

第4章

时序电路Verilog设计

4.1 基本时序元件的Verilog表述

4.1.1 基本D触发器单元及其Verilog表述

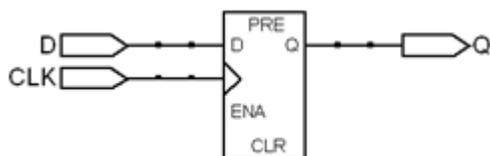


图 4-1 边沿触发型 D 触发器



图 4-2 D 触发器时序波形

【例 4-1】

```
module DFF1(CLK,D,Q);  
    output Q;  
    input  CLK, D;  
    reg Q;  
    always @(posedge CLK )  
        Q <= D;  
endmodule
```

4.1 基本时序元件的Verilog表述

4.1.2 用UDP表述D触发器

【例4-2】

```
primitive EDGE_UDP (Q,D,CLK,RST);
input D,CLK,RST; output Q; reg Q;
table // D CLK RST : Q : Q+
    0 (01) 0 : ? : 0;
    1 (01) 0 : ? : 1;
    ? (1?) 0 : ? : -;
    ? (?0) 0 : ? : -;
    1 0 1 : ? : 0;
    1 1 1 : ? : 0;
    0 0 1 : ? : 0;
    0 1 1 : ? : 0;
endtable
endprimitive
```

【例4-3】

```
module DFF_UDP (Q,D,CLK,RST) ;
input D,CLK,RST;
output Q;
EDGE_UDP U1 (Q, D,CLK,RST);
Endmodule
```

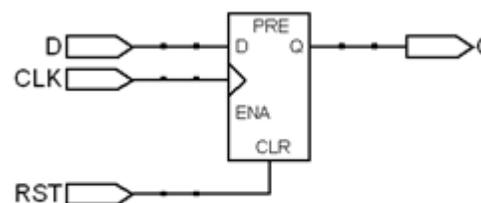


图 4-3 含异步复位的边沿触发型 D 触发器

4.1 基本时序元件的Verilog表述

4.1.3 含异步复位和时钟使能的D触发器及其Verilog表述

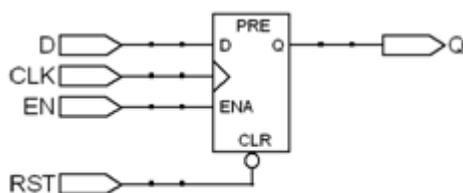


图 4-4 含使能和复位控制的 D 触发器

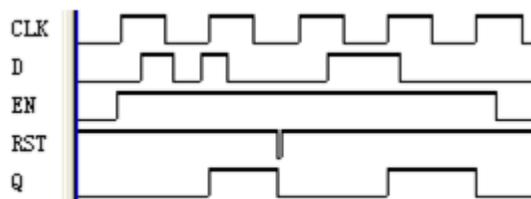


图 4-5 图 4-4 的 D 触发器的时序图

【例 4.4】

```
module DFF2 (CLK, D, Q, RST, EN);  
    output Q;  
    input CLK, D, RST, EN;  
    reg Q;  
    always @(posedge CLK or negedge RST)  
    begin  
        if (!RST) Q <= 0;  
        else if (EN) Q <= D;  
    end  
endmodule
```

4.1 基本时序元件的Verilog表述

4.1.4 含同步复位控制逻辑的D触发器及其Verilog表述

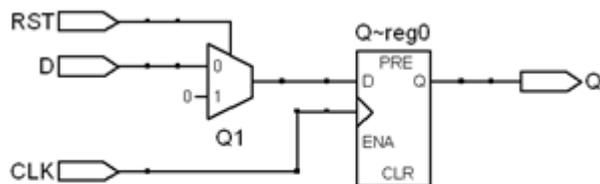


图 4-6 含同步清 0 控制的 D 触发器

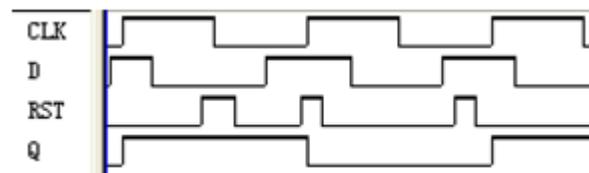


图 4-7 含同步清 0 控制 D 触发器的时序图

4.1 基本时序元件的Verilog表述

4.1.4 含同步复位控制逻辑的D触发器及其Verilog表述

【例4-5】

```
module DFF3 (CLK, D, Q, RST);
    output Q;
    input CLK, D, RST;
    reg Q;
    always @(posedge CLK)
        if (RST==1) Q = 0;
    else if (RST==0) Q = D;
        else      Q = Q;
endmodule
```

【例4-6】

```
module DFF1 (CLK, D, Q, RST);
    output Q; input CLK, D, RST;
    reg Q, Q1; //注意定义了Q1信号
    always @(RST) //纯组合过程
        if (RST==1) Q1=0;
            else Q1=D;
    always @(posedge CLK)
        Q <= Q1;
endmodule
```

```
module DFF2 (input CLK, input D, input RST, output reg Q);
    always @(posedge CLK)
        Q <= RST ? 1'b0 : D;
endmodule
```

4.1 基本时序元件的Verilog表述

4.1.5 基本锁存器及其Verilog表述

【例 4-7】

```
module LATCH1(CLK,D,Q);  
    output Q ; input CLK,D;  
    reg Q;  
    always @(D or CLK)  
        if(CLK) Q <= D;  
endmodule
```

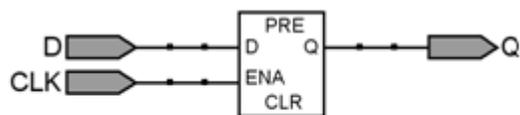


图 4-8 锁存器模块

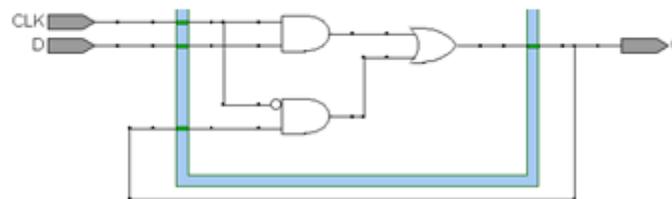


图 4-9 锁存器模块内部逻辑电路

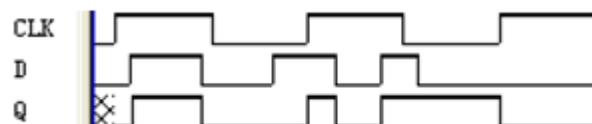


图 4-10 例 4-7 锁存器的时序波形

4.1 基本时序元件的Verilog表述

4.1.6 含清0控制的锁存器及其Verilog表述

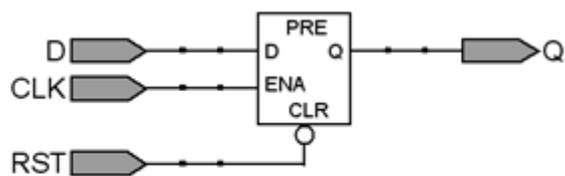


图 4-11 含异步清 0 的锁存器

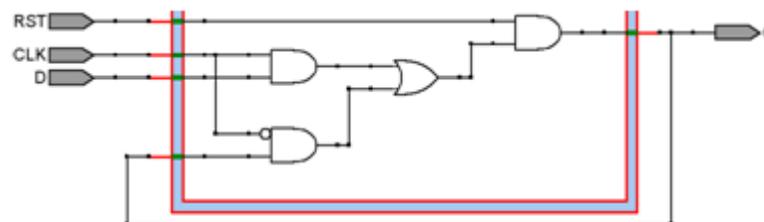


图 4-12 含异步清 0 锁存器的逻辑电路图

4.1 基本时序元件的Verilog表述

4.1.6 含清0控制的锁存器及其Verilog表述

【例 4-8】

```
module LATCH2 (CLK,D,Q,RST);  
    output Q ;    input CLK,D,RST;  
    assign Q = (!RST)? 0:(CLK ? D:Q);  
endmodule
```

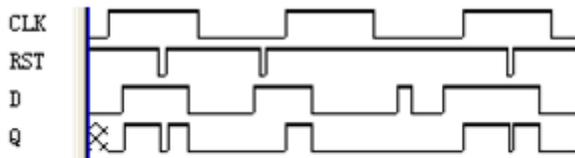


图 4-13 含异步清 0 的锁存器的仿真波形

【例 4-9】

```
module LATCH3 (CLK,D,Q,RST);  
    output Q ;  
    input CLK,D,RST;  
    reg Q;  
    always @(D or CLK or RST)  
        if(!RST) Q<=0;  
        else  
            if(CLK) Q<=D;  
endmodule
```

4.1 基本时序元件的Verilog表述

4.1.7 异步时序电路的Verilog表述特点

【例 4-10】

```
module AMOD(D,A,CLK,Q);  
    output Q ; input A,D,CLK;  
    reg Q,Q1;  
    always @(posedge CLK)  
        begin Q1 = ~(A | Q); end  
    always @(posedge Q1 )  
        begin Q = D;          end  
endmodule
```

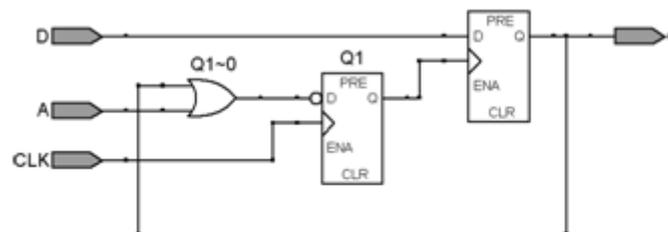
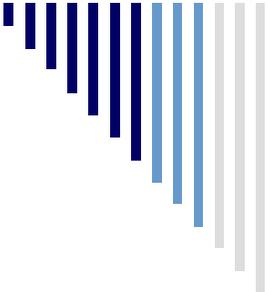


图 4-14 例 4-10 的时序电路图



4.1 基本时序元件的Verilog表述

4.1.8 时钟过程表述的特点和规律

`always @(posedge CLK or RST), 或 always @(posedge CLK or negedge RST or A)` ✘

`always @(posedge CLK or negedge RST) begin if (! RST) ...`

`always @(posedge CLK or negedge RST) begin if (RST==0) ...`

`always @(posedge CLK or negedge RST) begin if (! RST==1) ...`

`always @(posedge CLK or DIN)`

4.1 基本时序元件的Verilog表述

4.1.8 时钟过程表述的特点和规律

【例 4-11】

```
module DFF5 (CLK, D, Q, RST, DIN, OUT);  
output Q, OUT; input CLK, D, RST, DIN;  
reg Q, OUT;  
always @(posedge CLK ) begin  
    OUT = !DIN ;  
    if (RST==1) Q=0;  
    else if(RST==0) Q=D;  
end  
endmodule
```

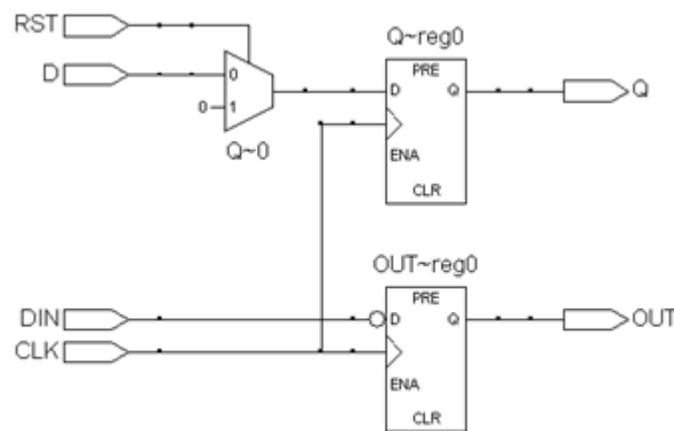


图 4-15 例 4-11 的 RTL 图

4.2 二进制计数器及其Verilog表述

4.2.1 简单加法计数器及其Verilog表述

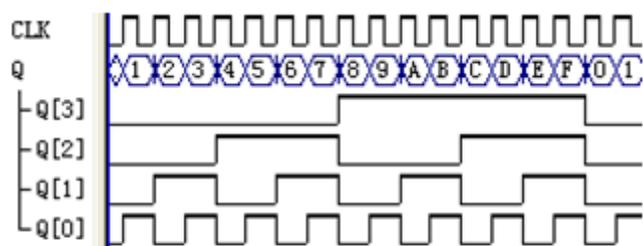


图 4-16 4 位加法计数器工作时序

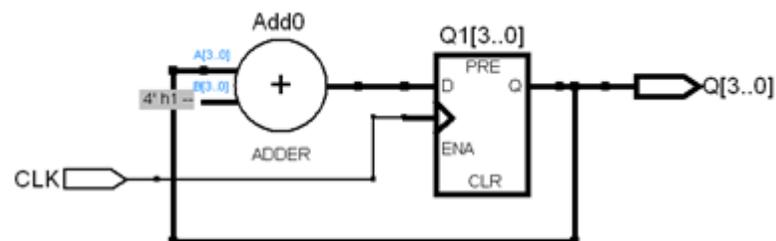


图 4-17 4 位加法计数器 RTL 电路图



4.2 二进制计数器及其Verilog表述

4.2.1 简单加法计数器及其Verilog表述

【例4-12】

```
module CNT4(CLK,Q);  
    output [3:0] Q;    input  CLK;  
    reg [3:0] Q1 ;  
    always @(posedge CLK)  
        Q1 <= Q1+1 ;  
    assign Q=Q1;  
endmodule
```

【例4-13】

```
module CNT4 (CLK,Q);  
    output [3:0] Q ;  
    input  CLK;  
    reg [3:0] Q ;  
    always @(posedge CLK)  
        Q <= Q+1 ;  
endmodule
```

4.2 二进制计数器及其Verilog表述

4.2.2 实用加法计数器设计

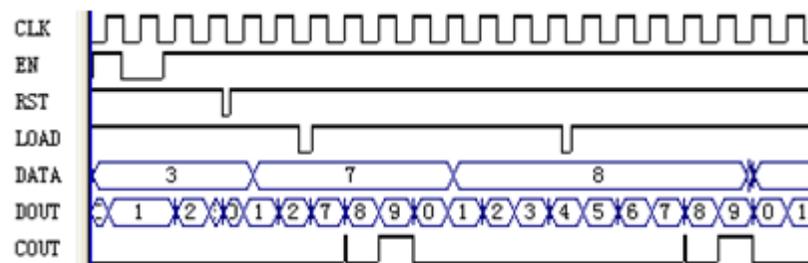


图 4-18 例 4-14 的仿真波形图

【例 4-14】

```
module CNT10 (CLK,RST,EN,LOAD,COUT,DOUT,DATA);
    input CLK,EN,RST,LOAD ; // 时钟, 时钟使能, 复位, 数据加载控制信号;
    input [3:0] DATA ; // 4 位并行加载数据
    output [3:0] DOUT ; // 4 位计数输出
    output COUT ; // 计数进位输出
    reg [3:0] Q1 ; reg COUT ;
    assign DOUT = Q1; // 将内部寄存器的计数结果输出至 DOUT
    always @(posedge CLK or negedge RST) //时序过程
        begin
            if (!RST) Q1 <= 0; //RST=0 时, 对内部寄存器单元异步清 0
            else if (EN) begin //同步使能 EN=1, 则允许加载或计数
                if (!LOAD) Q1<=DATA; //当 LOAD=0, 向内部寄存器加载数据
                else if (Q1<9) Q1 <= Q1+1; //当 Q1 小于 9 时, 允许累加
                else Q1 <= 4'b0000; end //否则一个时钟后清 0 返回初值
            end
        end
    always @(Q1) //组合过程
        if (Q1==4'h9) COUT = 1'b1; //当 Q1=1001 时, COUT 输出进位标志 1
        else COUT = 1'b0; //否则, 输出进位标志 0
endmodule
```

4.2 二进制计数器及其Verilog表述

4.2.2 实用加法计数器设计

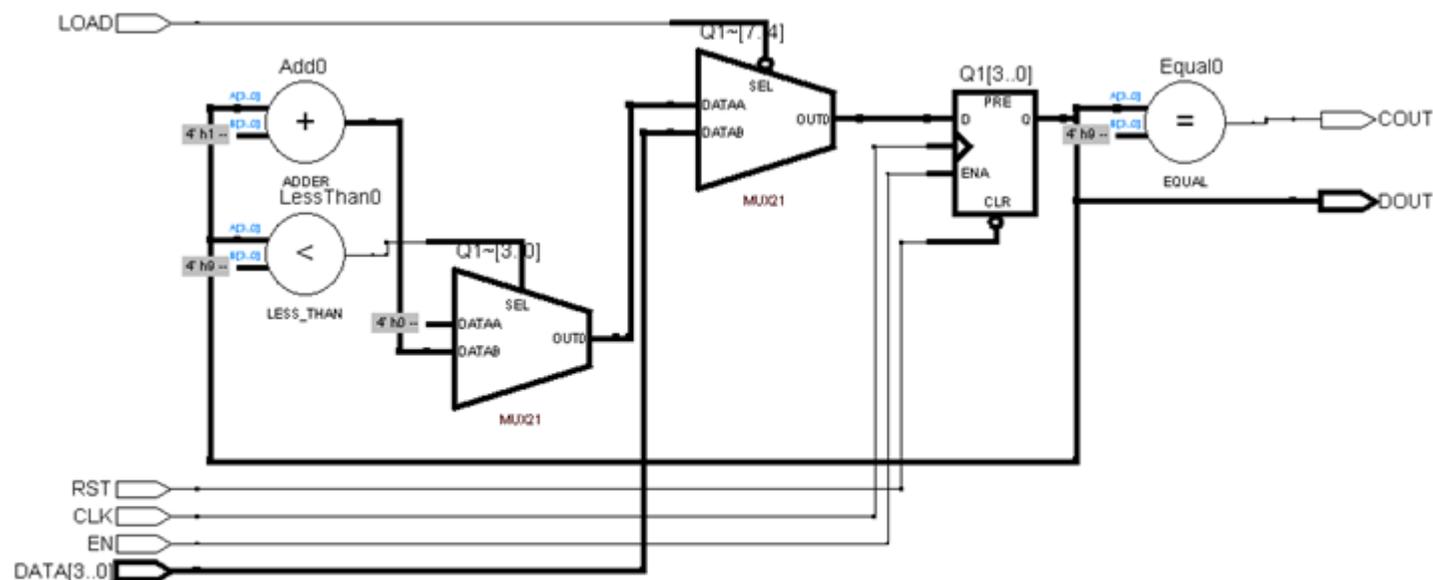


图 4-19 QuartusII 对例 4-14 综合和后得到的 RTL 电路图

4.3 移位寄存器的Verilog表述与设计

4.3.1 含同步预置功能的移位寄存器设计

【例4-15】

```
module SHFT1 (CLK, LOAD, DIN, QB);  
    output QB; input CLK, LOAD;  
    input[7:0] DIN; reg[7:0] REG8;  
    always @(posedge CLK )  
        if (LOAD)    REG8<=DIN ;  
        else REG8[6:0]<=REG8[7:1];  
    assign QB = REG8[0] ;  
endmodule
```

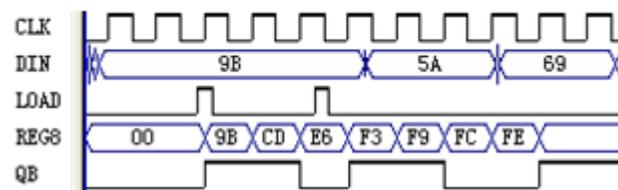


图 4-20 例 4-15 的工作时序图

4.3 移位寄存器的Verilog表述与设计

4.3.2 模式可控的移位寄存器设计

【例 4-16】

```
module SHFT2 (CLK,C0,MD,D,QB,CN);
output CN ; output [7:0] QB;    //进位输出和移位数据输出
input CLK,C0;                  //时钟和进位输入
input [7:0] D; input [2:0] MD; //待加载移位的数据输入和移位模式控制字
reg[7:0] REG ; reg CY ;
always @(posedge CLK ) begin
case (MD)
1 : begin REG[0]<=C0; REG[7:1]<=REG[6:0]; CY<=REG[7]; end//带进位循环左移
2 : begin REG[0]<=REG[7] ; REG[7:1]<=REG[6:0] ; end //自循环左移
3 : begin REG[7]<=REG[0] ; REG[6:0]<=REG[7:1] ; end //自循环右移
4 : begin REG[7]<=C0; REG[6:0]<=REG[7:1];CY<=REG[0]; end //带进位循环右移
5 : begin REG<=D ; end //加载待移数
default : begin REG <= REG ; CY <= CY ; end //过程结束
endcase end
assign QB = REG ; assign CN = CY ; //移位后输出, 及移位后输出
endmodule
```

4.3 移位寄存器的Verilog表述与设计

4.3.2 模式可控的移位寄存器设计

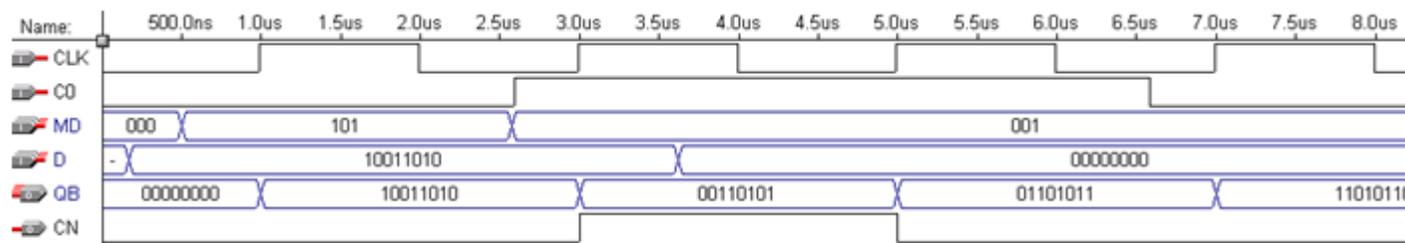


图 4-21 例 4-16 中带进位循环左移仿真波形(MD =“001”)



4.3 移位寄存器的Verilog表述与设计

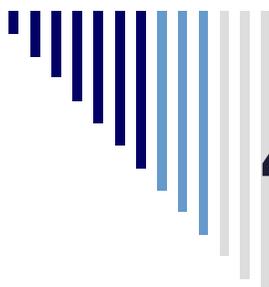
4.3.3 使用移位操作符设计移位寄存器

【例 4-17】

```
module SHIF4(DIN,CLK,RST,DOUT);
    input CLK,DIN,RST;
    output DOUT;
    reg [3:0] SHFT;
    always@(posedge CLK or posedge RST)
        if(RST) SHFT<=4'B0;
        else begin    SHFT[3]<=DIN;
            SHFT[2:0] <= SHFT[3:1];
        end
        assign DOUT=SHFT[0];
endmodule
```

【例 4-18】

```
module SHIF4 (DIN,CLK,RST,DOUT);
    input CLK,DIN,RST;    output DOUT;
    reg [3:0] SHFT;
    always@(posedge CLK or posedge RST)
        if(RST) SHFT<=4'B0;
        else begin
            SHFT <= (SHFT >> 1);
            SHFT[3] <= DIN;
        end
        assign DOUT = SHFT[0];
endmodule
```



4.4 时序电路硬件设计与仿真示例

【例 4-19】

```
module CNT_DCD
(CLK,EN,COUT,LED);
    input CLK,EN ;
    output COUT;
    output[6:0] LED ;
    (*synthesis, keep*) wire [3:0]net1;
    CNT10  U1 (.CLK(CLK), .EN(EN),
    .COUT(COUT),.DOUT(net1) );
    DCOD7S U2 (.A(net1), .LED7S(LED));
endmodule
```

【例 4-20】

```
module CNT10 (CLK,EN,COUT,DOUT);
    input CLK,EN ; output [3:0]DOUT;
    output COUT ; reg [3:0] Q1 ;
    reg COUT ; assign DOUT = Q1;
    always @(posedge CLK)
        if(EN) if(Q1<9) Q1<=Q1+1;
        else Q1<=4'b0000; else Q1 <= Q1;
    always @(Q1)
        if (Q1==4'h9) COUT = 1'b1;
        else COUT = 1'b0;
endmodule
```

4.4 时序电路硬件设计与仿真示例

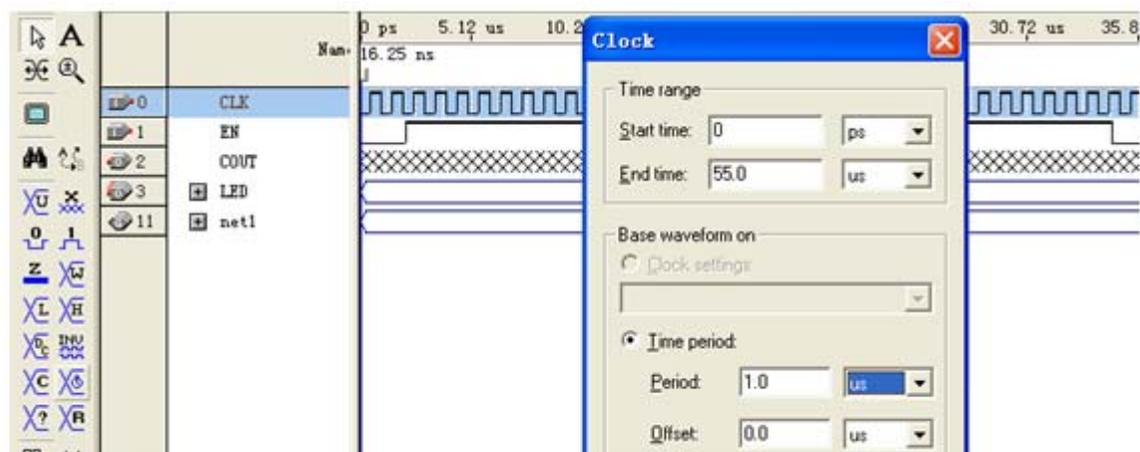


图 4-22 设置时钟 CLK 的周期

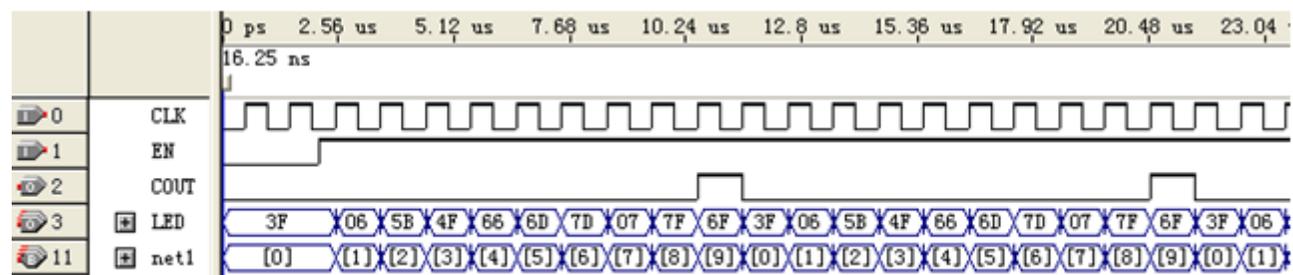


图 4-23 仿真波形输出报告 Simulation Report

4.5 SignalTap II的使用方法

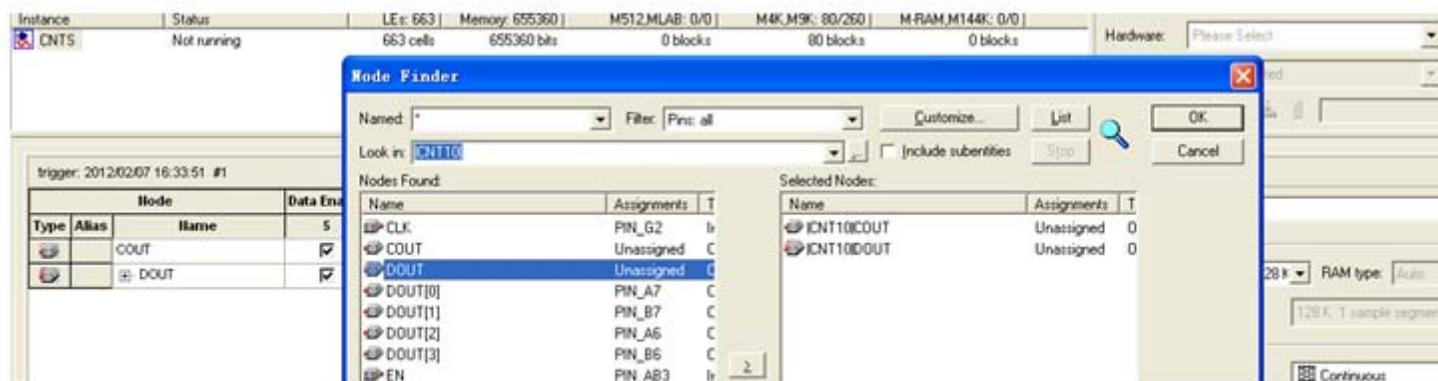


图 4-24 输入逻辑分析仪测试信号

4.5 SignalTap II的使用方法

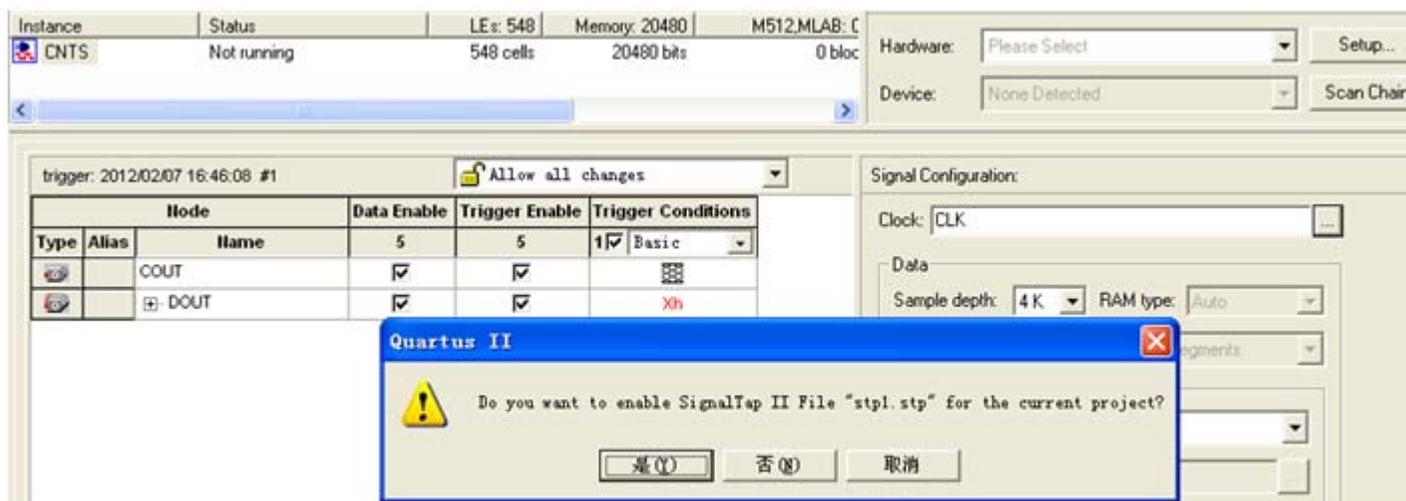


图 4-25 SignalTap II 编辑窗口

4.5 SignalTap II的使用方法

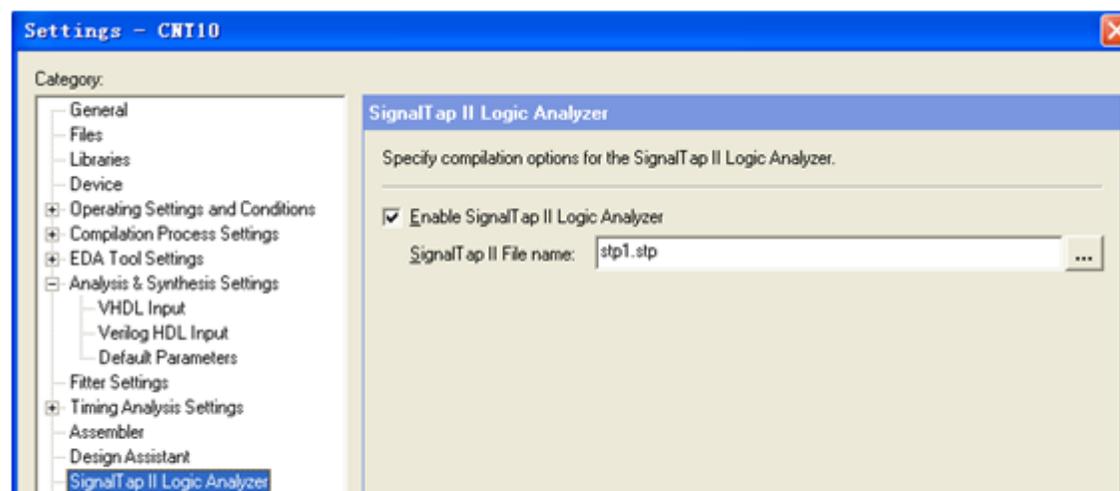


图 4-26 选择或删除 SignalTap II 文件加入综合编译

4.5 SignalTap II 的使用方法

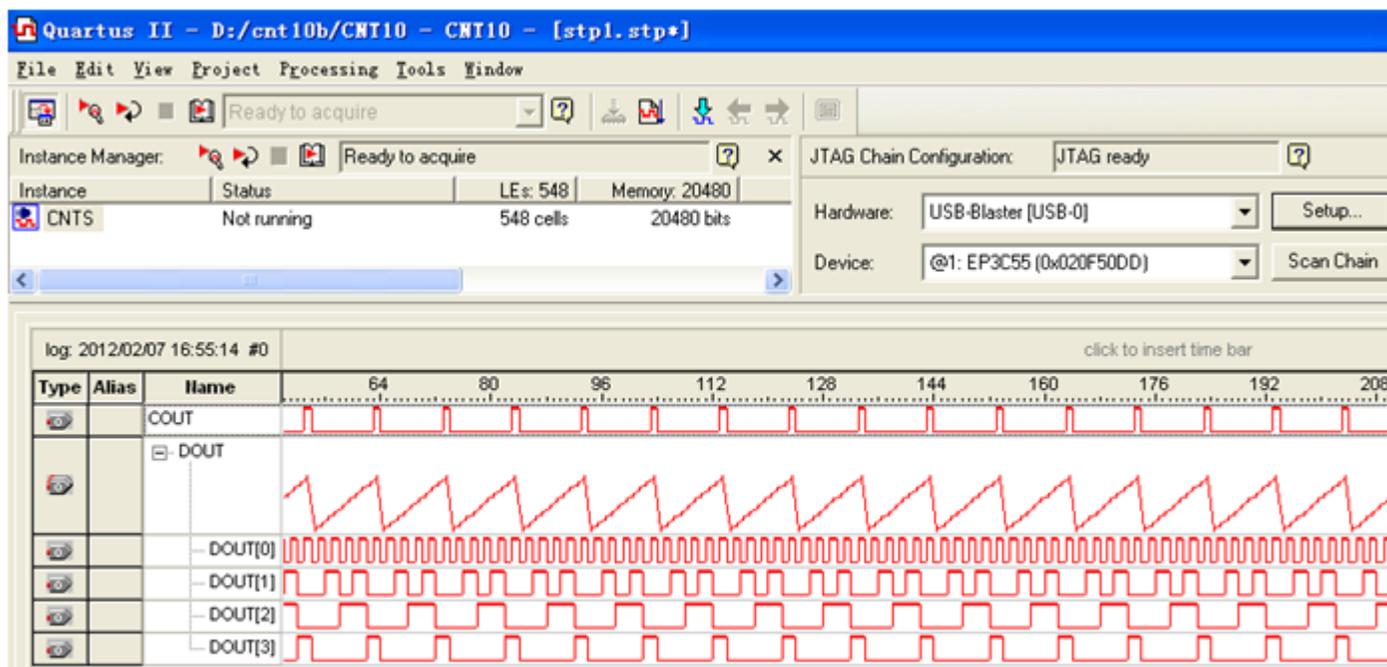
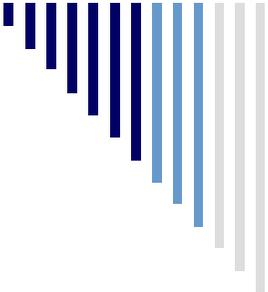


图 4-27 SignalTap II 数据窗口设置后的信号波形



4.5 SignalTap II的使用方法

【例 4-21】

```
module CNT10 (CLK,EN,COUT,DOUT,CLK0);  
    input CLK    /* synthesis chip_pin = "G2" */ ; // 计数器工作时钟  
    input CLK0  /* synthesis chip_pin = "G21" */; // 逻辑分析仪采样时钟
```

习题

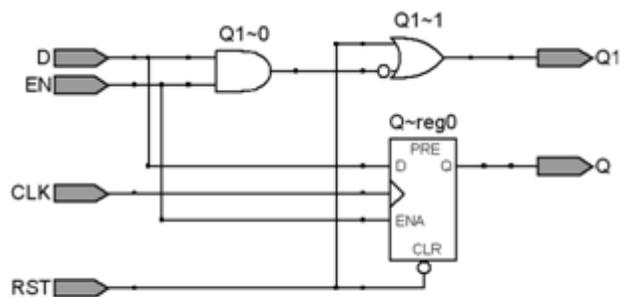


图 4-28 RTL 图 1

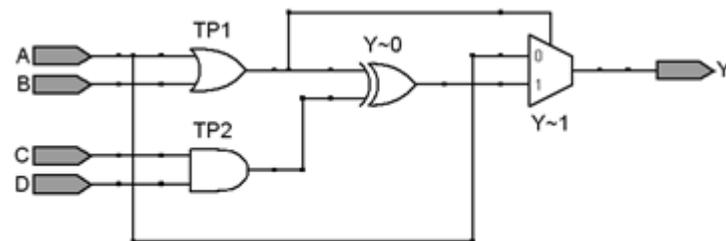


图 4-29 RTL 图 2

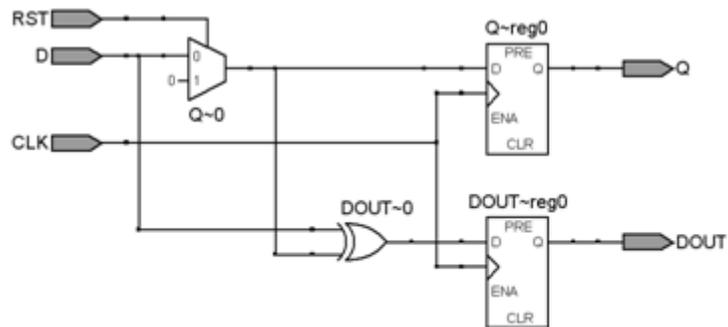


图 4-30 RTL 图 3

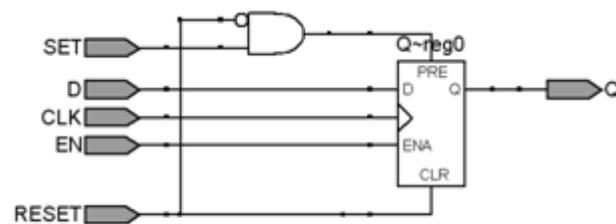


图 4-31 RTL 图 4

EDA实验

4-1 数字计数器设计实验

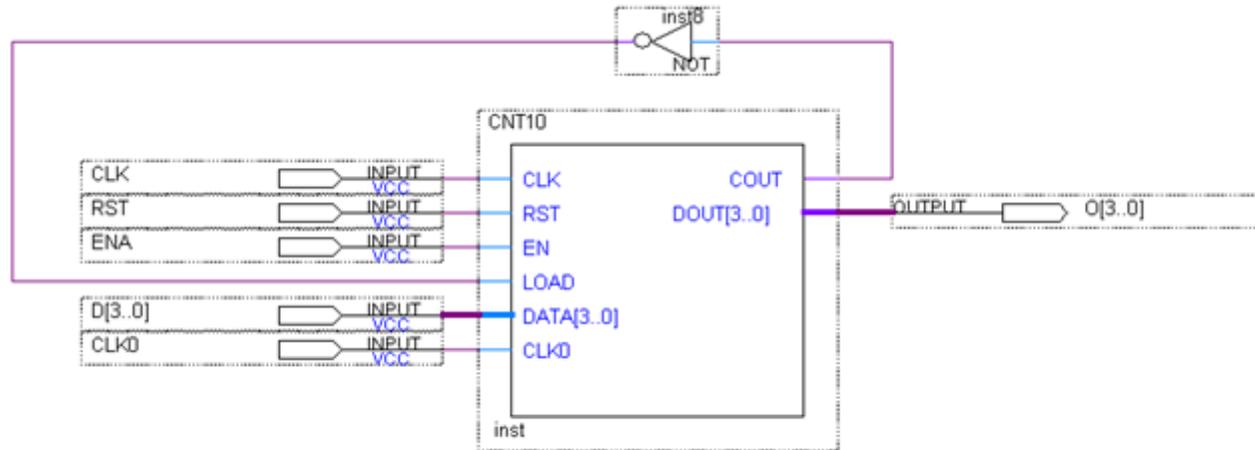


图 4-32 原理图示意图

EDA实验

4-2 十六进制7段数码显示译码器设计

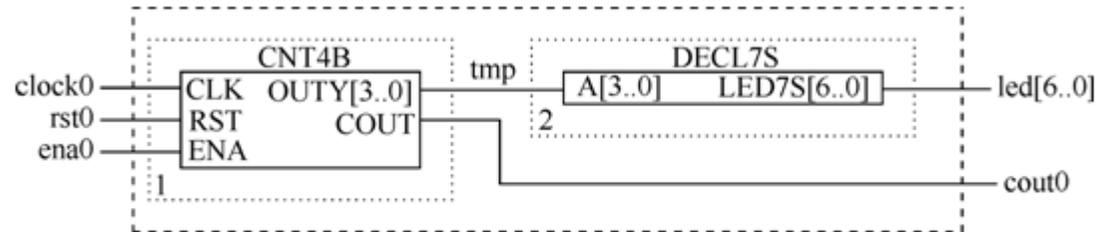


图 4-33 计数器和译码器连接电路的顶层文件原理图

EDA实验

4-3 数码扫描显示电路设计

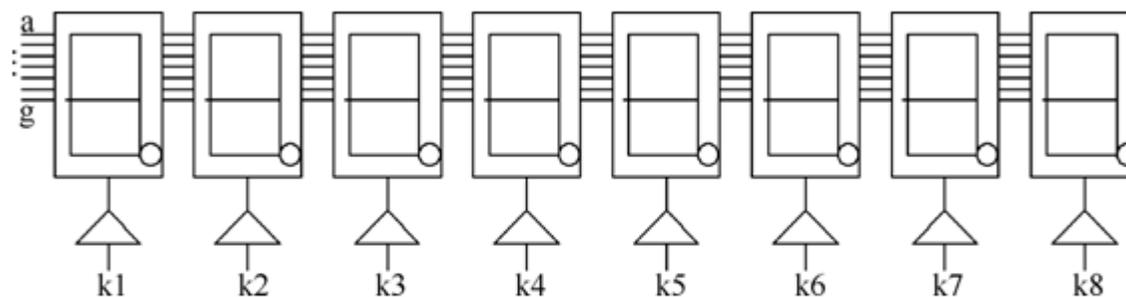


图 4-34 8 位数码扫描显示电路

EDA实验

4-4 模可控计数器设计

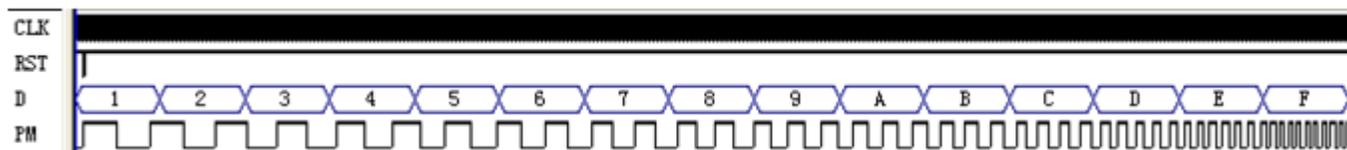


图 4-35 针对不同预置数，占空比均衡后的分频器输出

4-5 移位寄存器设计

EDA实验

4-6 串行静态显示控制电路设计

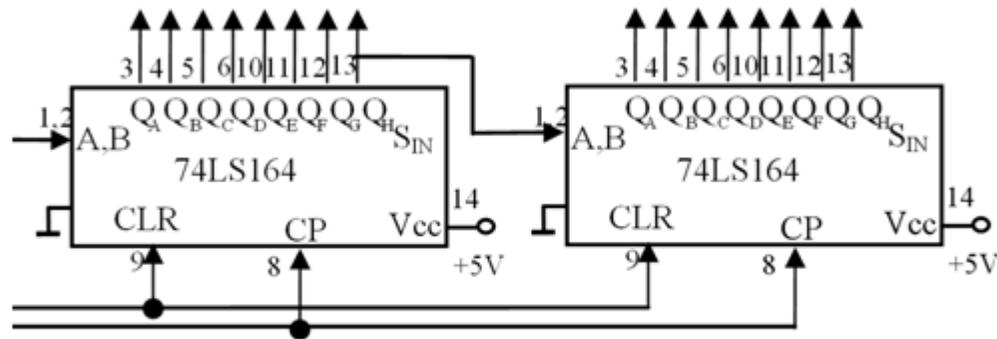


图 4-36 串/并转换数码管静态显示电路

EDA实验

4-7 应用宏模块设计频率计

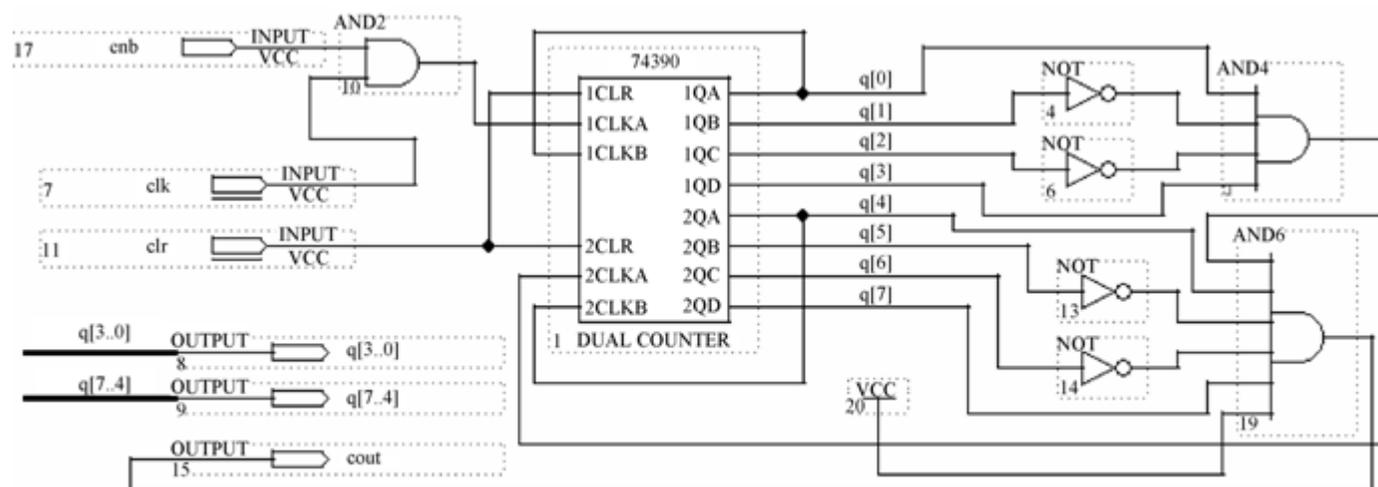


图 4-37 含有时钟使能的 2 位十进制计数器

EDA实验

4-7 应用宏模块设计频率计

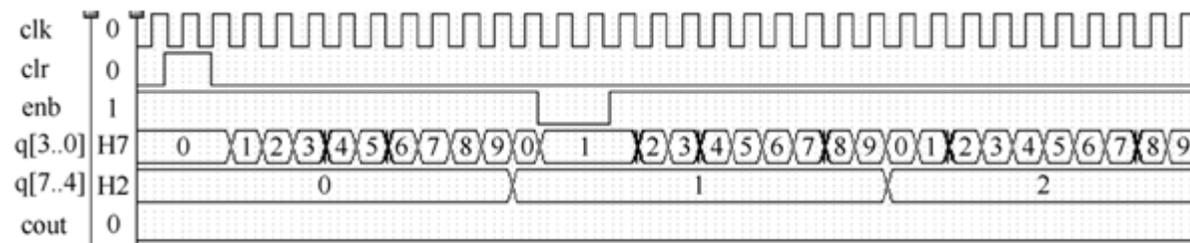


图 4-38 两位十进制计数器工作波形

EDA实验

4-7 应用宏模块设计频率计

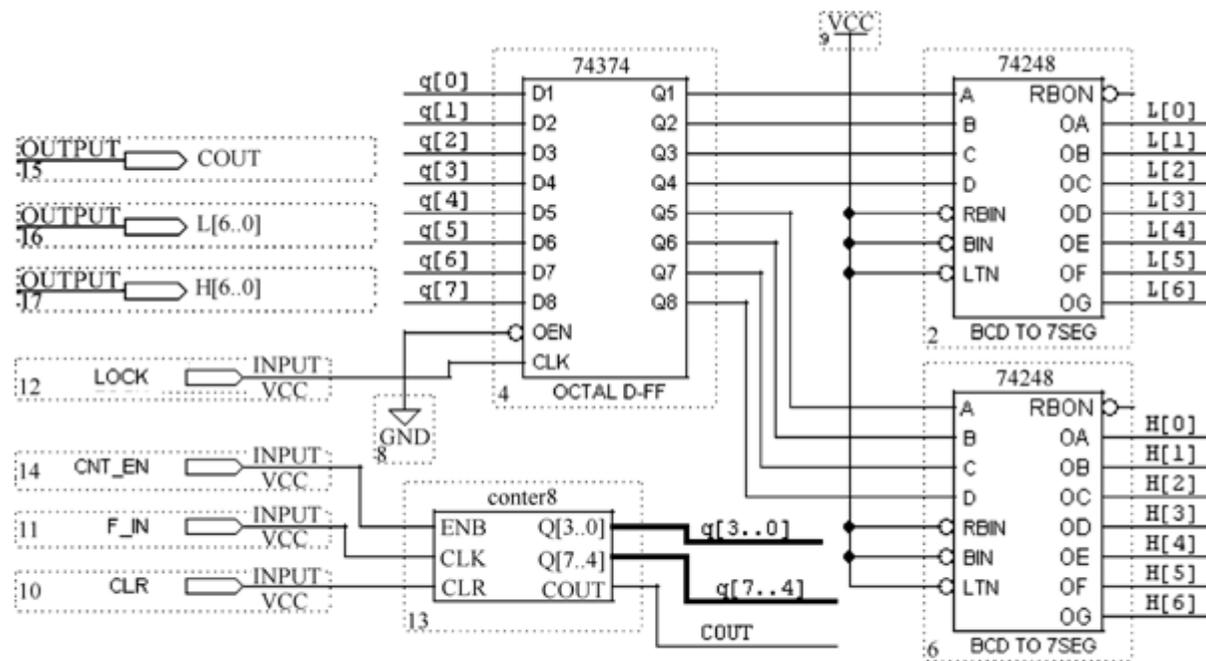


图 4-39 2 位十进制频率计顶层设计原理图文件

EDA实验

4-7 应用宏模块设计频率计

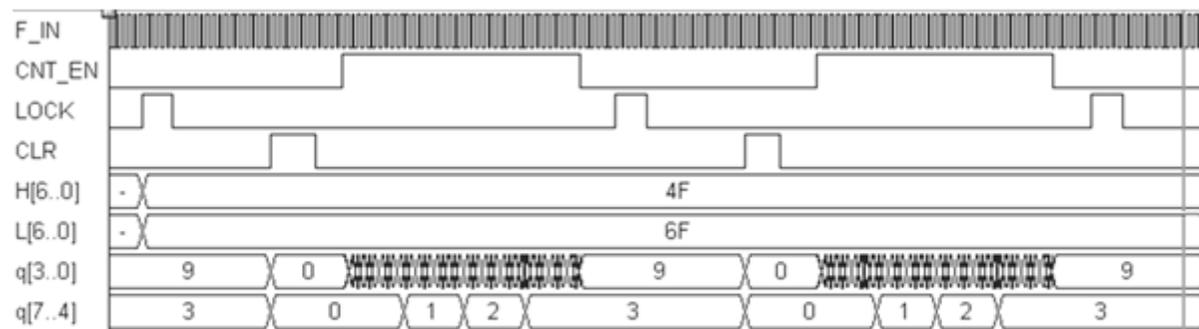


图 4.40 两位十进制频率计测频仿真波形

EDA实验

4-7 应用宏模块设计频率计

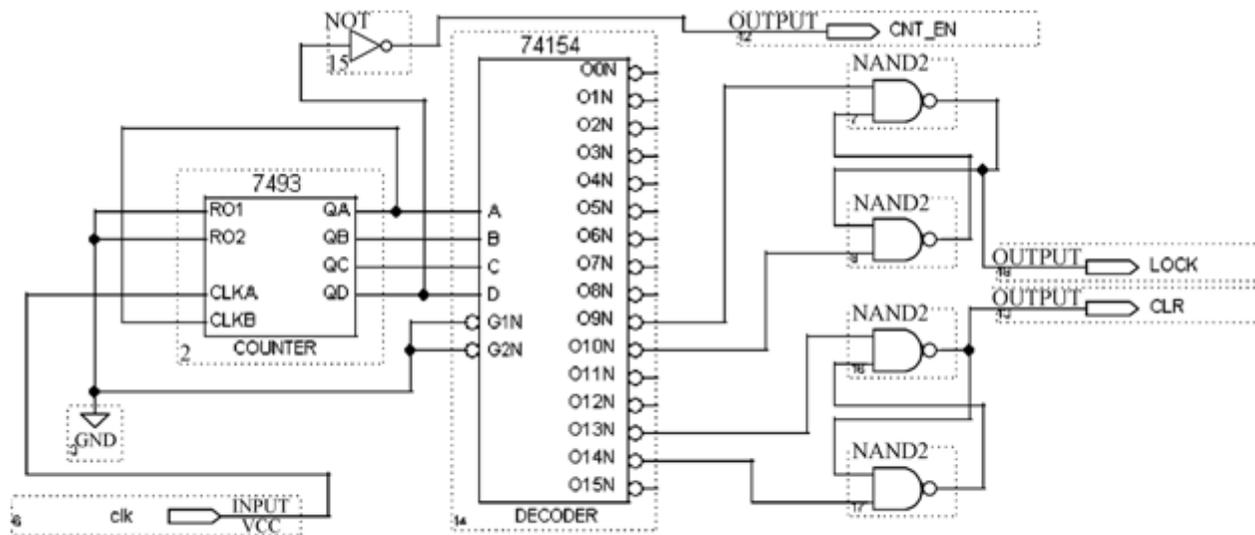


图 4-41 测频时序控制电路

EDA实验

4-7 应用宏模块设计频率计

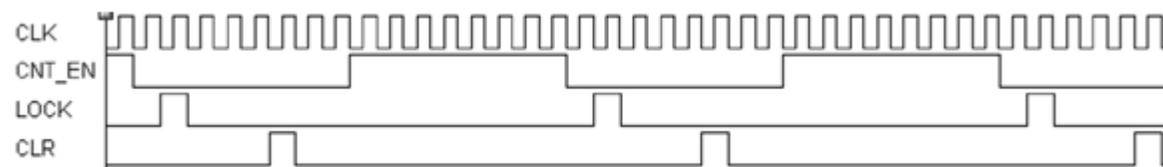


图 4.42 测频时序控制电路工作波形

EDA实验

4-7 应用宏模块设计频率计

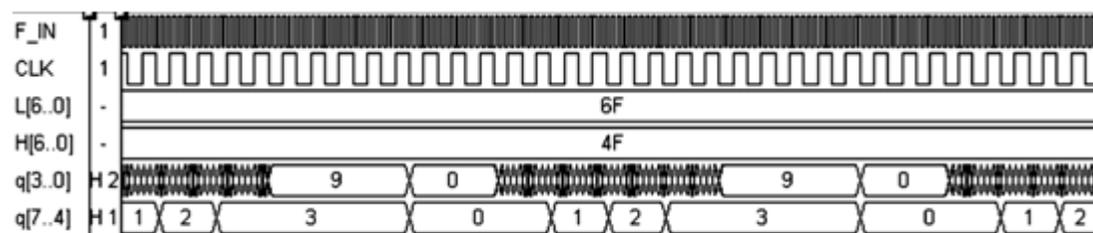


图 4-44 频率计工作时序波形