lab1

ALU模块代码

lab1 > src > alu.v

```
37  module alu (a,b,aluc,s,z);
36    input [31:0] a,b;
35    input [3:0] aluc;
34    output [31:0] s;
33    output        z;
32    wire z;
31    wire [31:0] s;
30    assign  s = (aluc == 4'b0000)? a + b:
29                (aluc == 4'b1000)? a - b:
28                (aluc == 4'b0111)? a & b:
27                (aluc == 4'b0110)? a | b:
26                (aluc == 4'b0100)? a ^ b:
25                (aluc == 4'b0010)? b    :
24                (aluc == 4'b0001)? a << b:
23                (aluc == 4'b0101)? a >> b:
22                (aluc == 4'b1101)? $signed($signed(a) >>> $signed(b)):
21                0;
20   assign z = (s == 0);
19
18  /*
17    reg [31:0] s;
16    reg        z;
15    always @ (a or b or aluc)
14      begin
13        casex (aluc)  //
12          4'b0000: s = a + b;             //0000 ADD
11          4'b1000: s = a - b;             //1000 SUB
10          4'b0111: s =      ;             //0111 AND
 9          4'b0110: s =     ;              //0110 OR
 8          4'b0100: s =      ;             //0100 XOR
 7          4'b0010: s = b;                 //0010 LUI:{lui_imm ,12'b0}
 6          4'b0001: s =       ;            //0001 SLL: rd <- (rs1 << rs2)
 5          4'b0101: s =       ;            //0101 SRL: rd <- (rs1 >>rs2) (logical)
 4          4'b1101: s =        ;           //1101 SRA: rd <- (rs1 >> rs2) (arithmetic)
 3          default: s = 0;
 2        endcase
 1        if (s == 0 )  z = 1;
38        else z = 0;
 1      end
 2      */
 3
 4  endmodule
```

ALU实例化

```
1    //例化alu
43   alu alu_inst(
1      .a(32'h80000005),
2      .b(alub),
3      .aluc(aluc),
4      .s(aluout),
5      .z(zero)
6    );
```

例化的指令ROM