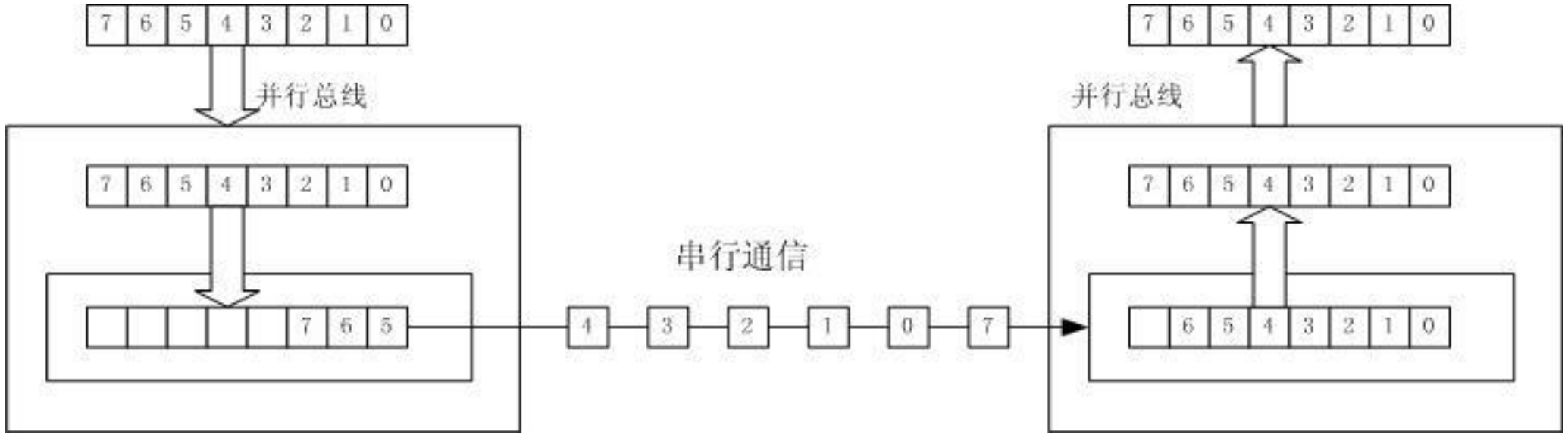
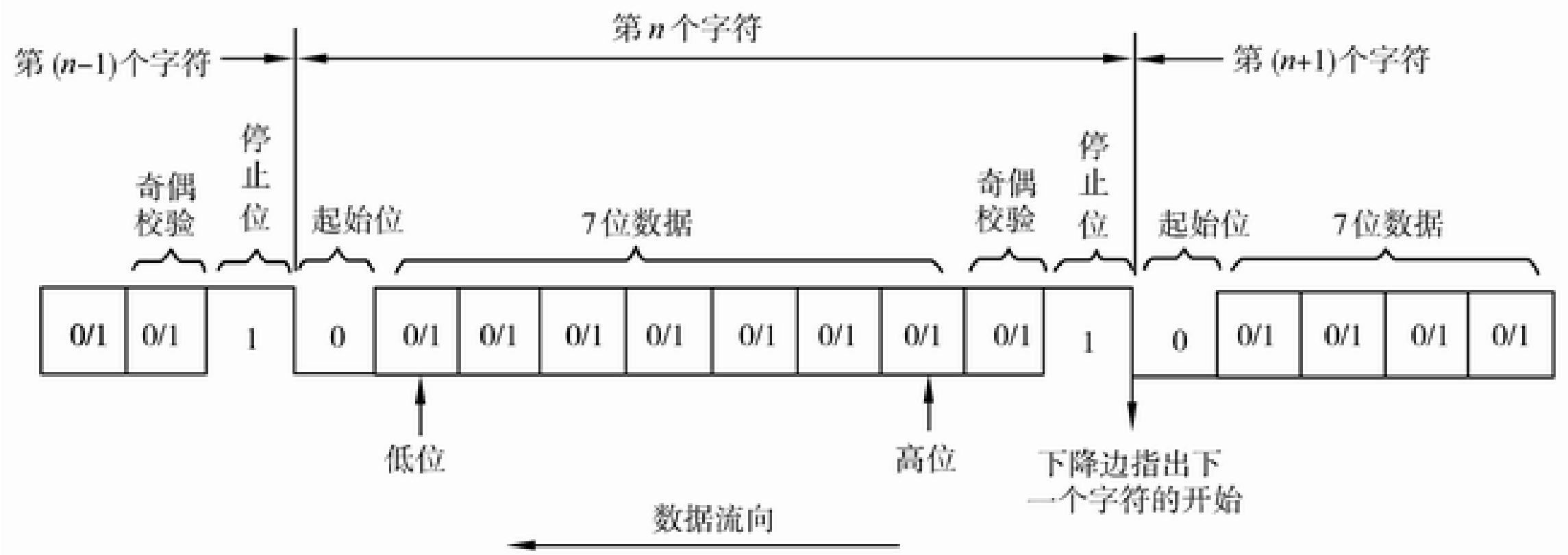
接口技术

第八章

串行接口及应用


7 6 5 4 3 2 1 0
并行总线
7 6 5 4 3 2 1 0
7 6 5
串行通信
4 3 2 1 0 7
7 6 5 4 3 2 1 0
并行总线
7 6 5 4 3 2 1 0
6 5 4 3 2 1 0

异步串行通信协议


第 (n−1)个字符
第 n 个字符
第 (n+1)个字符
奇偶校验
停止位
起始位
7位数据
奇偶校验
停止位
起始位
7位数据
0/1
0/1
1
0
0/1
0/1
0/1
0/1
0/1
0/1
0/1
0/1
1
0
0/1
0/1
0/1
0/1
低位
高位
下降边指出下一个字符的开始
数据流向

波特率与比特率

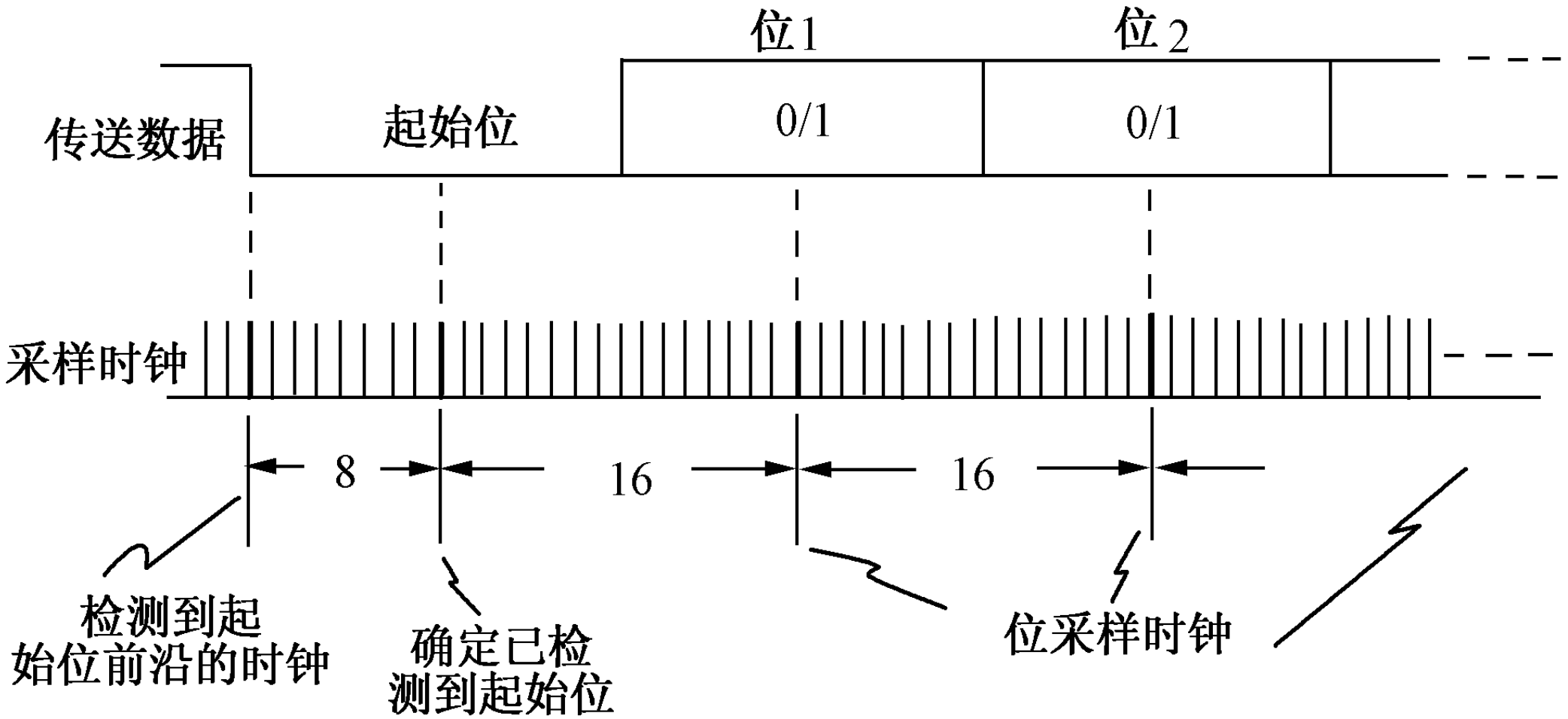
- 比特率是数字信号的传输速率，它用单位时间内传输的二进制代码的有效位（bit）数来表示，其单位为每秒比特数bps（bit per second）、每秒千比特数（Kbps）或每秒兆比特数（Mbps）来表示。
- 波特率指每秒传输的符号数，指数据信号对载波的调制速率，它用单位时间内载波调制状态改变次数来表示，其单位为波特（Baud）。
- 波特率与比特率的关系为：比特率=波特率×单个调制状态对应的二进制位数。

- 通信产品中有一个标准波特率系列，即最常用的波特率，标准波特率系列为110、300、600、1200、1800、2400、4800、9600、19200、38400、57600、115200。波特率提高后，数据传输的速度加快，但是信号传输的距离则相应地缩短。

- 假定比特率为9600bps，异步方式下，每个字符对应1个起始位、7个数据位、1个奇偶校验位和1个停止位。
- ①每传输一个二进制位需要的时间？
- ②数据传输效率是多少？
- ③每秒钟能传输的最大字符数为多少？
- ④每秒钟有效数据传输位是多少？

- ①每传输一个二进制位需要的时间为1÷9600 = 0.0001042秒=0.1042毫秒。
- ②数据传输效率是7 /（1+7+1+1）= 70%。
- ③传送一个字符就需要10个二进制信息位。每秒钟能传输的最大字符数为9600÷10=960个字符。
- ④每秒钟有效数据传输位=9600×70%=6720 bit。

波特率因子



奇偶校验

- 奇偶校验是以字符为单位进行校验。在每一个字符传输过程中，增加一位作为校验位。发送方和接收方可以约定是否采用奇偶校验、以及采用奇校验还是偶校验。
- 采用奇校验时，总数必须为奇数；采用偶校验时，总数必须为偶数。如果总数不匹配，则说明数据传送过程中出现了错误。
- 偶校验时，发送字符10010001b，则校验位为1。

同步串行通信协议

- SDLC / HDLC协议
- 规定以01111110为标志字段作为起止的标志。

可编程串行通信接口

- 8250/16550功能和基本原理
- 通用异步收发传输器（Universal Asynchronous Receiver/Transmitter）
- 二者的区别在于8250的发送和接收数据缓冲器只有一个字节，每发送或接收一个字节都要求CPU来干预，而16550增加了16个字节的FIFO（First Input First Output）发送和接收数据缓冲器，可以连续发送或接收16个字节的数据；16550传输更快，更适合高速系统通信接口应用。8250的最大通信速率为19200bps，而16550的最大通信速率可达115200bps。

特点

- 是可编程的串行异步通信接口，支持全双工通信，内部结构分为发送模块，接收模块和控制模块。
- 可编程设置异步通信格式，如字符数据位数、奇偶校验模式和停止位宽度等。
- 内部有时钟发生器电路，可编程选择数据传输率，8250的数据传输速率最大为19200bps，16550的数据传输速率最大为115200bps。
- 具有自动奇偶校验、溢出检查和帧格式检查等电路。
- 具有中断优先级控制逻辑，支持4级中断。
- 具有控制MODEM功能和完整的状态报告功能。
- 16550增加了FIFO模式。

引脚信号线

- （1）与CPU或系统连接的信号
- D7~D0： 8位双向数据线，与计算机系统数据总线直接相连，用于CPU和芯片之间命令、状态和数据的传送。
- CS0、CS1、CS2#：输入，片选信号。当CS0=1，CS1=1，CS2#＝0这3个条件同时满足时，芯片工作。
- A2~A0：输入，地址线，当片选信号有效时，由A2～A0组合选择内部寄存器。
- ADS#：输入，地址选通信号。当ADS#从高电平变为低电平时，锁存CS0、CS1、CS2#以及A2～A0的输入状态，保证读写操作期间的地址稳定。

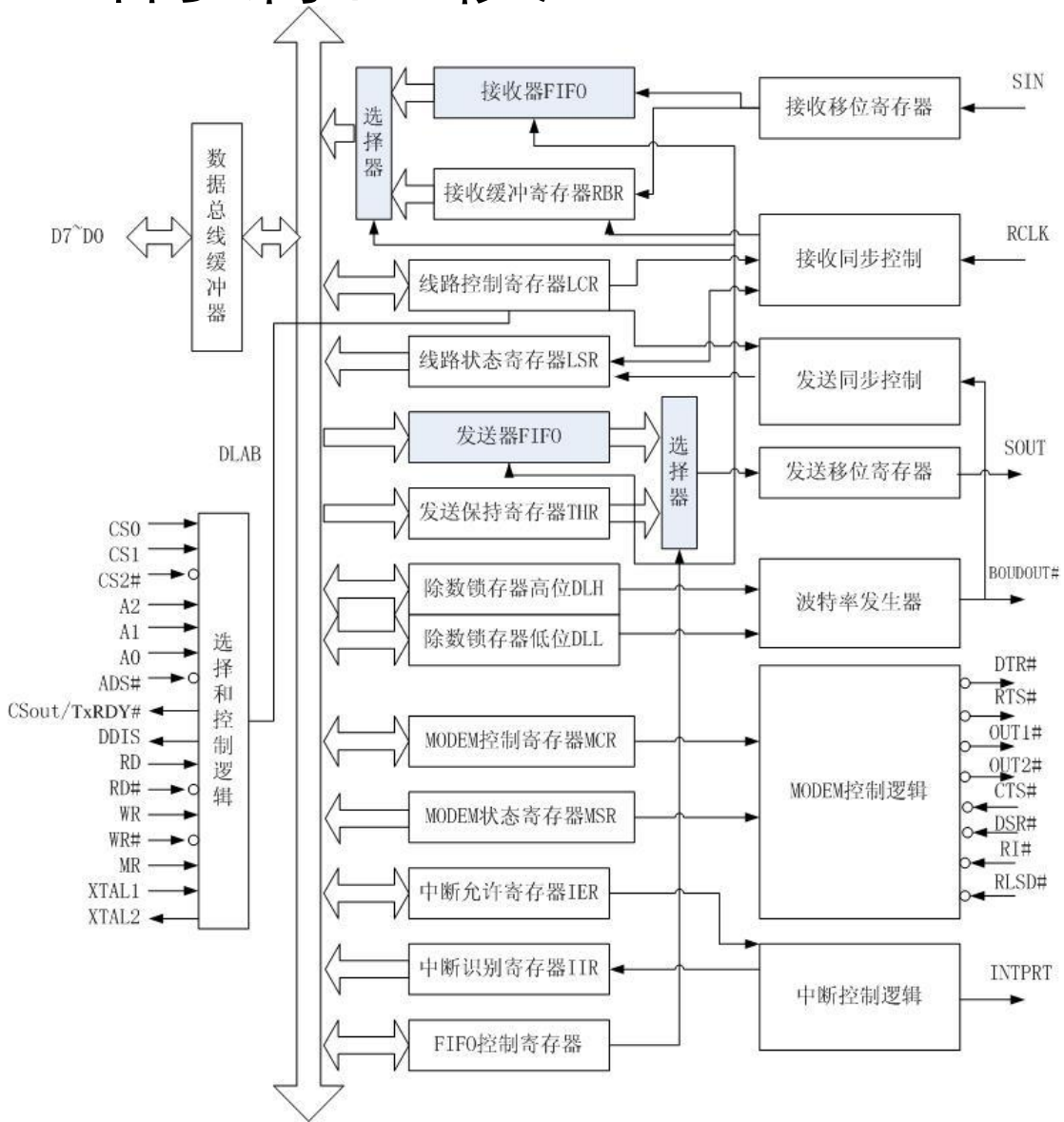
- CSOUT/TxRDY#：输出，片选输出信号，CSOUT=1时表示8250被选中，进行工作，通常将其悬空。在16550中表示发送准备就绪，常用于DMA方式数据传输。
- NC/RxRDY#：8250中此引脚没用到，悬空。16550中为输出，低有效，表示输出接收数据就绪，常用于DMA方式数据传输。
- RD和RD#：输入，数据输入选通信号，即读控制信号，这两个信号功能一样，信号电平不同。在芯片被选中时，如果RD为高电平，或者RD#为低电平，CPU就从芯片的寄存器中读出数据，寄存器由A2 ~ A0决定。
- WR和WR#：输入，数据输出选通信号，即写控制信号，这两个信号功能一样，信号电平不同。在芯片被选中时，如果WR为高电平，或者WR#为低电平，CPU把数据写入芯片的寄存器，寄存器由A2 ~ A0决定。

- DDIS：输出，驱动器禁止信号。
- INTRPT：输出，中断请求信号。在满足一定条件下（如接收数据准备好，发送保持寄存器空以及允许中断时）变成高电平，产生中断请求。
- OUT1、OUT2：输出，由MODEM控制寄存器的第2、3位决定。
- MR：输入，复位信号。一般接到系统的RESET，使芯片和系统同时复位。

- （2）时钟与传送速率控制
- XTAL1：输入，时钟信号。外部晶体振荡电路产生的1.8432MHz信号送到芯片的XTAL1端，作为芯片的基准工作时钟。
- XTAL2：输出，时钟信号。
- BAUDOUT#：输出，时钟信号。外部输入的基准时钟，经芯片内部波特率发生器（分频器），分频后产生发送时钟，并经BAUDOUT#引脚输出。
- $f_{\text{工作时钟}} = f_{\text{基准时钟}} \div$ 除数锁存器 = 波特率 × 16
- RCLK：输入，时钟信号。可接收由外部提供的接收时钟信号。若采用芯片内部的发送时钟作为接收时钟，则只要将RCLK引脚和BAUDOUT#引脚直接相连即可。

- （3）串行数据发送和接收信号
- SOUT：输出，串行数据输出。与TxD连接，但中间需要经过一个TTL电平到RS-232电平的转换。
- SIN：输入，串行数据输入信号。与RxD连接，但中间需要经过一个RS-232电平到TTL电平的转换。

8250/16550的编程模型



- 8250内部有11个可访问的寄存器，16550有12个，多了一个FIFO控制寄存器。由于芯片只有3根地址线A2～A0，也就是3位地址码，只能产生8个地址，不够每个寄存器分配一个地址，因此部分寄存器需要共用地址。共用地址的寄存器用线路控制寄存器LCR的D7位DLAB位作为标志区分。

- DLAB=1时，A2、A1、A0=000B表示波特率除数寄存器低字节DLL，A2、A1、A0=001B表示波特率除数寄存器高字节DLH；
- DLAB=0时，A2、A1、A0=000B对应于发送保持寄存器（THR）和接收缓冲寄存器（RBR）。CPU写入该地址的字节作为发送保持寄存器；而CPU从该地址读入的字节是接收缓冲寄存器。
- DLAB=0时，A2、A1、A0=001B对应于中断允许寄存器（IER）。
- A2、A1、A0=010B对应于中断识别寄存器（IER）和FIFO控制寄存器（FCR）。CPU写入该地址的字节作为FIFO控制寄存器；而CPU从该地址读入的字节是中断识别寄存器。

A2～A0	DLAB	访问的寄存器	基址为 3F8 时各寄存器地址	基址为 2F8 时各寄存器地址
000	0	接收缓冲寄存器 RBR(读)， 发送保持寄存器 THR(写)	3F8	2F8
	1	波特率除数寄存器 DLL(低字节)	3F8	2F8
001	0	中断允许寄存器 IER	3F9	2F9
	1	波特率除数寄存器 DLM(高字节)	3F9	2F9
010	X	中断识别寄存器 IIR(读) FIFO 控制寄存器 FCR(写)（16550 专有）	3FA	2FA
011	X	线路控制寄存器 LCR	3FB	2FB
100	X	MODEM 控制寄存器 MCR	3FC	2FC
101	X	线路状态寄存器 LSR	3FD	2FD
110	X	MODEM 状态寄存器 MSR	3FE	2FE
111	X	暂存寄存器	3FF	2FF

寄存器的格式

- 发送保持寄存器（Transmitter HoldIng Register，THR）
- CPU将要发送的数据字节输出到这个寄存器，串行接口电路就将这个数据字节转换成串行信号传送给对方。数据字节的有效位可以是5位、6位、7位或8位。
- 发送时，CPU将待发送的字符写到THR后，然后进入发送移位寄存器，在发送时钟的作用下，按起始位、数据位、校验位、停止位的顺序，从SOUT引脚逐位输出。一旦THR的内容送到发送移位寄存器TSR后，THR变空，就使LSR的THRE位置1，产生中断请求。THRE位置1时，表示CPU可以发送下一个字符。CPU向THR写入一个字符后，THRE置0。

- 接收缓冲寄存器（Receiver Buffer Register，RBR）
- 串行接口电路接收到发送方的一个数据字节后，CPU可以从这个寄存器读取这个字节。接收时，串行数据在接收时钟作用下，从SIN引脚以串行移位的方式输入到接收移位寄存器RSR，然后由RSR并行输入到接收缓冲寄存器RBR，一旦RBR变满，就使LSR的DR位置1。DR位为1时，表示CPU可以读取数据字符。CPU从RBR读取一个字符后，DR置0。

7	6	5	4	3	2	1	0
RFE	TEMT	THRE	BI	FE	PE	OE	DR

RFE	接收 FIFO 出错（只对 16550 有效）。等于 1 时，表示接收 FIFO 出错
TEMT	发送移位器空。等于 1 时，表示发送器空。
THRE	发送保持寄存器空。等于 1 时，表示 THR 中的数据字节已被取走。数据字节写入到 THR 时，此位被清零。
BI	间断识别指示。等于 1 时，接收线 SIN 空闲的时间超过了传送一个字符的时间，对方发送过程出现了间断。
FE	帧格式错。等于 1 时，表示传输的数据格式错误。
PE	奇偶校验错。等于 1 时，表示奇偶校验错误。
OE	覆盖错。等于 1 时，表示接收到有效的数据但被丢失。
DR	接收缓冲寄存器有效。等于 1 时，已接收到一个数据字节放入到 RBR 中。读取 RBR 后，此位被清零。

- LSR的第0位DR和第5位THRE是最基本的指示位。只有DR=1时，CPU从RBR中读取的数据字节才是从发送方发出的有效数据字节。DR=0时，CPU从RBR中读出的是“旧的”数据字节或无效的数据字节。只有THRE=1时，CPU写到THR的数据字节才会被正确地发送出去。THRE=0时，THR中的数据字节还没有被取走，不能向其中写入新的数据字节。

- 假定8250/16550基地址为3F8H（对应于A2、A1、A0=000B），那么发送保持寄存器、接收缓冲寄存器的地址为3F8H，而线路状态寄存器的地址为3FDH（A2、A1、A0=101B）。在不考虑串口发送、接收出错的情况下，试编写程序从串行接口发送和接收一个字符AL。

;发送程序片段

```
SendByte        PROC
                PUSH    AX                  ;要发送的数据压栈
                MOV     DX, 3fdh            ;LSR端口号—> DX
SendByteBusy:
                IN      AL, DX              ;读入端口状态—>AL
                TEST    AL, 20h             ;测试状态中的第5位THRE
                JZ      SendByteBusy        ;THRE=0, 继续查询
                POP     AX                  ;要发送的数据出栈
                MOV     DX, 3f8h            ;THR端口号—> DX
                OUT     DX, AL              ;数据发送到THR
 RET
SendByte        ENDP
```

;接收一个字符到AL中的程序片段

```
ReceiveByte     PROC
                MOV     DX, 3FDH            ;LSR端口号—>DX
NoByteReceived:
                IN      AL, DX              ;读入端口状态—> AL
                TEST    AL, 01H             ;测试状态中的第0位DR
                JZ      NoByteReceived      ;DR=0, 继续查询
                MOV     DX, 3F8H            ;RBR端口号—> DX
                IN      AL, DX              ;读入RBR中的数据字节
                RET
ReceiveByte     ENDP
```

7	6	5	4	3	2	1	0
DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0

WLS1 WLS0	WLS1 WLS0=00b，字符长度为 5 位； =01b，字符长度为 6 位； =10b，字符长度为 7 位； =11b，字符长度为 8 位。
STB	=0，停止位长度为 1 位；=1，1.5 位或 2 位（字符长度为 5 位时采用 1.5 位停止位，字符长度为 6、7、8 位时采用 2 位停止位）。
PEN	=0，不使用奇偶校验。发送接收时没有校验位。
EPS	=0，奇校验； =1，偶校验。EP=0 时，此位无效。
SP	=1 时，奇偶校验位固定为 0 或 1。=0 时，设置校验位。
SB	=1 时，发送线 SOUT 设为 0 并保持至少一个字符的时间，即产生一个间断，进入发送间断状态。=0 时，退出间断状态。
DLAB	=1，访问除数寄存器；DLAB=0，访问其他寄存器。

D5、D4 和 D3 组合

D5(SP)	D4(EPS)	D3(PEN)	说明
x	x	0	无校验位
1	0	1	校验位恒为 1
1	1	1	校验位恒为 0
0	0	1	奇校验规则计算校验位
0	1	1	偶校验规则计算校验位

- D1~D0组合设定异步传输帧格式中传送一个字符需要二进制位数。
- D2设定停止位的位数。
- D5、D4和D3的组合设定校验位,当D3（PEN）=0时，无校验位，D4和D5无效。当D3（PEN）=1时，表示有校验位。此时若D5（Stick Parity，SP）=1，那么采用固定值校验位，D4（EPS）来确定校验位的值：PEN=1、SP=1、EPS=0时，校验位恒为1；PEN=1、SP=1、EPS=1时，校验位恒为0。若SP=0，此时根据数据信息位的值来计算校验位，当PEN=1，SP=0，EPS=0时，按奇校验规则计算校验位，当PEN=1，SP=0，EPS=1时，按偶校验规则计算校验位。

- 第6位SB（Set Break）是设置中止方式选择位，若SB位置1，则发送端连续发送空号（逻辑“0”），当发空号的时间超过一个完整的字符传送时间，接收端就认为发送设备发送了一个中止字符。此时接收设备一方的BI位被置为1，并且可以发送中断请求，由CPU进行处理。

- 8250地址范围为03F8H~03FFH，试编写程序设置发送字符长度为8位，2位停止位，偶校验。
- 线路控制寄存器的地址为3FBH（A2、A1、A0=011B），控制字应为00011111B。
- 参考程序段如下:
- MOV　　DX, 3FBH　　;LCR口地址
- MOV　　AL, 00011111B　　;LCR的内容，数据格式参数
- OUT　　DX, AL

- 除数锁存器（Divisor Latch LSB/MSB，DLL/DLM）
- 8250/16550芯片传输数据的速率是由除数锁存器控制的。计算机异步串行通信接口外接的1.8432MHz基准时钟，通过除数寄存器给定的分频值，可以在8250内部产生不同的波特率，然后通过BAUDOUT#引脚输出到RCLK，控制接收传输速率。
- $f_{工作时钟}$ = $f_{基准时钟}$ ÷ 除数锁存器 = 波特率 × 16
- 这里$f_{基准时钟}$= 1.8432MHz = 1843200Hz。
- 除数锁存器= $f_{基准时钟}$ ÷（波特率 × 16）
- = 1843200 ÷（波特率 × 16）= 115200 ÷ 波特率

- 8250/16550的除数锁存器有2个字节，低字节（Least Significant Byte，LSB）为DLL，高字节（Most Significant Byte，MSB）为DLM。
- 写入DLM和DLL时，必须设置LCR中的DLAB为1。

编写程序，设置波特率为2400bps

- 波特率为2400，则除数锁存器=115200 ÷2400=48=0030H。将00H写入DLM，30H写入DLL。
- MOV　　DX，3FBH　;置LCR口地址
- MOV　　AL，80H　　;DLAB=1
- OUT　　DX，AL　　;之后，3F8H、3F9H对应于DLL、DLM
- MOV　　DX，3F8H　;DLL的I/O地址
- MOV　　AL，30H　　;商的低字节
- OUT　　DX，AL　　;写入DLL

- MOV DX，3F9H ;DLM的I/O地址
- MOV AL，00H ;商的高字节
- OUT DX，AL ;写入DLM
- MOV DX, 3FBH ;LCR的I/O地址
- MOV AL, 00011111B ;LCR的内容，数据格式参数，DLAB=0
- OUT DX, AL ;之后，3F8H对应于THR/RBR，3F9H对应于IER

中断允许寄存器 IER 的格式

7	6	5	4	3	2	1	0
0	0	0	0	EDSSI	ELSI	ETBEI	ERBFI

ERBFI	接收缓冲器满以后，是否产生中断。=0，禁止；=1，允许。
ETBEI	发送保持寄存器空以后，是否产生中断。=0，禁止；=1，允许。
ELSI	接收数据出错后，是否产生中断。=0，禁止；=1，允许。
EDSSI	MODEM 的控制信号状态改变，是否产生中断。=0，禁止；=1，允许。

编写程序，允许8250/16550中所有的中断

- 中断允许控制字为：00001111B=0FH
- 中断允许寄存器地址为：3F9H（A2A1A0=001），注意此时要DLAB=0。
- MOV　　AL, 0FH　　;中断允许控制寄存器控制字
- MOV　　DX, 3F9H　　;中断允许控制寄存器端口地址
- OUT　　DX, AL　　;写入中断允许控制寄存器

中断识别寄存器 IIR 的格式

7	6	5	4	3	2	1	0
0	0	0	0	0	S2	S1	NINT

NINT	=0，产生了中断。=1，没有产生中断。
S2、S1	=00b，MODEM 的控制信号状态改变。读取 MSR 后，该中断被清除。 =01b，发送保持寄存器空。读取 IIR 或者写入 THR 后，该中断被清除。 =10b，接收缓冲器满。读取 RBR 后，该中断被清除。 =11b，接收数据出错(OE、PE、FE 或 BI 为 1)。读取 LSR 后，中断该被清除。

- 当两种中断同时产生时，S2、S1被设为最高优先级的中断源（11B最高，00B最低）。例如，当某一时刻接收数据出错和接收缓冲器满中断源都发出中断请求时，IIR的内容为06H。处理完高优先级的中断后，NINT仍然为0，而S2、S1被设为低优先级的中断源，直到所有的中断都被处理。所以编写程序时应注意，若同一时间内允许有一个以上中断请求（IER中至少有2位为1），则在处理完高优先级的中断之后，还要再检查中断识别寄存器IIR的NINT是否为0，即是否尚有未被处理的中断源，如果不检查可能会造成某些中断不被响应。

- 如果以中断方式发送或接收数据，在初始化时必须设置中断控制器，设置对应的中断请求允许，同时还要设置发送或接收中断服务程序的入口地址（中断向量）。查询方式下可以往中断允许寄存器里面写入00H，禁止所有中断。

7	6	5	4	3	2	1	0
RT1	RT2	0	0	DMA	XMIT	REVC	EN

EN	FIFO 允许，=1 时，表示允许，=0 时表示禁止。
REVC	接收器 FIFO 复位，=1 时，表示复位，=0 时无效。
XMIT	发送器 FIFO 复位，=1 时，表示复位，=0 时无效。
DMA	DMA 控制。=1 时，FIFO 方式，=0 时表示 16450 方式
RT1，RT2	接收器触发器值。00：FIFO 中有 1 字节；01：FIFO 中有 4 字节；10：FIFO 中有 8 字节；11：FIFO 中有 14B

FIFO 控制寄存器格式

- 8255/16550初始化编程
- 在使用8255/16550进行串行通信以前，必须要对其进行初始化编程。串口初始化在系统复位后，在芯片工作以前使用。初始化工作主要包括设置串口芯片的通信格式、波特率、是否使用中断、是否进行自检测试等操作。

- 写除数锁存器，设置数据传输率。注意写除数锁存器时要先使通信线控制寄存器的最高位（DLAB）置“1”。
- 写入通信线路控制寄存器，确定异步串行通信的数据帧格式，注意要将DLAB位清“0”，以便接下来能对中断允许寄存器初始化，以及在串行数据传送中能对接收缓冲器和发送保持寄存器进行操作。
- 写入MODEM控制寄存器（可省略）。
- 写入中断允许寄存器，设置中断允许或屏蔽位。
- 16550的初始化编程和8250类似，如果打开了FIFO的话，还需要增加下一步。
- 设置FIFO控制寄存器。

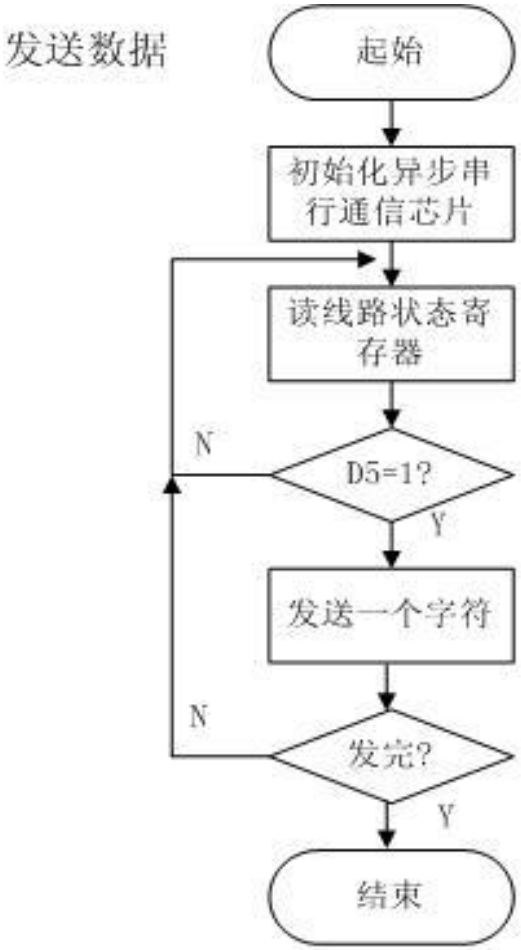
- 假定16550的端口地址为3F8 ~ 3FFH。16550以波特率为9600bps进行串行通信，字符格式为7个数据位、2个停止位、奇校验方式，允许所有中断，试编写初始化程序。
- 波特率为9600bps，则除数锁存器=115200 ÷9600=12=000CH。将00H写入DLM， 0CH写入DLL。
- 根据要求的数据帧格式，LCR=00001110B=0EH
- MCR=00001011B=0BH，表示使用中断，并且使DTR#和RTS#两个信号为有效电平。
- 中断允许字为：00001111B=0FH，开放所有中断。
- FCR控制字为：10010111B=87H，表示FIFO缓冲中有8个字节触发，发送和接收FIFO复位。

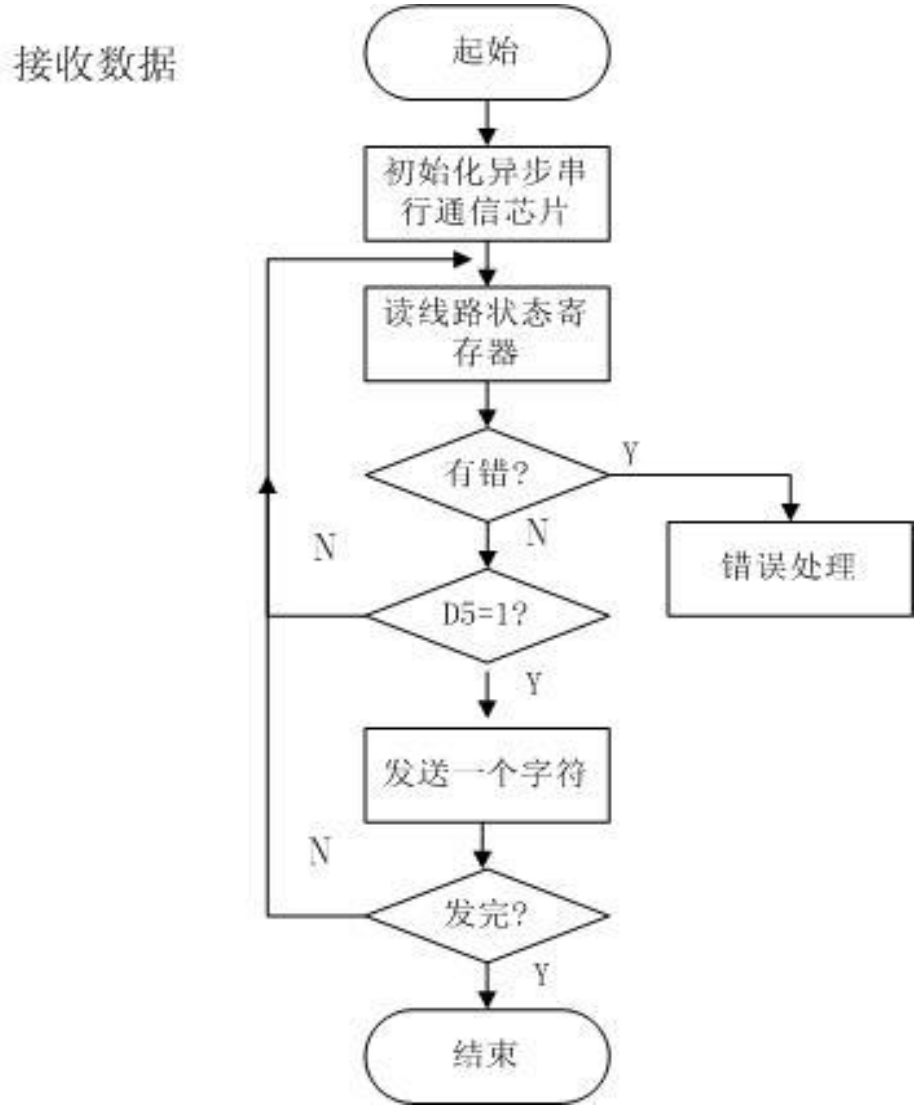
```
;置DLAB=1
MOV     DX, 03FBH       ;DX指向16550的通信线控制寄存器地址
MOV     AL, 80H
OUT     DX, AL
;置除数锁存器
MOV     DX, 03F8H       ;除数寄存器（低字节）地址
MOV     AL, 0CH         ;对应波特率为9600的除数为000CH
OUT     DX, AL          ;送除数低字节
INC     DX              ;除数寄存器（高字节）地址
MOV     AL, 0
OUT     DX, AL          ;送除数高字节
```

```
;置通信线路控制寄存器
MOV     AL, 0EH          ;通信控制寄存器控制字：0→DLAB
MOV     DX, 03FBH
OUT     DX, AL
;置Modem控制寄存器
MOV     DX, 03FCH        ;指向Modem控制寄存器
MOV     AL, 0BH          ;Modem控制字
OUT     DX, AL
MOV     DX, 03F9H        ;指向中断允许寄存器地址
MOV     AL, 0FH          ;中断允许控制字：允许所有的中断
OUT     DX, AL
```

```
;置FIFO控制寄存器
MOV     DX, 03FAH       ;DX指向FIFO控制寄存器
MOV     AL, 87H         ;FIFO控制字
OUT     DX, AL
```

8255/16550应用编程-查询方式





- 编写程序段，实现串行异步全双工通信，采用查询方式，通信过程中检测到数据传输错误就显示一个问号“?”，没有错误则接收数据并显示。同时可以从键盘输入发送字符发送数据（用户没有输入字符就不发送）。按下ESC键返回系统。地址范围为3F8H~3FFH。

```
;查询通信线路状态
statue:     MOV  DX, 3FDH
;读通信线路状态寄存器
            IN     AL, DX
            TEST   AL, 1EH          ;检测是否有错误
            JNZ    error            ;有错，则转错误处理
            TEST   AL, 01H          ;检测是否接收到数据
            JNZ    receive          ;是，转接收处理
            TEST   AL, 20H          ;检测是否可以发送数据
            JZ     statue           ;不，循环查询
```

```
;检测键盘输入
        MOV  AH, 0BH        ;检测键盘有无输入字符
        INT  21H
        CMP  AL, 0
        JZ   statue         ;无输入字符，循环等待
        MOV  AH, 0          ;有输入字符，读取字符
        INT  16H            ;键盘服务
;采用01号DOS功能调用，则有回显
        CMP  AL, 1BH
        JZ   done           ;是ESC键，程序返回DOS
```

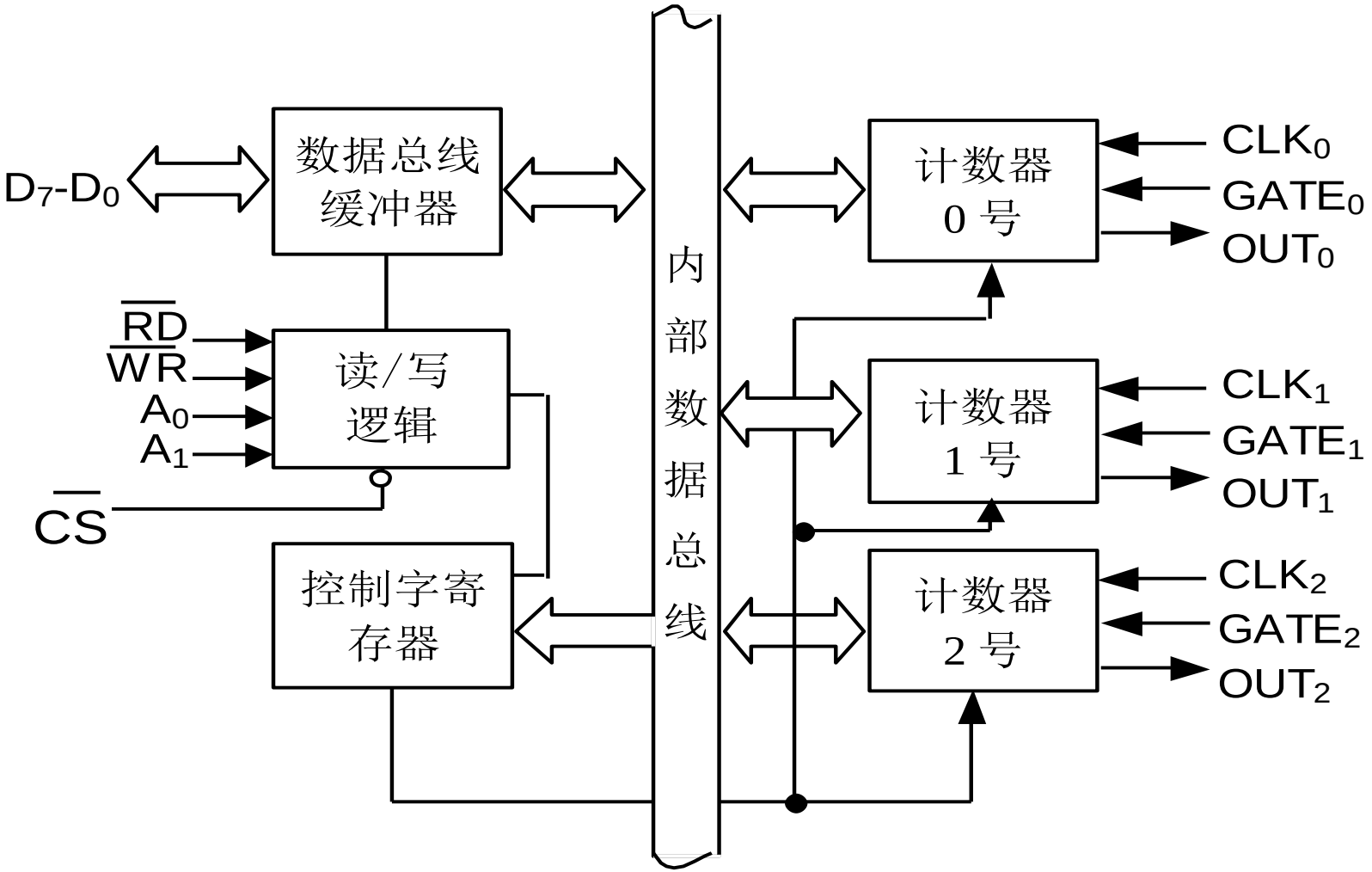
```
;发送数据
        MOV  DX, 3F8H      ;将字符输出给发送保持寄存器
        OUT  DX, AL        ;串行发送数据
        JMP  statue        ;继续查询
;接收数据
receive:
        MOV  DX, 3F8H      ;从输入缓冲寄存器读取字符
        IN   AL, DX
        AND  AL, 7FH       ;传送标准ASCII码（7个数据位）， 故取低7位
        PUSH AX            ;保存数据
```

```
;显示数据
        MOV   DL, AL        ;屏幕显示该数据
        MOV   AH, 2
        INT   21H
        POP   AX            ;恢复数据
        CMP   AL, 0DH       ;判断数据是不是回车符
        JNZ   statue        ;不是，则循环
        MOV   DL, 0AH       ;是，再输出换行
        MOV   AH, 2
        INT   21H
        JMP   statue        ;继续查询
```

```
;错误处理
error:      MOV  DX, 3F8H     ;读出接收有误的数据，丢掉
            IN   AL, DX
            MOV  DL, '?'      ;显示问号
            MOV  AH, 2
            INT  21H
            JMP  statue       ;继续查询
```

定时与计数技术

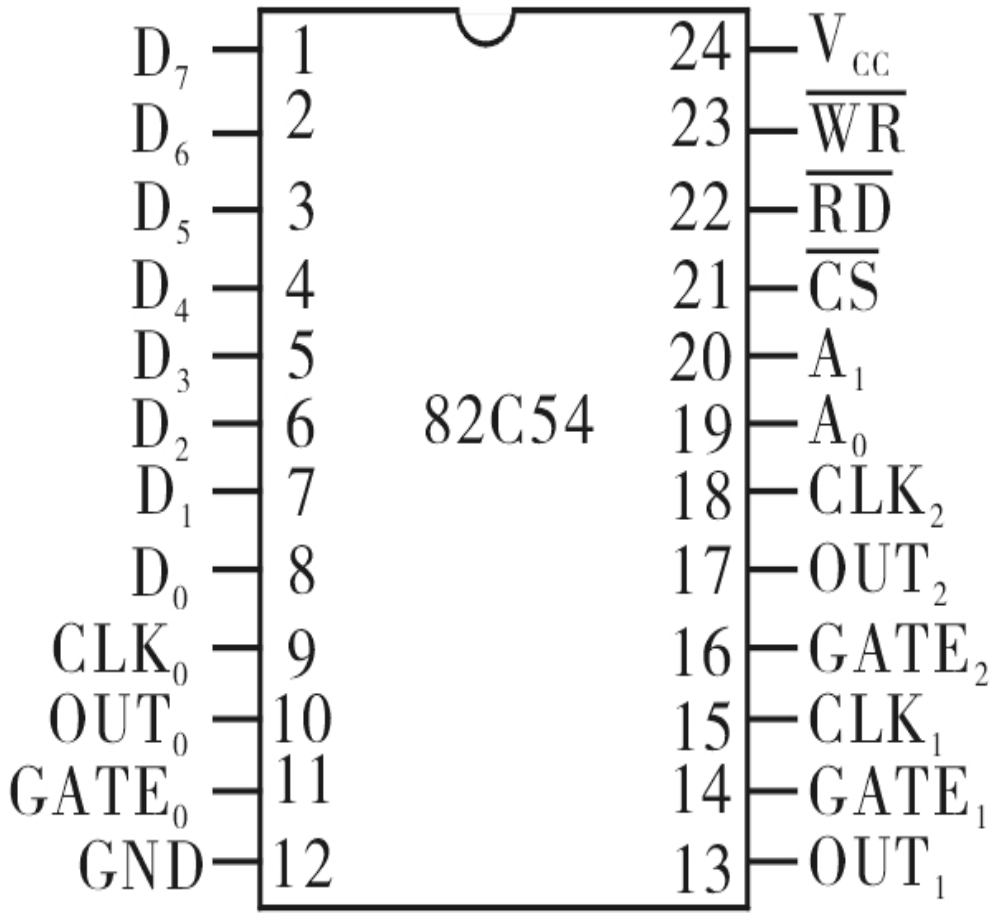
- 可编程定时器芯片
- 8254/8253功能基本类似，常用在实现定时和计数功能的外围电路中，拥有3个独立的16位计数器，每个计数器都可通过程序设计的方法设定为6种计数方式。二者最大的区别在于8253不支持写读回控制字。


D7-D0
数据总线
缓冲器
RD
WR
A0
A1
CS
读/写
逻辑
控制字寄
存器
内
部
数
据
总
线
计数器
0 号
计数器
1 号
计数器
2 号
CLK0
GATE0
OUT0
CLK1
GATE1
OUT1
CLK2
GATE2
OUT2

- （1）数据总线缓冲器
- 该缓冲器为8位双向三态的缓冲器，可直接挂在数据总线上。通过它，一方面可以向控制寄存器写入控制字，向计数器写入计数初值；另一方面也可由CPU通过该缓冲器读取计数器的当前计数值。
- （2）读写逻辑
- 读写逻辑的功能是接收来自CPU的控制信号，包括读信号RD#、写信号WR#、片选信号CS#和芯片内部寄存器的寻址信号A1、A0，并完成对8254各计数器的读写操作。

- （3）控制字寄存器
- 接收来自CPU的控制字，并由控制字D7、D6位决定将该控制字写入哪一个计数器的控制寄存器中。
- （4）计数器
- 8254有3个独立的计数器通道，每个通道的结构完全相同，每一个通道有一个16位减法计数器；还有对应的16位初值寄存器和输出锁存器。计数开始前写入的计数初值保存在初值寄存器中，如果计数初值为16位则需要分两次写入；计数过程中，减法计数器的值不断递减，而初值寄存器中的初值不变。输出锁存器则用于写入锁存命令时锁定当前计数值。

8254的引脚


D7 1
D6 2
D5 3
D4 4
D3 5
D2 6
D1 7
D0 8
CLK0 9
OUT0 10
GATE0 11
GND 12
82C54
24 VCC
23 WR
22 RD
21 CS
20 A1
19 A0
18 CLK2
17 OUT2
16 GATE2
15 CLK1
14 GATE1
13 OUT1

- （1）与CPU连接的引脚
- D0～D7，三态双向数据线。与CPU数据总线相连，用于传递CPU与8254之间的数据信息、控制信息和状态信息。
- CS#，片选信号，输入，低电平有效。有效时，表示8254被选中，允许CPU 对其进行读写操作。通常连接到I/O端口地址译码电路的输出端。
- WR#，写信号，输入，低电平有效。用于控制CPU对8254的写操作，可与A1、A0信号配合以决定是写入控制字还是计数初值。

8254 的读写操作逻辑

CS#	RD#	WR#	A1	A0	操作功能
0	1	0	0	0	计数初值装入计数器 0
0	1	0	0	1	计数初值装入计数器 1
0	1	0	1	0	计数初值装入计数器 2
0	1	0	1	1	写控制寄存器
0	0	1	0	0	读计数器 0
0	0	1	0	1	读计数器 1
0	0	1	1	0	读计数器 2

- （2）与外部设备的接口信号
- CLK0、CLK1、CLK2，时钟脉冲输入端。用于输入定时脉冲或计数脉冲信号。CLK可以是系统时钟脉冲，也可以由其他脉冲源提供。
- GATE0、GATE1、GATE2，门控输入端。用于外部控制计数器的启动计数和停止计数的操作。两个或两个以上计数器连用时，可用此信号来同步，也可用于与外部某信号的同步。
- OUT0、OUT1、OUT2，计数输出端。在不同方式的计数过程中，OUT引脚上输出相应的信号。

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

D7、D6	=00b，设定计数器 0 的工作参数。 =01b，设定计数器 1 的工作参数。 =10b，设定计数器 2 的工作参数。 =11b，锁存计数器的当前计数值。
D5、D4	=00b，锁存计数器当前值，供 CPU 使用 =01b，只读/写低 8 位计数值，高 8 位自动置零。 =10b，只读/写高 8 位计数值，低 8 位自动置零。 =11b，使用 16 位计数值。先读/写低 8 位，后读/写高 8 位。。
D3、D2、D1	=000b~101b，设定该计数器的工作方式为方式 0~5。
D0	=0，二进制计数模式；=1，BCD 计数模式。

8254 方式控制字的格式

- 8254的每个计数通道都有二进制和十进制（BCD）两种计数模式。由于计数器是先减1，再判断是否为0，所以写入0实际代表最大计数值。采用16位二进制计数时，写入初值的范围为0000H～FFFFH，其中0000H是最大值，表示计数器初值为65536。BCD计数制时，写入初值范围为0000～9999，其中0000表示最大值l0000。

8254的编程

- 首先把相应的方式控制字写入到控制字寄存器中，再根据控制字中数据读写格式的规定，将计数初值写入对应的计数通道。方式控制字地址为A1、A0=11B；计数器0的初值应写入A1、A0=00B的地址；计数器1的初值应写入A1、A0=01B的地址；计数器2的初值应写入A1、A0=10B的地址。使用16位计数值时，高8位和低8位写入的是同一个地址，应先写入低8位，后写入高8位。

- 假设8254地址为40H~43H，试编写程序，将计数器0初始化为工作方式3，采用二进制计数模式，计数初值为2000。
- 由地址范围可知：控制口的地址为43H，计数器0、1、2分别使用地址40H、41H、42H。
- 计数初值2000D = 07D0H，需要采用16位计数，故计数器0的控制字=00110110B=36H。

```
;二进制方式
MOV     AL, 36H
OUT     43H, AL          ;写入方式控制字
MOV     AL, 0D0H         ;2000D = 07D0H，取低8位
OUT     40H, AL          ;写入计数初值的低8位
MOV     AL, 07H          ;2000D = 07D0H，取高8位
OUT     40H, AL          ;写入计数初值的高8位
```

```
;如果采用BCD方式，初始化程序为：
MOV     AL, 00110111B     ;D0=1: 使用BCD计数
OUT     43H, AL                   ;写入方式控制字
MOV     AL, 00H                   ;2000的BCD码为2000H。
OUT     40H, AL                   ;写入计数初值的低8位
MOV     AL, 20H                   ;2000D = 07D0H，取高8位
OUT     40H, AL                   ;写入计数初值的高8位
```

读取当前计数值

- 8254的计数器16位的，但数据总线是8位的，即每次只能读取8位，所以16位数据CPU要分2次读，先读取低8位，再读取高8位。此时，如果不锁存计数器的当前计数值，那么在两次读取操作之间，计数值的高8位可能已经发生变化了。
- 因此在读取计数器的当前值之前，要先把当前值锁存到锁存寄存器，然后由CPU读取锁存寄存器的值。

- 锁存当前计数值有下面3种方法：
- ①利用GATE信号使计数过程暂停。
- 这种方法需要硬件电路配合，读取前将GATE信号置为低电平，不再计数，读取后将GATE信号恢复为高电平。因为这种方法会中断计数过程，因此一般不采用这种方法。
- ②锁存一个计数器。
- 每一个计数器都有一个16位输出锁存器，它的值随着计数器的值不断变化。向8254写入一个方式控制字，令其D5、D4=00B，则8254锁存由D7、D6指定的计数器的当前值。当锁存控制字写入时，完成的操作是把计数器的当前值锁存到输出锁存器中，进入锁存状态，计数器继续计数时不再影响输出锁存器的内容。CPU读取输出锁存器后，自动解除锁存状态，它的值又随减法计数器变化。

- 假定8254端口地址为40H~43H，编写程序锁存并读取计数器0的当前计数值。
- 控制字地址为43H，计数器0的地址为40H。
- 锁存计数器0的当前计数值的方式控制字为00000110B=06H
- MOV　　AL, 06H
- OUT　　43H, AL　　;写入方式控制字，锁存计数值
- IN　　AL, 40H　　;读入输出锁存器的低8位
- MOV　　AH, AL　　;暂存在AH中
- IN　　AL, 40H　　;读入输出锁存器的高8位
- XCHG　　AH, AL　　;AX=输出锁存器的16位值

7	6	5	4	3	2	1	0
1	1	COUNT#	STATUS#	CNT2	CNT1	CNT0	0

COUNT#	=0，锁存当前计数值。=1，不锁存当前计数值。
STATUS#	=0，锁存当前状态。=1，不锁存当前状态。
CNT2	=1，对计数器 2 进行锁存操作。=0，不对计数器 2 进行锁存操作。
CNT1	=1，对计数器 1 进行锁存操作。=0，不对计数器 1 进行锁存操作。
CNT0	=1，对计数器 0 进行锁存操作。=0，不对计数器 0 进行锁存操作。

8254 读回控制字的格式

- 当COUNT#和STATUS#都设置为0时，计数器的计数值和状态同时被锁存，读取该计数器的地址时，首先读取到的是这个计数器的状态，然后再读取它的计数当前值。
- 假定8254端口地址为40H~43H，利用读回控制字读取计数器0的当前计数值。
- 控制字地址为43H，计数器0的地址为40H。
- 锁存计数器0的当前计数值的方式控制字为11010010B=D2H。

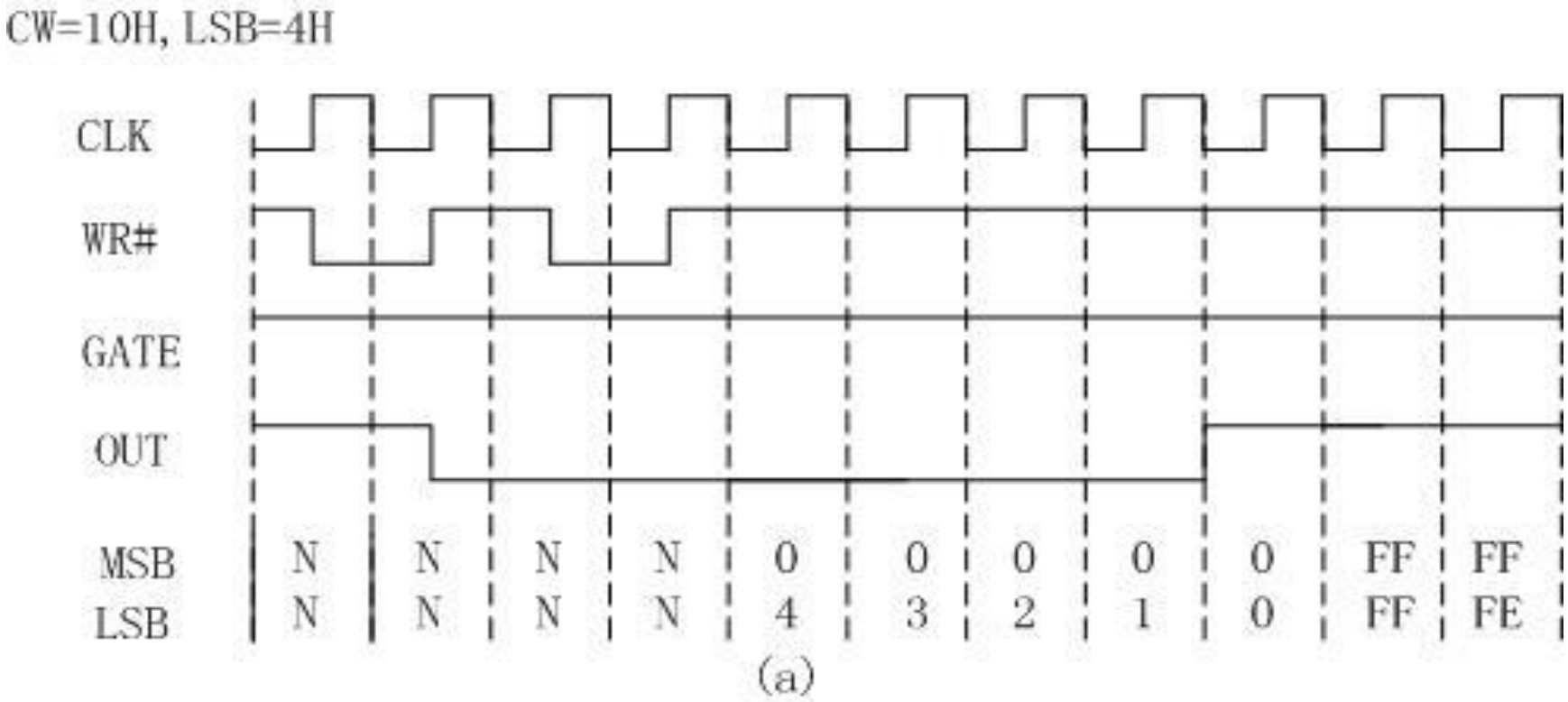
```
MOV     AL, D2H
OUT     43H, AL         ;写入方式控制字，锁存计数值
IN      AL, 40H         ;读入输出锁存器的低8位
MOV     AH, AL          ;暂存在AH中
IN      AL, 40H         ;读入输出锁存器的高8位
XCHG    AH, AL          ;AX=输出锁存器的16位值
```

7	6	5	4	3	2	1	0
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

OUTPUT	计数器 OUT 输出管脚的状态。=0，低电平；=1，高电平。
NULL COUNT	=0，输出锁存器的内容有效。=1，输出锁存器的内容无效。
RW1RW0	计数器的方式控制字的 D5 D4。即读写格式。
M2M1M0	计数器的方式控制字的 D3 D2 D1。即工作方式 0~5。
BCD	计数器的方式控制字的 D0。=0，二进制计数模式；=1，BCD 模式。

8254的六种工作方式

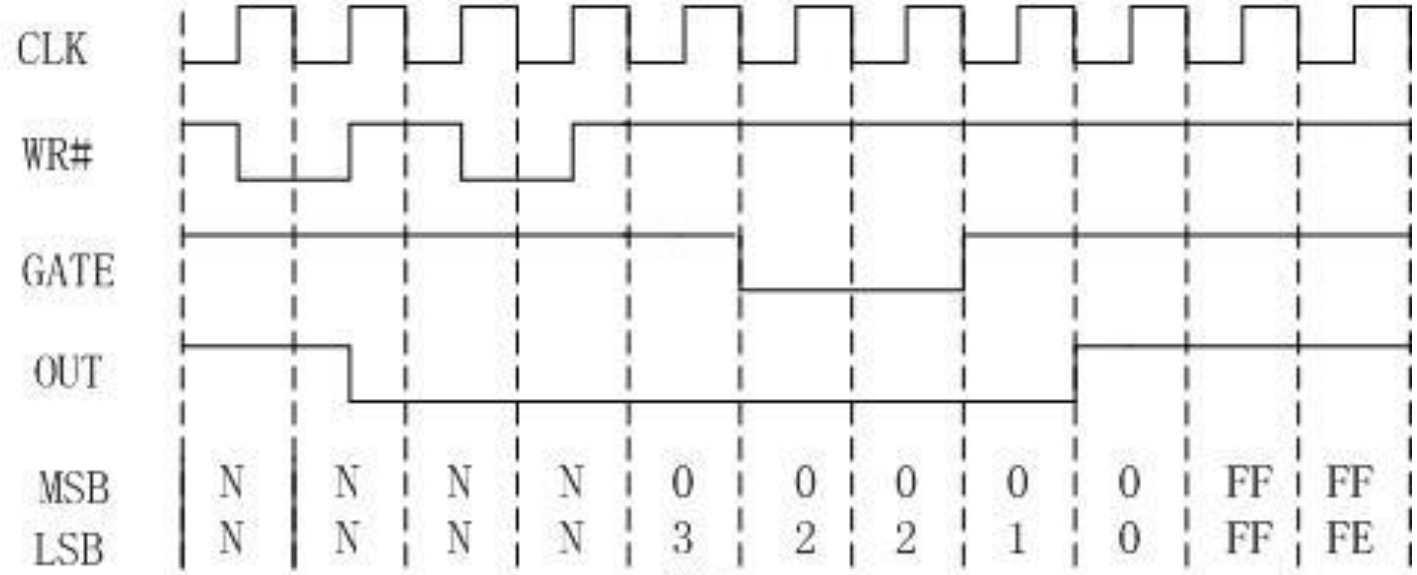
- （1）方式0（计数结束中断方式）



- 当写入方式0控制字后，OUT立即变为低电平，并且在计数过程中一直维持低电平。写入初值n后，CLK第1个下降沿使计数初值装入计数器。随后每一个CLK脉冲的下降沿都使计数器减1。计数器减到零时，即OUT引脚在n+1个CLK后变为高电平，并且一直保持到该通道重新装入计数值或重新设置工作方式为止。在方式0下，写一次计数初值，只计数一遍，计数器不会自动重装初值和重新开始计数。

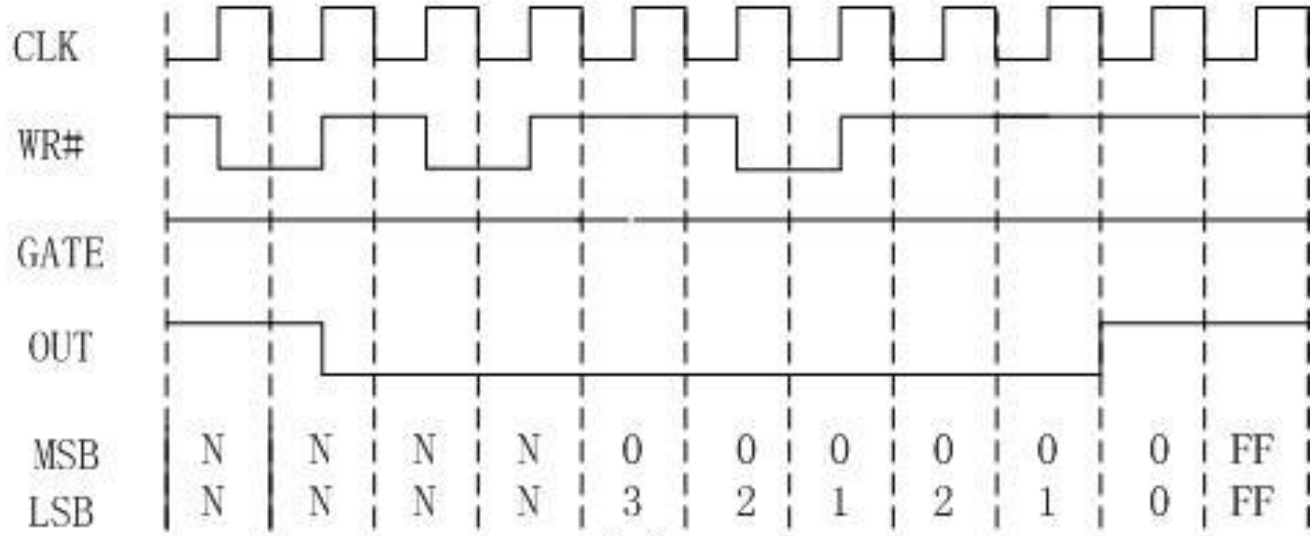
- 门控信号GATE用来控制计数过程。当GATE保持为低电平时暂停计数，当GATE变为高电平时又恢复计数。
- 如果在计数过程中写入新的计数初值，则在写入新值后的下一个时钟下降沿计数器将按新的初值计数，即新的初值是立即有效的，不必等待第一个计数过程的结束。

CW=10H, LSB=3H



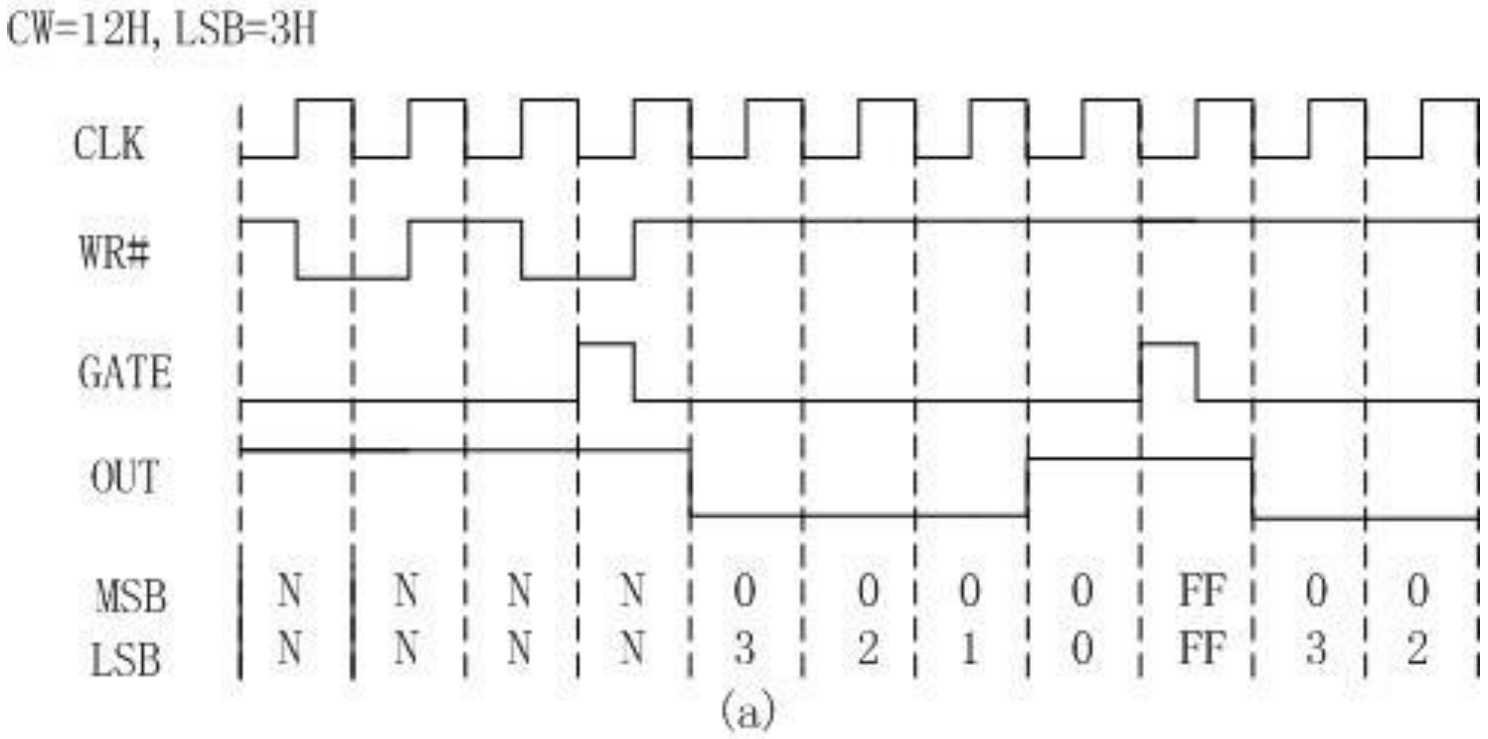
(b)

CW=10H, LSB=3H, LSB=2H



(c)

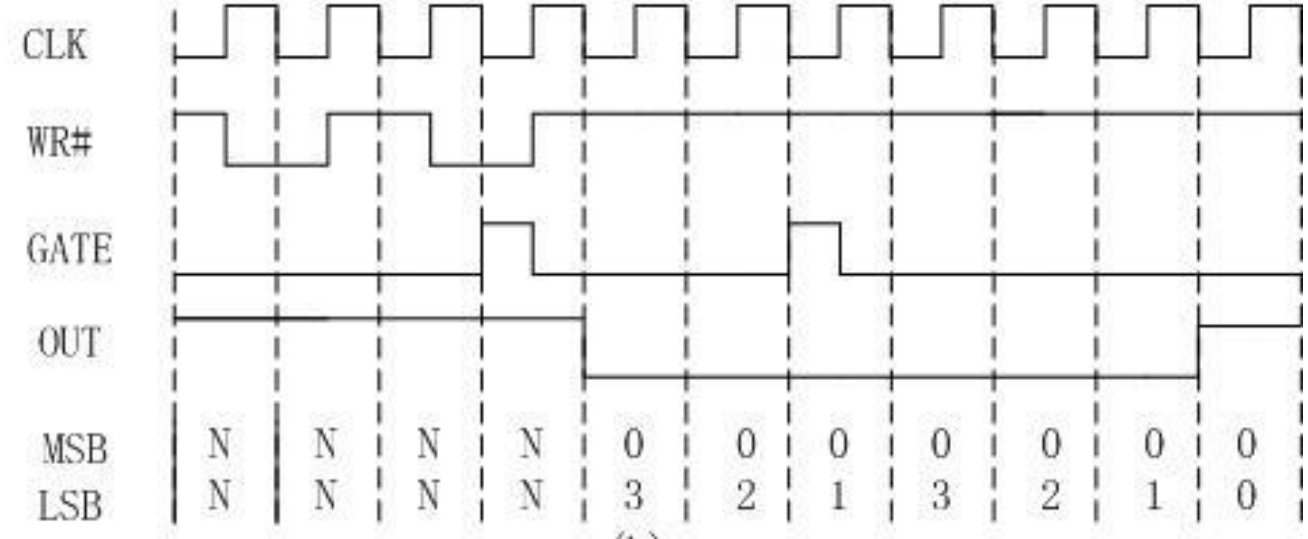
- （2）方式1 （可编程单稳态触发器）
- 这种方式由外部门控信号GATE上升沿触发，产生一单拍负脉冲信号，脉冲宽度由计数初值决定。



- 当写入方式1控制字后，OUT输出变为高电平。写入计数初值n之后，计数器并不立即开始计数，而要等到GATE上升沿后的下一个CLK输入脉冲的下降沿，OUT输出变低电平，计数才开始。计数到0时结束，OUT输出变高，从而产生一个宽度为n个CLK周期的负脉冲。

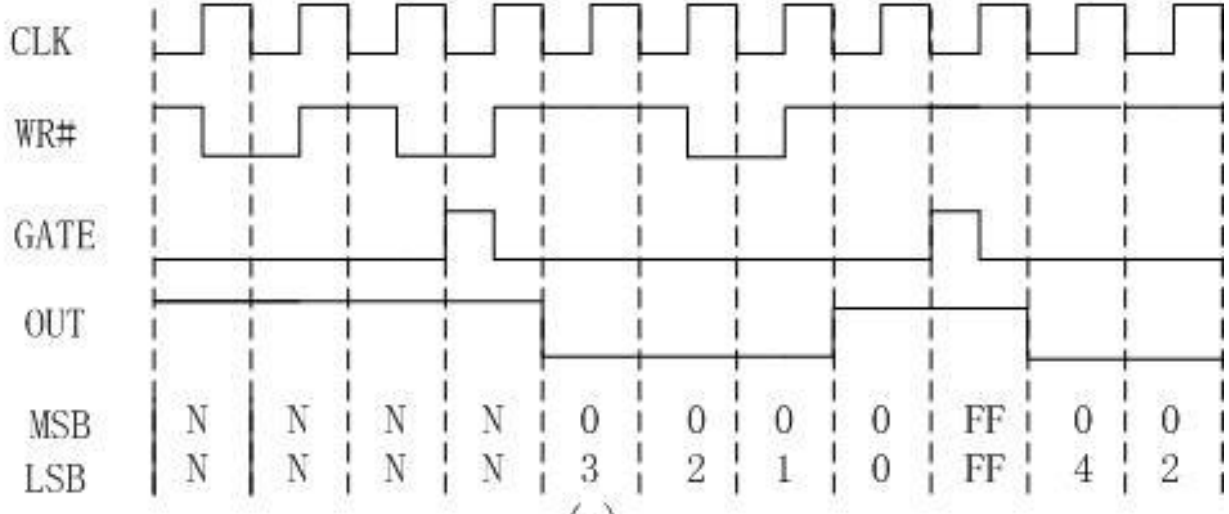
- 在方式1中，GATE信号的作用包括两个方面。第一，在计数结束后，若再来一个GATE信号上升沿，则在下一个时钟周期的下降沿又从初值开始计数，而不需要像方式0那样重新写入初值，即门控信号可重新触发计数；第二，在计数过程中，若再来一个门控信号的上升沿，也在下一个时钟下降沿从重新装入初值开始计数，即终止原来的计数过程，开始新的一轮计数。
- 在计数过程中写入新的初值，不会立即影响计数过程，只有下一个门控信号到来后才按新值开始计数。即在计数过程写入的初值要到下一次计数才有效。

CW=12H, LSB=3H



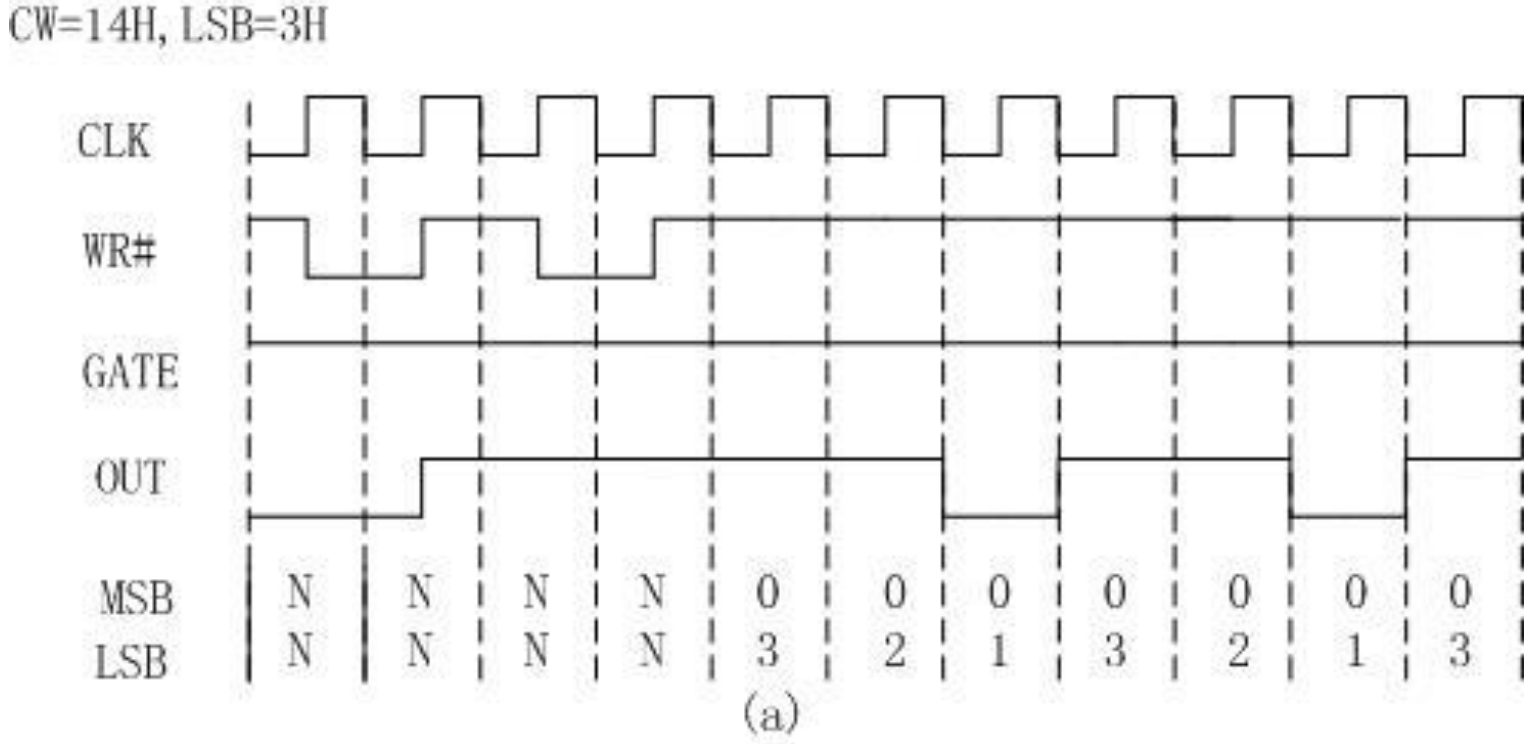
(b)

CW=12H, LSB=3H, LSB=4H



(c)

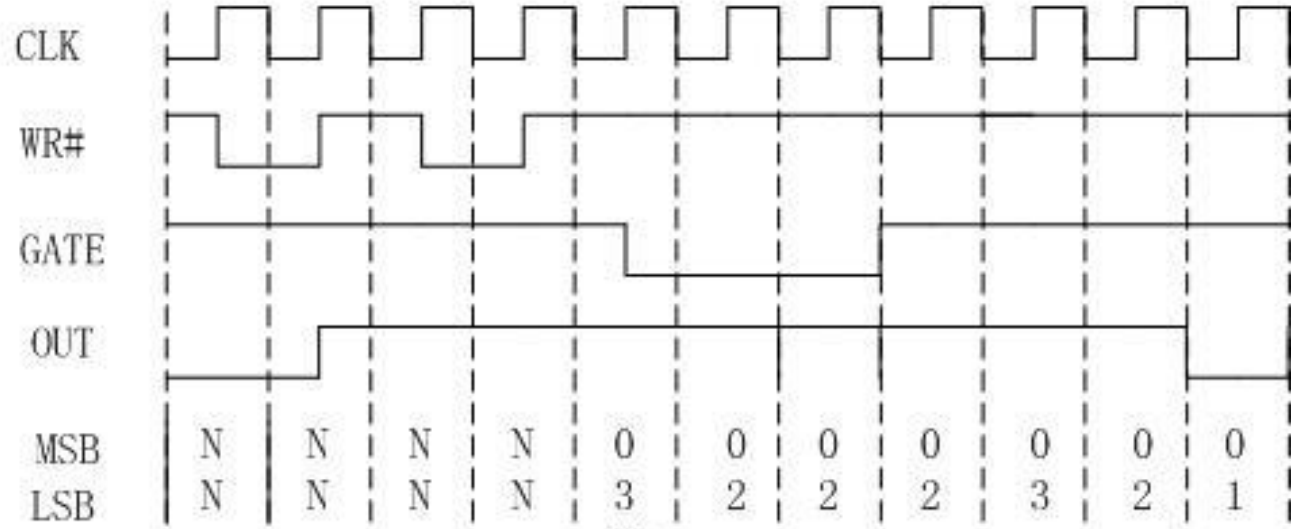
- （3）方式2（脉冲波发生器、分频器）
- 这种方式的功能如同一个N分频计数器，输出是输入时钟按照计数值N分频后的一个连续脉冲。



- 写入控制字后的第一个CLK时钟上升沿，输出端OUT 变成高电平。若GATE=1，写入计数初值后的第一个时钟下降沿开始减1计数，减到1时，输出端OUT变为低电平，减到0时，输出OUT又变成高电平，同时从初值开始新的计数过程，即计数到1时输出一个CLK脉冲宽度的负脉冲。方式2能自动重装初值，给定计数初值后计数通道连续工作。

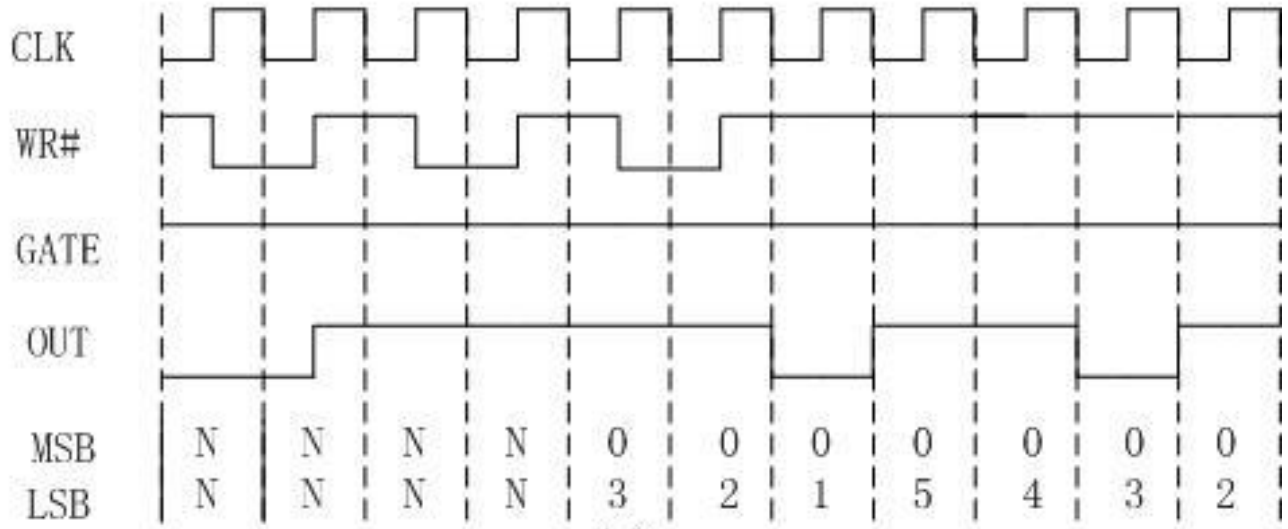
- 方式2中，GATE信号为低电平时终止计数，而由低电平恢复为高电平后的第一个时钟下降沿重新从初值开始计数。
- 在计数过程中写入新的初值，且GATE信号一直维持高电平，则新的初值不会立即影响当前的计数过程，但在计数结束后的下一个计数周期将按新的初值计数，即新的初值下次有效。

CW=14H, LSB=3H



(b)

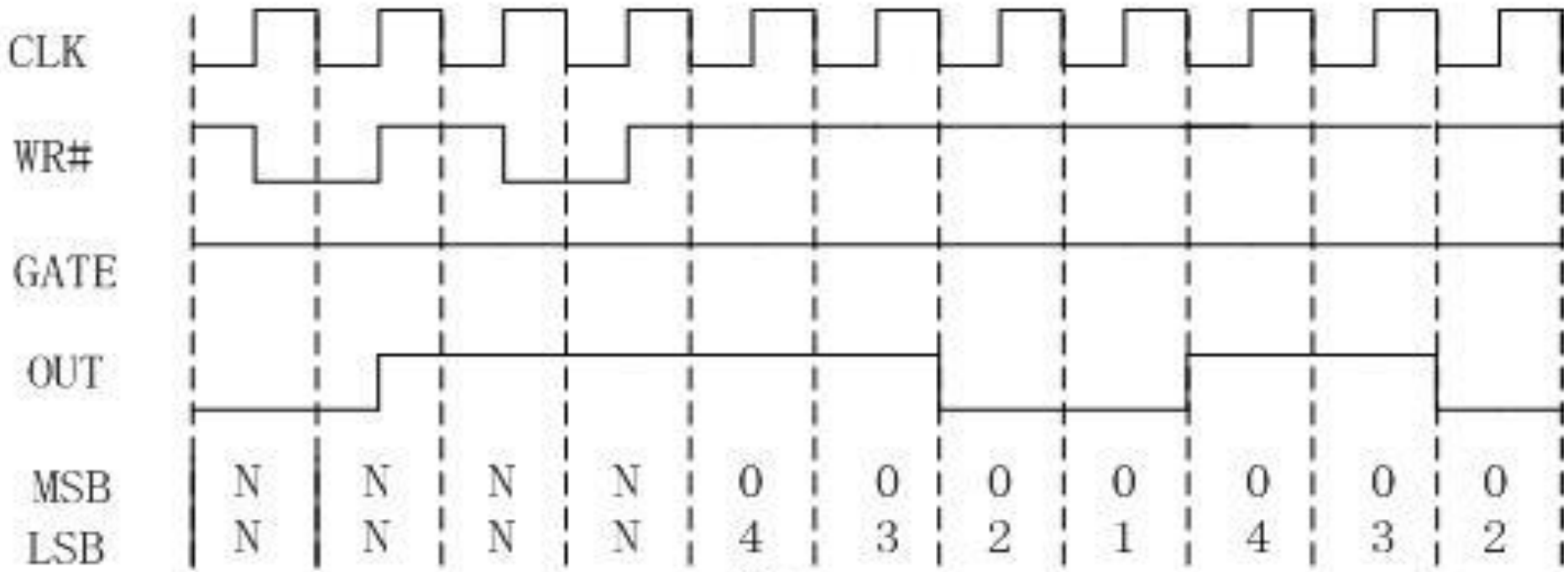
CW=14H, LSB=3H, LSB=5



(c)

- （4）方式3 （方波发生器）
- 方式3与方式2类似，所不同的是它们的OUT输出波形不同：方式2在计数过程结束前输出一个CLK时钟的负脉冲；而方式3输出一个方波。当计数初值n为偶数时，方波的高电平和低电平的维持时间为n/2个CLK时钟；当计数初值n为奇数时，方波的高电平维持时间为（n+1）/2个CLK时钟，低电平维持时间为（n-1）/2个CLK时钟。即输出端OUT的波形是连续的方波（或近似方波）

CW=16H, LSB=4H

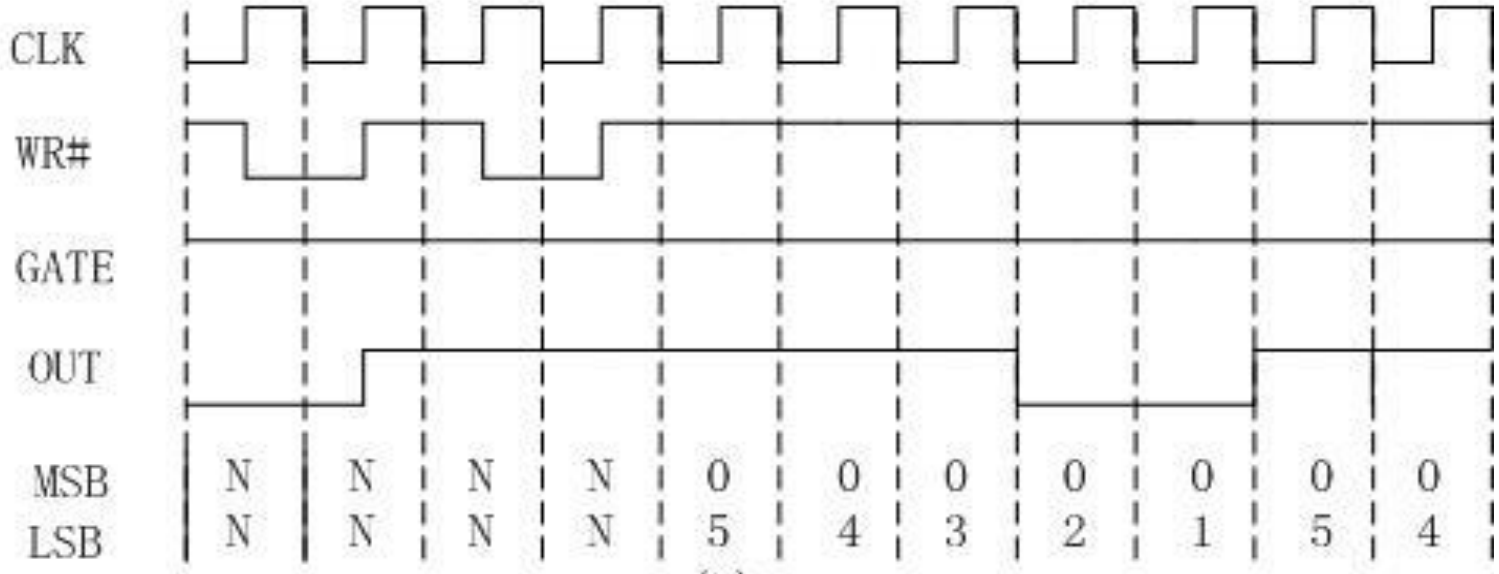


(a)

- 写入控制字后的第一个时钟上升沿，输出端OUT变成高电平。写入计数初值后的第一个时钟下降沿开始减1计数。计数初值为偶数4，减到4/2=2时，输出端OUT变为低电平；减到0时，输出端OUT又变成高电平，并重新从初值开始新的计数过程。

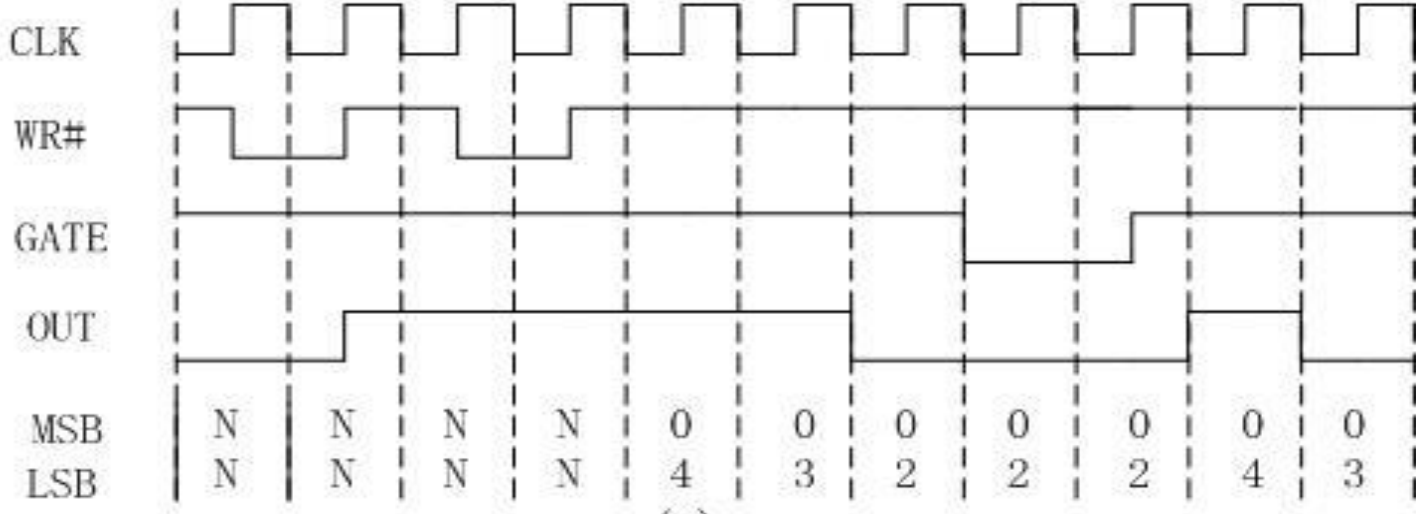
- 给出当计数初值n为奇数5时，输出为3个CLK宽度的高电平和2个CLK宽度的低电平的近似方波组合。
- 计数过程中，如果GATE变为低电平时，计数暂停。当GATE变高以后的下一个时钟脉冲下降沿，计数初值被重新装入，并开始计数。

CW=16H, LSB=5H



(b)

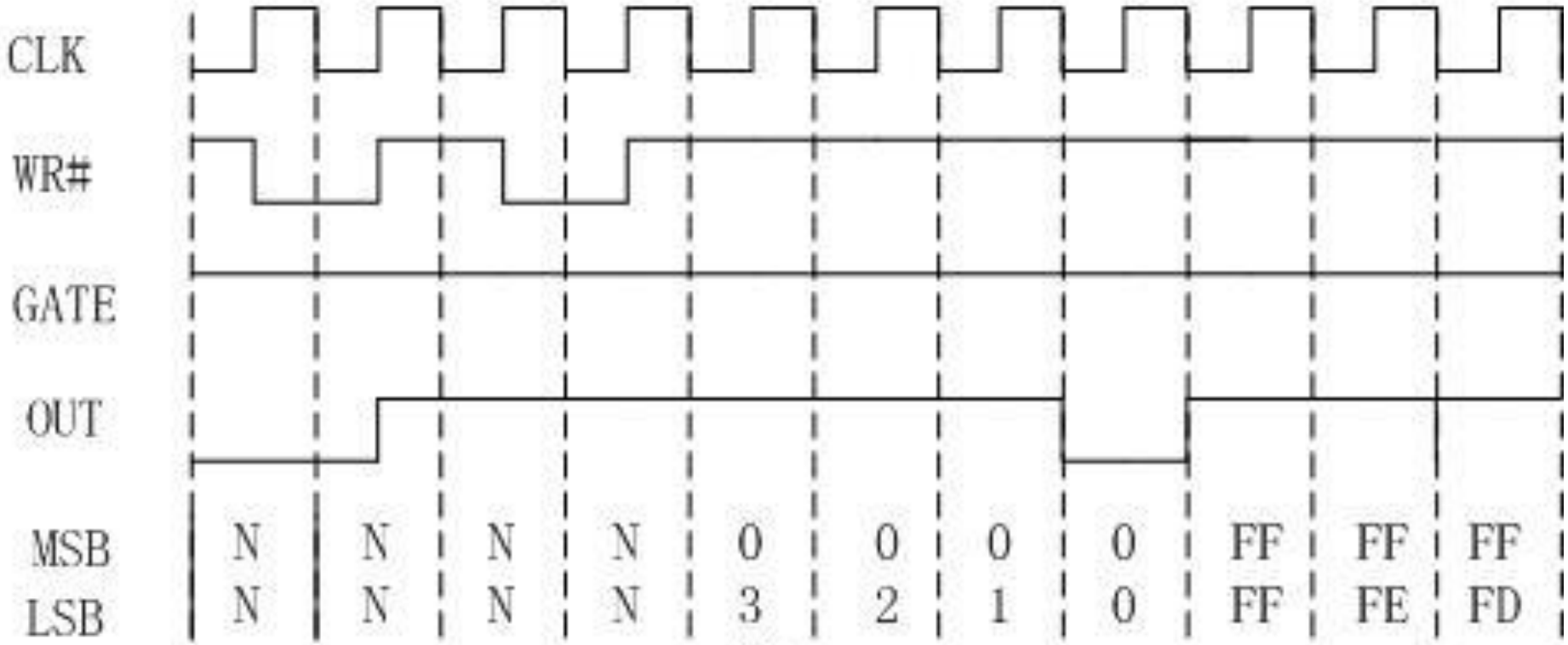
CW=16H, LSB=4H



(c)

- （5）方式4 （软件触发选通方式）
- 写入方式控制字后，OUT输出高电平。若GATE=1，写入初值后的下一个CLK脉冲开始减1计数，计数到达0值（不是减到1）后，OUT输出为低电平，持续一个CLK脉冲周期后再恢复到高电平。输出负脉冲可以用作选通脉冲，它是通过用软件写入计数初值触发而获得的，所以叫软件触发选通方式。

CW=18H, LSB=3H

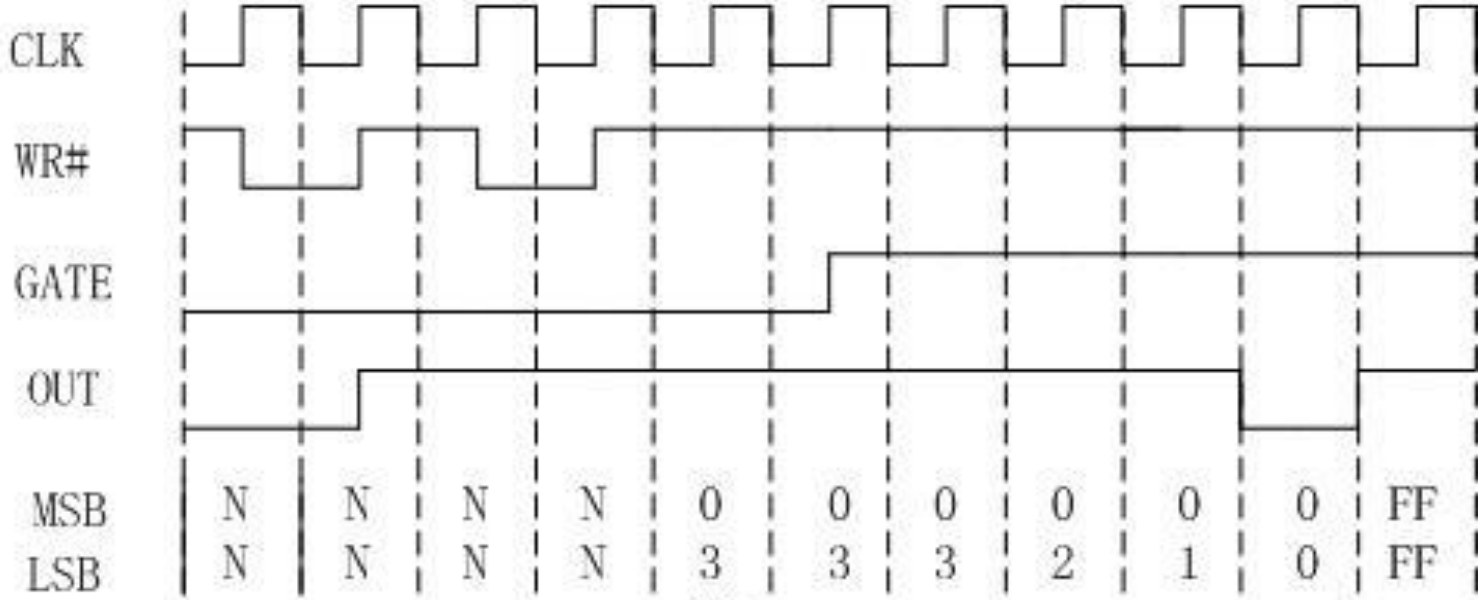


(a)

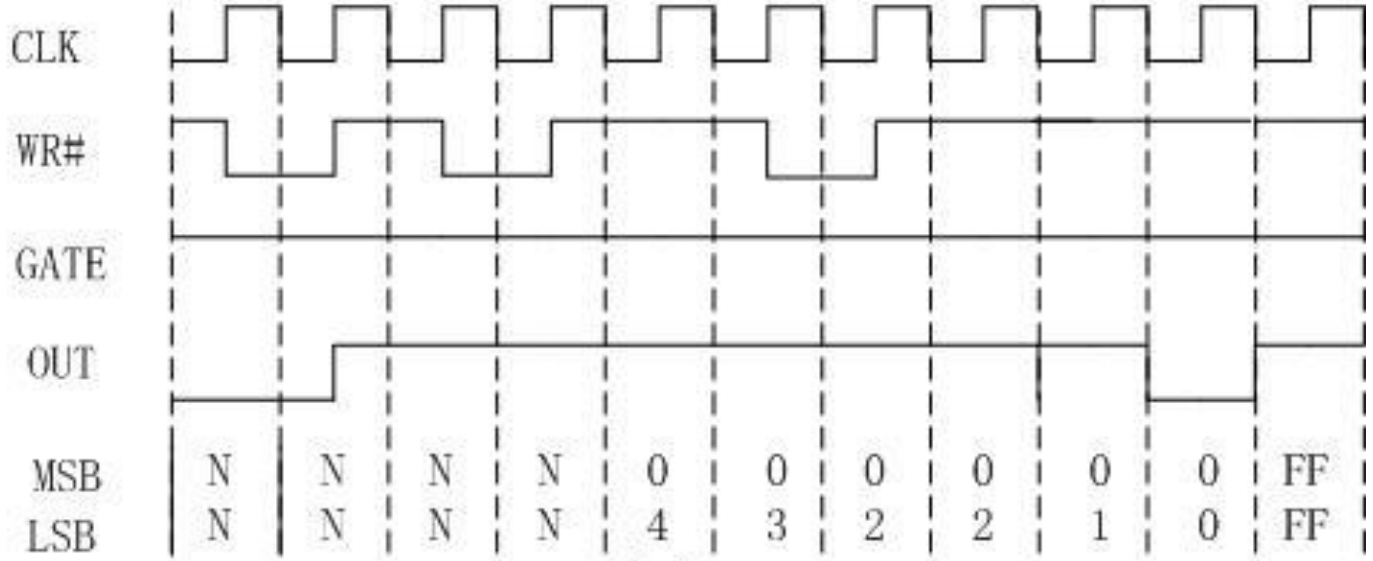
- 写入控制字后的第一个时钟上升沿，输出端OUT变成高电平。写入计数初值后的第一个时钟下降沿开始减1计数。减到0时，输出端OUT输出一个CLK脉冲宽度的负脉冲，计数停止。

- 写入计数初值后，GATE为低电平时，禁止计数，输出维持当时的电平。当GATE变高以后，允许计数。
- 在计数过程中写入新的初值，在写入新值后的下一个时钟下降沿计数器将按新的初值计数，即新的初值立即生效。

CW=18H, LSB=3H



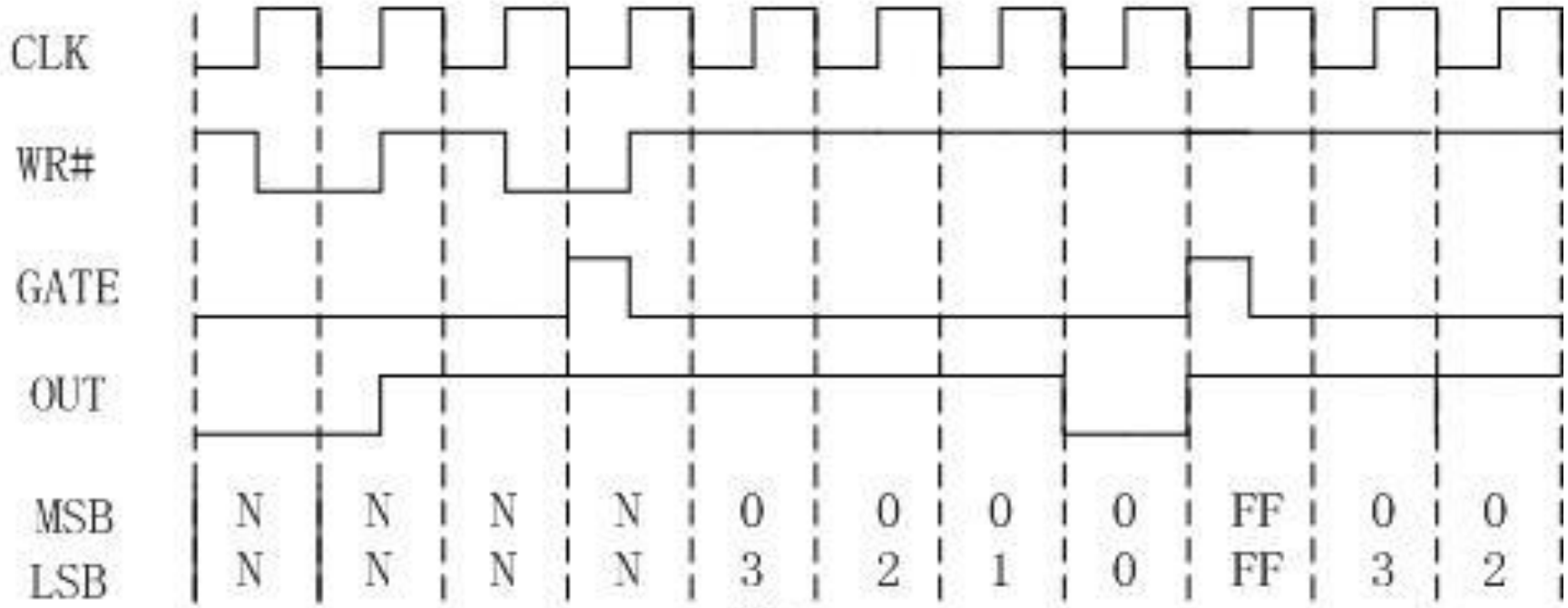
CW=18H, LSB=4H, LSB=2H



(c)

- （6）方式5 （硬件触发选通方式）
- 写入控制字后，输出OUT即为高电平。写入计数初值后，计数器并不立即开始计数，而是由GATE门控脉冲的上升沿触发。计数结束 （计数器减到0），输出一个持续时间为一个CLK时钟周期的负脉冲，然后输出恢复为高电平。输出负脉冲是通过硬件电路产生的门控信号上升沿触发得到的，所以叫硬件触发选通方式。

CW=1AH, LSB=3H

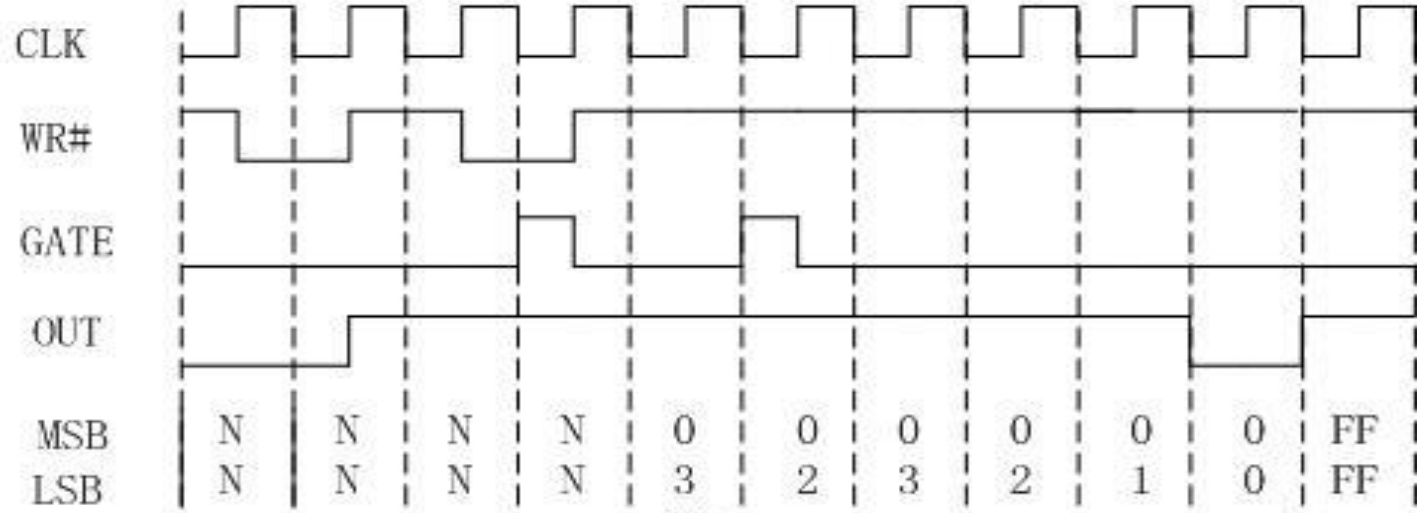


(a)

- 写入控制字后的第一个时钟上升沿，输出端OUT变成高电平。写入计数初值后，等待GATE信号的上升沿触发，然后下一个时钟信号下降沿开始减1计数。减到0时，输出端OUT输出一个CLK脉冲宽度的负脉冲，计数停止。当GATE上升沿到来时，触发计数初值重新装入，开始新一轮计数。

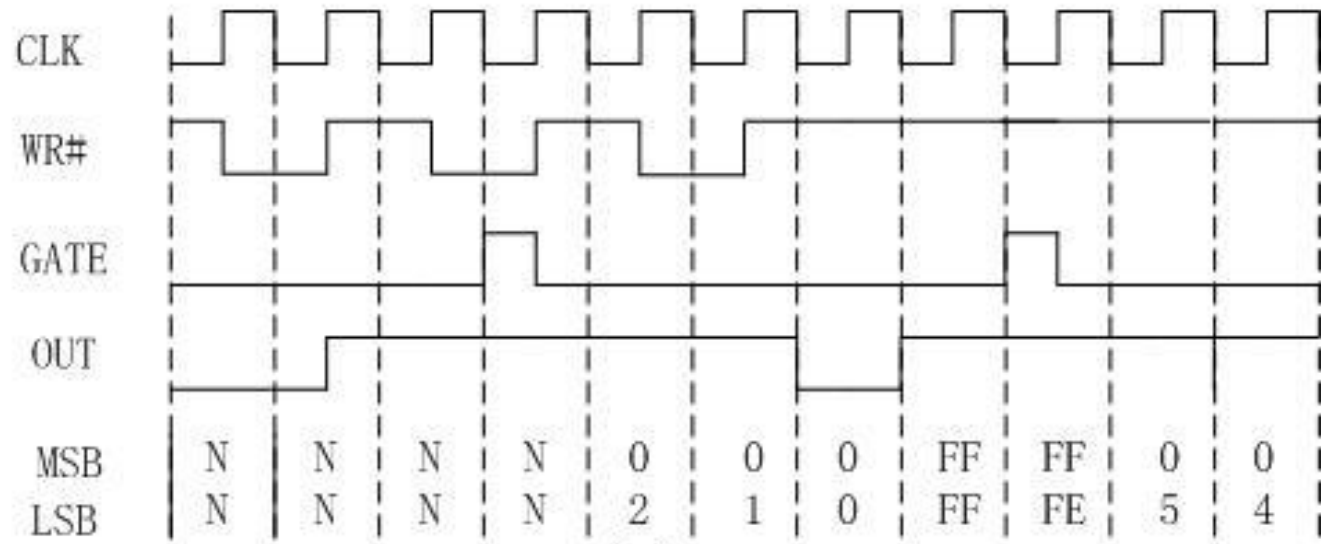
- 在计数过程中，若又输入一个门控信号的上升沿，则立即终止当前的计数过程，在下一个时钟下降沿，又从初值开始重新计数。也就是说，门控信号的上升沿到来后，会立即触发一个新的计数过程。
- 在计数过程中写入新的初值，新的初值不会立即影响当前的计数过程，即新的初值不会生效。只有在下一个门控信号上升沿到来后，才从新的初值开始计数。即新的计数初值需要门控信号上升沿触发后才有效。

CW=1AH, LSB=3H



(b)

CW=1AH, LSB=2H, LSB=5



(c)

六种工作方式总结

		方式 0	方式 1	方式 2	方式 3	方式 4	方式 5
OUT 输出状态	写控制字后	变 0	变 1	变 1	变 0	变 1	变 1
	波形宽度	N+1	N	N	N	N+1	N+1
初值是否自动重装		否	否	是	是	否	否
计数过程改变初值		立即有效	GATE 触发后有效	计数结束或 GATE 触发后有效	计数结束或 GATE 触发后有效	立即有效	GATE 触发后有效
GATE	0	禁止计数	无影响	禁止计数	禁止计数	禁止计数	无影响
	下降沿	暂停计数	无影响	停止计数	停止计数	停止计数	无影响
	上升沿	继续计数	从初值开始重新计数	从初值开始重新计数	从初值开始重新计数	从初值开始重新计数	从初值开始重新计数
	1	允许计数	无影响	允许计数	允许计数	允许计数	无影响