

单片微控制机手册之一

8096

单片微控制机



力试验所

6-62

《舰船导航》编辑部

一九八五年五月

目 录

第一章 MCS-96的绪言	(1)
1.0 微控制机的发展	(1)
1.1 MCS-96的绪言	(1)
1.2 MCS-96的应用	(2)
1.3 MCS-96系列开发工具	(3)
1.4 MCS-96系列产品	(5)
第二章 体系结构概述	(6)
2.0 前言	(6)
2.1 CPU的操作	(6)
2.2 基本定时	(8)
2.3 存贮空间	(9)
2.4 RAM空间	(12)
2.5 中断结构	(13)
2.6 定时器	(16)
2.7 高速输入	(17)
2.8 高速输出	(19)
2.9 模拟输入	(20)
2.10 脉冲宽度调制输出(D/A)	(22)
2.11 串行端口	(22)
2.12 I/O端口0、1、2、3和4	(25)
2.13 状态和控制寄存器	(26)
2.14 监视定时器	(28)
2.15 复位	(29)
2.16 引脚说明	(30)
2.17 引脚表	(32)
第三章 MCS-96软件设计说明	(34)
3.0 前言	(34)

3.1	操作数类型	(34)
3.2	操作数寻址	(36)
3.3	程序状态字	(38)
3.4	指令系统	(40)
3.5	软件标准和约定	(46)
3.6	中断系统的使用	(47)
3.7	I/O程序设计研究	(50)
3.8	例1 串行I/O通道的程序设计	(53)
3.9	例2 用HSO部件产生PWM	(53)
3.10	例3 用HSI部件测量脉冲	(53)
3.11	例4 A/D通道扫描	(53)
3.12	例5 表检查和插入	(53)
3.13	指令系统详述	(53)
第四章	硬件设计资料	(98)
4.0	硬件接口概述	(98)
4.1	硬件连接要求	(98)
4.2	驱动和接口电平	(102)
4.3	模拟接口	(105)
4.4	I/O定时	(106)
4.5	串行端口定时	(107)
4.6	总线定时和存储器接口	(109)
4.7	噪音防护的提示	(115)
第五章	MCS-96数据表	(117)

第一章 MCS-96的绪言

1.0 微控制机的发展

自从1976年推出世界标准的8048 (MCS-48) 微控制机以来, Intel公司继续发展了单片微控制机。1980年Intel公司推出了8051 (MCS-51), 其性能更高于8048。出现8051后, 微控制机的应用水平显著地提高了。这些多功能片的应用范围, 从键盘控制和终端设备到汽车发动机控制。8051迅速地加强了第二代世界标准微控制机的地位。

现在, 半导体工艺已发展到一个新的水平, 已经能在一个单硅片上集成100,000个以上的晶体管。Intel公司的微控制机设计人员已取得当代工艺技术的新成就, 研制出了新一代单片微控制机MCS-96。8096 (MCS-96系列有代表性的器件编号) 是迄今单片微控制机所达到的最高程度的系统集成。它使用120,000个以上的晶体管, 实现了高性能的16位CPU、8K字节的程序存储器、232字节的数据存储器、以及模拟和数字的I/O部件。图1—1示出了Intel公司单片微控制机的发展情况。

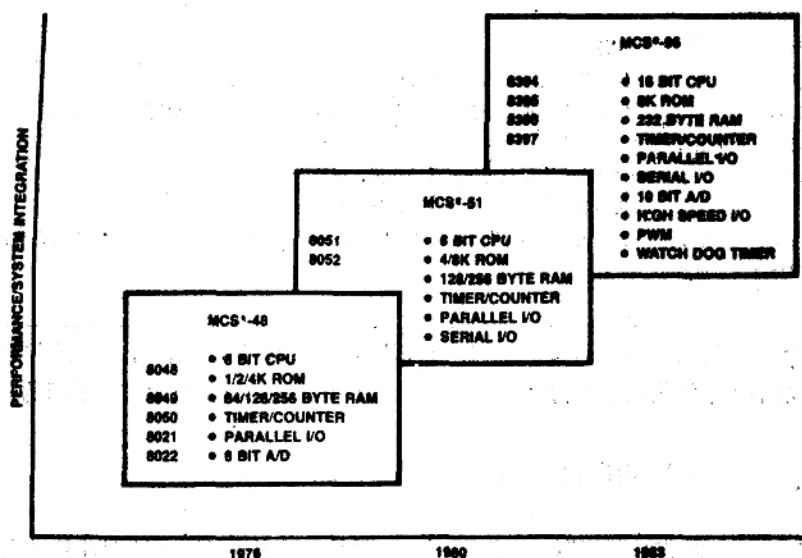


图1—1 Intel公司微控制机的发展情况

1.1 MCS-96的绪言

8096由16位高性能的CPU及与其紧密相联结的程序和数据存储器、以及几个I/O部件所组成, 所有这些都集成在一个单硅片上。CPU支持位、字节和字的操作。指令系统的子集也支持32位双字操作。在输入频率为12兆赫时, 8096完成16位加法的时间是1微秒, 完成16×16乘法或32/16除法的时间为6.5微秒。

四个高速触发脉冲输入，用来记录分辨率为2微秒的外部事件发生的时间（在12兆赫晶体频率时）。多达6个高速脉冲发生器输出，用来在预定时间内触发外部事件。这种高速输出部件可同时完成定时器功能。多达四个这样的16位软件定时器可同时操作，但两个16位硬件定时器除外。

任选的片上A/D转换器，把四个（48脚封装型式）或8个（68脚封装型式）模拟输入通道转换成10位数字量。还提供片上串行端口、监视定时器和脉冲宽度调制输出信号。表1-1列出了MCS-96的简要特性和优点。

表1-1 MCS-96的特性和优点

特 性	优 点
16位CPU	大吞吐量的高效率机器
8K字节ROM	对较复杂的大程序有大的程序空间
232字节RAM	板上寄存器堆多
硬件乘/除	提供良好的运算能力，在0.5微秒内执行 16×16 乘法和除法（在输入频率为12兆赫时）
6种寻址方式	提供较大的编程和数据操作灵活性
高速I/O部件	可以高分辨率测量和产生脉冲（在输入频率12兆赫时分辨率为2微秒）
4根专用I/O线	
4根可编程I/O线	
10位A/D转换器	读取外部模拟输入
全双工串行口	提供与其它处理器或系统的异步串行连接
多达40个I/O端口	提供TTL与标准的8位或16位外围设备兼容的数字数据I/O（包括系统扩充）
可编程的8级优先中断系统	响应异步事件
脉冲宽度调制输出	用可变占空度提供可编程脉冲序列，还能产生模拟输出
监视定时器	提供因软件故障或硬件失灵后的恢复能力
48脚（DIP）和68脚（扁平组件，引脚网格排列）封装型式	为便于安装，对I/O数量和封装尺寸有特殊使用要求的，可选择不同的封装型式

在单硅片上集成有16位CPU、全部I/O部件与接口设备的8096，代表了微控制机领域中最高水平的系统集成。它为过去不得不采用多片方法的应用技术打开了新的一页。

1.2 MCS-96的应用

MCS-96产品是为复杂的实时应用（如工业控制、仪表测试和灵活的计算机外围设备等）而设计的高性能单片微控制机。它的广阔的应用范围几乎占据了所有工业领域（见表1-2）。带有16位CPU、高速运算处理器和高速I/O的8096，用于复杂的电机控制和轴控制系统是理想的。例如能用于三相大马力交流电机和自动装置。

在选用带有10位A/D转换器的情况下，这种器件应用于数据采集系统和闭环模拟控制器中。把模拟和数字I/O处理组合在单片上，能做到相当大的系统集成。

这种芯片能理想地用于仪器产品，例如气体色谱仪，它是把模拟处理同高速数字声响结合起来的仪器。这些特点使8096成为适合于空间应用（如导弹制导和控制）的器件。

表1-2 MCS-96的应用范围

工业应用
电机控制
自动装置
断续和连续过程控制
数字控制
智能传感器
仪器装置
医学仪器
液体和气体色谱仪
示波器
消费装置
视频信号记录器
激光磁盘驱动器
高档视频游戏
制导与控制
导弹控制
鱼雷制导控制
智能弹药装填
空间制导系统
数据处理
绘图机
彩色和黑白复示器(显示器)
磁栅特画盘驱动器
击打和非击打式打印机
电讯
调制解调器
智能线路板控制
机动控制
点火控制
发送控制
反刹车制动
发射控制

1.3 MCS-96系列开发工具

产品系列得到一整套Intel公司软件和硬件开发工具的支持。这些工具缩短了产品研制周期,从而使产品成为市场竞争的胜利者。

1.3.1 MCS-96软件开发程序包

8096软件开发程序包为MCS-96系列单片微控制机提供专门设计的开发系统支援。程序包包括符号宏汇编程序ASM-96、连接再定位实用程序RL-96和程序库管理程序

LIB-96。在高级语言中，PLM-96同浮点数学程序包一起提供。目前正在为MCS-96产品系列开发其它的高级语言。

1.3.2 ASM-96宏汇编程序

8096宏汇编程序把符号汇编语言指令翻译成机器可执行的目标代码。ASM-96允许程序员以模块形式编写程序。模块程序把相当复杂的程序分成较小的功能块，以便易于编码、调整和修改。于是，使用RL-96连接再定位实用程序可以把分离的模块连接和定位成一个程序模块。这个实用程序把所选的输入目标模块组合成单个的输出目标模块。它还把存储器分配给输入程序段，并把浮动地址连接到绝对地址。然后它生成打印文件。打印文件是由连接摘要、符号表列表和前后参照列表组成的。LIB-96是另一个实用程序，它帮助生成、改进和检查程序库文件。ASM-96在Intellec系列Ⅲ或Ⅳ上运行。

1.3.3 PL/M-96

PL/M-96编译程序把PL/M-96语言译成8096浮动目标模块。这就提高了程序员的效力和应用可靠性。这种高级语言经过有效的设计，变成机器结构格式，因此就不需要更高级的程序员来使用效率很低的代码。因为这种语言和编译程序对8096及其应用环境来说是最优的，所以用PL/M-96开发软件是一种“低危险”方案。

1.3.4 硬件开发支援：iSBE-96

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/调试器组成。这种开发系统，经过两条带状电缆同用户的8096系统连接：其中一条电缆接8096 I/O端口，而另一条则接存储器总线。Intellec系列Ⅲ或其它计算机系统通过串行连接线路控制iSBE-96。把iSBE-96插入MULTIBUS插件槽内，便可得到电源，也可由外部电源供电。iSBE-96装在一块标准的MULTIBUS板上。

iSBE-96对实时硬件仿真提供最常用的功能。用户可以显示和修改存储器，设置断点，使用或不用断点执行程序，并改变存储器分配。此外，用户可以单步执行系统程序。

1.3.5 MCS-96训练班 (Workshop)

训练班为设计工程师和系统设计师提供使用MCS-96系列产品的技巧经验。训练班的内容包括解释Intel公司8096的体系结构、系统定时、输入/输出设计。实验室会话能使参加者深入地了解MCS-96系列产品和支援工具。

1.3.6 Insite程序库

Intel公司的Insite程序库包括几种应用程序。Insite中很有用的程序是8096的软件模拟程序SIM-96。它能帮助用户系统进行软件模拟。模拟程序能设置断点，检查和修改存储器，分解目标代码和单步执行代码。

1.4 MCS-96系列产品

尽管这本手册中MCS-96产品经常使用8096代表性器件编号，但整个产品系列是由8种器件编号（包括8096在内）的8种型号组成的。这种多样化的产品在I/O数量和组件尺寸方面都最能满足用户的使用要求。可选的内容包括板上8K字节掩模程序存储器、10位A/D转换器和48脚或68脚封装型式。

表1-3

备 选 器 件		68脚	48脚
数字I/O	无ROM	8096	8094
	ROM	8396	8394
模拟和数字I/O	无ROM	8097	8095
	ROM	8397	8395

表1-3列出了MCS-96产品系列的所有现行产品。

48脚型式用于双列直插式封装（DIP）。

68脚型式用于两种封装：塑料扁平封装和管脚网格排列封装。

第二章 体系结构概述

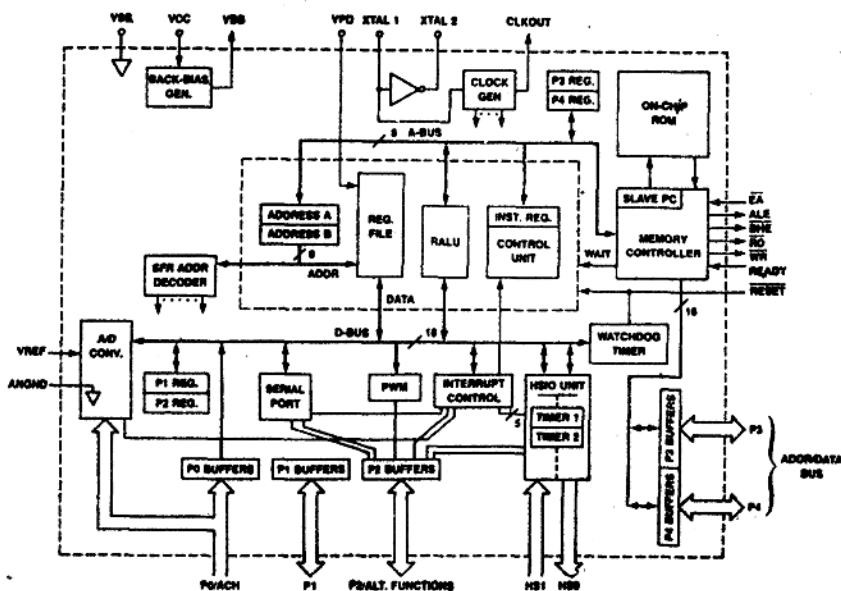
2.0 前言

为了描述8096的工作情况,我们可把8096分成几个部分,包括CPU、可编程高速I/O部件、模拟-数字转换器、串行端口和用作数字-模拟转换器的脉冲宽度调制(PWM)输出。除了这些功能部件外,还有支持整个芯片工作的几个部分,如时钟脉冲发生器和反偏压发生器。CPU和可编程序I/O的存在,使得8096与其它任何微控制机有显著的差别。下面我们首先研究CPU。

2.1 CPU的操作

对8096来说,CPU的主要元件是寄存器堆和RALU(寄存器/运算逻辑部件)。同外部世界的通信要通过特殊功能寄存器(SFR)或存贮控制器。RALU不采用累加器,它直接操作在构成寄存器堆和SFR的256字节寄存器空间。通过SFR直接控制I/O,就能实现有效的I/O操作。这种结构的主要优点是能快速地改变前后关系,无累加器阻塞,运算速度快,输入/输出时间短。

2.1.1 CPU总线



“控制部件”和两个总线连接寄存器堆和RALU。图2-1示出了CPU及其主要总线的连接情况。两个总线是8位宽的“A-总线”和16位宽的“D-总线”。D-总线仅在RALU与寄存器堆或特殊功能寄存器SFR之间传输数据。A-总线用作上述传输的地址总线，或用作连接“存储控制器”的复用地址/数据总线。内部ROM或外部存储器的任何访问都要通过存储控制器。

在内部，存储控制器是“从”程序控制器（从属的PC），在CPU中它跟踪PC。由于大多数程序是存储器参考从属PC取出指令，处理器很少把地址送给存储控制器，从而节省了处理器的时间。如果地址转移，则从属的PC便装入新值，并继续处理。从存储器取数据，也通过存储控制器实现，但对这一操作，从属的PC将不起作用。

2.1.2 CPU寄存器堆

寄存器堆包含232字节的RAM，它可以字节、字或双字的形式存取。因为这些存储单元中的每一个都可供RALU使用，所以实质上具有232个“累加器”。寄存器堆中的第一个字保留作为栈指示器，因此在出现栈处理时，它不能用作数据字。访问寄存器堆和SFR的地址，由CPU硬件暂时存储在两个8位地址寄存器中。

2.1.3 RALU控制

送RALU的指令来自A-总线，并暂存在指令寄存器中。控制部件将指令译码，并产生正确的信号序列，以使RALU执行所要求的功能。图2-1示出了指令寄存器和控制部件。

2.1.4 RALU

RALU能完成8096执行的大多数计算。如图2-2所示，RALU包含一个17位ALU、

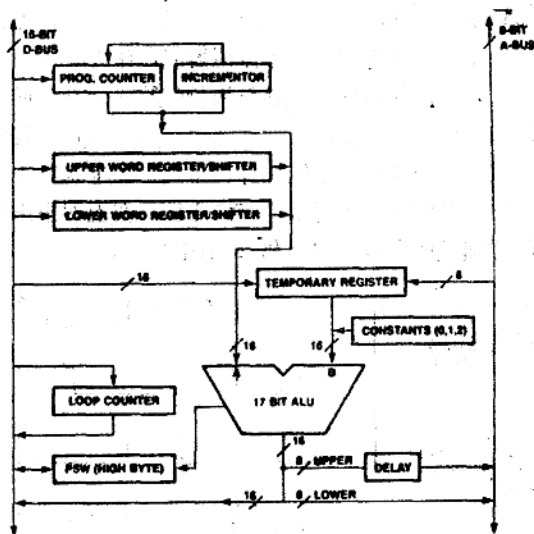


图2-2 RALU方块图

程序状态字 (PSW)、程序计数器 (PC)、一个循环计数器和三个暂时寄存器。所有寄存器都是16位或17位 (16位+符号扩充位)。某些寄存器能执行简单操作, 为ALU卸载。

单个的增量用于程序计数器PC; 然而, 转移必须通过ALU处理。两个暂时寄存器有它们自己的移位逻辑。这些寄存器用于规格化、乘、除这样一些需要逻辑移位的操作。“低位字”寄存器只在双字移位时使用; “高位字”寄存器在每次执行移位时都使用, 并对大多数指令, 它还作为暂时寄存器使用。重复移位由5位“循环计数器”计数。

暂时寄存器用来保存两个操作数指令的第二个操作数。它包括执行乘法运算的乘法器和执行除法运算的除法器。在执行减法运算时, 这个寄存器的输出在装入ALU的“B”端输入前能变补。

图2—2所示的延迟器用来把16位总线转换成8位总线。这在地址和指令都在8位A—总线上传送时才要求。某些常数, 如0、1和2都存在RALU, 以加快某些计算的速度。在RALU必须取2的补码数或执行加1和减1指令时, 这是有用的。

2.2 基本定时

8096要求输入时钟脉冲频率在6.0兆赫~12兆赫之间。这个频率可直接加到XTAL 1端。换句话说, 因为XTAL 1和XTAL 2都是反相器的输入和输出, 所以也可以用晶体来产生时钟脉冲。振荡器部分的方块图示于图2—3。电路的详细说明及其使用建议在4.1节中介绍。

2.2.1 内部定时

晶体或外部振荡器频率除以3, 产生三个内部定时相位脉冲, 如图2—4所示。每一内部相位脉冲都重复三个振荡器周期, 三个振荡器周期作为一个“状态时间”, 用作8096操作的基本测量时间。大多数内部操作都与相位脉冲A、B、或C同步, 其中每一个相位脉冲都有33%的占空度。相位脉冲A在外部用CLKOUT表示, 在68脚封装的器件上有此信号引脚。相位脉冲B和C不能在外部得到。XTAL 1、CLKOUT同相位脉冲A、B、C的关系如图2—4所示。应该注意, 在上图中未考虑传输延迟。对这些和其它定时关系的详细说明在4.1、4.4和4.6节中介绍。

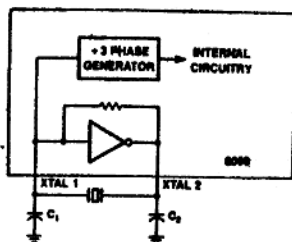


图2—3 振荡器方块图

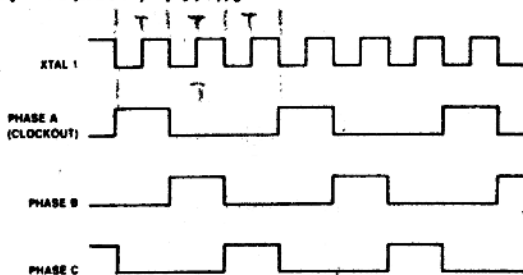


图2—4 相对于XTAL 1的内部定时

R E S E T线可用来以精确时间启动8096, 以保证试验设备与多片系统同步。这一特性的应用在2.15和4.1节的R E S E T项内有详细说明。

2.3 存贮空间

8096可寻址的存贮空间为64K字节, 对用户来说, 其中大部分可作为程序存贮器或

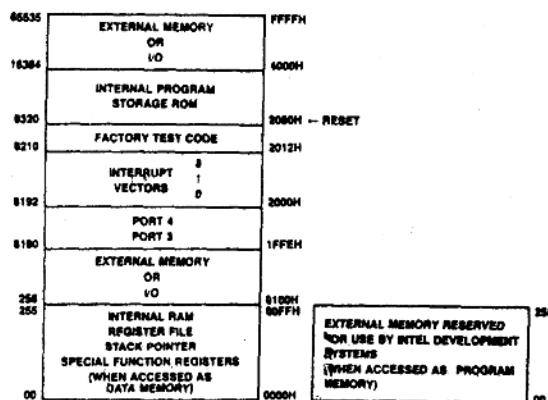


图 2—5 存贮器分配

数据存贮器使用。专用单元地址是0000H~00FFH和1FFE H~2010H。其它所有单元地址都可以用于程序或数据存贮器, 或者用于存贮器映象外围设备。存贮器安排如图2—5所示。

2.3.1 寄存器堆

000H~0FFH单元作为寄存器堆和SFR使用。对这一部分存贮空间的完整说明可以在“RAM空间”一节中找到。这一部分内部RAM不能执行代码。如果企图从000H~0FFH单元执行指令, 则指令将从外部存贮器取得。外部存贮器的这一部分由Intel开发工具保留使用。执行非屏蔽中断(NMI), 将强迫调用外部地址0000H, 因此, NMI指令也要保留给Intel开发工具。

2.3.2 保留的存贮空间

1FFE H和1FFFH单元分别保留给端口3和端口4。如果系统使用外部存贮器, 很容易改变这些端口。在4.6.7节中给出了改变I/O端口的示例。如果不改变端口3和端口4, 则这些单元就可以作为任何其它外部存贮器单元来处理。

九个中断向量存贮在2000H~2011H单元。第九个向量用于Intel开发系统, 如2.5节所述。内部单元2012H~2077H保留供Intel公司厂方测代码用。这些单元可以在外部存贮器中使用。

复位8096, 系统就从2080H取出指令。选择这一地址使系统8K RAM紧连着寄存器堆。对复位的进一步说明见2.15节。

2.3.3 内部ROM

在安排ROM时,内部存储器2000H~3FFFH单元是由用户根据2000H~2011H中的中断向量规定的。从内部ROM取指令和数据,只有在器件上有ROM,EA接为高电位,并且地址处在2000H和3FFFH之间时才能实现。在其它所有时间,数据由内部RAM空间或外部存储器存取,并且从外部存储器取指令。

2.3.4 存储控制器

RALU同存储器对话(除寄存器堆和SFR空间地址单元外)要通过存储控制器。存储控制器是由A—总线和几条控制线同RALU相连的。因为A—总线为8位宽,所以存储控制器采用了“从”程序计数器,以免总要从RALU给出指令地址。这一从属的PC在每次取指令后地址“+1”。在出现转移和调用时,从属的PC必须在连续取指令前从A—总线装入地址值。

除有从属的PC外,存储控制器还包含了字节指令排队机构,以加速指令执行。这种排队机构对RALU和用户都是透明的,除非在外部总线周期中强制产生等待状态。表3—3和3—4所列的指令执行时间,是不附加等待状态时的正常执行时间。重装从属的PC和取新指令流的第一字节,需要4个状态时间。表3—4反映了有转移/无转移的不同时间。

2.3.5 系统总线

外部存储器通过AD0~AD15线寻址,后者形成16位复用(地址/数据)数据总线。这些线同I/O端口3和端口4共用引脚。地址锁存允许(ALE)线的下降沿为透明锁存(74LS373)提供时钟脉冲,以便使总线信号分离。典型的线路和所要求的定时如4.6节所述。因为8096的外部存储器可以用字节和字来寻址,所以译码受两条线[总线高位允许(BHE)和地址/数据线0(AD0)]的控制。BHE线必须象地址那样透明地锁存。

为避免在解释存储系统时造成混乱,应适当地为复用地址/数据信号命名。地址信号称为MA0~MA15(存储器地址);数据信号称为MD0~MD15(存储器数据)。

在BHE有效(低)时,应选择连接高字节数据总线的存储器。当MA0低时,应选择连接低字节数据总线的存储器。这样,对16位字长的存储器访问时,访问低(偶)字节只要MA0=0, BHE=1;访问高(奇)字节只要MA0=1, BHE=0;访问双字节只要MA0=0, BHE=0。当存储区只读时, BHE与MA0不一定要译码。

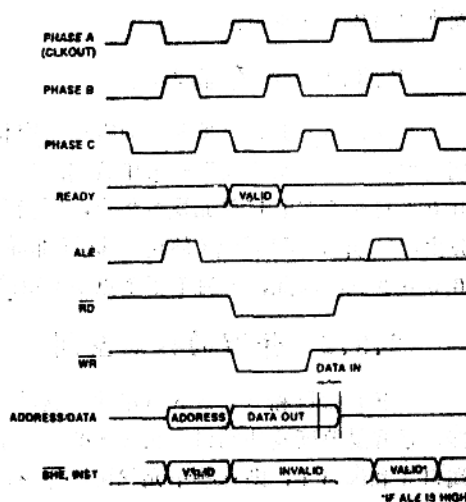


图2—6 外部存储器定时

图2—6示出了有关下面将要描述的外部存储器操作的理想波形。精确的定时数据，请查阅最后的数据表。在外部存储器取数开始时，地址锁存允许（ALE）线上信号变高，地址放到AD0~AD15，并把BHE置于要求状态。然后ALE下降，地址送出引脚，RD（读）信号变低。然后READY线变低，以便用几个附加的状态时间把处理器保持在这一状态下。

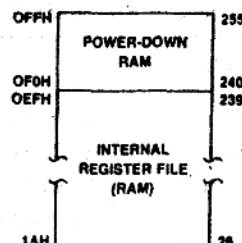
2.3.6 准备就绪

READY线能把处理器保持在上一状态，以便访问慢速存储器，或满足DMA要求。READY线在产生CLKOUT信号的内部相位2时采样。有一个在CLKOUT变低之前READY必须稳定的最小时间。如果没有确保这个准备时间，这一部件将转向“未准备就绪”的状态，则这一部件便会失去预期的操作能力。

尽管READY线在任何时间都能处于高电位，但由于READY与CLKOUT同步，所以8096在等于CLKOUT的某一倍数的周期内将处于“未准备就绪”的状态。保持8096处于“未准备就绪”状态的最长时间，典型的数据约1微秒。对所要求的特定器件和温度范围，数据表中规定了精确的时间。

来自外部存储器的数据必须在总线上，并在RD上升沿之前用最短的规定准备时间稳定。RD的上升沿把信息锁存在8096内。如果是读数据，则在地址有效时，INST引脚应处于低电位。如果是读指令，则在这一时间内INST引脚应是高电位。48引脚封装的器件没有INST引脚。

对外部存储器的写入所要求的定时，与由它读出所要求的定时相似。主要差别在于用写（WR）信号代替RD信号。在WR线下降沿之前，定时是相同的。在WR下降沿，8096撤去（断开）地址，并把数据置于总线上。如上所述，这时READY线必须保持在要求状态。在WR线变高时，数据应锁存在外部存储器内。在写入时，INST总是处于低电位，因而指令不能写入。存储器访问时的精确定时数据可在数据表中找到。



19H	STACK POINTER	25	STACK POINTER
18H		24	
17H		23	PWM_CONTROL
16H	IOS1	22	IOC1
15H	IOS0	21	IOC0
14H		20	
13H	RESERVED	19	RESERVED
12H		18	
11H	SP_STAT	17	SP_CON
10H	IO PORT 2	16	IO PORT 2
0FH	IO PORT 1	15	IO PORT 1
0EH	IO PORT 0	14	BAUD_RATE
0DH	TIMER2 (HI)	13	
0CH	TIMER2 (LO)	12	RESERVED
0BH	TIMER1 (HI)	11	
0AH	TIMER1 (LO)	10	WATCHDOG
09H	INT_PENDING	9	INT_PENDING
08H	INT_MASK	8	INT_MASK
07H	SBUF (RX)	7	SBUF (TX)
06H	HSI_STATUS	6	HSO_COMMAND
05H	HSI_TIME (HI)	5	HSO_TIME (HI)
04H	HSI_TIME (LO)	4	HSO_TIME (LO)
03H	AD_RESULT (HI)	3	HSI_MODE
02H	AD_RESULT (LO)	2	AD_COMMAND
01H	RD (HI)	1	RD (HI)
00H	RD (LO)	0	RD (LO)

(WHEN READ)

(WHEN WRITTEN)

图2—7 寄存器堆存储器分配

2.4 RAM空间

8096的内部寄存器地址单元分成两组:寄存器堆和特殊功能寄存器组(SFR)。R - L U可以在任何256内部寄存器单元上操作。00H~17H单元用来访问SFR。18H和19H单元作为栈指示器。这里对使用剩下的230单元没有限制,除非代码不根据它们执行。

2.4.1 特殊功能寄存器组

8096的所有I/O都通过SFR组控制。SFR中的许多寄存器有两种功能:一种是读出;另一种是写入。图2-7示出了这些寄存器的单元地址和名称。每一寄存器的性能摘要列于图2-8。完整的说明见后几章。

寄存器	说 明	章 节
RO	零寄存器—总读作零,变址时对基地址有效,计算和比较时是常数	3.2.7
AD—结果	A/D结果高/低—A/D的低位/高位数据结果	2.9.3
AD—命令	A/D命令寄存器—控制A/D	2.9.2
HSI—方式	HSI方式寄存器—设置高速输入部件的方式	2.7.1
HSI—时间	HSI时间高/低—保存高速输入部件所触发的时间	2.7.4
HSO—时间	HSO时间高/低—设置高速输出时间,以执行指令寄存器的命令	2.8.3
HSO—命令	HSO指令寄存器—确定在给HSO时间寄存器加载时将发生的情况	2.8.2
HSI—状态	HSI状态寄存器—指示当时HSI时间寄存器中哪个HSI引脚被检测	2.7.4
SBUF(RX)	接收串行端口的缓冲,保持待输出的内容	2.11
SBUF(TX)	发送串行端口的缓冲,保持由串行端口接收的字节	2.11
INT—屏蔽	中断屏蔽寄存器—允许或禁止单个中断	2.5.2
INT—未决	中断未决寄存器—指示在一个中断源上何时出现中断信号	2.5.2
监视器	监视器定时寄存器—周期性写入,以在每64K状态时间停止自动复位	2.1.4
定时器1	定时器1高/低—定时器1的高字节和低字节	2.6.1
定时器2	定时器2高/低—定时器2的高字节和低字节	2.7.8
IO端口0	端口0寄存器—端口0引脚上的电平	2.6.2
波特率	保存波特率的寄存器,这个寄存器顺序加载	2.7.8
IO端口1	端口1寄存器—用于对端口1读和写	2.12.1
IO端口2	端口2寄存器—用于对端口2读和写	2.11.4
SP—状态	串行端口状态—指示串行端口的状态	2.12.2
SP—控制	串行端口控制—设置串行端口的方式	2.12.3
IOS0	I/O状态寄存器0—保存HSO状态的信息	2.11.3
IOS1	I/O状态寄存器1—保存定时器和HSI的状态信息	2.11.1
IOC0	I/O控制寄存器0—控制HSI引脚的变换功能:定时器2作为复位源还是作为时钟源	2.13.4
IOC1	I/O控制寄存器1—控制端口2引脚的变换功能:定时器中断还是HSI中断	2.13.5
PWM—控制	脉冲宽度调制寄存器—设置PWM脉冲的宽度	3.7.2
		2.13.2
		2.13.3
		2.10
		4.3.2

图2-8 SFR摘要

在SFR空间有几个标有“被保留”的寄存器。这些寄存器保留用于扩充或测试。它们的读出和写入可能产生不希望有的结果。例如，对000CH地址单元的写入会把两个定时器置成0FFFXH，这种特性用于测试器件，而不适用于程序。

2.4.2 掉电

RAM中靠上面的16个单元(0F0H~0FFH)由VCC引脚和VPD引脚同时供电。如果在掉电情况下要求保持这些单元的存储器信息，则只要保持VPD引脚上有电压即可。为使RAM保持信息所要求的电流约为1毫安(精确的数值参考数据表)。

为把8096置于掉电模式，RESET引脚电平应变低。两个状态时间过后，器件便复位。这是为防止器件在掉电时对RAM误写所必须的。这时，可从VCC引脚断开电源，而VPD引脚电源继续保持在规定的指标之内。8096可以在任何时间保持这种状态，而16个RAM字节将维持它们的信息。

为使8096脱离掉电模式，在加VCC时，RESET应保持低电位。在振荡器和反向偏压发生器稳定后的两个状态时间里(约1毫秒)，RESET引脚电平变高。在RESET电平升高后的10个状态时间内，8096开始从02080H单元执行代码。图2-9示出掉电序列定时图。为了保证满足2个状态时间的最小复位时间(与CLKOUT同步)，建议采用10个XTAL1周期。对实际硬件连接的建议，见4.1节。复位在2.15节讨论。

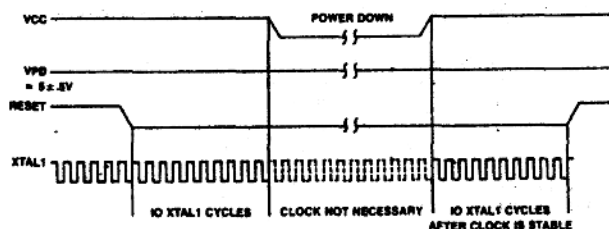


图2-9 掉电定时

2.5 中断结构

2.5.1 中断源

8096可有8个中断源。在中断允许时，这些中断源上任一个出现中断都会强迫转移到存有该中断源向量地址的存储器单元。中断源及其各自的向量地址示于图2-10。除了8个标准的中断外，还有TRAP指令，用作产生软件中断。这一指令通常不用MCS-96汇编程序的支援，而留给Intel开发系统使用。某些中断源可用几种方法启动。图2-11示出所有可能的中断源。

Source	Vector Location		Priority
	(High Byte)	(Low Byte)	
Software	2011H	2010H	Not Applicable
Extint	200FH	200EH	7 (Highest)
Serial Port	200DH	200CH	6
Software Timers	200BH	200AH	5
HSI.0	2009H	2008H	4
High Speed Outputs	2007H	2006H	3
HSI Data Available	2005H	2004H	2
A/D Conversion Complete	2003H	2002H	1
Timer Overflow	2001H	2000H	0 (Lowest)

图 2—10 中断向量地址

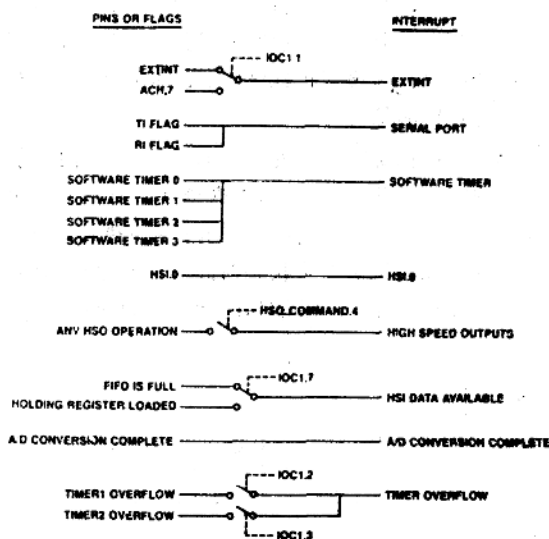


图 2—11 所有可能的中断源

2.5.2 中断控制

中断系统的方块图如图 2—12 所示。每个中断源都用 0—1 跃变来测试。如果出现这种跃变，则将地址为 0009H 的中断未决寄存器的相应位置位。因为这一寄存器可以写，所以通过在寄存器内置相应位，可以产生软件中断，或者通过清除这一寄存器中的相应位，可以去掉未决的中断。即使中断禁止了，中断未决寄存器仍能置位。

在写中断未决寄存器以清除中断时必须谨慎。如果在清除这一位时中断早已响应，则会出现 4 个状态时间的“局部”中断周期。这是因为 8096 将取正常指令流的下一条指令。对程序的影响，实际上是附加 NOP 的影响。如果 8096 在“读/改/写”指令时离开响应的中断，则用 2 操作数直接逻辑清除这一位，便能防止上述影响。

每个中断的允许和禁止，是通过位于 0008H 地址的中断屏蔽寄存器实现的。如果屏蔽寄存器中的这一位是“1”，则中断允许，否则便禁止。即使一个中断被屏蔽掉，它仍可能变成未决的。因此，在未屏蔽一个中断之前，希望清除未决位。

中断屏蔽寄存器也是 PSW 的低字节。在采用“EI”（允许中断）和“DI”