

2.5 定点运算器的组成

提纲

2.5.1

逻辑运算

2.5.2

多功能算术/逻辑运算单元(ALU)

2.5.3

内部总线

2.5.4

定点运算器的基本结构



2.5.1 逻辑运算

- 计算机中除了进行加、减、乘、除等基本算术运算外，还可对两个或一个逻辑数进行逻辑运算
- 所谓逻辑数，是指不带符号的二进制数。利用逻辑运算可以进行两个数的比较，或者从某个数中选取某几位等操作
- 计算机中的逻辑运算，主要是指逻辑非、逻辑加、逻辑乘、逻辑异四种基本运算。



2.5.1 逻辑运算

■ 1. 逻辑非运算

- 逻辑非也称求反。对某数进行逻辑非运算，就是按位求它的反，常用变量上方加一横来表示。

■ 2. 逻辑加运算

- 两数进行逻辑加，就是按位求它们的“或”，所以逻辑加又称逻辑或，常用记号“V”或“+”来表示。

■ 3. 逻辑乘运算

- 两数逻辑乘，就是按位求它们的“与”，所以逻辑乘又称“逻辑与”，常用记号“^”或“.”来表示。

■ 4. 逻辑异运算

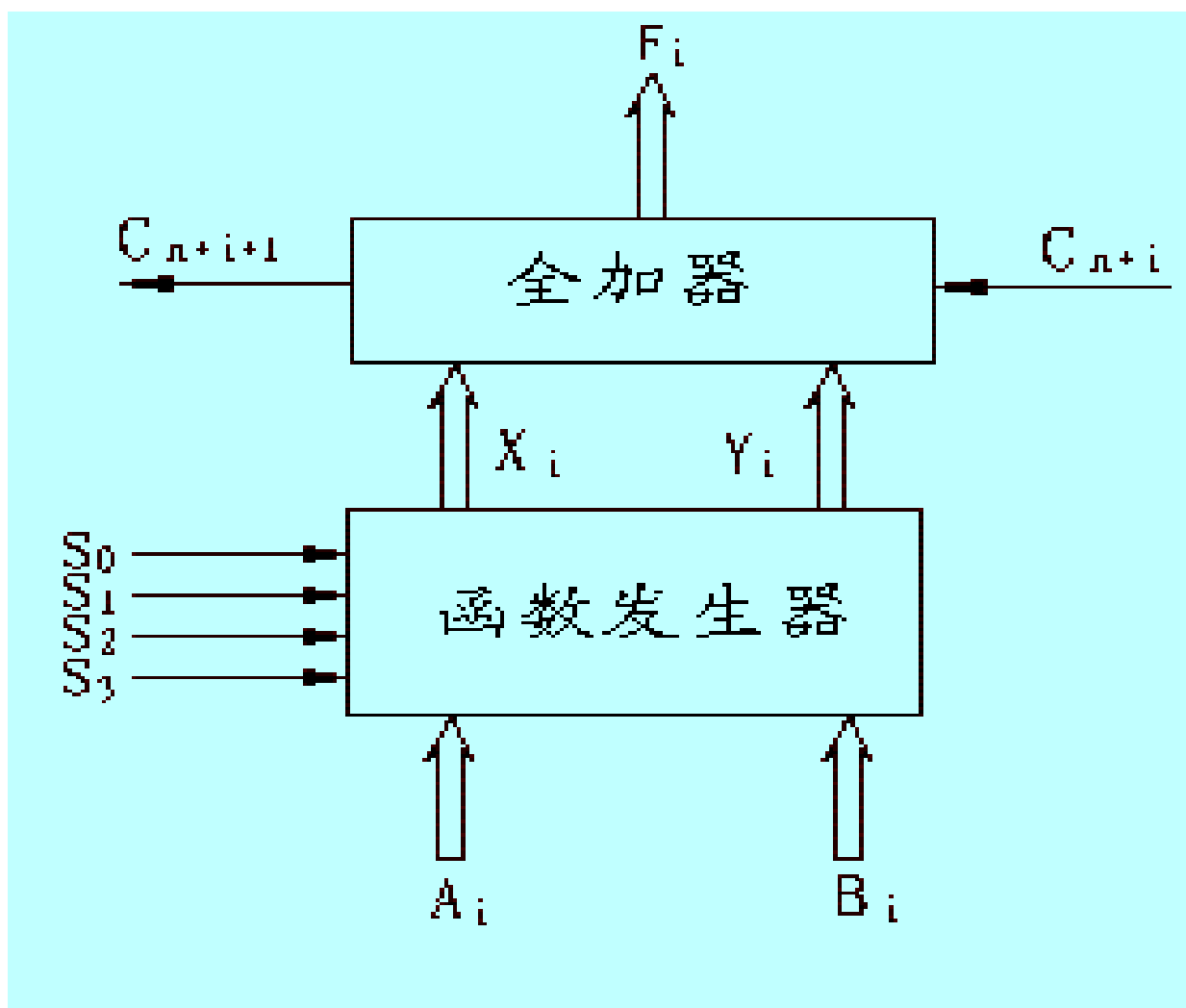
- 对两数进行异或就是按位求它们的模2和，所以逻辑异又称“按位加”，常用记号“ \oplus ”表示。

2.5.2 多功能算术/逻辑运算单元(ALU)

- 由一位全加器(FA)构成的行波进位加法器，它可以实现补码数的加法或减法运算。但是这种加法/减法器存在两个问题：一是由于串行进位它的运算时间很长。假如加法器由 n 位全加器构成，每一位的进位延迟时间为 20ns ，那么最坏情况下，进位信号从最低位传递到最高位而最后输出稳定至少需要 $n \times 20\text{ns}$ ，这在高速计算中显然是不利的。二是就行波进位加法器本身来说，它只能完成加法或减法两种操作而不能完成逻辑操作。
- 本节我们介绍的多功能算术/逻辑运算单元(ALU)不仅具有多种算术运算和逻辑运算的功能，而且具有先行进位逻辑，从而能实现高速运算。

2.5.2 多功能算术/逻辑运算单元(ALU)

■ ALU的逻辑结构原理框图



2.5.2 多功能算术/逻辑运算单元(ALU)

■ 一、基本思想

- 一位全加器(FA)的逻辑表达式为

$$F_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = A_i B_i + B_i C_i + C_i A_i$$

- 我们将 A_i 和 B_i 先组合成由控制参数 S_0, S_1, S_2, S_3 控制的组合函数 X_i 和 Y_i , 然后再将 X_i, Y_i 和下一位进位数通过全加器进行全加
- 不同的控制参数可以得到不同的组合函数, 因而能够实现多种算术运算和逻辑运算。

2.5.2 多功能算术/逻辑运算单元(ALU)

■ 一、基本思想

- 一位算术/逻辑运算单元的逻辑表达式为：

$$F_i = X_i \oplus Y_i \oplus C_{n+i}$$

$$\begin{aligned} C_{n+i+1} &= X_i Y_i + Y_i C_{n+i} + C_{n+i} X_i \\ &= X_i Y_i + (X_i \oplus Y_i) C_{n+i} \end{aligned}$$

- 上式中进位下标用 $n + i$ 代替原来一位全加器中的 i ， i 代表集成在一片电路上的ALU的二进制位数。对于4位一片的ALU， $i = 0, 1, 2, 3$ 。 n 代表若干片ALU组成更大字长的运算器时每片电路的进位输入，例如当4片组成16位字长的运算器时， $n = 0, 4, 8, 12$ 。

2.5.2 多功能算术/逻辑运算单元(ALU)

■ 二、逻辑表达式

- 控制参数 S_0, S_1, S_2, S_3 分别控制输入 A_i 和 B_i ，产生 Y 和 X 的函数。其中 Y_i 是受 S_0, S_1 控制的 A_i 和 B_i 的组合函数，而 X_i 是受 S_2, S_3 控制的 A_i 和 B_i 组合函数，其函数关系如下表所示。

$S_0 \ S_1$	Y_i	$S_2 \ S_3$	X_i
0 0	\bar{A}_i	0 0	1
0 1	$\bar{A}_i B_i$	0 1	$\bar{A}_i + \bar{B}_i$
1 0	$\bar{A}_i \bar{B}_i$	1 0	$\bar{A}_i + B_i$
1 1	0	1 1	\bar{A}_i

X_i, Y_i 与控制参数和输入量的关系

2.5.2 多功能算术/逻辑运算单元(ALU)

- 4位之间采用先行进位公式，根据上式，每一位的进位公式可递推如下：

- 第0位向第1位的进位公式为

$$C_{n+1} = Y_0 + X_0 C_n \quad \text{其中 } C_n \text{ 是向第0位(末位)的进位。}$$

- 第1位向第2位的进位公式为

$$C_{n+2} = Y_1 + X_1 C_{n+1} = Y_1 + Y_0 X_1 + X_0 X_1 C_n$$

- 第2位向第3位的进位公式为

$$C_{n+3} = Y_2 + X_2 C_{n+2} = Y_2 + Y_1 X_2 + Y_0 X_1 X_2 + X_0 X_1 X_2 C_n$$

- 第3位的进位输出(即整个4位运算进位输出)公式为

$$C_{n+4} = Y_3 + X_3 C_{n+3} = Y_3 + Y_2 X_3 + Y_1 X_2 X_3 + Y_0 X_1 X_2 X_3 + X_0 X_1 X_2 X_3 C_n$$

- 设 $G = Y_3 + Y_2 X_3 + Y_1 X_2 X_3 + Y_0 X_1 X_2 X_3$, $P = X_0 X_1 X_2 X_3$

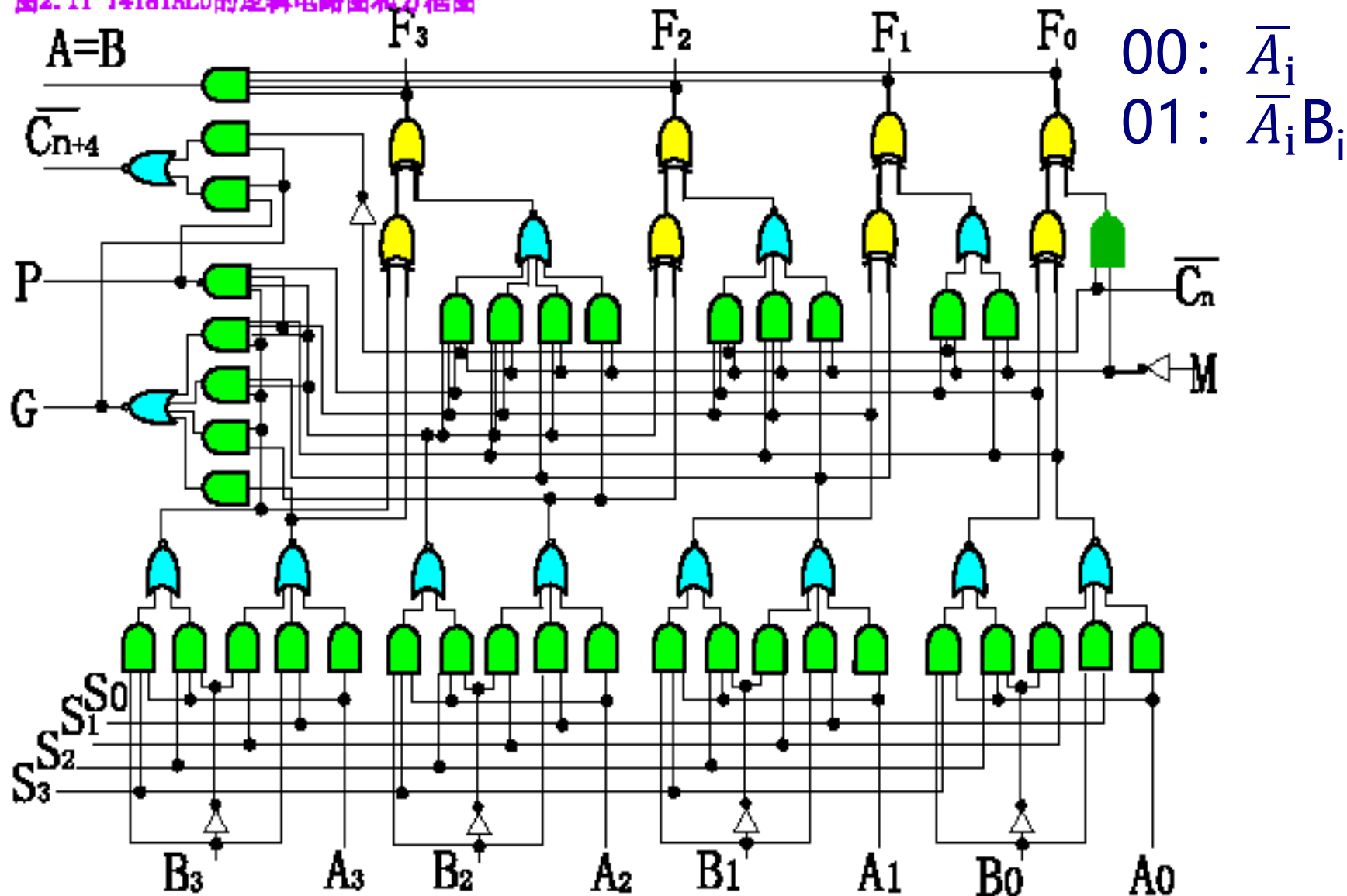
- 则 $C_{n+4} = G + PC_n$

2.5.2 多功能算术/逻辑运算单元(ALU)

- 这样对一片ALU来说，可有三个进位输出。其中G称为进位发生输出，P称为进位传送输出。在电路中多加这两个进位输出的目的，是为了便于实现多片(组)ALU之间的先行进位，为此还需一个配合电路称之为先行进位发生器(CLA)。
- C_{n+4} 是本片(组)的最后进位输出。逻辑表达式表明，这是一个先行进位逻辑。换句话说第0位的进位输入 C_n 可以直接传送到最高位上去，因而可以实现高速运算。
- 用正逻辑表示的4位算术/逻辑运算单元(ALU)的逻辑电路图如下，它是根据上面的原始推导公式用TTL电路实现的。这个期间的商业标号为74181ALU。

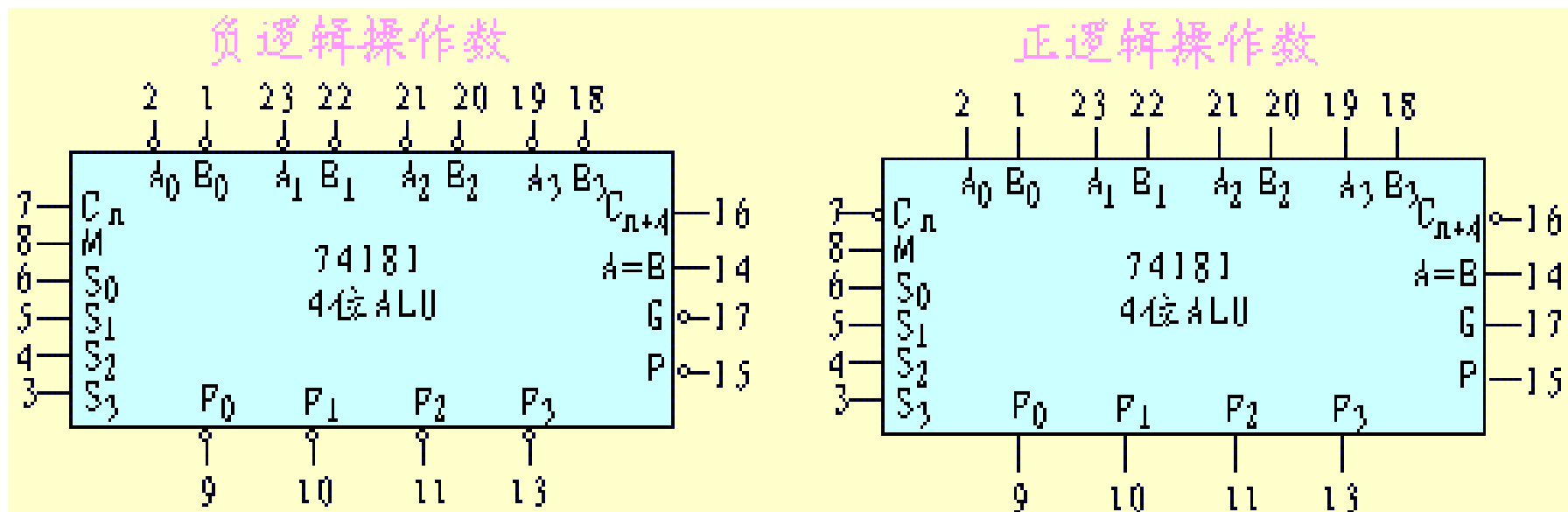
2.5.2 多功能算术/逻辑运算单元(ALU)

图2.11 74181ALU的逻辑电路图和方框图



2.5.2 多功能算术/逻辑运算单元(ALU)

- 下图示出了工作于负逻辑和正逻辑操作数方式的74181ALU方框图



(b) 负逻辑或正逻辑操作数方式的74181ALU方框图

- 。表示低电平有效

2.5.2 多功能算术/逻辑运算单元(ALU)

- 书中第47页的表2.5列出了74181ALU的运算功能表
- 它有两种工作方式。对正逻辑操作数来说，算术运算称高电平操作，逻辑运算称正逻辑操作(即高电平为“1”，低电平为“0”)。对于负逻辑操作数来说，正好相反
- 由于 $S_0 \sim S_3$ 有16种状态组合，因此对正逻辑输入与输出而言，有16种算术运算功能和16种逻辑运算功能。同样对于负逻辑输入与输出而言，也有16种算术运算功能和16种逻辑运算功能。

2.5.2 多功能算术/逻辑运算单元(ALU)

- 四、两级先行进位的ALU
- 前面说过, 74181ALU设置了P和G两个本组先行进位输出端。如果将四片74181的P,G输出端送入到74182先行进位部件(CLA), 又可实现第二级的先行进位, 即组与组之间的先行进位。
- 假设4片(组)74181的先行进位输出依次为 $P_0, G_0, G_1, P_1, P_2, G_2, P_3, G_3$, 那么参考式(2.37)的进位逻辑表达式, 先行进位部件74182CLA所提供的进位逻辑关系如下:

2.5.2 多功能算术/逻辑运算单元(ALU)

- $C_{n+x} = G_0 + P_0 C_n$
- $C_{n+y} = G_1 + P_1 C_{n+x} = G_1 + G_0 P_1 + P_0 P_1 C_n$
- $C_{n+z} = G_2 + P_2 C_{n+y} = G_2 + G_1 P_2 + G_0 P_1 P_2 + P_0 P_1 P_2 C_n$
- $C_{n+4} = G_3 + P_3 C_{n+z}$
 $= G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3 + P_0 P_1 P_2 P_3 C_n$
 $= G^* + P^* C_n$

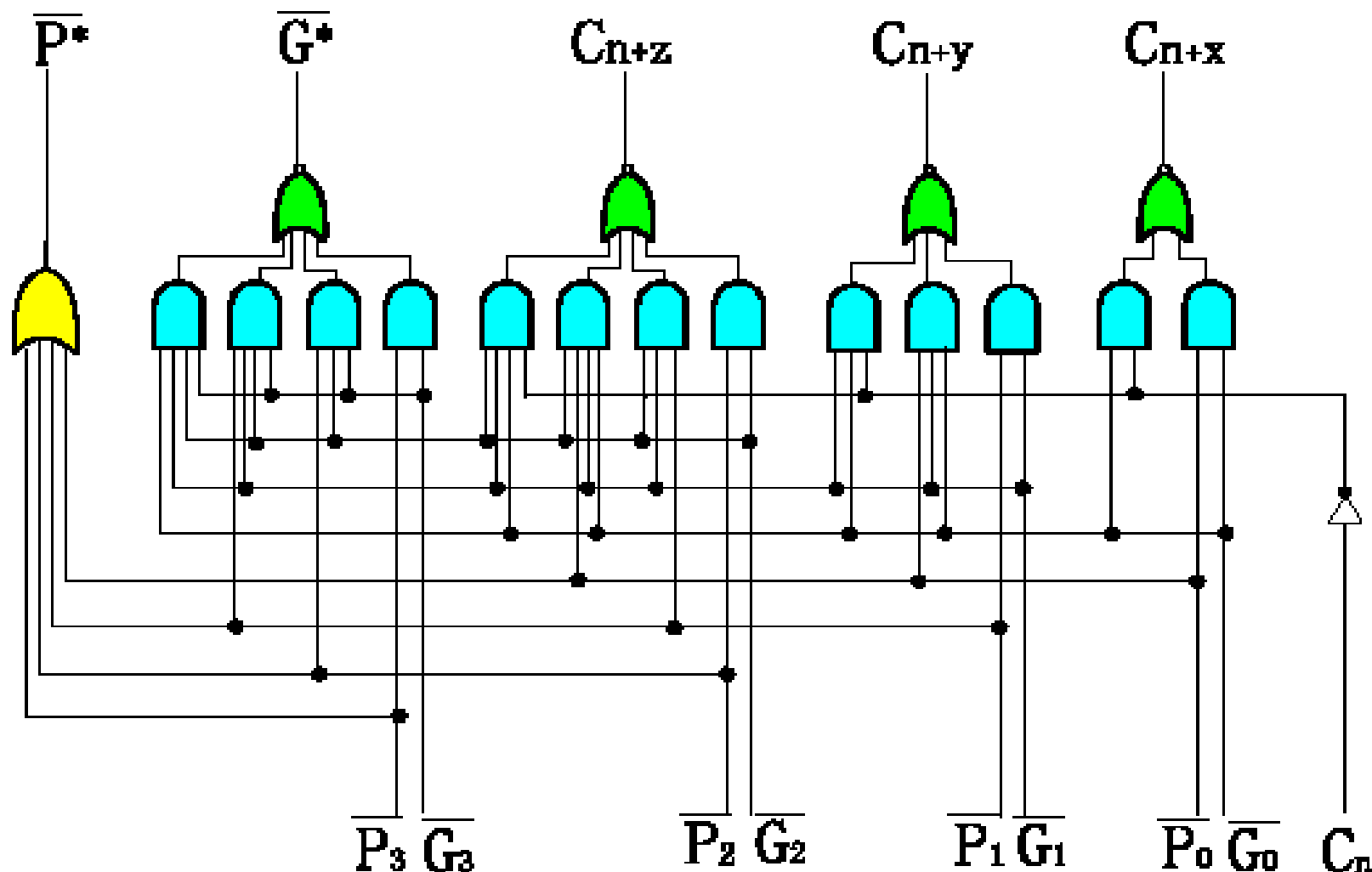
- 其中

$$P^* = P_0 P_1 P_2 P_3$$

$$G^* = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3$$

- 根据以上表达式，用TTL器件实现的成组先行进位部件74182的逻辑电路图如下，其中 G^* 称为成组进位发生输出， P^* 称为成组进位传送输出。

2.5.2 多功能算术/逻辑运算单元(ALU)



$$P^* = P_0 P_1 P_2 P_3$$

$$G^* = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0$$

图2.12 成组先行进位部件CLA的逻辑电路图

2.5.2 多功能算术/逻辑运算单元(ALU)

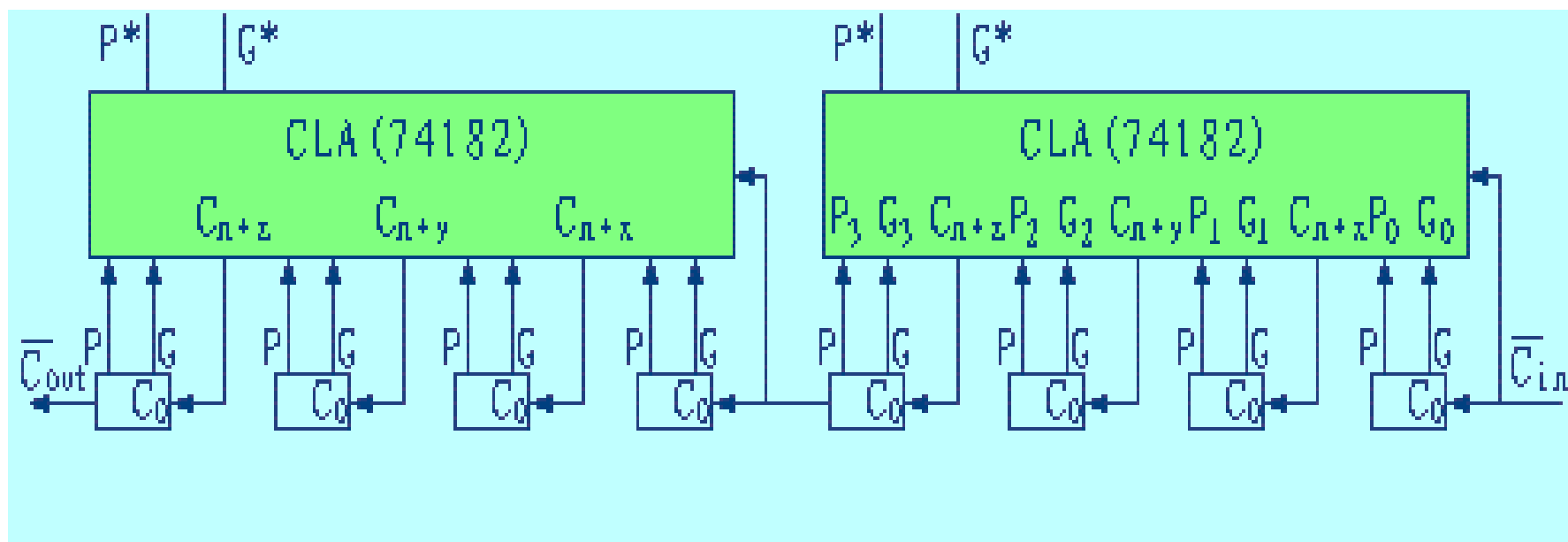
$$G_0 + P_0 C_n$$

$$\overline{P_0} * \overline{G_0} + \overline{C_n} * \overline{G_0}$$

G_0	P_0	C_n	C_{n+1}		$\overline{G_0}$	$\overline{P_0}$	$\overline{G_0}$	C_{n+1}
0	0	0	0		1	1	1	0
0	0	1	0		1	1	0	0
0	1	0	0		1	0	1	0
0	1	1	1		1	0	0	1
1	0	0	1		0	1	1	1
1	0	1	1		0	1	0	1
1	1	0	1		0	0	1	1
1	1	1	1		0	0	0	1

2.5.2 多功能算术/逻辑运算单元(ALU)

- 下图示出了用两个16位全先行进位部件级联组成的32位ALU逻辑方框图。在这个电路中使用了八个74181ALU和两个74182CLA器件。很显然对一个16位来说，CLA部件构成了第二级的先行进位逻辑，即实现四个小组(位片)之间的先行进位，从而使全字长ALU的运算时间大大缩短。



2.5.3 内部总线

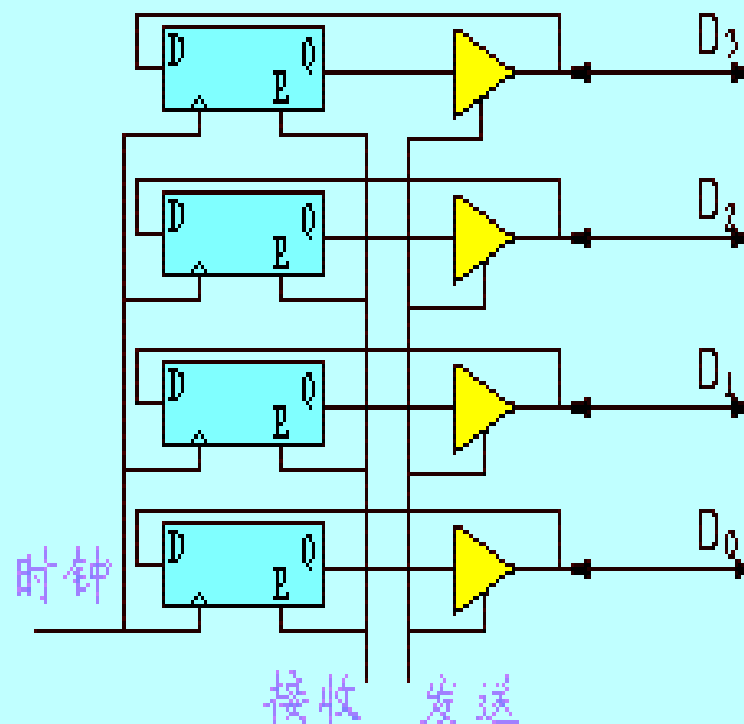
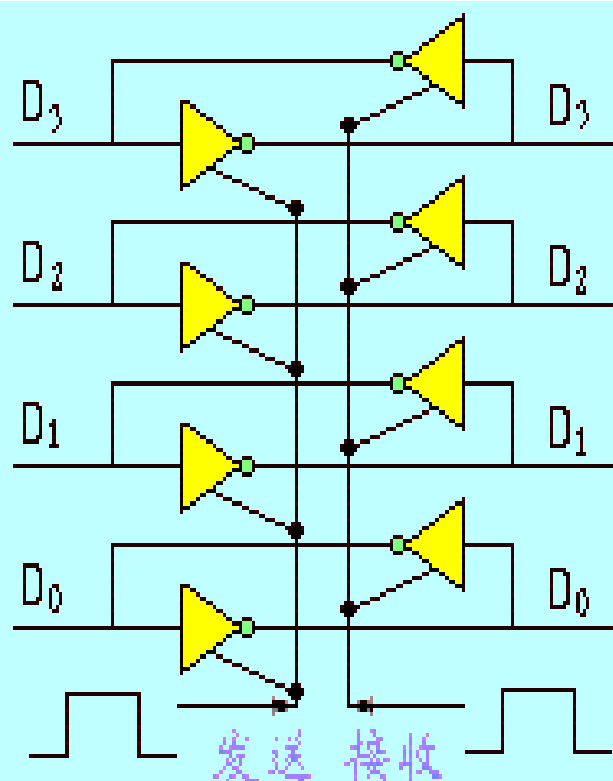
- 由于计算机内部的主要工作过程是信息传送和加工的过程，因此在机器内部各部件之间的数据传送非常频繁。为了减少内部的传送线并便于控制，通常将一些寄存器之间数据传送的通路加以归并，组成总线结构，使不同来源的信息在此传输线上分时传送。
- 根据总线所在位置，总线分为内部总线和外部总线两类。**内部总线**是指CPU内各部件的连线，而**外部总线**是指系统总线，即CPU与存储器、I/O系统之间的连线。
- 本节只讨论内部总线。



2.5.3 内部总线

- 按总线的逻辑结构来说，总线可分为单向传送总线和双向传送总线。所谓**单向总线**就是信息只能向一个方向传送。所谓**双向总线**就是信息可以分两个方向传送，既可以发送数据，也可以接收数据。
- 下图(a)是带有缓冲驱动器的4位双向数据总线。其中所用的基本电路就是三态逻辑电路。当“发送”信号有效时，数据从左向右传送。反之当“接收”信号有效时，数据从右向左传送。这种类型的缓冲器通常根据它们如何使用而叫作总线扩展器、总线驱动器、总线接收器等等。

2.5.3 内部总线



(a) 带有缓冲器的双向数据总线 (b) 带有锁存器的4位双向数据总线

由三态门组成的双向数据总线

2.5.3 内部总线

- 上图 (b)中所示的是带有锁存器的4位双向数据总线。它主要由一个DE触发器和一个三态缓冲器组成。DE触发器是在一个普通D触发器上另加一个E输入端（允许端）而构成的。此处E输入端用以控制D的输入。若 $E = 0$ ，即使D为“1”，也不能输入。当接收数据时， $E = 1$ 三态门被禁止，因而数据总线上的数据被接收到锁存器。当发送数据时， $E = 0$ ，三态门被允许，因而锁存器的数据发送至数据总线上。

2.5.4 定点运算器的基本结构

- 运算器包括ALU\阵列乘除器\寄存器\多路开关\三态缓冲器\数据总线等逻辑部件。
- 运算器的设计，主要是围绕ALU和寄存器同数据总线之间如何传送操作数和运算结果进行的。
- 在决定方案时，需要考虑数据传送的方便性和操作速度，在微型机和单片机中还要考虑在硅片上制作总线的工艺。计算机的运算器大体有如下三种结构形式：



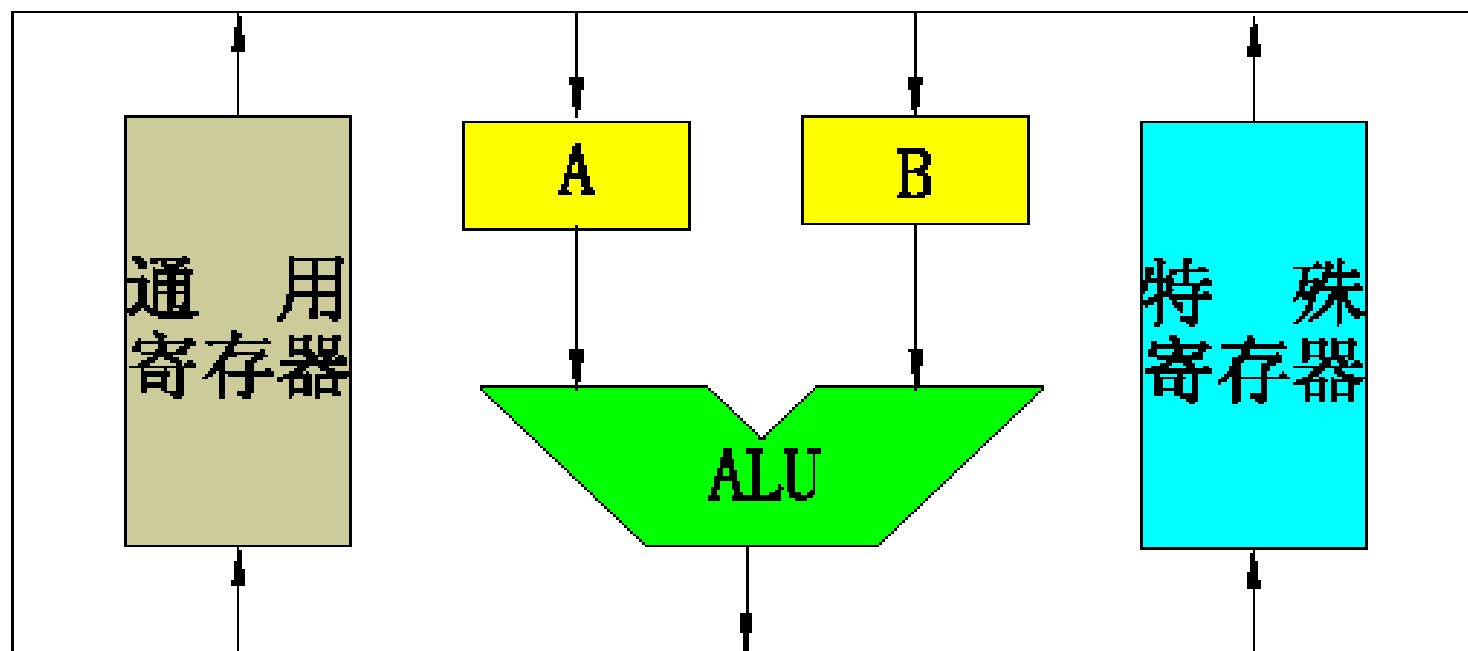
2.5.4 定点运算器的基本结构

■ 一、单总线结构的运算器

- 单总线结构的运算器如下图所示。由于所有部件都接到同一总线上，所以数据可以在任何两个寄存器之间，或者在任一个寄存器和ALU之间传送。如果具有阵列乘法器或除法器，那么它们所处的位置应与ALU相当。对这种结构的运算器来说，在同一时间内，只能有一个操作数放在单总线上。为了把两个操作数输入到ALU，需要分两次来做，而且还需要A, B两个缓冲寄存器。这种结构的主要缺点是操作速度较慢，但是由于它只控制一条总线，故控制电路比较简单。

2.5.4 定点运算器的基本结构

单总线



(a) 单总线结构的运算器

对单总线结构的运算器来说，在同一个时刻，只能有一个操作数放在单总线上，为了将两个操作数输入到ALU，需分两次来做，而且需要两个缓冲寄存器。例如执行一个加法操作，第一个操作数先放入A缓冲寄存器，然后把第二个操作数放入B缓冲寄存器。然后，ALU执行加法。当结果出现在单总线上时，由第三个传送动作把加法的结果选通到目的寄存器。由此可见，单总线结构操作速度慢，但控制电路较简单。

图2.15 运算器的三种基本结构形式

2.5.4 定点运算器的基本结构

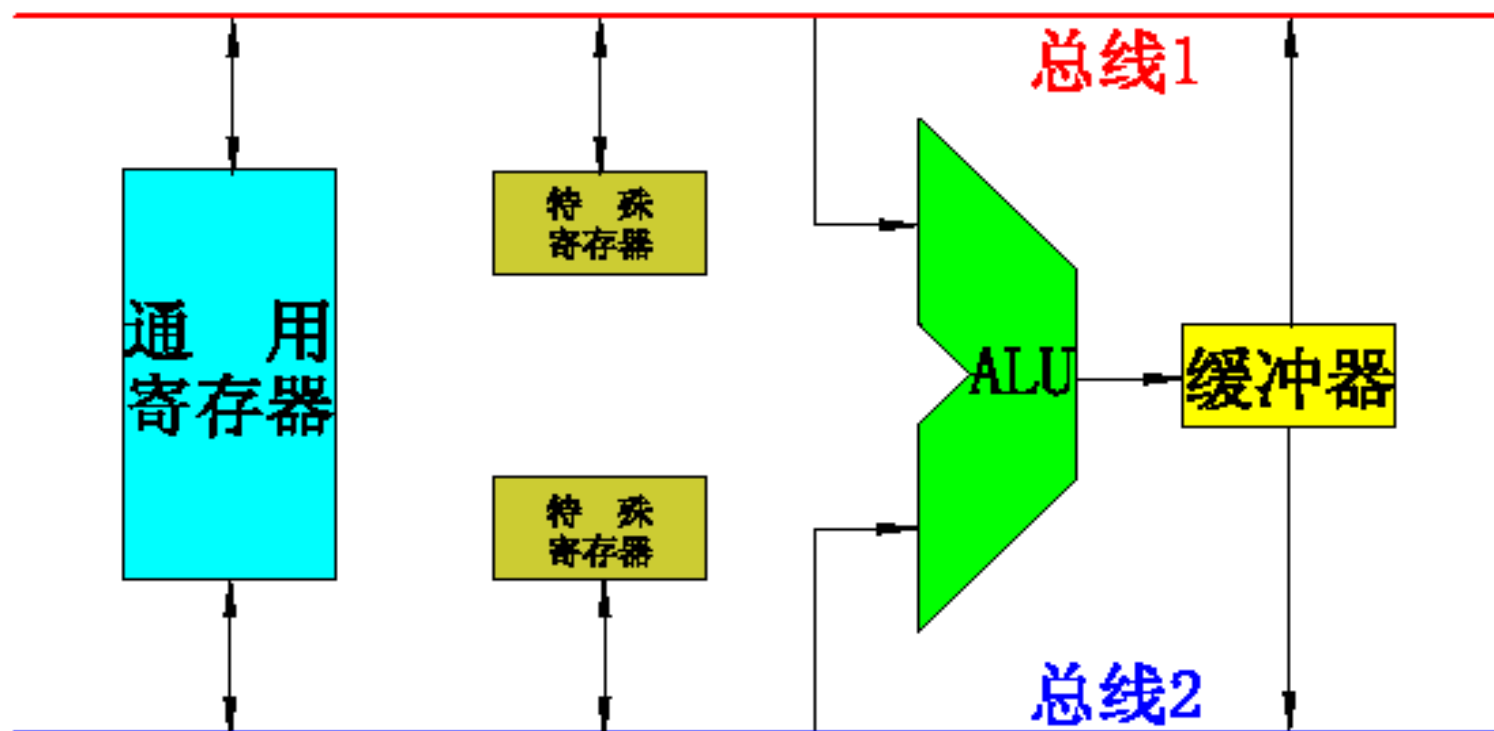
■ 二、双总线结构的运算器

- 双总线结构的运算器如下图所示。在该结构中，两个操作数同时加到ALU进行运算，只需一次操作控制，而且马上就可以得到运算结果。图中两条总线各自把其数据送至ALU的输入端。特殊寄存器分为两组，它们分别与一条总线交换数据。这样通用寄存器中的数就可进入到任一组特殊寄存器中去，从而使数据传送更为灵活。ALU的输出不能直接加到总线上去。这是因为当形成操作结果的输出时，两条总线都被输入数占据，因而必须在ALU输出端设置缓冲寄存器。为此操作的控制要分两步完成：

2.5.4 定点运算器的基本结构

- 第一步：在ALU的两个输入端输入操作数，形成结果并送入缓冲寄存器;
- 第二步：把结果送入目的寄存器
- 假如在总线1, 2和ALU输入端之间再各加一个输入缓冲寄存器，并把两个输入数先放至这两个缓冲寄存器，那么ALU输出端就可以直接把操作结果送至总线1或总线2上去。

2.5.4 定点运算器的基本结构



(b) 双总线结构的运算器

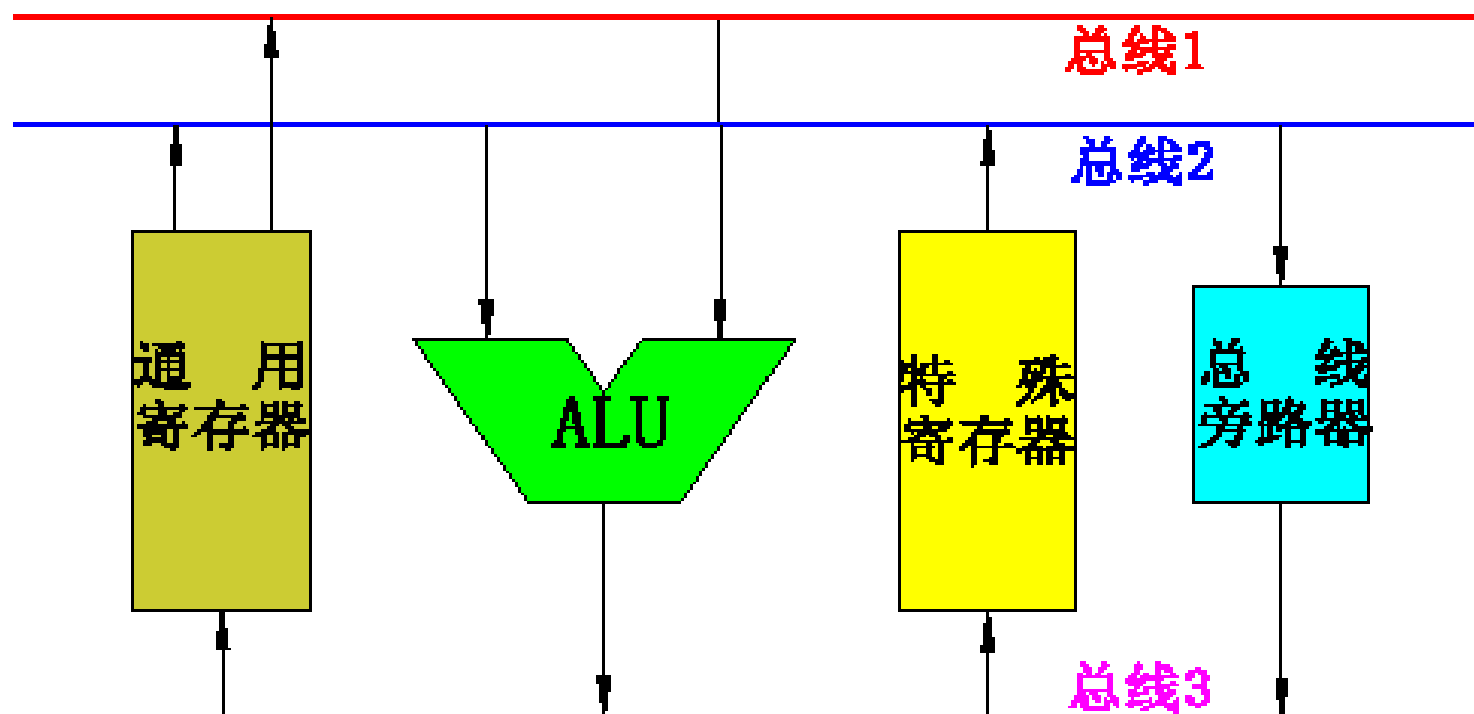
双总线结构中，两个操作数同时加到ALU进行运算，只需一次操作控制，而且马上就可以得到结果，两条总线各自把操作数送至ALU的输入端，ALU的输出端必须设置缓冲器，操作控制分两步：第一步，在ALU的两个输入端输入操作数，形成结果并送至缓冲器；第二步，把结果送入目的寄存器。

图2.15 运算器的三种基本结构形式

2.5.4 定点运算器的基本结构

- 三、三总线结构的运算器
- 三总线结构的运算器如下图所示。在三总线结构中，ALU的两个输入端分别由两条总线供给，而ALU的输出则与第三条总线相连。这样算术逻辑操作就可以在一步控制之内完成。
- 另外设置了一个总线旁路器。如果一个操作数不需要修改而直接从总线2传送到总线3，那么可以通过控制总线旁路器把数据传出；如果一个操作数传送时需要修改那么就借助于ALU。很显然三总线结构的运算器的特点是操作时间快。

2.5.4 定点运算器的基本结构



(c) 三总线结构的运算器

在三总线结构中，ALU的两个输入端分别由两条总线供给，而ALU的输出则和第三条总线相连。总线旁路器的作用是如果一个操作数不需修改，那么可直接通过总线旁路器从总线2传送到总线3；如果需要修改，那么就借助于ALU。

图2.15 运算器的三种基本结构形式