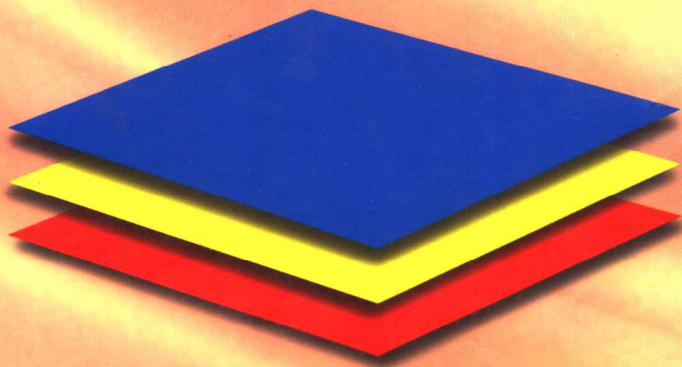




单片机原理 及接口技术

(简明修订版)

李朝青 编著



北京航空航天大学出版社

单片机原理及接口技术

(简明修订版)

李朝青 编著

北京航空航天大学出版社

内 容 简 介

本书是在北航出版社1994年出版的《单片机原理及接口技术》的基础上,根据近四年来从各方面获得的反馈信息,删改了许多陈旧内容并代之以新的内容而成的。全书变动内容超过50%。作者在保持原书精华的同时,对内容做了适当的压缩和调整,力求使书的内容更加系统而简明。

全书共分为十一章,深入浅出地介绍了8051单片机和89C51系列单片机的原理、接口及应用技术。主要内容包括:单片机外围芯片知识,微机的组成及工作过程,8051的结构原理、指令系统,系统配置及接口技术,实用程序设计举例,通信原理及C语言程序,与8051兼容的80C552、89C51和89C2051的原理及应用。

本书内容丰富、新颖、通俗、实用,适合自学,可用作高等院校本、专科生的教材,也可供从事单片机产品开发的工程技术人员参考。

图书在版编目(CIP)数据

单片机原理及接口技术/李朝青编著. -北京:北京航空航天大学出版社,1998.11

ISBN 7-81012-819-1

I. 单… II. 李… III. 单片微型计算机-基本知识 IV. TP368.1

中国版本图书馆CIP数据核字(98)第19362号

单片机原理及接口技术(简明修订版)

李朝青 编著

责任编辑 冯学民

责任校对 李宝田

*

北京航空航天大学出版社出版发行

北京市学院路37号,邮编100083 发行部电话(010)82317024

<http://www.buaapress.cn.net>

E-mail: pressell@publica.bj.cninfo.net

天津铁道部十八局印刷厂印装 各地书店经销

*

开本:787×1092 1/16 印张:19 字数:482千字

1999年3月第1版 1999年3月第1次印刷 印数:5000册

ISBN 7-81012-819-1/TP·302 定价:26.00元

前 言

《单片机原理及接口技术》是北航出版社于 1994 年出版的,至今已印刷了四次,仍不能满足读者需求。鉴于本书已出版了近四年的时间,有些内容已经过时,加上篇幅过多(约 70 万字),用于教与学所花时间较多,因此,有必要对原书进行修订,同时对内容进行适当的压缩和调整。作者根据来自各方面的反馈信息,经过反复斟酌,对原书内容进行了很大变动,删掉了已经过时的和不太重要的内容,补充了很多新内容。增加的内容包括:实用程序设计举例;与 8051 兼容的 89C51 系列单片机原理及应用;用 C 语言编写的 PC 机与 8051 通信程序实例;另外还增加了一些例题和习题,以方便教学。

本书可作为大、中专院校“单片微机及接口”、“微机原理”和“单片机原理及应用”等课程的教材,也可供从事单片机产品开发的技术人员参考。

本书内容丰富、新颖、通俗、实用,适于自学。

参加本书编写工作的人员还有王洪涛、张秋燕、刘艳玲、曹文嫣、韩继增、卢晋、许居衡、郝廷柱等。

由于作者水平有限,书中难免有错误和不妥之处,恳请读者予以指正或提出修改意见。

李朝青

于天津理工学院电子系

邮编 300191

1998 年 6 月

目 录

第一章 微机基础知识	(1)
1.1 微处理器、微机和单片机的概念	(1)
1.1.1 微处理器(机)的组成	(1)
1.1.2 存储器和输入/输出接口	(4)
1.2 微机的工作过程	(5)
1.2.1 执行一条指令的顺序	(6)
1.2.2 执行一条指令的过程	(6)
1.2.3 执行一个程序的过程	(7)
1.3 思考题与习题	(10)
第二章 MCS-51 单片机的结构和原理	(11)
2.1 MCS-51 系列单片机的结构	(11)
2.1.1 MCS-51 单片机的基本组成	(11)
2.1.2 MCS-51 单片机内部结构	(12)
2.2 MCS-51 单片机引脚及其功能	(15)
2.3 8051 存储器配置	(17)
2.3.1 程序存储器地址空间	(18)
2.3.2 数据存储器地址空间	(19)
2.4 CPU 时序	(26)
2.4.1 片内振荡器及时钟信号的产生	(26)
2.4.2 机器周期和指令周期	(27)
2.4.3 CPU 取指、执行周期时序	(27)
2.5 复位及复位电路	(28)
2.5.1 复位操作	(28)
2.5.2 复位信号及其产生	(29)
2.5.3 复位电路	(29)
2.6 输入/输出端口结构	(30)
2.6.1 P0 口	(30)
2.6.2 P1 口	(32)
2.6.3 P2 口	(32)
2.6.4 P3 口	(33)
2.6.5 端口的负载能力和接口要求	(34)
2.7 思考题与习题	(35)

第三章 8051 指令系统	(36)
3.1 汇编语言	(36)
3.1.1 指令和程序设计语言	(36)
3.1.2 指令格式	(37)
3.2 寻址方式	(38)
3.2.1 七种寻址方式	(38)
3.2.2 寻址空间及符号注释	(41)
3.3 MCS-51 单片机的指令系统	(43)
3.3.1 数据传送指令	(43)
3.3.2 算术运算指令	(49)
3.3.3 逻辑操作指令	(53)
3.3.4 控制程序转移类指令	(56)
3.3.5 位操作(布尔处理)类指令	(63)
3.4 思考题与习题	(67)
第四章 汇编语言程序设计知识	(71)
4.1 编程的步骤、方法和技巧	(71)
4.1.1 编程步骤	(71)
4.1.2 编程的方法和技巧	(73)
4.1.3 汇编语言程序的基本结构	(74)
4.2 汇编语言源程序的编辑和汇编	(76)
4.2.1 源程序编辑	(76)
4.2.2 源程序的汇编	(76)
4.2.3 伪指令	(77)
第五章 中断系统	(80)
5.1 微机的输入/输出方式	(80)
5.1.1 无条件传送方式	(80)
5.1.2 查询传送方式	(80)
5.1.3 直接存储器存取(DMA)方式	(81)
5.2 中断的概念	(81)
5.3 8051 中断系统结构及中断控制	(82)
5.3.1 8051 中断源	(83)
5.3.2 中断控制	(84)
5.4 中断处理过程	(87)
5.4.1 中断响应	(88)
5.4.2 中断处理	(90)
5.4.3 中断返回	(90)

5.5	外部中断扩展方法	(91)
5.5.1	利用定时器扩展外部中断源	(91)
5.5.2	中断加查询扩展中断源	(91)
5.6	中断程序举例	(92)
5.6.1	主程序	(92)
5.6.2	中断服务程序	(92)
5.7	思考题与习题	(95)
第六章	定时器及应用	(97)
6.1	定时器概述	(97)
6.2	定时器的控制	(98)
6.2.1	工作模式寄存器 TMOD	(98)
6.2.2	控制寄存器 TCON	(99)
6.3	定时器的四种模式及应用	(100)
6.3.1	模式 0 及应用	(100)
6.3.2	模式 1 及应用	(103)
6.3.3	模式 2 及应用	(104)
6.3.4	模式 3 及应用	(106)
6.3.5	综合应用举例	(108)
6.4	思考题与习题	(111)
第七章	单片机系统扩展及接口技术	(112)
7.1	扩展三总线的产生	(113)
7.1.1	片外三总线结构	(113)
7.1.2	系统扩展的实现	(114)
7.2	扩展程序存储器	(115)
7.2.1	访问片外程序存储器的操作时序	(115)
7.2.2	扩展 8KB/16KB EPROM	(117)
7.3	扩展数据存储器	(119)
7.3.1	常用的数据存储器芯片	(119)
7.3.2	访问片外 RAM 的操作时序	(122)
7.3.3	8051 扩展 2KB RAM	(123)
7.3.4	8031 外扩 32KB EPROM 和 32KB RAM	(124)
7.3.5	8031 扩展 8KB E ² PROM	(125)
7.4	简单并行 I/O 口的扩展	(126)
7.4.1	I/O 口的直接输入/输出	(127)
7.4.2	简单 I/O 接口的扩展方法	(128)
7.5	扩展 8155 可编程外围并行接口芯片	(129)
7.5.1	8155 的结构及引脚	(129)

7.5.2	8155 的 RAM 和 I/O 口寻址	(131)
7.5.3	8155 的寄存器	(131)
7.5.4	8155 芯片的使用	(134)
7.5.5	8031 单片机与 8155 的接口及简单程序	(137)
7.6	思考题与习题	(138)
第八章	应用系统配置及接口技术	(140)
8.1	人-机通道配置与接口技术	(140)
8.1.1	键盘接口及处理程序	(140)
8.1.2	LED 显示器接口及显示程序	(149)
8.1.3	键盘/LED 显示器与 8155 接口及键盘扫描子程序	(153)
8.1.4	串行口控制的键盘/LED 显示器接口电路	(155)
8.1.5	打印机及接口	(158)
8.2	单片机测控系统前向通道配置——传感器及小信号放大电路	(159)
8.2.1	传感器	(160)
8.2.2	模拟小信号放大	(161)
8.2.3	放大电路实例	(162)
8.3	前向通道中的 A/D 转换器及接口技术	(163)
8.3.1	逐次逼近型 A/D 转换器及接口	(164)
8.3.2	双积分 A/D 转换器及接口技术	(167)
8.4	系统后向通道配置及接口技术	(170)
8.4.1	后向通道中的功率开关器件及接口	(170)
8.4.2	后向通道中的 D/A 转换及接口技术	(174)
8.5	思考题与习题	(178)
第九章	8051 串行口及串行通信技术	(179)
9.1	串行通信基本知识	(179)
9.1.1	数据通信	(179)
9.1.2	串行通信的传输方式	(180)
9.1.3	异步通信和同步通信	(180)
9.1.4	串行通信的过程及通信协议	(182)
9.2	串行口及应用	(184)
9.2.1	8051 串行口	(184)
9.2.2	8051 串行口的应用	(195)
9.3	RS-232C 标准接口总线及串行通信硬件设计	(203)
9.3.1	RS-232C 标准接口总线	(204)
9.3.2	信号电气特性与电平转换	(205)
9.3.3	RS-232C 的应用	(206)
9.3.4	单片机与 PC 机通信的接口电路	(208)

9.4	8051 与 8051 点对点异步通信	(210)
9.4.1	通信协议	(210)
9.4.2	波特率设置	(211)
9.4.3	通信程序举例	(212)
9.5	8051 与 PC 机间通信软件的设计	(216)
9.5.1	PC 机通信软件设计	(216)
9.5.2	8051 通信软件设计	(224)
9.6	PC 机与多个单片机间的通信	(228)
9.6.1	采用 RS-232C 标准总线通信	(228)
9.6.2	采用 RS-422A 标准总线的通信系统	(231)
9.7	思考题与习题	(234)
第十章	系统实用程序	(235)
10.1	主程序和子程序的概念	(235)
10.1.1	主程序	(235)
10.1.2	子程序及参数传递	(236)
10.1.3	中断服务子程序	(237)
10.2	数据采集及简单控制程序	(238)
10.2.1	$3\frac{1}{2}$ 位数据采集程序	(238)
10.2.2	航标灯控制程序	(240)
10.2.3	水位控制程序	(242)
10.2.4	蜂鸣音报警子程序	(244)
10.3	数据处理程序	(245)
10.3.1	排序程序	(245)
10.3.2	数字滤波程序	(246)
10.3.3	标度变换(工程量变换)	(249)
10.4	代码转换程序	(252)
10.5	软件抗干扰	(255)
10.5.1	软件陷阱技术	(255)
10.5.2	软件看门狗	(257)
10.6	定时程序	(259)
10.6.1	软件定时程序	(259)
10.6.2	硬件定时程序	(261)
10.7	最短程序	(262)
第十一章	80C51 系列单片机	(263)
11.1	PHILIPS 83C552 单片微控制器简介	(263)
11.2	ATMEL 89C51 系列单片机	(270)

11.3 89C2051 单片机.....	(276)
附录 A MCS-51 指令表	(284)
附录 B MCS-51 指令矩阵(汇编/反汇编表)	(289)
主要参考资料.....	(290)

第一章 微机基础知识

1.1 微处理器、微机和单片机的概念

首先,我们介绍一下微处理器(Microprocessor,简称 μP)、微型计算机(Microcomputer,简称微机, μC)和单片机(Single-Chip Microcomputer)的概念。

微处理器(芯片)本身不是计算机,但它是小型计算机或微型计算机的控制和处理部分。

微机则是具有完整运算及控制功能的计算机,它除了包括微处理器(作为它的中央处理单元CPU——Central Processing Unit)外,还包括存储器、接口适配器(即输入/输出接口电路)以及输入/输出(I/O)设备等。图 1-1 所示为微机的各组成部分。其中,微处理器由控制器、运算器和若干个寄存器组成;I/O设备与微处理器的连接需要通过接口适配器(即 I/O 接口);存储器是指微机内部的存储器(RAM,ROM 和 EPROM 等芯片)。

将微处理器、一定容量的 RAM 和 ROM 以及 I/O 口、定时器等电路集成在一块芯片上,构成单片微型计算机,简称单片机。

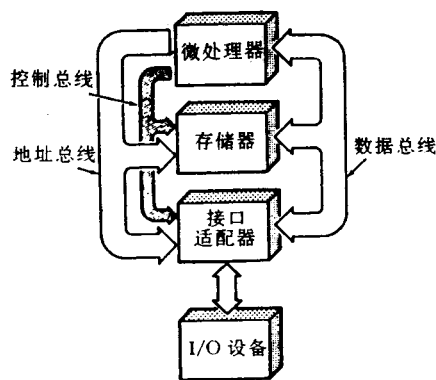


图 1-1 微机的组成

1.1.1 微处理器(机)的组成

微处理器包括两个主要部分:运算器和控制器。

图 1-2 所示是一个较详细的由微处理器、存储器和 I/O 接口组成的计算机模型。为了简化问题,在 CPU(或 MPU)中只画出了主要的寄存器和控制电路,并且假设所有的计数器、寄存器和总线都是 8 位宽度,即要求多数主要寄存器和存储器能保存 8 位(bit)数据,传送数据的总线由 8 根并行导线组成。

在计算机术语中,数据单元是一组二进制数,是计算机中使用的基本信息单元。它可以作为数据,也可以是计算机完成某操作的一条指令码,还可以是 ASCII 码字符等。

在 8 位微处理器中,数据单元由一个字节(Byte)组成;在 16 位机中,数据单元由二个字节组成。图 1-3 表示了组成计算机数据单元的位数。

一、运算器

运算器由运算部件——算术逻辑单元(Arithmetic & Logical Unit,简称 ALU)、累加器和寄存器等几部分组成。ALU 的作用是把传送到微处理器的数据进行算术或逻辑运算。ALU 有两个主要的输入来源,一个来自累加器,另一个来自数据寄存器。ALU 能够完成这两个输入

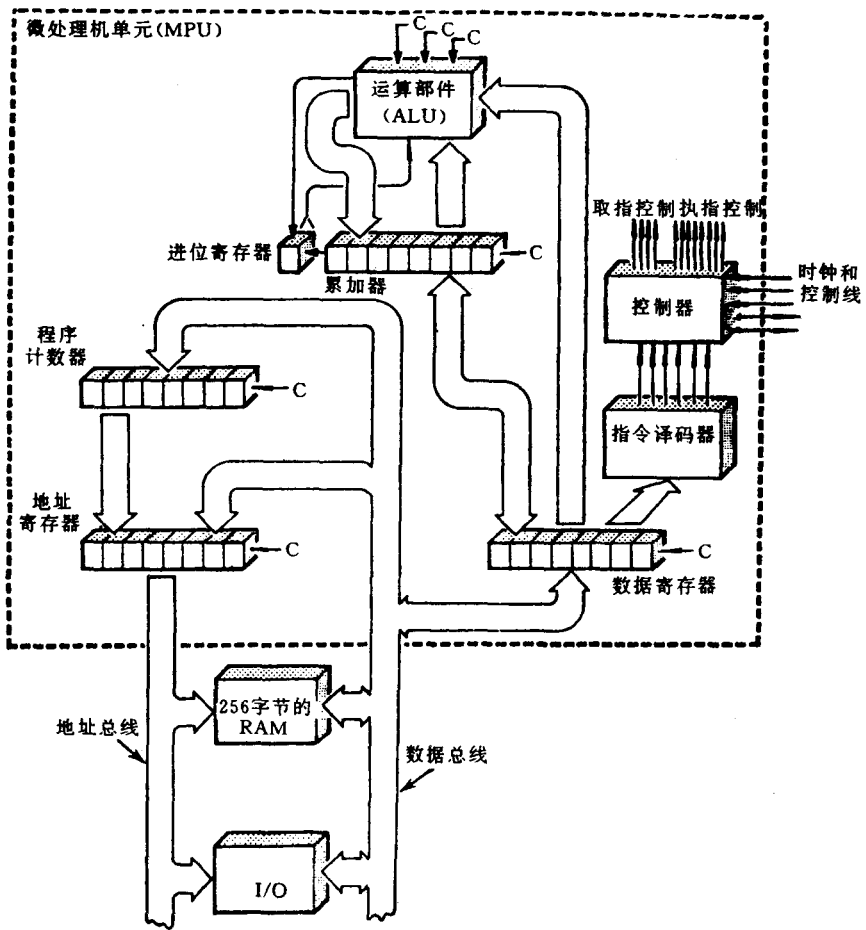


图 1-2 一个计算机模型

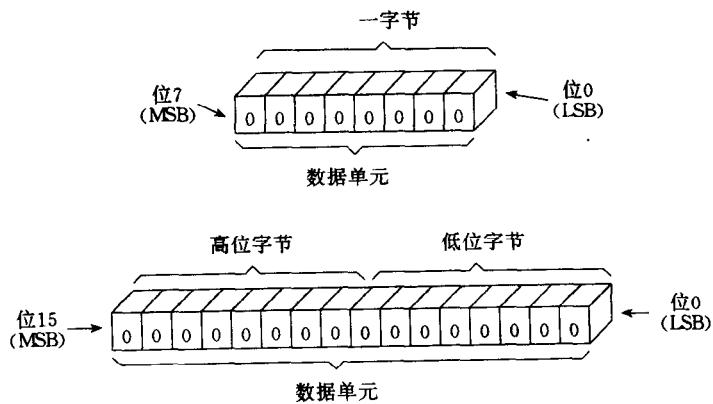


图1-3 计算机中的数据单元

数据的相加或相减运算,也能够完成某些逻辑运算。ALU 执行不同的运算操作是由不同控制线上的信号(在图 1-2 方框图上的标志为 C)所确定的。

通常,ALU 接收来自累加器和数据寄存器的两个 8 位二进制数,因为要对这些数据进行某些操作,所以将这两个输入的数据均称为操作数。

ALU 可对两个操作数进行加、减、与、或、比较大小等操作,最后将结果存入累加器。例如,两个数 7 和 9 相加,在相加之前,操作数 9 放在累加器中,7 放在数据寄存器中,执行两数相加运算的控制线发出“加”操作信号,ALU 即把两个数相加并把所得结果 16 存入累加器,取代累加器原来存放的数 9。总之,运算器有两个主要功能:

- ① 执行各种算术运算。
 - ② 执行各种逻辑运算,并进行逻辑测试,如零值测试或两个值的比较。
- 通常,一个算术操作产生一个运算结果,而一个逻辑操作产生一个判决。

二、控制器

控制器由程序计数器、指令寄存器、指令译码器、时序发生器和操作控制器等组成,是发布命令的“决策机构”,即协调和指挥整个计算机系统的操作。控制器的主要功能有:

- ① 从内存中取出一条指令,并指出下一条指令在内存中的位置。
- ② 对指令进行译码或测试,并产生相应的操作控制信号,以便执行规定的动作。比如一次内存读/写操作,一个算术/逻辑运算操作或一个输入/输出操作等。
- ③ 指挥并控制 CPU、内存和输入/输出设备之间数据流动的方向。

相对控制器而言,运算器接收控制器的命令而进行动作,即运算器所执行的全部操作都是由控制器发出的控制信号来指挥的。

ALU、计数器、寄存器和控制部分除在微处理器内通过内部总线相互联系以外,还通过外部总线与外部的存储器和输入/输出接口电路联系。外部总线一般分为数据总线 DB、地址总线 AB 和控制总线 CB,统称为系统总线。存储器包括 RAM 和 ROM。微型计算机通过输入/输出接口电路可与各种外围设备连接。

三、CPU 中的主要寄存器

1. 累加器(A)

累加器是微处理器中最繁忙的寄存器。在算术和逻辑运算时,它具有双重功能:运算前,用于保存一个操作数;运算后,用于保存所得的和、差或逻辑运算结果。

2. 数据寄存器(DR)

数据(缓冲)寄存器是通过数据总线向存储器和输入/输出设备送(写)或取(读)数据的暂存单元。它可以保存一条正在译码的指令,也可以保存正在送往存储器中存储的一个数据字节等等。

3. 指令寄存器(IR)及指令译码器(ID)

指令寄存器用来保存当前正在执行的一条指令。当执行一条指令时,先把它从内存取到数据寄存器中,然后再传送到指令寄存器(图中未画出)。指令分为操作码和地址码字段,由二进制数字组成。为执行给定的指令,必须对操作码进行译码,以便确定所要求的操作。指令译码器就是负责这项工作的。指令寄存器中操作码字段的输出就是指令译码器的输入。操作码一经译码后,即可向操作控制器发出具体操作的特定信号。

4. 程序计数器(PC)

为了保证程序能够连续地执行下去,CPU 必须采取某些手段来确定下一条指令的地址。程序计数器正是起到了这种作用,所以通常又称其为指令地址计数器。在程序开始执行前,必须将其起始地址,即程序的第一条指令所在的内存单元地址送入 PC;当执行指令时,CPU 将自动修改 PC 的内容,使之总是指示出将要执行的下一条指令的地址。由于大多数指令都是按顺序执行的,所以修改的过程通常只是简单的加 1 操作。

5. 地址寄存器(AR)

地址寄存器用于保存当前 CPU 所要访问的内存单元或 I/O 设备的地址。由于内存和 CPU 之间存在着速度上的差别,所以必须使用地址寄存器来保持地址信息,直到内存读/写操作完成为止。

显而易见,当 CPU 和内存进行信息交换(即 CPU 向/从存储器存/取数据或者 CPU 从内存读出指令)时,都要使用地址寄存器和数据寄存器。同样,如果把外围设备的地址作为内存地址单元来看待的话,那么,当 CPU 和外围设备交换信息时,也需要使用地址寄存器和数据寄存器。

1.1.2 存储器和输入/输出接口

一、存储器

如图 1-4 所示,假设某台微型计算机使用 256 字节的 8 位随机存储器(RAM)与 CPU 交换数据,经常把这种规格的存储器称作 256×8 位读/写存储器。

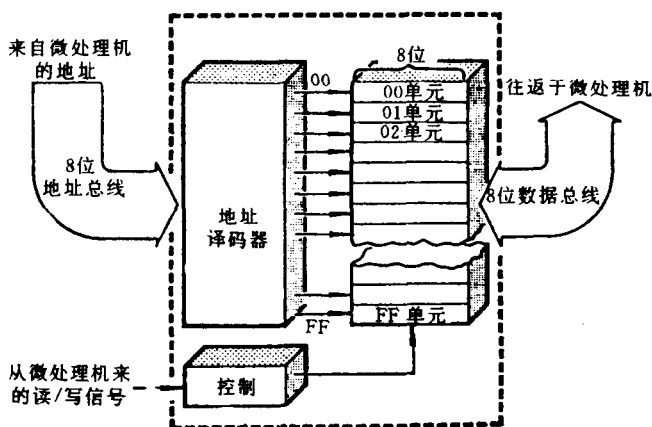


图 1-4 随机存取存储器

两根 8 位总线和若干控制线把存储器和微处理器(机)连接起来。地址总线将一组 8 位二进制数(能表示 256 个单元)从 CPU 送到存储器的地址译码器。每个存储单元被赋予一个唯一的地址,规定第一单元地址为 0,最后一单元的地址为 255(用二进制表示为 11111111B,用十六进制表示为 FFH)。在地址总线上,通过 8 位地址线选择指定的单元。地址译码器的输出可以唯一确定被选择的存储单元。

存储器还从 CPU 接收控制信号,从而确定存储器执行何种操作。“读”信号表明要读出被

选单元的内容,并将数据放到数据总线上,由总线送到 CPU。“写”信号表明要把数据总线上的数据写入指定的存储单元中。

二、I/O 接口及外设

从图 1-2 可以看到,I/O 接口与地址总线、数据总线的连接同存储器一样,而每个外部设备与微处理器的连接必须经过接口适配器(I/O 接口)。每个 I/O 接口及其对应的外部设备都有一个固定的地址,在 CPU 的控制下实现对外部设备的输入(读)和输出(写)操作。

1.2 微机的工作过程

计算机采取“存储程序”的工作方式,即事先把程序加载到计算机的存储器中,当启动运行后,计算机便自动进行工作。计算器虽然也有运算和控制的功能,但它不是“存储程序”式的自动工作方式,所以不能称为计算机。

任何计算机都有它的指令系统,有十几条至一百多条指令,并有若干种寻址方式。我们假设图 1-2 所示的模型计算机有四条指令,并只有一种寻址方式——直接寻址方式,模型机的指令及其说明如表 1-1 所示。

表 1-1 模型机指令表

名称	助记符	操作码	注 释
取入累加器	LDA	1001 0110 (96H)	将存储单元的内容取入累加器,其单元地址由下一个字节给出。
加法	ADD	1001 1011 (9BH)	将存储单元的内容和累加器的现有内容相加,结果放在累加器中,存储单元的地址由下一字节给出。
累加器送存	STA	1001 0111 (97H)	累加器内容送存,存储单元的地址由下一字节给出。
停机	HLT	0011 1110 (3EH)	停止全部操作。

寻址方式是指用什么方法寻找指令的操作数。上述四条指令除 HLT 外,LDA, ADD 和 STA 都有操作数。直接寻址方式的指令格式如图 1-5 所示。

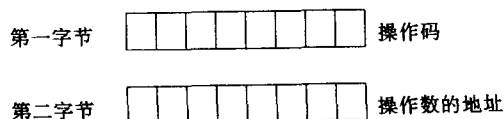


图1-5 直接寻址方式的指令格式

指令中应有一部分数位(8位,即一节)用于指明所执行的特定操作,这部分(图 1-5 中的第一字节)叫操作码。该模型机的操作有数据传送(LDA),相加(ADD),送存(STA)和停机(HLT)四种。它们的操作码如表 1-1 所示。

指令中还应有一部分数位(图 1-5 中的第二字节)用于说明被操作的数据来自什么地方,这一部分叫操作数的地址。

在这种寻址方式中,一条指令(如 LDA,ADD 和 STA)需要两个字节:第一个字节是操作码,第二个字节不是操作数,而是存放操作数的内存单元的地址。例如,指令

LDA 23 ;将地址为 23 的内存单元中的内容 7 装入累加器 A 中。23 为操作数的地址。

在图 1-6 所示的内存单元 23 中存放的 7 为操作数。执行上述指令后就将 7 装入累加器 A 中。

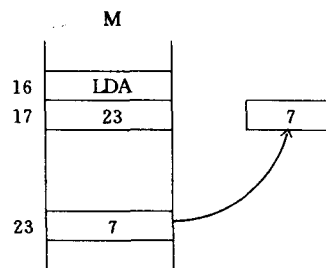


图1-6 执行“LDA 23”指令

1.2.1 执行一条指令的顺序

计算机执行程序是一条指令一条指令执行的。执行一条指令的过程可分为两个阶段,如图 1-7 所示。

在计算机中,“存储程序”第一条指令的第一个字节一定是操作码。这样,CPU 首先进入取指阶段,从存储器中取出指令并通过 CPU 译码后,转入执指阶段,在这期间,CPU 执行指令指定的操作。

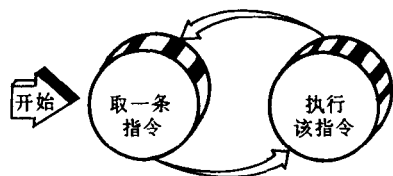


图 1-7 取指令、执行指令序列

取指阶段是由一系列相同的操作组成的,因此,取指阶段的时间总是相同的。而执行指令的阶段是由不同的事件顺序组成的,它取决于被执行指令的类型。执行完一条指令后接着执行下一条指令。所以,程序的执行顺序是取指→执指,取指→执指……如此反复直至程序结束。

1.2.2 执行一条指令的过程

指令“LDA 23”的执行过程是怎样的呢?这是一条直接寻址方式的指令,执行的过程如图 1-8 所示。

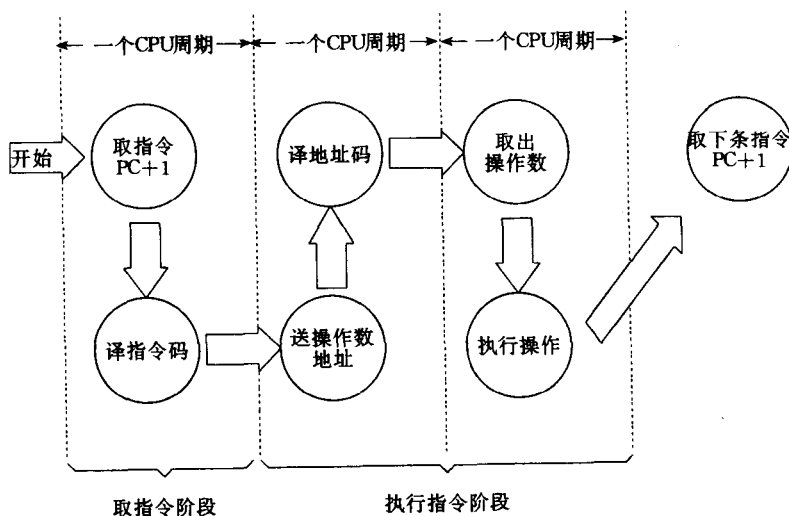


图1-8 直接访问内存指令的指令周期

LDA 指令的指令周期由三个 CPU 周期(即机器周期)组成。其中,第一个 CPU 周期为取指令阶段;执行指令阶段由两个 CPU 周期组成,第二个 CPU 周期中将操作数的地址送往地

址寄存器并完成地址译码,在第三个 CPU 周期中,从内存取出操作数并执行装入的操作。

1.2.3 执行一个程序的过程

采用直接寻址方式,执行一个“7+10”的程序实例如表 1-2 所示。

表 1-2 “7+10”程序执行过程

地 址	内 容	助记符/内容
0001 0000	1001 0110	LDA
0001 0001	0001 0111	23
0001 0010	1001 1011	ADD
0001 0011	0001 1000	24
0001 0100	1001 0111	STA
0001 0101	0001 1001	25
0001 0110	0011 1110	HLT
0001 0111	0000 0111	7
0001 1000	0000 1010	10
0001 1001	0000 0000	保存和

假如程序存放在起始地址为 00010000B(16)的存储单元中。地址 16 和 17 存放第一条指令“LDA 23”,执行第一条指令的过程如图 1-9 所示。计算机启动运行后,PC 中的地址即为 16,将 16 送地址寄存器,接着 16 被放入地址总线上,找到操作码地址 00010000B(16),PC 自动加 1 为 17,做好取下一字节的准备;取出“LDA 23”的操作码 10010110B 放入数据总线;操作码经数据总线装入数据寄存器,因为是操作码,所以还需要装入指令译码器进行指令译码,得到“装入”的操作。

此时,PC 中内容已是 17,地址 17 送入地址寄存器并放到地址总线,找到操作码地址 23,PC 又自动加 1,做好取第二条指令“ADD 24”的准备。如图 1-10 所示。

找到操作数地址 23 后,因为是直接寻址,取出 23 单元中的操作数 7 放到数据总线上,再装入数据寄存器中,经数据寄存器将操作数 7 装入累加器。如图 1-11 所示。至此,第一条指令“LDA 23”执行完毕。

第二、三条指令也是直接寻址方式,执行的过程与第一条类似,这里不再详述。最后一条指令 HLT 为固有寻址,无操作数,取出译码结果后执行停机操作,这个程序就执行完了。