

教材：《数字逻辑基础与Verilog硬件描述语言》 贾熹滨 等 编著 2012.8. 清华大学出版社

《数字逻辑》 2015—2016 （1）总复习

复习宗旨：

基本理论、基本知识、基本分析方法、基本设计方法

复习方法：

1. 按照教材，逐章整理知识点，形成知识点网络；
2. 以典型例题，带动复习，拾遗补缺。

避免：

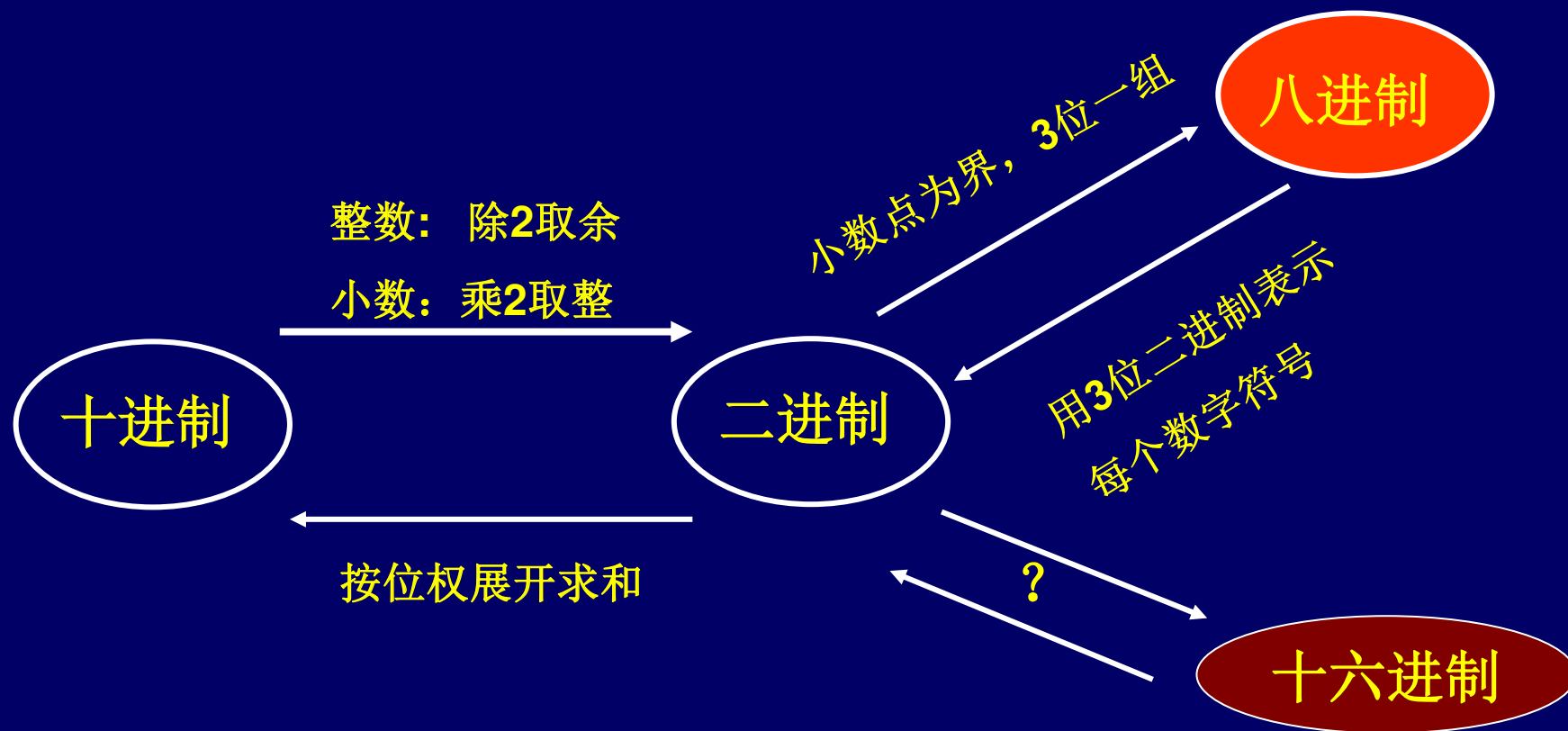
盲目做以往的试卷，不总结概念、方法。

只看以往试卷，阅读“答案”，不动手做题。

第一章 数制和码制

知识点1:

十进制、二进制、八进制、十六进制数之间的转换。



例1：完成下列数制转换。

$$(127.25)_{10} = (1111111.01)_2 = (177.2)_8 = (7F.4)_{16}$$

整数： $2^7 - 1$

小数：？

位权试凑！

$$(88.375)_{10} = (1011000.011)_2$$

整数： 11×8

$(1011)_2$ 左移三位

$$\left(\frac{13}{64}\right)_{10} = (0.001101)_2$$

$(1101)_2$ 小数点左移6位

$$(237)_8$$

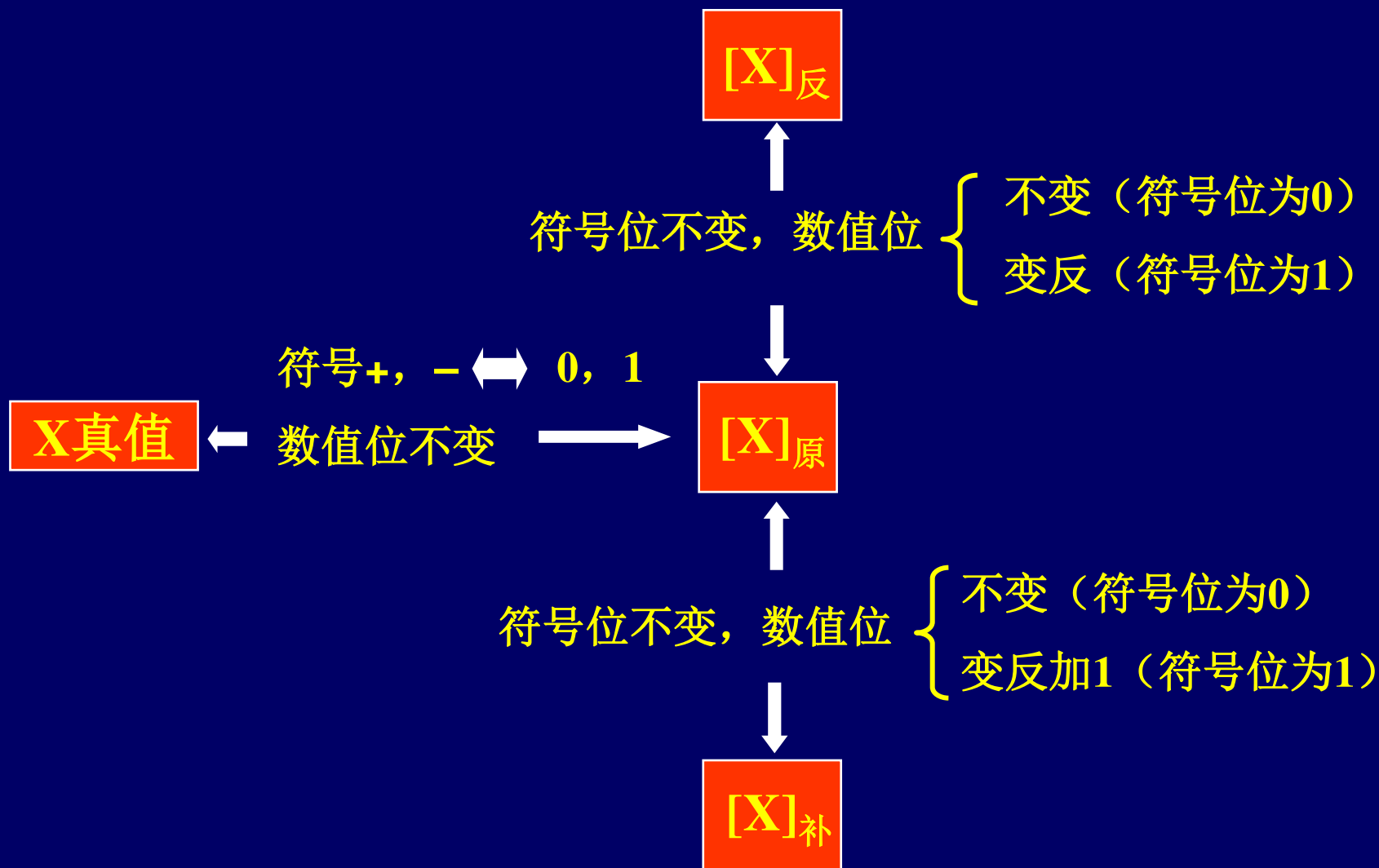
借助二进制

$$(010\ 011\ 111)_2$$

$$(9F)_{16}$$

知识点2:

真值、原码、反码、补码



例2: 已知真值 $x = -1010$, 求字长8位时, $[x]_{\text{原}}$ 、 $[x]_{\text{反}}$ 、 $[x]_{\text{补}}$ 。

$x = -0001010$

$[x]_{\text{原}} = 10001010$	$\left. \begin{array}{l} \left. \begin{array}{l} \leftarrow \\ \leftarrow \end{array} \right\} \text{数值按位变反} \\ \leftarrow \end{array} \right\} \text{加1}$	$\left. \begin{array}{l} \leftarrow \end{array} \right\} \text{数值按位变反, 加1}$
$[x]_{\text{反}} = 11110101$		
$[x]_{\text{补}} = 11110110$		

引申:

$[-x]_{\text{原}} = [-x]_{\text{反}} = [-x]_{\text{补}} = 00001010$

$[x]_{\text{补}} = 10000000, x = -10000000$

知识点3:

BCD码——8421码、余3码、2421码

前5减3，后5加3



十进制数字符号	8421码	余3码	2421码
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0010
3	0011	0110	0011
4	0100	0111	0100
5	0101	1000	1011
6	0110	1001	1100
7	0111	1010	1101
8	1000	1011	1110
9	1001	1100	1111

加 3

前5个不变，后5个加6

四位二进制	8421码	余3码	2421码
0000	0		0
0001	1		1
0010	2		2
0011	3	0	3
0100	4	1	4
0101	5	2	
0110	6	3	
0111	7	4	
1000	8	5	
1001	9	6	
1010		7	
1011		8	5
1100		9	6
1101			7
1110			8
1111			9

对应的十进制数字符号

例3：完成下列代码转换。

$$(237.69)_{10} \longrightarrow \left\{ \begin{array}{l} (0010\ 0011\ 0111\ .\ 0110\ 1001)_{8421} \\ (0010\ 0011\ 1101\ .\ 1100\ 1111)_{2421} \\ (0101\ 0110\ 1010\ .\ 1001\ 1100)_{\text{余3}} \end{array} \right.$$

注意：首尾的0不能省略！！！！

知识点4:

4位格雷码（循环码）和格雷BCD码

十进制	四位二进制 $B_3B_2B_1B_0$	循环码 $G_3G_2G_1G_0$
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

$$G_3=B_3$$

$$G_2=B_3\oplus B_2$$

$$G_1=B_2\oplus B_1$$

$$G_0=B_1\oplus B_0$$

知识点5:

奇（偶）校验码 = 信息码 + 校验位

$n + 1$ 位

n 位

1 位

奇校验约定：奇校验码中有奇数个“1”。

$$\text{奇校验位} = D_0 \oplus D_1 \oplus \dots \oplus D_n \oplus 1$$

偶校验约定：偶校验码中有偶数个“1”。

$$\text{偶校验位} = D_0 \oplus D_1 \oplus \dots \oplus D_n \oplus 0$$

例5：已知某奇校验码01110011，校验位的状态是“1”。则
信息码为0111001，信息码中“1”的个数是偶数个，

数字逻辑

第二章 小结

北京工业大学计算机学院系统结构系

彭建朝

第二章 逻辑代数基础

本章内容庞杂，应在通读的基础上，进行归纳总结。并由典型逻辑命题入手，整理知识点。

知识点

反演规则和对偶规则

应用要点：原函数不要变形；

“大”非号不变，非号下按规则；
适当添加括号。

例：已知 $F = \overline{\overline{A}B + D} \oplus C + B\overline{D}$

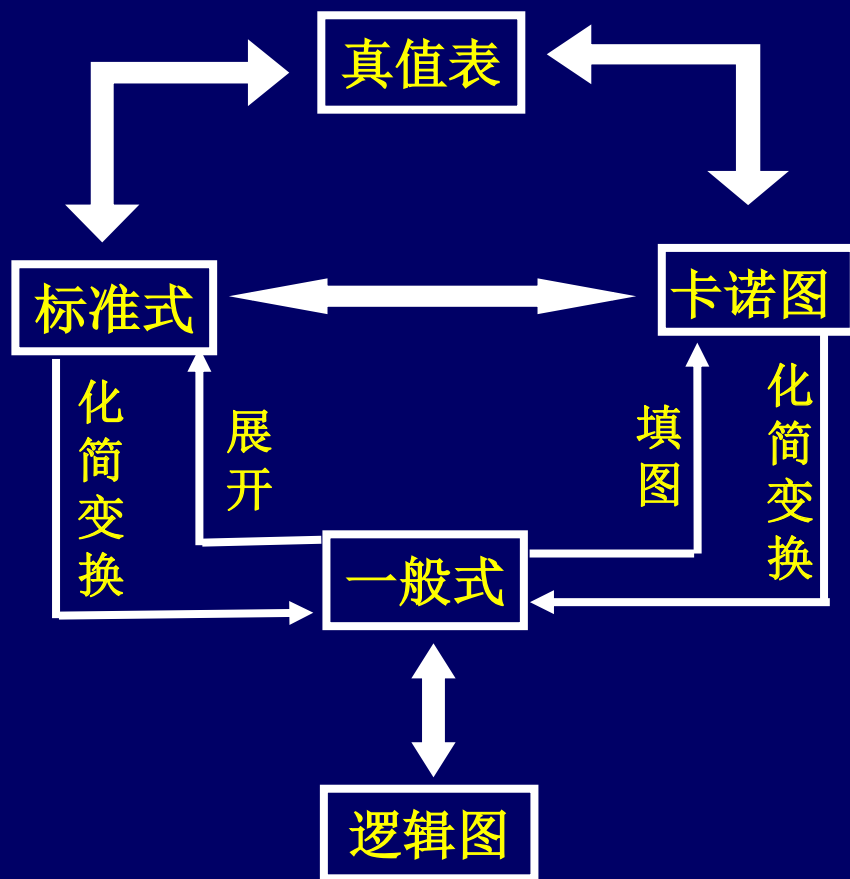
按规则求 \overline{F}, F'

$$\overline{F} = (\overline{(\overline{A} + B)\overline{D}} \odot \overline{C})(\overline{B} + \overline{D})$$

$$F' = (\overline{(A + \overline{B})D} \odot C)(B + \overline{D})$$

知识点

逻辑问题的描述可用真值表、逻辑表达式、逻辑图、卡诺图和时序图，它们各具特点又相互关联。



知识点

逻辑函数的化简

代数化简法：

卡诺图化简法：重点

常用编码的卡诺图表示
及化简（格雷码、格雷
BCD码、8421码、2421
码、余3码）



真值表
标准式
一般与或式
一般或与式
其它形式

利用无关项
进行化简的
原则：尽量
利用与尽量
不用

最简与或式
最简与非式
最简或与式
最简或非式
最简与或非式

由下列表达式，你能想到什么？

$$F = \sum m^4(0,2,7,8,10,12,14,15)$$

- (1) 四变量（设变量顺序**ABCD**）**最小项表达式**
- (2) 显性给出使**F=1**的变量取值组合；隐性给出使**F=0**的变量取值组合。
- (3) 可列函数的**真值表**
- (4) **标准与或式** $F(A,B,C,D) = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \cdots + ABCD$
- (5) **代数法化简逻辑函数**（运用公理、定理、规则）

由下列表达式，你能想到什么？

$$F = \sum m^4(0,2,7,8,10,12,14,15)$$

(6) 填卡诺图 (很重要)

设变量顺序ABCD

(7) 卡诺图化简逻辑函数

		AB			
		00	01	11	10
CD	00	1		1	1
	01				
	11		1	1	
	10	1		1	1

圈“1”，求F的最简与或式

还原律、摩根定理

最简与非式

圈“0”，求 \overline{F} 的最简与或式

取反

F最简与或非式

反演规则

F最简或与式

F最简或非式

还原律、摩根定理

由下列表达式，你能想到什么？(续)

$$F = \sum m^4(0,2,7,8,10,12,14,15)$$

(8) 根据五种最简表达式画出对应的逻辑图

(9) 根据最简与或式，判断有无逻辑险象。

$$F = \overline{B}\overline{D} + \overline{A}\overline{D} + BCD$$

代数法判断

卡诺图法判断

消除险象：增加冗余项

$$F = \overline{B}\overline{D} + \overline{A}\overline{D} + BCD + \boxed{ABC}$$

AB \ CD		00	01	11	10
00	1			1	1
01					
11			1	1	
10	1			1	1

四角

由下列表达式，你能想到什么？

$$F = \sum m^4(0,2,7,8,10,12,14,15)$$

(10) 最大项表达式

$$F = \prod M^4(1,3,4,5,6,9,11,13)$$

●显性给出使F=0的变量取值组合；隐性给出使F=1的变量取值组合。

●标准或与式

$$F(A,B,C,D) = (A+B+C+\bar{D})(A+B+\bar{C}+\bar{D}) \cdots (\bar{A}+\bar{B}+C+\bar{D})$$

●填卡诺图

由下列表达式，你能想到什么？（续）

$$F = \sum m^4(0,2,7,8,10,12,14,15)$$

(11) 反函数的最小项表达式

$$\overline{F} = \sum m^4(1,3,4,5,6,9,11,13)$$

(12) 反函数的最大项表达式

$$\overline{F} = \prod M^4(0,2,7,8,10,12,14,15)$$

知识点:

含有无关项的逻辑函数化简。

无关项使用原则：尽量使用；尽量不用。

例题：已知逻辑函数

$$F(A,B,C,D)=\sum m(3,5,6,7,13)+\sum d(2,4,12,15)$$

求最简与或式和最简或与式。

AB \ CD	00	01	11	10
00		d	d	
01		1	1	
11	1	1	d	
10	d	1		

圈“1”

$$F = \overline{A}C + BD$$

圈“0”

$$\overline{F} = \overline{B}\overline{C} + AC$$

$$F = (B + C)(\overline{A} + \overline{C})$$

谢 谢 收 看

第三章 Verilog HDL基础

知识点1:

模块的概念

模块声明

端口定义

变量类型

功能描述

module 模块名（端口列表）；

input 输入端口；

output 输出端口；

parameter 标志符常量；

wire 变量；

reg 变量；

integer 变量；

门级建模

数据流描述

行为描述

endmodule

不同位宽？

不同位宽？

知识点2:

门级描述方式

多输入门元件模型

module 模块名（端口列表）；

input 输入端口；

output 输出端口；

wire 变量；

门级元件实例语句1；

门级元件实例语句2；

.....

门级元件实例语句n；

endmodule

<元件名>（输出，输入1，.....，输入n）

**and、nand、or、nor、
xor、xnor**

多输出门元件模型

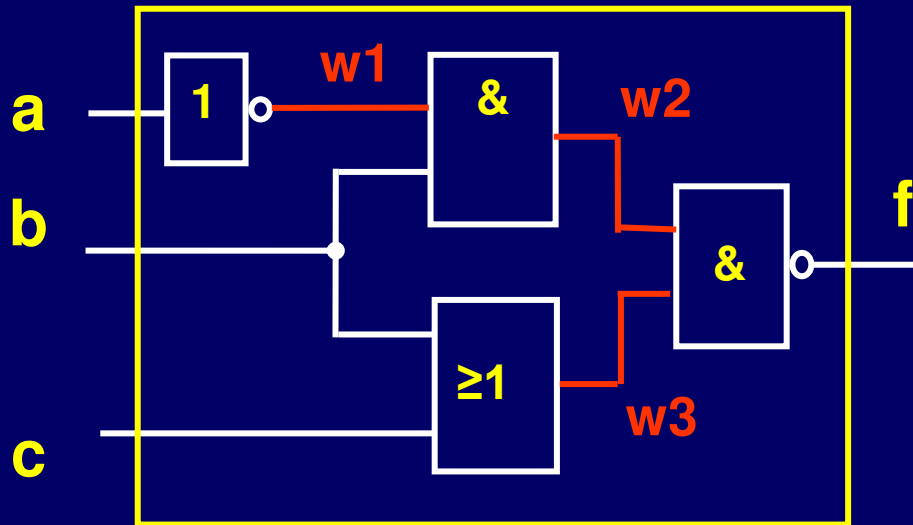
<元件名>（输出1，.....，输出n，输入）

not

门级元件实例语句的格式

<元件名> <实例名>（端口连接表）；

例：用Verilog HDL建立下列电路的门级模型。



分析：电路输入？

电路输出？

增加内部连线变量！

```
module example (a,b,c,f);
    input a,b,c;
    output f;
    wire w1,w2,w3;
    not u1 (w1,a);
    and u2 (w2,w1,b);
    or u3 (w3,b,c);
    nand u4 (f,w2,w3);
endmodule
```

知识点3:

数据流描述方式

module 模块名（端口列表）；

input 输入端口；

output 输出端口；

wire 变量；

assign <表达式1>;

assign <表达式2>;

.....

assign <表达式n>;

endmodule

运算符：

需要认真整理一下。

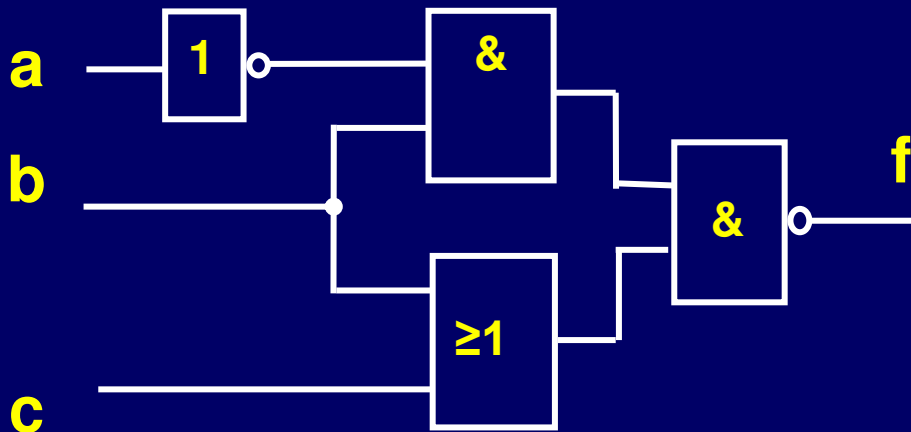
要点：

求得逻辑表达式，

运用**Verilog HDL**运算符，

用持续赋值语句**assign**描述。

例：用Verilog HDL建立下列电路的数据流模型。



$$f = \overline{ab} \bullet (b + c)$$

```
module example (a,b,c,f);  
    input a,b,c;  
    output f;  
    assign f = ~ ( ~ a & b & ( b | c ) );  
endmodule
```

知识点4:

行为描述方式

module 模块名（端口列表）；

input 输入端口；

output 输出端口；

reg 变量；

integer 变量；

parameter 标志符常量；

always @ (敏感信号列表)

begin

过程赋值语句（=、<=）

高级程序语句（if_else、case、for）

end

endmodule

要点：

所有**always**中**被赋值变量**都要定义为**reg**型或**integer**型。

integer只定义32位

多句，应使用**begin_end**

组合电路？

时序电路？

知识点4—1:

组合电路的行为描述方式

module 模块名（端口列表）；

input 输入端口；

output 输出端口；

reg 变量；

integer 变量；

parameter 标志符常量；

always @ (电平型敏感信号列表)

begin

阻塞赋值语句（=）

高级程序语句（if_else、case、for）

end

endmodule

所有影响输出的输入信号，
多个时，用**or**。

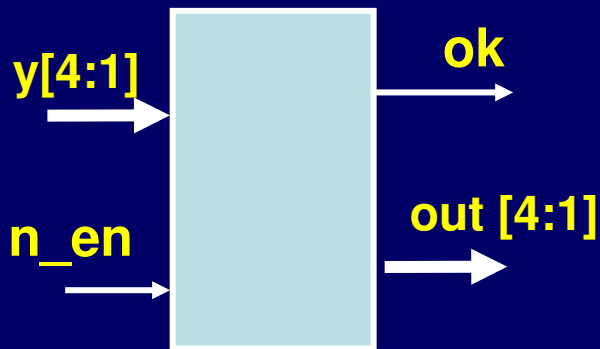
为什么？

组合电路的特点。

例：用**Verilog HDL**描述满足下列要求的代码转换电路：

- (1) 输入余**3**码，输出**8421**码；
- (2) 电路有一个低有效的使能端；
- (3) 电路有一个高有效的输出有效标志。

分析命题，
画框图，
再描述。



```
module y_8421 (y,n_en,ok,out);  
    input n_en;    input [4:1]y;  
    output ok;     output [4:1]out;  
    reg ok;        reg [4:1]out;  
    always @ ( y or n_en )  
        if (n_en==1) ok=0;  
        else begin  
            ok=1; out = y - 4'b0011;  
        end  
endmodule
```


知识点4—2:

时序电路的行为描述方式

module 模块名（端口列表）；

input 输入端口；

output 输出端口；

reg 变量；

integer 变量；

parameter 标志符常量；

always @ (边沿型敏感信号列表)

begin

非阻塞赋值语句（<=）

高级程序语句（if_else、case、for）

end

endmodule

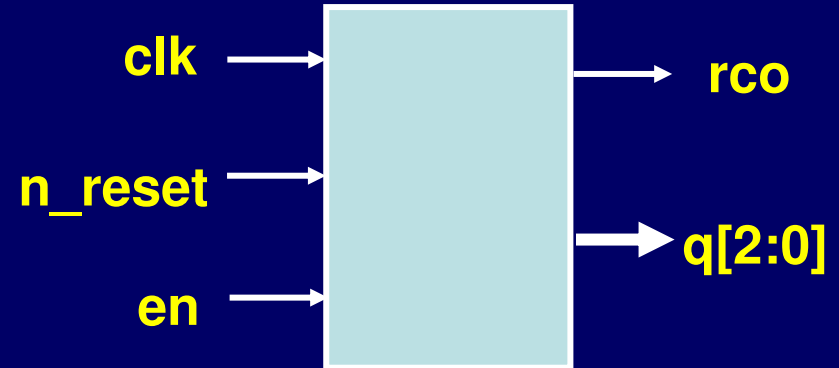
时钟信号、异步置位信号、
异步清零信号的有效沿
（posedge、negedge）。

为什么？

同时采样，
一起赋值。

例：用Verilog HDL描述一个具有下列功能的计数器：

- (1) 3位二进制减1计数；
- (2) 有一个低有效异步清零端；
- (3) 有一个高有效的计数使能端；
- (4) 具有循环借位输出。



```
module counter ( clk, n_reset, en, rco, q );
  input  clk, n_reset, en;
  output rco;
  output [2:0] q;    reg [2:0] q;
  assign rco = ( en & q==0 ) ? 1 : 0 ;
  always @ ( posedge clk or negedge n_reset )
    if ( ! n_reset ) q <= 0 ;
    else if ( en ) q <= q - 1 ;
    else q <= q ;
endmodule
```

同步清零？

知识点5:

Verilog HDL运算符

要点提示：逻辑运算符和位运算符的区别；

归约运算的特点；

拼接运算符 { }；

条件运算符 ? :

知识点6:

if_else语句和case语句

要点:

(1) 语法格式

(2) 如何避免隐含触发器

第四章 组合电路的分析与设计

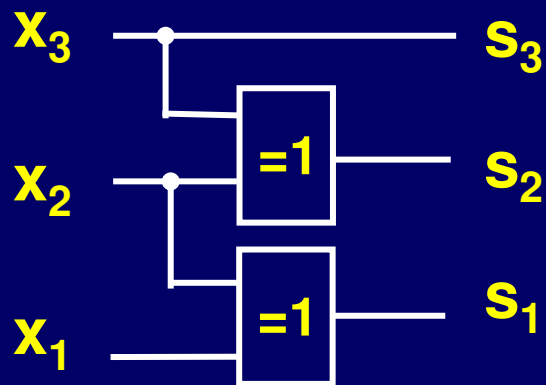
组合电路的分析

1. 基于逻辑门的组合电路分析

分析步骤:

- 列表表达式
- 作真值表
- 分析功能
- 转化为Verilog HDL模型

例：分析电路功能，建Verilog HDL模型。



$$S_3 = X_3$$

$$S_2 = X_3 \oplus X_2$$

$$S_1 = X_2 \oplus X_1$$

功能：

3位二进制码转换成3位格雷码。

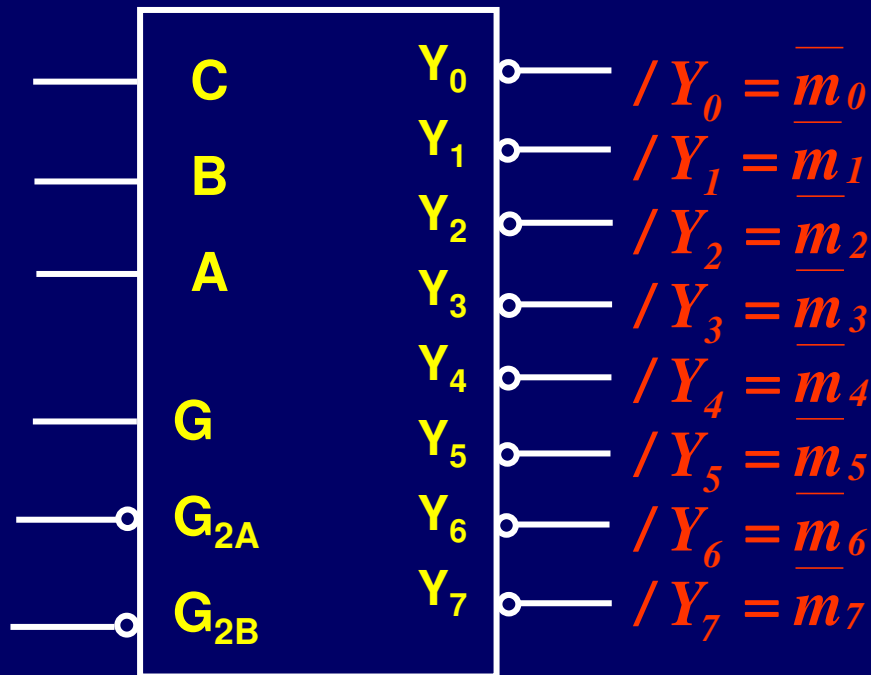
$X_3 X_2 X_1$	$S_3 S_2 S_1$
0 0 0	0 0 0
0 0 1	0 0 1
0 1 0	0 1 1
0 1 1	0 1 0
1 0 0	1 1 0
1 0 1	1 1 1
1 1 0	1 0 1
1 1 1	1 0 0

```
module B_G(x,s);
    input [3:1] x ;
    output [3:1] s ;
    assign s[3]=x[3];
    assign s[2]=x[3]^x[2];
    assign s[1]=x[2]^x[1];
endmodule
```

组合电路的分析

2. 基于译码器74LS138的电路分析

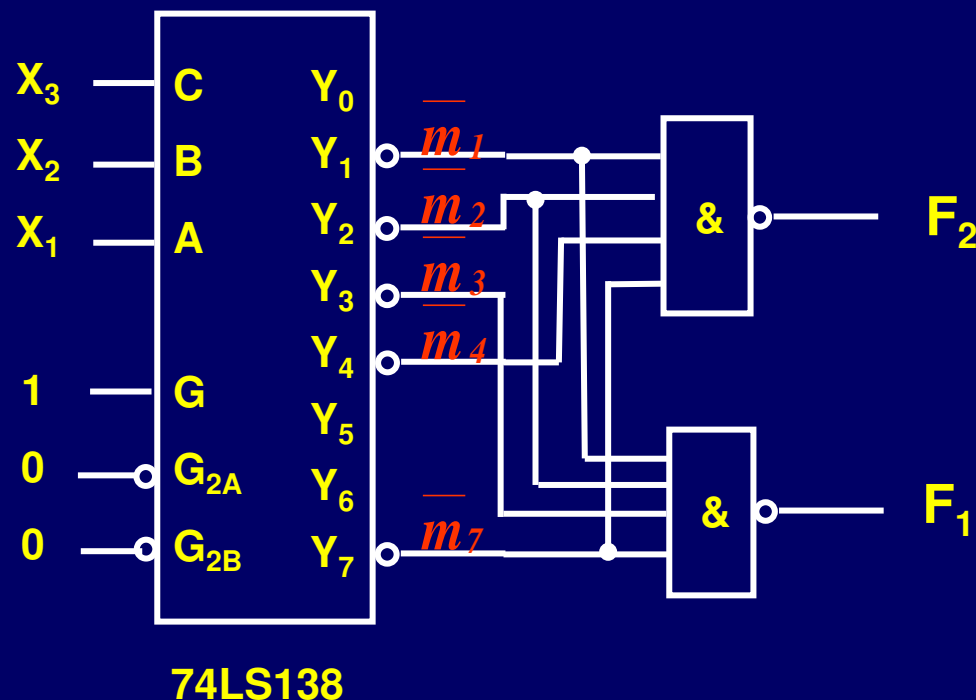
要点：熟悉74LS138的逻辑符号、逻辑功能、输出表达式。



$$/Y_i = \overline{m_i}$$

m_i 为关于
C、B、A
的最小项

例：分析电路的逻辑功能（综合考虑 F_2 和 F_1 ）。



标注译码器输出

写函数表达式

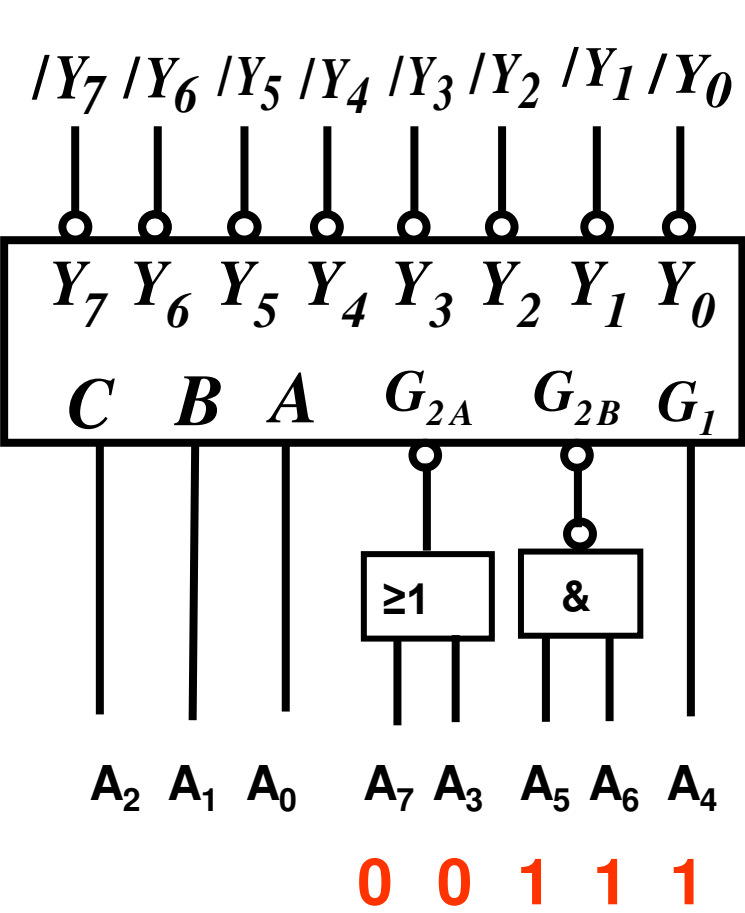
$$F_2(X_3, X_2, X_1) = \overline{m_1} \bullet \overline{m_2} \bullet \overline{m_4} \bullet \overline{m_7} \\ = m_1 + m_2 + m_4 + m_7$$

$$F_1(X_3, X_2, X_1) = \overline{m_1} \bullet \overline{m_2} \bullet \overline{m_3} \bullet \overline{m_7} \\ = m_1 + m_2 + m_3 + m_7$$

$X_3 X_2 X_1$	F_2	F_1
0 0 0	0	0
0 0 1	1	1
0 1 0	1	1
0 1 1	0	1
1 0 0	1	0
1 0 1	0	0
1 1 0	0	0
1 1 1	1	1

全加器？全减器？

例： 由74LS138译码器及逻辑门构成的组合逻辑电路如下，其中输入信号A7~A0为地址变量。试填写表格。



	$A_7A_6A_5A_4A_3A_2A_1A_0$	16进制
$\overline{Y_0}$ 有效时	0 1 1 1 0 0 0 0	7 0
$\overline{Y_1}$ 有效时	0 1 1 1 0 0 0 1	7 1
$\overline{Y_4}$ 有效时	0 1 1 1 0 1 0 0	7 4
$\overline{Y_6}$ 有效时	0 1 1 1 0 1 1 0	7 6
$\overline{Y_7}$ 有效时	0 1 1 1 0 1 1 1	7 7

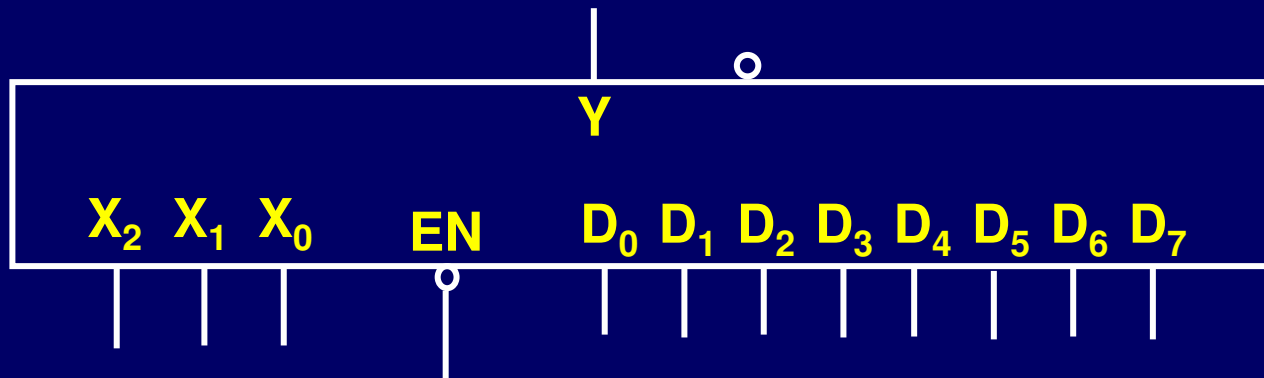
组合电路的分析

3. 基于数据选择器74LS151的电路分析

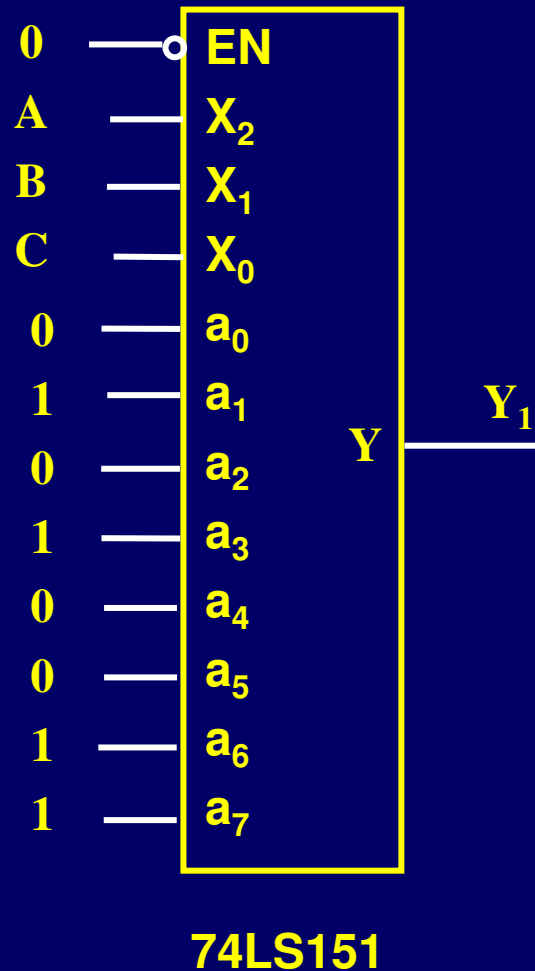
要点：熟悉74LS151的逻辑符号、逻辑功能、输出表达式。

当使能有效

$$Y = \overline{X_2} \overline{X_1} \overline{X_0} \bullet D_0 + \overline{X_2} \overline{X_1} X_0 \bullet D_1 + \cdots + X_2 X_1 X_0 \bullet D_7$$



例：分析电路，写出表达式，作卡诺图，求无险象与或式。



根据74LS151输出表达式写出 Y_1 表达式

$$Y_1 = \overline{A}\overline{B}C + \overline{A}BC + A\overline{B}\overline{C} + ABC$$

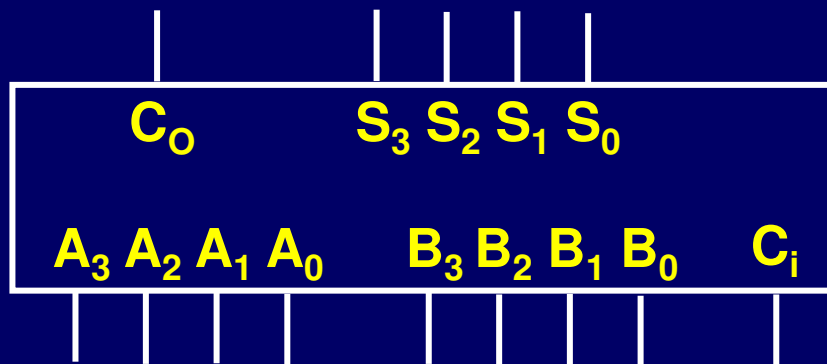
		AB			
		00	01	11	10
C	0			1	
	1	1	1	1	

$$Y_1 = AB + \overline{A}C + BC$$

组合电路的分析

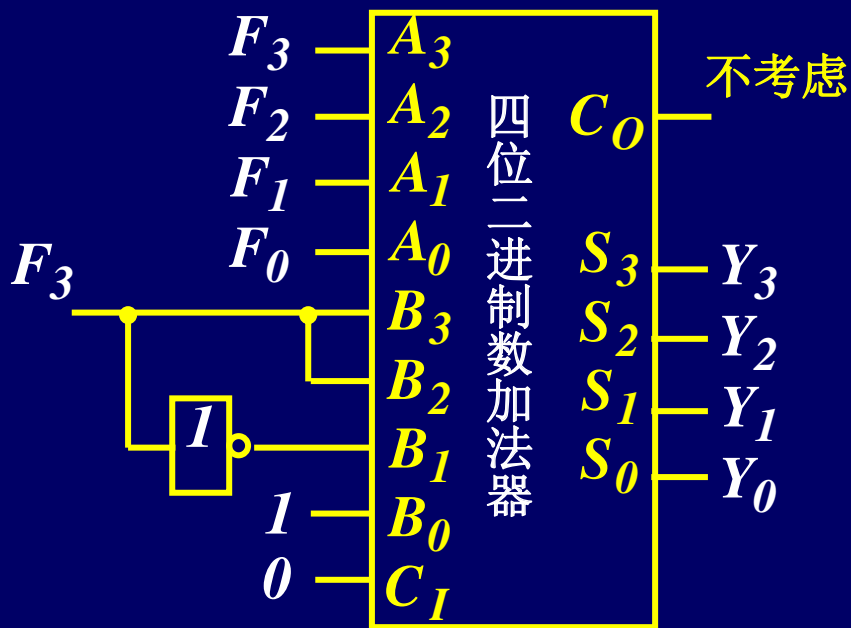
4. 基于加法器74LS283的电路分析

要点：熟悉74LS283的逻辑符号、逻辑功能。



$$\begin{array}{r} A_3 A_2 A_1 A_0 \\ B_3 B_2 B_1 B_0 \\ + \quad \quad \quad C_i \\ \hline C_o S_3 S_2 S_1 S_0 \end{array}$$

例：分析下列电路的逻辑功能。（ $F_3F_2F_1F_0$ 为2421码输入）



$F_3F_2F_1F_0$	$B_3B_2B_1B_0$	$Y_3Y_2Y_1Y_0$
0000		
0001		
0010		
0011		
0100		
1011		
1100		
1101		
1110		
1111		

功能：2421码转换为余3码。

组合电路的设计

教学要求：分析逻辑命题，建立Verilog HDL模型。

供参考的设计方法：

- (1) 建立真值表，在**always**中用**case**语句描述；
- (2) 得到函数表达式，用**assign**描述；
- (3) 归纳行为特性，在**always**中用**if_else**、**case**语句描述；

组合电路的设计举例

例1：设计一个3变量非一致电路。

A B C	F
0 0 0	0
1 1 1	0
其他	1

```
module exam_1(A,F);  
  input [3:1] A ;  
  output F ;  
  assign F=(A==0 | A==7) ? 0 : 1 ;  
endmodule
```

```
module exam_1 ( a, b, c, f ) ;  
  input a, b, c ;  
  output f ;  
  reg f ;  
  always @ ( a or b or c )  
    case ( {a,b,c} )  
      3'b000 : f = 0 ;  
      3'b111 : f = 0 ;  
      default : f = 1 ;  
    endcase  
endmodule
```

组合电路的设计举例

例2：按照下列要求，设计一个代码转换电路：

- (1) 输入8421码，输出余3码，作卡诺图；
- (2) 写出函数表达式；
- (3) 建立数据流描述模型。

CD \ AB	AB			
	00	01	11	10
00	0011	0111	d	1011
01	0100	1000	d	1100
11	0110	1010	d	d
10	0101	1001	d	d

$Y_3 Y_2 Y_1 Y_0$

$$Y_3 = A + BD + BC$$

$$Y_2 = \overline{B}D + \overline{B}C + B\overline{C}\overline{D}$$

$$Y_1 = CD + \overline{C}\overline{D}$$

$$Y_0 = \overline{D}$$

$$Y_3 = A + BD + BC$$

$$Y_2 = \overline{B}D + \overline{B}C + B\overline{C}\overline{D}$$

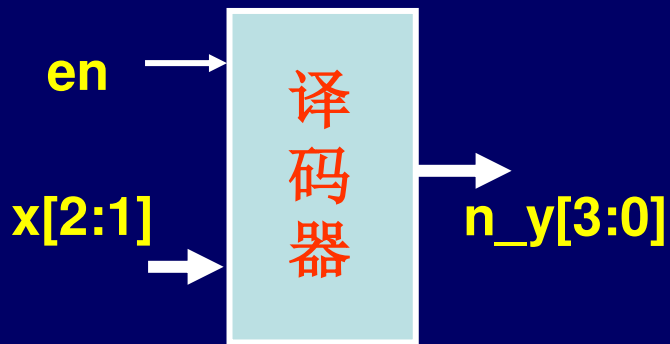
$$Y_1 = CD + \overline{C}\overline{D}$$

$$Y_0 = \overline{D}$$

```
module exam_2 (a,b,c,d,y) ;  
    input a,b,c,d;  
    output [3:0] y;  
  
    assign y[3]=a|(b&d)|(b&c) ;  
    assign y[2]=~b&d|~b&c|b&~c&~d ;  
    assign y[1]=c&d|~c&~d ;  
    assign y[0]=~d ;  
  
endmodule
```


组合电路的设计举例

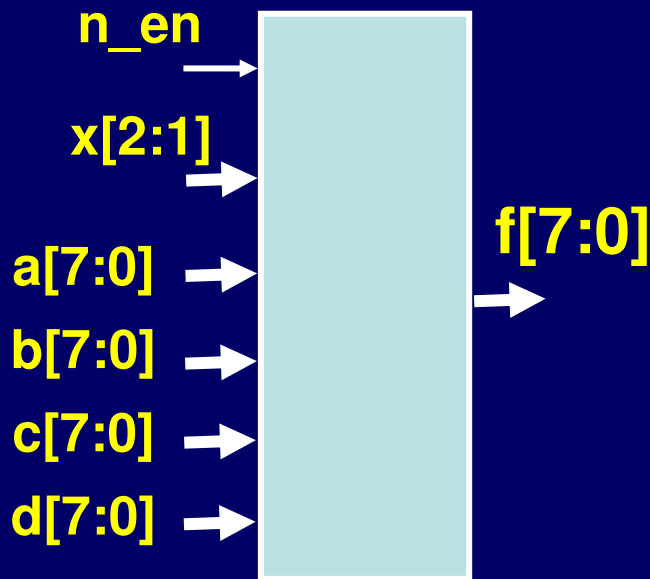
例3：设计一个高有效使能、低有效输出的2—4译码器。



```
module exam_3(en,x,n_y);  
    input en;    input [2:1]x;  
    output [3:0] n_y;  
    reg [3:0] n_y;  
    always @(en or x)  
        if (!en) n_y=4'b1111;  
        else case (x)  
            2'b00 : n_y=4'b1110;  
            2'b01 : n_y=4'b1101;  
            2'b10 : n_y=4'b1011;  
            2'b11 : n_y=4'b0111;  
        endcase  
endmodule
```

组合电路的设计举例

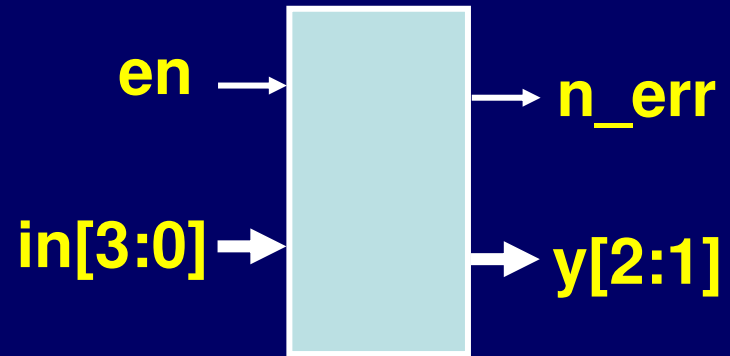
例4：设计一个低有效使能的8位四选一数据选择器。



```
module exam_4(n_en,x,a,b,c,d,f);  
    input n_en;    input [2:1] x;  
    input [7:0] a,b,c,d;  
    output [7:0] f; reg [7:0] f;  
    always @(n_en or x or a or b or c or d)  
        if ( n_en ) f = 0;  
        else case (x)  
            0 : f = a ;  
            1 : f = b ;  
            2 : f = c ;  
            3 : f = d ;  
        endcase  
endmodule
```

组合电路的设计举例

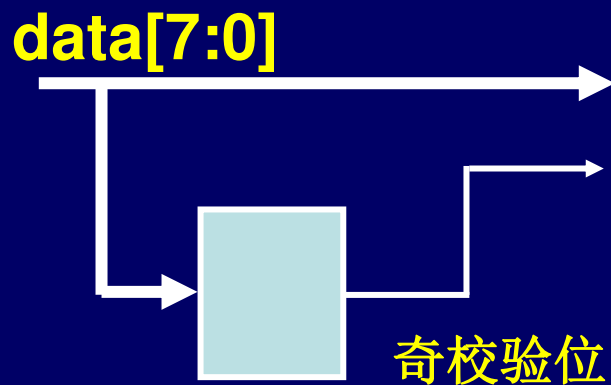
例5：设计一个高有效使能、
具有编码输出有效标志
的4—2优先权编码器。



```
module exam_5 ( en, in, n_err, y ) ;
    input en;  input [3:0] in;
    output n_err;  output [2:1] y;
    reg n_err;  reg [2:1] y;
    always @(en or in)
        if ( !en )  n_err=0;
        else if ( in[3] )  {n_err,y}=3'b1_11;
            else if ( in[2] )  {n_err,y}=3'b1_10;
                else if ( in[1] )  {n_err,y}=3'b1_01;
                    else if ( in[0] )  {n_err,y}=3'b1_00;
                        else  n_err=0 ;
endmodule
```

组合电路的设计举例

例6：设计一个8为数据并行传输时，满足奇校验要求的奇校验位发生器。



```
module exam_6(data, o_bit);  
    input [7:0] data ;  
    output o_bit ;  
    assign o_bit = ! (^data) ;  
endmodule
```

第五~八章

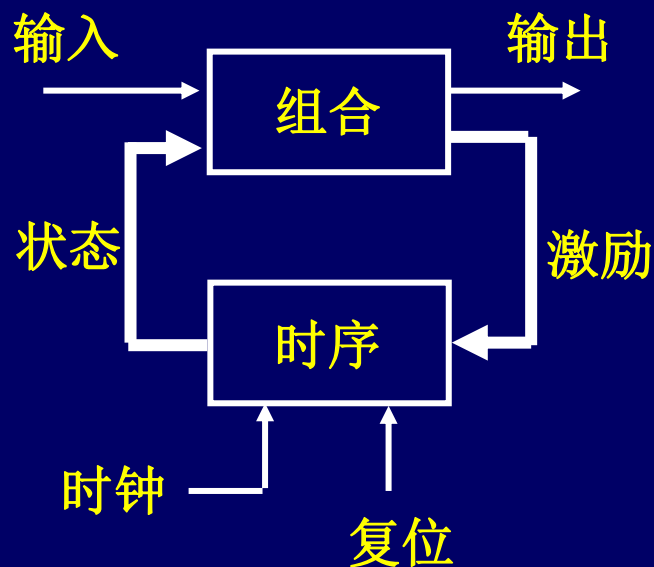
同步时序电路

应对5~8章的内容进行整理、归纳，形成知识网络，掌握基本概念、分析方法和设计要点。

知识点1:

同步时序电路的两种模型

Mealy型



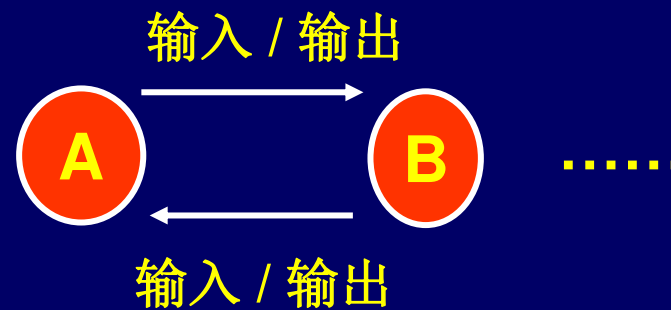
状态 { 时钟作用前, 现态;
时钟作用后, 次态;

$$\text{输出} = f(\text{现态}, \text{输入})$$

$$\text{激励} = f(\text{现态}, \text{输入})$$

$$\begin{aligned} \text{次态} &= f(\text{激励}) \\ &= f(\text{现态}, \text{输入}) \end{aligned}$$

Mealy型状态图

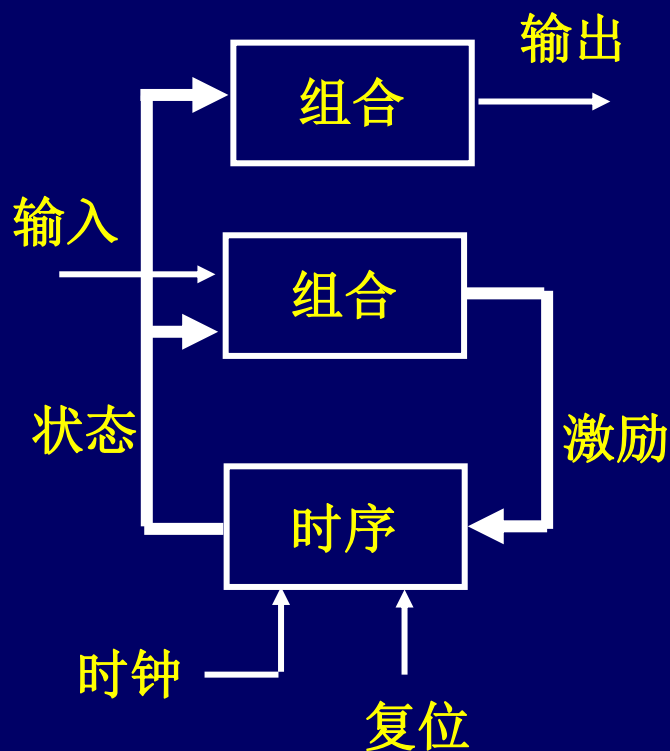


Mealy型状态表

现态Q(t) \ 输入X	0	1
A	B / 0	D / 1
B	C / 0	A / 0
C	D / 0	B / 0
D	A / 1	C / 0

次态Q(t+1) / 输出Z

Moore型

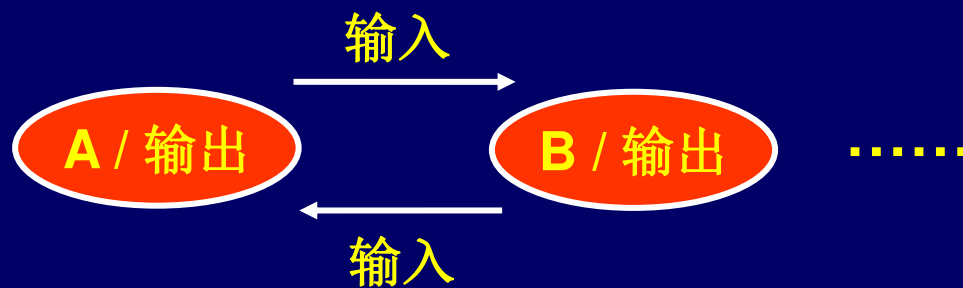


输出 = f (现态)

激励 = f (现态, 输入)

次态 = f (激励)
= f (现态, 输入)

Moore型状态图



Moore型状态表

现态Q(t) \ 输入X	输入X		输出Z
	0	1	
A	B	D	0
B	C	A	0
C	D	B	0
D	A	C	1

次态Q(t+1)

共用表格——次态真值表

输入 现态	次态	输出
所有组合		

共用表格——激励/状态转换表

输入 现态	激励	次态	输出
所有组合			

共用表格——状态转换表

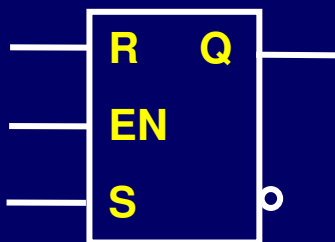
现态		次态
初态	→	次态1
现态1	←	次态2
现态2	←

知识点2:

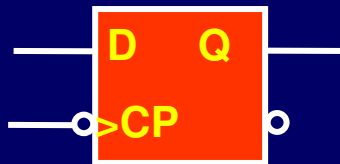
锁存器和触发器

锁存器——在使能信号有效期间，输入影响输出。

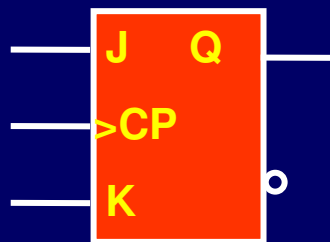
触发器——在时钟信号的有效沿，完成状态的改变，其它时刻，输入不影响输出。



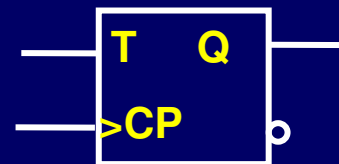
RS锁存器



下降沿D
触发器



上升沿JK
触发器



上升沿T
触发器

$$\begin{cases} Q_{t+1} = S + \bar{R} \bullet Q_t \\ R \bullet S = 0 \end{cases}$$

$$Q_{t+1} = D$$

$$Q_{t+1} = J\bar{Q}_t + \bar{K}Q_t$$

$$Q_{t+1} = T \oplus Q_t$$

R	S	Q_{t+1}
0	0	Q_t
0	1	1
1	0	0
1	1	d

D	Q_{t+1}
0	0
1	1

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	\bar{Q}_t

T	Q_{t+1}
0	Q_t
1	\bar{Q}_t

知识点3:

时序波形图

画波形图要点:

- (1) 画时钟，标定有效沿；
- (2) 画输入，有效沿保持稳定；
- (3) 分段画出状态波形；
- (4) 最后画输出波形。

以 $Q_{t+1}=D$ 、 $Z=D\oplus Q_t$ 为例。（上升沿有效，初态为0）

知识点4:

基于D、JK触发器的同步时序电路分析

分析方法:

- (1) 写出触发器输入端的激励函数表达式和电路的输出函数表达式;
- (2) 代入触发器的特性方程, 得到电路的状态方程 (组);
- (3) 填写卡诺图形式的状态 / 输出表; (**Mealy? Moore?**)
- (4) 画状态图; (**Mealy? Moore?**)
- (5) 分析电路的功能;
- (6) 根据给定输入序列和电路初态, 画时序图;
- (7) 根据电路功能建立**Verilog HDL**模型。

应根据题目的要求, 灵活运用!

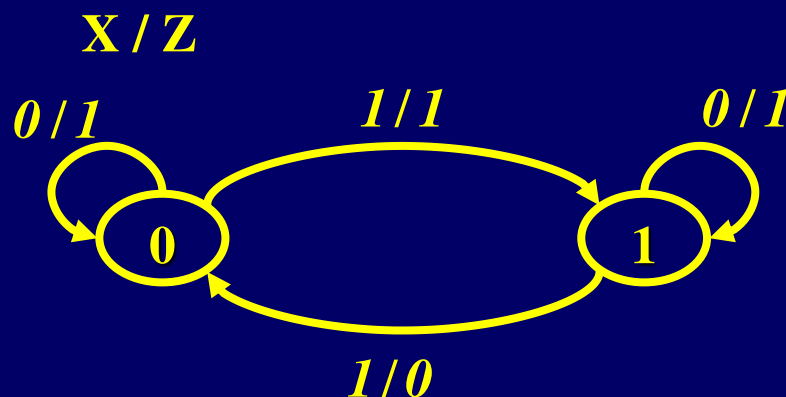
基于D、JK触发器的同步时序电路分析举例

例1：根据给定状态/输出表，画出状态图和初态为0且输入为10110的时序图。

$X \backslash Q$	0	1
0	0 / 1	1 / 1
1	1 / 1	0 / 0

$Q_{(t+1)} / Z_{(t)}$

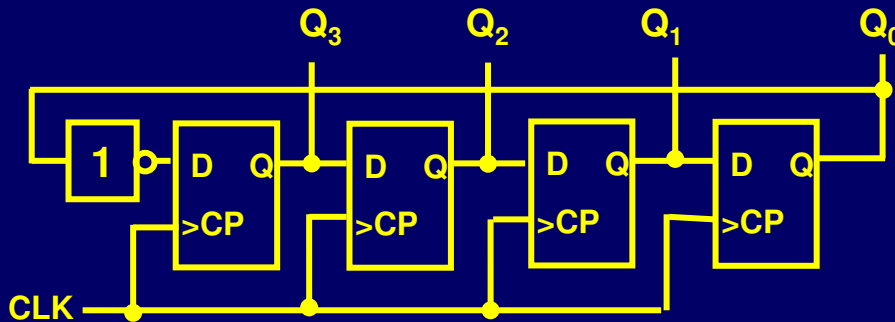
状态图



在黑板上画出波形图

基于D、JK触发器的同步时序电路分析举例

例2：填写下列同步时序电路的初态为0000时的状态转换表。



$Q_3 \sim Q_0(t)$	$Q_3 \sim Q_0(t+1)$
0000	

引申：

电路的功能？

自启动特性？

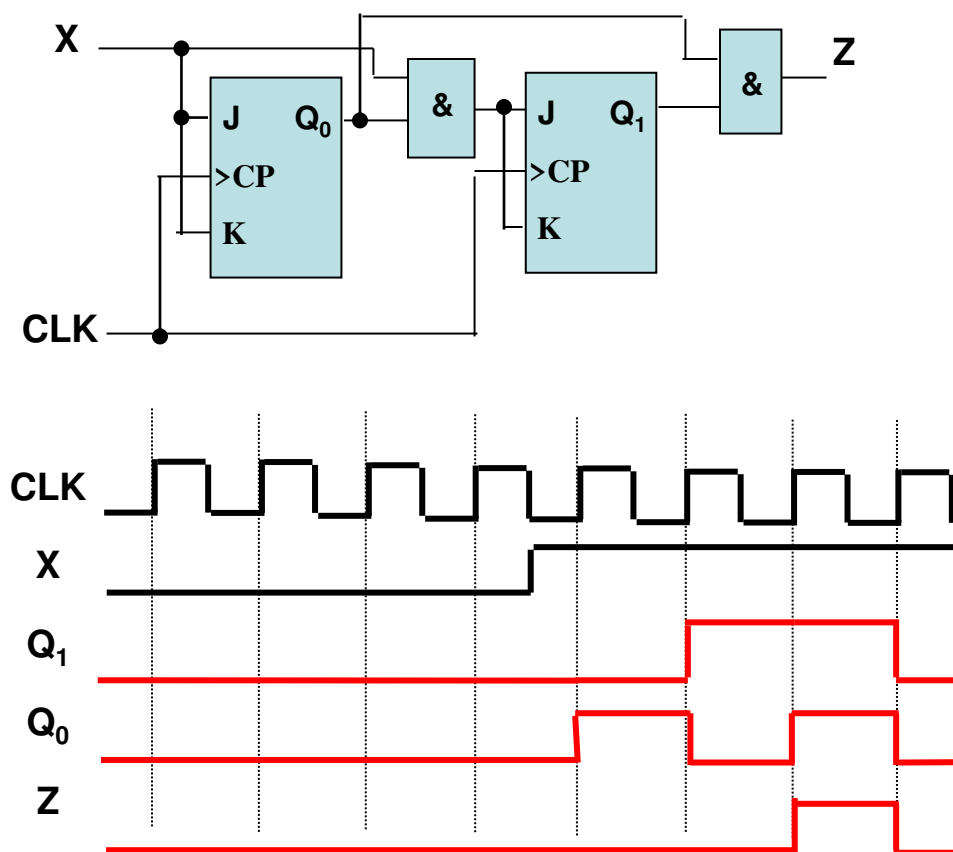
波形图？

现态1100，经3个脉冲，电路状态？

基于D、JK触发器的同步时序电路分析举例

例3：分析图示同步时序电路。

- (1) 写出激励方程和输出方程；
- (2) 作激励 / 状态转换表；
- (3) 画初态 $Q_1Q_0=00$ 时，输入x为00001111时， Q_1 、 Q_0 、Z的波形图。
- (4) 说明电路功能。



$$J_0=K_0=X \quad J_1=K_1=XQ_0 \quad Z=Q_1Q_0$$

X	Q_1	Q_0	J_1	K_1	J_0	K_0	$Q_{1(t+1)}$	$Q_{0(t+1)}$	Z
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	0
0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	0	1	0
1	0	1	1	1	1	1	1	0	0
1	1	0	0	0	1	1	1	1	0
1	1	1	1	1	1	1	0	0	1

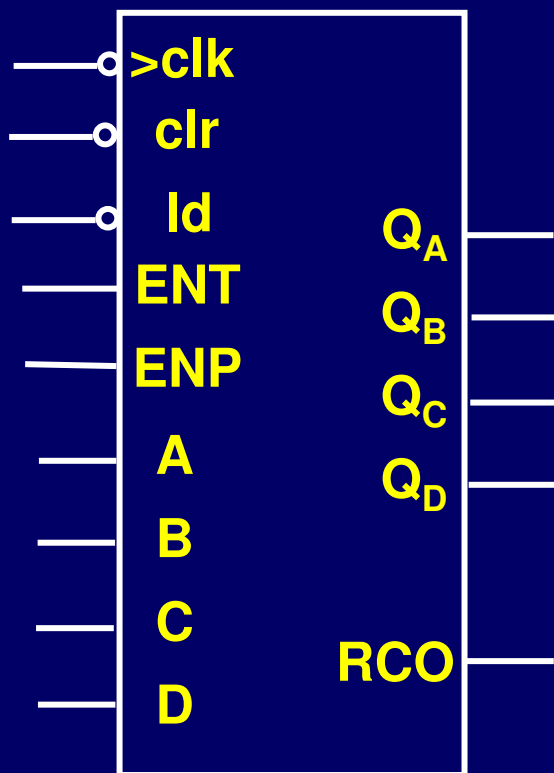
功能： $X=0$ ，停止计数；
 $X=1$ ，模4加1计数，Z为循环进位。

知识点5

基于74LS163和74LS194的同步时序电路分析

要点：熟悉逻辑符号、逻辑功能，掌握分析方法。

74LS163



逻辑功能：

同步清零；

同步置数；

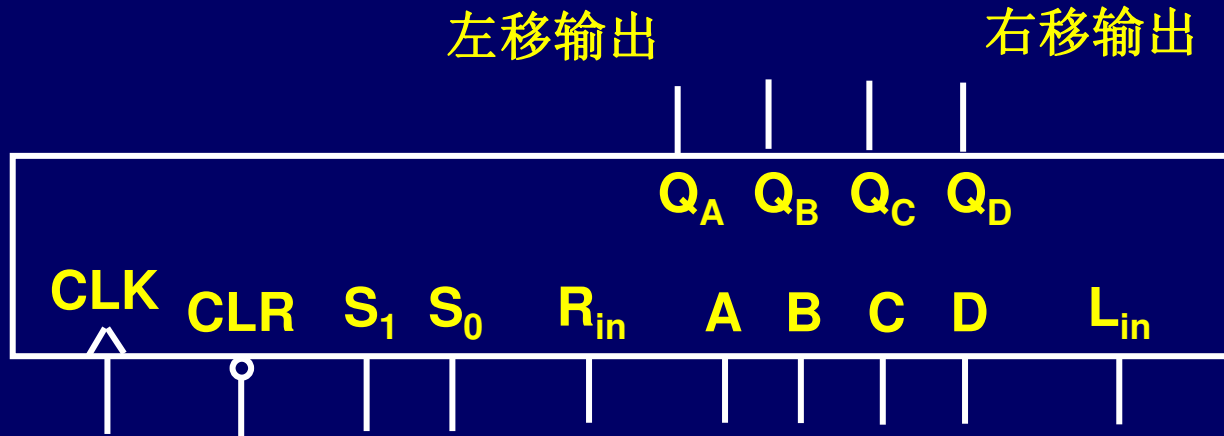
加1计数；

保持。

RCO=1的条件？

$Q_D Q_C Q_B Q_A = 1111$ & $ENT = 1$

74LS194



异步
清零

0	0	保持
0	1	右移
1	0	左移
1	1	置数

基于74LS163和74LS194的同步时序电路分析

设定电路初态（现态）；

计算反馈；

查功能表；

获得次态；



次态作为现态

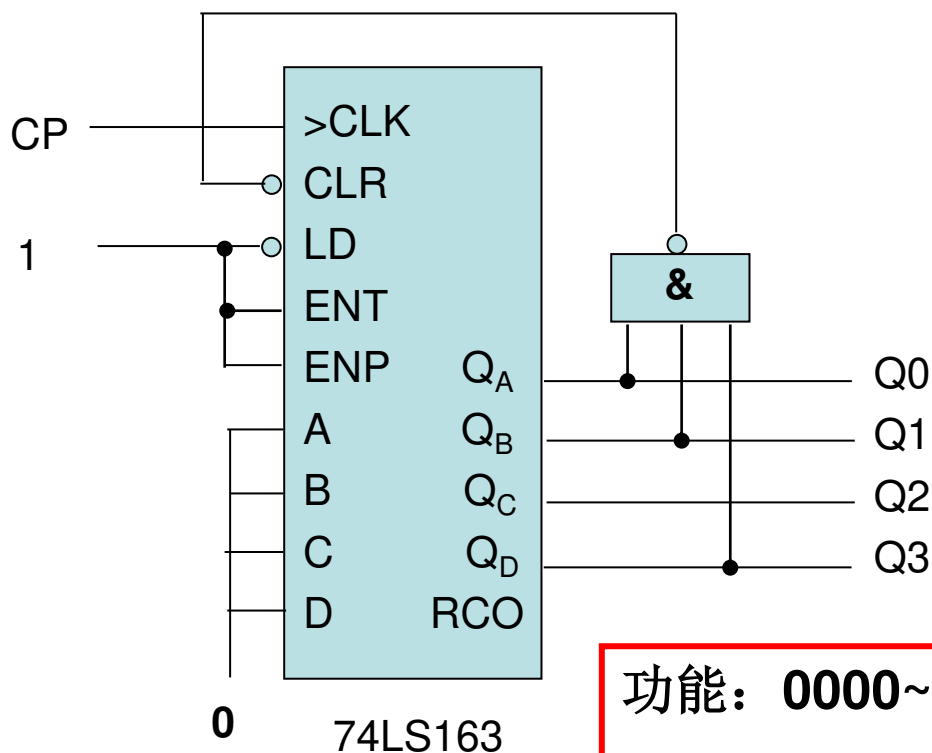
直到出现状态循环

基于74LS163和74LS194的同步时序电路分析举例

例1. 分析74LS163构成的电路功能。

(1) 画出上电清0后，电路的状态转换序列；

(2) 说明电路功能。



功能：0000~1011

模12加1计数器。

$Q_D Q_C Q_B Q_A$

0000

0001

0010

0011

0100

0101

0110

0111

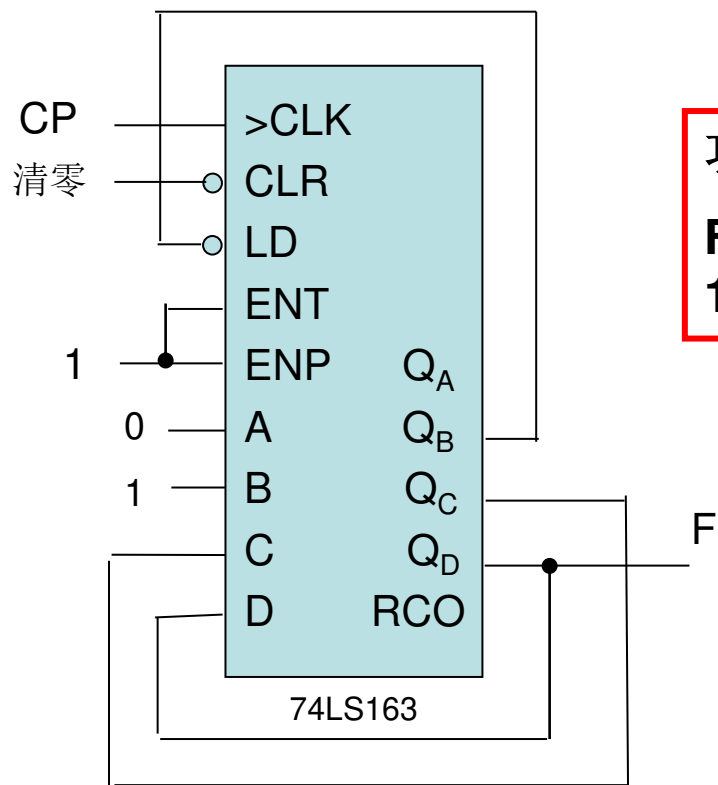
1000

1001

1010

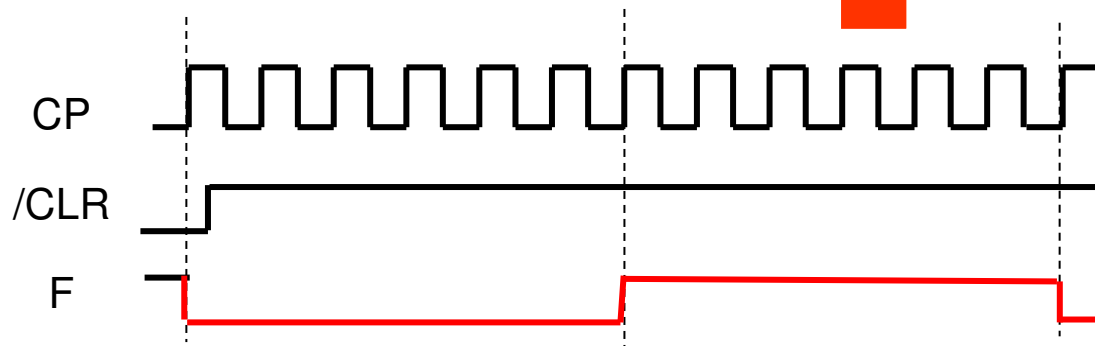
1011

例2. 分析图示电路的逻辑功能，并画出F的波形图。

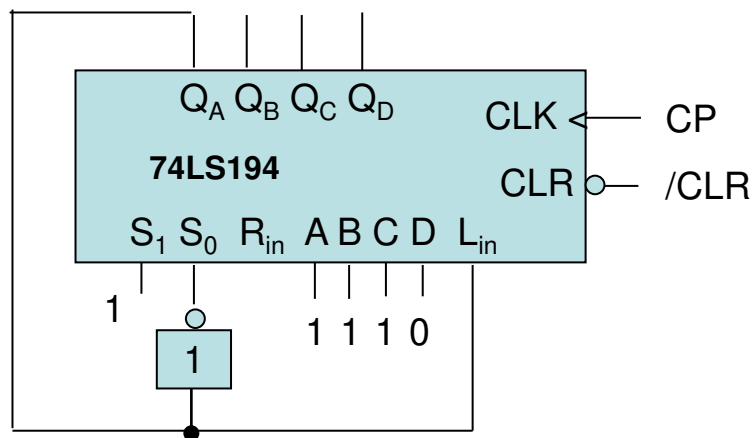


功能：
F为CP的
12分频。

Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	1	0
0	0	1	1
0	1	0	0
0	1	1	0
0	1	1	1
1	0	0	0
1	0	1	0
1	0	1	1
1	1	0	0
1	1	1	0
1	1	1	1



例3： 分析图示电路，写出启动清零后电路的状态转换序列，说明功能。



顺序很重要

功能：四位左移
循环一个0

Q _A	Q _B	Q _C	Q _D	
0	0	0	0	置数
1	1	1	0	左移
1	1	0	1	左移
1	0	1	1	左移
0	1	1	1	左移

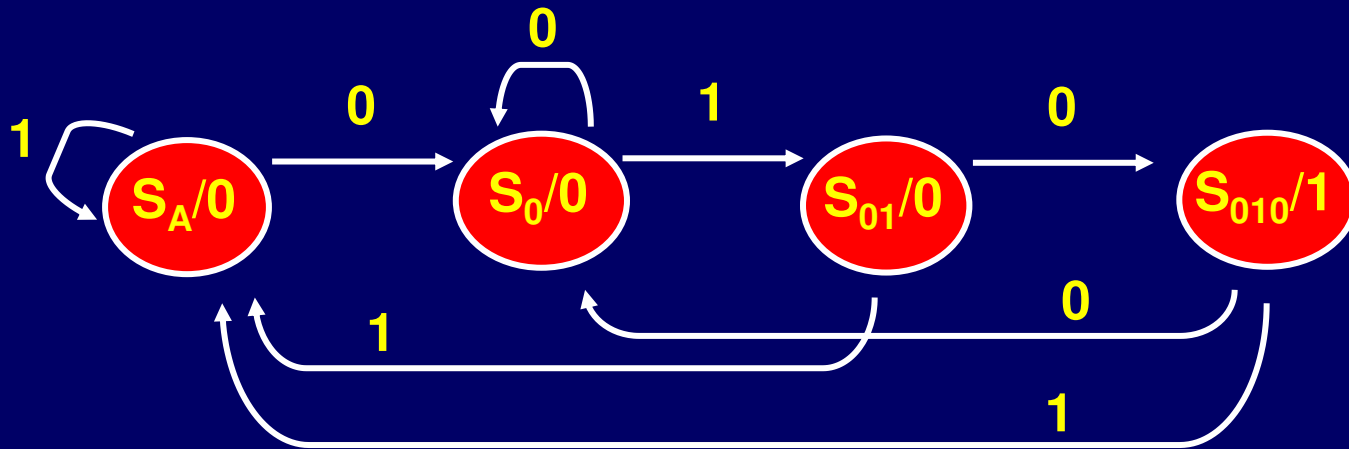
知识点6

序列检测器的原始状态图和原始状态表的建立

方法:

- (1) Moore型? Mealy型?
- (2) 设定一个初态;
- (3) 先画出主序列; 可重? 不可重!
- (4) 再画出辅助序列;
- (5) 建立原始状态表。

例：画出“010”（不可重）序列检测器的Moore型原始状态图 and 原始状态表。



X		Z		
		0	1	z
S	S _A	S ₀	S _A	0
	S ₀	S ₀	S ₀₁	0
	S ₀₁	S ₀₁₀	S _A	0
	S ₀₁₀	S ₀	S _A	1
		S _{t+1}		

知识点7

原始状态表的化简（隐含表法）

涉及的概念：等效对、等效类、最大等效类

隐含表的构成：梯形表，“缺头少尾”

化简过程：顺序比较、关联比较；

确定等效对；

根据等效的传递性，确定等效类；

确定最大等效类；

将每个最大等效类命名一个新状态，形成最小状态表。

状态等效的判定原则：

（1）输入相同时的输出相同；

（2）次态满足下列条件之一：次态相同；次态交错或维持；
后继状态等效；次态循环。

例：化简给定原始状态表。

$X \backslash Y$	0	1
A	C / 1	B / 0
B	C / 1	E / 0
C	B / 1	E / 0
D	D / 1	B / 1
E	D / 1	B / 1

$Y_{(t+1)} / Z$

B	BE			
C	BC BE	√		
D	X	X	X	
E	X	X	X	√
	A	B	C	D

$X \backslash S$	0	1
S_1	$S_2 / 1$	$S_2 / 0$
S_2	$S_2 / 1$	$S_3 / 0$
S_3	$S_3 / 1$	$S_2 / 1$

$S_{(t+1)} / Z$

等效对： (B, C) , (D, E)

最大等效类： {A}, {B, C}, {D, E}

令 $S_1 = \{A\}$, $S_2 = \{B, C\}$, $S_3 = \{D, E\}$

知识点8

同步时序电路的设计

考核基于**Verilog HDL**的建模方法。

重点内容：

触发器、计数器、寄存器、移位寄存器和序列检测器。

关键步骤：

- (1) 分析逻辑命题的要求，确定逻辑功能；
- (2) 画出逻辑框图（或状态转换图）；
- (3) 用**Verilog HDL**建模。

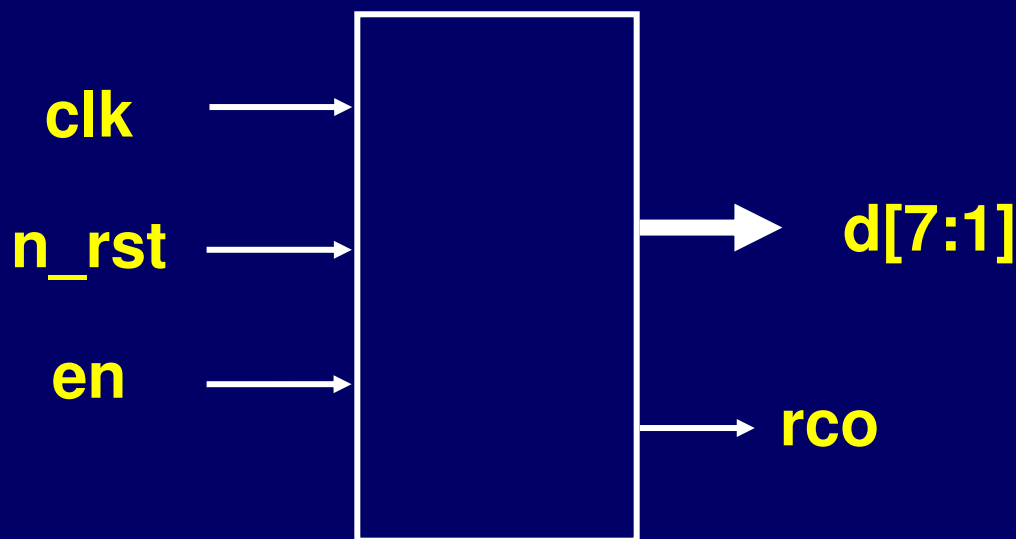
例1： 建立一个具有异步清零、异步置位功能的上升沿
JK触发器的Verilog HDL模型。

```
module m_JK ( clk, rst, set, j, k ,q ) ;  
    input clk, rst, set, j, k ;  
    output q ;  
    reg q ;  
    always @ ( posedge clk or negedge rst or negedge set )  
        if ( ~rst ) q<=0 ;  
        else if ( set==0 ) q<=1 ;  
        else case ( { j, k } )  
            0 : q<=q ;  
            1 : q<=0 ;  
            2 : q<=1 ;  
            3 : q<=~q ;  
        endcase  
endmodule
```

若同步清零、置位？

例2： 用**Verilog HDL**描述一个满足下列要求的计数器：

- (1)** 下降沿 $(0\sim99)_{10}$ 加1计数；
- (2)** 电路具有一个低有效的异步清零端；
- (3)** 电路具有一个高有效的计数使能端；
- (4)** 电路具有一个高有效的循环进位 (**RCO**) 输出端。



设计框图

```
module M_100 ( clk, n_rst, en, d, rco ) ;
```

```
    input  clk, n_rst, en ;
```

```
    output [7:1] d ;
```

```
    output rco ;
```

```
    reg [7:1] d ;
```

```
    assign rco = ( d==99 & en==1 ) ? 1 : 0 ; // 进位
```

```
    always @ ( posedge clk or negedge n_rst )
```

```
        if ( n_rst==0 ) d<=0 ;                // 异步清零
```

```
        else if ( en==0 ) d<=d ;              // 计数保持
```

```
            else if ( d==99 ) d<=0 ;          // 模控制
```

```
                else d<d+1 ;                  // 加1计数
```

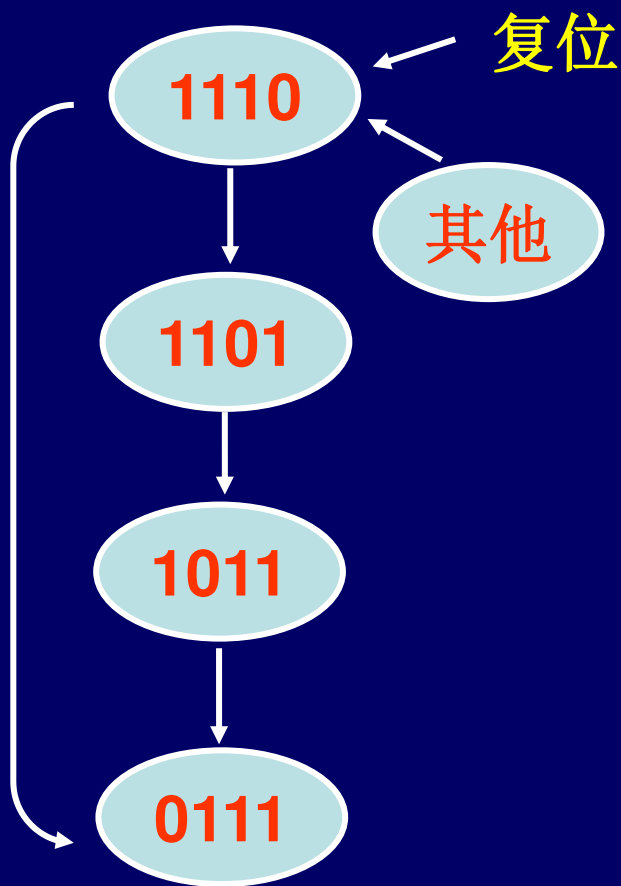
```
endmodule
```

思考：减1计数？

双向可逆计数？

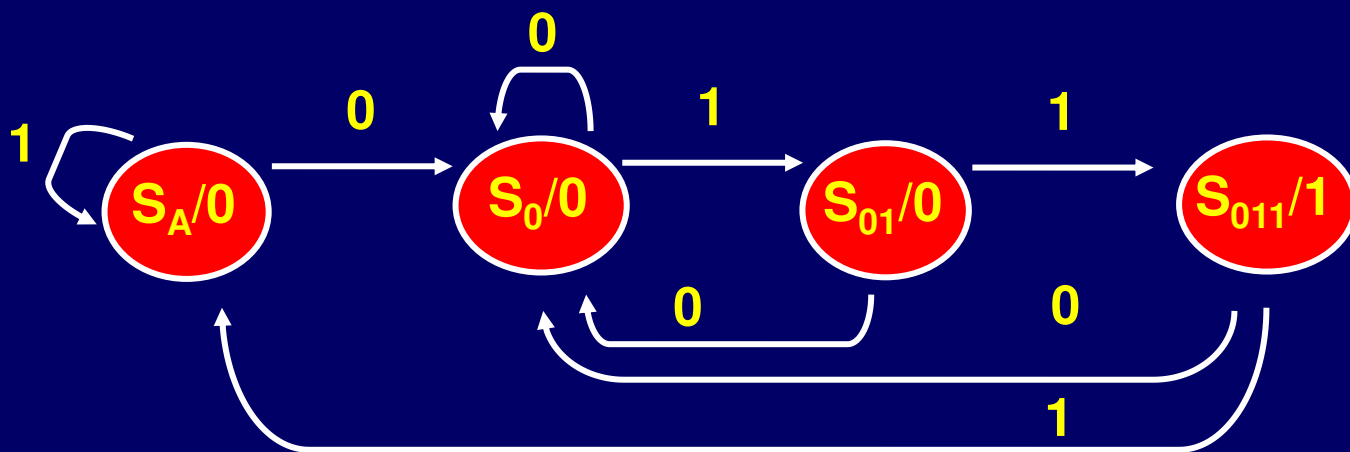
5~99加1计数？

例3： 用**Verilog HDL**描述一个左移循环一个“0”的4位环形计数器。
要求先画出能自启动的状态图，再进行描述。



```
module X_0 ( clk, rst, q );  
    input clk, rst ;  
    output [4:1] q ;  
    reg [4:1] q ;  
    always @ ( posedge clk )  
        if ( rst ) q<=4'b1110 ;  
        else case ( q )  
            4'b1110 : q<=4'b1101 ;  
            4'b1101 : q<=4'b1011 ;  
            4'b1011 : q<=4'b0111 ;  
            4'b0111 : q<=4'b1110 ;  
            default : q<=4'b1110 ;  
        endcase  
    endmodule
```

例4：画出“011”序列检测器的Moore型原始状态图，再用Verilog HDL建模。



X \ S		X		
		0	1	z
S	SA	S ₀	S _A	0
	S ₀	S ₀	S ₀₁	0
	S ₀₁	S ₀	S ₀₁₁	0
	S ₀₁₁	S ₀	S _A	1
		S _{t+1}		

		X		
		0	1	z
S	S _A	S ₀	S _A	0
	S ₀	S ₀	S ₀₁	0
	S ₀₁	S ₀	S ₀₁₁	0
	S ₀₁₁	S ₀	S _A	1

```

module M_011 (clk, rst, x, z);
  parameter
    SA=2'b00,  S0=2'b01,
    S01=2'b10, S011=2'b11;
  input clk, rst, x ;
  output z ;
  reg z ;
  reg [2:1] now, next ;
  always @ ( posedge clk )
    if ( rst ) now<=SA;
    else now<=next ;

```

always @ (now or x)

case (now)

SA : if (x==0) next =S0 ; else next=SA ;

S0 : if (x==0) next =S0 ; else next=S01 ;

S01 : if (x==0) next =S0 ; else next=S011 ;

S011 : if (x==0) next =S0 ; else next=SA ;

default : next=SA ;

endcase

always @ (now)

if (now==S011) z=1 ;

else z=0 ;

endmodule

答疑时间：

？ 月 ？ 日，每天9：00至17：00

答疑地点： 信西303

信南308（西）

祝同学们取得好成绩！！！！