



```
module runningLED_tb(input clk, input [0:0] sw, output [3:0] led);
runningLEDs UUT(.clk(clk),.rst(sw[0]),.led(led[3:0]));
endmodule
~
~
~
~
~
~
~
~
~
~
~
```

```
module runningLEDs(input clk, input rst, output reg [3:0] led);
parameter cntmax = 32'd100000000; // 100M for 100MHz on-board oscillator crystal
reg [31:0] cnt;
always @ (posedge clk or negedge rst)
begin
if(rst)
begin
cnt<=32'd0;
led <= 4'b0001;
#100;
end
else if (cnt == cntmax)
begin
cnt<=32'd0;
led<={led[2:0],led[3]};
#100;
end
else
begin
cnt<=cnt+1'd1;
led<=led;
#100;
end
end
endmodule
```



```
module ssDisplay_count(digit, out, an, dp);
```

```
input [3:0] digit;  
output reg [6:0] out;  
output reg [3:0] an;  
output reg dp;
```

```
initial begin
```

```
    dp = 1; // off  
    an = 4'b0000; //all four on
```

```
end
```

```
always @ (digit)
```

```
begin
```

```
    case(digit)
```

```
        //MSB to LSB
```

```
4'b0000 : out = 7'b1000000; // 0000  
4'b0001 : out = 7'b1111001; // 0001  
4'b0010 : out = 7'b0100100; // 0010  
4'b0011 : out = 7'b0110000; // 0011  
4'b0100 : out = 7'b0011001; // 0100  
4'b0101 : out = 7'b0010010; // 0101  
4'b0110 : out = 7'b0000010; // 0110  
4'b0111 : out = 7'b1111000; // 0111  
4'b1000 : out = 7'b0000000; // 1000  
4'b1001 : out = 7'b0010000; // 1001
```

```
default: out = 7'b0001000; // A
```

```
    endcase
```

```
end
```

```
endmodule
```

```
module ssDisplay_top(sw,seg,an,dp);
```

```
input [3:0] sw;  
output [6:0] seg;  
output [3:0] an;  
output dp;
```

```
ssDisplay_count UUT(.digit(sw[3:0]),.out(seg[6:0]),.an(an[3:0]),.dp(dp));
```

```
endmodule
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```



```
module ssDisplay_count(digit, out, an, dp);
```

```
input [3:0] digit;  
output reg [6:0] out;  
output reg [3:0] an;  
output reg dp;
```

```
initial begin
```

```
    dp = 1; // off  
    an = 4'b1110; //only one on
```

```
end
```

```
always @ (digit)
```

```
begin
```

```
    case(digit)
```

```
        //MSB to LSB
```

```
        4'b0000 : out = 7'b1000000; // 0000
```

```
        4'b0001 : out = 7'b1111001; // 0001
```

```
        4'b0010 : out = 7'b0100100; // 0010
```

```
        4'b0011 : out = 7'b0110000; // 0011
```

```
        4'b0100 : out = 7'b0011001; // 0100
```

```
        4'b0101 : out = 7'b0010010; // 0101
```

```
        4'b0110 : out = 7'b0000010; // 0110
```

```
        4'b0111 : out = 7'b1111000; // 0111
```

```
        4'b1000 : out = 7'b0000000; // 1000
```

```
        4'b1001 : out = 7'b0010000; // 1001
```

```
        default: out = 7'b0001000; // A
```

```
    endcase
```

```
end
```

```
endmodule
```

```
module ssDisplay_top(sw,seg,an,dp);
```

```
input [3:0] sw;  
output [6:0] seg;  
output [3:0] an;  
output dp;
```

```
ssDisplay_count UUT(.digit(sw[3:0]),.out(seg[6:0]),.an(an[3:0]),.dp(dp));
```

```
endmodule
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```