Ghosh & Sreedhar

CHAPTER 9

Programmable Interval Timer: 8253

1. The 8253 Programmable Interval Timer can be programmed to generate accurate time delays and waveforms under *software control*. It has three counters: Counter 0, Counter 1, and Counter 2, each of which can be independently programmed for the nature of the output.

2. In Mode 0, it generates an interrupt on the Terminal Count; in Mode 1, it provides a negative pulse of controllable width; in Mode 2, it generates a symmetric square wave of controllable frequency; in Mode 3, symmetric square waves are generated; in Mode 4, a negative pulse of one clock period duration is the output after a software controlled delay; and in Mode 5, a delayed negative pulse output of one clock period duration results following a positive going trigger input.

3. To program the 8253, an 8-bit control word is sent to its Control Register. The most significant two bits select a counter, the next two bits define whether it is a Read or Load Count operation and whether it is the LSB or the MSB of the count that is involved. The next three bits define the Mode of operation, and the last bit determines whether the counting is to be performed in binary or in BCD. After loading the Control Register with the Control word, the relevant counter is loaded with the desired count number.

The 8253 is a Programmable Interval Timer/Counter which can be used to generate a range of accurate time delays and waveforms (DC to 2MHz) using *software control*. This is achieved by using one or more of the three 16-bit, *presettable down counters* (the *down* counter starts with a given count number and counts *down* to zero) each of which can operate in either binary or BCD. The counter can be independently programmed for a desired *mode* and *count number* by writing an appropriate *control word* in the *control word register*. Because of this facility, timing loops in system software can be avoided; any one of these counters could be configured and set up to do such jobs and could be made to interrupt the CPU (or provide the output elsewhere) at the end of the termination of the count. This facility allows the CPU to carry out other tasks in the meantime. The counters and the Control Word Register can be considered by the system software

as an array of peripheral I/O ports with characteristic port addresses. Besides the data bus D_0-D_7, the 8253 needs A_0 and A_1 lines of the address bus, $\overline{RD}$, $\overline{WR}$, and a $\overline{CS}$. The $\overline{CS}$ can be derived from the address bus either by a Linear Select method or from a decoder. Each of the three counters has 2 inputs: the Clock and the Gate, and one Output. Figs. 9.1 and 9.2 show respectively the pin diagram and the functional block diagram of the 8253.

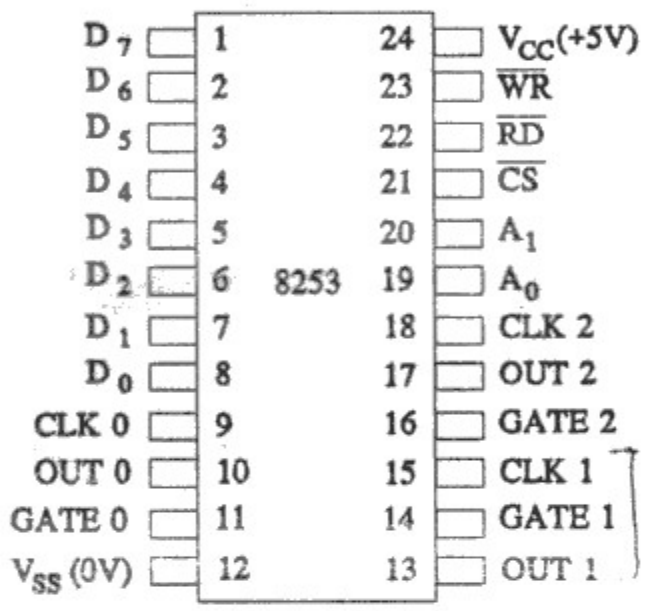

Figure 9.1 8253 Pin Diagram
(Source: Intel Corporation)

The six alternative modes (mode 0-5) in which the counters can be programmed to operate are described first. Six different types of signals, specified by Modes 0-5, can be generated by the three counters of 8253; all these counters have identical attributes.

9.1 THE SIX MODES OF OPERATION

Mode 0: Interrupt on Terminal Count

The output goes *low* on *setting* the mode, and goes *high after* the desired count.

Mode 1: Programmable One-Shot

The output goes *low* on a Gate input, and goes *high* on Terminal Count.

Mode 2: Rate Generator

It is equivalent to a '*Clock pulse divide by n*' counter. The output pulse frequency is 1/n of the input pulse frequency, where *n* is the count number.

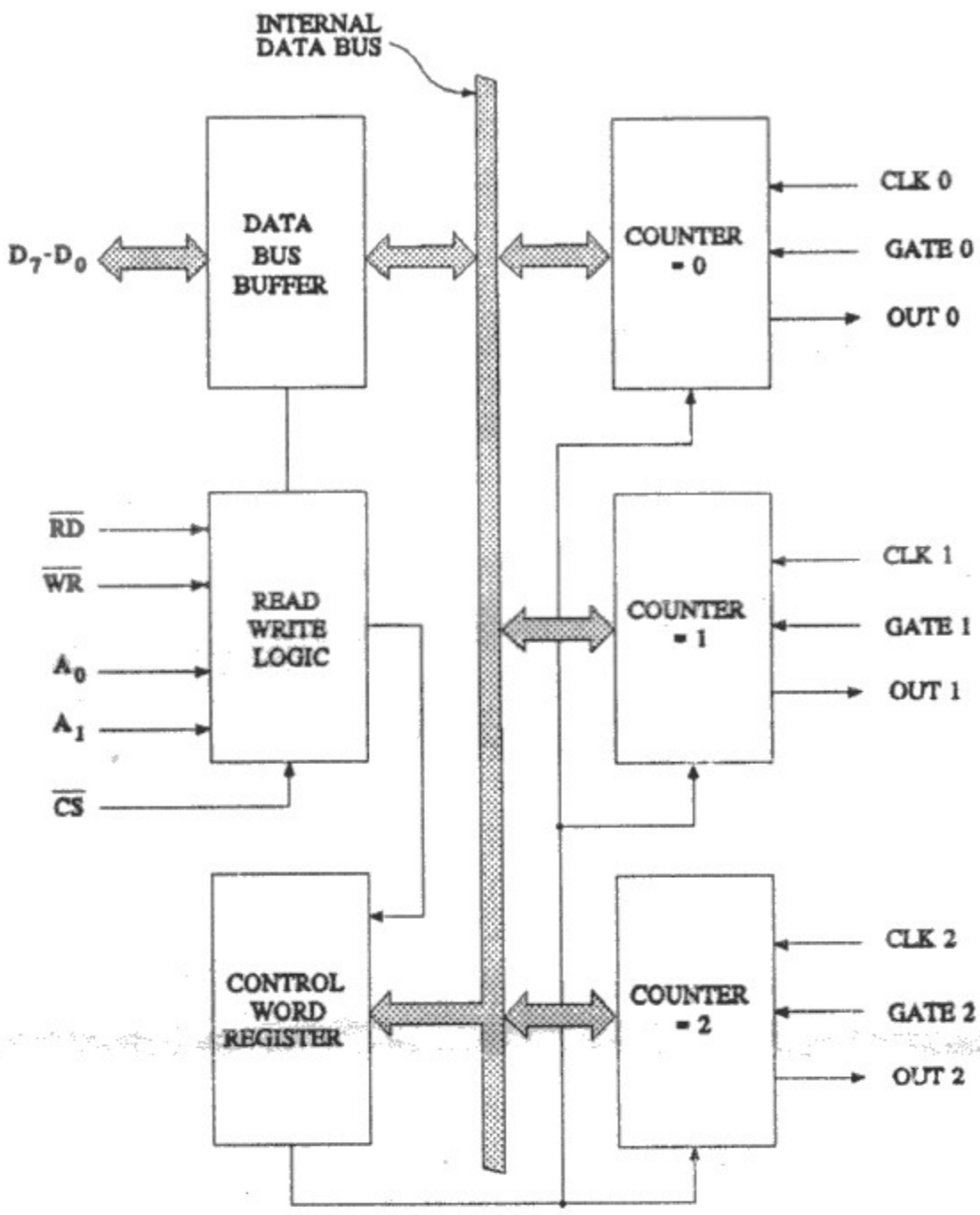

Figure 9.2 8253 Functional Block Diagram
(Source: Intel Corporation)

Mode 3: Square Wave Rate Generator

This is similar to Mode 2 with difference in minor operating details.

Mode 4: Software-Triggered Strobe

The output goes *high* on *setting* the mode. After Terminal Count, the output goes *low for one clock period* and *then* goes *high* again.

Mode 5: Hardware-Triggered Strobe

This is similar to Mode 4, but the counting is initiated by a trigger at the Gate.

9.1.1 Detailed Mode Description

In this section the six modes are described in detail; mode setting and count loading operations are described in a later section in this Chapter.

Mode 0: Interrupt On Terminal Count

Fig. 9.3 shows the operation of a counter in Mode 0.

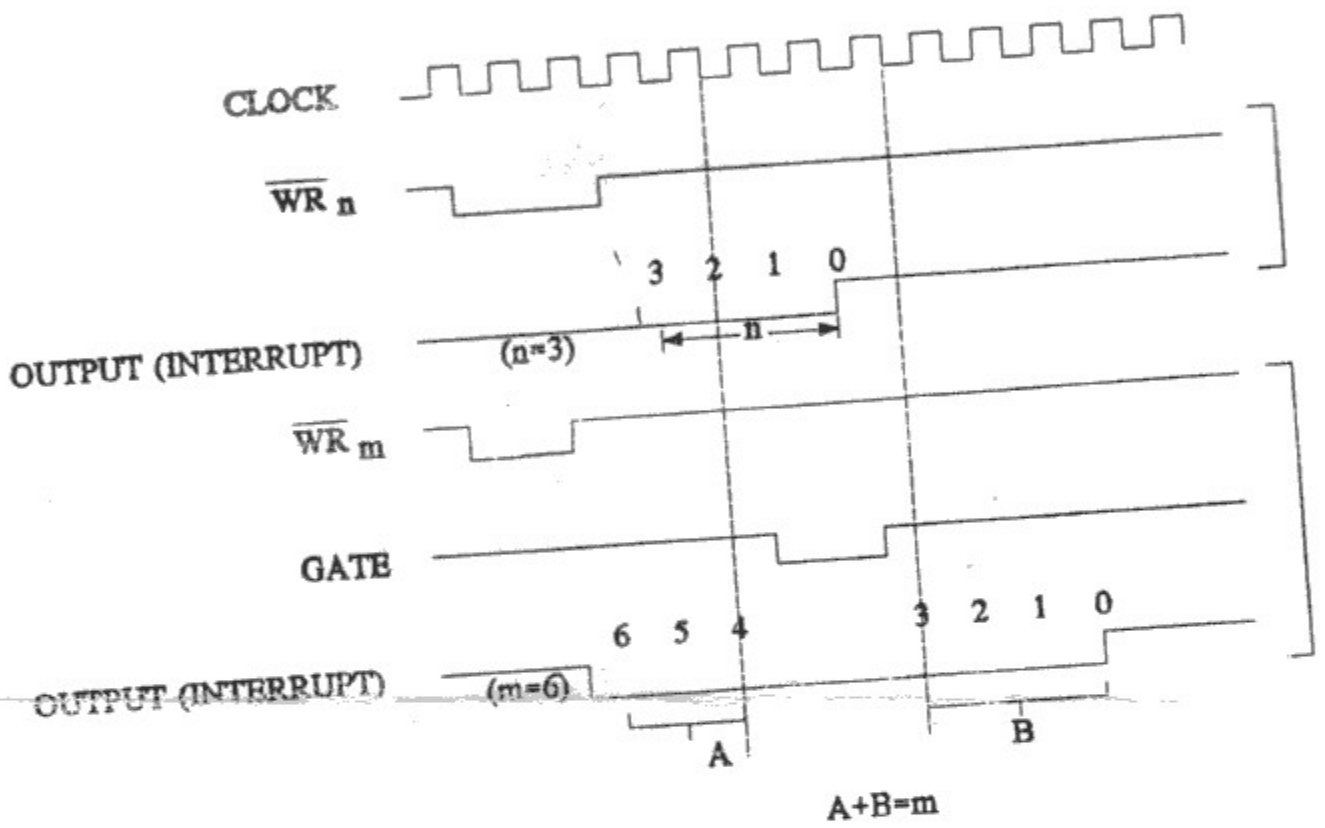

Figure 9.3 Mode 0 Operation
(Source: Intel Corporation)

OUTPUT

After the mode is set, the output becomes low and remains low while the selected counter is loaded with the count. The counter decrements the loaded count starting from the falling edge of the clock pulse which occurs just after the completion of the loading operation ($\overline{WR}$ goes high at the end of the loading operation) and continues to decrement the count at each subsequent clock pulse. When the terminal count is reached, the output becomes *high* and *stays high* till the count register is loaded with a new count or the mode of operation is changed.

GATE

If the Gate pin is made *low* while the counter decrements, the counting *stops*, and the then current contents are held. The process resumes only after the Gate pin is made *high* again. The counter decrement starts again from the time of the falling edge of the clock pulse subsequent to the rising edge of the Gate signal.

Further, if the count register is reloaded while the counting is on, then (i) after the first byte of the count is written, the current counting stops, and (ii) after the second byte of the count is written, the counting restarts with the new count number.

Mode 1: Programmable One-Shot (A Negative Pulse of Controllable Width)

Fig. 9.4 shows the operation of a counter in Mode 1.

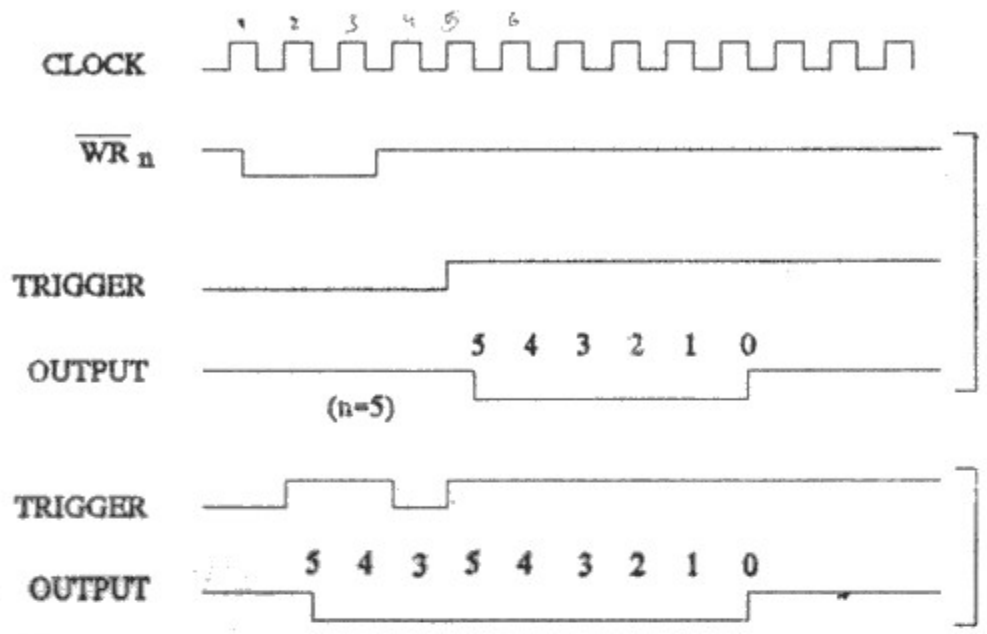

Figure 9.4 Mode 1 Operation
(Source: Intel Corporation)

OUTPUT

The output remains *high* while the mode is set and the count is loaded. It goes *low* after the rising edge of the Gate input (as shown in Fig. 9.4, the trigger goes high between the 4th and 5th clock pulse, while the output goes low on the falling edge of the 5th clock pulse). The output remains low till the terminal count is reached at which it becomes high again.

TRIGGER

If a trigger (Gate) signal appears *while* the output is low, the count is automatically reloaded into the counter and it is decremented *from the full count* again. The output remains low for the full count after an intermediate rising edge of the Gate input.

A new count may be added while the output is low, but it will *not* affect the duration of the one-shot pulse until the next trigger. Also, the current count can be read at any time and it will not affect the one-shot pulse.

Mode 2: Rate Generator

Fig. 9.5 shows the operation of a counter in Mode 2.

IN & OUT instructions perform I/O operation. Contents of AL or AX are transferred b/w the I/o device & the µP

IN & OUT
- Fixed port &
- Variable port
  - → IN AL, port
  - IN AX, port
  - → IN AL, DX
  - IN AX, DX

PROGRAMMABLE INTERVAL TIMER: 8253

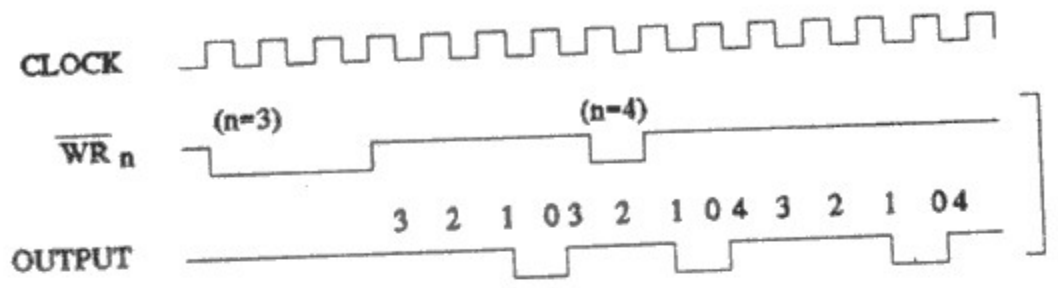

Figure 9.5 Mode 2 Operation
(Source: Intel Corporation)

OUTPUT

In this mode, depending on what is loaded for *n* as the count, after *n* pulses, the output goes *low* for *one* clock period (clock ON time + clock OFF time). Then it again becomes *high* for *n pulses* and low for *one* clock period and so on. It therefore works as a *divide by n* counter. If the counter is working in this mode and a new *n* value is loaded, the current output pulse timing is *not* affected; the *next one* is, according to the new value of *n*.

GATE

When the Gate input (Reset) goes low, it forces the output to go high and inhibits the counting. When the Gate goes high, the output starts its count from *that point* onwards. The Gate input can therefore be used to synchronise the counter with the Gate input pulse. The output can also be synchronised through software. The output goes high as soon as the mode is set. A count needs to be loaded and it starts counting only after the count is loaded.

Mode 3: Square Wave Rate Generator

This is similar to Mode 2 with the exception that the output remains low for half the count and is high for the other half (for even n).

There are two possibilities depending on whether *n* is even or odd.

(i) If n is even

The counter is decremented by *two* on the falling edge of each clock pulse till the Terminal Count is reached. The state of the output changes at this time (high to low or vice-versa), the count is

then reloaded with the full count and the process is again repeated. Fig. 9.6 shows the output of counter in Mode 3.

(ii) If n is odd

When the output is high, the first clock pulse decrements the count by *one* and subsequent clock pulses decrement the count by *two* till the Terminal Count is reached. The output then is made low and the full count is reloaded. When the output is low, the first clock pulse decrements the count by *three* and subsequent clock pulses decrement the count by *two* till the Terminal Count is reached (see Fig. 9.6).

In summary, if the count is odd, for (n+1)/2 pulses the output is high, and for (n-1)/2 pulses the output is low.

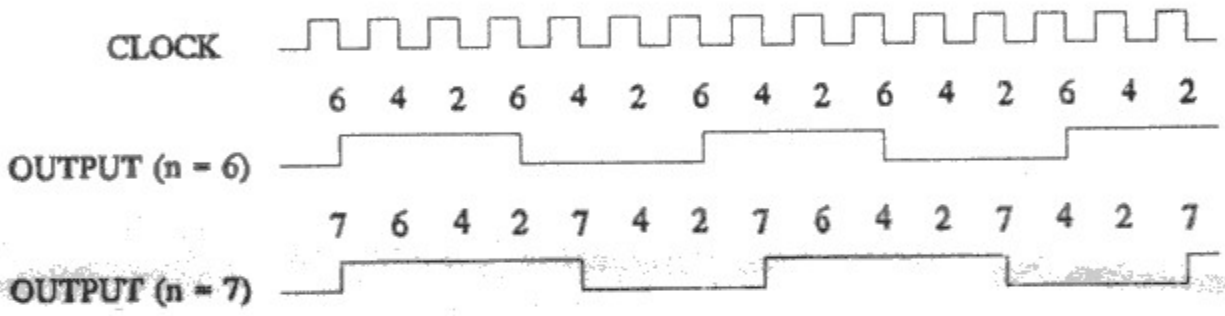

Figure 9.6 Mode 3 Operation
(Source: Intel Corporation)

Mode 4: Software Triggered Strobe

OUTPUT

A software-controlled delayed *negative* pulse of *one* clock period duration with or without synchronisation is generated in this mode. Fig. 9.7 shows a counter operating in Mode 4. In this mode, after the mode is set, the output becomes *high*. Counting starts after the counter is loaded. On reaching Terminal Count, the output goes low for one clock period and goes high again. If during counting, the count register is reloaded, the new count is loaded during the next clock pulse.

GATE

The counting is inhibited for the time the Gate input stays low. The Gate input can therefore be used for synchronisation.

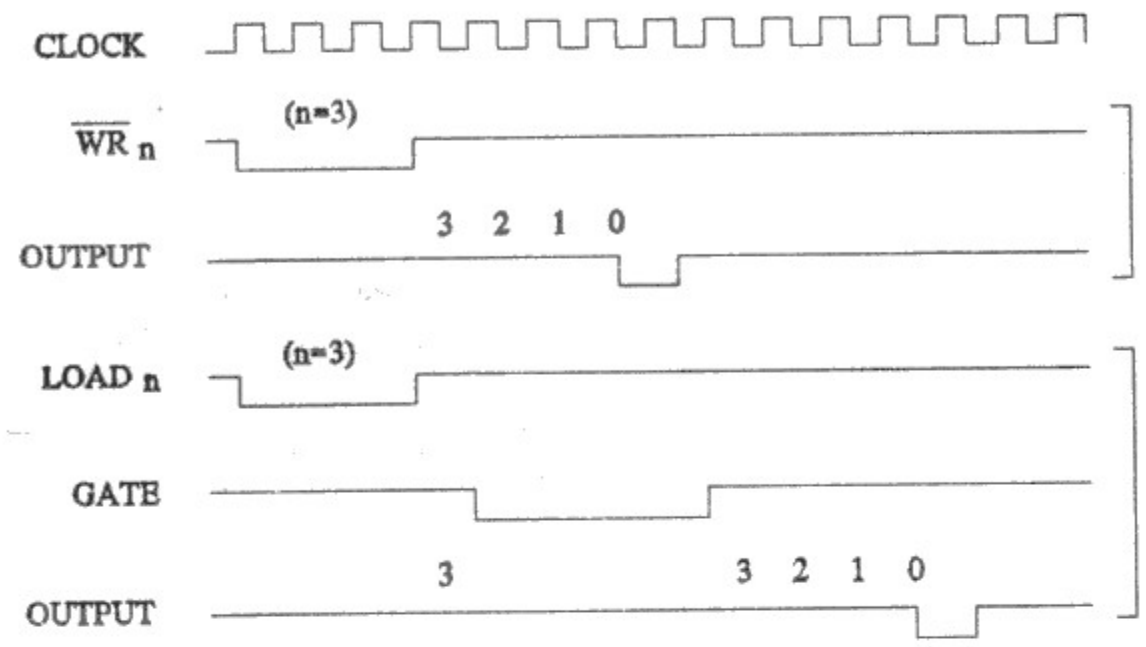

Figure 9.7 Mode 4 Operation
(Source: Intel Corporation)

Mode 5: Hardware Triggered Strobe

In this mode of operation, a delayed negative pulse with a width of one clock period is generated following a positive going *trigger* input at the Gate. Fig. 9.8 shows the output of a counter in Mode 5.

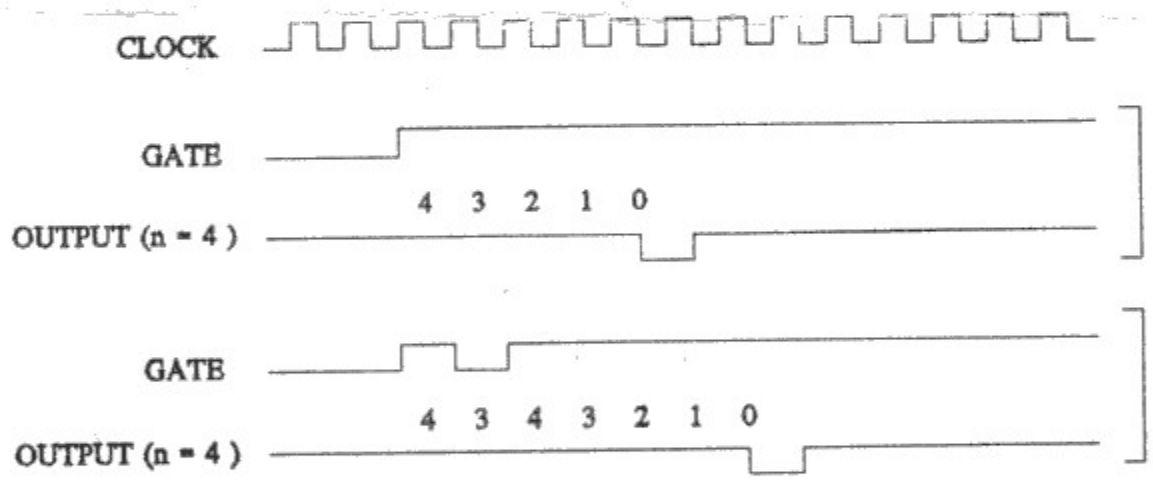

Figure 9.8 Mode 5 Operation
(Source: Intel Corporation)

OUTPUT

The counting starts after the *rising edge* of the Gate input, and on Terminal Count, the output goes *low* for one clock period. The counter can be retriggered. After the rising edge of any trigger, the output does not go low until the full count.

9.2 INTERNAL ORGANISATION AND INTERFACING

The following attributes of 8253 are described with respect to the functional block diagram of 8253 given in Fig. 9.2.

Data Bus Buffer

The tri-state bidirectional 8-bit data buffer interfaces with the system data bus; transmits data to the CPU on the IN instruction and receives data from the CPU on the OUT instruction (in I/O Mapped I/O). It has three basic functions: (i) programming the modes of the 8253, (ii) loading the count registers, and (iii) reading the count values.

Read/Write Logic

It accepts inputs from the system bus A_0, A_1, $\overline{RD}$, $\overline{WR}$, and generates control signals for overall device operation. Its operation/selection is enabled/disabled by the $\overline{CS}$. $\overline{RD}$ low from the CPU informs of a data input to the CPU, and $\overline{WR}$ low implies a data output by the CPU.

Control Word Register

When $A_0=1$ and $A_1=1$, as shown on row 4 in Table 9.1, the control register is selected and it accepts data from the data bus buffer. The information stored pertains to (i) the operation mode of each counter, (ii) selection of binary or BCD, and (iii) loading of each count register. Information can *only be written into* this register; it is *not possible* to *read* its contents.

Table 9.1 Counters/Control Register Selection (Source: Intel Corporation)

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	LOAD COUNTER NO. 0
0	1	0	0	1	LOAD COUNTER NO. 1
0	1	0	1	0	LOAD COUNTER NO. 2
0	1	0	1	1	WRITE MODE WORD
0	0	1	0	0	READ COUNTER NO. 0
0	0	1	0	1	READ COUNTER NO. 1
0	0	1	1	0	READ COUNTER NO. 2
0	0	1	1	1	NO-OPERATION 3-STATE
1	X	X	X	X	DISABLE 3-STATE
0	1	1	X	X	NO-OPERATION 3-STATE

Counters 0, 1, 2

The three 16-bit presettable down counters can be *independently* programmed through the control word sent to the control word register for (i) either binary or BCD operation, (ii) specific mode

selection (this configures its input, Gate, and output), and (iii) loading of the count value. It is possible to read the contents of each counter *without* inhibiting the clock input.

The requirement of specific $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, A_0, and A_1 states for various operations (writing to a counter or control register, reading a counter or the status word) are given in Table 9.1.

9.3 PROGRAMMING THE 8253

A set of control words from the CPU initialises the mode, selects binary or BCD counting, and determines the loading sequence of the count. The control word format and the definition of control word bits are given in Fig. 9.9.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

DEFINITION OF CONTROL:

SC - SELECT COUNTER:

SC1	SC0	
0	0	SELECT COUNTER 0
0	1	SELECT COUNTER 1
1	0	SELECT COUNTER 2
1	1	ILLEGAL

RL - READ/LOAD:

RL1	RL0	
0	0	COUNTER LATCHING OPERATION (SEE READ / WRITE PROCEDURE SECTION)
1	0	READ/LOAD MOST SIGNIFICANT BYTE ONLY
0	1	READ/LOAD LEAST SIGNIFICANT BYTE ONLY
1	1	READ/LOAD 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)

Figure 9.9 Control Word Format and Mode Definitions
(Source: Intel Corporation)

Counter Loading

For a counter to be considered *loaded*, it is necessary that (i) the count value be written (one or two bytes, depending on the mode selected by the RL bits) and (ii) it be followed by a rising *and* a falling edge of the clock. Data that is read before the falling edge of the clock may be invalid.