南京理工大学计算机学院计算机逻辑基础实验报告

姓名：阚东 学号：919106840420

实验一：译码器（或编码器）的设计及应用实验

1. 实验目的

学习组合逻辑电路译码器或编码器的设计方法及应用。

2. 实验内容

（1）用 verilog 来实现 2-4 译码器或 3-8 译码器或七段译码器；

（2）用 verilog 来实现 4-2 编码器或 8-3 编码器；

3. 实验原理

将某一特定的代码译成原始的信息，称为译码过程。译码过程可以通过译码器电路实现。译码的过程实际上就是编码过程的逆过程，即把一组一定规律排开列的二进制数还原为原始信息的过程，如 3-8 译码器、七段译码器等。

真值表：

输入in[7:0]								输出out[2:0]		
in[7]	in[6]	in[5]	in[4]	in[3]	in[2]	in[1]	in[0]	out[2]	out[1]	out[0]
1	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0

4. 实验过程

(1)Verilog 代码

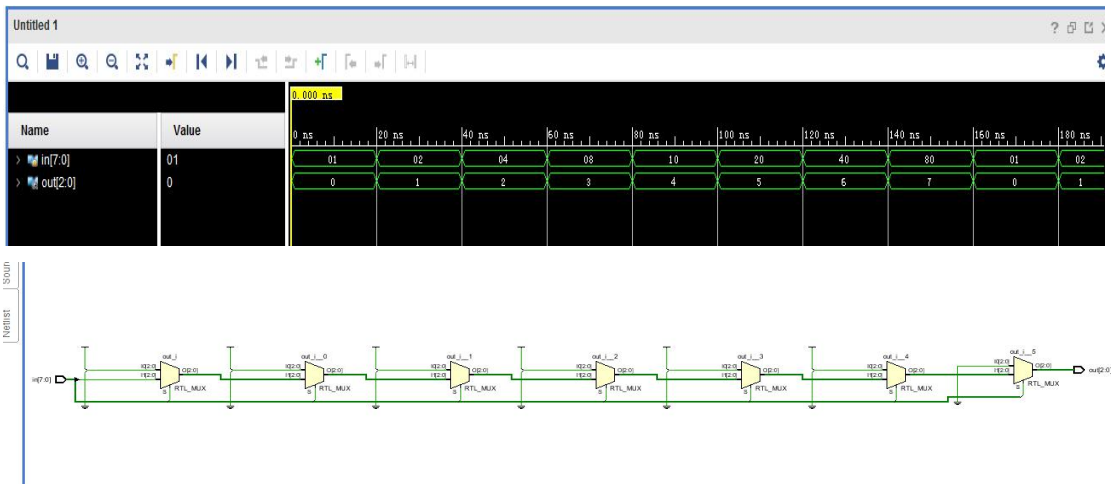
```
module project83(
input [7:0] in,
output [2:0] out
    );
```

```
    reg [2:0] out;
    always@(in)
    begin
        if(in[0]) out = 3'b000;
        else if (in[1]) out = 3'b001;
        else if (in[2]) out = 3'b010;
        else if (in[3]) out = 3'b011;
        else if (in[4]) out = 3'b100;
        else if (in[5]) out = 3'b101;
        else if (in[6]) out = 3'b110;
        else if (in[7]) out = 3'b111;
        else out = 3'bxxx;
    end
endmodule
```

(2)仿真文件

```
module project83_tb();
reg[7:0] in;
wire[2:0] out;
project83 u0 (.in(in),.out(out));
always begin
    in = 8'b00000001;#20;
    in = 8'b00000010;#20;
    in = 8'b00000100;#20;
    in = 8'b00001000;#20;
    in = 8'b00010000;#20;
    in = 8'b00100000;#20;
    in = 8'b01000000;#20;
    in = 8'b10000000;#20;
end
endmodule
```

5. 实验结果记录



实验二：数据通道选择器的设计及应用实验

1.实验目的

学习组合逻辑电路数据选择器的设计方法及应用。

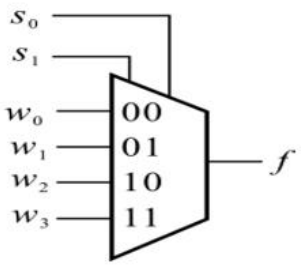
2.实验内容

用 verilog 来实现 2 选 1 或 4 选 1 或 8 选 1 等数据选择器。

3.实验原理

在数字系统中，经常需要把多个不同通道的信号发送到公共的信号通道上，通过多路选择器可以完成这一功能。在数字系统设计中，常用 CASE 语句和 IF 语句描述多路选择器。多路选择器是由几路数据输入、一位或多位的选择控制，和一路数据输出所组成的，如 2 选 1、4 选 1、8 选 1 等多路选择器，它们的符号和真值表如下所示（例）：

逻辑符号及真值表：



(a) 逻辑符号

s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

(b) 真值表

4.实验过程

(1)Verilog 代码

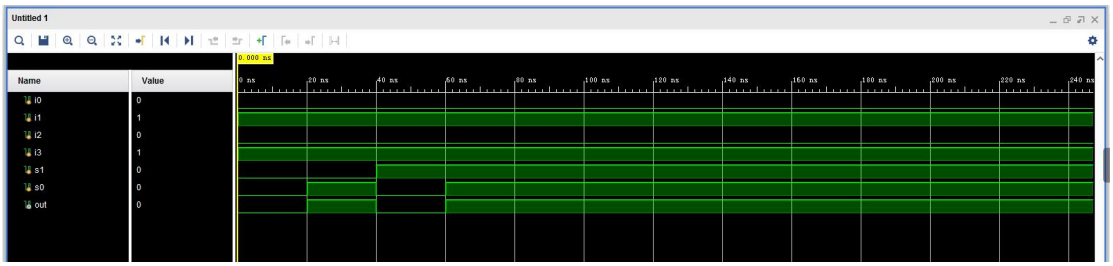
```
module mux4_1(out,i0,i1,i2,i3,s1,s0);
output out;
input i0,i1,i2,i3;
input s1,s0;
reg out;
always @(s1 or s0 or i0 or i1 or i2 or i3)
```

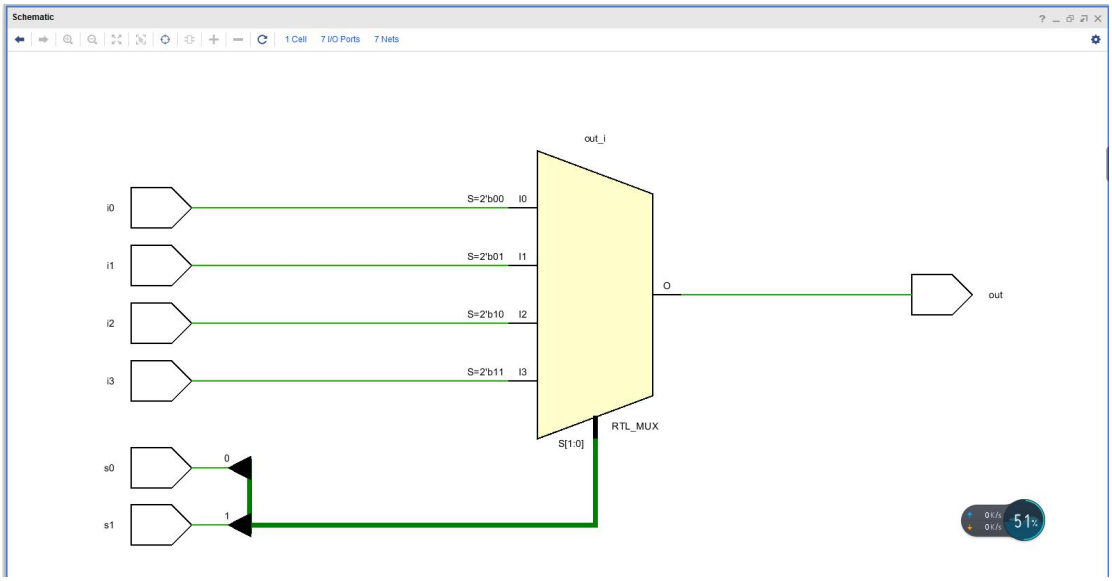
```
begin
    case({s1,s0})
        2'b00:out = i0;
        2'b01:out = i1;
        2'b10:out = i2;
        2'b11:out = i3;
        default:out=1'bx;
    endcase
end
endmodule
```

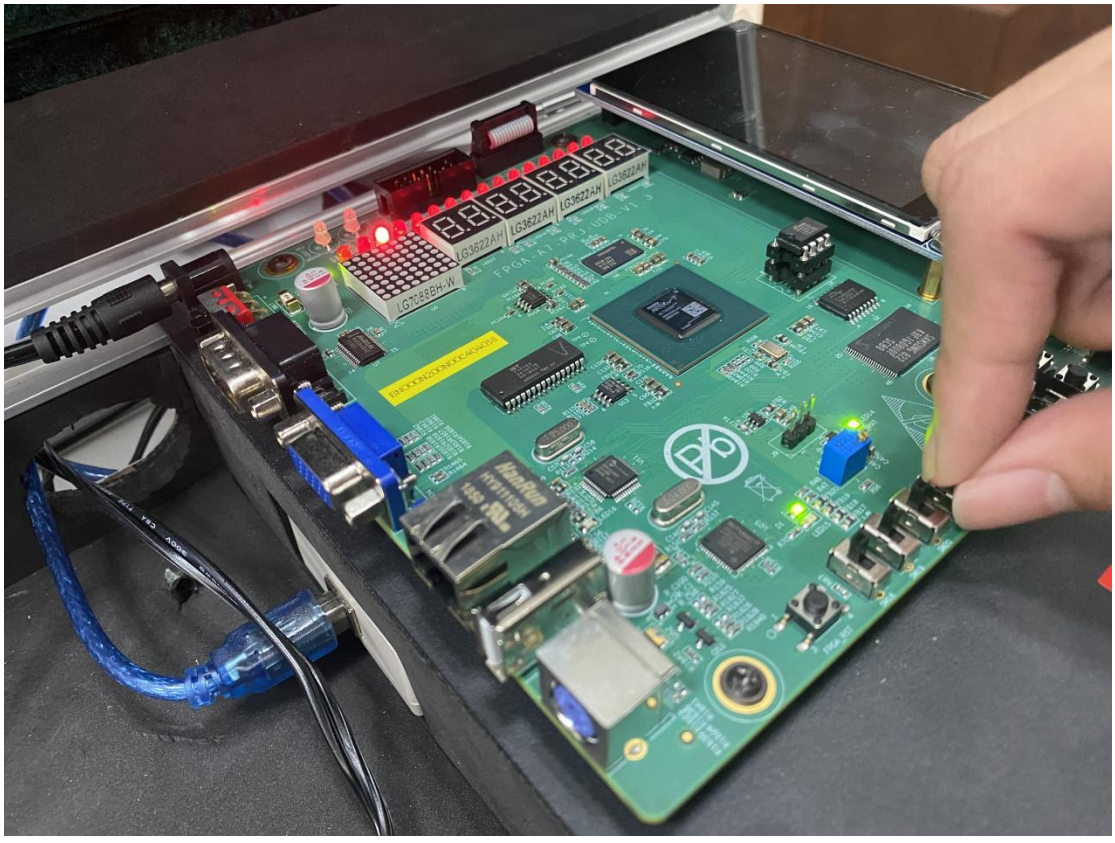
(2)仿真文件

```
module mux4_1_tb;
reg i0,i1,i2,i3;
reg s1,s0;
wire out;
mux4_1 unit(
    .i0(i0),
    .i1(i1),
    .i2(i2),
    .i3(i3),
    .s1(s1),
    .s0(s0),
    .out(out)
);
initial
    begin
    i0=2'b00;
    i1=2'b01;
    i2=2'b10;
    i3=2'b11;
    s1=1'b0;s0=1'b0;
    #20
    s1=1'b0;s0=1'b1;
    #20
    s1=1'b1;s0=1'b0;
    #20
    s1=1'b1;s0=1'b1;
    #20;
    end
endmodule
```

5. 实验结果记录


Untitled 1
Name
Value
i0
i1
i2
i3
s1
s0
out
0
1
0
1
0
0
0
0 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
140 ns
160 ns
180 ns
200 ns
220 ns
240 ns


Schematic
1 Cell
7 I/O Ports
7 Nets
out_i
i0
i1
i2
i3
s0
s1
S=2'b00
S=2'b01
S=2'b10
S=2'b11
I0
I1
I2
I3
O
out
RTL_MUX
S[1:0]
0
1
51%


LG3622AH
LG7288BH-N

实验三：计数器（或移位寄存器）的设计及应用实验

1.实验目的

学习时序逻辑电路计数器的设计方法及应用。

2.实验内容

(1)用 verilog 生成一个四位二进制计数器，同步清零；

(2)用 verilog 生成一个十进制计数器，异步清零；

(3)用 verilog 生成一个异步可逆十进制计数器，具有同步置数功能；

(4)用 verilog 生成一个模 100 的计数器，具有清零、预置功能；

(5)用 Verilog 生成一个移位寄存器（左移或右移）；

(6)用 Verilog 实现分频器。

3.实验原理

在数字系统设计中，计数器和寄存器是最常用的两类功能器件，也是最常见的时序逻辑元件。计数器是一种能统计输入脉冲个数的时序电路，它可以记录特定事件的发生次数，产生控制系统中不同任务的时间间隔，记录特定事件之间的时间间隔等，它可以用于定时器、分频器、程序控制器、信号发生器等多种数字设备中。计数器按脉冲的作用方式分类，可分为同步计数器和异步计数器。在同步计数器中，各个触发器的时钟输入端和同一个时钟脉冲源相连，因而所有触发器状态（即计数器状态）的改变都与时钟脉冲同步。而在异步计数器中，有的触发器直接受输入计数脉冲控制，有的是利用其他触发器输出作为时钟输入信号，因此所有触发器状态的改变有先有后，是异步的

4. 实验过程

(1)Verilog 代码

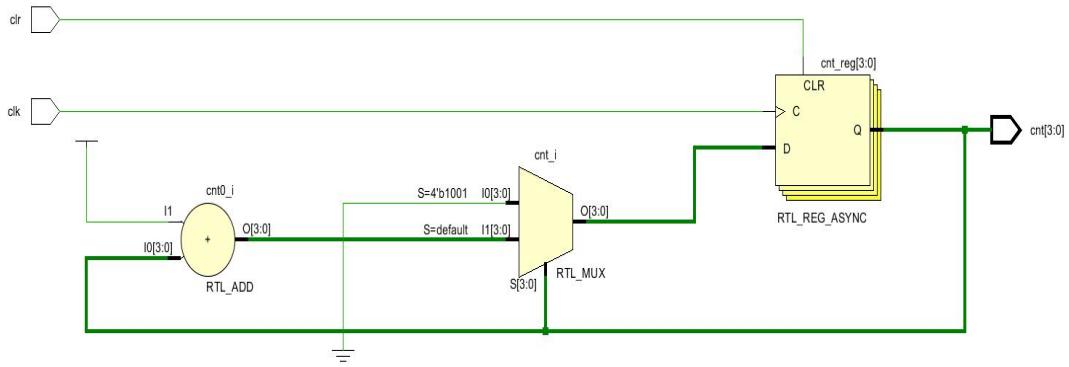
```
module count10(
input clk,input clr, output[3:0] cnt
    );
    reg[3:0] cnt;

    always @(posedge clk or posedge clr)
    begin
        if(clr)
            cnt <= 4'b0000;
        else
            if(cnt==4'b1001)
                cnt <= 4'b0000;
            else
                cnt<= cnt+1;
    end
//    assign q=cnt;
endmodule
```

(2)仿真文件

该实验只上板，未仿真

5. 实验结果记录



实验四：状态机的设计及应用实验

1. 实验目的

学习时序逻辑电路中 Mealy 和 Moore 状态机的设计方法及应用。

2. 实验内容

(1) 设计一个序列检测状态机，实现 Mealy 型或 Moore 型。

(2) 设计一个让发光灯显示不同状态的状态机。比如：

要求有 3 个显示状态， 且 3 个状态循环切换：a->b->c->a

a. 选 2 个 led 灯按 1s 的间隔闪烁 5 次；

b. 选 3 个 led 灯按模 8 递增计数；

c. 选 4 个 led 灯按右移方式逐个点亮；

3. 实验原理

Mealy 型状态机的输出由状态机的输入和状态机的状态共同决定。Moore 型状态机与 Mealy 型状态机区别在于，Moore 型状态机的输出仅与状态机的状态有关，与状态机的输入无关。序列检测器用于由二进制码组成的脉冲序列信号，当序列检测器连续收到一组串行二进制码后，如果这组码与检测器中的目标值相同则输出 1，否则输出 0。

4. 实验过程

⑴ Verilog 代码

```
module moore_0420(
    input clk, input clr, input din, output dout
    );
    reg dout;
    reg [2:0] cs,ns;
    parameter
s0=3'b000,s1=3'b001,s2=3'b010,s3=3'b011,s4=3'b100,s5=3'b101
;
    always @(posedge clk or posedge clr)
    begin
        if(clr==1) cs<=s0;
        else cs<=ns;
    end
    always @(cs or din)
    begin
        case(cs)
            s0:if(din==1) ns<=s1; else ns<=s0;
            s1:if(din==0) ns<=s2; else ns<=s1;
            s2:if(din==1) ns<=s3; else ns<=s0;
            s3:if(din==0) ns<=s4; else ns<=s1;
            s4:if(din==1) ns<=s5; else ns<=s0;
            s5:if(din==0) ns<=s0; else ns<=s1;
            default: ns<=s0;
        endcase
    end
    always @(cs)
    begin
        if(cs==s5)
```

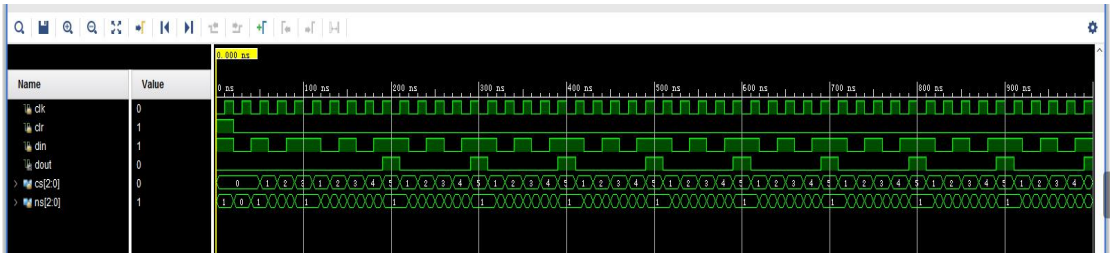
```
            dout<=1'b1;
        else
            dout<=1'b0;
    end
endmodule
```

(2)仿真文件

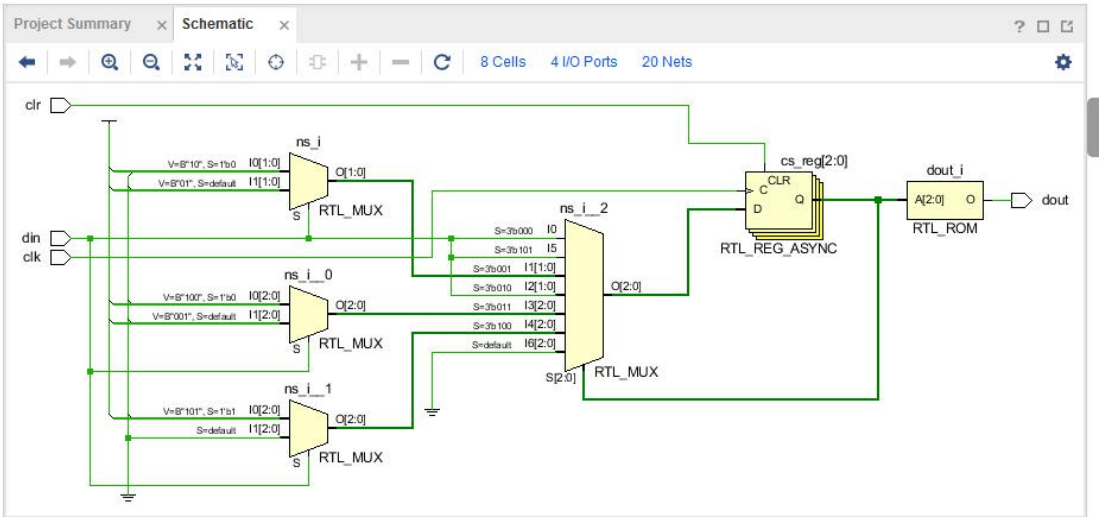
```
module moore_0420_tb(

    );
    reg clk,clr,din;
    wire dout;
    moore_0420 u0
(.clk(clk),.clr(clr),.din(din),.dout(dout));
    initial begin
        clk=0;#10;
        clr=1;#10;
        din=0;#10;
    end
    always begin
        clk=0;#10;
        clk=1;#10;
    end
    always begin
        clr=1;#20;
        clr=0;#10000;
    end
    always begin
        din=1;#20;
        din=0;#20;
        din=1;#20;
        din=0;#20;
        din=1;#20;
    end
endmodule
```

5. 实验结果记录



成功检测 10101 序列



实验小结

学习计算机逻辑基础知识后，在实验中强化所学内容，通过 vivado 编写 verilog 代码，加强了我的动手实践能力，加深对计算机逻辑基础的理解。