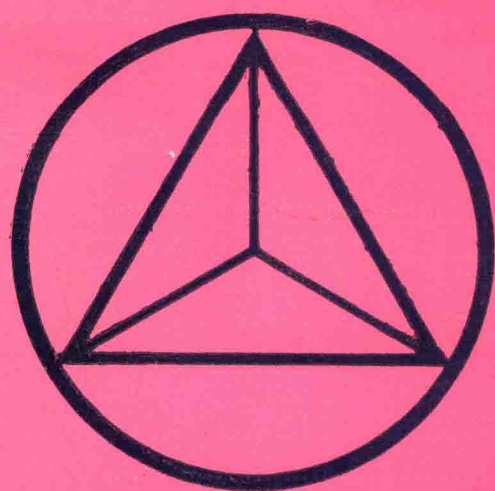


晶峰器件应用



11

上海元件五廠

上 海 元 件 五 厂

生 产 经 营 总 纲

确立经营为中心 讲求实效为重点

* 产 *

* 品 *

* 方 *

* 向 *

坚决贯彻大规模集成电路为主，大规模电路和中小规模电路结合的方针。大力开发各类大规模集成电路，努力生产优质线性电路、MOS电路和低噪声器件，加速发展CMOS模拟电路，认真搞好高速和超高速集成电路，以形成一个综合性大规模集成电路开发和中试生产基地。

《经营 管理 百 字 方 针》

* 经 *

* 营 *

* 思 *

* 想 *

立 足 长 远 ，	着 眼 当 前 。
打 好 基 础 ，	快 上 水 平 。
开 拓 市 场 ，	加 速 更 新 。
巩 固 老 户 ，	发 展 新 户 。
胸 怀 全 局 ，	面 向 世 界 。

* 经 *

* 营 *

* 方 *

* 针 *

质 量 可 靠 ，	确 保 经 久 耐 用 。
性 能 优 良 ，	达 到 国 际 标 准 。
品 种 齐 全 ，	形 成 系 列 配 套 。
价 格 低 廉 ，	实 行 薄 利 多 销 。
应 用 广 泛 ，	渗 透 各 行 各 业 。
服 务 周 到 ，	满 足 用 户 需 要 。

“晶 峰” 器 件 应 用

第 十 一 期

1 9 8 4 年 3 月 出 版

目 录

- (一) 5G8085A单片 8 位中央处理单元 (CPU)温杰西 (1)
- (二) 5G8253 可编程计时器陈丽珍 纪兰花 (15)
- (三) 5G8257 可编程的DMA控制器的功能和用途.....张守桂 (40)
- (四) ADC0801 与 8 位微处理器兼容的 A/D转换器(译文).....周燕群 (54)
- (五) CC4000B系列产品特性简介.....王福奎 (84)
- (六) 巧妙地测试ADC(模/数转换器)的特性.....周燕群 王海麟 (97)

用 户 园 地

- (七) 5G7650 的使用体会华东电子仪器厂 沈慧珍 (100)
- (八) 5G7556—CMOS双时基电路用于报警装置
.....上海铁路局科研所通号室 黄肖英 钟永昌 (101)
- (九) ES16/14500 一位微处理机上海自力电子设备厂 宋 庆 (102)
- (十) 模拟温(湿)度试验用的数字逻辑控制单元
.....邮电部一所 王裕春 (106)
- (十一) 时钟, 报时, 告警电路.....上海市电报局 高世云 (115)
- (十二) 555电路在收音机调整中的应用.....中国人民解放军89766部队 张开基 (118)

计 算 机 应 用

- (十三) RTBMMS大规模集成电路计算机辅助制版软件系统 (一)
.....傅镇康 董伯明 王兴汉 (121)
- 附: 晶峰器件应用 1—10期总目录..... (125)

5G8085A单片8位中央处理单元 (CPU)

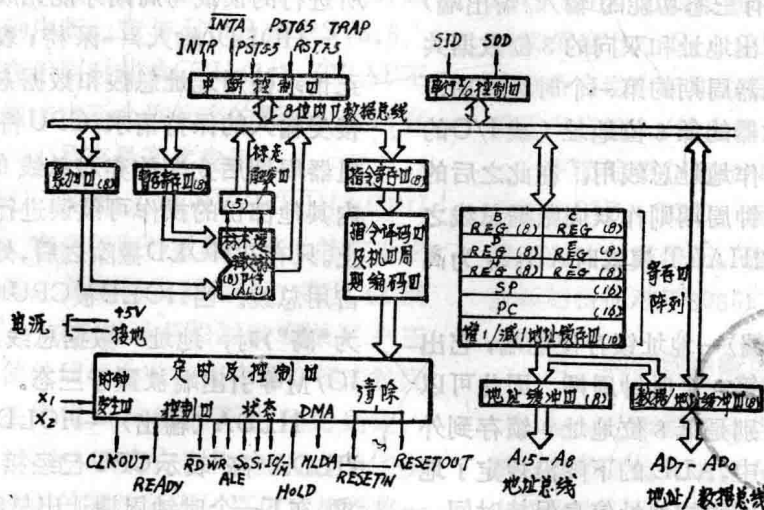
温杰西

一、概述

8085A是美国intel公司为了继续保持自己在微型处理机领域中的领先地位,于1977对原开发的8位微机中央处理单元(CPU)著名的intel8080作了较大改进,推出了新的8位微型机中央处理单元。作为一个完整的8位并行的CPU。8085A无论在工作速度,内部功能,系统的简化诸方面均较原来的8080有了较大的改善,给现在8080的用户提供了一个性能更高,使用更方便的微处理机,是目前公认的8位微处理机优良品种之一。8085在设计上充分利用了8080系统已开发的丰富的软件和硬件资源,在原指令的基础上,利用8080未用的机器码增加了两条新指令,所以直至机器码这一级,8085与8080软件完全兼容,程序编制人员不必进行任何新的培训就能适应由8085组成的系统。在制造技术方面,intel采用本公司开创的HMOS工艺,使工作频率提高至3MC,芯片集成密度

提高,在芯片内加进了时钟发生器一位I/O口以及中断优先级处理电路,使功能强化。采用8085及其二片系统专用部件RAM/IO,ROM/IO就能构成一个微型机系统。在系统逻辑设计上采用了复用总线技术,数据(8位)与低8位地址共用一条8位地址/数据总线,节省下来的引出线则用于扩充芯片功能,除肩负8224,8228的功能的引出端外,还增加了四条向量中断输入端和一位串行输入、输出端,这样,与原8080相比同为40条引出线,芯片功能明显地增加。同时由于采用N沟E/D技术,故可用+5V单电源工作,给用户带来了极大的方便。为了完善和发展我国的8位微型计算机,1983年我厂承接了清华大学微电子学研究所的科研成果试制出了全仿intel8085A的单片8位微处理机5G8085,实际使用证明5G8085无论在使用频率,功耗,电源拉偏等方面均与进口intel8085相同能直接转换使用。

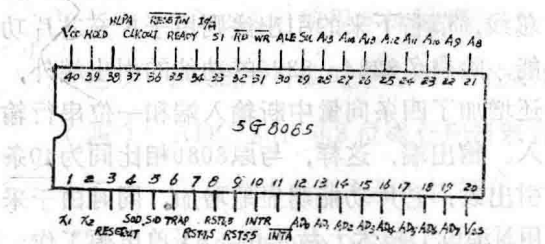
二、8085功能框图



图一 8085ACP功能框图

8085ACPU 的功能框图看起来并不陌生，它和原8080A的功能框图基本上是一样的，只是增设了中断控制电路和串行 I/O 控制电路。定时及控制部份的功能也显然得到了强化，同时地址总线被分裂为高 8 位 A_{15-8} 总线和低 8 位地址/数据复用总线 AD_{7-0} 。除此之外 CPU 的组成没有发生任何变化。寄存器阵列的寄存器原封不动址保留着。由于 W、Z 寄存器只是在指令执行过程中用于暂存中间结果，不能由指令编程调用，故框图中未画出。

三、8085的管脚排列图：



四、8085的管脚功能：

A_{8-15} （具有三态功能的输出端）—为 8 位的输出地址总线，它输出存贮器的高 8 位地址或 I/O 的 8 位地址。在 HOLD（保持）和 HALT（暂停）模式时，此 8 个输出端均处于高阻状态。

AD_{0-7} （具有三态功能的输入/输出端）—是低 8 位的输出地址和双向的 8 位数据共用的总线；在机器周期的第一个时钟周期，该总线送出存贮器的低 8 位地址（或 I/O 的 8 位地址），即作地址总线用。在此之后的第二，第三个时钟周期则作双向数据总线之用。在 HOLD 和 HALT 模式时 AD_{0-7} 为高阻。

ALE（输出端）—地址锁存使能端；它出现于机器周期的第一个时钟周期，因此可以用它将地址（特别是低 8 位地址）锁存到外围电路的锁存器中，ALE 的下降沿规定了地址的建立时间和所需的地址信息保持时间。

ALE 也被用来选通状态信息。ALE 不可能处于三态。

S_0 、 S_1 （输出端）—数据总线状态信息，总线周期的编码状态如下：

S_1	S_0	状态信息	注： S_1 可以看作所有总线传送条件下的 $R/\overline{W}$ ，因此可用作 $R/\overline{W}$ 的先期信息。
0	0	暂停	
0	1	写	
1	0	读	
1	1	取指	

$\overline{RD}$ （三态输出端）—读， $\overline{RD}$ 低电平表示被选中的存贮器或 I/O 设备要作读出操作，并表示数据总线此时用作数据传送。 $\overline{RD}$ 端在保持和暂停模式时处于三态。

$\overline{WR}$ （三态输出端）—写控制， $\overline{WR}$ 低电平表示数据总线上的数据是要写入所选的存贮器或被选中的 I/O 单元，数据在 $\overline{WR}$ 的后沿之前写入存贮器或 I/O 单元，在 HOLD 或 HALT 模式时 $\overline{WR}$ 端为三态。

READY（输入）—准备好，在读或写周期中，如果 READY 是“高”则表示存贮器或外设已准备好发送或接收数据。若 READY 是“低”（即表示尚未准备好）则 CPU 必须等待数个时钟周期，直至 READY 变高之后所进行的读或写周期才能完成。

HOLD（输入）—保持；表示另一个用户正请求使用地址总线 and 数据总线，一旦 CPU 接受输入的保持请求，CPU 将在完成现行的机器周期后立即放弃对总线的占用。CPU 内其他部份的操作可继续进行并不因此而停止。只有在 HOLD 撤除之后，处理机方能再度占用总线。当 HOLD 被 CPU 响应（即 HLDA 为“高”）时，地址—数据总线， $\overline{RD}$ 、 $\overline{WR}$ 和 $IO/\overline{M}$ 等引出端被置于三态。

HLDA（输出）—HOLD 的响应讯号，HLDA “高”表示 CPU 已经接受 HOLD 的请求，在下一个时钟周期让出总线。HOLD 请求

撤除之后HLDA变低,在HLDA变为低电平之后,一个半时钟周期 CPU重新获得总线。

INTR(输入)—中断请求,该端被用于一般性中断请求。该中断请求只能在现行指令的倒数第二个时钟周期或是在HOLD、HALT状态时被采样。若中断请求有效,则程序计数器(PC)将停止向上计数并发出INTA讯号。此时RESTART(重新启动)或(CALL)(调用)指令可以被插入从而转入中断服务程序,INTR允许与否由软件决定,复位讯号Reset可以禁止INTR请求。而且中断接受之后INTR请求立刻被禁止。

INTA—中断响应输出端,INTA低电平表示CPU已响应中断请求。在INTR被接受之后的指令周期中,INTA用以代替RD(它具有与RD同样的定时关系。)用于中断指令的读取。INTA还可用来启动8259中断芯片或其他中断器。

RST5.5
RST6.5
RST7.5 } 重新启动中断输入端—这三
个输入端具有和INTR一样的
定时,不同之处在于它们能产生一个自动插入的内部重新启动动作。在这三者中RST7.5是最高优先级,RST6.5次之,RST5.5是最低优先级,但它们的优先级别均高于INTR。用SIM(置中断屏蔽)指令可以对它们进行分别屏蔽。

TRAP(输入)—陷阱中断是一个非屏蔽的重新启动中断,它与INTR或RST5.5,6.5,7.5在相同时刻被CPU采样。TRAP不受任何屏蔽或中断允许状态的影响,在所有中断中TRAP具有最高优先级。

RESET IN(输入)—RESET IN是CPU的复位端,它使程序计数器置零,并使中断允许触发器,暂停触发器复位。在复位期间,数据,地址以及各控制线都被置于三态。复位使处理机内的各寄存器,标志触发器内容发生变化,其结果不能预先断定,RESET IN输入具有史密特功能,故为了实现电源启动时的复位延迟,可以直接连接一

个RC网络。只要存在复位讯号,CPU就保持在复位状态。

RESET out(输出端)—该输出端高电平表示CPU处于复位状态。此讯号也可用作系统复位的讯号。RESET out与处理机时钟同步,持续时间是时钟周期的整数倍。

X₁, X₂(输入端)—此二端与外部晶体或R/C网络相连接以建立内部时钟。当不用晶体时,X₁还可作为一个来自外部逻辑门的时钟输入端,该输入在8085内部二分频之后才是内部的工作频率。

CLK(输出端)—当晶体或R/C网路接入CPU的X₁X₂输入端时,CLK输出系统时钟讯号。CLK的周期是X₁X₂输入周期的两倍。频率是振荡频率的 $\frac{1}{2}$ 。

IO/ $\overline{M}$ (输出端)—该讯号指出进行读写的是存储器还是I/O外设。在HOLD和HALT模式时IO/ $\overline{M}$ 为三态。

SID—串行数据输入端。在执行读中断屏蔽指令RIM时,SID线上的数据被送到累加器的第七位。

SOD—串行数据输出端。SOD按照设置中断屏蔽指令SIM的规定置位或复位。

从上功能端的说明便可知,除了8080的所有特征之外,8085还包括内部时钟发生器,时钟输出,完全同步的READY,有史密特功能的RESET IN输入端,作为系统复位的Reset输出,总线控制讯号RDWR和IO/ $\overline{M}$,状态信息S₀, S₁,复用地址/数据总线,直接重新启动中断和直接重新启动非屏蔽中断以及串行输入/输出线。

下面简要介绍8085的中断功能,串行输入输出SID, SOD,系统定时,时钟驱动。等8085的主要特征。在8080的基础上了解这些新的特征对于8085的使用是有帮助的。

五、8085的中断功能:

8085的中断处理功能由于扩充了四个重新启动中断输入端而大大地加强了,在整个中断之中INTR的级别最低,它与8080的

INTR中断输入端功能完全相同,主要用于一般中断, $\overline{\text{INTA}}$ 是8085为INTR中断专设的应答信号,当CPU“开”中断,且INTR是唯一有效的中断时,CPU即发出与RD具有相同定时关系的 $\overline{\text{INTA}}$,利用 $\overline{\text{INTA}}$ 将外部电路提供的中断指令(如RST指令或CALL指令)通过数据总线读到指令寄存器IR,然后转入中断服务程序的入口地址。一个 $\overline{\text{INTA}}$ 讯号一次只能读进一个字节。所以中断指令有多少字节CPU就根据第一次读进的操作码决定发多少个INTA讯号。使用 $\overline{\text{INTA}}$ 中断需要附加外部电路以提供中断指令。为了方便地组成具有中断功能的小系统,8085新设了四级重新启动的向量中断输入(TRAP, RST7.5 RST6.5 RST5.5)其中RST7.5、RST6.5、RST5.5和INTR一样,受片内的中断允许触发器IEF/F的控制,IEF/F用开中断指令EI置位(“开”中断),用关中断指令DI复位(关中断)。此三个重新启动中断还受到片内各自的中断屏蔽触发器IM7.5, IM6.5, IM5.5的控制,这些屏蔽触发器可以通过新增加的指令SIM(设置中断屏蔽指令)的对应屏蔽码分别进行置位或复位。用以设置和解除相应的屏蔽状态。TRAP是不接受屏蔽的中断。同时也不受片内的中断允许触发器的控制,因而TRAP的中断请求总是挡不住的,无论片内中断允许和中断屏蔽的状态如何,也无论另外四个中断是否有请求,CPU总是优先响应TRAP中断的请求。TRAP、RST7.5、RST6.5、RST5.5都是重新启动中断,它们的特点是CPU在响应这些中断之后,不是等待外部提供中断指令,而是由CPU内部的编码电路自动地产生RST(重新启动)指令操作码(11100111)直接送至指令寄存器,执行重新启动指令功能,即将指令计数器内容压入堆栈,之后再修改栈顶指示器,在完成这些保护断点的工作之后,再自动地产生相应的向量地址送至地址总线,从而转入相应的服务程序的入口,而不是像INTR那样,先由CPU发出 $\overline{\text{INTA}}$ 中断响应讯号,

然后由外电路提供中断指令及服务程序的入口地址。当不只一个RST中断请求同时被采样时,8085片内的优先权排队电路会自动地按规定的优先级别排队,并自动地转入其中级别最高的中断请求所对应的中断处理程序的入口地址。四个RST中断所对应的中断处理程序的入口地址按如下规定:

TRAP——0024_H
RST7.5——003C_H
RST6.5——0034_H
RST5.5——002C_H

8085的中断是在每条指令结束前的倒数第二个机器状态进行采样的,任何中断一旦被CPU接受,即自动复位中断允许触发器,从而关闭中断。复位后的中断允许触发器只能由软件(EI指令)来重新置位,使CPU重新进入中断允许,即“开中断”状态。

INTR的输入是高电平有效,与8080的INT是一样的。RST5.5、RST6.5,也是高电平输入有效。RST7.5则设计为上跳变输入有效,RST7.5输入端的一个上跳变使内部的RST7.5的请求触发器置位,从而产生内部的中断请求,该触发器的状态要一直保持到该请求得到响应时为止,之后自动复位。RST7.5的输入请求触发器还可以用软件(即通过SIM指令的第四位)来复位。TRAP中断输入是上跳变加上电平触发输入有效,即输入在上跳变后还须保持高电平直到现行指令结束该中断得到响应之后才能撤消,所以短的尖脉冲干扰对它是不起作用的,抗干扰能力较强。TRAP被响应之后会自动复位内部的输入触发器。所以此时即使TRAP还保持在高电平,CPU也不会对它作第二次响应,TRAP再次响应只能在第一次输入变低,并再次出现第二次上跳变之后才有可能,从而避免了对同一事件作出多次响应。TRAP中断常用于象电源故障或总线出错等灾难性事件作紧急处理以保护断点等场合。应注意任何一中断一旦被响应,就会禁止以后出现的所有其它中断(TRAP中断除外)直

到EI指令执行。8085规定的中断优先级别顺序并不意味着较高级优先级别中断所启动的子程序也具有同样的优先级别,所以RST5.5能够中断由RST7.5所启动的子程序,只要在这条子程序结束前,中断重新开放。另外应说明的一点是,TRAP中断不受中断允许状态约束,但它会将原有的中断允许状态保存下来,并在TRAP中断完成后用一条RIM指令恢复CPU原来的中断允许状态,使原来的中断处理在TRAP完成之后就能继续进行,不受TRAP的影响。

六、系统的定时:

8085的任何一条指令的执行过程均由一系列的机器周期组成,每个机器周期(又称M)包括3~6个时钟周期(又称T状态)。每条指令最多不超过5个机器周期,每个机器周期执行一次读操作或写操作,每个机器周期的性质是由8085直接输出的三条状态线S₀、S₁、IO/M以及三条控制线RD、WR、INTA来决定的。8085机器周期按执行的功能划分,共有七种类型。详见表一

表一 8085机器周期表

机器周期	状态信息			控制讯号		
	Io/ $\overline{m}$	S ₁	S ₀	RD	WR	INTA
取操作码 (OF)	0	1	1	0	1	1
读存储器 (mR)	0	1	0	0	1	1
写存储器 (mw)	0	0	1	1	0	1
读I/O (IOR)	1	1	0	0	1	1
写I/O (IOW)	1	0	1	1	0	1
中断响应 (INA)	1	1	1	1	1	0
总线空闲 (BI)	0	1	0	1	1	1

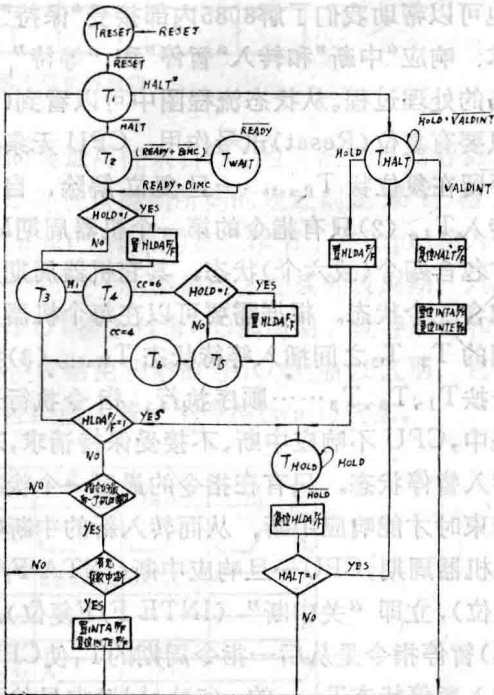
定义机器周期类型的状态信息Io/ $\overline{m}$ 、S₀、S₁在每个机器周期一开始即送出,并在整个周期机器时间内保持稳定。

指令的第一个机器周期(M₁)总是取操作码周期(OP周期)一般包含四个时钟周期,个别指令的M₁也包含6个时钟周期,除此之外的其它机器周期均只包含三个时钟周期。一条指令需要多少个机器周期以及周期的执行顺序均取决于M₁中所取出的

指令操作码,当然由于READY和HOLD线用可以置CPU为等待(WAIT)状态或保持(HOLD)状态,从而使指令规定的周期时间由于插入这些状态而加长。根据状态信息、控制讯号,系统总线所处的状态的不同,CPU在任一时刻只能处于如表二所示的十种状态之中的一个机器状态。

表二 8085机器状态表及说明

机器状态	状态信息及总线状态				控制信号		
	S ₁ , S ₀	Io/ $\overline{m}$	A8-15	A0-7	RD, WR	INTA	ALE
T ₁	X	X	X	X	1	1	1
T ₂	X	X	X	X	X	X	0
T _{WAIT}	X	X	X	X	X	X	0
T ₃	X	X	X	X	X	X	0
T ₄	1	0*	X	Ts	1	1	0
T ₅	1	0*	X	Ts	1	1	0
T ₆	1	0*	X	Ts	1	1	0
T _{Reset}	X	Ts	Ts	Ts	Ts	1	0
T _{HALT}	0	Ts	Ts	Ts	Ts	1	0
T _{HOLD}	X	Ts	Ts	Ts	Ts	1	0



图二 8085CPU状态流程图

注符: 号定义
(Tx) = CPU的状态。所有CPU的状态转换均发生在时钟的下降沿。

⊗ = 判定流程中一个可选择的路径。

[X] = 执行X功能

→ 顺序流程线

≠ 有条件的顺序流程线

CC = 现行机器周期T状态的数目

BIMC — “总线空闲机器周期”即不使用系统总线的机器周期。

VALIDINT = “有效中断” — 被允许而且没有被屏蔽（屏蔽仅对RST7.5、6.5、5.5的输入而言）的中断，即挂起来只等时间一到即被响应的中断。

HLDAF/F = 片内响应HOLD的触发器。8085的系统总线在HLDAF/F置位之后的一个时钟周期成为三态。

8085的任何指令均包含一系列的机器周期，这些机器周期总是由流程中的机器状态组成的，因而机器状态的流程就反映了指令的执行过程，机器状态流程图可以帮助我们清楚地理解CPU的机器状态的转换过程，也可以帮助我们了解8085内部接受“保持”请求，响应“中断”和转入“暂停”和“等待”状态的处理过程。从状态流程图中可以看到(1)只要有复位(Reset)讯号作用，CPU无条件停留在复位状态 T_{Reset} 。一旦复位解除，自动转入 T_1 。(2)只有指令的第一个机器周期 M_1 才包含四个(或六个)状态。其它机器周期只包含三个状态。根据需要可以在每个机器周期的 T_2 、 T_3 之间插入等待状态 T_{wait} 。(3)指令按 T_1 、 T_2 、 T_3 ……顺序执行，指令执行过程中，CPU不响应中断、不接受保持请求，不转入暂停状态。只有在指令的最后一个状态结束时才能响应中断，从而转入新的中断响应机器周期，CPU一旦响应中断(INTA F/F置位)，立即“关中断”(INTE F/F复位)。(4)暂停指令是从后一指令周期的 T_1 使CPU转入暂停状态 T_{HALT} 的。(5)hold请求虽然可以在指令的执行过程中被检测，但最终要在机器周期结束后才能进入保持状态 T_{HOLD} 。(6)在暂停状态 T_{HALT} 时可以随时响应中断而使CPU脱离暂停状态(HALT F/F被复位)

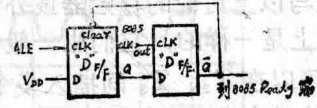
进入中断响应机器周期。在暂停状态时也可以接受hold保持请求暂时地转入保持状态，由于HALTF/F未被复位，所以保持结束后还会自动地返回暂停状态。

七、各机器周期定时的特点

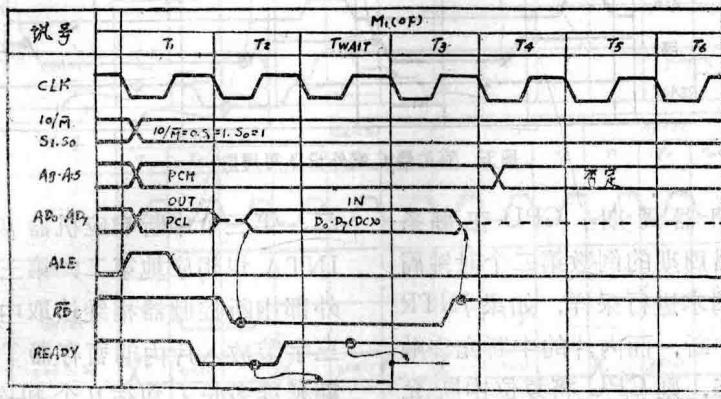
1. 取操作码机器周期

(1)在每条指令的第一个机器周期总是取操作码周期，它是唯一多于三个状态 T 的机器周期的，它主要工作是从存贮器中取操作码，如图这是在 T_3 结束前完成的，从 $T_1 \sim T_3$ 的过程与读存贮器机器周期完全相同。 T_4 或可能包括的 T_5 、 T_6 状态则用于指令的译码和部份单机器周期指令的执行，从图中可见，(2)地址 A_{8-15} 、 A_{0-7} ，以及状态信息输出 S_1 、 S_2 、 $IO/\overline{M}$ 从 T_1 开始就分别出现在各自的输出端，不过 S_1 、 S_0 、 $IO/\overline{M}$ 是在整个机器周期内保持有效，以提供稳定机器周期的状态，(3)高8位地址在整个读存贮器过程($T_1 \sim T_3$)中保持恒定，而低8位地址只是在 T_1 期间保持有效，在读信号 $\overline{RD}$ 出现之后，该 AD_{0-7} 总线则将作为接收存贮器读出指令码的数据总线，所以必须用ALE的后沿(出现在 T_1 内)将 A_{0-7} 锁存，以保证 $T_1 \sim T_3$ 地址稳定同时由于 AD_{0-7} 在 T_1 用于提供地址，只有等待 AD_{0-7} 变为数据总线之后，存贮器才可以执行读操作，所以8085的读写信号 $\overline{RD}$ 、 $\overline{WR}$ 是在 T_2 开始后发出。(4)同时注意到、 $\overline{RD}$ 、 $\overline{WR}$ 总是比 S_1 出现晚，因此 S_1 可用作R/ $\overline{W}$ 的先期信息，提供给外设接口，从而使总线传送的考虑更为方便。取操作码的过程在 T_3 内结束，因此从 T_4 开始地址总线 A_{8-15} 以及数据/地址总线 AD_{0-7} 不再用作访问。(5)8085的每一机器周期只安排 T_2 、 T_3 两个机器状态用于访问存贮器或外设。若存贮器或外设的响应速度较慢，可以利用Ready讯号在 T_2 和 T_3 之间插入若干个 T_{wait} (等待)状态，使之适于多类存取速度的存贮器和外设的工作。利用 $IO/\overline{M}$ 状态线来控制Ready信号，可以对存贮器或外设等待状态的产生与

否进行选择。具有等待状态的取操作码机器周期如图三，顺便提一下，Ready (准备)线的另一个经常性的用途是用手动开关使8085执行单步操作，即在每个机器周期中执行到 T_2 时利用Ready使之停在 T_{wait} 状态或待需要继续执行。如图电路可用于在每个8085机器周期中插入一个等待状态。两个“D”F/F均是上升边触发，Clear清零端是低电平有效的。Ready“低”电平脉冲是在 T_2 时建立的，使之后的一个状态为等待状态 T_{wait} 。



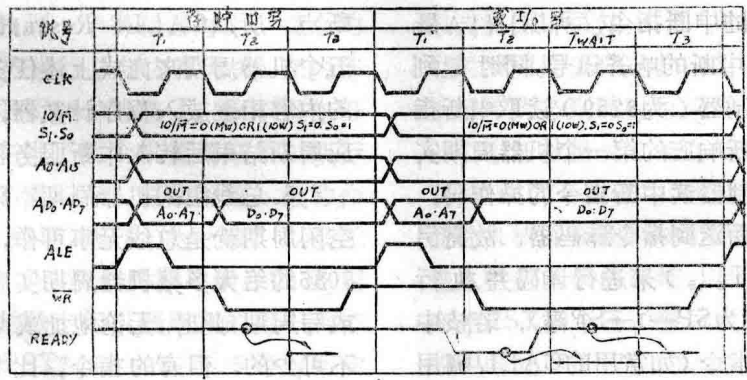
取操作码机器周期的 T_1 总是指令计数器内容送地址总线， T_2 执行 $PC+1$ ，而 T_3 完成取操作码即操作码送指令寄存器， T_4 进行指令译码。特殊指令包含的 T_5 、 T_6 主要用作CPU内部的双字节数据交换和处理。



图三 具有一个等待状态的取操作码机器周期

2. 存贮器或外设读机器周期：这两种类型的机器周期的时序关系与取操作码周期相同，不同的是它们只包括三个T状态，而且 S_0 、 S_1 、 $IO/\overline{M}$ 的状态信息内容与取操作码周期不一样（见前机器周期状态表），同时，取操作码周期在 T_1 时送出的存贮器地址总是来自指令计数器PC，而存贮器读周

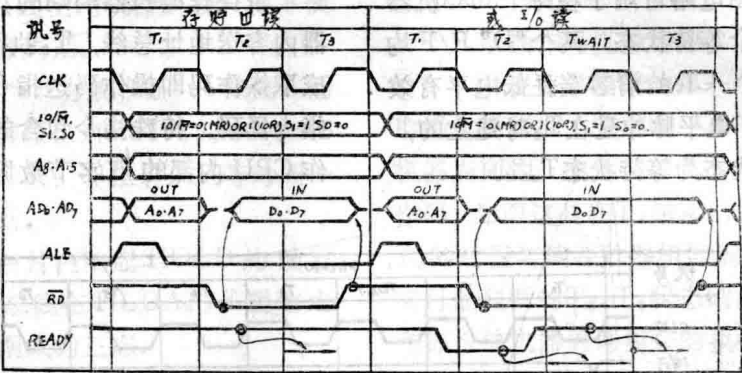
期的地址可能来自PC，也可能来自寄存器，I/O 外设读周期所需的地址则是来自输入（input）指令的第二字节，在input输入指令的第三个机器周期才是读外设周期。只有INPUT指令包含I/O读周期。读存贮器和读外设周期由状态信号 $IO/\overline{M}$ 加以区别，它们都可以插入若干个等待(wait)状态。



图四 存贮器（或外设）读周期

3. 写存储器或写外设周期：存储器或外设的写入与以上讨论的存储器或外设读机器周期基本上是一样的，它们也一般包括三个T状态，可以在T₂、T₃间插入多个Wait状态，不同的是在发RD时改发WD，从T₂开始

AD₀~AD₇由CPU驱动，以提供写入存储器或外设的数据。只有out（输出）指令包含写外设机器周期，写外设所需的地址同样由out指令的第二字节提供。

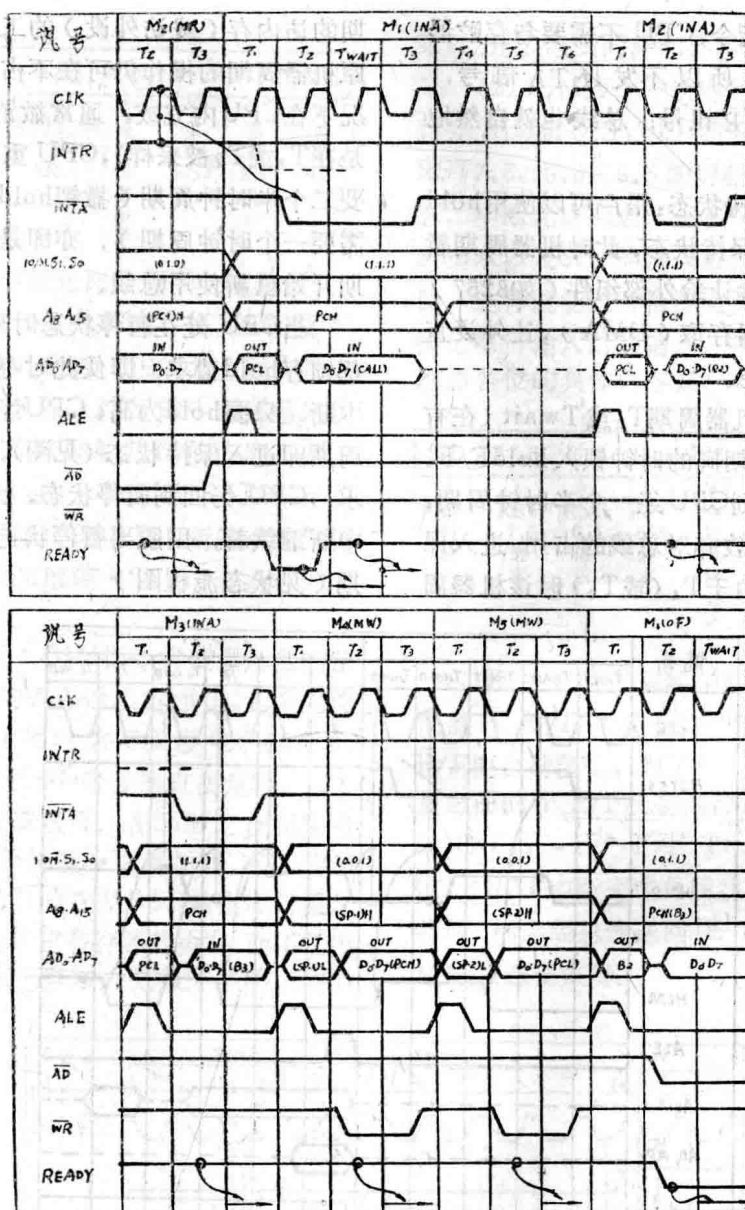


图五 存储器（或外设）写周期

4. 中断响应机器周期：CPU在每条指令最后一个机器周期的倒数第二个时钟周期对所有的中断请求进行采样，如果INTR作为唯一有效的中断，而内片的中断允许触发器处于置位状态，则CPU将复位中断允许触发器（即内部关中断）而进入如图的中断响应机器周期（INA），在INA周期，RD、WR均为高电平，虽然程序计数器的内容在T₁时就送到了地址总线，但并不起任何作用，因为无论存储器或外设此时都不能进行读/写操作。但是CPU却发出了一个低电平的中断应答信号INTA以代替RD，INTA具有RD相同的定时关系，它被送往外部的中断控制逻辑电路，表示CPU的数据总线可以接受外部送入的中断指令，所以INTA既可作为CPU接受中断的应答信号，同时起到从外部的中断控制器（为8259）读取中断指令的作用。在中断响应的第一个机器周期实质上是从中断控制器读中断指令的操作码，它经过数据总线而送到指令译码器，故需另外的状态（T₄或T₅）来进行译码并执行中断的其它操作（为SP-1→SP等），若该中断指令是多字节指令（如常用的CALL调用指令、Restart重新启动指令），则有第

二、第三个中断响应机器周期、中断响应INTA也相应地第二、第三次地发出用以从外部中断控制器相继读取中断指令的第二、三字节放入片内的暂存器TMP，中断响应周期延续多长，包括几个机器周期则由M₁读取的中断指令的操作码决定。在取得完整的中断指令以后，CPU将执行中断指令，为适应不同用途的中断，8085内部逻辑已设计成可以允许任何指令用于中断，但是CALL和Restart是最合理的选择，因为它们具有在转到新的调用子程序之前强制CPU将PC的内容（注意在整个取中断指令的过程中程序计数器已被禁止加1）推入堆栈的功能，使之在中断处理子程序结束后能准确地返回断点，所以CALL、Restart还要求第四、第五个机器周期来完成上述任务，寄存器WZ的内容相继放入程序计数器，以便在中断响应周期结束后转入中断服务程序的入口，

5. 总线空闲机器周期：顾名思义，总线空闲周期就是总线无事可作，处于闲置状态。8085的绝大多数机器周期实质上不外都是读或写周期，此时，无论地址或数据总线都是必不可少的。但有的指令，比如DAD_{RP}（寄存器对相加）指令，是CPU内的两对寄存器的



图六 中断响应机器周期（中断响应后为 CALL 调用指令）

内容相加结果放到一个寄存器对中，指令的执行是在CPU内进行，与内存无关，在内部完成这一功能需要10个状态T，除取操作码的4个，8085再安排了两个机器周期用以执行指令，由于它不需访存，总线闲置，故称作总线空闲机器周期。除RD为“高”以及无ALE讯号之外，这两个周期与读存贮器周期相同。出现总线空闲机器周期的另一种情况是TRAP、RST7.5、6.5、5.5等重新启动中断、请求被接受时发生的。从8085的状

态流程图中可知，CPU只能在指令的最后一个机器周期采样这些中断申请，当CPU处于暂停状态则随时可采样，这些中断只要有一个中断被接受（条件是内部的中断允许触发器IE F/F置位“1”即处于“开中断”状态，而且未被屏蔽注TRAP不能屏蔽），则CPU将进入特定的机器周期，以便在CPU内部自动产生RST指令码，随后自动地转入各自固定的中断处理子程序的入口地址。这些中断并不像一般中断INTR一样需要从外部中断控

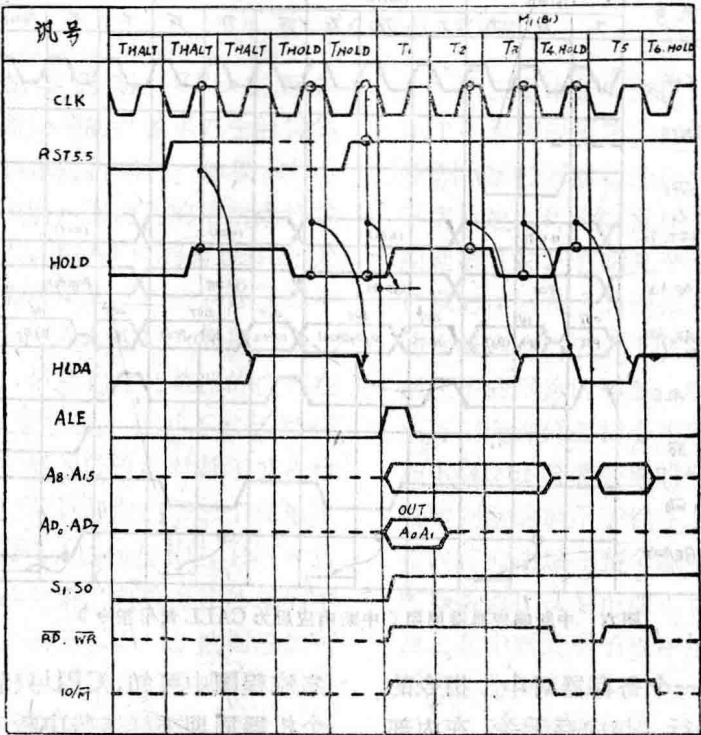
制电路接收中断指令,CPU 不需要与存储器等外电路打交道,所以不发 $\overline{INTA}$ 信号,也不发出 $\overline{RD}$ 、 $\overline{WR}$ 讯号,总线也就自然地处于闲置状态。

6. 8085的保持状态:用户可以使用hold输入使CPU进入保持状态,此时机器周期暂时停止执行,总线让给外部组件(如8257)去执行直接存储器存取(DMA),让外设直接与内存交换信息。

通常hold由机器周期 T_2 或 T_{wait} (在有6T的 M_1 时是 T_4)期间的时钟锁入holdF/F,若HOLD为“高”则CPU经一个半时钟周期,即在 T_4 (或 T_6)时放弃对总线的占用,进入保持状态(见图)。由于 T_4 (或 T_6)时该机器周

期的访内存(或访外设)的工作已经结束,原机器周期的操作仍可在不占用总线的情况下在CPU内完成。通常撤销hold请求(也是在 T_2 或 T_4 被采样),CPU重新获得总线需要二个半时钟周期(撤销hold到hlda变“低”需要一个时钟周期),亦即是从下一机器周期开始重新使用总线。

当CPU处在暂停状态时刻在每一时钟都锁存hold请求,即使此时存在一个有效的中断,只要hold为高,CPU经过一个半时钟周期即进入保持状态(见图)。撤销hold请求,CPU仍回到暂停状态。此时若有效的中断继续着,则脱离暂停状进入中断响应周期(见状态流程图)



图七 保持状态的建立

八、SIM、RIM指令及串行输出SOD, 输入SID

8085包含8080的全部指令,并且为了适应增加的功能,8085正式增加了两条指令,目的是用软件对新增的三级RST中断7.5、6.5、5.5,进行控制和管理,同时也对新增

的一位串行输出SOD,输入SID进行控制。
(1)SIM指令:设置中断屏蔽指令是一个包含四状态的单机器周期指令,执行SIM指令时累加器的部份内容用来设置RST7.5、6.5、5.5中断的屏蔽状态,和实施控制。累加器的另外两位则用于控制SOD的输出。累加器各位的具体内容是:

7	6	5	4	3	2	1	0
SOD	SOE	X	R7.5	MSE	M7.5	M6.5	M5.5
↑ 串行 输出 数据	↑ 允许 串行 输出	↑ 不定	↑ RST7.5 复位	↑ 允许 设置 屏蔽	↑ 屏蔽 RST 7.5	↑ 屏蔽 RST 6.5	↑ 屏蔽 RST 5.5

如果设置屏蔽允许位 MSE 为“1”，则累加器的第 2—0 位的内容将分别置位（内容为 1 时）或复位（内容为 0 时）片内 RST7.5、6.5、5.5 各自对应的中断屏蔽触发器，MSE 为“0”原屏蔽触发器的内容不变。

第四位 R7.5 为 1，则不管 RST7.5 是否屏蔽，将对 RST7.5 的内部中断请求触发器复位。即用软件来解除 RST7.5 在内部的中断请求。

若第六位为 1，SIM 指令将使累加器的第 7 位送到 SOD 的寄存器然后输出，若第六位为 0，SOD 寄存器内容不受影响。8085 的 RESCT IN 可使 SOD 寄存器直接复零。

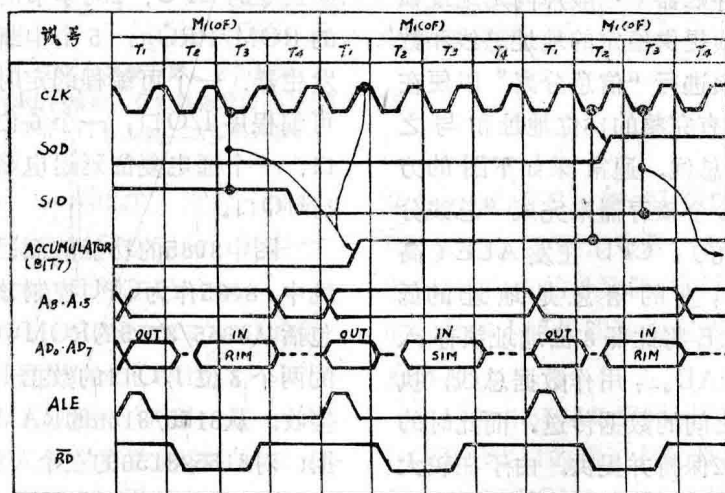
显然若要设置蔽屏，在 SIM 之前还需有一条传送指令，将设置位送累加器。有了 SIM 指令还可利用软件直接从 SCD 输出串行累加器内容或 CPU 内其它寄存器的内容，而不需要其它接口，这在一些场合是很方便的。同时

第 6 位还能起到同步和异步的控制作用。

(2) RIM 指令—读中断屏蔽指令。在执行完 RIM 指令后，CPU 将有关中断的信息：RST7.5、6.5、5.5 的屏蔽中断触发器的状态，中断允许标志，正在等待响应的 RST 中断请求以及 SID 的输入数据分别送入累加器。这样就能随时检测有关中断的情况，并执行串行输入口的输入，执行 RIM 指令后累加器各位的具体内容是：

7	6	5	4	3	2	1	0
SID	I7.5	I6.5	I5.5	IE*	M7.5	M6.5	M5.5
↑ 串行 输入 数据	↑ RST 7.5 请求	↑ RST 6.5 请求	↑ RST 5.5 请求	↑ 中断 允许 标志	↑ RST 7.5 屏蔽 F/F 状态	↑ RST 6.5 屏蔽 F/F 状态	↑ RST 5.5 屏蔽 F/F 状态

IE*—TRAP 陷阱中断不受中断允许触发器 IE F/F 的控制，在响应 TRAP 后 IE F/F 要自动复位，为了在 TRAP 执行后能恢复之前的 IE F/F 状态，内部设置了 IE* 标志，平时 IE F/F = IE* F/F，TRAP 时 IE* 不被复位，TRAP 中断完成之后执行一条 RIM，以便将 IE* 标志经总线送回 IE F/F 和累加器，原中断状态被恢复。



图八 讯号 SID、SOD 与指令 RIM、SIM 的关系

8085 组成系统的一些特点：

在以 8085 为 CPU 的微型计算机系统中，8085 要协调所有总线上信息的传送，并对这

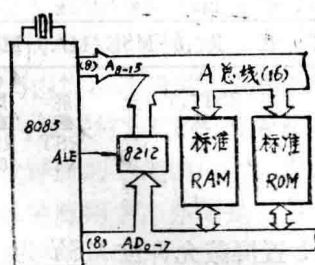
些信息进行处理，以执行“指令”。8085 作为中央处理单元要产生信号告诉外部器件在总线上的是什么样的信息以及从何时开始进

行操作。

8085既继承了8080的全部软件资源,也继承了它的硬件资源,使8085组成系统有了丰富的遗产。同时为了推广微型机使用,还设计了几种专用的外围部件,这些部件把存贮器(RAM、ROM、EPROM), I/O接口, 定时器等的功能结合而组成在同一芯片上,并使它们能直接与8085的复用总线相连接。这样,对大多数系统而言,8085就可以组成需要外部器件的最少的系统。这些系统小型、灵活、廉价,易于开发和使用。下面将提供一个用8085以及两片采用外围电路8355/8755和8156组成的三片系统的例子。

在组成系统方面,8085能共享8080系列已有的丰富的外围电路接口资源,除8257—5DMA控制器之外,8085能直接享用这些接口。由于8080的这些外围接口都是可编程序的,所以根据程序都有一定的自我管理能力的,从而减轻了系统软件在分时和对I/O进行登记方面的负担。应说明的一点是,如果用户的使用频率在2mc以上,则应选用825X—5的器件以保证其频率性能。

8085由于复用总线,为了与标准存贮器,特别是静态存贮器(一般片内无地址锁存器)接口,必须提供稳定的地址总线 and 数据总线。为此必须进行“信息分离”以便在存贮器读/写时具有完整的16位地址和与分离的8位数据总线。通常采如下图的方式,利用8212的8个锁存器来完成“总线分离”任务(见图九)。CPU在发ALE(高电平)时,AD₀₋₇上的信息是地址的低8位,故可用ALE将此低8位地址锁存入8212。ALE之后AD₀₋₇用作数据总线,执行存贮器和8085之间的数据传送。而此时的低8位地址由8212保持并提供。由于在较大的内存容量的系统中,往往需要用8212提高总线驱动能力,所以对于这些系统8212的“信息分离”作用是顺便得到的,并不使系统的附加器件数增加。对于I/O口,由于IN、OUT指令提供的8位I/O口地址同时从地址



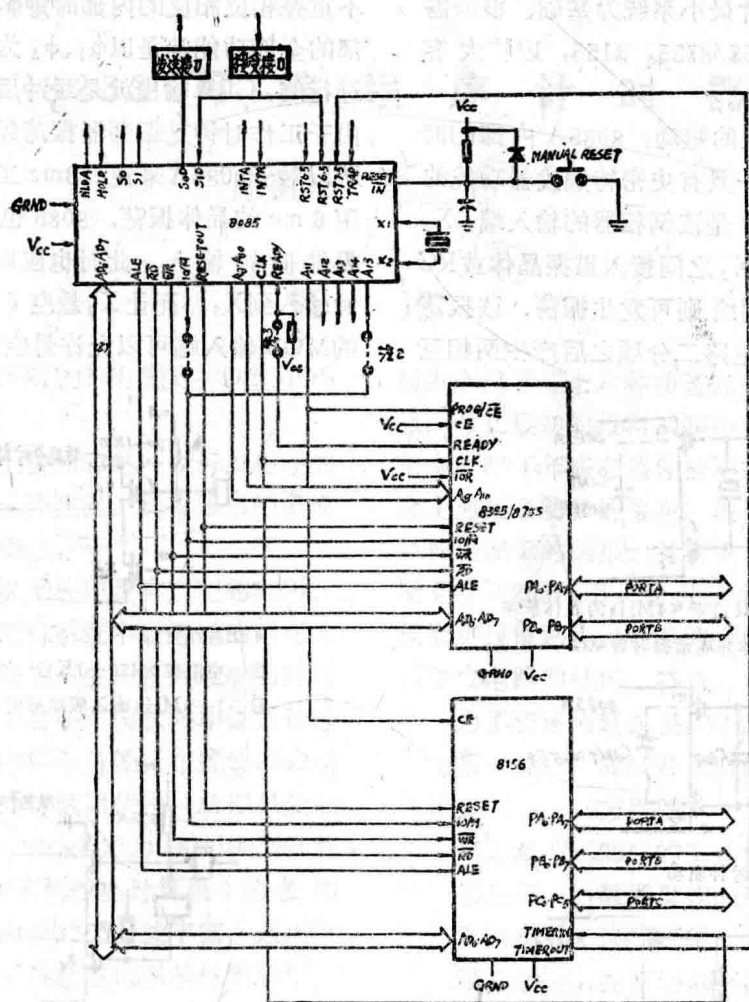
图九

的高8位和低8位总线上输出,所以不存在以上的“信息分离”问题。

8085的最小系统:

由于芯片设计的改进,集成度的提高,intel公司推出了由三片大规模电路组成的最小系统(如图)除8085作中央处理单元外,还包括专门为8085系统设计的两只专用存贮器接口芯片,8155/8156、8355/8755、intel还提供了该系统印刷电路板,该系统除三片LSI之外,只须一组5V电源,一个晶体、一个电容、一个二极管、二只电阻。在实际应用中,应根据实际适当连接所需的控制讯号(如INTR,HOLD、TRAP等),也可根据系统规模扩充存贮容量和I/O口的数目。但就此最小系统而言,已包含具有74条指令的CPU,256字节的RAM,2K字节的ROM/PRON,5个中断输入,一个时钟发生器,一个可编程的定时器,四个8位的可编程序I/O口,一个6位的可编程序I/O口,一个通电复位延迟电路,和一个一位串行I/O口。

图中8085的功能前面已经介绍,在该系统中,8085作为CPU控制整个系统的动作,包括从8355/8755的ROM中读数,8355/8755的两个8位I/O口的数据I/O控制,发送和接收。从8155/8156的RAM中读数或写入数据,对8155/8156的三个可编程序I/O口进行程序控制,以及对8155/8156的14位计数器/定时器进行程序控制。此外,8085本身提供了一个程序控制的一位串行输入和输出口,必要时还可通过HOLD、TRAP、INTR等进行DMA和中断控制。X₁、X₂的外接晶



图十 8085 最小系统结构图

- 注1 TRAP INTK HOLD如果不用,必须接地。
 注2 用IO/M对标准I/O编址,存储器编址的I/O则采用A15。
 注3 仅当要求一个TWAIT状态时,才需要连接。

体以产生内部时钟,外接电阻、电容、网络以提供电源接通时的暂时复位讯号,使系统延迟一段时间之后才启动相应的功能。

图中可见8155/8156以及8355/8755均能直接与8085的复用数据地址总线接口,因为它们的内部均设置了地址锁存器,在ALE的后沿地址被锁存以提供存储器及I/O口的稳定的地址信息。根据IO/M的极性,此地址或作存储器的地址,或作I/O口的地址,在RD、WD的控制下、AD₀₋₇的8位数据可以写/读8155/8156的RAM或I/O口,(8155 CE“低”电平有效,8156是CE“高”电平有效。)

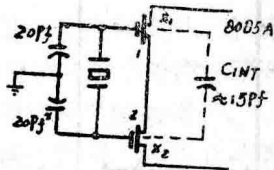
两个8位I/O是一般的I/O口,6位的I/O既可作I/O口,又可作为两个8位口的状态标志寄存器。14位的计数器/定时器接在编所设置的计数终点可按四种规定的模式提供一方波或脉冲。

8355和8755是管脚相容的带有五个8位I/O通用接口的唯读存储器,8755是紫外光擦写的EPROM,8355是固定ROM,ROM容量均为2K字节,往往在固定程序调试过程中采用8755,待系统调试完毕后,再用硬化了的8355直接取代8755,既加快了系统试制周期,又可降低大量生产后的成本。

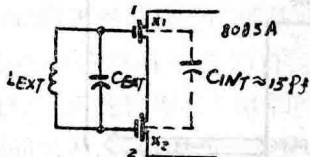
以上面的三片最小系统为基础、根据需
要可以再增添 8355/8755, 8155, 以扩大容
量, 增加接口。

X_1 、 X_2 输入的驱动: 8085A 内部的时
钟发生器是内部一具有史密特触发器功能的
倒相器充任的, X_1 是该倒相器的输入端, X_2
是输出端, 在 X_1 、 X_2 之间接入谐振晶体或 RC
网络或 LC 谐振网络 则可发生振荡, 该振荡
器的输出经内部电路二分频之后产生两相互

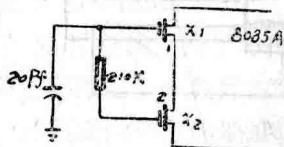
不重叠相位相反的内部时钟 ϕ_1 、 ϕ_2 、8085A 内
部的全部功能就是以 ϕ_1 、 ϕ_2 为基本定时节拍
进行的, CLK 输出就是缓冲后的内部时钟。
由于工作时钟是振荡器振荡频率的二分频,
所以对于 8085A 来说, 3mc 工作频率要求采
用 6 mc 的晶体振荡。8085 也可由外面直接
提供时钟输入, 此时也应以两倍频率的时
钟输入到 X_1 , 而让 X_2 悬空 (X_2 不同于一般
的 MOS 输入端可以允许悬空)。以下是 8085



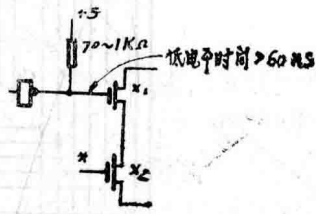
• 20 Pf 电容是只适于 $\leq 4\text{MHz}$ 的晶体频率
A. 石英晶体并联谐振时钟驱动



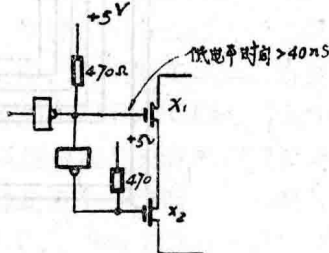
B. LC 谐振时钟驱动



C. RC 时钟驱动



• 让其浮空
注: 电阻应为 $470 \sim 1\text{K}\Omega$
D. 1—6MHz 输入频率外时钟驱动电路



E. 1—10MHz 输入频率外时钟驱动电路

图十一 8085 X_1 X_2 的几种主要驱动方式

X_1 、 X_2 的几种主要驱动方式及其特点:

8085 的驱动频率至少应高于 1mc。晶体
并联谐振 (图 A) 时要求 C_L (负载电容) $\leq$
30Pf、 C_s (分流电容) $\leq 7\text{Pf}$, R_s (等效分
流电阻) ≤ 75 欧姆, 驱动电平 10mw, 频率误
差: $\pm 0.005\%$, L_C 并联谐振 (图 B) 的频率
偏差接近于 $\pm 10\%$, 其元件按

$$f = \frac{1}{2\pi\sqrt{L(C_{ext} + C_{int})}}$$

为减小频
率偏离建议 C_{ext} 至少应选为 C_{int} 的两倍。当
频率高到约 5MC 时最好不选用 LC 调谐时钟
驱动。

当频率的精度并不重要时, 可用图 C 的

RC 回路来产生得 8085A 所需的时钟。当使
用 RC 方式时, 片内定时发生器的性能的偏
差会使频率产生大范围的偏离, 此法的优点
在于元件价格低。图 C 电路产生约 3mc 驱动
频率, 当驱动频率明显地高于或低于这一频
率时, 建议不用 RC 振荡方法。

图 D 示出外时钟输入驱动电路, 接入上
接电阻是保证输入高电平不低于 $4V$ 。如果需
用 6~10MHz 频率脉冲来驱动 8085A, 可用图
E 的推挽源来驱动 X_1 和 X_2 以改善时钟发生
器的稳定性, 此时为了防止 8085 产生自激振
荡, 应设法使 X_2 不会经过驱动电路偶合回
 X_1 。