

## 7.4 加法器和比较器

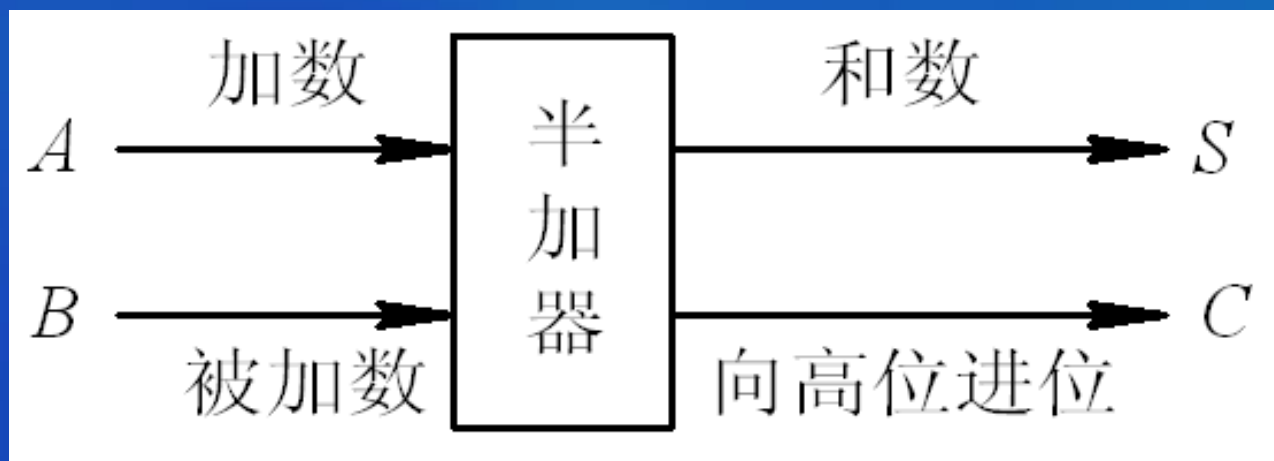
### 加法器

数字运算是数字系统基本的功能之一，加法器(adder)是执行算术运算的重要逻辑部件，在数字系统和计算机中，二进制数的加、减、乘、除等运算都可以转换为若干步加法运算。

# 1.半加器

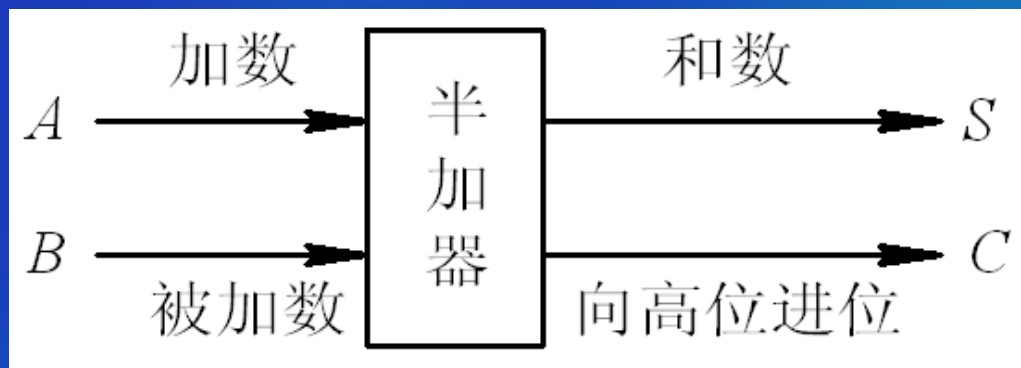
两个一位二进制数 $A$ 和 $B$ 相加，不考虑低位进位的加法器称为半加器(Half Adder, 简称HA)。

## (1) 半加器功能框图



## (2) 半加器真值表

<i>A</i>	<i>B</i>	<i>S</i>	<i>C</i>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

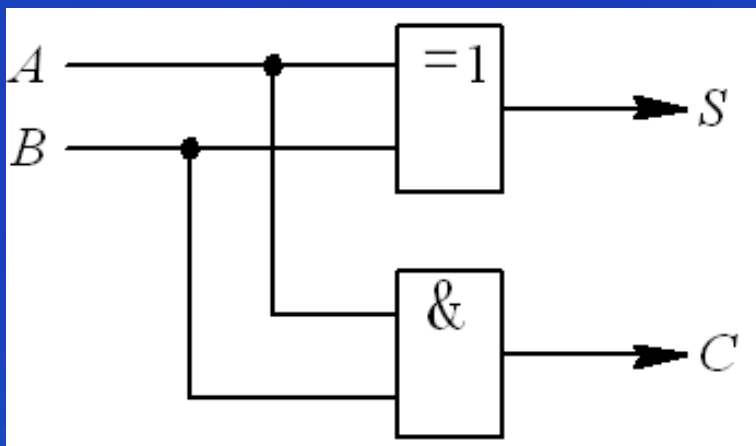


## (3) 半加器逻辑表达式

$$S = A \oplus B$$

$$C = AB$$

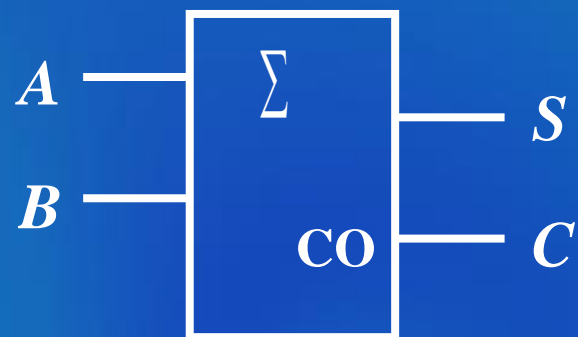
## (4) 半加器逻辑图



$$S = A \oplus B$$

$$C = AB$$

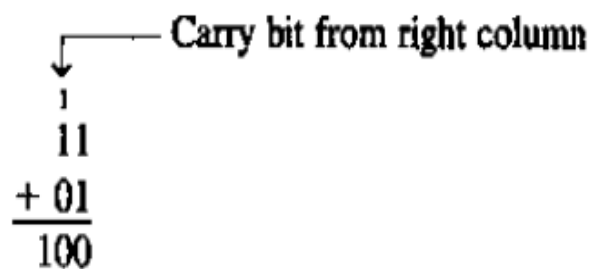
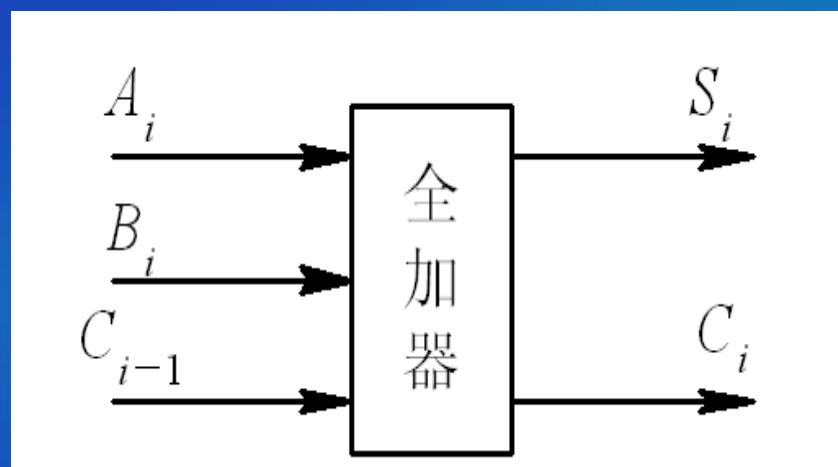
## (5) 半加器符号图



## 2. 全加器

两个一位二进制数 $A_i$ 和 $B_i$ 相加，考虑到相邻低位的进位 $C_{i-1}$ 的加法器称为全加器(Full Adder, 简称FA)。

### (1) 全加器框图



In this case, the carry bit from second column becomes a sum bit.

## (2) 全加器真值表

## (3) 全加器逻辑表达式

$$S_i = \bar{A}_i \bar{B}_i C_{i-1} + \bar{A}_i B_i \bar{C}_{i-1} + A_i \bar{B}_i \bar{C}_{i-1} + A_i B_i C_{i-1}$$

$$= (\bar{A}_i B_i + A_i \bar{B}_i) \bar{C}_{i-1} + (\bar{A}_i \bar{B}_i + A_i B_i) C_{i-1}$$

$$= (A_i \oplus B_i) \bar{C}_{i-1} + \overline{A_i \oplus B_i} C_{i-1}$$

$$= A_i \oplus B_i \oplus C_{i-1}$$

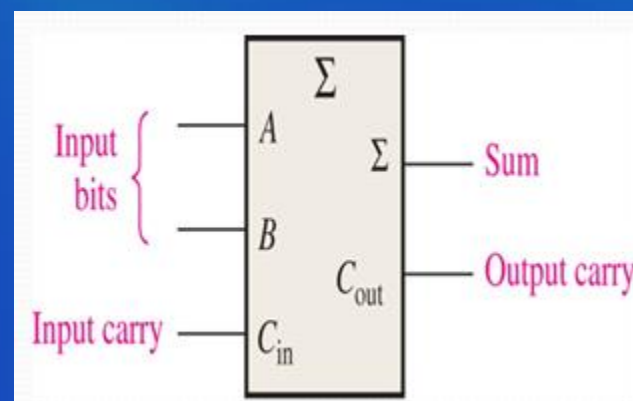
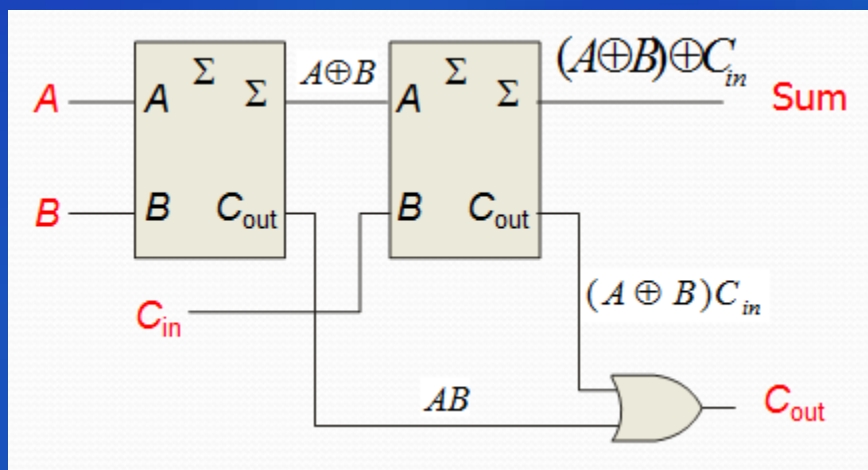
$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$C_i = A_i \bar{B}_i C_{i-1} + \bar{A}_i B_i C_{i-1} + A_i B_i \bar{C}_{i-1} + A_i B_i C_{i-1}$$

$$= (A_i \bar{B}_i + \bar{A}_i B_i) C_{i-1} + A_i B_i$$

$$= (A_i \oplus B_i) C_{i-1} + A_i B_i$$

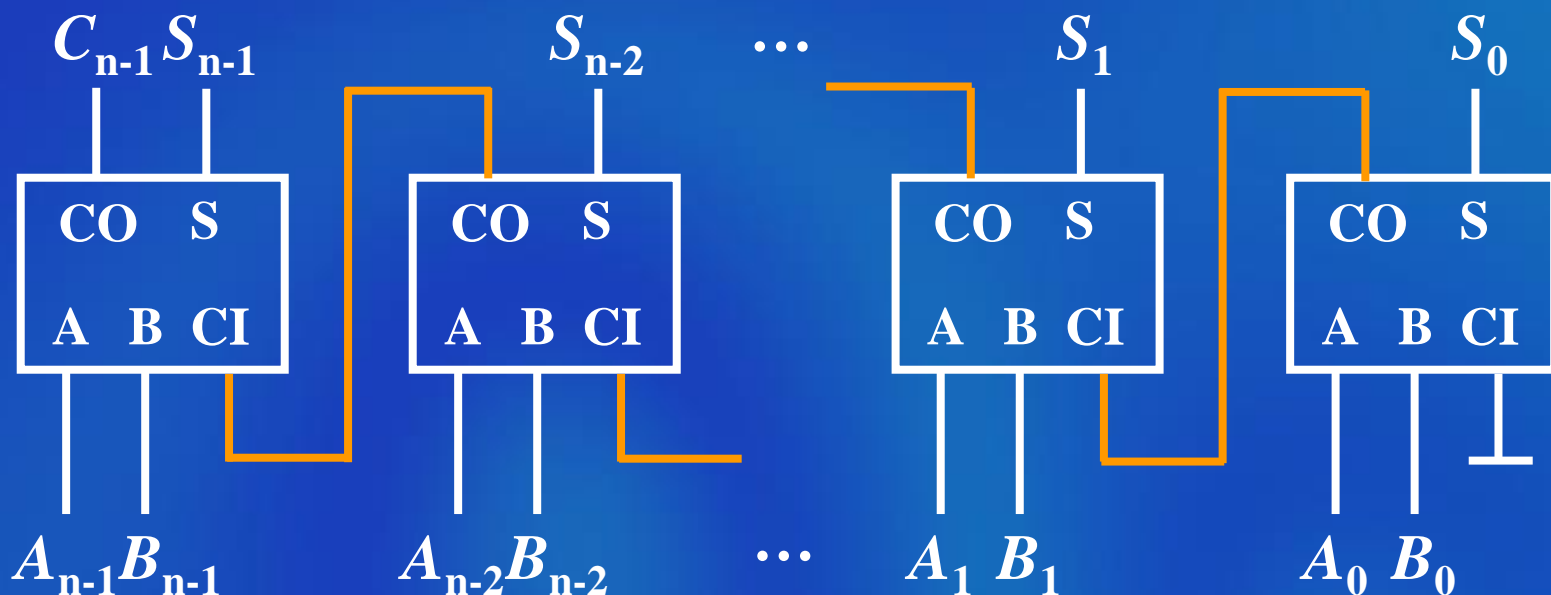
$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1



### 3. 多位二进制加法

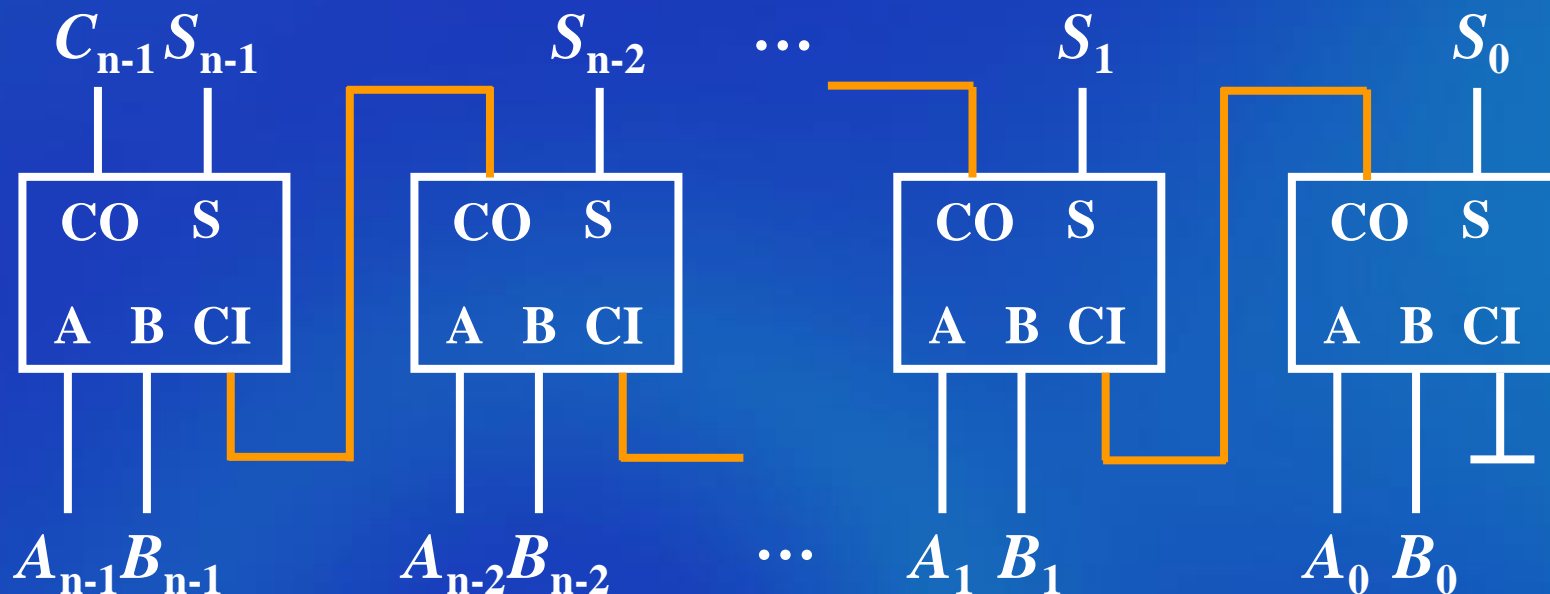
#### (1) 串行进位

将 $n$ 个全加器级联实现两个 $n$ 位二进制数相加电路





## 串行进位加法器的特点:



高位相加的结果只有等到低位进位产生后才能建立起来。

这种结构的电路称为串行进位加法器或行波加法器。

缺点是运算速度慢,优点是结构简单。

## 2.超前进位加法器

采用超前进位技术。

超前进位全加器内部进位信号 $C_i$ 可写为如下表达式

$$C_i = f_i(A_1, \dots, A_4, B_1, \dots, B_4, CI)$$

即 进位信号仅由加数和被加数决定，从而有效地提高了运算速度。

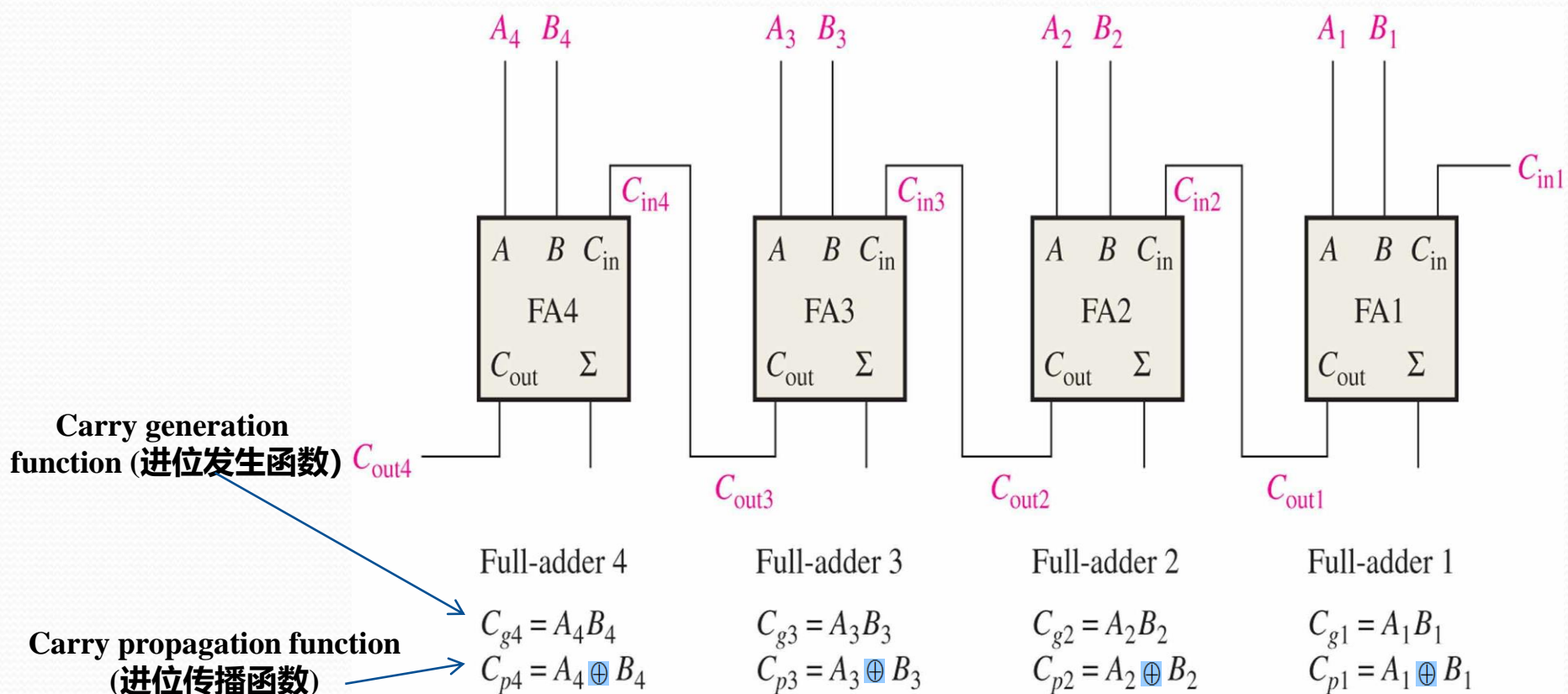
运算速度的提高是靠增加电路复杂程度换取的，而且随着位数的增多，电路复杂程度更甚。

目前中规模集成超前进位全加器多为4位。

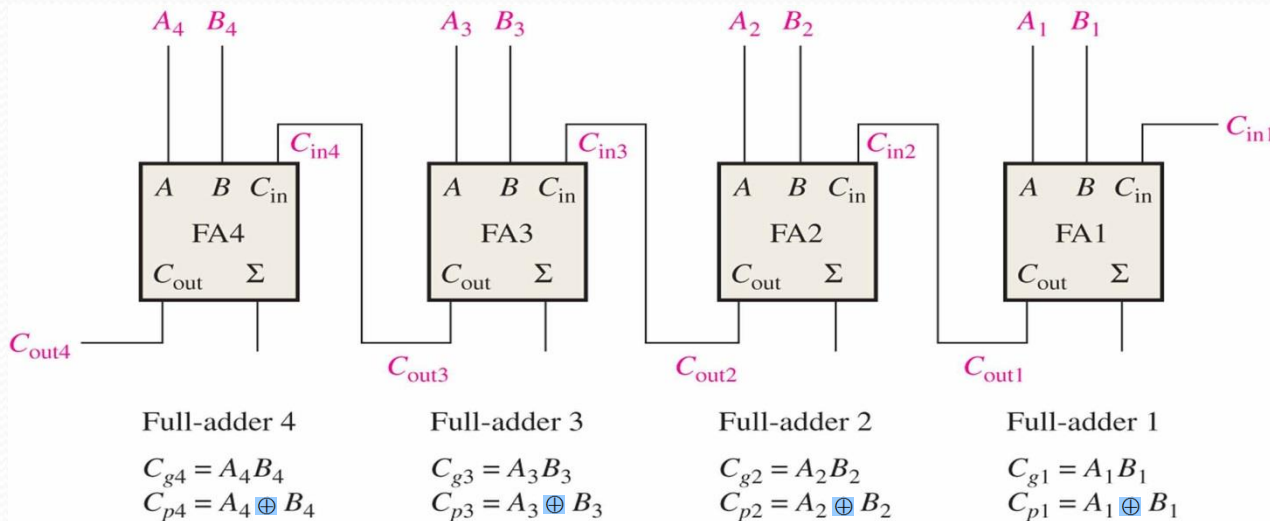
# The look-ahead carry adder

Principles:

$$C_{out} = (A \oplus B)C_{in} + AB$$



# The look-ahead carry adder



The output carry for each adder is dependent only on the initial input carry  $C_{in1}$ , and the A and B inputs to that adder. So all the output carries are immediately available.

Full-adder 1:

$$C_{out1} = C_{g1} + C_{P1}C_{in1}$$

Full-adder 2:

$$C_{out2} = C_{g2} + C_{P2}C_{in2} = C_{g2} + C_{P2}C_{g1} + C_{p2}C_{p1}C_{in1}$$

Full-adder 3:

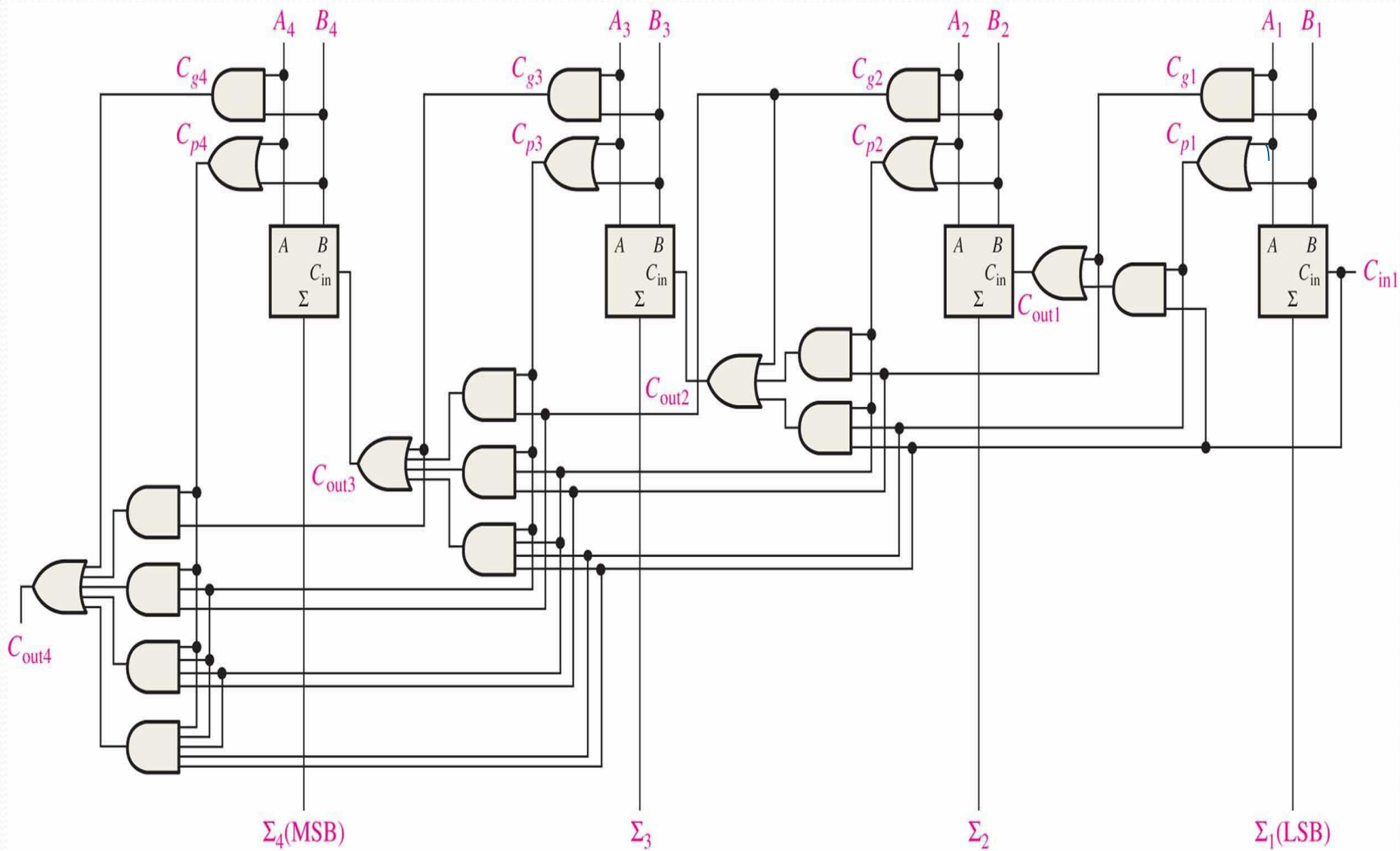
$$C_{out3} = C_{g3} + C_{P3}C_{in3} = C_{g3} + C_{P3}C_{g2} + C_{p3}C_{p2}C_{g1} + C_{p3}C_{p2}C_{p1}C_{in1}$$

Full-adder 4:

$$C_{out4} = C_{g4} + C_{P4}C_{in4}$$

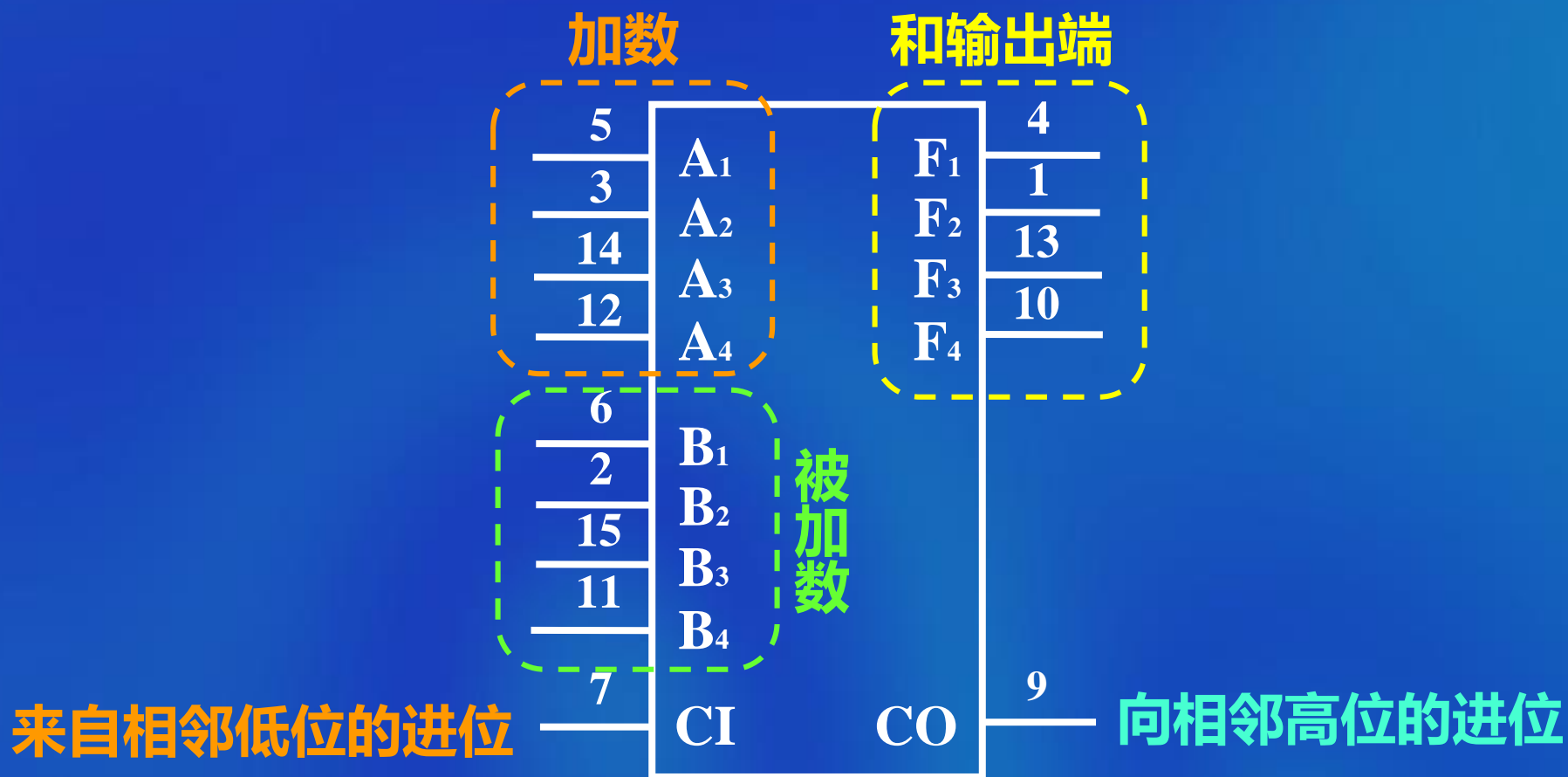
$$= C_{g4} + C_{P4}C_{g3} + C_{p4}C_{p3}C_{g2} + C_{p4}C_{p3}C_{p2}C_{g1} + C_{p4}C_{p3}C_{p2}C_{p1}C_{in1}$$

## CLA: carry look-ahead (进位超前) 部件





# 4位超前进位全加器74LS283的符号图



## 加法器的应用

加法器除可构成加法运算电路外，还可构成减法器、乘法器和除法器等多种运算电路。

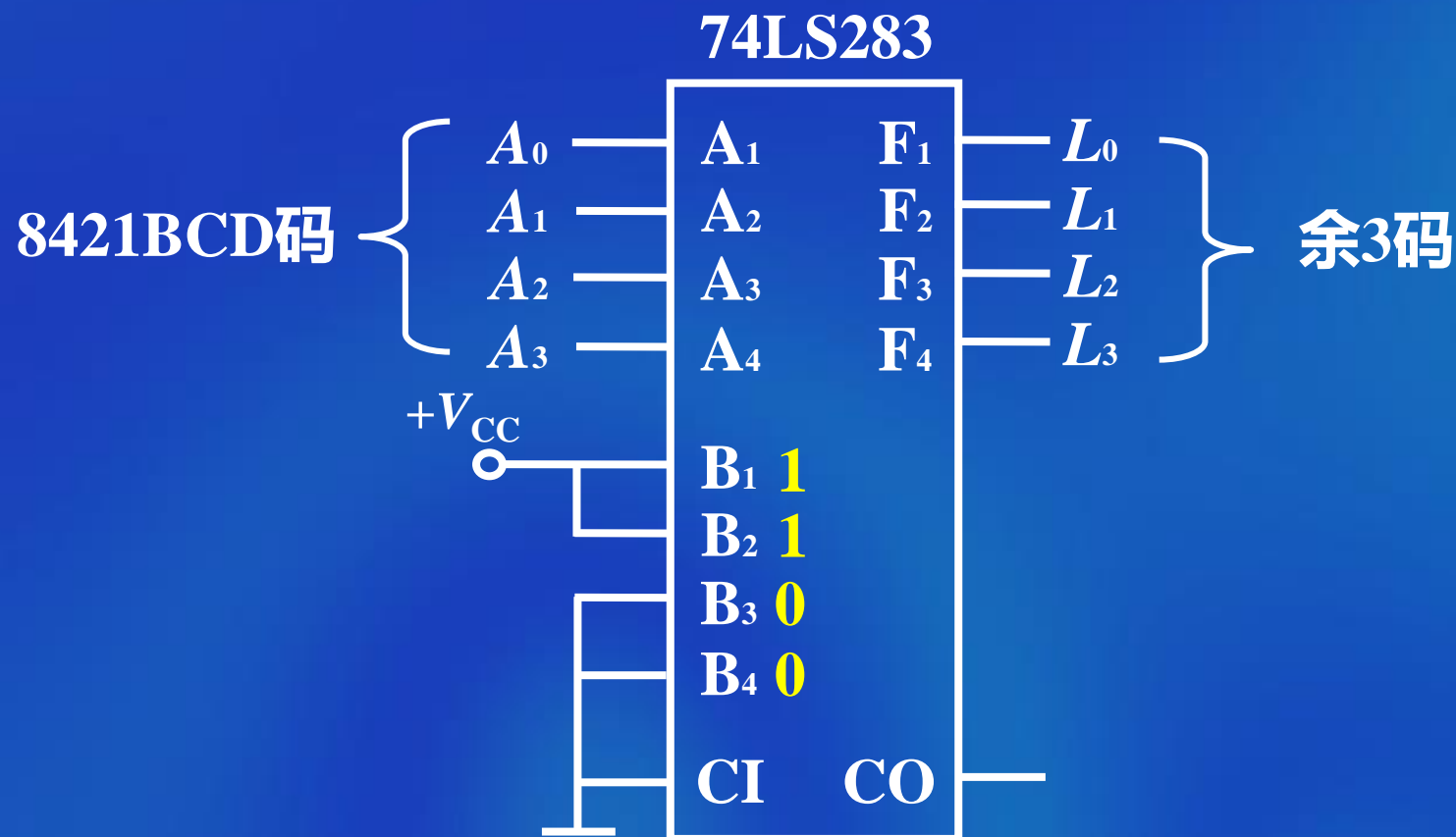
[例] 设计将8421BCD码转换为余3码的逻辑电路。

[解] 由于 余3码 = 8421BCD + 0011

故 输出逻辑表式

$$L_3L_2L_1L_0 = A_3A_2A_1A_0 + 0011$$

# 逻辑电路



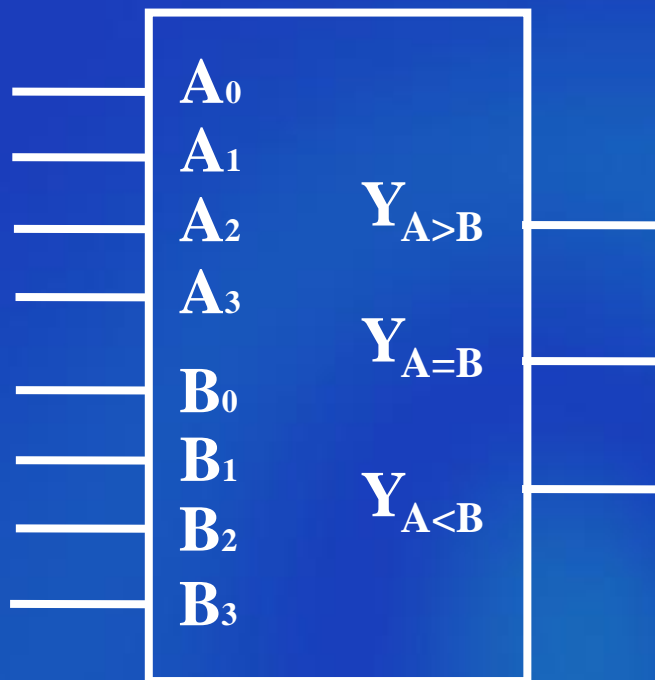


## 数值比较器

在数字系统和计算机中，经常需要比较两个数的大小或是否相等，完成这一功能的逻辑电路称为数值比较电路，相应的器件称为比较器(Digital Comparator)。

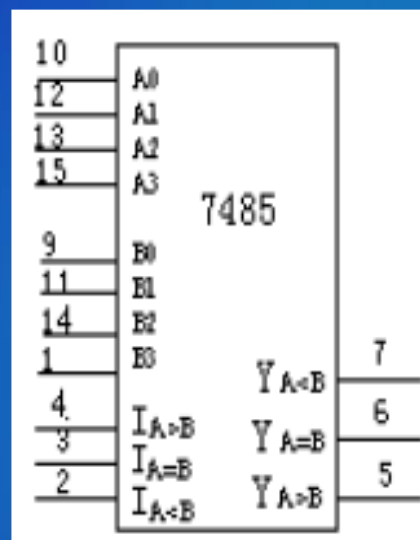
## 四位数值比较器

### 功能框图



其中,  $A_3 \sim A_0$ 、 $B_3 \sim B_0$ 是相比较的两组4位二进制数的输入端,  $Y_{A<B}$ 、 $Y_{A=B}$ 、 $Y_{A>B}$ 是比较结果输出端。

## 4位MSI比较器--74LS85

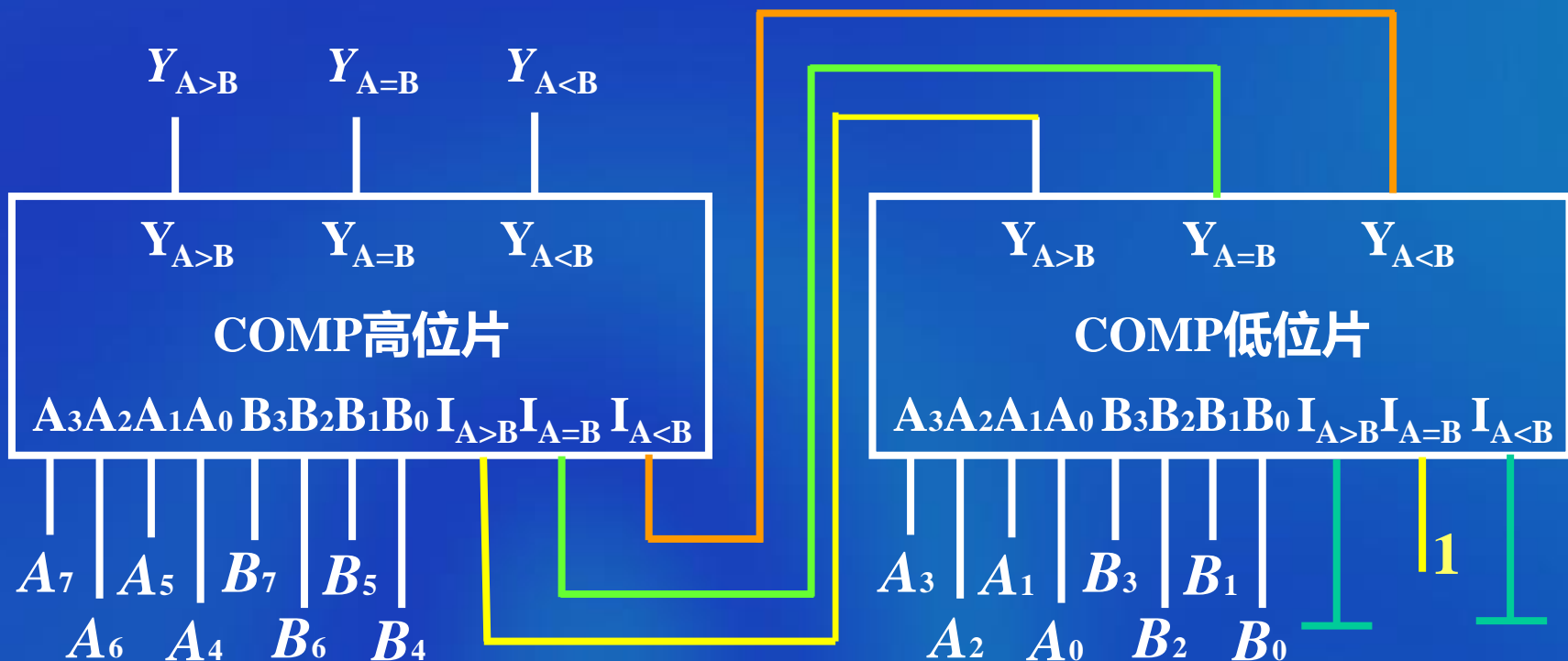


级联时低位片：

 $I_{A<B}=0,$  $I_{A>B}=0$  $I_{A=B}=1。$ 

比 较 输 入				级 联 输 入			输 出		
$A_3B_3$	$A_2B_2$	$A_1B_1$	$A_0B_0$	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$Y_{A>B}$	$Y_{A<B}$	$Y_{A=B}$
$A_3 > B_3$	x	x	x	x	x	x	1	0	0
$A_3 < B_3$	x	x	x	x	x	x	0	1	0
$A_3 = B_3$	$A_2 > B_2$	x	x	x	x	x	1	0	0
$A_3 = B_3$	$A_2 < B_2$	x	x	x	x	x	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	x	x	x	x	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	x	x	x	x	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	x	x	x	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	x	x	x	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1	0	0	1

# 74LS85的扩展



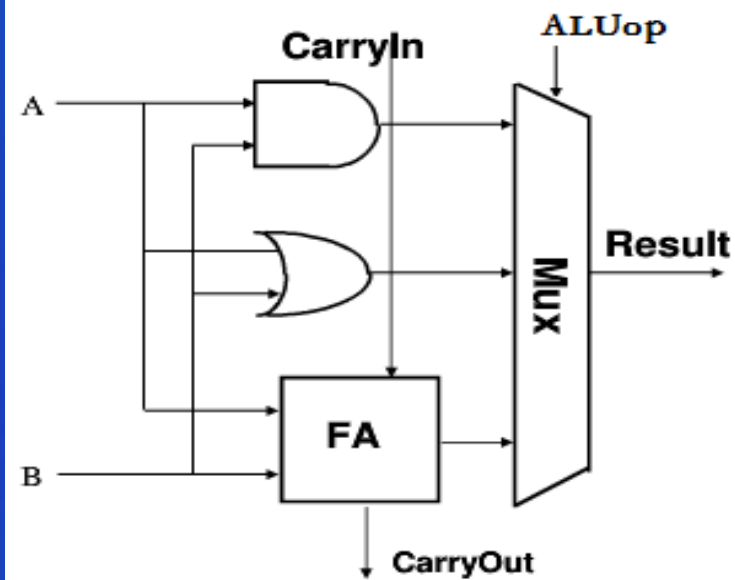
## 7.5 算术/逻辑运算单元(ALU)

**CPU:** 是微处理器数字系统的核心。负责从存储器内读取程序指令，通过算术逻辑单元 (Arithmetic-Logic Unit, ALU) 计算结果并保存到存储器，同时通过接口电路与外界交换数据。

**算术逻辑单元 (ALU):** 是CPU的执行单元和核心，主要完成算术与逻辑运算，即加、减、乘、除算术运算以及比较、判断等逻辑运算。

现代的计算机或者微处理器中甚至有多CPU核，一个核中有多个ALU。

## 1-bit ALU

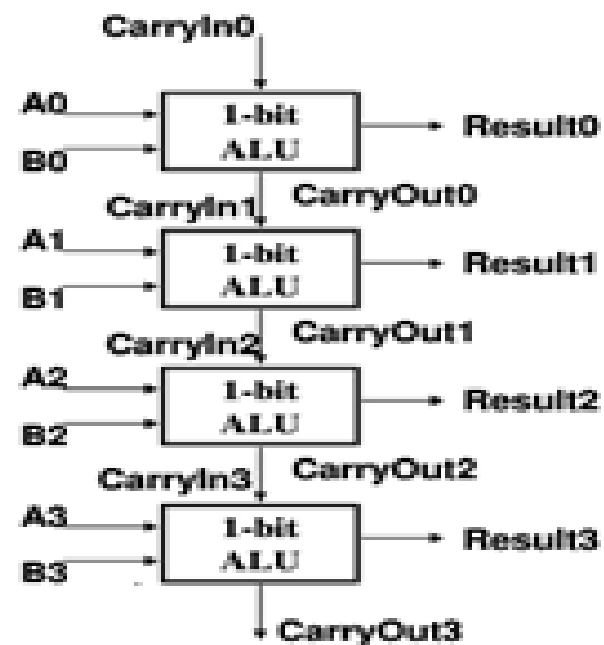


由门、全加器FA和多路选择器构成的一位ALU的运算电路。

根据来自CPU的指令操作码ALUop, 可以实现输入A和B的与、或运算, 以及A和B的算术求和运算。

ALU串联构成4位ALU。速度慢。

## 4-bit 串行 ALU



# 总结MSI的学习方法与使用

任何器件，其pdf文件提供最详细的器件信息。

以**DM74LS47**为例

- 1、名称BCD to 7-Segment Decoder/Driver with Open-Collector Outputs
- 2、符号图或原理图 (Logic Symbols)
- 3、功能表（使能端无效时输出状态；输入的高低顺序，比如138的ABC输入等细节）
- 4、个别引脚不理解看内部门电路
- 5、实验验证

# 作业

## 自练题:

- 7.10
- 7.13

## 作业题:

- 7.8



## 本章小结

### ● 组合电路

任何时刻的输出仅决定于当时的输入，而与电路原来的状态无关；它由基本门构成，不含存贮电路和记忆元件，且无反馈线。

### ● 组合电路的分析

根据已经给定的逻辑电路，描述其逻辑功能。

### ● 组合电路的设计

根据设计要求构成功能正确、经济、可靠的电路

### ● 常用的中规模组合逻辑器件及应用

编码器、译码器、数据选择器、比较器和加法器等。

## 重点难点

重点:

- ◆ 组合逻辑电路的概念
- ◆ 常用组合模块的功能及应用
- ◆ MSI组合逻辑电路分析与设计方法

难点:

- ◆ 灵活运用模块进行电路设计