



第 5 章

I/O 接口中的定时/计数技术

无论何种计算机，其工作都是在标准的时钟脉冲控制下完成的，因此，定时电路是任何计算机系统中必不可少的基本电路。尤其是在实时计算机测控系统中，常需要定时/计数电路，为模拟 I/O 通道和其他 I/O 设备提供实时时钟，以实现定时中断、定时检测、定时扫描、定时显示等定时或延时控制，或者对外部事件进行计数等。可见，定时/计数是实现计算机测控系统或其他计算机应用系统的一项重要技术。本章将在介绍实现定时的一般方法的基础上，重点介绍可编程定时器/计数器 8254 及应用。

5.1 实现定时的一般方法

实现定时或延时控制的方法通常有三种，即软件定时、不可编程硬件定时和可编程硬件定时。

1. 软件定时

软件定时就是通过执行一段固定的循环程序来实现定时。例如，执行下列循环程序：

```

MOV  CX, 0FFFH
LOP:  NOP
      NOP
      NOP
      LOOP LOP

```

} 循环体，时间固定

由于 CPU 执行每条指令都需要一定的时间，因此执行一个固定的程序段（循环体）就需要一个固定的时间，而控制循环体的执行次数即可方便地控制定时或延时时间的长短。这种软件定时方式实现起来很简单，但要占用大量 CPU 时间，会降低 CPU 的利用率，且定时精度不高。所以，软件定时主要适合于定时时间不长，精度要求不高的场合。

2. 不可编程硬件定时

不可编程硬件定时是采用中小规模集成电路器件来构成定时电路。例如利用单稳触发器、555 定时器等常见的定时器件外接电阻和电容构成定时电路；也可利用加法或减法计数器对周期一定的时钟脉冲计数来实现定时或延时控制。这种硬件定时方案的优点是不占用 CPU 时间，电路也较简单；缺点是灵活性较差，且电路一经连接好后，定时值就不便控制和改变。

3. 可编程硬件定时

可编程硬件定时就是在上述不可编程硬件定时的基础上加以改进，使其在不改变硬件的情况下，可通过软件编程改变其定时值来控制定时或延时时间。可编程定时电路一般都是用可编程计数器来实现，因为它既可计数又可定时，故称之为可编程定时器/计数器电路。

可编程定时器/计数器电路的典型结构如图 5.1 所示。控制字寄存器和计数初值寄存器分别用于设置计数器的工作方式和控制定时时间。计数工作单元是一个减法计数器，在门控信号 GATE 允许（或启动）计数的条件下，对 CLK 输入端的时钟脉冲进行减法计数，计数结束通过 OUT 端输出相应波形，用作中断请求或其他定时信号。输出锁存器用于锁存计数器的当前计数值，以供 CPU 读取。状态寄存器用于寄存计数器的现行工作状态，CPU 可通过读状态锁存器获取计数器的工作状态。

目前，各种微机和微机系统中大多采用可编程定时器/计数器来满足计数和定时、延时控制的需要。如各种 PC 系列机中普遍采用的是 Intel 公司的 8253/8254 定时器/计数器芯片，因此本章重点介绍它和它的应用。

5.2 可编程定时器/计数器 8254

8253/8254 是 Intel 公司为解决与微处理器设计有关的公共时间问题而专门设计的可编程间隔定时器（Programmable Interval Timer, PIT）。8254 是 8253 的改进型，两者的基本功能相同，硬件组成、外部引脚和编程特性完全兼容，但 8254 比 8253 具有更优良的性能，下面主要以 8254 为例

介绍其功能、结构、引脚信号、工作方式和编程，而对 8254 与 8253 不同的地方随时予以说明。

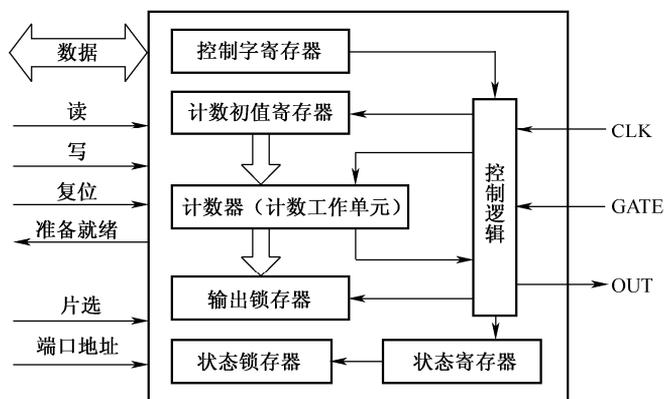


图 5.1 可编程定时器/计数器典型结构框图

5.2.1 基本功能

8254 具有以下基本功能：

- 1) 有 3 个独立的 16 位计数器通道。
- 2) 每个计数器可按二进制或十进制（BCD）计数。
- 3) 每个计数器可工作于 6 种不同工作方式。
- 4) 每个计数器允许的最高计数频率为 10MHz（8253 为 2MHz，8253-5 为 5MHz）。
- 5) 有读回命令（8253 没有），可以读出当前计数单元的内容和状态寄存器内容。
- 6) 每个计数器通道的逻辑功能完全相同，既可作定时器用，又可作计数器用。

5.2.2 内部结构与外部引脚

8254 是一个具有 24 个引脚的双列直插式芯片，其内部结构与外部引脚如图 5.2 所示。从图可见，8254 芯片由与微处理器的接口、控制字寄存器和 3 个独立的计数器通道组成。

1. 与微处理器的接口

这部分电路由数据总线缓冲器和读/写控制逻辑组成。

数据总线缓冲器是一个 8 位双向三态缓冲器，8 位数据线 $D_7 \sim D_0$ 与 CPU 的系统数据总线连接，构成 CPU 与 8254 之间的数据传送通道。CPU 对 8254 发出的控制命令、计数初值，以及从 8254 读出的计数值或状态信息，都是通过这个缓冲器传送的。

读/写控制逻辑接收 CPU 系统总线的读、写控制信号和端口选择信号，用于控制 8254 内部寄存器的读/写操作。5 根控制线功能如下：

\overline{WR} ：写信号，输入，低电平有效。当 CPU 对 8254 执行写操作时，该信号有效。

\overline{RD} ：读信号，输入，低电平有效。当 CPU 对 8254 执行读操作时，该信号有效。

\overline{CS} ：片选信号，输入，低电平有效。 \overline{CS} 有效时 8254 被选中。

A_1A_0 ：端口选择信号，由 8254 片内译码，选择控制寄存器和 3 个计数器通道。

这些信号线与 CPU 的连接方法和 8255A、8259A 等可编程接口芯片与 CPU 的连接特性类似，

也是数据线 $D_0 \sim D_7$ 与 CPU 的数据总线直接相连; 端口选择线 A_1 和 A_0 与 CPU 的低位地址线直接相连; 片选线 \overline{CS} 则与余下 CPU 高位地址线经译码后产生的片选信号相连; 读/写控制线 $\overline{RD}/\overline{WR}$ 与 CPU 控制总线组合形成的读/写控制信号 (如 \overline{IOR} 和 \overline{IOW}) 直连。

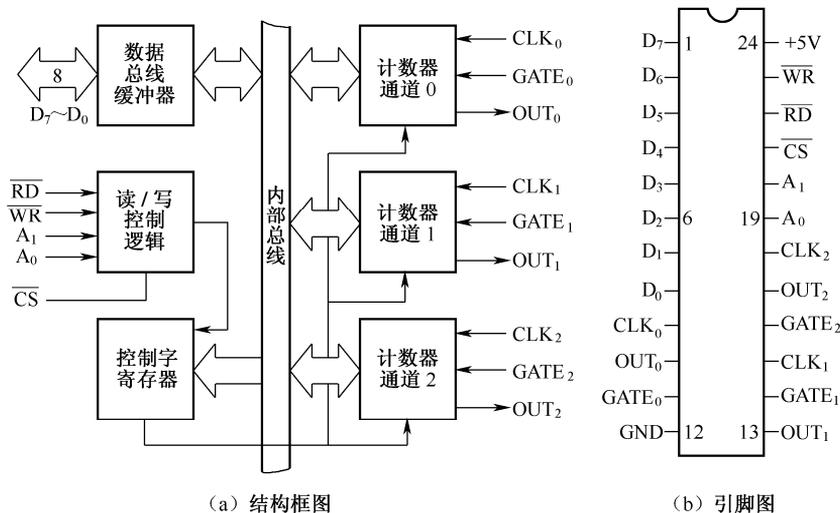


图 5.2 8254 的内部结构与引脚排列

2. 控制字寄存器

控制字寄存器与 3 个计数器通道通过内部总线相连, 由 CPU 写入控制字规定各计数器通道的工作方式或控制计数值和状态值的锁存。

3. 计数器通道

3 个计数器通道相互间是完全独立的, 但结构和功能完全相同。每个计数器通道的内部结构与图 5.1 给出的可编程定时器/计数器典型结构类似, 均包含一个 16 位的计数初值寄存器 (CR)、一个 16 位的减法计数工作单元 (CE)、一个 16 位的输出锁存器 (OL)、一个 8 位的状态寄存器 (SR) 和一个 8 位的状态锁存器 (SL) 以及相关的控制逻辑。

计数器通道的基本工作原理是: 首先写入控制字, 设置计数器的工作方式, 同时清除计数初值寄存器的内容。然后预置计数初值寄存器 (CR), 把计数初值传送至计数工作单元 (CE), 计数脉冲经 CLK 输入端对计数器进行减法计数。

每个通道都有 3 个和外界联系的引脚信号: OUT、CLK 和 GATE。OUT 引脚是计数器的输出信号端, 对于 6 种不同的工作方式, OUT 输出不同的波形。CLK 引脚用来引入基准时钟或外部事件脉冲信号。门控信号 GATE 有多种控制作用, 如允许/禁止计数、启动/中止计数等, 其作用因工作方式不同而异。

每个计数器通道的逻辑功能完全相同, 既可作定时器用, 又可作计数器用。当用作计数器时, 应将要求计数的次数预置到该通道的 CR 中, 被计数的事件应以脉冲方式从 CLK_i 端输入, 每输入一个计数脉冲, 计数器内容减 1, 当减至 0 时, OUT_i 端将有信号输出, 表示计数次数到。当用作定时器时, 由 CLK_i 端输入一定周期的时钟脉冲, 同时根据定时的时间长短确定所需的计数值, 并预置到 CR 中, 每输入一个时钟脉冲, 计数器内容减 1, 待计数值减到 0 时, OUT_i 端将有输出, 表示定时时间到。



可见,任一通道无论作计数器用或作定时器用,其内部操作完全相同,区别仅在于前者是由计数脉冲(间隔不一定相同)进行减 1 计数,而后者是由周期一定的时钟脉冲作减 1 计数。

5.2.3 内部端口寻址与读/写控制

8254 内部有 4 个 I/O 端口,分别对应于 A_1A_0 的 00、01、10、11 四种状态,实际中通常将它们分别称为 0 口、1 口、2 口和 3 口。CPU 对 3 个计数器通道和控制字寄存器的寻址及读/写操作是通过对这 4 个端口的寻址和读/写来实现的,有关控制作用如表 5.1 所示。

表 5.1 8254 各端口寄存器的寻址和读/写控制作用

\overline{CS}	\overline{RD}	\overline{WR}	A_1	A_0	读/写操作说明
0	1	0	0	0	写计数器通道 0 的 CR
0	1	0	0	1	写计数器通道 1 的 CR
0	1	0	1	0	写计数器通道 2 的 CR
0	1	0	1	1	写控制寄存器
0	0	1	0	0	读通道 0 的 OL 或状态锁存器
0	0	1	0	1	读通道 1 的 OL 或状态锁存器
0	0	1	1	0	读通道 2 的 OL 或状态锁存器
0	0	1	1	1	无操作
1	×	×	×	×	禁止使用
0	1	1	×	×	无操作

需要注意的是,由于计数工作单元(CE)和计数初值寄存器(CR)、输出锁存器(OL)均为 16 位,而内部总线的宽度为 8 位,因此 CR 的写入和 OL 的读出都必须分两次进行。若在初始化时只写入 CR 的一个字节,则另一个字节的内容保持为 0。CE 是 CPU 不能直接读/写的,需要修改其初值时,只能通过写入 CR 实现;需要读 CE 的当前内容时,必须先写入读回命令,将 CE 的内容锁存于 OL,然后再读出 OL 内容。经锁存后的 OL 内容将一直保持至 CPU 读出时为止。在 CPU 读出 OL 之后,OL 又跟随 CE 变化。状态寄存器也是 CPU 不能直接读的,同样必须先锁存到状态锁存器,才允许 CPU 读取(8253 中没有状态寄存器和状态锁存器,这是 8254 和 8253 的主要区别之一)。

5.2.4 六种工作方式

8254/8253 中各计数器通道均有 6 种工作方式可供选择。不同方式的工作特点是不相同的,理解 6 种工作方式时,要重点注意门控信号 GATE 的功能,输出信号 OUT 的波形,以及启动计数的条件。据此学会为不同的应用场合选择合适的工作方式。

下面分别介绍各种工作方式。在说明各种方式的波形图中,一律假定已经写入了控制字,通道已经进入了相应的工作方式,波形全部从写初值至 CR(用 n 值表示)开始画起。因为各波形图中的 n 值都很小,所以只需写一次初值的低字节,而不必写其高字节 00H。

1. 方式 0——计数结束中断方式(Interrupt On Terminal Count)

方式 0 是典型的事件计数用法,当计数单元 CE 计至 0 时,OUT 信号由低变高,可作为中断



请求信号。

其工作过程是：当写入控制字后，OUT 信号变为低电平，并维持低电平至 CE 的内容到达 0，此后 OUT 信号变为高电平，并维持高电平至再次写入新的计数值或重新写入控制字。门控信号 GATE 用于开放或禁止计数，GATE 为“1”允许计数，为“0”则停止计数。

方式 0 的工作特点可用图 5.3 所示的波形图来表示：

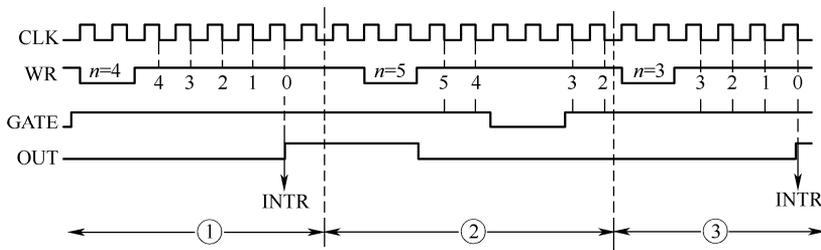


图 5.3 工作方式 0 定时波形图

1) 图中①表示 GATE=1 的正常计数情况。计数由软件启动，每次写入计数初值，只启动一次计数。当计数到 0 后，并不恢复计数初值，也不重新开始计数，OUT 端保持高电平。只有再次写入计数值后，OUT 变低，才开始新一轮的计数。CPU 写计数初值到 CR 后，CR 内容并不立即装入 CE，而是在其后的下一个 CLK 脉冲下降沿才将 CR 内容装入 CE，并从此开始作减 1 计数，因此，若计数初值为 n，则必须在出现 n+1 个 CLK 脉冲之后，OUT 信号才变为高电平。

2) 图中②表示计数过程中 GATE=0 时暂停计数的情况。在计数过程中，如果 GATE=0，则暂停计数，直到 GATE 变为 1 后再接着计数。

3) 图中③表示计数过程中重写了新的初值的计数情况。在计数过程中可写入新的计数初值。从写入后的下一个时钟脉冲开始，以新的初值计数。如是 8 位计数，在写入新的初值（仅低字节）后即按新值开始计数；如是 16 位计数，则在写入第一字节后，计数器停止计数，在写入第二字节后，计数器才按新值开始计数。

2. 方式 1——硬件可重触发单稳方式（Hardware Retriggerable One-Shot）

在方式 1 下，计数器相当于一个可编程的单稳态电路，触发输入为 GATE 信号。

其工作过程是：当写入控制字后，OUT 信号变为高电平，并保持不变；写入计数初值后，CR 内容并不装入 CE，也不开始计数过程，必须当 GATE 信号由低变高（上升沿起触发作用）之后的一个 CLK 脉冲出现，才将 CR 内容装入 CE，同时使 OUT 信号变为低电平，从而形成输出单脉冲的前沿。OUT 信号在 CE 不为 0 时一直保持低电平，当 CE 到达 0 时，OUT 信号恢复为高电平，形成输出单脉冲的后沿。因此，由 OUT 端输出的单脉冲宽度为 CLK 脉冲周期的 n 倍（n 为计数初值）。

方式 1 的工作特点可用图 5.4 所示的波形图来表示：

1) 图中①为一般情况。设定 CR 的初值为 n=3，于是输出负脉冲宽度为 3 个 CLK 周期。

2) 图中②为重触发情况。一旦控制字设置为方式 1 并向 CR 置入了初值，在没有新的初值置入之前，这个初值在 CR 中保持不变。以后在 OUT 端没变高电平之前，如果 GATE 端又加入了触发信号（由低变高），则又将初值从 CR 装入 CE，又从初值开始计数，而且每触发一次，该过程都重复一次。其结果将延长 OUT 端输出的负脉冲宽度。图 5.4 中第②段反映的即是这种情况，虽

然计数初值为 $n=3$ ，但 OUT 输出的脉冲宽度并不是 3 个 CLK 周期，而是 5 个 CLK 周期。GATE 由高电平变为低电平对计数过程没有影响。

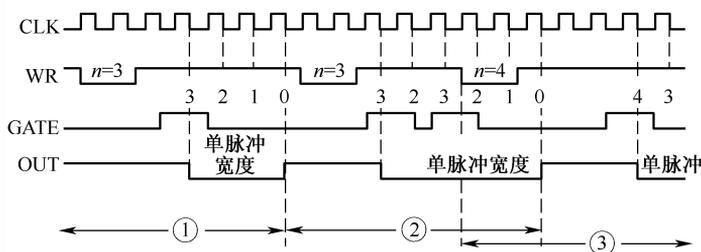


图 5.4 工作方式 1 定时波形图

3) 图中③表示在形成单脉冲的过程中，再次写入新的计数初值，对正在进行的计数过程不产生任何影响，必须等到下一个 GATE 触发信号（上升沿）出现后，才将新的 CR 内容（ $n=4$ ）装入 CE，从此开始计数。

3. 方式 2——速率波发生器方式 (Rate Generator)

方式 2 也称为 n 分频方式。在该方式下，允许自动重装计数初值，进行周期性重复计数。

方式 2 的工作过程是：当写入控制字，使 8254 进入这种工作方式后，OUT 输出高电平，这时若写入计数初值 n ，则在其后的下个 CLK 下降沿将初值装入 CE，并从初值开始作减 1 计数，OUT 保持为高电平不变；待计数值减到 1 时，OUT 将输出宽度为一个 CLK 周期的负脉冲，至计数值为 0 时，自动重新装入计数初值 n 至 CE，实现循环计数。如果 CLK 为周期性的脉冲序列，则 OUT 端也输出周期性的脉冲序列，其负脉冲的宽度为一个 CLK 脉冲周期，脉冲频率为 CLK 信号频率的 $1/n$ (n 为 CR 初值)，即为 CLK 的 n 分频信号。

方式 2 的工作特点可用图 5.5 所示的波形图来表示：

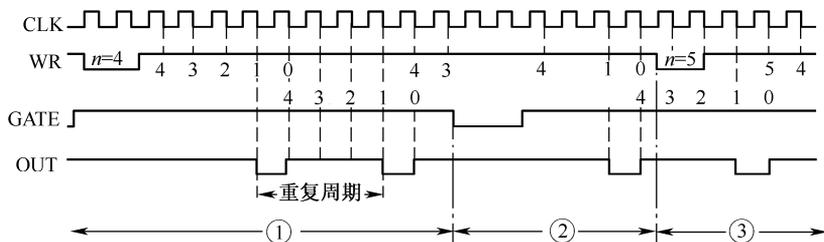


图 5.5 工作方式 2 定时波形图

1) 图中①反映的是正常工作情况，即 GATE 保持为高，且不改变 CR 内容的计数情况。这时 OUT 输出周期性的负脉冲序列，其负脉冲占空比为 $1/n$ 。图中所示例子的分频系数为 4，负脉冲占空比为 $1/4$ 。

2) 图中②反映的是启动分频过程之后，GATE 信号变低的情况。这时停止对 CLK 计数。但当 GATE 由低重新变高时，CR 内容重新置入 CE，即由 GATE 上升沿重新启动分频过程。如果此后 GATE 保持在高电平不变，则此后的过程与①情况相同。

由 1) 和 2) 可知，方式 2 的计数启动有两种方式：一是在 GATE 恒为高电平时，通过由软件

写入计数初值启动；二是写入计数初值后由 GATE 上升沿启动。

3) 图中③反映的是 GATE 信号保持高电平不变时重新写入了 CR 初值的情况。这种情况下，在按原来的初值计数使 CE 减为 0 时才将新的初值从 CR 装入 CE，改变分频系数。

4. 方式 3——方波方式 (Square Wave Mode)

在方式 3 下，OUT 端输出的是方波或近似方波信号，它的典型用法是作波特率发生器。

方式 3 的工作过程比较复杂。当写入控制字，使 8254 进入工作方式 3 后，OUT 输出低电平，这时若装入计数初值 n 且 GATE 为高电平，则在其后的下一个 CLK 脉冲下降沿，OUT 跳变为高电平，同时开始减法计数。依据写入的计数初值 n 是奇数还是偶数，减法计数的过程和 OUT 端得到的波形均有所不同。

下面结合图 5.6 所示的定时波形图进行说明。图中反映了三种不同条件下的计数过程。

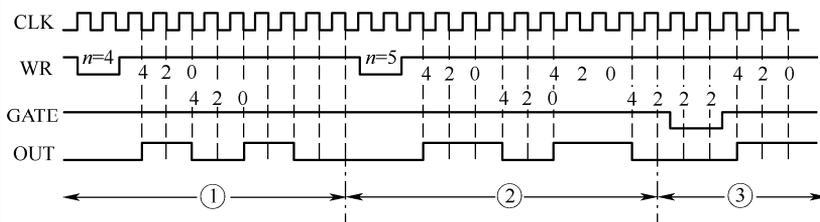


图 5.6 工作方式 3 定时波形图

1) 图中①表示计数初值为偶数 ($n=4$) 的情况。如果置入 CR 的初值 n 是偶数，减法计数对每个 CLK 脉冲减 2，经过 $n/2$ 个 CLK 脉冲，计数值达到 0 值，OUT 输出变为低；然后，CR 内的初值自动再装入 CE 并继续减 2 计数，经过 $n/2$ 个 CLK 脉冲，计数值达 0 值，OUT 输出又立即变高。如此周而复始，OUT 端得到的是完全对称 (占空比为 1/2) 的方波信号。

2) 图中②表示初值为奇数 (图中为 $n=5$) 的情况。置入的初值是奇数，则计数过程略有不同：在 OUT 变为高电平瞬间，CR 内的初值向 CE 装入时减 1 成为偶数，然后对 CLK 减 2 计数，减至 0 时 OUT 不立即变低，而是再经过一个 CLK 脉冲后变低。这就是说，方波的高电平持续时间为 $(n-1)/2+1=(n+1)/2$ 个脉冲周期。OUT 从高变低瞬间，CR 内初值向 CE 装入时减 1，然后对 CLK 减 2 计数，计数到 0 值时，OUT 输出立即变高。这就是说，方波的低电平持续 $(n-1)/2$ 个 CLK 脉冲周期。如此周而复始，OUT 端得到的是近似对称的方波信号。

3) 图中③表示计数过程中出现 GATE=0 的情况。这时 CE 暂停对 CLK 计数，直到 GATE 再次由低变高 (出现上升沿) 时，重新启动计数过程，从初值 n 开始计数。

如果要求改变输出方波的速率，则 CPU 可在任何时候向 CR 重新写入新的计数初值 n ，并从下一个计数操作周期开始起作用。而如果在新的初值写入之后又收到 GATE 上升沿，则虽然原来的方波半周期尚未结束，CE 也将将在下一个 CLK 脉冲时装入新初值，并从它开始计数。

5. 方式 4——软件触发选通方式 (Software Triggered Strobe)

方式 4 和方式 0 十分相似，都是由软件触发的计数方式。差别只是当定时时间或计数次数到时 OUT 端输出波形不同。

方式 4 的工作过程是：当写入控制字，使 8254 进入方式 4 后，OUT 信号变为高电平。这时，如果 GATE 为高电平，则在写入 CR 初值 n 后的下一个 CLK 脉冲下降沿将 CR 内容装入 CE，并从它开始对 CLK 脉冲减 1 计数。这相当于软件启动计数。当 CE 计到 0 时，OUT 端输出一个宽度

为 1 个 CLK 周期的负脉冲。可见必须经过 $n+1$ 个 CLK 脉冲周期，才产生一个负选通脉冲。这种由软件装入的计数值只一次有效，如果要继续操作，必须重新置入计数初值 n 。

方式 4 的工作特点可用图 5.7 所示的波形图来表示：

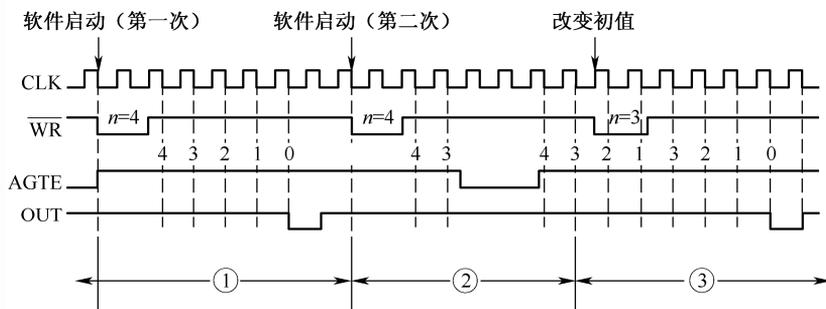


图 5.7 工作方式 4 定时波形图

1) 图中①表示 GATE 为高电平时的正常软件启动计数过程。

2) 图中②表示 GATE 对计数过程的控制作用，如果在操作过程中，GATE 变为无效（低电平），则停止减 1 计数，直到 GATE 恢复有效时，重新从初值开始减 1 计数。

3) 图中③表示计数过程中改变（重写）计数初值对计数过程的影响。这时将从改变之后的下一个 CLK 脉冲起，按新计数初值重新开始计数。

显然，利用工作方式 4 可以完成定时功能。定时时间从装入计数初值 n 开始，当定时时间到时，OUT 输出一负脉冲。定时时间 $T=n \times \text{CLK 周期}$ 。这种工作方式也可完成计数功能，只要将要求计数的事件以脉冲方式从 CLK 端输入，将计数次数作为计数初值装入 CR，即可利用 GATE 信号启动计数过程，对 CLK 端输入的脉冲作减 1 计数。当 CE 计数至 0 时，OUT 输出一负脉冲，表示计数次数到。

6. 方式 5——硬件触发选通方式（Hardware Triggered Strobe）

方式 5 与方式 1 十分相似，也是一种由 GATE 端引入的触发信号控制的计数或定时工作方式，只不过 CE 计数到 0 时 OUT 端产生的是负选通脉冲。

其工作过程是：当写入控制字，使 8254 进入方式 5 后，OUT 输出高电平。硬件触发信号由 GATE 端引入。开始时 GATE 一般应输入为 0，装入计数初值 n 后，减 1 计数并不开始，一定要等到 GATE 端出现一个正跳变触发信号后才开始。待计数值计到 0，OUT 将输出一个负脉冲，其宽度固定为一个 CLK 周期，表示定时时间或计数次数到。

方式 5 的工作特点可用图 5.8 所示的波形图来表示：

1) 图中①表示硬件触发选通的正常工作情况。在这种工作方式下，当计数值计到 0 后，系统将自动重新装入计数初值 n ，但并不开始计数，一定要等到 GATE 端出现上升沿，才会开始进行减 1 计数。如果由 CLK 输入的是一定频率的时钟脉冲，那么可完成定时功能，定时时间从 GATE 上升沿开始，到 OUT 端输出负脉冲结束。如果从 CLK 端输入的是要求计数的事件，则可完成计数功能，计数过程也从 GATE 上升沿开始，到 OUT 输出负脉冲结束。GATE 可由外部电路或控制现场产生。

2) 图中②表示改变计数初值对工作过程的影响情况。当在前一计数周期中装入新的计数初值 n 时，它并不影响正在进行的操作过程，而是到下一个计数操作周期才会按新的计数值进行操作。

不过如果写入新的初值后，在计数值到达 0 之前加了触发信号（GATE 上升沿），则下一个 CLK 脉冲将使新的初值装入 CE，并从它开始新的计数。

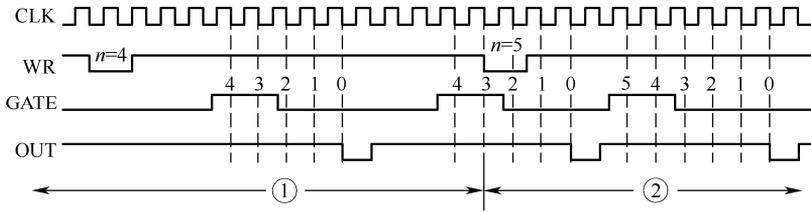


图 5.8 工作方式 5 定时波形图

综上所述可知：对于各种不同的工作方式，作为 8254 各通道门控信号的 GATE 端，所起的作用各不相同。在 8254 的应用中，必须正确使用 GATE 信号，才能保证各通道的正常工作。

5.2.5 8254 的编程

和其他可编程芯片一样，为了使用 8254，必须通过读/写操作对它编程。8254 的编程包括初始化编程和工作编程两个方面。

1. 初始化编程

初始化编程是指在工作之前写入控制字以确定每个计数器通道的工作方式和向每个计数器通道写入计数初值。

(1) **方式控制字**。方式控制字的格式如图 5.9 所示。最高两位 SC_1 、 SC_0 用于指明写入本控制字的计数器通道。 $SC_1SC_0=00, 01, 10$ 分别表示选择通道 0，通道 1，通道 2。所以，每写一个控制字，只能选择一个通道的工作方式；要设置三个通道的工作方式，必须对同一地址（控制字寄存器）写入三个控制字才行。 $SC_1SC_0=11$ 在 8253 中没有意义，在 8254 中则表示写入控制字寄存器的是后面将要讲到的读回命令字。

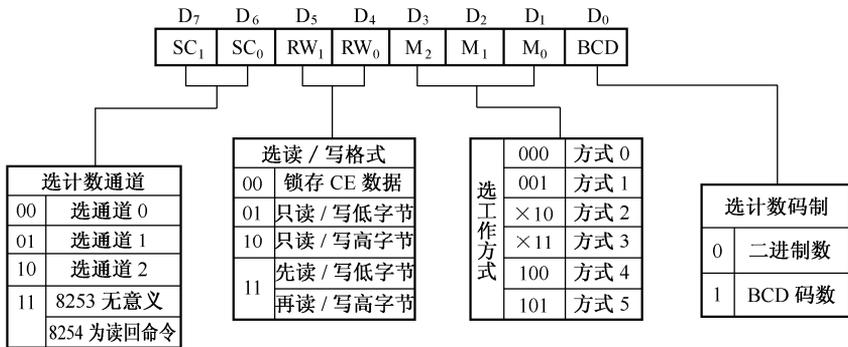


图 5.9 8254 控制字格式

RW_1 、 RW_0 用于定义对所选计数通道的读/写操作格式，即指明是只读出 OL 或写入 CR 的低字节；还是只读/写其高字节；还是先读/写其低字节，再读/写其高字节。例如，如果向计数通道 2 写入的控制字的高 4 位为 1011，那么以后向其 CR 预置初值时必须用两条输出指令，先后将初值的低字节和高字节写入 CR 的低 8 位和高 8 位；同样，从 OL 读数时，也必须相继用两条输入指令



先后将其低 8 位和高 8 位读出。而如果控制字的 RW_1RW_0 两位为 01 或 10, 则向 CR 写入初值或从 OL 读出数值时, 每次只需一条 OUT 指令或 IN 指令, 写入或读出指定的低字节或高字节的内容。 $RW_1RW_0=00$ 是将所选通道中 CE 的当前内容锁存到输出锁存器 OL 中, 为 CPU 读取当前计数值做准备。这时的控制字实际上就是后面将讲到的锁存命令字。

M_2 、 M_1 、 M_0 三位用于指定所选通道的工作方式。

BCD 位是计数码制选择位, 用于定义所选通道是按二进制计数还是按 BCD 码计数。

(2) **计数初值**。根据计数码制的不同, 计数初值可以以二进制数码或 BCD 数码形式写入, 前者取值为 0000H~FFFFH, 后者取值为 0000~9999。无论是二进制计数还是 BCD 计数, 都是写入 0000H 时表示初值最大, 前者表示初值为 65536, 后者表示初值为 10000。

用作定时器时, 其计数初值 (也叫定时系数) 可用如下公式计算:

$$\text{定时系数} = \frac{\text{要求定时的时间}}{\text{时钟脉冲周期}} = \frac{\text{输入时钟频率}}{\text{输出信号频率}}$$

对 8254 各通道的初始化, 均必须按先写入方式控制字, 后写入计数初值的顺序进行。要特别注意的是, 用了 8254 中多少个通道, 就必须写多少个控制字和多少个计数初值, 只是各控制字写入的是同一地址 (3 口), 而各计数初值写入的是各通道自己的地址 (0 口, 1 口或 2 口)。另外, 采用 BCD 计数时, 计数值必须用 BCD 编码表示, 否则写入的就可能不是等值的十进制数。

例 5.1 已知 8254 定时/计数器通道 0、通道 1、通道 2 和控制字寄存器端口地址分别为 200H、201H、202H 和 203H, CLK_0 接有 1MHz 的时钟, 通道 0 输出 OUT_0 接通道 1 时钟输入 CLK_1 , 现希望 OUT_1 输出一个高电平为 200ms, 低电平为 1ms 的连续信号, 编写初始化程序。

编写 8254 的初始化程序, 首先必须确定各通道的工作方式和计数初值, 然后按规定写入各通道的方式控制字和计数初值。

此例, 要使 OUT_1 输出高电平为 200ms, 低电平为 1ms 的连续信号, 即得到负脉冲宽度为 1ms、周期为 201ms 的连续负脉冲, 所以计数器 1 应工作在方式 2, CLK_1 的时钟周期应为 1ms (负脉冲宽度), 计数初值为: $201\text{ms}/1\text{ms}=201$ 。

计数器 0 要为通道 1 提供周期为 1ms 的时钟信号, 所以应工作在方式 2 或方式 3, 计数初值为:

$$\text{计数初值} = \frac{1\text{ms}}{1\mu\text{s}} = 1000$$

相应的初始化程序如下:

```
MOV  DX, 203H
MOV  AL, 37H          ; 通道 0 方式 3 BCD 计数
OUT  DX, AL
MOV  DX, 200H        ; 指向通道 0 端口地址
MOV  AL, 1000H       ; 通道 0 计数值为 BCD 数 1000
OUT  DX, AL          ; 写低字节
MOV  AL, AH
OUT  DX, AL          ; 写高字节
MOV  DX, 203H
MOV  AL, 54H         ; 通道 1 方式 2 二进制计数
OUT  DX, AL
MOV  DX, 201H        ; 指向通道 1 端口地址
```



```
MOV AL, 201 ; 通道 1 计数值为 201
OUT DX, AL ; 只写低字节
```

若用 C 语言编程，则相应的 8254 初始化子程序如下：

```
init8254(){
    Outportb(0x203,0x37); /* 写通道 0 方式控制字 */
    Outportb(0x200,0x0); /* 写通道 0 计数值低字节 */
    Outportb(0x200,0x10); /* 写通道 0 计数值高字节 */
    Outportb(0x203,0x54); /* 写通道 1 方式控制字 */
    Outportb(0x201,201); /* 写通道 1 计数值 */
}
```

2. 工作编程

工作编程是指工作过程中改变某通道的计数初值和写入命令字以读出某计数通道的当前 CE 内容或状态寄存器内容。

8254 有两个命令字：计数器锁存命令字和读回命令字。锁存命令是和 8253 兼容的。读回命令是 8254 才有的，它包含有锁存命令的功能但比它有所扩充。

锁存命令和读回命令使用和控制字相同的地址写入 8254。

(1) **锁存命令字**。计数器锁存命令用来将当前的 CE 内容锁存到输出锁存器 OL，以供 CPU 读出。其格式如图 5.10 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC ₁	SC ₀	0	0	×	×	×	×

图 5.10 锁存命令字格式

其中 D₅D₄=00 为锁存命令特征值；SC₁、SC₀ 的含义和控制字相同，是计数通道选择位，也不能同时为 1；D₃~D₀ 位可为任意状态，在锁存命令中无任何意义。

(2) **读回命令字**。读回命令用来将指定计数器通道的 CE 当前内容锁存入 OL 或/和将状态寄存器内容锁存入状态锁存器。和锁存命令不同，它能同时规定锁存几个计数器通道的当前 CE 内容和状态寄存器内容。读回命令的格式如图 5.11 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	1	COUNT	STATUS	CNT ₂	CNT ₁	CNT ₀	0
0: 锁存当前计数器内容						1: 选中通道 0	
0: 锁存状态寄存器内容			1: 选中通道 2		1: 选中通道 1		

图 5.11 读回命令字格式

COUNT 位为 0 时，凡 CNT₂、CNT₁、CNT₀ 位选中的通道的当前 CE 内容均予以锁存，以备 CPU 读取。当某一个计数器被读取后，该计数器自行失锁，但其他计数器并不受其影响。如果对同一个计数器发出多次读回命令，但并不立即读取计数值，那么只有第一次发出的读回命令是有效的，后面的无效。也就是说，以后读取的计数值仅是第一个读回命令锁存的数。

同样，若 STATUS 位为 0，则凡是 CNT₂、CNT₁、CNT₀ 位指定的计数器通道的状态寄存器内容都将被锁存入相应通道的状态锁存器，供 CPU 读取。

(3) **状态字**。CPU 通过读各通道的状态锁存器获取状态字，以了解相应通道的现行工作状态。

状态字格式如图 5.12 所示。

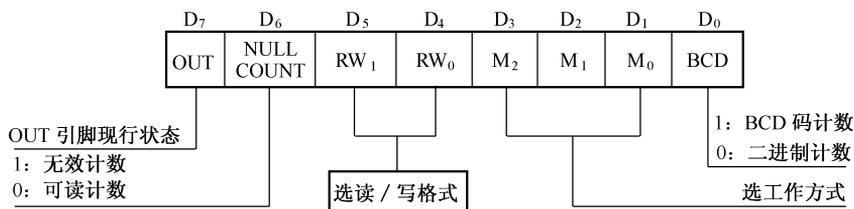


图 5.12 状态字格式

其中 $D_5 \sim D_0$ 意义与前面控制字的对应位意义相同。 D_7 位 (OUT) 反映了相应计数器通道 OUT 端的现行状态, 利用它就可以通过软件来监视计数器输出, 减少系统的硬件开销。 D_6 位 (NULL COUNT) 指示 CR 内容是否已装入 CE, 若最后写入 CR 的内容已装入 CE, 则 D_6 位为 0, 表示可读计数; 若 CR 内容未装入 CE, 则 D_6 位为 1, 表示无效计数, 读取的计数值将不反映刚才写入的那个新计数值。

和对当前 CE 内容的读回规则一样, 若对同一个状态寄存器发了多次读回命令, 但每次命令后并未当即读取其状态, 那么除第一次读回命令引起的锁存操作有效外, 其余均无效。即是说, 发多次读回命令后读取的状态, 总是第一次命令发出时刻计数器的状态。

如果读回命令的 \overline{COUNT} 、 \overline{STATUS} 都为 0, 则被选定计数通道的现行 CE 内容和状态同时被锁存, 它等价于发出两条单独的 CE 值和状态的读回命令。

最后要说明一点, 若通道的计数值和状态都已锁存, 则该通道第一次读出的将是状态字, 而不管先锁存的究竟是计数值还是状态。下一次或下两次再读出的才是计数值 (一次还是两次由编程时方式控制字所规定的计数值字节数而定)。以后的读操作又回到无锁存的计数。

5.3 8254 应用举例

8254 是目前微机系统中应用非常广泛的可编程定时器/计数器芯片。本节通过几个实际例子说明 8254 的应用及编程方法。

例 5.2 在 8086 系统中, 用 8254 构成一个智能化频率计。

要构成智能化频率计, 需用一个计数器通道对外部脉冲进行计数, 同时要用另一个计数器通道进行定时, 以控制计数时间。假定控制计数的时间为 1s, 用 8254 构成的智能化频率计原理如图 5.13 所示。

通道 0 工作于方式 0, 用于对外部脉冲计数, 计数初值设置为 0。通道 1 工作于方式 3, 用于产生 1kHz 的方波分频。通道 2 工作于方式 2, OUT_2 接 8259A 的 IR_2 , 用于产生 1s 定时检测 (中断) 信号, 控制计数时间。工作原理是: 当定时时间到时, OUT_2 输出负脉冲通过 8259A 向 CPU 发出中断请求, 在中断处理程序中, 读出通道 0 计数值 N , 则可计算出脉冲频率为:

$$\text{脉冲频率} = \frac{65536 - N}{\text{定时时间}} = 65536 - N \text{ (Hz)}$$

根据图中 CLK_1 输入 2.5MHz 的时钟脉冲, 可以计算出计数器通道 1 和通道 2 的计数初值, 分别为:

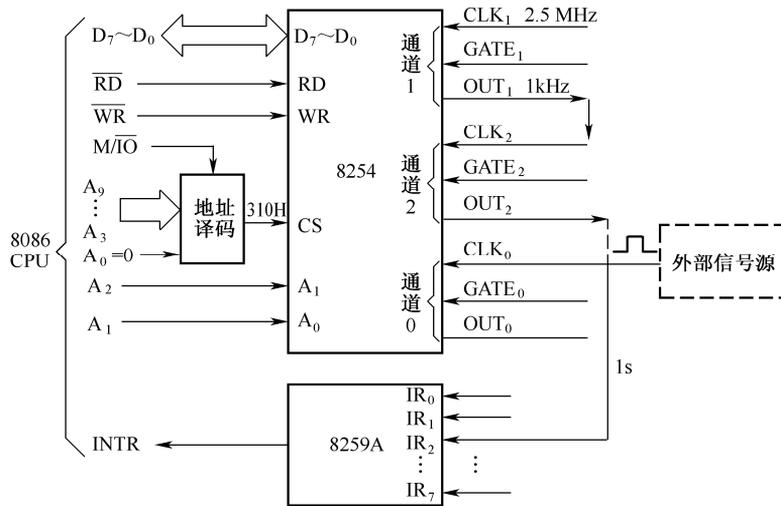


图 5.13 用 8254 构成的智能频率计

$$\text{通道 1 计数初值} = \frac{T_{\text{OUT}_1}}{T_{\text{CLK}_1}} = \frac{1\text{ms}}{0.4\mu\text{s}} = 2500$$

$$\text{通道 2 计数初值} = \frac{T_{\text{OUT}_2}}{T_{\text{CLK}_2}} = \frac{1\text{s}}{1\text{ms}} = 1000$$

对图 5.13 的 CPU 接口侧连接情况还要注意，8254 的端口地址线 A₁、A₀ 与系统地址线 A₂、A₁ 相连，而系统地址线的 A₀ 固定为“0”，参加高位地址译码，以形成对 8254 的片选信号 $\overline{\text{CS}}$ 。这样连接的目的是保证 CPU 访问 8254 的 4 个端口地址均为偶数地址。根据图中译码器输出的基地址为 310H，可以确定通道 0 端口、通道 1 端口、通道 2 端口和控制字寄存器端口的地址分别应为 310H、312H、314H 和 316H。

依据上述测量原理，不难设计智能频率计的程序。程序应包括 8254 初始化、8259A 初始化、中断向量表填写和一个用来处理 1s 定时检测的中断处理程序。假定系统已对 8259A 初始化，则用汇编语言编写的该频率计驱动程序如下：

```

DATA SEGMENT
    TIME DW ? ; 保存脉冲频率值
    IRQ2 EQU 0AH ; IR2 中断类型码
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START: MOV AX, DATA
        MOV DS, AX
        CLI ; 关中断，以填写中断向量表
        MOV DI, IRQ2*4 ; 取 IRQ2 的中断向量地址
        CLD
        XOR AX, AX
        MOV ES, AX ; ES 指向中断向量表段基址
        MOV AX, OFFSET INT_ROUT ; 填写中断向量的偏移地址
        STOSW
    
```



```

MOV AX, SEG INT_ROUT      ; 填写中断向量的段基址
STOSW
INI8253: MOV AL, 30H        ; 通道 0 方式 0, 二进制计数
MOV DX, 316H
OUT DX, AL                ; 写入通道 0 方式控制字
MOV AL, 0                 ; 写入计数器 0 的初值
MOV DX, 310H
OUT DX, AL
OUT DX, AL
MOV AL, 77H              ; 通道 1 方式 3, BCD 计数
MOV DX, 316H
OUT DX, AL                ; 写入通道 1 方式控制字
MOV AX, 2500H            ; 写入计数器 1 的初值
MOV DX, 312H
OUT DX, AL
MOV AL, AH
OUT DX, AL
MOV AL, 0B5H            ; 通道 2 方式 2, BCD 计数
MOV DX, 316H
OUT DX, AL                ; 写入通道 2 方式控制字
MOV AX, 1000H            ; 写入计数器 2 的初值
MOV DX, 314H
OUT DX, AL
MOV AL, AH
OUT DX, AL
STI                       ; 开中断
AGAIN: MOV AH, 01H        ; 有键按下?
INT 16H
JZ AGAIN                 ; 无键按下程序等待, 否则程序结束
MOV AH, 4CH              ; 返回 DOS
INT 21H

INT_ROUT PROC            ; 1s 检测中断处理程序
PUSH DX
PUSH CX
PUSH AX
READ: MOV AL, 11010010B   ; 写读回命令, 锁存计数器 0 的计数值
MOV DX, 316H
OUT DX, AL
MOV DX, 310H
IN AL, DX                ; 读通道 0 当前计数值
MOV CL, AL
IN AL, DX                 ; 读高位
MOV CH, AL
NEG CX                    ; 计算脉冲频率 (65536-N)
MOV TIME, CX             ; 保存测量结果

```



```

MOV    AL, 0                ; 重装通道 0 计数值, 开始下一轮计数
MOV    DX, 310H
OUT    DX, AL
OUT    DX, AL
MOV    AL, 20H              ; 发中断结束命令
OUT    20H, AL
POP    AX
POP    CX
POP    DX
IRET                          ; 开中断, 中断返回
INT_ROUT ENDP

```

CODE ENDS

END START

若用 C 语言编写, 则实现例 5.2 功能的驱动程序如下:

```

#include "stdio.h"
#include "conio.h"
#include "dos.h"
#define IRQ2 0x0a           /* 定义 IRQ2 对应的中断向量号 */
void interrupt (*oldhandler)(void), handler(void);
int time;

main(){
    int temp;
    disable();              /* 关中断 */
    oldhandler = getvect(IRQ2); /* 获取系统中断向量 */
    setvect(IRQ2, handler); /* 设置新的中断向量 */
    temp=inportb(0x21);     /* 读系统中断屏蔽字 */
    outportb(0x21,temp&0xfb); /* 开 IRQ2 中断 */
    outportb(0x316,0x30);   /* 初始化 8254 通道 0 */
    outportb(0x310,0);
    outportb(0x310,0);
    outportb(0x316,0x77);   /* 初始化 8254 通道 1 */
    outportb(0x312,0);
    outportb(0x312,0x25);
    outportb(0x316,0xb5);   /* 初始化 8254 通道 2 */
    outportb(0x314,0);
    outportb(0x314,0x10);
    enable();              /* 开中断 */
    While(!kbhit());      /* 无键按下, 循环 */
    setvect(IRQ2, oldhandler); /* 恢复系统中断向量和屏蔽字 */
    outportb(0x21,temp);
}

void interrupt handler(void){
    unsigned char time_al,time_ah;
    outportb(0x316,0xd2); /* 发读通道 0 计数值读回命令 */
    time_al=inportb(0x310);
}

```

```

time_ah=inportb(0x310);
time=time_ah*0x10+time_al;
time=0-time;
outputb(0x310,0);          /* 重装通道 0 计数值 */
outputb(0x310,0);
outputb(0x20,0x20);
}

```

例 5.3 用 8253/8254 实现啤酒包装流水线控制中的定时和计数功能。

某啤酒包装流水线中，每个包装箱能装入 24 罐，希望每通过 24 罐，流水线要停 2s，等待装箱完毕再继续传送装箱。其工作原理是：流水线工作啤酒罐通过时，光电传感器发出脉冲信号，计数器对该脉冲计数；计满 24 罐时，使流水线暂停 2s，此时无啤酒罐通过，因而光电传感器也无脉冲输出，相当于使计数器停止计数；当 2s 定时时间到时，重新启动流水线，重复上述过程。

要实现上述啤酒包装流水线控制中的定时和计数功能，可利用 8253/8254 的一个通道作为计数器，用作 24 罐的计数，另一个通道作为定时器，用作 2s 定时。具体实现方法有多种方案。图 5.14 给出了一种具体实现方案。

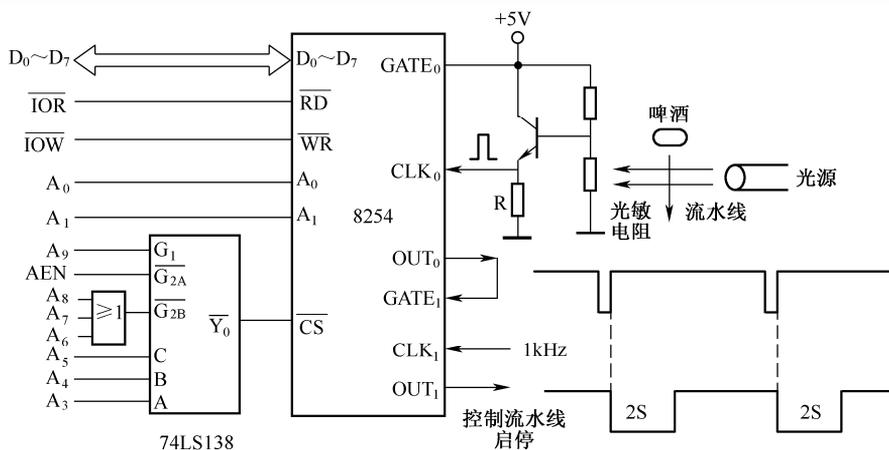


图 5.14 啤酒包装流水线的定时和计数控制

8254 的通道 0 作为计数器，工作于方式 2（方式控制字为 14H，计数初值为 24），用作 24 罐的计数。8254 的通道 1 作为定时器，工作于方式 1，用作 2s 定时（方式控制字为 72H，计数值 $= \frac{2s}{1/1kHz} = \frac{2s}{1ms} = 2000$ ）， OUT_1 输出用于控制流水线的启停。

该电路定时计数控制的原理是：流水线工作，啤酒罐经过光电传感器产生计数脉冲，加到 8254 的 CLK_0 端，通道 0 对该信号计数，当计数值达到 23 时，通道 0 的输出 OUT_0 为低电平，在第 24 罐啤酒通过传感器时，通道 0 的输出端 OUT_0 变为高电平，利用该 OUT_0 的上升沿来触发通道 1 的 2s 定时器开始定时，同时 $OUT_1=0$ 使流水线暂停，传感器无脉冲输出使计数停止。当 2s 定时时间到时， OUT_1 变为高电平，流水线重新启动。这时，啤酒罐经过传感器产生计数脉冲加到 CLK_0 ，通道 0 又开始 24 罐啤酒的计数，重复以上的工作过程。

据此，用汇编语言编写的 8254 工作程序如下：



3 个计数器通道的时钟输入均为 1.1931816MHz 方波(由系统时钟分频得到)。系统分配给 8254 的端口地址范围为 0040H~005FH, 每个端口可用 8 个不同的地址访问。实际一般使用 0040H~0043H 作为端口 0~端口 3 的地址。系统 8255 的 PB_0 、 PB_1 分别用作通道 2 的门控 ($GATE_2$) 和控制 OUT_2 的输出, 4 个端口地址为 0060H~0063H。

1. 通道 0

通道 0 是一个产生实时时钟信号的系统计时器。系统利用它完成日历时钟计数, 控制软盘驱动器读/写操作后的电动机的自动延迟停机以及为用户提供定时中断调用。用户可使用这个中断调用运行自己的中断处理程序。它的 $GATE_0$ 接+5V, CLK_0 输入为 1.1931816MHz 方波, 工作于方式 3, CE 的初值(即置入 CR 的内容)为 0(即 65536), 输出信号 OUT_0 连接到系统板上 8259 的 IRQ_0 中断请求输入线(最高级可屏蔽中断)。于是, 在 OUT_0 引脚上输出 $1.1931816\text{MHz}/65536=18.2065\text{Hz}$ 的方波脉冲序列, 即每经过 54.925ms 产生一次 0 级中断请求。根据该中断请求, 系统直接调用固化在 BIOS 中的中断处理程序, 调用命令为 INT 8H(即该中断的类型码为 8)。

INT 8H 中断服务程序的第一项功能是完成日时钟的计时。BIOS 数据区的 40:6CH 和 40:6FH 是一个双字的系统计时器。每次中断计时操作就是对该系统计时器进行加 1 操作。因为通道 0 中断频率为每秒 18.2065 次, 计满 24 小时需要中断 $18.2065 \times 3600 \times 24=1573042$ (001800B2H) 次。每次中断总是对低字进行加 1, 当低字计满为 0 时, 高字加 1; 当高字计到 0018H、低字计到 00B2H 时, 表示计满 24 小时, 双字复位清零, 并建立计满 24 小时标志, 置 40:70H 单元为 1。任何一次对中断 INT 1AH 的调用, BIOS 中的中断服务程序将撤销其标志, 将 40:70H 单元复位为 0。

INT 8H 中断服务程序的第二项功能是实现软盘驱动器的马达开启时间管理, 使其开启一段时间、完成数据存取操作后自动关闭。在系统初始化时, 系统设定的延迟时间为 2s。系统控制延迟停机的工作原理是, 在软盘存取操作后从磁盘基数区域读取一个延迟常数到 BIOS 数据区单元 40:40H, 然后利用通道 0 的每秒 18.2065 次的中断, 对 40:40H 单元值进行减 1 操作, 当减为 0 时, 发出关闭软盘驱动电机的命令。由于通道 0 的中断间隔时间为 54.925ms, 达到延迟 2s 所需要的延迟常数就为 37 ($54.925\text{ms} \times 37=2\text{s}$)。INT 8H 服务程序处理了日时钟计时操作后, 紧接着对 40:40H 单元减 1, 并判断是否为 0 操作。

INT 8H 中断服务程序的最后一项功能是进行 INT 1CH 软中断调用。AT 机系统设置 INT 1CH 的目的在于建立一个用户可用的定时操作服务程序入口。如果用户没有编制新的 INT 1CH 中断服务程序, 并修改 1CH 的中断向量, 则 INT 8H 调用了 1CH 号中断后立即从 INT 1CH 中断返回, 因为 AT 系统原来的 INT 1CH 中断服务程序仅由一条中断返回指令 IRET 组成。

2. 通道 1

通道 1 专门用作动态存储器刷新的定时控制。它的控制端 $GATE_1$ 同样接高电平+5V, CLK_1 端的信号也和通道 0 相同, 为 1.1931816MHz 方波脉冲串, 工作于方式 2。CE 的初值预置为 18(即 0012H), 于是在 OUT_2 端输出一负脉冲序列, 其周期为 $18/1.1931816=15.08\mu\text{s}$, 该信号用作 D 触发器的触发时钟信号, 使每隔 $15.08\mu\text{s}$ 产生一个正脉冲, 周期性地对系统的动态存储器刷新。由于 PC/AT 机使用的是 $128\text{KB} \times 1$ 位的 DRAM 芯片, 其行列分配为 256 行 \times 512 列, 所以从上述刷新周期可知, 它的刷新周期为 3.86ms ($15.08\mu\text{s} \times 256=3.86\text{ms}$)。

3. 通道 2

通道 2 用于为系统机箱内的扬声器发声提供音频信号。工作于方式 3, 系统中 CR 的内容预置为 0533H(即十进制 1331), 于是当 $GATE_2$ 为高电平时, OUT_2 将输出频率为 $1.1931816\text{MHz}/1331$



≈900Hz 的方波, 该方波信号经放大和滤波后推动扬声器。送到扬声器的信号实际上受到从并行接口芯片 8255 来的双重控制, 8255 的 PB_0 位接到通道 2 的 $GATE_2$ 引脚, 通道 2 的 OUT_2 信号和 8255 的 PB_1 同时作为与门的输入。 PB_0 和 PB_1 位可由程序决定为 0 或 1, 显然只有 PB_0 和 PB_1 都是 1, 才能使扬声器发出声音。改变计数初值, 可改变 OUT_2 输出方波信号的频率, 从而改变扬声器发声的音调。实现硬件控制发声的例程如下:

```

IN    AL, 61H
AND   AL, 0FCH           ; 使  $PB_1$ 、 $PB_0$  为 0, 关闭扬声器
OUT   61H, AL
MOV   AL, 0B6H          ; 设置通道 2 方式控制字, 使之工作于方式 3
OUT   43H, AL
MOV   AX, 1352           ; 按 A 调设置计数初值
OUT   42H, AL           ; 写初值低字节
MOV   AL, AH             ; 写初值高字节
OUT   42H, AL
IN    AL, 61H           ; 使  $PB_1$ 、 $PB_0$  为 1, 启动扬声器工作
OR    AL, 03H
OUT   61H, AL
    
```

4. BIOS 对 8254 的初始化编程

根据 8254 各计数器通道在定时系统中的功能, PC/AT 机在上电后, BIOS 对它的初始化程序段为:

```

MOV   AL, 36H           ; 设置通道 0 方式控制字, 选择双字节写, 方式 3
OUT   43H, AL           ; 二进制计数
MOV   AL, 0             ; 计数初值设定为 65536
OUT   40H, AL           ; 写入低字节
OUT   40H, AL           ; 写入高字节
MOV   AL, 01010100B     ; 设置通道 1 方式控制字, 定义只写低位字节
OUT   43H, AL           ; 方式 2, 二进制计数
MOV   AL, 18            ; 预置计数初值
OUT   41H, AL
MOV   AL, 10110110B     ; 设置通道 2 方式控制字, 定义双字节写
OUT   43H, AL           ; 方式 3, 二进制计数
MOV   AX, 533H          ; 写计数初值
OUT   42H, AL           ; 先写低位字节
MOV   AL, AH            ; 再写高位字节
OUT   42H, AL
IN    AL, 61H           ; 以下使 8255 的  $PB_1$ 、 $PB_0$  为 1, 控制扬声器发声
MOV   AH, AL            ; 将 8255 B 口的内容保存于 AH
OR    AL, 03H
OUT   61H, AL
    
```

在上述程序中, 8255 端口 B 内容保存在 AH 寄存器中, 当关闭扬声器时, 应再把存放在 AH 中的内容送回 8255 的端口 B。



习题与思考题五

5.1 选择题

- (1) 8254 的各种工作方式中, 利用 $GATE=0$ 来控制计数过程的暂停, 而当 $GATE$ 信号从 0 变 1 时, 又接着计数的是 ()。
- A. 方式 0 B. 方式 2 C. 方式 3 D. 方式 4
- (2) 可编程计数器/定时器电路 8254 有 () 种可编程设置的工作方式和 () 个 I/O 端口地址。
- A. 3, 4 B. 4, 5 C. 6, 3 D. 6, 4
- (3) 8254 在计数过程中, 改变计数初值, 必须当外部 $GATE$ 信号触发后, 新的计数值才能有效的工作方式有 ()。
- A. 方式 0 和方式 4 B. 方式 2 和方式 3
C. 方式 1 和方式 5 D. 方式 0 和方式 2
- (4) 8254 某计数/定时通道采用 BCD 计数, 写入的计数初值为 0000H, 则该计数/定时通道的计数初值为 ()。
- A. 0 B. 10000 C. 65536 D. 65535
- (5) 若 8254 用读回命令先后锁存了某通道的计数值和状态值, 但均未读出, 则以后第一次读出的是其 ()。
- A. 状态值 B. 计数值
C. 两者中先锁存的值
- (6) 8253/8254 有多种工作方式, 下面哪种说法不正确? ()
- A. 可以从 OUT 上输出周期重复的窄脉冲
B. 可以从 OUT 上输出周期重复的对称方波
C. 可以从 OUT 上输出单个宽度可编程的负脉冲
D. 可以从 OUT 上输出单个宽度可编程的正脉冲
- (7) 8253/8254 定时器/计数器中, 门控信号 $GATE$ 所起的作用是 ()。
- A. 为定时器/计数器提供计数脉冲输入
B. 控制计数过程的启/停
C. 输出定时信号
D. 控制各端口的读/写
- (8) 计数器/定时器 8254 的计数器 0 工作于方式 2, 即分频工作方式, CLK_0 的时钟输入频率为 1MHz, 计数初值为 0064H, 则 OUT_0 的输出负脉冲周期为 ()。
- A. 0.01ms B. 0.064ms
C. 0.1ms D. 0.64ms
- (9) 若要在 8253/8254 的计数过程中读取计数值, 方法是 ()。
- A. 直接用 IN 指令读出 CE 的内容
B. 先写入一个锁存命令, 将 CE 内容锁存入 OL, 然后再用 IN 指令读出 OL 的内容
C. 先用读回命令将 CE 的内容锁存于 OL, 然后再用 IN 指令读出 OL 的内容



(10) 某计算机应用系统中采用 8254 的计数器 0 作频率发生器, 计数器 0 的计数时钟频率为 1.2775MHz, 欲使它输出一个频率为 2500Hz 的信号, 则在初始化时, 送到计数器 0 的计数初值为 ()。

- A. 0FFFH
- B. 02FFH
- C. 01FFH
- D. 00FFH

5.2 试简述微机系统中定时器/计数器的必要性和重要性, 以及定时实现的常用方法。

5.3 可编程定时器/计数器 8253/8254 有几个通道? 各通道有几种工作方式? 各种工作方式的主要特点是什么? 8254 与 8253 有什么区别?

5.4 8254 的初始化编程包括哪几项内容? 它们在顺序上有无要求, 如何要求?

5.5 要读回 8254 工作过程中的当前状态和当前 CE 内容, 分别有几种方法? 试简述之。

5.6 8253/8254 既可作定时器又可作计数器。作定时器和计数器用时区别何在?

5.7 下表给出了对 8254 依次发出的 8 条读回命令, 试将各条命令的作用及其执行后结果填入表中。

次序	读回命令								命令作用	执行结果
	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀		
1	1	1	0	1	0	1	0	0		
2	1	1	0	0	1	0	0	0		
3	1	1	1	0	1	0	1	0		
4	1	1	1	0	1	1	0	0		
5	1	1	0	0	0	1	1	0		
6	1	1	0	1	0	0	1	0		
7	1	1	0	0	1	1	1	0		
8	1	1	1	1	1	0	1	0		

5.8 用 8254 通道 2 作为 DRAM 刷新定时器, 动态存储器要求在 2ms 内对全部 128 行存储单元刷新一遍, 假定计数用的时钟频率为 2MHz, 问该通道应工作在什么方式? 请确定控制字和计数值。

5.9 若 8254 在工作时有某计数器的 OUT 输出波形如图 5.16 所示。波形中脉冲周期为 40ms, 负脉冲宽度为 2μs。试问该计数器的外部计数时钟频率为多少? 计数器工作在方式几? 计数值为多少? 此时的 GATE 上的状态是什么?

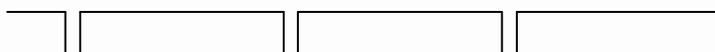


图 5.16 OUT 输出波形

5.10 某系统中 8254 芯片的通道 0~通道 2 和控制口地址为 90H~93H, 定义通道 0 工作在方式 2, CLK₀=5MHz, 要求输出 OUT₀=1kHz; 定义通道 1 工作在方式 4, 用 CLK₁ 作计数脉冲, 计数值为 1000。请画出硬件连接图, 并编写初始化程序。

5.11 某 IBM PC/XT 应用系统中, 当某一外部事件发生时 (给出一高电平信号), 1s 后向主机申请中断, 若用 8254 实现延迟, 试分析 8254 应采用哪种工作方式, 说明实现方法并对 8254 进行初始化编程。设 8254 的端口地址为 330H~333H, 输入时钟为 1.19MHz。

5.12 8254CLK₀ 的时钟频率为 2.5MHz。该通道的最大定时时间是多少? 若要产生 1s 的定时中断, 说明实现方法 (画图表示有关通道信号的硬件连接, 说明有关通道的工作方式)。

5.13 试简要说明 8254 应用于 8 位 8088、16 位 8086/80286 和 32 位 80386/80486 等不同字长的 PC 机系统时, 与地址总线的接口有什么不同? 在 16 位和 32 位系统中, 数据线的连接对地址总线接口有影响吗? 若有, 如何影响?

5.14 试利用 PC/AT 机系统板上的 8254 通道 0 产生年、月、日、时、分、秒的时间记录并显示于 CRT 屏幕上。在键盘上输入当前的年、月、日、时、分、秒后启动 8254 计时操作。

5.15 试编写一程序, 使 PC/AT 系统板上的发声电路发出 200Hz 至 900Hz 频率连续变化的报警声。

5.16 某应用系统中, 有一个 4MHz 时钟提供给 8254 使用, 系统中 8254 通道 2 接一发光二极管, 要使发光二极管以亮 2 秒、灭 2 秒的间隔工作, 假定 8254 的通道地址分别为 3F8H~3FBH, 设计 8254 的接口电路, 并写出工作程序。

5.17 试说明如何利用 8254 测量从同一信号线送来的两个脉冲的时间间隔, 测量的最大时间间隔为 1h, 读时精度为 1ms。假定时钟频率为 5MHz, 8254 的地址为 05C8H~05CBH。

5.18 已知 8254 的应用电路如图 5.17 所示。相应的初始化程序如下, 试分析 8254 的计数器 0 和计数器 1 各输出什么波形, 频率各是多少?

```
MOV  DX, 203H
MOV  AL, 25H
OUT  DX, AL
MOV  AL, 56H
OUT  DX, AL
MOV  DX, 200H
MOV  AL, 1
OUT  DX, AL
INC  DX
MOV  AL, 64H
OUT  DX, AL
```

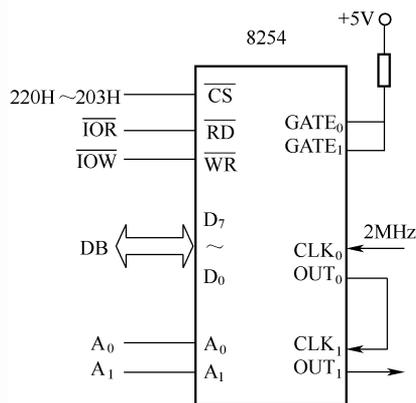


图 5.17 8254 应用电路