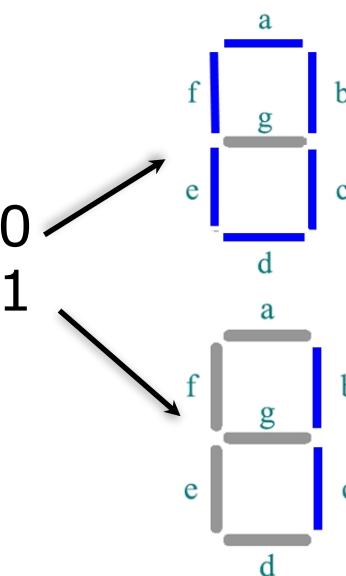


System Block Diagram



bcd_to_seg7.v

- module bcd_to_seg7(bcd_in, seg7);
- input[3:0] bcd_in;
- output[6:0] seg7;
- reg[6:0] seg7;
- always@ (bcd_in)begin
- case(bcd_in) // abcdefg
- 4'b0000: seg7 = 7'b1111110; // 0
- 4'b0001: seg7 = 7'b0110000; // 1
- **~~~Type your code~~~~~**
- default: seg7 = 7'b0000000;
- endcase
- end
- endmodule



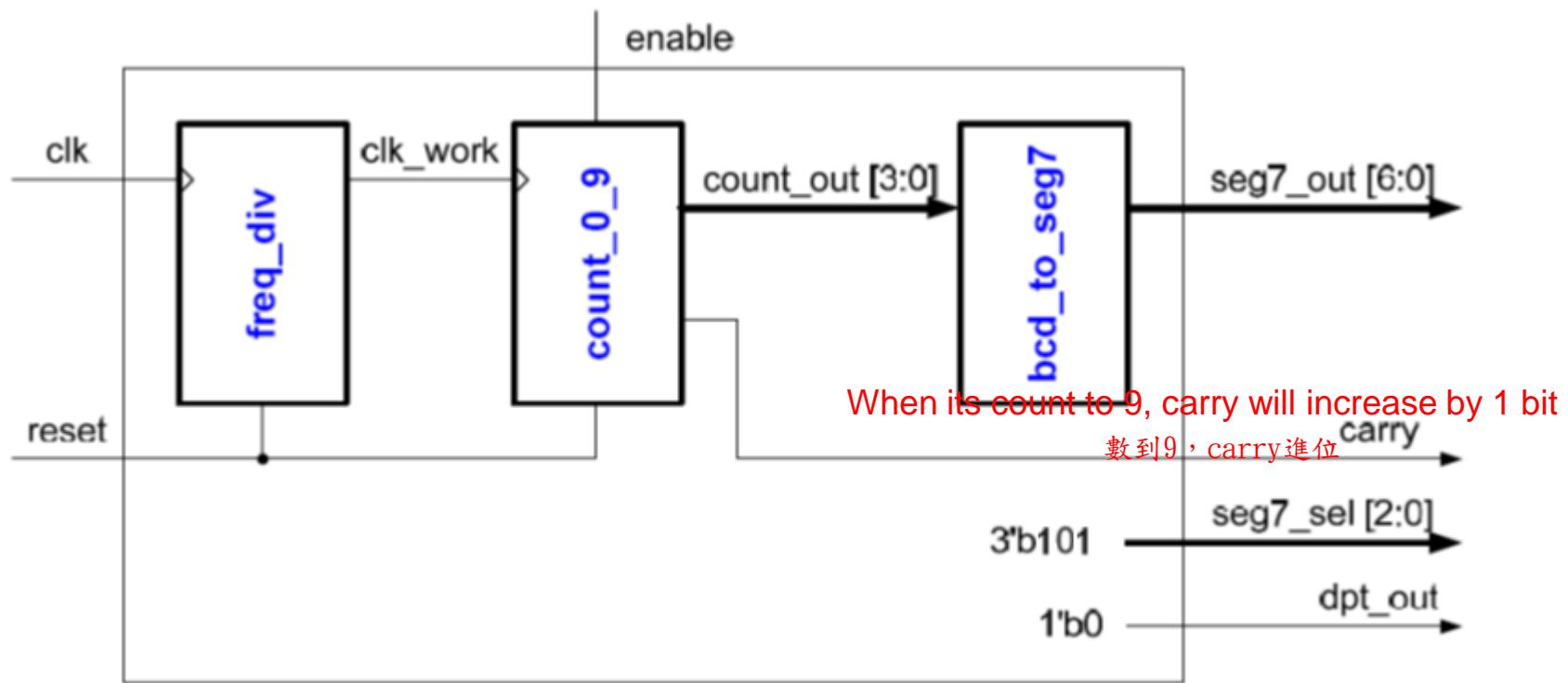
bcd_to_seg7_top.v

最上層程式命名記得改

- **module** bcd_to_seg7_top(bcd_in, seg7_sel,
seg7_out, dpt_out);
- **input** [3:0] bcd_in; // pinAA15,AA14,AB18,AA18
- **output** [2:0] seg7_sel; //pin AB10,AB11,AA12
- **output** [6:0] seg7_out;
- // pin AB7,AA7,AB6,AB5,AA9,Y9,AB8
- **output** dpt_out; // pinAA8
- **bcd_to_seg7 M1 (Your code);**
- **assign** seg7_sel = 3'b101; // Use the rightmost
segment
- **assign** dpt_out= 1'b0;
- **endmodule**

Experiment 2: Displaying 0-to-9 on a seven-segment display with an up-counter

- Program Architecture: Frequency Division, Counter, binary to 7 segment decoder



count_0_9.v

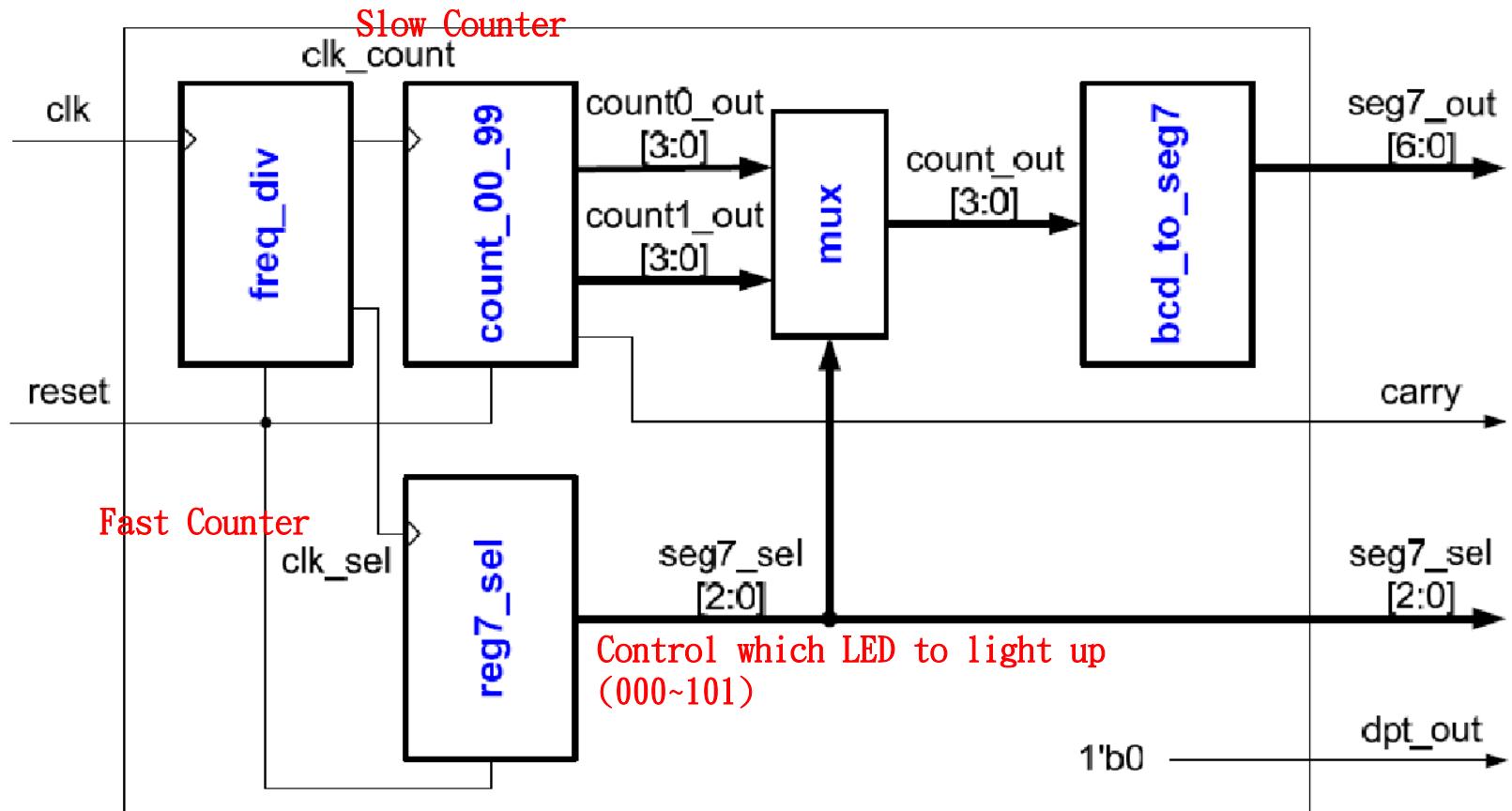
```
○ module count_0_9(clk, reset, enable, count_out, carry);
○ input clk, reset, enable;
○ output[3:0] count_out;
○ output carry;
○ reg[3:0] count_out;
○ assign carry = (count_out== 4'b1001) ? 1 : 0;
○ always@ (posedge clk or posedge reset)begin
○ if(reset)
○   count_out= 4'b0;
○ else if(enable == 1) begin
○   if(count_out== 4'b1001)
○     //~~~~~your code~~~~~/count_out back to 0
○   else
○     //~~~~~your code~~~~~/count_out add 1
○ end
○ end
○ endmodule
```

count_0_9_top.v

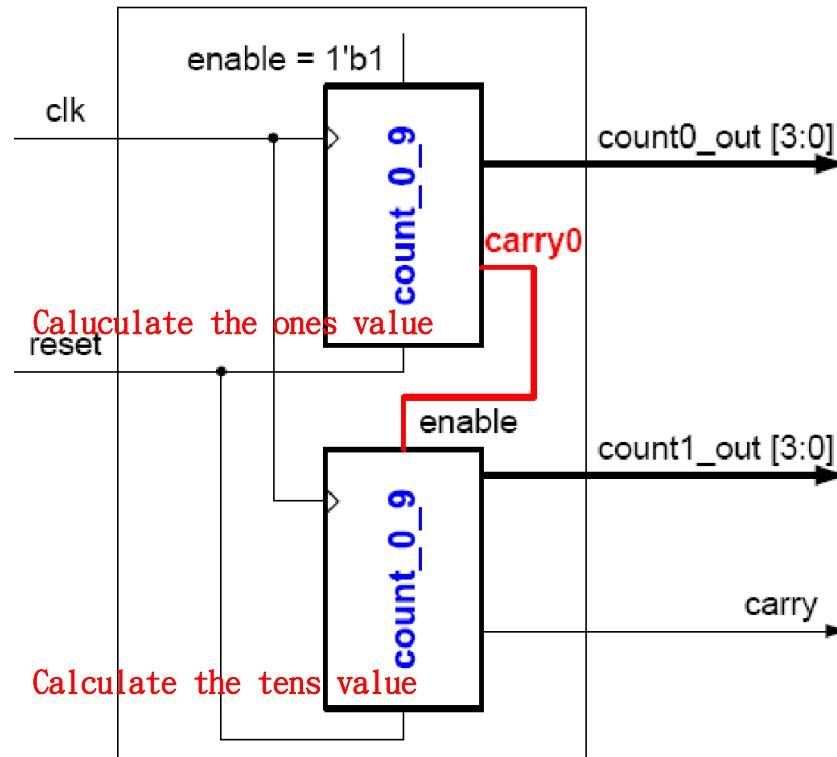
Change the main module

```
○ module count_0_9_top(clk, reset, enable, seg7_sel,  
seg7_out, dpt_out, carry, led_com);  
○ input clk, reset, enable;          //pin W16,C16,AA15  
○ output[2:0] seg7_sel;            //pin AB10,AB11,AA12  
○ output[6:0] seg7_out;           // pin AB7,AA7,AB6,AB5,AA9,Y9,AB8  
○ output dpt_out, carry, led_com;//pinAA8,E2,N20  
○ wire clk_work;  
○ wire[3:0] count_out;  
○ freq_div# (21)M1 (your code) ;  
○ count_0_9 M2 (your code) ;  
○ bcd_to_seg7 M3 (your code);  
○ assign seg7_sel = 3'b101;  
○ assign dpt_out = 1'b0;          //7 segment small dot display  
○ assign led_com = 1'b1;          //Light up top row LED  
○ endmodule
```

實驗三：在七段顯示器上顯示00-to-99的上數計數器



00-99計數器架構



count_00_99.v

- module count_00_99(clk, reset, enable,
count1_out, count0_out, carry);
- input clk, reset, enable;
- output[3:0] count1_out, count0_out;
- output carry = carry1 & carry0;
- wire carry0, carry1;
- count_0_9 C1(**your code**);
- count_0_9 C2(**your code**);
- endmodule

seg7_select.v

(Choose which 7 segment to
light up)

```
module seg7_select(clk, reset, seg7_sel);
parameter num_use= 6; //number of 7 segment
input clk, reset;
output[2:0] seg7_sel;
reg [2:0] seg7_sel;
always@ (posedge clk or posedge reset) begin
if(reset == 1)
    seg7_sel = 3'b101; // the rightmost one
else
    if(seg7_sel == 6 -num_use)
        seg7_sel = 3'b101;
    else
        seg7_sel = seg7_sel-3'b001; // shift left
end
endmodule
```

count_00_99_top.v

Change the main module

- module count_00_99_top(clk, reset, seg7_sel, enable, seg7_out, dpt_out, carry, led_com);
- input clk, reset, enable; //pin W16,C16,AA15
- output[2:0] seg7_sel; //pin AB10,AB11,AA12
- output[6:0] seg7_out; // pin AB7,AA7,AB6,AB5,AA9,Y9,AB8
- output dpt_out, led_com, carry;
- wire clk_count, clk_sel;
- wire[3:0] count_out, count1, count0;
- assign dpt_out= 1'b0;
- assign led_com= 1'b1;
- assign count_out= (seg7_sel == 3'b101)? count0 : count1; //MUX
- freq_div #(21) (your code); // slow
- freq_div #(17) (your code); // high
- count_00_99 (your code);
- bcd_to_seg7 (your code);
- seg7_select #(2) (your code);
- endmodule

Display 00-to-99 on 7 segment LED

