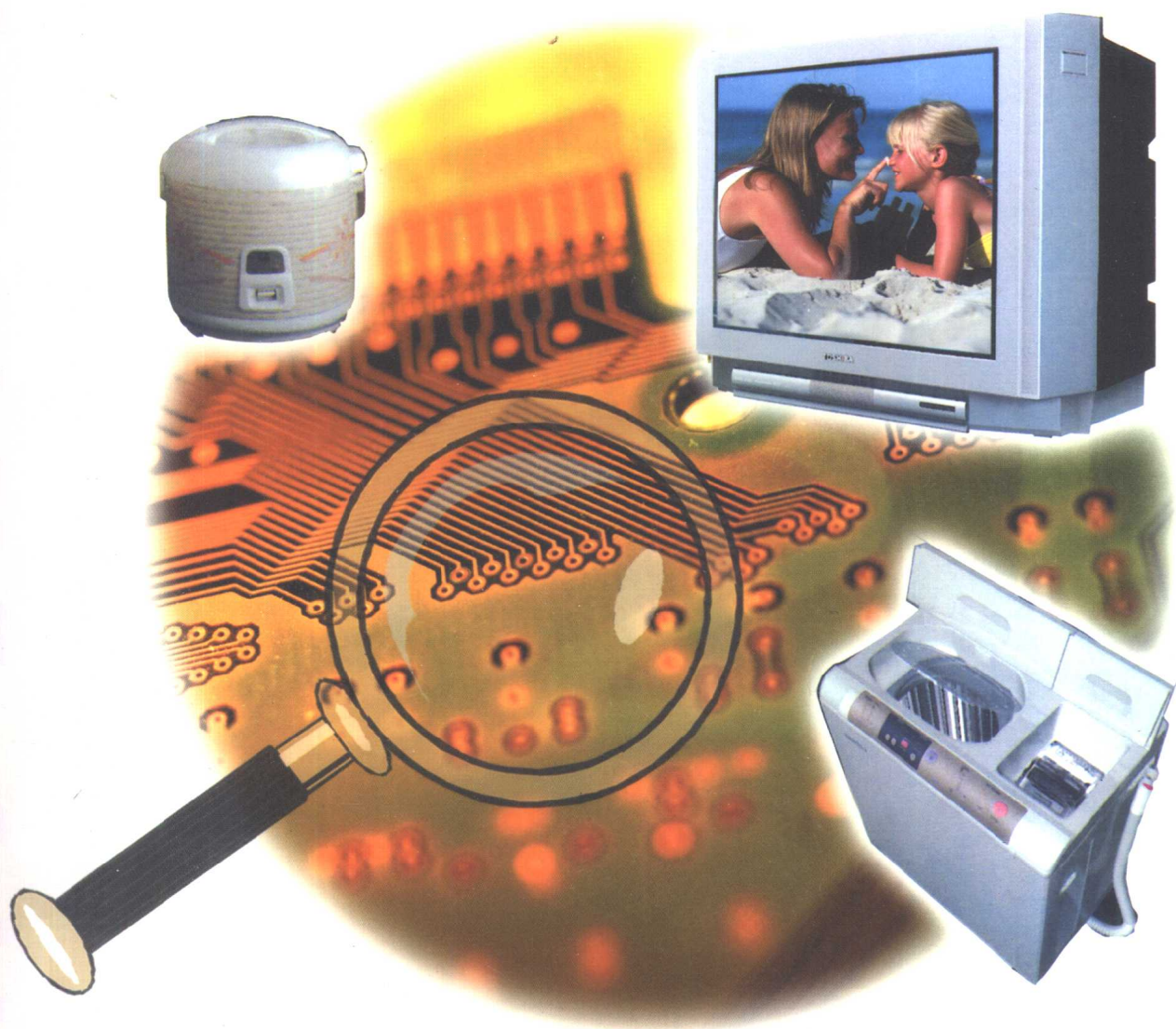


怎样看 家电微电脑 控制电路

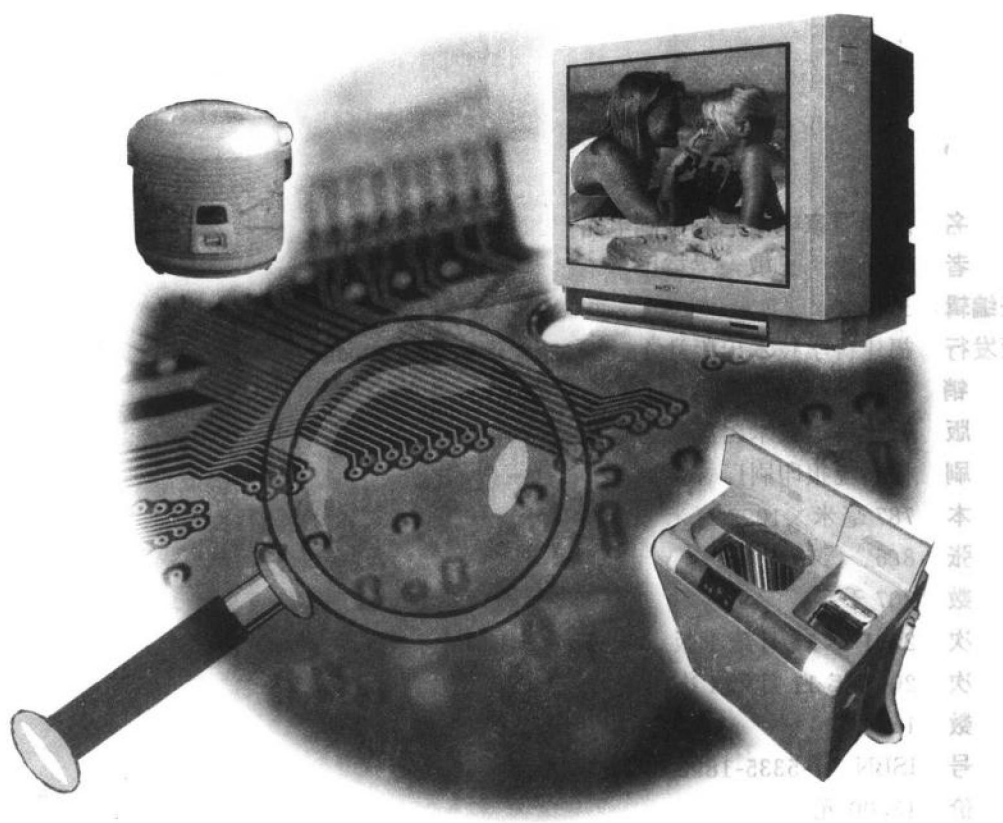
福建科学技术出版社



怎样看 家电微电脑 控制电路

王乃成 王敏 梁裕民

福建科学技术出版社



责任编辑：陈国良 封面设计：陈国良

图书在版编目 (CIP) 数据

怎样看家电微电脑控制电路/王乃成, 王敏, 梁裕民
编著. —福州: 福建科学技术出版社, 2001. 11
ISBN 7-5335-1892-6

I. 怎… II. ①王…②王…③梁… III. 日用电
气器具—微处理器—控制电路—基本知识 IV. TM925

中国版本图书馆 CIP 数据核字 (2001) 第 062319 号

书 名 怎样看家电微电脑控制电路
作 者 王乃成 王敏 梁裕民
责任编辑 王健文
出版发行 福建科学技术出版社 (福州市东水路 76 号, 邮编 350001)
经 销 各地新华书店
排 版 福建科学技术出版社照排室
印 刷 福州晚报印刷厂
开 本 787 毫米×1092 毫米 1/16
印 张 8.5
字 数 207 千字
版 次 2001 年 11 月第 1 版
印 次 2001 年 11 月第 1 次印刷
印 数 1—4 000
书 号 ISBN 7-5335-1892-6/TP·73
定 价 13.00 元



书中如有印装质量问题, 可直接向本社调换

前 言

伴随着科学技术的发展，微电脑型的家用电器已成为当前家电产品的主流。这类家电尽管品种繁多，用途各异，消耗功率大小不等，精密程度不同，但其微电脑控制电路却存在相同之处。微电脑控制电路的正常运行，必须具备硬件和软件两个部分。软件部分，在产品出厂时，根据家电的特性、运行特点和要求，厂家已将运行程序固化在存储器中。使用时，只需按动操作键，微电脑即可完成相应的功能，所以本书对软件部分不作介绍。硬件部分是完成各种功能的电路保障，它是以微电脑芯片为核心，配置适当的外围电路构成。各种外围电路在微电脑的统一指挥下，有序地完成应有的功能。

本书在简要介绍微电脑的基本结构和工作原理的基础上，选编了部分具有代表性的家用电器微电脑控制电路，通过对这些电路的分析，希望能帮助读者提高识图、读图的能力，进而掌握电路的工作原理、简单故障的分析和检修方法。

由于编者水平所限，经验不足，书中难免存在缺点和错误，敬请读者给予批评指正。

编著者

2001.5

目 录

第 1 章 微电脑基础知识	(1)
1.1 微电脑的基本结构	(1)
1.2 微电脑的定时系统	(7)
1.3 微电脑的指令系统	(9)
1.4 微电脑工作过程概述	(10)
1.5 微电脑与家用电器的接口电路	(12)
第 2 章 红外线遥控器微电脑控制电路	(21)
2.1 红外线遥控器的组成	(21)
2.2 红外线遥控器微电脑控制电路	(21)
第 3 章 房间空调器微电脑控制电路	(28)
3.1 分体式空调器的控制系统	(28)
3.2 微电脑控制系统所能实现的功能	(29)
3.3 采用 C75P036 组成的微电脑控制电路	(29)
3.4 采用 8031 组成的微电脑控制电路	(33)
3.5 空调器的变频器控制	(36)
3.6 海尔 KFR 50LW/(BP)变频空调器微电脑控制电路	(37)
3.7 空调器微电脑控制电路一般故障的检修	(42)
第 4 章 电冰箱微电脑控制电路	(44)
4.1 采用 8031 组成的微电脑控制电路	(44)
4.2 采用 MC6805R2P 组成的微电脑控制电路	(48)
4.3 双鹿三门电冰箱微电脑控制电路	(52)
4.4 电冰箱模糊控制系统	(53)
4.5 电冰箱微电脑控制电路一般故障的检修	(58)
第 5 章 全自动洗衣机微电脑控制电路	(60)
5.1 洗衣机微电脑控制基本原理	(60)
5.2 采用 DJ2001 组成的微电脑控制电路	(62)
5.3 采用 MCS8049 组成的微电脑控制电路	(65)
5.4 LY362 型全自动洗衣机微电脑控制电路	(70)
5.5 全自动洗衣机的模糊技术控制	(75)
5.6 全自动洗衣机微电脑控制电路一般故障的检修	(82)

第 6 章 微波炉微电脑控制电路	(85)
6.1 概述.....	(85)
6.2 采用 TMS1000 组成的微电脑控制电路	(86)
6.3 WP600 型微电脑控制电路	(88)
6.4 采用 MC6805R ₃ 组成的微电脑控制电路	(92)
6.5 微波炉微电脑控制电路一般故障的检修.....	(96)
第 7 章 电饭锅微电脑控制电路	(98)
7.1 概述.....	(98)
7.2 采用 MH8841 组成的微电脑控制电路	(98)
7.3 采用 8031 组成的微电脑控制电路.....	(101)
7.4 电饭锅的模糊技术控制	(103)
第 8 章 电风扇微电脑控制电路	(105)
8.1 概述	(105)
8.2 采用 MH8822 组成的遥控电风扇微电脑控制电路	(105)
8.3 采用 PC—2122 组成的微电脑控制电路	(109)
8.4 采用 COP411L 组成的微电脑控制电路	(111)
第 9 章 彩电微电脑控制电路.....	(113)
9.1 遥控彩电概述	(113)
9.2 TC—M25C 彩电控制电路	(114)
9.3 遥控彩电控制系统一般故障的检修	(128)

第 1 章 微电脑基础知识

随着大规模集成电路技术的高度发展，人们已能在一块半导体芯片上集成一台计算机的全部部件，如中央处理器、只读存储器、随机存储器、定时/计数器、输入/输出及 A/D 转换器等。就其组成而言，一块芯片就是一台计算机，故称作单片机，或称作微电脑。微电脑由于体积小，可以装在设备的内部，实现智能化控制，因此在家用电器和其他许多行业中，都得到了广泛应用。

在家用电器微电脑控制电路中，微电脑是电路的指挥中心。我们只有对微电脑的基本结构、工作原理以及常用接口电路有所了解，才能看懂控制电路，掌握整机的工作原理以及故障的排除方法。

1.1 微电脑的基本结构

微电脑的内部结构非常复杂，对一个初学者来说，如果一开始就去了解一个完整的微型计算机，会有很多困难。为此，我们构造了一个微电脑的简化模型，以帮助读者了解微电脑的基本结构和工作原理。

微电脑的基本结构如图 1.1 所示。由图可见，微电脑主要由运算器、控制器、存储器、I/O 接口以及总线等构成。

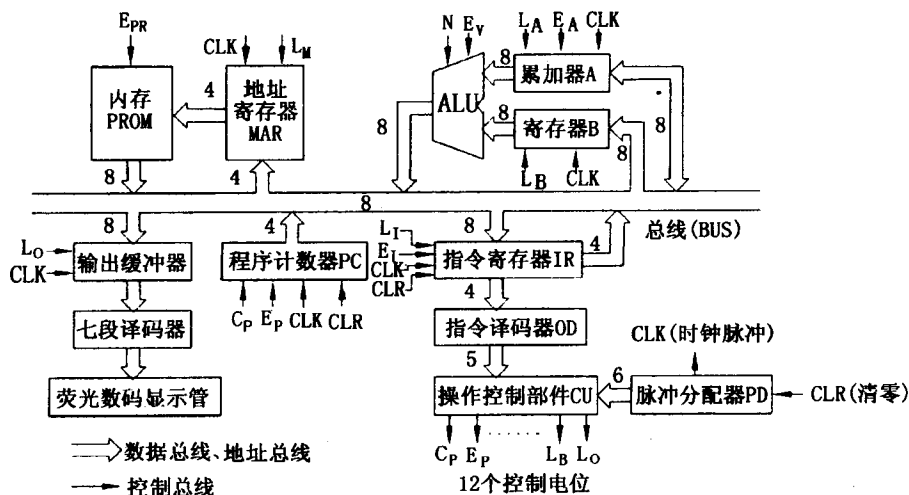


图 1.1 微电脑的基本结构

运算器由算术逻辑部件 ALU、累加器 A 以及寄存器 B 组成。它是用来完成算术运算和逻辑运算的部件。也就是说，它是计算机数据的变换部分。

控制器由程序计数器 PC、指令寄存器 IR、指令译码器 OD、操作控制部件 CU、脉冲分配器 PD 等组成。控制器根据指令的要求，对内部和外部各部件发出相应的控制信号，使它们

协调工作，从而完成对整个微电脑系统的控制。

运算器、控制器和寄存器构成了微电脑的核心，统称中央处理器，简称 CPU。

存储器是用来存放原始数据、程序及运算结果的部件。它由许多存储单元组成，每一存储单元用以存放一个数据代码。为了区分不同的存储单元，全部存储单元按一定顺序编号，这个编号就称作单元的地址。

计算机内部 CPU 与外部设备必须通过输入/输出 (I/O) 接口电路建立联系。因为外部设备输入到计算机的信号，计算机未必能识别，而计算机输出的信号未必能对外部设备进行控制，故两者之间必须通过接口电路对信号进行变换才能达到互相识别的目的。

总线 BUS 是用来传送信息的一组公用线。总线分单向总线和双向总线。双向总线既可以发送数据，也可以接收数据，即可以朝两个方向传送数据。单向总线只能朝一个方向传送数据。微电脑一般均采用总线结构，即控制器、运算器、存储器及接口电路都连到总线上，通过总线来交换信息。根据传递信息的不同，总线可分为地址总线、数据总线和控制总线。

微电脑在运算时，为了提高速度，总是一次运算由几位构成的二进制数据，这时运算器的位数应与处理数据的位数相等。运算器的位数多少就称为微电脑的字长。一般最小为 4 位，通常为 8 位、16 位，也有 32 位。字长也决定了内部总线的条数，8 位字长微电脑的总线有 8 根。

一、存储器

存储器是用来存放二进制信息的部件。一个存储器包括很多存储单元，每个存储单元都有相应的地址。存储器的示意图如图 1.2 所示。存储器分为读/写存储器 RAM 和只读存储器 ROM。

(一) 读/写存储器 RAM

读/写存储器也称为随机存储器，其结构方框图如图 1.3 所示。当地址线为 P 条时，则可有 $N=2^P$ 个字。若每个字具有 M 位，存储单元数为 $N \times M$ ，双向数据线数等于位数 M。

0000	00001111
0001	00111110
0010	1110XXXX
0011	1111XXXX
⋮	⋮
1110	00100010
1111	01000010

图 1.2 存储器示意图

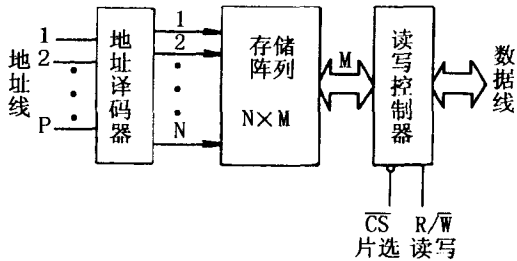


图 1.3 RAM 结构方框图

读/写存储器既可以写入信息，也可以读出信息，每个存储单元由 MOS 管双稳态触发器组成，可以存储一位二进制数（“0”或“1”）。当用地址线通过地址译码器选中相对应的单元时，在片选信号 $\overline{CS}$ 和读写信号 $R/\overline{W}$ 控制下，从双向数据线上可以将数据写入 ($\overline{CS}=0, R/\overline{W}=0$) 该单元，或从该单元读出 ($\overline{CS}=0, R/\overline{W}=1$) 数据。

读/写存储器常用于存放正在运行的程序及处理的数据。其工作特点是，当向一个单元写入新的数据时，则该单元原有数据将被充掉，新的数据被保存下来。当向一个单元读出数据时，单元中的数据不会消失。RAM 存储器的缺点是一旦断电，所存放的数据就消失了。显然，

RAM 不能用来存放家用电器的程序。但它又是微电脑中不可缺少的, RAM 容量越大, 微电脑的功能越强。

(二) 只读存储器 ROM

只读存储器中的数据只能读出而不能重新改写, 但断电后数据不会消失, 可实现永久性存储。家用电器中的工作程序、字符数据就存放在 ROM 中, 程序和数据在微电脑出厂时由制造厂家固化在 ROM 中。

二极管矩阵存储器如图 1.4 所示。ROM 由存储阵列、地址译码器和输出缓冲器三部分组成, 图中的存储阵列有 16 个字线 ($R_0 \sim R_F$) 和 8 个位线 ($D_0 \sim D_7$), 能存储 16 个字, 每个字有 8 位。字线和位线的相交处称为存储单元, 该处若有二极管, 存储内容为 “1”, 否则为 “0”。由图可见, 该存储阵列的容量为 $16 \times 8 = 128$ 个存储单元。

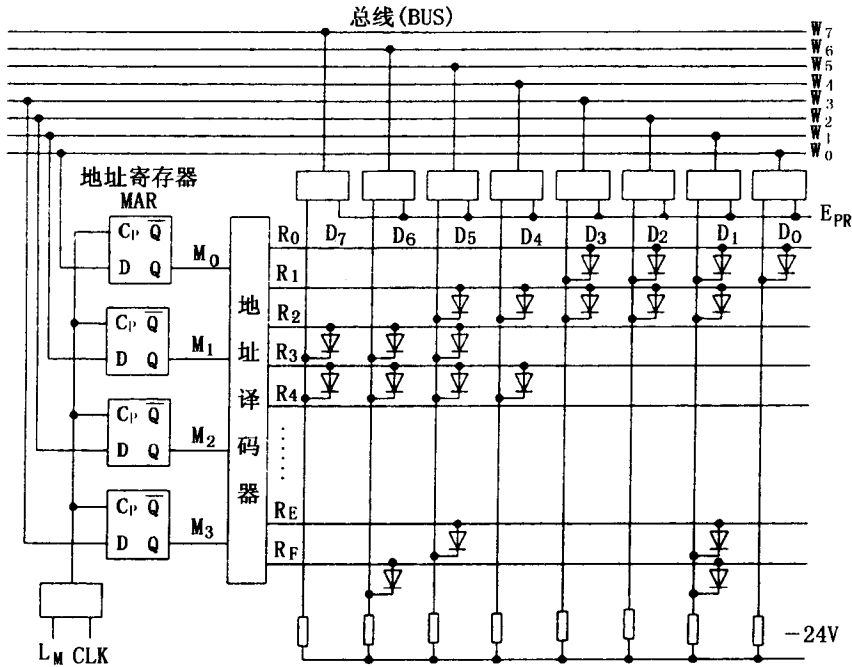


图 1.4 只读存储器 ROM 原理

工作时, 来自总线 ($W_0 \sim W_7$) 低 4 位的地址码 ($W_0 \sim W_3$) 首先送入地址寄存器 MAR 的输入端。地址寄存器由 4 个 D 触发器组成。当输入控制端 L_M 为高电平时, 在时钟脉冲 CLK 上升沿的作用下, D 触发器打开, 地址码被地址寄存器接收。地址寄存器输出端的地址码经地址译码器译码, 可产生 1 个地址译码信号, 让 $R_0 \sim R_F$ 中某一个信号为 “1”, 其余为 “0” 时, 表示选中了信号为 “1” 所对应的字。例如, $M_3 M_2 M_1 M_0 = 0001$, 经译码, $R_1 = 1$, $R_0, R_2 \sim R_F$ 均为 “0”, 与字线 R_1 相连的 5 个二极管导通, 相应的位线 (D_5, D_4, D_3, D_2, D_1) 为高电平 “1”, 没有二极管的位线 (D_7, D_6, D_0) 仍为低电平 “0”, 所选中字的内容为 00111110。若给出输出控制信号 E_{PR} 为高电平, 则输出控制门被打开, 选中字的内容被送到总线, 即 $W_7 W_6 W_5 W_4 W_3 W_2 W_1 W_0 = 00111110$, 以便被其他寄存器所接收。

只读存储器的种类很多, 按使用的器件类型分, 有二极管 ROM、三极管 ROM 和 MOS 管 ROM; 按数据的写入方式又可分为固定 ROM、可编程 ROM (简称 PROM) 和可擦可编程

ROM (简称 EPROM) 三种。固定 ROM 的存储内容在制造时已完全固定下来, 使用时无法再更改。PROM 的内容则由用户自己根据需要写入, 但只能写一次, 一经写入也不能再修改。EPROM 具有较大的使用灵活性, 它的存储内容不仅可以按照用户的需要写入, 而且还能擦去重写。

二、运算器

运算器是完成各种算术运算和逻辑运算的部件。这里我们仅以一个简单的运算器为例说明运算器的工作原理, 虽然它只能进行二进制加法、减法运算, 但工作过程是典型的。

运算器的逻辑原理图如图 1.5 所示, 它由全加器 ALU、累加器 A 和寄存器 B 组成。累加器 A 用于寄存并向加法器提供参加运算的操作数之一, 同时也用于寄存运算结果, 并在需要时将此结果经总线送往输出缓冲器。当输入控制电平 L_A 为高电平, 时钟脉冲 CLK 为上升沿时, 总线上的操作数或运算结果被送至累加器 A 中。累加器 A 中的操作数直接加到 8 位全加器 ALU 的一组输入端上, 也可以在输出控制 E_A 为高电平时, 经输出控制门送至总线。

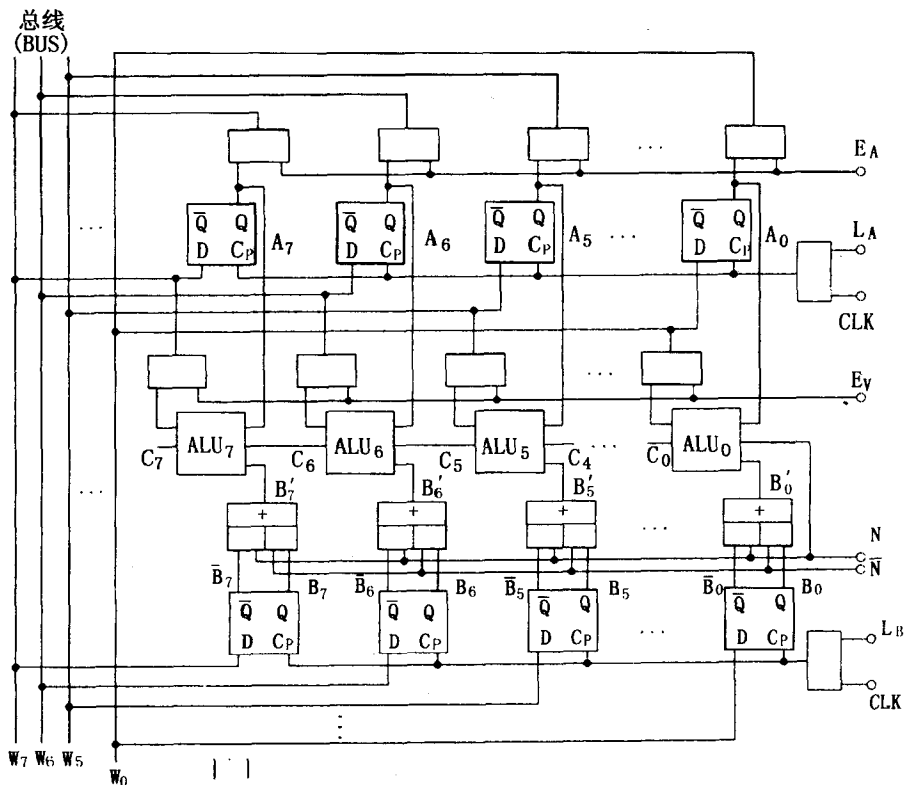


图 1.5 运算器逻辑原理图

寄存器 B 用于向全加器提供另一个操作数。当输入控制 L_B 为高电平, CLK 上升沿到来的时刻, 总线上的加数或减数被送入 B 寄存器。B 寄存器的输出端接有加减控制门, 加减控制门的输入与输出的逻辑关系为:

$$B_i' = \bar{N}B_i + N\bar{B}_i$$

其中 B_i 为寄存器 B 的第 i 位的 Q 端输出, $\bar{B}_i$ 为 $\bar{Q}$ 端输出, B_i' 为送往全加器的第 i 位操作数。当 $N=0$, 则 $B_i' = B_i$, 表示做加法运算; 当 $N=1$, 则 $B_i' = \bar{B}_i$, 表示做减法运算。

全加器 ALU 接收来自累加器 A 和寄存器 B 的操作数, 并完成加法 (或减法) 运算。当控制电位 E_V 为高电平时, 将运算结果送往总线, 以便被累加器 A 接收。

三、总线

总线是微电脑内部信息传送的公共线。由图 1.4 和图 1.5 可以看出, 总线上信号与各部件输出信号之间关系为:

$$\begin{aligned} W_7 &= E_A \cdot A_7 + E_V \cdot ALU_7 + E_{PR} \cdot D_7 + \dots \\ W_6 &= E_A \cdot A_6 + E_V \cdot ALU_6 + E_{PR} \cdot D_6 + \dots \\ &\vdots \\ W_0 &= E_A \cdot A_0 + E_V \cdot ALU_0 + E_{PR} \cdot D_0 + \dots \end{aligned}$$

式中 E_{PR} 、 E_R 、 E_V 分别为存储器 ROM、累加器 A、全加器 ALU 的使能端控制电平。

若 E_A 为高电平, 其余控制端为低电平, 则得 $W_7 = A_7$, $W_6 = A_6$, $\dots$, $W_0 = A_0$ 。即总线上的信号完全反映了累加器 A 的信号状态。

如果 E_A 和 E_{PR} 同时为高电平, 则得: $W_7 = A_7 + D_7$, $W_6 = A_6 + D_6$, $\dots$, $W_0 = A_0 + D_0$ 。总线上的信号既不反映累加器 A 的输出信号, 也不反映存储器的输出信号, 而是二者的“或”。显然这是不允许出现的。这就是说, 在各部件向总线传输信息时, 只能有一个控制端为高电平。

各部件从总线接收信息时, 只要相应的输入控制电平为高电平, 即可在时钟脉冲 CLK 作用下接收来自总线的信号。如 L_A 为高电平, 当 CLK 上升沿到来时, $A_7 = W_7$, $A_6 = W_6$, $\dots$, $A_0 = W_0$, 即总线上的信息被传送到累加器中。

由上述可见, 采用总线可以使信息很方便地传送, 既可以单向传送, 又可以双向传送。

如果两个操作数存放在 ROM 中, 则执行一个加法运算要经过下列步骤:

1. 控制器发出 E_{PR} 、 L_A 两个控制高电平, 被加数从 ROM 中取出, 送到累加器 A 中, 累加器 A 直接把数送到全加器 ALU 的一组输入端上。
2. 控制器发出 E_{PR} 和 L_B 两个控制高电平, 加数从 ROM 中取出, 送到寄存器 B 中, 寄存器 B 将加数直接送到加减控制门, 由于 $N=0$, 全加器 ALU 的另一组输入端得到来自 B 寄存器的加数, 并在全加器中将加数与被加数相加, 其和立即出现在全加器 ALU 的输出端上。
3. 控制器发出 E_V 、 L_V 两个控制高电平, 全加器输出端的和数经总线送到累加器 A 中, 累加器 A 中原有的内容被充掉, 而寄存器 B 中的加数却保持不变。

四、控制器

微电脑完成某一功能控制的过程, 就是执行相应程序的过程。具体地讲, 就是取指令、执行指令、再取下一条指令……周而复始重复运行。完成这一过程的任务由控制器承担。下面根据图 1.1 介绍控制器各部件的结构及工作原理。

(一) 程序计数器

结构简单的程序计数器如图 1.6 所示。它是一个 4 位二进制计数器, 最多可产生 16 个地址码。也就是说, 使用该程序计数器的微电脑最多只能有 16 条指令, 显然, 在实际的微电脑中是不够的。程序计数器的功能是用来指出现行指令在内存中的地址, 该指令地址在 E_P 和 L_M (见图 1.4) 控制电平控制下, 送到内存地址寄存器 MAR 中, 经 ROM 地址译码器译码后, 选中现行指

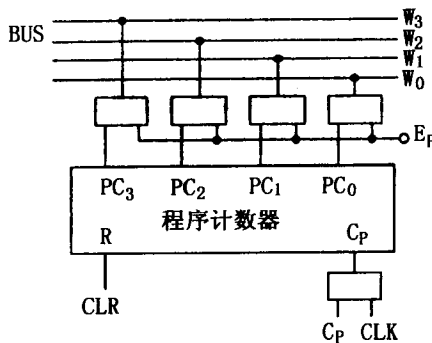


图 1.6 程序计数器

令的内存地址。与此同时，在现行指令地址被送到 MAR 后，在 C_p 控制端为高电平、时钟脉冲 CLK 上升沿到来时，程序计数器 PC 自动加 1，以形成下条指令地址。当清零信号 CLR 为低电平时，程序计数器清零。

（二）指令寄存器

指令寄存器 IR 的逻辑图如图 1.7 所示。当 L_i 为高电平、时钟脉冲上升沿到来时，指令寄存器通过总线接收来自 ROM 的指令码并寄存起来。该机器指令的高 4 位为操作码部分，低 4 位为操作数地址码部分。当清零信号 CLR 为低电平时，指令寄存器 IR 清零。工作时，高 4 位操作码直接送操作码译码器，低 4 位操作数地址码在 E_i 为高电平时送回到总线。当 L_M 为高电平、时钟脉冲 CLK 上升沿到来时，低 4 位操作数地址码又被内存地址寄存器 MAR 所接收，以便经译码取出相应操作数。

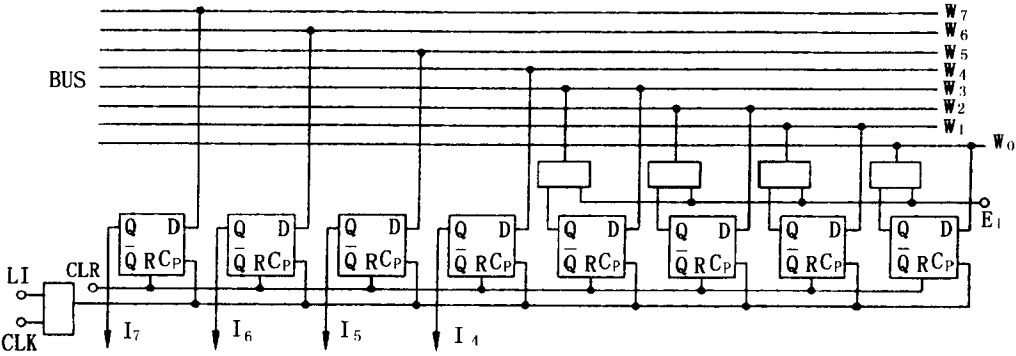


图 1.7 指令寄存器

（三）指令译码器

指令译码器来自指令寄存器 IR 高 4 位操作码进行译码，产生对应于本指令的译码信号。此译码信号被送往操作控制部件，和时序电位相配合产生执行本条指令相应的控制电平序列。

指令译码器的电路如图 1.8 所示。它是一个 5 中取 1 的组合逻辑电路。它直接接收来自指令寄存器 IR 高 4 位操作码。例如，当操作码为 $I_7I_6I_5I_4=0000$ ，则对应输出端为 LD 的“与”门的 4 个输入端为 $\bar{I}_7\bar{I}_6\bar{I}_5\bar{I}_4=1111$ ，所以输出端 LD 为高电平，而其他输出端均为低电平。又如，当 $I_7I_6I_5I_4=0011$ 时，只有 ADD 译码输出为高电平，其他输出端均为低电平。

（四）操作控制部件

由上述可知，微电脑的工作过程就是执行程序的过程，即按照顺序，从存储器中取出当前的指令，并送到指令寄存器 IR，然后对所取出的指令进行译码分析，根据译码结果，操作控制部件 CU 发出完成操作所需要的一系列控制电平。所以，操作控制部件是由时钟脉冲发生器与微操作控制电路组成的组合逻辑电路。它将指令译码器送来的指令信号与时钟发生器产生的时序信号经微操作控制电路组合，产生各功能部件的控制信号。这些控制信号按一定的时间顺序先后发出，以指

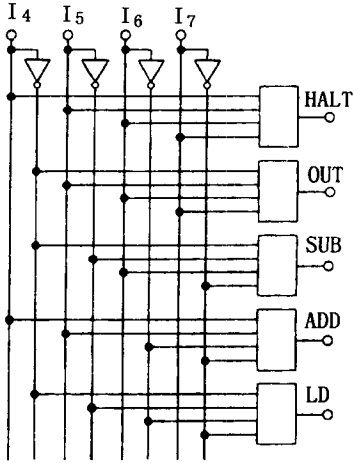


图 1.8 指令译码器

挥各功能部件协调的工作，完成指令所规定的操作。

五、输入/输出接口 (I/O 口)

输入/输出接口是微型计算机的重要组成部分。通过 I/O 口，微电脑能与外部设备连接起来，实现信号的输入和输出。输入/输出接口可根据接口的作用以及传递信息的方式等进行分类。

(一) 按数据传递的方式可分为并行接口和串行接口

1. 并行接口。能实现将信息单元的字符同时在一组通路上传输的接口，叫并行接口。其通道数目等于同时传送的数据字符的位数。例如，8 位微电脑的数据，通过 8 根数据线同时从 CPU 经并行接口传送到被控制对象。

2. 串行接口。能将信息流的字符在单一通道上按序传送，完成数据逐位传送功能的接口，叫串行接口。采用串行接口可大大减少传输线的数目。

(二) 按输入、输出信息的性质，可分为数字接口和模拟接口

1. 数字接口。如果通过某接口传递的信息是二进制代码形式的数字量或控制命令，则称该接口为数字接口，如键盘接口等。

2. 模拟接口。完成把某一连续变化的模拟量转换为相应的数字量或把某一数字量转换成相应模拟量的接口，称为模拟接口。

一个模拟量要输入微电脑，首先要进行模/数 (A/D) 转换，把它转换成数字量后，才能被微电脑所接收，然后才可能对该信号进行数据处理。一般在微电脑芯片的内部设置 A/D 转换器来完成。若微电脑连接的是模拟量控制的外设，则必须先把微电脑输出的数字量，经数/模 (D/A) 转换器，把该数字量转化为相应的电压值，实现对外设的控制。一般微电脑芯片中也设置有 D/A 转换电路。

(三) 按接口的作用可分为内务操作接口、用户通信接口、数据采集接口和控制接口

1. 内务操作接口。内务操作接口是指微电脑的处理和控制功能处于最基本水平上所必需的接口，如总线驱动器、数据锁存器、三态缓冲器等。

2. 用户通信接口。用户通信接口是指微电脑把数据传送给用户指定的外设或接收用户输入信息的那些接口，如键盘接口、显示器接口。

3. 数据采集接口。当微电脑用于过程控制时，必须监视控制对象的变化，如现场的温度、压力等参数。通常把从现场获得有关参数信息传送给微电脑的接口称为数据采集接口。

4. 控制接口。控制接口是指把来自微电脑的控制数据或命令，经过适当处理后，传递给执行部件的接口。

1.2 微电脑的定时系统

由前面的分析我们已经知道，微电脑中的各功能部件都接在内部总线上，因此，任何两个部件间的联系，都要通过总线来进行。然而，在同一时刻，总线上只能出现唯一的一组信息，否则将会发生混乱，这也就决定了微电脑必须分时有节拍地工作。这些节拍都需要精确的定时。微电脑的定时系统是以时钟脉冲为基准而形成的。

微电脑为了连续地执行程序，必须有时钟脉冲，它是微电脑的基本定时信号，为微电脑的所有操作提供基准。时钟脉冲可以内部供给，也可以外部供给。通常微电脑芯片内部带有时钟振荡电路，仅需在片外提供时钟参考频率即可。参考频率一般由晶体振荡器来产生，以便获得可靠稳定的频率，其接线图如图 1.9 所示。晶体的振荡频率范围为 1.2~12MHz，振荡器产生的脉冲经分频、整形后得到所需要的时钟脉冲。

在时钟脉冲的作用下，微电脑将产生一系列时序脉冲，如图 1.10 所示。其中， T_0 、 T_1 、 $\dots$ 、 T_5 为环形计数器的 CLK 端输入矩形时钟脉冲，环形计数器输出端 Q_0 、 Q_1 、 $\dots$ 、 Q_5 的输出波形。由环形计数器的工作原理可知，当第一个时钟脉冲上升沿到来之时， T_0 由低电平跳变为高电平，其他为低电平。当第二个时钟脉冲上升沿到来时， T_0 由高电平跳变为低电平，而 T_1 则由低电平跳变为高电平。以此类推，第七个时钟脉冲上升沿到来时， T_0 又由低电平跳变为高电平……形成不同的时序脉冲。

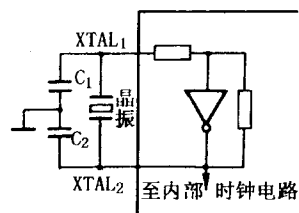


图 1.9 振荡电路

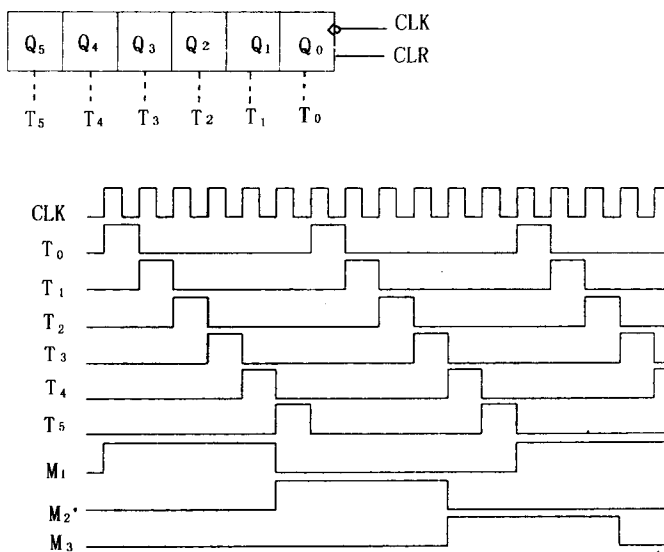


图 1.10 T 信号、M 信号定时关系图

微电脑有关定时的概念有：

1. 时钟周期 T 。时钟周期 T 是时钟脉冲两个相邻脉冲前沿的时间间隔，是 CPU 执行微操作的最小时间单位。也叫机器节拍。

2. 机器周期 M 。机器周期是指 CPU 完成某一明确规定的基本操作所需要的时间，通常指 CPU 与外界打一次交道的的时间。例如，取指令的操作码、读一个字节的操作数以及向存储单元写入一个字节的的数据等所需要的时间，称为机器周期。每个机器周期由若干个时钟周期组成。根据操作任务的不同，各种机器周期所含的时钟周期的数目也不同。图 1.10 中有 3 种机器周期 ($M_1 \sim M_3$)。

3. 指令周期。指令周期为指令从开始取出到执行完毕所需要的时间。由于指令功能的繁简不同，指令周期也长短不一，因此，一个指令周期有若干个机器周期。任一指令周期的第一个机器周期都用来取本条指令，指令规定的操作将在本指令周期的其他机器周期中执行。

微电脑处理指令的速度，受到存储器的存储时间限制，这是因为，一旦微电脑向存储器送出一个地址信号，存储器要有足够的时间才能建立稳定的信息，只有稳定后才能进行其他处理。因此，作为定时的时钟振荡器的频率要受到存储器存取时间的限制。

定时系统中，复位电平起着计数器清零的作用。每次开机或微电脑受干扰无法继续运行

时，复位电路将计数器从头计数，以免微电脑功能紊乱。

可见，定时系统的作用就是使微电脑在唯一的时钟信号控制下，严格地按时序进行工作。

1.3 微电脑的指令系统

指令是控制微电脑进行各种操作和运算的命令。CPU 所能执行的各种指令的总和称为指令系统。微电脑芯片类型不同，其指令系统也不同。指令系统是在设计芯片时就规定下来，是由计算机生产厂商定义的。因此，指令系统是用户必须理解和遵循的准则，是学习和使用微电脑的基础和工具，也是必须掌握的知识。

一、表达指令的语言

微电脑的指令主要使用机器语言和汇编语言。

机器语言是指令的二进制编码表示形式。由于微电脑所能识别的语言是用“0”与“1”两个代码书写的，所以微电脑能够直接执行机器语言。由于二进制编码书写指令很不方便，因此常用十六进制形式。采用机器语言编写程序的优点是程序送入微电脑后可直接执行，运算的速度较快。但其缺点是程序长、不易书写、难以阅读和调试、容易出错且不易查找等，为使用带来一定的不方便。

汇编语言是指令的符号表示形式。为了克服机器语言的缺点，人们用一组字母、数字和符号来代替机器语言，这些符号称为助记符。以助记符表示的指令称为汇编语言，使用汇编语言编写的程序称之为汇编语言程序。为起到助记的作用，指令常以其英文名称或缩写形式来作助记符。例如助记符 SUB 意为减去，是英文中 SUBTRACT 的缩写。

与机器语言相比，汇编语言具有简洁、明确、易读等特点。然而这种汇编语言程序，机器不能直接识别，必须将其翻译成机器语言才能为微电脑所接受。通常把用汇编语言编写的程序称为源程序，而相应的机器语言程序称为目标程序。将源程序翻译成目标程序的过程称为汇编。汇编的方法有两种，一是手工汇编，即用人工查表的方法，逐条地将指令改写成相应的机器码；另一种是机器汇编，即给计算机配备能够执行翻译工作的程序，让计算机完成这一工作。这种执行翻译工作的程序称为汇编程序。

家用电器所用的专用微电脑芯片，工作程序已固化在内存中，均为机器语言编写的程序。因此，无需再用汇编语言编写程序。

二、指令格式

指令的表示方法称之为指令格式，其内容包括指令的长度和指令内部信息的安排等。一条指令通常由两部分组成，即操作码和操作数。操作码用来规定指令进行什么操作，而操作数则表示指令操作的对象。操作数可能是一个具体的数据，也可能是指出到哪取得数据的地址或符号。

微电脑中，通常把存储器一个单元所包含的二进制代码的位数称为字长。微电脑的字长有 4 位、8 位、16 位和 32 位等几种，习惯上把 8 位字长称为字节，16 位字长称为字，即一个字有两个字节。

微电脑由于字长短不同，因此指令都是不定长的，即所谓变长指令。例如在 MCS-51 指令系统中，有一字节、二字节、三字节等不同长度的指令。

1. 一字节指令。一字节指令只有一个字节，操作码和操作数信息同在其中，前 4 位为操作码，后 4 位为操作数。例如：

1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

操作码

操作数

2. 二字节指令。二字节指令包括两个字节，其中第一个字节为操作码，第二个字节是操作数。例如：

0	1	1	1	0	1	0	0
---	---	---	---	---	---	---	---

操作码

立即数

操作数

3. 三字节指令。三字节指令中，操作码占一个字节，操作数占两个字节，其中操作数既可能是数据，也可能是地址。例如：

0	1	0	1	0	0	1	1
---	---	---	---	---	---	---	---

操作码

直接地址

立即数

} 操作数

三、指令的分类

微电脑指令系统均有数十条指令，按其功能大致可分为数据传送指令、数据处理指令、控制指令及 I/O 指令等。

1. 数据传送指令。数据传送指令是使用最多最频繁的一类指令。这类指令主要用于寄存器、存储器和外设之间传送数据。

2. 数据处理指令。数据处理指令有算术运算指令，如加法、减法、乘法、除法等；逻辑运算指令，如“与”、“或”、“异或”等。不论做哪种运算，其结果在指令作用下均送累加器 A 中。

3. 控制指令。程序控制指令的功能是改变程序计数器的内容，从而改变程序执行的顺序和方向。这类指令包括转移、调用和返回指令。

4. I/O 指令。I/O 指令是在寄存器和外设之间用于输入数据或输出数据的指令。此类指令分别用 IN、OUT 表示输入和输出。

1.4 微电脑工作过程概述

以上介绍了微电脑各部件的结构与工作原理，现在通过一个简单的例子来说明图 1.1 所示的微电脑模型机的工作过程。

为简单起见，设微电脑的指令系统共有 5 条指令，字长为 8 位，其中高 4 位为操作码，低 4 位为操作数，指令系统列于表 1.1。

表 1.1 指令系统

指令类别	指令名称	助记符	操作码	操作数	功能
传送类	取数指令	LDA, R	0000	R	$(R) \rightarrow A$
处理类	加法指令	ADDA, R	0011	R	$A + (R) \rightarrow A$
处理类	减法指令	SUBR	0100	R	$A - (R) \rightarrow A$
I/O 类	输出指令	OUT	1110		$A \rightarrow \text{OUT}$
控制类	停机指令	HALT	1111		停机

例题：66+34=？

解题程序如下：

地址	指令码	助记符	注释
R ₀	00001111	LDA, R _F	；被加数(R _F)→A
R ₁	00111110	ADDA, R _E	；加数(R _E)+A→A
R ₂	1110××××	OUT	；和数输出显示
R ₃	1111××××	HALT	；停机
⋮			
R _E	00100010		；加数
R _F	01000010		；被加数

该程序在 ROM 中的存储情况见图 1.4 和图 1.2。

合上电源，按清零（复位）按钮 CLR，使程序计数器 PC 和指令寄存器 IR 清零，并使节拍发生器停在 T₀ 节拍。按启动按钮，时钟电路产生时钟脉冲 CLK 和时序脉冲 T，使时序脉冲按 T₀、T₁…T₅、T₀…顺序循环出现。

一、LDA, R_F 指令执行过程

1. T₀ 节拍。操作控制部件 CU 发出控制电平 E_P 和 L_M，当 CLK 上升沿到来时，程序计数器 PC 的状态 0000 经总线送入内存地址 MAR，经译码选中 ROM 中的 R₀ 存储单元（见图 1.4）。

2. T₁ 节拍。操作控制部件 CU 发出控制电平 E_{PR} 和 L_I，当 CLK 上升沿到来时，R₀ 单元中的内容 00001111 经总线送往 IR（见图 1.7），其中操作码部分经译码后得到译码信号 LD=1（见图 1.8）。

3. T₂ 节拍。操作控制部件 CU 发出控制电平 C_P，使程序计数器 PC 自动加 1，其状态变成 00000001（见图 1.6）。

4. T₃ 节拍。操作控制部件 CU 发出控制电平 E_I 和 L_M，将指令寄存器 IR 中的低 4 位被加数地址经总线送 MAR，经地址译码器译码 R_F=1。

5. T₄ 节拍。操作控制部件 CU 发出控制电平 E_{PR} 和 L_A，在 CLK 上升沿到来时，将 ROM 中的 R_F 单元内容经总线送往累加器 A，累加器 A 中的内容为 01000010，同时，累加器 A 中的内容直接送到全加器一组输入端上。至此完成了 LD 指令的整个执行过程。

二、ADDA, R_E 指令的执行过程

取指令和分析指令阶段的 T₀、T₁、T₂ 节拍过程与 LD 指令完全一样，下面介绍执行指令阶段的 T₃、T₄、T₅ 节拍过程。

1. T₃ 节拍。操作控制部件 CU 产生控制电平 E_I 和 L_M，将指令寄存器 IR 中的低 4 位加数地址码经总线送内存地址寄存器（见图 1.7、图 1.4），经译码后得到 R_E=1，即选中加数存储单元。

2. T₄ 节拍。操作控制部件 CU 发出控制电平 E_{PR} 和 L_B，将 ROM 中的 R_E 单元内容经总线送往寄存器 B（见图 1.5），寄存器 B 中的内容为 00100010，寄存器 B 中的内容直接送到全加器另一组输入端，全加器做加法运算。

3. T₅ 节拍。操作控制部件 CU 发出 E_V 和 L_A 控制电平，全加器输出的和数 01100100 送往