Segment 7

Programmable Interval Timer 8254

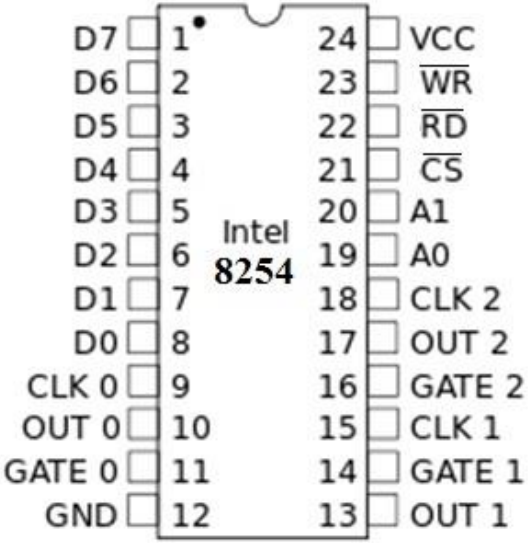
Contents:

- ✓ Why 8254?
- ✓ 8254 Vs 8253.
- ✓ Features and Application of 8254.
- ✓ Pin functions.
- ✓ Read and Write Operation.
- ✓ Modes of operation.
- ✓ Problems on 8086-8254 based system.

Prepared by:
Mohammed Abdul kader
Lecturer, EEE
International Islamic University Chittagong.

What is 8254?

The Intel 8254 is a counter/timer device designed to solve the common timing control problems in microcomputer system design. 8254 is the high speed version of the 8253.



8254 Vs 8253

Characteristics	8254	8253
Frequency Range	DC to 8 MHz DC to 10 MHz (8254-2)	DC to 2 MHz
Special Command	Status read-back command	No such command

Application

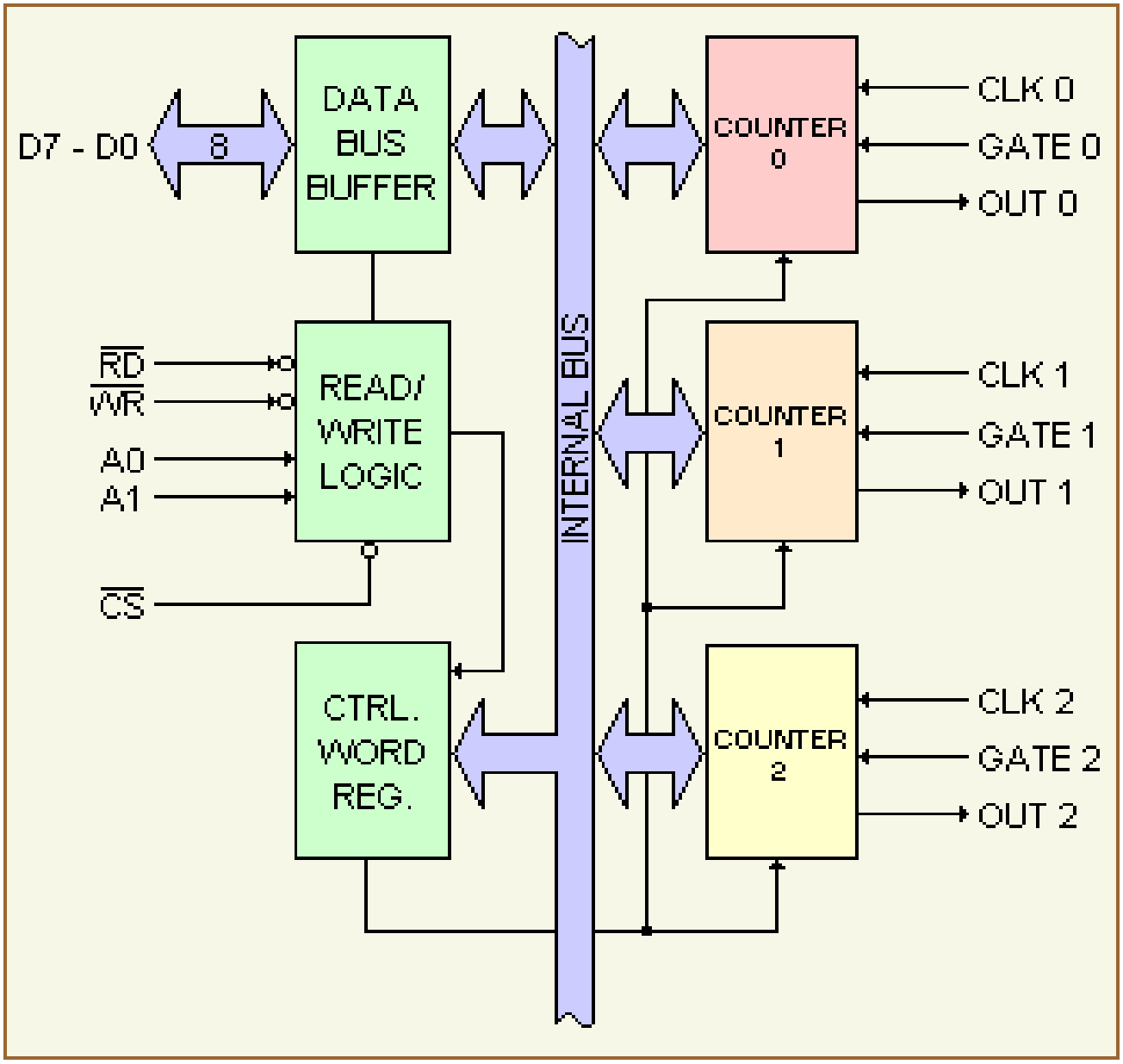
Some of the other counter/timer functions common to microcomputers which can be implemented with the 8254 are:

- ✓ Real time clock
- ✓ Event-counter
- ✓ Digital one-shot
- ✓ Programmable rate generator
- ✓ Square wave generator
- ✓ Binary rate multiplier
- ✓ Complex waveform generator
- ✓ Complex motor controller

8254 Features

- ✓ It includes three 16-bit counters that can work independently in 6 different modes.
- ✓ It is packaged in a 24-pin DIP(Dual in-line package) and requires +5V power supply.
- ✓ It can count either in binary or BCD.
- ✓ It's counters can operate at a maximum frequency of 10 MHz.

Functional Diagram



Pin functions

A0, A1: The address inputs select one of the four internal registers within the 8254.

A1	A0	Function
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word

$\overline{\text{CS}}$: Chip select enables the 8254 for programming and for reading or writing a counter.

Vcc: Power connects to the +5V power supply.

GND: Ground connects to the system ground bus.

GATE : The gate input controls the operation of the counter in some modes of operation.

GATE 0	Gate input of counter 0
GATE 1	Gate input of counter 1
GATE 2	Gate input of counter 2

Pin functions (Continued)

D0-D7: Bidirectional three state data bus lines connected to system data bus.

CLK : The clock input is the timing source for each of the internal counters. This input is often connected to the PCLK signal from the microprocessor system bus controller.

CLK0	Clock input of counter 0
CLK1	Clock input of counter 1
CLK2	Clock input of counter 2

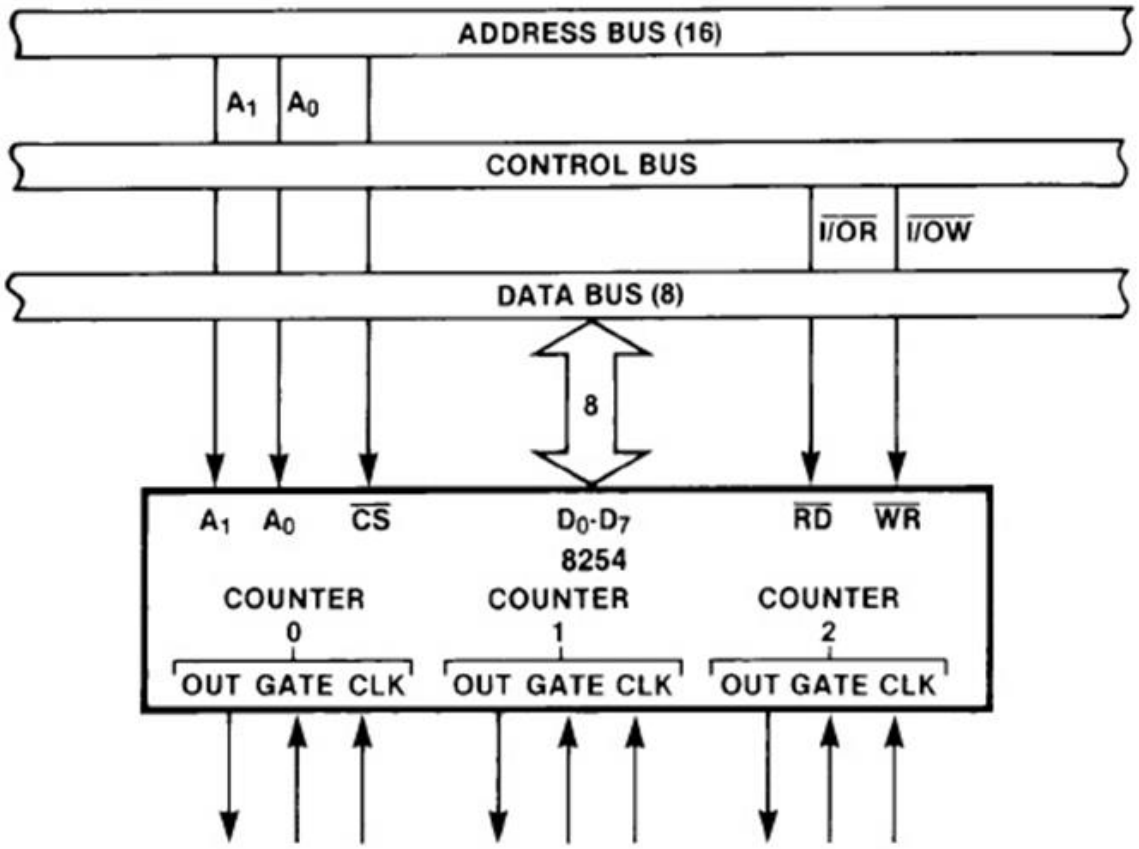
OUT: A counter output is where the waveform generated by the counter is available.

OUT 0	Output of counter 0
OUT 1	Output of counter 1
OUT 2	Output of counter 2

$\overline{\text{RD}}$: Read causes data to be read from the 8254 and often connected to the $\overline{IORC}$ signal.

$\overline{\text{WR}}$: Write causes data to be written to the 8254 and often connects to the write strobe ($\overline{IOWC}$)

8254 System Interface



Programming the 8254 (Control Word Format)

$A_1, A_0 = 11$ $\overline{CS} = 0$ $\overline{RD} = 1$ $\overline{WR} = 0$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC—Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (see Read Operations)

M—Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW—Read/Write

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first, then most significant byte

BCD

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

8254 Write operation

The programming procedure for the 8254 is very flexible. Only two conversion need to be remember.

1) For each Counter, the Control Word must be written before the initial count is written.

2) The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

With a clock and an appropriate gate signal to one of the counters, the above steps should start the counter and provide appropriate output according to the control word.

8254 Read Operations

There are three possible methods for reading the counters:

- ✓ A simple read operation
- ✓ The Counter Latch Command, and
- ✓ The Read-Back Command.

Simple Read Operation

- ✓ This operation read the counter after stopping.
- ✓ To read the Counter, which is selected with the A1, A0 inputs, the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in the process of changing when it is read, giving an undefined result.
- ✓ Two I/O read operation are performed by the MPU

    1. The first I/O operation reads the low order byte.
    2. The second I/O operation reads high order byte.

Counter Latch Command

- ✓ This allows reading the contents of the Counters "on the fly" without affecting counting in progress.
- ✓ The selected Counter's output latch (OL) latches the count at the time the *Counter Latch Command* is received.
- ✓ This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE).

Counter Latch Command (Continued)

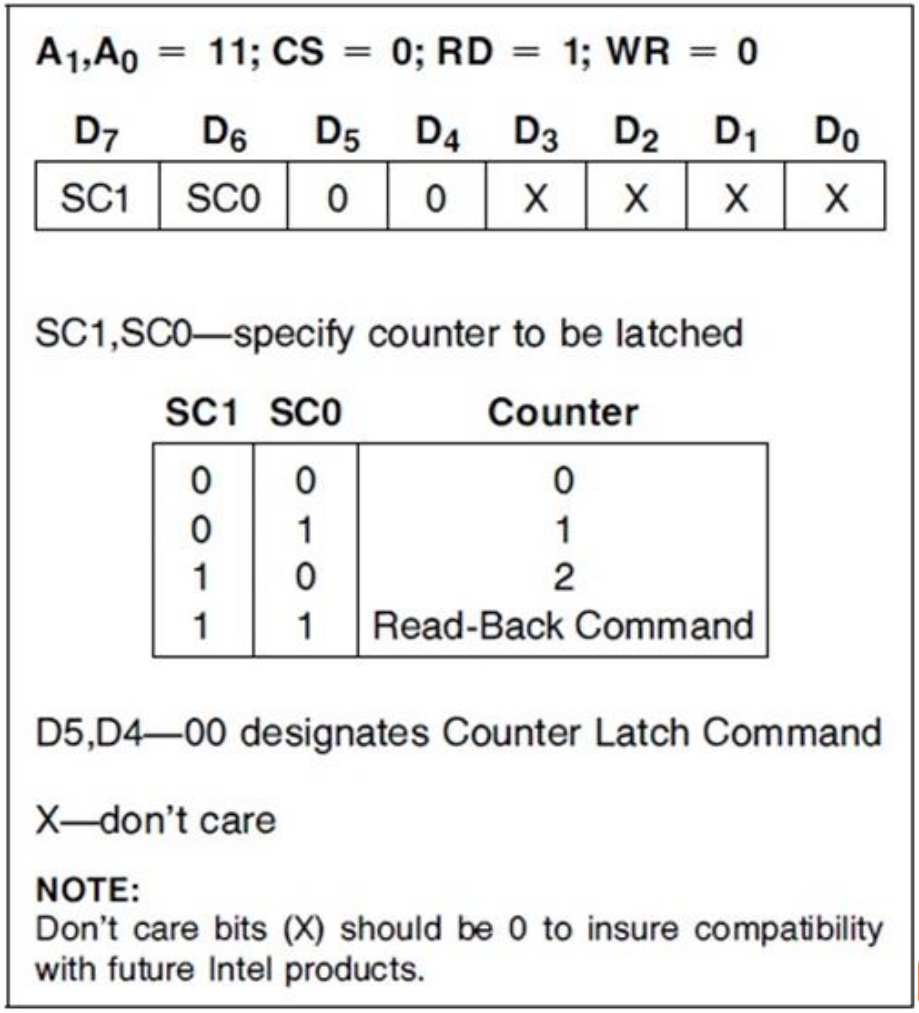
- ✓ Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

- ✓ Example:

```
; Latching counter0
MOV DX, C_REG
MOV  AL, 00000000B ; count latched
                     for counter 0.
OUT DX, AL

;Reading counter0

MOV DX, CNTR0
IN AL, DX
```



A_1, A_0 = 11; CS = 0; RD = 1; WR = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	0	0	X	X	X	X

SC1,SC0—specify counter to be latched

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Read-Back Command

D5,D4—00 designates Counter Latch Command

X—don't care

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Fig: Counter Latching Command Format

Read-Back Command

- ✓ This command is used to read several counters at a time. It eliminates the need of writing separate counter-latch commands for different counters.
- ✓ It allows the user to check the count value, programmed Mode, and current states of the OUT pin and Null Count flag of the selected counter/ counters.
- ✓ The read back command is written to the Control Word Register.
- ✓ The command is written into the Control Word Register and has the format shown in Figure.
- ✓ The read-back command may be used to latch multiple counter output latches (OL) by setting the COUNT bit D5 =0 and selecting the desired counter(s).
- ✓ A single **read back command** is functionally equivalent to several **counter latch commands.**

A0, A1 = 11 **$\overline{CS}$ = 0** **$\overline{RD}$ = 1** **$\overline{WR}$ = 0**

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT 2	CNT 1	CNT 0	0

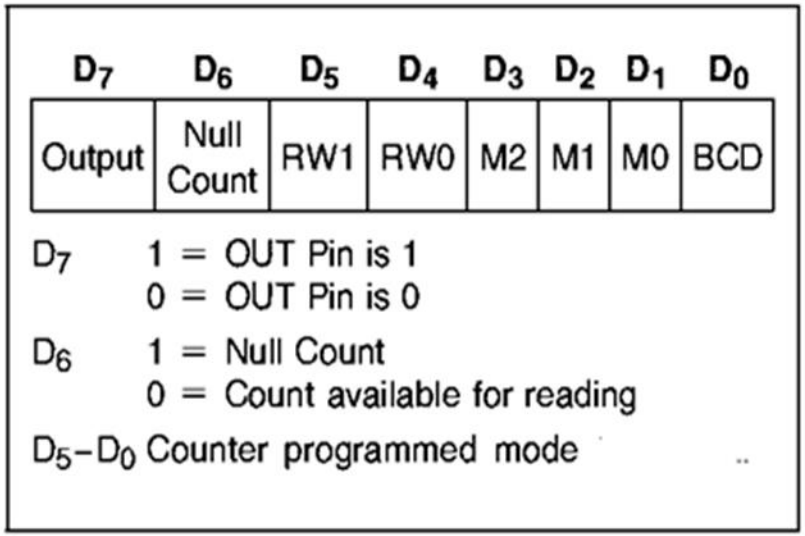
D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counters(s)
D_3: 1 = Select Counter 2
D_2: 1 = Select Counter 1
D_1: 1 = Select Counter 0
D_0: Reserved for future expansion; Must be 0

Figure Read-Back Command Format

- ✓ Each counter's latched count is held in the OL until it is read (or the counter is reprogrammed). The counter is automatically unlatched when read, but other counters remain latched until they are read.

Read-Back Command (Continued)

- ✓ The read-back command may also be used to latch status information of selected counter(s) by setting STATUS bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.
- ✓ The counter status format is shown in Figure below.



D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Output	Null Count	RW1	RW0	M2	M1	M0	BCD

D_7 1 = OUT Pin is 1
0 = OUT Pin is 0

D_6 1 = Null Count
0 = Count available for reading

D_5–D_0 Counter programmed mode

- ✓ Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word.
- ✓ OUTPUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

Read-Back Command (Continued)

Example:

```
; Count and Status latched for count 0

MOV DX, C_REG
MOV AL, 11000010B ; count latched for count 0
OUT DX, AL

; Reading the latched status for count 0

MOV DX, TRM0
IN AL, DX  ; Reading Status
MOV AH, AL


; Reading the latched count for counter 0

IN AL, DX       ; Reading LSB of counter 0
MOV BL, AL
IN AL, DX       ; Reading MSB of counter 0
MOV BH, AL
```

Modes of Operation

Mode 0: Interrupt on terminal count.

Mode 1: Hardware Retriggerable One-Shot.

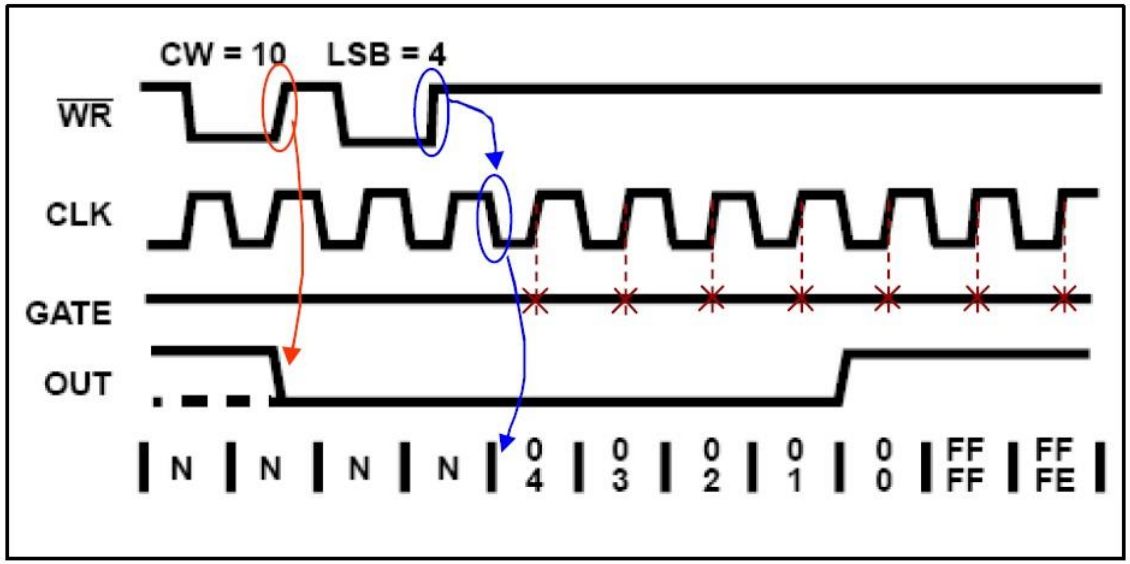
Mode 2: Rate Generator.

Mode 3: Square Wave Mode.

Mode 4: Software Triggered Mode.

Mode 5: Hardware Triggered Mode.

Mode 0: Interrupt on terminal count.

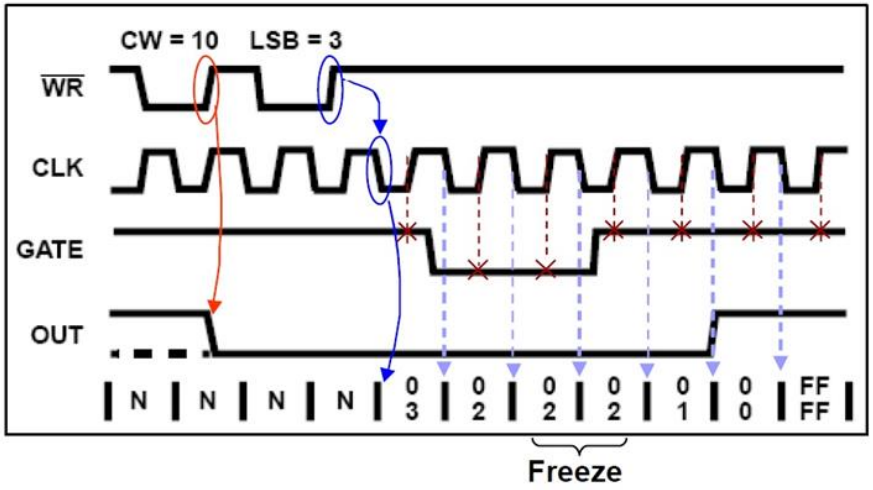


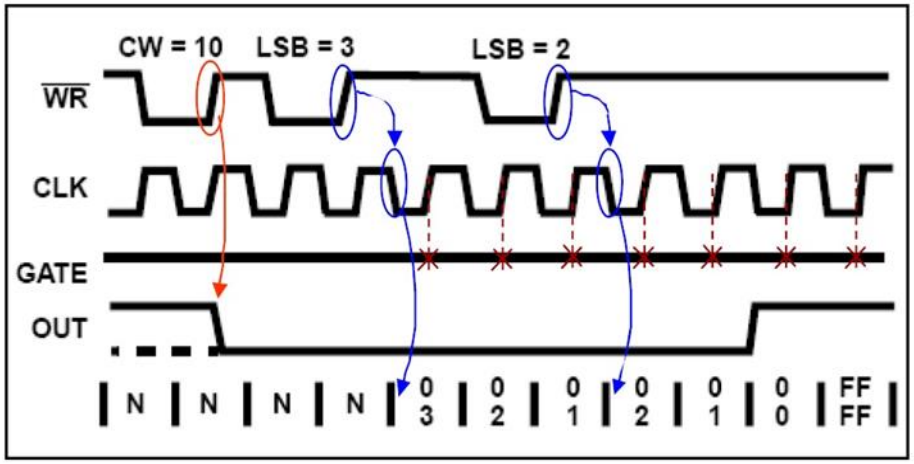
* N stands for an undefined count.

- ✓ Mode 0 is typically used for event counting.
- ✓ After the Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the Counter.
- ✓ After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not go high until N +1 CLK pulses after the initial count is written.

Mode 0: Interrupt on terminal count. (Continued)

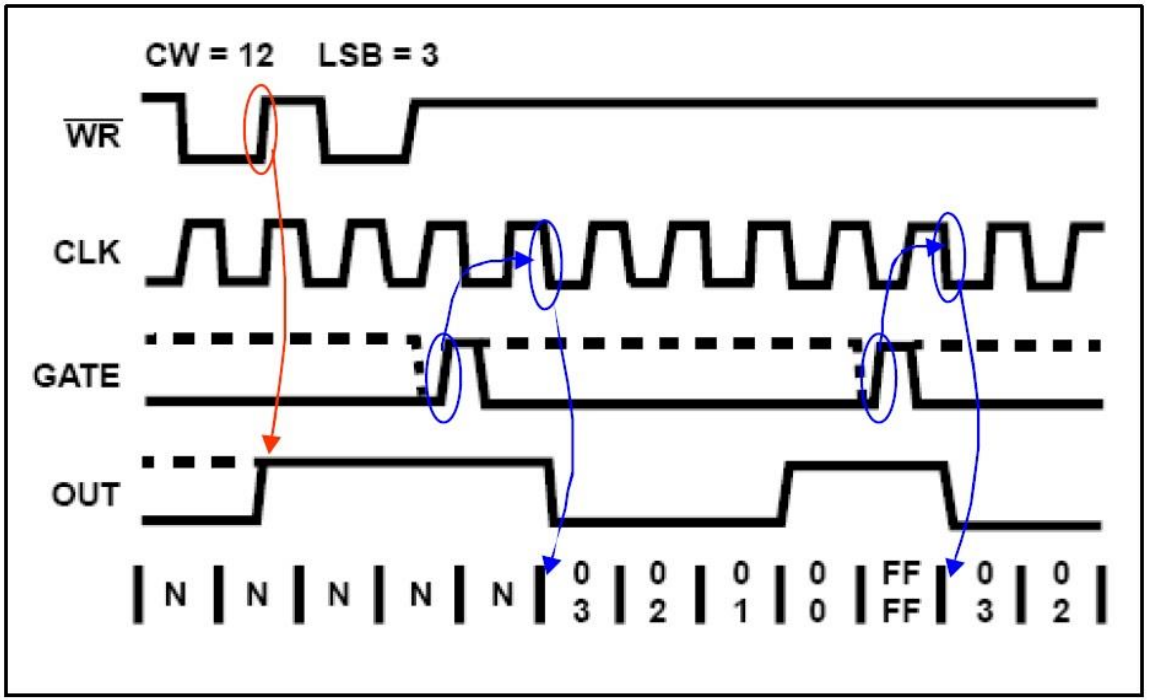
- ✓ GATE =1 enables counting; GATE = 0 disables counting.
- ✓ GATE has no effect on OUT. If G becomes a logic 0 in the middle of the count, the counter will remain stop until G again becomes a logic 1.
- ✓ If a new count is written to the Counter, it will be loaded on the next CLK pulse and counting will continue from the new count.





- ✓ At the rising edge of WR(CW) OUT becomes high.
- ✓ At the first falling edge of clock after first rising edge of WR(LSB), counter starts counting.

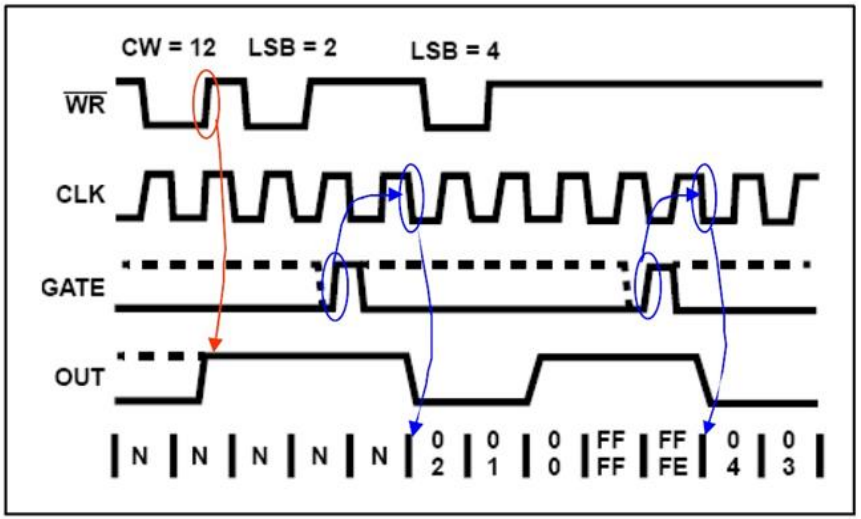
Mode 1: Hardware Retriggerable One-Shot.

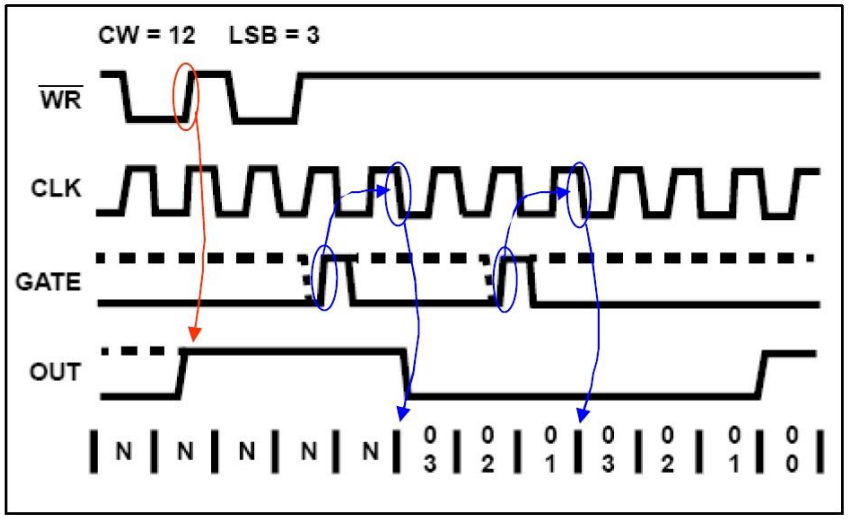


- ✓ Causes the counter to function as a retriggerable, monostable multivibrator (one-shot).
- ✓ OUT is initially (after loading CW) high. Also remain high when count is written.
- ✓ When gate is triggered, OUT goes low and will remain low until the Counter reaches zero. On completion of count OUT goes high again.

Mode 1: Hardware Retriggerable One-Shot. (Continued)

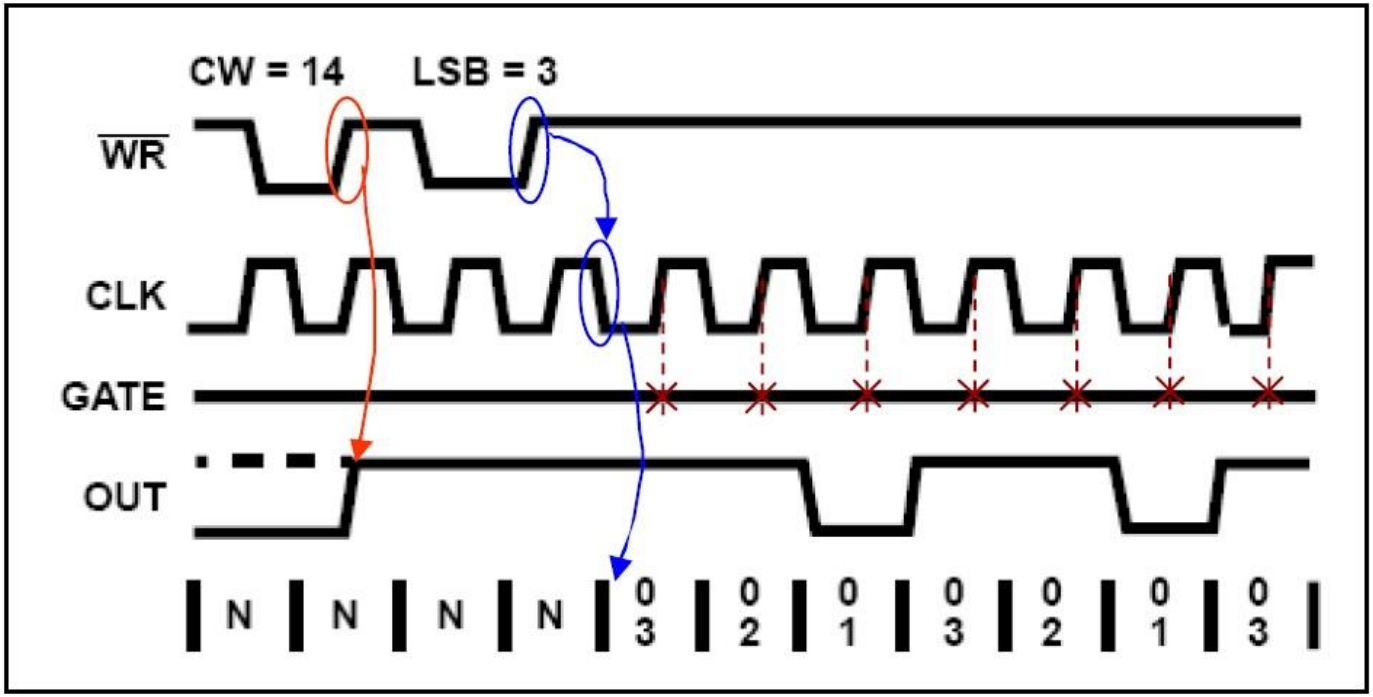
- ✓ If the GATE input occurs within the duration of counting, the counter is again reloaded with the count and start counting from the beginning.





- ✓ At the rising edge of WR(CW) OUT becomes high.
- ✓ At the first falling edge of clock after first rising edge of GATE counter starts counting.

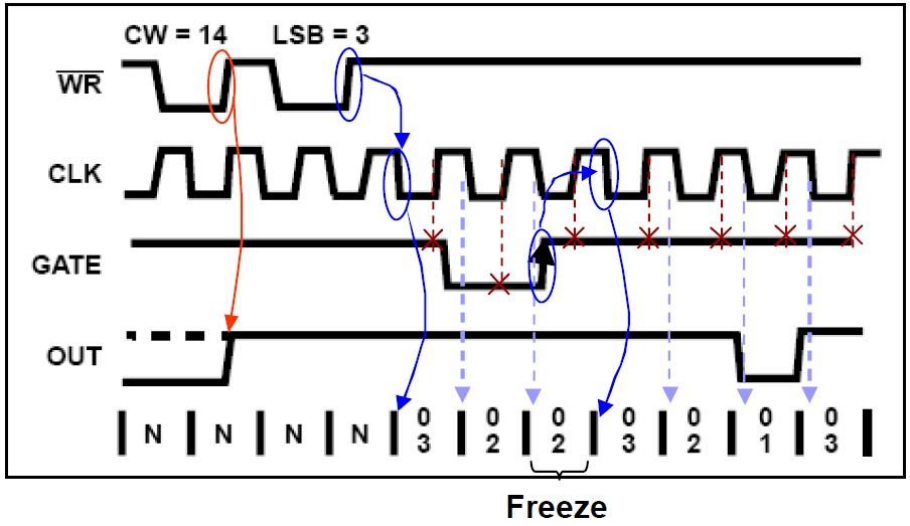
Mode 2: RATE GENERATOR

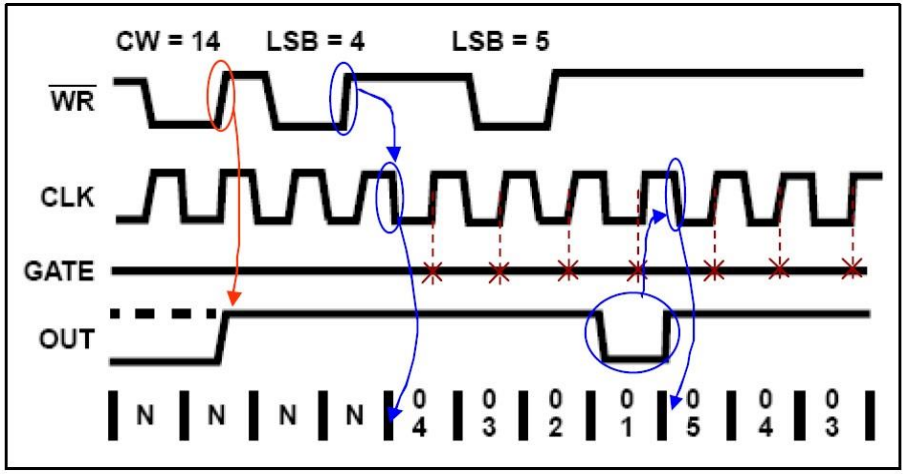


- ✓ Allows the counter to generate a series of continuous pulses that are one clock pulse wide.
- ✓ The separation between pulses is determined by the count.
- ✓ If count N is loaded then, output will remain high for (N-1) clock period and low for 1 clock period.

Mode 2: RATE GENERATOR (Continued)

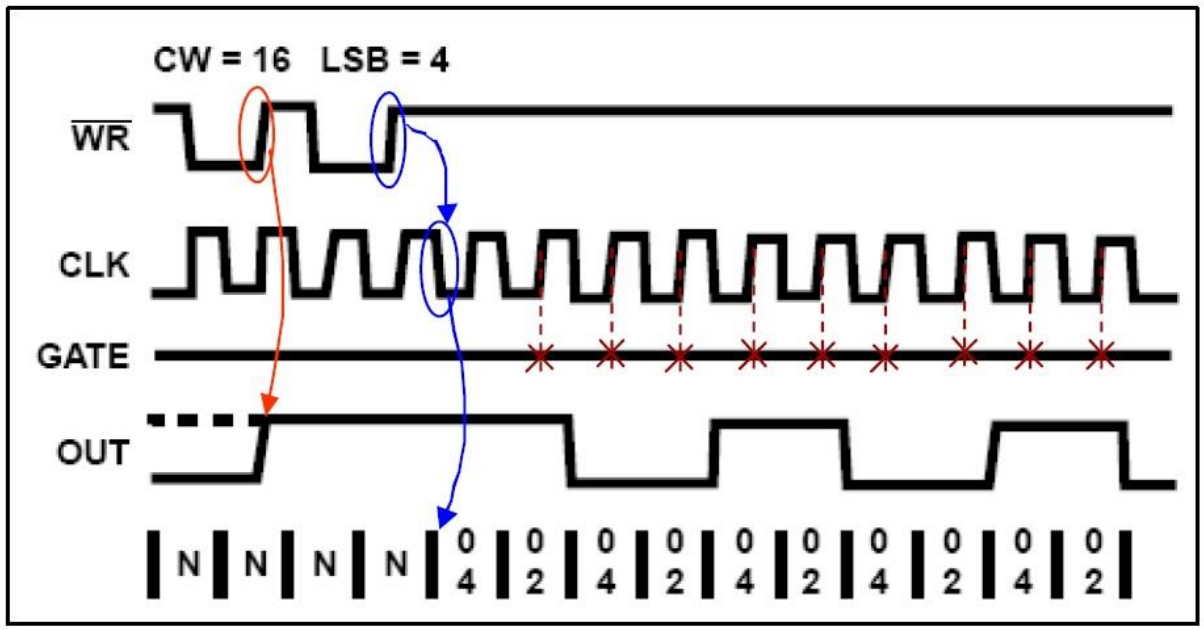
- ✓ For example, for a count of 10, the output is a logic 1 for nine clock period and low for 1 clock period. This cycle is repeated until the counter is programmed with a new count or until G pin is placed at a logic 0 level.
- ✓ The G input must be logic 1 for this mode to generate a continuous series of pulses.
- ✓ In mode 2, a COUNT of 1 is illegal.





- ✓ At the rising edge of WR(CW) OUT becomes high.
- ✓ At the first falling edge of clock after first rising edge of WR(LSB), counter starts counting.

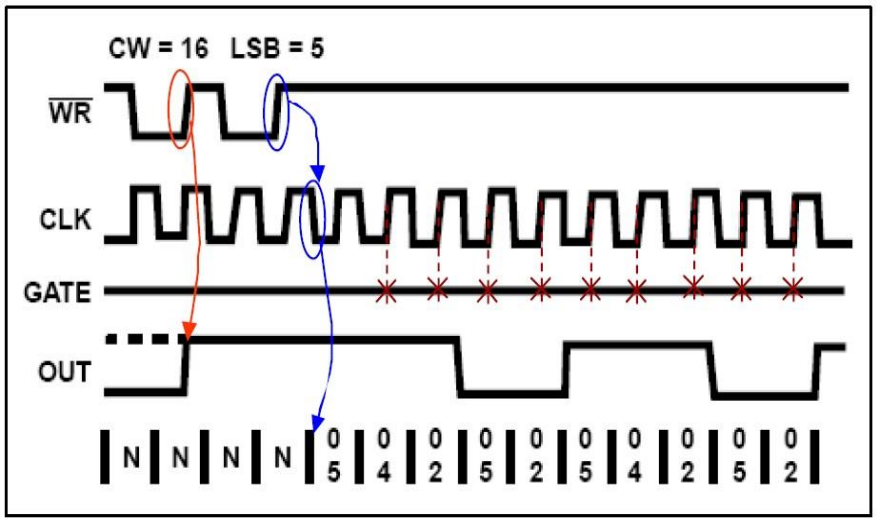
Mode 3: Square Wave Mode.

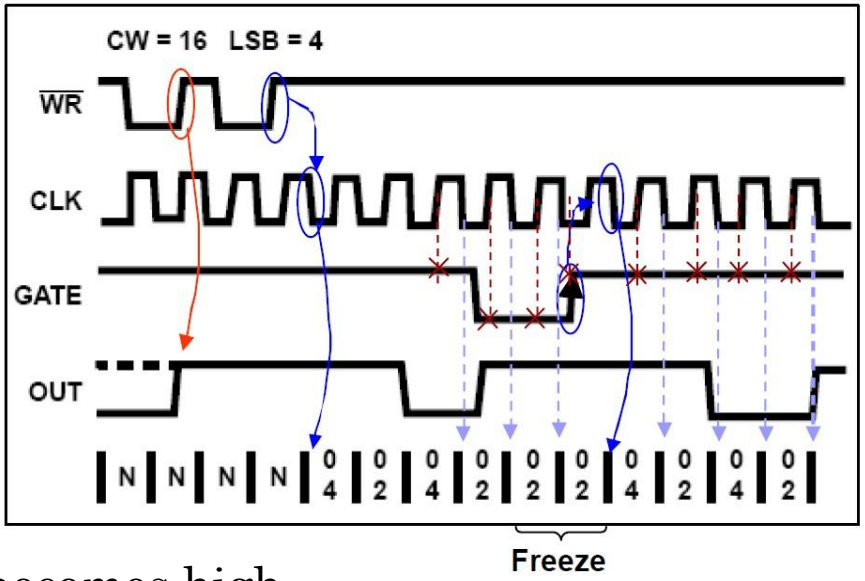


- ✓ Generates a continuous square wave at the out connection.
- ✓ Mode 3 is similar to Mode 2 except for the duty cycle of OUT.
- ✓ If the count (N) is even, the output is high for one half (N/2) of the count and low for one half (N/2) of the count.
- ✓ If the count (N) is odd, the output is high for one clocking period longer than it is low i.e. high for (N+1)/2 clock pulses and low for (N-1)/2 clock pulses.

Mode 3: Square Wave Mode. (Continued)

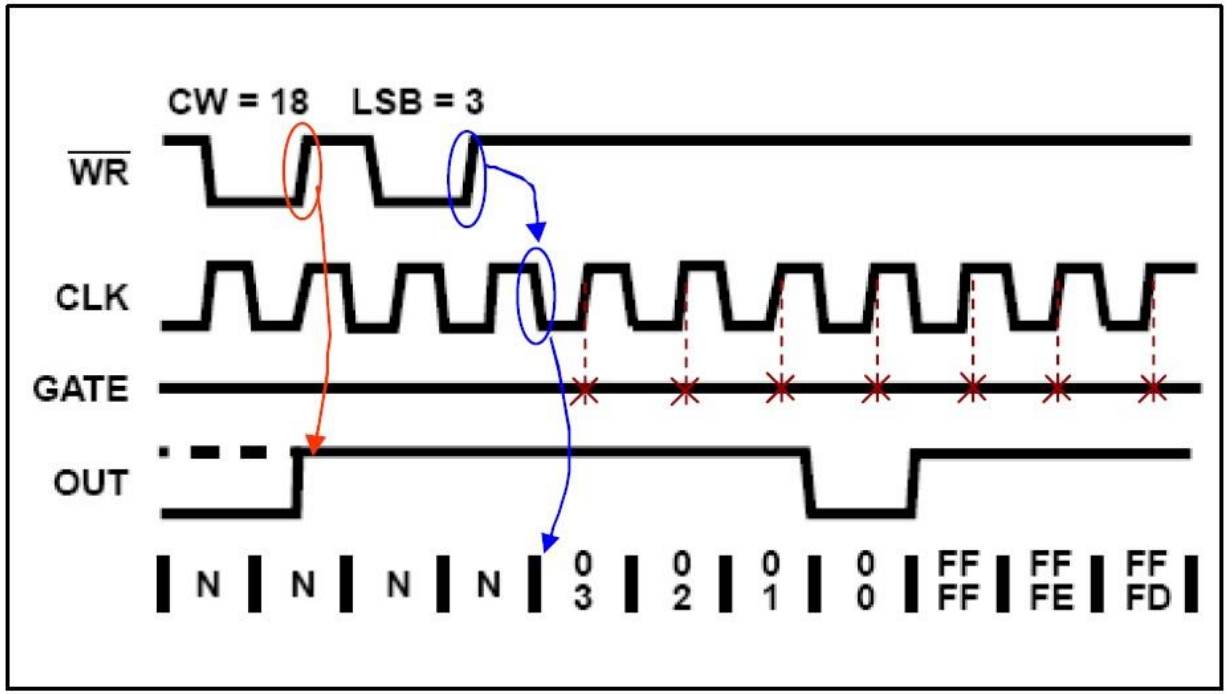
- ✓ For example, if the count is programmed for a count of 5, the output is high for three clocks and low for two clocks.
- ✓ Gate should be maintained at logic 1 always (GATE =1 enables counting; GATE =0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required).





- ✓ At the rising edge of WR(CW) OUT becomes high.
- ✓ At the first falling edge of clock after first rising edge of WR(LSB), counter starts counting.

Mode 4: Software Triggered One-shot.

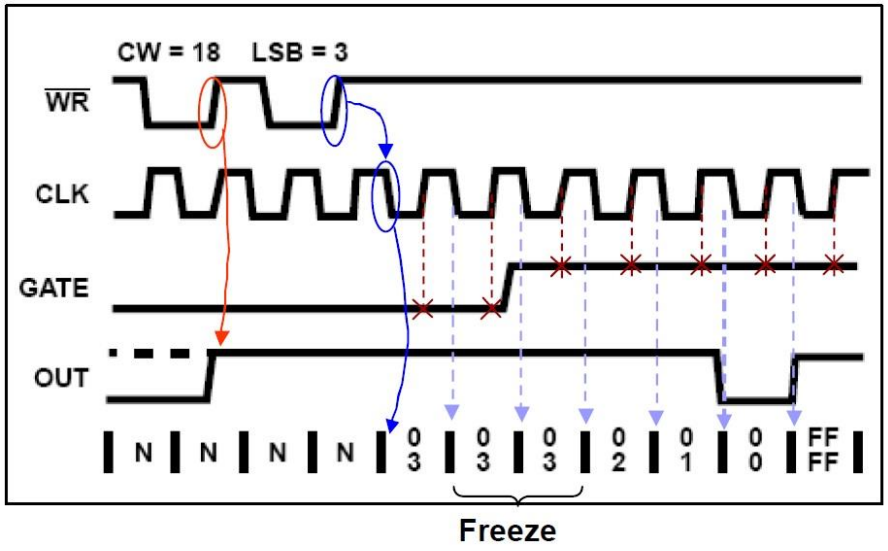


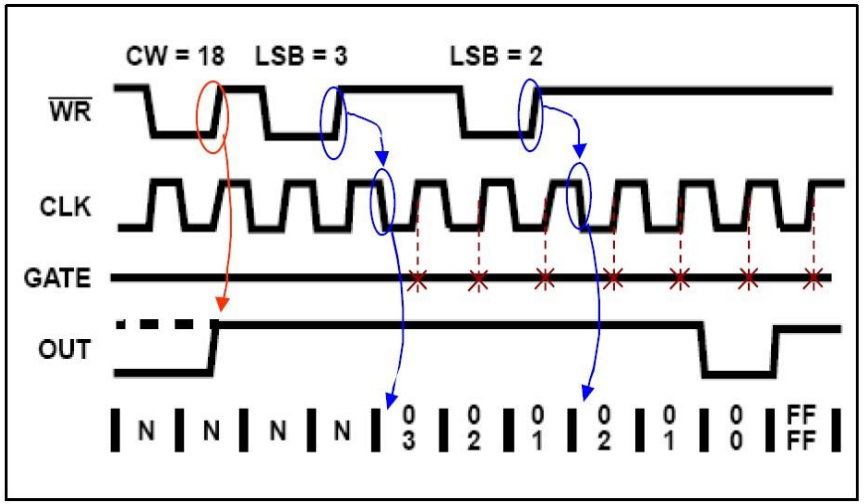
- ✓ Allows the counter to produce a single pulse at the output.
- ✓ If count of N is loaded, then OUT will be high for N clock cycles and low for one clock cycle at the end.
- ✓ The cycle does not begin until the counter is loaded again.

Mode 4: Software Triggered One-shot. (Continued)

- ✓ G input must be maintained at logic 1 throughout the operation.
- ✓ This mode operates as a software triggered one-shot.

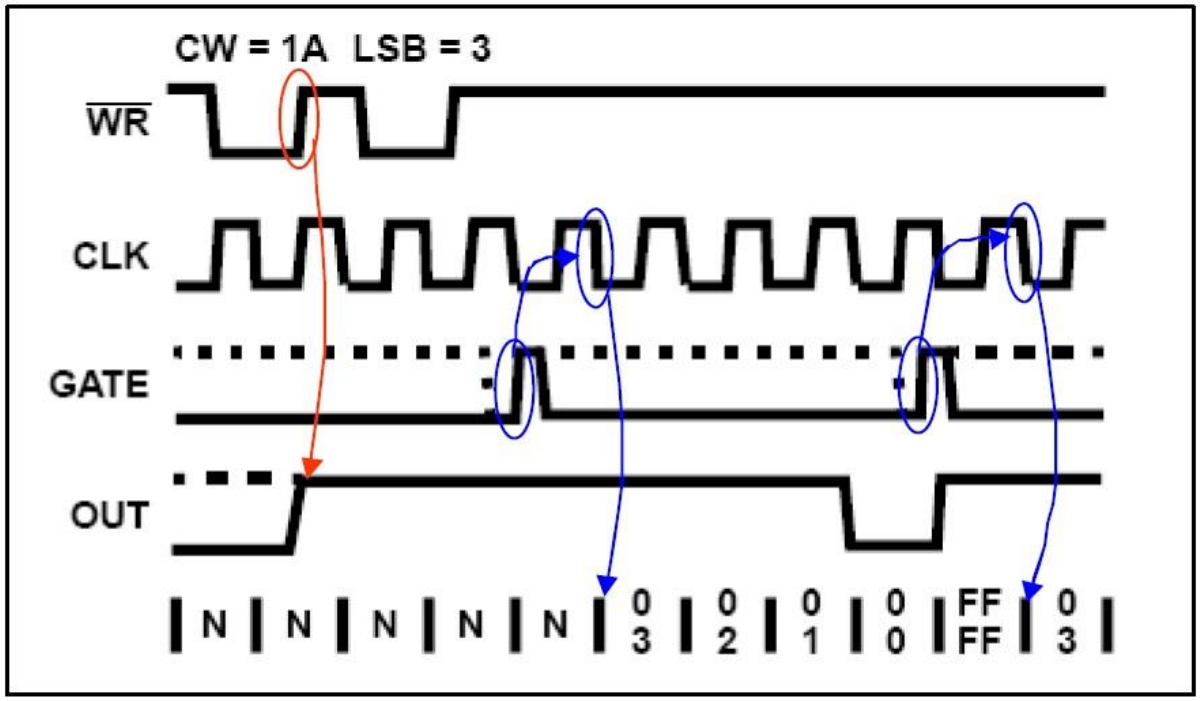
N.B. The G input must be a logic 1 for the counter to operate for these three modes (Mode 2, 3, 4)





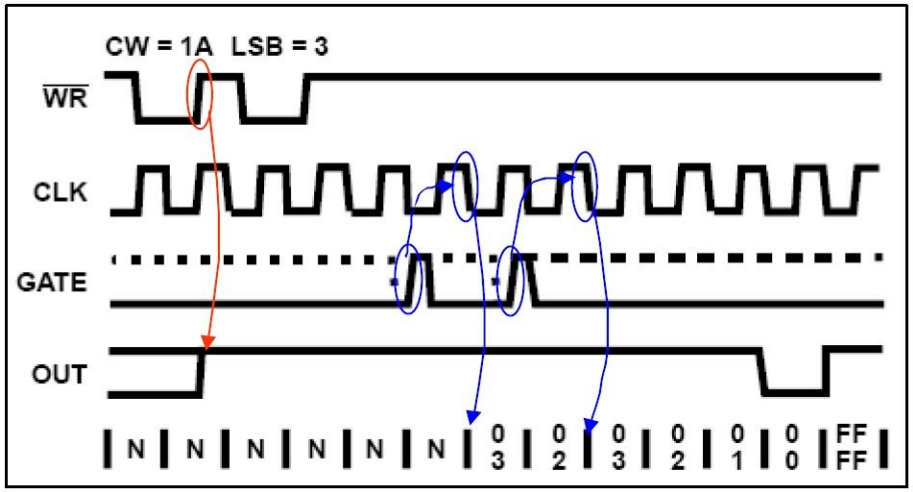
- ✓ At the rising edge of WR(CW) OUT becomes high.
- ✓ At the first falling edge of clock after first rising edge of WR(LSB), counter starts counting.

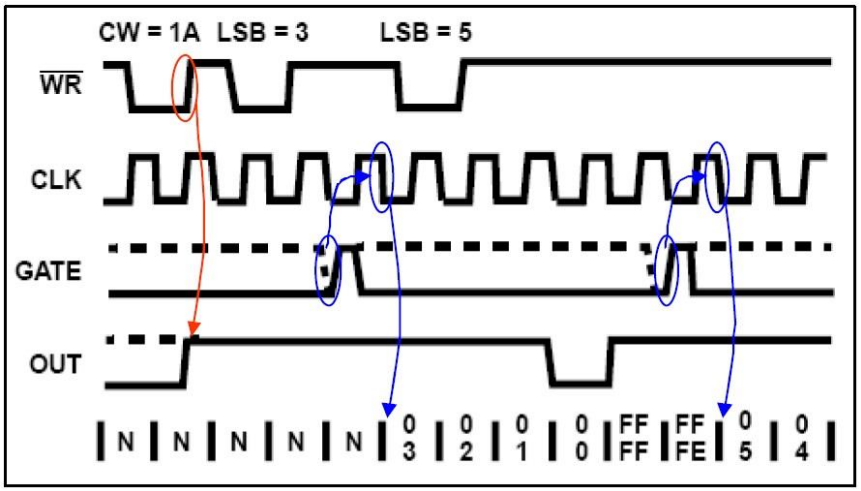
Mode 5: Hardware Triggered Mode.



- ✓ A hardware triggered one-shot that function as mode 4, except that it is started by a trigger pulse on the G pin instead of by software.
- ✓ When the GATE pulse is triggered from low to high the count begins. At the end of the count OUT goes low for one clock period.
- ✓ This mode is also called HARDWARE TRIGGERED STROBE (RETRIGGERABLE)

Mode 5: Hardware Triggered Mode.

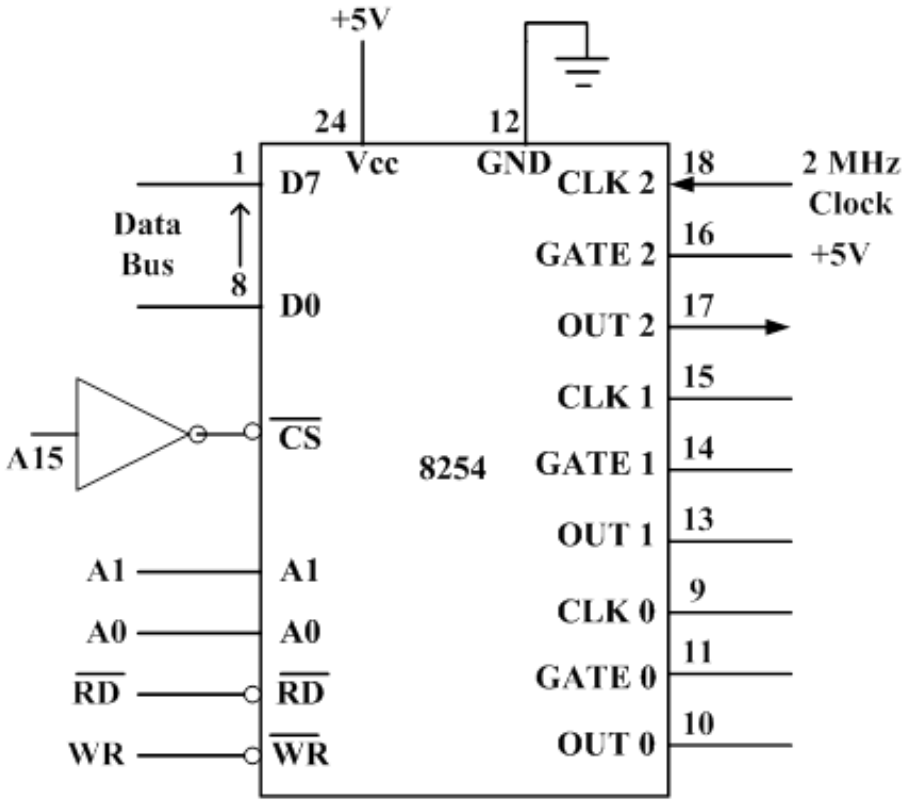




- ✓ At the rising edge of WR(CW) OUT becomes high.
- ✓ At the first falling edge of clock after first rising edge of GATE , counter starts counting.

Problem 1

(a) Identify the port address of the control register and counter 2 in figure.
(b) Write a subroutine to initialize counter 2 in mode 0 with a count of 50,000. The subroutine should also include reading counts on the fly; when count reaches zero, it should return to the main program.
(c) Write a main program to display seconds by calling the subroutine as many times as necessary.



Problem 1 (Continued)

Solution (a):

A15	A14	A13	A12	A11.....A5	A4	A3	A2	A1	A0	Address	
1	0	0	0	.....	0	0	0	0	0	8000H	Counter 0
1	0	0	0	.....	0	0	0	0	1	8001H	Counter 1
1	0	0	0	.....	0	0	0	1	0	8002H	Counter 2
1	0	0	0	.....	0	0	0	1	1	8003H	Control Register

Address of counter 2 = 8002H
Address of control register = 8003H

Solution (b):

We have to initialize counter 2 in Mode0.
Count=$(50,000)_{10}$ = C350H
Control Word to initialize counter 2 in mode0 and to load 16-bit count:

1	0	1	1	0	0	0	0	B0H
Counter 2		Load 16 bit count		Mode 0			Count in binary	Control Word

To read count 2 on the fly counter latch command is:

1	0	0	0	0	0	0	0	80H
Counter 2		Counter latch command		Don't care				Counter latch command

Problem 1 (Continued)

Solution (b)
Subroutine:

```
COUNTER  PROC  NEAR
     CNT2   EQU   8002H
     CNTR  EQU   8003H
     MOV  AL, B0H
     OUT   CNTR, AL
     MOV   AL, 50H
     OUT  CNTR2, AL
     MOV AL, C3H
     OUT  CNTR2, AL
  READ: MOV AL, 80H
          OUT  CNTR, AL
          IN  AL, CNT2
          MOV  DL, AL
          IN  AL, CNT2
          OR  AL, DL
         JNZ  READ
          RET
COUNTER  ENDP
```

Problem 1 (Continued)

Solution (c)

Clock frequency, $f_c = 2$ MHz

Time period of each clock cycle,

$$t_c = \frac{1}{2 \times 10^6} = 5 \times 10^{-7} sec$$

Every time subroutine is called then, $50000 \times 5 \times 10^{-7} = 25\ ms$ is counted.

To count 1 sec subroutine needed to be called, $\left(\frac{1s}{25ms}\right) = 40\ times = 28H\ times$

Main Program:

Assuming segment registers are already initialized.

```
MOV  BL, 00H
SECOND:   MOV CL, 28H
WAIT:  CALL  COUNTER
          DEC  CL
          JNZ   WAIT
 MOV  AL, BL
ADD   AL, 01
DAA
OUT  25H, AL ; assuming 8 bit port 25H
MOV  BL, AL
JMP  SECOND
HLT
```

Problem 2

Write instructions to generate a pulse in every 50 us later from counter 0. Consider the figure of problem1.

Solution

To generate a pulse in every 50 us later, we should initialize counter0 in mode 2. Gate0 should be high.

Count:

Clock frequency, $f_c = 2$ MHz

Needed count, $N = \frac{pulse\ time}{clock\ period} = 100 = 64H$

Control Word

0	0	0	1	0	1	0	0	14H
Counter 0		Load least significant byte only		Mode 2			Count in binary	Control Word

Instructions:

```
CNT0   EQU   8000H
CNTR   EQU   8003H
MOV    AL, 14H
OUT    CNTR, AL
MOV    AL, 64H
OUT    CNT0, AL
```

Problem 3

Write instructions to generate a 1 KHz square wave from Counter1. Assume the gate of counter1 is tied to +5V through a 10K resistor. Explain the significance of connecting the gate to +5V. (use figure of problem 1)

Solution

To generate a square wave from counter1, it should be initialize in Mode 3.
Needed count = 2000 = $07D0H$
Control Word:

0	1	1	1	0	1	1	0	76H Control Word
Counter 1		Load both byte		Mode 3			Count in binary	

Instructions:

```
CNT1   EQU   8001H
CNTR  EQU   8003H
MOV   AL, 76H
OUT   CNTR, AL
MOV   AL, D0H
OUT   CNT1, AL
MOV   AL, 07H
OUT    CNT1, AL
```

Gate should be maintained at logic 1 always (GATE =1 enables counting; GATE =0 disables counting.

Problem 4

Write a subroutine to generate an interrupt every 1sec. Consider the figure of problem1.

Solution

To obtain a pulse every 1 sec, the count should be 2×2^6, which is too large for one 16-bit counter. We can divide this counter as follows:

Counter 1 [gate pulse=2 MHz]--- Mode 2 ----- Count 50,000 (C350H)

Counter 2 [gate pulse=out1 (output of counter 1)]---- Mode 2 ---- Count 40 (28H)

So, finally we get $50{,}000 \times 40 = 2 \times 10^6$ count and from OUT2 we get the desired output.

Control word:

Counter 1= 01 11 010 0 = 74H

Counter 2= 10 01 010 0=94H

Subroutine:

```
INTRP   PROC    NEAR
CNT1    EQU    8001 H
CNT2    EQU    8002H
CNTR    EQU    8003H
MOV AL, 74H
OUT CNTR, AL
MOV AL, 94H
OUT CNTR, AL
MOV AL, 50H
OUT CNT1, AL
MOV AL, C3H
OUT CNT1, AL
MOV AL, 28H
OUT CNT2, AL
RET
INTRP   ENDP
```

Problem 5:

Write the instruction to generate a 100 KHz square-wave at OUT0 and a 200 KHz continuous pulse at OUT1 of 8254. Consider a clock of 8 MHz at clk0 and clk1. Again address pins A_0 and A_1 of 8254 are directly connected to A_0 and A_1 of 8086 and A_{15} of 8086 is connected to $\overline{CS}$ of 8254 through an inverter.

Solution

To generate 100 KHz square wave at OUT0,
No. of count: x×100 KHz=8MHz; ≫x=80=50H

Control Word:

0	0	0	1	0	1	1	0	16H
Counter 0		Load lest significant byte		Mode 3			Count in binary	Control Word

To generate 200 KHz continuous pulse at OUT1,
No. of count, $x \times 200\ KHz = 8MHz;\ \gg x = 40 = 28H$

Control Word:

0	1	0	1	0	1	0	0	54H
Counter 1		Load list significant byte		Mode 3			Count in binary	Control Word

Problem 5 (Continued)

Address:

As address pins A_0 and A_1 of 8254 are directly connected to A_0 and A_1 of 8086 and A_{15} of 8086 is connected to $\overline{CS}$ of 8254 through an inverter. So,

Address of Counter 0 = 8000H
Address of Counter 1= 8001H
Address of counter 2=8002H
Address of Control Register = 8003H

Subroutine:

```
CNT0   EQU   8000 H
CNT1   EQU   8001H
CNTR   EQU   8003H
MOV AL, 16H
OUT CNTR, AL
MOV AL, 50H
OUT CNT0, AL
MOV  AL, 54H
OUT CNTR, AL
MOV AL, 28H
OUT CNT1, AL
```