Unit 9 8086 Interfacing – 2

Structure:

9.1 Introduction

In the previous unit, you studied about the interfacing of keyboard and display devices. You also studied the interfacing of stepper motor with 8086. The 8253/54 the programmable interval timer that generates accurate time delays under software control solving one of most common problem in any microcomputer system. As a programmer you can configure the 8253/54 to match your requirements instead of setting up timing loops in system software. In this unit you will study about the programmable counter 8253/54 and its interfacing with 8086. Most of the microprocessors are designed for parallel communications. But for long distance data communication, this will increase cost of cabling. Again for CRT terminal, this parallel communication is not practicable. So for these types of situations, serial communication is preferred. In this unit you will also study about the serial communication concepts and the 8251 USART (Universal Synchronous Asynchronous Receiver Transmitter.

Objectives:

After studying this unit, you should be able to:

- explain the functional block diagram of 8253/54 timer
- explain the interfacing of 8253/54 with 8086
- explain block diagram of the 8251 USART
- discuss on the use of RS232 standard.

9.2 Programmable Interval Timer 8253/54

The 8254 is a programmable interval timer/counter designed for use with Intel microcomputer systems. The 8253/54 generates accurate time delays under software control. You can configure the 8253/54 to match your requirements, initializing one of the counters of the 8253/54 with the desired quantity.

The 8253/54 has three identical 16 bit counters and you can operate them independently. For example, to operate a counter, a 16-bit count is loaded in its register and on giving suitable command, it begins to decrement the count until it reaches 0. At the end of the count, it generates a pulse that can be used to interrupt the processor. You can operate the counter to count either in binary or BCD. In this unit, we are going to study two timer ICs 8253 and 8254. The 8254 is a superset of 8253. The functioning of these two ICs are almost similar along with the pin configuration. Only the differences are shown in the table 9.1

Table 9.1: Differences between 8253 and 8254

8253	8254
1. Operating frequency 0 - 2.6 MHz.	1. Operating frequency 0 - 10 MHz.
2. Uses N-MOS technology.	2. Uses H-MOS technology.
3. Read-Back command not available.	3. Read-Back command available.
4. Reads and writes of the same counter can not be interleaved.	4. Reads and writes of the same counter can be interleaved.

In this unit, things that are common to 8253/54 are mentioned and 8254 is mentioned specifically for giving only information about 8254.

Features: The important features are:

1) Three independent 16-bit down counters.
2) 8254 can handle inputs from DC to 10 MHz (5MHz 8254-5 8MHz 8254 10MHz 8254-2) where as 8253 can operate upto 2.6 MHz.
3) Three counters are identical presettable, and can be programmed for either binary or BCD count.
4) Counter can be programmed in six different modes.
5) Compatible with all Intel and most other microprocessors.
6) 8254 has powerful command called READ BACK command which allows the user to check the count value, programmed mode and current mode and current status of the counter.

9.2.1 Pin Diagram

The pin diagram of 8254 is shown in figure 9.1.

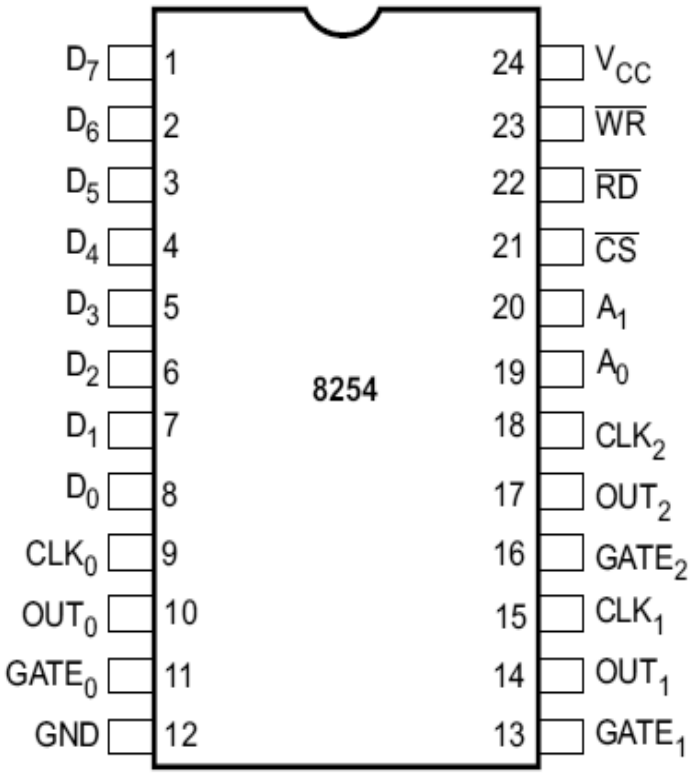


Figure 9.1: Pin diagram of 8254

Table 9.2 gives you the pin description of 8254.

Table 9.2: Pin Description of 8254

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

CLK 2	18	I	**CLOCK 2:** Clock input of Counter 2.
OUT 2	17	O	**OUT 2:** Output of Counter 2.
GATE 2	16	I	**GATE 2:** Gate input of Counter 2.
CLK 1	15	I	**CLOCK 1:** Clock input of Counter 1.
GATE 1	14	I	**GATE 1:** Gate input of Counter 1.
OUT 1	13	O	**OUT 1:** Output of Counter 1.

9.2.2 Block Diagram

Figure 10.2 shows the block diagram of 8253/54. It includes three counters, a data bus buffer, Read/Write control logic, and a control register. Each counter has two input signals CLOCK and GATE and one output signal OUT.

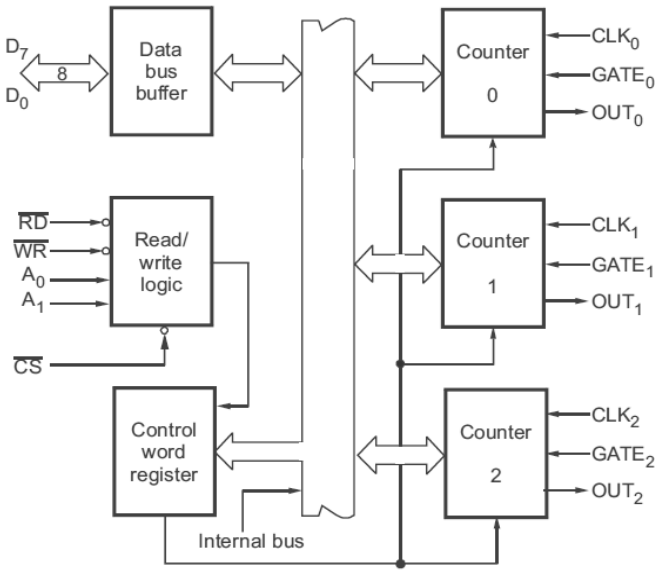


Figure 9.2: Block diagram of 8254

Data Bus Buffer:

This is the tri-state, bi-directional, 8-bit buffer and is used to interface the 8253/54 to the system data bus. Its basic functions are:

1. Programming the modes of 8253/54.
2. Loading the count registers.
3. Reading the count values.

Read/Write Logic: The Read/Write logic has five signals: $\overline{RD}$, $\overline{WR}$, $\overline{CS}$ and the address lines A0 and A1. In the peripheral I/O mode, the $\overline{RD}$, and $\overline{WR}$ signals are connected to $\overline{IOR}$ and $\overline{IOW}$, respectively. In memory-mapped I/O, these are connected to MEMR and MEMW. Address lines A0 and A1 of the CPU are usually connected to lines A0 and A1 of the 8253/54, and $\overline{CS}$ is connected to a decoded address. A0 and A1 lines are used to select control word register and counters as shown in the table 9.3.

Table 9.3: Selection of Counters and Control Register

A_1	A_0	Selection
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control word Register

Control Word Register: This register can be selected when lines A0 and A1 are at logic 1. It is used to write a command word which specifies the counter to be used (binary or BCD), its mode, and either a read or write operation.

Counters: These three counters are identical in operation. Each counter consists of a single, 16 bit, pre-settable, down counter. The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of modes stored in the control word register. The counters are fully independent. You can read the contents of any of the three counters without disturbing the actual count in process.

9.2.3 Operational Description

The complete functional definition of the 8253/54 is programmed by the system software. The 8253/54 performs whatever timing tasks it is assigned to accomplish once it is programmed.

Programming the 8253/54: You can program individually each counter of the 8253/54 by writing a control word into the control word register (A0 - A1 = 11). The figure 9.3 shows the control word format.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC_1	SC_0	RW_1	RW_0	M_2	M_1	M_0	BCD

SC - Select counter

SC_1	SC_0	
0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2
1	1	Illegal for 8253 Read -Back command for 8254 (See Read operations)

M - Mode

M_2	M_1	M_0	
0	0	0	Mode 0
0	0	1	Mode 1
x	1	0	Mode 2
x	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW - Read /Write

RW_1	RW_0	
0	0	Counter latch command (See Read operations)
0	1	Read / Write least significant byte only
1	0	Read / Write most significant byte only
1	1	Read / write least significant byte first, then most significant byte

BCD :

0	Binary counter 16 - bits
1	Binary coded decimal (BCD) Counter (4 Decades)

Note : Don't care bits (x) should be 0 to ensure compatibility with future Intel products

Figure 9.3: Control Word Format

Write Operation: You can perform the write operation as follows:

1. Write a control word into control register.
2. Load the low-order byte of a count in the counter register.
3. Load the high-order byte of count in the counter register.

Read Operation: In some applications, especially in event counters, it is necessary to read the value of the count in process. This can be done by three possible methods:

1. **Simple Read:** It involves reading a count after inhibiting the counter by controlling the gate input or the clock input of the selected counter, and two I/O read operations are performed by the processor. The first I/O operation reads the low-order byte, and the second I/O operation reads the high order byte.
2. **Counter Latch Command:** In this method, an appropriate control word is written into the control register to latch a count in the output latch, and two I/O read operations are performed by the processor. The first I/O operation reads the low-order byte, and the second I/O operation reads the high order byte.
3. **Read-Back Command (Available only for 8254):** This method uses the Read-Back command. This command allows the user to check the count value, programmed Mode, and current status of the OUT pin and Null count flag of the selected counter(s). Figure 9.4 shows the format of the control word register for Read-Back command.

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	$\overline{COUNT}$	$\overline{STATUS}$	CNT_2	CNT_1	CNT_0	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 9.4: Control Word Register for read-back command

The Read-Back command may be used to latch multiple counter output latches by setting the COUNT bit D5 = 0 and selecting the desired counter (s). Each counter's latch count in held until it is read. That counter is automatically unlatched when read.

Other Features of Read - Back Command (only for 8254):

The Read-Back command may also be used to latch status information of selected counter(s) by setting Status bit D4 = 0. The contents of the counter must be latched before reading. The status of a counter is then accessed by a read from that counter. The figure 9.5 shows the counter status format.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
OUTPUT	NULL COUNT	RW_1	RW_0	M_2	M_1	M_0	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

Figure 9.5: Counter status Format

Bit D5 - D0 contains the counter's programmed mode. Bit D7 contains the current status of the output pin. In 8254, it is not possible to read count from the counter, if the count is not loaded into the counting element (CE). The Bit D6 indicates whether the counting element has count or not. If D6 = 0, counting element has count otherwise null count.

Interleaved Read and Write: It is possible in 8254 that reads and writes of the same counter may be interleaved. For example, if the counter is programmed for the 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.

Mode 0: Interrupt on terminal count

a) Normal Operation:

1) The output will be initially low after the mode set operation.
2) After the count is loaded into the selected count Register the output will remain low and the counter will count.
3) When the terminal count is reached the output will go high and remain high until the selected count is reloaded.

b) Gate Disable

1) Gate = 1 enables counting.
2) Gate = 0 disables counting.

Note: Gate has no effect on OUT.

Figure 9.6 shows the waveforms of operation of Mode 0: interrupt on terminal count.

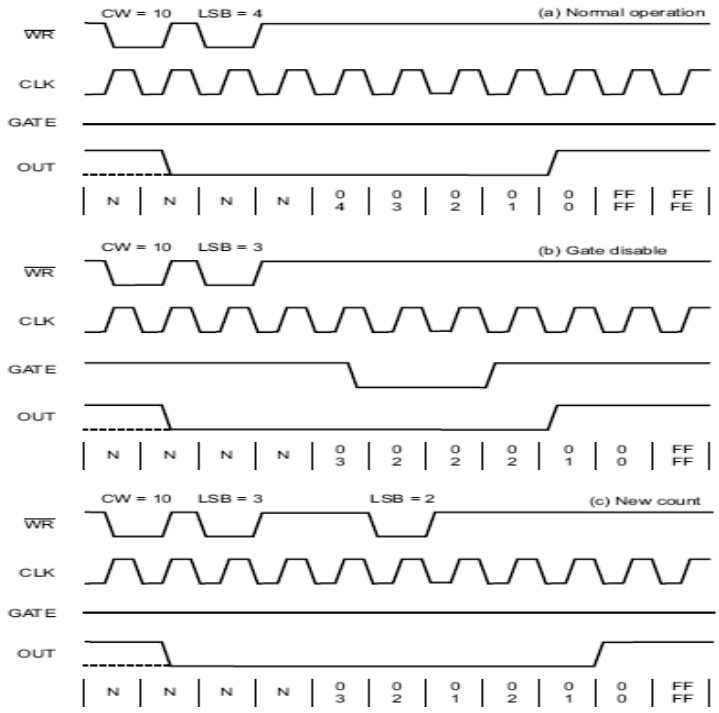


Figure 9.6 Mode 0: Interrupt on terminal count

c) New Count

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

In case of two byte count:

1) Writing the first byte disables counting.
2) Writing the second byte loads the new count on the next CLK pulse and counting will continue from the new count.

MODE 1: Hardware Retriggerable One-shot

a) Normal operation

1) The output will be initially high

2) The output will go low on the CLK pulse following the rising edge at the gate input.
3) The output will go high on the terminal count and remain high until the next rising edge at the gate input.

b) Retriggering

The one shot is retriggerable, hence the output will remain low for the full count after any rising edge of the gate input.

c) New count

If the counter is loaded during one shot pulse, the current one shot is not affected unless the counter is retriggered. If retriggered, the counter is loaded with the new count and the one-shot pulse continues until the new count expires.

Figure 9.7 shows the waveforms of operation of Mode 1: hardware retriggerable one-shot.

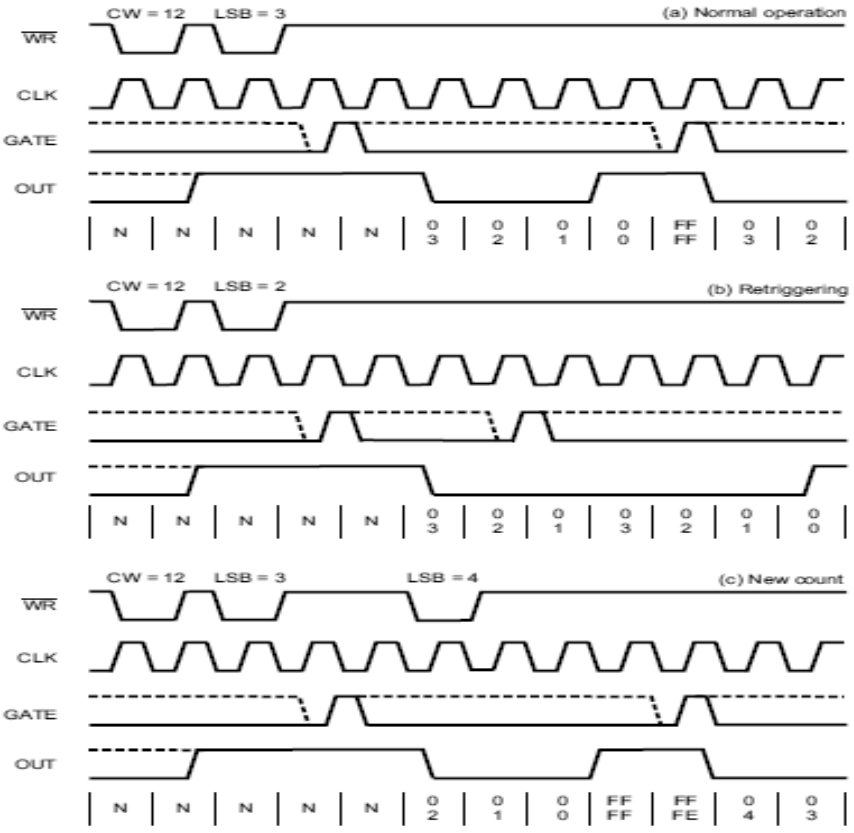


Figure 9.7: Mode 1 hardware retriggerable one-shot

MODE 2: Rate generator

This mode functions like a divide by-N counter.

a) Normal Operation

1) The output will be initially high.
2) The output will go low for one clock pulse before the terminal count.
3) The output then goes high, the counter reloads the initial count and the process is repeated.
4) The period from one output pulse to the next equals the number of input counts in the count register.

b) Gate Disable

1) If Gate = 1 it enables a counting otherwise it disables counting (Gate = 0).
2) If Gate goes low during a low output pulse, output is set immediately high. A trigger reloads the count and the normal sequence is repeated.

c) New count:

The current counting sequence does not affect when the new count is written. Suppose if a trigger is received after writing a new count but before the end of the current period, the new count 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. Figure 9.8 shows the waveforms of Mode 2 rate generator.

Note: In mode 2, a count of 1 is illegal.

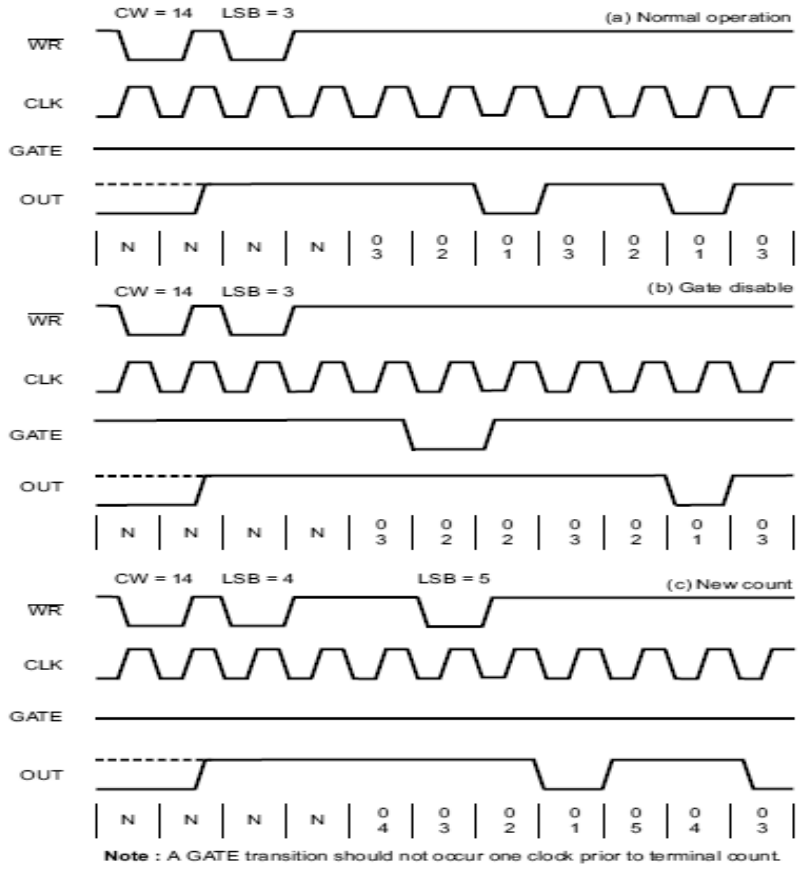


Figure 9.8: Mode 2 rate generator

MODE 3: Square Wave Rate Generator

a) Normal operation

1) Initially output is high.
2) For even count, counter is decremented by 2 on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.
3) If the count is odd and the output is high the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the clock by 2. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the count by 3 and subsequent clock pulse

decrement the count by two. Then the whole process is repeated. In this way, if the count is odd, the output will be high for (n+1)/2 counts and low for (n-1)/2 counts.

b) Gate Disable

If Gate is 1 counting is enabled otherwise it is disabled. If Gate goes low while output is low, output is set high immediately. After this, when Gate goes high, the counter is loaded with the initial count on the next clock pulse and the sequence is repeated.

c) New Count

The current counting sequence does not affect when the new count is written. 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 end of the current half-cycle.

Figure 9.9 shows the waveforms of Mode 3: Square Wave generator mode.

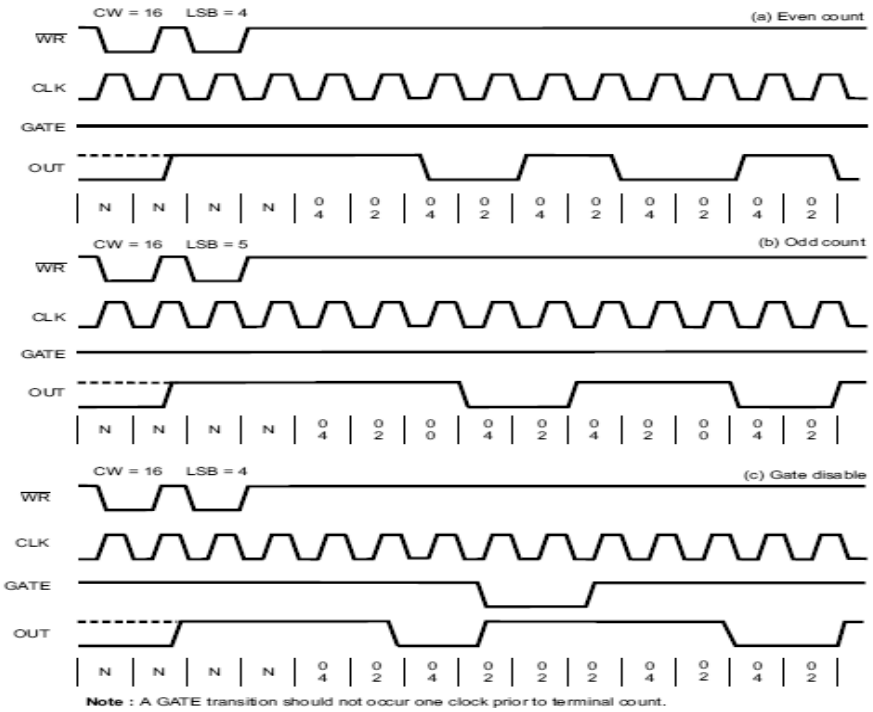


Figure 9.9: Mode 3- Square Wave Generator Mode

MODE 4: Software Triggered Strobe.

a) Normal operation

1) Initially the output will be high
2) The output will go low for one CLK pulse after the terminal count (TC).

b) Gate Disable

If Gate is one the counting is enabled otherwise it is disabled. The Gate has no effect on the output.

c) New count

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 the count is two byte then

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.

Figure 9.10 shows the waveforms of Mode 4 software triggered strobe

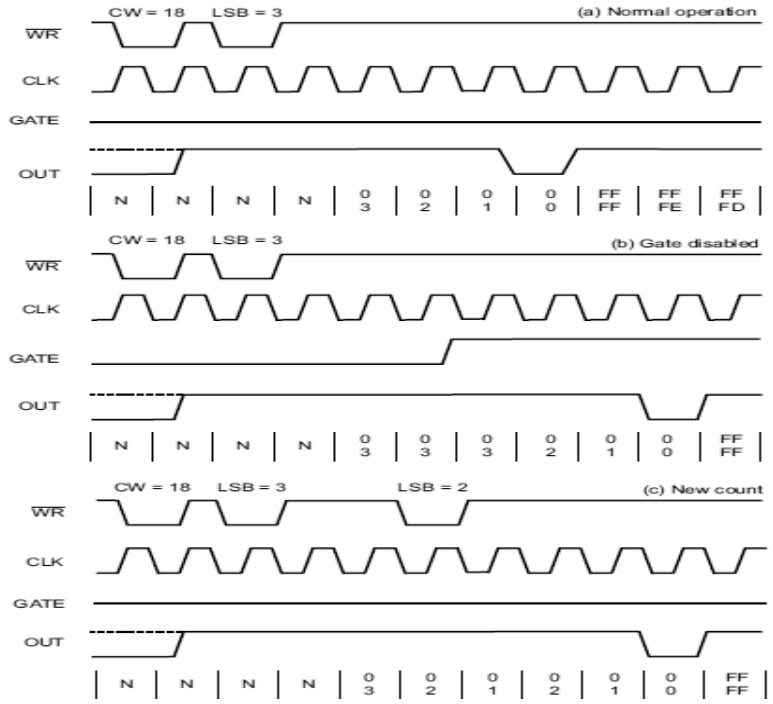


Figure 9.10: Mode 4 software triggered strobe

MODE 5: Hardware triggered strobe (Retriggerable).

a) Normal operation

1) Initially the output will be high.
2) The counting is triggered by the rising edge of the Gate.
3) The output will go low for one CLK pulse after the terminal count (TC).

b) Retriggering

If the triggering occurs on the Gate input during the counting, the initial count is loaded on the next CLK pulse and the counting will be continued until the terminal count is reached.

c) New count

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

Figure 9.11 shows the waveforms of operation of Mode 5 hardware triggered strobe

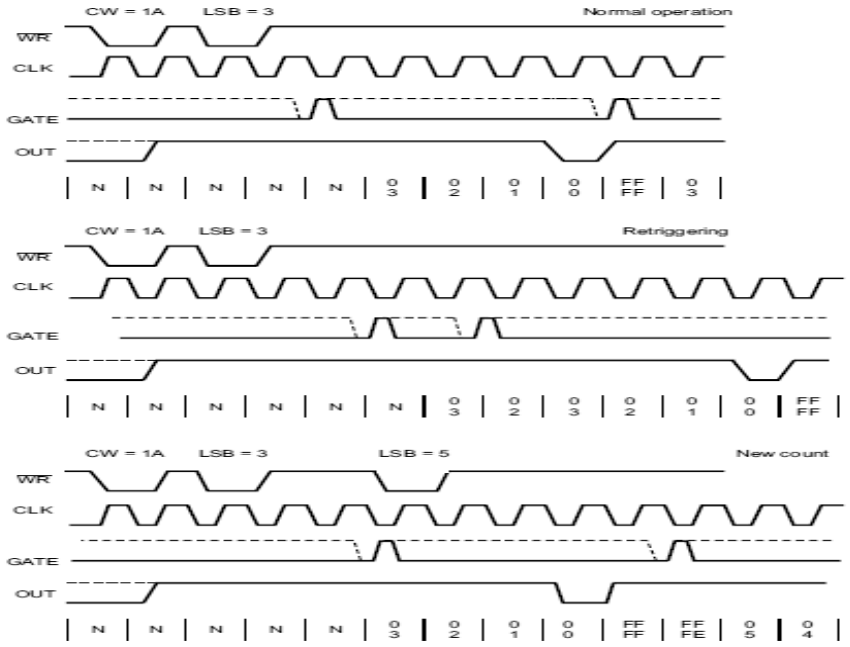


Figure 9.11: Mode 5 hardware triggered strobe

Table 9.4 shows the gate pin operations.

Table 9.4: Gate pin operations

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

Programming Examples:

Example 1: Write a program to initialize counter 2 in mode 0 with a count of C030H. Assume address for control register = 0BH, counter 0 = 08H, counter 1 = 09 Hand counter 2 = 0AH.

Solution: Control word

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC_1	SC_2	RW_1	RW_0	M_2	M_1	M_0	BCD
1	0	1	1	0	0	0	0

= B0H

Source Program

```
MOV AL, B0H
OUT 0BH, AL        ; Loads control word (B0H) in the control register.
```

```
MOV AL, 30H
OUT 0AH, AL        ; Loads lower byte of (30H) the count.
MOV AL, 0C0H
OUT 0AH, AL        ; Loads higher byte (C0H) of the count.
```

Example 2: Write a program to generate a square wave of 1 KHz frequency on OUT 1 pin of 8253/54. Assume CLK1 frequency is 1MHz and address for control register = 0BH, counter 1 = 09H and counter 2 = 0AH.

Solution: To get square wave mode 3 is selected count should be 1 MHz/ 1KHz = 1000

Control word:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SC_1	SC_2	RW_1	RW_0	M_2	M_1	M_0	BCD	
0	1	1	1	0	1	1	1	= 77H

Source program

```
MOV AL, 77H
OUT 0BH, AL        ; Loads control word (77H) in the control register.
MOV AL, 00H        ; loads lower byte (00) of the count
OUT 09H, AL
MOV AL, 10         ; Loads higher byte (10) of the count
OUT 09H, AL
```

Self Assessment Questions

1. The _____ is a programmable interval timer/counter designed for use with Intel microcomputer systems.
2. The 8253/54 has _____ identical 16 bit counters and you can operate them independently.
3. Control Word Register can be selected when lines A0 and A1 are at logic 1. (True/False)
4. _____ is called Square Wave Rate Generator.
5. Which mode is the Hardware triggered strobe (Retriggerable)?

9.3 Interfacing of 8253/54 with 8086

With 8-Bit Address:

We know that, 8253/54 has two address and eight data lines. So, it is necessary to interface lower byte of demultiplexed data bus to 8253/54. The address lines A1 and A2 are connected to the address lines of 8253/54. The 8253/54 IC decodes A1 and A2 lines internally to select one of its ports or control register. The remaining address lines (A7-A3) can be used to generate chip select signal. Figure 9.12 shows the interfacing of 8253/54 with 8086.

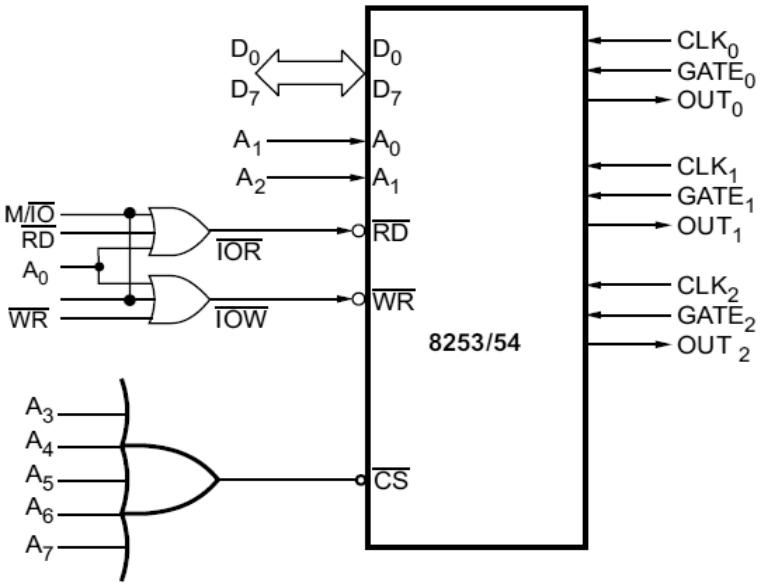


Figure 9.12: Interfacing of 8253/54 with 8-bit address

Address Map:

Ports / control Register	Address lines								Address
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Counter 0	0	0	0	0	0	0	0	0	00H
Counter 1	0	0	0	0	0	0	1	0	02H
Counter 2	0	0	0	0	0	1	0	0	04H
Control Register	0	0	0	0	0	1	1	0	06H

With 16-Bit Address

Figure 9.13 shows the interfacing of 8253/54 with 8086 with 16-bit address. Here $\overline{RD}$ and $\overline{WR}$ signals are activated when M/$\overline{IO}$ =0, indicating I/O bus cycle. To get absolute address, all remaining address lines (A3 - A15) are used to decode the address for 8253/54.

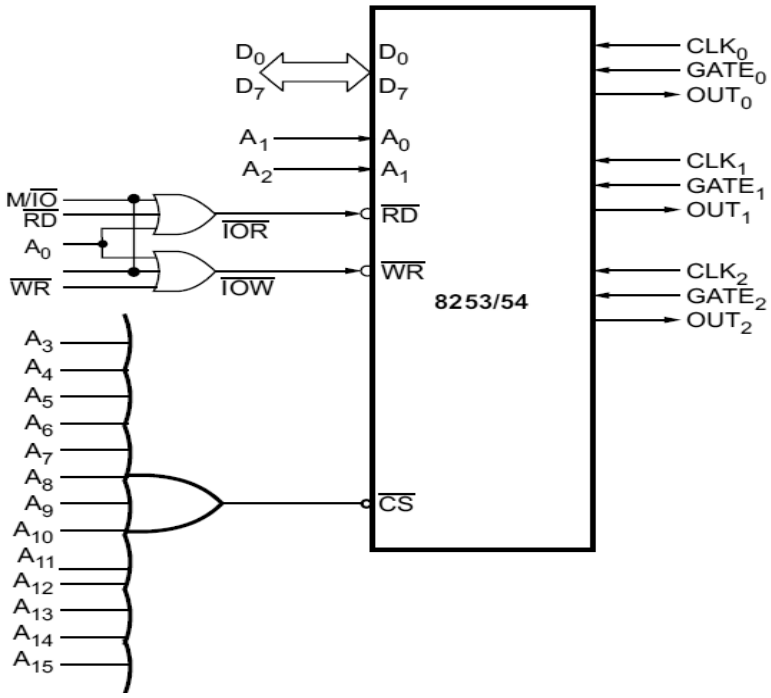


Figure 9.13: Interfacing 8253/54 with 16-bit address

Note: When 16-bit address is used it is necessary to use indirect addressing mode to access 8253/8254.

Address Map:

Ports/Control Register	Address lines																Address
	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Counter 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
Counter 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002H
Counter 2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004H
Control Register	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006H

9.3.1 Interfacing 8253/54 to 8086 in Memory Mapped I/O

In this type of I/O interfacing the 8086 uses 20 address lines to identify an I/O device, an I/O device is connected as if it is a memory register. The 8086 uses same control signals and instructions to access I/O as those of memory. Figure 9.14 shows the interfacing of 8253/54 with 8086 in memory mapped I/O technique. Here $\overline{RD}$ and $\overline{WR}$ signals are activated when M/$\overline{IO}$ signal is high, indicating memory bus cycle. The (A1 - A2) address lines are used by 8253/54 for internal decoding. To get absolute address, all remaining address lines (A3 - A19) are used to decode the address for 8255. Other signal connections are same as in I/O mapped I/O.

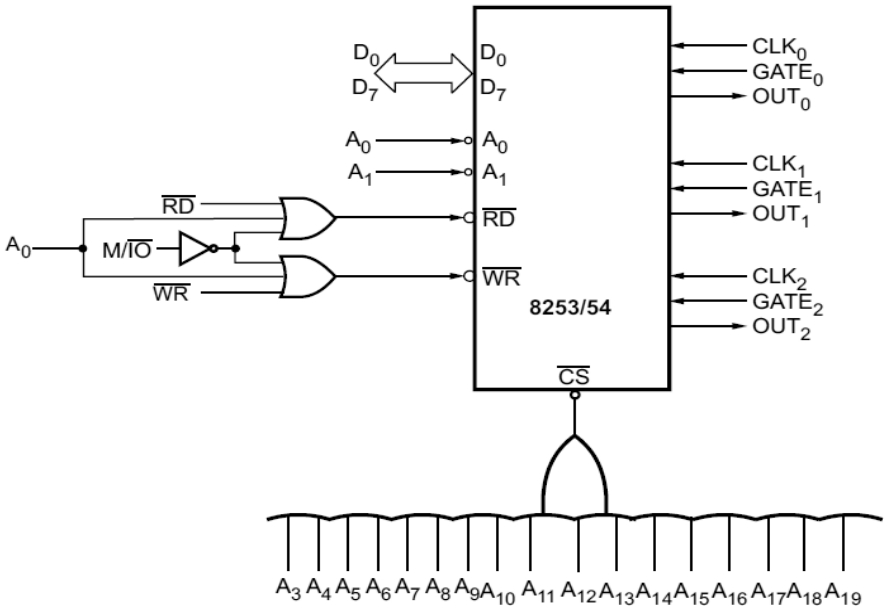


Figure 9.14: Interfacing 8253/54 with 8086 in memory mapped I/O

I/O Map:

Register	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Address
Counter 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
Counter 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002H
Counter 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004H
Control register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006H

9.4 Data Communication

As we know, in telecommunications serial transmission is the sequential transmission of signal elements of a group representing a character or other entity of data. Digital serial transmissions are bits sent sequentially over a single wire, or optical path or channel. The transfer rate of each individual path may be faster and also it requires less signal processing and less chances for error than parallel transmission. So, you can use this over longer distances. In digital communications, parallel transmission requires multiple electrical wires and can transmit multiple bits simultaneously, which allows for higher data transfer rates than that can be achieved with serial transmission. This method is used internally within the computer, for example the internal buses, and sometimes externally for such things as printers, the major issue with this is "skewing" because the wires in parallel data transmission have slightly different properties. So some bits may arrive before others, which may corrupt the message. A parity bit can help to reduce this. However, electrical wire parallel data transmission is less reliable for long distances because corrupt transmissions are far more likely, in this case serial transmission is preferred. Serial communication port is the interface used in microprocessor based system for connecting devices like printer, modems, CRT terminals. It uses only two lines one for data transfer and another line for receiving data.

The figure 9.15 shows the concept of serial communication.

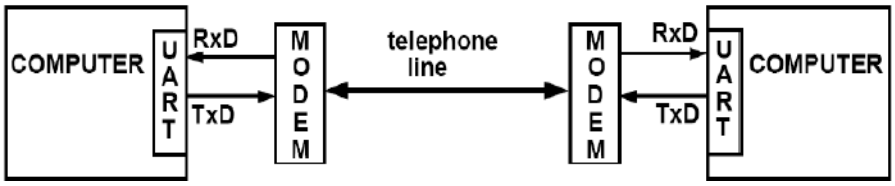


Figure 9.15: Serial Data Communication

The data transmission could be asynchronous or synchronous. Asynchronous transmission (refer figure 9.16) eliminates the need for clock signal. It uses start and stop bits as shown in figure 9.16(b). ASCII character would actually be transmitted using 10 bits e.g.: A "0100 0001" would become "1 0100 0001 0".

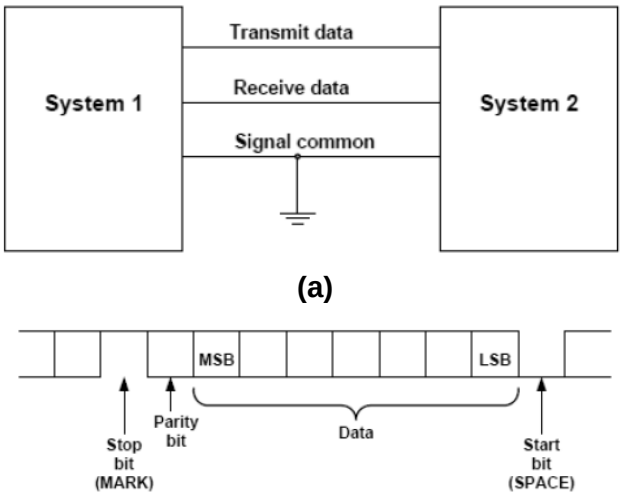


Figure 9.16: (a) Asynchronous Communication (b) Asynchronous Data Transmission Format

The extra one (or zero depending on parity bit) at the start and end of the transmission tells the receiver first that a character is coming and secondly that the character has ended.

Synchronous transmission shown in figure 9.17 uses no start and stop bits but instead synchronizes transmission speeds at both the receiving and sending end of the transmission using clock signal(s) built into each component. A continual stream of data is then sent between the two nodes.

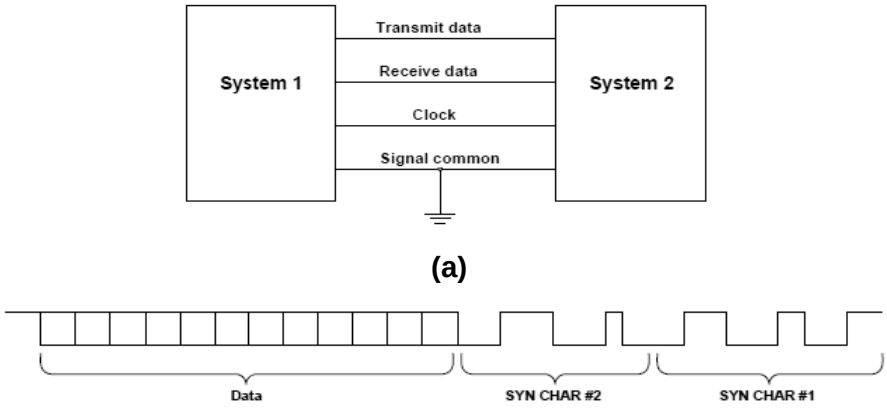


Figure 9.17: (a) Synchronous Communication Interface (b) Synchronous Data Transmission Format

Since no start and stop bits used, the data transfer rate is quicker. But more errors may occur, when the clocks get out of sync, so some bytes could become corrupted by losing bits. Ways to get around this problem include re-synchronization of the clocks and use of check digits to ensure the byte is correctly interpreted and received.

The table 9.5 shows the difference between Synchronous and Asynchronous serial data transfer

Table 9.5: Comparison between Synchronous and Asynchronous serial data transfer.

Sr. No.	Asynchronous Serial Communication	Synchronous Serial Communication
1.	Transmitters and receivers are not synchronized by clock.	Transmitter and receivers are synchronized by clock.
2.	Bits of data are transmitted at constant rate.	Data bits are transmitted with synchronisation of clock.
3.	Character may arrive at any rate at receiver.	Character is received at constant rate.
4.	Data transfer is character oriented.	Data transfer takes place in blocks.
5.	Start and stop bits are required to establish communication of each character.	Start and stop bits are not required to establish communication of each character; however, synchronisation bits are required to transfer the data block.
6.	Used in low-speed transmissions at about speed less than 20 kbits/sec.	Used in high-speed transmissions

We need two devices to for serial communication in microprocessor based system. They are: parallel to serial converter and serial to parallel converter.

Parallel to serial converter are required to convert parallel data to serial format to transmit data serially and serial to parallel converter are required to convert the received serial data to parallel format. UART (Universal Asynchronous Receiver Transmitter) and USART (Universal Synchronous Asynchronous Receiver Transmitter) are the devices used for this purpose.

The UART performs the "overhead" tasks necessary for asynchronous serial communication. Parity bits as shown in figure 9.18 are also often employed to ensure that the data sent has not been corrupted.

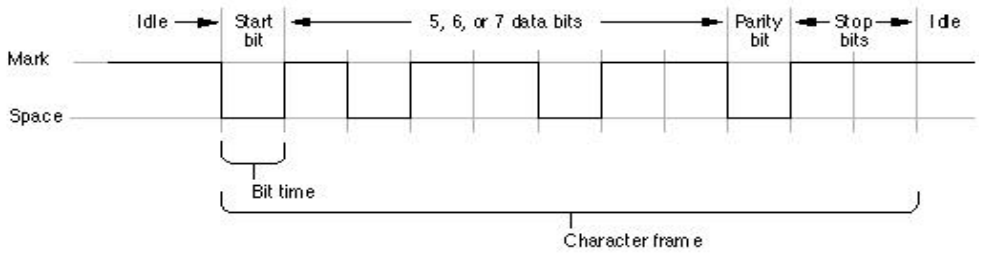


Figure 9.18: Frame Format

The UART usually generates the start, stop, and parity bits when transmitting data, and can detect communication errors upon receiving data. 8250 is an example of UART. The device which provides synchronous as well as asynchronous transmission and reception is called as Universal Synchronous Asynchronous Receiver Transmitter (USART). 8251 is an example of USART. We will discuss on this USART in the sub-section 9.4.3.

9.4.1 RS232 Standard

RS232 standard is developed by the Electronic Industry Association (EIA) in 1962 and was revised and renamed as RS232C. RS stands for "recommended standard." This is a standard hardware interface used for implementing asynchronous serial data communication ports on devices such as CRT terminals, printers, modems and keyboards. RS-232 can be plugged straight into the computer's serial port (know as COM or Comm port). RS-232 has been around as a standard for decades as an electrical interface between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) such as modems. DTE stands for Data Terminal Equipment, and DCE stands for Data Communications Equipment. Your computer is a DTE device, while most other devices such as modem and other serial devices are usually DCE devices. RS-232 is the interface that your computer uses to talk to and exchange data with your modem and other serial devices. The serial ports on most computers use a subset of the RS-232C standard. The RS-232 standard defines the voltage levels that correspond to logical one and logical zero levels for the data transmission and the control signal lines. Valid signals are either in the range of +3 to +15 volts, or the range -3 to -15 volts; the range between -3 to +3 volts is not a valid RS-232 level. Data signals and control signals use opposite polarity to represent a "true" or logic 1 asserted state. RS-232C, EIA RS-232, or simply RS-232, refers to the same standard.

The figure 9.19 shows the use of RS232 in Communication.

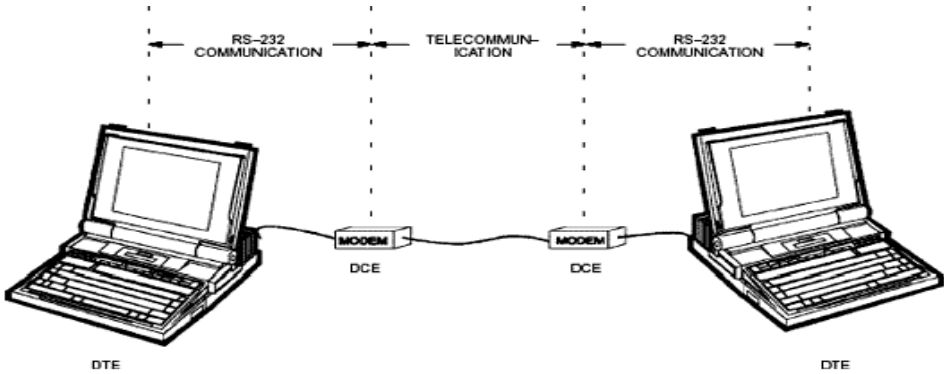
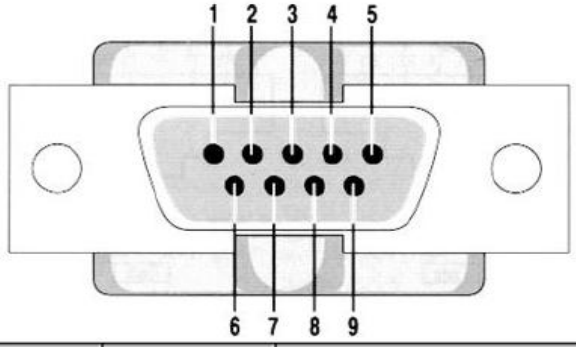


Figure 9.19: Use of RS232 in Communication

RS232 on DB9 (9-pin D-type connector)

Figure 9.20 shows standardized pin out for RS-232 on DB9 (9-pin D-type connector).

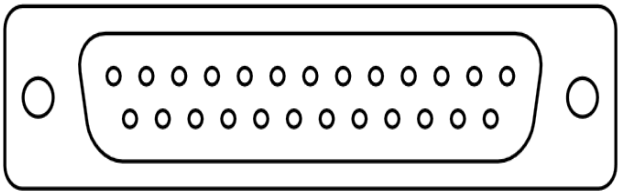


Pin Number	Signal	Description
1	DCD	Data carrier detect
2	RxD	Receive Data
3	TxD	Transmit Data
4	DTR	Data terminal ready
5	GND	Signal ground
6	DSR	Data set ready
7	RTS	Ready to send
8	CTS	Clear to send
9	RI	Ring Indicator

Figure 9.20: 9-Pin D- type Connector Pin Assignment

RS232 on DB25 (25-pin D-type connector)

Most of the pins in DB-25 connector are not needed for normal PC communications, and indeed, most new PCs are equipped with male D type connectors having only 9 pins. You can extend normal cable limitation of 50 feet to several hundred feet with high-quality cable by using a 25-pin DB-25 or 9-pin DB-9 connector. RS-232 defines the purpose and signal timing for each of the 25 lines; however, many applications use less than a dozen. The standardized pin out for RS-232 on a DB25 connector is as shown in figure 9.21.



Pin Number	Signal	Description
1	PG	Protective ground
2	TD	Transmitted data
3	RD	Received data
4	RTS	Request to send
5	CTS	Clear to send
6	DSR	Data set ready
7	SG	Signal Ground
8	CD	Carrier detect
9	+	Voltage (testing)
10	-	Voltage (testing)
11		
12	SCD	Secondary CD
13	SCS	Secondary CTS
14	STD	Secondary TD
15	TC	Transmit Clock
16	SRD	Secondary RD
17	RS	Receiver clock
18		Ready to Send
19	SRS	Secondary RTS
20	DTR	Data Terminal Ready
21	SQD	Signal Quality Detector
22	RI	Ring Indicator
23	DRS	Data rate select
24	XTC	External Clock
25		

Figure 9.21: 25-pin D-type Connector Pin Assignment

Signal Description

TxD (Transmit Data): This pin carries data from the computer to the serial device

RXD (Receive Data): This pin carries data from the serial device to the computer

DTR signals (Data Terminal Ready): DTR is used by the computer to signal that it is ready to communicate with the serial device like modem.

DSR: Similarly to DTR, Data set ready (DSR) is an indication from the Dataset that it is ON.

DCD: Data Carrier Detect (DCD) indicates that carrier for the transmit data is ON.

RTS (Request to Send): This pin is used to request clearance to send data to a modem

CTS (Clear to Send): This pin is used by the serial device to acknowledge the computer's RTS Signal. In most situations, RTS and CTS are constantly on throughout the communication session.

Clock signals (TC, RC, and XTC): The clock signals are only used for synchronous communications. The modem or DSU extracts the clock from the data stream and provides a steady clock signal to the DTE. Note that the transmit and receive clock signals do not have to be the same, or even at the same baud rate.

CD: CD stands for Carrier Detect. Carrier Detect is used by a modem to signal that it has a made a connection with another modem, or has detected a carrier tone. In other words, this is used by the modem to signal that a carrier signal has been received from a remote modem.

RI: RI stands for Ring Indicator. A modem toggles (keystroke) the state of this line when an incoming call rings your phone. In other words, this is used by an auto answer modem to signal the receipt of a telephone ring signal

Note: The Carrier Detect (CD) and the Ring Indicator (RI) lines are only available in connections to a modem.

9.4.2 IEEE-488 Standard: IEEE-488 refers to the Institute of Electrical and Electronics Engineers (IEEE) Standard number 488. IEEE-488 was created as HP-IB (Hewlett-Packard Interface Bus), and is commonly called **GPIB**

(**General Purpose Interface Bus**). The IEEE-488, General Purpose Interface Bus (GPIB), is a general purpose digital interface system that can be used to transfer data between two or more devices. The GPIB system is a parallel communication system, which can communicate with several devices through the same interface port. The standard defines three types of devices that are connected to this bus:

- controller
- listener
- Talker

All of which are connected through the IEEE-488 connector. A Talker sends data messages to one or more Listeners, which receive the data. The Controller manages the flow of information on the GPIB by sending commands to all devices. A digital voltmeter, for example, is a Talker and is also a Listener. The GPIB or IEEE 488 bus is a very flexible system and it allows the data to flow between any of the instruments on the bus, at a speed suitable for the slowest active instrument. It is possible to purchase GPIB cards to incorporate into computers that do not have the interface fitted. As GPIB cards are relatively cheap, this makes the inclusion of a GPIB card into the system a very cost effect method of installing it.

The important key features are:

- Up to 15 devices may be connected to one bus.
- Total bus length may be up to 20 m and the distance between devices may be up to 2 m
- Communication is digital (as opposed to analog) and messages are sent one byte (8 bits) at a time.
- Data rates may be up to 1 Mbyte/sec

The connector used for the IEEE 488 bus is standardized as a 24-way Amphenol 57 series type. This provides an ideal physical interface for the standard. The IEEE 488 or GPIB connector is very similar in format to those that were used for parallel printer ports on PCs. Figure 9.22 shows the IEEE-488 Connector.

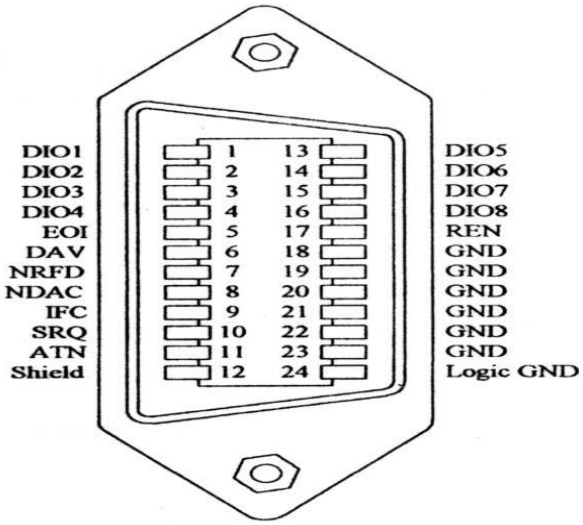


Figure 9.22: IEEE-488 Connector

The GPIB is like an ordinary computer bus, except that a computer has its circuit cards interconnected via a backplane -the GPIB has stand-alone devices interconnected by standard cables. The role of the GPIB Controller is comparable to the role of a computer CPU, but a better analogy is to compare the controller to the switching center of a city telephone system.

Self Assessment Questions

6. In _____ type of I/O interfacing the 8086 uses 20 address lines to identify an I/O device and an I/O device is connected as if it is a memory register.
7. _____ port is the interface used in microprocessor based system for connecting devices like printer, modems, CRT terminals.
8. RS232 standard is developed by the Electronic Industry Association (EIA) in 1962 and was revised and renamed as _____.
9. DTE stands for _____.
10. IEEE-488 was created as HP-IB (Hewlett-Packard Interface Bus), and is commonly called GPIB (General Purpose Interface Bus).(True/False)

9.4.3 8251 USART

The 8251 is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication. It is also called a programmable communications interface (PCI). The 8251 receives parallel data from the processor and transmits serial data after conversion. This device also receives serial data from the outside and transmits parallel data to the processor after conversion.

Main Features of 8251 are

1) Both Synchronous and Asynchronous operation
2) False start bit detection
3) Automatic break detect and handling
4) Error detection - parity, overrun and framing errors.
5) Break characters generation
6) It has built-in baud generator
7) Allows full duplex transmission and reception
8) It is compatible with extended range of Intel microprocessors
9) It is fabricated in 28 pin DIP package and all its inputs and outputs are TTL compatible.

Pin Description of 8251:

Intel 8251A or simply 8251 is a 28-pin programmable IC available as a DIP package. Its physical and functional pin diagrams are shown in figure 9.23 and 9.24 respectively.

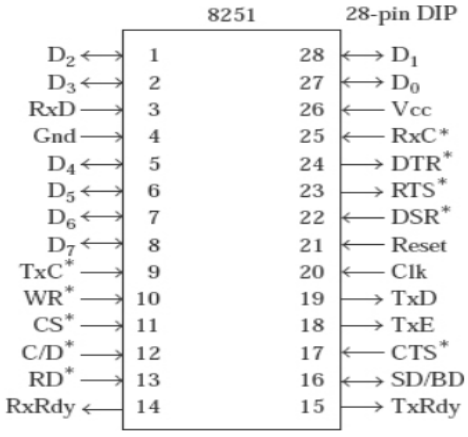


Figure: 9.23 Pin diagram of Intel 8251

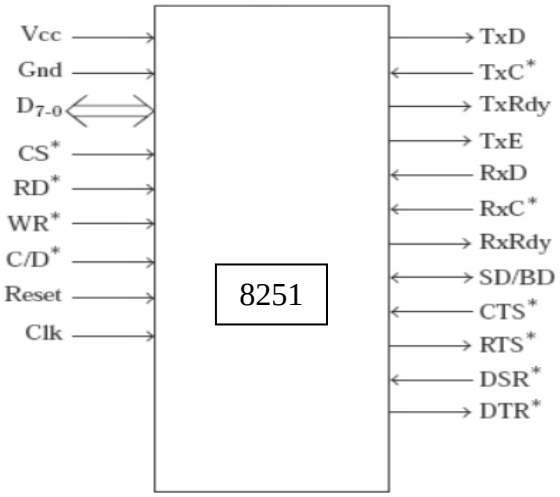


Figure: 9.24: Functional Pin diagram of Intel 8251

Vcc and Gnd: Power supply and ground pins. 8251 uses 15V power supply.

D 0 to D 7 (I/O terminal): This is bidirectional data bus which receives control words and transmits data from the CPU and sends status words and received data to CPU.

RESET: A "High" on this input forces the 8251 into "reset status." The device waits for the writing of "mode instruction." The min. reset width is six clock inputs during the operating status of CLK.

CLK (Clock): CLK signal is used to generate internal device timing. CLK signal is independent of RXC or TXC. However, the frequency of CLK must be greater than 30 times the RXC and TXC at Synchronous mode and Asynchronous "x1" mode, and must be greater than 5 times at Asynchronous "x16" and "x64" mode.

$\overline{WR}$ (or) **WR* (Write):** This is the "active low" input terminal which receives a signal for writing transmit data and control words from the CPU into the 8251.

$\overline{RD}$ (or) **RD* (Read):** This is the "active low" input terminal which receives a signal for reading receive data and status words from the 8251.

C/$\overline{D}$ -Control/ Data: This is an input terminal which receives a signal for selecting data or command words and status words when the 8251 is accessed by the CPU. If C/D = low, data will be accessed. If C/D = high, command word or status word will be accessed.

$\overline{CS}$ (or) **CS*** **(Chip Select)-:** This is the "active low" input terminal which selects the 8251 When this is low.

TXD (output terminal)
This is an output terminal for transmitting data from which serial-converted data is sent out. The device is in "mark status" (high level) after resetting or during a status when transmit is disabled. It is also possible to set the device in "break status" (low level) by a command.

TXRDY (Transmitter Ready): This is an output terminal which indicates that the 8251is ready to accept a transmitted data character. But the terminal is always at low level if CTS = high or the device was set in "TX disable status" by a command.

Note: TXRDY status word indicates that transmit data character is receivable, regardless of CTS or command. If the CPU writes a data character, TXRDY will be reset by the leading edge or WR signal.

TXE (Transmitter Empty): This is an output terminal which indicates that the 8251 has transmitted all the characters and had no data character. In "synchronous mode," the terminal is at high level, if transmit data characters are no longer remaining and sync characters are automatically transmitted. If the CPU writes a data character, TXE will be reset by the leading edge of WR signal. Note: As the transmitter is disabled by setting CTS "High" or command, data written before disable will be sent out. Then TXD and TXE will be "High". Even if a data is written after disable, that data is not sent out and TXE will be "High". After the transmitter is enabled, data will be sent out.

TXC (Transmitter Clock): This is a clock input signal which determines the transfer speed of transmitted data. In "synchronous mode," the baud rate will be the same as the frequency of TXC. In "asynchronous mode", it is possible to select the baud rate factor by mode instruction. It can be 1, 1/16 or 1/64 the TXC. The falling edge of TXC shifts the serial data out of the 8251.

RXD (Receive Data): This is a terminal which receives serial data.

RXRDY (Receiver Ready): This is a terminal which indicates that the 8251 contains a character that is ready to READ. If the CPU reads a data character, RXRDY will be reset by the leading edge of RD signal. Unless the CPU reads a data character before the next one is received completely, the preceding data will be lost. In such a case, an overrun error flag status word will be set.

$\overline{RXC}$ **(Receiver Clock):** This is a clock input signal which determines the transfer speed of received data. In "synchronous mode," the baud rate is the same as the frequency of RXC. In "asynchronous mode," it is possible to select the baud rate factor by mode instruction. It can be 1, 1/16, 1/64 the RXC.

SD/BD (Sync Detect/Break Detect): This is a terminal whose function changes according to mode. In "internal synchronous mode" this terminal is at high level, if sync characters are received and synchronized. If a status word is read, the terminal will be reset. In "external synchronous mode" this is an input terminal. A "High" on this input forces the 8251 to start receiving data characters.

In "asynchronous mode," this is an output terminal which generates "high level" output upon the detection of a "break" character if receiver data contains a "low-level" space between the stop bits of two continuous characters. The terminal will be reset, if RXD is at high level. After Reset is active, the terminal will be output at low level.

$\overline{DSR}$ **(Data Set Ready):** This is an input port for MODEM interface. The input status of the terminal can be recognized by the CPU reading status words.

$\overline{DTR}$ **(Data Terminal Ready):** This is an output port for MODEM interface. It is possible to set the status of DTR by a command.

$\overline{CTS}$ **(Clear to Send):** This is an input terminal for MODEM interface which is used for controlling a transmit circuit. The terminal controls data transmission if the device is set in "TX Enable" status by a command. Data is transmittable if the terminal is at low level.

$\overline{RTS}$ **(Request to Send):** This is an output port for MODEM interface. It is possible to set the status RTS by a command.

Block diagram of the 8251 USART is shown in figure 9.25.

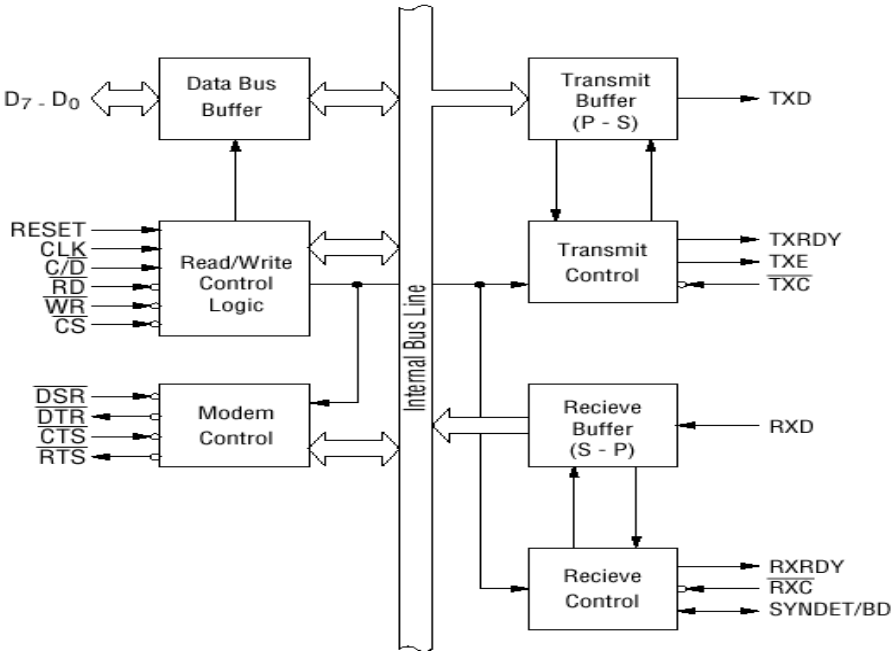


Figure 9.25: Block Diagram of the 8251 USART

The functional block diagram of 8251A consists five sections. They are: Data bus buffer, Read/Write control logic, Modem control, Transmitter Receiver

Data Bus Buffer: This is a tri-state, bi-directional 8 bit buffer used to interface 8251 to the system bus. Along with the data, control word, command words and status information are also transferred through the data bus buffer.

Read/Write control logic: The Read/Write Control logic interfaces the 8251A with CPU, determines the functions of the 8251A according to the control word written into its control register. It also monitors the data flow. This block has three registers: control register, status register and data buffer. The active low signals $\overline{RD}$, $\overline{WR}$, $\overline{CS}$ and C/$\overline{D}$ are used for read/write operations with these three registers. When C/$\overline{D}$ is high, the

control register is selected for writing control word or reading status word. When C/$\overline{D}$ is low, the data buffer is selected for read/write operation. When the reset is high, it forces 8251A into the idle mode. Here the clock input is necessary for 8251A for communication with CPU, but this clock does not control either the serial transmission or the reception rate.

MODEM Control: The MODEM control unit allows to interface a MODEM to 8251A and to establish data communication through MODEM over telephone lines. This block also takes care of handshake signals for MODEM interface. Operation between the 8251 and a CPU is executed by program control.

Transmitter section: This section accepts parallel data from CPU and converts them into serial data. This section is double buffered which means it has a buffer register to hold an 8-bit parallel data and another register called output register to convert the parallel data into serial bits. When output register is empty, the data is transferred from buffer to output register. Now the processor can again load another data in buffer register. If buffer register is empty, then TxRDY goes to high. If output register is empty then TxE goes to high. The clock signal, $\overline{TxC}$ controls the rate at which the bits are transmitted by the USART. The clock frequency can be 1,16 or 64 times the baud rate.

Receiver Section: This section accepts serial data and convert them into parallel data. The receiver section is also double buffered, i.e., it has an input register to receive serial data and convert to parallel, and a buffer register to hold the parallel data. When the RxD line goes low, the control logic assumes it as a START bit, waits for half a bit time and samples the line again. If the line is still low, then the input register accepts the following bits, forms a character and loads it into the buffer register. The processor reads the parallel data from the buffer register. When the input register loads a parallel data to buffer register, the RXRDY line goes high. The clock signal $\overline{RxC}$ controls the rate at which bits are received by the USART. During asynchronous mode, the signal SYNDET/BD will indicate the break in the data transmission. During synchronous mode, the signal SYNDET/BD will indicate the reception of synchronous character.

The 8251 functional configuration is programmed by software. Operation between the 8251 and a CPU is executed by program control. Table 9.6 shows the operation between a CPU and the device.

Table 9.6: Operation between a CPU and the device.

$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	
1	×	×	×	Data Bus 3-State
0	×	1	1	Data Bus 3-State
0	1	0	1	Status → CPU
0	1	1	0	Control Word ← CPU
0	0	0	1	Data → CPU
0	0	1	0	Data ← CPU

Programming the 8251

The complete functional definition of the 8251 is programmed by the system's software. A set of control words must be sent out by the CPU to initialize the 8251 to support the desired communication format. These words must immediately follow a reset (internal/external).

Control Words

There are two types of control word.

1. Mode instruction (setting of function)
2. Command (setting of operation)

1) Mode Instruction

Mode instruction is used for setting the function of the 8251. Mode instruction will be in "wait for write" at either internal reset or external reset. That is, the writing of a control word after resetting will be recognized as a "mode instruction."

Items set by mode instruction are as follows:

- Synchronous/asynchronous mode
- Stop bit length (asynchronous mode)
- Character length
- Parity bit
- Baud rate factor (asynchronous mode)

- Internal/external synchronization (synchronous mode)
- Number of synchronous characters (Synchronous mode)

The bit configuration of mode instruction is shown in figures 9.26 and 9.27. In the case of synchronous mode, it is necessary to write one-or two byte sync characters. If sync characters were written, a function will be set because the writing of sync characters constitutes part of mode instruction.

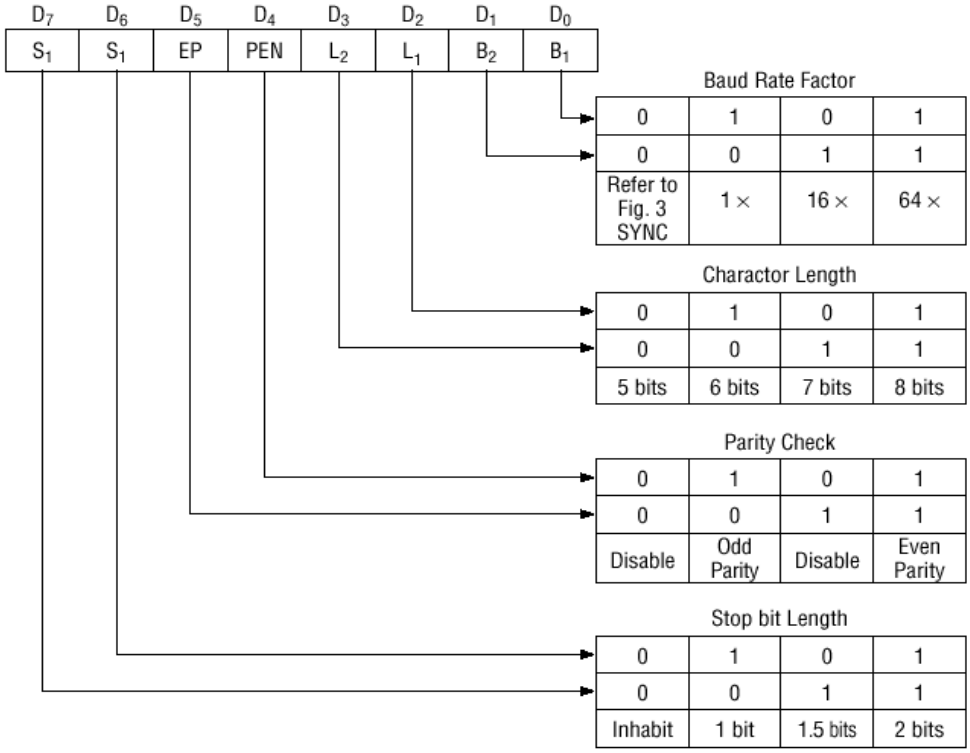


Figure 9.26: Bit configuration of Mode instruction (Asynchronous)

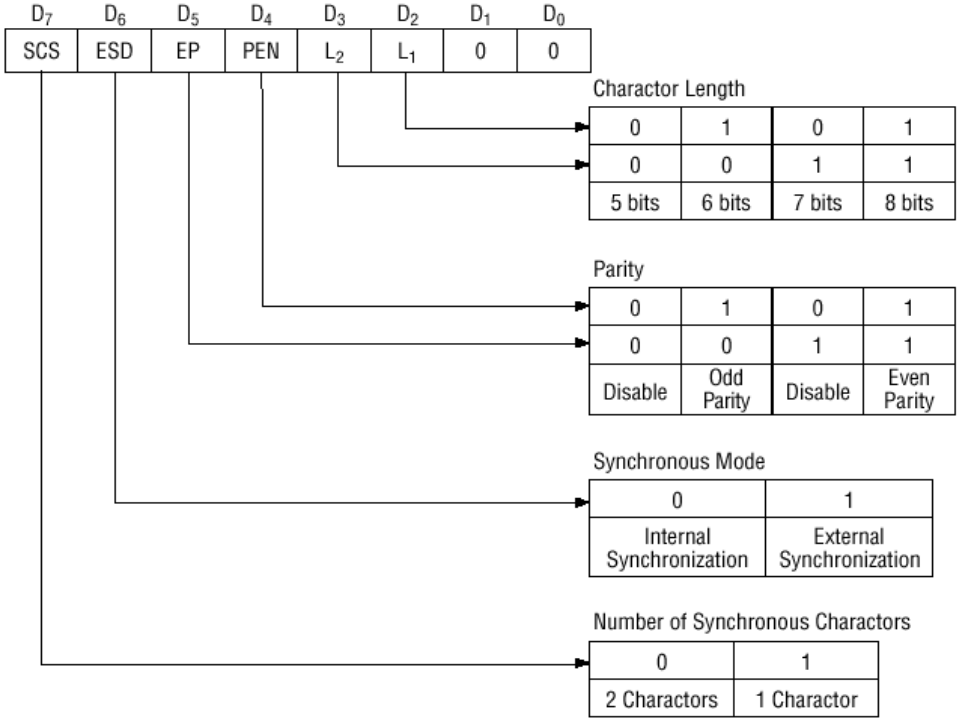


Figure 9.27: Bit configuration of Mode Instruction (Synchronous)

2) Command

Command is used for setting the operation of the 8251. It is possible to write a command whenever necessary after writing a mode instruction and sync characters.

Items to be set by command are as follows:

- Transmit Enable/Disable
- Receive Enable/Disable
- DTR, RTS Output of data.
- Resetting of error flag.
- Sending to break characters
- Internal resetting
- Hunt mode (synchronous mode)

Bit configuration of command is shown in figure 9.28.

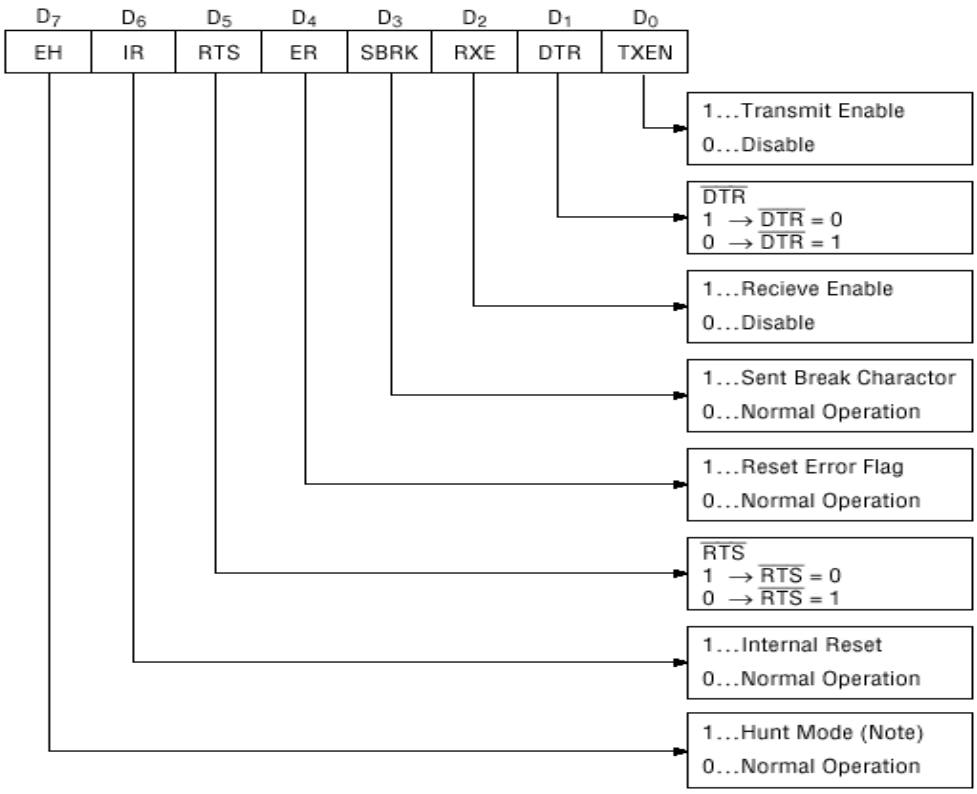


Figure 9.28: Bit configuration of Command

Status Word

The status word gives the status of the general operating information of 8251. It is possible to see the internal status of the 8251A by reading a status word. The bit configuration of status word is shown in figure 9.29

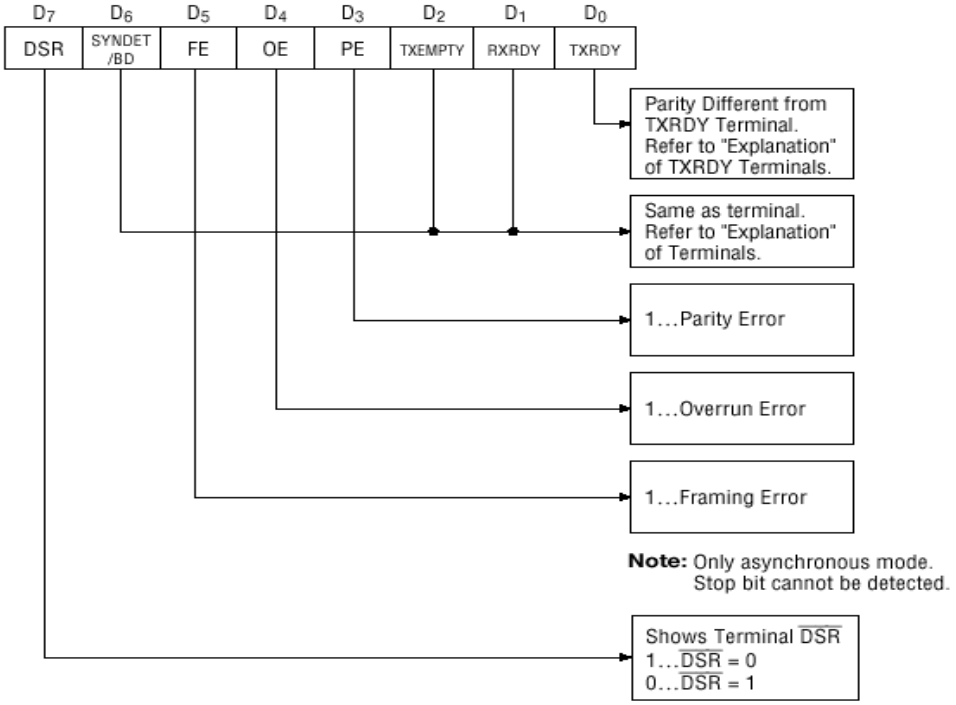


Figure 9.29: Bit configuration of Status Word

The 8251 as we know is a USART, so communication takes place in four different ways. They are: asynchronous transmission, asynchronous reception, synchronous transmission and synchronous reception. You can enable these communication modes by writing suitable mode and command instructions.

Interfacing 8251 to 8086 in Memory mapped I/O:

Figure 9.30 shows the Memory mapped I/O technique of interfacing of 8251 with 8086 which uses 20 bit address lines to identify an I/O device.

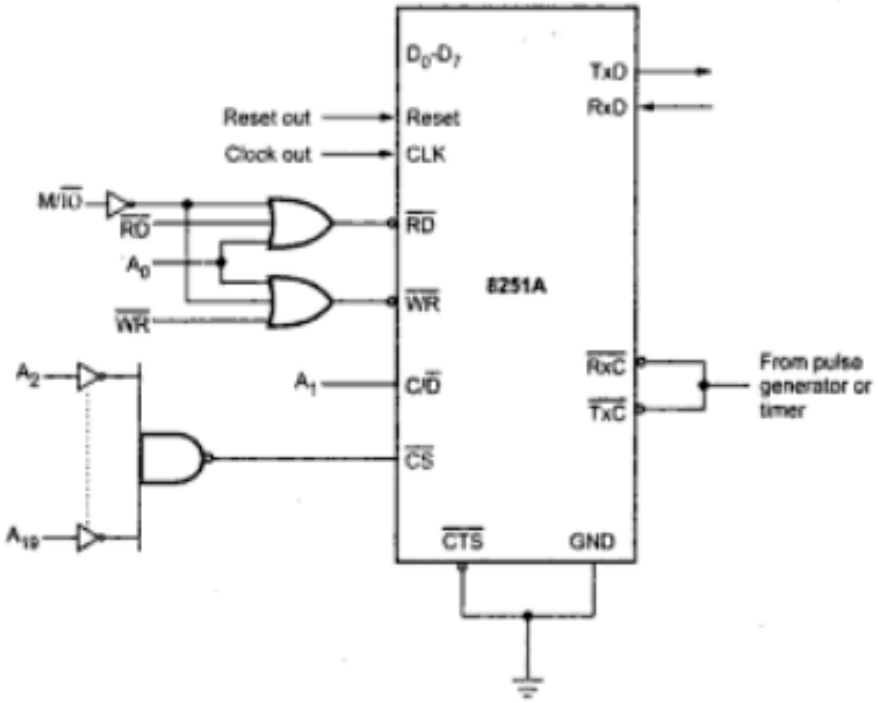


Figure 9.30: Interfacing 8251 with 8086 in Memory mapped I/O

8086 uses same Memory related instructions to access I/O devices. When M/$\overline{IO}$ signal is High, $\overline{RD}$ and $\overline{WR}$ signals get activated indicating memory bus cycle.

I/O Map:

Register	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Address
Data Register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
Control Register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	00002H

Example 3: Write an 8086 instruction sequence for transmitting 50 characters which are stored from the location 2010H using 8251.

Solution: Assume 8251 as asynchronous mode Even parity, 2 stop bits, 8 bit char length, frequency 160 kHz, and baud rate 10k D7 D6 D5 D4 D3 D2 D1 D0 1 1 1 1 1 1 1 0

```
ASSUME CS: CODE
CODE SEGMENT
START: MOV AX, 2010H
       MOV DS, AX          ; DS points to byte string segment
       MOV SI, 5000H       ; SI points to byte string charter
       MOV CL, 50H         ; length of string
       MOV AL, 0FEH        ; mode control word out to
       OUT 0FEH, AL        ; D0-D7
       MOV AX, 11H         ; load command word
       OUT 0FEH, AL        ; to transmit enable and error reset
WAIT: IN AL, 0FEH          ; read status
       AND AL, 01H         ; check transmitter enable
       JZ WAIT             ; if zero wait for transmitter is to be ready
       MOV AL, [SI]        ; if ready first charter is transmitted
       OUT 0FCH, AL
       INC SI              ; point to next byte or charter
       DEC CL              ; decrement counter
       JNZ WAIT            ; if CL not zero, go for next byte
       MOV AH, 4CH
       INT21H CODE ENDS
END START
```

Example 4: Design the hardware interface circuit for interfacing 8251 with 8086.Set the 8251in synchronous mode as a transmitter and receiver with even parity enabled, 2 stop bits,8-bit character length, frequency 160 kHz and baud rate 10k.

(a) Write an ALP to transmit 100 bytes of data string at location 2000:5000H

(b) Write an ALP to receive 100 bytes of data string and store it at the location 3000:4000H.

Solution: The figure 9.31 shows the Interfacing of 8251 with 8086

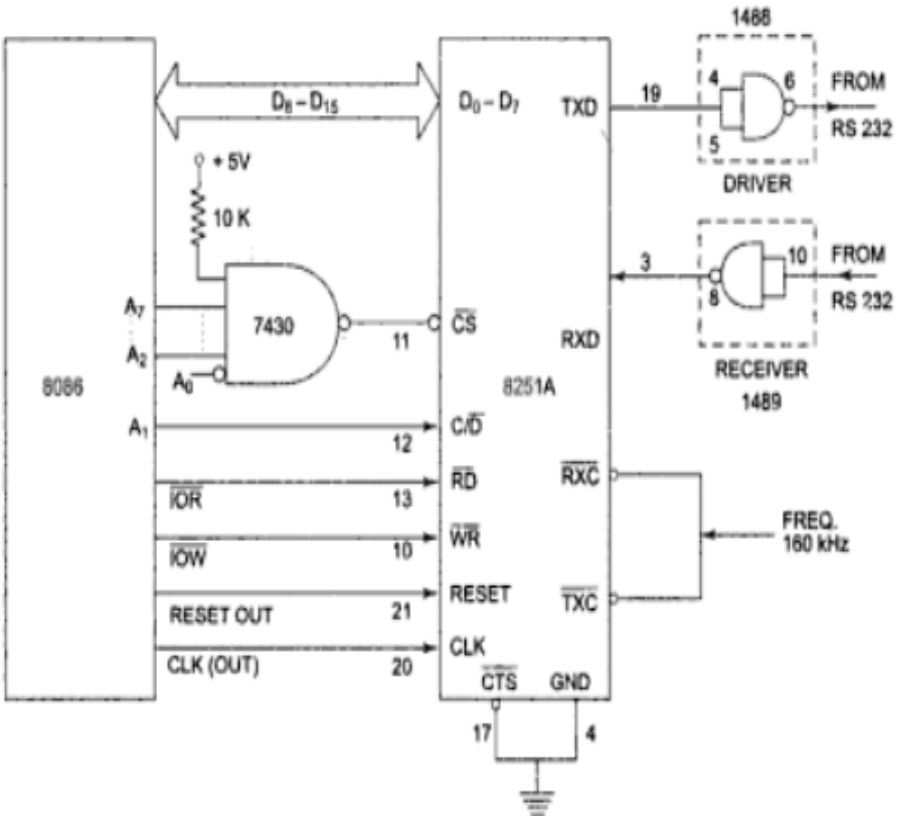


Figure 9.31: Interfacing of 8251 with 8086.

(a) Asynchronous mode control word for problem (a) is:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	1	1	1	1	1	1	0	= 0FE H
2 stop bits		Even parity enabled		8-bit format		CLK scaled by 16		

ALP to initialize 8251 and transmit 100 bytes of data is:

```
ASSUME      CS : CODE
CODE        SEGMENT
START:      MOV AX,2000H      ;
            MOV DS,AX         ; DS points to byte string segment

            MOV CL,64H        ; length of the string in CL(hex)
            MOV AL,0FEH       ; Mode control word out to
            OUT 0FEH,AL       ; D0-D7.
            MOV AX,11H        ; Load command word
            OUT 0FEH,AL       ; to transmit enable and error reset
WAIT:       IN AL,0FEH        ; Read status,
            AND AL,01H        ; check transmitter enable
            JZ WAIT           ; bit,if zero wait for the transmitter to be ready
            MOV AL,[SI]       ; If ready,first byte of string data
            OUT 0FCH,AL       ; is transmitted.
            INC SI            ; Point to next byte.
            DEC CL            ; Decrement counter.
            JNZ WAIT          ; If CL is not zero, go for next byte.
            MOV AH,4CH        ; If CX is zero,return to DOS
            INT 21H
CODE        ENDS
            END START
```

(b) The command word for the problem (b) can be calculated as 14H. ALP to initialize 8251 and receive 100 bytes is:

```
ASSUME    CS : CODE
CODE      SEGMENT
START:    MOV AX,3000H      ;
          MOV DS,AX         ; Data segment set to 3000H
          MOV SI,4000H      ; Pointer to destination offset
          MOV CL,64H        ; Byte count in CL
          MOV AL,7EH        ; Only one stop bit for
          OUT 0FEH,AL       ; receiver is set
          MOV AL,14H        ; Load command word to enable
          OUT 0FEH,AL       ; the receiver and disable transmitter
NXTBT:    IN AL,0FEH        ; Read status
          AND 38H           ; Check FE, OE and PE,
          JZ READY          ; If zero,jump to READY
          MOV AL,14H        ; If not zero ,clear them
          OUT 0FEH,AL       ;

READY:    IN AL,0FEH        ; Check RXRDY.If the
          AND 02H           ; receiver is not ready,
          JZ READY          ; wait
          IN AL,0FCH        ; If it is ready,
          MOV [SI],AL       ; receive the character
          INC SI            ; Increment pointer to next byte
          DEC CL            ; Decrement counter
          JNZ NXTBT         ; Repeat,if CL is not zero
          MOV AH,4CH        ; If CL is 0, return to DOS
          INT 21H
CODE      ENDS
          END START
```

Self Assessment Questions

11. The _____ is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication.
12. _____ is an output terminal which indicates that the 8251is ready to accept a transmitted data character.
13. Which pin indicates that the 8251 has transmitted all the characters and had no data character?

    (a) TXEMPTY (b) TEPTY (c)TXC (d) RXRDY
14. In 8251 there are four types of command words. (True or false).

9.5 Summary

Let us recapitulate the important concepts discussed in this unit:

- The 8254 is a programmable interval timer/counter designed for use with Intel microcomputer systems.
- The 8253/54 has three identical 16 bit counters and you can operate them independently.
- Each counter of the 8253/54 can be programmed individually by writing a control word into the control word register.
- Counter can be programmed in six different modes. MODE 2 is called Rate generator. This mode functions like a divide by-N counter.
- Serial data transmission is preferred for long distances.
- Synchronous transmission uses no start and stop bits but instead synchronizes transmission speeds at both the receiving and sending end of the transmission using clock signal(s).
- The 8251 is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication.
- RS232C is a standard hardware interface used for implementing asynchronous serial data communication ports on devices such as CRT terminals, printers, modems and keyboards.
- IEEE-488 was created as HP-IB (Hewlett-Packard Interface Bus), and is commonly called GPIB (General Purpose Interface Bus

9.6 Terminal Questions

1. List out the important features of 8253/54 programmable interval timer.
2. Explain the functions of various blocks of functional block diagram of 8253/54 programmable interval timers.
3. Explain how 8253/54 is interfaced to 8086 using memory mapped I/O technique.
4. Write a note on (a) RS 232 standard (b) IEEE 488 standard.
5. Discuss about 8251 USART.

9.7 Answers

Self Assessment Questions

1. 8254
2. Three
3. True

4. Mode 3
5. Mode 5
6. Memory mapped I/O
7. Serial communication
8. RS232C
9. Data Terminal Equipment
10. True
11. 8251
12. TXRDY
13. (a) TXEMPTY
14. False

Terminal Questions

1. The 8253/54 has three identical 16 bit counters. Refer to section 9.2 for details.
2. The block diagram of 8253/54 includes three counters, a data bus buffer, Read/Write control logic, and a control register. Each counter has two input signals CLOCK and GATE and one output signal OUT. Refer to sub-section 9.2.2 for more details.
3. In memory mapped I/O type of interfacing, the 8086 uses 20 address lines to identify an I/O device and an I/O device is connected as if it is a memory register. Refer to sub-section 9.3.1 for more details.
4. RS232 standard is developed by the Electronic Industry Association (EIA) in 1962 and was revised and renamed as RS232C. IEEE-488 was created as HP-IB (Hewlett-Packard Interface Bus), and is commonly called GPIB (General Purpose Interface Bus).Refer to sub-sections 9.4.1 and 9.4.2 for more details.
5. The 8251 is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication. Refer to sub-section 9.4.3 for more details.