# CA82C54

## PROGRAMMABLE INTERVAL TIMER

- **A high performance device featuring pin and functional compatibility with the industry standard 8254**
- **Supports 8086/88 and 80186/188 microprocessors**
- **High Speed: zero wait state 10 MHz and 8 MHz versions available**
- **Low power CMOS implementation**
- **TTL input/output compatibility**
- **Compatible with 8080/85, 8086/88, 80286/386 and 68000 μP families**
- **Fully static operation**
- **Three independent 16 bit counters**
- **Six programmable counter modes**
- **Status read-back command**
- **Binary or BCD counting**

The CA82C54 is a counter/timer device that includes complete pin and functional compatibility with the industry standard 8254. Designed for fast 10 MHz operation, it has three independently programmable 16 bit counters and six programmable counter modes. Counting can be performed in both binary and BCD formats.

The CA82C54 offers a very flexible, hardware solution to the generation of accurate time delays in microprocessor systems. A general purpose, multi-timing element, it can be used to implement event counters, elapsed time indicators, waveform generators plus a host of other functions.

The low power consumption of the CA82C54 makes it ideally suited to portable systems or those with low power standby modes.

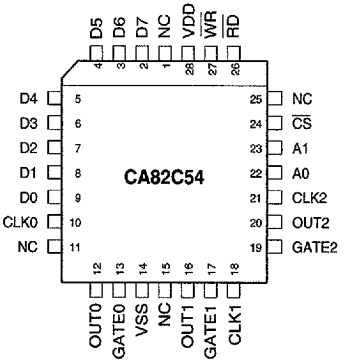



**Figure 2-1: PLCC Pin Configurations**

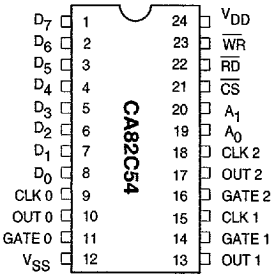



**Figure 2-2: PDIP Pin Configurations**

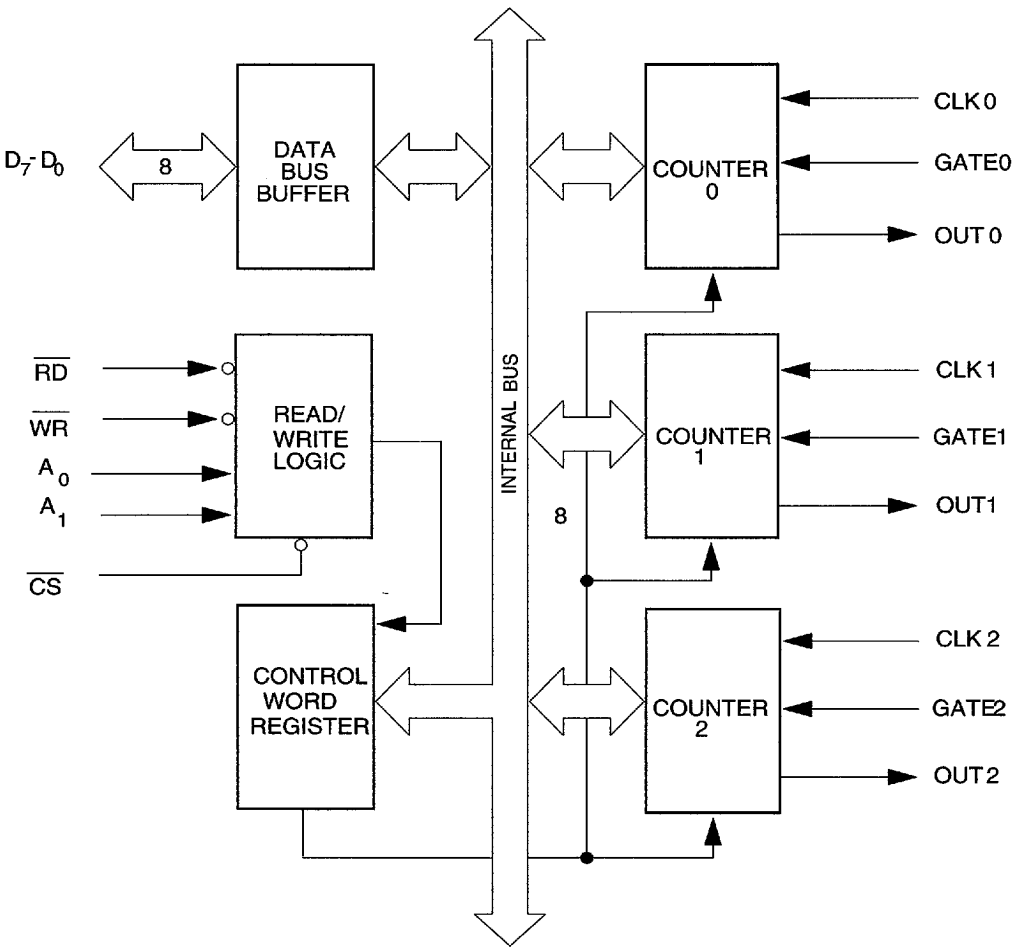



**Figure 2-3: CA82C54 Block Diagram**

**Table 2-1: Pin Descriptions**

| Symbol | Pins | | Type | Name and Function |
|---|---|---|---|---|
| | PLCC | PDIP | | |
| $A_0$, $A_1$ | 22, 23 | 19, 20 | I | **Address**: These two address pins are used to select the Control Word Register (for read or write operations), or one of the three Counters. They are normally connected to the system address bus.<br><br>$A_1$ $A_0$ **Selects:**<br>0 0 Counter 0<br>0 1 Counter 1<br>1 0 Counter 2<br>1 1 Control Word Register |
| CLK 0 | 10 | 9 | I | **Clock 0**: Clock input of Counter 0. |
| CLK 1 | 18 | 15 | I | **Clock 1**: Clock input of Counter 1. |
| CLK 2 | 21 | 18 | I | **Clock 2**: Clock input of Counter 2. |
| $\overline{CS}$ | 24 | 21 | I | **Chip Select**: Active LOW control signal to enable the CA82C54 to respond to $\overline{RD}$ and $\overline{WR}$ signals. If $\overline{CS}$ is not LOW, $\overline{RD}$ and $\overline{WR}$ are ignored. |
| $D_7$ - $D_0$ | 2 - 9 | 1 - 8 | I/O | **Data**: Bi-directional 3-state data bus lines, connected to system data bus. |
| GATE 0 | 13 | 11 | I | **Gate 0**: Gate input of Counter 0. |
| GATE 1 | 17 | 14 | I | **Gate 1**: Gate input of Counter 1. |
| GATE 2 | 19 | 16 | I | **Gate 2**: Gate input of Counter 2. |
| OUT 0 | 12 | 10 | O | **Output 0**: Output of Counter 0. |
| OUT 1 | 16 | 13 | O | **Output 1**: Output of Counter 1. |
| OUT 2 | 20 | 17 | O | **Output 2**: Output of Counter 2. |
| $\overline{RD}$ | 26 | 22 | I | **Read Control**: Active LOW control signal used to enable the CA82C54 for read operations by the CPU. |
| VDD | 28 | 24 | - | **Power**: 5 v ± 10% DC Supply |
| VSS | 14 | 12 | - | **Ground**: 0 v |
| $\overline{WR}$ | 27 | 23 | I | **Write Control**: Active LOW control signal used to enable the CA82C54 to be written to by the CPU. |

## FUNCTIONAL DESCRIPTION

The CA82C54 is a versatile programmable interval timer/counter designed for use in high speed 8, 16 and 32-bit microprocessor systems. It provides a means of generating accurate time delays in hardware that is fully software configurable. It can be treated as an array of I/O ports, with minimal software overhead.

The internal structure of the CA82C54 is illustrated in the block diagram of Figure 2-3. Major functional blocks include a data bus buffer, read/write logic, control word register, and three programmable counters.

### Data Bus Buffer Block

The 8-bit, 3-state data bus buffer provides controllable, bi-directional interface between the CA82C54 and the microprocessor system bus.

### Read/Write Logic Block

The read/write logic block generates internal control signals for the different functional blocks using address and control information obtained from the system. The active LOW signals; $\overline{CS}$, $\overline{RD}$ and $\overline{WR}$ are used to select the CA82C54 for operation, read a counter, and write to a counter (or the control word register) respectively. $\overline{CS}$ must be LOW for $\overline{RD}$ or $\overline{WR}$ to be recognized. Note that $\overline{RD}$ and $\overline{WR}$ must NOT be active at the same time.

The inputs $A_0$ and $A_1$ are used to select the control word register, or one of the three counters that is to be written to or read from (see Table 2-1). $A_0$ and $A_1$ connect directly to the corresponding signals of the microprocessor address bus, while $\overline{CS}$ is derived from the address bus using either a linear select method, or an address decoder device.

### Control Word Register

The control word register is a write only register that is selected by the read/write logic block when $A_0$ and $A_1 = 1$. When $\overline{CS}$ and $\overline{WR}$ are LOW, data is written into the CA82C54 control word register from the CPU via the data bus buffer. Control word data is interpreted as a number of different commands which are used to program the various device functions. For example, status information is available with the Read-Back Command. These are discussed further in the section on programming.

### Counter Blocks

The CA82C54 contains three identical, independent counter blocks. Each counter provides the same functions, but can be programmed to operate in different modes relative to each other. A typical CA82C54 counter is illustrated in Figure 2-4, and contains the following functional elements: control logic, counter, output latches, count registers and status register.

The Control Logic provides the interface between the counter proper, the program instructions contained in the control word register and the external signals CLK n, GATE n and OUT n. It also keeps the status register information current, controls the access of OL and CR to the internal data bus, and the loading of CE from the CR registers.

The Counter proper (shown in the Figure 2-4 as CE, for counting element) is a 16-bit presettable synchronous down counter.

The Output Latches (shown as $OL_M$ and $OL_L$) provide a mechanism whereby the CPU can read the current contents of the CE. These two 8-bit latches (M for most significant byte and L for least significant byte) together form a 16-bit latch capable of holding the complete contents of the CE. Note that this arrangement is also convenient for communicating 16-bit values over the 8-bit internal data bus.

During normal operation, the contents of OL track with the contents of CE. When a Counter Latch Command is issued by the CPU to a particular counter, its OL latches the current value of CE so that it can be read by the CPU (the CE cannot be read directly). OL then returns to tracking with CE. Note that only one latch ($OL_M$ followed by $OL_L$) at a time is enabled by the counter's control logic.

The Count Registers (shown as $CR_M$ and $CR_L$) behave as input latches to the CE, and provide a mechanism whereby the initial count value can be downloaded from the CPU to the CE. Similar in operation to OL, CR is controlled by the counter control logic. When a two byte initial count is to be downloaded, it is transferred one byte at a time across the internal CA82C54 data bus to the appropriate register ($CR_M$ if the most significant byte, $CR_L$ otherwise). CE is loaded by transferring both bytes simultaneously from CR. Note that CR is the interface between CE and the data bus, since CE cannot be accessed directly.

Both $CR_M$ and $CR_L$ are cleared automatically when the counter is programmed and a new initial count is to be written. Thus, regardless of the counter's previous programming, both CR bytes will be initialized to a known zero state. This is important in the case where one byte counts are programmed (either most significant or least significant byte), so that the unused byte is always zero, and won't corrupt the initial count value loaded into CE.

The Status Register and Latch is used to hold the current contents of the control word register and the status of the output and null count flag (see section on Programming). The contents of the status register must be latched to become available to the data bus, where they can be read by the CPU.

Note that the Control Word Register is also shown in the Counter block diagram. While not a part of the counter proper, its contents determine the functional operation of the counter, including mode selections programmed.

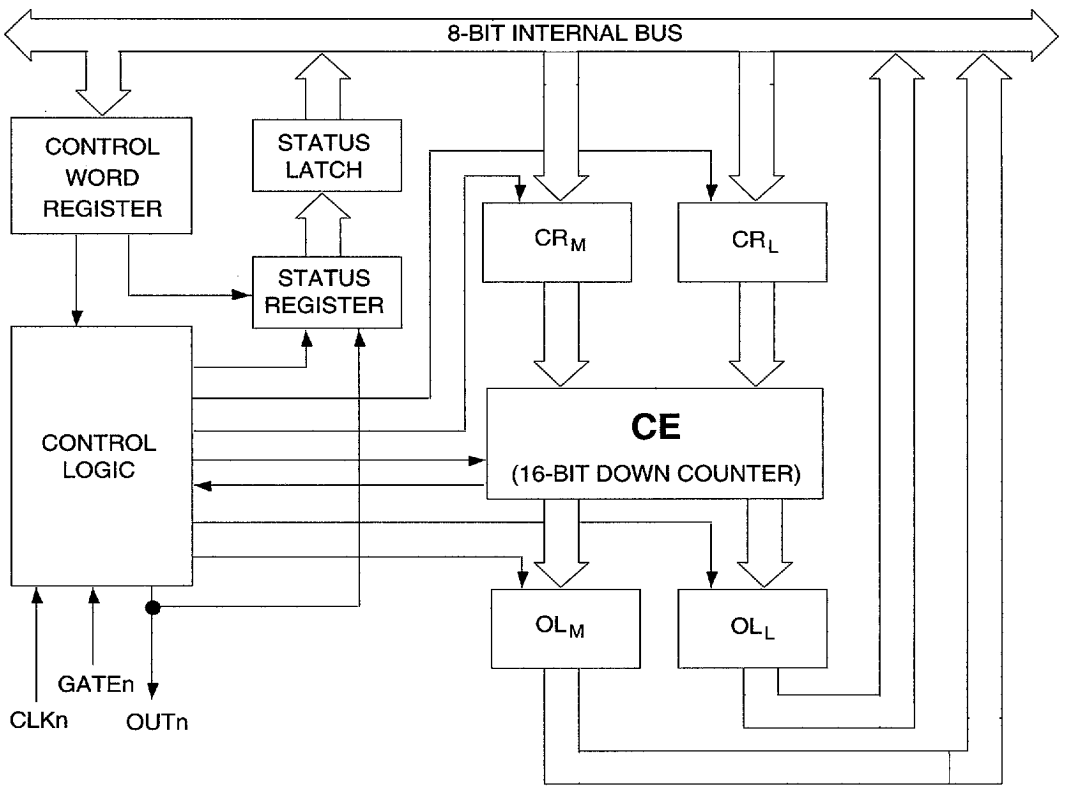



**Figure 2-4: Block Diagram of a Counter**

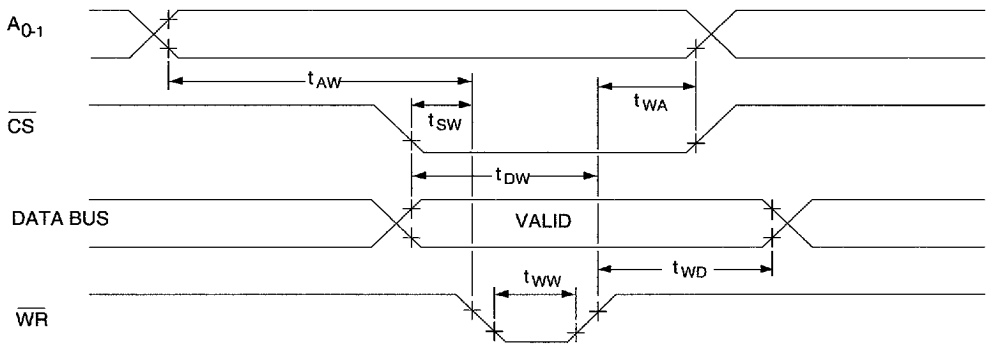
**Table 2-2: AC Characteristics ($T_A$ = -40° to +85°C, $V_{DD}$ = 5V ± 10%, $V_{SS}$ = 0V) Bus Parameters[1]**

| Symbol | Parameter | Test Condition | Limits (8 MHz) | | Limits (10 MHz) | | Units |
|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | |
| $t_{AD}$ | Data delay from address | | - | 220 | - | 185 | ns |
| $t_{AR}$ | Address stable before $\overline{RD}$ ↓ | | 45 | - | 30 | - | ns |
| $t_{AW}$ | Address stable before $\overline{WR}$ ↓ | | 0 | - | 0 | - | ns |
| $t_{CL}$ | CLK setup for count latch | | -40 | 45 | -40 | 40 | ns |
| $t_{CLK}$ | Clock period | | 125 | DC | 100 | DC | ns |
| $t_{DF}$ | $\overline{RD}$ ↑ to data floating | | 5 | 90 | 5 | 65 | ns |
| $t_{DW}$ | Data setup time before $\overline{WR}$ ↑ | | 120 | - | 95 | - | ns |
| $t_F$ | Clock fall time | | - | 25 | - | 25 | ns |
| $t_{GH}$ | Gate hold time after CLK ↑ | Note 2 | 50 | - | 50 | - | ns |
| $t_{GL}$ | Gate width low | | 50 | - | 50 | - | ns |
| $t_{GS}$ | Gate setup time to CLK ↑ | | 50 | - | 40 | - | ns |
| $t_{GW}$ | Gate width high | | 50 | - | 50 | - | ns |
| $t_{OD}$ | Output delay from CLK ↓ | | - | 150 | - | 100 | ns |
| $t_{ODG}$ | Output delay from Gate ↓ | | - | 120 | - | 100 | ns |
| $t_{PWH}$ | High pulse width | Note 3 | 60 | - | 30 | - | ns |
| $t_{PWL}$ | Low pulse width | Note 3 | 60 | - | 50 | - | ns |
| $t_R$ | Clock rise time | | - | 25 | - | 25 | ns |
| $t_{RA}$ | Address hold time after $\overline{RD}$ ↑ | | 0 | - | 0 | - | ns |
| $t_{RD}$ | Data delay from $\overline{RD}$ ↓ | | - | 120 | - | 85 | ns |
| $t_{RR}$ | $\overline{RD}$ pulse width | | 150 | - | 95 | - | ns |
| $t_{RV}$ | Command recovery time | | 200 | - | 165 | - | ns |
| $t_{SR}$ | CS stable before $\overline{RD}$ ↓ | | 0 | - | 0 | - | ns |
| $t_{SW}$ | CS stable before $\overline{WR}$ ↓ | | 0 | - | 0 | - | ns |
| $t_{WA}$ | Address hold time $\overline{WR}$ ↑ | | 0 | - | 0 | - | ns |
| $t_{WC}$ | CLK delay for loading | | 0 | 55 | 0 | 55 | ns |
| $t_{WD}$ | Data hold time after $\overline{WR}$ ↑ | | 0 | - | 0 | - | ns |
| $t_{WG}$ | Gate delay for sampling | | -5 | 50 | -5 | 40 | ns |
| $t_{WO}$ | OUT delay from Mode Write | | - | 260 | - | 240 | ns |
| $t_{WW}$ | $\overline{WR}$ pulse width | | 150 | - | 95 | - | ns |

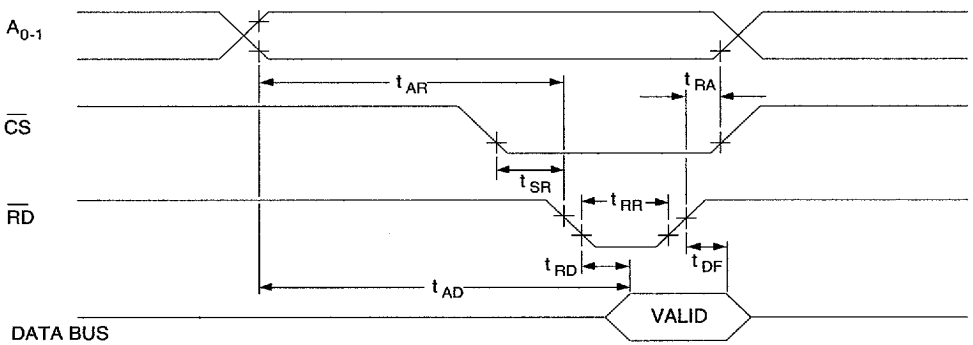
Notes: 1. AC timings measured at $V_{OH}$ = 2.0 V, $V_{OL}$ = 0.8 V
2. In Modes 1 and 5 triggers are sampled on each rising clock edge. A second trigger within 120 ns of the rising clock edge may not be detected, (70 ns for CA82C54-10).
3. Low-going glitches that violate $t_{PWH}$, $t_{PWL}$ may cause errors requiring counter reprogramming.

**Figure 2-5: Timing Diagrams**

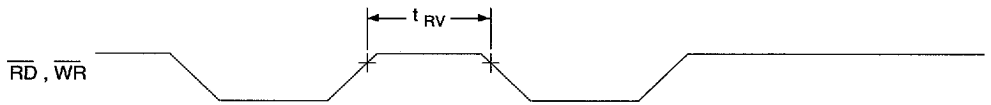
**a) Write Timing**



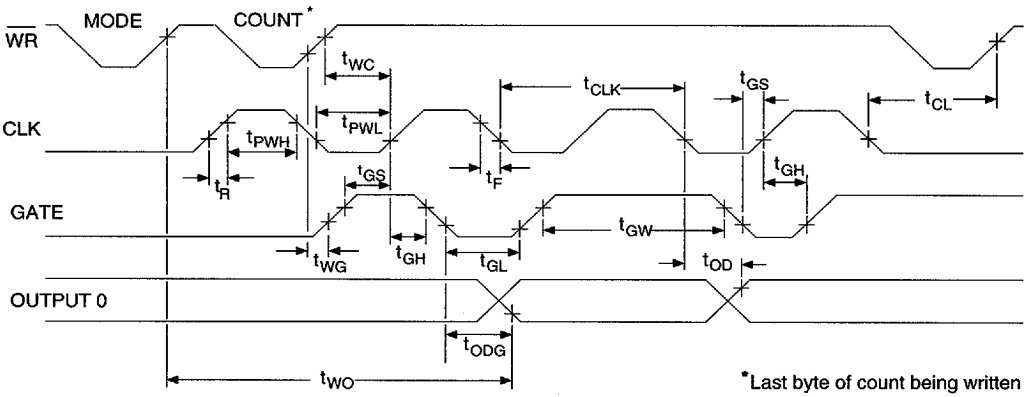

**b) Read Timing**

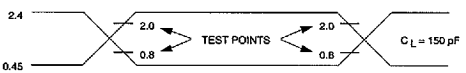


**c) Recovery Timing**



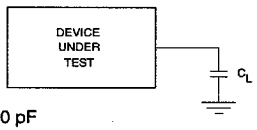

**d) Clock and Gate Timing**




All input signals must switch between 0.45V and 2.4V.
All timing measurements are made at 0.8V and 2.0V

**Figure 2-6: AC Testing I/O Waveform**




$C_L$ = 150 pF
$C_L$ includes jig capacitance

**Figure 2-7: AC Testing Loading Circuit**

**Table 2-3: DC Characteristics ($T_A$ = -40°C to +85°C, $V_{DD}$ = 5V ± 10%, $V_{SS}$ = 0V)**

| Symbol | Parameter | Test Conditions | Limits | | Units |
|---|---|---|---|---|---|
| | | | Min | Max | |
| $I_{DD}$ | $V_{DD}$ Supply Current | CLK Freq = 5 MHz, CA82C54-5<br>CLK Freq = 8 MHz, CA82C54-8<br>CLK Freq = 10 MHz, CA82C54-10 | - | 20 | mA |
| $I_{IL}$ | Input Load Current | $V_{IN} = V_{DD}$ or $V_{SS}$ | -1.0 | 1.0 | µA |
| $I_{OFL}$ | Output Float Leakage | $V_{OUT} = V_{DD}$ to 0.45V | -10 | 10 | µA |
| $V_{IH}$ | Input High Voltage | | 2.0 | $V_{DD}$ + 0.3V | V |
| $V_{IL}$ | Input Low Voltage | | -0.3 | 0.8 | V |
| $V_{OH}$ | Output High Voltage | $I_{OH}$ = -400 µA | $V_{DD}$ - 0.4V | - | V |
| | | $I_{OH}$ = -2.5 mA | 3.0 | - | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ = 2.5 mA | - | 0.4 | V |

**Table 2-4: Capacitance ($T_A$ = 25°C, $V_{DD}$ = $V_{SS}$ = 0V, $V_{IN}$ = +5 V or $V_{SS}$)**

| Symbol | Parameter | Test Conditions | Min | Max | Units |
|---|---|---|---|---|---|
| $C_{I/O}$ | I/O Capacitance | FREQ = 1 MHz<br>Unmeasured pins returned to $V_{SS}$ | - | 15 | pF |
| $C_{IN}$ | Input Capacitance | | - | 10 | pF |
| $C_{OUT}$ | Output Capacitance | | - | 10 | pF |

**Table 2-5: Recommended Operating Conditions**

| | | |
|---|---|---|
| DC Supply Voltage | | +4.5 V to +5.5 V |
| Operating Temperature Range | Commercial | 0°C to +70°C |
| | Industrial | -40°C to +85°C |

**Table 2-6: Absolute Maximum Ratings**

| | |
|---|---|
| DC Supply Voltage | -.03 to +7.0 V |
| Input, Output or I/O Voltage Applied | $V_{SS}$ - 0.3 V to $V_{DD}$ + 0.3 V |
| Storage Temperature Range | -55°C to +150°C |
| Maximum Package Power Dissipation | 1 W |
| Input Pin Current (@ 25°C) | -10.0 mA to +10.0 mA |
| Lead Temperature (soldering: 10 sec.) | 300°C |

Stresses beyond those listed above may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

## PROGRAMMING

When installed in microprocessor systems the CA82C54 Programmable Interval Timer appears to the system software as an array of peripheral I/O ports, namely: three counters, and a control register for MODE programming.

After power-up, the state of the CA82C54 is undefined. Thus, the mode, the count value and the output of all the counters are undefined. The subsequent operation of each counter is determined when it is programmed, and each counter must be programmed before it can be used. Unused counters need not be programmed.

Counters are programmed by writing a Control Word and then an initial count value for the counter in question. 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. It is illustrated in Figure 2-8.

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

A number of commands are available to the user which allow access to the programmable features of the CA82C54. These are described in detail below.

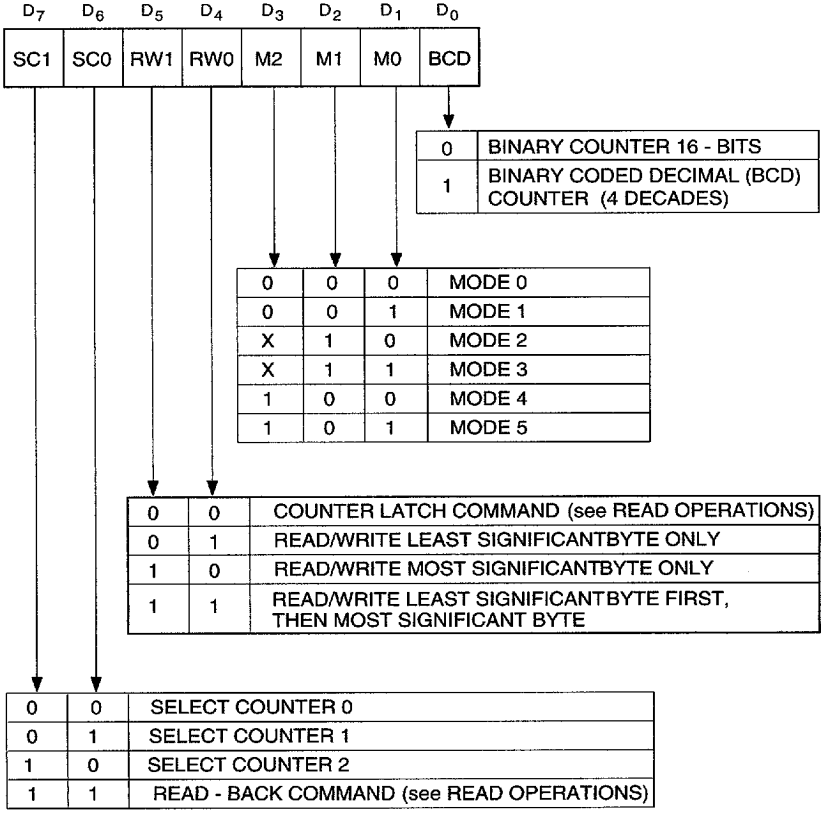



**Figure 2-8: Control Word Format**

### Write Operations

Programming for the CA82C54 is very flexible, and requires that only two conventions be followed:

- For each Counter, the Control Word must be written before the initial count is written.
- 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 2-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 the same Counter. Otherwise, the Counter will be loaded with an incorrect count.

**Table 2-7: Sample Programming Sequences**

| Example | Step | Operation | $A_1$ | $A_0$ |
|---|---|---|---|---|
| 1 | 1 | Control Word-Counter 0 | 1 | 1 |
| | 2 | LSB of count-Counter 0 | 0 | 0 |
| | 3 | MSB of count-Counter 0 | 0 | 0 |
| | 4 | Control Word-Counter 1 | 1 | 1 |
| | 5 | LSB of count-Counter 1 | 0 | 1 |
| | 6 | MSB of count-Counter 1 | 0 | 1 |
| | 7 | Control Word-Counter 2 | 1 | 1 |
| | 8 | LSB of count-Counter 2 | 1 | 0 |
| | 9 | MSB of count-Counter 2 | 1 | 0 |
| 2 | 1 | Control Word-Counter 1 | 1 | 1 |
| | 2 | Control Word-Counter 0 | 1 | 1 |
| | 3 | LSB of count-Counter 1 | 0 | 1 |
| | 4 | Control Word-Counter 2 | 1 | 1 |
| | 5 | LSB of count-Counter 0 | 0 | 0 |
| | 6 | MSB of count-Counter 1 | 0 | 1 |
| | 7 | LSB of count-Counter 2 | 1 | 0 |
| | 8 | MSB of count-Counter 0 | 0 | 0 |
| | 9 | MSB of count-Counter 2 | 1 | 0 |

Note: In both examples, all counters are programmed to read/write 2-byte counts. These are two of many programming sequences.

### Read Operations

It is often desirable to read the value of a counter without disturbing the count in progress, a procedure easily accomplished in the CA82C54.

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 the CLK input of the selected counter must be inhibited by using either the GATE input or external logic; or the count must first be latched. Otherwise, the count may be in process of changing when it is read, giving an undefined result. The third method involves a special software command called the Counter Latch Command, described in more detail below.

### Counter Latch Command

The Counter Latch Command, like a Control Word, 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, $D_5$ and $D_4$, distinguish this command from a Control Word.

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.

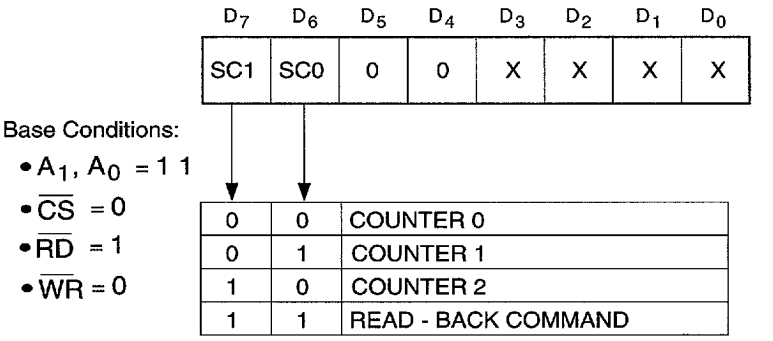
With either method, the count must be read according to the programmed format; specifically, if the counter is

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 CA82C54 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 and write two-byte counts, the following precaution applies: a program must not transfer control (between reading first and second byte) to another routine which also reads from that same counter. Otherwise, an incorrect count will be read.




SC1, SC0 - specify counter to be latched
$D_5$, $D_4$ - 00 designates Counter Latch Command
X - don't care

Note: Don't care bits (X) should be 0 to insure compatibility with future Newbridge Microsystem products

**Figure 2-9: Counter Latch Command Format**

### Read-Back Command

The Read-Back Command allows the user to check the count value, the programmed Mode and the 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 2-10. The command applies to the counters selected by setting their corresponding bits $D_3$, $D_2$, $D_1 = 1$.

The Read-Back Command may be used to latch multiple counter output latches (OL) by setting the COUNT bit $D_5 = 0$ 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. That is, the count which will be read is the count at the time the first Read-Back Command was issued.

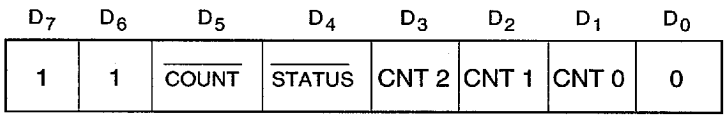
The Read-Back Command may also be used to latch status information of selected counter(s) by setting bit $D_4 = 0$ (STATUS). Status must be latched to be read; as counter status is obtained by a read from that counter.

The counter status format is shown in Figure 2-11. Bits $D_5$ through $D_0$ contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUTPUT bit $D_7$ contains the current state of the OUT pin. This allows the user to monitor counter output via software, thus eliminating some hardware from a system.

NULL COUNT bit $D_6$ 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. 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 Table 2-8.

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored. That is, 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 $D_5$, $D_4 = 0$. 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 Table 2-9.




Base Conditions:

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

$D_5$: 0 = Latch count of selected counter(s)
$D_4$: 0 = Latch status of selected counter(s)
$D_3$: 1 = Selected Counter 2
$D_2$: 1 = Selected Counter 1
$D_1$: 1 = Selected Counter 0

**Figure 2-10: Read-Back Command Format**

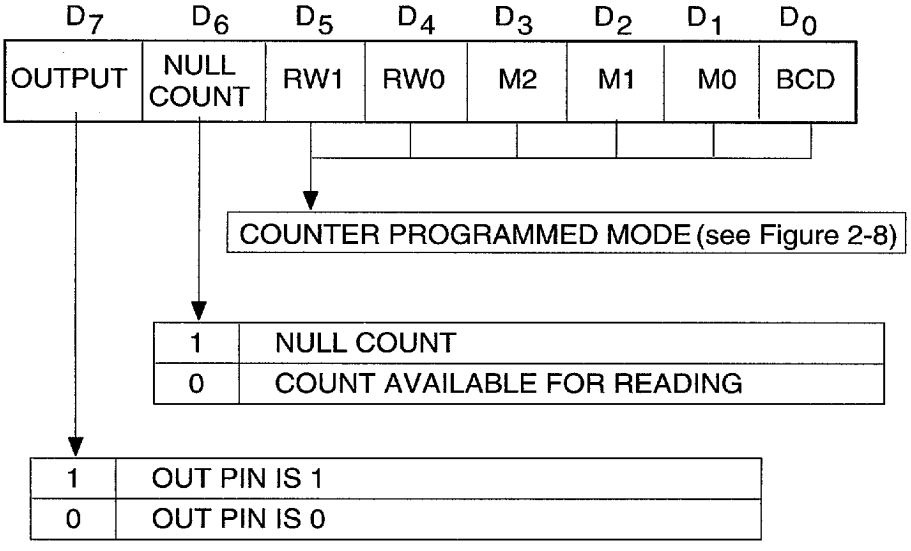



**Figure 2-11: Status Byte**

**Table 2-8: Null Count Operation**

| | 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 |

Notes: 1. Only the counter specified by the control word will have its null count set = 1. Null counts 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 second byte is written.

**Table 2-9: Read-Back Command Example**

| Command Description | Result | Command Word | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
| Read back count and status of Counter 0 | Count and status latched for Counter 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| Read back status of Counter 1 | Status latched for Counter 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| Read back status of Counters 2, 1 | Status latched for Counter 2 only | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| Read back count of Counter 2 | Count latched for Counter 2 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| Read back count and status of Counter 1 | Count latched for Counter 1, but not status | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| Read back status of Counter 1 | Command ignored, status already latched for Counter 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

## OPERATIONAL DESCRIPTION

The following operations are common to all modes.

**Control Word:** When a Control Word is written to a Counter, all Control Logic is Reset, and OUT is initialized to a known state. No CLK pulses are needed.

**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 sampled on the next rising edge of CLK, then is immediately reset. In this way, a trigger will be detected no matter when it occurs and a high logic level does not have to be maintained until the next CLK pulse. A summary is given in Table 2-12.

Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive. If a CLK source other than the system clock is used in Modes 2 and 3, GATE should be pulsed immediately after the $\overline{WR}$ for a new count value.

**Table 2-10: Minimum and Maximum Initial Counts**

| Mode | Minimum Count | Maximum Count* |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 0 |
| 2 | 2 | 0 |
| 3 | 2 | 0 |
| 4 | 1 | 0 |
| 5 | 1 | 0 |

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

**Counter:** New counts are loaded, with the largest possible initial count being zero (0), equivalent to $2^{16}$ for binary counting and $10^4$ for BCD counting, as in Table 2-10.

Counters decremented on the falling edge of CLK do not stop when they reach zero. In Modes 0, 1, 4, and 5 the Counters wrap around to the highest count (either FFFF hex for binary counting or 9999 for BCD counting), then continue counting. Modes 2 and 3 are periodic; the Counters reload themselves with the initial count, then continue counting from there.

If both the count and status registers of a counter are latched, the first read operation of that counter will return the latched status, regardless of which was latched first. The next one or two reads (the counter can be programmed for one or two type counts) will return the latched count. Subsequent reads will return an unlatched count. Read and Write operations are summarized in Table 2-11.

**Table 2-11: Read/Write Operations Summary**

| $\overline{CS}$ | $\overline{RD}$ | $\overline{WR}$ | $A_1$ | $A_0$ | |
|---|---|---|---|---|---|
| 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) |

**Table 2-12: Gate Pin Operations Summary**

| Signal Status Modes | Low, or Going Low | Rising | High |
|---|---|---|---|
| 0 | •Disables counting | - | •Enables counting |
| 1 | - | •Initiates counting<br>•Resets output after next clock | - |
| 2 | •Disables counting<br>•Sets output immediately high | •Initiates counting | •Enables counting |
| 3 | •Disables counting<br>•Sets output immediately high | •Initiates counting | •Enables counting |
| 4 | •Disables counting | - | •Enables counting |
| 5 | - | •Initiates counting | - |

## MODE DEFINITIONS

The following terms are useful in describing the operation of the CA82C54.

- CLK pulse: A rising edge, followed by a falling edge, of a Counter's CLK input.
- Trigger: A rising edge of a Counter's GATE input.
- Counter loading: Transfer of a count from the CR to the CE (see Functional Description)

### Mode 0: Interrupt on Terminal Count

Mode 0 is typically used for event counting. After the Control Word is written, OUT is set low, and remains 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.

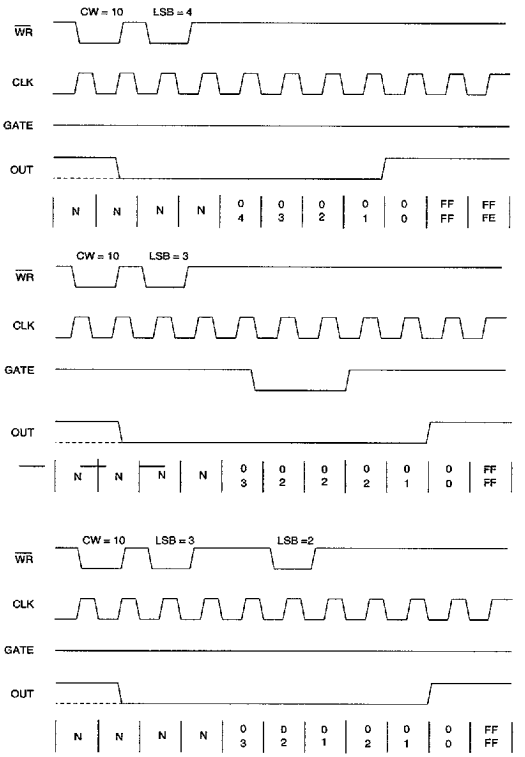
After a Control Word and initial count are written to a Counter, the initial count is loaded on the next CLK pulse. Since this CLK pulse does not decrement the count, OUT does not go high until N + 1 CLK pulses after the initial count is written (where N is the initial count value).

If a new count is written to the Counter, it is loaded on the next CLK pulse and counting continues 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 be 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. A CLK pulse is not required to load the Counter as this has already been done.




Notes: These 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 is the 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 2-12: Mode 0 Timing**

### Mode 1: Hardware Retriggerable One-Shot

OUT is initially high. To begin the one-shot pulse, OUT goes low on the CLK pulse following a trigger and remains low until the Counter reaches zero. OUT then goes high and remains high until the CLK pulse following the next trigger.

After a Control Word and initial count have been written, the Counter is armed. A trigger causes the Counter to be loaded and OUT to be set low on the next CLK pulse, starting the one-shot pulse. An initial count of N results in a one-shot pulse N CLK cycles long. Since the one-shot is retriggerable, OUT remains 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 this case, the new count is loaded into the Counter and the one-shot pulse continues for the duration of the count.

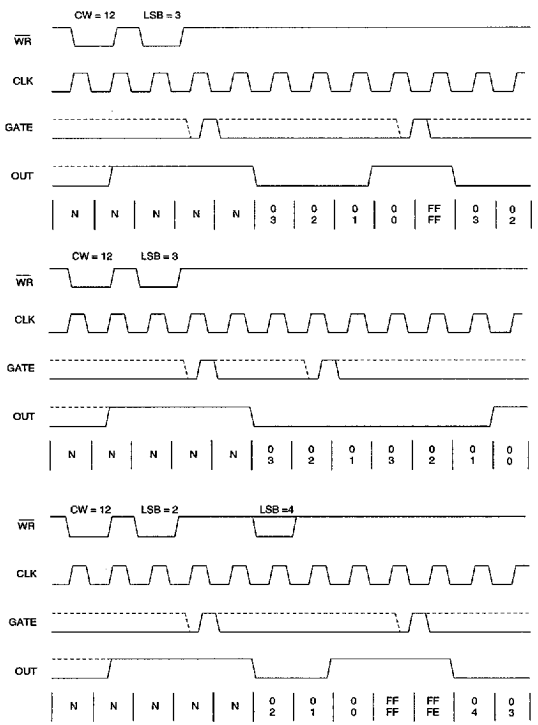



**Figure 2-13: Mode 1 Timing**

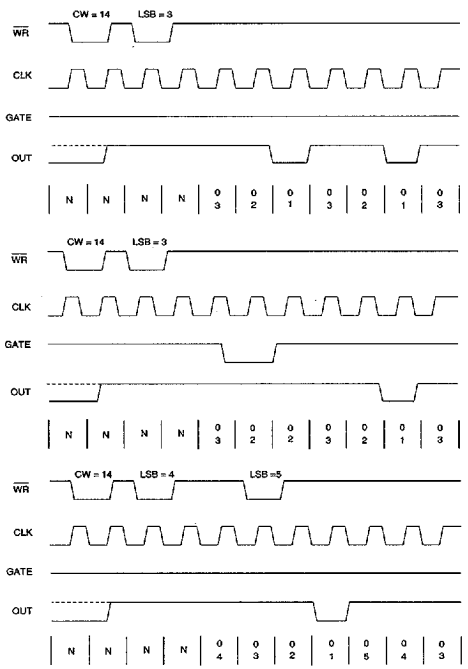
### Mode 2: Rate Generator

This mode functions like a divide-by-N counter and is typically used for generating Real Time Clock Interrupts. OUT is initially high. When the initial count has decremented to 1, OUT goes low for one CLK pulse, then high again. The Counter reloads the initial count and the process is

repeated. Mode 2 is periodic, with the same sequence 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 initial count into the Counter 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 a Control Word and initial count have been written, the Counter is loaded on the next CLK pulse. OUT goes low NCLK pulses after the initial count is written, which allows the Counter to be synchronized by software.




Note: A GATE transition should not occur one clock cycle prior to reaching the terminal count (TC).

**Figure 2-14: Mode 2 Timing**

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after a new count is written, but before the end of the current period, the Counter is loaded with the new count on the next CLK pulse and counting continues from the new count. Else, the new count is 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, and is similar to Mode 2 except for the duty cycle of OUT. OUT is initially high. When half the initial count has expired, OUT goes low for the remainder of the count. Mode 3 is also periodic, with the sequence above 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 needed). 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.

The Counter is loaded on the next CLK pulse after a Control Word and initial count have been written. This allows the Counter to be synchronized by software.

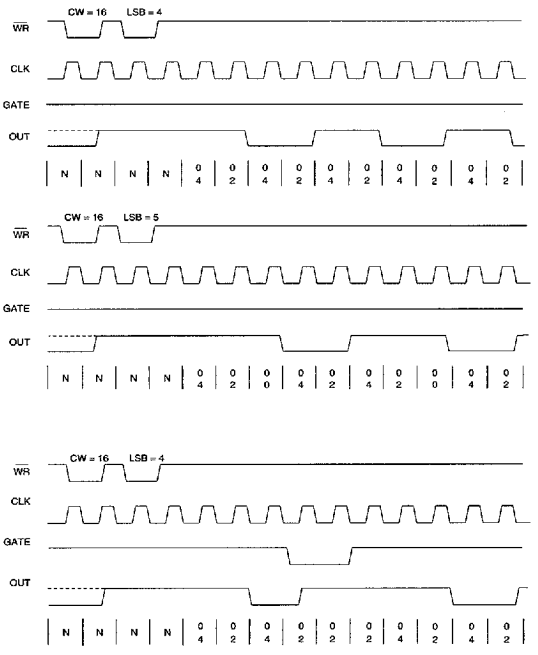
Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter is loaded with the new count on the next CLK pulse and counting continues from the new count.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

Mode 3 is implemented as follows according to whether the initial count value is even or odd:

Even counts: OUT is initially high. The initial count is loaded on one CLK pulse and then decremented by two on succeeding CLK pulses. When the count expires, OUT goes low 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 (to give an even number) is loaded on one CLK pulse and then 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 is high for (N + 1)/2 counts and low for (N - 1)/2 counts.




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

**Figure 2-15: Mode 3 Timing**

### Mode 4: Software Triggered Strobe

OUT is initially high. When the initial count expires, OUT goes low for one CLK pulse and then goes high again. Counting sequence is triggered by writing initial count.

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

The Counter is loaded on the next CLK pulse after a Control Word and initial count have been written. 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 is loaded on the next CLK pulse and counting continues from the new count. If a two-byte count is written, the following events occur:

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.

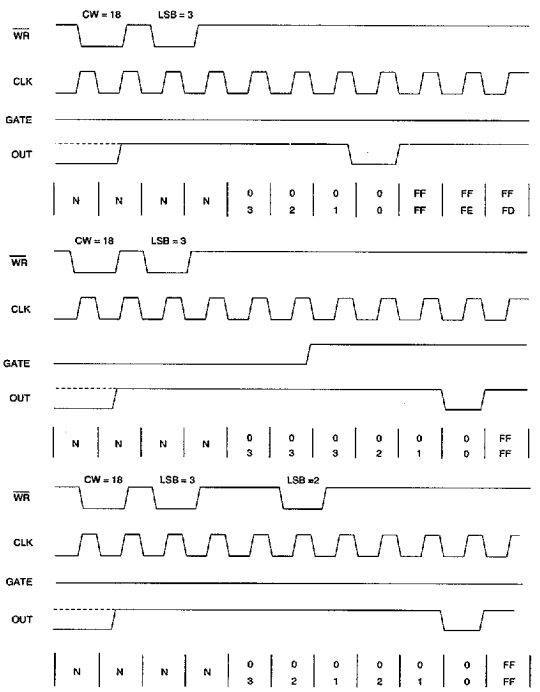



**Figure 2-16: Mode 4 Timing**

### Mode 5: Retriggerable Hardware Triggered Strobe

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

After a Control Word and initial count has been written, the counter is loaded on the first CLK pulse following a trigger. This CLK pulse does not decrement the count, so, given an initial count of N, OUT does not strobe low until N + 1 CLK pulses after a trigger.

A trigger causes the Counter to be loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable, so OUT will not go low until 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.

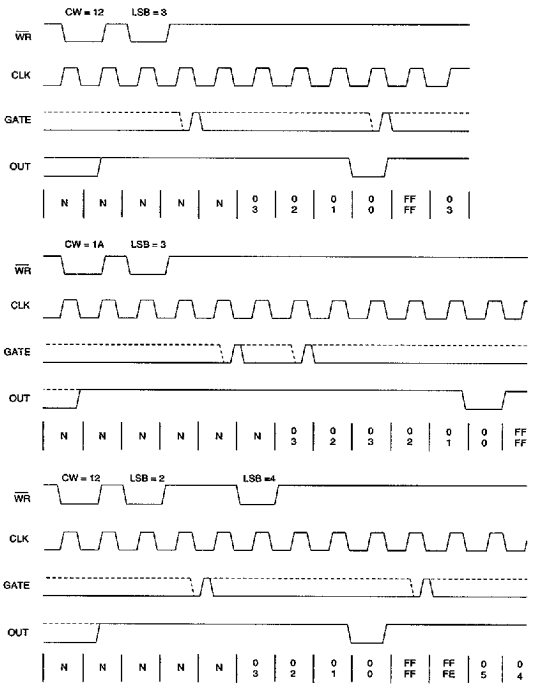



**Figure 2-17: Mode 5 Timing**