UM8254

Programmable Interval Timer

Features

- Compatible with most microprocessor including 8080A, 8085A, iAPX88 and iAPX86*
- Handles inputs from DC to 8 MHz
- Six programmable counter modes
- Status read-back command
- Three independent 16-bit counters
- Binary or BCD counting
- Single +5V supply

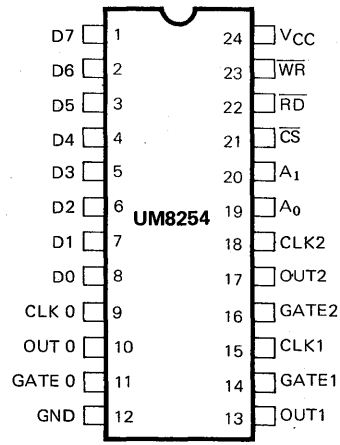
General Description

The UM8254 is a counter/timer device designed to solve the common timing control problems in microcomputer system design. It provides three independent 16-bit counters, each capable of handling clock inputs up to 8 MHz. All modes are software programmable. The UM8254 is a superset of the UM8253.

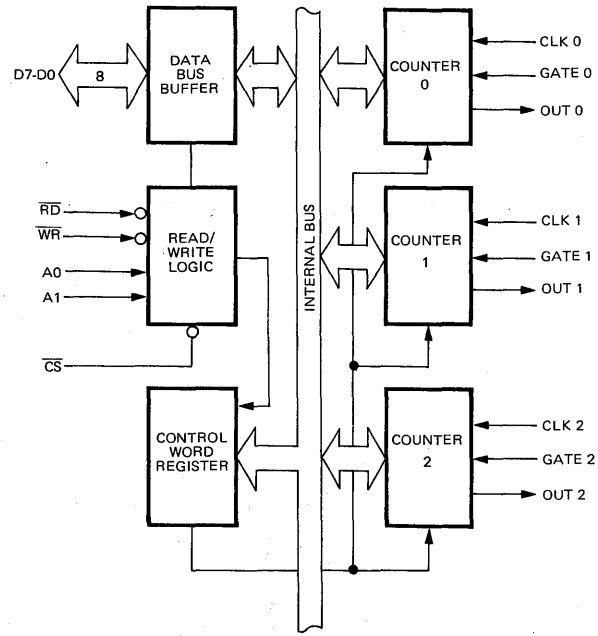
The UM8254 uses HMOS technology and comes in a 24-pin plastic package.

*iAPX88 and iAPX86 are all trademarks of Intel microsystem.

Pin Configuration



Block Diagram



Absolute Maximum Ratings*

Ambient Temperature Under Bias T_A . . . . . 0°C to 70°C

Storage Temperature T_{STG} . . . . . –65°C to +150°C

Voltage on Any Pin with Respect to Ground . . . . . –0.5V to +7V

Power Dissipation . . . . . 1 Watt

*Comments

Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied and exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

(T_A = 0°C to 70°C, V_{CC} = 5V + 10%)

Symbol	Parameter	Min.	Max.	Units	Conditions
V_{IL}	Input Low Voltage	–0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5V	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = –400 μA
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0.45V
I_{CC}	V_{CC} Supply Current		140	mA	

Capacitance

(T_A = 25°C, V_{CC} = GND = 0V)

Symbol	Parameter	Min.	Max.	Unit	Conditions
C_{IN}	Input Capacitance		10	pF	f_C = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to V_{SS}

A.C. Characteristics

(T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, GND = 0V)

Bus Parameters (Note 1)

READ CYCLE

Symbol	Parameter	UM8254		Unit
		Min.	Max.	
t_{AR}	Address Stable Before $\overline{RD}\downarrow$	45		ns
t_{SR}	$\overline{CS}$ Stable Before $\overline{RD}\downarrow$	0		ns
t_{RA}	Address Hold Time After $\overline{RD}\uparrow$	0		ns
t_{RR}	RD Pulse Width	150		ns
t_{RD}	Data Delay from $\overline{RD}\downarrow$		120	ns
t_{AD}	Data Delay from Address		220	ns
t_{DF}	$\overline{RD}\uparrow$ to Data Floating	5	90	ns
t_{RV}	Command Recovery Time	200		ns

Note: AC timings measured at V_{OH} = 2.0V, V_{OL} = 0.8V

A.C. Characteristics (Continued)

WRITE CYCLE

Symbol	Parameter	UM8254		Units
		Min.	Max.	
t_{AW}	Address Stable Before $\overline{WR}\downarrow$	0		ns
t_{SW}	$\overline{CS}$ Stable Before $\overline{WR}\downarrow$	0		ns
t_{WA}	Address Hold Time $\overline{WR}\uparrow$	0		ns
t_{WW}	$\overline{WR}$ Pulse Width	150		ns
t_{DW}	Data Setup Time Before $\overline{WR}\uparrow$	120		ns
t_{WD}	Data Hold Time After $\overline{WR}\uparrow$	0		ns
t_{RV}	Command Recovery Time	200		ns

Clock and Gate

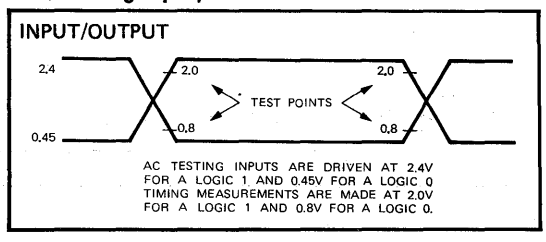
(T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, GND = 0V)

Symbol	Parameter	UM8254		Units
		Min.	Max.	
t_{CLK}	Clock Period	125	DC	ns
t_{PWH}	High Pulse Width	60(3)		ns
t_{PWL}	Low Pulse Width	60(3)		ns
t_R	Clock Rise Time		25	ns
t_F	Clock Fall Time		25	ns
t_{GW}	Gate Width High	50		ns
t_{GL}	Gate Width Low	50		ns
t_{GS}	Gate Setup Time to CLK↑	50		ns
t_{GH}	Gate Hold Time After CLK↑	50(2)		ns
t_{OD}	Output Delay from CLK↓		150	ns
t_{ODG}	Output Delay from Gate↓		120	ns
t_{WC}	CLK Delay for Loading	0	55	ns
t_{WG}	Gate Delay for Sampling	−5	50	ns
t_{WO}	OUT Delay from Mode Write		260	ns
t_{CL}	CLK Set Up for Count Latch	−40	45	ns

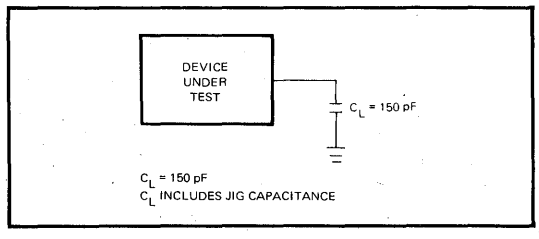
Note 2: In modes 1 and 5 triggers are sampled on each rising clock edge. A second trigger within 120 ns of the rising clock edge may not be detected.

Note 3: Low-going glitches that violate tPWH, tPWL may cause errors requirring counter reprogramming.

A.C. Testing Input, WTPUT Waveform

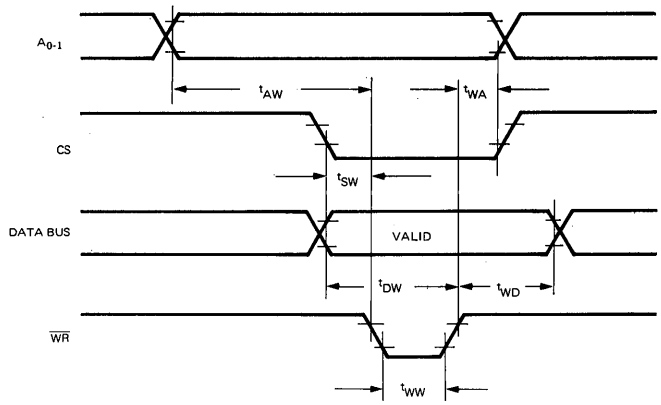


A.C. Testing Load Circuit

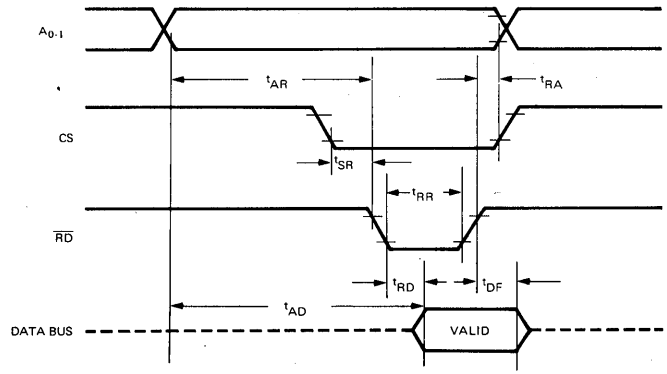


Waveforms

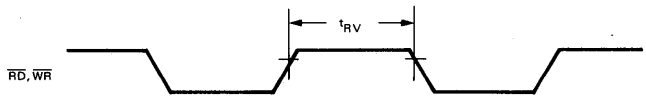
WRITE



READ



RECOVERY



CLOCK AND GATE

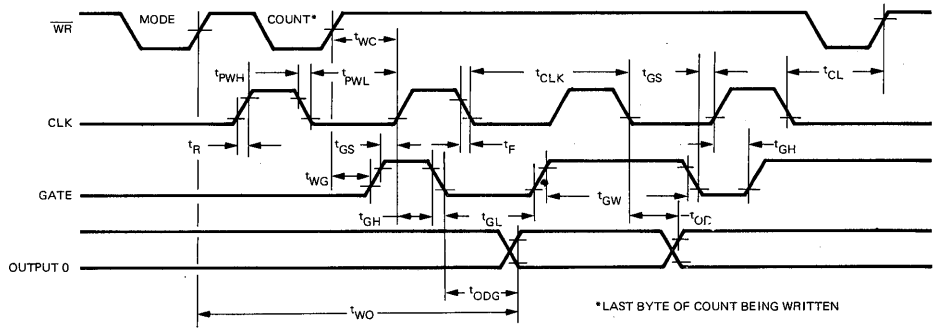


Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
D_7-D_0	1-8	I/O	Data: Bi-directional three state data bus lines, connected to system data bus.
CLK 0	9	I	Clock 0: Clock Input of Counter 0.
OUT 0	10	O	Output 0: Output of Counter 0.
GATE 0	11	I	Gate 0: Gate Input of Counter 0.
GND	12		Ground: Power supply connection.
V_{CC}	24		Power +5V power supply connection.
$\overline{WR}$	23	I	Write Control: This Input is low during CPU write operations.
$\overline{RD}$	22	I	Read Control: This input is low during CPU read operations.
$\overline{CS}$	21	I	Chip Select: A low on this input enables the 8254 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.
A_1, A_0	20-19	I	Address: Used to select one of the three Counters or the Control Word Register for read or write operations. Normally connected to the system address bus.
			A_1 A_0 Selects 0 0 Counter 0 0 1 Counter 1 1 0 Counter 2 1 1 Control Word Register
CLK2	18	I	Clock 2: Clock Input of Counter 2.
OUT 2	17	O	Out 2: Output of Counter 2.
GATE2	16	I	Gate 2: Gate Input of Counter 2.
CLK 1	15	I	Clock 1: Clock Input of Counter 1.
GATE1	14	I	Gate 1: Gate Input of Counter 1.
OUT 1	13	O	Out 1: Output of Counter 1.

Functional Description

General

The UM8254 is a programmable interval timer/counter designed for use with microcomputer systems. It is a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The UM8254 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the UM8254 to match his requirements and programs one of the counters for the desired delay. After the desired delay, the UM8254 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other counter/timer functions common to microcomputers which can be implemented with the UM8254 are:

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

Block Diagram

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the UM8254 to the system bus (see Figure 1).

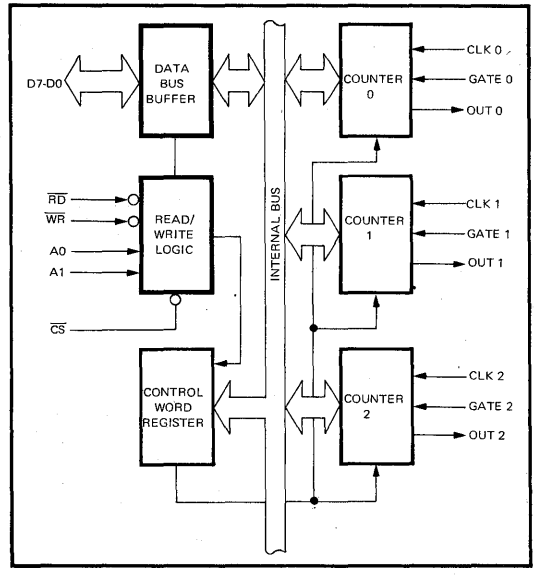


Figure 1. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the UM8254. A_1 and A_0 select one of the three counters or the Control Word Register to be read from/written into. A "low" on the $\overline{RD}$ input tells the Um8254 that the CPU is reading one of the counters. A "low" on the $\overline{WR}$ input tells the UM8254 that the CPU is writing either a Control Word or an initial count. Both $\overline{RD}$ and $\overline{WR}$ are qualified by $\overline{CS}$; $\overline{RD}$ and $\overline{WR}$ are ignored unless the UM8254 has been selected by holding $\overline{CS}$ low.

Control Word Register

The Control Word Register (see Figure 2) is selected by the Read/Write Logic when A_1, A_0 = 11. If the CPU then does a write operation to the UM8254, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

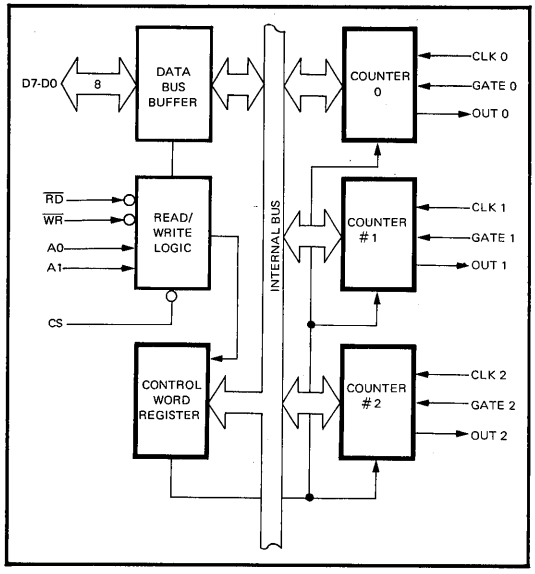


Figure 2. Block Diagram Showing Control Word Register and Counter Functions

Counter 0, Counter 1, Counter 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a single counter is shown in Figure 3.

The Counters are fully independent. Each Counter may operate in a different Mode.

The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates.

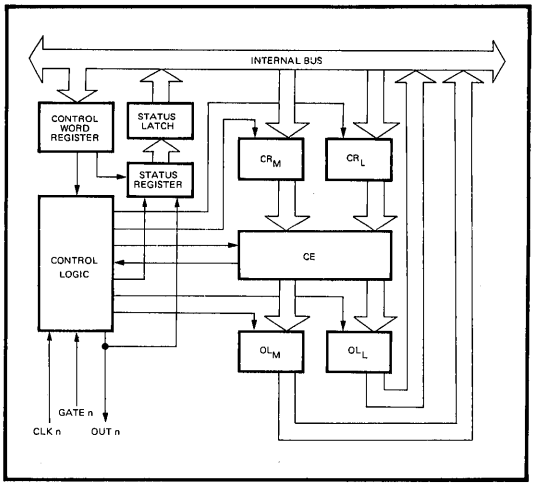


Figure 3. Internal Block Diagram of a Counter

The status register, shown in the Figure, when latched, contains the current contents of the Control Word Register and status of the output and null count flag. (See detailed explanation of the Read-Back command.)

The actual counter is labelled CE (for "Counting Element"). It is a 16-bit presettable synchronous down counter.

OL_M and OL_L are two 8-bit latches. OL stands for "Output Latch"; the subscripts M and L stand for "Most significant byte" and "Least significant byte" respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter Latch Command is sent to the UM8254, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CR_M and CR_L (for "Count Register"). Both are normally referred to as one until and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CR_M and CR_L are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

Write Operations

The programming procedure for the UM8254 is very flexible. Only two conventions need to be remembered:

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).

Since the Control Word Register and the three Counters have separate addresses (selected by the A_1, A_0 inputs), and each Control Word specifies the Counter it applies to (SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions above is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be affected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

	A_1	A_0		A_1	A_0
Control Word – Counter 0	1	1	Control Word – Counter 2	1	1
LSB of count – Counter 0	0	0	Control Word – Counter 1	1	1
MSB of count – Counter 0	0	0	Control Word – Counter 0	1	1
Control Word – Counter 1	1	1	LSB of count – Counter 2	1	0
LSB of count – Counter 1	0	1	MSB of count – Counter 2	1	0
MSB of count – Counter 1	0	1	LSB of count – Counter 1	0	1
Control Word – Counter 2	1	1	MSB of count – Counter 1	0	1
LSB of count – Counter 2	1	0	LSB of count – Counter 0	0	0
MSB of count – Counter 2	1	0	MSB of count – Counter 0	0	0

	A_1	A_0		A_1	A_0
Control Word – Counter 0	1	1	Control Word – Counter 1	1	1
Control Word – Counter 1	1	1	Control Word – Counter 0	1	1
Control Word – Counter 2	1	1	LSB of count – Counter 1	0	1
LSB of count – Counter 2	1	0	Control Word – Counter 2	1	1
LSB of count – Counter 1	0	1	LSB of count – Counter 0	0	0
LSB of count – Counter 0	0	0	MSB of count – Counter 1	0	1
MSB of count – Counter 0	0	0	LSB of count – Counter 2	1	1
MSB of count – Counter 1	0	1	MSB of count – Counter 0	0	0
MSB of count – Counter 2	1	0	MSB of count – Counter 2	1	0

Note: In all four examples, all counters are programmed to Read/Write two-byte counts. These are only four of many possible programming sequences.

Figure 4. A Few Possible Programming Sequences

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the UM8254.

There are three possible methods for reading the Counters. The first is through the Read-Back command. The second is a simple read operation of the Counter, which is selected with the A_1, A_0 inputs. The only requirement is that 1) the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic; or 2) the count must first be latched. Otherwise, the count may be in process of changing when it is read, giving an undefined result.

UM8254 System Interface

The UM8254 is a component of Microcomputer Systems and interfaces in the same manner as all other peripherals of the family. It is treated by the systems software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A_0, A_1 connect to the A_0, A_1 address bus signals of the CPU. The CS can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an UM8205 for larger systems.

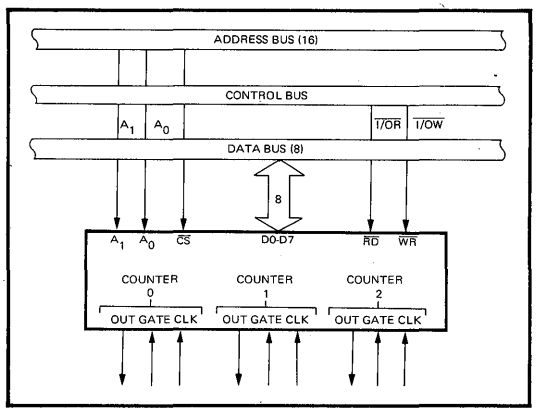


Figure 5. UM8254 System Interface

Operational Description

General

After power-up, the state of the UM8254 is undefined. The Mode, count value, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused counters need not be programmed.

Programming the UM8254

Counters are programmed by writing a Control Word and then an initial count.

All Control Words are written into the Control Word Register, which is selected when A_1, A_0 = 11. The Control Word itself specifies which Counter is being programmed.

By contrast, initial counts are written into the Counters, not the Control Word Register. The A_1, A_0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

Control Word Format

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

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.

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

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 UMC products.

Figure 6. Control Word Format

Counter Latch Command

The other method involves a special software command called the "Counter Latch Command" Like a Control Word, this command is written to the Control Word Register, which is selected when A_1, A_0 = 11. Also like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D5 and D4, distinguish this command from a Control Word.

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 UMC products.

Figure 7. Counter Latching Command Format

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). This allows reading the contents of the Counters "on the fly" without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until it is read. Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read or write or programming operations of other Counters may be inserted between them.

Another feature of the UM8254 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two byte counts, the following sequence is valid.

1. Read least significant byte.
2. Write new least significant byte.
3. Read most significant byte.
4. Write new most significant byte.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

Read-Back Command

The read-back command allows the user to check the count value, programmed Mode, and current state of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in Figure 8. The command applies to the counters selected by setting their corresponding bits D3, S2, D1 = 1.

A_0, A_1 = 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	$\overline{COUNT}$	$\overline{STATUS}$	CNT2	CNT1	CNT0	0

D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counter(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 8. Read-Back Command Format

The read-back command may be used to latch multiple counter output latches (OL) by setting the $\overline{COUNT}$ bit D5 = 0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). That counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting $\overline{STATUS}$ 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 9. 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.

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 (See Figure 7)

Figure 9. Status Byte

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the Mode of the counter and is described in the Mode Definitions, but until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown in Figure 10.

This Action:	Causes:
A. Write to the control word register.(1)	Null count = 1
B. Write to the count register (CR):(2)	Null count = 1
C. New count is loaded into CE (CR→CE):	Null count = 0

(1) Only the counter specified by the control word will have its null count set to 1. Null count bits of other counters are unaffected.

(2) If the counter is programmed for two byte counts (least significant byte then most significant byte) null count goes to 1 when the second byte is written.

Figure 10. Null Count Operation

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is the status of the counter at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both COUNT and STATUS bits D5, D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored. This is illustrated in Figure 11.

Command								Descriptions	Results
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	1	0	0	0	0	1	0	Read back count and status of Counter 0	Count and status latched for Counter 0
1	1	1	0	0	1	0	0	Read back status of Counter 1	Status latched for Counter 1
1	1	1	0	1	1	0	0	Read back status of Counters 2, 1	Status latched for Counter 2, but not Counter 1
1	1	0	1	1	0	0	0	Read back count of Counter 2	Count latched for Counter 2
1	1	0	0	0	1	0	0	Read back count and status of Counter 1	Count latched for Counter 1, but not status
1	1	1	0	0	0	1	0	Read back status of Counter 1	Command ignored, status already latched for Counter 1

Figure 11. Read-Back Command Example

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

CS	RD	WR	A_1	A_0	
0	1	0	0	0	**Write into Counter 0**
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3-State)
1	X	X	X	X	No-Operation (3-State)
0	1	1	X	X	No-Operation (3-State)

Figure 12. Read/Write Operations Summary

Mode Definitions

The following are defined for use in describing the operation of the UM8254.

CLK pulse: a rising edge, then a falling edge, in that order, of a Counter's CLK input.

trigger: a rising edge of a Counter's GATE input.

Counter loading: the transfer of a count from the CR to the CE (refer to the "Functional Description")

MODE 0: Interrupt on Terminal 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.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

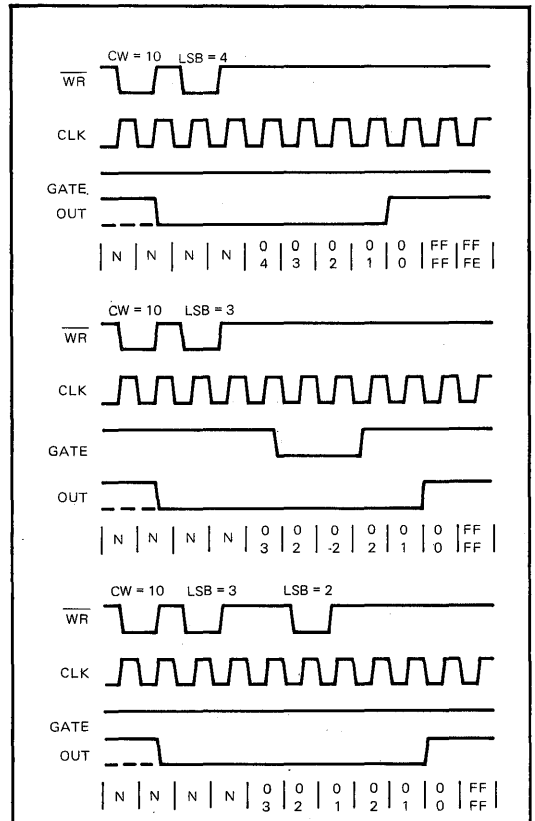
After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not go high untill N + 1 CLK pulses after the initial count is written.

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. If a two-byte count is written, the following happens:

1) Writing the first byte disables counting. OUT is set low immediately (no clock pulse required).
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the counting sequence to by synchronized by software. Again, OUT does not go high until N + 1 CLK pulses after the new count of N is written.

If an initial count is written while GATE = 0, it will still be loaded on the next CLK pulse. When GATE goes high, OUT will go high N CLK pulses later; no CLK pulse is needed to load the Counter as this has already been done.



Note: THE FOLLOWING CONVENTIONS APPLY TO ALL MODE TIMING DIAGRAMS:
1. COUNTERS ARE PROGRAMMED FOR BINARY (NOT BCD) COUNTING AND FOR READING/WRITING LEAST SIGNIFICANT BYTE(LSB) ONLY.
2. THE COUNTER IS ALWAYS SELECTED (CS ALWAYS LOW).
3. CW STANDS FOR "CONTROL WORD". CW=10 MEANS A CONTROL WORD OF 10, HEX IS WRITTEN TO THE COUNTER.
4. LSB STANDS FOR "LEAST SIGNIFICANT BYTE" OF COUNT.
5. NUMBERS BELOW DIAGRAMS ARE COUNT VALUES.
   THE LOWER NUMBER IS THE LEAST SIGNIFICANT BYTE.
   THE UPPER NUMBER IS THE MOST SIGNIFICANT BYTE. SINCE THE COUNTER IS PROGRAMMED TO READ/WRITE LSB ONLY, THE MOST SIGNIFICANT BYTE CANNOT BE READ.
   N STANDS FOR AN UNDEFINED COUNT.
   VERTICAL LINES SHOW TRANSITIONS BETWEEN COUNT VALUES.

Figure 13. Mode 0

MODE 1: Hardware Retriggerable One-Shot

Out will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the Counter is retriggered. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

MODE 2: Rate Generator

This Mode functions like a divide-by-N counter. It is typically used to generate a Real Time Clock interrupt. Out will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. OUT goes low N CLK Pulses after the initial count is written. This allows the Counter to be synchronized by software also.

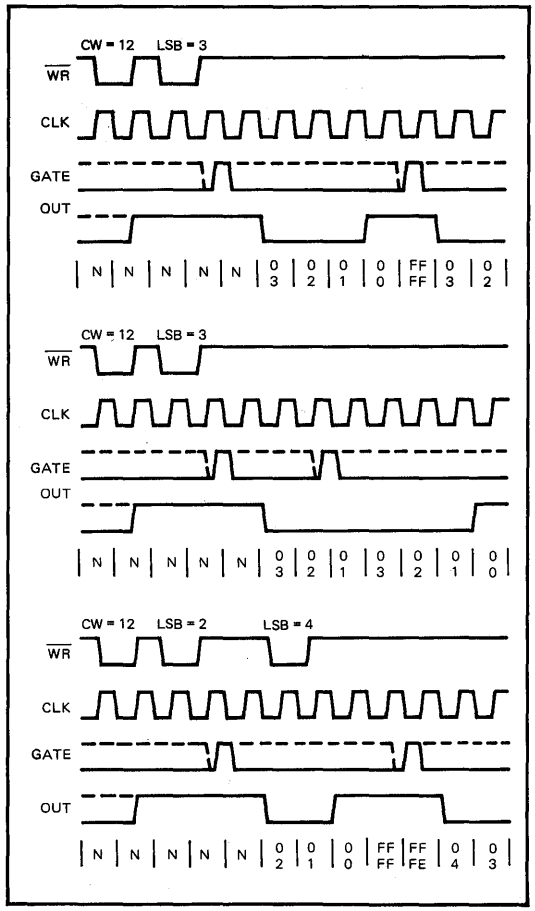
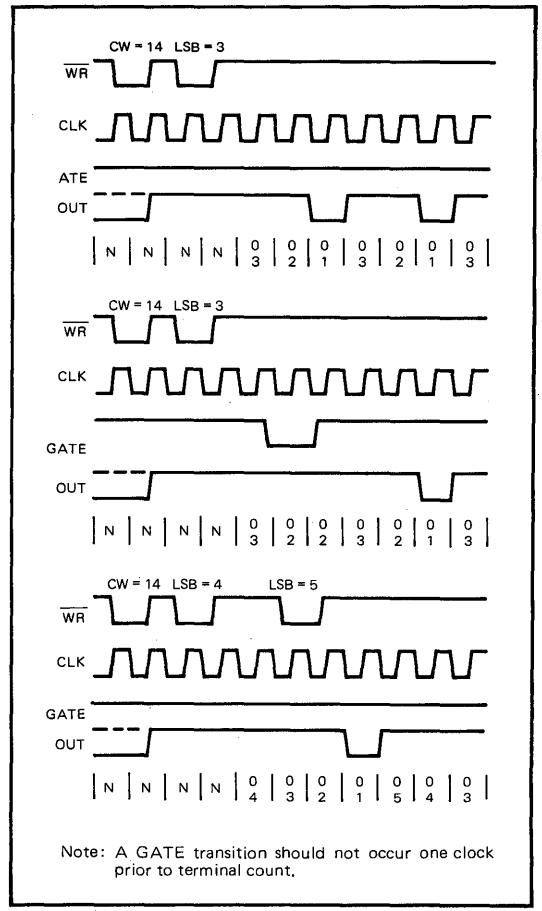


Figure 14. Mode 1



Note: A GATE transition should not occur one clock prior to terminal count.

Figure 15. Mode 2

Peripheral IC

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current counting cycle. In mode 2, a COUNT of 1 is illegal.

MODE 3: Square Wave Mode

Mode 3 is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

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. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

Mode 3 is implemented as follows:

Even counts: OUT is initially high. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires OUT changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

Odd counts: OUT is initially high. The initial count minus one (an even number) is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. One CLK pulse after the count expires, OUT goes low and the Counter is reloaded with the initial count minus one. Succeeding CLK pulses decrement the count by two. When the count expires, OUT goes high again and the Counter is reloaded with the initial count minus one. The above process is repeated indefinitely. So for odd counts, OUT will be high for (N+1)/2 counts and low for (N−1)/2 counts.

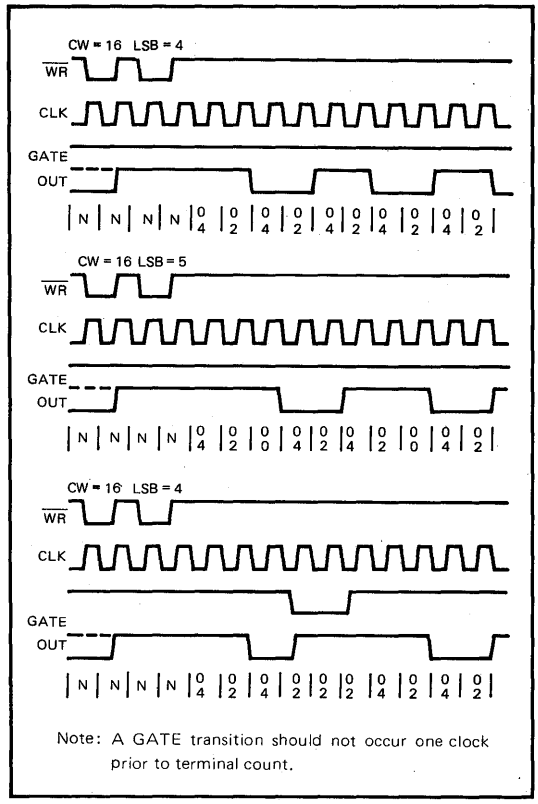


Figure 16. Mode 3

MODE 4: Software Triggered Strobe

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse and then go high again. The counting sequence is "triggered" by writing the initial count.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After writing a Control Word and initial count, the Counter 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 strobe low until N+1 CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte has no effect on counting.
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. OUT strobes low N+1 CLK pulses after the new count of N is written.

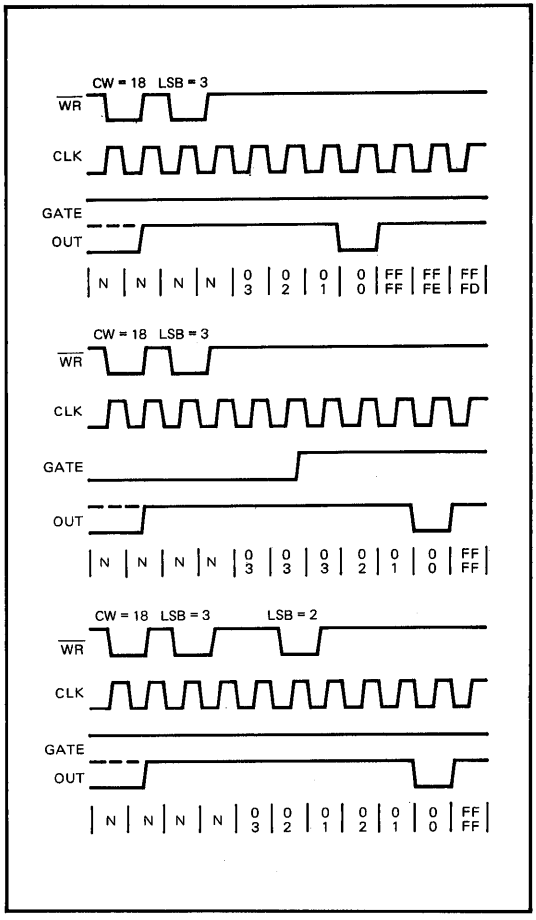


Figure 17. Mode 4

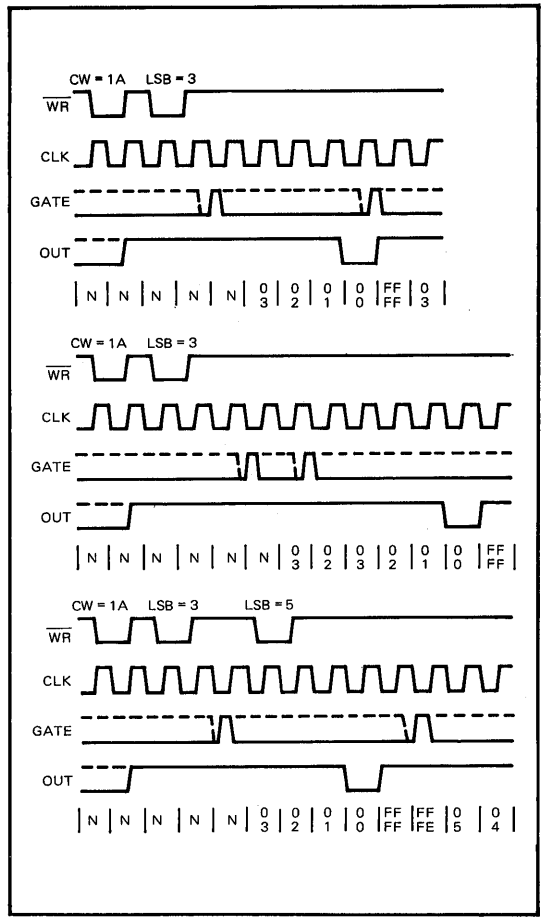


Figure 18. Mode 5

MODE 5: Hardware Triggered Strobe (Retriggerable)

OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again.

After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N+1 CLK pulses after a trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. OUT will not strobe low for N+1 CLK pulses after any trigger. GATE has no effect on OUT.

If a new count is written during counting, the current counting sequence will not be affected. If a trigger occurs after the new count is written but before the current count expires, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from there.

Signal Status Modes	Low or Going Low	Rising	High
0	Disables counting		Enables counting
1	—	1) Initiates counting 2) Resets output after next clock	—
2	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

Figure 19. Gate Pin Operations Summary

Mode	Min Count	Max Count
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0
5	1	0

Note: 0 is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

Figure 20. Minimum and Maximum Initial Counts

Operation Common to All Modes

Programming

When a Control Word is written to a Counter, all Control Logic is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

Gate

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3, and 4 the GATE input is level sensitive, and the logic level is sampled on the rising edge of CLK. In Modes 1, 2, 3, and 5 the GATE input is rising-edge sensitive. In these Modes, a rising edge of GATE (trigger) sets an edge-sensitive flip-flop in the Counter. This flip-flop is then sampled on the next rising edge of CLK; the flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs – a high logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level- sensitive. In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following WR of a new count value.

Counter

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The Counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5 the Counter "wraps around" to the highest count, either FFFF hex for binary counting or 9999 for BCD counting, and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.