《数字逻辑》
Digital Logic

组合逻辑（3）

北京工业大学软件学院
王晓懿

常用组合逻辑电路

- 编码器 (Encoder)

编码器 (Encoder)

- 编码器(Encoder)原理
- 优先编码器(Priority Encoder)
- 8-3优先编码器
- 扩展应用：16-4 优先编码器

编码器(Encoder)原理

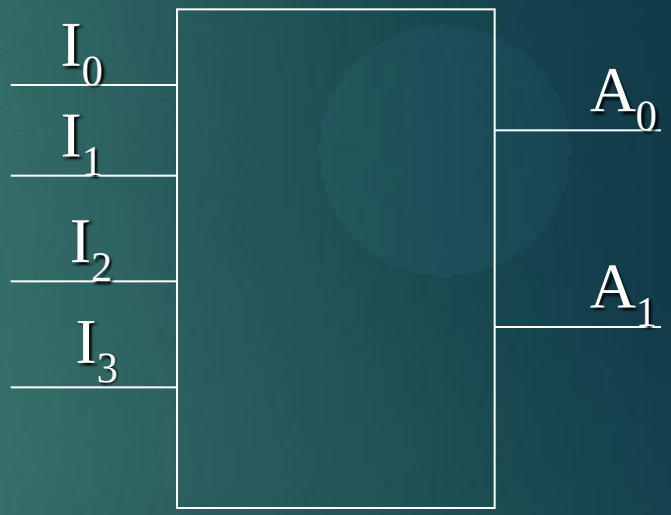
- 功能：将译码器反过来，对应输入的每一个状态，输出一个编码。
- 4-2编码，将输入的4个状态编成2位二进制数码； 8-3编码，将输入的8个状态编成3位二进制数码；BCD编码，将10个输入编成BCD码。

编码器(Encoder)原理

例：4-2编码器

功能表

I_0	I_1	I_2	I_3	A_0	A_1
0	1	1	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	0	1	1



$$\begin{cases} A_0 = I_0\overline{I_1}I_2I_3 + I_0I_1I_2\overline{I_3} = \overline{\overline{I_0}I_1I_2I_3 + I_0I_1\overline{I_2}I_3} \\ A1 = I_0I_1\overline{I_2}I_3 + I_0I_1I_2\overline{I_3} = \overline{\overline{I_0}I_1I_2I_3 + I_0\overline{I_1}I_2I_3} \end{cases}$$

8421码编码器

$$Y_3 = X_8 + X_9$$

$$Y_2 = X_4 + X_5 + X_6 + X_7$$

$$Y_1 = X_2 + X_3 + X_6 + X_7$$

$$Y_0 = X_1 + X_3 + X_5 + X_7 + X_9$$

X_9	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	1

8421码编码器

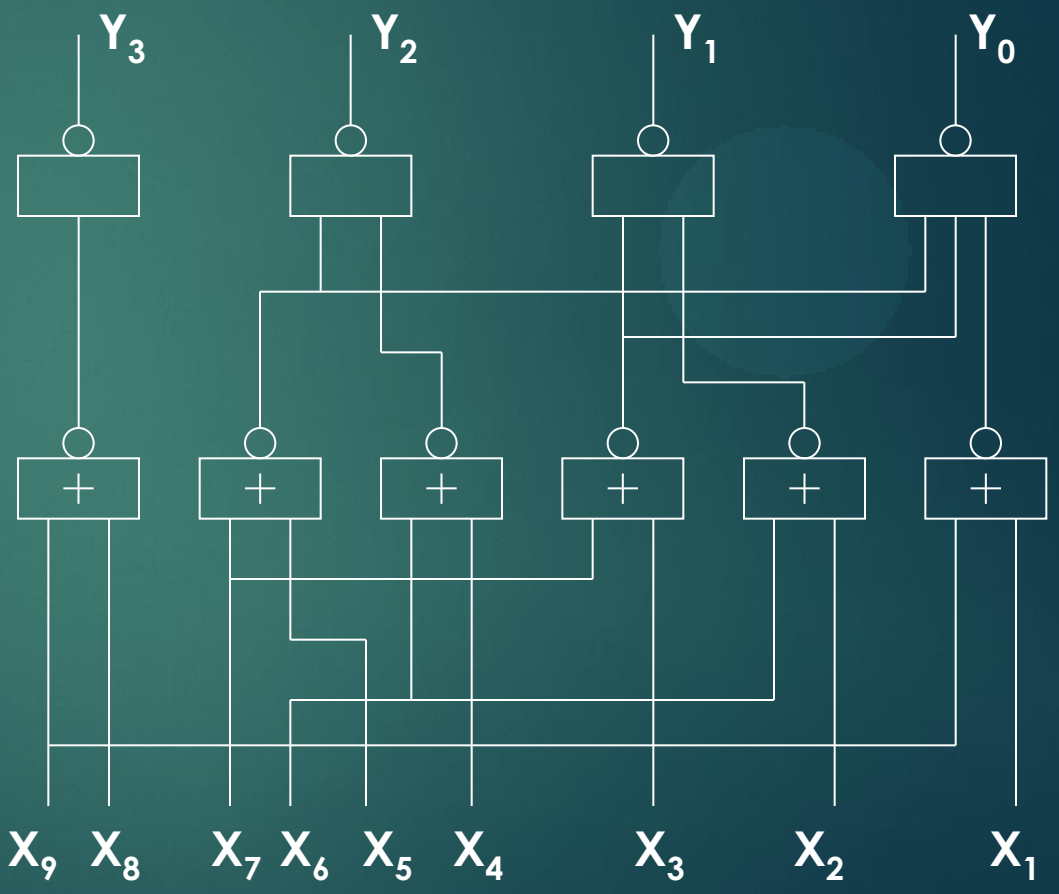
$$Y_3 = X_8 + X_9$$

$$Y_2 = X_4 + X_5 + X_6 + X_7$$

$$Y_1 = X_2 + X_3 + X_6 + X_7$$

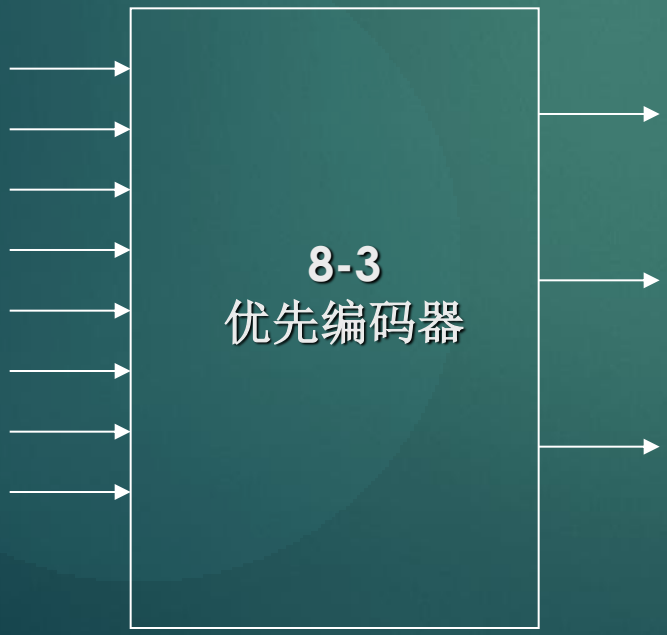
$$Y_0 = X_1 + X_3 + X_5 + X_7 + X_9$$

局限：只有互斥输入时，才能用这种编码器。即在任一时刻所有输入线中只允许有一个为“1”，否则编码器会发生混乱。必须用优先编码器。



优先编码器

当两条或两条以上线为“0”时，优先按输入编号大的编码，称优先编码器(Priority Encoder)。
以8-3优先编码器为例。



优先编码功能表

$\overline{E_i}$	0	1	2	3	4	5	6	7	A_0	A_1	A_2	G_S	E_0	
0	X	X	X	X	X	X	X	0	0	0	0	0	1	7
0	X	X	X	X	X	X	0	1	1	0	0	0	1	6
0	X	X	X	X	X	0	1	1	0	1	0	0	1	5
0	X	X	X	X	0	1	1	1	1	1	0	0	1	4
0	X	X	X	0	1	1	1	1	0	0	1	0	1	3
0	X	X	0	1	1	1	1	1	1	0	1	0	1	2
0	X	0	1	1	1	1	1	1	0	1	1	0	1	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0	
1	X	X	X	X	X	X	X	X	1	1	1	1	1	

(A_2,A_1,A_0用反码编码，Gs为编码输出，Eo为使能输出,E_i为使能输入)

$$\overline{A_0} = \overline{E_i}(\bar{7} + 76\bar{5} + 765\bar{4}3 + 7654321) = \overline{E_i}(\bar{7} + 6\bar{5} + 6\bar{4}3 + 6421)$$

$$\overline{A_1} = \overline{E_i}(\bar{7} + 7\bar{6} + 765\bar{4}3 + 7654321) = \overline{E_i}(\bar{7} + \bar{6} + 5\bar{4}3 + 542)$$

$$\overline{A_2} = \overline{E_i}(\bar{7} + 7\bar{6} + 76\bar{5} + 765\bar{4}) = \overline{E_i}(\bar{7} + \bar{6} + \bar{5} + \bar{4})$$

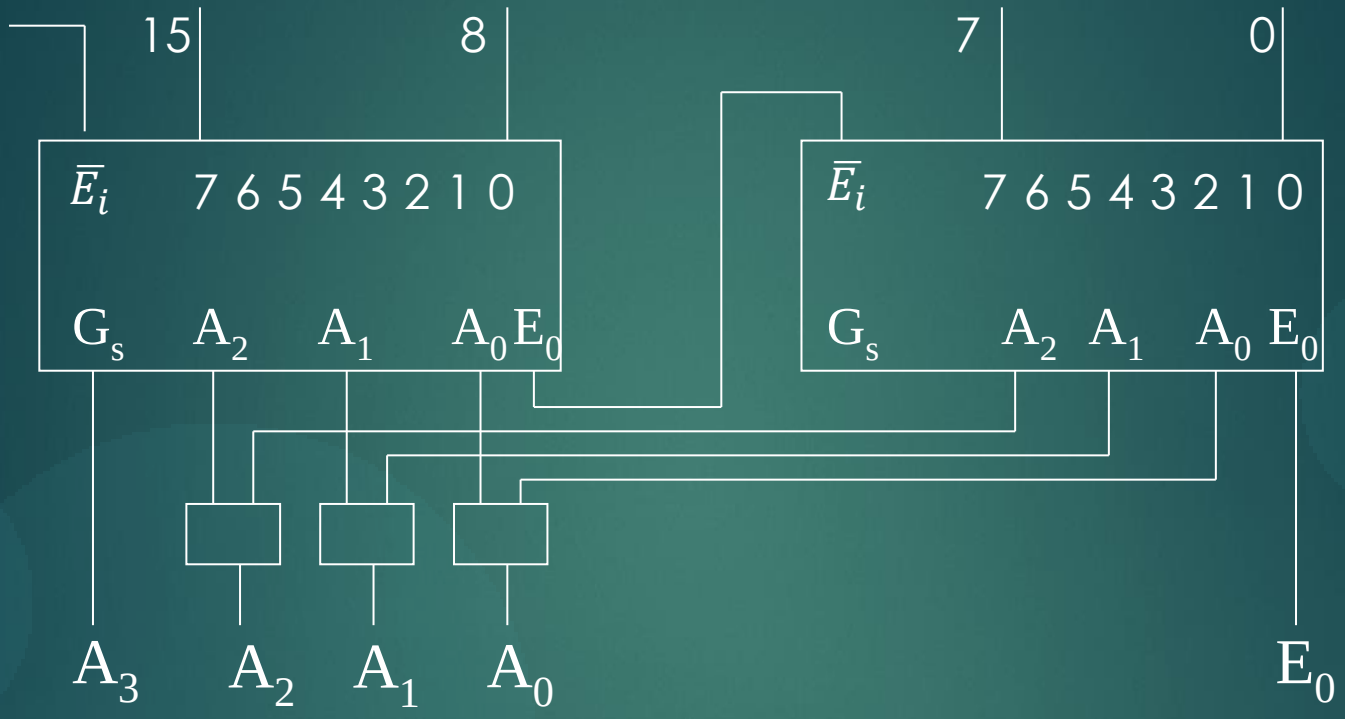
$$E_0 = \overline{E_i}\,\overline{76543210}$$

(E_0=0，表示本片没有编码，多片相连时低位可以编码)

用公式化简：

$A+\overline{A}B=A+B$

3-8优先编码器扩展为16-4优先编码器



若高位片有“0”输入，高位Eo=1,应禁止低位片，以$(A_{2\sim0})_{高}$作为$(A_{2\sim0})_{16\text{-}4}$，高位片的Gs(=0)作为$A_3$

若高位片无“0”输入，高位Eo=0,低位片工作，以$(A_{2\sim0})_{低}$作为$(A_{2\sim0})_{16\text{-}4}$，高位片的Gs(=1)作为$A_3$

编码器的Verilog描述

```
module encoder83(output x,
                 output y,
                 output z,
                 input [7:0] d);
 or(x,d[4],d[5],d[6],d[7]);
 or(y,d[2],d[3],d[6],d[7]);
 or(z,d[1],d[3],d[5],d[7]);
endmodule
```

```
module pri_encoder83 (
  output [2:0] binary_out,
  input  [7:0] encoder_in,
  input  enable );

  assign  binary_out  = ( ! enable) ? 3'd7 : (
        (~encoder_in[7]) ? 3'd7 :
        (~encoder_in[6]) ? 3'd6 :
        (~encoder_in[5]) ? 3'd5 :
        (~encoder_in[4]) ? 3'd4 :
        (~encoder_in[3]) ? 3'd3 :
        (~encoder_in[2]) ? 3'd2 :
        (~encoder_in[1]) ? 3'd1 :
        (~encoder_in[0]) ? 3'd0 : 3'd7);

endmodule
```

常用组合逻辑电路

- 数据比较器

数据比较器

$A_3A_2A_1A_0$
$B_3B_2B_1B_0$

从高位开始比较，

若$A_3>B_3$ 则A>B，
若$A_3<B_3$ 则A<B，
若$A_3=B_3$ 则再比较低位

功能：比较A、B两数大小，判断A>B、A<B、A=B

$A_i>B_i$的条件：$A_i=1$，$B_i=0$；即 $A_i \bullet \overline{B_i}=1$或$Z=A_i \bullet \overline{A_iB_i}=1$

$A_i<B_i$的条件：$A_i=0$，$B_i=1$；即 $\overline{A_i} \bullet B_i=1$或$W=B_i \bullet \overline{A_iB_i}=1$

$A_i=B_i$的条件：$\overline{A_i \oplus B_i}=1$或$Y=\overline{A_i \bullet \overline{A_iB_i}+B_i \bullet \overline{A_iB_i}}=1$

数据比较器功能表

A_3 ， B_3	A_2 ， B_2	A_1 ， B_1	A_0 ， B_0	A>B	A<B	A=B
$A_3 > B_3$	X	X	X	1	0	0
$A_3 < B_3$	X	X	X	0	1	0
$A_3 = B_3$	$A_2 > B_2$	X	X	1	0	0
$A_3 = B_3$	$A_2 < B_2$	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1

表达式：

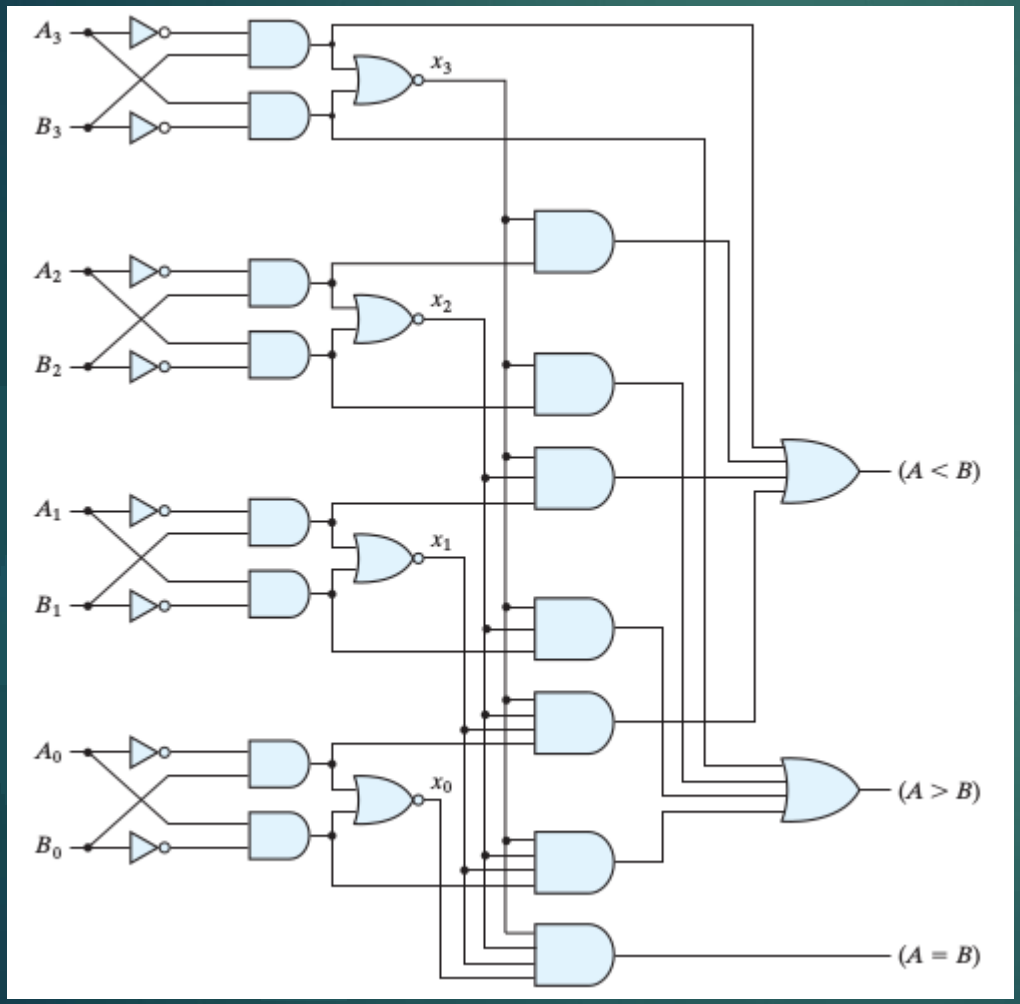
$$(A > B) = \overline{W_3 + Y_3 W_2 + Y_3 Y_2 W_1 + Y_3 Y_2 Y_1 W_0 + Y_3 Y_2 Y_1 Y_0}$$

$$(A < B) = \overline{Z_3 + Y_3 Z_2 + Y_3 Y_2 Z_1 + Y_3 Y_2 Y_1 Z_0 + Y_3 Y_2 Y_1 Y_0}$$

$$(A = B) = Y_3 Y_2 Y_1 Y_0$$

(Y_i表示$A_i = B_i$; W_i表示$A_i < B_i$; Z_i表示$A_i > B_i$

数据比较器逻辑图



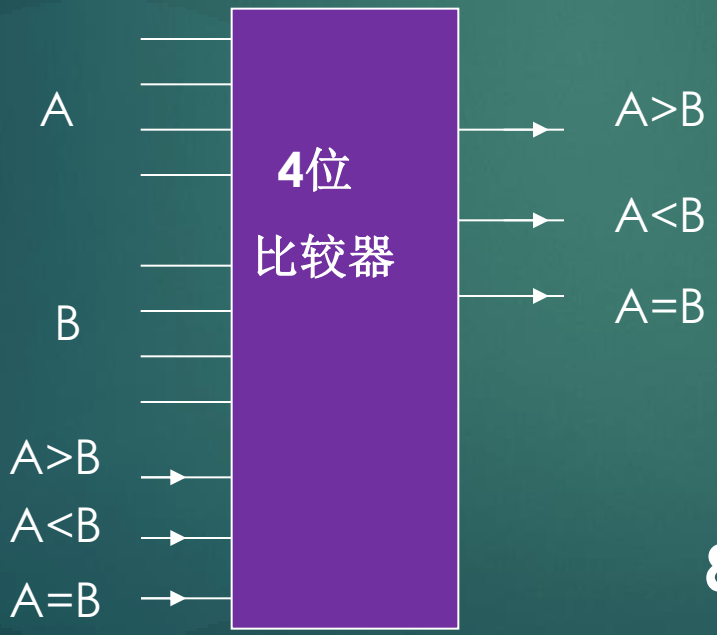
这是4位并行比较器，一次判断4位数大小。

按此思想可以推广到多位并行比较。

但是位数多了以后会出现组合爆炸，因此用分段比较。

分段比较:
多片比较器构成更长位数的方法

比较器不仅输出比较结果，还要能接受其它片输出的结果。



74LS85

8位比较器：P130 图4—45

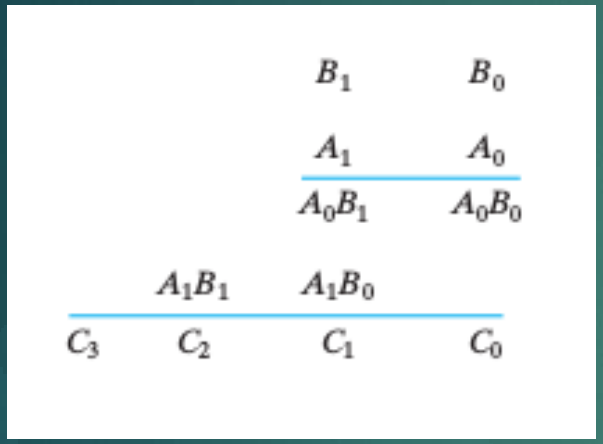
数据比较器的Verilog描述

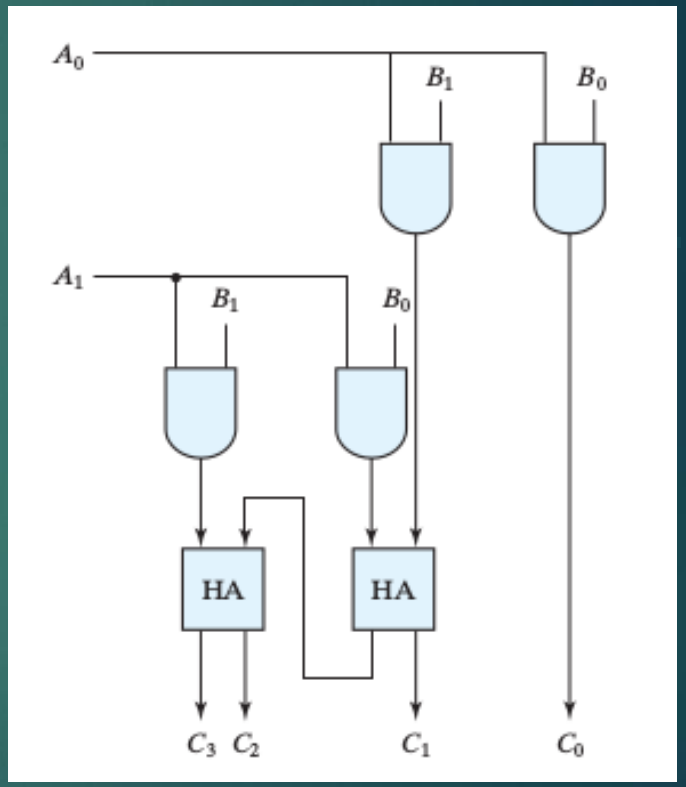
```
module compare(output A_lt_B, A_eq_B, A_gt_B,
                      input [3: 0] A, B );
   assign A_lt_B = (A < B);
   assign A_gt_B = (A > B);
   assign A_eq_B = (A == B);
endmodule
```

其他运算器

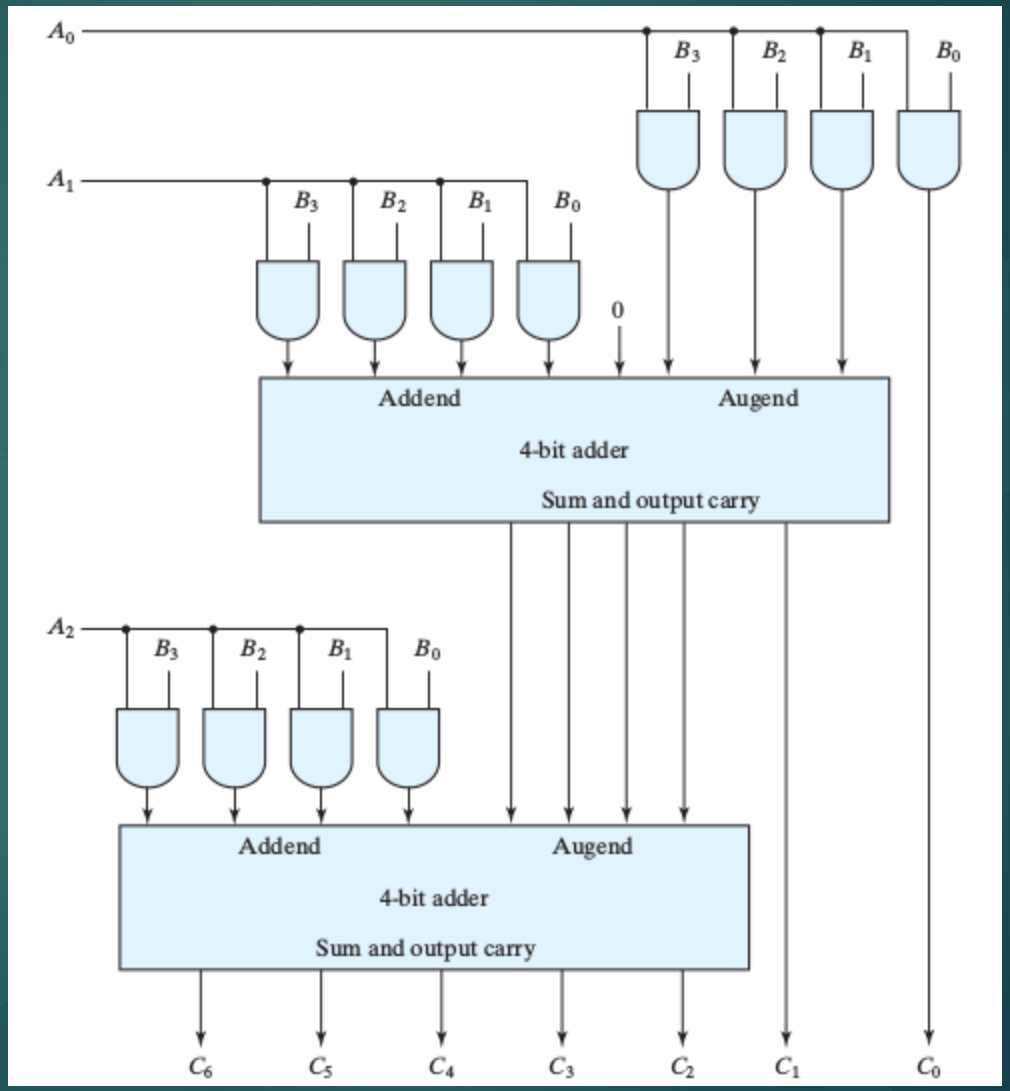
- 乘法器
- 算术逻辑部件ALU

乘法器


B_1 B_0
A_1 A_0
A_0B_1 A_0B_0
A_1B_1 A_1B_0
C_3 C_2 C_1 C_0


A_0
B_1
B_0
A_1
B_1
B_0
HA
HA
C_3 C_2
C_1
C_0

四位乘法器


A_0
B_3 B_2 B_1 B_0
A_1
B_3 B_2 B_1 B_0
0
Addend
Augend
4-bit adder
Sum and output carry
A_2
B_3 B_2 B_1 B_0
Addend
Augend
4-bit adder
Sum and output carry
C_6 C_5 C_4 C_3 C_2 C_1 C_0

乘法器的Verilog结构描述

```
module product(output [6:0] res,
        input [3:0] a, input [2:0] b);
  wire [4:0] r1, i1;
  wire [3:0] r2,r3;

  assign r1[0]=a[0]&b[0]; assign r1[1]=a[1]&b[0];
  assign r1[2]=a[2]&b[0]; assign r1[3]=a[3]&b[0];
  assign r2[0]=a[0]&b[1]; assign r2[1]=a[1]&b[1];
  assign r2[2]=a[2]&b[1]; assign r2[3]=a[3]&b[1];
  assign r3[0]=a[0]&b[2]; assign r3[1]=a[1]&b[2];
  assign r3[2]=a[2]&b[2]; assign r3[3]=a[3]&b[2];

  assign r1[4]=1'b0;
  assign res[0]=r1[0];
  adder a1(r1[4:1],r2,i1[3:0],i1[4],1'b0);
  assign res[1]=i1[0];
  adder a2(i1[4:1],r3,res[5:2],res[6],1'b0);
endmodule
```

算术逻辑部件ALU

- 实现多个算数逻辑运算
- 计算机的核心部件
- 《计算机组成原理》课程将进一步详细介绍

4位ALU功能表 ： 16种算数逻辑运算

S_3	S_2	S_1	S_0	正逻辑 $M=H$ 逻辑运算	正逻辑 $M=L$ 算术运算 $C_n=1$	正逻辑 $M=L$ 算术运算 $C_n=0$
L	L	L	L	$\overline{A}$	A	$A+1$
L	L	L	H	$\overline{A+B}$	$A+B$	$(A+B)$加 1
L	L	H	L	$\overline{A}\cdot B$	$A+\overline{B}$	$(A+\overline{B})$加 1
L	L	H	H	“0”	减 1	“0”
L	H	L	L	$\overline{A\cdot B}$	A 加$(A\cdot\overline{B})$	A 加$(A\cdot\overline{B})$加 1
L	H	L	H	$\overline{B}$	$(A\cdot\overline{B})$加$(A+B)$	$(A\cdot\overline{B})$加$(A+B)$加 1
L	H	H	L	$A\oplus B$	A 减 B 减 1	A 减 B
L	H	H	H	$A\cdot\overline{B}$	$(A\cdot\overline{B})$减 1	$A\cdot\overline{B}$
H	L	L	L	$\overline{A}+B$	A 加$(A\cdot B)$	A 加$(A\cdot B)$加 1
H	L	L	H	$\overline{A\oplus B}$	A 加 B	A 加 B 加 1
H	L	H	L	B	$(A\cdot B)$加$(A+\overline{B})$	$(A\cdot B)$加$(A+\overline{B})$加 1
H	L	H	H	$A\cdot B$	$(A\cdot B)$减 1	$A\cdot B$
H	H	L	L	“1”	A 加 A	A 加 A 加 1
H	H	L	H	$A+\overline{B}$	A 加$(A+B)$	A 加$(A+B)$加 1
H	H	H	L	$A+B$	A 加$(A+\overline{B})$	A 加$(A+\overline{B})$加 1
H	H	H	H	A	A 减 1	A

(b) 功能表(正逻辑)

图 2.9　四位 ALU 逻辑图及功能表