一、设计题目

每位同学自选下列题目之一。

（1）时序信号发生器设计

（2）地址译码电路设计

（3）自选（内容要求参见（1）、（2））

二、设计内容要求

2.1、时序信号发生器设计

用Verilog实现一个满足设计要求的时序信号发生器完整电路。

2.1.1设计要求

时序信号是使计算机能够准确、迅速、有条不紊地工作的时间基准。CPU每读出并执行

一条指令所需要的时间通常叫做一个指令周期，一个指令周期一般由若干个CPU周期（通常定义为从内存中读取一个指令字的最短时间，又称为机器周期）组成。

这里的时序信号用节拍表示，一个节拍表示一个CPU周期时间，在一个节拍中又包含若干个节拍脉冲，节拍脉冲表示较小的时间单位。

时序信号发生器的功能就是产生一系列的节拍脉冲，它一般由：

时钟脉冲源；

时序信号产生电路；

启停控制电路等部分组成。

**要求设计一个用于实验系统的简单时序信号发生器，具体功能如下：**

（1）由时钟脉冲源提供频率稳定的方波信号作为系统的主频信号（即时序发生器的输入信号），要求系统的主频信号可以在100MHz，50MHz两种不同频率间进行选择；

（2）规定一个CPU周期（一个节拍）由4个时钟周期组成，即要求在一个CPU周期中产生4个等间隔的节拍脉冲；

（3）为了保证系统可靠地启动和停止，必须对时序信号进行有效的控制。

此外，由于启动信号和停止信号都是随机产生的，考虑到节拍脉冲的完整性，所以要求时序信号发生器启动时从第一个节拍脉冲的前沿开始工作，停止时在第四个节拍脉冲的后沿关闭。

2.1.2 功能描述

根据设计要求可知，时序信号发生器由时钟脉冲源、时序信号产生电路、启停控制电路3部分组成，其结构框图如图2-1所示。

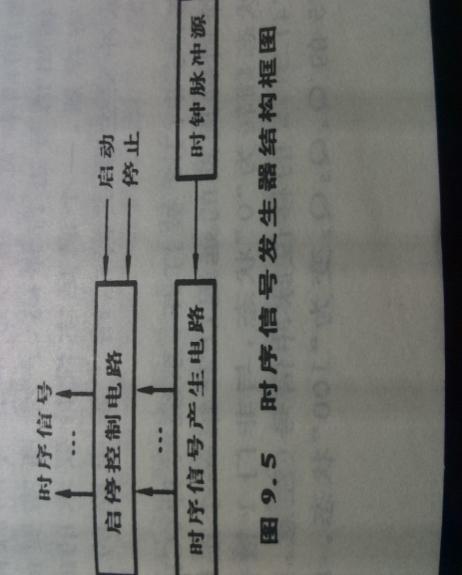


图2-1 时序信号发生器结构框图

假定节拍脉冲信号用T1，T2，T3，T4表示，可画出时序信号发生器产生的波形，如图2-2所示。

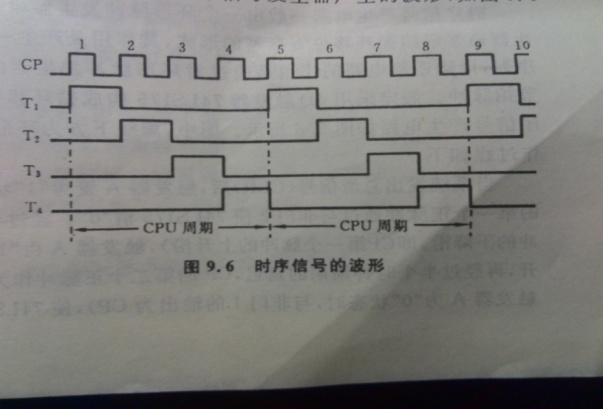


图2-2 时序信号的波形

2.1.3 电路设计

**（1）时钟电路（时钟源）**

由于要求时序发生器的输入信号能在100MHz、50MHz两种不同频率的方波信号之间进行选择。

所以时钟电路应由：信号源和分频电路两部分组成：

a．具体电路图：

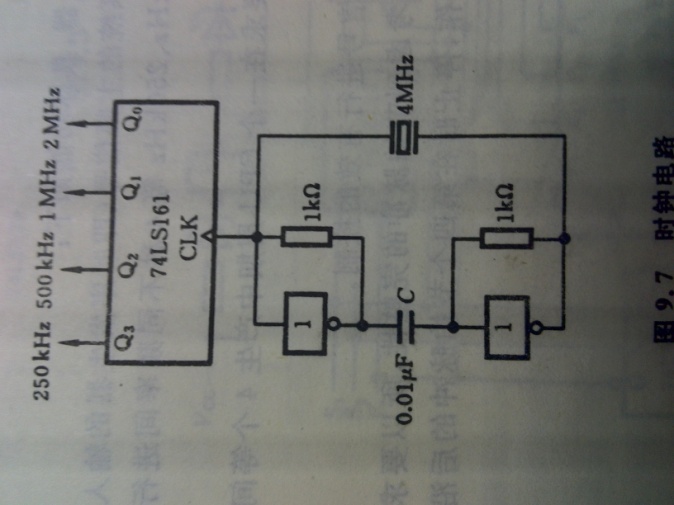


图2-3 信号源与分频电路

b. 该电路的工作过程如下：

由石英晶体发出的200MHz时钟源，作为2位模4计数器的时钟输入端，使得其2个输出端的时钟频率降频为100MHz与50MHz的时钟源。

c. Verilog程序以及RTL级的电路图和仿真图：

1.signal\_source

`timescale 1ns / 1ps

module signal\_source

#(parameter CYCLE = 5)

(

output reg clk

);

initial begin

clk = 0;

end

always begin

#(CYCLE/2.0) clk = ~clk;

end

endmodule

2.binary\_counter

`timescale 1ns / 1ps

module binary\_counter

#(parameter LEN = 4)

(

input clk,

output reg [(LEN-1): 0] Q

);

initial begin

Q <= {(LEN){1'b0}};

end

always @(posedge clk) begin

Q <= Q + 1;

end

endmodule

3.clock\_generator

`timescale 1ns / 1ps

module clock\_generator

#(parameter CYCLE = 5, LEN = 4)

(

output [(LEN-1):0] clk

);

wire src;

signal\_source #(CYCLE) SOURCE (.clk(src));

binary\_counter #(LEN) DIVIDER (.clk(src), .Q(clk));

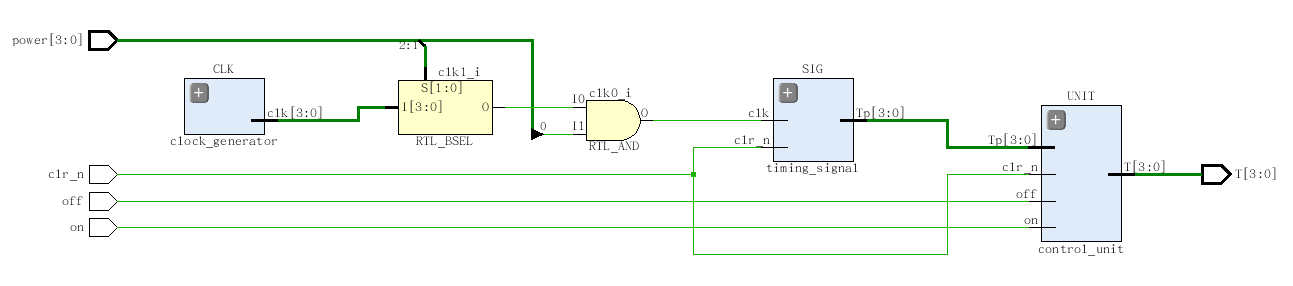
endmodule \

图2-4 RTL级电路图

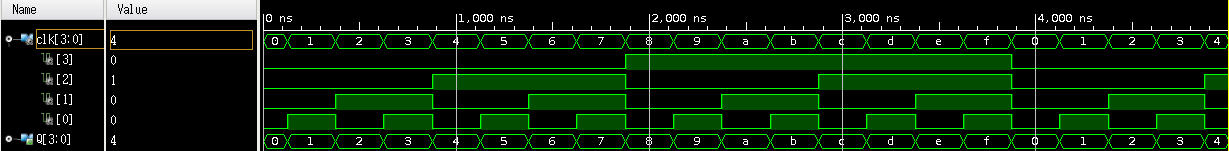


图2-5 仿真图

**（2）时序信号产生电路**

时序信号产生电路一般由一个环形脉冲发生器和一个译码逻辑电路组成。

1. 具体电路图：

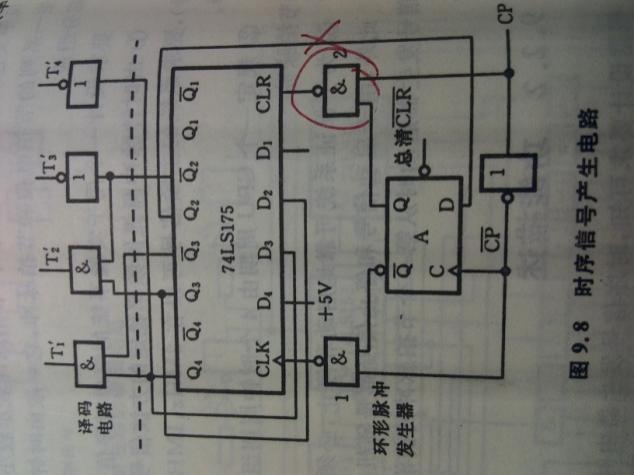


图2-6 环形脉冲发生器与译码逻辑电路

b. 该电路的工作过程如下：

当系统发出总清信号后，触发器A置为”1”状态，与非门2被打开，系统时钟的第一个正脉冲通过与非门2使74LS175清”0。经过半个时钟周期(CP第一个正脉冲下降沿)，触发器A由”1”状态翻转为”0”状态，与非门1被打开；再经过半个时钟周期，CP的第二个正脉冲作为74LS175的移位脉冲信号，使74LS175的Q4Q3Q2变为“100”状态。此后，CP的第三个、第四个正脉冲连续通过门1形成移位脉冲信号，使Q4Q3Q2相继变为“110”、“111”状态。当Q2变为“1”状态，触发器A的输入端D为“1”，因而在CP第四个正脉冲的下降沿又将触发器A置为“1”状态，与非门2再次被打开，CP的第五个正脉冲通过与非门2使74LS175再次清零，开始下一轮循环。

c. 时序信号产生电路的波形图：

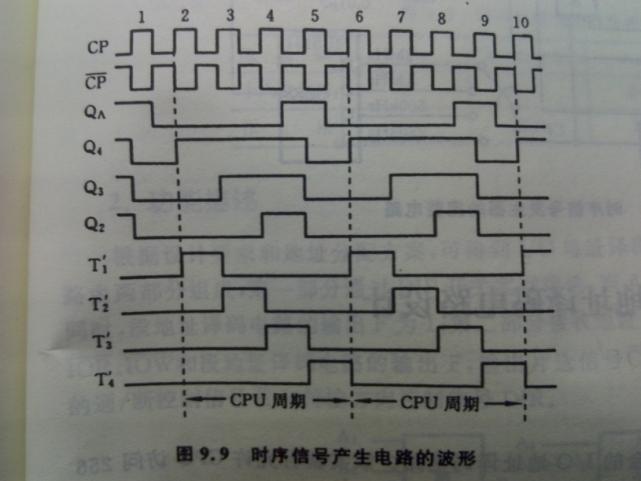


图2-7 时序信号产生电路的波形图

d. Verilog程序以及RTL级的电路图和仿真图：

1.D\_ff

`timescale 1ns / 1ps

module D\_ff

#(parameter LEN = 1)

(

input clk,

input clr\_n,

input [(LEN-1):0] D,

output reg [(LEN-1):0] Q,

output [(LEN-1):0] Qbar

);

assign Qbar = ~Q;

always @(negedge clr\_n or posedge clk) begin

if (!clr\_n) begin

Q <= {(LEN){1'b1}};

end

else begin

Q <= D;

end

end

endmodule

2.rotate\_shift\_register

`timescale 1ns / 1ps

module rotate\_shift\_register

#(parameter LEN = 4)

(

input clk,

input clr,

input [(LEN-1):0] D,

output reg [(LEN-1):0] Q,

output [(LEN-1):0] Qbar

);

assign Qbar = ~Q;

initial begin

Q <= {(LEN){1'b1}};

end

always @(posedge clr or posedge clk) begin

if (clr) begin

Q <= {(LEN){1'b0}};

end

else begin

Q <= D;

end

end

endmodule

3.timing\_signal

`timescale 1ns / 1ps

module timing\_signal

#(parameter LEN = 4)

(

input clk,

input clr\_n,

output [(LEN-1):0] Tp

);

// D\_ff

wire D\_Q, D\_Qbar;

// rotate register

wire [(LEN-1):0] Q, Qbar;

// output timing signal

assign Tp = {Qbar[3], Q[1], Q[2]&Qbar[1], Q[3]&Qbar[2]};

D\_ff #(.LEN(1)) D (

.clk(~clk),

.clr\_n(clr\_n),

.D(Q[1]),

.Q(D\_Q),

.Qbar(D\_Qbar));

rotate\_shift\_register #(LEN) ROTATE (

.clk(~(~clk&D\_Qbar)),

.clr(clk&D\_Q),

.D({1'b1, Q[3], Q[2],1'b0}),

.Q(Q),

.Qbar(Qbar));

endmodule

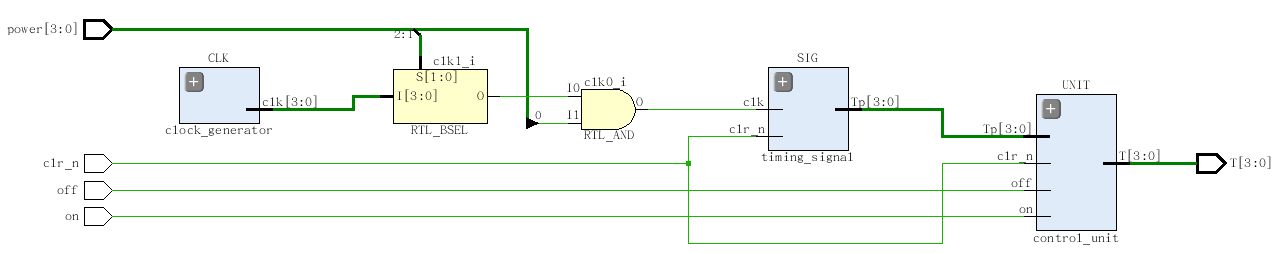


图2-8 RTL级电路图

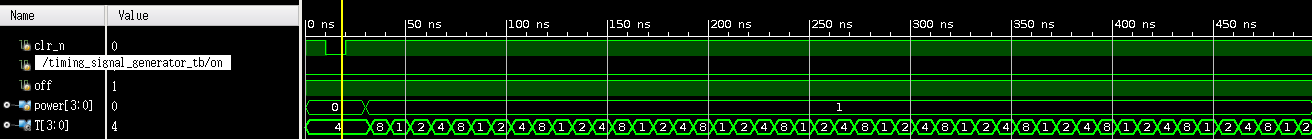


图2-9 仿真图

**（3）启停控制电路**

a. 具体电路图：

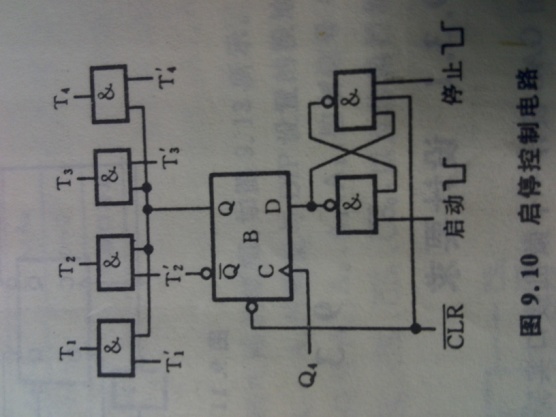


图2-10 启停控制电路

1. 该电路的工作过程如下：

当运行标志触发器为“1”时，允许节拍脉冲T’1~T’4发出；当运行标志触发器为“0”时，节拍脉冲T’1~T’4被封锁。

c. Verilog程序以及RTL级的电路图和仿真图：

1.D\_ff

`timescale 1ns / 1ps

module D\_ff

#(parameter LEN = 1)

(

input clk,

input clr\_n,

input [(LEN-1):0] D,

output reg [(LEN-1):0] Q,

output [(LEN-1):0] Qbar

);

assign Qbar = ~Q;

always @(negedge clr\_n or posedge clk) begin

if (!clr\_n) begin

Q <= {(LEN){1'b1}};

end

else begin

Q <= D;

end

end

endmodule

2.control\_unit

`timescale 1ns / 1ps

module control\_unit

#(parameter LEN = 4)

(

input on,

input off,

input clr\_n,

input [(LEN-1):0] Tp,

output [(LEN-1):0] T

);

// D\_ff

wire Q;

// R\_S latch

wire on\_and, off\_and;

assign on\_and = ~(on & off\_and);

assign off\_and = ~(on\_and & off & clr\_n);

assign T = Tp & {(LEN){Q}};

D\_ff #(.LEN(1)) D (.clk(~Tp[3]), .clr\_n(clr\_n), .D(on\_and), .Q(Q), .Qbar());

endmodule

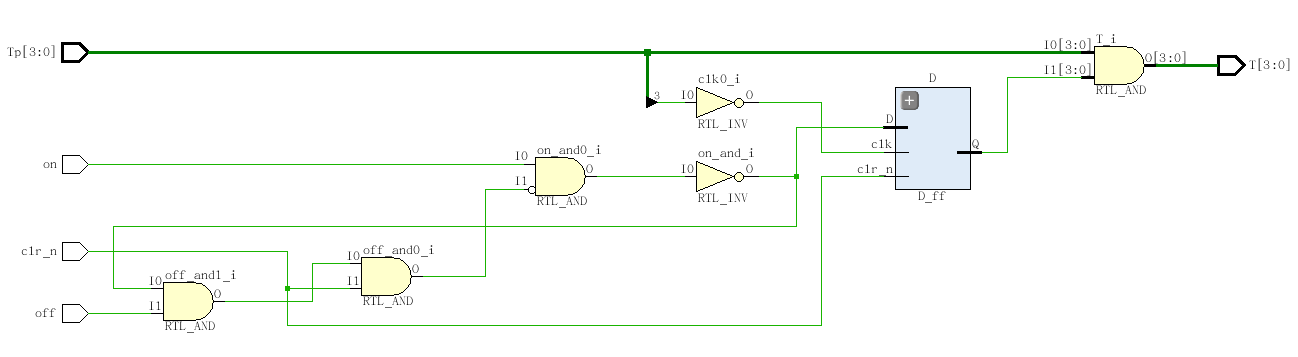


图2-11 RTL级电路图

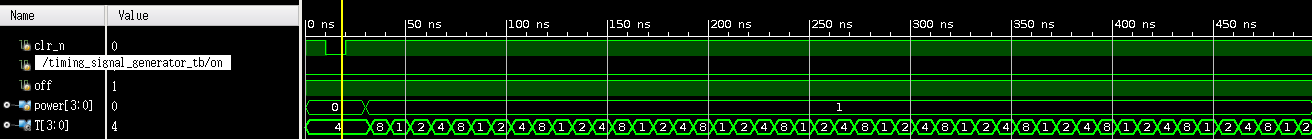


图2-12 仿真图(on/off)

**（4）总电路实现**

`timescale 1ns / 1ps

/\*

\* top module

\*/

module timing\_signal\_generator

#(parameter CYCLE = 5, LEN = 4)

(

input clr\_n,

input on,

input off,

input [3:0] power,

output [(LEN-1):0] T

);

// power[1]: Hz select

// power[0]: on/off

wire [(LEN-1):0] clk\_group, Tp;

clock\_generator #(.CYCLE(CYCLE), .LEN(LEN)) CLK (

.clk(clk\_group));

timing\_signal #(.LEN(LEN)) SIG (

.clk(clk\_group[power[3:1]]&power[0]),

.clr\_n(clr\_n),

.Tp(Tp));

control\_unit #(.LEN(LEN)) UNIT (

.on(on),

.off(off),

.clr\_n(clr\_n),

.Tp(Tp),

.T(T));

endmodule