**数字逻辑作业**

**计科卓越班 20164500 赵一霖**

**题目：整理以下器件的基本工作原理、用途、对应的Verilog代码（对于有位数的，可以以4位的为具体例子，但要考虑扩展问题）。  
1、编码器  
2、译码器  
3、BCD七段数码管字符译码器  
4、数据分配器（DEMUX）  
5、数据选择器（MUX）  
6、数值比较器  
7、加法器**

# 常见器件总结

## 编码器（Encoder）

**分类：普通编码器和优先编码器**

### 普通编码器（Common Encoder）

1. **特点**

**任何时刻只允许输入一个编码信号，否则输出将发生混乱。**

1. **基本工作原理**

**对于n位二进制编码器为例，其最多对应有2n个输入，每个输入的各位最多只有一位为1而其余均为0。根据真值表我们可以将每种符合条件的输入信号转换为对应的特定输出从而将其转换为n位二进制编码，而其余的输入信号无效，为“约束项”。**

1. **逻辑功能（用途）**

**把输入的每一个高、低电平信号变成一个对应的二进制编码。**

1. **Verilog代码**

*1 module encoder8\_3(a,b,c,d,e,f,g,h,out1,out2,out0,none\_on);*

*2 input a,b,c,f,d,e,g,h;*

*3 output out0,out1,out2,none\_on;*

*4 wire [3:0]outvec;*

*5 assign outvec = h?4'b0111:g?4'b0110:f?4'b0101:e?4'b0100:d?4'b0011:c?4'b0010:b?4'b0001:a?4'b0000:4'b1000;*

*6 assign out0=outvec[0];*

*7 assign out1=outvec[1];*

*8 assign out2=outvec[2];*

*9 assign none\_on=outvec[3];*

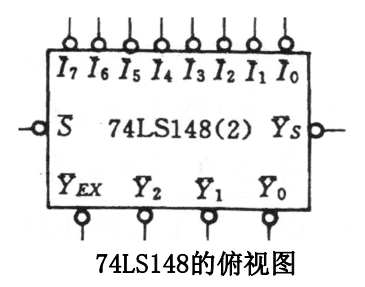
*10 endmodule*

1. **优先编码器（Priority Encoder）**
2. **特点**

**允许同时输人两个以上的编码信号，不过在设计优先编码器时已经将所有的输人信号按优先顺序排了队，当几个输人信号同时出现时，只对其中优先权最高的一个进行编码。**

1. **基本工作原理**

**以74LS148为例（俯视图如下）：**



1. **‾S为选通输入端，S=1时编码器正常工作，S=0时所有输出端被封锁在高电平。**
2. **‾Ys为选通输出端，‾Ys=0（低电平）时表示编码器正常工作但无编码输入，此时S=1，且所有输入端均为高电平（无信号输入）。**
3. **‾YEX为扩展端，‾YEX=0（低电平）时表示编码器正常工作且有编码输入，此时S=1，且至少有一个输入端为低电平。**
4. **在正常工作且有编码输入信号中，‾I0--‾I7中可以有多个输入端输入低电平，但其中的优先度按‾I7 --‾I0递减，以此便保证了每个信号输入都有对应的编码输出。**
5. **逻辑功能（用途）**

**把输入的每一个高、低电平信号变成一个对应的二进制编码。**

1. **Verilog代码**

module code8\_3(I,Y);  
input[7:0]I;  
output[2:0]Y;  
function[2:0]code;  
input[7:0]I;  
if(I[7]) code=3'b111;  
else if(I[6]) code=3'b110;  
else if(I[5]) code=3'b101;  
else if(I[4]) code=3'b100;  
else if(I[3]) code=3'b011;  
else if(I[2]) code=3'b010;  
else if(I[1]) code=3'b001;  
else code=3'b000;  
endfunction  
assign Y=code(I);  
endmodule

1. **译码器（Decoder）**

**（一）分类：二进制译码器、二—十进制译码器和显示译码器**

1. **二进制译码器（最小项译码器）**
2. **特点**

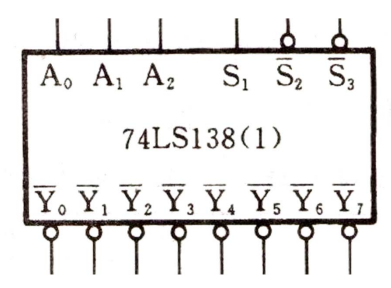
**输入一组二进制代码，输出一组与输入代码一一对应的高、低电平信号。**

1. **基本工作原理**

**二进制译码器可采用二极管与门阵列或三极管集成门电路构成。**

**二极管与门阵列常用于大规模（LSI）集成电路中，通过大量的二极管和若干与门构成，将输入的电平Vcc通过电路输出到给定数目的输出端口从而实现每个输出端口对应输入的二进制中的某一位高、低电平信号。**

**三极管集成门电路常用于中规模（MSI）集成电路中，以如下的74LS138为例：**



1. **S1=1时编码器正常工作，S=0时所有输出端被封锁在高电平（‾Yn=1 ）。**
2. **‾S2+‾S3=0时编码器正常工作，‾S2+‾S3=1时所有输出端被封锁在高电平（‾Yn=1 ）。**
3. **编码器正常工作时编码器根据输入端A0、A1、A2的输入输出一个对应的电平信号，该信号中只有一个输出端口输出为低电平其余均为高电平。通过真值表我们发现其输出满足最小项输出（‾Yn=‾mn ），故又称为最小项译码器。**
4. **逻辑功能**

**每个输入的二进制代码译成对应的输出高、低电平信号或另外一个代码,译码是编码的反操作。**

1. **Verilog代码**

module decoder\_38(E,A\_in,Y\_out);

input [2:0] A\_in;

input [2:0] E;

output [7:0] Y\_out;

reg [7:0] Y\_out;

always @(A\_in or E )

begin

if(E == 3'b100)

begin

case(A\_in)

3'b000: Y\_out = 8'b1111\_1110;

3'b001: Y\_out = 8'b1111\_1101;

3'b010: Y\_out = 8'b1111\_1011;

3'b011: Y\_out = 8'b1111\_0111;

3'b100: Y\_out = 8'b1110\_1111;

3'b101: Y\_out = 8'b1101\_1111;

3'b110: Y\_out = 8'b1011\_1111;

3'b111: Y\_out = 8'b0111\_1111;

endcase

end

else

begin

Y\_out = 8'b1111\_1111;

end

end

endmodule

1. **二—十进制译码器**
2. **特点**

**输入一组BCD代码，输出10个与输入代码一一对应的高、低电平信号以表示十进制数字。**

1. **基本工作原理**

**以常用的74LS42为例：**

1. **用四位二进制输入高、低电平信号，输出10个与输入代码对应的高、低电平信号以表示十进制数字，其中只有一位输出为低电平（‾Yn=0 ）。**
2. **当输入信号为伪码、不对应一个十进制数字时，输出全为高电平。**
3. **逻辑功能**

**将输入的BCD码的10个代码译成10个高、低电平信号输出信号，属于码制变换译码器中的一种。**

1. **显示译码器：详见第三部分**

**（二）用途**

1. **用作地址译码器或指令译码器，输入地址码，输出存储单元地址。**
2. **进行大容量编码时可扩展芯片。**
3. **实现一些逻辑函数。**
4. **可作数据分配器使用。**
5. **BCD七段数码管字符译码器**
6. **特点**

**将数字（0-9）、文字、符号（A-F）等的二进制代码翻译并显示出来，包括驱动电路和数码显示器两部分。属于译码器里显示译码器的一种。**

1. **基本工作原理**

**74LS48除了有实现7段显示译码器基本功能的输入（DCBA）和输出（Ya～Yg）端外，还引入了灯测试输入端（LT）和动态灭零输入端（RBI），以及既有  
输入功能又有输出功能的消隐输入/动态灭零输出（BI/RBO）端。**

**（1）七段译码功能（LT=1,RBI=1）**

**74LS48通过将DCBA四个高低电平输入进行译码，把对应的七位高低电平信号输出到abcdefg七个LED灯管上，从而构成了对应的图案来与输入对应。**

**在灯测试输入端（LT）和动态灭零输入端（RBI）都接无效电平时，输入DCBA经7448译码，输出高电平有效的7段字符显示器的驱动信号，显示相应字符。除DCBA= 0000外，RBI也可以接低电平，见表中1～16行。**

**（2）消隐功能（BI=0）**

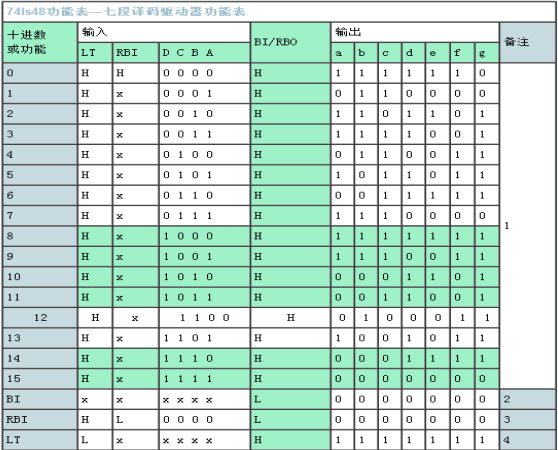
**此时BI/RBO端作为输入端，该端输入低电平信号时，表倒数第3行，无论LT 和RBI输入什么电平信号，不管输入DCBA是什么状态，输出全为“0”，7段显示器熄灭。该功能主要用于多显示器的动态显示。**

**（3）灯测试功能（LT=0）**

**此时BI/RBO端作为输出端，灯测试端输入低电平信号时，表最后一行，与及DCBA输入无关，输出全为“1”，显示器7个字段都点亮。该功能用于7段显示器测试，判别是否有损坏的字段。**

**（4）动态灭零功能（LT=1,RBI=1）**

**此时BI/RBO端也作为输出端， LT端输入高电平信号， RBI 端输入低电平信号，若此时DCBA=0000，表倒数第2行，输出全为“0”，显示器熄灭，不显示这个零。 DCBA≠0，则对显示无影响。该功能主要用于多个7段显示器同时显示时熄灭高位的零。**



1. **用途**
2. **七段译码功能**

**输出高电平有效的7段字符显示器的驱动信号，显示相应字符。**

**（2）消隐功能**

**BI/RBO端输入低电平信号时，输出全为“0”，7段显示器熄灭。该功能主要用于多显示器的动态显示。**

**（3）灯测试功能**

**LT端输入低电平信号时，输出全为“1”，显示器7个字段都点亮。该功能用于7段显示器测试，判别是否有损坏的字段。**

**（4）动态灭零功能**

**若DCBA=0000，输出全为“0”，显示器熄灭。 DCBA≠0，则对显示无影响。该功能主要用于多个7段显示器同时显示时熄灭高位的零。**

1. **Verilog代码**

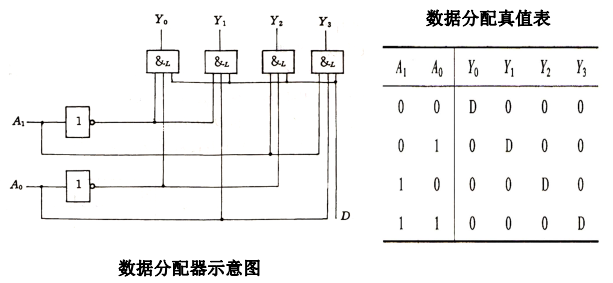
module bin27seg (data\_in ,EN ,data\_out );  
input [3:0] data\_in ;  
input EN ;  
output [6:0] data\_out ;  
reg [6:0] data\_out ;  
always @(data\_in or EN )  
begin  
data\_out = 7'b1111111;  
if (EN == 1)  
case (data\_in )  
4'b0000: data\_out = 7'b1000000; // 0  
4'b0001: data\_out = 7'b1111001; // 1  
4'b0010: data\_out = 7'b0100100; // 2  
4'b0011: data\_out = 7'b0110000; // 3  
4'b0100: data\_out = 7'b0011001; // 4  
4'b0101: data\_out = 7'b0010010; // 5  
4'b0110: data\_out = 7'b0000011; // 6  
4'b0111: data\_out = 7'b1111000; // 7  
4'b1000: data\_out = 7'b0000000; // 8  
4'b1001: data\_out = 7'b0011000; // 9  
4'b1010: data\_out = 7'b0001000; // A  
4'b1011: data\_out = 7'b0000011; // b  
4'b1100: data\_out = 7'b0100111; // c  
4'b1101: data\_out = 7'b0100001; // d  
4'b1110: data\_out = 7'b0000110; // E  
4'b1111: data\_out = 7'b0001110; // F  
default: data\_out = 7'b1111111;  
endcase  
end  
endmodule

1. **数据分配器（多路分配器、多路调节器）（DEMUX）**
2. **特点**

**能将数据分配到不同的数据通道上，其电路为单输入、多输出形式。**

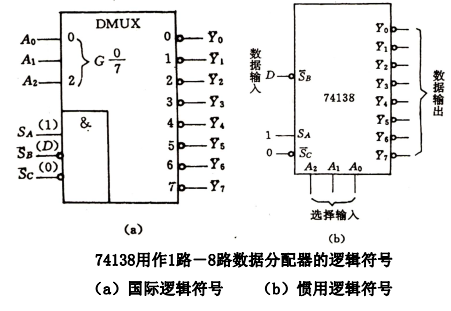
1. **基本工作原理**
2. **四路逻辑分配器**

**如图，D为被传输的数据，A0、A1是选择输入端，Y0-Y3为数据输出端。**



1. **1路－8路数据分配器**

**如图所示，74138不仅可以作3线－8线译码器，而且还可用作1路－8路数据分配器。**



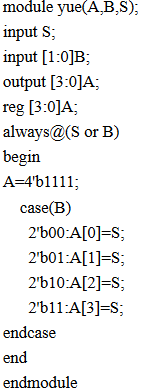
1. **用途**

**（1）与数据选择器连接组成分时数据传送系统。**

**（2）与计数器结合组成脉冲分配器。**

**（3）将一台PC机与多台外部设备连接，将计算机的数据分送到外部设备中。**

1. **Verilog代码**

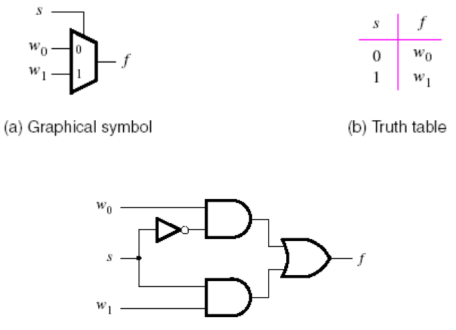


1. **数据选择器（多路调制器、多路开关、多路选择器）（MUX）**
2. **特点**

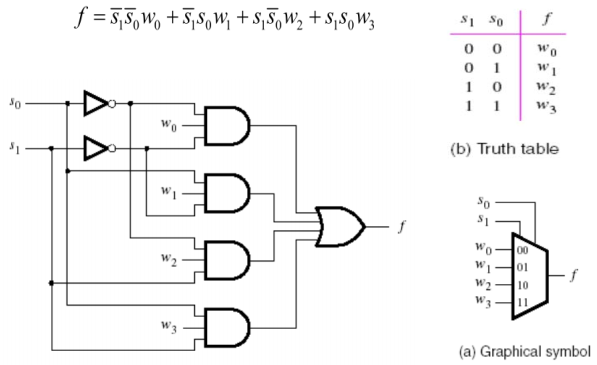
**能从多路数据中选择一路进行传输，其电路为多输入、但输出形式，与数据分配器相反。**

**2.基本工作原理**

1. **二选一多路选择器**

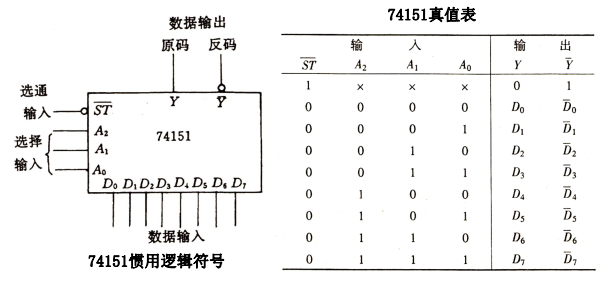


1. **四选一多路选择器**



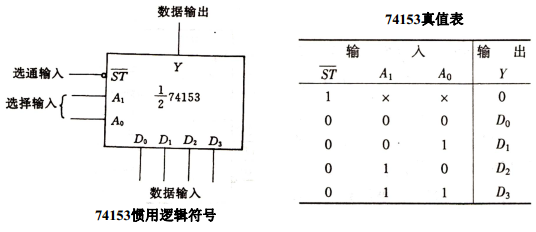
1. **八选一多路选择器74151**

**八选一MUX需要3个选择输入端，8个数据输入端，并有互补的原码和反码两种输出形式。74151的惯用逻辑符号和真值表如下所示：**



1. **双四选一多路选择器74153**

**74153包含两个完全相同的4选一MUX，两个MUX有公共的地址输入端，而数据输入和输出端各自独立。通过给定不同的地址代码（A1A0）,即可从4个输入数据中选出所需要的一个，并送至输出端Y。74153的惯用逻辑符号和真值表如下：**



**3.用途**

1. **与数据分配器连接组成分时数据传送系统。**
2. **实现组合逻辑函数。**
3. **对已有的数据选择器进行扩展。**

**4.Verilog代码**

module mux(d1,d2,d3,d4,[se](https://www.baidu.com/s?wd=se&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBrHT4uyP-myc1mHRzuAc40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1ndnHc4PWRd)1,[se](https://www.baidu.com/s?wd=se&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBrHT4uyP-myc1mHRzuAc40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1ndnHc4PWRd)2,dout);  
 input d1;  
 input d2;  
 input d3;  
 input d4;  
 input [se](https://www.baidu.com/s?wd=se&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBrHT4uyP-myc1mHRzuAc40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1ndnHc4PWRd)1;  
 input se2;  
 output dout;  
 reg dout;  
 always @ (d1 or d2 or d3 or d4 or se1 or se2)  
 case({se1,se2})  
 2'b00 : dout=d1;  
 2'b01 : dout=d2;  
 2'b10 : dout=d3;  
 2'b11 : dout=d4;  
 [end](https://www.baidu.com/s?wd=end&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBrHT4uyP-myc1mHRzuAc40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1ndnHc4PWRd)case  
[end](https://www.baidu.com/s?wd=end&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBrHT4uyP-myc1mHRzuAc40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1ndnHc4PWRd)module

1. **数值比较器**

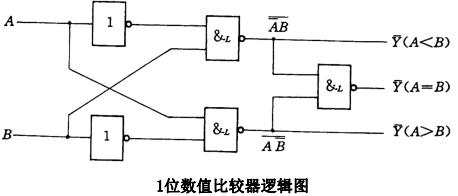
**分类：一位数值比较器和多位数值比较器**

1. **一位数值比较器**
2. **特点**

**输入两个1位二进制数比较大小，并有三种输出信号。**

1. **基本工作原理**

**两个1位二进制数A,B相比的情况有以下几种：  
 ①A>B（即A=1,B=0）,则A‾B=1，所以可用A‾B作为A>B的输出信号。  
 ②同理可用‾AB作为A<B的输出信号。  
 ③同理可用A⊙B作为A=B的输出信号。**



1. **用途**

**能比较两个1位二进制数的大小。**

1. **Verilog代码**

1 module compare ( Y ,A ,B );

2 input A ;

3 wire A ;

4 input B ;

5 wire B ;

6 output [2:0] Y ;

7 reg [2:0] Y ;

8 always @ ( A or B )

9 begin

10 if ( A > B )

11 Y <= 3'b001;

12 else if ( A == B)

13 Y <= 3'b010;

14 else

15 Y <= 3'b100;

16 end

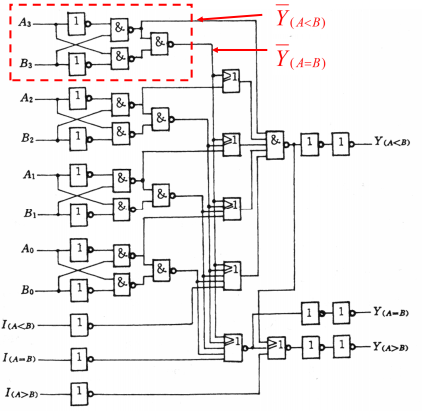
17 endmodule

1. **多位数值比较器**
2. **特点**

**输入两个多位二进制数比较大小，并有三种输出信号。**

1. **基本工作原理**

**在比较两个多位数的大小时，必须自高而低的逐位比较，而且只有在高位相等时，才需比较较低位。  
 下图示出了4位比较器CC14585的逻辑图。在比较两个4位以上的二进制数时，应将两片以上的CC14585级联，组合成位数更多的比较电路。**



1. **用途**

**能比较两个多位二进制数的大小。**

1. **Verilog代码**

1 module compare ( Y ,A ,B );

2 input [3:0] A ;

3 wire [3:0] A ;

4 input [3:0] B ;

5 wire [3:0] B ;

6 output [2:0] Y ;

7 reg [2:0] Y ;

8 always @ ( A or B )

9 begin

10 if ( A > B )

11 Y <= 3'b001;

12 else if ( A == B)

13 Y <= 3'b010;

14 else

15 Y <= 3'b100;

16 end

17 endmodule

1. **加法器（Adder）**

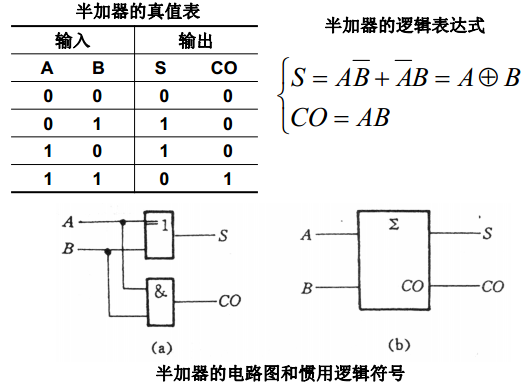
**分类：半加器、全加器、串行进位加法器和超前进位加法器**

1. **半加器（Half Adder）**
2. **特点**

**一位加法器的一种，能够实现半加（不考虑有来自低位进位的两个1位二进制数相加）功能。**

1. **基本工作原理**

**半加器具有两个输入A、B和两个输出S（表示加和后该位数字）、CO（表示进位）。其原理图如下。**



1. **用途**

**实现半加（不考虑有来自低位进位的两个1位二进制数相加）功能。**

1. **Verilog代码**

module h\_adder (A, B, SO, CO);

input A, B;

output SO, CO;

assign SO = A ^ B;

assign CO = A & B;

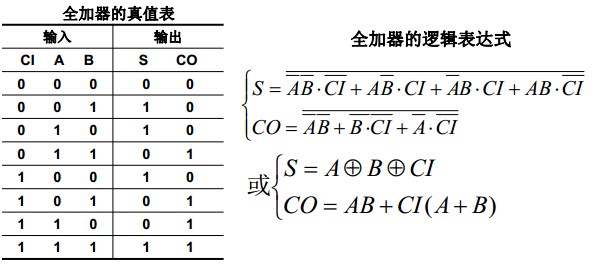
endmodule

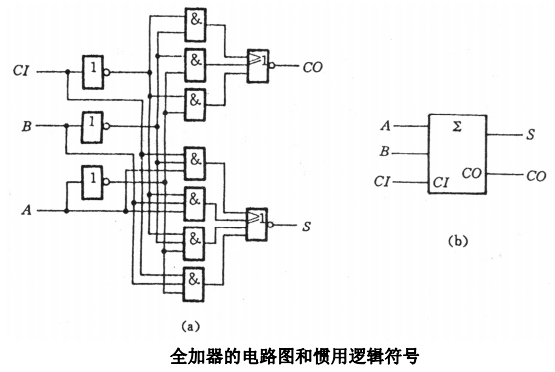
1. **全加器（Full Adder）**
2. **特点**

**一位加法器的一种，能够实现全加（两个多位二进制数相加时将两个对应的加数和来自低位的进位3个数相加）功能。**

1. **基本工作原理**

**以1位全加器为例，其具有三个输入A、B（表示加数）、CI（表示低位进位）和两个输出S（表示加和后该位数字）、CO（表示该位进位）。其原理图如下。**





1. **用途**

**实现全加（两个多位二进制数相加时将两个对应的加数和来自低位的进位3个数相加）功能。**

1. **Verilog代码**

module Fadd(x,y,Cin,Cout,Sum);

input x,y,Cin;

output Cout,Sum;

wire a,b,c;

xor xor1(a,x,y);

xor2(Sum,a,Cin);

and and1(b,x,y);

and2(c,Cin,a);

or or1(Cout,b,c);

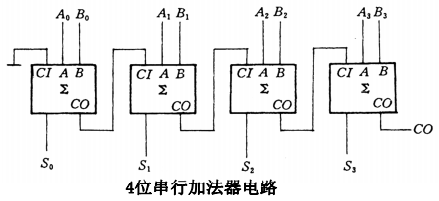
Endmodule

1. **串行进位加法器**
2. **特点**

**多位加法器的一种，n位串行进位加法器共有2n个输入和n+1个输出。**

1. **基本工作原理**

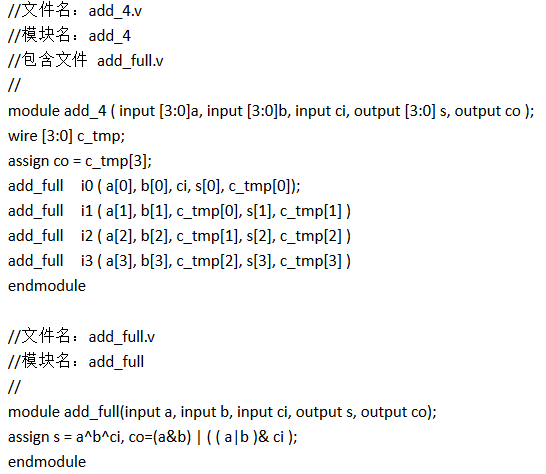
**依次将低位全加器的进位输出端CO接到高位全加器的进位输入端CI。四位串行进位加法器的原理图如下。**



1. **用途**

**实现多位数全加功能，可用于多人表决电路。**

1. **Verilog代码**

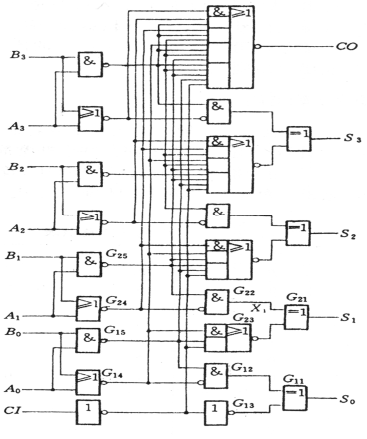


1. **超前进位加法器**
2. **特点**

**多位加法器的一种，n位串行进位加法器共有2n+1个输入和n+1个输出。其采用超前进位加法进行全加运算，效率较高。**

1. **基本工作原理**

**为了提高运算速度，须减小或消除由于进位信号逐级传递所耗费的时间，便可以采用超前进位加法的方法进行改进：由于第i位的进位输入信号(CI)i一定能由**IMG_256**和**IMG_256**唯一确定，所以可先得出每一位全加器的进位输入信号，而无需再从最低位开始向高位逐级传递进位信号，这就有效的提高了运算速度。下图示出了4位超前进位加法器74LS283的逻辑图：**



**3.用途**

**实现多位数全加功能，并大大提升了运算效率、节省了运算时间。**

**4.Verilog代码**

1. module Design\_Code(
2. input [3:0] num\_1,
3. input [3:0] num\_2,
4. output reg [3:0] s,
5. output reg CF
6. );
8. reg [3:0] P;
9. reg [3:0] G;
10. integer i;
12. always @ (\*)
13. begin
14. for(i=0; i<4; i=i+1)
15. begin
16. G[i] = num\_1[i]&num\_2[i];
17. P[i] = num\_1[i]^num\_2[i];
18. end
20. s[0] = P[0];
21. s[1] = P[1]^G[0];
22. s[2] = P[2]^(G[1]|P[1]&G[0]);
23. s[3] = P[3]^(G[2]|(P[2]&G[1])|(P[2]&P[1]&G[0]));
24. CF = G[3]|(P[3]&G[2])|(P[3]&P[2]&G[1])|(P[3]&P[2]&P[1]&G[0]);
25. end
27. endmodule