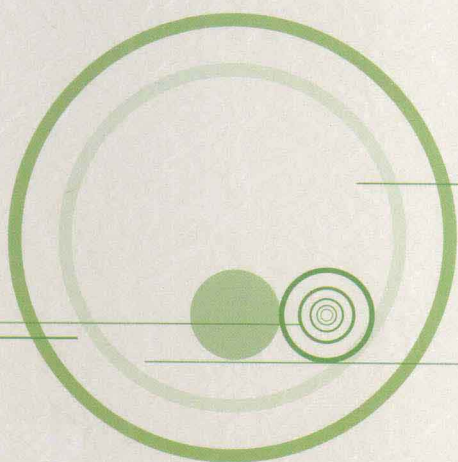




普通高等教育“十一五”国家级规划教材
教育部“2008年度普通高等教育精品教材”

单片机原理及接口技术

(第4版)



MCU

MCU

MCU

李朝青 刘艳玲 编著



配有课件



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS



普通高等教育“十一五”国家级规划
教育部“2008年度普通高等教育精品教材”

单片机原理及接口技术

(第4版)

李朝青 刘艳玲 编著

北京航空航天大学出版社

内 容 简 介

本书以 89C51/S51 为典型机,深入浅出地讲述单片机原理、接口及应用技术。主要内容包括:微机基础知识、89C51/S51 单片机硬件结构、指令系统、汇编语言程序设计知识、中断系统、定时器及应用、89C51/S51 串行口通信及串行通信技术、89C51/S51 单片机小系统及片外扩展、应用系统配置及接口技术、系统应用程序实例和 C51 程序设计。本书对第 3 版教材中的部分内容进行更新和修订,删去 EPROM 扩展口及片外 RAM 芯片扩展的内容,用户可根据需要选择 89 系列不同容量 Flash ROM 的产品;增加了无线单片机及其点到多点无线通信、RFID 技术与物联网的应用以及 C51 程序设计的内容。

本书内容新颖、实用,可用作大中专院校微机原理、单片机及接口技术的教材,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, 刘艳玲编著. --
4 版. --北京:北京航空航天大学出版社,2013.7
ISBN 978-7-5124-1190-6

I. ①单… II. ①李… ②刘… III. ①单片微型计算机—基础理论—教材②单片微型计算机—接口—教材
IV. TP368.1

中国版本图书馆 CIP 数据核字(2013)第 150055 号

版权所有,侵权必究。

单片机原理及接口技术(第 4 版)

李朝青 刘艳玲 编著

责任编辑 刘 星

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:20.75 字数:442 千字

2013 年 7 月第 4 版 2013 年 7 月第 1 次印刷 印数:10 000 册

ISBN 978-7-5124-1190-6 定价:36.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

前 言

《单片机原理及接口技术(第3版)》自2005年出版以来,得到各大中专院校同行的认可,在此表示诚挚的谢意。本书2006年被教育部评为普通高等教育“十一五”国家级规划教材,2008年又被教育部评为“教育部2008年度普通高等教育精品教材”。从1999年第1版至今,现已第39次印刷。鉴于单片机及嵌入式系统技术发展迅速,该教材很多内容已经陈旧,现决定对本教材再次修订。

这次修订内容变化较大,从以往的89C51芯片为主转变为以89C51/S51芯片为主的典型机。89C51/S51都具有80C51内核,仍属于MCS-51系列单片机,简称51系列单片机。89S51片内具有硬件看门狗,抗干扰功能更强。本书删去EPROM扩展口及片外RAM芯片扩展的内容,用户可根据需要选择89系列不同容量Flash ROM的产品。为了节省89C51/S51的I/O口线,选择了一些串口(SPI或I²C)A/D、D/A、E²PROM、看门狗、键盘和显示器的实例。这样,89C51/S51在不扩展片外I/O口芯片的情况下,即可构成完整的测控系统。本书还增加了无线单片机及其点到多点无线通信、RFID技术与物联网的应用以及C51程序设计的内容。

各章习题与原教材区别不大,其习题解答及题库可在北京航空航天大学出版社2005年出版的《单片机学习指导》(与本书配套)一书中找到,敬请关注。

参加本教程编写的还有刘晓培(天津商业大学)、孙学超、曹玉珍(天津大学)、王志勇、沈怡琳、袁其平、张秋燕、曹文嫣、李运等。

由于作者水平所限,难免出现错误和不妥之处,敬请同行及读者提出宝贵意见。

李朝青

天津理工大学电信学院

2013年5月

本教材还配有教学课件,需要用于教学的教师,请与北京航空航天大学出版社联系。北京航空航天大学出版社联系方式如下:

通信地址:北京市海淀区学院路37号北京航空航天大学出版社嵌入式系统事业部
邮编:100191

电话:010-82317035

传真:010-82328026

E-mail:emsbook@gmail.com

目 录

第 1 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.....	5
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	7
1.2.3 执行一个程序的过程	8
1.3 常用数制和编码.....	11
1.3.1 数制及数制间转换.....	12
1.3.2 计算机中常用编码.....	18
1.4 数据在计算机中的表示.....	19
1.4.1 有符号数.....	19
1.4.2 无符号数.....	20
1.5 89C51/S51 单片机.....	21
1.5.1 AT89C51/S51 系列单片机	22
1.5.2 STC89 系列单片机	23
1.5.3 SST89 系列单片机	23
1.6 思考题与习题.....	24
第 2 章 89C51/S51 单片机的硬件结构和原理	25
2.1 89C51/S51 单片机的内部结构及特点	25
2.1.1 89C51/S51 单片机的基本组成.....	25
2.1.2 89C51/S51 单片机芯片内部结构.....	26
2.2 89C51/S51 单片机的引脚及其功能.....	29
2.3 89C51/S51 单片机的存储器配置.....	34
2.3.1 程序存储器地址空间.....	35
2.3.2 数据存储器地址空间.....	37
2.4 89C51/S51 CPU 时序	44
2.5 复位操作.....	46

2.5.1	复位操作的主要功能	46
2.5.2	复位电路	48
2.6	思考题与习题	49
第3章	指令系统	50
3.1	汇编语言	50
3.1.1	指令和程序设计语言	50
3.1.2	指令格式	51
3.2	寻址方式	52
3.2.1	7种寻址方式	53
3.2.2	寻址空间及符号注释	57
3.3	89C51/S51单片机的指令系统	58
3.3.1	数据传送指令	59
3.3.2	算术运算指令	63
3.3.3	逻辑操作指令	67
3.3.4	控制程序转移类指令	69
3.3.5	位操作(布尔处理)类指令	77
3.4	思考题与习题	80
第4章	汇编语言程序设计知识	85
4.1	编程的步骤、方法和技巧	85
4.1.1	编程步骤	85
4.1.2	编程的方法和技巧	87
4.1.3	汇编语言程序的基本结构	88
4.2	伪指令	96
第5章	中断系统	99
5.1	微机的输入/输出方式	99
5.1.1	无条件传送方式	99
5.1.2	查询传送方式	99
5.1.3	直接存储器存取(DMA)方式	100
5.2	中断的概念	100
5.3	89C51/S51中断系统结构及中断控制	102
5.3.1	89C51/S51中断源	103
5.3.2	中断控制	103
5.4	中断响应及中断处理过程	108

5.4.1	中断响应	108
5.4.2	中断处理	110
5.4.3	中断返回	111
5.4.4	关于具体的中断服务程序	111
5.5	中断程序举例	112
5.5.1	主程序	112
5.5.2	中断服务程序	113
5.6	思考题与习题	116
第 6 章	定时器及应用	118
6.1	定时器概述	118
6.1.1	什么是计数和定时	118
6.1.2	定时器/计数器的组成	119
6.2	定时器的控制	120
6.2.1	工作模式寄存器 TMOD	120
6.2.2	控制寄存器 TCON	121
6.3	定时器的 4 种模式及应用	122
6.3.1	模式 1 及应用	123
6.3.2	模式 2 及应用	123
6.3.3	模式 3 及应用	126
6.3.4	综合应用举例	128
6.4	思考题与习题	134
第 7 章	89C51/S51 串行口及串行通信技术	136
7.1	串行通信基本知识	136
7.1.1	数据通信	136
7.1.2	串行通信的传输方式	137
7.1.3	异步通信和同步通信	138
7.1.4	串行通信的过程及通信协议	140
7.2	串行口及应用	142
7.2.1	89C51/S51 串行口	142
7.2.2	89C51/S51 串行口的应用	153
7.3	89C51/S51 与 89C51/S51 点对点异步通信	158
7.3.1	通信协议	158
7.3.2	波特率设置	158
7.3.3	通信程序举例	160

7.4	89C51/S51 与 PC 机间通信	164
7.4.1	单片机与 PC 机通信的接口电路	164
7.4.2	PC 机通信软件	165
7.4.3	89C51 通信软件设计	170
7.5	无线单片机及其点到多点无线通信	174
7.5.1	无线单片机	174
7.5.2	无线单片机实现点到多点的无线通信	175
7.5.3	多点无线测温系统	176
7.6	RFID 技术与物联网的应用	178
7.6.1	物联网定义	178
7.6.2	RFID 技术	178
7.7	思考题与习题	181
第 8 章	单片机小系统及片外扩展	182
8.1	串行扩展总线接口技术	182
8.1.1	SPI 串行外设接口总线	182
8.1.2	I ² C 总线	188
8.1.3	单总线	188
8.2	并行扩展三总线的产生	194
8.2.1	片外三总线结构	194
8.2.2	系统扩展的实现	195
8.3	扩展数据存储器	196
8.4	简单并行 I/O 口的扩展	196
8.4.1	I/O 口的直接输入/输出	196
8.4.2	简单 I/O 接口的扩展方法	197
8.5	思考题与习题	199
第 9 章	应用系统配置及接口技术	201
9.1	人-机通道配置与接口技术	201
9.1.1	键盘接口及处理程序	201
9.1.2	LED 显示器接口及显示程序	209
9.1.3	串行口控制的键盘/LED 显示器接口电路	213
9.2	系统前向通道中的 A/D 转换器及接口技术	217
9.2.1	8 位串行 A/D 芯片 TLC0831 与单片机接口及编程	217
9.2.2	8 位 2 通道串行 A/D 芯片 ADC0832 与单片机接口及编程	219
9.2.3	10 位单通道串行输出 A/D 芯片 TLC1549 接口及编程	221

9.2.4	12 位串行 A/D 芯片 AD7893 与单片机接口及编程	224
9.2.5	16 位低速串行 A/D 芯片 AD7705 与单片机接口及编程	227
9.2.6	16 位高速串行 A/D 芯片 AD7683 与单片机接口及编程	232
9.2.7	8 位并行输出 A/D 芯片 ADC0809 与单片机接口及编程	236
9.3	系统后向通道配置及接口技术	239
9.3.1	后向通道中的功率开关器件及接口	240
9.3.2	双向晶闸管(可控硅)温度控制系统	243
9.3.3	串行输入 D/A 芯片 TLC5615 接口技术	243
9.3.4	并行输入 D/A 芯片及接口技术	248
9.4	思考题与习题	252
第 10 章	系统实用程序	254
10.1	主程序和子程序的概念	254
10.1.1	主程序	254
10.1.2	子程序及参数传递	254
10.1.3	中断服务子程序	256
10.2	数据采集及简单控制程序	257
10.2.1	数据采集程序	257
10.2.2	航标灯控制程序	257
10.2.3	水位控制程序	259
10.2.4	蜂鸣音报警子程序	262
10.3	数据处理程序	262
10.3.1	排序程序	263
10.3.2	数字滤波程序	263
10.3.3	标度变换(工程量变换)	264
10.4	代码转换程序	267
10.5	抗干扰技术	269
10.5.1	软件陷阱技术	269
10.5.2	软件看门狗	272
10.5.3	单片机片内硬件看门狗	274
10.6	最短程序	276
第 11 章	C51 程序设计	277
11.1	C51 程序设计基础	277
11.1.1	C51 的标识符和关键字	277
11.1.2	C51 的数据类型	279

11.1.3	C51 变量的存储方式	280
11.1.4	C51 运算符、表达式及规则	284
11.2	C51 语句	286
11.2.1	说明语句	286
11.2.2	表达式语句	286
11.2.3	复合语句	287
11.2.4	条件语句	287
11.2.5	开关与跳转语句	288
11.2.6	循环语句	290
11.2.7	函数调用语句	291
11.2.8	返回语句	291
11.2.9	空语句	292
11.3	C51 的流程控制结构	292
11.4	C51 函数	292
11.4.1	函数的定义	293
11.4.2	函数的调用	293
11.5	数组和指针	297
11.5.1	数组的定义和引用	297
11.5.2	字符数组	298
11.5.3	数组元素赋初值	299
11.5.4	数组作为函数的参数	299
11.5.5	指针的概念	300
11.5.6	数组的指针	303
11.5.7	指针的地址计算	304
11.5.8	指针数组与指针型指针	305
11.6	思考题与习题	306
附录 A	80C51 指令表	308
附录 B	89C51 指令矩阵(汇编/反汇编表)	313
附录 C	8255A 可编程外围并行接口芯片及接口	314
参考文献		322

第 1 章 微机基础知识

1.1 微处理器、微机和单片机的概念

首先介绍一下微处理器(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,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)。

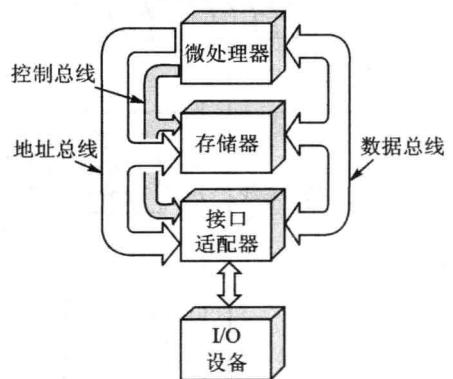


图 1-1 微机的组成

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。

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 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(bit)宽度,即要求多数主要寄存器和存储器能保存 8 位数据,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。

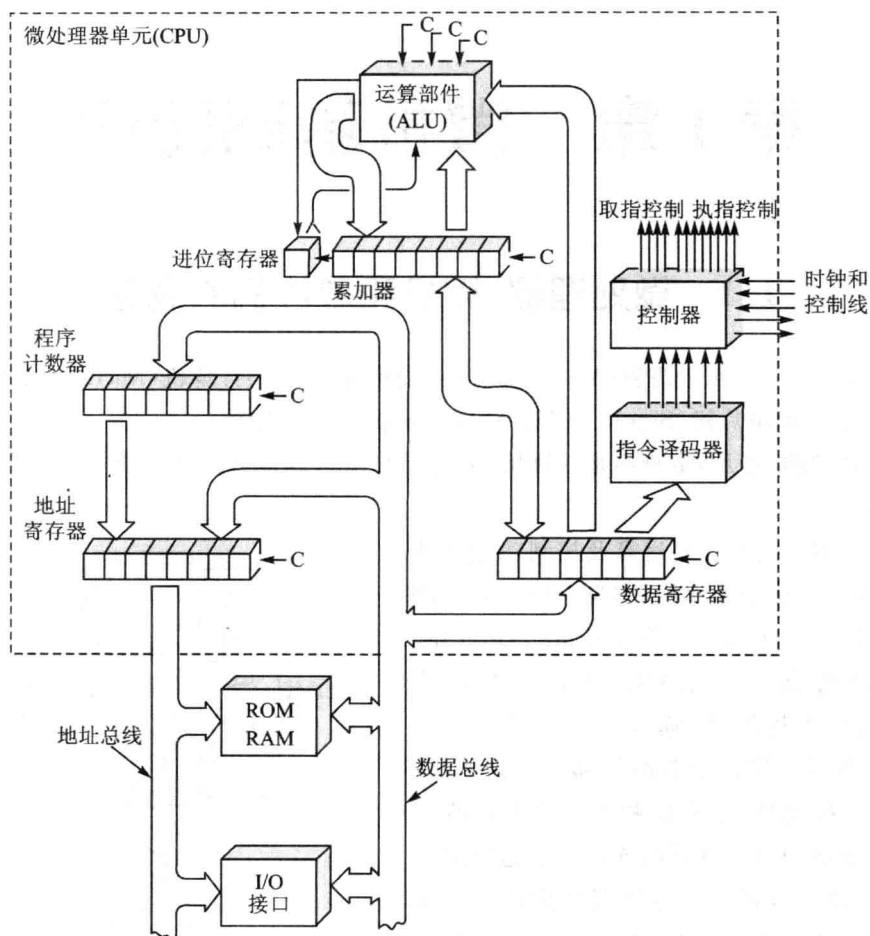


图 1-2 一个计算机模型

在 8 位微处理器中,数据单元由 1 字节(Byte)组成;在 16 位机中,数据单元由 2 字节组成。图 1-3 表示了组成计算机数据单元的位数。

1. 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 具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。

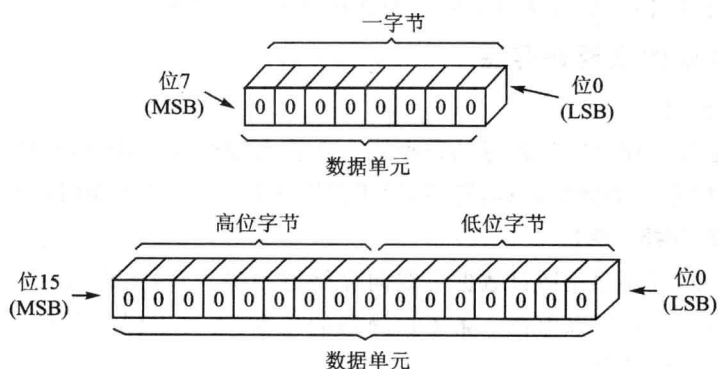


图 1-3 计算机中的数据单元

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、或和比较大小等操作,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:

- 执行各种算术运算;
- 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。

2. 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:

- 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;
- 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;
- 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操作,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、寄存器和控制器除在微处理器内通过内部总线相互联系外,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。

3. CPU 中的主要寄存器

1) 累加器(A)

累加器是微处理器中最忙碌的寄存器。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。

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1-2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,即程序第1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与存储器进行信息交换(即CPU从RAM存/取数据,或者CPU从ROM中读出指令)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

1. 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。

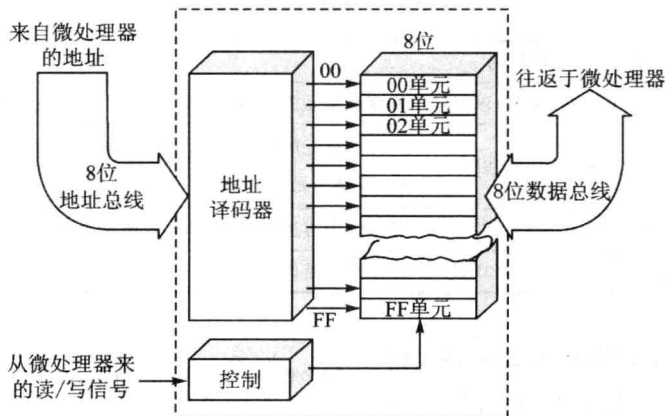


图 1-4 随机存取存储器

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。

2. 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 4 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 所列。

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 4 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。

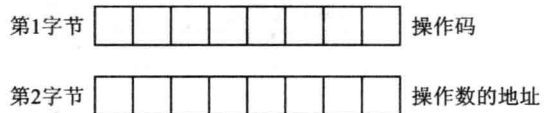


图 1-5 直接寻址方式的指令格式

指令中应有一部分数位(8 位,即 1 字节)用于指明所执行的特定操作,这部分(图 1-5 中的第 1 字节)称为操作码。该模型机的操作有数据传送(LDA)、相加(ADD)、送存(STA)和停机(HLT)4 种。它们的操作码如表 1-1 所列。

指令中还应有一部分数位(图 1-5 中的第 2 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。

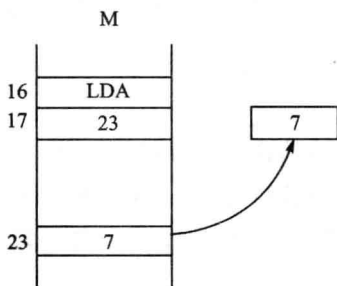


图 1-6 执行“LDA 23”指令

在这种寻址方式中,一条指令(如 LDA、ADD 和 STA)需要 2 个字节:第 1 个字节是操作码,第 2 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:

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。

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 1 条指令的第 1 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,转入执行阶段,在这期间,CPU 执行指令指定的操作。

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。

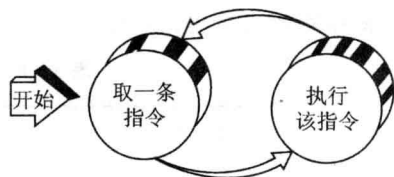


图 1-7 取指令、执行指令序列

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。

LDA 指令的指令周期由 3 个 CPU 周期(即机器周期)组成。其中,第 1 个 CPU 周期为取指令阶段;执行指令阶段由 2 个 CPU 周期组成,第 2 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 3 个 CPU 周期中,从内存取出操作数并执行装入的操作。

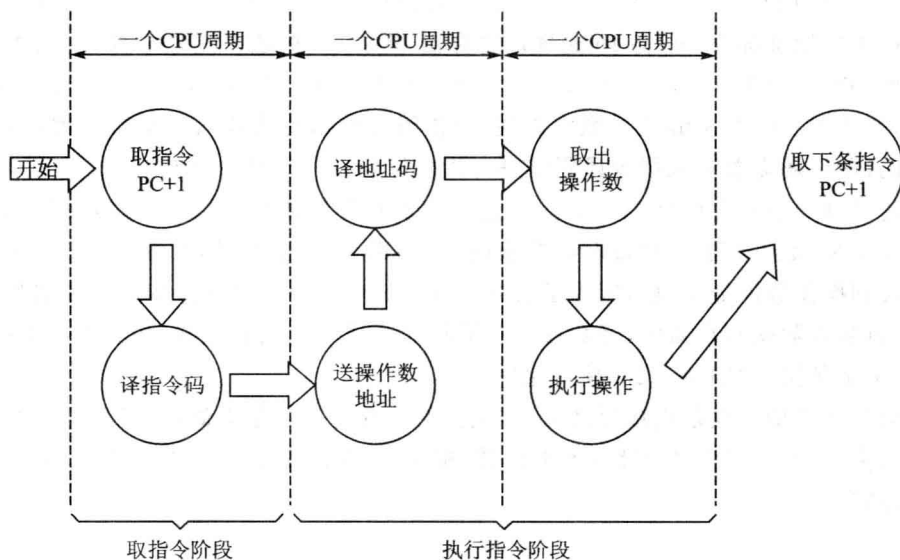


图 1-8 直接访问内存指令的指令周期