南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

计算机逻辑实验报告

班　　级 9191062301

学生姓名 孙傲歆

学　　号 919106840333

任课教师 余立功

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：

用 verilog 来实现 3-8 译码器

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。

3-8 译码器的真值表如下图：

输入			输出							
A3	A2	A1	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

4.简要操作步骤：

使用 Vivado 工具进行代码的编写及运行。首先选择芯片环境为 Aartix -7 系列（XC7A200T-FBG676-2）。先建立源代码文件进行源代码的编写，之后再编写相应的 tb 文件进行仿真代码的编写。最后点击仿真即可获得仿真波形图。

使用实验箱进行上板验证之前，先要选择合适的开关引脚建立约束文件 xdc，生成 bit 文件；正确连接实验箱，下载 bit 文件，观察实验结果。

5.实验代码：

1）源代码：

```
module decoder38(input [2:0] a,output [7:0] y);
reg [7:0] y;
always@(a)
begin
case(a)
0:y=8'b00000001;
1:y=8'b00000010;
```

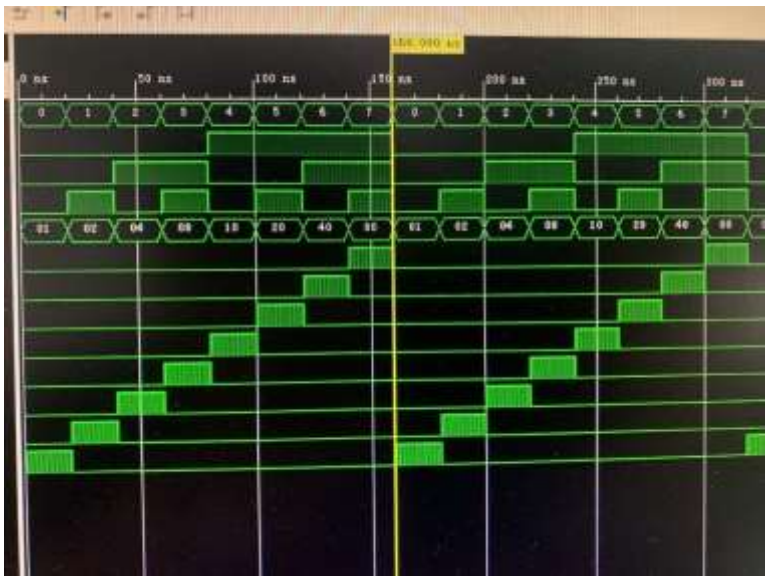
```
2:y=8'b00000100;
3:y=8'b00001000;
4:y=8'b00010000;
5:y=8'b00100000;
6:y=8'b01000000;
7:y=8'b10000000;
default:y=8'b00000000;
endcase
end
endmodule
```

2）仿真代码

```
module decoder38_tb();
reg [2:0] a;
wire [7:0] y;
decoder38 u0(a,y);
always begin
a=0; #20;
a=1; #20;
a=2; #20;
a=3; #20;
a=4; #20;
a=5; #20;
a=6; #20;
a=7; #20;
end
endmodule
```

6. 实验结果：

1）波形图：



2)实验箱结果:(拨码开关为:011,结果输出为 11101111)

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 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。

2 选 1 选择器的真值表如下所示：

输入	输出
S	Y
0	A
1	B

4. 简要操作步骤

使用 Vivado 工具进行代码的编写及运行。首先选择芯片环境为 Aartix -7 系列（XC7A200T-FBG676-2）。先建立源代码文件进行源代码的编写，之后再编写相应的 tb 文件进行仿真代码的编写。最后点击仿真即可获得仿真波形图。

5. 实验代码

1）源代码

```
module mux2 #(parameter WIDTH = 8)
      (a,b,s,y);
       input s;
       input [WIDTH-1:0] a,b;
       output[WIDTH-1:0] y;
       assign y= (s==0) ? a : b;
endmodule
```

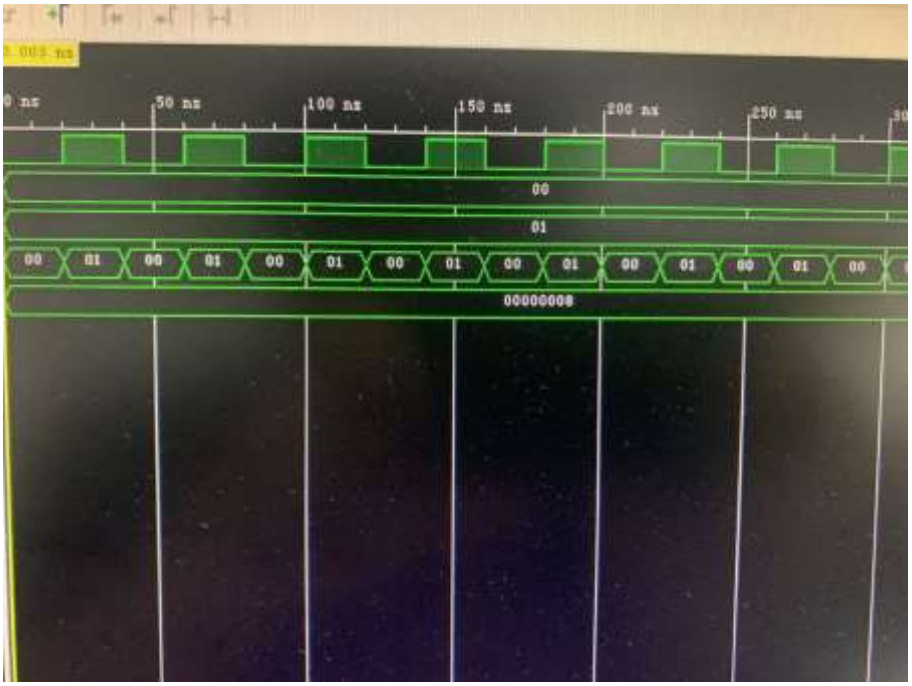
2）仿真代码

```
module mux2_tb #(parameter WIDTH = 8)();
reg s;
reg [WIDTH-1:0] a=8'b00000000;
reg [WIDTH-1:0] b=8'b00000001;
wire [WIDTH-1:0] y;
mux2 u1(a,b,s,y);
always begin
```

```
s=0; #20;
s=1; #20;
end
endmodule
```

6. 实验结果

波形图:



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：

用 verilog 生成一个十进制计数器，异步清零；

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。

下图为异步可逆十进制计数器:（本实验所制作十进制计数器仅有异步清零以及加计数功能）

CLK	CLR	LD	EN	功能
X	0	X	X	异步清零
↑	1	0	X	同步置数
↑	1	1	0	减计数
↑	1	1	1	加计数

4.简要操作步骤

使用 Vivado 工具进行代码的编写及运行。首先选择芯片环境为 Aartix -7 系列（XC7A200T-FBG676-2）。建立源代码文件进行源代码的编写。

使用实验箱进行上板验证之前，先要选择合适的开关引脚建立约束文件 xdc，生成 bit 文件；正确连接实验箱，下载 bit 文件，观察实验结果。

5.实验代码

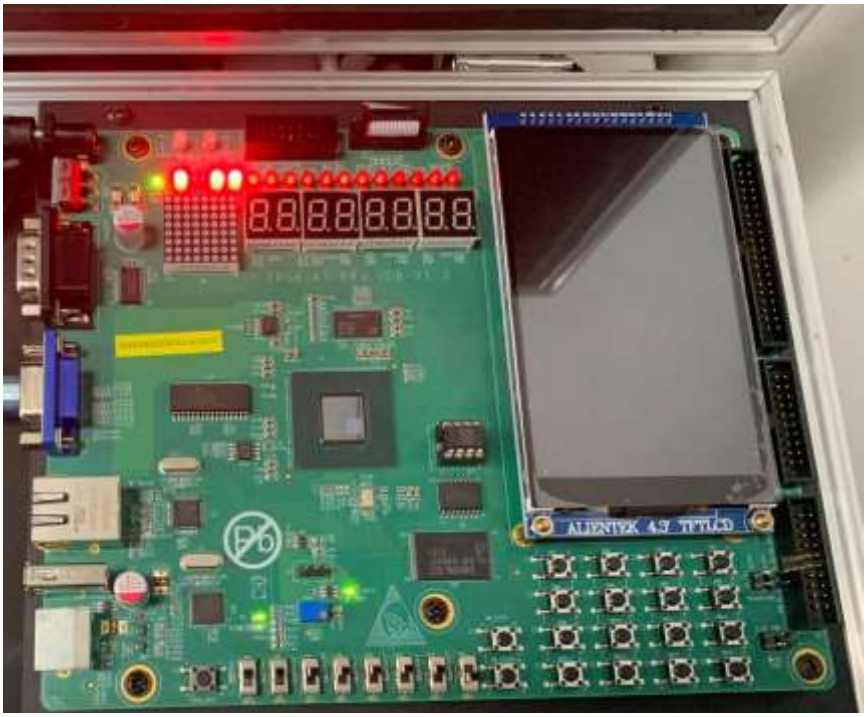
源代码：

```
module count10(input clkin, input clr,output [3:0]q);
reg[3:0]cnt;
always@(posedge clkin or posedge clr)
begin
    if(clr)
```

```
            cnt<=4'b0000;
        else
            if(cnt==4'h9)
                cnt<=4'h0;
            else
                cnt<=cnt+1;
end
assign q=cnt;
endmodule
```

6. 实验结果

实验箱结果：



实验四　状态机的设计及应用实验

1.实验目的：

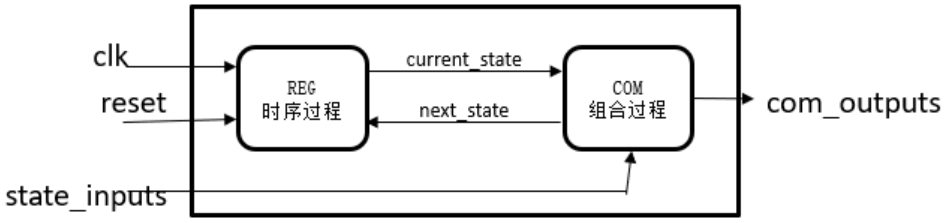
学习时序逻辑电路中 Mealy 和 Moore 状态机的设计方法及应用。

2.实验内容：

设计一个 Moore 型序列检测状态机，检测序列为 10101。

3.实验原理：

有限自动状态机 FSM（Finate State Machine)是复杂数字系统设计中非常重要的一部分，是实现高效率高可靠性逻辑控制的重要途径。大部分数字系统都是由控制单元和数据单元组成的。数据单元负责数据的处理和传输，而控制单元主要是控制数据单元的操作顺序。在数字系统中，控制单元往往通过使用有限状态机实现，有限状态机接受外部信号以及数据单元产生的状态信号，从而产生控制信号序列。有限状态机的一般结构如下图：



4.简要操作步骤

使用 Vivado 工具进行代码的编写及运行。首先选择芯片环境为 Aartix -7 系列（XC7A200T-FBG676-2)。建立源代码文件进行源代码的编写。

使用实验箱进行上板验证之前，先要选择合适的开关引脚建立约束文件 xdc，生成 bit 文件；正确连接实验箱，下载 bit 文件，观察实验结果。

5.实验代码

1）源代码

```
module moore3(input clk,input clr,input din,output dout);
      reg dout;
reg[3:0]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）仿真代码

```
module moore3_tb( );
        reg clk,clr,din;
        wire dout;
        moore3 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
```

6. 实验结果

波形图: