

# 《可编程定时器/计数器芯片》

## 总体方案设计

# 版本信息

序号	版本号	修改信息说明	修改人	时间
1	V 1.0	起草总体方案	韩高飞	2008.10.7
2	V 1.1	a. 修改计数器模块划分 b. 修改计数方式	韩高飞	2008.10.8
3	V 1.2	a. 修改工作时序 b. 修改概述	韩高飞	2008.10.9
4	V 1.3	a. 修改六种工作方式 b. 修改接口说明	韩高飞	2008.10.10
5	V 1.4	a. 修改验证方案	韩高飞	2008.10.11
6	V 1.5	a. 修改数据缓冲器模块 b. 修改计数器模块的数据读出模块 c. 修改验证方案	韩高飞	2008.10.25
7	V 1.6	加入全局异步复位信号	韩高飞	2008.11.18

# 目录

1 概述 .....	5
2 设计目标 .....	5
2.1 功能定义 .....	5
2.2 引脚说明 .....	6
2.3 基本时序 .....	7
2.3.1 写入操作 .....	7
2.3.2 读出操作 .....	8
3 顶层方案设计 .....	9
3.1 模块划分 .....	9
3.2 数据总线缓冲模块 .....	11
3.2.1 功能 .....	11
3.2.2 接口说明 .....	11
3.2.3 时序说明 .....	12
3.2.4 实现方案 .....	12
3.3 读写逻辑模块 .....	13
3.3.1 功能 .....	13
3.3.2 接口说明 .....	13
3.3.3 时序说明 .....	13
3.3.4 实现方案 .....	13
3.4 控制字寄存器模块 .....	14
3.4.1 功能说明 .....	14
3.4.2 接口说明 .....	14
3.4.3 时序说明 .....	14
3.4.4 实现方案 .....	14
3.5 计数器模块 .....	15
3.5.1 功能 .....	15
3.5.2 接口说明 .....	15
3.5.3 时序说明 .....	15
3.5.4 实现方案 .....	16
4 计数器六种工作方式说明 .....	19
4.1 计数结束输出正跃变信号 .....	19
4.1.1 工作特点 .....	19
4.1.2 工作时序 .....	20
4.2 单脉冲发生器 .....	21
4.2.1 工作特点 .....	21
4.2.2 工作时序 .....	21
4.3 分频器 .....	22
4.3.1 工作特点 .....	22
4.3.2 工作时序 .....	23
4.4 方波发生器 .....	24
4.4.1 工作特点 .....	24

4.4.2 工作时序.....	24
4.5 软件触发的单脉冲发生器.....	25
4.5.1 工作特点.....	25
4.5.2 工作时序.....	26
4.6 硬件触发的单脉冲发生器.....	26
4.6.1 工作特点.....	26
4.6.2 工作时序.....	27

# 1 概述

可编程定时器/计数器 8254 是一个应用于英特尔微处理器系统的可编程定时器/计数器芯片。它也可以应用于其它任何微处理器系统来解决一种最常见的问题：在软件控制下产生精确的时延，而代替在写软件时利用多重循环来实现的时延。使用者可以通过配置参数来满足自己的时延要求。该芯片会在一定的时延之后产生一个特定的信号来提醒微处理器进行下一步的操作。

然而，标准的 8254 最大工作频率为 10M。对于工作频率高于此的系统来说，8254 无法正常工作。本文旨在设计一个高速的可以应用于片上系统的可编程定时器/计数器 IP 核。

## 2 设计目标

### 2.1 功能定义

1. 内部含有 3 个可独立计数的 16 位计数器。
2. 支持 8 位数和 16 位数的计数要求。
3. 支持三种读写方式：
  - (1) 只读写低 8 位；
  - (2) 只读写高 8 位；
  - (3) 先读写低 8 位，再读写高 8 位。
4. 支持 2 进制计数方式和 10 进制计数方式。
5. 提供 6 种计数方式：
  - (1) 计数结束输出正跃变信；
  - (2) 单脉冲发生器；
  - (3) 分频器；
  - (4) 方波发生器；
  - (5) 软件触发的单脉冲发生器；
  - (6) 硬件触发的单脉冲发生器。
6. 可由编程者通过写入方式控制字来配置其所需要的计数方式。
7. 可由编程者通过写入读回命令字来配置起所需要的读写方式。

8. 可由编程者通过写入方式控制字和读回命令字来选择使用和读出三个计数器中的一个。

## 2.2 引脚说明

表1: 引脚名称及功能描述

序号	引脚名称	引脚方向	引脚编号	频率	功能说明
1	D0	I/O	8		双向数据总线, 用于传输控制字, 计数值和状态字。
2	D1	I/O	7		双向数据总线, 用于传输控制字, 计数值和状态字。
3	D2	I/O	6		双向数据总线, 用于传输控制字, 计数值和状态字。
4	D3	I/O	5		双向数据总线, 用于传输控制字, 计数值和状态字。
5	D4	I/O	4		双向数据总线, 用于传输控制字, 计数值和状态字。
6	D5	I/O	3		双向数据总线, 用于传输控制字, 计数值和状态字。
7	D6	I/O	2		双向数据总线, 用于传输控制字, 计数值和状态字。
8	D7	I/O	1		双向数据总线, 用于传输控制字, 计数值和状态字。
9	CLK0	I	9	50M	计数器 0 时钟信号
10	GATE0	I	11		计数器 0 控制信号, 外部输入, 当 GATE0 为高时计数, 当其为低时停止计数; 也可以在其上升沿完成初值重装的功能。
11	OUT0	O	10		计数器 0 输出, 1 位, 在六种不同的工作模式下, 含义不同。
12	CLK1	I	15	50M	计数器 1 时钟信号
13	GATE1	I	14		计数器 1 控制信号, 外部输入, 当 GATE1 为高时计数, 当其为低时停止计数; 也可以在其上升沿完成初值重装的功能。
14	OUT1	O	13		计数器 1 输出, 1 位, 在六种不同的工作模式下, 含义不同。
15	CLK2	I	18	50M	计数器 2 时钟信号
16	GATE2	I	16		计数器 2 控制信号, 外部输入, 当 GATE2 为高时计数, 当其为低时停止计数; 也可以在其上

					升沿完成初值重装的功能。
17	OUT2	0	17		计数器 2 输出, 1 位, 在六种不同的工作模式下, 含义不同。
18	RD	I	22		读控制信号, 低电平有效
19	WR	I	23		写控制信号, 低电平有效
20	A0	I	19		地址总线, 用于选择控制字寄存器和计数器。
21	A1	I	20		地址总线, 用于选择控制字寄存器和计数器。
22	CS	I	21		片选信号, 低电平有效
23	V <sub>cc</sub>	I	24		电源
24	GND	0	12		接地
25	Rst	I	25		全局异步复位信号
26	保留				

## 2.3 基本时序

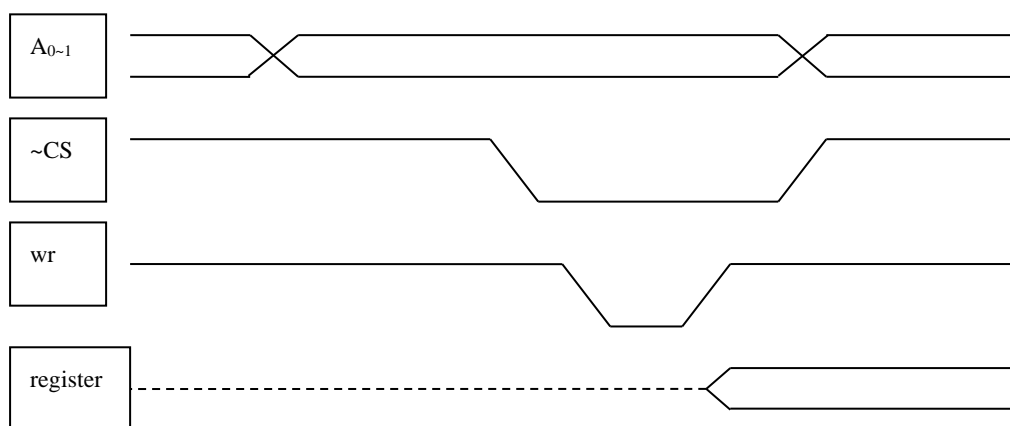
### 2.3.1 写入操作

当片选信号为高时, CPU 对芯片不作操作; 当片选信号为低时, CPU 可以对芯片进行操作。对于每一个计数器来说, 必须先写入方式控制字, 再写入计数初值。在地址总线选通某一个计数器的方式控制字缓冲寄存器或者读回命令控制字缓冲寄存器时, 并且写信号为低时, 该芯片随时接收数据。比如程序员控制 CPU 通过数据总线写入方式控制字, 并在有效建立时间之后写信号为低, 该芯片就会将数据存入控制字缓冲寄存器中。然后通过判断计数器选择字段再存入所选择计数器的方式控制字缓冲寄存器或读回命令字缓冲控制字中。再通过判断读回命令控制字的计数器选择字段将数据存入某计数器的读回命令控制字中。

然后写入计数初值, 直接存放在某选通计数器的数据寄存器中。当然该芯片会根据方式控制字的读写字段来控制只写入低 8 位, 或者只写入高 8 位, 以及先写入低 8 位再写入高 8 位。

当写信号为高时, 不执行任何写入操作。

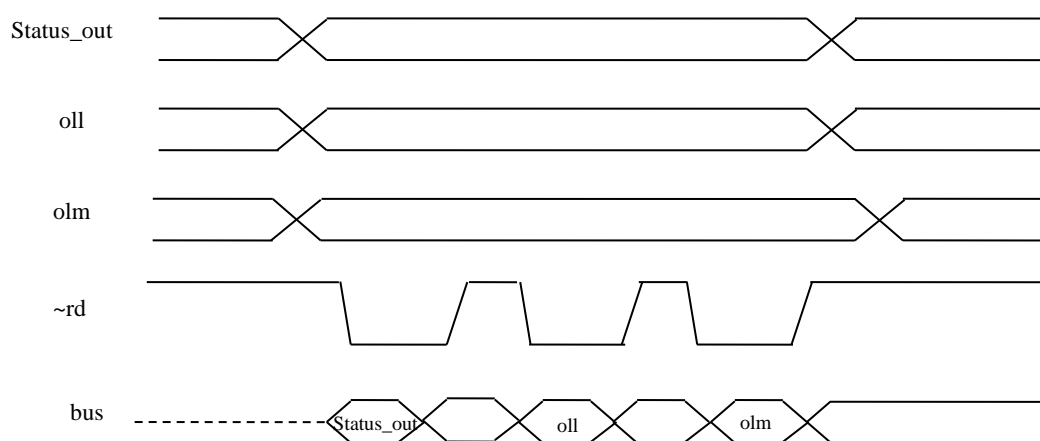
时序图如下：



### 2.3.2 读出操作

当读信号为低时，芯片分次把状态字寄存器和数据寄存器的值读出到数据总线上。在读出数据寄存器的数值时，芯片会根据读回命令控制字来锁存状态字寄存器和数据读出寄存器。并根据读写方式字段来控制只读出低 8 位，或者只读出低 8 位，以及先读出低 8 位再读出高 8 位。当读信号位高时，不执行任何读出操作。

时序图如下：





## 3 顶层方案设计

### 3.1 模块划分

根据可编程定时器要实现的根据不同工作方式计数和存储控制字,以及接收和输出计数值和状态字等等功能,将其划分为以下四个逻辑模块:

- 数据总线缓冲模块: 用于传输 cpu 和计数器之间需要交互的数据,数据总线为 8 位双向端口。
- 读写逻辑模块: 此模块为读写信号控制模块,并有根据两根地址总线对控制字寄存器和三个计数器进行选择的功能。
- 控制字寄存器模块: 用于存储来自于数据总线的方式控制字和都会命令字。并根据控制字中的计数器字段对三个计数器进行控制。
- 计数器模块: 完成计数控制和计数功能。其内部又有四个小模块,分别是:

(1) 控制逻辑模块: 根据输入的 gate 和 clk 信号以及控制字来控制计数单元计数。

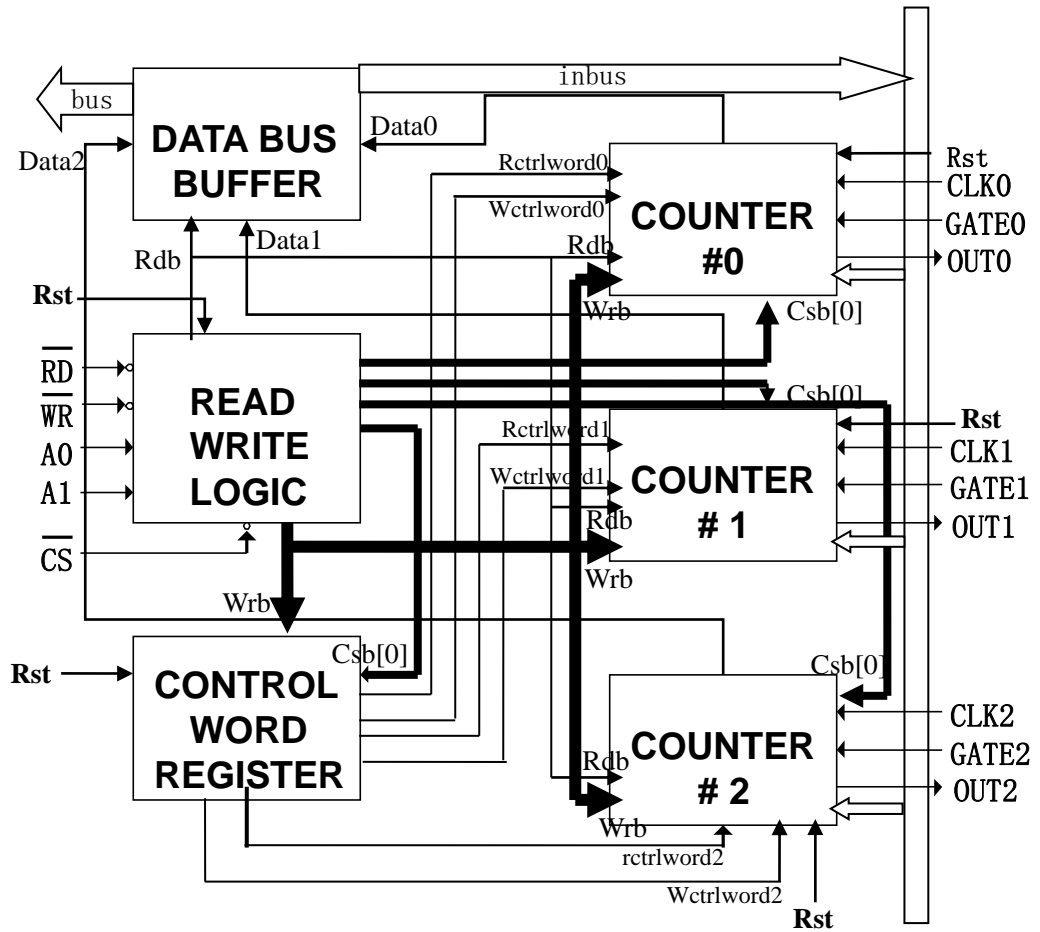
(2) 状态寄存器模块: 存储计数器状态信息。

(3) 初值写入寄存器模块: 存储计数初值。

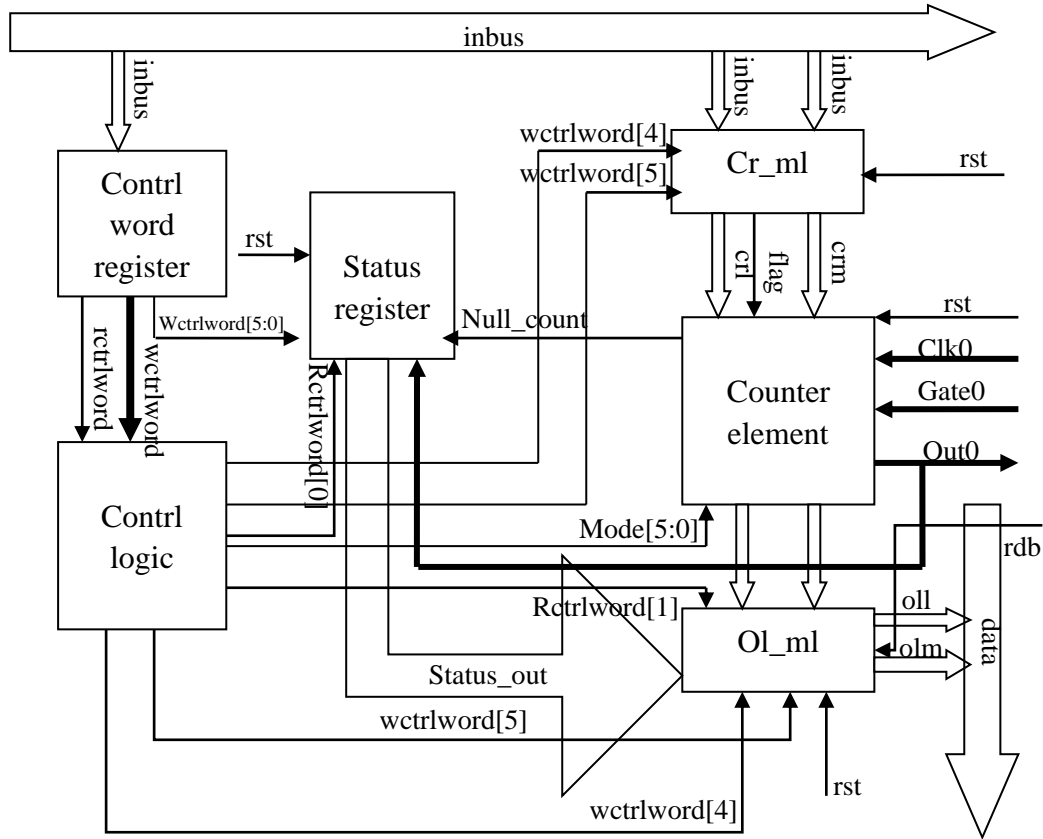
(4) 计数单元: 根据方式控制字完成六种计数方式。

(5) 计数值读出寄存器: 用于存储计数值停止计数时的计数值,以备读出。

总体模块划分图如下：



计数器模块的模块划分：



### 3.2 数据总线缓冲模块

#### 3.2.1 功能

数据总线缓冲器模块 (data\_bus\_buffer)：数据总线缓冲器的基本功能：

- (1) 外部数据总线无条件将数据写入内部输入数据线。
- (2) 在读信号 rd 为低时，数据输出到总线，在读出时，根据读回命令字中的计数器选择字来判断读那个计数器中的状态字和计数值，将 data 输出到 bus 中。

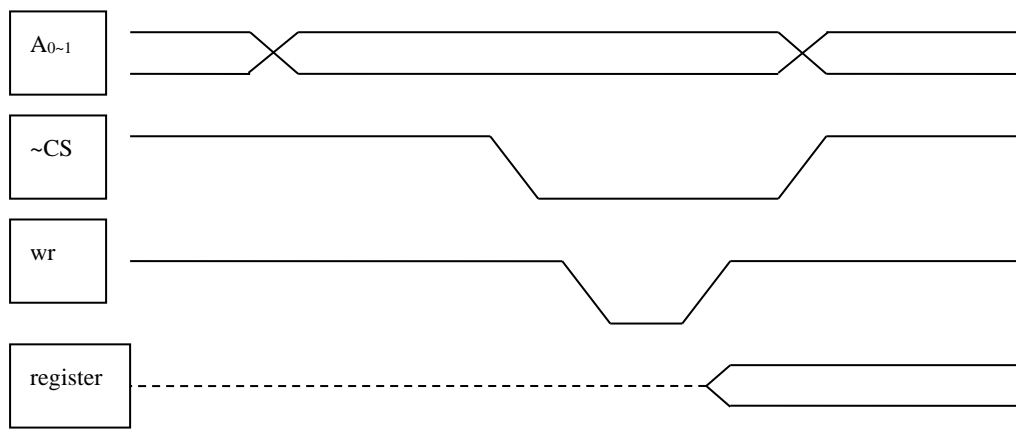
#### 3.2.2 接口说明

序号	信号名称	输入/输出	含义
1	outBus[7:0]	0	输出数据总线

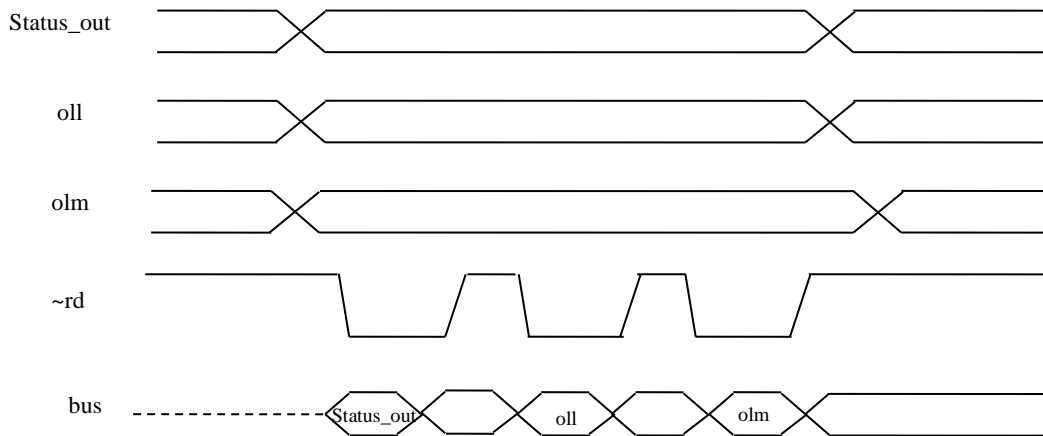
2	Rdb	I	数据写入控制信号，低电平有效
3	Data0[7:0]	I	计数器 0 的输出
4	Data1[7:0]	I	计数器 1 的输出
5	Data2[7:0]	I	计数器 2 的输出
6	Csrdcounter[2:0]	I	读回命令字中的计数器选择

### 3.2.3 时序说明

写时序:



读时序:



### 3.2.4 实现方案

设置一个三态门，写入数据时，无条件将数据从双向总线 bus 接收到写入内部数据线 inbus。读出数据，用 Rdb 控制，根据读回命令字的计数器选择字来判断，将数据从内部输出数据线 data0, data1, data2 按照选择读出到 bus。

### 3.3 读写逻辑模块

#### 3.3.1 功能

8254 内部的控制模块，当片选信号  $cs=1$  时，cpu 对 8254 不操作。当  $cs=0$  时，由 A1, A0 (接 CPU 地址线 A1, A0) 信号通过 2-4 译码器选择三个计数器的初值寄存器或控制字寄存器。读信号 rd 和写信号 wr 在  $\sim cs$  时分别产生 rdb 信号和 wrb 以便控制对选定寄存器的读/写操作。

#### 3.3.2 接口说明

信号名称	输入/输出	类型	含义
Rd	I	wire	读控制信号，低电平有效
Wr	I	wire	写控制信号，低电平有效
A0A1	I	wire	地址总线
Rdb	O	wire	产生的读控制信号
Wrb	O	wire	产生的写控制信号
Cs	I	wire	片选信号，低电平有效
Csc[3:0]	O	reg	译码输出

#### 3.3.3 时序说明

此电路为纯组合电路。无时序。

#### 3.3.4 实现方案

用一个 2-4 译码器对输入的二位地址进行译码，产生四位译码信号并连接到三个计数器和一个控制寄存器。在片选信号为低时，对输入地址进行译码，并接收写控制信号和读控制信号，并产生读写控制信号来控制数据总线，控制字寄存器和计数器。

## 3.4 控制字寄存器模块

### 3.4.1 功能说明

初始化编程时，由 CPU 写入方式控制字和读回命令字，以决定计数器的工作方式，和读写模式。此寄存器只能写入，不能读出。

### 3.4.2 接口说明

信号名称	输入/输出	类型	含义
Csc[3]	I	wire	片选信号，低电平有效
Wrb	I	wire	写控制信号，上升沿有效
Inbus[7:0]	I	Reg	内部总线，用于输入控制字
Wctrlword0[5:0]	0	Reg	存储计数器 0 的方式控制字
Wctrlword1[5:0]	0	Reg	存储计数器 1 的方式控制字
Wctrlword2[5:0]	0	Reg	存储计数器 2 的方式控制字
Rctrlword0[1:0]	0	Reg	存储计数器 0 的读回命令字
Rctrlword1[1:0]	0	Reg	存储计数器 1 的读回命令字
Rctrlword2[1:0]	0	wire	存储计数器 2 的读回命令字
Csrdcounter[2:0]	0	Reg	读回命令字中的计数器选择
Rst	I	Wire	全局异步复位信号

### 3.4.3 时序说明

在片选信号 csc[3]为低电平时，写信号上升沿到来，将数据写入，并根据其计数器选择位不同，存入不同的方式控制字寄存器和读回命令控制字寄存器中。

### 3.4.4 实现方案

当 csc[3]=1 时，将 inbus 的数据存入 ctrlword 寄存器中。再根据接收的控制字的最高两位判断是方式控制字或读回命令字，再转存入 wctrlword0~wctrlword2 和 rctrlword 中。如果是读回命令字，再产生一个已存

入信号 ralreg，高有效。再根据读回命令字的 D3~D1 来判断是写给哪个计数器的控制字，然后存入 rctrlword0~rctrlword2 中。将读回命令字的 d3~d1 存入 Csrddcounter，以备读出时提供计数器的选择信号。

## 3.5 计数器模块

### 3.5.1 功能

计数器模块：8254 有 3 个独立的计数器，每个计数器的结构完全相同。完成 16 位和 8 位计数，以及完成 2 进制和 10 进制计数方式。并根据计数方式不同 out 输出也不相同。

### 3.5.2 接口说明

以计数器 0 为例，其他计数器与此类同。

信号名称	输入/输出	类型	含义
Clk0	I	Wire	时钟, 下降沿计数
Gate0	I	Wire	计数使能信号
Out0	O	Reg	输出信号
inbus[7:0]	I	Wire	内部总线
Wctrlword0[5:0]	I	Wire	方式控制字
Rctrlword0[1:0]	I	Wire	读回命令字
Csc[0]	I	Wire	片选信号
Data[7:0]	O	Reg	输出状态字
wrb	I	wire	写控制信号, 上升沿有效
Rdb	I	wire	读控制信号, 下降沿有效
Rst	I	Wire	全局异步复位信号

### 3.5.3 时序说明

计数器模块的时序根据不同工作方式时序图不同，所以在第四节有不同工作

方式的时序图。

### 3.5.4 实现方案

计数器模块有 3 个独立的计数器，每个计数器的结构完全相同。计数器内部包括以下模块：

#### 3.5.4.1 控制逻辑模块：

##### (1) 接口说明

序号	信号名称	输入/输出	含义
1	Mode[5:0]	0	六种工作方式的独热码编址，高有效
6	wctrlword[2:0]	I	方式控制字

##### (2) 实现方案

把控制寄存拆分成读写方式字 `rd_mode`，计数方式字 `work_mode`，和计数数制 `bcd`。当 `A0A1` 有效并选择此计数器时，此控制器产生写入控制命令 `wl_en` 和 `wm_en`，或读出控制命令 `rl_en` 和 `rm_en`，用于控制写入寄存器和读出寄存器。当 `gate=1` 时，当 `clk` 上升沿到来时，启动计数单元。计数单元将按照计数方式开始计数。

#### 3.5.4.1 状态字寄存器模块

##### (1) 接口说明

序号	信号名称	输入/输出	含义
1	wctrlword[5:0]	I	方式控制字
2	status_latch	0	状态字锁存信号，低有效
3	status_out[7:0]	0	状态字输出信号
4	Out	I	计数器输出信号
5	null_count	I	计数值是否有效信号，低为有效

##### (2) 实现方案

在时钟上升沿到来时，写入控制字的 `d5—d0` 位直接赋给状态寄存器的 `d5—d0` 位，`out` 赋给 `d7` 位，`d6` 位表示数据是否写入计数器中。当读出控制到来时，如果其 `d4=0` 时（锁存计数器状态），将状态寄存器的内容锁入锁存器中以供读信号到来时，将锁存器的内容输出到内部总线。



### 3.5.4.1 初值写入寄存器模块

#### (1) 接口说明

序号	信号名称	输入/输出	含义
1	Crm[7:0]	0	高8位初值寄存器
2	cr1[7:0]	0	低8位初值寄存器
3	inbus[7:0]	I	内部数据线
4	cr1_en	I	低8位写使能信号，高有效
5	crm_en	I	高8位写使能信号，高有效
6	flag	0	初值写完标志信号，高有效
7	Csc[0]	I	片选信号，低有效
8	wr	I	写时钟，上升沿触发
9	Rst	I	全局异步复位信号

#### (2) 实现方案

当 csc[0]=0 时，在 cr1\_en 或（和）crm\_en 有效且当 wrb 上升沿到来，将内部数据线的数存入数据写入寄存器中，并使初值写完标志信号置高。

### 3.5.4.1 计数单元

#### (1) 接口说明

序号	信号名称	输入/输出	含义
1	olm_in[7:0]	0	计数当前值的高8位
2	oll_in[7:0]	0	计数当前值的低8位
3	crm[7:0]	I	高8位计数锁存值
4	cr1[7:0]	I	低8位计数锁存值
5	Flag	I	初值写完标志信号，高有效
6	mode[5:0]	I	六种工作方式的独热码编址，高有效
7	Gate	I	计数使能信号，高有效
8	Clk	I	计数时钟
9	Out	0	计数器的输出
10	null_count	0	计数值是否有效，低表示有效
11	Rst	I	全局异步复位信号

#### (2) 实现方案

在初值写完标志信号为高时，把初值寄存器中的高 8 位和低 8 位分别写入计数单元的 16 位寄存器 ce\_reg 中，当 gate=1 有效时，开始计数。本单元包含六种工作方式。根据工作方式字选择一种。并根据需要将 ce\_reg 中的内容输出到输出寄存器中。

### 3.5.4.1 计数值读出寄存器模块

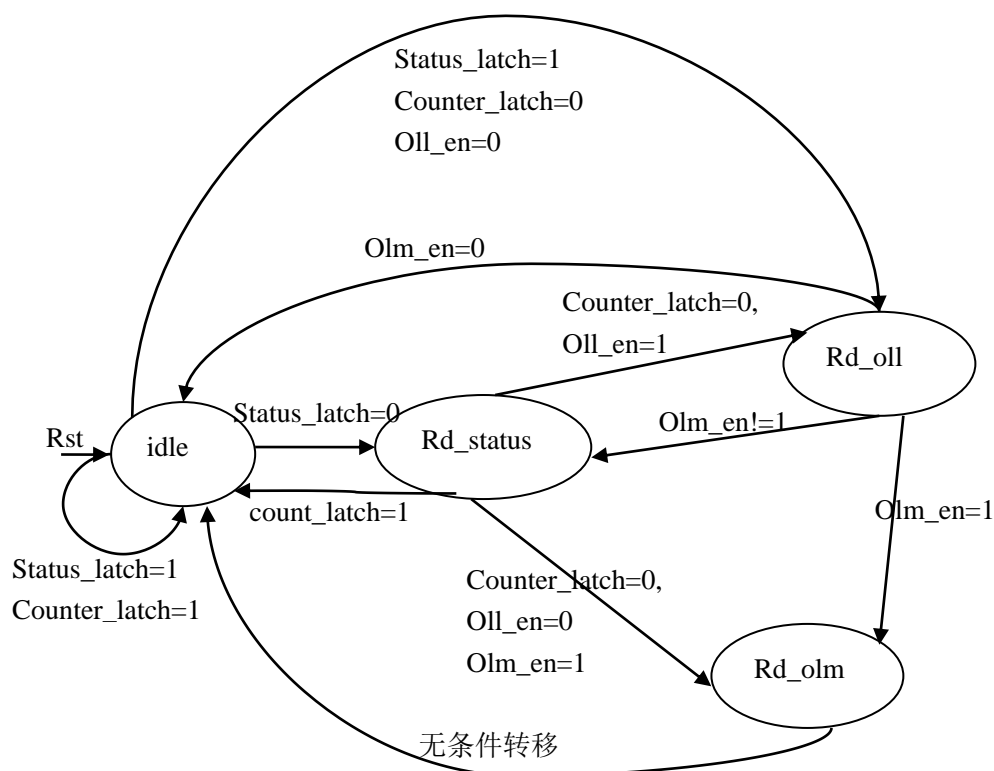
#### (1) 接口说明

序号	信号名称	输入/输出	含义
1	Data[7:0]	0	输出数据线
2	Status_out[7:0]	0	状态字寄存器
3	olm_in[7:0]	I	计数当前值的高 8 位
4	oll_in[7:0]	I	计数当前值的低 8 位
5	count_latch	I	计数值锁存信号，低有效
6	status_latch	I	锁存状态字控制信号，低电平有效
7	Rdb	I	数据读出控制信号，低电平有效
8	oll_en	I	低 8 位计数值读出控制信号，高电平有效
9	olm_en	I	高 8 位计数值读出控制信号，高电平有效
10	Rst	I	全局异步复位信号

#### (2) 实现方案

当读出方式字的 d5=0 (count\_latch=0) 时，将计数器当前值锁存到锁存器 oll 和 olm 中。当读信号下降沿到来时，将状态字和计数值输出到输出数据线 data 中，在输出时，可根据计数器的状态字和计数值是否锁存来判断是否读出；通过读出方式，来确定是只读低 8 位，或者只读高 8 位，又或者先读低 8 位，后读高 8 位。

用一个状态机实现：



## 4 计数器六种工作方式说明

### 4.1 计数结束输出正跃变信号

#### 4.1.1 工作特点

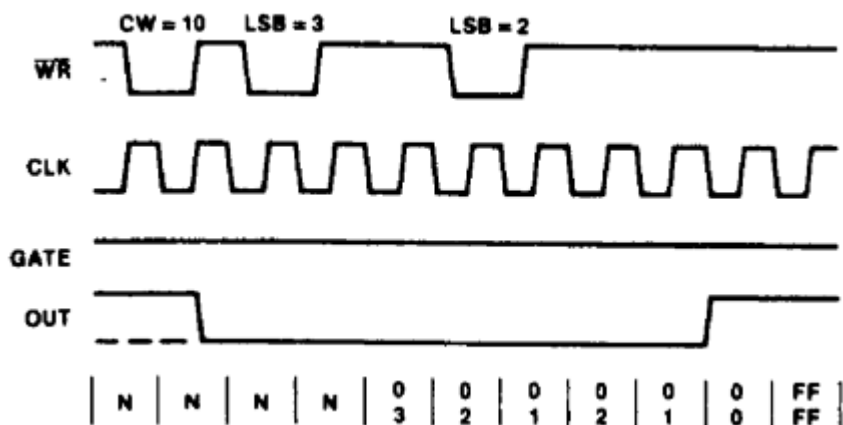
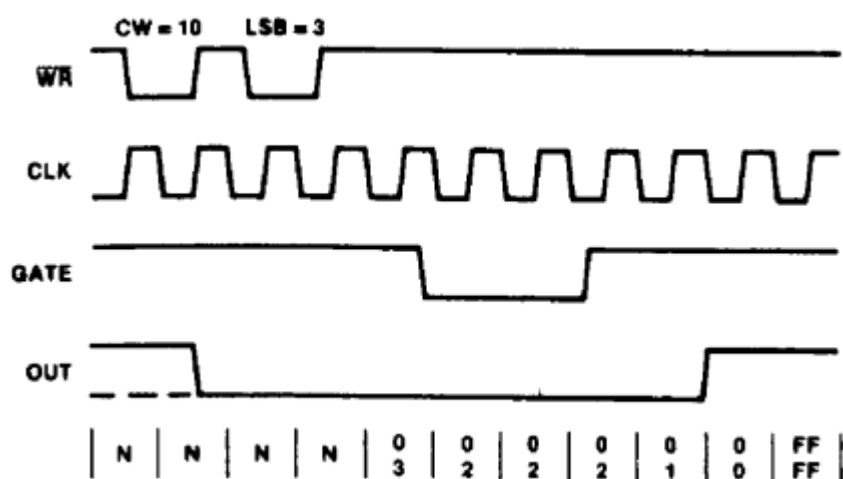
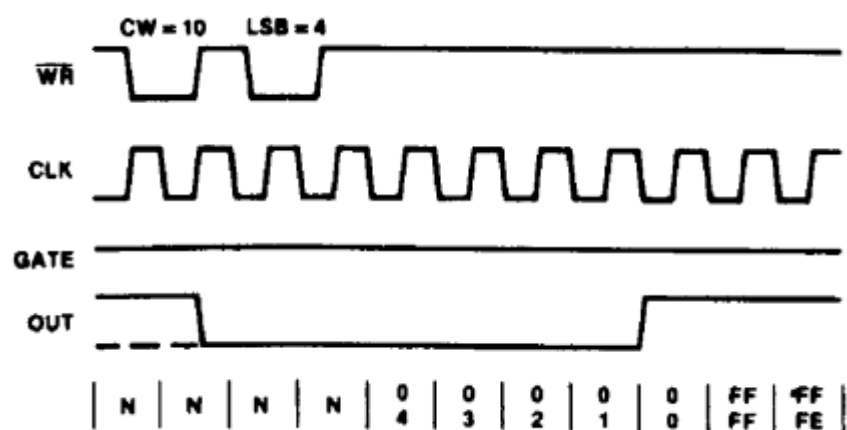
(1) 写入控制字后，OUT 端输出为低电平。写入计数初值后，OUT 端保持低电平，计数器开始对 CLK 脉冲进行减 1 计数。当计数值减至为 0 时，OUT 端变为高电平，此信号可用于向 CPU 发出中断请求。方式 0 不具备“初值自动重装”的功能。当计数值减至为 0 以后，再来一个 CLK 脉冲，计数值减 1 变成-1，再来一个 CLK 脉冲，计数器再次减 1，计数值变成-2；……。

(2) 如果在计数过程中，改变计数初值，刚在写入新计数初值后，计数器将以该值为计数初值，重新开始作减 1 计数。

(3) GATE 为计数控制信号，当 GATE=1 时，允许计数；GATE=0 时，停止计

数。

### 4.1.2 工作时序



## 4.2 单脉冲发生器

### 4.2.1 工作特点

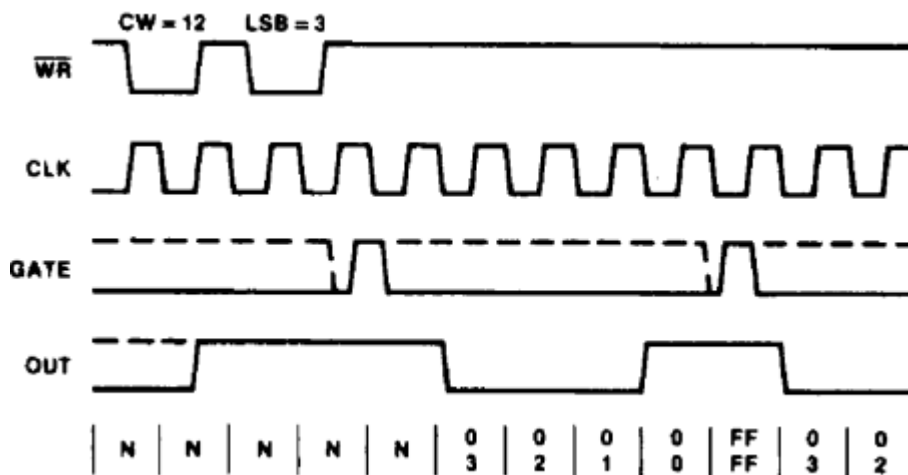
该方式是由外部门控脉冲（硬件）启动计数。其特点是：

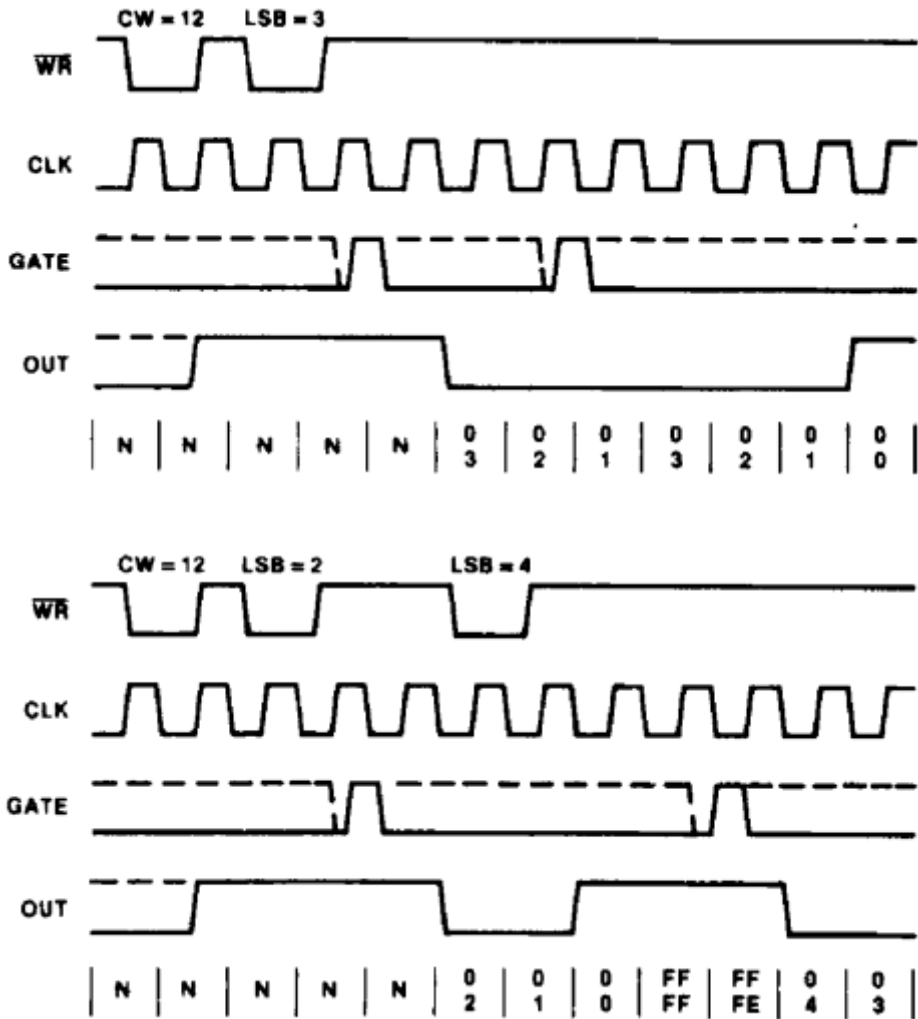
写入控制字后，OUT 端输出高电平。写入计数初值后，OUT 端保持高电平，计数器由 GATE 的上升沿启动。GATE 启动之后，OUT 变为低电平，每来一个 CLK，计数器减 1，当计数值减到 0 时，OUT 输出高电平，从而在 OUT 端输出一个负脉冲，负脉冲宽度为计数初值乘以 CLK 脉冲周期。方工 1 没有“初值自动重装”的功能，只有当 GATE 再次出现 0→1 的跃变时，才能完成初值重装。

在计数器未减到 0 时，门控信号 GATE 又来一个正脉冲，计数初值将重新装入计数器计数器从初始值开始重新作减 1 计数。

在计数过程中，程序员可装入新的计数初值，此时计数过程不受影响。只有当 GATE 再次出现 0→1 的跃变后，计数器才能按新的计数初值作减 1 计数。

### 4.2.2 工作时序





## 4.3 分频器

### 4.3.1 工作特点

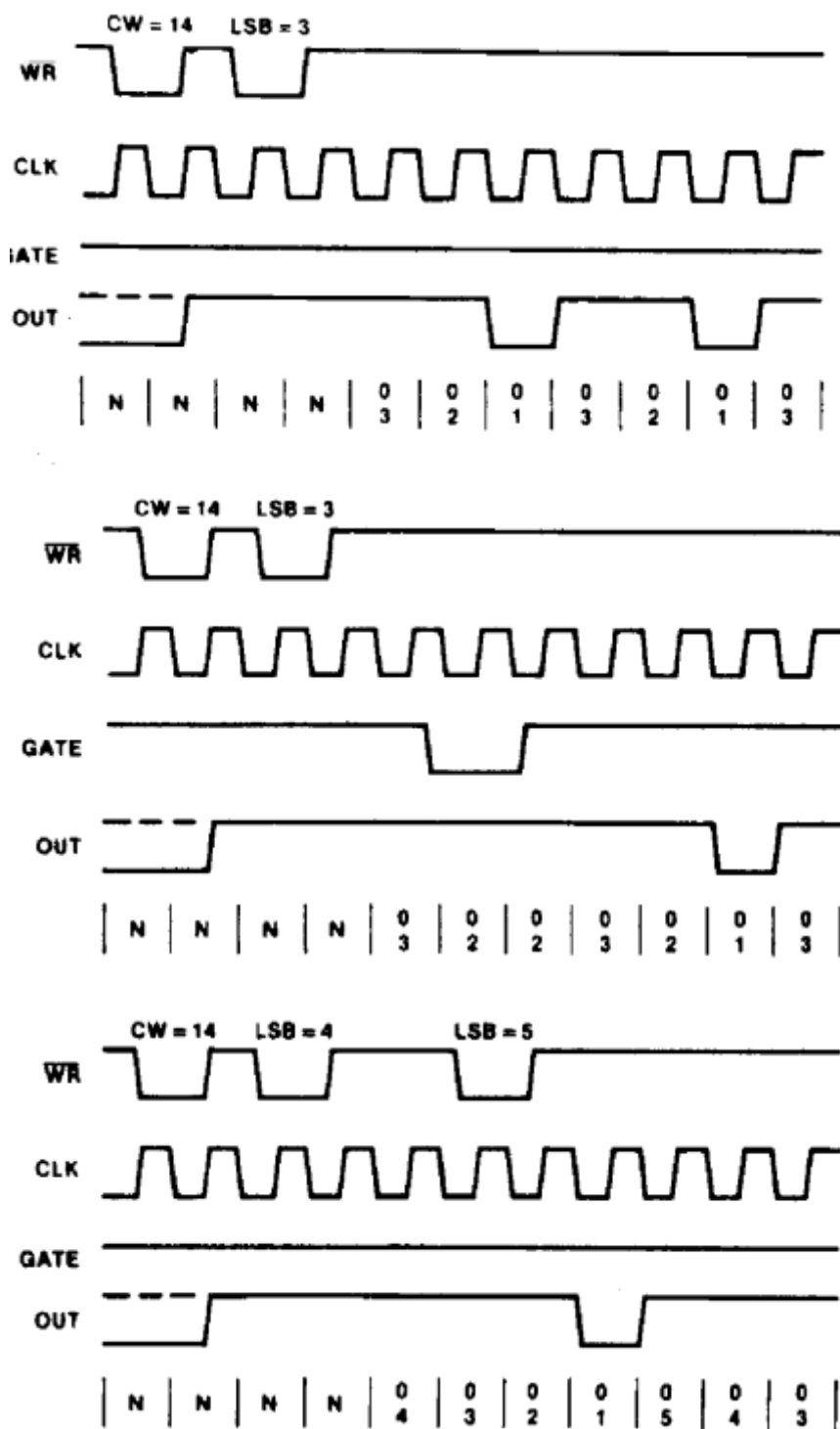
方式 2 的特点是计数器有“初值自动重装”的功能，所以能够输出固定频率的脉冲。其工作特点如下：

(1) 写入控制字后，OUT 输出为高电平。写入计数初值后，如果 GATE 为高电平，计数器开始减 1 计数，当计数值减到 1（注意 1）时，OUT 输出为低电平，维持一个 CLK 周期，又变为高电平，且计数初值自动重装，计数器开始重新计数，周而复始。OUT 端输出连续的负脉冲，负脉冲宽度为一个 CLK 周期。

(2) 如果在减 1 计数的过程中，GATE 变低，则暂停计数，GATE 的上升沿使计数器恢复初值，并从初值开始减 1 计数。

(3) 在计数过程中，如果 GATE 为高电平，程序员写入新的计数初值，不会影响正在进行的减 1 计数过程，只有计数器减到 1 之后，计数器才装入新的计数初值，并且按新的计数初值开始计数。

### 4.3.2 工作时序

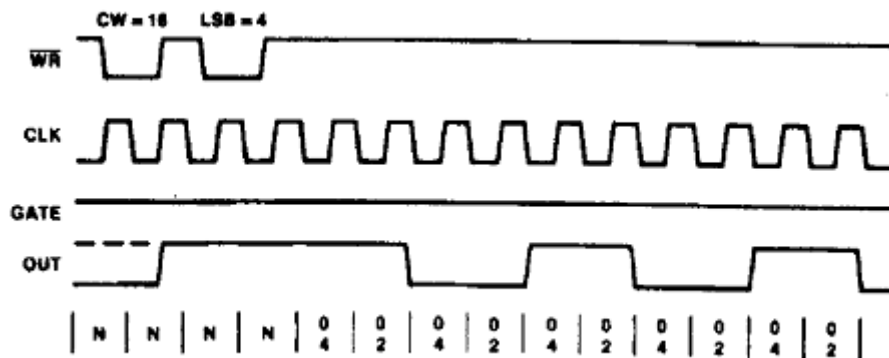


## 4.4 方波发生器

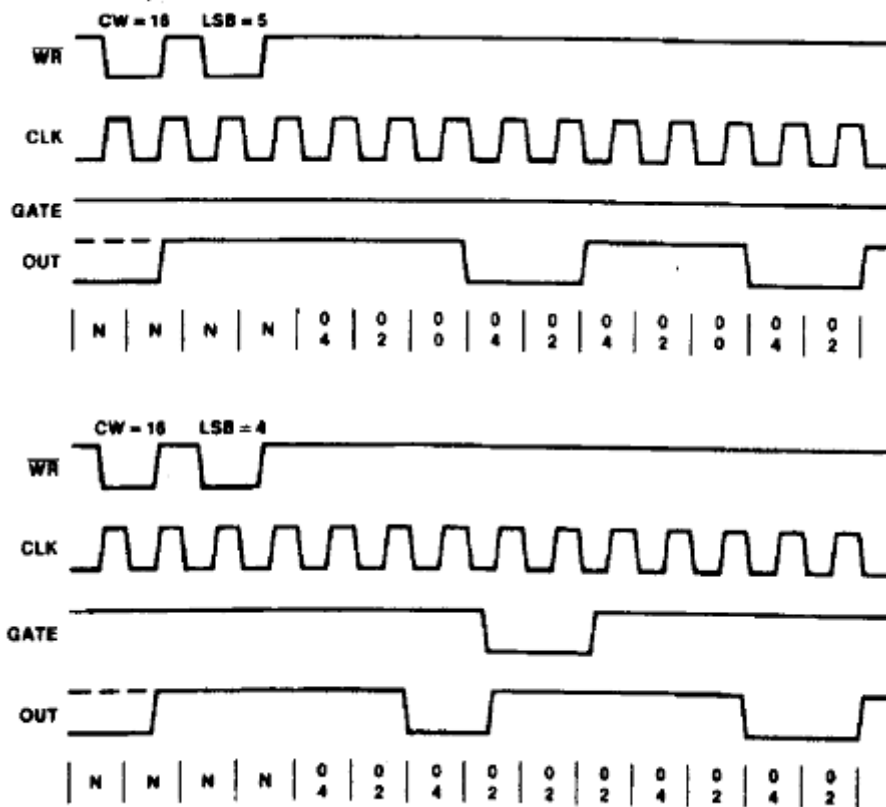
### 4.4.1 工作特点

这种方式是在编程后重复地循环计数，输出波形为方波。如果初始计数值为偶数，每个时钟输入脉冲使计数器减 2，达到计数终点时输出电平改变。如果初始计数值为奇数，则输出高电平时第一个时钟输入脉冲使计数器减 1，随后每个输入脉冲使计数器减 2；输出为低电平时第一个时钟输入脉冲使计数器减 3，随后每个输入脉冲使计数器减 2，到达计数终点时输出电平改变，计数器自动初始化后继续计数。用 GATE 的上升沿初始化并开始计数，GATE 为低电平时停止计数。

### 4.4.2 工作时序





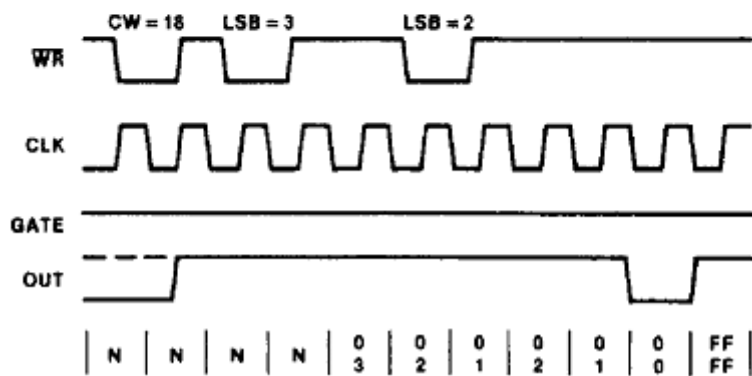
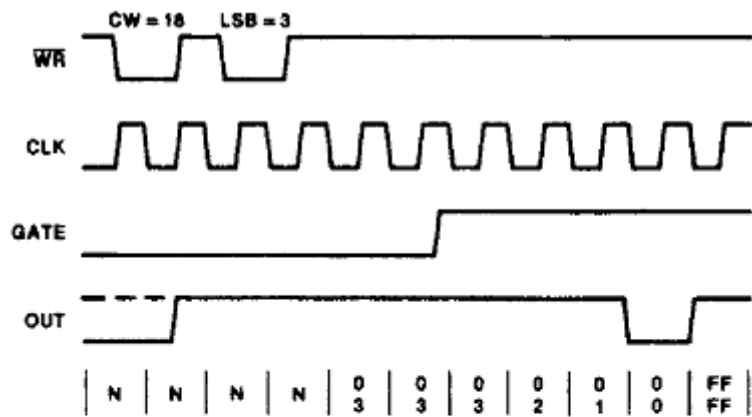
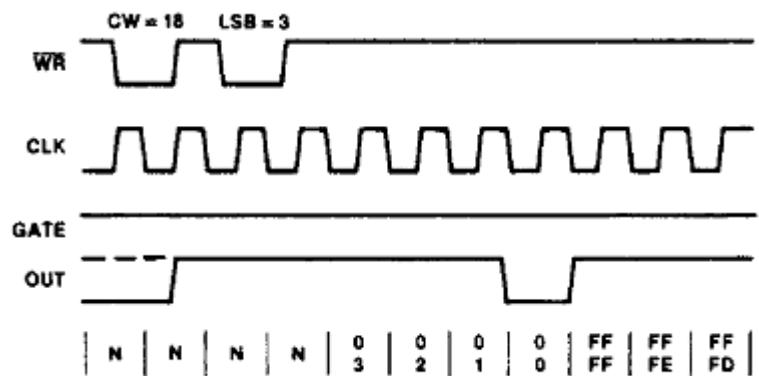


## 4.5 软件触发的单脉冲发生器

### 4.5.1 工作特点

软件启动选通脉冲输出。编程后自动启动，计数到终点后输出一个时钟周期的低电平脉冲。用 GATE 的上升沿初始化并开始计数，GATE 为低电平时停止计数。

## 4.5.2 工作时序



## 4.6 硬件触发的单脉冲发生器

### 4.6.1 工作特点

硬件启动选通脉冲输出。编程后，等待 GATE 上升沿进行初始化并开始计数，

计数到终点后输出一个时钟周期的低电平脉冲，计数器开始计数后不受 GATE 信号电平的影响，这种选通脉冲的输出可用 GATE 的上升沿多次触发。在工作方式控制字中，如果设置计数器锁存操作，则该控制字中工作方式选择位 M1、M0 和计数方式选择位 BCD 无效。即设置锁存操作时不影响计数器的工作方式，计数器锁存操作，是在计数器计数过程中，在不影响正在进行的计数操作的条件下，把当前的计数值锁存到寄存器，供 CPU 读取，这时在工作方式控制字中，SC1、SC0 指定要锁存的计数器，RL1、RL0=00 表示锁存操作，其余 4 位无效，计数器按原来设定的方式工作。

### 4.6.2 工作时序

