

同济大学计算机系

数字逻辑课程实验报告



学 号 2350752

姓 名 田思宇

专 业 计算机科学与技术

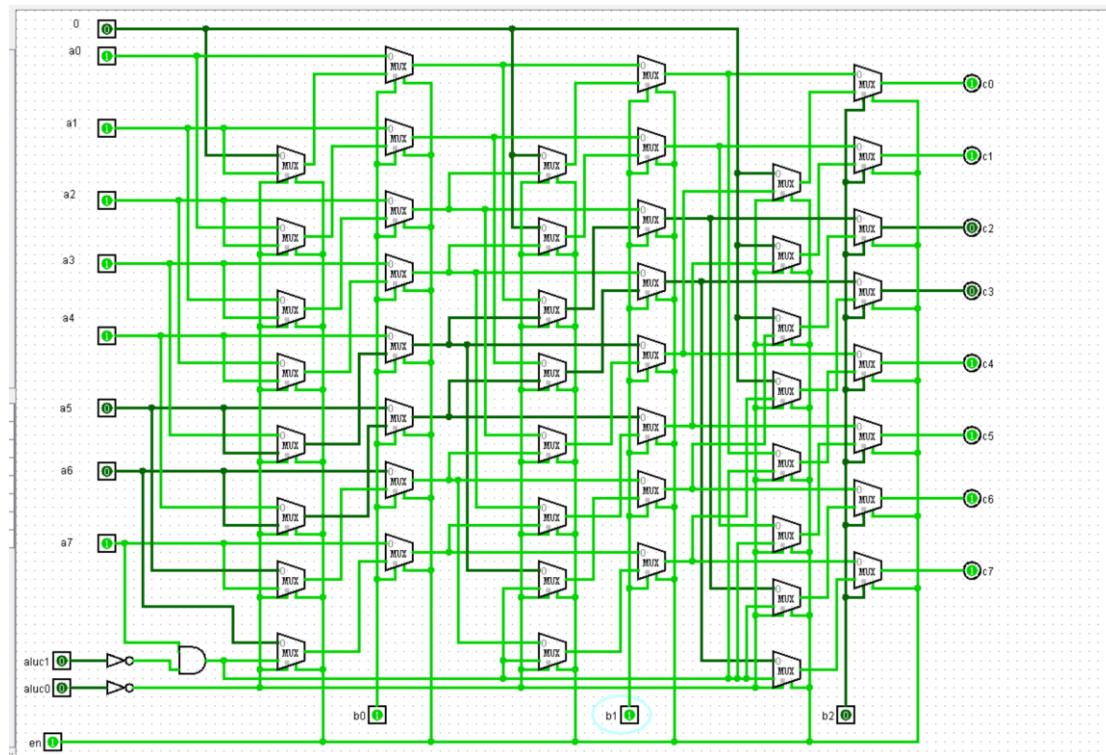
授课老师 张冬冬

一、实验内容

深入了解桶形移位器的原理，在学习完 4 位桶形移位器原理后使用 logisim 软件搭建一个 8 位的桶形移位器，学习使用 Verilog HDL 语言设计实现一个 32 位桶形移位器。

二、硬件逻辑图

(实验步骤中要求用 logisim 画图的实验，在该部分给出 logisim 原理图，否则该部分在实验报告中不用写)



三、模块建模

(该部分要求对实验中建模的所有模块进行功能描述，并列出各模块建模的 verilog 代码)

模块功能描述：

这个模块是 32 位桶形移位器模块，首先定义 32 位输入 a，因为要产生 32 位的位移，所以位移信号是 5 位的 b，其次是两位的控制信号，aluc[1]控制算数位移还是逻辑位移、aluc[0]控制左移还是右移，然后就是 32 位的输出信号 c，采用行为级描述，由于模块中采用 if else 语言来实现位移的逻辑描述，所以要定义一个中间变量 temp，代码逻辑是先判断 aluc 的信号（00 代表算数右移，10 代表逻辑右移，01 和 11 都代表左移）。然后写第二层 if else 语言，为了便于读程序，用了 begin end 语句块把第二层 if else 包起来了。算数右移是通过逻辑右移和与运算结合来实现的。最后实现的功能是根据控制信号 b 转成十进制来决定移动几位。

```
module barrelshifter32(  
    input [31:0] a,      //32 位原始输入数据  
    input [4:0] b,       //5 位位移量  
    input [1:0] aluc,   //2 位控制信号  
    output [31:0] c     //32 位输出结果  
)
```

```

input [4:0] b,      //5 位输入信号, 控制移位的位数
input [1:0] aluc,    //2 位输入信号, 控制移位的方式
output reg [31:0] c  //32 位移位后的输出数据
);
reg [31:0] temp;
always@(*)
begin
if(aluc[1]==1'b0&&aluc[0]==1'b0)
begin
if(b[0]==1'b1)
temp=((a>>1)|32'b10000000000000000000000000000000);
else
temp=a;
if(b[1]==1'b1)
temp=((temp>>2)|32'b11000000000000000000000000000000);
if(b[2]==1'b1)
temp=((temp>>4)|32'b11110000000000000000000000000000);
if(b[3]==1'b1)
temp=((temp>>8)|32'b11111110000000000000000000000000);
if(b[4]==1'b1)
temp=((temp>>16)|32'b11111111111110000000000000000000);
end
if(aluc[1]==1'b1&&aluc[0]==1'b0)
begin
if(b[0]==1'b1)
temp=(a>>1);
else
temp=a;
if(b[1]==1'b1)
temp=(temp>>2);
if(b[2]==1'b1)
temp=(temp>>4);
if(b[3]==1'b1)
temp=(temp>>8);
if(b[4]==1'b1)
temp=(temp>>16);
end
if((aluc[1]==1'b0&&aluc[0]==1'b1)|(aluc[1]==1'b1&&aluc[0]==1'b1))
begin
if(b[0]==1'b1)
temp=(a<<1);
else
temp=a;
if(b[1]==1'b1)

```

```

temp=(temp<<2);
if(b[2]==1'b1)
temp=(temp<<4);
if(b[3]==1'b1)
temp=(temp<<8);
if(b[4]==1'b1)
temp=(temp<<16);
end
c=temp;
end
endmodule

```

四、测试模块建模

(要求列写各建模模块的 test bench 模块代码)

```

`timescale 1ns/1ns
module shifter32_tb;
reg [31:0] a;
reg [3:0] b;
reg [1:0]aluc;
wire [31:0] temp;
initial
begin
a=32'b1111000000000000000000000000000010111;
b=4'b0110;
aluc=2'b00;
#50
b=4'b0110;
aluc=2'b10;
#50
b=4'b0110;
aluc=2'b01;
#50
b=4'b0110;
aluc=2'b11;
#50
b=4'b0001;
aluc=2'b00;
#50
b=4'b0001;
aluc=2'b10;
#50
b=4'b0001;

```

```

aluc=2'b01;
end
barrelshifter32 uut(a,b,aluc,temp);
endmodule

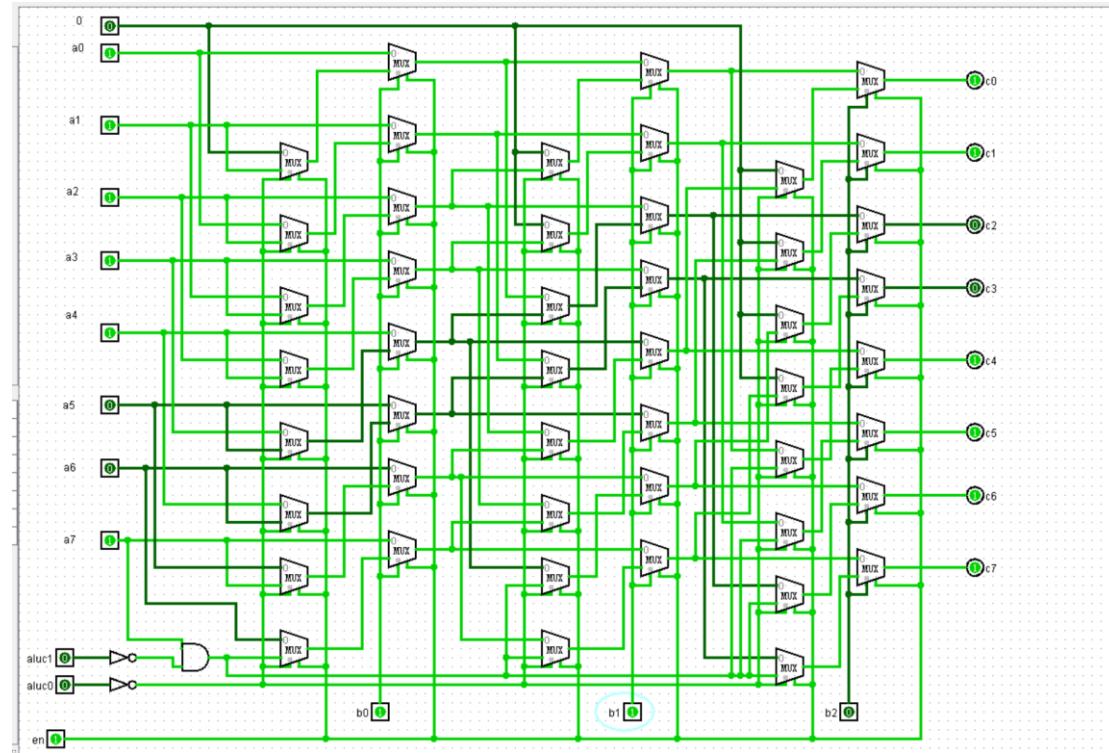
```

五、实验结果

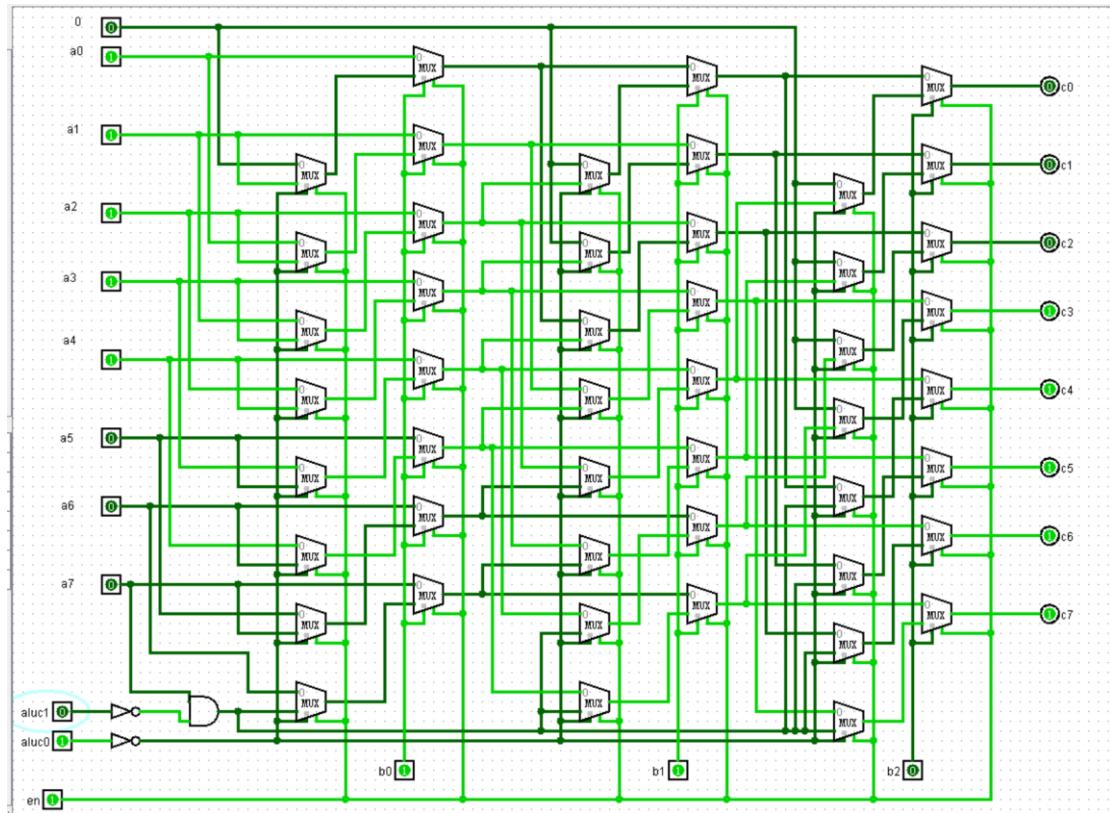
(该部分可截图说明，要求 logisim 逻辑验证图、modelsim 仿真波形图、以及下板后的实验结果贴图（实验步骤中没有下板要求的实验，不需要下板贴图）)

logisim 逻辑验证图：

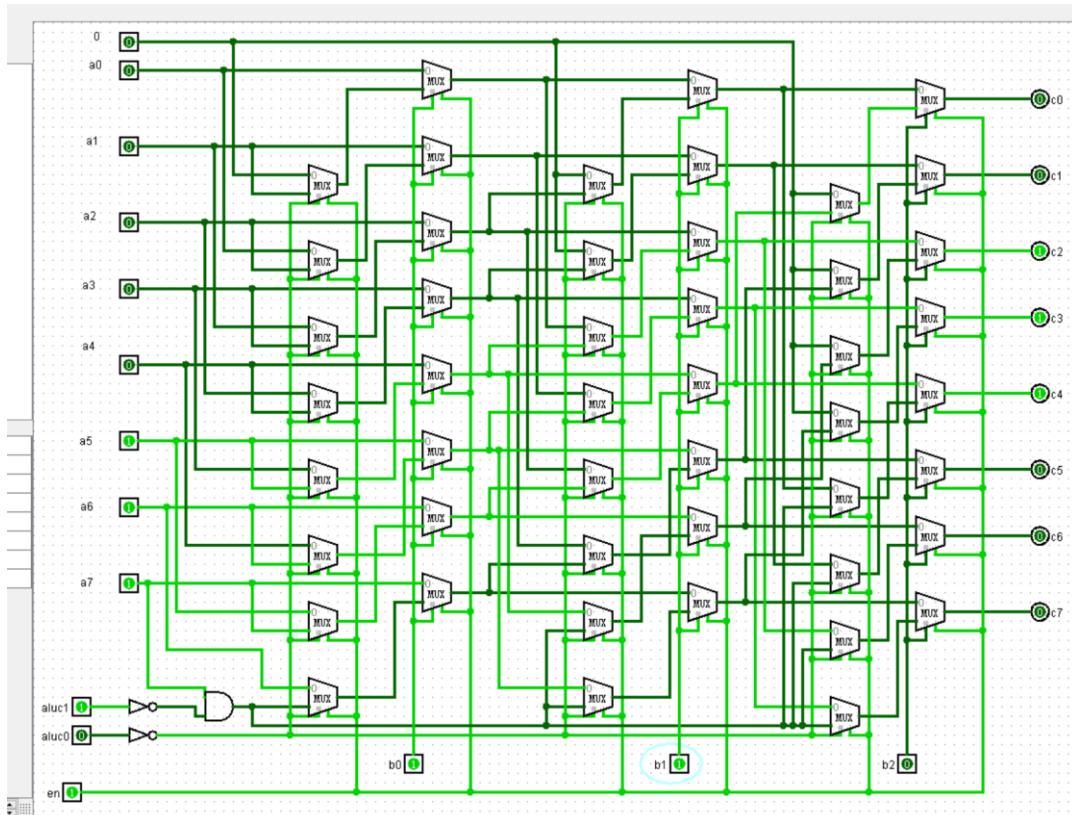
1、这里是八位的桶形移位器，输入是 10011111，控制信号表示算数右移，移动信号位 011，代表移动 3 位，最终结果为 11110011，结果正确



2、输入是 00011111，控制信号表示左移，移动信号位 011，代表移动 3 位，最终结果为 11111000，结果正确



3、输入是 11100000，控制信号表示逻辑右移，移动信号位 011，代表移动 3 位，最终结果为 00011100，结果正确



由于情况太多，这里只列举出三种不同数据以及不同移动方式的逻辑验证，均正确。

modelsim 仿真波形图：

