



高职高专“十二五”规划示范教材

# 单片机原理及接口技术

## (高职高专版)

李朝青 编著

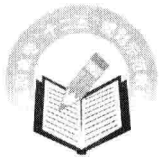
- 由“精品教材”补充、完善和优化而成
- 基础知识+实训案例
- 讲解透彻，学以致用
- 配有课件，易于教学



北京航空航天大学出版社  
BEIHANG UNIVERSITY PRESS



配有课件



高职高专“十二五”规划示范教材

# 单片机原理及接口技术

## (高职高专版)

李朝青 编著

北京航空航天大学出版社

## 内 容 简 介

本书以 89C51 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 单片机硬件结构、指令系统、汇编语言程序设计知识、中断与定时器、89C51 串行口及串行通信、89C51 小系统及片外扩展、应用系统配置及接口技术和系统应用程序实例。

该书为教育部 2008 年度普通高等教育“精品教材”——《单片机原理及接口技术(第 3 版)》精简更新而成,称高职高专版。书中增加了一些串行外设芯片扩展的内容,如 SPI、串行 A/D、D/A、键盘和显示器等实例;同时增加了 8 个实训内容。本书可用作高职高专院校微机原理、单片机原理及应用课程的教材,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:高职高专版 / 李朝青编著

—北京:北京航空航天大学出版社,2011.1

ISBN 978-7-5124-0257-7

I. ①单… II. ①李… III. ①单片微型计算机—基础理论—高等学校:技术学校—教材②单片微型计算机—接口—高等学校:技术学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2010)第 216096 号

版权所有,侵权必究。

### 单片机原理及接口技术(高职高专版)

李朝青 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

北京时代华都印刷有限公司印装 各地书店经销

\*

开本:787×960 1/16 印张:17.5 字数:392 千字

2011 年 1 月第 1 版 2011 年 1 月第 1 次印刷 印数:5 000 册

ISBN 978-7-5124-0257-7 定价:29.00 元

## 前 言

本教材结合我国高职高专教学改革的实践,针对高职高专学生的学习特点,在原有李朝青编著的《单片机原理及接口技术(第3版)》(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)的基础上,进行了精简、更新和改编。诸作者在广泛调研的基础上,经过与多所高职院校教学一线教师的深入讨论,对原有教材的内容进行了有机整合,降低了理论难度,丰富了实训内容。

本教材在内容的选择和讲解方面,遵循“必需、够用”的原则,以当前高等职业院校学生对知识实际接受能力和就业技能要求为依据,力求体现针对性和实用性。本教材通过增加实训内容着重培养学生独立分析实际问题和解决实际问题的能力,增强学生的创新意识和团队精神,为就业打下坚实的基础。

该书以89C51为典型机讲述单片机原理及接口技术。删去了传统的EPROM扩展及I/O口线扩展的内容,用户可根据需要选择89系列不同容量Flash的产品。为了节省89C51的口线,增加了一些串行口外设芯片扩展的内容,如SPI、A/D、D/A、E<sup>2</sup>PROM、键盘和显示器的实例。这样,89C51在不扩展I/O口线及EPROM芯片的情况下即可构成完整的测控系统。

本教材由天津理工大学李朝青教授担任主编,参加编写的还有刘艳玲、沈怡琳、贾宝会、张文、王志勇、袁其平、曹文嫣、张秋燕、李运等。

为了方便教师教学,本教材还提供了丰富的电子课件、习题答案、考题库及解答(在本书的配套教材《单片机学习指导》一书中)。需要用于教学的教师,请与北京航空航天大学出版社联系。

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

2010年11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/传真:010-82317035/82317034

E-mail: [emsbook@gmail.com](mailto: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	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	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 单片机	21
1.6 思考题与习题	22
1.7 实训一 Keil $\mu$ Vision2 集成开发环境简介	23
1.7.1 实训目的	23
1.7.2 实训说明	23
1.7.3 实训内容及步骤	24
第 2 章 单片机的硬件结构和原理	28
2.1 89C51 单片机的内部结构及特点	28
2.1.1 单片机的基本组成	28
2.1.2 89C51 单片机芯片内部结构	29
2.2 89C51 单片机的引脚及其功能	32
2.3 89C51 单片机的存储器配置	36
2.3.1 程序存储器地址空间	37

2.3.2 数据存储器地址空间·····	38
2.4 89C51 CPU 时序·····	45
2.5 复位操作·····	47
2.5.1 复位操作的主要功能·····	47
2.5.2 复位电路·····	49
2.6 思考题与习题·····	49
2.7 实训二 硬件实时在线仿真·····	50
2.7.1 实训目的·····	50
2.7.2 实训说明·····	50
2.7.3 实训内容及步骤·····	51
<b>第3章 89C51 单片机指令系统</b> ·····	<b>54</b>
3.1 汇编语言·····	54
3.1.1 指令和程序设计语言·····	54
3.1.2 指令格式·····	54
3.2 寻址方式·····	55
3.2.1 7种寻址方式·····	56
3.2.2 寻址空间及符号注释·····	60
3.3 89C51 单片机的指令系统·····	61
3.3.1 数据传送指令·····	61
3.3.2 算术运算指令·····	66
3.3.3 逻辑操作指令·····	69
3.3.4 控制程序转移类指令·····	71
3.3.5 位操作(布尔处理)类指令·····	80
3.4 思考题与习题·····	82
3.5 实训三 片内外 RAM 间数据的传送·····	87
3.5.1 实训目的·····	87
3.5.2 实训说明·····	87
3.5.3 实训内容及步骤·····	87
<b>第4章 汇编语言程序设计知识</b> ·····	<b>89</b>
4.1 编程的步骤、方法和技巧·····	89
4.1.1 编程步骤·····	89
4.1.2 编程的方法和技巧·····	91

---

4.1.3 汇编语言程序的基本结构	92
4.2 伪指令	100
4.3 实训四 I/O 口的简单输出使用	102
4.3.1 实训目的	102
4.3.2 实训说明	103
4.3.3 实训内容及步骤	103
4.3.4 思考题	104
4.3.5 实训电路图	104
<b>第 5 章 中断系统</b>	<b>105</b>
5.1 微机的输入/输出方式	105
5.1.1 无条件传送方式	105
5.1.2 查询传送方式	105
5.1.3 直接存储器存取(DMA)方式	106
5.2 中断的概念	106
5.3 89C51 中断系统结构及中断控制	107
5.3.1 89C51 中断源	108
5.3.2 中断控制	109
5.4 中断响应及中断处理过程	114
5.4.1 中断响应	114
5.4.2 中断处理	116
5.4.3 中断返回	117
5.4.4 关于具体的中断服务程序	117
5.5 中断程序举例	118
5.5.1 主程序	118
5.5.2 中断服务程序	119
5.6 定时器及其控制	122
5.6.1 工作模式寄存器 TMOD	123
5.6.2 控制寄存器 TCON	124
5.7 定时器的 4 种模式及应用	125
5.7.1 模式 1 及应用	126
5.7.2 模式 2 及应用	127
5.7.3 模式 3 及应用	127
5.8 思考题与习题	129

5.8.1	中断习题	129
5.8.2	定时器习题	130
5.9	实训五 定时器的使用	131
5.9.1	实训目的	131
5.9.2	实训说明	131
5.9.3	实训内容及步骤	132
5.9.4	思考题	133
5.9.5	实训电路图	133
<b>第6章</b>	<b>89C51 串行口及串行通信技术</b>	<b>134</b>
6.1	串行通信基本知识	134
6.1.1	数据通信	134
6.1.2	串行通信的传输方式	135
6.1.3	异步通信和同步通信	136
6.1.4	串行通信的过程及通信协议	138
6.2	串行口及应用	139
6.2.1	89C51 串行口	140
6.2.2	89C51 串行口的应用程序	151
6.3	89C51 与 89C51 点对点异步通信	156
6.3.1	通信协议	156
6.3.2	波特率设置	157
6.3.3	通信程序举例	159
6.4	89C51 与 PC 机间通信	163
6.4.1	单片机与 PC 机通信的接口电路	163
6.4.2	PC 机及单片机通信程序	163
6.5	思考题与习题	164
6.6	实训六 89C51 与 PC 机串行口通信	165
6.6.1	实训目的	165
6.6.2	实训说明	165
6.6.3	实训内容及步骤	165
6.6.4	实训电路图	166
<b>第7章</b>	<b>单片机小系统及片外扩展</b>	<b>168</b>
7.1	串行扩展总线接口技术	168



---

7.1.1	SPI 串行外设接口总线 .....	168
7.1.2	I <sup>2</sup> C 总线 .....	173
7.1.3	单总线 .....	174
7.2	并行扩展三总线的产生 .....	179
7.2.1	片外三总线结构 .....	180
7.2.2	系统扩展的实现 .....	180
7.3	扩展数据存储器 .....	181
7.3.1	常用的数据存储器芯片 .....	181
7.3.2	访问片外 RAM 的操作时序 .....	183
7.3.3	89C51 扩展 2 KB RAM .....	185
7.4	思考题与习题 .....	185
7.5	实训七 片外数据存储器扩展 .....	188
7.5.1	实训目的 .....	188
7.5.2	实训说明 .....	188
7.5.3	实训内容与步骤 .....	188
7.5.4	实训电路图 .....	190
<b>第 8 章</b>	<b>应用系统配置及接口技术</b> .....	<b>192</b>
8.1	人-机通道配置与接口技术 .....	192
8.1.1	键盘接口及处理程序 .....	192
8.1.2	LED 显示器接口及显示程序 .....	201
8.1.3	串行口控制的键盘/LED 显示器接口电路 .....	205
8.2	系统前向通道中的 A/D 转换器及接口技术 .....	208
8.2.1	单通道串行输出 A/D 芯片 TLC1549 接口及编程 .....	209
8.2.2	8 位串行 A/D 芯片 TLC0831 与单片机接口及编程 .....	211
8.2.3	逐次逼近型并行输出 A/D 转换器及接口 .....	214
8.3	系统后向通道中的 D/A 转换技术 .....	217
8.3.1	串行输入 D/A 芯片 TLC5615 接口技术 .....	217
8.3.2	并行输入 D/A 芯片及接口技术 .....	221
8.4	思考题与习题 .....	226
8.5	实训八 A/D、D/A 接口实训 .....	227
8.5.1	实训目的 .....	227
8.5.2	实训说明 .....	227
8.5.3	实训内容与步骤 .....	228

8.5.4	实训电路图 .....	228
8.5.5	思考题 .....	229
<b>第9章</b>	<b>系统实用程序 .....</b>	<b>231</b>
9.1	主程序和子程序的概念 .....	231
9.1.1	主程序 .....	231
9.1.2	子程序及参数传递 .....	231
9.1.3	中断服务子程序 .....	234
9.2	数据采集及简单控制程序 .....	234
9.2.1	数据采集程序 .....	234
9.2.2	航标灯控制程序 .....	235
9.2.3	水位控制程序 .....	237
9.2.4	蜂鸣音报警子程序 .....	239
9.3	数据处理程序 .....	240
9.3.1	排序程序 .....	240
9.3.2	数字滤波程序 .....	241
9.3.3	标度变换(工程量变换) .....	242
9.4	代码转换程序 .....	244
9.5	软件抗干扰 .....	246
9.5.1	软件陷阱技术 .....	247
9.5.2	软件看门狗 .....	249
9.5.3	硬件(专用芯片)看门狗 .....	251
9.6	最短程序 .....	254
<b>附录 A</b>	<b>89C51 指令表 .....</b>	<b>255</b>
<b>附录 B</b>	<b>89C51 指令矩阵(汇编/反汇编表) .....</b>	<b>260</b>
<b>附录 C</b>	<b>8255A 可编程外围并行接口芯片及接口 .....</b>	<b>261</b>
	<b>参考文献 .....</b>	<b>269</b>

# 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 $\mu\text{P}$ )、微型计算机(Microcomputer,简称微机, $\mu\text{C}$ )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)。

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。

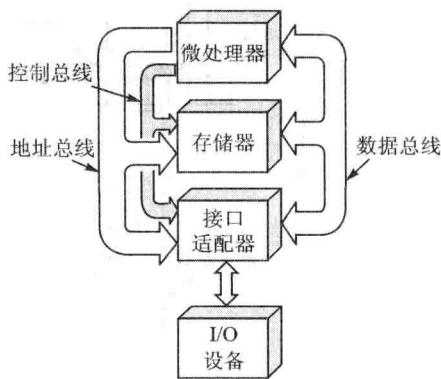


图 1-1 微机的组成

### 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 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。

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

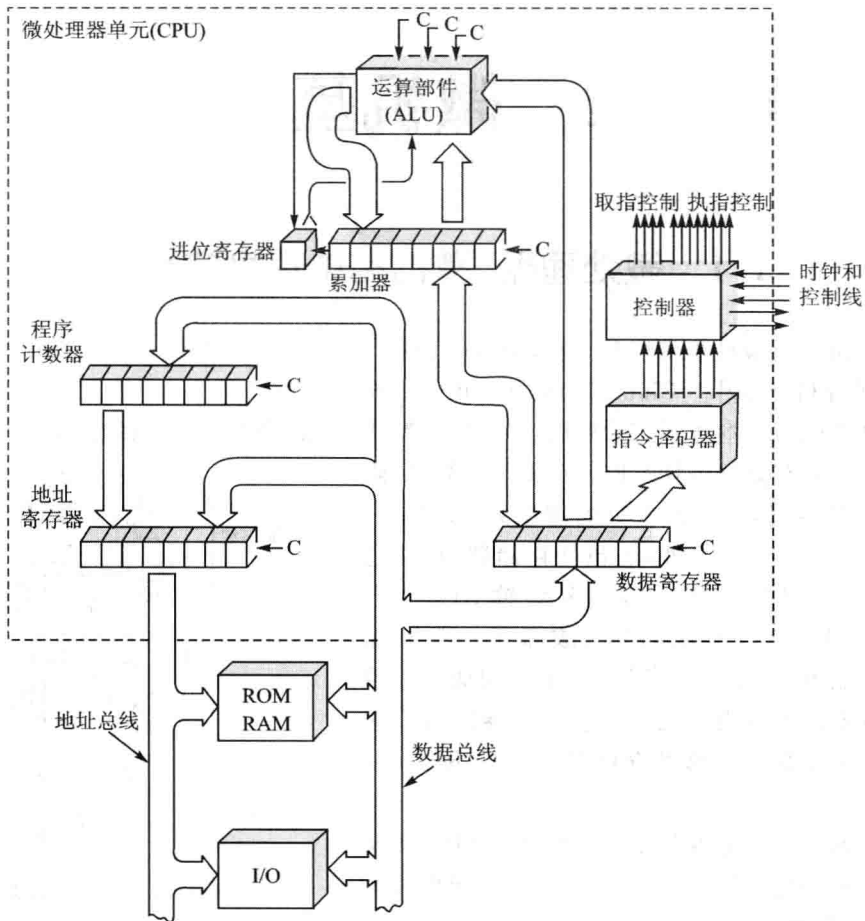


图 1-2 一个计算机模型

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。

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

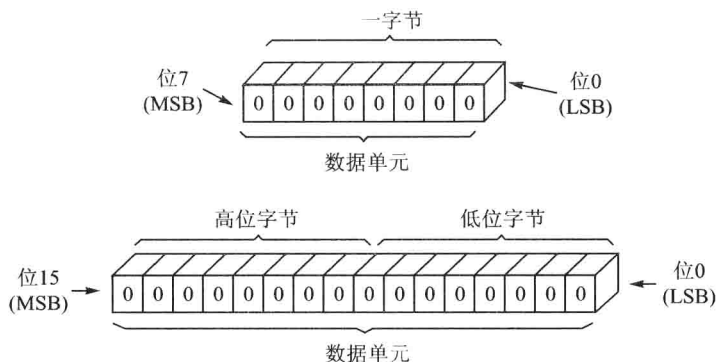


图 1-3 计算机中的数据单元

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 向 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  $256 \times 8$  位读/写存储器。

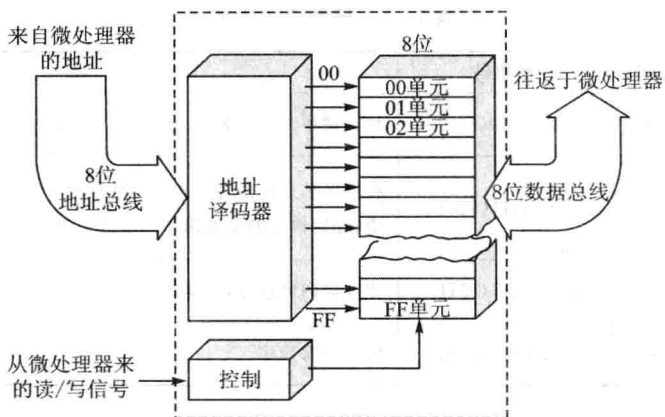


图 1-4 随机存取存储器

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。计算机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。

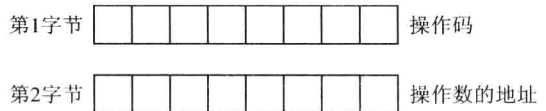


图 1-5 直接寻址方式的指令格式

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。

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,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。

### 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,转入执指阶段,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。

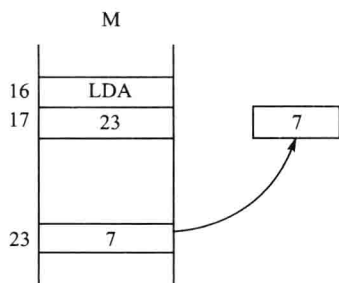


图 1-6 执行“LDA 23”指令

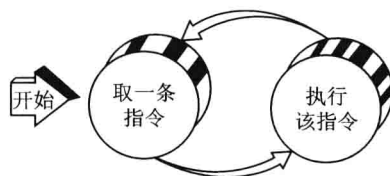


图 1-7 取指令、执行指令序列

## 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。

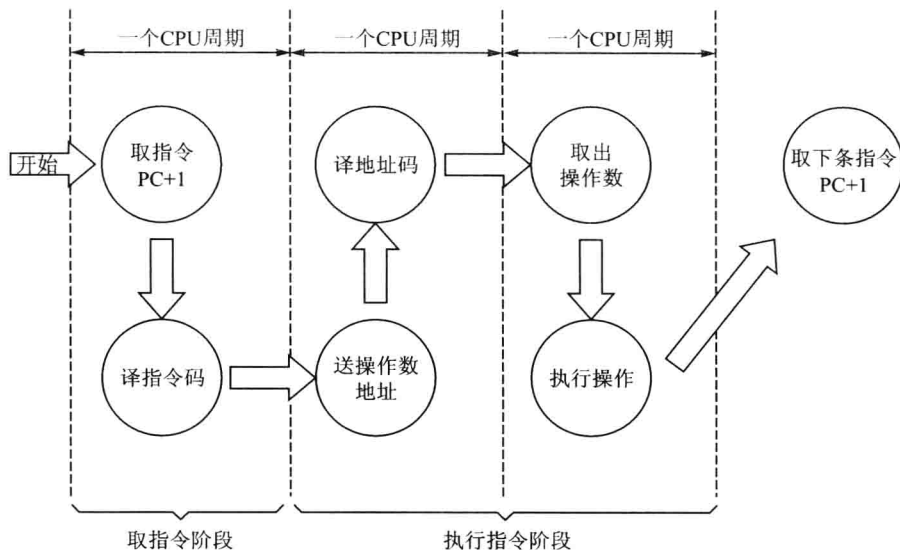


图 1-8 直接访问内存指令的指令周期

LDA 指令的指令周期由 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