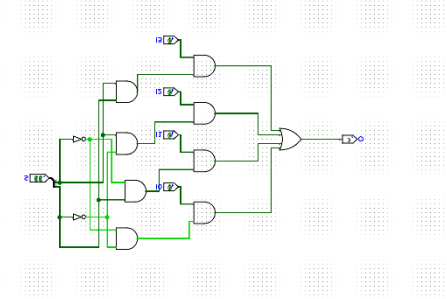
Lab1-1 Report

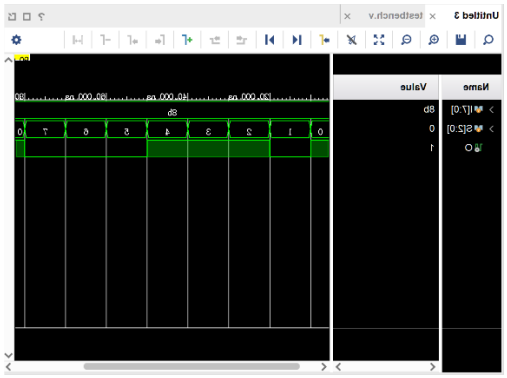


图表 1 四路选择器原理图

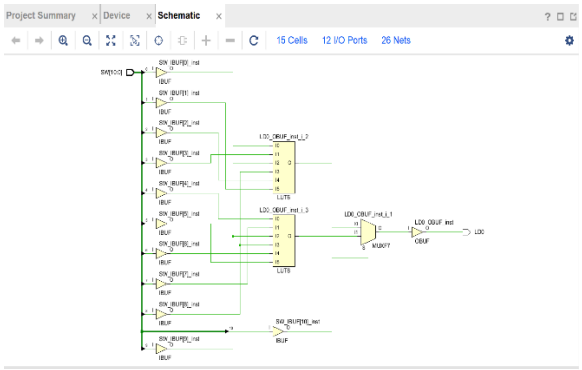
```
wire [1:0] Mux4_out;
MUX4T1_1 fir(//第一个四路选择器
   .I0(I[0]),
   .I1(I[1]),
   .I2(I[2]),
   .I3(I[3]),
   .S(S[1:0]),
   .O(Mux4_out[0])
);

MUX4T1_1 sec(//第二个四路选择器
   .I0(I[4]),
   .I1(I[5]),
   .I2(I[6]),
   .I3(I[7]),
   .S(S[1:0]),
   .O(Mux4_out[1])
);
```

```
assign O = S[2] ? Mux4_out[1] : Mux4_out[0] ; //另建一个两路选择器将结果链接
```

图表 2 八路选择器代码解释



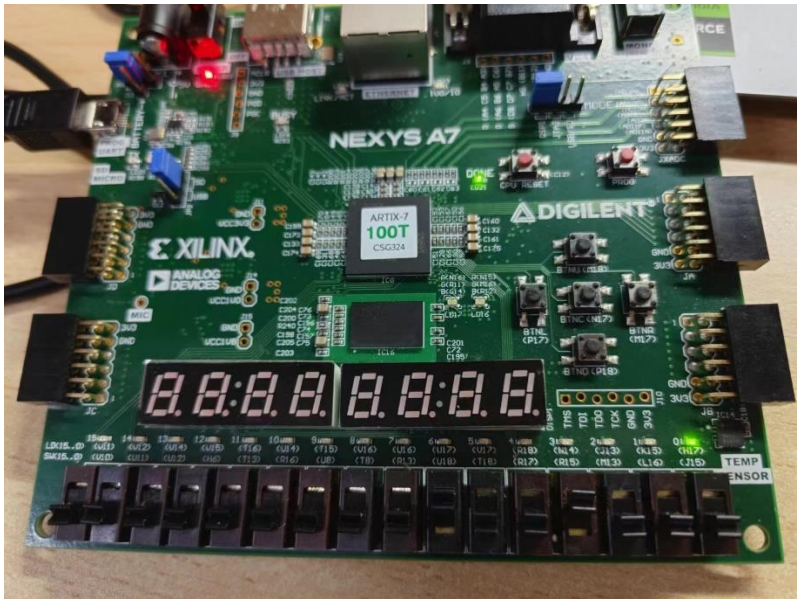
图表 3Vivado 仿真



图表 4 电路图

```verilog
module Testbench;

    reg [7:0] I;
    reg [2:0] S;
    wire O;

    initial begin
        I = 8'b10001011;
        S = 3'b000;
        repeat (8) begin
            #10
            S = S + 1;
        end
        #10
        $finish;
    end

    Mux8T1_1 dut(
        .I(I),
        .S(S),
        .O(O)
    );
```

图表 5testbench 代码，使用循环，每次循环 s+1，遍历八次



图表 6 上板验证

思考问题:

1. 先对 S 取反，后再用第一个与门处理 A 和 S，再用第二个与门处理 B 和 S 非，最后再使用或门将两个结果结合。
2. 使用两个 MUX2T1_1，每个分别处理两个输入信号。第一个 MUX2T1_1 处理输入 A 和 B，第二个 MUX2T1_1 处理输入 C 和 D。这两个 MUX2T1_1 的输出然后被送入第三个 MUX2T1_1，这个 MUX2T1_1 使用第二个控制信号进行选择。
3. 先使用两个 MUX4T1_1，再使用一个 MUX2T1_1 将两个 MUX4T1_1 所得结果综合，得到最终结果。
4. 由两个 MUX2^(m-1)T1_n，在使用一个 MUX2T1_1 将两个门所得结果综合。

或者使用 2^m-1 个 MUX2T1_1（即将所有多输入选择器拆解）