# 时序逻辑设计

时序电路设计，时序电路是指在任一时刻的输出信号不仅与当时的输入信号有关，而且还与电路的原来状态有关。常用的时序逻辑电路有**计数器、寄存器、锁存器和存储器**等。

## 一、同步计数器

基本原理是将几个触发器按照一定的顺序连接起来，然后根据触发器的组合状态，按照一定的技术规律随着时钟脉冲变化来记忆时钟脉冲的个数。

按照计数方向分为加法，减法和可逆计数器。

按照其中触发器是否与时钟同步又分为同步计数器和异步计数器。

### 1.同步4位计数器，同步清零，同步置数。

*module cnt16 (cout, q, clk, clr, load, en, d);*

*output[3:0] q; //输出*

*output cout; //进位信号*

*input clk, clr, load, en;*

*input[3:0] d;*

*reg[3:0] q;*

*reg cout;*

*always @ (posedge clk) begin*

*if (clr) begin q <= 0; end*

*else if (load) begin q <= d; end*

*else if (en) begin*

*q <= q + 1;*

*if(q == 4’b1111) begin cout <= 1; end*

*else begin cout <= 0; end*

*end*

*else begin q <= q; end*

*end*

*endmodule*

### 同步24进制BCD码计数器，同步清零。

*module cnt24 (ten, one, cout, clk, clr);*

*output[3:0] ten, one; //输出*

*output cout; //进位信号*

*input clk, clr;*

*reg[3;0] ten, one;*

*reg cout;*

*always @ (posedge clk) begin*

*if (clr) begin ten <= 0; one <= 0; end*

*else begin*

*if({ten, one}) == 8’b0010\_0011) //24十进制*

*begin ten <= 0; one <= 0; cout <= 1; end*

*else if(one==4’b1001)*

*begin one <= 0; ten<=ten+1; cout <= 0; end*

*else begin one <= one + 1; cout <=0; end*

*end*

*end*

*endmodule*

### 模为60的BCD码加法计数器counter60.v

*module count60(qout, cout, data, load, cin, reset, clk);*

*output [7:0] qout;*

*output cout;*

*input [7:0] data;*

*input load, cin, clk, reset;*

*reg [7:0] qout;*

*always @ (posedge clk)*

*begin*

*if (reset) qout <= 0;*

*else if (load) qout <= data;*

*else if (cin) begin*

*if(qout[3:0] == 9) begin*

*qout[3:0] <= 0;*

*if(qout[7:4] == 5) qout[7:4] <= 0;*

*else qout[7:4] <= qout[7:4]+1;*

*end*

*else qout[3:0] <= qout[3:0]+1;*

*end*

*end*

*assign cout = ((qout == 8’h59)&cin)?1:0;*

*endmodule*

### 异步计数器：

异步计数器是指构成计数器的低位计数器触发的输出作为相邻计数器的时钟，这样逐级串联起来的一类计数器，时钟信号的这种接法又叫行波计数，异步计数器的技术延迟增加，从而影响了它的范围。

*// 异步递增计数器*

*// 如果是递减计数器，则qn[i]<=q[i]; (i=0,1,2,3)*

*module yb\_cnt4 (q, clk, rst);*

*output[3:0] q;*

*input clk, rst;*

*reg[3:0] q;*

*reg[3:0] qn;*

*always @ (posedge clk)*

*begin*

*if(!rst) begin q[0] = 0; qn[0] = 1; end*

*else begin q[0] = ~q[0]; qn[0] =~q[0]; end*

*end*

*always @ (posedge qn[0])*

*begin*

*if(!rst) begin q[1] = 0; qn[1] = 1; end*

*else begin q[1] = ~q[1]; qn[1] =~q[1]; end*

*end*

*always @ (posedge qn[1])*

*begin if(!rst) begin q[2] = 0; qn[2] = 1; end*

*else begin q[2] = ~q[2]; qn[2] =~q[2]; end*

*end*

*always @ (posedge qn[2])*

*begin if(!rst) begin q[3] = 0; qn[3] = 1; end*

*else begin q[3] = ~q[3]; qn[3] =~q[3]; end*

*end*

*endmodule*

*// 最好都非阻塞赋值吧*

### 可变模计数器

可以通过模值控制端来改变计数器的模值

*module yb\_cnt16(q, clk, clr, load, m);*

*output[6:0] q;*

*input clk, clr, load;*

*input[6:0] m;*

*reg[6:0] q;*

*reg[6:0] md;*

*always @ (posedge clk)*

*begin*

***md <= m-1;***

*if(!clr) begin q <= 0; end*

*else begin if(load) begin q <= md; end*

*else begin if(q == md)*

*begin q <= 0; end*

*else begin q <= q + 1; end end*

*end*

*end*

*end*

*endmodule*

### 递增递减计数器

module updowncount(clk,load,enable,updown,R,Q);

parameter n=4;

input clk,load,enable,updown;

input [n-1:0]R;

output reg [n-1:0]Q;

always@(posedge clk)

if(load) Q<=R;

else if(enable) Q<=Q+(updown?1:-1);

endmodule

## 寄存器

是数字电路中的基本模块，许多复杂的时序逻辑电路都是由它构成的。在数字系统中，寄存器是一种在某一特定信号的控制下用于存储一组二进制数据的时序电路。通常由触发器购得寄存器，把多个D触发器的时钟端连接起来就可以构成一个存储多位二进制代码的寄存器。

*module reg8\_1 (q, d, clk, oe);*

*output[7:0] q;//数据输出*

*input[7:0] d; //数据输入*

*input oe, clk; //三态控制端，时钟信号*

*reg[7:0] q;*

*always @ (posedge clk)*

*begin*

*if(oe) begin*

*q <= 8’bz;*

*end*

*else begin*

*q <= d;*

*end*

*end*

*endmodule*

## 锁存器

是一种与寄存器类似的器件。与寄存器采用同步时钟信号控制不同，锁存器是采用**电位信号**来进行控制的。

*module reg8\_1 (q, d, g, oe);*

*output[7:0] q;//*

*input[7:0] d; //*

*input oe, g; //三态控制端，控制信号*

*reg[7:0] q;*

*always @ (\*)*

*begin*

*if(oe) begin*

*q <= 8’bz;*

*end*

*else begin*

*if(g) q <= d;*

*end*

*end*

*Endmodule*

## 触发器

### 含异步复位端的D触发器

*module dflipflop(dout,din,clk,rst);*

*output reg dout;*

*input din,clk,rst;*

*always@(negedge rst,posedge clk) begin //D触发器是只受clk影响*

*if(!rst) dout<=0;*

*esle dout<=din;*

*end*

*endmodule*

### 含同步复位端的D触发器

*Module Dflipflop(dout,din,clk,rst);*

*Output reg dout;*

*Input din,clk,rst;*

*always@(posedge clk) begin*

*If(!rst) dout<=0;*

*Esle dout<=din;*

*End*

*Endmodule*

### 2选1多路器的D触发器

*Module Dflipflop(dout,din1,din2,clk,rst);*

*Output reg dout;*

*Input din1,din2,clk,rst;*

*always@(posedge clk) begin*

*If(!rst) dout<=din1;*

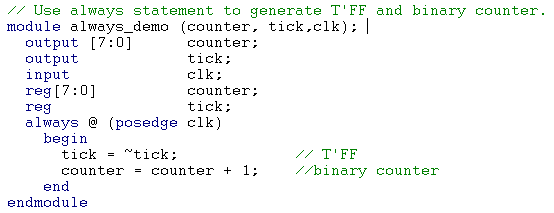
*Esle dout<=din2;*

*End*

*Endmodule*

### T’FF（T’触发器实现）

按照特征方程来写就行，



## 移位寄存器

是指寄存器里面存储的二进制数据能够在时钟信号的控制下一次左移或者右移，在数字电路中通常用于数据的串并转换、并串转换、数值运算等。包括双向移位寄存器、串入（并入）/串出（并出）移位寄存器。

### 串入串出

*module siso4 (dout, clk, din);//串入串出*

*output dout;// 串出 标量*

*input clk;*

*input din;// 串入 标量*

*reg dout;*

*reg[3:0] q;*

*always @ (posedge clk)*

*begin*

*q[0] <= din;*

*q[3:1] <= q[2:0];*

*dout <= q[3];*

*// q[0]<=din;*

*// q[1]<=q[0];*

*// q[2]<= q[1];*

*// q[3]<= q[2];*

*// dout<=q[3];*

*end*

*endmodule*

### 串入并出

*module sipo (dout, din,clr,clk);//串入并出*

*output[4:0] dout; // 并出，矢量*

*input clk, din,clr; // 串入，标量*

*reg[4:0] dout; //五位*

*always @ (posedge clk )*

*begin*

*if(clr) begin*

*dout <= 0;*

*end*

*else begin*

*dout <= {dout, din};*

*end*

*Endmodule*

### 并入串出

*module piso4 ( dout, clk,clr,din);//并入串出，有一个counter计时，串出完了再并入*

*output dout; //串出 标量*

*input clk, clr;*

*input[3:0] din; // 并入 矢量*

*reg dout;*

*reg[1:0] cnt; //初始为0，串出计数*

*reg[3:0] q;*

*always @ (posedge clk)*

*begin*

*if(clr) begin q <= 4’b0000; end*

*else begin*

*if(cnt > 0) begin*

*q[3:1] <= q[2:0];*

*cnt <= cnt -1;*

*end*

*else if(cnt == 2’b00) begin*

*q <= din; cnt<=2’b11;*

*end*

*end*

*dout <= q[3]; // NOTE\_lyq: 感觉这里不应该这样赋值，貌似也可以（可以试试assign dout=q[3]; 以实时更新*

*end*

*endmodule*

### 并入并出

就是一个赋值，😄，想多了

[n-1:0]din, [n-1:0]dout, always模块里面dout=din;

双向移位寄存器

module d\_reg (dout\_l, dout\_r, clk,

din, left\_right);

output dout\_l, dout\_r;

input clk, din, left\_right;

reg dout\_l, dout\_r;

reg[7:0] q\_tempt; //内部有八位

integer i;

always @ (posedge clk) //双向移位寄存器

begin

if(left\_right) begin

q\_tempt[7] <= din;

for (i = 7; i >= 1; i = i - 1)

begin q\_tempt[i-1] <= q\_tempt[i]; end

end

else begin

q\_tempt[0] <= din;

for (i = 1; i <= 7; i = i + 1)

begin q\_tempt[i] <= q\_tempt[i-1]; end

end

dout\_r <= q\_tempt[0];

dout\_l <= q\_tempt[7];

end

endmodule

## 分频器：

在数字电路的设计中，分频器是一种应用十分广泛的电路，其功能就是对较高频率的信号进行分频。本质上，**分频器是加法计数器的变种**，其计数值由分频系数**N=fin/fout**决定，其输出不是一般计数器的结果，而是根据分频常数对输出信号的高、低电平进行控制通常来说，分频器常用于数字电路中的时钟信号进行分频，从而得到较低频率的时钟信号、选通信号、中断信号等。

偶数分频器:指分频系数是偶数的分频器，分频系数N = 2的n次方（n=1,2,…,n）,如果输入信号的频率为f，则分频器的输出信号为：f/（2的n次方）

如2分频是指分频后得到的一个周期占了2个clk的周期，4分频是占了clk的4个周期

### 分频系数是2的整数次幂

*module div248(div2,div4,div8,clk);*

*output div2, div4, div8;*

*input clk;*

*reg div2, div4, div8;*

*reg[2:0] cnt; //初始化*

*always @ (posegdge clk)*

*begin*

*cnt <= cnt+1;*

*div2 <= cnt[0];*

*div4 <= cnt[1];*

*div8 <= cnt[2];*

*end*

*endmodule*

### 位01变化分频

*//实际应用不看占空比的分频，q直接在0和1间等时变化*

*module clkdiv (*

*input mclk, //50MHz*

*input clr,*

*output clk190,*

*output clk48*

*);*

*reg [24:0] q; // 25 位计数器*

*always @ (posedge mclk or posedge clr)*

*begin*

*if (clr == 1)*

*q <= 0;*

*else*

*q <= q + 1;*

*end*

*assign clk190 = q[17]; // 190 Hz*

*assign clk48 = q[19]; // 47.7 Hz*

*endmodule*

### 分频系数不是2 的整数次幂

*module div6(div6, clk);*

*output div6;*

*input clk;*

*reg div6;*

*reg[2:0] cnt;*

*always @ (posedge clk)*

*begin*

*if (cnt == 3’b101) begin*

*div6 <= ~ div6;*

*cnt <= 0;*

*end*

*else begin*

*cnt <= cnt + 1;*

*end*

*end*

*endmodule*

### 占空比≠1:1即1占时≠0占时

*//占空比不是1：1*

*module div6(div6, clk);*

*output div6;*

*input clk;*

*reg div6;*

*reg[2:0] cnt;*

*always @ (posedge clk)*

*begin*

*if (cnt == 3’b101) cnt <= 0;*

*else cnt <= cnt + 1;*

*end*

*always @ (posedge clk)*

*begin*

*if (cnt == 3’b000) div6 <= 0;*

*else if (cnt == 3’b010) div6<=1;*

*end*

*endmodule*