

INTEL

8086

微处理机与微计算机

(下册)

王兆全 编写

国营第五七四厂研究所

一九八三年七月

876
39

目 录

第三章 8086 外围支持电路

3.1 微处理机与微计算机	(1)
3.2 微型机的存贮器	(2)
3.2.1 RAM 存贮器	(2)
3.2.2 ROM 存贮器	(7)
3.2.3 存贮器框图	(11)
3.3 时钟产生/驱动器	(11)
3.3.1 8224时钟产生/驱动器	(11)
3.3.2 8284时钟产生/驱动器	(13)
3.4 总线管理	(16)
3.4.1 总线驱动器8216/8226	(16)
3.4.2 8288总线控制器	(18)
3.4.3 8289总线仲裁器	(22)
3.5 I/O 接口控制电路	(33)
3.5.1 外部设备与处理机之间的信息交换	(33)
3.5.2 串行数据通讯	(34)
3.5.3 8251可编程串行通讯接口 (PCI)	(37)
3.5.4 并行数据通讯	(47)
3.5.5 8255A可编程序外设接口 (PPI)	(48)
3.5.6 8253可编程序定时器 (PIT)	(63)
3.5.7 8259A可编程序中断控制器 (PIC)	(77)

第四章ISBC86/12A单板计算机

4.1 ISBC 86/12A 概述	(105)
4.1.1 一般说明	(105)
4.1.2 RAM和ROM/EPROM的扩展选择	(108)
4.1.3 系统软件的开发	(108)
4.1.4 ISBC86/12A技术说明	(108)
4.2 ISBC86/12A单板机逻辑功能块简介	(111)
4.2.1 时钟电路	(111)
4.2.2 中央处理单元	(111)
4.2.3 定时器	(111)

4.2.4 串行I/O	(112)
4.2.5 并行接口	(112)
4.2.6 中断控制器	(112)
4.2.7 RAM 结构	(112)
4.2.8 ROM/EPROM结构	(113)
4.2.9 总线结构	(113)
4.2.10 多总线接口	(114)
4.3 电路分析	(114)
4.3.1 初始化	(115)
4.3.2 时钟电路	(115)
4.3.3 8086cpu 时序	(116)
4.3.4 地址总线	(118)
4.3.5 数据总线	(118)
4.3.6 总线定时输出	(119)
4.3.7 内部控制信号	(119)
4.3.8 双端口控制逻辑	(119)
4.3.9 多总线仲裁	(122)
4.3.10 I/O 操作	(122)
4.3.11 ROM/EPROM 操作	(123)
4.3.12 RAM 操作	(124)
4.3.13 中断操作	(125)

第三章 8086外围支持电路

3.1 微处理机与微计算机

微处理机是一个单独的芯片,具有一定逻辑功能的电路,实际上只是微型计算机的运算器和控制器的部分,或者说是微型计算机的中央处理单元,所以它只是微型机的一个组成部分。而微计算机又称微型机,组成系统后,被称做微型计算机系统,它由四部分组成:中央处理单元(CPU)、存储器、输入/输出(I/O)装置和系统软件。由于微处理机不具有存贮能力和输入输出能力,所以它只能被用来做一些特定的逻辑控制单元,而只有在以微处理机为中心,加上只读存储器芯片,随机存储器芯片, I/O接口芯片,以及一些必要的输入/输出设备,才能构成一个软件能在其上运行的计算机系统。

微型计算机在原理上、结构上和功能上与其它计算机并无原则的差别。在运算原理、控制原理、数据传送原理、程序设计原理等方面,微型机和其它计算机基本上相同或是完全相同。比较起来微型机的系统设计及结构原理和小型机关系密切。微型机继承和发展了小型机的一些特点,如小型机的总线方式,通用寄存器阵列、固定或浮动堆栈技术,并行处理,以及微程序控制方式等先进技术都在微型机中得到运用和发展。但是微型机又不是小型机的简单微型化,微型机要克服集成电路带来的许多制约,要在有限的硅片尺寸内实现较高的性能,必须采用精巧的设计技术。使得微型机在结构上更加简单、灵活,在性能上更加可靠。

微型机具有简单的科学计算、数据处理和过程控制等功能。虽然在功能上,微型机和其它计算机比较起来功能较弱,但基本功能微型机和其它计算机是完全一样的。凡其它计算机能完成的任务,给以足够的时间和足够的存贮容量,微型机都能完成。微型机中的高档机在性能上已接近小型机的水平。比较起来微型机的程序设计比小型机要困难和麻烦一些,微型机的硬件和软件两者之间牵涉的程度更深一些,软硬件的界线更加模糊。

下面简单介绍一下微型机的特点。

1. 体积小,重量轻

微处理机以及其它外围支持芯片的硅片尺寸只有几十平方毫米(如8086 CPU 225平方密尔的芯片),经过封装芯片也只不过绘图橡皮大小。重量只有十几克,组装成微型计算机所需要的芯片数目也不多。例如ISBC86/12A单板计算机仅用99块集成电路。关于ISBC86/12A的详细情况以后我们还要介绍。目前已经出现单片式微型计算机,在一个芯片上就集成了一台计算机,其中包括运算、控制、存贮和I/O接口全部电路,尺寸上更进一步缩小。因此,有可能把微型机看成是控制对象的一个配件,这使得很多无法和不便使用计算机的场合可以很方便地使用上微型机。

2. 成本低

国外每块大规模集成电路芯片的售价只有几美元到几十美元,半导体存储器的成本已降到每位0.1美分,比磁芯存储器的成本还低。而且随着大规模集成电路工艺的不断发展,其成本还要进一步降低。不但企业用得起,而且私人也用得起,有些家庭中自己就有INTEL公司的开发系统,很多青少年都会编制较大的微处理机程序。与微型机芯片的价格相比,外围设备的价格显得太高了。有一个这样的比喻,微型机(不包括外部设备)的价格相当于等边三角

形的角，而外部设备的价格相当于这个三角形两腰中点的连线，而软件的价格却相当于一个边。据报导如果买系统的话微处理机主机部分以后能够白送。对微型机系统来说，如何降低外部设备的价格是十分关键的了。

3. 简单、灵活、可靠

微型机有如芯片的积木式组合，可以根据实际的需要，规模可大可小，便于扩大和缩小，十分灵活方便。

控制程序可以存放在ROM中，掉电时不会被破坏。由于采用大规模集成电路芯片，焊点接线大量减少（约减少1—2个数量级）。电路功耗少，发热量小，使微型机的可靠性大大提高。

但微型机也有缺点，如速度低，用微型机本身研制软件比较困难等。

下面我们介绍一下组成微型机所用的一些必要的电路，如存贮器、时钟产生/驱动器、总线管理和I/O接口控制电路。

3.2 微型机的存贮器

除个别情况外，微型计算机都是使用半导体存贮器。半导体存贮器从工艺技术上分有两种类型：MOS型半导体存贮器和TTL型半导体存贮器。微型计算机使用的半导体存贮器大都是MOS型的。MOS型半导体存贮器有P沟道和N沟道两种，比较先进的是N沟道的。

存贮器是存放指令和数据的地方，一个计算机的性能在很大程度上取决于存贮器容量的大小和存贮器的存取速度。存贮器有两种基本形式。通用微型机中的最常用的一种存贮器是随机存取的存贮器，既可以存入数据，也可以将存入的数据取出，一般简称RAM，另一种是只读存贮器ROM。ROM只允许从存贮器中读出数据，不允许新的数据写入该存贮单元，所以ROM不运用来存放随机使用的数据，而可以用来存放固定的常用程序。

3.2.1 RAM存贮器

RAM存贮器一般有两种，一种是“动态”的，一种是“静态”的。

1. 静态RAM存贮器存贮单元工作原理

静态RAM存贮器的存贮单元如图3—1所示，其中采用了六只晶体管，由交叉耦合的触

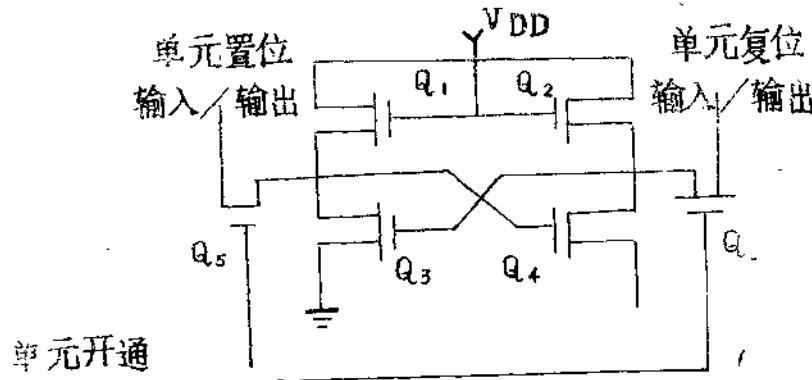


图3 - 1 静态RAM存贮单元

发器存贮“1”和“0”的信息。晶体管 Q_3 和 Q_4 组成一个简单的RS触发器，它是基本存贮单元。晶体管 Q_1 与 Q_2 作为 Q_3 与 Q_4 的晶体管负载。 Q_3 与 Q_4 将该单元的信息接至外部电路。若要对此单元存入信息时，要在单元开通端加一个选通信号，然后由单元的置位或复位端加入信息脉冲，使此单元成为“1”状态或为“0”状态。在读取此单元时，同样要在单元开通端加一选通信号，就可在单元置位（或复位）输入/输出端上读取了此单元中所存放的信息。在电源不断电的情况下，此单元中存放的信息将保持不变，即不会丢失。由于每个存贮单元要由六只晶体管组成，所以单片的存贮容量比较小。

2. 动态RAM存贮单元工作原理

动态RAM存贮单元，有的采用三个晶体管（如图3—2a所示）有的只用一个晶体管（如图3—2b所示），对于动态RAM的存贮单元，其数字信息是靠MOS管的栅极电容上的电荷来存贮的。如图3—2a所示。当要对此单元存入数据时，先发一个字写信号，然后在数据输入端将要写入的信息“0”或“1”加在数据输入端上。根据信息是“1”还是“0”决定对电容是否充电。在读此单元的信息时，在字读信号端加一信号，即可在数据输出端上读取存放在此单元中的信息。而图3—2b所示的是一个单晶体管、单电容器的动态存贮单元。单元存贮电容上充有电荷表示逻辑“1”，放了电的电容器代表逻辑“0”， Q_1 用来把该电容器连接到输入/输出线。由于动态RAM存贮单元的信息是靠电容上的电荷保存的，而任何电容都会有漏电，并且对于MOS的栅极电容来说，电容上的电荷大约只能维持若干毫秒。因此，为了维持存贮的信息，每隔1或2毫秒就必须对动态RAM存贮器刷新一次，其刷新过程是，先读出存贮单元信息，然后根据这个信息，再将它写回到存贮单元中去。

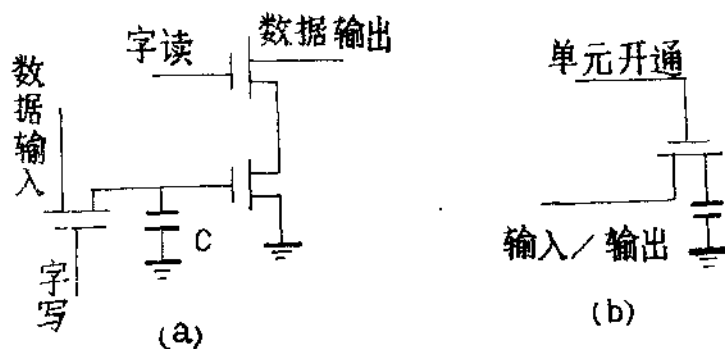


图3 - 2 动态RAM存贮单元

3. 动态RAM的刷新

动态RAM的刷新一般是-行一行地进行的。每一行所有的读选择线(字线)都是公用的。当这条线被选中时，所有各位的存贮单元都被读出，并且，按照原来所存贮的内容重新恢复栅极电容存贮的电荷。对于1024位的动态存贮器（ 32×32 ）来说，在2ms的时间内，要对整个矩阵的所有32根读选择线依次加上信号，使32根线轮流作用一遍。由于在刷新时不必加片选信号，所以对由1024位存贮器片组成的 $16 \times 64K$ 字的存贮器，只需要加32次读选信号就可将其全部刷新。图3—3是典型的动态存贮器刷新线路。假如现在对第i行的存贮单元进行刷新，也就是说，现已给第i行的读选择线加上信号。在这种情况下，如果K列的存贮单元原

来存“1”，则读出线上此时的信号为“0”。于是，刷新放大器的 T_a 关闭。如果控制端和预充电端 P 都为负，则写入数据信号将为 V_{DD} 电平（即逻辑信号为1）。若给写选择线加上信号，则将使 T_2 管的栅极电容重新充电，从而变为“1”信号。同理，若存储单元原来存“0”，则 T_2 管的栅极电容将仍旧处于放电状态。该线路中的刷新放大器，可以为一列的存储单元所共用。

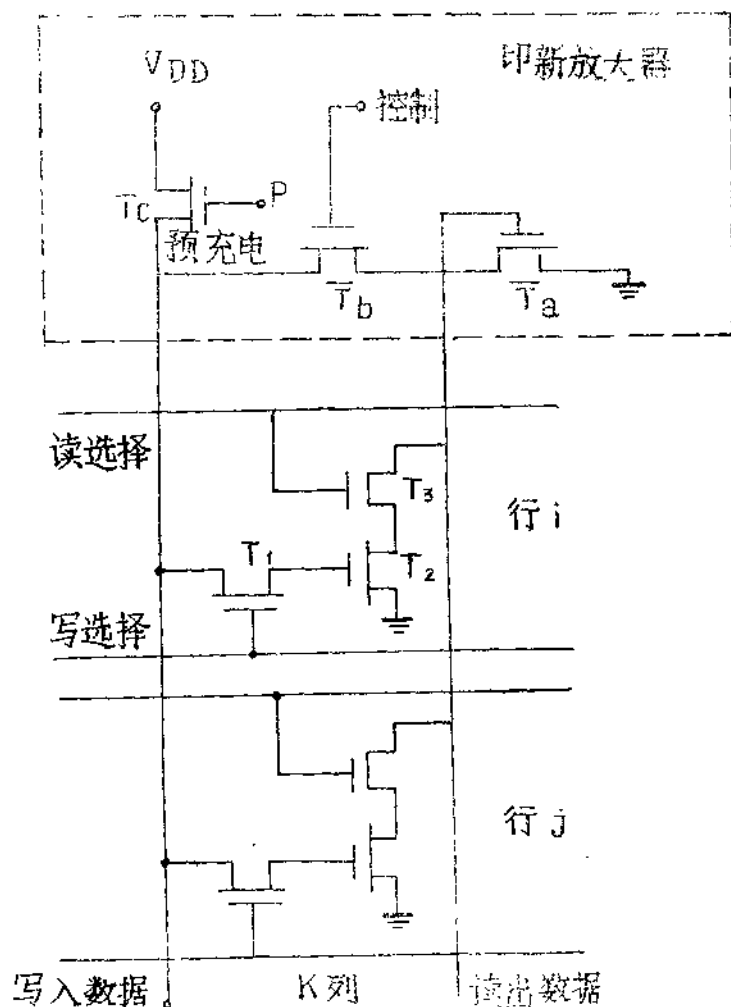


图3-3 动态存储器刷新线路

必须指出，刷新操作有下述两个特点：（1）必须在一定时间间隔内刷新。一般选择为2ms，这个时间还和温度有关。如果在高温环境下工作，那么，由于栅极漏电流将随温度升高而增大，所以刷新的时间间隔要缩短。（2）刷新操作要优先于读写操作。由于刷新是定时产生的，而读出或写入操作则是随机的。这样就有可能在进行刷新操作的同时，正好要求读写操作。在这种情况下，如果刷新正在进行，则要求刷新操作完成以后再进行读写操作。

4. 2117、16384×1 bit动态RAM简介

2117是INTEL公司生产的高性能、高可靠性和高存储密度的16384×1位的动态MOS RAM存储器。2117是用单个晶体管作为动态的存储单元，是一种先进的动态电路，具有高速和低功耗，这种电路所用的晶体管数目最少。2117在系统环境中具有高抗干扰性，并且能够允许电源电压波动±10%。2117采用工业标准的16线双列直插式封装，其封装引脚和逻辑符号及2117电路方框图如图3—4所示。2117电路由7位的行和列锁存器，地址选通电路、两个时钟发生器、允许写缓冲器、数据输出锁存器、两个64—1的行译码器、2个64×128单元的存储阵列、64—1列译码的128位的读出放大器、输出选通电路和输出缓冲器组成。

将14根地址分成多路复用的二组地址信号，每组7根，分时使用 $A_0 \sim A_7$ 。通过两个TTL时钟将这两个7位的地址字锁存起来。一组被称为行地址，而另一组被称为列地址。 $\overline{RAS}$ 为行的选通脉冲， $\overline{CAS}$ 为列的选通脉冲。

2117具有三态输出，输出信号由 $\overline{CAS}$ 控制。在读或者是读改为写周期以后，当 $\overline{CAS}$ 为低电平的时候将数据锁存在输出端。当 $\overline{CAS}$ 成为高电平时，数据输出端呈高阻抗。当 $\overline{RAS}$ 信号被用在执行“ $\overline{RAS}$ -仅刷新周期”中时，2117的隐藏刷新特性允许 $\overline{CAS}$ 保持低电平以维持被锁存的数据。隐藏刷新是指在输出端上正保持着有效数据时执行刷新周期。这样可使刷新周期“隐藏”在数据周期中而不影响数据的使用。

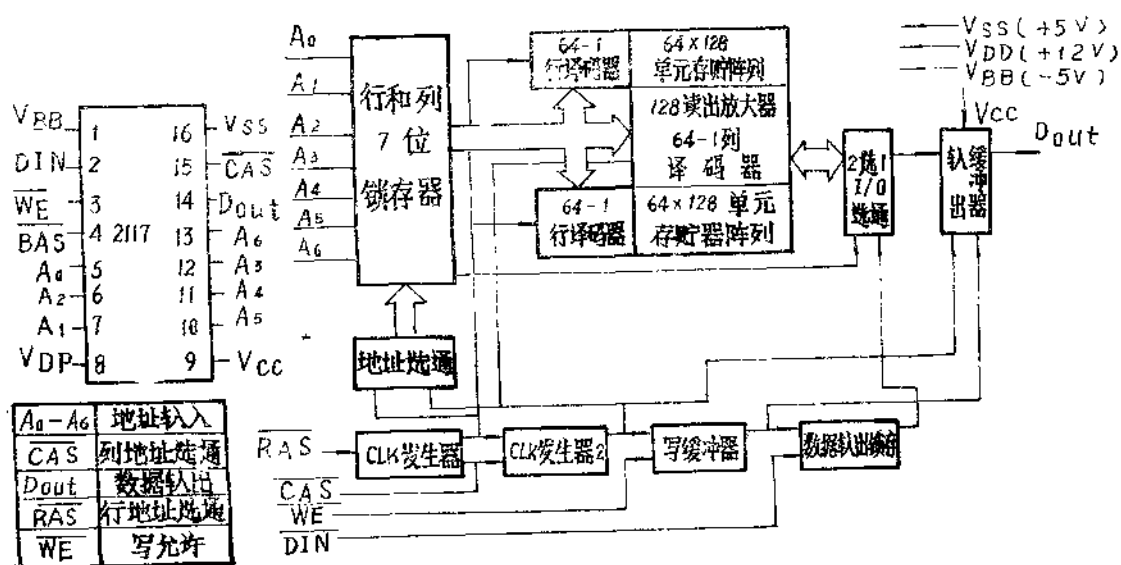


图3-4 2117 RAM动态存储器封装与逻辑框图

为了保持数据，单晶体管的动态存储单元要求对数据进行刷新。刷新是通过执行“ $\overline{RAS}$ -仅刷新”周期，或在执行读写周期时的隐藏刷新周期来完成的。

5. 8202动态RAM控制器

8202是动态RAM系统控制器，提供了对2014A、2117或者2118动态存储器控制的必须的所有信号。8202的封装和逻辑框图如图3—5所示。8202在不附加外部驱动器的情况下能直接寻址和驱动128K字节的能力。提供了地址多路复用和地址选通信息。可从内部或外部请求刷新周期。提供了传输刷新的能力，完全和8080A、8085A和8086微处理机相兼容。给出了系统询问和传输询问信号、即可由外部提供时钟也可由内部产生时钟。

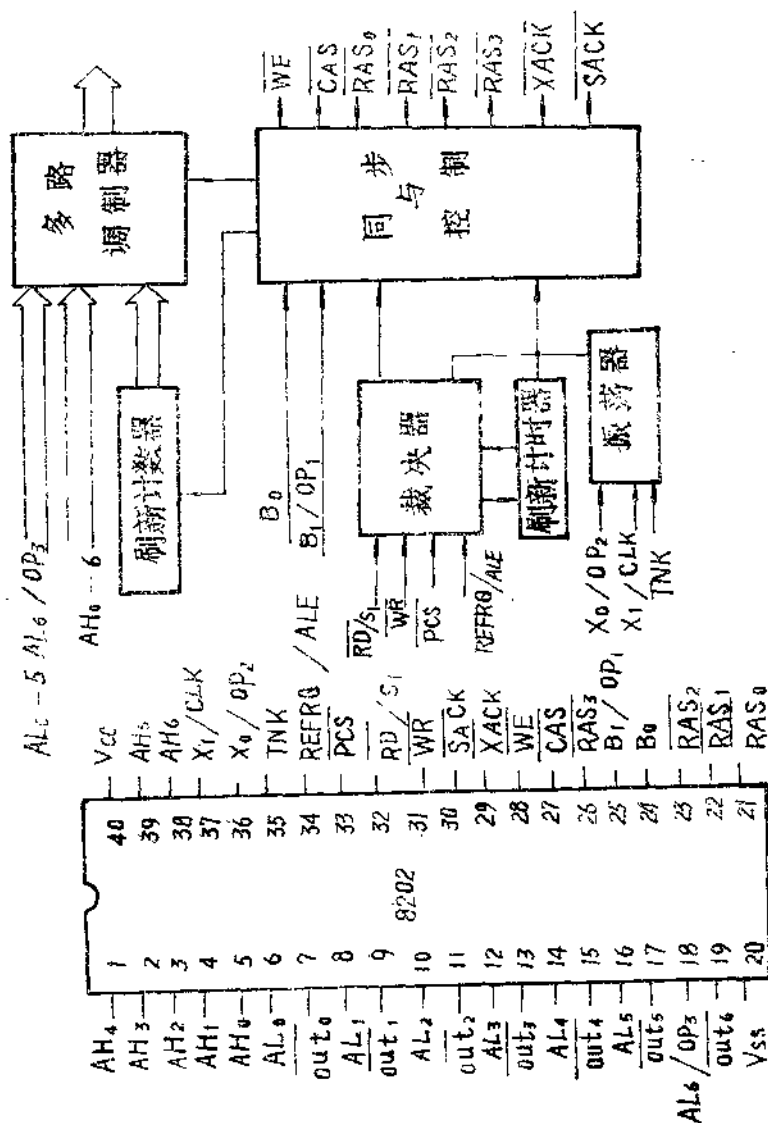


图3—5 动态RAM控制器封装与框图

3.2.2 ROM只读存储器

只读存储器在微处理机领域里起着极为重要的作用。不少微处理机在应用中只要一个固定的程序就够了，而且这一程序是反复使用的。例如，用于控制交通指挥灯的微处理机就是这种情况。控制交通指挥灯的程序主要是计时、定时和决定红绿黄灯什么时候开启。又如微处理机控制的某种玩具游戏，它的控制程序也可以是固定的。只读存储器实际上是一个单向选择的开关阵列。图3—6表示一个16位的阵列。4位地址译码中的两位用来选择四行中的一行，而另外两位地址码经过译码器，用来选择四列中的一列，在行和列同时被选中的地方，也就是存储单元被选址时，在存在二极管的地方存储器便产生逻辑“1”的输出，而在二极管被断开的地方，则只读存储器便产生逻辑“0”的输出。

只读存储器有很多不同的形式。一种可进行大量生产的最常用的方法是掩模可编程只读存储器。该存储器的内容由用户指定，然后由厂商根据要求进行生产。蚀刻掉集成电路中某些金属接线。一旦掩模按照顾客的要求被蚀刻后，存储在只读存储器中的程序是永久不变的，只有另刻一新掩模时，才能改变它的程序，也就是需要重新做。

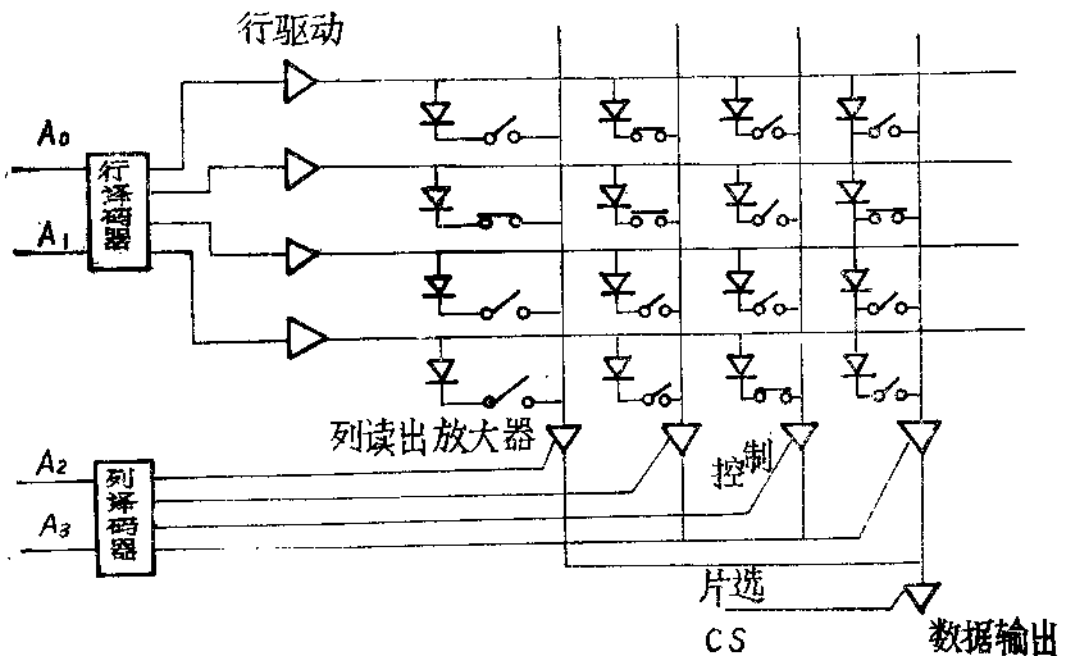


图3-6 16位简化的ROM阵列

只读存贮器的第二种型式是易熔只读存贮器，易熔只读存贮单元如图3—7所示。既用户可编程序只读存贮器，在原始状态时，易熔只读存贮器的每一位都是逻辑“1”，而用户根据自己的具体使用程序，在要求切断的地方“熔断”其熔丝。使其产生逻辑“0”。这样的只读存贮器一旦经过“熔断”处理，它就被永久性地编码了。这种存贮器也称为可编程序只读存贮器（PROM）。

掩模式ROM大都用在有固定程序的，生产批量很大的产品中。PROM一般用在用户可以修改程序，生产批量不太大的产品中。

通常PROM还包括另一种型式的存贮器，这种形式较确切的名称应该是 EPROM，即可擦除可编程只读存贮器。它可按程序需要写入信息，并可将其长久保存，不用时，可用紫外线照射，擦去原来写入的信息。EPROM有两种常用的技术，最常用的老式技术是紫外线可擦除型。紫外线可擦除只读存贮器采用一种特殊的石英罩来覆盖集成电路。当紫外线照到集成电路片上时，被“摧毁”的零又重新“治愈”了，另一种较新的技术是用电流来恢复。这种可擦除程序只读存贮器有时候也叫做电可变只读存贮器。

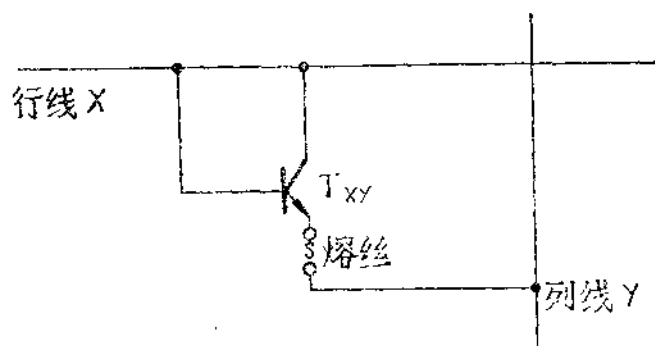


图3 - 7 熔丝式 PROM 单元

下面以INTEL2708为例，详细说明PROM存贮器工作过程。

2708是MOS的8K(1024×8)的可擦洗PROM，电源电压为±5V和±12V；读数时间最大为450ns。在每次擦洗以后，所有单元都存“1”信息，编程时在18引线端加上26V脉冲，加过脉冲的单元的信息变为“0”，2708的封装图和框图见图3—8。这里，单元阵列排列为64行×128列，128列又分成16组，每组8位“数位”。当行译码器选中64行中的一行时，列译码器选择16组之一，即在这一行中，有8列的单元应从数位线（8位）上输出信息。如果这个单元原来为“1”，则该单元晶体管导通，数位线上出现低电位（接近地电位）；如果这种单元原来为“0”，数位线上为高电位，则该单元晶体管不导通。

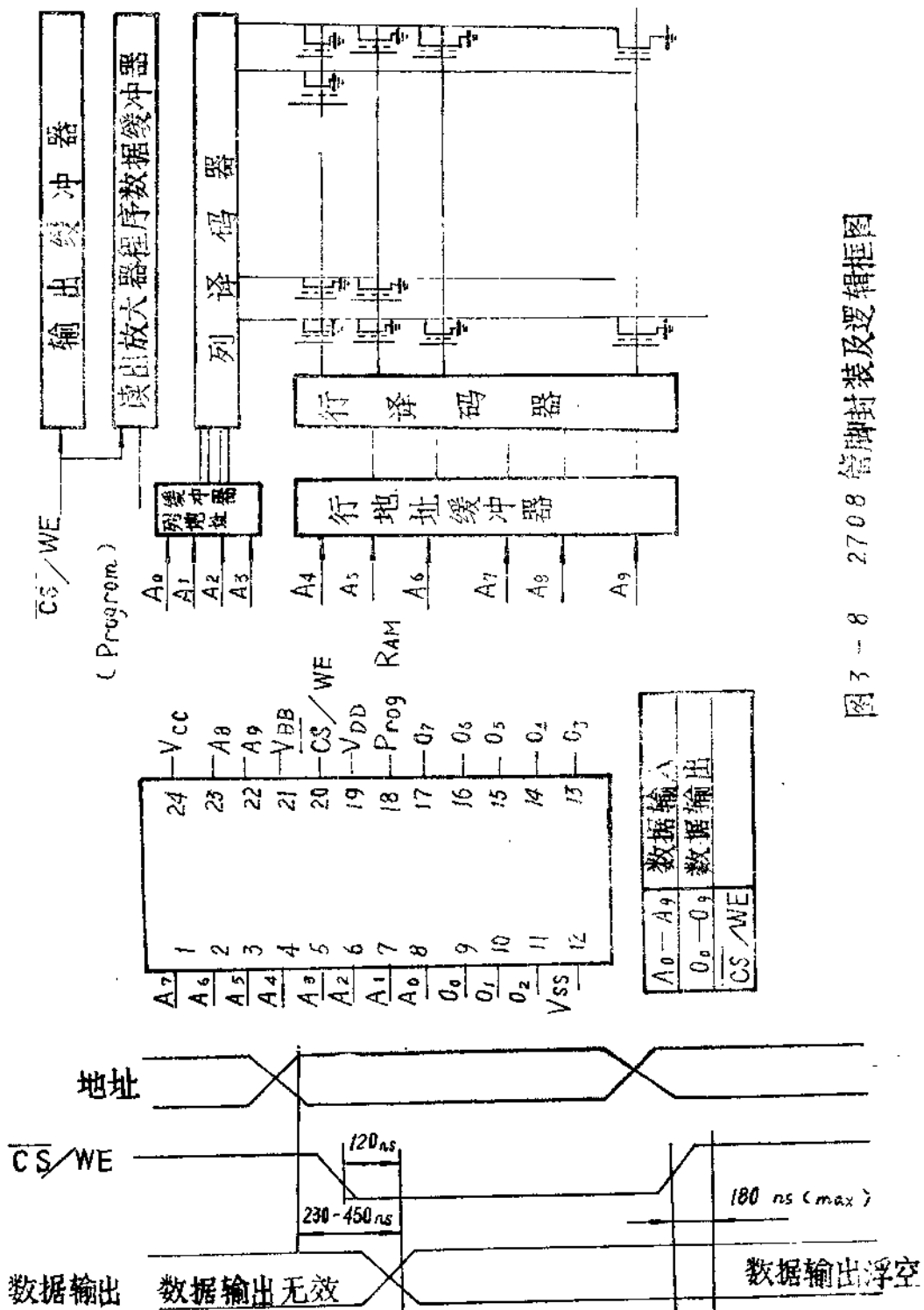


图 3 - 8 2708 管脚封装及逻辑框图

图 3 - 9 2708 读数工作波形

读数时的工作信号波形如图3—9所示。当地址线选上时，经过160~330ns后片选信号 $\overline{CS}/\overline{WE}$ （片选/允许写入）变负。接着再经过120ns以后，数据输出线上出现有效数据。值得注意的是，片选线恢复为正常高电平以后，要经过120ns，数据输出线上的缓冲器才呈现浮离的高阻抗状态。其它部件必须在这个时间以后才能使用这条数据输出线。

当编写程序写入时，在程序输入端18加26V脉冲，使浮动栅中注入电荷。所加的每次脉冲宽度为0.6~1ms，每个晶体管上所加脉冲的总时间在100ms以上，例如，每次脉冲宽度为0.6ms，共加256次，则其总时间为153.6ms。在编写程序写入时，1024个地址必须全部顺序扫描。因此，若要加256次，则1024个地址需要扫描256遍，总共时间约需100多秒。对于16K位的EPROM INTEL 2716(2K×8)，其编写程序的方法与上相同，在这种情况下，写入信息的脉冲幅度为25V，每一个地址并行写8位。不过，写入脉冲宽度需要50ms，每个单元只需要接受写入脉冲作用一次就可以了，不必重复写多次。因此，对于2716的2K个地址，全部写一次共需时间约100秒。

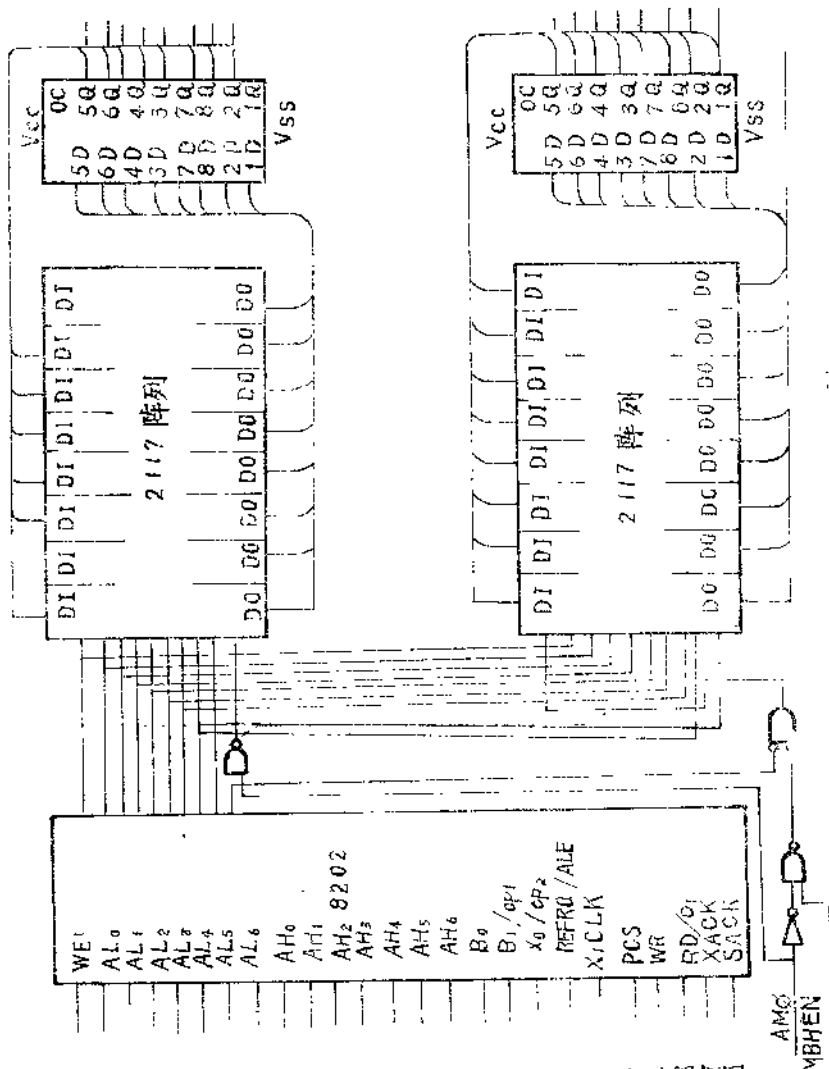


图3 - 10 32 K字节动态RAM存储器框图

3.2.3 存储器框图

图3—10是典型的存储器模块的框图,由8202动态RAM控制器、16片16384×1位的动态RAM存储器2117和两片8位的D-型锁存器74S373组成。16片的2117分成两组,每组就构成了一个16K的字节,所以总的存储容量达32K字节,8202是动态RAM控制器由它提供2117所需要的所有信号。74S373 8位锁存器实现数据的输入/输出锁存。

3.3 时钟产生/驱动器

下面我们介绍两种8080/8085、8086 CPU芯片和其它电路芯片所使用的时钟产生和驱动器。

3.3.1 8224时钟产生/驱动器

INTEL8224是8080/8085、8259总线仲裁器和8253可编程序内部计时器等芯片所使用的时钟产生/驱动器。8224是一个单片电路,受一个晶体的控制。设计者可以根据不同的用途来选择晶体,以满足各类系统对速度的要求。

1. 8224的逻辑结构

8224的逻辑框图及封装引线排列如图3—11所示。现将引线端做简要说明:

RESIN: 外部“复位”信号输入,低电平有效。

RESET: “复位”信号输出,高电平有效。

RDYIN: 外部“准备完毕”信号输入,高电平有效。

SYNC: “同步”信号输入,高电平有效。

STSTB: 周期选通信号,低电平有效。

ϕ_1 、 ϕ_2 : 8080时钟。

XTAL₁)
XTAL₂): 晶体联接端

TANK: 和XTAL配合使用,用于谐波振荡

OSC: 振荡器输出

ϕ_2 (TTL): ϕ_2 时钟, TTL电平输出

VCC: 电源+5V

VDD: 电源+12V

GND: 0V

2. 功能说明:

8224里面有一个受晶体控制的振荡器,一个九分频的计数器,两个高电平驱动器和几个辅助的逻辑电路。

振荡器用来为时钟发生器提供振荡信号。其振荡频率由外接的以串型谐振式连接的石英晶体来决定,选择晶体的频率时,应按所要求的时钟频率的九倍来确定。如8080A的时钟频率为2MC,则振荡器振荡信号的频率,也就是晶体的频率应为:

$$2MC \times 9 = 18MC$$

另一个振荡器的输入是TANK,这个输入允许8224使用谐波型晶体。这种类型的晶体通常比基础型具有低得多的“增益”,所以需一个附加的LC网络为专用的振荡器回路提供附加的“增益”。这个附加的LC网络连接到TANK输入端并且交流耦合接地,见图3—12。振荡器的输出经过缓冲,在OSC端输出。

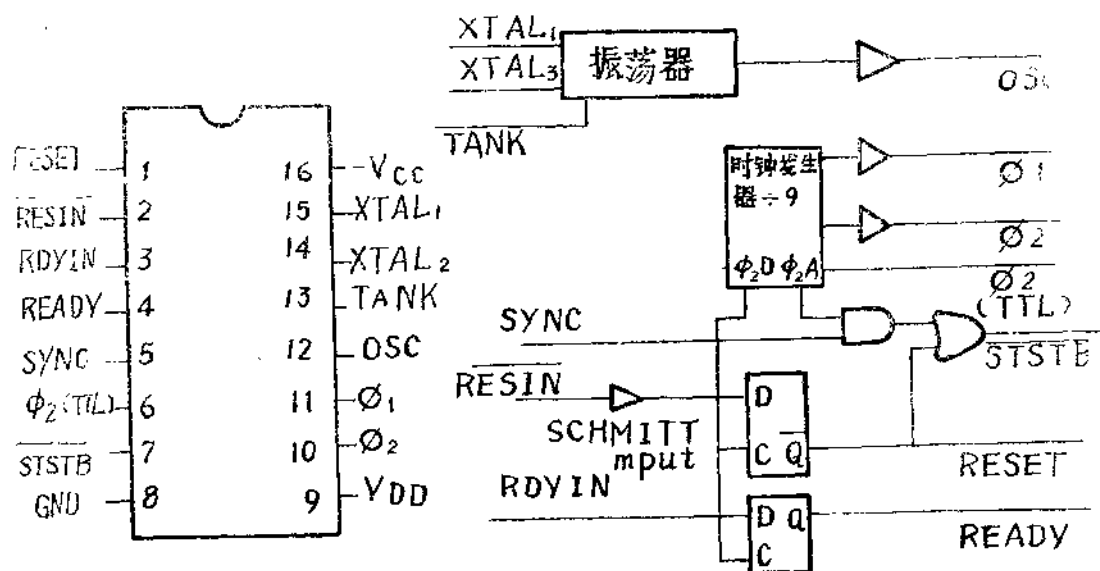


图3 - 11 8224 封装及框图

时钟发生器用于产生8080A的时钟信号 ϕ_1 和 ϕ_2 。九分频的计数器是时钟发生器的主要组成部分。

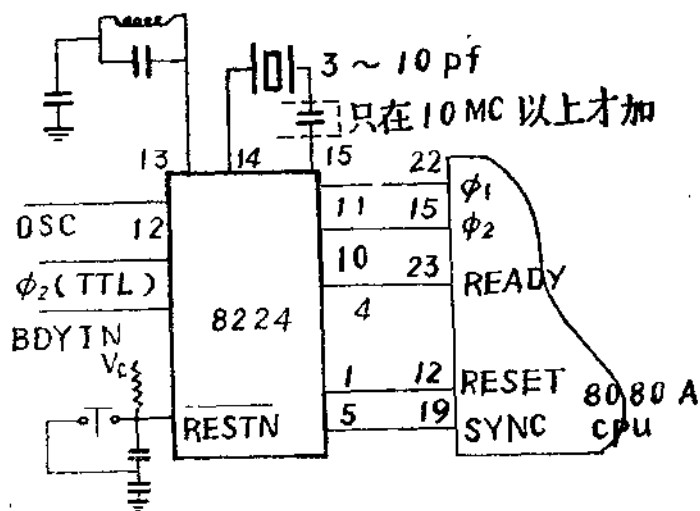


图3 - 12 8224 外部接线

内部还产生其它几个信号,以便得到辅助触发器的最佳定时和周期的选通 ($\overline{STSTB}$) 信号。 $\overline{STSTB}$ 周期选通信号在8080A CPU的每个机器周期开始,就在它的数据总线上发出周期信息,这个信息告诉在本机器周期将做什么动作。从CPU引来SYNC信号并用一个内部定时信号(ϕ_1 A)来作门控,就能得到一个低电平有效的选通信号,它发生在每个机器周期的起始时,当周期信息在总线上能稳定的尽可能早的瞬间。 $\overline{STSTB}$ 信号直接连到8228系统控制和总线驱动器上。

开启电源的复位信号(RESET)也能产生 $\overline{STSTB}$,但显然时间间隔较长。这样可使8228能自动复位而不需为此增加引线端。

连接在 $\overline{RESIN}$ 输入端的是一个外部的RC网络,电源上升边的慢速瞬变由内部的一个施密特触发器来监视,当输入电平到达预定值时,这个电路把它的慢速瞬变转换为快速跳变的边沿,施密特触发器的输出连接到D型触发器,此触发器用 ϕ_2 D(一个内部的定时信号)作为时钟输入,这个触发器使复位被同步并产生符合8080A输入规格的电平。对手动开关类型的系统复位线路,一个有效显低电平的开关闭合,被连接到 $\overline{RESIN}$ 输入端,附加在电源接通的RC网络上。

8224芯片信号输入输出以及联接情况见图3—12。

3.3.2 8284时钟产生/驱动器

8284是一种双极型的时钟产生与驱动器,可提供8086 CPU及其它外部设备所需的时钟信号,该器件还具有READY逻辑以供具有两个多重总线的系统操作,并能提供8086所需的READY同步化及时序信号。同时还提供了带有迟后作用及同步化的复位逻辑。

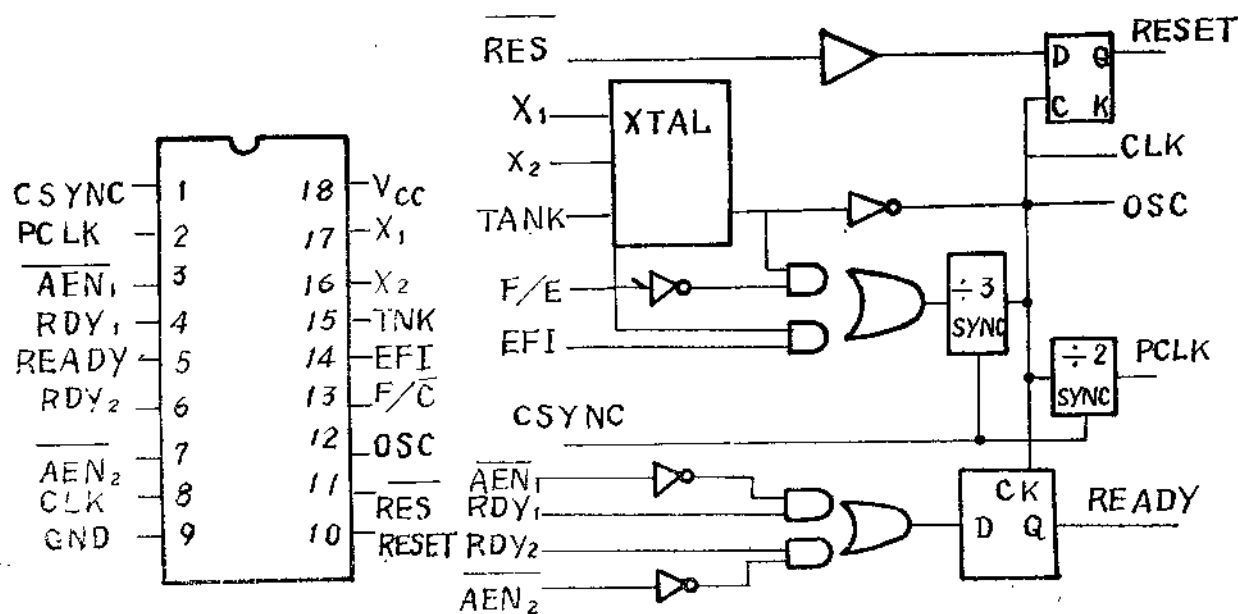


图3-13 8284封装与框图

下面给出8284的引脚名称(图3—13)

X_1 、 X_2 : 晶体联接端

TANK: 在用谐波振荡时与 X_1 、 X_2 配合使用

EFI: 外部时钟输入

CSYNC: 时钟同步化输入

RD1/1、RD1/2: 来自两个多重总线的准备好信号

$\overline{AEN}_1$ 、 $\overline{AEN}_2$: RD1/1、2的地址允许制约信号

RES: 复位输入

RESET: 同步复位输出

OSC: 振荡器输出

CLK: 8086用MOC时钟

PCLK: 外部设备用TTL时钟

READY: 同步准备好输出

Vcc: +5V

GND: 0V

下面介绍一下引脚的定义:

$\overline{AEN}_1$ 、 $\overline{AEN}_2$: 地址允许信号, 由外部提供, AEN是低电平有效, 用来制约相应的总线准备好信号RDY₁和RDY₂。 $\overline{AEN}_1$ 使RDY₁生效, 而 $\overline{AEN}_2$ 使RDY₂生效。在允许处理机访问两个多总线管理的系统总线使用这两个 $\overline{AEN}$ 信号。在非多总线管理的配置方式中, 此两 $\overline{AEN}$ 信号输入端接低电平。

RDY₁、RDY₂: 总线准备好信号。由外部提供, RDY是高电平有效的信号, 该信号来自系统数据总线上的某一个设备, 指示数据已经收到, 或表示数据已准备好可供使用。RDY₁受 $\overline{AEN}_1$ 制约。

READY: 准备好信号, 由8284输出, READY是高电平有效的信号, 它由输入信号RDY同步后得到。由于RDY是相对处理机时钟异步产生的, 因此在把它送到处理机之前应当使它同步, 在处理机所需的保持时间过后, READY被清除。

X_1 、 X_2 、TNK: X_1 和 X_2 是联接荡体的两个输入端。在使用谐波晶体时, TNK要接LC网络, 晶体频率应为处理机所需频率的三倍。

F/ $\overline{C}$: 当频率/晶体选择端, 是一个用接线来选择工作方式的输入端。当F/ $\overline{C}$ 接低电平时, 处理机所需的时钟由晶体产生。当F/ $\overline{C}$ 接高电平时, CLK由EFI输入产生。

EFI: 外加频率输入端。当F/C接高电平时, CLK由此端输入的频率产生, 输入的频率是方波信号, 为CLK输出频率的三倍。

CLK: 8284输出供处理机使用的时钟信号。CLK是处理机和其它直接与处理机局部总线相联的设备(即双极型支持芯片和其它MOS器件)所用的时钟信号的输出端。CLK的频率是晶体频率或EFI所输入频率的1/3, 占空度为1/3, 此端所提供的输出高电平为4.5V, 用来驱动MOS器件。

PCLK: 供外部设备用的时钟。PCLK是TTL电平信号。输出频率为CLK的1/2, 其占空度为50%。

OSC: 振荡器输出端。OSC是TTL电平的内部振荡器的输出, 其频率等于晶体的频率。