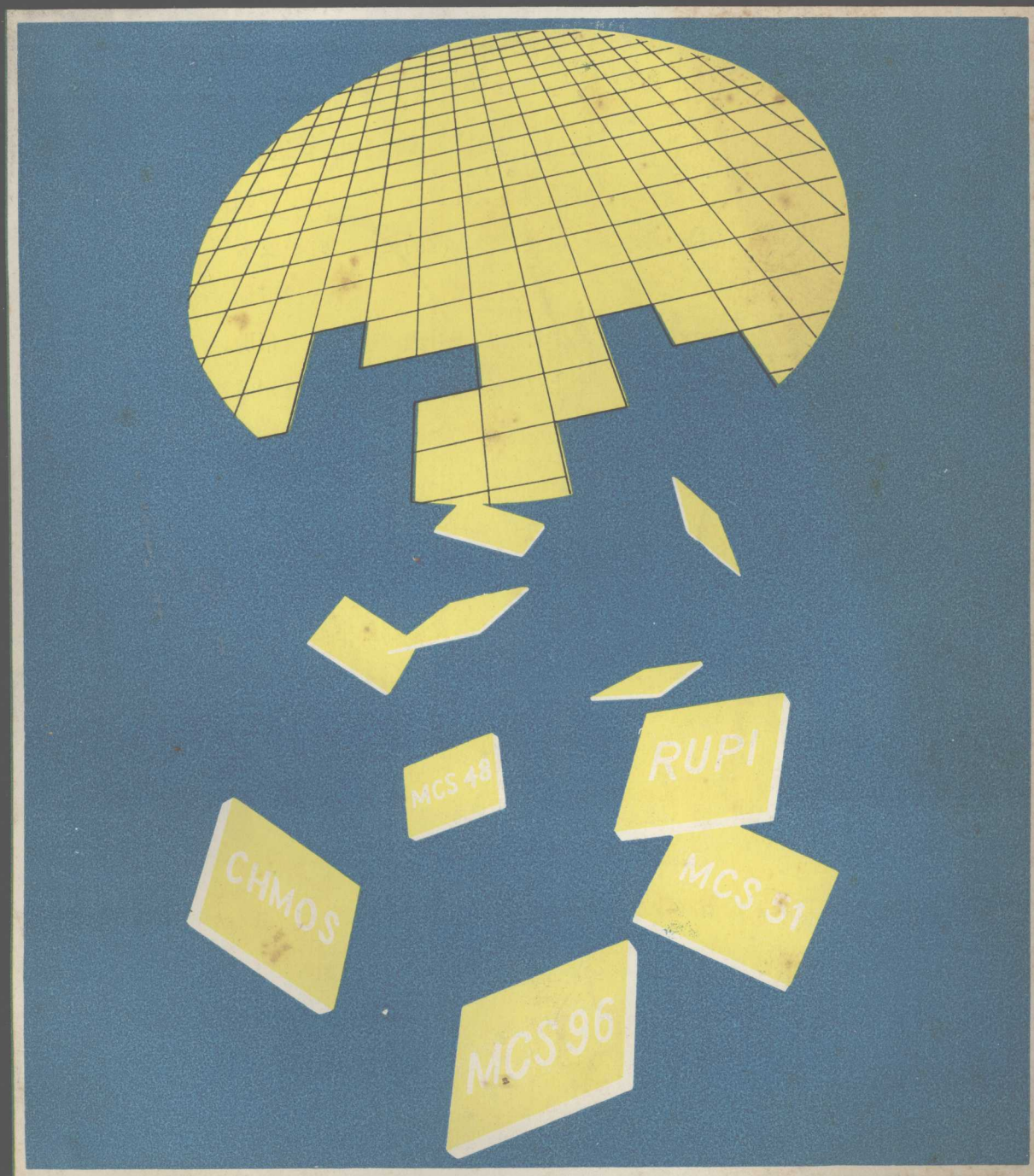


Intel 微型控制器硬件手册

MCS 48 MCS 51 MCS 96 ■ 技术设计说明书



机械部北京机械工业自动化所

目 录

ISC-96系列

- 第 一 章 MCS-96简介.....(1-1)
- 第 二 章 结构概述.....(2-1)
- 第 三 章 MCS-96软件设计资料.....(3-1)
- 第 四 章 硬件设计资料.....(4-1)
- 第 五 章 MCS-96数据表.....(5-1)

MCS-51系列

- 第 六 章 MCS-51结构.....(6-1)
- 第 七 章 MCS-51存储器结构, 寻址方式和布尔处理器.....(7-1)
- 第 八 章 MCS-51指令系统.....(8-1)
- 第 九 章 MCS-51应用实例.....(9-1)
- 第 十 章 CMOS微型控制器.....(10-1)
- 第十一章 单片8位微型计算机.....(11-1)
 - 8031AH/8051AH
 - 8032AH/8052AH
 - 8751H

CMOS

- 第十二章 使用CHMOS的设计注意事项.....(12-1)
 - CMOS内部工艺

第一章 MCS-96简介

1.0继续发展微控制器

英特尔公司自1976年推出世界标准的8048 (MCS-40) 微控制器以来, 继续推进单片微控制器的发展。1980年英特尔公司又推出性能水平大大高于8048的8051 (MCS-51)。随着8051的出现, 微控制器应用突飞猛进。从键盘和终端一直到控制汽车发动机广阔的应用领域中都使用这种通用的芯片。8051迅速地巩固了第二代世界标准微控制器的地位。

由于半导体工艺技术正被推向一个新的极限值, 因此在单个硅片上集成十万个以上的晶体管已经不成为问题。英特尔公司的微控制器充分发挥今天的工艺技术成就并精心开发了称之为MCS-96的新一代单片微控制器。8096 (MCS-96的通用型号) 具有单片微控制器所达到的最高水平系统集成度。它使用十二万个以上的晶体管, 实现高性能16位CPU, 8K字节的程序存贮, 以及模拟和数字两种类型的I/O部件。图1-1为英特尔公司单片微控制器的发展。

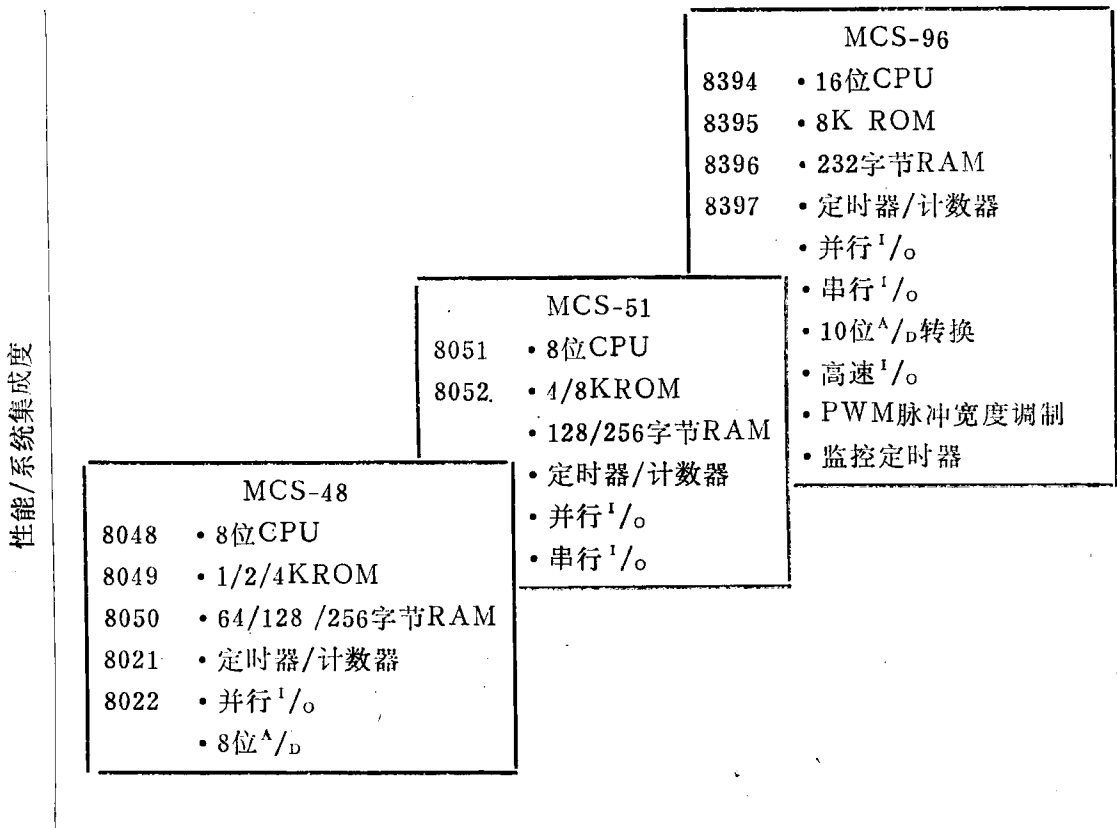


图1-1 英特尔公司微控制器的发展

1.1MCS-96简介

8096是由功效很强的16位CPU组成的，它紧密地与程序和数据存储器以及一些I/O部件在一起，并且所有这些都集成在单一硅片上。CPU支持位、字节和字操作，对于一部份指令系统还支持32位的双倍字长操作。若输入频率为12MHZ，8096执行16位加法的时间为1.0μs，16×16位的乘法或32/16位的除法时间为6.5μs。

8096提供4个高速触发脉冲输入，能以2MS的分辨率（在12MHZ晶体振荡频率之下）记录外部事件发生的时间，还提供最多可达6个的高速脉冲发生器输出，可在预定的时间启动外部事件。高速输出单元能同时执行定时器功能，除了二个16位硬件定时器以外，还有最多可达4个这样的16位软件定时器能同时操作。

片上附加的一个A/D转换器最多可将4个（48腿型）或8个（64腿型）模拟量输入通道转换为10位数字值。除此以外，片上还提供一个串行口，一个监控定时器以及一个脉宽调制输出信号。表1-1为MCS-96的特性和优点的摘要。

在单一硅片上带16位CPU和所有I/O部件以及接口资源的8096代表了当今微控制器系统集成度的最高水平。它将开辟在过去要用好几个总片才能实现的许多新的应用。

表1-1 MCS-96特性以及优点的摘要

特 性	优 点
16位CPU 8K字节ROM 232字节RAM 硬件乘/除	具有较高吞吐量的高效机器 对较复杂较大的程序有较大的程序空间 较大的片上寄存器堆 提供优良的数学运算能力，在12MHz时16×16位乘法或32/16位除法执行时间为6.5μs
6种寻址方式 高速I/O单元 4条专用I/O线 4条可编程I/O线	提供编程和数据处理方面较大的灵活性 能以高分辨率（在12MHz时为2μs）测量和产生脉冲信号
10位A/D转换器 全双工串行口 最多可达40个I/O口	读出外部模拟量输入 提供到其他处理器或系统的异步串行连接 提供与TTL兼容的数字数据I/O包括以标准的8式16位外国芯片进行系统扩展
8种可编程中断源的优先 中断系统 脉宽调制输出 监控定时器	响应异步事件 提供可变空度的可编程脉冲序列，也可作用产生模拟量输出提供恢复软件出错或硬件失常的能力
48腿（双列直插封装）与48腿（扁平封装，四面出腿）型	提供各种封装形式，以便选择I/O的个数以及封装尺寸，更好地满足特定的应用需要

1.2 MCS-96的应用

MCS-96产品是独立的高性能单片微控制器，可以用于诸如工业控制、测量以及计算机智能外设这样一些要求高度实时性的应用上。它的应用面涉及几乎所有工业领域（见表1.2）。由于具有16位CPU高速数学运算和高速I/O的能力使得8096很适合作复杂的发动机控

制和轴控制系统。例如三相、大马力交流电机和机器人。

由于可选用带有10位A/D转换器的芯片，因此可用于数据采集系统和闭环模拟量控制器，模拟量和数字I/O处理结合在一起，使单一芯片达到相当大的系统集成度。

这一芯片完全适用于象气相色谱仪这样一些代表产品的领域，这些产品将模拟量处理与高速数字压碾（number crunching）结合在一起。这些特性还使它成为航天应用，象导弹制导和控制方面理想的文件。

表1-2 MCS-96广泛的应用领域

工业

电机控制

机器人

离散和连续过程控制

数控

智能传感器

仪表

医疗仪器

液相和气相色谱

示波器

家用电器

电视录象机

激光唱片驱动器

High-end电视游戏机

制导和控制

导弹控制

鱼雷制导控制

智能化武器

航天制导系统

数据处理

绘画机

彩色和黑白复印机

温氏磁盘驱动器

磁带驱动器

撞击和非撞击打印机

远程通讯

调制解调器

智能化线路标牌控制

汽车

点火控制

变速控制

防滑制动

1.3 MCS-96系列开发支持工具

MCS-96系列产品是由一系列英特尔公司的软件和硬件开发工具所支持的。这些工具大大地缩短了产品的开发周期,因此使产品更快地进入市场。

1.3.1 MCS-96软件开发包

8096软件开发包提供专用于MCS-96单片微控制器系列的开发系统支持。这一程序包是由符号宏汇编程序ASM-96,连接重定位程序RL-96以及子程序库LIB-96所组成的。在高级语言中提供连同浮点数学运算程序包的PLM-96。对于MCS-96产品系列另外一些高级语言也正在开发中。

1.3.2 ASM-96宏汇编

8096的宏汇编程序将符号汇编语言指令翻译为机器可执行的目标码。ASM-96使程序员能以模块方式编写程序。这些模块程序将一个比较复杂的程序分成比较小的功能单元,以便易于编码,调试和修改。紧接着这些单独的模块能用RL-96实用程序连接起来,并且定位成为一个程序模块。这一实用程序将选择的输入目标模块合并成一个单一输出的目标模块,同时还 将存储器分配给输入段,并将可重定位地址代入绝对地址。然后它产生一个由连接一览表、符号表以及中间交叉访问表所组成的打印文件。另一个实用程序LIB-96帮助产生,修改和检查程序库文件。ASM-96运行在Intellec系列Ⅲ或Ⅳ。

1.3.3 PL/M-96

PL/M-96编译程序将PL/M-96语言翻译为8096可重定位的目标模块,因此大大地提高了程序员的效率以及应用的可靠性。这一高级语言已经有效地被设计成反映到机器的结构中,以致不必再费心考虑在较高的程序员效率和不经济的代码之间进行取舍。由于这一语言及其编译程序对于8096以及它的应用环境都是最优化的,因此以PL/M-96来开发软件是很少冒什么风险的方案。

1.3.4 硬件开发支持

iSBE-96是MCS-96产品的硬件执行和调试工具。它由驻留在8096系统中的监控程序/调试程序所组成的。这一开发系统通过二条带状电缆与用户的8096系统接口,其中之一是对于8096I/O口的;而另一条是对于存储总线的。iSBE-96通过一条串行连接线受Intellec系列Ⅲ或其他的计算机系统控制。iSBE-96的电源是将它插入MULTIBUS插件座供给的,也可以由外部电源供给。iSBE-96包含在一块标准的MULTIBUS板上。

iSBE-96提供实时硬件仿真最常用的部件。用户能显示,修改存储单元,设定断点,执行带时不带断点的程序以及修改存储器分配。除此之外用户也能单步地执行系统程序。

1.3.5 MCS-96讲习班

讲习班向设计工程师和系统设计师提供MCS-96系列产品丰富的操作经验。教程包括英特尔公司8096的结构、系统定时、输入/输出设计等。讲习期间会使参加者大大地提高对于MCS-96产品系列以及支持工具深入的了解。

1.3.6 Insite 程序库

英特尔公司Insite程序库包含一些应用程序,其中一个很有用的程序是SIM-96,8096的软件模拟程序。模拟程序具有设定断点,检查和修改存储器,及汇编目标码和单步执行代码的能力。

1.4MCS-96系列产品

虽则8096是MCS-96产品在本手册中最常用的一般元件号，然而这一产品系列却是由8种形式组成的，并具有包括8096在内的8个元件号。之所以要提供如此广泛的种类，其原因是为了能更好地在I/O个数，封装尺寸这样一些项目上满足用户的应用要求。注意选择的项目包括板上的8K字节掩膜编程存储器、十位A/D转换器以及48或68腿封装类型。

表1-3扼要摘录了所有在MCS-96产品系列现有的产品。

表1-3 MCS-96产品系列

任 选 项		68腿	48腿
数字 ¹ /。	不带 ROM	8096	8094
	带 ROM	8396	8394
模拟量和数字 ¹ /。	不带 ROM	8097	8095
	带 ROM	8397	8395

48腿型是双列直插式封装

68腿型有两种封装形式：塑料扁平 and 四面出腿式封装。

第二章 结构概述

2.0引言

为了说明8096的操作可将其分成若干部份：CPU、可编程高速I/O单元、模数转换器、串行口以及作为数模转换的脉宽调制（PWM）输出。除了这些功能单元之外，还有支持片上全部操作的一些部份，如象时钟发生器以及反馈偏压发生器。CPU以及可编程I/O使得8096大大不同于任何其他的微控制器，让我们首先来分析CPU。

2.1CPU操作

8096上CPU的主要元件是寄存器堆和RALU。同外部世界的通讯是通过特殊功能寄存器（SFR）或存贮控制器。RALU（寄存器/算术逻辑单元）不用累加器，它直接在寄存器堆和SFR所组成的256字节的寄存器空间上进行操作。直接通过SFR控制I/O来进行有效的I/O操作是可能的，其主要好处是能很快地修改前后关系，不存在累加器阻塞以及很高的吞吐量和I/O次数。

2.1.1CPU总线

一个控制单元和二条总线连接寄存器堆和RALU。图2-1为CPU和它的主要总线连线。二条总线是8位的A总线和16位的D总线。D总线仅仅只能在RALU和寄存器堆或特殊功能寄存器（SER）之间传送数据，A总线用作上述传送的地址总线或连到存贮控制器的多重切换地址/数据总线。内部ROM或外存贮器的任何访问都是通过存贮控制器完成的。

存贮控制器内部是一个从属程序计数器（从属PC），它将PC的踪迹保持在CPU内。由于大多数都是根据从属PC从存贮器取指令因此节省了处理器的时间，因为很少需要将地址发送到存贮控制器。若地址跳出顺序则从属PC装入一个新值并且继续进行处理。从存贮器取数也是通过存贮控制器完成的，然而对于这一操作从属PC被旁路。（图2-1见下页）

2.1.2CPU寄存器堆

寄存器堆包含一个能以字节、字式双倍字进行访问的232字节RAM。由于RALU能使用这些单元的每一字节，因此基本上等于有232个累加器。寄存器堆中的第一个字是留作堆栈指示器用的。因此当发生堆栈操作时不能用作数据。访问寄存器堆和SFR的地址是由CPU硬件暂时存贮在8位的地址寄存器内。

2.1.3RALU控制

RALU的命令是取自A总线并且暂时存贮在指令寄存器。控制单元将指令译码并产生正确的信号序列使RALU执行要求的功能。图2-1为指令寄存器和控制单元。

2.1.4RALU

由8096执行的极大多数计算都发生在RALU。图2-2所示的RALU包含一个17位的ALU、程序状态字（PSW）、程序计数器（PC）、循环计数器以及三个暂存寄存器。所有这些寄存器都是16位或17位（16位+符号位）的。有些寄存器具有抛开ALU而执行简单操作的能力。

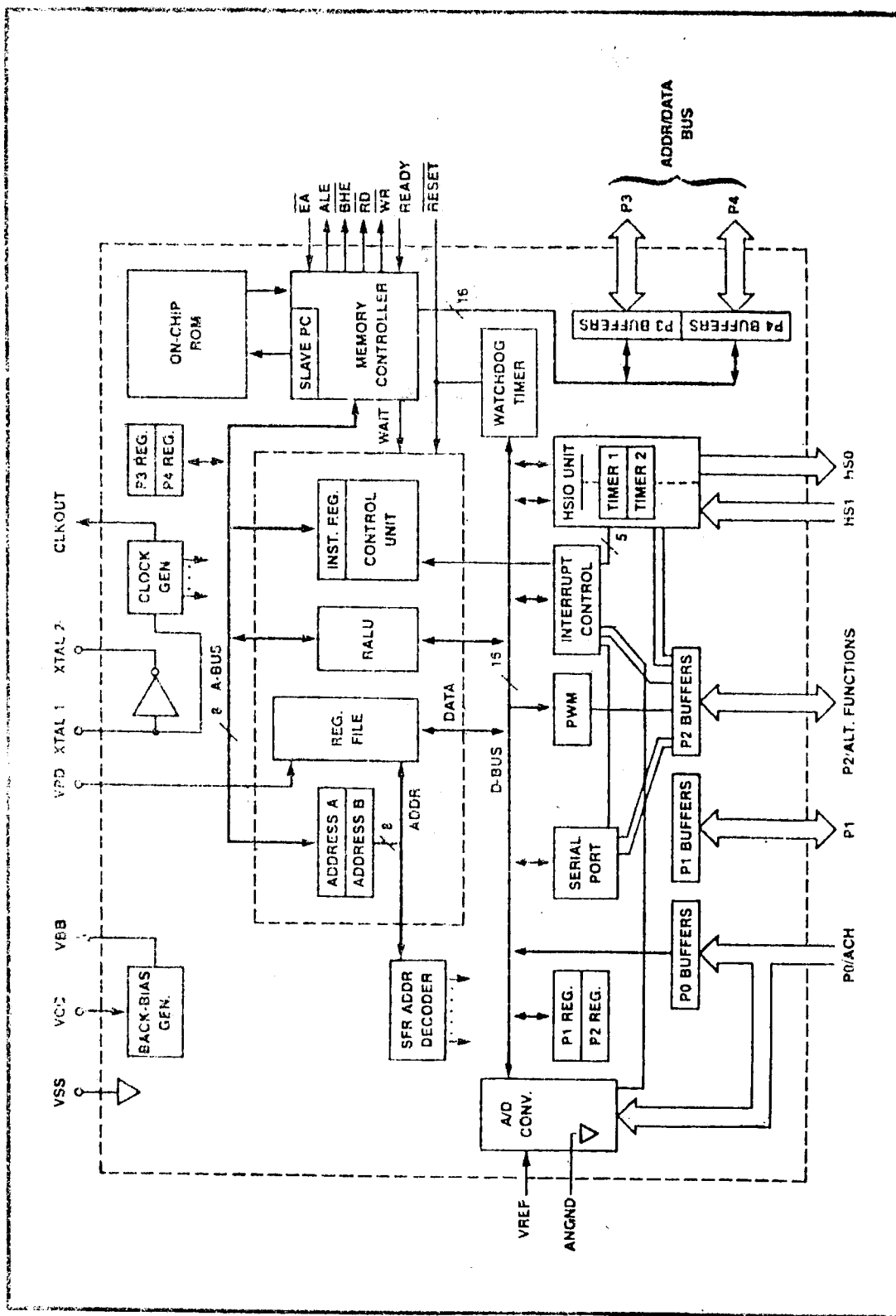


图2-1 框图 (为了简单起见, 出入口寄存器到出入口缓冲器的连线未予表示)

一个单独的增量寄存器被用于PC，然而转移必须通过ALU。暂存寄存器中，其中二个有它们自己的移位逻辑。这些寄存器用于包括规格化、乘法和除法这样一些需要移位逻辑的操作中。下字寄存器仅仅当进行双字长移位时才用到；而上字寄存器每当执行移位时都要用到，或者在许多指令中被用作一个暂存寄存器。

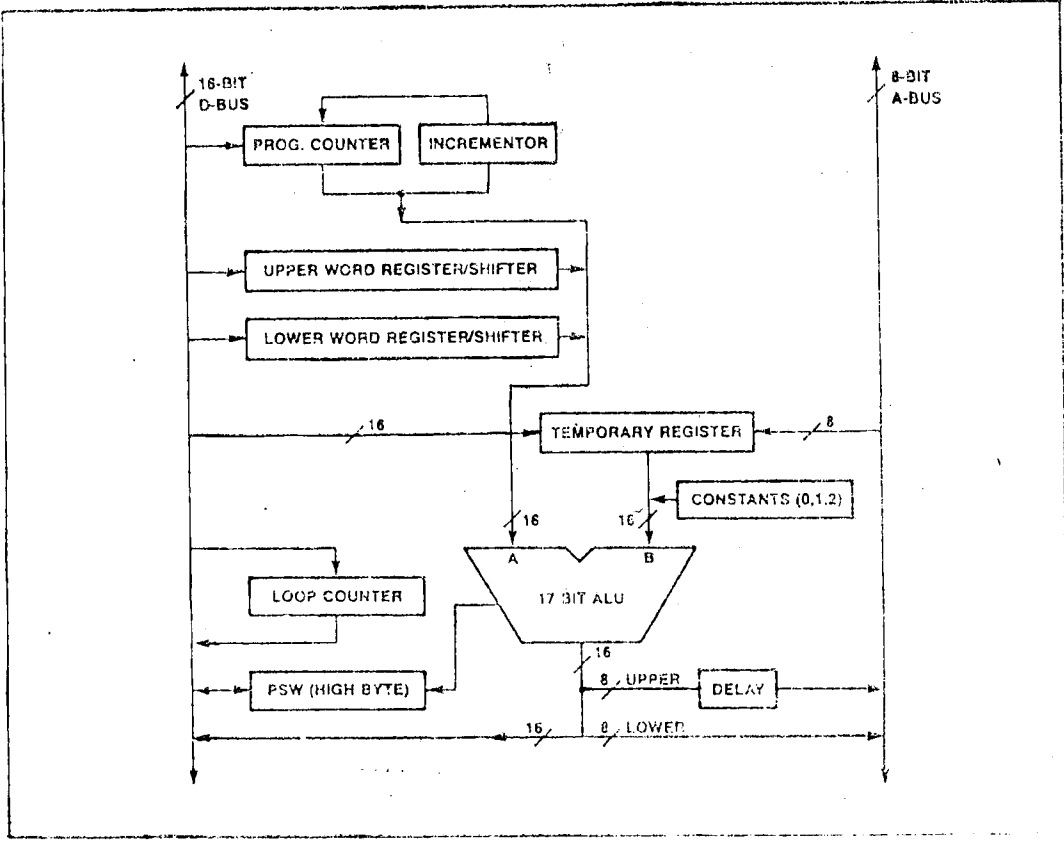


图2-2 RALU框图

一个暂存寄存器用来存储双操作数指令的第二个操作数，包括乘法操作的乘数和除法操作时的除数。为了执行减法这个寄存器输出能在放进ALU的B输入端前进行取补。

示于图2-2的延时电路是用来转换16位的总线成为8位总线。当所有地址和指令在8位A总线上传送时就需要这个。象0、1和2这样一些常数存放在RALU中，以便用来加速某些计算。这些常数当RALU需要作2的补码或执行增量或减量指令时是迟早有用的。

2.2基本定时

8096进行工作需要一个6.0MHE到12MHE之间的输入时钟频率。这个频率能直接加到XTAL1。另一方面，由于XTAL1和XTAL2是一个反相器的输入和输出，因此也可以用一个晶体来产生时钟。振荡器部份的框图如图2-3所示。有关它的用法的电路和建议的是在细节4.1节中。（见下页2-3图）

2.2.1内部定时

晶体式外部振荡频率3分频后产生如图2-4所示的三个内部定时节拍。各个内部节拍每隔三个振荡周期重复出现。3个振荡周期称作为一个状态时间，它是8096操作的基本时间量度单位。极大多数内部操作是与占空度为33%的节拍A、B或C同步的。节拍A在外部是以

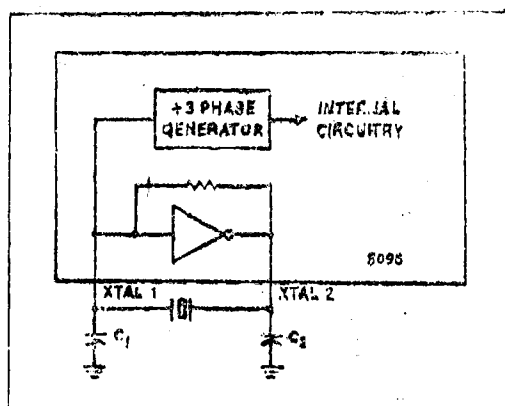


图2-3 振荡器框图

CLKOVT表示的，在68腿之作中是一个可供利用的信号。节拍B和节拍C外部是不能用的。XTAL1，CLKOUT和节拍A、B和C的关系示于图2-4。注意：在框图中并没有把传输的延迟考虑进去。有关这些以及其他定时关系的细节是在4.1，4.4和4.6节中。

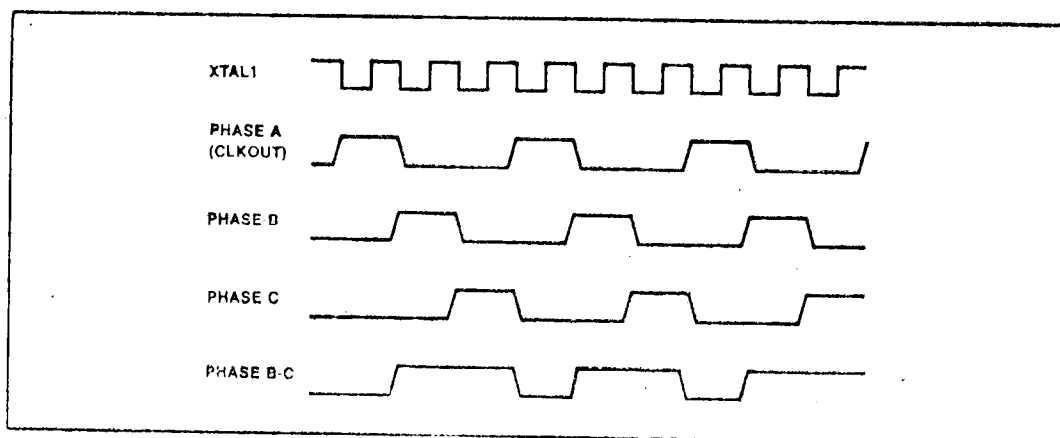


图2-4 相对于XTAL1的内部定时

RESET线可以用来在精确的时间下启动8096提供测试设备与多片系统的同步。这一特性的用法在2.15节和4.1节RESET下面给予更充分的解释。

2.3 存储空间

8096中可寻址的存储空间由64K字节组成。它们中间绝大多数可供用户用作存储程序或数据。具有特殊用途的单元是0000H到00FFH和1FFE H到2010H。所有其他单元能用于存储程序或数据，亦可用于按存储器分配地址的外部设备。存储器映象图示于图2-5。（见下页2-5图）

2.3.1 寄存器堆

00H到0FFH单元包含寄存器堆和SFR。存储空间这一部份的全部信息能在“RAM空间”这一节中找到。在内部RAM区中不能执行代码。若在00H到0FFH单元中应各执行指令，则指令将取自外存储器。这一部份外存储器留给英特尔公司的开发工具使用。执行一个非屏蔽中断（NMI）将强制调用外部0000H单元，因此NMI指令也被留给英特尔公司的开发工具。

2.3.2 保留的存储空间

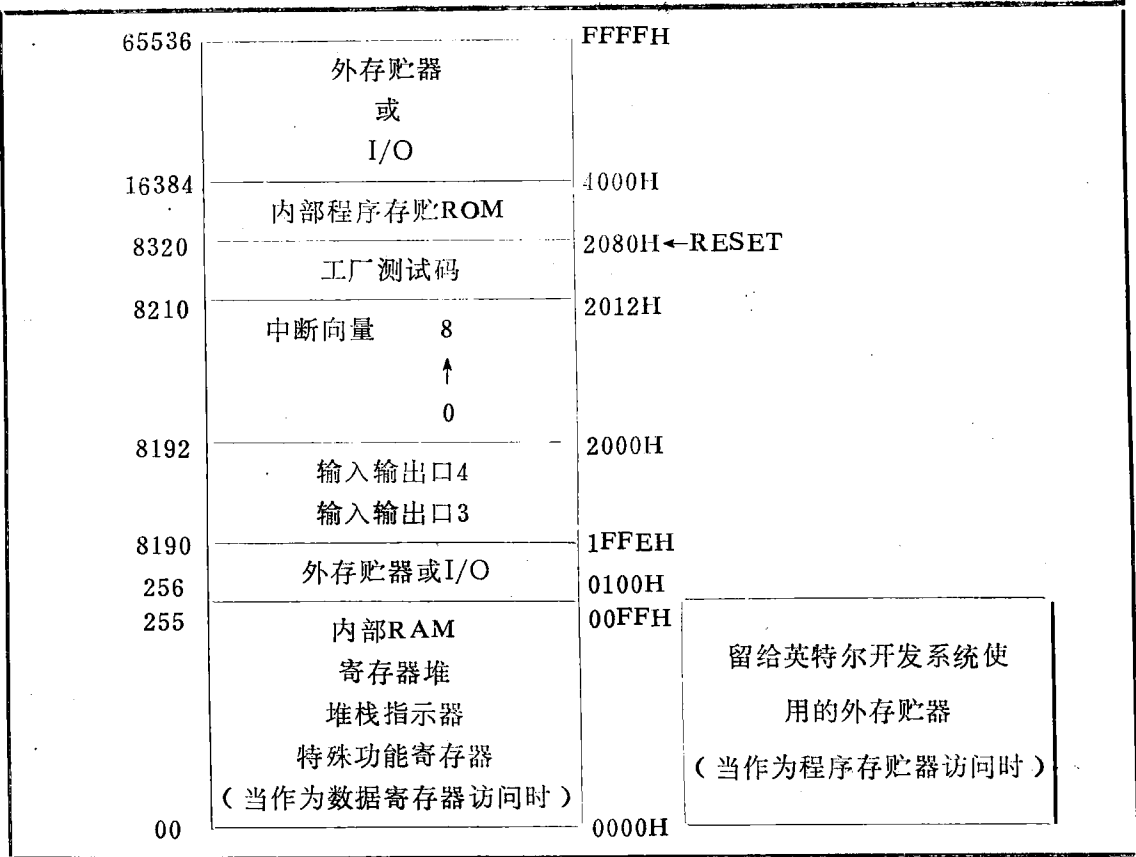


图2-5 存贮器映象图

1FFE H和1FFF H单元是分别留给3和4出入口的。这是为了允许容易地重新构造这些I/O口。若系统中使用外存贮器，在4.6.7节给出了重新构造出入口的例子。若3和4出入口并不打算要重新构造，则这些单元可以看作任何其他外部的存贮单元。

9个中断向量是存放在2000H到2011H。第9个向量按2.5节所说明的那样是由英特尔公司开发系统使用的。内部的2012H后2077H单元是留给英特尔公司的工厂测试码。这些单元在外存贮器中是可以使用的。

8096的复位使要取的指令自2080H单元开始。选择这一单元 是为了允许系统具有与寄存器堆邻接的最多可达8K的RAM。有关复位进一步的资料在2.15节。

2.3.3内部ROM

当一个ROM元件被订货时，与2000H到2011H单元是中断向量一样，内部存贮单元2080H到3FFFH是用户指定的。仅当这元件有ROM，EA处于 高并且 地 址是在2000H 和 3FFFH之间，指令和 数据才 取自于ROM，而在所有其他 时间，以 内部RAM空间或外存贮器 访问数据，指令取自于外存贮器。

2.3.4存贮控制器

RALU经存贮控制器与存贮器（除了在寄存器堆和SFR空间以外）通讯。存贮控制器通过A总线和几条控制线连接到RALU。由于A总线是8位的，因此存贮控制器使用一个从属程序计数器，避免经常要从RALU得到指令单元。从属PC在每次取指令后增量。当发生转移或调用时，从属程序计数器必须在能继续取指令以前从A总线装入新值。

除了包含一个从属PC以外，存贮控制器还包含一个三字节排队寄存器，以帮助提高执行的速度。除非在外部总线周期被强制于等待状态，否则这个排队寄存器对RALU和用户是透明的。示于3-3和3-4的指令执行时间为没有附加等待状态的正常执行时间。重新装入从属PC并且取新指令流的第一字节要花费4个状态时间，这反映在示于表3-4中的转移发生或不发生的时间。

2.3.5系统总线

外存贮器通过构成一条16位多重切换的（地址/数据）数据总线AD0到AD15线进行寻址。这些线与I/O口3和I/O口4共用连接腿。为了使这一总线分路，地址门锁使能（ALE）线的下降沿用来形成一个给透明的门锁电路（74LS373）的时钟。典型的电路和需要的定时示于4.6节。由于8096的外存贮器能按字节或字进行寻址，所以要用二条线来控制译码，总线高使能（ $\overline{\text{BHE}}$ ）和地址/数据线0（AD0）。 $\overline{\text{BHE}}$ 线显而易见必须被锁定，就象地址线一样。

在解释存贮系统时，为了避免混淆，有必要对分路的地址/数据信号给以命名。地址信号就称之为MA0到MD15（存贮器地址），而数据信号则称之为MD0到MD15（存贮器数据）。

当 $\overline{\text{BHE}}$ 为有效低时，应选择存贮器连接到数据总线的高字节，而当MA0为低时则应选择存贮器连到数据总线的低字节。以这种方法，访问16位存贮器能仅为低字节（偶）（MA0=0, $\overline{\text{BHE}}=1$ ），仅为高字节（奇）（MA0=1, $\overline{\text{BHE}}=0$ ）或为双字节（MA0=0, $\overline{\text{BHE}}=0$ ）。另一个存贮区正在读出，则 $\overline{\text{BHE}}$ 和MA0就不需要进行译码。

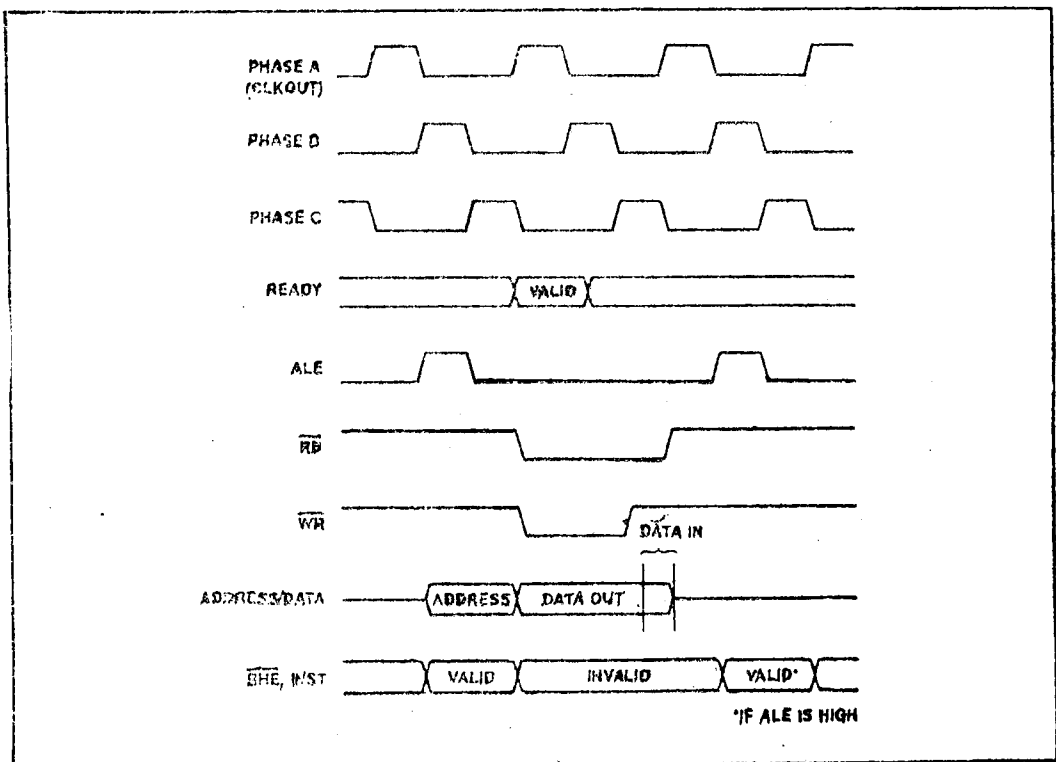
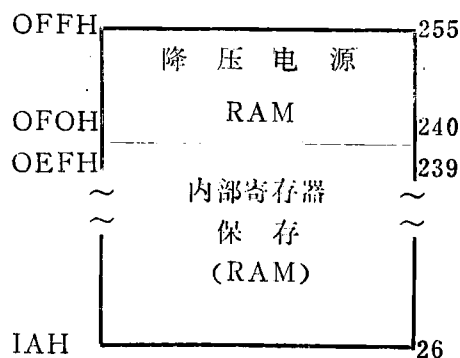


图2-6 外存贮器定时

图2-6为与下列说明外存贮器操作有关的理想化波形图。对于精确的定时说明请参阅最



19H	
18H	堆 栈 指 针
17H	
16H	IOS1
15H	IOS0
14H	
13H	储 备
12H	
11H	SP-STAT
10H	IO口2
0FH	IO口1
0EH	IO口0
0DH	计时器2 (HI)
0CH	计时器2 (LO)
0BH	计时器1 (HI)
0AH	计时器1 (LO)
09H	INT-PENDING
08H	INT-MASK
07H	SBUF (RX)
06H	HSI-STATUS
05H	HSH-TIME (HI)
04H	HSI-TIMF (LO)
03H	AD-结果 (HI)
02H	AD-结果 (LO)
01H	R0 (HI)
00H	R0 (LO)

当读时

25	
24	堆 栈 指 针
23	PWM-控制
22	IOC1
21	IOC0
20	
19	储 备
18	
17	SP-CON
16	IO口2
15	IO口1
14	波德一率
13	
12	储 备
11	
10	WATCHDOG
9	INT-PENDING
8	INT-MASK
7	SBVF (TX)
6	HSO-COMMAND
5	HSO-TIME (HI)
4	HSO-TIMF (LO)
3	HSI-MODE
2	AD-COMMND
1	R0 (HI)
0	R0 (LO)

当写时

图2-7 寄存器外存图

后的数据表。当开始取补存贮器时，地址门锁使能（ALE）线就上升，地址就置于AD0—AD15上并且 $\overline{\text{BHE}}$ 被设定所需要的状态。然后ALE下降，地址从这些腿移走，并且 $\overline{\text{RD}}$ （读出）信号趋于低，READY线则被拉低，以保持处理器在几个额外的状态时间条件下。

2.3.6就绪

READY线能用来保持处理器在上述条件，以便允许访问低速存贮器或为了DMA目的。当产生CLKOUT信号的节拍2时，在内部发生READY线的采样。在CLKOUT趋于低以前，READY必须在一段最小时间内保持低状态。若这元件正趋向于非就绪状态时，这个设定的时间不符合要求，则这元件就不可能操作。

虽则READY线能在任何时间带到高，但是由于READY是与CLKOUT同步的，所以8096在几倍于CLKOUT的周期内处在非就绪条件下。保持8096在非就绪状态有一个最大的时间限制，典型的大约为1 μs 。对于特殊的元件和要求的温度范围内，精确的时间关系在数据表里有所说明。

来自外存贮器的数据必须在总线上，并且在 $\overline{\text{RD}}$ 上升沿以前在所指定的最小设定时间之内是稳定的。 $\overline{\text{RD}}$ 的上升沿锁定进入8096的信息。若读出的是数据，则在地址有效时INST腿是低的。若读出的是指令，则INST腿在此期间为高的。在48腿的元件是没有INST腿的。

写到外存贮器需要类似于自外存贮器中读出所需要的定时，主要的不是用到的是字信号（ $\overline{\text{WR}}$ ）而不是 $\overline{\text{RD}}$ 信号。直到 $\overline{\text{WR}}$ 线的下降沿定时都是一样的。在这一时间8096去掉地址，并将数据置于总线上。在那时READY线必须如同上述那样保持在要求的状态下。当 $\overline{\text{WR}}$ 线趋于高时，数据就应被锁定在外存贮器。在字期间，INST始终是低的，因为指令是不能被写入的，对于存贮器访问的精确定时在数据表内有所说明。

2.4RAM空间

8096中的内部寄存器单元分成二组：一个寄存器堆和一组特殊功能寄存器。RALU能与256个内部寄存器单元中的任何一个进行操作。00H到17H单元是用作访问特殊功能寄存器的。18H和19H单元包含一个堆栈指示器。对于其余230个单元的用法除了不能由此执行代码以外就没有什么区别。

2.4.1特殊功能寄存器

所有的8096的I/O是由SFR控制的。这些寄存器中的一部份有两种功能：即它们被读出时功能和它们被写入时功能。图2-7为这些寄存器的单元和名字。有关每个寄存器的功能的摘要以及给出完整的说明的各章节号示于图2-8。

寄 存 器	说 明	章 节
RO	零寄存器—始终读出0可用于变化时的巷数以及计算和比较时的常数	3.2.7
AD-RESULT	A/D结果高/低—A/D转换器的低部和高部结果	2.9.3
AD-COMMAND	A/D命令寄存器—控制A/D转换	2.9.2
HSI-MODE	HSI方式寄存器—设定高速输入单元方式	2.7.1
HSI-TIME	HSI时间高/低—包含高速输入单元触发的时间	2.7.4
HSO-TIME	HSO时间高/低—设定高速输出的时间以便执行命令寄存器的命令	2.8.3
HSO-COMMAND	HSO命令寄存器—决定装入HsO时间寄存器的时间将	2.8.2

发些什么		
HSI-STATU	HSI状态寄存器—指出在HSI时间寄存器的时间检测哪一条腿	2.7.4
SBUF (RX)	串行口的接收缓冲器：包含要输出的内容	2.11
SBUF (TX)	串行口的发送缓冲器：包含刚从串行口接收的字节	2.11
INT-MASK	中断屏蔽寄存器—使能或禁止各别的中断	2.5.2, 3.6.2
INT-PENDING	中断未决寄存器—指示某一中断源已经发生中断信号	2.5.2, 3.6.2
WAT0HDOG	监控定时寄存器—周期性的写入，以便抑制每64K状态时间的自动复位	2.14
TIMER1	定时器1高/低—定时器1的高字节和低字节	2.6.1, 2.7.8
TIMER2	定时器2高/低—定时器2的高字节和低字节	2.6.2, 2.7.8
IOPORT0	输入输出0寄存器—输入输出0腿上的电平	2.12.1
BAUD-RATE	包含波特率的寄存器，该寄存器是顺序装入的	2.11.4
IOPORT1	输入输出1寄存器—用于输入输出1的读出或写入	2.12.2
IOPORT2	输入输出2寄存器—用于输入输出2的读出或写入	2.12.3
SP-STAT	串行口状态—指示串行口的状态	2.11.3
SP-CON	串行口控制—用来制定串行口的方式	2.11.1
IOSO	I/O状态寄存器—包含HSO状态信息	2.13.4
IOSI	I/O状态寄存器1—包含定时器和HSI的状态信息	2.13.5
IOCO	I/O控制寄存器0—控制HSI出腿的不同功能，定时；2的复位源和定时器2的时钟源	2.13.2
IOCI	I/O控制寄存器1—控制输入输出2的不同功能，定时器中断和HSI中断	2.13.3
LPWM-CONTRO	脉宽调制控制寄存器—设定PWM脉冲的持续时间	2.10, 4.3.2

在SFR空间内是一些标作保留的寄存器。这些寄存器是留给将来扩展或测试用的，这些寄存器的读写可能产生不希望的结果。例如：写到000CH单元将要定二个定时器为0FFXH，这个特性是用来测试这一元件的，在程序中不应使用。

2.4.2电源下降

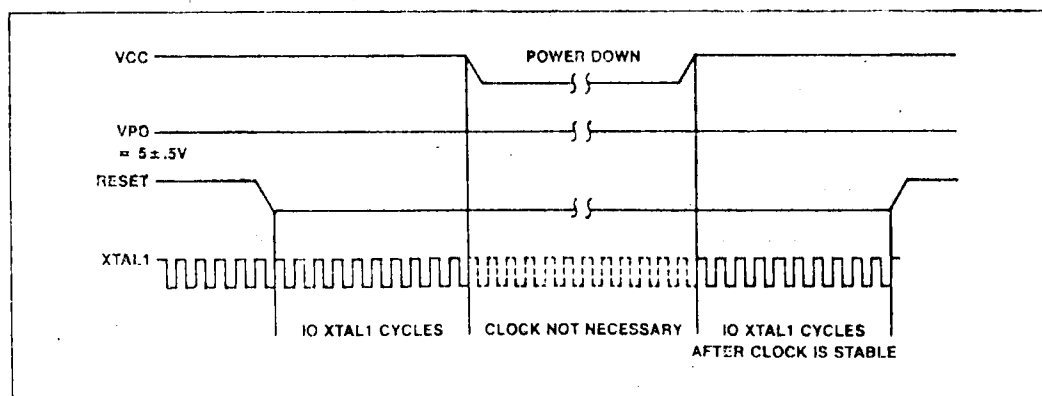


图2-9 电源下降定时