

微处理机

编程与接口

浙江省科技情报研究所
浙江省计算技术研究所

Z—80

微处理机程序设计和外围接口

第 二 册

美国布拉克斯堡、连续教育丛书

目 录

第一章	Z—80接口	(1)
第二章	NEZ—80实验台	(24)
第三章	同步脉冲发生: 地址和设备选择脉冲	(52)
第四章	总线, 三状态缓冲器和 Z—80输入, 输出	(146)
第五章	毫微计算机硬件和软件系统	(221)
第六章	中断处理	(298)
第七章	Z—80并行 I/O电路 (PIO)	(359)
第八章	TTL 集成电路测试器	(433)
第九章	Z—80计数器/定时器电路(CTC)	(456)

附录 A

在实验中使用的软件汇编	(493)
-------------------	---------

附录 B

实验软件明细表 (略)	(514)
-------------------	---------

附录 C

使用 MOS 器件注意事项	(515)
---------------------	---------

附录 D

毫微计算机简图	(516)
---------------	---------

第一章 Z—80 接 口

引言

本章将要介绍微处理机接口问题，对所涉及的概念给出定义，概述一些接口技术。

目 标

学完本章后，读者将能做到：

- 定义什么是微处理机接口，并给出微处理机接口应用实例。
- 说明微处理机接口的目的。
- 说明 Z—80 微处理机使用的一些接口技术
- 讨论以 Z—80 为基础的微计算机的三条重要的总线。
- 讨论 Z—80 CPU 发出的四个重要的接口控制信号： $\overline{RD}$ ， $\overline{WR}$ ， $\overline{MREQ}$ ， $\overline{IORQ}$ 。
- 讨论 Z—80 微处理机的引出线。
- 能够看懂时间图。人们用时间图说明 Z—80 芯片内部事件的发生与“外部世界”事件的发生之间的同步关系。
- 给出与接口有关的一些定义：同步，I/O 设备，CPU 和存储器。

接口是什么？

接口可以定义为一个组织（如若干人、仪器等）的成员的结合方式，在这种方式下，他们能以兼容而协调的方式* 工作。所谓“兼容而协调的方式”常常就是指同步。下面给出一些常用术语的定义：

* 详见附录 E

同步——对于两个设备或机器，同步就是步调一致或相位一致。而用于计算机则意味着用同一个时钟信号或时钟脉冲控制一系列操作的执行。

同步计算机——所有普通操作都受一个主时钟控制的数字计算机。

同步操作——系统在同一时钟控制下的操作。

同步逻辑——一个系统的操作由时钟脉冲触发而发生，以这种方式工作的数字逻辑叫同步逻辑。

同步(Sync)——这是同步、同步的、同步化等等的缩写。

同步化——迫使系统一个元件的工作与其它元件的工作保持步调一致。

同步脉冲——由发送设备产生送到接收设备，以使这些设备工作保持同步的脉冲。

同步输入——触发器的某些输入，它们不象在门电路那样直接控制输出，而只是在时钟脉冲允许时才起作用。

上述定义来自参考书目 2。下面定义计算机接口：

在计算机和外部装置*（包括存储器和输入输出设备）之间的数字式数据传输的同步叫做计算机接口。

虽然计算机接口的细节随着计算机种类不同而不同，但对于各种各样的计算机总还是有一个接口的一般原则。Z—80微处理器CPU的接口的基本目标见图 1—1，概述如下：

- 输入：从外部装置传送数据到微处理机。
- 输出：从微处理机传送数据到外部装置。
- 同步脉冲的产生：产生适当的输入、输出数据传送同步脉冲，叫设备选择脉冲，使外部装置和微处理机工作协调。
- 中断处理：检测并处理由外部装置发到微处理机的中断信

* external device 在本书中含义较广，为免与通常所说外部设备混淆，故译为外部装置。——译注

号。

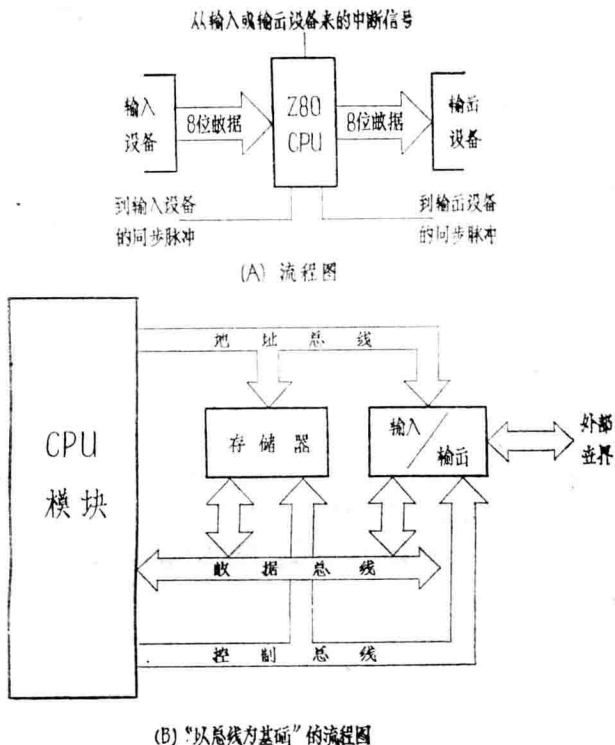


图 1—1 接口的四项主要任务：输入，输出，设备选择脉冲的产生和中断处理

CPU 和外部装置之间数据传送通过数据总线进行。Z—80 的数据总线为 8 位宽的双向总线，即信息交换是通过 8 条线并行地将数据传入或送出 Z—80 CPU。通过地址总线选择指定进行数据传输的外部装置。Z—80 CPU 用 16 位地址总线中的低 8 位来选择 I/O 设备，全部 16 位总线用于访问内存。控制总线上传送同步信号，使得 CPU 和外部装置工作时数据总线和地址总线上

的信息保持同步。因此，在 CPU 和外部装置之间交换信息时，三条总线：数据、地址和控制总线都要参与。

本章及以下各章的目的在于使读者掌握一些接口技术，实现以 Z—80 为基础的毫微计算机同其它设备的接口。但在具体谈到接口之前，先要研究一下为什么要为微计算机配接口。

为什么微计算机接口很重要？

为了回答“为什么接口很重要？”这个问题，让我们从一篇优秀的微计算机接口论文中引证几段。这篇文章叫“聪明的机器革命：使产品智能化”，登载于一九七六年七月五日 McGraw-Hill 出版的“商业周刊”（Business Week）。

“这是第二次工业革命，” TRW 公司副总经理西德尼·韦布说：“正如第一次工业革命使人的体力成倍增强一样，它使人的脑力也成倍增强。”

这次革命的动力是微处理机，或单片计算机，即做在微小硅片上的计算机的算术逻辑运算心脏。应用“微处理机头脑”的产品第一个浪潮正涌向市场，证明了这是前所未有的一种强有力的工具，可以构成“聪明的”机器——除了通常功能外还能作出决定、运算和存储信息的机器。

由单片计算机构成的新产品对于消费者将是最感动人心的东西。微处理机将进入家庭、汽车、各种器械以及其它消费品，其数量将远远超过进入其它产品领域的数量。“到 1980 年以前每个家庭将拥有 7 到 10 台微处理机”霍尼韦尔公司领导微处理机事业的安德鲁·A·佩劳斯基这样说。他的公司已致力于家用能源管理和安全系统。

在工厂，单片计算机使智能电子产品价格降得很低，以至于最小的生产设备也能改成智能机器。智能生产设备、传感器和其它仪器可以联接起来构成分布式数据采集和控制系统，这就加速了完全自动化的工厂的实现。

微处理机销售额猛增和新的智能机器浪潮的出现的关键因素很简单，就是价格因素。仙童摄象机和仪器公司副总经理C·莱斯特·霍根几周前在波士顿会议上戏剧性地说明了这个因素。他从口袋里掏出18个微处理机，抛向听众。“这些微处理机是具有相当于二十年前价值1800万美元的计算机效能的东西”，他说。霍根解释说，他的20美元的微处理机的功能相当于IBM的第一台商用计算机的功能，而后者在五十年代初价值100万美元。“我要说明的一点”，霍根说“就是今天计算机的功能基本上是可以免费得到的”。

在发展以微处理机为基础的产品方面：

从常规的电子元器件(如集成电路)转变为MPU*，由于它代替了成百个集成电路和其它元件，节省了设计时间和制造成本。一旦设计的新产品中包含有MPU，该产品即具有巨大的市场优势。如要改变该产品的功能，不再需要重新进行电子线路的花费巨大的设计工作，而只要简单地改变存在MPU存储器中的那些指令或软件即可。可以几乎不增加成本地增加新的功能，而且新的智能机器能完成很多从前不能经济地完成的工作。

今天软件不仅是MPU用户的最大问题，而且也是成本中最主要部份，“实际上软件价格比重，对于微计算机比对于小型计算机还大。”理查德·马勒这样说，他是一个新汉普夏公司(New Hampshire)顾问，他已为几家小公司设计了一些智能产品。他说，他花在每一项软件设计上的费用达10万美元，而硬件设计费用降到大约仅2万美元。

从这些引证可以看出微处理机的应用一定会很快遍布各处，重要之点是要认识到要利用微处理机的功能就总是要用到接口技术。

* MPU即微处理机——译注

接口如何实现？

前面已谈过接口的四个主要任务：输入，输出，同步脉冲发生，中断处理。现在要对一些术语给出正式定义。

总线(bus)——这是一种数字信息传送的通路，从任何一个源设备到任何一个目的地设备的信息传送都是通过总线，但在一个时候只能有一个这种信息传送在进行。当这样的信息传送在进行时，总线上所有其它的源设备都被禁止发送。

双向数据总线——数字信息在上沿两个方向都可以传送的总线。在 Z—80 微处理机系统中，用双向数据通路在 CPU、内存和其它外设之间传送数据。

地址总线——是一条单向总线，用以寻找特定的存储单元或 I/O 设备的数字信息由此总线传输，Z—80 微处理机的地址总线由 16 条并行的线构成。

地址——用以识别特定存储单元外设的一组二进制码。Z—80 微处理机 CPU 用 16 位来识别存储单元，而外设则只用 8 位地址识别。

控制器——计算机中按适当顺序解释指令，产生适当的同步信号以执行指令的部份。

控制总线——管理微计算机系统（包括内存和外设）操作的信号传输所经过的一组线，这些信号可以是 CPU 产生的，也可以是从外部装置产生。Z—80 微计算机系统控制总线宽度为 13 个二进制位，其上传送的信号有：使 CPU 和内存、外设之间的 I/O 操作同步的信号，控制 CPU 的信号如中断、等待和停止，还有控制使用地址总线 and 数据总线的信号。

I/O——输入/输出(input—output)的缩写符号。

I/O 设备——输入/输出设备。指卡片读出机、磁带机、打印机或其它类似设备，用以与计算机或辅助存储器交换数据。从更一般意义上讲，指任何数字设备，包括一个集成电路块，它们发

送数据给计算机，或从计算机接收数据或选通脉冲。

CPU——中央处理机的缩写。

中央处理机（大计算机）——计算机系统中包含控制器、运算器和特殊寄存器组的部份。它控制指令处理、执行算术运算，提供同步信号和其它内部操作信号。

中央处理机（微处理机）——一种单片集成电路，执行数据传送、控制、运算、逻辑、中断处理等操作，其操作按照由存储器取得的指令进行。

存储器——能存放逻辑 0 和逻辑 1 信息的设备。其存入或取出方式可以是按单独一位或一组二进位（叫一个字）进行写入或读出。

Z—80中央处理机 (CPU)

记住以上定义，然后更详细地看图 1—1。特别要仔细地看看CPU组件。毫微计算机的 CPU 组件是一个40条引线的集成电路片，叫做Z—80CPU。在上册我们已经研究过 Z—80CPU的指令系统，现在我们要把注意重点由 Z—80软件转向 Z—80硬件，首先我们要研究一下 Z—80微处理机的框图，再看看作为 Z—80电子线路外壳的双列直插式封装的40条外引线，最后研究一下时间图，这些图精确地说明 Z—80和存储器以及外设是如何通讯的。

图 1—2 是Z—80CPU的功能框图，此图来自SGS—ATES出版的Z—80CPU技术手册。当 Z—80CPU 执行存储在存储器中的程序时，将存储器中指令顺序读出、将程序计数器 (PC) 内的指令地址送到地址总线上，再产生适当的控制信号送到控制总线以启动内存工作，然后将数据总线上的数据读进 CPU 中相应寄存器内。显然，一定要严格同步才能保证，当CPU从数据总线读数时，该单元的存储内容已经送到了数据总线。CPU控制器的功能就是要协调这些任务并保证指令操作码 (OP) 放入指令寄存器

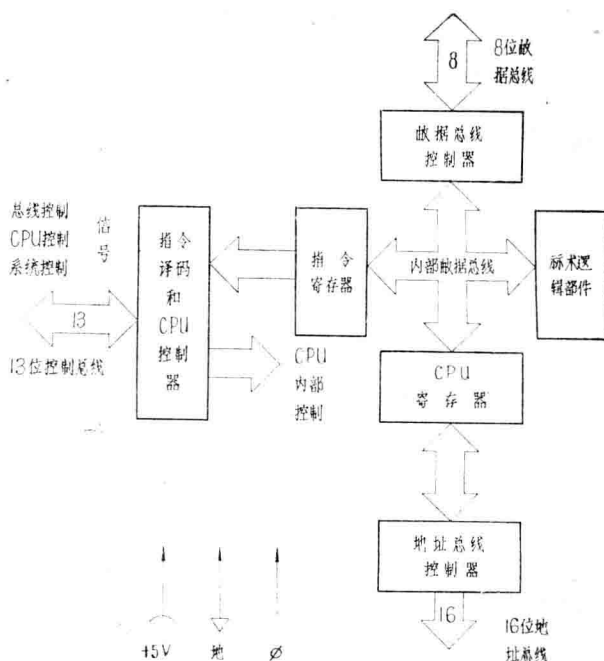


图 1—2 Z—80功能框图

并正确译码，而且要控制 ALU 执行所有 Z—80指令系统规定的算术和逻辑操作。这些操作包括加、减、逻辑与、逻辑或、逻辑异或、比较、左右移和旋转、递增、递减、位置位、位复位、位测试等。ALU 在执行这些操作时，通过内部数据总线与22个内部寄存器、指令寄存器和数据总线控制器通讯。数据总线和地址总线的控制器管理CPU 外部世界通过相应的总线交换数据的活动。注意，虽然数据总线是双向的，但地址总线都是单向的，由CPU 输出。CPU 不从地址总线接收任何数据，CPU 寄存器结构图见图 1—3。图中每个小方块是 8 位寄存器，每个长方块是16位寄存器，注意12个 8 位寄存器可以成对使用，而形成 6 个16位寄存

器。

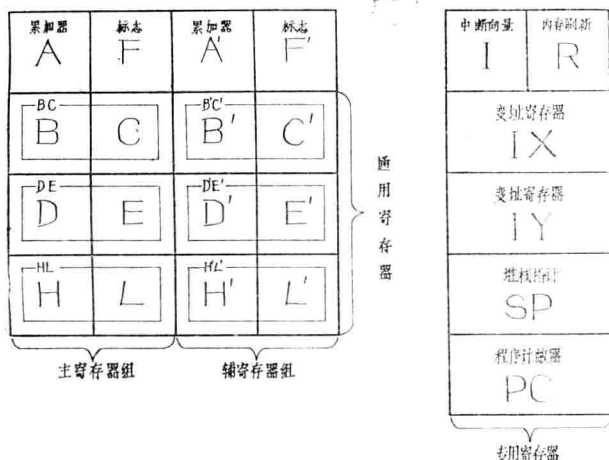
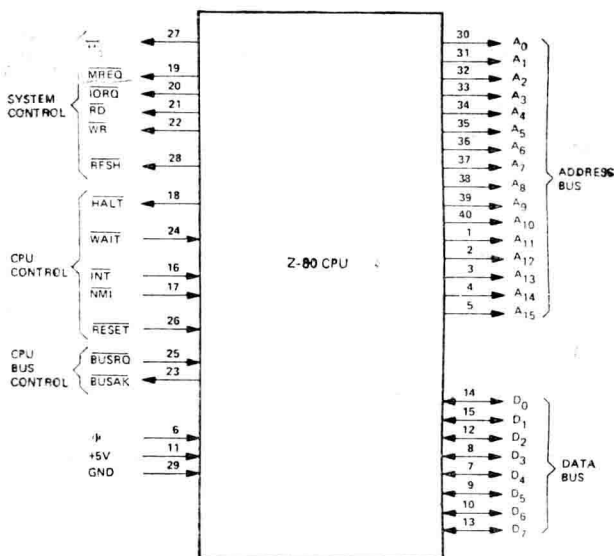


图 1—3 Z—80 CPU寄存器

Z—80CPU 引线说明见图 1—4，它来自 SGS—ATES 出版的 Z—80技术手册，要特别注意地址总线、数据总线和系统控制信号的说明。在本章及下两章，我们将要仔细研究这些信号。其余信号也将或简或繁地加以介绍。因此不要指望这时由此引线说明即能弄明白一切。本书下册目的之一就是在学完后能够理解和运用这些知识于 Z—80接口的实际应用中。在此提出的目的只是告诉读者一个大概情况，使读者有个概念，Z—80CPU只不过是一个40条引线（图1—4）的集成电路，没什么可怕的。

每一条引线功能如下：

A0—A15(地址总线)三状态输出，有效时输出高电平。A0—A15构成16位的地址总线为同存储器(可达64K字节 = 2^{16}) 交换数据和同 I/O 设备交换数据提供地址。I/O 寻址利用 8 个低位地址，可以直接选择 $256 = 2^8$ 输入或输出端口。A0 是最低位地址位。在刷新(refresh)时，低 7 位存放的是有效刷新地址。(存



Courtesy SGS-ATES Componenti, Elettronici SpA
Fig. 1-4. Z-80 CPU pin configuration.

图 1—4 Z—80CPU 引线图

存储器刷新将在第 3 章讨论)。

D0—D7 (数据总线)：三状态输入/输出，有效时高电平。D0—D7 构成 8 位双向数据总线，数据总线用于同存储器 and I/O 设备交换数据。

$\overline{M1}$ (机器周期 1)：输出，有效时低电平。 $\overline{M1}$ 指示现行机器周期是指令执行的取操作码周期。注意，对于二字节操作码指令，在取每一操作码字节时都要产生 $\overline{M1}$ 。这些二字节操作码总是以 CB、DD、ED、或 FD (十六进制) 开始。 $\overline{M1}$ 也总是伴随着 $\overline{IORQ}$ 出现以指示中断认可 (acknowledge) 周期。

$\overline{MREQ}$ (存储器请求)：三状态输出，有效时输出低电平，存储器请求信号，表明地址总线上为存储器读写操作的有效地

址。

$\overline{\text{IORQ}}$ (输入, 输出请求): 三状态输出, 有效时低电平, $\overline{\text{IORQ}}$ 信号表明地址线的低位部份为 I/O 读写操作所需有效 I/O 地址。当中断被接受时, 与 $\overline{\text{M1}}$ 信号同时也产生 $\overline{\text{IORQ}}$ 信号, 表明中断响应向量可以放到数据总线上。在 $\overline{\text{M1}}$ 期间可以出现中断认可操作, 但不可能出现 I/O 操作。

$\overline{\text{RD}}$ (存储器读): 三状态输出, 有效时低电平。 $\overline{\text{RD}}$ 表明 CPU 要从存储器或 I/O 设备读取数据。被访问的 I/O 设备或存储器应该用此信号将数据送到 CPU 数据总线上。

$\overline{\text{WR}}$ (存储器写) 三状态输出, 有效时低电平, $\overline{\text{WR}}$ 表明 CPU 数据总线上为将要写入存储器或 I/O 设备的有效数据。

$\overline{\text{RFSH}}$ (刷新): 输出, 有效时低电平, $\overline{\text{RFSH}}$ 表明地址总线的低 7 位为动态存储器的刷新地址, 现行 $\overline{\text{MREQ}}$ 信号应该用于使所有动态存储器进行刷新读出。

$\overline{\text{HALT}}$ (停机状态): 输出, 有效时低电平, $\overline{\text{HALT}}$ 表明 CPU 已执行 $\overline{\text{HALT}}$ 软件指令, 正在等待发生不可屏蔽或可屏蔽中断 (同时中断触发器置位), 然后才能恢复操作。虽然是停机, 但 CPU 在执行 NOP 以维持存储器刷新。

$\overline{\text{WAIT}}$ (等待): 输入, 有效时低电平。 $\overline{\text{WAIT}}$ 通知 Z-80 CPU 被访问存储器或 I/O 设备尚未作好数据传送的准备, CPU 继续处于等待状态直至此信号消失。此信号使任何速度的存储器或 I/O 设备都可以与 CPU 同步。

$\overline{\text{INT}}$ (中断请求): 输入, 有效时低电平。中断请求信号由 I/O 设备发出, 在现行指令结束时, 如果内部的软件控制的允许中断触发器 (IFF) 置位, 而且 $\overline{\text{BUSRQ}}$ 信号未出现, 则中断请求可接受。当 CPU 接受了中断, 在下一指令周期初即送出一个认可信号 (在 $\overline{\text{M1}}$ 期间的 $\overline{\text{IORQ}}$) CPU 可以按三种不同方式响应中断) 这在下面将详细说明。

$\overline{\text{NMI}}$ (不可屏蔽中断): 输入, 负边缘触发, 不可屏蔽中断请求线比 $\overline{\text{INT}}$ 优先权更高, 总是在现行指令结束时被接受, 而不管允许中断触发器状态如何。 $\overline{\text{NMI}}$ 使Z—80CPU自动执行重新起始转到0066(十六进制)单元。程序计数器被自动存入外部堆栈, 以便能够返回到被中断的程序, 注意, 连续的WAIT周期会造成现行指令不能执行完毕, 并且 $\overline{\text{BUSRQ}}$ 会使 $\overline{\text{NMI}}$ 无效。

$\overline{\text{RESET}}$: 输入, 有效时低电平。 $\overline{\text{RESET}}$ 将程序计数器清0并使CPU置成初始状态, CPU的初始状态包括以下内容:

- (1) 使允许中断触发器复位。
- (2) 将寄存器I置成00(十六进制)
- (3) 将寄存器R置成00(十六进制)
- (4) 将中断方式置成0方式。

在复位期间, 地址总线 and 数据总线进入高阻抗状态, 所有控制输出信号都消失。

$\overline{\text{BUSRQ}}$ (总线请求): 输入, 有效时低电平。总线请求信号用于请求将CPU的地址总线、数据总线 and 三状态输出控制信号线都变成高阻状态, 以便其它设备能控制这些总线, 当 $\overline{\text{BUSRQ}}$ 信号出现后, 现行CPU机器周期一结束, CPU就立即将这些线都置成高阻状态。

$\overline{\text{BUSAk}}$ (总线认可): 输出, 有效时低电平, 总线认可信号用以通知发出总线请求的外部装置, CPU地址总线、数据总线 and 三状态控制总线信号都已进入高阻状态, 它们可交由该设备使用。

ϕ : 单相TTL电平时钟输入。

Z—80指令周期: 机器周期和T周期

现在要一步步地介绍在一条指令执行期间, Z—80芯片内和在其40条引线上出现的一系列变化。每一条Z—80指令由一系列

叫做机器周期的基本操作组成。Z—80只有7个基本操作（机器周期）。这7个Z—80机器周期是：

1. 读取指令操作码（M1周期）
2. 存储器读或写周期。
3. I/O 读或写周期。
4. 总线请求/认可周期。
5. 中断请求/认可周期。
6. 不可屏蔽中断请求/认可周期。
7. 从 HALT 指令转出周期。

显然，头6个周期与接口的四项主要任务直接有关。

本章及下面几章将限于讨论前三类机器周期。

执行存放于0100单元的指令 LDA, 00H（十六进制3E00）时，Z—80首先执行一个M1周期，从存储器取出指令操作码。这就必须将地址0100送到16位的地址总线上，并发出控制信号 $\overline{\text{MREQ}}$ 和 $\overline{\text{RD}}$ （第19脚和21脚）。存储器对这两个控制信号的解释是：

$\overline{\text{MREQ}}$ ——有效时低电平（逻辑0），意味着要访问存储器。

$\overline{\text{RD}}$ ——有效时低电平（逻辑0），意味着这是一次读操作，地址总线上的0100说明要从哪个单元读出。

然后将此存储单元0100的内容读出到数据总线上。CPU即从数据总线取走此0100单元的内容。当进行读取指令字第一个字节操作时，发出 $\overline{\text{M1}}$ 信号表明此字节一定是操作码。当指令操作码为两个字节时，这就要执行两个M1周期以完全取出操作码。在此例中，读操作码3E只需要一个M1周期。

Z—80对3E译码为将立即数传送到累加器的指令，于是马上执行一个存储器读周期将0101单元内容读进累加器。一个存储器读周期几乎和M1周期完全一样，只是 $\overline{\text{M1}}$ 信号不出现，以及

时间关系稍有点不同。

在研究M 1 机器周期的时间图之前，必须先介绍一下T周期的概念，每个机器周期又进而分为T周期。T周期与Z—80组件的第六脚的时钟脉冲输入 ϕ ——对应。经过挑选的Z—80的最高时钟频率可达4兆赫，即每秒4,000,000周。所以T周期为250毫微秒宽。每个机器周期包含的T周期数随着机器周期功能不同而不同，功能越复杂，T周期数越多。一个M 1 机器周期由4个T周期组成，而一个存储器读机器周期则由3个T周期组成。M 1 周期比较复杂些是因为需要将指令译码。

概括地说，指令由机器周期组成，而机器周期又由T周期组成，指令的第一个字节是操作码字节，所以Z—80在执行任何指令时总要至少执行一个M 1 周期，虽然有各种不同机器周期，但T周期却只有一种，也就是4MHZ时每250毫微秒发生一次逻辑转变，从逻辑0到逻辑1，又到逻辑0。（见图1—5）

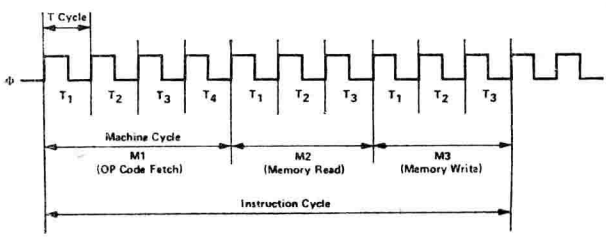


Fig. 1-5. Breakdown of Z-80 instruction into Machine cycles and T-cycles.

图 1—5 Z—80指令周期分割为机器周期和T周期

Z—80CPU时间关系图

图1—6粗略地表示了一个M1周期内Z—80CPU组件各引线上的地址、数据和控制信号相互之间的时间关系。和其余Z—80机器周期的时间图一样，这个时间图在Z—80CPU技术手册中也可找到。如果了解Z—80CPU引线上各信号之间的绝对时间