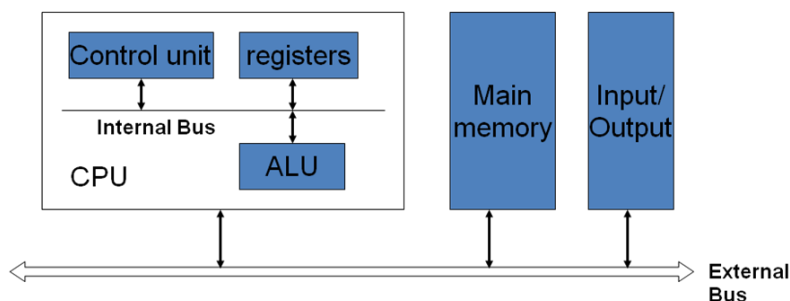


第 5 章 算术逻辑单元

(一) 冯诺依曼模型



The basic organization of a stored-program computer

图 5-1 冯诺依曼模型

现今使用中的大多数计算机系统都是在冯 诺依曼计算机模型上构造的。该模型于 1946 年由冯 诺依曼提出。该计算机被看作是一个存储程序计算机。一道程序是一个指令序列，其中每一条指令执行一个基本操作。执行前，程序和将要由它加工的数据一起存放到存储器中。在程序执行中，它的指令一条一条地从存储器读出，送到处理单元中去。处理单元译码、取数，执行，并写回结果。因此冯 诺依曼机型典型组成包含：存储器，CPU，I/O。

冯诺依曼模型如图 5-1 所示。中央处理器组成为一个有限状态机，它的状态随指令执行的结果而改变。像 CPU 这样的有限状态机应包含一个存储部分和一个组合部分，前者是寄存器，后者是算术逻辑单元和控制器，用以实现指令的序列控制。

输入输出子系统作为计算机和外部世界之间的接口。

外总线让所有这些函数部件连接起来并相互通信。

(二) 并行快速加法器

1. 串行进位加法器

并行加法器可以同时数据的各位进行相加，一般用 n 个全加器来实现 2 个操作数的各位同时相加。其操作数的各位是同时提供的，由于进位是逐位形成，低位运算所产生的进位会影响高位的运算结果。

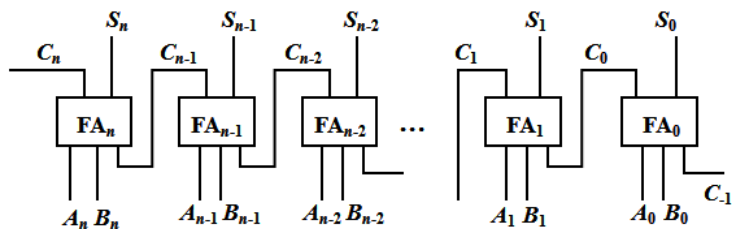


图 5-2 串行进位加法器

串行进位(也称波形进位)加法器, 逻辑电路比较简单, 但是最高位的加法运算, 一定要等到所有低位的加法完成之后才能进行, 低位的进位要逐步的传递到高位, 逐级产生进位, 因此运算速度比较慢。

2. 并行进位加法器

怎样才能提高加法器的工作速度? 只有通过改变进位逐位传送的路径来解决。解决办法之一是采用“超前进位产生电路”来同时形成各位进位, 从而实现快速加法。我们称这种加法器为超前进位加法器。超前进位产生电路是根据各位进位的形成条件来实现的。只要满足下述两条件中任一个, 就可形成 C_1 :

- (1) X_1, Y_1 均为 “1”;
- (2) X_1, Y_1 任一个为 “1”, 且进位 C_0 为 “1”。

由此, 可写得 C_1 的表达式为: $C_1 = X_1 Y_1 + (X_1 + Y_1) C_0$

只要满足下述条件中任一个即可形成 C_2 :

- (1) X_2, Y_2 均为 “1”;
- (2) X_2, Y_2 任一为 “1”, 且 X_1, Y_1 均为 “1”;
- (3) X_2, Y_2 任一为 “1”, 同时 X_1, Y_1 任一为 “1”, 且 C_0 为 “1”。

由此可得 C_2 表达式为: $C_2 = X_2 Y_2 + (X_2 + Y_2) X_1 Y_1 + (X_2 + Y_2) (X_1 + Y_1) C_0$

同理, 可有 C_3, C_4 表达式如下:

$$C_3 = X_3 Y_3 + (X_3 + Y_3) X_2 Y_2 + (X_3 + Y_3) (X_2 + Y_2) X_1 Y_1 + (X_3 + Y_3) (X_2 + Y_2) (X_1 + Y_1) C_0$$

$$C_4 = X_4 Y_4 + (X_4 + Y_4) X_3 Y_3 + (X_4 + Y_4) (X_3 + Y_3) X_2 Y_2 + (X_4 + Y_4) (X_3 + Y_3) (X_2 + Y_2) X_1 Y_1 + (X_4 + Y_4) (X_3 + Y_3) (X_2 + Y_2) (X_1 + Y_1) C_0$$

下面我们引入进位传递函数 P_i 和进位产生函数 G_i 的概念。它们的定义为:

$G_i = X_i Y_i$ 称为进位产生函数

$P_i = X_i + Y_i$ 称为进位传递函数

G_1 的意义是：当 X_1, Y_1 均为“1”时，不管有无进位输入，定会产生向高位的进位。
 P_1 的意义是：当 X_1, Y_1 中有一个为“1”时，若有进位输入，则本位向高位传送进位，这个进位可看成是低位进位越过本位直接向高位传递的。

将 $P_1、G_1$ 代入 $C_1 \sim C_4$ 式，便可得：

$$C_1 = G_1 + P_1 C_0 \quad (\text{低位})$$

$$C_2 = G_2 + P_2 G_1 + P_2 P_1 C_0$$

$$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_0$$

$$C_4 = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 C_0$$

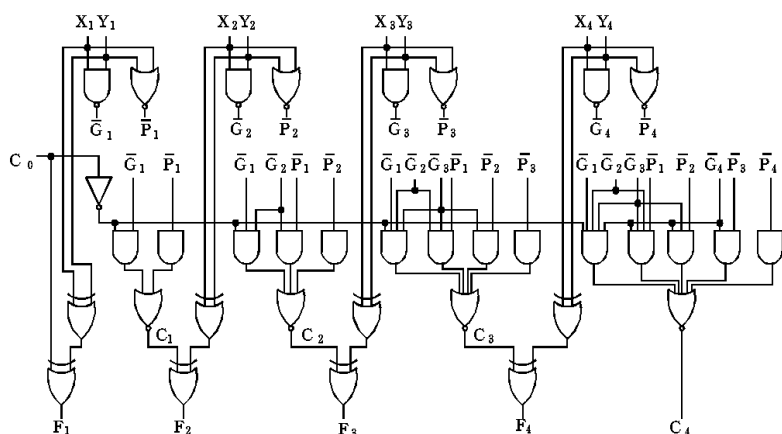


图 5-3 4 位超前进位加法器

并行进位加法器的运算速度很快，形成最高进位输出的延迟时间很短，但是以增加硬件逻辑线路为代价。对于长字长的加法器，往往将加法器分成若干组，在组内采用并行进位，组间则采用串行进位或并行进位，由此形成多种进位结构。

(1) 单级先行进位

单级先行进位方式将 n 位字长分为若干组，每组内采用并行进位方式，组与组之间采用串行进位方式。如图 5-4 所示。

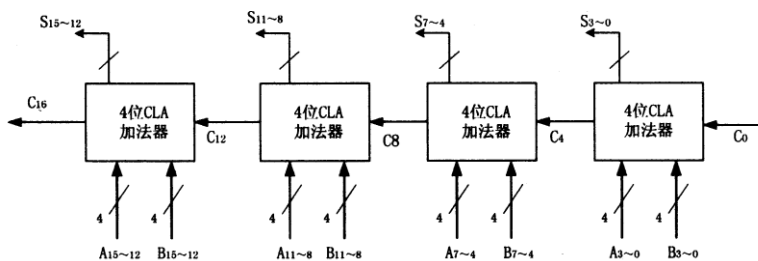
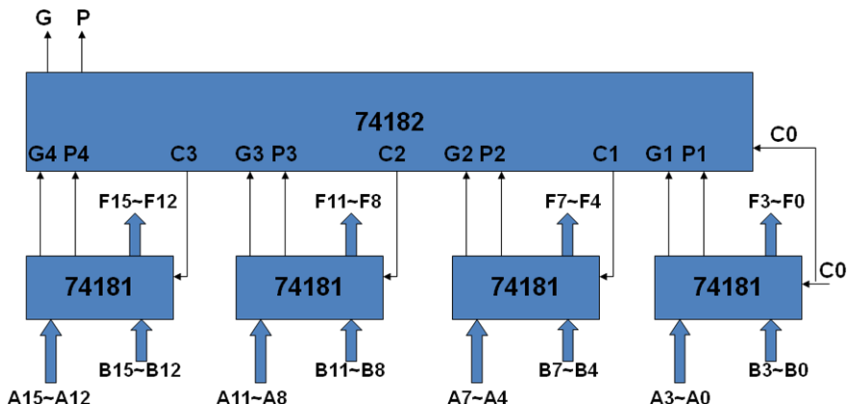


图 5-4 16 位单级先行进位加法器

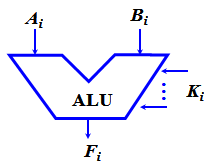
(2)多级先行进位

多级先行进位在组内和组间都采用先行进位方式。



(三) 算术逻辑单元 ALU

1. ALU 电路



组合逻辑电路
K_i 不同取值
F_i 不同

四位 ALU 74181

M = 0 算术运算
M = 1 逻辑运算
S₃ ~ S₀ 不同取值, 可做不同运算

ALU 是一种功能较强的组合逻辑电路。它能进行多种算术运算和逻辑运算。ALU 的基本逻辑结构是超前进位加法器, 它通过改变加法器的进位产生函数 G 和进位传递函数 P 来获得多种运算能力。

2. 算术逻辑单元 ALU 的功能和机构

ALU 部件是运算器中的主要组成部分, 又称多功能函数发生器, 主要用于完成各种算术运算和逻辑运算。

ALU 的算术运算部件包含加法器, 减法器, 乘法器, 除法器, 增量器(+1), 减量器(-1), BCD 码运算器等组件。

ALU 的主要工作是根据 CPU 指令要求执行各种指定运算, 如加法, 减法, 乘法, 除法, 比较, 逻辑移位等操作。

通用寄存器组是一组存取速度最快的存储器, 用于保存参加运算的操作数和中间结果。访问寄存器无需高速缓存, 也不需要运行总线周期, 因此指令的执行速度很快。几乎所有的指令都要将寄存器指定为一个操作数, 有些指令还要求将操作数

存放在专用的寄存器中。

专用寄存器通常用于表示 CPU 所处于某种系统状态，ALU 中有两个重要的状态寄存器：指令指针寄存器 IP(即程序计数器 PC)和标志寄存器 FLAGS。