



武汉纺织大学
WUHAN TEXTILE UNIVERSITY

单片机原理及应用

课程性质：必修（考试） 学时：48 学分：3
考核方式：闭卷考试



导 言

1. 《单片机原理及应用》难学怎么办？
担心考试不合格怎么办？

2. 为什么要学51单片机？

3. 为什么要学基于汇编语言的编程？

[视频：汇编无用吗？](#)



目 录

第1章	微机基础知识
第2章	89C51单片机硬件结构和原理
第3章	指令系统
第4章	汇编语言程序设计知识
第5章	中断系统
第6章	定时器及应用
第7章	89C51串行口及串行通信技术
第8章	单片机小系统及扩展
第9章	应用系统配置及接口技术
第10章	系统实用程序
第11章	C51程序设计



第一章 微机基础知识

§ 1.1 微处理器、微机和单片机的概念

§ 1.2 微机的工作过程

§ 1.3 常用数制和编码

§ 1.4 数据在计算机中的表示

§ 1.5 89C51单片机

§ 1.6 思考题与习题



§ 1.1 微处理器、微机和单片机的概念

§ 1.1.1 微处理器(机)的组成

§ 1.1.2 存储器和输入输出接口



1. 微处理器
2. 微型计算机
3. 单片机



微处理器

微处理器 (Microprocessor)

是小型计算机或微型计算机的控制和处理部分。

又称中央处理单元CPU (Central Processing Unit) 。



微型计算机

微型计算机 (Microcomputer, 简称微机 MC)

是具有完整运算及控制功能的计算机。

包括 {
 微处理器 (CPU)
 存储器
 接口适配器 (输入输出接口电路)
 输入/输出 (I/O) 设备。

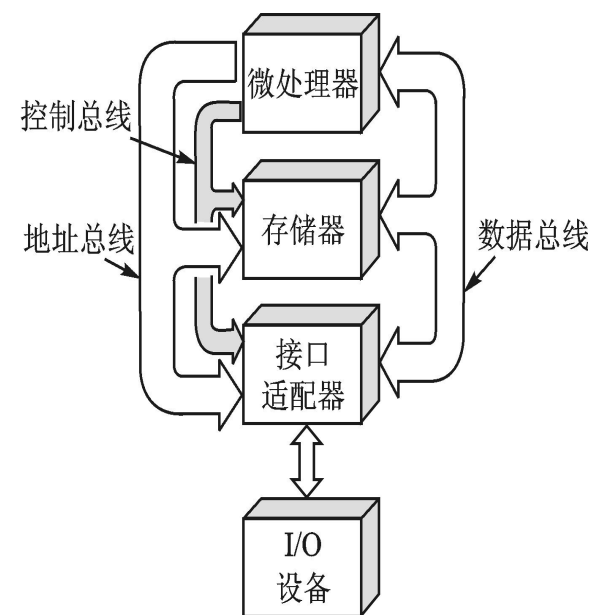


图1-1 微机的组成

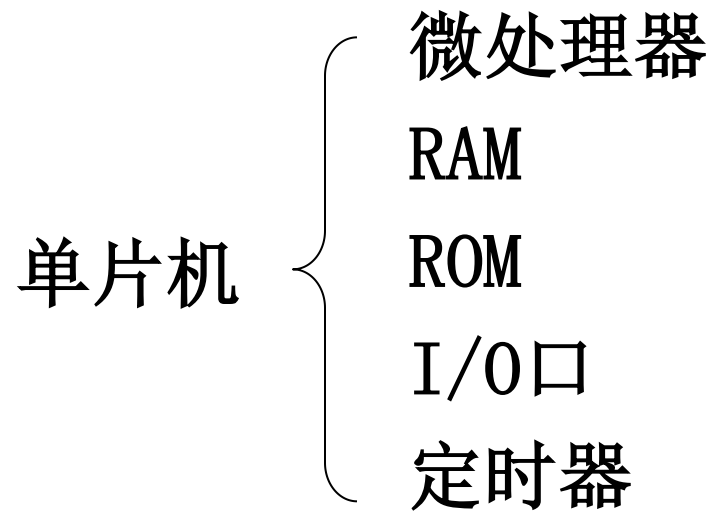
- **微处理器**由控制器、运算器和若干个寄存器组成;
- **存储器**是指微机内部的存储器(RAM、ROM和EPROM等芯片);
- I / O 设备与微处理器的连接需要通过接口适配器(即 **I / O 接口**)。



单片机

单片机 (Single-Chip Microcomputer)

是将微处理器、一定容量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。





计算机的模型

- 微处理单元与存储器及I/O接口组成的计算机模型如图1-2所示。
- 图中只画出CPU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**。
- 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**和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、地址总线和控制总线**，统称为系统总线。
- 存储器包括**RAM和ROM**。
- 微计算机通过输入/输出接口电路可与各种外围设备联接。

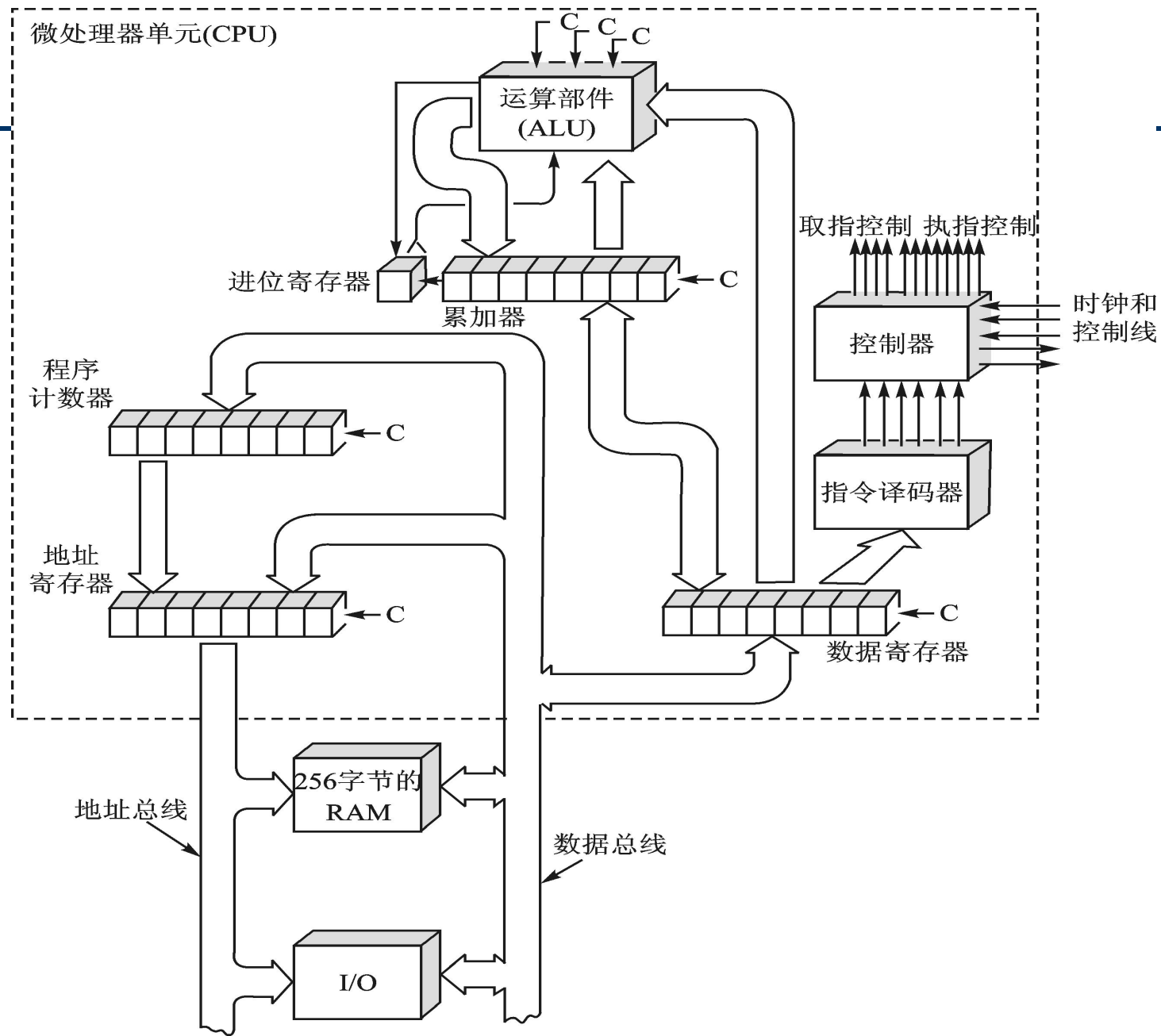


图1-2 一个计算机模型



CPU的组成

1. 运算器
2. 控制器
3. CPU中的主要寄存器

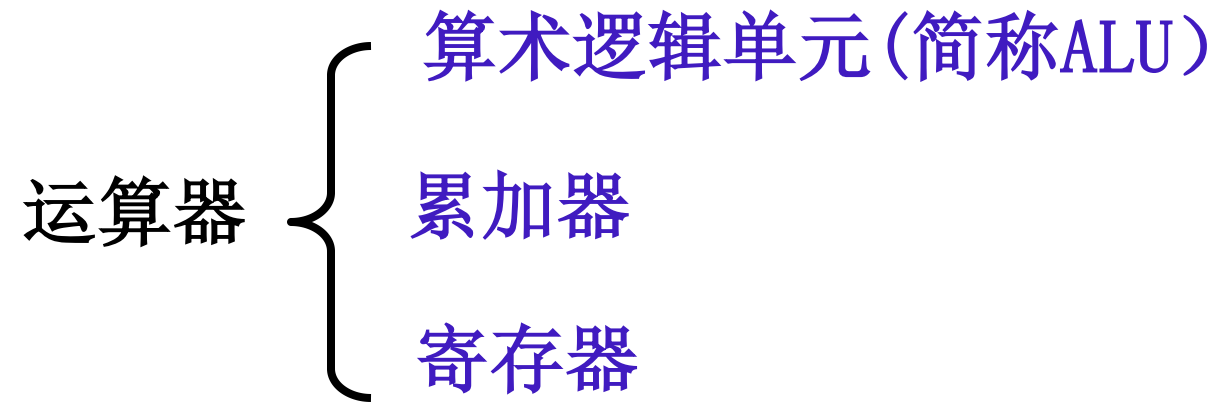


1. 运算器

- 1) 组成
- 2) 作用
- 3) ALU的两个主要的输入来源
- 4) 运算器的两个主要功能



1) 运算器的组成



2) 运算器的作用



- 是把传送到微处理器的数据进行运算或逻辑运算。
- 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。
- ALU执行不同的运算操作是由不同控制线上的信息所确定的。

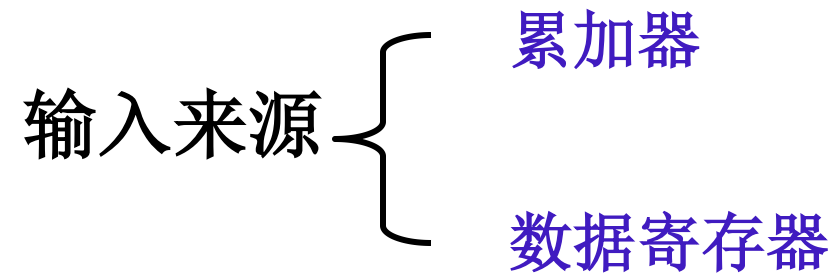


例如：

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结果（16）存入累加器，取代累加器前面存放的数9。



3) ALU的两个主要的输入来源





4) 运算器的两个主要功能

- (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。



2. 控制器

- 1) 控制器的组成
- 2) 控制器的作用
- 3) 控制器的主要功能



1) 控制器的组成

控制器的组成

程序计数器

指令寄存器

指令译码器

时序产生器

操作控制器



2) 作用

它是发布命令的“决策机构”，即协调和指挥整个计算机系统的操作。



3) 控制器的主要功能

- 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，以便启动规定的动作。
- 指挥并控制CPU、内存和输入/输出设备之间数据流动的方向。



3. CPU中的主要寄存器

- 1) 累加器 (A)
- 2) 数据寄存器 (DR)
- 3) 指令寄存器 (IR)
- 4) 指令译码器 (ID)
- 5) 程序计数器 (PC)
- 6) 地址寄存器 (AR)



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（DR）是通过数据总线（DBUS）向存储器（M）和输入/输出设备I/O送（写）或取（读）数据的暂存单元。



3) 指令寄存器 (IR)

- 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，然后再传送到指令译码器中。



4) 指令译码器 (ID)

- 指令分为操作码和地址码字段，由二进制数字组成。当执行任何给定的指令，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，即可向控制器发出具体操作的特定信号。



5) 程序计数器 (PC)

- 通常又称为指令地址计数器。
- 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到PC。
- 当执行指令时，CPU将自动修改PC的内容，使之总是保存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是简单的加1操作。



6) 地址寄存器 (AR)

- 地址寄存器用来保存当前CPU所要访问的内存单元或I/O设备的地址。
- 因为内存（I/O设备）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存地址信息，直到内存（I/O设备）读/写操作完成为止。



§ 1. 1. 2 存储器和输入输出接口

1. 存储器

2. I/O接口及外设



1. 存储器

- 如图1-4所示。
- 地址总线、数据总线和若干控制线把存储器和微处理器连接起来。
- 存储器从CPU接收控制信号，以确定存储器执行读/写操作。
- 地址总线将8位地址信息送入地址译码器，地址译码器的输出可以确定唯一的存储单元。
- 数据总线用来传送存储器到CPU或CPU到存储器的数据信息。

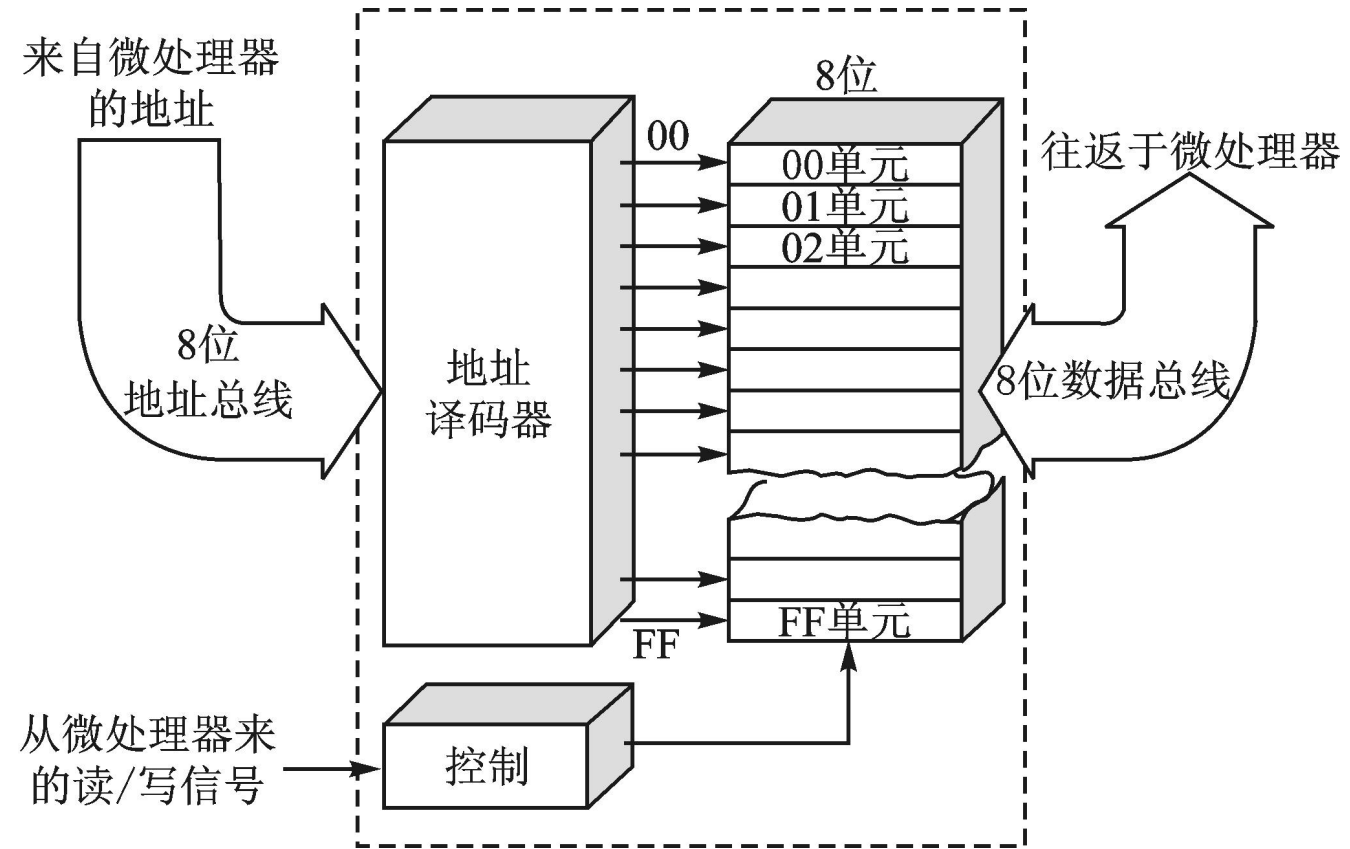


图1-4 随机存取存储器



2. I/O接口及外设

- 每个外设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接口及其对应的外设都有一个固定的地址，在CPU的控制下实现对外设的输入（读）和输出（写）操作。



§ 1.2 微机的工作过程

寻址：寻找指令的操作数

指令：操作码+操作数

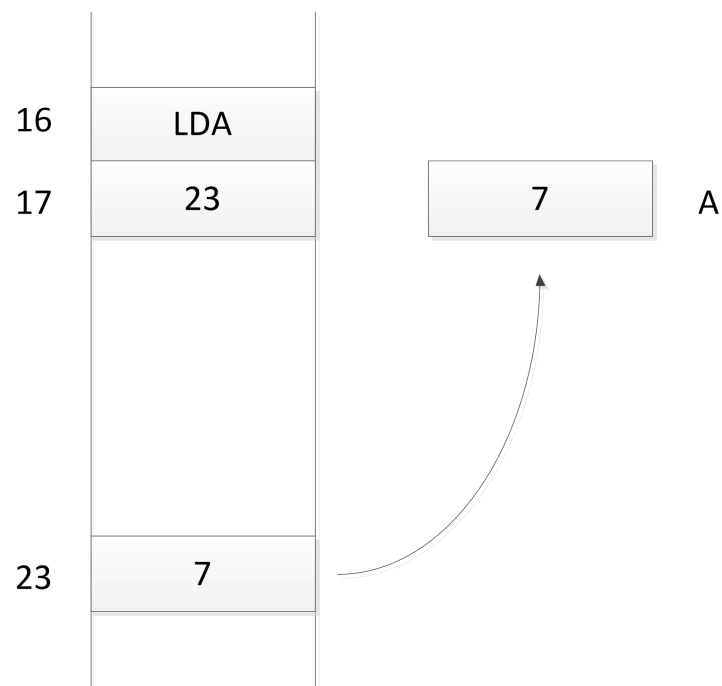


图1-6 执行 LDA 23 指令



§ 1.2 微机的工作过程

§ 1.2.1 执行一条指令的顺序

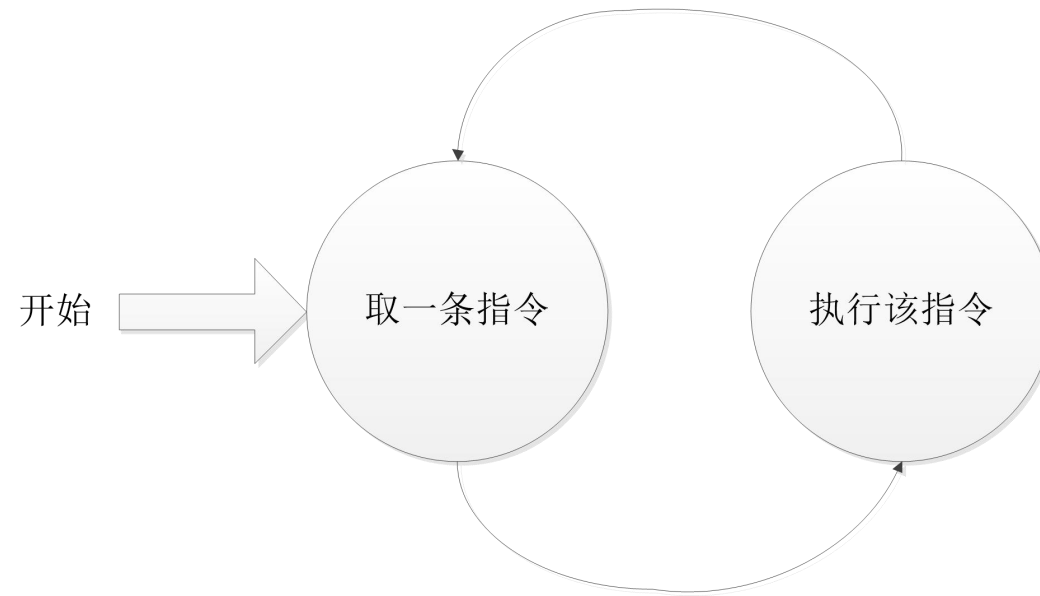


图1-7 取指令、执行指令序列



§ 1.2 微机的工作过程

§ 1.2.2 执行一条指令的过程

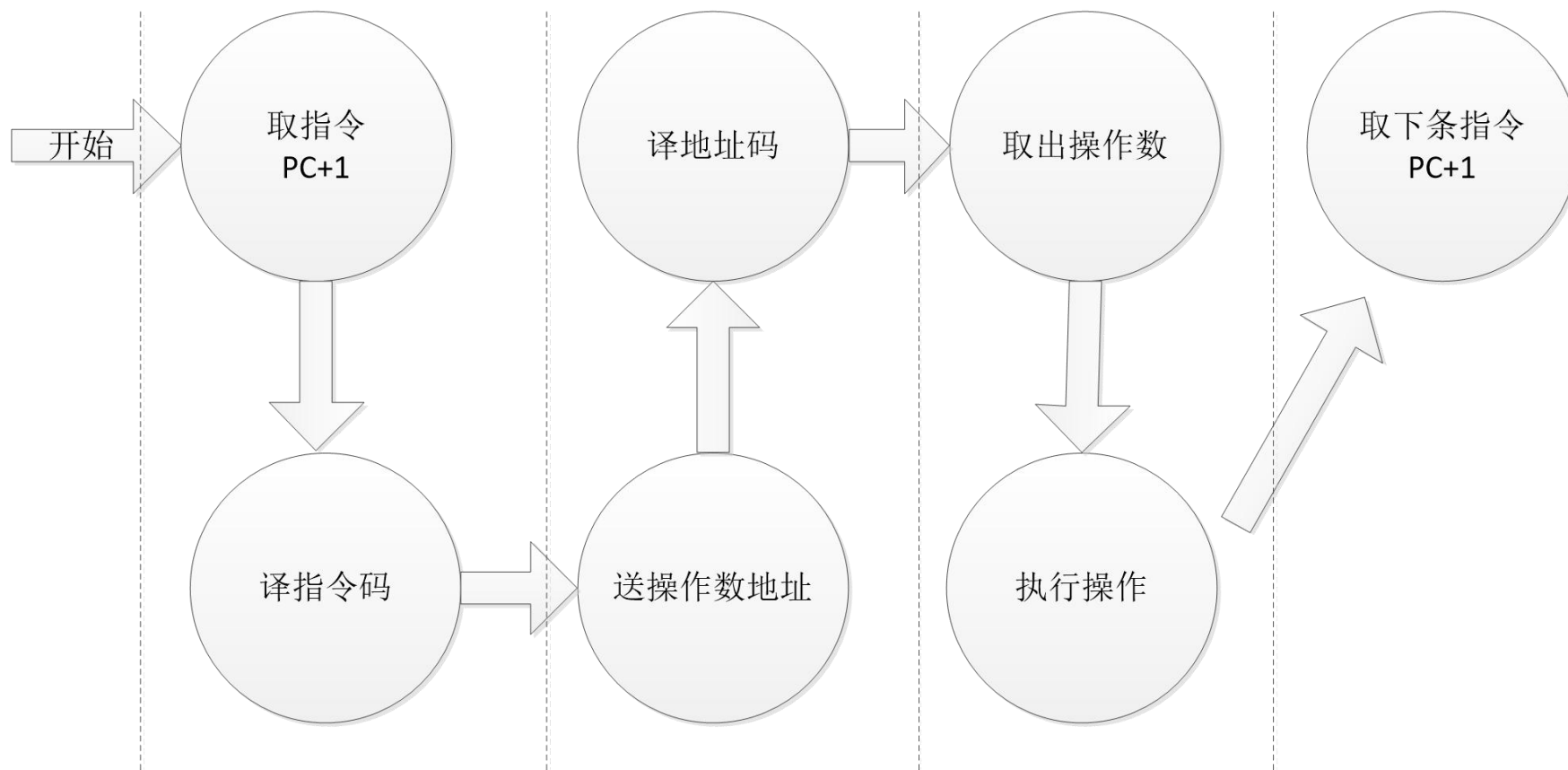


图1-8 直接访问内存指令周期



§ 1.2 微机的工作过程

§ 1.2.3 执行一条程序的过程

表1-2 “7+10”程序执行过程

地址	内容	助记符/内容
0001 0000 (16)	1001 0110	LDA
0001 0001 (17)	0001 0111	23
0001 0010 (18)	1001 1011	ADD
0001 0011 (19)	0001 1000	24
0001 0100 (20)	0001 0111	STA
0001 0101 (21)	0001 1001	25
0001 0101 (22)	0011 1110	HLT
0001 0111 (23)	0000 0111	7
0001 1000 (24)	0000 1010	10
0001 1001(25)	0000 0000	保存和



§ 1.2 微机的工作过程

§ 1.2.3 执行一条程序的过程

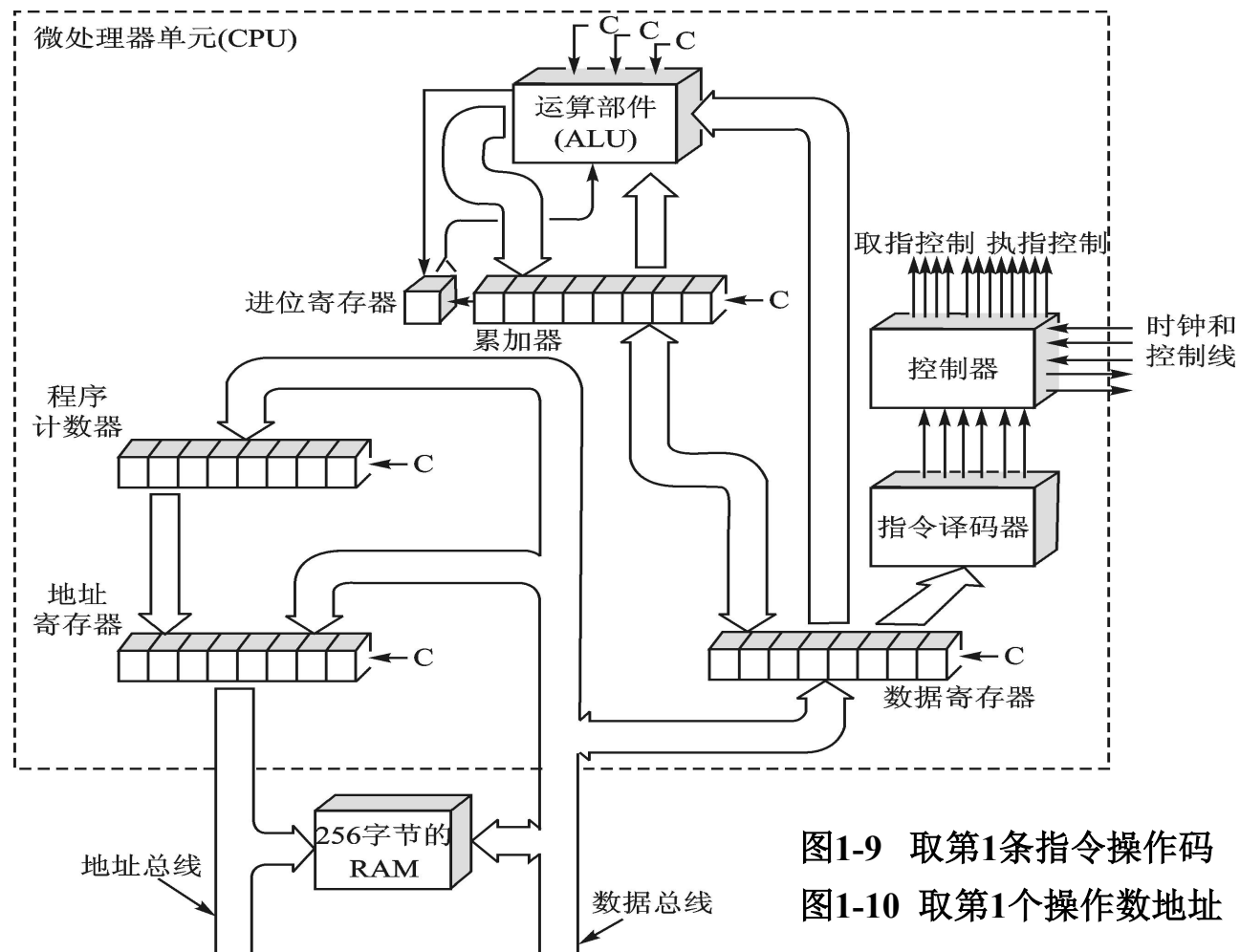


图1-9 取第1条指令操作码

图1-10 取第1个操作数地址

图1-11 取第1个操作数并装入A

§ 1.3 常用数制和编码



§ 1.2.1 数制及数制间转换

§ 1.2.2 计算机中常用编码



§ 1.3.1 数制及数制间转换

1. 数制——计数的进位制
2. 不同数制之间的转换



1. 数制——计数的进位制

1. 二进制：是“0”和“1”这样的数、逢2进位。按权展开时权的基数为2。用后缀字母“B”表示。

如：1001B = $1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9D$ （十进制数）

2. 十进制：是“0” — “9”之间的数、逢10进位。按权展开时权的基数为10。用后缀字母“D”表示。

如：1135D = $1 \times 10^3 + 1 \times 10^2 + 3 \times 10^1 + 5 \times 10^0$

3. 十六进制：是“0” — “9”，“A, B, C, D, E, F”之间的数、逢16进位。按权展开时权的基数为16。用后缀字母“H”表示。

如：1C5H = $1 \times 16^2 + 12 \times 16^1 + 5 \times 16^0 = 453D$



2. 不同数制之间的转换

1、二进制、十六进制转化成十进制：

将二、十六进制数按权展开相加即为相应的十进制数。

如： $1101 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13D$

如： $1FH = 1 \times 16^1 + 15 \times 16^0 = 31D$

2、十进制转换成二进制数：

将十进制数除2取余，商为0止余数倒置。

如： $11D = 1011B$

2		11	余数
2		5	1
2		2	1
2		1	0
		0	1

3、十进制转换成十六进制数：

将十进制数除16取余，商为0止余数倒置。

如： $100D = 64H$

16		100	余数
16		6	4
		0	6



2、不同数制之间的转换

4. 二进制转换成十六进制数：

将二进制数以小数点为界四位一分，不足补0，用一位十六进制数代替四位二进制数。

如：1 0011 1100 B=0001 0011 1100 B= 13C H

5. 十六进制转换成二进制数：

将十六进制数以小数点为界，用四位二进制数代替一位十六进制数。

如：D4E H=1101 0100 1110 B



不同数制之间的转换(带小数点)

【例1-2】 将十进制0.625转换成二进制数。

解：将十进制数乘2取整，小数部分为0止或满足精度，整数部分正的排列。

$$0.625 \times 2 = 1.25 \quad 1 \text{ (高位)}$$

$$0.25 \times 2 = 0.5 \quad 0$$

$$0.5 \times 2 = 1.00 \quad 1 \text{ (低位)}$$

$$\mathbf{0.625D = 0.101B}$$

【例1-4】 将十进制0.359375转换成十六进制数。

解：将十进制数乘16取整，小数部分为0止或满足精度，整数部分正的排列。

$$0.359375 \times 16 = 5.75 \quad 5 \text{ (高位)}$$

$$0.75 \times 16 = 12.0 \quad 12 \text{ (低位)}$$

$$\mathbf{0.359375D = 0.5CH}$$



不同数制之间的转换(带小数点)

【例】 将二进制0.01011011, 0.1011001101转换成十六进制数。

解:

(高位) (低位)

$$0.01011011\text{B} = 0.0101 \quad 1011 = 0.5\text{BH}$$

$$0.1011001101\text{B} = 0.1011 \quad 0011 \quad 01\textcolor{red}{00} = 0.\text{B34H}$$

【例】 将十六进制8F.41, 175.4E转换成二进制数。

解:

$$8\text{F}.41\text{H} = 1000 \quad \textcolor{red}{1111} . 0100 \quad \textcolor{red}{0001} \text{ B}$$

$$175.4\text{EH} = 0001 \quad 0111 \quad \textcolor{red}{0101} . 0100 \quad \textcolor{red}{1110} \text{ B}$$

§ 1.3.2 计算机中常用编码



1. **BCD (Binary Coded Decimal) 码——二进制码**
2. **ASCII (American Standard Code for Information Interchange) 码**



1. BCD（Binary Coded Decimal）码——二进制码

- BCD码是一种二进制形式的十进制码，也称二进制码。它用4位二进制数表示1位十进制数，最常用的是**8421BCD码**，见表1-2。
- **8421BCD码用0000H~1001H代表十进制数0~9，运算法则是逢十进一。8421BCD码每位的权分别是8，4，2，1，故得此名。**
- 例如，1 649的BCD码为0001 0110 0100 1001。



表 1-2 8421 BCD码表

十进制数	8421BCD码	二进制数	十进制数	8421BCD码	二进制数
0	0000	0000	8	1000	1000
1	0001	0001	9	1001	1001
2	0010	0010	10	0001 0000	1010
3	0011	0011	11	0001 0001	1011
4	0100	0100	12	0001 0010	1100
5	0101	0101	13	0001 0011	1101
6	0110	0110	14	0001 0100	1110
7	0111	0111	15	0001 0101	1111

2. ASCII (American Standard Code for Information Interchange) 码



- ASCII码是一种字符编码，是美国信息交换标准代码的简称，见表1-3。它由7位二进制数码构成，共有128个字符。
- ASCII码主要用于微机与外设通信。当微机与ASCII码制的键盘、打印机及CRT等连用时，均以ASCII码形式进行数据传输。
- 例如，当按微机的某一键时，键盘中的单片机便将所按的键码转换成ASCII码传入微机进行相应处理。



表 1-3 ASCII码字符表

高位		低位															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	DEL	BS	HT	LF	VT	FF	CR	SO	SI
1	001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	010	SP	!	“	#	\$	%	&	‘	()	*	+	,	-	。	、
3	011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	↑	←
6	110	、	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL



§ 1.4 数据在计算机中的表示

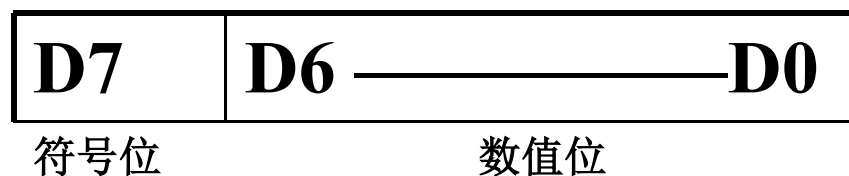
§ 1.4.1 有符号数

§ 1.4.2 无符号数



§ 1.4.1 有符号数

- 有符号的8位二进制数用最高位**D7**表示数的正或负，
- 0代表“+”，1代表“-”，
- **D7**称为符号位，**D6~D0**为数值位。



- 上述的8位带符号二进制数又有3种不同表达形式，即原码、反码和补码。
- 在计算机中，所有有符号数都是以**补码**形式存放的。



1. 原码

一个二进制数，用最高位表示数的符号，其后各位表示数值本身，这种表示方法称为原码。

原码的表示范围是-127~+127

例如：

$X = +1011010B$ $[X]_{\text{原}} = 01011010B$;

$X = -1011010B$ $[X]_{\text{原}} = 11011010B$



2. 反码

正数的反码与原码相同。

符号位一定为0，其余位为数值位。

负数的反码符号位为1，数值位将其原码的数值位逐位求反。

反码的表示范围是-127~+127

例如：

$X = -1011010B$ $[X]_{\text{原}} = 11011010B$

$[X]_{\text{反}} = 10100101B$



3. 补码

正数的补码与原码相同。

负数的补码符号位为1，数值位将其原码的数值位逐位求反后加1，即负数的反码加1。

补码的表示范围是-128~+127

例如：

$$X = -1011010B \quad [X]_{\text{反}} = 10100101B$$

$$[X]_{\text{补}} = 10100110B$$

通常计算机中的数用补码表示，用补码进行运算。一个很明显的优点是减法可以用补码的加法来运算。

这里还要特别提示“**溢出**”的概念。溢出与进位不同，溢出是指有符号数的运算结果超出了数-128~+127的表示范围，破坏了符号位。



4 机器数与真值

- 机器数：计算机中以二进制形式表示的数。
- 真值：机器数所代表的数值。

例如，机器数10001010B，它的真值为

138（无符号数）

-10（原码）

-117（反码）

-118（补码）

【例1-5】怎样根据真值求补码，或根据补码求真值？

答：只有两种求补码的方法：

- 1.是求负数的补码，用绝对值“取反加1”来求补码；
- 2.是求负数（补码）的真值，可先将该补码数用“取反加1”的方法得到其绝对值，再在绝对值前添加一负号。



§ 1.4.2 无符号数

- 无符号的8位二进制数没有符号位，从D7～D0皆为数值位，所以8位无符号二进制数的表示范围是0～+255。
- 8位二进制数码的不同表达含义见表1-4。



表 1-4 数的表示方法

8位二进制数	无符号数	原码	反码	补码
0000 0000	0	+0	+0	+0
0000 0001	1	+1	+1	+1
0000 0010	2	+2	+2	+2
0111 1100	124	+124	+124	+124
0111 1101	125	+125	+125	+125
0111 1110	126	+126	+126	+126
0111 1111	127	+127	+127	+127
1000 0000	128	-0	-127	-128
1000 0001	129	-1	-126	-127
1000 0010	130	-2	-125	-126
1111 1100	252	-124	-3	-4
1111 1101	253	-125	-2	-3
1111 1110	254	-126	-1	-2
1111 1111	255	-127	-0	-1



§ 1.5 89C51单片机

- 89系列单片机已经在片内增加4 KB或8 KB的Flash ROM, 而且整个89C51/89C52芯片比87C51便宜得多。所以现在已经没有人使用80C31或87C51开发产品了。
- 单片机是典型的嵌入式系统, 从体系结构到指令系统都是按照嵌入式应用特点专门设计的, 能最好地满足面对控制对象、应用系统的嵌入、现场的可靠运行以及非凡的控制品质要求。因此, 单片机是发展最快、品种最多、数量最大的嵌入式系统。



§ 1.5 89C51单片机

- 嵌入式系统与单片机已深入到国民经济众多技术领域，从天上到地下，从军事、工业到家庭日常生活。在人类进入信息时代的今天，难以想像，没有单片机的世界将会怎样！
- 本教程以ATMEL、PHILIPS和SST等公司的89系列单片机中的89C51/P89C51/SST89E554（以下简称为89C51）为典型机，讲述单片机的硬件结构、原理、接口技术、编程及其应用技术。舍弃80C31扩展EPROM的传统模式，而依据目标任务选择所需不同档次（片内不同存储器容量）的89系列单片机。



§ 1.6 思考题与习题

1. 什么是微处理器、CPU、微机和单片机？
2. 单片机有哪些特点？
3. 微型计算机怎样执行一个程序？
4. 将下列各二进制数转换为十进制数及十六进制数。
① 11010B ② 110100B ③ 10101011B ④ 11111B
5. 将下列各数转换为十六进制数及ASCII码。
① 129D ② 253D ③ 01000011BCD ④ 00101001BCD
6. 将下列十六进制数转换成二进制数和十进制数。
① 5AH ② 0AE7.D2H ③ 12BEH ④ 0A85.6EH
7. 将下列十进制数转换成8421BCD码。
① 22 ② 986.71 ③ 1234 ④ 678.95



§ 1.6 思考题与习题

8. 什么叫原码、反码及补码？

9. 已知原码如下，写出其补码和反码（其最高位为符号位）。

① $[X]_{\text{原}} = 01011001$

② $[X]_{\text{原}} = 00111110$

③ $[X]_{\text{原}} = 11011011$

④ $[X]_{\text{原}} = 11111100$

10. 当微机把下列数看成无符号数时，它们相应的十进制数为多少？若把它们看成是补码，最高位为符号位，那么相应的十进制数是多少？

① 10001110

② 10110000

③ 00010001

④ 01110101



参考资料

- 1.李朝青. 单片机原理及接口技术(简明修订版). 北京: 北京航空航天大学出版社, 1999
- 2.李朝青. 单片机学习辅导测验及解答讲义. 北京: 北京航空航天大学出版社, 2003
- 3.李朝青. 单片机&DSP外围数字IC技术手册. 北京: 北京航空航天大学出版社, 2002
- 4.何立民. 单片机高级教程. 北京: 北京航空航天大学出版社, 1999
- 5.何立民. I2C总线应用系统设计. 北京: 北京航空航天大学出版社, 2004
- 6.张俊谟. 单片机中级教程. 北京: 北京航空航天大学出版社, 1999
- 7.张迎新, 等. 单片机初级教程. 北京: 北京航空航天大学出版社, 1999
- 8.余永权. Flash单片机原理及应用. 北京: 电子工业出版社, 1997
- 9.潘琢金, 等. C8051F $\times\times\times$ 高速SOC单片机原理及应用. 北京: 北京航空航天大学出版社, 2002
- 10.李刚. AD μ C8 $\times\times$ 系列单片机原理与应用技术. 北京: 北京航空航天大学出版社, 2002
- 11.李群芳, 等. 单片微型计算机与接口技术. 北京: 电子工业出版社, 2001
- 12.朱定华, 等. 单片微机原理与应用. 北京: 清华大学出版社, 北京: 北方交通大学出版社, 2003
- 13.李维祥. 单片机原理与应用. 天津: 天津大学出版社, 2001
- 14.肖洪兵, 等. 跟我学用单片机. 北京: 北京航空航天大学出版社, 2002
- 15.钱逸秋. 单片机原理与应用. 北京: 电子工业出版社, 2002



见作业集



第一章结束