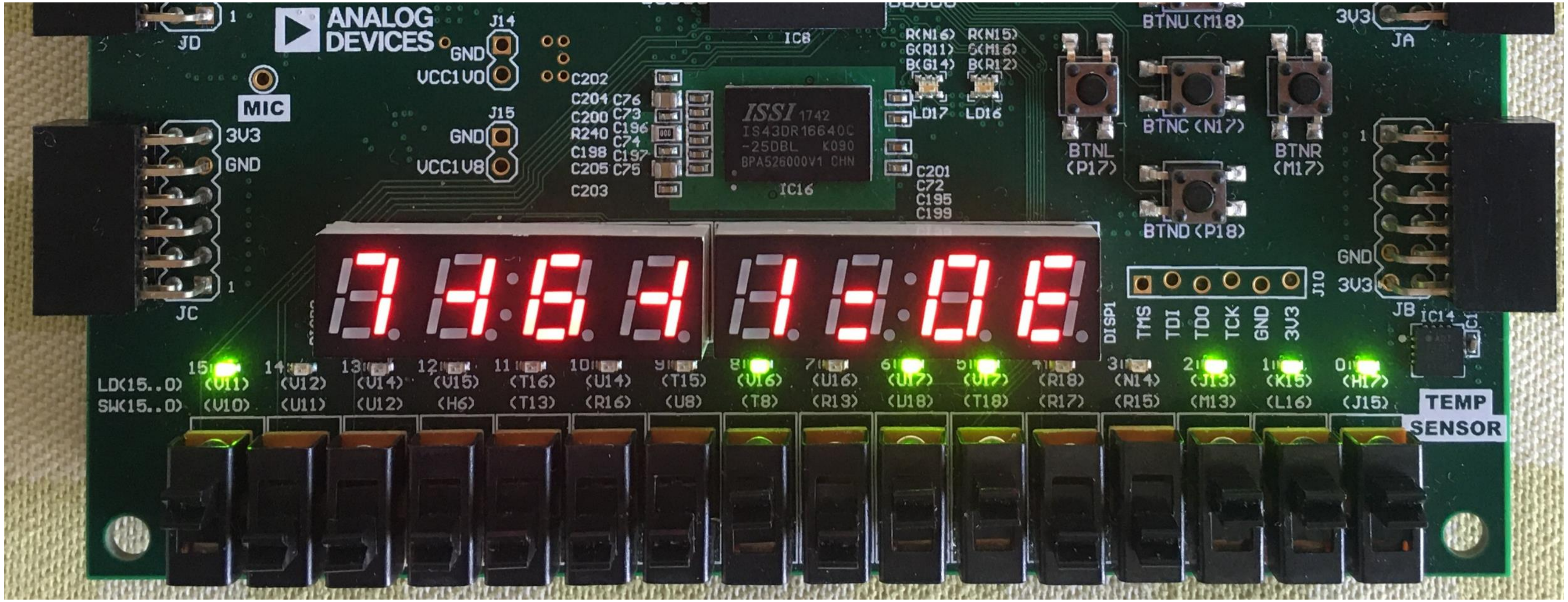
实验3：ALU 参考答案



ANALOG DEVICES
BTNU (M18)
BTNC (N17)
BTNL (P17)
BTNR (M17)
BTND (P18)
LD(15..0)
SW(15..0)
TEMP SENSOR

层次化、模块化 设计方法

∵ 对于较复杂的组合逻辑电路，往往不适合用一组方程直接描述。

- **层次化**设计方法：自顶向下对整个设计任务进行分层和分块的划分，

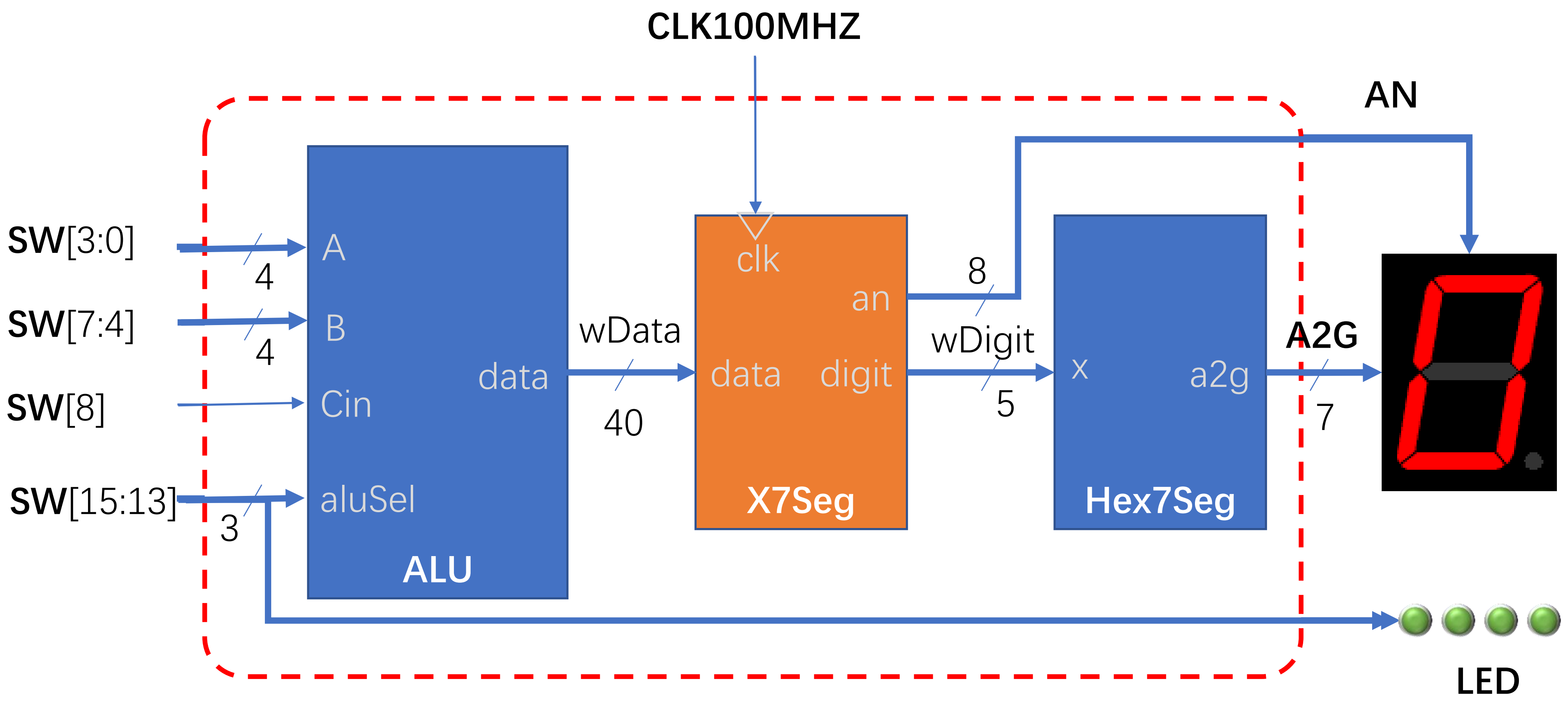
  降低每层的复杂度，简化每个模块的功能；

  或自底向上地对每个有限复杂度的模块进行实现或调用。

- **模块化**设计方法：将经过设计和验证的能完成一定功能的逻辑电路

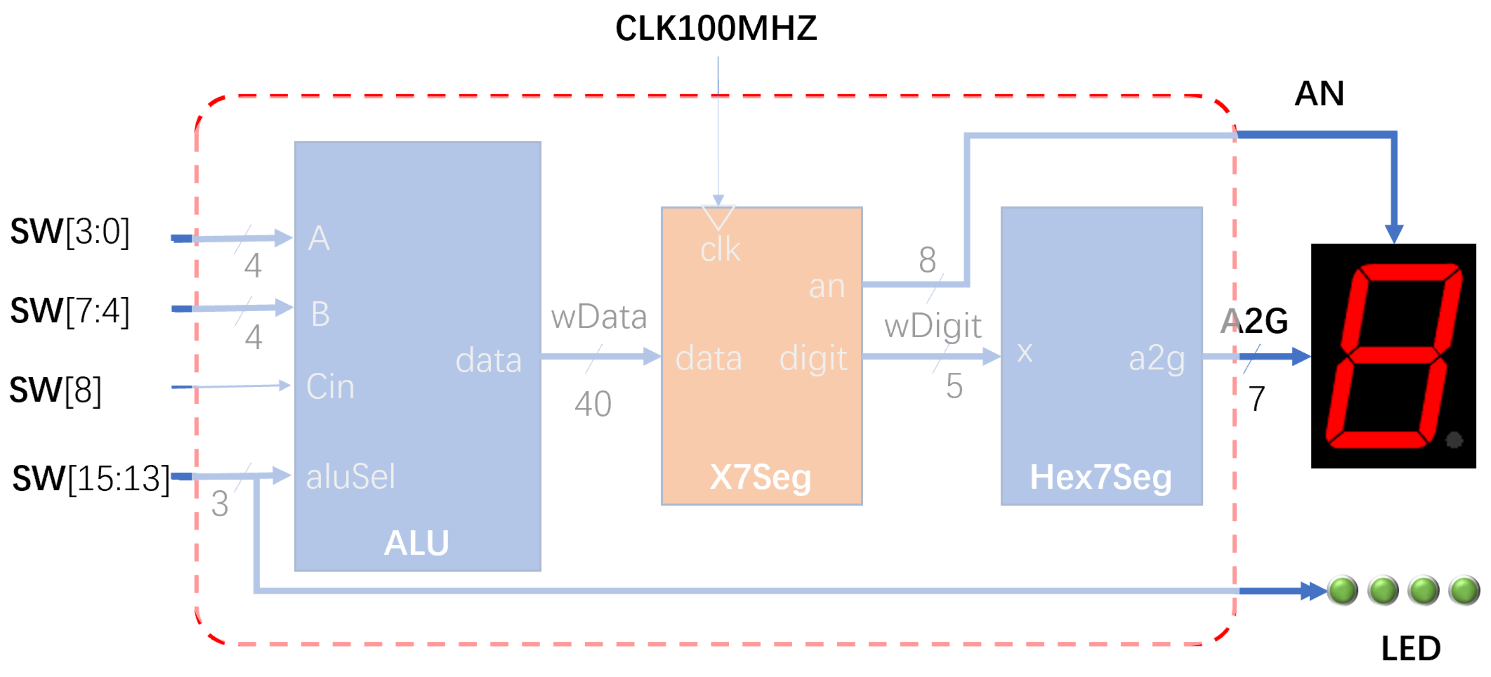
  封装成为模块，在后续的设计中都可反复使用。

设计思路



CLK100MHZ
AN
SW[3:0]
SW[7:4]
SW[8]
SW[15:13]
4
4
3
A
B
Cin
aluSel
data
ALU
wData
40
clk
an
data
digit
X7Seg
8
wDigit
5
x
a2g
Hex7Seg
A2G
7
LED

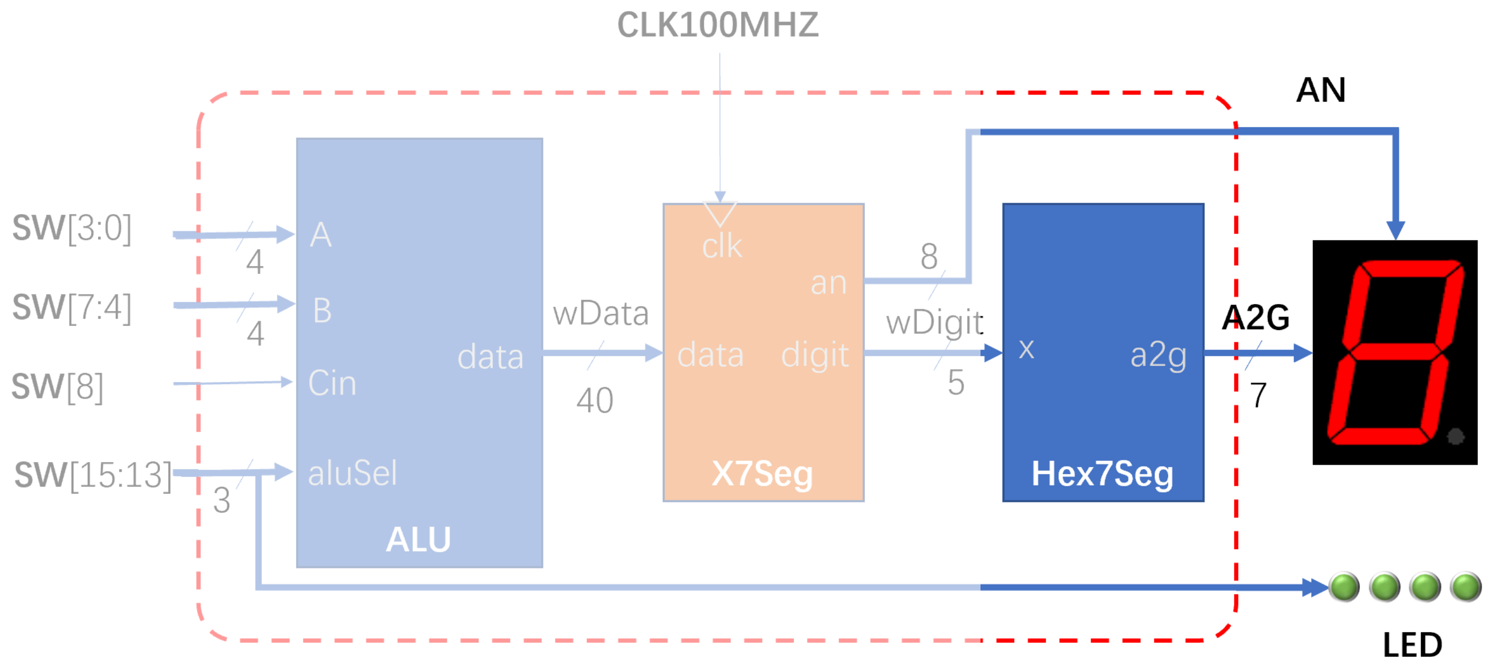
顶层模块



```
module ALU_Top(input logic         CLK100MHZ,
               input logic [15:0] SW,
              output logic [15:0] LED,
              output logic [7:0]  AN,
              output logic [6:0]  A2G );

    logic [39:0] wData;
    logic [4:0]  wDigit;

    assign LED = SW;   // LED

    ALU a1(.A      (SW[3:0]),
           .B      (SW[7:4]),
           .Cin    (SW[8]  ),
           .aluSel(SW[15:13]),//M(SW15),S1,S0(SW14,13)
           .data   (wData)   );

    // 分时复用数码管显示
    X7Seg x1(.clk   (CLK100MHZ),
             .data  (wData),
             .digit(wDigit),
             .an    (AN) );

    // 单个七段数码管显示
    Hex7Seg h1(.x(wDigit), .a2g(A2G) );
endmodule
```

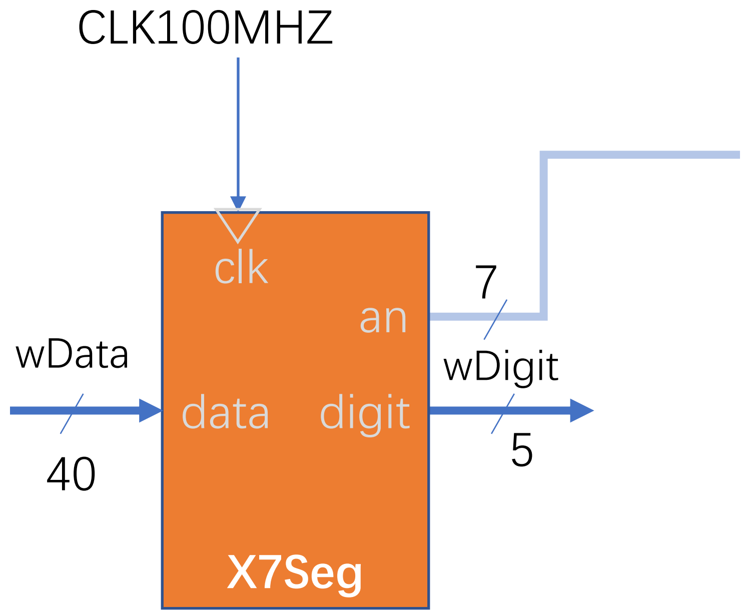
单个七段数码管显示模块

```
module Hex7Seg( input  logic [4:0] x,
                output logic [6:0] a2g  );
    // a2g format {a, b, c, d, e, f, g}
    always_comb
        case(x)
            5’b00000 : a2g = 7’b0000001; //0
            5’b00001 : a2g = 7’b1001111; //1
            5’b00010 : a2g = 7’b0010010; //2
            5’b00011 : a2g = 7’b0000110; //3
            5’b00100 : a2g = 7’b1001100; //4
            5’b00101 : a2g = 7’b0100100; //5
            5’b00110 : a2g = 7’b0100000; //6
            5’b00111 : a2g = 7’b0001111; //7
            5’b01000 : a2g = 7’b0000000; //8
            5’b01001 : a2g = 7’b0000100; //9
            5’b01010 : a2g = 7’b0001000; //A
            5’b01011 : a2g = 7’b1100000; //B
            5’b01100 : a2g = 7’b0110001; //C
            5’b01101 : a2g = 7’b1000010; //D
            5’b01110 : a2g = 7’b0110000; //E
            5’b01111 : a2g = 7’b0111000; //F
            5’b10000 : a2g = 7’b0110111; //=
            5’b10001 : a2g = 7’b1001110; //+
            5’b10010 : a2g = 7’b1111110; //-
        default      : a2g = 7’b1111111; //全灭
      endcase
endmodule
```



数码管分时复用模块

```
module X7Seg( input logic        clk,
              input logic [39:0] data,
             output logic [4:0]  digit,
             output logic [7:0]  an  );

    logic [19:0] clkdiv;
    logic [2:0]  s;
    assign s = clkdiv[19:17];

    //8个七段数码管要显示的数据
    always_comb
        case(s)
            0: digit = data[4:0];
            1: digit = data[9:5];
            2: digit = data[14:10];
            3: digit = data[19:15];
            4: digit = data[24:20];
            5: digit = data[29:25];
            6: digit = data[34:30];
            7: digit = data[39:35];
        endcase

    //8个七段数码管的分时显示
    always_comb
        case(s)
            0: an = 8'b1111_1110;
            1: an = 8'b1111_1101;
            2: an = 8'b1111_1011;
            3: an = 8'b1111_0111;
            4: an = 8'b1110_1111;
            5: an = 8'b1101_1111;
            6: an = 8'b1011_1111;
            7: an = 8'b0111_1111;
        endcase

    //时钟分频器
    always @(posedge clk)
        clkdiv <= clkdiv + 1;
endmodule
```



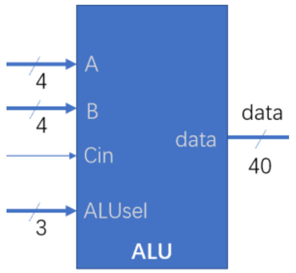
ALU

```
module ALU(input  logic [3:0]  A,
           input  logic [3:0]  B,
           input  logic        Cin,
           input  logic [2:0]  aluSel,
           output logic [39:0] data  );

    logic [4:0] temp;
    logic [3:0] y; //4位计算结果

    always_comb
    begin
        //初始化
        data         = {40{1'b1}}; // 全灭
        data[39:35] = {1'b0,A};    // A
        data[14:10] = 5'b10000;    // =
        temp = 5'b00000;

        case(aluSel) //M(SW15),S1(SW14),S0(SW13)
            3'b000:  // =======  NOT A
            begin
                y = ~A;
                data[4:0] = {1'b0,y}; //计算结果
            end
```

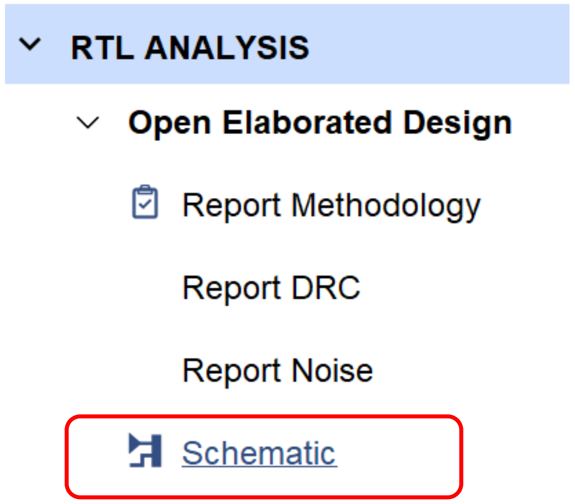
```
            3'b001:  // ======= A AND B
            begin
                y = A & B;
                data[29:25] = {1'b0,B}; // B
                data[4:0]   = {1'b0,y}; //计算结果
            end

            3'b010:  //========= A OR B
            begin
                y = A | B;
                data[29:25] = {1'b0,B}; // B
                data[4:0]   = {1'b0,y}; //计算结果
            end

            3'b011:  // ======== A XOR B
            begin
                y = A ^ B;
                data[29:25] = {1'b0,B}; //B
                data[4:0]   = {1'b0,y}; //计算结果
            end
```

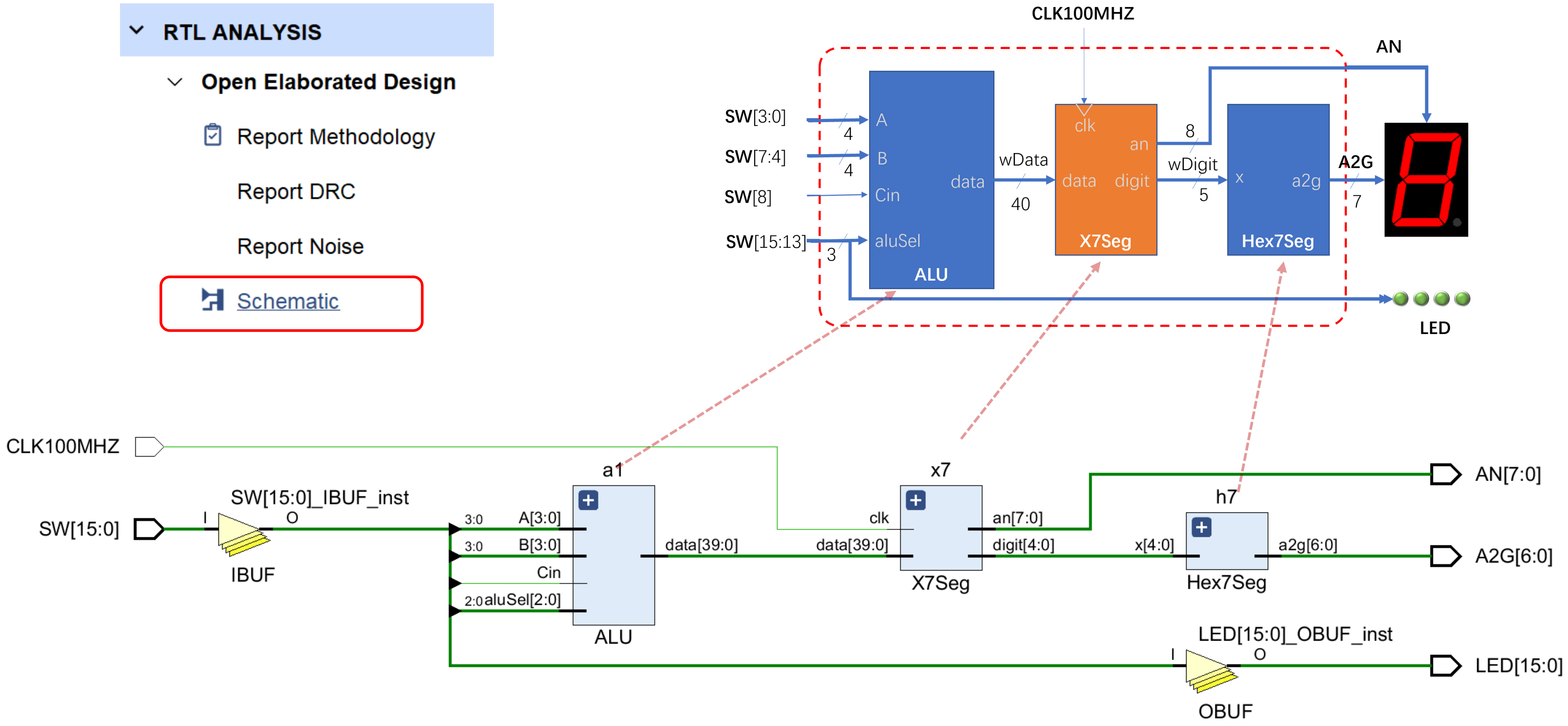
```
            3'b100: // ======= A + B
            begin
                temp = {1'b0, A} + {1'b0, B} + {4'b0000, Cin};
                y    = temp[3:0];

                data[34:30] = 5'b10001;          // +
                data[29:25] = {1'b0,B};          // B
                data[24:20] = 5'b10001;          // +
                data[19:15] = {4'b0000,Cin};     //Cin
                data[9:5]   = {4'b0000,temp[4]};//Cout
                data[4:0]   = {1'b0,y};          //计算结果
            end

            3'b101: // A - B    默认A大于B
            begin
                temp = {1'b0, A} - {1'b0, B} - {4'b0000, Cin};
                y    = temp[3:0];

                data[34:30] = 5'b10010;           // -
                data[29:25] = {1'b0,B};           // B
                data[24:20] = 5'b10010;           // -
                data[19:15] = {4'b0000, Cin};     //Cin
                data[9:5]   = {4'b00000,temp[4]};//Cout
                data[4:0]   = {1'b0,y};           //计算结果
            end
```

```
            3'b110, 3'b111: // 没有运算
                data = {40{1'b1}}; // 全灭
        endcase
    end // always_comb
endmodule
```

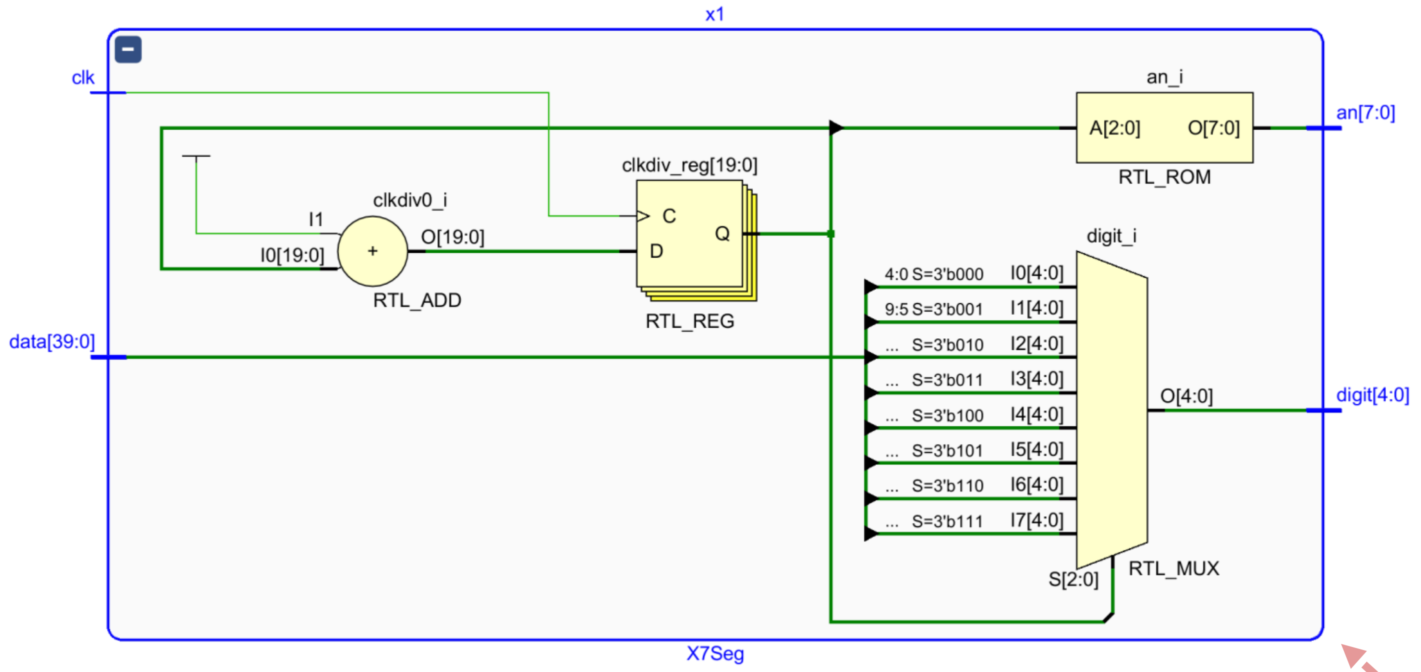


RTL 电路图


RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic

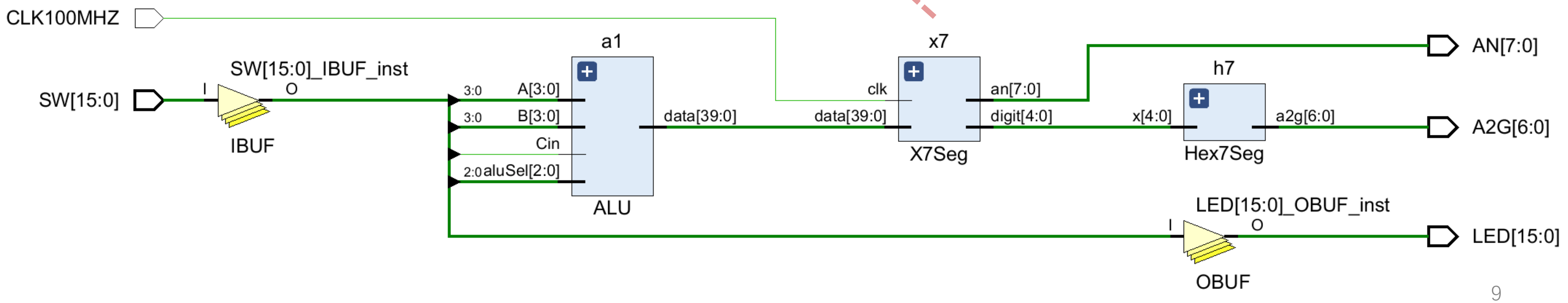

CLK100MHZ
AN
SW[3:0]
SW[7:4]
SW[8]
SW[15:13]
4
4
3
A
B
Cin
aluSel
data
ALU
wData
40
clk
an
data
digit
X7Seg
8
wDigit
5
x
a2g
Hex7Seg
A2G
7
LED
CLK100MHZ
SW[15:0]
SW[15:0]_IBUF_inst
I
O
IBUF
a1
3:0
A[3:0]
3:0
B[3:0]
Cin
2:0
aluSel[2:0]
ALU
data[39:0]
x7
clk
data[39:0]
an[7:0]
digit[4:0]
X7Seg
h7
x[4:0]
a2g[6:0]
Hex7Seg
AN[7:0]
A2G[6:0]
LED[15:0]_OBUF_inst
I
O
OBUF
LED[15:0]

RTL 电路图

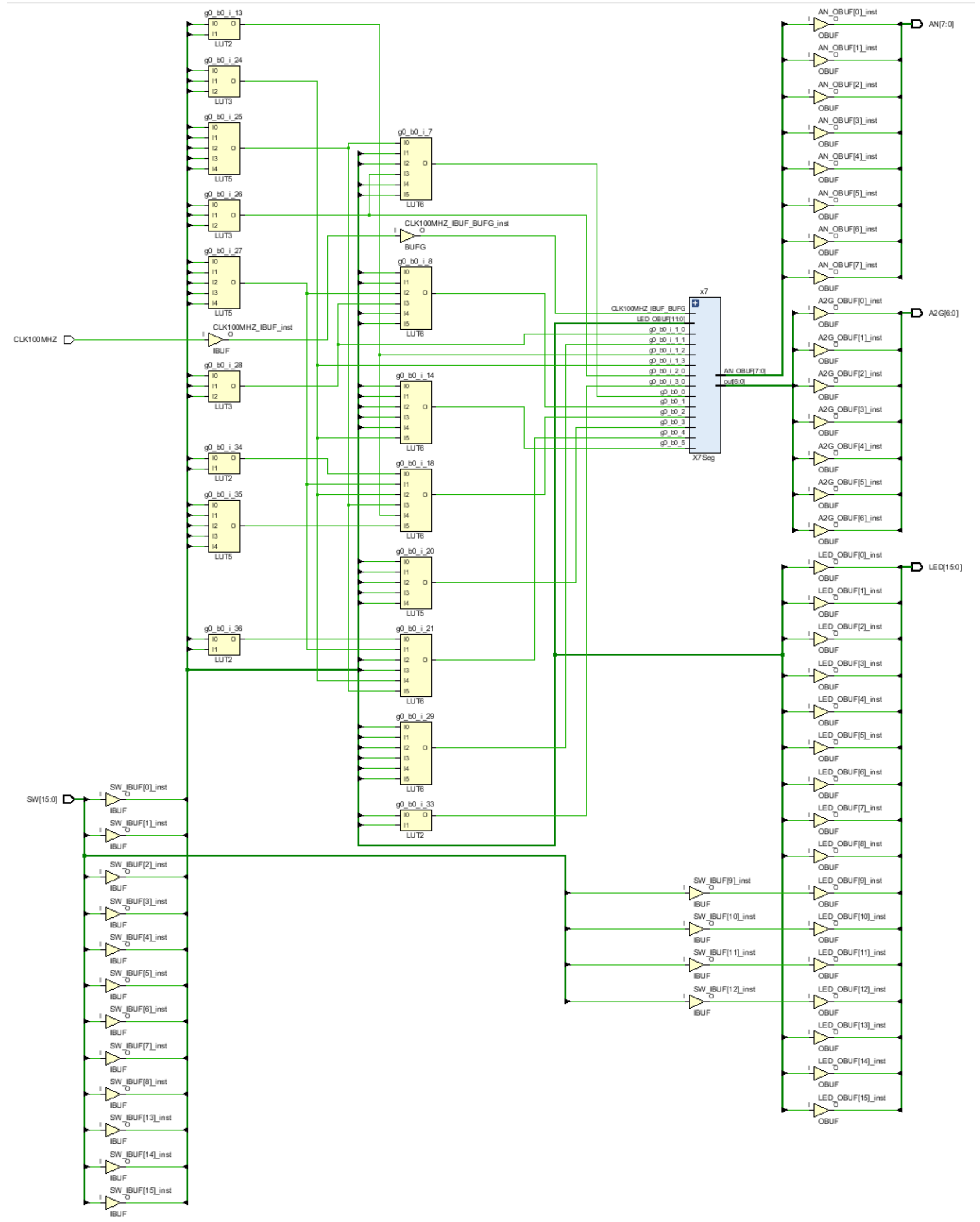


由基本的与门、或门、非门、MUX、比较器、运算单元、触发器等组成。

鼠标双击



“综合 “后 电路图



已将RTL级电路图的基本逻辑部件映射到FPGA的部件，即由:查找表**LUT**、**MUX**、**触发器**、**存储器**、**进位链**等组成。