**Synchronous vs. Asynchronous**

* **Synchronous data transfer:** 
  + sender and receiver use the same clock signal
  + supports high data transfer rate
  + needs clock signal between the sender and the receiver
  + requires master/slave configuration
* **Asynchronous data transfer:** 
  + Sender provides a synchronization signal to the receiver before starting the transfer of each message
  + does not need clock signal between the sender and the receiver
  + slower data transfer rate

**Notes:**

There are many serial data transfer protocols. The protocols for serial data transfer can be grouped into two types: synchronous and asynchronous. For synchronous data transfer, both the sender and receiver access the data according to the same clock. Therefore, a special line for the clock signal is required. A master (or one of the senders) should provide the clock signal to all the receivers in the synchronous data transfer.   
  
For asynchronous data transfer, there is no common clock signal between the sender and receivers. Therefore, the sender and the receiver first need to agree on a data transfer speed. This speed usually does not change after the data transfer starts. Both the sender and receiver set up their own internal circuits to make sure that the data accessing is follows that agreement. However, just like some watches run faster than others, computer clocks also differ in accuracy. Although the difference is very small, it can accumulate fast and eventually cause errors in data transfer. This problem is solved by adding synchronization bits at the front, middle or end of the data. Since the synchronization is done periodically, the receiver can correct the clock accumulation error. The synchronization information may be added to every byte of data or to every frame of data. Sending these extra synchronization bits may account for up to 50% data transfer overhead and hence slows down the actual data transfer rate.

The Intel 8253 and 8254 are Programmable Interval Timers (PTIs) designed for microprocessors to perform timing and counting functions using three 16-bit registers. Each counter has 2 input pins, i.e. Clock & Gate, and 1 pin for “OUT” output. To operate a counter, a 16-bit count is loaded in its register. On command, it begins to decrement the count until it reaches 0, then it generates a pulse that can be used to interrupt the CPU.

**Difference between 8253 and 8254**

The following table differentiates the features of 8253 and 8254

|  |  |
| --- | --- |
| **8253** | **8254** |
| Its operating frequency is 0 - 2.6 MHz | Its operating frequency is 0 - 10 MHz |
| It uses N-MOS technology | It uses H-MOS technology |
| Read-Back command is not available | Read-Back command is available |
| Reads and writes of the same counter cannot be interleaved. | Reads and writes of the same counter can be interleaved. |

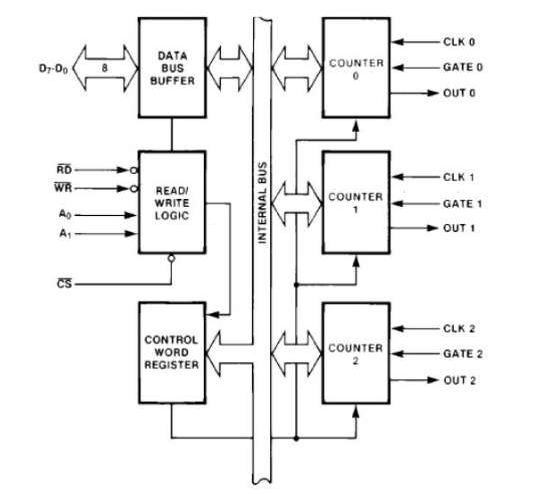
**Features of 8253 / 54**

The most prominent features of 8253/54 are as follows −

* It has three independent 16-bit down counters.
* It can handle inputs from DC to 10 MHz.
* These three counters can be programmed for either binary or BCD count.
* It is compatible with almost all microprocessors.
* 8254 has a powerful command called READ BACK command, which allows the user to check the count value, the programmed mode, the current mode, and the current status of the counter.

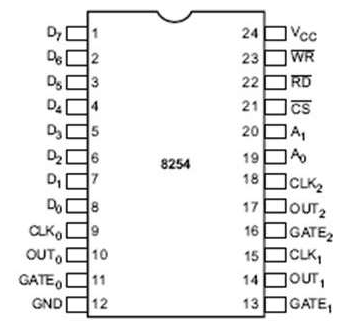
**8254 Architecture**

The architecture of 8254 looks as follows –



**8254 Pin Description**

Here is the pin diagram of 8254



In the above figure, there are three counters, a data bus buffer, Read/Write control logic, and a control register. Each counter has two input signals - CLOCK & GATE, and one output signal - OUT.

**Data Bus Buffer**

It is a tri-state, bi-directional, 8-bit buffer, which is used to interface the 8253/54 to the system data bus. It has three basic functions −

* Programming the modes of 8253/54.
* Loading the count registers.
* Reading the count values.

**Read/Write Logic**

It includes 5 signals, i.e. RD, WR, CS, and the address lines A0 & A1. In the peripheral I/O mode, the RD and WR signals are connected to IOR and IOW, respectively. In the memorymapped I/O mode, these are connected to MEMR and MEMW.

Address lines A0 & A1 of the CPU are connected to lines A0 and A1 of the 8253/54, and CS is tied to a decoded address. The control word register and counters are selected according to the signals on lines A0 & A1.

|  |  |  |
| --- | --- | --- |
| **A1** | **A0** | **Result** |
| 0 | 0 | Counter 0 |
| 0 | 1 | Counter 1 |
| 1 | 0 | Counter 2 |
| 1 | 1 | Control Word Register |
| X | X | No Selection |

**Control Word Register**

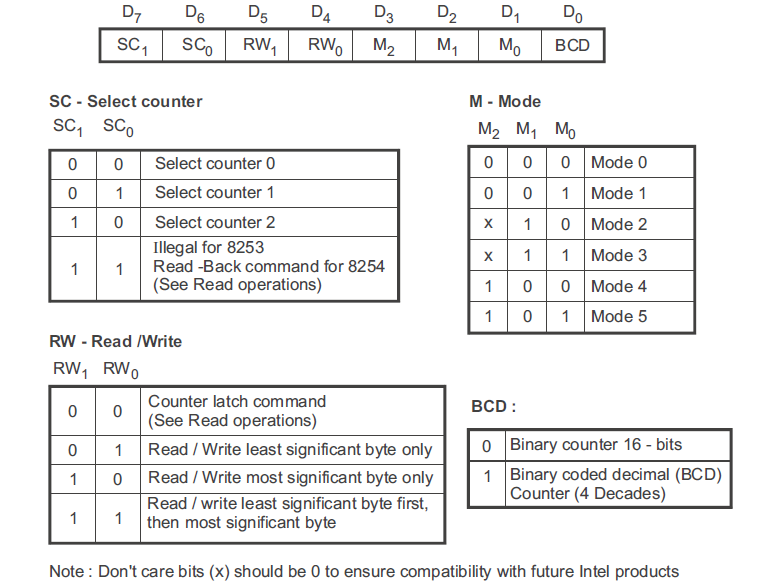
This register is accessed when lines A0 & A1 are at logic 1. It is used to write a command word, which specifies the counter to be used, its mode, and either a read or write operation. Following table shows the result for various control inputs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A1** | **A0** | **RD** | **WR** | **CS** | **Result** |
| 0 | 0 | 1 | 0 | 0 | Write Counter 0 |
| 0 | 1 | 1 | 0 | 0 | Write Counter 1 |
| 1 | 0 | 1 | 0 | 0 | Write Counter 2 |
| 1 | 1 | 1 | 0 | 0 | Write Control Word |
| 0 | 0 | 0 | 1 | 0 | Read Counter 0 |
| 0 | 1 | 0 | 1 | 0 | Read Counter 1 |
| 1 | 0 | 0 | 1 | 0 | Read Counter 2 |
| 1 | 1 | 0 | 1 | 0 | No operation |
| X | X | 1 | 1 | 0 | No operation |
| X | X | X | X | 1 | No operation |

**WRITE Operation :**

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.

**Fig. Control word format**

**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 CPU. 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 the second 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 CPU. 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)** : The third 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). Fig. 9.4 shows the format of the control word register for Read-Back command.

**Counters**

Each counter consists of a single, 16 bit-down counter, which can be operated in either binary or BCD. Its input and output is configured by the selection of modes stored in the control word register. The programmer can read the contents of any of the three counters without disturbing the actual count in process.

8253/54 can be operated in 6 different modes. In this chapter, we will discuss these operational modes.

## Mode 0 ─ Interrupt on Terminal Count

* It is used to generate an interrupt to the microprocessor after a certain interval.
* Initially the output is low after the mode is set. The output remains LOW after the count value is loaded into the counter.
* The process of decrementing the counter continues till the terminal count is reached, i.e., the count become zero and the output goes HIGH and will remain high until it reloads a new count.
* The GATE signal is high for normal counting. When GATE goes low, counting is terminated and the current count is latched till the GATE goes high again.

## Mode 1 – Programmable One Shot

* It can be used as a mono stable multi-vibrator.
* The gate input is used as a trigger input in this mode.
* The output remains high until the count is loaded and a trigger is applied.

## Mode 2 – Rate Generator

* The output is normally high after initialization.
* Whenever the count becomes zero, another low pulse is generated at the output and the counter will be reloaded.

## Mode 3 – Square Wave Generator

* This mode is similar to Mode 2 except the output remains low for half of the timer period and high for the other half of the period.

## Mode 4 − Software Triggered Mode

* In this mode, the output will remain high until the timer has counted to zero, at which point the output will pulse low and then go high again.
* The count is latched when the GATE signal goes LOW.
* On the terminal count, the output goes low for one clock cycle then goes HIGH. This low pulse can be used as a strobe.

## Mode 5 – Hardware Triggered Mode

* This mode generates a strobe in response to an externally generated signal.
* This mode is similar to mode 4 except that the counting is initiated by a signal at the gate input, which means it is hardware triggered instead of software triggered.
* After it is initialized, the output goes high.
* When the terminal count is reached, the output goes low for one clock cycle.
* **INTEL 8255: (Programmable Peripheral Interface)**
* The 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It consists of three 8-bit bidirectional I/O ports (24I/O lines) that can be configured to meet different system I/O needs. The three ports are PORT A, PORT B & PORT C. Port A contains one 8-bit output latch/buffer and one 8-bit input buffer. Port B is same as PORT A or PORT B. However, PORT C can be split into two parts PORT C lower (PC0-PC3) and PORT C upper (PC7-PC4) by the control word. The three ports are divided in two groups Group A (PORT A and upper PORT C) Group B (PORT B and lower PORT C). The two groups can be programmed in three different modes. In the first mode (mode 0), each group may be programmed in either input mode or output mode (PORT A, PORT B, PORT C lower, PORT C upper). In mode 1, the second’s mode, each group may be programmed to have 8-lines of input or output (PORT A or PORT B) of the remaining 4-lines (PORT C lower or PORT C upper) 3-lines are used for hand shaking and interrupt control signals. The third mode of operation (mode 2) is a bidirectional bus mode which uses 8-line (PORT A only for a bidirectional bus and five lines (PORT C upper 4 lines and borrowing one from other group) for handshaking.
* The block diagram is shown below:
* 
* 
* **Functional Description:**
* This support chip is a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. It is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.
* **Data Bus Buffer:**
* It is a tri-state 8-bit buffer used to interface the chip to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer. The data lines are connected to BDB of micro processor
* **Read/Write and logic control:**
* The function of this block is to control the internal operation of the device and to control the transfer of data and control or status words. It accepts inputs from the CPU address and control buses and in turn issues command to both the control groups.
* **Chip Select:**
* A low on this input selects the chip and enables the communication between the 8255 A & the CPU. It is connected to the output of address decode circuitry to select the device when it (Read). A low on this input enables the 8255 to send the data or status information to the CPU on the data bus.
* **(Write):**
* A low on this input pin enables the CPU to write data or control words
* into the 8255 A.
* **A1, A0 port select:**
* These input signals, in conjunction with the and inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A0 and A1). Following Table gives the basic operation,
* Following Table gives the basic operation,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A1 | A0 |  |  |  | Input operation |
| 0 | 0 | 0 | 1 | 0 | PORT A Data bus |
| 0 | 1 | 0 | 1 | 0 | PORT B Data bus |
| 1 | 0 | 0 | 1 | 0 | PORT C Data bus |
| 0 | 0 | 1 | 0 | 0 | Output operation  Data bus PORT A |
| 0 | 1 | 1 | 0 | 0 | Data bus PORT B |
| 1 | 0 | 1 | 0 | 0 | Data bus PORT C |
| 1 | 1 | 1 | 0 | 0 | Data bus control |

* All other states put data bus into tri-state/illegal condition.
* **RESET:**
* A high on this input pin clears the control register and all ports (A, B & C) are initialized to input mode. This is connected to RESET OUT of 8255. This is done to prevent destruction of circuitry connected to port lines. If port lines are initialized as output after a power up or reset, the port might try to output into the output of a device connected to same inputs might destroy one or both of them.
* **PORTs A, B and C:**
* The 8255A contains three 8-bit ports (A, B and C). All can be configured in a variety of functional characteristic by the system software.
* **PORTA:**
* One 8-bit data output latch/buffer and one 8-bit data input latch.
* **PORT B:**
* One 8-bit data output latch/buffer and one 8-bit data input buffer.
* **PORT C:**
* One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signals inputs in conjunction with ports A and B.
* **Group A & Group B control:**
* The functional configuration of each port is programmed by the system software. The control words outputted by the CPU configure
* the associated ports of the each of the two groups. Each control block
* accepts command from Read/Write content logic receives control words from the internal data bus and issues proper commands to its associated ports.
* Control Group A – Port A & Port C upper
* Control Group B – Port B & Port C lower
* The control word register can only be written into No read operation if
* the control word register is allowed.
* **Operation Description:**
* **Mode selection:**
* There are three basic modes of operation that can be selected by the
* system software.
* Mode 0: Basic Input/output
* Mode 1: Strobes Input/output
* Mode 2: Bi-direction bus.
* When the reset input goes HIGH all poets are set to mode’0’ as input which means all 24 lines are in high impedance state and can be used as normal input. After the reset is removed the 8255A remains in the input mode with no additional initialization. During the execution of the program any of the other modes may be selected using a single output instruction.
* The modes for PORT A & PORT B can be separately defined, while PORT C is divided into two portions as required by the PORT A and PORT B definitions. The ports are thus divided into two groups Group A & Group B. All the output register, including the status flip-flop will be reset whenever the mode is changed. Modes of the two group may be combined for any desired I/O operation e.g. Group A in mode ‘1’ and group B in mode ‘0’. The basic mode definitions with bus interface and the mode definition format are given in fig (a) & (b),
* 