

同济大学计算机系

数字逻辑课程实验报告



学 号 2154312

姓 名 郑博远

专 业 计算机科学与技术

授课老师 郭玉臣

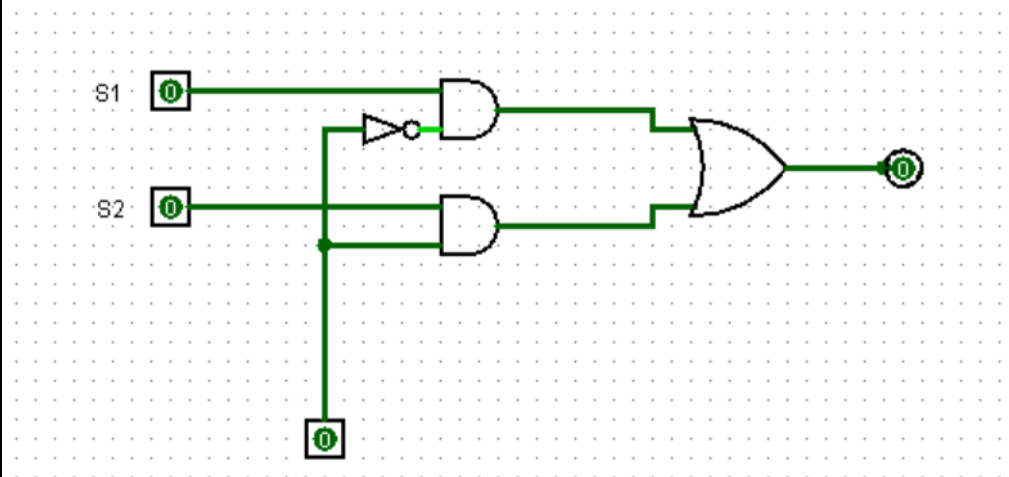
一、实验内容

(1) 桶形移位器实验：桶形移位器是一种组合逻辑电路，通常作为微处理器 CPU 一部分。它具有 n 个数据输入和 n 个数据输出，以及指定如何移动数据的控制输入，指定移位方向、移位类型（循环、算术还是逻辑移位）以及移动的位数等。本次实验探究使用 logisim 画出 8 位桶形移位器原理图，并用 verilog 编写模块实现 32 位桶形移位器。

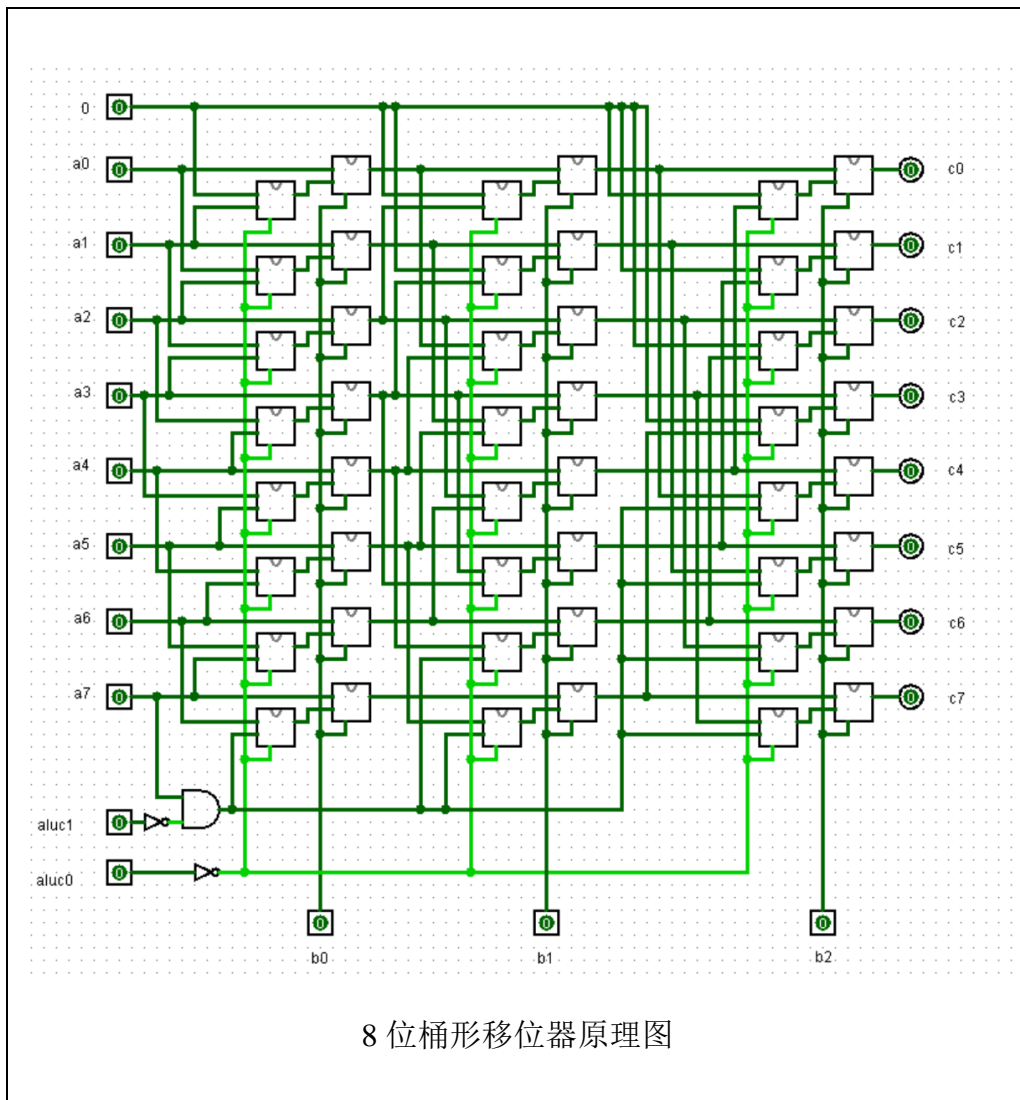
二、硬件逻辑图

(1) 桶形移位器实验：

本题中选择了自己设计子电路来实现 MUX，也可以使用 logisim 自带的 MUX 来组件桶形移位器电路。



2 位数据选择器子电路设计原理图



三、模块建模

(1) 桶形移位器实验：

a 为 32 位的输入数据，c 为移位后的 32 位输出数据；aluc 为控制移位方式的输入（其中 aluc[0]选择左移或右移、aluc[1]控制右移时进行算术右移还是逻辑右移）；5 位输入信号 b 控制移位的位数。

```
module barrelshifter32(
    input signed [31 : 0] a,
    input [4 : 0] b,
    input [1 : 0] aluc,
    output reg[31 : 0] c
```

```

);

always@(*)
begin
    case(aluc)
        2'b00 : c = a >>> b;
        2'b10 : c = a >> b;
        2'b01 : c = a <<< b;
        2'b11 : c = a << b;
    endcase
end
endmodule

```

四、测试模块建模

(1) 桶形移位器实验:

```

`timescale 1ns / 1ps

module barrelshifter32_tb();
    reg [31 : 0] a;
    reg [5 : 0] b;
    reg [2 : 0] aluc;
    wire [31 : 0] c;

    initial
    begin
        a = 32'b1111_0000_1100_0011_1001_0110_0001_1110;
        for(b = 0; b <= 4'b1111; b = b + 4'b00001)
            for(aluc = 0; aluc <= 2'b11; aluc = aluc + 2'b01)

```

```

begin
    #5;
end

end

barrelshifter32 barrelshifter_inst(a, b, aluc, c);

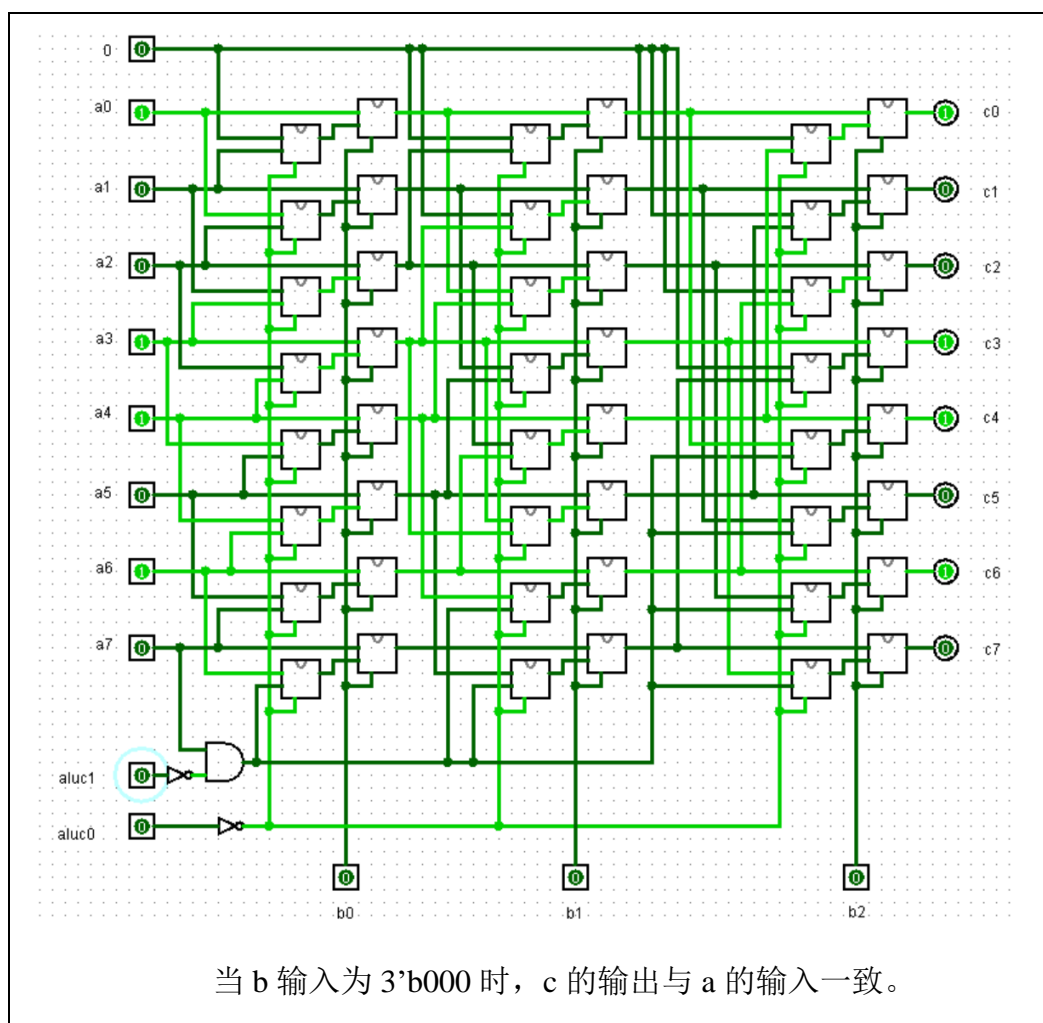
endmodule

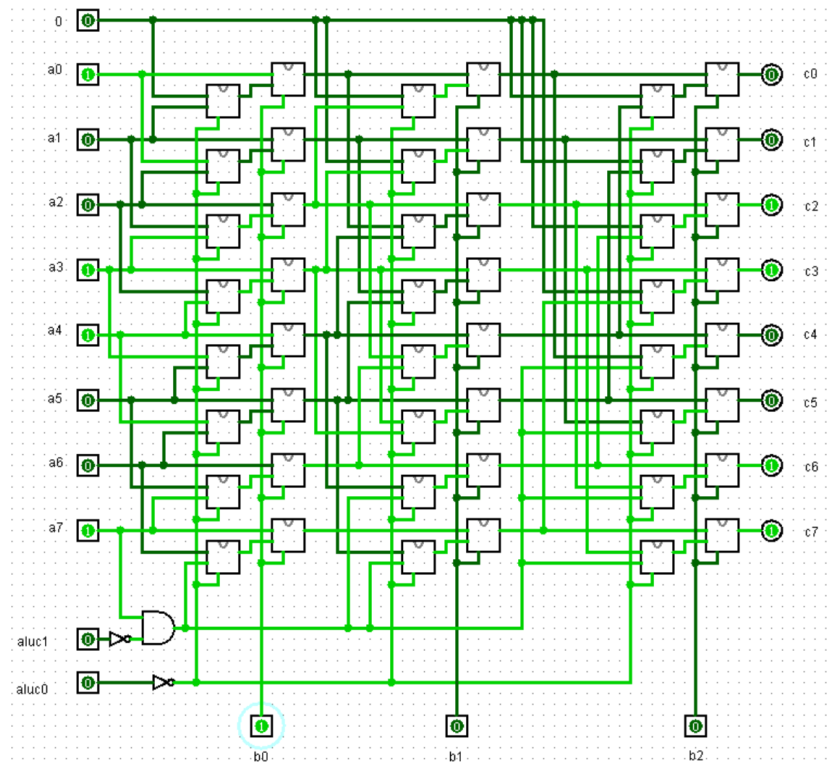
```

五、实验结果

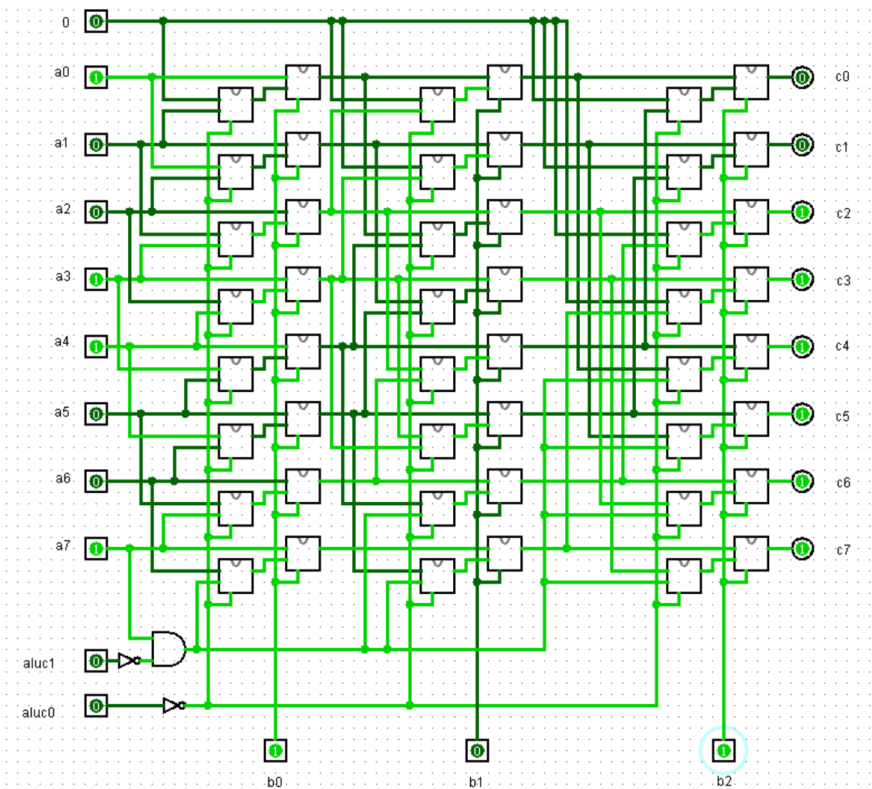
(1) 桶形移位器实验:

a) logisim 逻辑验证图

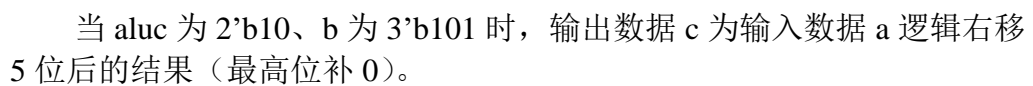
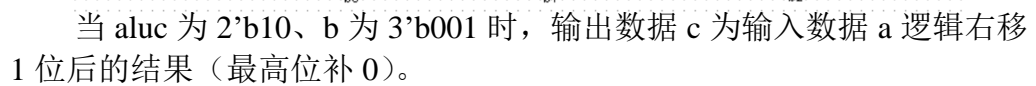


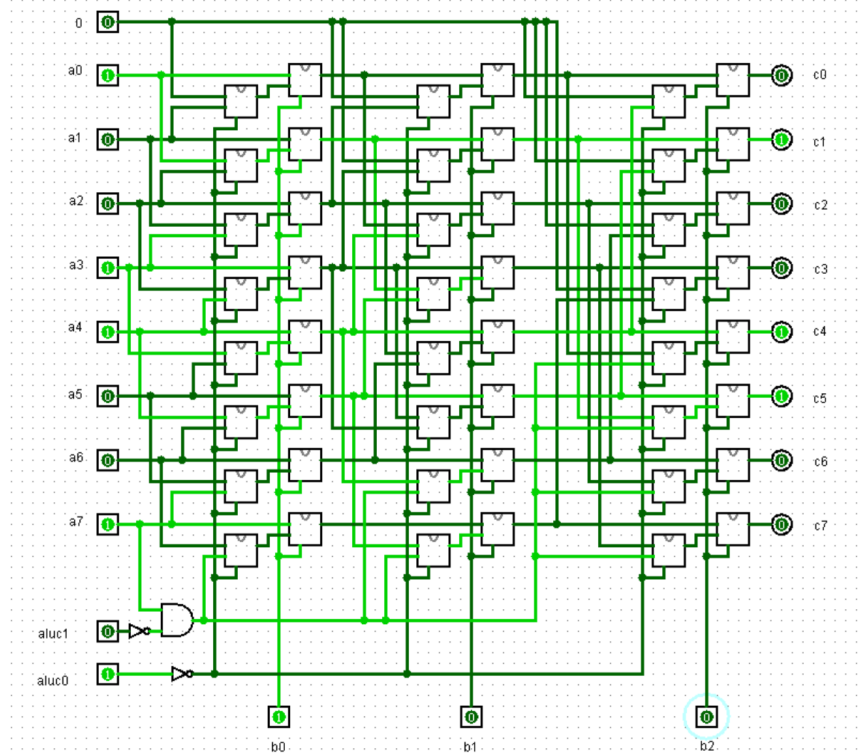


当 aluc 为 $2'b00$ 、b 为 $3'b001$ 时，输出数据 c 为输入数据 a 算术右移 1 位后的结果（最高位补符号位 1）。

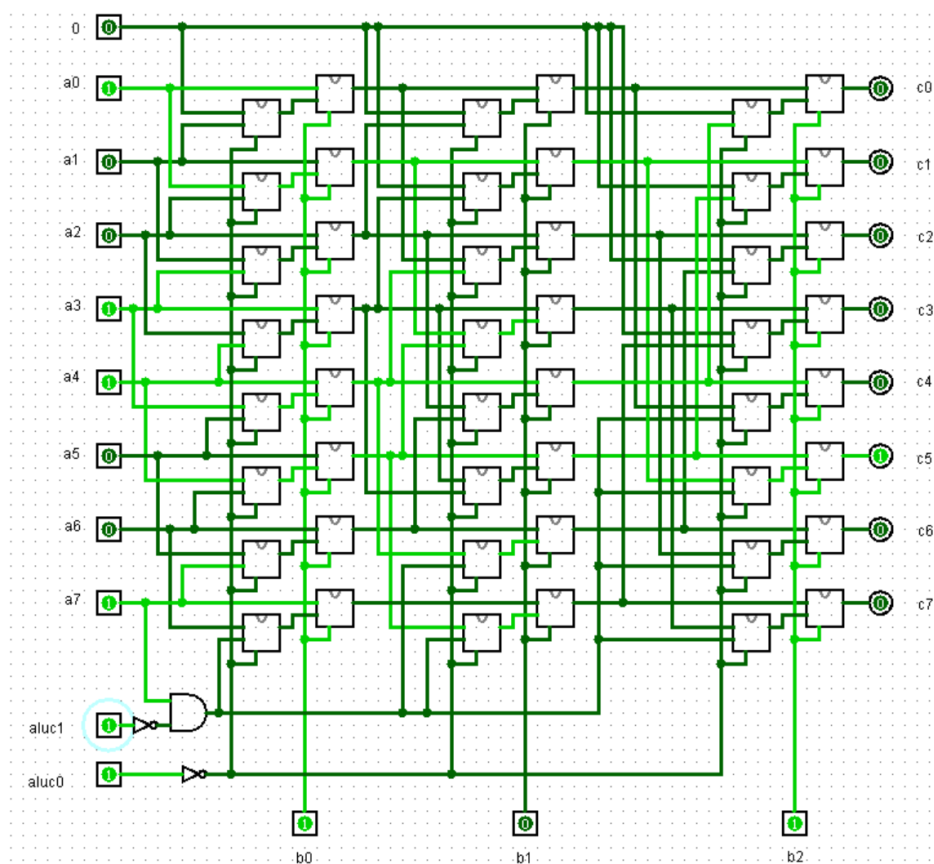


当 aluc 为 $2'b00$ 、b 为 $3'b101$ 时，输出数据 c 为输入数据 a 算术右移 5 位后的结果（最高位补符号位 1）。

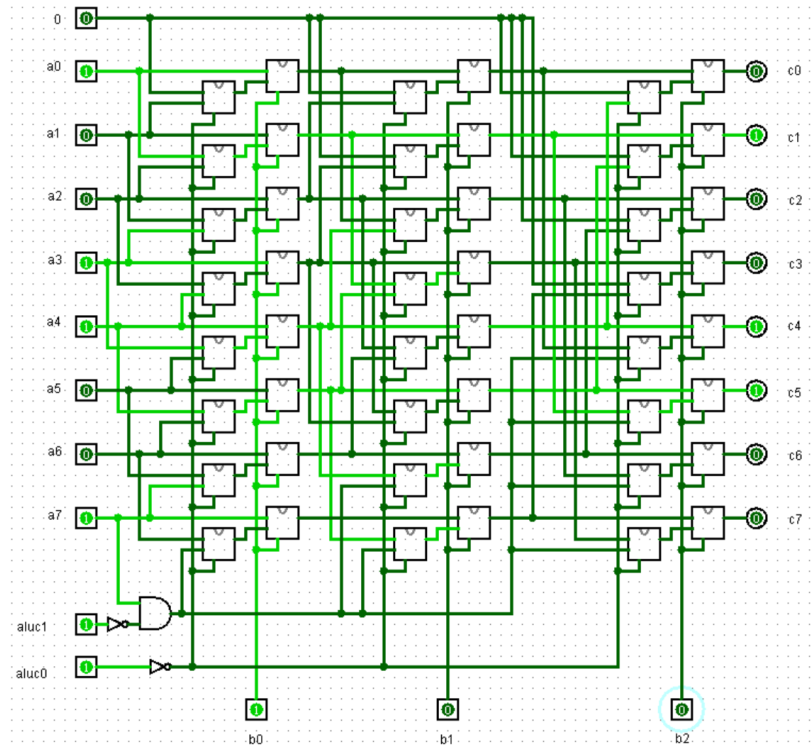




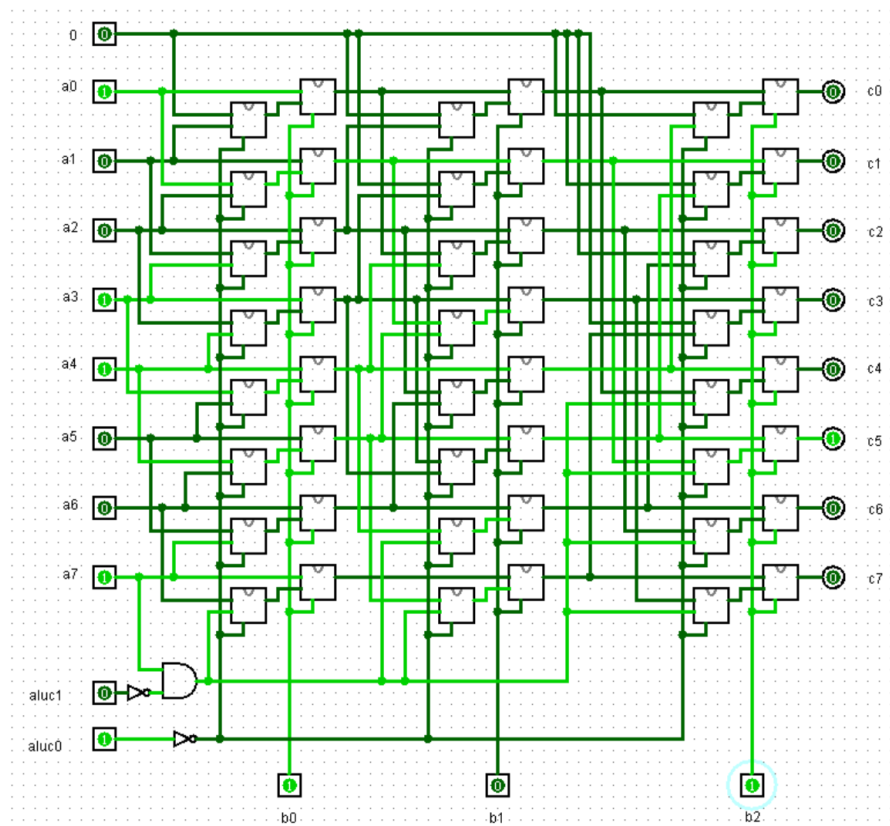
当 aluc 为 $2'b11$ 、b 为 $3'b001$ 时，输出数据 c 为输入数据 a 逻辑左移 1 位后的结果（最低位补 0）。



当 aluc 为 $2'b11$ 、b 为 $3'b101$ 时，输出数据 c 为输入数据 a 逻辑左移 5 位后的结果（最低位补 0）。

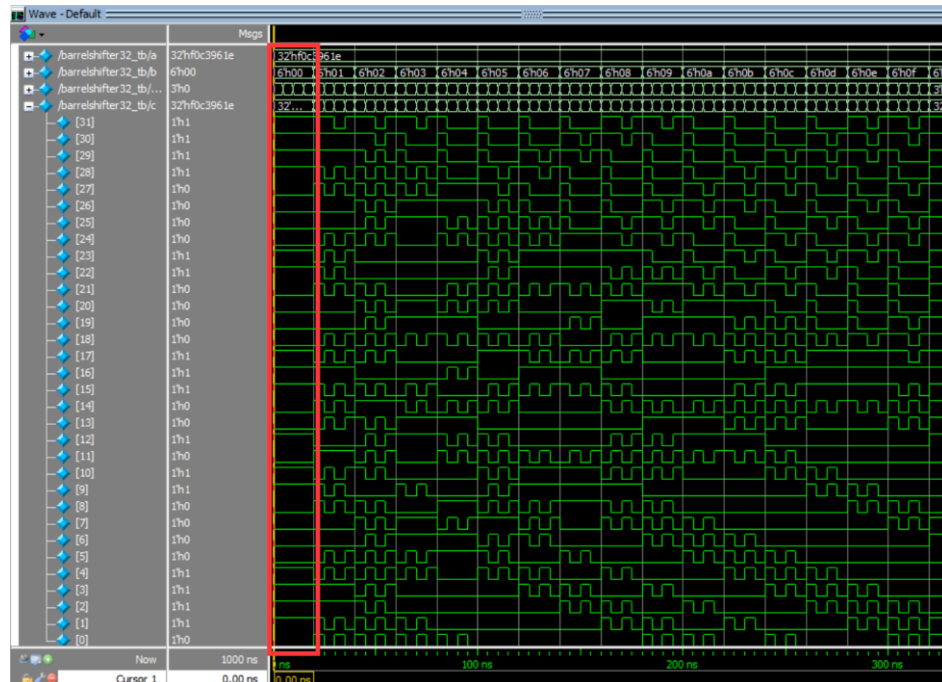


当 aluc 为 2'b01、b 为 3'b001 时，输出数据 c 为输入数据 a 算数左移 1 位后的结果（最低位补 0）。



当 aluc 为 2'b01、b 为 3'b101 时，输出数据 c 为输入数据 a 算数左移 5 位后的结果（最低位补 0）。

b) modelsim 仿真波形图



观察到当输入信号 `b` 为 `5'b00000` 时，输入数据 `a` 不发生位移，输出数据 `c` 与输入数据 `a` 保持一致。



`aluc` 在每次循环中由 `2'b00` 变为 `2'b11` (`tb` 中设为三位是为了防止在 `for` 循环终止条件判断时因为溢出回到 `2'b00` 而产生死循环)，分别对应算术右移、算术左移、逻辑右移、逻辑左移。

