南京郵電大學
Nanjing University of Posts and Telecommunications

电工电子实验报告

课程名称：电工电子实验（二）

实验项目：存储器实验

学　　院：贝尔英才学院

班　　级：

学　　号：

姓　　名：

学　　期：2022-2023学年第1学期

存储器实验

一、实验目的

1.了解利用ISE CORE Generator进行设计的方法。
2.了解块ROM的读写操作方法。
3.下载并测试实现的逻辑功能。

二、实验仪器

ISE Design Suite 14.7

三、实验原理

74194功能表

功能	$\overline{CR}$	M_1	M_0	$CP\uparrow$	D_{SR}	D_{SL}	$D_0\ D_1\ D_2\ D_3$	$Q_0^{n+1}\ Q_1^{n+1}Q_2^{n+1}Q_3^{n+1}$
消除	0	ϕ	ϕ	ϕ	ϕ	ϕ	$\phi\ \ \phi\ \ \phi\ \ \phi$	0 0 0 0
并入	1	1	1	1	ϕ	ϕ	$d_0\ d_1\ d_2\ d_3$	$d_0\ d_1\ d_2\ d_3$
保持	1	ϕ	ϕ	0	ϕ	ϕ	$\phi\ \ \phi\ \ \phi\ \ \phi$	$Q_0^n\ \ Q_1^n\ Q_2^n\ Q_3^n$
	1	0	0	ϕ	ϕ	ϕ	$\phi\ \ \phi\ \ \phi\ \ \phi$	
右移	1	0	1	1	1	ϕ	$\phi\ \ \phi\ \ \phi\ \ \phi$	$1\ \ Q_0^n\ \ Q_1^n\ Q_2^n$
	1	0	1	1	0	ϕ	$\phi\ \ \phi\ \ \phi\ \ \phi$	$0\ \ Q_0^n\ \ Q_1^n\ Q_2^n$
左移	1	1	0	1	ϕ	1	$\phi\ \ \phi\ \ \phi\ \ \phi$	$Q_1^n\ Q_2^n\ Q_3^n\ 1$
	1	1	0	1	ϕ	0	$\phi\ \ \phi\ \ \phi\ \ \phi$	$Q_1^n\ Q_2^n\ Q_3^n\ 0$

四、实验内容

1.用Verilog HDL设计一个具有74LS161（同步4位二进制加法计数器）集成电路功能的时序逻辑电路，完成设计模块、Testbench、并保存仿真波形。Testbench能够测试到电路的所有功能、下载到FPGA、完成硬件调测并实现电路功能。

2.用Verilog HDL设计一个具有94194集成电路功能的时序逻辑电路（即4位双向移位寄存器，具有异步清零、同步置数、左移、右移和保持功能），完成设计模块、Testbench、并保存仿真波形。Testbench能够测试到电路的所有功能、下载到FPGA、完成硬件调测并实现电路功能。

3. 观看预习视频，完成教材186页 存储器实验，应用IP核调用的方式，完成设计代码及测试代码 ，记录仿真结果（截图），写出设计过程。

4.拓展要求: 如果要求用k1k2按键控制这四路序列码可以分别单独输出，完成设计代码及测试代码 。记录仿真结果

五、实验数据

1.用Verilog HDL设计一个具有74LS161（同步4位二进制加法计数器）集成电路功能的时序逻辑电路，完成设计模块、Testbench、并保存仿真波形。Testbench能够测试到电路的所有功能、下载到FPGA、完成硬件调测并实现电路功能。

（1）设计代码

```
module addcounter(clr,ld,p,t,clk,D,Q,co
    );
input clr,ld,p,t,clk;
input [3:0] D;
output reg [3:0]Q;
output reg  co;
   always@(posedge clk or negedge clr)
     begin
      if(!clr)
         Q<=4'b0000;
      else if(!ld)
            Q <=D;
           else
           casex({p,t})
           2'b11:
             begin
                          co <= Q[3]&Q[2]&Q[1]&Q[0];
                         Q <=Q+1'b1;

             end
           2'b01:begin
                     Q<=Q;
                     co <= Q[3]&Q[2]&Q[1]&Q[0];
                end
           2'bx0:begin
                     Q<=Q;
                     co <=0;
                end
            endcase
      end
endmodule
```

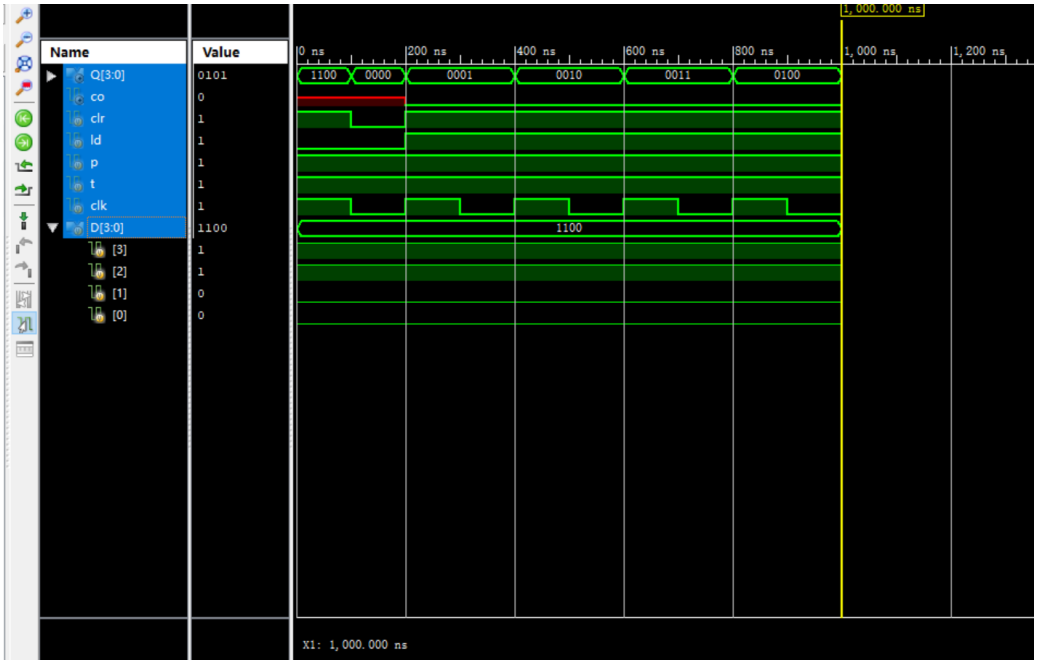
图（1-1）设计代码

（2）测试代码

```
module addcounter1;

   // Inputs
   reg clr;
   reg ld;
   reg p;
   reg t;
   reg clk;
   reg [3:0] D;

   // Outputs
   wire [3:0] Q;
   wire co;

   // Instantiate the Unit Under Test (UUT)
   addcounter uut (
      .clr(clr),
      .ld(ld),
      .p(p),
      .t(t),
      .clk(clk),
      .D(D),
      .Q(Q),
      .co(co)
   );
always #100 clk=~clk;
   initial begin
      // Initialize Inputs
      clr = 1;
      p = 1;
      t = 1;
      clk = 1;
      D = 4'b1100;

      // Wait 100 ns for global reset to finish
      ld=0;
      #100;//置数
      clr=0;
      #100;
         #100;
         //置零
         clr=1;ld=1;p=1;t=1;
         #3400;//计数
         ld=1;clr=1;p=0;t=1;
         #100;
         ld=1;clr=1;p=0;t=0;
         #100;
         // Add stimulus here10
    end

endmodule
```

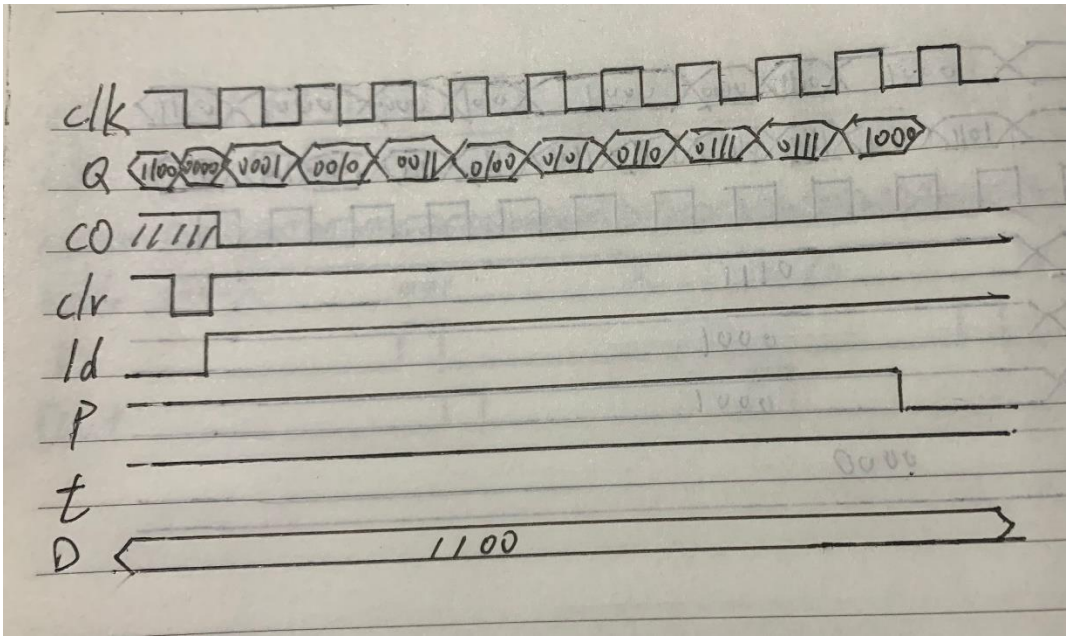
图（1-2）测试代码

（3）仿真结果

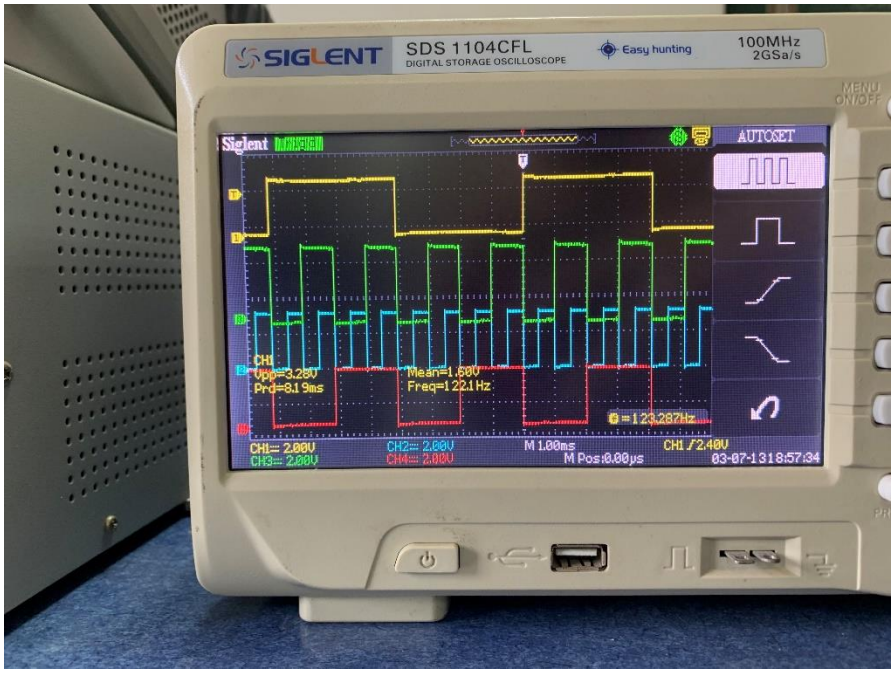


图（1-3）74161仿真截图

（4）手绘图



图（1-4）74161手绘波形图



图（1-5）74161计数功能示波器照片

2.用Verilog HDL设计一个具有74194集成电路功能的时序逻辑电路（即4位双向移位寄存器，具有异步清零、同步置数、左移、右移和保持功能），完成设计模块、Testbench、并保存仿真波形。Testbench能够测试到电路的所有功能、下载到FPGA、完成硬件调测并实现电路功能

（1）设计代码

```
////////////////////////////////////////////
module a74194(clr,M1,M0,clk,SR,SL,D,Q
    );
input clr,M1,M0,clk,SR,SL;
input [3:0] D;
output reg [3:0] Q;
always@(posedge clk or negedge clr)
begin
    if(!clr)
    Q<=4'b0000;
    else
    begin
       case({M1,M0})
       2'b00:Q<=Q;
       2'b01:Q<={SR,Q[3:1]};
       2'b10:Q<={Q[2:0],SL};
       2'b11:Q<=D;
       endcase
    end
end
endmodule

```

图（2-1）74194设计代码

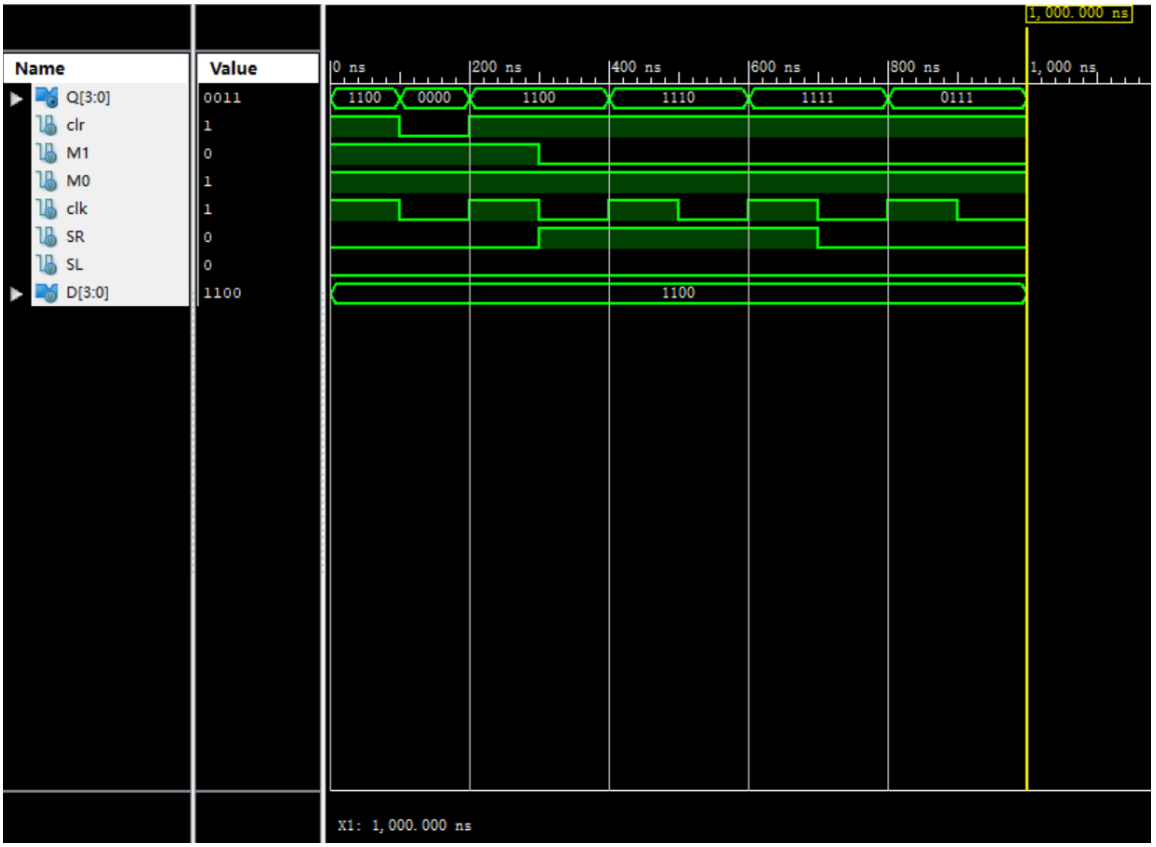
（2）测试代码

```
module a74194t;

    // Inputs
    reg clr;
    reg M1;
    reg M0;
    reg clk;
    reg SR;
    reg SL;
    reg [3:0] D;

    // Outputs
    wire [3:0] Q;

    // Instantiate the Unit Under Test (UUT)
    a74194 uut (
        .clr(clr),
        .M1(M1),
        .M0(M0),
        .clk(clk),
        .SR(SR),
        .SL(SL),
        .D(D),
        .Q(Q)
    );
always #100 clk=~clk;
    initial begin
        // Initialize Inputs
        clr = 1;
        M1 = 1;
        M0 = 1;
        clk = 1;
        SR = 0;
        SL = 0;
        D = 4'b1100;

        // Wait 100 ns for global reset to finish
        #100;
        clr=0;
#100;
```

```
clr=1;
#100;
M1=0;M0=1;SR=1;#400;
M1=0;M0=1;SR=0;#400;
M1=1;M0=0;SL=1;#400;
M1=1;M0=0;SL=0;#400;
M1=0;M0=0;
        // Add stimulus here

    end

endmodule
```

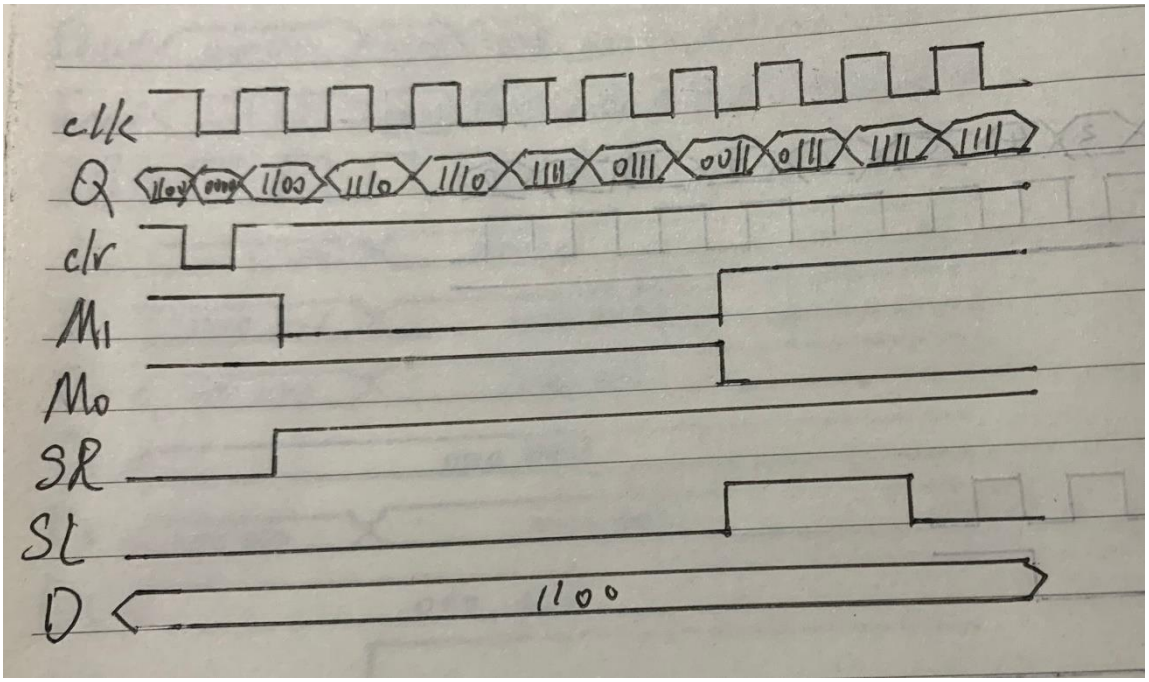
图（2-2）74194测试代码

（3）仿真结果



图（2-3）74194仿真截图

（4）手绘图



图（2-4）74194波形手绘图

（5）实验照片

右移功能


南京邮电大学电工电子实验教学中心
HTTP://EELAB.NJUPT.EDU.CN
逻辑电平显示
L1 L2 L3 L4 L5 L6 L7 L8
GND
+5V


南京邮电大学电工电子实验教学中心
HTTP://EELAB.NJUPT.EDU.CN
逻辑电平显示
L1 L2 L3 L4 L5 L6 L7 L8
GND
+5V


南京邮电大学电工电子实验教学中心
HTTP://EELAB.NJUPT.EDU.CN
逻辑电平显示
L1 L2 L3 L4 L5 L6 L7 L8
GND
+5V


南京邮电大学电工电子实验教学中心
HTTP://EELAB.NJUPT.EDU.CN
逻辑电平显示
L1 L2 L3 L4 L5 L6 L7 L8
GND
+5V

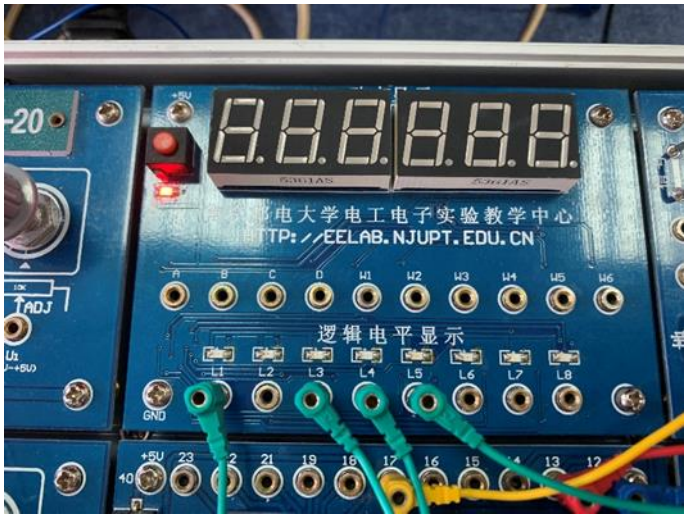
左移功能


南京邮电大学电工电子实验教学中心
HTTP://EELAB.NJUPT.EDU.CN
逻辑电平显示
L1 L2 L3 L4 L5 L6 L7 L8
GND
+5V


电大学电工电子实验教学中心
TTP://EELAB.NJUPT.EDU.CN
逻辑电平显示


电大学电工电子实验教学中心
TTP://EELAB.NJUPT.EDU.CN
逻辑电平显示


电大学电工电子实验教学中心
TTP://EELAB.NJUPT.EDU.CN
逻辑电平显示



3．观看预习视频，完成教材186页 存储器实验，应用IP核调用的方式，完成设计代码及测试代码 ，记录仿真结果，写出设计过程。

（1）设计代码

```
module rom(
    clk,ce,thresh0,q,w);
    input clk;
    input ce;
    output thresh0;
    output [2:0] q;
    output [3:0] w;
    aa aa(
    .clk(clk),
    .ce(ce),
    .thresh0(thresh0),
    .q(q)
    );
    bb bb(
    .clka(clk),
    .addra(q),
    .douta(w)
    );

endmodule
```

图（3-1）设计代码

（2）测试代码

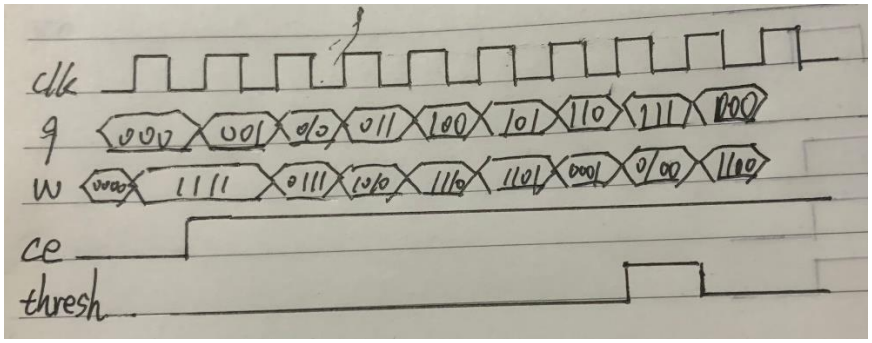
```
module rom_tb;

    // Inputs
    reg clk;
    reg ce;

    // Outputs
    wire thresh0;
    wire [2:0] q;
    wire [3:0] w;

    // Instantiate the Unit Under Test (UUT)
    rom uut (
        .clk(clk),
        .ce(ce),
        .thresh0(thresh0),
        .q(q),
        .w(w)
    );

    initial begin
        // Initialize Inputs
        clk = 0;
        ce = 0;      // Add stimulus here
       #300 ce=1;
    end
always #100 clk=~clk;

endmodule
```

图（3-2）测试代码

（3）仿真结果

图（3-3）仿真截图

（4）手绘图



图（3-4）手绘波形图

4.拓展要求：如果要求用k1k2按键控制这四路序列码可以分别单独输出，完成设计代码及测试代码 。记录仿真结果（截图）

（1）设计代码

```
module romk(
input [1:0] k,
input clk,
input en,
output reg d,
output reg [3:0] q

    );
 reg [2:0] count_8;
always@(posedge clk or negedge en)
begin
if(!en) count_8<=3'b000;
else
count_8<=count_8+1;
end
always@(count_8)
begin
case(count_8)
3'b000:q<=4'b1111;
3'b001:q<=4'b0111;
3'b010:q<=4'b1010;
3'b011:q<=4'b1110;
3'b100:q<=4'b1101;
3'b101:q<=4'b0001;
3'b110:q<=4'b0100;
default:q<=4'b1100;
endcase
end
always@(k or q)
begin
case(k)
2'b00: d<=q[3];
2'b01: d<=q[2];
2'b10: d<=q[1];
default: d<=q[0];
endcase
end

endmodule
```

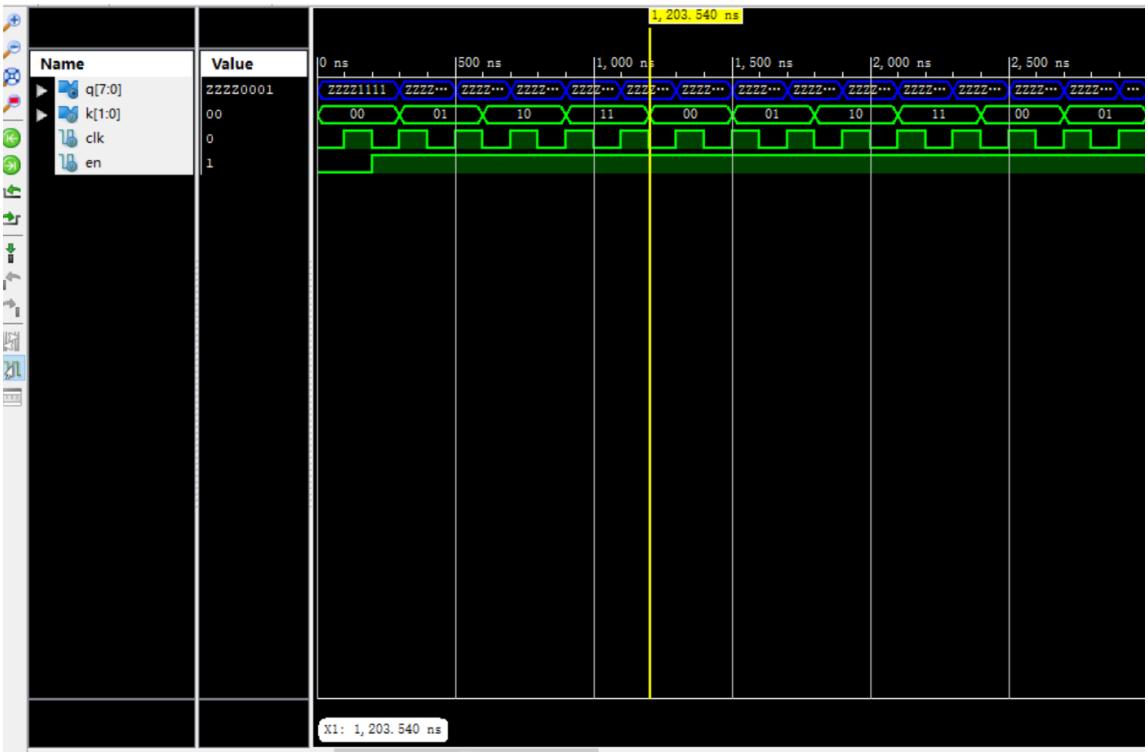
图（4-1）设计代码

（2）测试代码

```
module romk_tb;

   // Inputs
   reg [1:0] k;
   reg clk;
   reg en;

   // Outputs
   wire [7:0] q;

   // Instantiate the Unit Under Test (UUT)
   romk uut (
      .k(k),
      .clk(clk),
      .en(en),
      .q(q)
   );

initial begin
clk=0;
forever #100 clk=~clk;
end

initial begin
en=0;
#200 en=1;
end

initial begin
     k=2'b00;
#300 k=2'b01;
#300 k=2'b10;
#300 k=2'b11;
#300 k=2'b00;
#300 k=2'b01;
#300 k=2'b10;
#300 k=2'b11;
#300 k=2'b00;
#300 k=2'b01;
#300 k=2'b10;
#300 k=2'b11;
#300 k=2'b00;
#300 k=2'b01;
#300 k=2'b10;
#300 k=2'b11;
end
   /*initial begin
      // Initialize Inputs
      k = 0;
      clk = 0;
      en = 0;

      // Wait 100 ns for global reset to finish
      #100;

      // Add stimulus here

   end
     */
endmodule
```

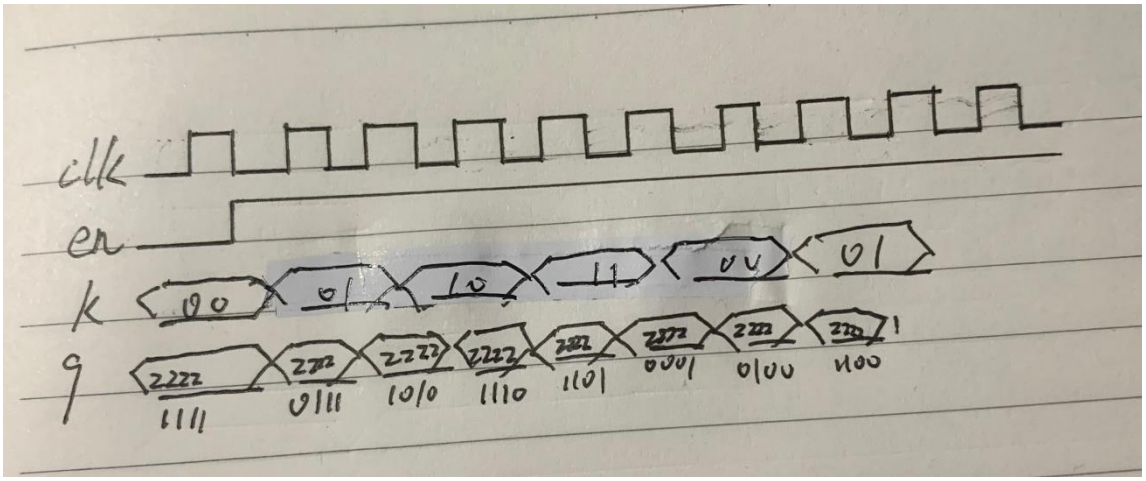
图（4-2）测试代码

（3）仿真结果



图（4-3）仿真截图

（4）手绘图



图（4-4）手绘波形图

六、实验小结

通过本次实验我学会了用IP核设计以及调用模块，。了解块ROM的读写操作方法。进行了74161以及74194的设计，对Verilog语言有了更深的理解。