附件：

BTN_Anti_Jitter 模块

```
module BTN_Anti_Jitter(
      input Clock,
      input BTN_IN,
      output reg BTN_Out
   );

    reg [3:0] cnt;
    reg BTN_Old;

    always @ (posedge Clock) begin
         if(BTN_IN != BTN_Old)
            begin
                  cnt <= 4'b0000; BTN_Old <= BTN_IN;
            end
         else
            begin
               if( cnt == 4'b1111)
                  begin
                     cnt <= 4'b0000; BTN_Out <= BTN_IN;
                  end
               else
                     cnt <= cnt + 1'b1;
            end
    end

endmodule
```

Hex7seg_decode 模块

```
module Hex7seg_decode(
      input wire [23:0] disp_num,
      input wire[2:0] Scanning,
      output wire [7:0] SEGMENT,
      output reg [5:0] AN
   );

   reg [3:0] digit;
   reg [7:0] digit_seg;

   assign SEGMENT = digit_seg;

   always @ (*) begin
         AN = 6'b000000;
         case (Scanning)
```

```
                3'h0: begin digit[3:0] = disp_num[3:0];  AN =
6'b000001; end
                3'h1: begin digit[3:0] = disp_num[7:4];   AN =
6'b100000; end
                3'h2: begin digit[3:0] = disp_num[11:8];  AN =
6'b010000; end
                3'h3: begin digit[3:0] = disp_num[15:12]; AN =
6'b001000; end
                3'h4: begin digit[3:0] = disp_num[19:16]; AN =
6'b000100; end
                3'h5: begin digit[3:0] = disp_num[23:20]; AN =
6'b000010; end
            endcase
    end

    always @ (*) begin
                case (digit)
                     4'h0: digit_seg = 8'b00111111;
                      4'h1: digit_seg = 8'b00000110;
                      4'h2: digit_seg = 8'b01011011;
                      4'h3: digit_seg = 8'b01001111;
                      4'h4: digit_seg = 8'b01100110;
                      4'h5: digit_seg = 8'b01101101;
                     4'h6: digit_seg = 8'b01111101;
                      4'h7: digit_seg = 8'b00000111;
                       4'h8: digit_seg = 8'b01111111;
                       4'h9: digit_seg = 8'b01101111;
                       4'hA: digit_seg = 8'b01110111;
                       4'hB: digit_seg = 8'b01111100;
                       4'hC: digit_seg = 8'b00111001;
                       4'hD: digit_seg = 8'b01011110;
                       4'hE: digit_seg = 8'b01111001;
                       4'hF: digit_seg = 8'b01110001;
                endcase
    end

endmodule
```