第 5 章　处理器总线时序和系统总线

5.1　8086 的引脚功能

8086 是一个双列直插式、40 个引脚的器件，它的引脚功能与系统的组态有关。

5.1.1　8086 的两种组态

当把 8086 CPU 与存储器和外设构成一个计算机的硬件系统时，根据所连接的存储器和外设的规模，8086 可以有最小和最大两种不同的组态。

目前常用的是最大组态，这要求有较强的驱动能力。此时，8086 要通过一组总线控制器 8288 来形成各种总线周期，控制信号由 8288 供给，如图 5-1 所示。

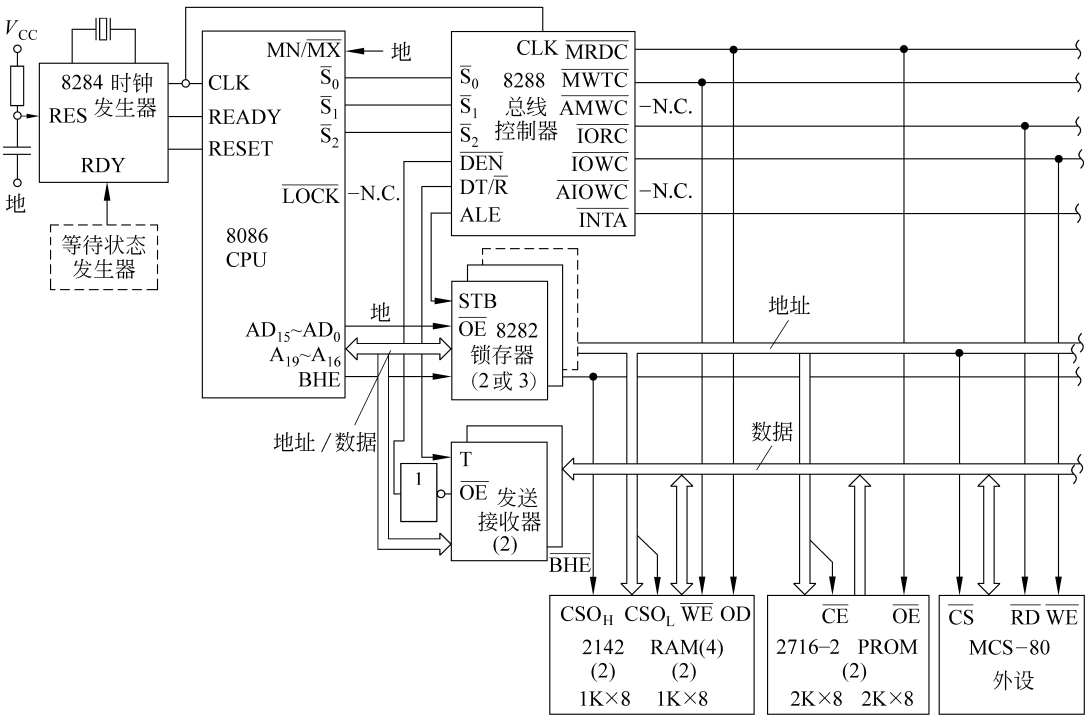

图 5-1　8086 的最大组态

在这两种组态下 8086 引脚中的引脚 24～引脚 31 有不同的名称和意义，所以需要有一个引脚 MN/$\overline{MX}$来规定 8086 处在什么组态。若把 MN/$\overline{MX}$引脚连至电源（+5V），则为最小组态；若把它接地，则 8086 处在最大组态。

当 8086 处在最大组态时，引脚 24～引脚 31 的含义如下。

- $\overline{S}_2$、$\overline{S}_1$、$\overline{S}_0$（输出，三态）

这些状态线的功能如表 5-1 所示。

表 5-1　最大组态下的总线周期

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	功　能
0(低)	0	0	中断响应
0	0	1	读 I/O 端口
0	1	0	写 I/O 端口
0	1	1	暂停(Halt)
1(高)	0	0	取指
1	0	1	读存储器
1	1	0	写存储器
1	1	1	无源

这些信号由 8288 总线控制器来产生有关存储器访问或 I/O 访问的总线周期和所需要的控制信号。

在时钟周期 T_4 状态期间，$\overline{S}_2$、$\overline{S}_1$、$\overline{S}_0$ 的任何变化，指示一个总线周期的开始；而它们在 T_3 或 T_W 期间返回到无源状态(111)，则表示一个总线周期的结束。当 CPU 处在 DMA 响应状态时，这些线浮空。

- $\overline{RQ}/\overline{GT}_0$，$\overline{RQ}/\overline{GT}_1$(输入/输出)

这些请求/允许(Request/Grant)引脚，是由外部的总线主设备请求总线并促使 CPU 在当前总线周期结束后让出总线用的。每一个引脚是双向的，$\overline{RQ}/\overline{GT}_0$ 比 $\overline{RQ}/\overline{GT}_1$ 有更高的优先权。这些线的内部有一个上拉电阻，所以允许这些引脚不连接。请求和允许的顺序如下：

① 由其他的总线主设备，输送一个宽度为一个时钟周期的脉冲给 8086，表示总线请求，相当于 HOLD 信号。

② CPU 在当前总线周期的 T_4 或下一个总线周期的 T_1 状态，输出一个宽度为一个时钟周期的脉冲给请求总线的设备，作为总线响应信号(相当于 HLDA 信号)，从下一个时钟周期开始，CPU 释放总线。

③ 当外设的 DMA 传送结束时，总线请求主设备输出一个宽度为一个时钟周期的脉冲给 CPU，表示总线请求的结束。于是 CPU 在下一个时钟周期开始又控制总线。

每一次总线主设备的改变，都需要这样的三个脉冲，脉冲为低电平有效。在两次总线请求之间，至少要有一个空时钟周期。

- $\overline{LOCK}$(输出，三态)

低电平有效，当其有效时，别的总线主设备不能获得对系统总线的控制。$\overline{LOCK}$ 信号由前缀指令 LOCK 使其有效，且在下一个指令完成以前保持有效。当 CPU 处在 DMA 响应状态时，此线浮空。

- QS_1、QS_0(输出)

QS_1 和 QS_0 提供一种状态(Queue Status)允许外部追踪 8086 内部的指令队列，如表 5-2 所示。

表 5-2　QS_1 和 QS_0 的功能

QS_1	QS_0	功　能
0(低)	0	无操作
0	1	从队列中取走操作码的第一个字节
1(高)	0	队列空
1	1	除第一个字节外,还取走队列中的其他字节

队列状态在 CLK 周期期间是有效的,在这以后队列的操作已完成。

- $\overline{BHE}$/ST(输出)

在总线周期的 T_1 状态,在$\overline{BHE}$/S_7 引脚输出$\overline{BHE}$信号,表示高 8 位数据线 AD_{15}～AD_8 上的数据有效;在 T_2、T_3、T_4 及 T_W 状态,$\overline{BHE}$/S_7 引脚输出状态信号 S_7。

5.1.2　8086 的引线

8086 的引线如图 5-2 所示。

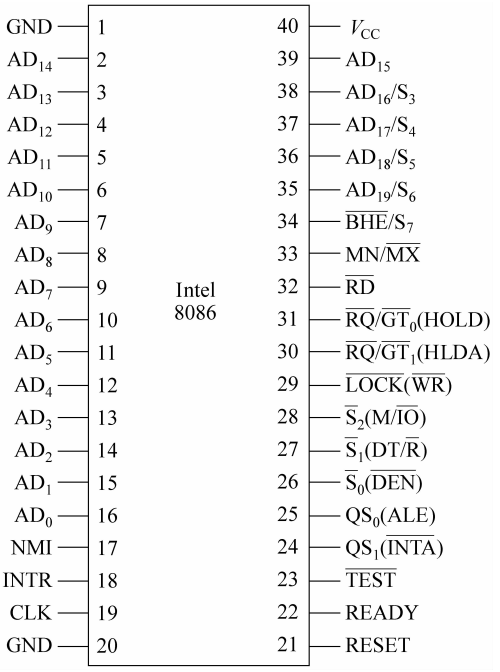

图 5-2　8086 的引脚

其中引脚 24～引脚 31 的含义已在前面介绍过了,在最小组态时的名称如图 5-2 的括号中所示。

- AD_{15}～AD_0(输入/输出,三态)

这些地址/数据引脚是多路开关的输出。由于 8086 只有 40 条引线,而它的地址线是 20 位,数据线是 16 位,因此 40 条引线的数量不能满足要求,于是在 CPU 内部用一些多路开关,使数据线与低 16 位地址线公用,从时间上加以区分。通常当 CPU 访问存储器或外设

时，先要给出所访问单元（或端口）的地址（在 T_1 状态），然后才是读写所需的数据（T_2、T_3、T_W 状态），它们在时间上是可区分的。只要在外部电路中有一个地址锁存器，把在这些线上先输出的 16 位地址锁存下来就可以了。在 DMA 方式时，这些引脚浮空。

- A_{19}/S_6、A_{18}/S_5、A_{17}/S_4、A_{16}/S_3（输出，三态）

这些引脚也是多路开关的输出，在存储器操作的总线周期的 T_1 状态时，这些引脚上是最高四位地址（也需要外部锁存）。在 I/O 操作时，这些地址不用，故在 T_1 状态时全为低电平。在存储器和 I/O 操作时，这些线又可以用来作为状态信息（在 T_2、T_3、T_W 状态时）。但 S_6 始终为低，表示 8086 当前与总线相连；S_5 是标志寄存器中中断允许标志的当前设置，它在每一个时钟周期开始时被修改；S_4 和 S_3 用以指示是哪一个段寄存器正在被使用，其编码如表 5-3 所示。

表 5-3　S_4、S_3 的功能

S_4	S_3	功　　能
0（低）	0	当前正在使用 ES
0	1	当前正在使用 SS
1	0	当前正在使用 CS，或者未用任何段寄存器
1	1	当前正在使用 DS

在 DMA 方式时，这些引脚浮空。

- $\overline{RD}$（输出，三态）

读选通信号，低电平有效。当其有效时，表示正在进行存储器读或 I/O 读。在 DMA 方式时，此引脚浮空。

- READY（输入）

准备就绪信号。这是从所寻址的存储器或 I/O 设备来的响应信号，高电平有效。当其有效时，将完成数据传送。CPU 在 T_3 周期的开始采样 READY 线，若其为低，则在 T_3 周期结束以后，插入 T_W 周期，直至 READY 变为有效。若 READY 有效，则在此 T_W 周期结束以后，进入 T_4 周期，完成数据传送。

- INTR（输入）

可屏蔽中断请求信号。这是一个电平触发输入信号，高电平有效。CPU 在每一个指令周期的最后一个 T 状态采样这条线，以决定是否进入中断响应周期。这条线上的请求信号，可以用软件复位内部的中断允许位来加以屏蔽。

- $\overline{TEST}$（输入）

这个检测输入信号是由 WAIT 指令来检查的。若此输入脚有效（低电平有效），则执行继续，否则处理器就等待进入空转状态。这个信号在每一个时钟周期的上升沿由内部同步。

- NMI（输入）

非屏蔽中断输入信号（Non-Maskable Interrupt），是一个边沿触发信号。这条线上的中断请求信号不能用软件来加以屏蔽，所以这条线上由低到高的变化，就在当前指令结束以后引起中断。

• RESET(输入)

复位输入引起处理器立即结束当前操作。这个信号必须保持有效(高电平)至少 4 个时钟周期,以完成内部的复位过程。当其返回为低电平时,它重新启动执行。

• CLK(输入)

时钟输入信号。它提供了处理器和总线控制器的定时操作。8086 的标准时钟频率为 8MHz。

• V_{CC}是 5V±10%的电源引脚。

• GND 是接地线。

5.2 8086 处理器时序

5.2.1 时序的基本概念

计算机的工作是在时钟脉冲 CLK 的统一控制下,一个节拍一个节拍地实现的。在 CPU 执行某一个程序之前,先要把程序(已变为可执行的目标程序)放到存储器的某个区域。在启动执行后,CPU 就发生读指令的命令;存储器接到这个命令后,从指定的地址(在 8086 中由代码段寄存器 CS 和指令指针 IP 给定)读出指令,把它送至 CPU 的指令寄存器中;CPU 对读出指令经过译码器分析之后,发出一系列控制信号,以执行指令规定的全部操作,控制各种信息在机器(或系统)各部件之间传送。简单地说,每条指令的执行由取指令(fetch)、译码(decode)和执行(execute)构成。对于 8086 微处理器来说,每条指令的执行有取指、译码、执行这样的阶段,但由于微处理器内有总线接口单元(BIU)和执行单元(EU),所以在执行一条指令的同时(在 EU 中操作),BIU 就可以取下一条指令,它们在时间上是重叠的。所以,从总体上来说,似乎不存在取指阶段。这种功能就称为“流水线”功能。目前,在高档微处理器中往往有多条流水线,使微处理器的许多内部操作“并行”进行,从而大大提高了微处理器的工作速度。

上述的这些操作以及执行一条指令的一系列动作,都是在时钟脉冲 CLK 的统一控制下一步一步进行的。它们都需要一定的时间(当然有些操作在时间上是重叠的)。如何确定执行一条指令所需要的时间呢?

执行一条指令所需要的时间称为指令周期(Instruction Cycle)。但是,8086 中不同指令的指令周期是不等长的。因为,首先 8086 的指令是不等长的,最短的指令是一个字节,大部分指令是两个字节,但由于各种不同寻址方式又可能要附加几个字节,8086 中最长的指令要 6 个字节。指令的最短执行时间是两个时钟周期,一般的加、减、比较、逻辑操作是几十个时钟周期,最长的为 16 位数乘除法大约需要 200 个时钟周期。

指令周期又分为一个个总线周期。每当 CPU 要从存储器或 I/O 端口读写一个字节(或字)就是一个总线周期(Bus Cycle)。所以,对于多字节指令,取指就需要若干个总线周期(当然,对于 8086 来说,取指可能与执行前面的指令在时间上有一定的重叠);在指令的执行阶段,不同的指令也会有不同的总线周期,有的只需要一个总线周期,而有的可能需要若干个总线周期。一个基本的总线周期的时序如图 5-3 所示。

每个总线周期通常包含 4 个 T 状态(T State),即图 5-3 中的 T_1、T_2、T_3、T_4,每个 T 状态是 8086 中处理动作的最小单位,它就是时钟周期(Clock Cycle)。早期的 8086 的时钟频

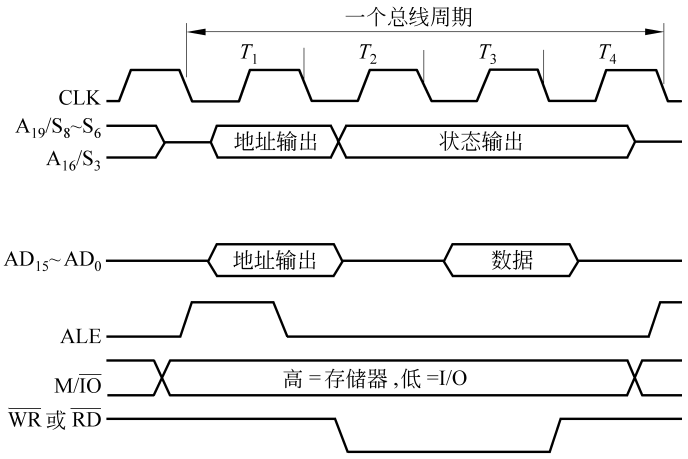

图 5-3　一个基本的指令周期时序图

率为 8MHz,故一个时钟周期或一个 T 状态为 125ns。

虽然各条指令的指令周期有很大差别,但它们仍然是由以下一些基本的总线周期组成的:

① 存储器读或写;

② 输入输出端口的读或写;

③ 中断响应。

如上所述,8086 CPU 的每条指令都有自己的固定的时序。例如,从存储器读一个字节(或字)的总线周期是由 4 个 T 状态组成的,如图 5-4 所示。

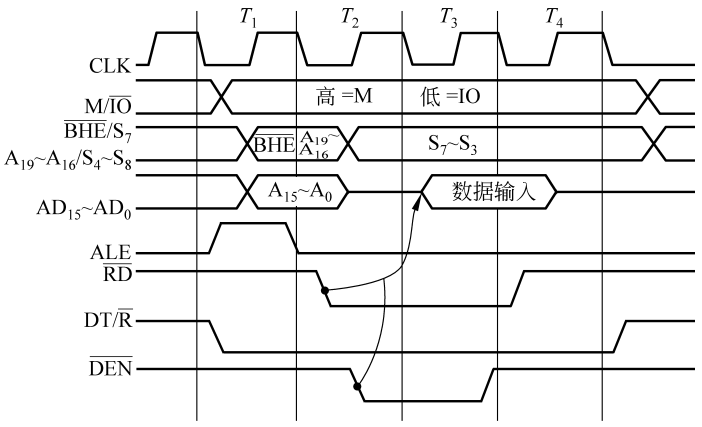

图 5-4　存储器读周期时序

CPU 希望能在 4 个 T 状态时间内,把存储单元的信息读出来,在 T_1 周期开始后一段时间(在 T_1 状态)把地址信息从地址线 A_{19}~A_{16}、AD_{15}~AD_0 上输出,且立即发出地址锁存信号 ALE,把在 A_{19}~A_{16} 上出现的高 4 位地址和在 AD_{15}~AD_0 上出现的低 16 位地址,在外部地址锁存器上锁存。这样,20 位地址信息就送至存储器。CPU 也是在 T_1 状态发出区分是存储器还是 I/O 操作的 M/$\overline{IO}$信号。在 T_2 状态,CPU 发出读命令信号(若使用接口芯片

8286，还有相应的控制信号 DT/$\overline{R}$ 和$\overline{DEN}$）。有了这些控制信号，存储器就可以实现读出操作（详见下面的有关存储器的分析）。在这些信号发出后，CPU 等待一段时间，到它的 T_4 状态的前沿（下降沿）采样数据总线 AD_{15}～AD_0 获取数据，从而结束此总线周期。这是 CPU 在执行存储器读时的时序，至于存储器在接到地址和读命令信号后，能否在 T_4 的前沿把数据读出送到数据总线，这是存储器本身的读写时间问题。

实际上存储器从接收到地址信号，要经过地址译码选择，选中所需要的单元；I/O 端口也是如此。从接收到 M/$\overline{IO}$信号和$\overline{RD}$信号（这些信号一般用作选通信号），到信息从被选中的单元读出送至数据总线也都是需要一定时间的，它是否能在 T_4 周期的前沿前完成，这完全取决于存储电路本身。所以，在 CPU 的时序和存储器或 I/O 端口的时序之间存在配合问题。

这个问题在早期的计算机设计中，是在设计 CPU 和存储器以及外设时协调解决的，因为当时 CPU 和存储器是统一设计的。而随着大规模集成电路生产的发展，以及计算机的专业化、产业化的发展，CPU 和存储器的生产企业都是一种大规模的系列化、标准化的企业。所以，在构成一个计算机硬件系统时，硬件系统的设计者要解决 CPU 的时序与存储器或 I/O 端口的时序之间的配合问题。

为解决此问题，在 CPU 中就设计了一条准备就绪——READY 输入线，这是由存储器或 I/O 端口输送给 CPU 的状态信号线在存储器或 I/O 端口对数据的读写操作完成时，使 READY 线有效（即为高电平）。CPU 在 T_3 状态的前沿（下降沿）采样 READY 线，若其有效，则为正常周期，在 T_3 状态结束后进入 T_4 状态，且 CPU 在 T_4 状态的前沿采样数据总线，完成一个读写周期；若 CPU 在 T_3 状态的前沿采样到 READY 为无效（低电平），则在 T_3 周期结束后，进入 T_W 周期（等待周期），且在 T_W 周期的前沿采样 READY 线，只要其为无效，就继续进入下一个 T_W 周期，直至在某一个 T_W 周期的前沿采样到 READY 为有效时，则在此 T_W 周期结束时进入 T_4 周期，在 T_4 状态的前沿采样数据线，完成一个读写周期，其过程如图 5-5 所示。

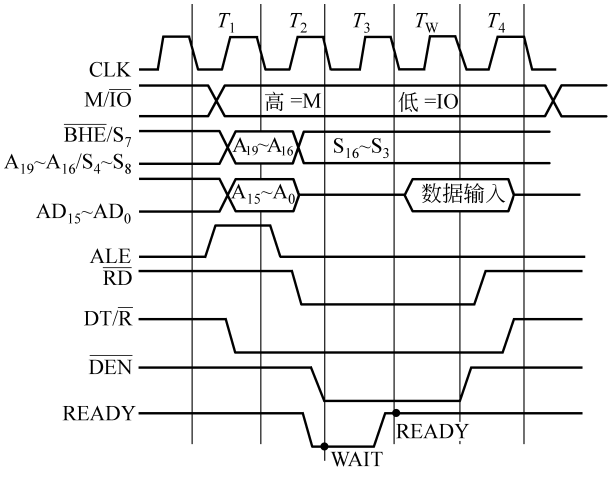

图 5-5　具有 T_W 状态的存储器读周期

因此，在设计系统的硬件电路时，要根据 CPU 与所选的存储器的读写速度，分析能否在时序上很好地配合，若需要插入 T_W 周期，就要设计一个硬件电路来产生适当的 READY 信号。

有了 READY 信号线，就可以使 CPU 与任何速度的存储器相连接（当然存储器的速度还是要由系统的要求来选定）。但是，这说明了当 CPU 与存储器或 I/O 端口连接时，要考虑相互之间的时序配合问题。

目前，在高档 CPU 的微型计算机（例如 386、486 以上的微型计算机）中，常有无等待（0 等待）、1 等待、2 等待等指标，就是指当 CPU 读写存储器时，是否需要插入以及插入多少个等待周期。

插入了 T_W状态，改变了指令的时钟周期数，使系统的速度变慢。若系统中使用了动态存储器（目前的系统中大量采用），则它要求周期性地进行刷新，所以，能插入的 T_W数也是有限制的。

5.2.2 8086 的典型时序

目前在构成微型计算机硬件系统时，所连接的存储器和 I/O 接口电路的数量较多，8086 微处理器通常工作在最大组态；下面所介绍的时序就是以 8086 工作在最大组态为基础的。

在最大组态下，8086 的基本总线周期由 4 个 T 状态组成。在 T_1状态时，8086 发出 20 位地址信号，同时送出状态信号 $\overline{S}_2$、$\overline{S}_1$、$\overline{S}_0$给 8288 总线控制器。8288 对 $\overline{S}_2 \sim \overline{S}_0$进行译码，产生相应的命令的控制信号输出。首先，8288 在 T_1期间送出地址锁存允许信号 ALE，将 CPU 输出的地址信息锁存至地址锁存器中，再输出到系统地址总线上。

在 T_2状态，8086 开始执行数据传送操作。此时，8086 内部的多路开关进行切换，将地址/数据线 $AD_{15} \sim AD_0$上的地址撤销，切换为数据总线，为读写数据作准备。8288 发出数据总线允许信号和数据发送/接收控制信号 DT/$\overline{R}$ 允许数据收发器工作，使数据总线与 8086 的数据线接通，并控制数据传送的方向。同样，把地址/状态线 $A_{19}/S_6 \sim A_{16}/S_3$切换成与总线周期有关的状态信息，指示若干与周期有关的情况。

在 T_3周期开始的时钟下降沿上，8086 采样 READY 线。如果 READY 信号有效（高电平），则在 T_3状态结束后进入 T_4状态，在 T_4状态开始的时钟下降沿，把数据总线上的数据读入 CPU 或写到地址选中的单元。在 T_4状态中结束总线周期。如果访问的是慢速存储器或是外设接口，则应该在 T_1状态输出的地址，经过译码选中某个单元或设备后，立即驱动 READY 信号到低电平。8086 在 T_3状态采样到 READY 信号无效，就会插入等待周期 T_W，在 T_W状态 CPU 继续采样 READY 信号；直至其变为有效后再进入 T_4状态，完成数据传送，结束总线周期。

在 T_4状态，8086 完成数据传送，状态信号 $\overline{S}_0 \sim \overline{S}_2$变为无操作的过渡状态。在此期间，8086 结束总线周期，恢复各信号线的初态，准备执行下一个总线周期。

1. 存储器读周期和存储器写周期

存储器读写周期由 4 个时钟组成，即使用 T_1、T_2、T_3和 T_4共 4 个状态。

对存储器读周期，在 T_1开始，8086 发出 20 位地址信息和 $\overline{S}_2 \sim \overline{S}_0$状态信息。8288 对 $\overline{S}_2 \sim \overline{S}_0$进行译码，发出 ALE 信号将地址锁存；同时判断为读操作，DT/$\overline{R}$ 信号输出为低电平。在 T_2期间，8086 将 $AD_{15} \sim AD_0$切换为数据总线，8288 发出读存储器命令$\overline{MRDC}$，此命令使地址选中的存储单元把数据送上数据总线；然后输出信号 DEN 有效（相位与最小模式下相反），接通数据收发器，允许数据输入至 8086。在 T_3状态开始时，8086 采样 READY，

当 READY 有效时，进入 T_4 状态，8086 读取在数据总线上的数据，到此读操作结束。在 T_4 之前时钟周期的时钟信号的上升沿，8086 就发出过渡的状态信息（$\overline{S}_2 \sim \overline{S}_0$ 为 111），使各信号在 T_4 期间恢复初态，准备执行下一个总线周期。存储器读周期的时序如图 5-6 所示。

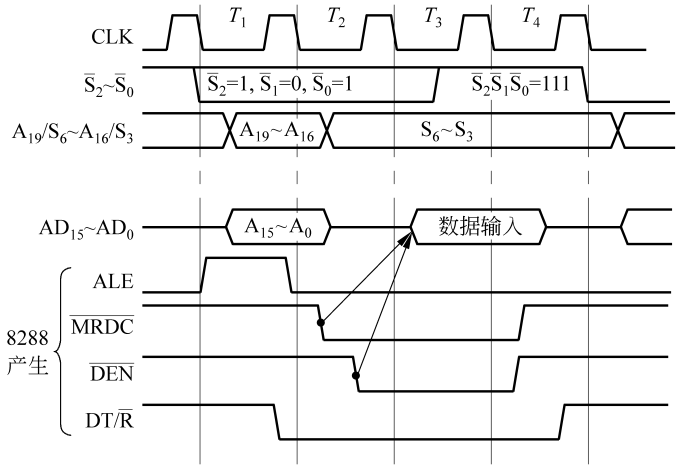

图 5-6　最大组态时存储器读周期时序

对于存储器写周期，大部分过程与读周期类似，但执行的是写操作。T_1 期间 8086 发出 20 位地址信息和 $\overline{S}_2 \sim \overline{S}_0$，8288 判断为写操作，则 DT/$\overline{R}$ 信号变为高电平。在 T_2 开始，8288 输出写命令$\overline{AMWC}$，命令存储器把数据写入选中的地址单元；同时$\overline{DEN}$信号有效，使 8086 输出的数据马上经数据收发器送到数据总线上。T_3 开始，采样 READY 线，当 READY 为高电平后，进入 T_4 状态，结束存储器写周期。存储器写周期的时序如图 5-7 所示。

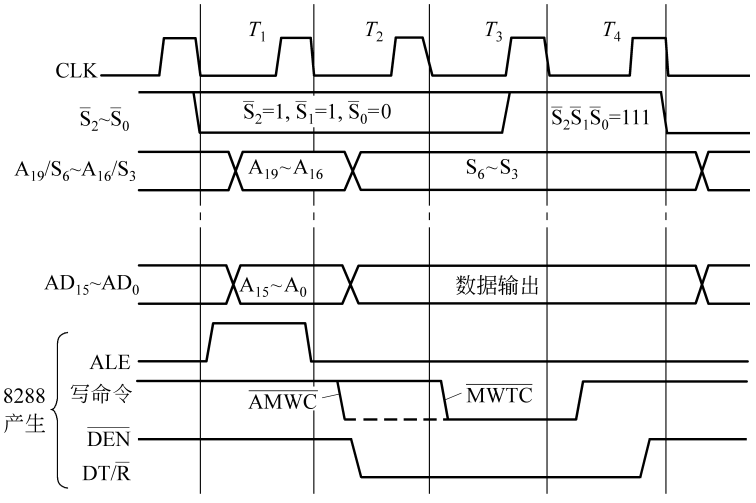

图 5-7　最大组态时存储器写周期时序

由图 5-7 中可看到在存储器写周期，8288 有两种写命令信号：存储器写命令$\overline{MWTC}$和提前写命令$\overline{AMWC}$，这两个信号大约差 200ns。

2. I/O 读和 I/O 写周期

8086 的基本 I/O 总线周期时序与存储器读写的时序是类似的。但通常 I/O 接口电路的工作速度较慢，往往要插入等待状态。例如在 IBM PC/XT 的 READY 信号设计在 I/O 操作时，要求插入一个 T_W 状态。即在 PC/TX 中，基本的 I/O 操作是由 T_1、T_2、T_3、T_W、T_4 组成，占用 5 个时钟周期。

这样的 I/O 读写周期和存储器读写周期的时序基本相同，不同之处为：

- T_1 期间 8086 发出 16 位地址信息，A_{19}～A_{16} 为 0。同时 $\overline{S}_2$～$\overline{S}_0$ 的编码为 I/O 操作。
- 在 T_3 时采样到的 READY 为低电平，插入一个 T_W 状态。
- 8288 发出的读写命令为 $\overline{IORC}$ 和 $\overline{AIOWC}$（$\overline{IOWC}$ 未用）。

I/O 读和 I/O 写周期的时序如图 5-8 所示。

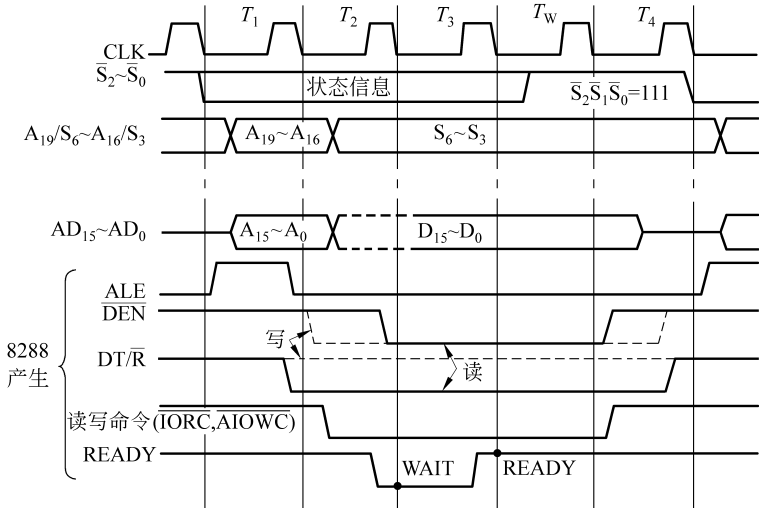

图 5-8　最大组态时的 I/O 读写时序

3. 空闲周期

若 CPU 不执行总线周期（不进行存储器或 I/O 操作），则总线接口执行空闲周期（一系列的 T_1 状态）。在这些空闲周期，CPU 在高位地址线上仍然驱动上一个机器周期的状态信息。

若上一个总线周期是写周期，则在空转状态，CPU 在 AD_{15}～AD_0 上仍输出上一个总线周期要写的数据，直至下一个总线周期的开始。

在这些空转周期，CPU 进行内部操作。

4. 中断响应周期

当外部中断源，通过 INTR 或 NMI 引线向 CPU 发出中断请求信号，若是 INTR 线上的信号，则只有在标志位 IF=1（即 CPU 处在开中断）的条件下，CPU 才会响应。CPU 在当前指令执行完以后，响应中断。在响应中断时，CPU 执行两个连续的中断响应周期，如图 5-9 所示。

在每一个中断响应周期，CPU 都输出中断响应信号 $\overline{INTA}$。在第一个中断响应周期，CPU 使 AD_{15}～AD_0 浮空。在第二个中断响应周期，被响应的外设（或接口芯片），应向数据总线输送一个字节的中断向量号，CPU 读入中断向量号后，就可以在中断向量表上找到该设备的服务程序的入口地址，转入中断服务。