

1.1 重点内容提要

1. 微型计算机系统硬件组成基础

微型计算机和其他类型的计算机一样,硬件上都是由各种功能的逻辑电路,以离散数学、逻辑代数为数学工具组合而成的。逻辑电路通常分为两大类:组合逻辑电路和时序逻辑电路。

1) 基本逻辑单元电路

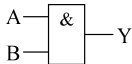
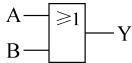
本质上,电路再复杂、功能再强大的数字系统或 IC 芯片,究其内部电路原理,归根到底都是由逻辑门、触发器和脉冲波形产生与整形电路三类基本逻辑单元构建而成的。逻辑门和脉冲波形产生与整形电路是构成任何数字电路所必要的,触发器则是构成时序逻辑电路所必要的。

(1) 组合逻辑基本单元——逻辑门

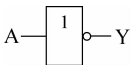
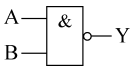
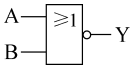
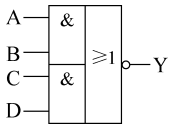
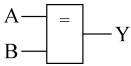
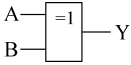
① 常用逻辑门

常用的逻辑门电路在逻辑功能上有与门、或门、非门、与非门、或非门、与或非门、异或门等。表 1.1 列出了三种基本逻辑门和五种常用复合门的逻辑功能及其表示方法。表中所有“与”和“或”都是相对于逻辑 1 而言的(1 有效,0 无效)。若从逻辑 0 的角度看(0 有效,1 无效),则“与”、“或”关系正好相反,即“与”变成“或”,“或”变成“与”,“与非”变成“或非”,“或非”变成“与非”,这就是所谓的“与”、“或”逻辑的相对性。

表 1.1 三种基本逻辑门和五种常用复合门

逻辑门	逻辑表达式	逻辑符号	真 值 表		逻辑功能
与门	$Y=A \cdot B$		A	B	全 1 出 1, 有 0 出 0
			0	0	
			0	1	
			1	0	
			1	1	
或门	$Y=A+B$		A	B	有 1 出 1, 全 0 出 0
			0	0	
			0	1	
			1	0	
			1	1	

续表

逻辑门	逻辑表达式	逻辑符号	真 值 表		逻辑功能
非门	$Y=\overline{A}$		A	Y	入 1 出 0, 入 0 出 1
			0	1	
与非门	$Y=\overline{A \cdot B}$		A B	Y	全 1 出 0, 有 0 出 1
			0 0	1	
			0 1	1	
			1 0	1	
或非门	$Y=\overline{A+B}$		A B	Y	有 1 出 0, 全 0 出 1
			0 0	1	
			0 1	0	
			1 0	0	
与或非门	$Y=\overline{AB+CD}$		A B C D	Y	有一组全 1 出 0, 每组有 0 出 1
			0 0 0 0	1	
			0 0 0 1	1	
			0 0 1 0	1	
			0 0 1 1	0	
			0 1 0 0	1	
			0 1 0 1	1	
			0 1 1 0	1	
			0 1 1 1	0	
			1 0 0 0	1	
			1 0 0 1	1	
			1 0 1 0	1	
			1 0 1 1	0	
			1 1 0 0	0	
			1 1 0 1	0	
			1 1 1 0	0	
			1 1 1 1	0	
同或门	$Y=A \odot B$		A B	Y	相同出 1, 不同出 0
			0 0	1	
			0 1	0	
			1 0	0	
异或门	$Y=A \oplus B$		A B	Y	相同出 0, 不同出 1
			0 0	0	
			0 1	1	
			1 0	1	
			1 1	0	

② 集电极开路门电路(OC 门)

在实际应用中,往往需要将多个门的输出端并联以实现“与”逻辑,这种功能通常被称为“线与”。但典型 TTL 门电路都有一个禁忌:不允许把多个门的输出端并联使用,否则将会使器件损坏。解决这一问题的方法就是把典型 TTL 门的输出级改为集电极开路的三极管结构,做成集电极开路的门电路,简称 OC 门(open collector gate)。各种逻辑功能的门电路都可做成 OC 门结构。图 1.1 给出了集电极开路与非门的逻辑符号。

需注意的是,OC 门在实际使用时必须在输出端通过外接负载电阻 R_L 接至 V_{CC} 或其他电源,如图 1.2 所示。

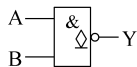


图 1.1 集电极开路与非门的逻辑符号

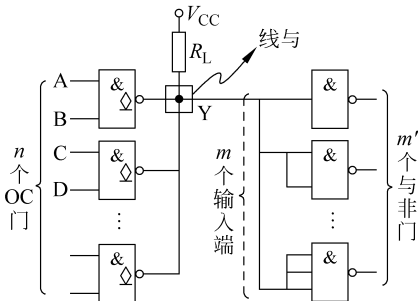


图 1.2 $Y = \overline{AB} \cdot \overline{CD} \cdot \dots$

③ 三态输出门电路(TS 门)

三态输出门电路常被简称为三态门(three state gate)。它是在普通门电路的基础上通过增加控制输入端和控制电路而成的,它的输出除了具有一般 TTL 门电路的两种状态,即输出电阻较小的高、低电平状态外,还具有高输出电阻的第三状态,称为高阻态。三态门同样可以有与门、或门、与非门、或非门等。图 1.3 给出了三态与非门的逻辑符号,其对应的真值表如表 1.2 所示。当三态与非门的控制端有效时,电路为正常的与非工作状态;当控制端无效时,电路为高阻态。

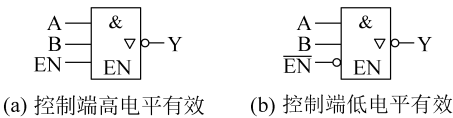


图 1.3 三态与非门的逻辑符号

表 1.2 三态与非门的真值表

控制端高电平有效				控制端低电平有效			
EN	A	B	Y	$\overline{EN}$	A	B	Y
1	0	0	1	0	0	0	1
	0	1	1		0	1	1
	1	0	1		1	0	1
	1	1	0		1	1	0
0	×	×	高阻 Z	1	×	×	高阻 Z

三态门最重要的一个用途就是可以实现一根导线分时传输若干个不同的数据或信号,如图 1.4 所示。图中,共用的那根导线称为总线(bus)。只要让各个门的控制端 $\overline{EN}_i$ 轮流定时地处于有效状态电平,即某一时刻只能有一个三态门处于工作状态,而其余的三态门处于

高阻状态,则总线就可轮流传送各三态门的输出信号。计算机系统中的数据传输基本上都采用这种总线结构。

此外,三态门还经常做成单向和双向总线驱动器,图 1.5 给出了利用三态门电路实现数据双向传输的示意图。

(2) 时序逻辑基本单元——触发器

触发器是构成各种时序逻辑电路的基础,它和逻辑门一样,是数字系统中的基本逻辑单元电路,它区别于逻辑门的最主要点是具有记忆功能,可以存储 1 位二值信号(0 或 1)。

触发器的逻辑符号如图 1.6 所示。它有两个互补的输出端 Q 和 $\bar{Q}$ (正常工作时, Q 和 $\bar{Q}$ 的状态总是相反的),还有 1~2 个输入端,有时也称为激励端或控制端。

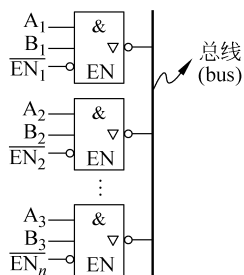


图 1.4 用三态输出门接成总线结构

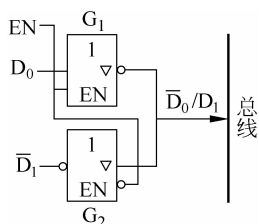


图 1.5 用三态输出门实现数据的双向传输

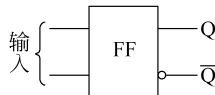


图 1.6 触发器的逻辑符号

为了能记忆 1 位二值信号,触发器必须具备以下三个基本特点:

① 具有两个稳定的状态,即“0”态和“1”态。可用于表示二进制数的 0 和 1。所以,触发器也叫双稳态触发器,有时简称为“双稳”。

② 具有触发翻转的特性。即两个稳态在外加输入信号的作用(触发)下可以相互转化。为叙述方便,一般把触发器原来的状态称为原态,用 Q^n 表示,而把改变后的状态称为次态,用 Q^{n+1} 表示。

③ 利用不同的输入信号可将触发器置成任一稳态,并在输入信号撤销后能保持该稳态不变,即有“自行保持”或“记忆”功能。

触发器按电路结构分,有基本 RS 触发器、同步 RS 触发器、主从触发器和边沿触发器等几种类型;若按触发器的逻辑功能分,又有 RS 触发器、D 触发器、JK 触发器、T 触发器和 T' 触发器等几种类型。

2) 基本组合逻辑部件

常见的基本组合逻辑部件有编码器、译码器、多路数据选择器、数码比较器、加法器以及算术逻辑单元等。

(1) 编码器

所谓编码,就是用二值代码(0 和 1)来表示给定的信号(数字或字符)。而编码器,就是能用来完成编码功能的数字电路。在数字计算机和其他数字系统中,编码器的应用极其广泛,如在指令编码、总线仲裁和多中断源识别中,均涉及编码的概念。

编码器有二进制编码器和二-十进制编码器两类,二进制编码器是把一般信号编为二进制代码的编码器;二-十进制编码器则是把十进制的十个状态编成十个 BCD 代码。

无论二进制编码器还是二-十进制编码器,又有普通编码器和优先编码器之分。两者的区别在于:普通编码器任何时刻只允许一个输入有效;而优先编码器允许同时输入两个以上的编码信号,当几个输入信号同时有效时,只对其中优先权最高的一个进行编码。

目前,常用的二进制编码器有 8 线-3 线优先编码器 74LS148,二-十进制编码器有 74LS147 优先编码器。

(2) 译码器

译码是编码的逆过程。有时也把译码器叫做解码器。其作用就是将代码的含意翻译出来,还原为特定的输出信号。

译码器同样有二进制译码器和二-十进制译码器之分。

① 二进制译码器是把二进制代码的各个状态按照原意译成对应的输出信号,即将 n 位二进制代码转换为 2^n 个输出。所以,二进制译码器又叫“ n 线- 2^n 线译码器”。

目前,应用最广的集成二进制译码器有“双 2 线-4 线”译码器 74LS139、“3 线-8 线译码器”74LS138、“4 线-16 线译码器”74LS154 等。

图 1.7 和表 1.3 分别给出了“3 线-8 线译码器”74LS138 的逻辑符号和功能表。

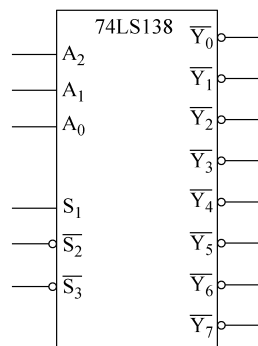


图 1.7 74LS138 的逻辑符号

表 1.3 3 线-8 线译码器 74LS138 的功能表

输 入					输 出							
S_1	$\overline{S_2 + S_3}$	A_2	A_1	A_0	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$	$\overline{Y_4}$	$\overline{Y_5}$	$\overline{Y_6}$	$\overline{Y_7}$
0	×	×	×	×	1	1	1	1	1	1	1	1
×	1	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

② 二-十进制译码器则是将输入 BCD 码的十个代码翻译成十进制数字信号的译码器。主要用于十进制数字的显示,常与数字显示器件配合使用。依数字显示器件的工作方式不同,对 BCD 译码器的要求也有所不同,从而使译码器的电路结构和工作原理也就不一样。

常用的二-十进制译码器有七段显示译码器 74LS46 和 74LS48 等。

(3) 数据选择器

数据选择器也叫多路选择器或多路复用开关。它是一种多输入单输出的组合逻辑部件。它在地址(选择)信号作用下,可从多路输入数据中选出所需要的一路送至输出端。数据选择器的逻辑功能可用图 1.8 所示的单刀多掷开关来模拟。图中, $D_0 \sim D_3$ 称为数据输入

端, A_1 、 A_0 为地址(选择)输入端, Y 为输出端。当 $A_1 A_0 = 00$ 时, 开关 K 与 D_0 相连, $Y = D_0$; 当 $A_1 A_0 = 01$ 时, 开关 K 与 D_1 相连, $Y = D_1$; 依此类推, 当 $A_1 A_0 = 11$ 时, 开关 K 与 D_3 相连, $Y = D_3$ 。

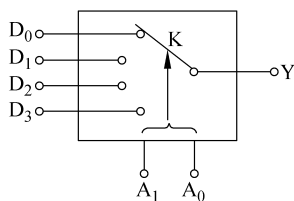


图 1.8 数据选择器示意图

目前, 常用的集成数据选择器有单/双/四 2 选 1、单/双 4 选 1、8 选 1、16 选 1 等多种类型。典型的有四 2 选 1 数据选择器 74LS158、双 4 选 1 数据选择器 74LS253、8 选 1 数据选择器 74LS151 等。

(4) 数码比较器

数码比较器是对两个数码进行比较, 以判断它们的相对大小以及是否相等的逻辑电路。通常, 在比较两个多位数的大小时, 必须自高而低地逐位进行比较。如果高位能比较出数值大小, 则比较结束; 如果高位数值相等, 再依次比较低位数值, 直至比较出结果。

目前, 常用的集成数码比较器有 CT7485 四位数码比较器等。

3) 基本时序逻辑部件

(1) 寄存器

寄存器是数字计算机和其他数字系统中常用的一类时序逻辑部件, 主要用于寄存一组二值代码。因为一个触发器只存储一位二值代码, 所以有多少位代码要存储, 寄存器就必须有多少个触发器。可见, 触发器组是寄存器的核心组成部分。除此之外, 寄存器通常还应有由门电路组成的控制电路, 用于控制寄存器的“接收”、“清零”、“保持”、“输出”等功能。

按照寄存数据的输入/输出方式不同, 寄存器可分为并行输入-并行输出、串行输入-并行输出、并行输入-串行输出和串行输入-串行输出 4 种类型。并行输入-并行输出的寄存器通常称为静态寄存器, 简称寄存器; 其余三种输入/输出形式的寄存器则统称为移位寄存器。

典型的寄存器芯片, TTL 集成电路有 74LS174、74LS175、74LS378、74LS374、74LS575、74LS824 和 74LS822 等, COMS 集成电路有 4076、40174 和 74HC374 等; 移位寄存器芯片有 8 位串入-并出移位寄存器 74LS164、4 位串/并入-串/并出移位寄存器 74LS195 和 4 位多功能集成移位寄存器 74LS194 等。

(2) 计数器

计数器是数字计算机和其他数字系统中使用最多的时序逻辑部件。它不仅可用来记录脉冲的个数, 而且还大量用作分频、程序控制及逻辑控制等。

计数器按计数脉冲触发方式可分为同步和异步两大类; 按计数制可分为二进制和非二进制(包括十进制和任意进制)两类; 按计数过程中数值的增减可分为加法、减法和可逆计数器; 此外还有有权码、无权码、移位型等其他名称的计数器。

计数器工作时, 经历的状态是有限的, 并且是周期性循环的, 表现为状态图一定有一个计数主循环, 它包含的状态数也就是计数器的模, 用 M 表示, 如十进制计数器的模为 10 ($M=10$), 称为模 10 计数器。

计数器有时也叫分频器, 虽然同是一个电路, 但使用时是有区别的: 分频器的时钟脉冲 CP 一定是周期性的信号, 所以输出信号也是周期性的, 输出信号的周期是输入信号周期的 M 倍, 反过来输出信号的频率是输入信号频率的 M 分之一, 这正是分频器得名的原因; 计数器的时钟脉冲 CP 不一定是周期性的信号, 可以是随机脉冲, 称为计数脉冲, 相应地输出

信号也不一定是周期性的,计数器的工作目的是记录计数脉冲个数(递增或递减)以及产生溢出(进位或借位)信号。

各种计数器的不同点主要表现在计数方式(同步计数或异步计数)、触发方式(上升沿或下降沿)、模及码制(自然二进制码或BCD码)、计数规律(加法、减法计数或加/减计数)、预置方式(同步预置或异步预置)以及复位方式(异步复位或同步复位)等几个方面。表1.4归纳总结了常用集成计数器的各种工作特点。

表 1.4 常用集成计数器及工作特点

型号	计数方式	触发方式	模及码制	计数规律	预置	复位
74LS160	同步	上升沿	模 10,8421 码	加法	同步	异步
74LS161	同步	上升沿	模 16,二进制	加法	同步	异步
74LS162	同步	上升沿	模 10,8421 码	加法	同步	同步
74LS163	同步	上升沿	模 16,二进制	加法	同步	同步
74LS190	同步	上升沿	模 10,8421 码	单时钟,加/减	异步	无
74LS191	同步	上升沿	模 16,二进制	单时钟,加/减	异步	无
74LS192	同步	上升沿	模 10,8421 码	双时钟,加/减	异步	异步
74LS193	同步	上升沿	模 16,二进制	双时钟,加/减	异步	异步
74LS196	异步	上升沿	模 2-5-10	加法	异步	异步
74LS197	异步	上升沿	模 2-8-16	加法	异步	异步
74LS290	异步	上升沿	模 2-5-10	加法	异步	异步
74LS293	异步	上升沿	模 2-8-16	加法	无	异步

2. 微型计算机系统的硬件结构

1) 冯·诺依曼结构

微型计算机(microcomputer,简称微机)系统,从体系结构来看采用的基本上是计算机系统的经典结构——冯·诺依曼结构。这种结构的特点是:

- (1) 计算机由运算器、控制器、存储器、输入设备和输出设备五大部分组成。
- (2) 数据和程序以二进制代码形式不加区别地存放在同一个存储器中,存放位置由地址指定,地址码也为二进制形式。
- (3) 控制器是根据存放在存储器中的指令序列即程序来工作的,并由一个程序计数器PC控制指令的执行。控制器具有判断能力,能根据计算结果选择不同的动作流程。

2) 哈佛结构

在冯·诺依曼体系结构中,程序和数据不加区别地存放在同一个存储器中,如图1.9所示。这种指令和数据共享同一总线的结构,使得指令的读取与数据的交换无法重叠执行,成为限制计算机性能的瓶颈。为此,人们又提出了“哈佛结构”。

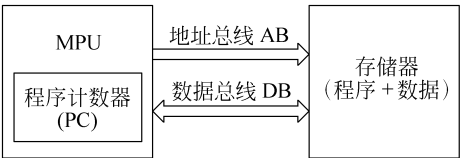


图 1.9 冯·诺依曼计算机的存储器结构图

与冯·诺依曼体系结构不同,哈佛结构是一种并行体系结构,如图 1.10 所示。其主要特点是:

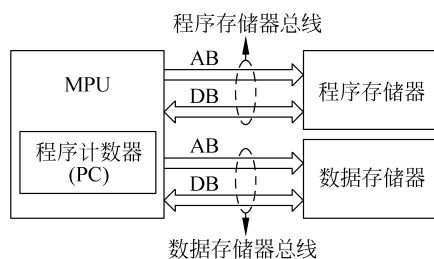


图 1.10 哈佛结构计算机的存储器结构图

(1) 程序和数据分别存储在不同的存储空间,即程序存储器和数据存储器是两个独立的存储器模块,每个存储模块独立编址、独立访问,且都不允许程序与数据并存。

(2) 程序存储器和数据存储器采用独立的两套总线,即程序存储器总线和数据存储器总线,分别作为微处理器与每个存储器模块之间的专用通信路径,这两套总线之间毫无关联,可以拥有不同的数据宽度。

显然,哈佛结构的上述特点,使得取指令和存取数据分别经由不同的存储空间和不同的总线,有利于缓解数据流传输的瓶颈;指令和数据能够同时访问,提高了 CPU 的执行速度和数据的吞吐率。

3) 三总线结构

无论冯·诺依曼结构还是哈佛结构,微机系统在硬件上都是由运算器、控制器、存储器、输入设备和输出设备五部分组成。而各部分硬件又是通过地址总线(AB)、数据总线(DB)和控制总线(CB)三大总线联系到一起的(如图 1.11 所示),故称为三总线结构。

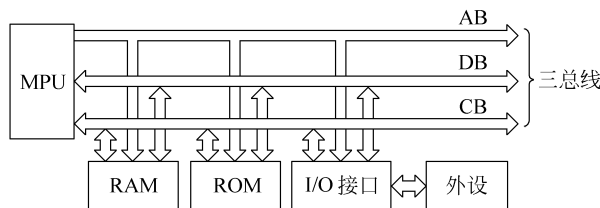


图 1.11 微机系统的硬件结构框图

由于使用总线,使硬件各模块之间的相互依赖关系变为模块与总线间的单向依赖关系,即只要满足相同总线规范的模块就可应用于系统,从而使微型计算机的系统构造比较简单,且具有更大的灵活性和更好的可扩充性、可维修性。根据总线组织方法不同,又可分为单总线、双总线、双重/多重总线三类。

3. 微型计算机的主要组成部分及功能

微型计算机主要由微处理器(MPU)、存储器、I/O 接口和总线四部分组成。

1) 微处理器

微处理器是微型计算机的运算和指挥控制中心。不同型号的微型计算机,其性能的差别首先在于其微处理器性能的不同,而微处理器性能又与它的内部结构、硬件配置有关。但无论哪种微处理器,其内部基本结构总是相同的,都由运算器、控制器和寄存器组三大部分,外加内部总线及缓冲器组成,每个部分又各由一些基本部件组成。

运算器主要由算术逻辑单元 ALU、累加器 ACC、累加锁存器、暂存器和标志寄存器 FR 等部件组成。其主要功能是完成各种算术运算和逻辑运算。

控制器主要由程序计数器 PC、指令寄存器 IR、指令译码器 ID、操作控制器 OC、堆栈指

针 SP 和一组通用寄存器等部件组成,是整个微处理器的指挥控制中心。控制器的主要功能是:

- 按照程序逻辑要求,控制程序中指令的执行顺序;
- 根据指令寄存器中的指令码控制每一条指令的执行过程。

寄存器组实质上是微处理器的内部 RAM,可分为专用寄存器和通用寄存器两类。**专用寄存器**(含堆栈指针 SP、程序计数器 PC、标志寄存器 FR 等)的作用是固定的;**通用寄存器**可由程序员规定其用途,一般用于暂时存放需重复使用的某些操作数或中间结果。

对其中一些主要组成部件的功能与工作特点要重点掌握:

(1) 算术逻辑单元 ALU 和累加器 ACC

ALU 是运算器的核心,是一种以全加器为核心的具有多种运算功能的组合逻辑电路。用于完成各种算术、逻辑及移位运算。累加器 ACC 提供送入 ALU 的两个运算操作数之一,并存放运算后的结果。而操作结果的某些重要状态或特征则存于标志寄存器 FR,有些机器的 FR 中还存放控制处理器工作方式的控制标志和系统标志。

(2) 程序计数器 PC

程序计数器 PC 是维持微处理器有序地执行程序的关键性的、不可缺少的寄存器。它存放着下一条要执行的指令的存放地址,并能自动修改指向下一指令的存放地址。

(3) 指令寄存器 IR、指令译码器 ID 和操作控制器 OC

指令寄存器 IR 存放从存储器中取出的各条指令的操作码。指令译码器 ID 对 IR 中存放的指令进行译码(分析),确定应该进行什么操作。操作控制器 OC 则依据指令译码器 ID 和时序电路的输出信号,产生执行指令所需的全部微操作控制信号,以控制计算机的各部件执行该指令所规定的操作。由于每条指令所执行的具体操作不同,所以每条指令都有一组不同的控制信号的组合,以确定相应的微操作序列。

(4) 堆栈和堆栈指针 SP

堆栈是计算机中一种先进后出的数据结构,由栈区和堆栈指针 SP 组成。栈区用来存放数据,既可由硬件寄存器组成,称为硬件堆栈,又可由 RAM 存储区构成,称为软件堆栈。堆栈指针是用来指示栈顶地址的寄存器,用于自动管理栈区,指示当前数据存入或取出的位置。

堆栈有两种操作:压栈(进栈)和弹栈(出栈),均只能在栈顶进行。堆栈作为数据的一种暂存结构,一般主要用于中断处理和过程调用。

2) 存储器

存储器又叫内存或主存,是微型计算机的存储和记忆部件,用以存放数据和程序。

(1) 内存单元的编址

内存是由一个个内存单元组成的,内存单元的总数目称为内存容量。

现代微机为适应变字长处理需求,都以字节单元编址来管理内存,如图 1.12 所示,每个存储单元存放一个 8 位字节数据,对应一个唯一地址编码,CPU 通过地址识别不同的内存单元,正确地内存单元进行操作。

地址	内容
00000H	10110010
00001H	11000111
00002H	00001100
⋮	⋮
F0000H	00111110
⋮	⋮
FFFFFH	01110010

图 1.12 内存单元的编址

(2) 内存操作

CPU 对内存的操作有读、写两种。从内存单元取数到 CPU 内部叫做读操作；从 CPU 内部送数到内存单元叫做写操作。图 1.13、图 1.14 分别给出了存储器读操作和写操作示意。

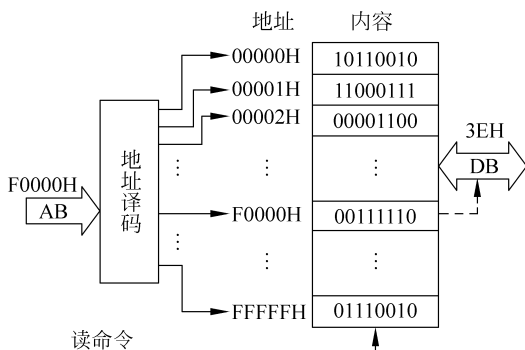


图 1.13 存储器读操作示意图

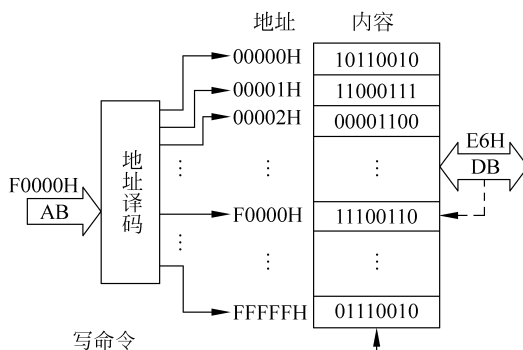


图 1.14 存储器写操作示意图

以读操作为例,CPU 首先通过地址总线送出被读单元的地址编码(F0000H),经地址译码找到被读单元 F0000H,当读命令有效时,F0000H 单元内存放的数据 3EH 被送上数据总线,原有内容保持不变。

写操作与读操作过程类似,CPU 也是先送出被写单元的地址编码,经译码找到被写单元(F0000H),当写命令有效时,数据总线上的数据 E6H 被送入 F0000H 单元,原有内容被刷新。

(3) 内存分类

按工作方式不同,内存可分为两大类: 随机存取存储器(random access memory, RAM)和只读存储器(read only memory, ROM)。

RAM 可以被 CPU 随机地读和写,所以又称为读/写存储器。这种存储器用于存放用户装入的程序、数据及部分系统信息。当机器断电后,所存信息消失。

ROM 中的信息只能被 CPU 随机读取,而不能由 CPU 任意写入。机器断电后,信息并不丢失。所以,这种存储器主要用来存放各种程序,如汇编程序、各种高级语言解释或编译程序、监控程序、基本 I/O 程序等标准子程序,也用来存放各种常用数据和表格等。

3) 输入/输出(I/O)设备的接口

(1) I/O 设备与 CPU 的连接

I/O 设备是计算机系统必不可少的组成部分。但任何外部 I/O 设备都不能直接与计算机相连,须通过 I/O 接口与 CPU 连接,这是因为:

① 两者的信息类型和格式可能不一样。外设的信息类型有数字量、模拟量、开关量和脉冲量信息,而 CPU 只能接收数字量信息;即使都是数字量,两者的信息格式也可能不一样,外设并行和串行数据之分,但 CPU 只能接收并行数据。

② CPU 与外设的信号传输速度往往不能匹配,其信号时序有很大差别。

所以,在 CPU 与外设间需引入“接口电路(也叫 I/O 适配器)”作为连接两者的桥梁,来缓冲和协调 CPU 与外设的传输速度,以及进行信息类型和格式变换。