# 数电实验9报告

17341015　数据科学与计算机学院
计科一班　陈鸿峥

## 1　内容一

### 1.1　实验目的

设计16进制异步加法计数器

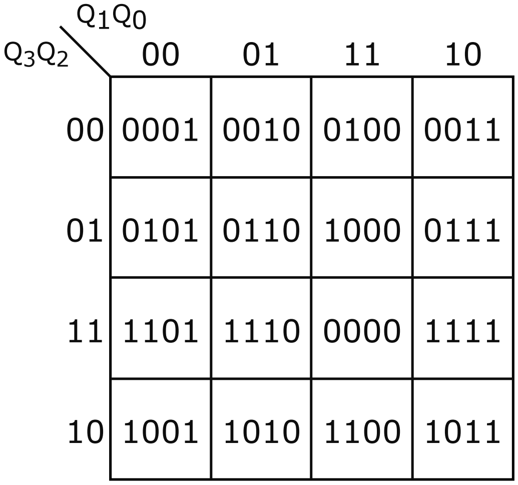
### 1.2　实验原理

1. 真值表如下

| Q3 | Q2 | Q1 | Q0 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |

| Q3 | Q2 | Q1 | Q0 |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |

2. 次态卡诺图



| $Q_3Q_2$ \ $Q_1Q_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0001 | 0010 | 0100 | 0011 |
| 01 | 0101 | 0110 | 1000 | 0111 |
| 11 | 1101 | 1110 | 0000 | 1111 |
| 10 | 1001 | 1010 | 1100 | 1011 |

3. 状态方程

$$
\begin{aligned}
Q_3^{n+1} &= Q_3\,\overline{Q_2} + Q_3\,\overline{Q_1} + Q_3\,\overline{Q_0} + \overline{Q_3}\,Q_2Q_1Q_0 \\
Q_2^{n+1} &= Q_2\,\overline{Q_1} + Q_2\,\overline{Q_0} + \overline{Q_2}\,Q_1Q_0 \\
Q_1^{n+1} &= \overline{Q_1}\,Q_0 + Q_1\,\overline{Q_0} \\
Q_0^{n+1} &= \overline{Q_0}
\end{aligned}
$$

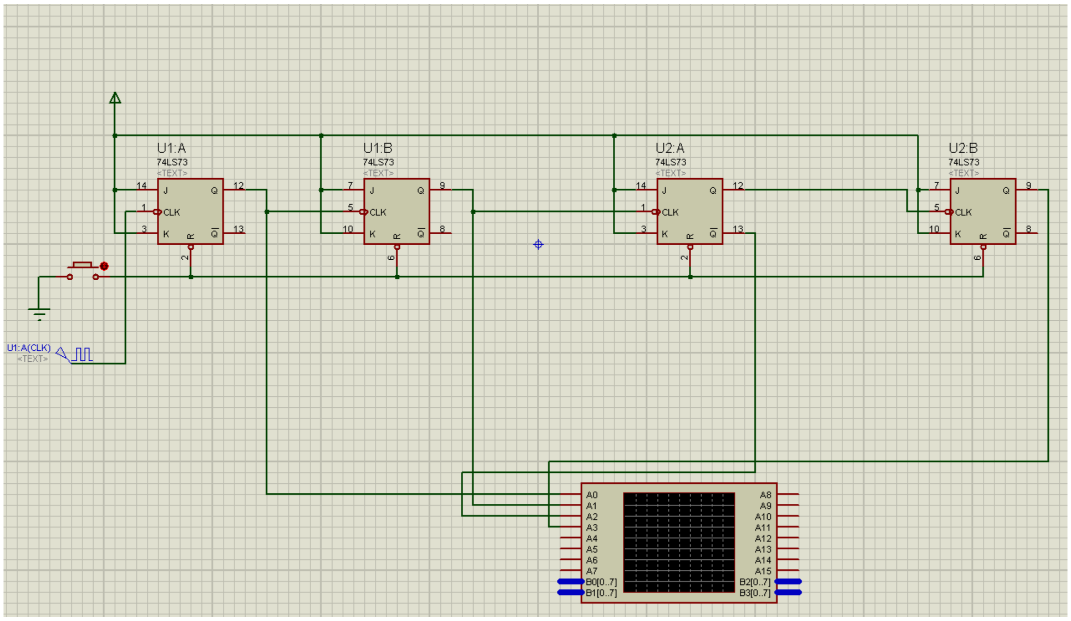
4. 驱动方程，将状态方程化为JK触发器形式可得

$$
\begin{aligned}
J_0 &= 1 & K_0 &= 1 \\
J_1 &= Q_0 & K_0 &= Q_0 \\
J_2 &= Q_1Q_0 & K_2 &= Q_1Q_0 \\
J_3 &= Q_2Q_1Q_0 & K_3 &= Q_2Q_1Q_0
\end{aligned}
$$

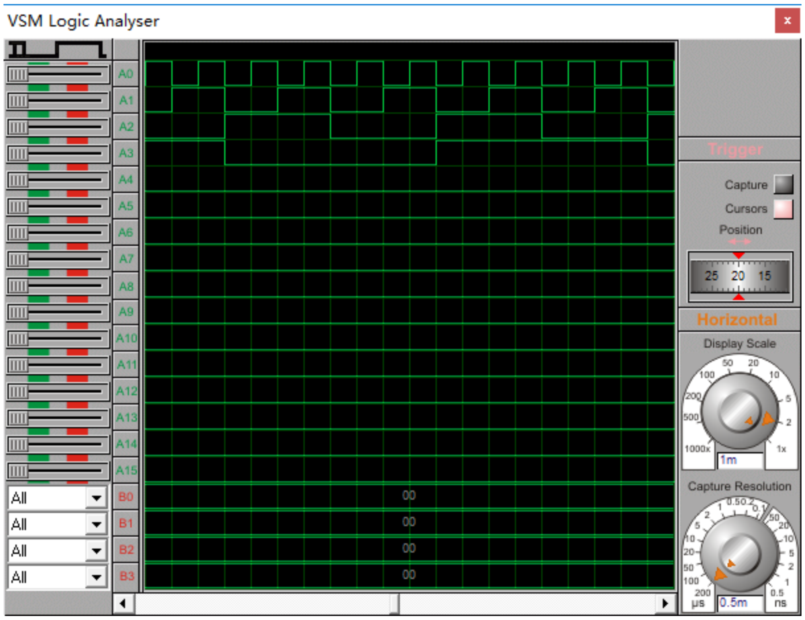
## 1.3 实验细节

### 1.3.1 Protues仿真

电路图如下所示



仿真结果如下



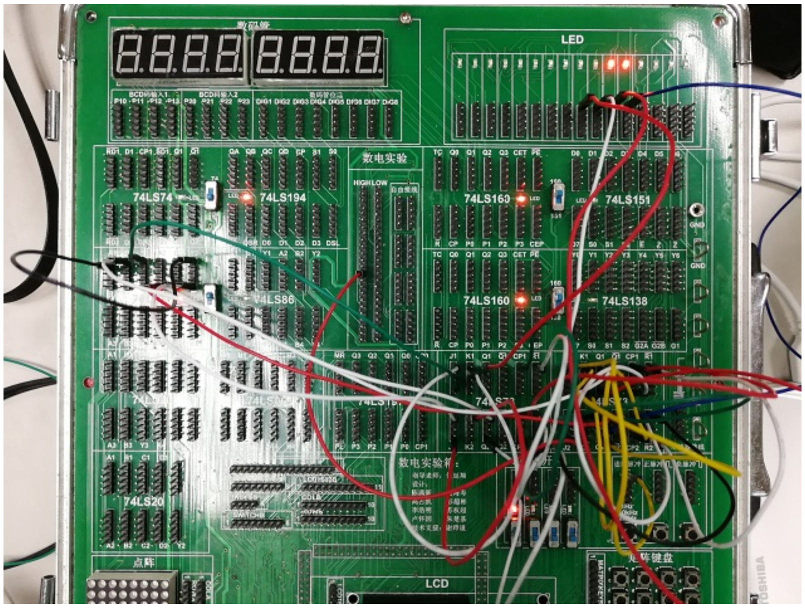
### 1.3.2　实验仪器及器件

1. 数字电路实验箱、示波器、导线若干

2. 74LS73*4

### 1.3.3　实验流程与结果分析

如Protues电路连线
实验箱连线如下



通过观察01显示器，符合波形

# 2　内容二

## 2.1　实验目的

实现16进制同步加法计数器

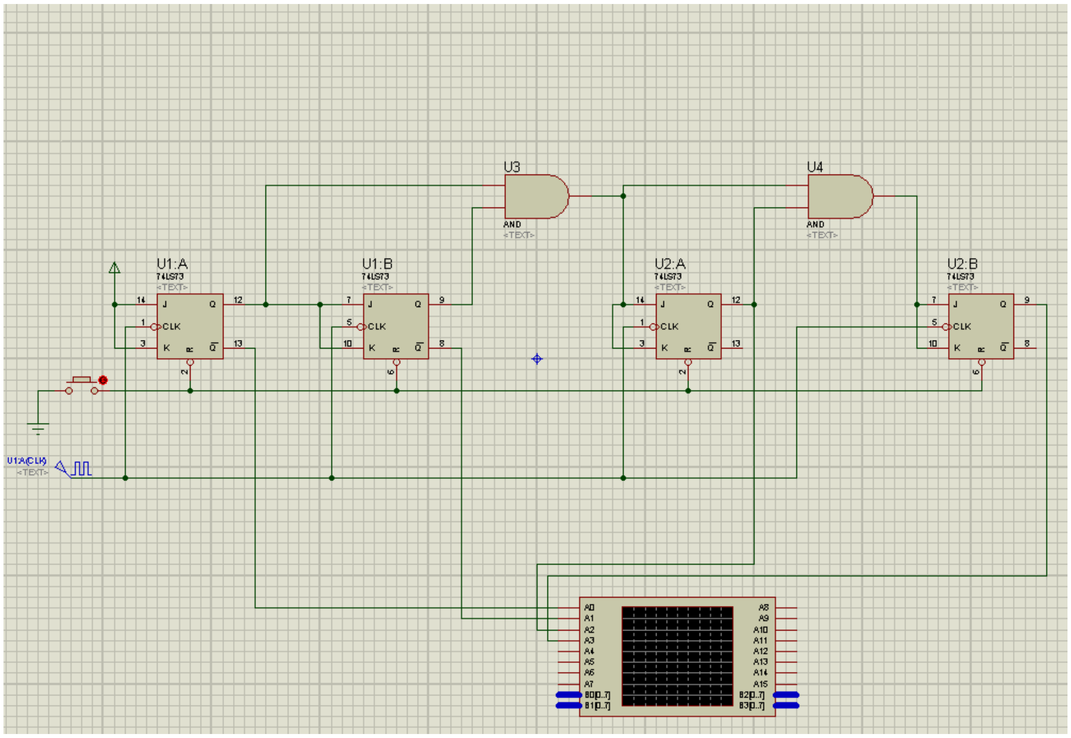
## 2.2　实验原理

同实验1

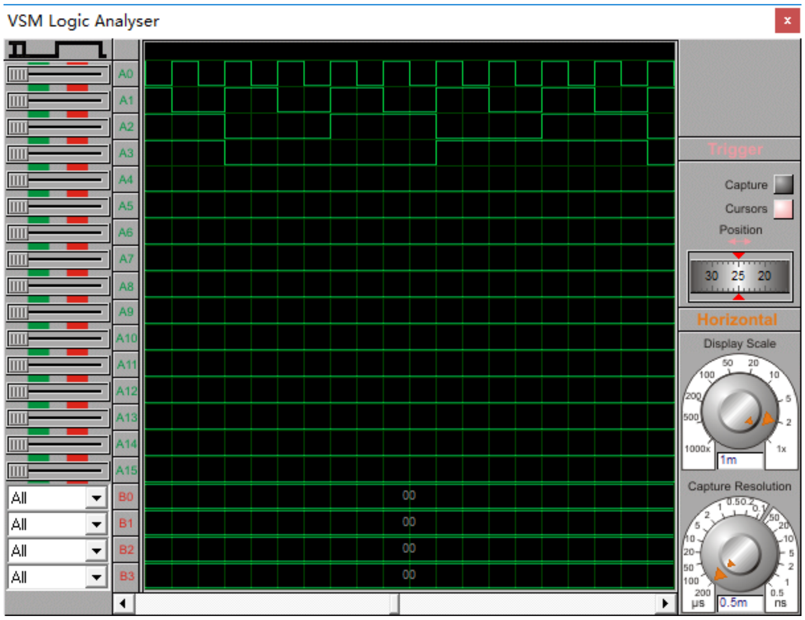
## 2.3　实验细节

### 2.3.1　Protues仿真

电路图如下所示



仿真结果如下



Display Scale 1m

Capture Resolution 0.5m

### 2.3.2 实验仪器及器件

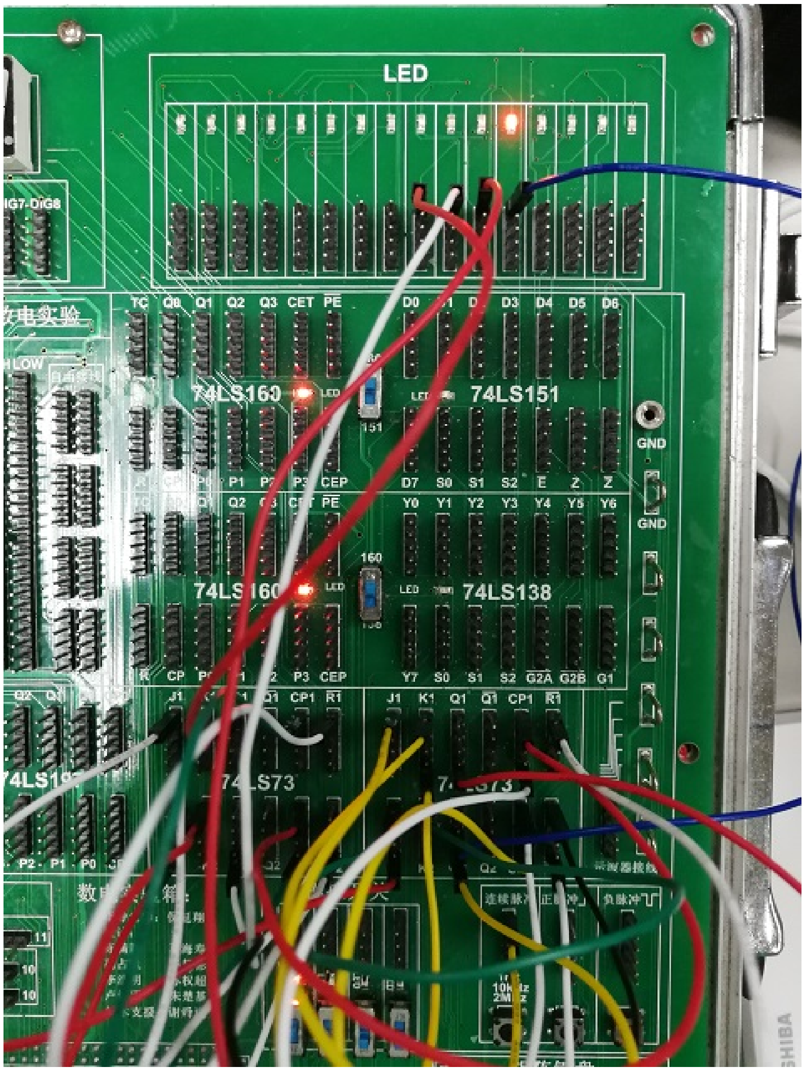
1. 数字电路实验箱、示波器、导线若干

2. 74LS73*4

### 2.3.3 实验流程及结果分析

如Protues电路连线

实验箱连线如下



通过观察01显示器，符合波形

# 3 内容三

## 3.1 实验目的

使用JK触发器和门电路设计实现一个二进制四位计数器模仿74LS194功能

## 3.2 实验原理

74LS194功能如下

| $\overline{Cr}$ | $S_1$ | $S_0$ | 功能 |
|---|---|---|---|
| 0 | X | X | 置零 |
| 1 | 0 | 0 | 保持 |
| 1 | 0 | 1 | 右移 |
| 1 | 1 | 0 | 左移 |
| 1 | 1 | 1 | 并行送数 |

仅左移右移次态卡诺图

| $Q_3Q_2$ \ $Q_1Q_0D$ | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | 00010 | 10001 | X | 00110 | 01110 | X | X | X |
| 01 | X | X | X | X | 11110 | X | X | X |
| 11 | X | 11101 | X | X | X | X | 11111 | X |
| 10 | X | 11001 | X | X | X | X | X | X |

状态方程

$$Q_3^{n+1} = Q_2 + D$$
$$Q_2^{n+1} = Q_3 + Q_1$$
$$Q_1^{n+1} = Q_2 + Q_1 + Q_0$$
$$Q_0^{n+1} = Q_1 + \overline{D}$$

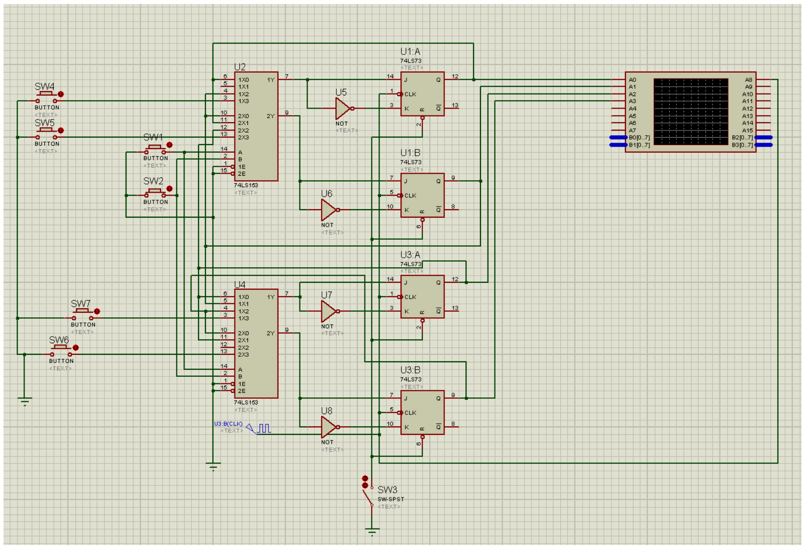
驱动方程，将状态方程化为JK触发器形式可得

$$J_0 = \overline{\overline{Q_1}\,D} \qquad K_0 = \overline{Q_1}\,D$$
$$J_1 = \overline{\overline{Q_2}\,\overline{Q_0}} \qquad K_0 = 0$$
$$J_2 = \overline{\overline{Q_3}\,\overline{Q_1}} \qquad K_2 = \overline{Q_3}\,\overline{Q_1}$$
$$J_3 = \overline{\overline{Q_2}\,\overline{D}} \qquad K_3 = \overline{Q_2}\,\overline{D}$$

## 3.3 实验细节

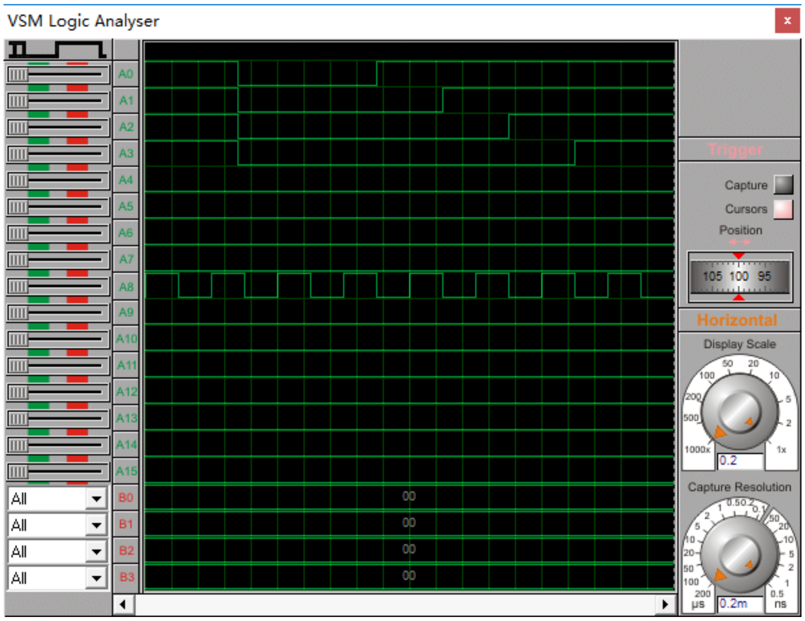
### 3.3.1 Protues仿真

全功能74LS194电路图如下

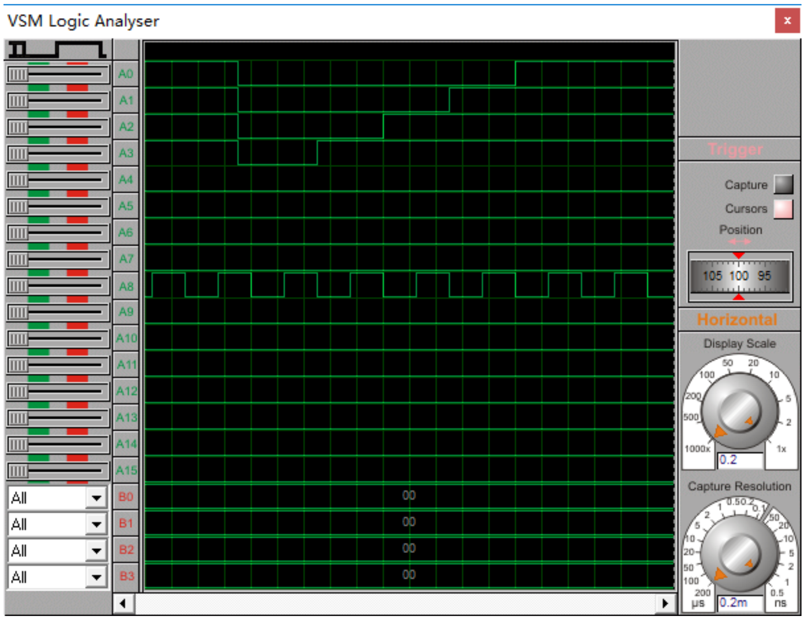


具体功能测试如下

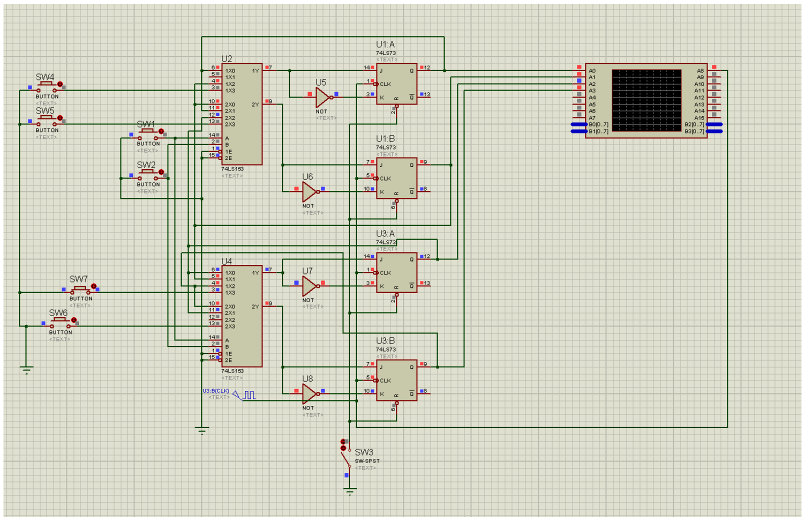
1. 左移



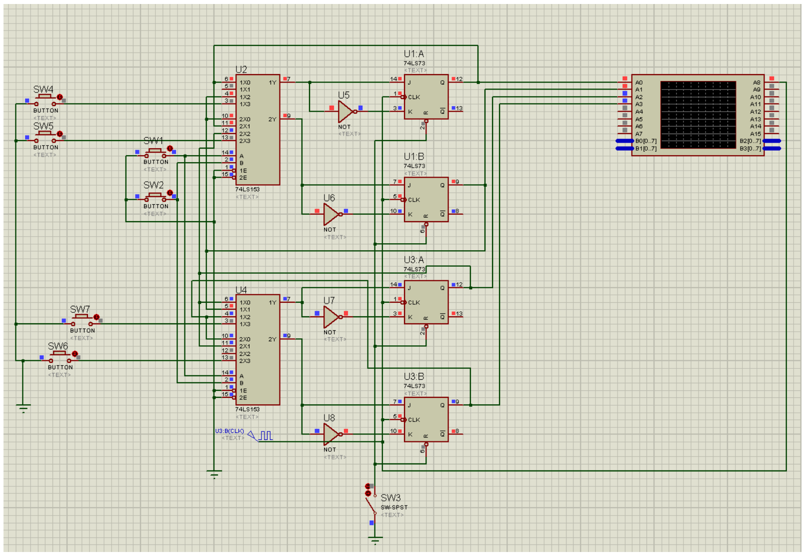
2. 右移



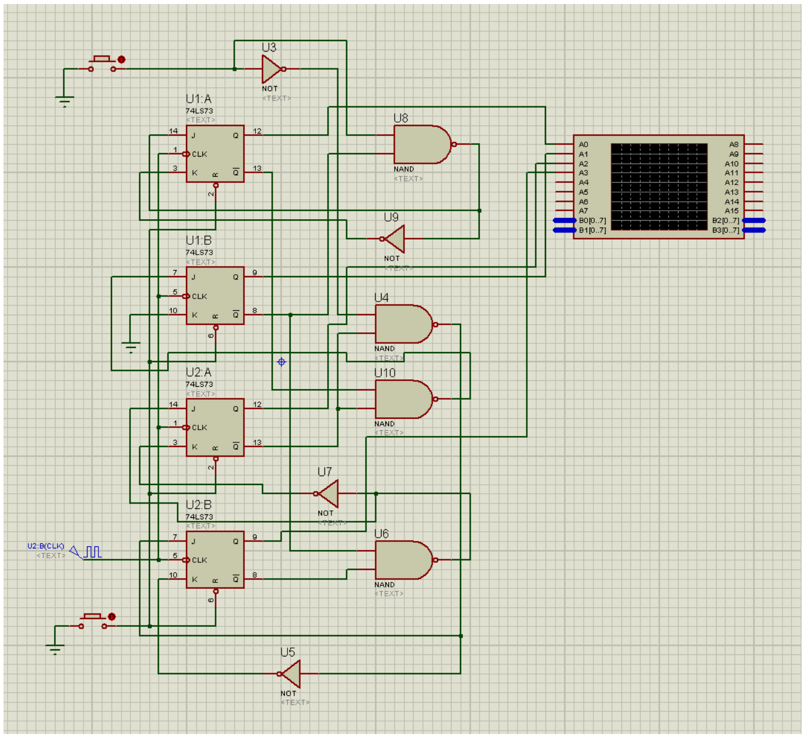
3. 并行送数，图为1011



4. 保持，图为左移过程中出现0011后立即切换状态为保持



仅左移右移74LS194电路图如下



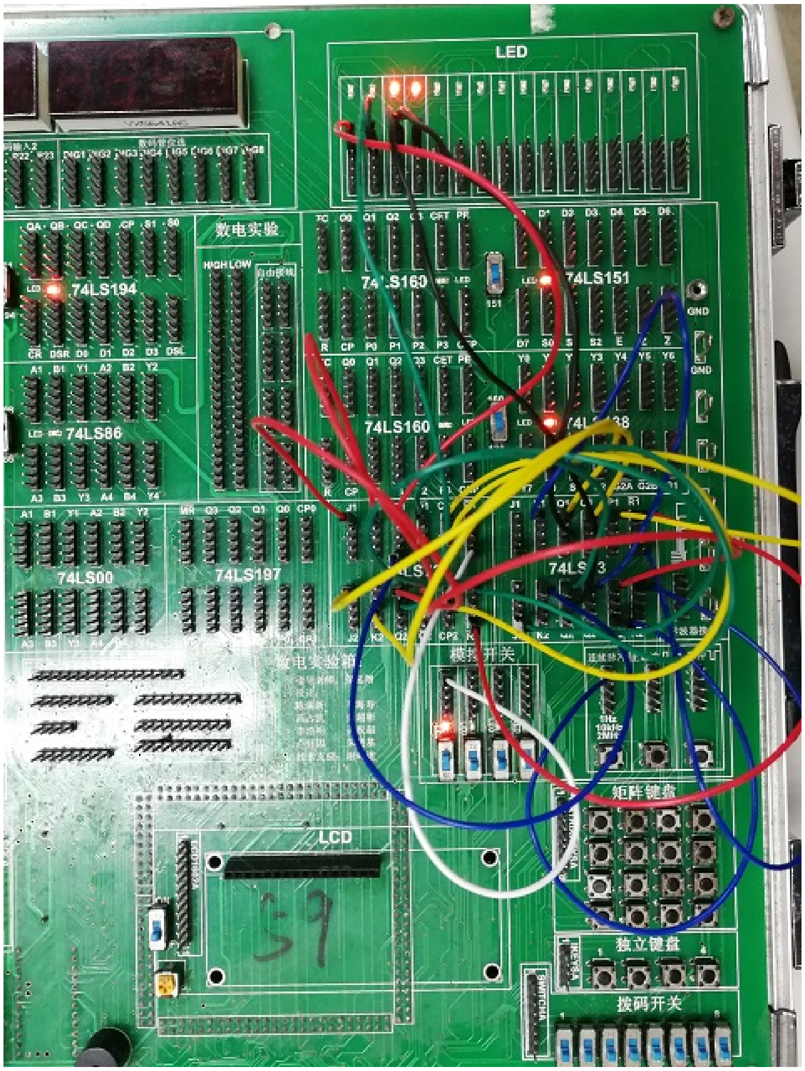
### 3.3.2 实验仪器及器件

1. 数字电路实验箱、示波器、导线若干

2. 74LS73*4

### 3.3.3 实验流程与结果分析

连线如Protues所示，结果如下图



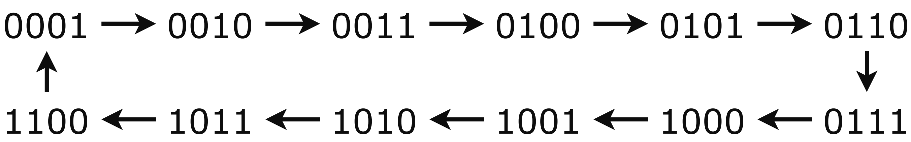
# 4 内容四

## 4.1 实验目的

用JK触发器实现12进制同步计数器

## 4.2 实验原理

1. 状态转换图



2. 确定电路所需触发器数目
   由于$2^4 = 16 > 12$，故需要4个JK触发器

3. 次态卡诺图

| $Q_1Q_0$<br>$Q_3Q_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 0010 | 0100 | 0011 |
| 01 | 0101 | 0110 | 1000 | 0111 |
| 11 | 0001 | X | X | X |
| 10 | 1001 | 1010 | 1100 | 1011 |

4. 触发器状态方程，由卡诺图可得

$$
\begin{aligned}
Q_0^{n+1} &= \overline{Q_0} \\
Q_1^{n+1} &= Q_0\overline{Q_1} + \overline{Q_0}Q_1 \\
Q_2^{n+1} &= Q_0Q_1\,\overline{Q_2} + \overline{Q_1}\,Q_2\,\overline{Q_3} + \overline{Q_0}\,Q_2\,\overline{Q_3} \\
Q_3^{n+1} &= \overline{Q_2}\,Q_3 + Q_0Q_1Q_2\,\overline{Q_3}
\end{aligned}
$$

5. 触发器驱动方程，由

$$Q^{n+1} = J\,\overline{Q^n} + \overline{K}\,Q^n$$

将状态方程整理为上式形式，可得

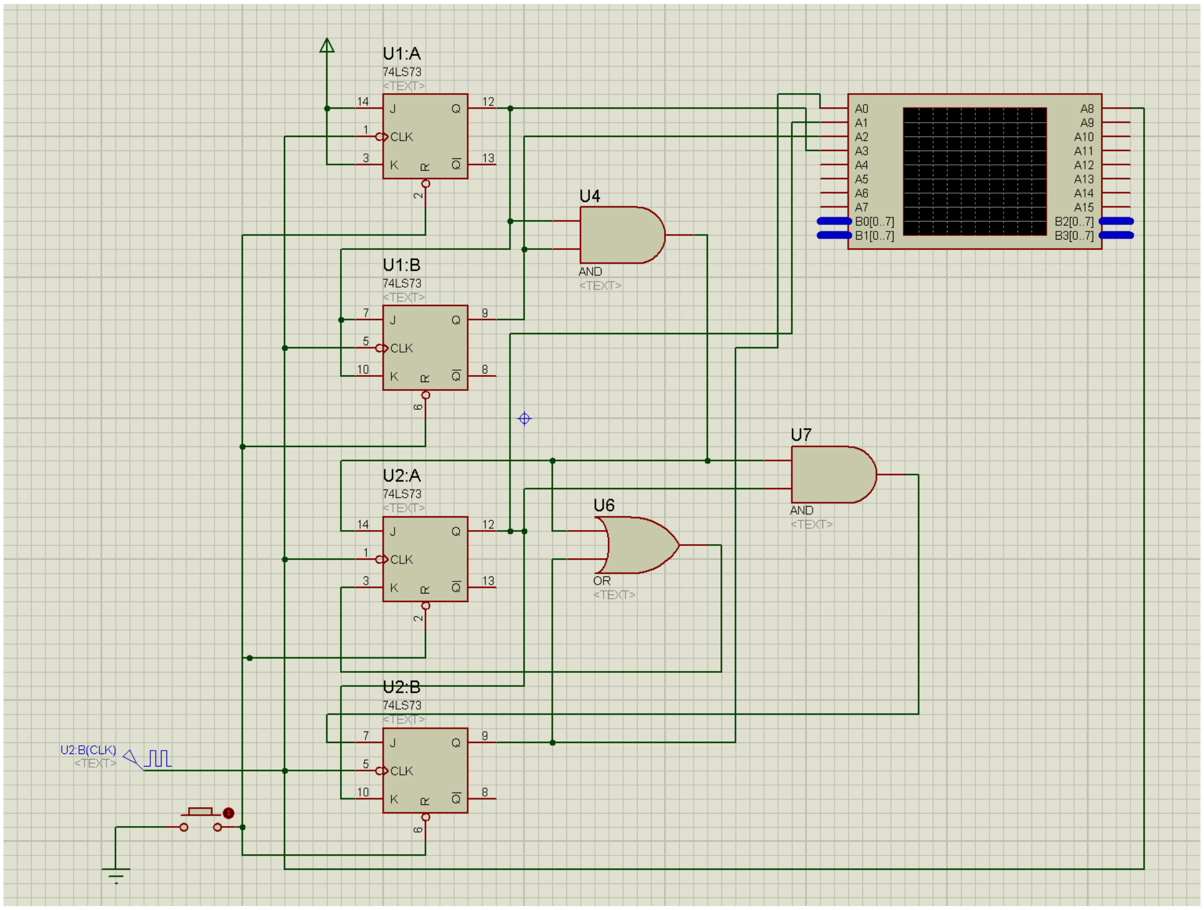
$$
\begin{aligned}
J_0 &= 1 & K_0 &= 1 \\
J_1 &= Q_0 & K_0 &= Q_0 \\
J_2 &= Q_1Q_0 & K_2 &= \overline{\overline{Q_3}\,\overline{Q_1} + \overline{Q_3}\,\overline{Q_0}} = \overline{Q_3} + Q_1Q_0 \\
J_3 &= Q_2Q_1Q_0 & K_3 &= Q_2
\end{aligned}
$$

6. 检查自启动

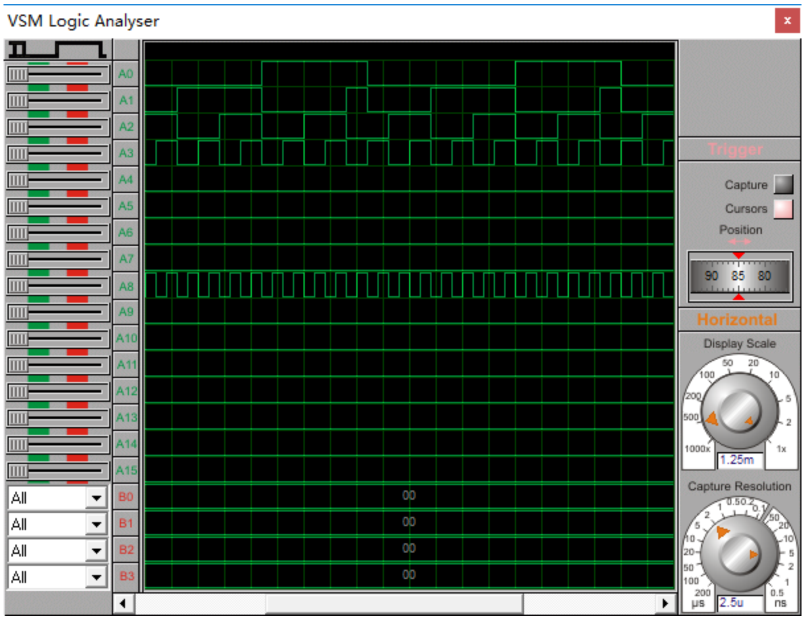
当输入为1111和0000时，可自动跳转至0001；输入为1101时，跳转至0010；输入为1110时，跳转至0011

### 4.3 Protues仿真

电路图连接如下



仿真结果如下



波形符合12进制转换

## 4.4 实验细节

### 4.4.1 实验仪器及器件

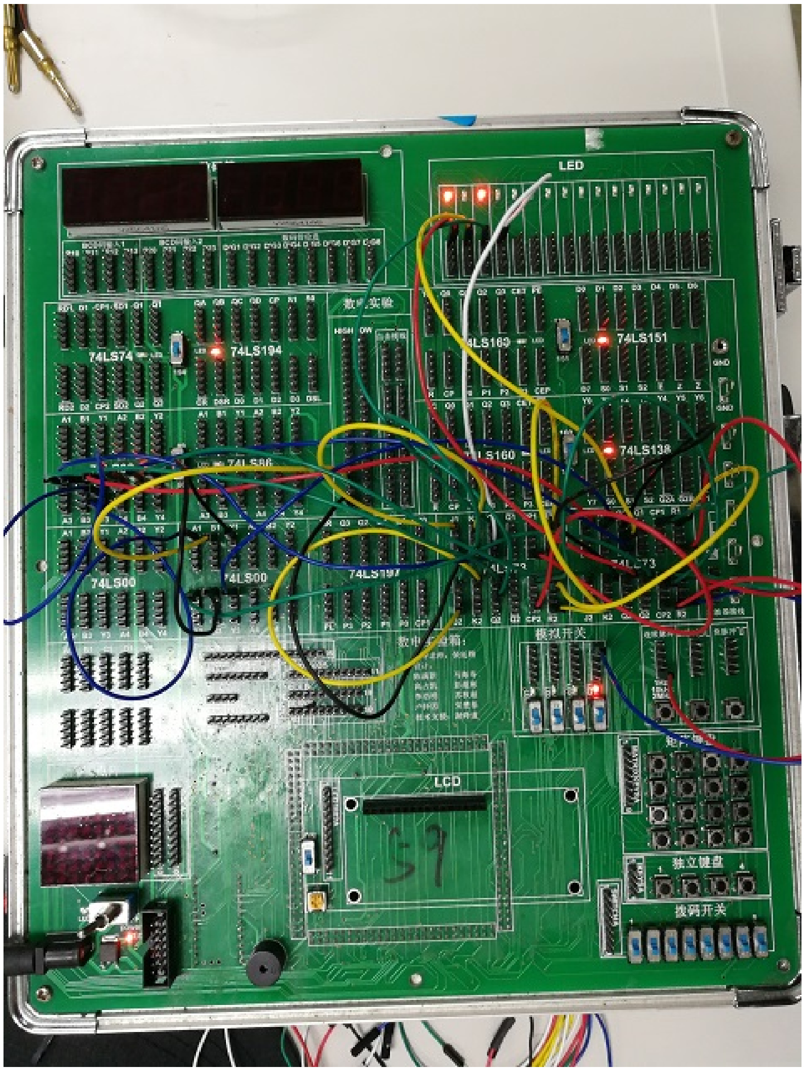
1. 数字电路实验箱、示波器、导线若干

2. 74LS73*4，74LS00*2，74LS08*2

### 4.4.2 实验流程与结果分析

如Protues电路图所示连线

将输出接01显示器，可见01显示器按照状态转换图顺序依次闪烁



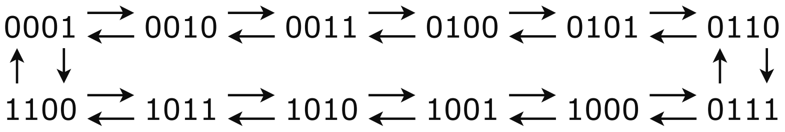
# 5 内容五

## 5.1 实验目的

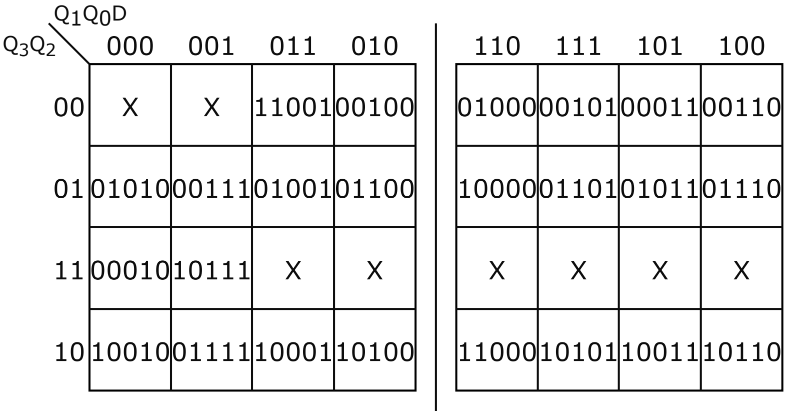
用Protues和Vivado分别实现一个有控制变量D的12进制计数器，并在七段数码管上显示计数结果

## 5.2 实验原理

1. 状态转移图



2. 次态卡诺图



| $Q_3Q_2$ \ $Q_1Q_0D$ | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | X | X | 11001 | 00100 | 01000 | 00101 | 00011 | 00110 |
| 01 | 01010 | 00111 | 01001 | 01100 | 10000 | 01101 | 01011 | 01110 |
| 11 | 00010 | 10111 | X | X | X | X | X | X |
| 10 | 10010 | 01111 | 10001 | 10100 | 11000 | 10101 | 10011 | 10110 |

3. 状态方程

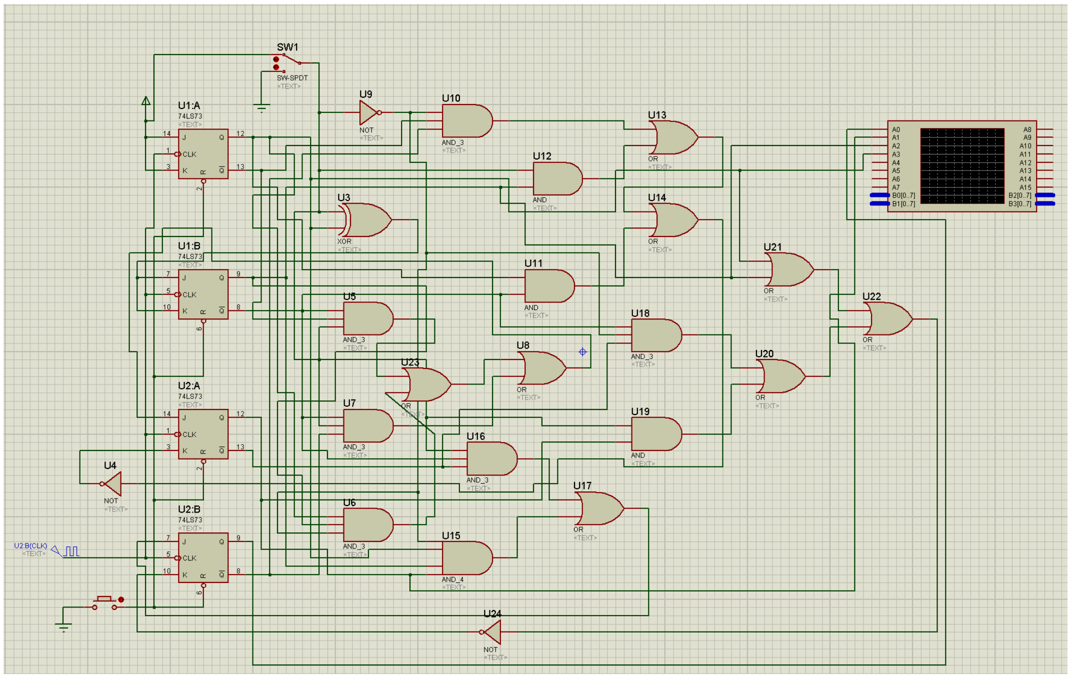
$$
\begin{aligned}
Q_0^{n+1} &= \overline{Q_0} \\
Q_1^{n+1} &= \overline{Q_1}\,\overline{Q_0}\,D + \overline{Q_1}\,Q_0\,\overline{D} + Q_1Q_0D + Q_1\,\overline{Q_0}\,\overline{D} = (Q_0 \oplus D)\,\overline{Q_1} + \overline{Q_0 \oplus D}\,Q_1 \\
Q_2^{n+1} &= \overline{Q_3}\,Q_2\,\overline{Q_0}\,\overline{D} + \overline{Q_2}\,\overline{Q_1}\,\overline{Q_0}\,D + Q_2\,\overline{Q_1}\,Q_0 + \overline{Q_3}\,\overline{Q_2}\,\overline{Q_1}\,D + Q_2Q_1D + \overline{Q_2}\,Q_1Q_0\,\overline{D} \\
&= (\overline{Q_1}\,\overline{Q_0}\,D + Q_1Q_0\,\overline{D} + \overline{Q_3}\,\overline{Q_1}\,D)\,\overline{Q_2} + (\overline{Q_3}\,\overline{Q_0}\,\overline{D} + \overline{Q_1}\,Q_0 + Q_1D)Q_2 \\
Q_3^{n+1} &= Q_3Q_1 + \overline{Q_3}\,Q_2Q_1Q_0\,\overline{D} + Q_3Q_0 + Q_3\,\overline{Q_2}\,\overline{Q_1}\,\overline{D} + Q_3Q_2D + \overline{Q_3}\,\overline{Q_2}\,\overline{Q_1}\,D \\
&= (Q_2Q_1Q_0\,\overline{D} + \overline{Q_2}\,\overline{Q_1}\,D)\,\overline{Q_3} + (Q_1 + Q_0 + \overline{Q_2}\,\overline{Q_1}\,\overline{D} + Q_2D)Q_3
\end{aligned}
$$

4. 驱动方程

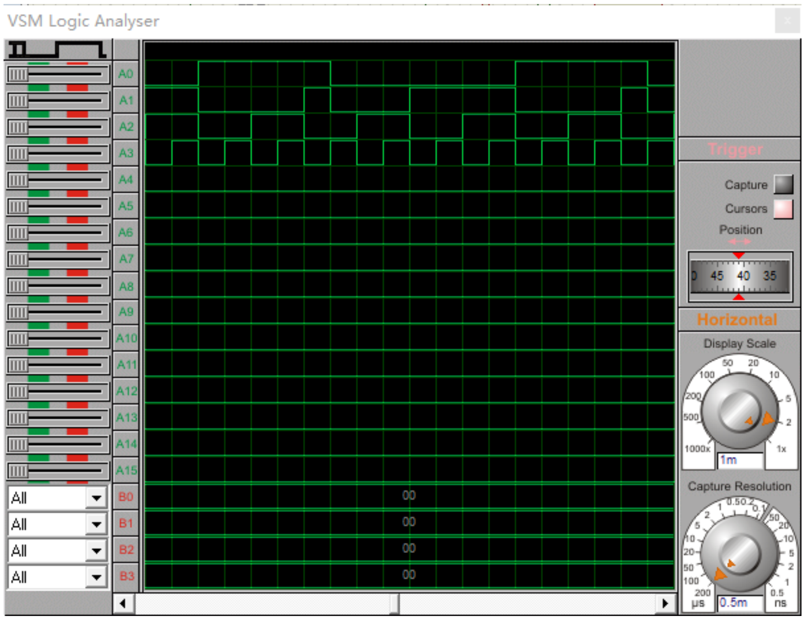
$$
\begin{aligned}
J_0 &= 1 & K_0 &= 1 \\
J_1 &= Q_0 \oplus D & K_1 &= Q_0 \oplus D \\
J_2 &= \overline{Q_1}\,\overline{Q_0}\,D + Q_1Q_0\,\overline{D} + \overline{Q_3}\,\overline{Q_1}\,D & K_2 &= \overline{\overline{Q_3}\,\overline{Q_0}\,\overline{D} + \overline{Q_1}\,Q_0 + Q_1D} \\
J_3 &= Q_2Q_1Q_0\,\overline{D} + \overline{Q_2}\,\overline{Q_1}\,D & K_3 &= \overline{Q_1 + Q_0 + \overline{Q_2}\,\overline{Q_1}\,\overline{D} + Q_2D}
\end{aligned}
$$

## 5.3 Protues仿真

电路图连接如下



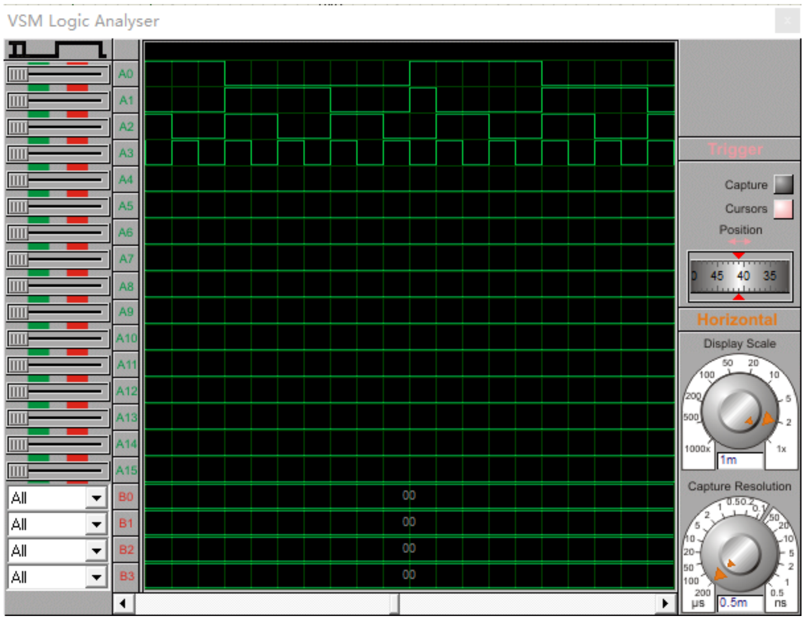
仿真结果如下，正向计数



Display Scale 1m

Capture Resolution 0.5m

逆向计数



波形符合12进制转换

## 5.4 Basys板电路实现

Verilog程序如下，12进制计数器部分，包括控制变量D的实现（变量名为fb）

```
module Counter(
    input clr, // clear, say reset
    input wire clk, // clock
    input wire fb, // add or minus
    output [6:0] seg,
    output [3:0] an
    );

    reg [3:0] out;
    parameter MAX_COUNT = 12;

    // time counter
    localparam MAX_COUNT_TIME = 50_000_000; // 0.5s
    reg [25:0] count; // 26 bits to store count: 2^26 > 5*10^7
    always @ (posedge clk or posedge clr)
    begin
        if (clr == 1) // reset
            count <= 0;
        else if (count == MAX_COUNT_TIME - 1) // return 0
            count <= 0;
        else
            count <= count + 1;
    end

    // frequency divisor (flip-flop)
    reg clk_div;
    always @ (posedge clk, posedge clr)
    begin
        if (clr == 1)
            clk_div <= 0;
        else if (count == MAX_COUNT_TIME - 1) // reset
            clk_div <= ~clk_div;
        else // set
            clk_div <= clk_div;
    end

    // 12 system counter
    always @ (posedge clk_div or posedge clr)
    if (fb == 0)
```

```
        begin
            if (clr == 1)
                out <= 0;
            if (out == MAX_COUNT)
                out <= 0;
            else
                out <= out + 1;
        end
    else
        begin
            if (clr == 1)
                out <= 0;
            if (out == 0)
                out <= 4'b1100;
            else
                out <= out - 1;
        end

    display disp1 (.digit(out),.seven_seg(seg));

    // only turn on the first seven segment display
    assign an = 4'b1110;

endmodule
```

分频计数器与七段数码管连接部分

```
\scriptsize
module display (
    input [3:0] digit,
    output reg [6:0] seven_seg
    );

    always @(digit)
        case(digit)
            1: seven_seg = 7'b100_1111;
            2: seven_seg = 7'b001_0010;
            3: seven_seg = 7'b000_0110;
            4: seven_seg = 7'b100_1100;
            5: seven_seg = 7'b010_0100;
            6: seven_seg = 7'b010_0000;
            7: seven_seg = 7'b000_1111;
            8: seven_seg = 7'b000_0000;
            9: seven_seg = 7'b000_0100;
            10: seven_seg = 7'b000_1000; // a
            11: seven_seg = 7'b110_0000; // b
            12: seven_seg = 7'b011_0001; // c
        endcase // clk

endmodule
```

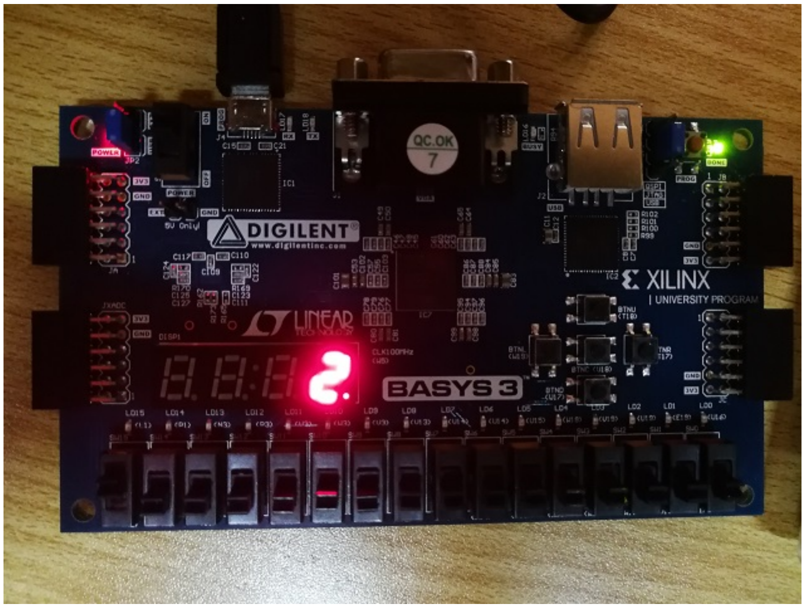
限制文件如下

```
set_property PACKAGE_PIN W4 [get_ports {an[3]}]
set_property PACKAGE_PIN V4 [get_ports {an[2]}]
set_property PACKAGE_PIN U4 [get_ports {an[1]}]
set_property PACKAGE_PIN U2 [get_ports {an[0]}]
set_property PACKAGE_PIN W7 [get_ports {seg[6]}]
set_property PACKAGE_PIN W6 [get_ports {seg[5]}]
set_property PACKAGE_PIN U8 [get_ports {seg[4]}]
set_property PACKAGE_PIN V8 [get_ports {seg[3]}]
set_property PACKAGE_PIN U5 [get_ports {seg[2]}]
set_property PACKAGE_PIN V5 [get_ports {seg[1]}]
set_property PACKAGE_PIN U7 [get_ports {seg[0]}]

set_property PACKAGE_PIN W5 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {seg[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports clk]

set_property PACKAGE_PIN V17 [get_ports clr]
set_property PACKAGE_PIN R2 [get_ports fb]
set_property IOSTANDARD LVCMOS33 [get_ports clr]
set_property IOSTANDARD LVCMOS33 [get_ports fb]
```

最终结果图如下



通过调整最左下角的拨码开关，可实现任意时刻计数器的正向计数和反向计数

## 6 心得体会

1. 学会了移位寄存器及JK触发器的使用及其功能
2. 明白同步异步时序电路如何进行设计
3. 更好地学会用Verilog进行电路设计分析，并成功上板实现