

Programmable Interrupt Controller - 8259A

Interrupts can be used for a variety of applications. Each of these interrupt applications requires a separate interrupt input. If we are working with an 8085 microprocessor we get TRAP, RST7.5, RST6.5, RST5.5, and INTR interrupt inputs. For applications where we have multiple interrupt sources (more than five), we have to use external device called a Priority Interrupt Controller (PIC). By connecting such a device it is possible to increase the interrupt handling capacity of the microprocessor. Fig. 10.1 shows the connection between 8085 and 8259A. The 8259A is a commonly used priority interrupt controller, which is specifically designed for use with interrupt signals INTR and INTA of Intel series. It is packaged in a 28 pin DIP. It uses NMOS technology and requires a single + 5 V supply.

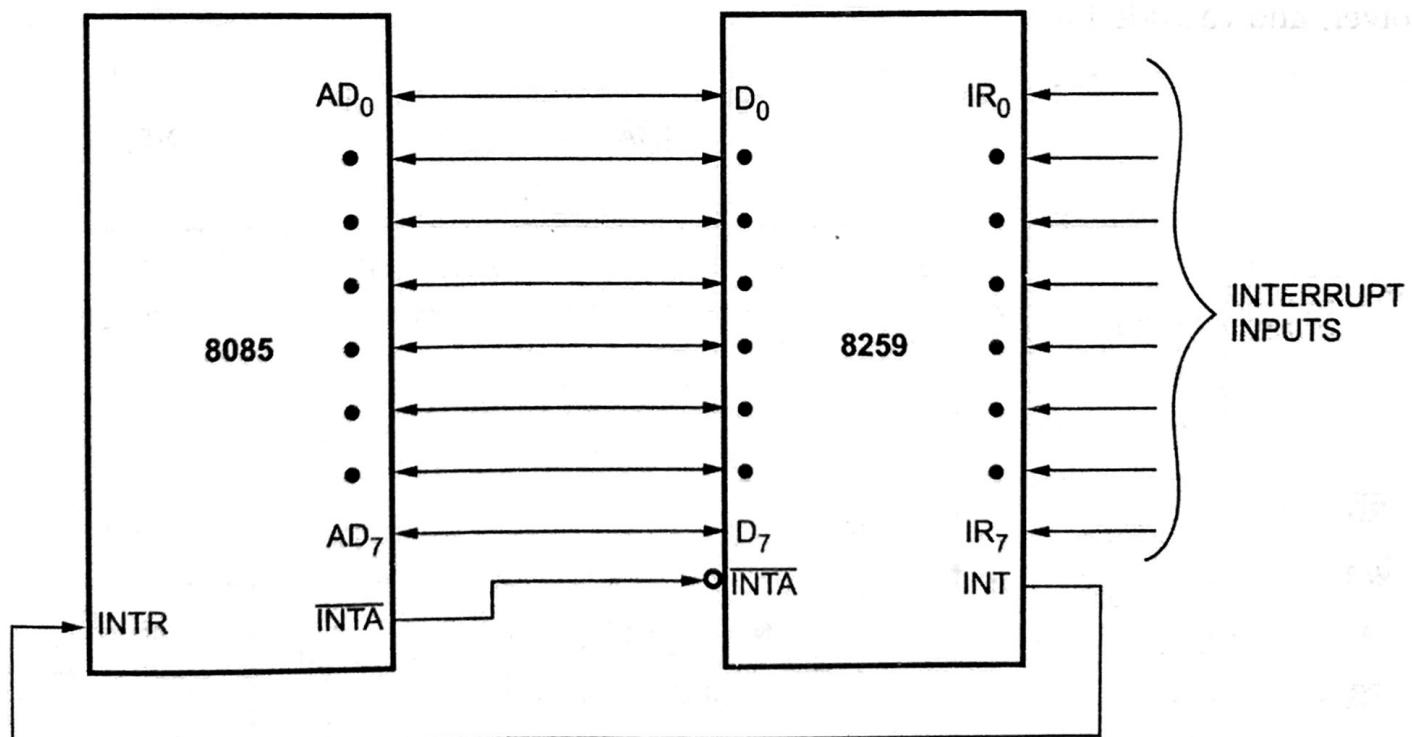


Fig. 10.1 Connection between 8085 and 8259A

10.1 Features of 8259A

1. It can manage eight priority interrupts. This is equivalent to providing eight interrupt pins on the processor in place of INTR pin.
2. It is possible to locate vector table for these additional interrupts anywhere in the memory map. However, all eight interrupts are spaced at the interval of either four or eight locations.
3. By cascading nine 8259s it is possible to get 64 priority interrupts.
4. Interrupt mask register makes it possible to mask individual interrupt request.
5. The 8259A can be programmed to accept either the level triggered or the edge triggered interrupt request.
6. With the help of 8259A user can get the information of pending interrupts, in-service interrupts and masked interrupts.
7. The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts.

10.2 Block Diagram of 8259A

Fig. 10.2 shows the internal block diagram of the 8259A. It includes eight blocks : data bus buffer, read/write logic, control logic, three registers (IRR, ISR and IMR), priority resolver, and cascade buffer.

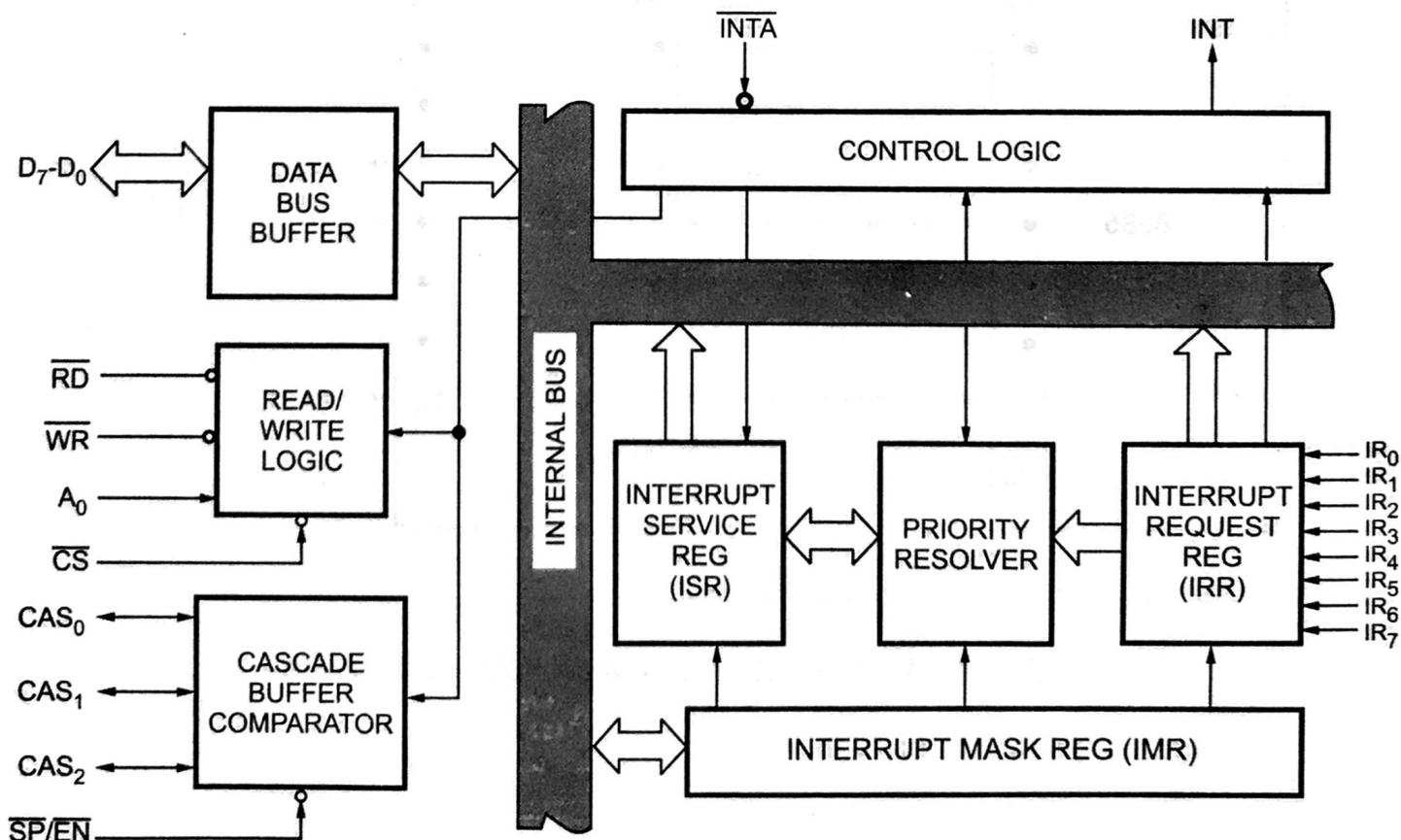


Fig. 10.2 Block diagram of 8259A

Data Bus Buffer

The data bus buffer allows the 8085 to send control words to the 8259A and read a status word from the 8259A. The 8-bit data bus buffer also allows the 8259A to send interrupt opcode and address of the interrupt service subroutine to the 8085.

Read/Write Logic

The \overline{RD} and \overline{WR} inputs control the data flow on the data bus when the device is selected by asserting its chip select (\overline{CS}) input low.

Control Logic

This block has an input and an output line. If the 8259A is properly enabled, the interrupt request will cause the 8259A to assert its INT output pin high. If this pin is connected to the INTR pin of an 8085 and if the 8085 Interrupt Enable (IE) flag is set, then this high signal will cause the 8085 to respond INTR as explained earlier.

Interrupt Request Register (IRR)

The IRR is used to store all the interrupt levels which are requesting the service. The eight interrupt inputs set corresponding bits of the Interrupt Request Register upon service request.

Interrupt Service Register (ISR)

The Interrupt Service Register (ISR) stores all the levels that are currently being serviced.

Interrupt Mask Register (IMR)

Interrupt Mask Register (IMR) stores the masking bits of the interrupt lines to be masked. This register can be programmed by an Operation Command Word (OCW). An interrupt which is masked by software will not be recognised and serviced even if it sets the corresponding bits in the IRR.

Priority Resolver

The priority resolver determines the priorities of the bits set in the IRR. The bit corresponding to the highest priority interrupt input is set in the ISR during the \overline{INTA} input.

Cascade Buffer Comparator

This section generates control signals necessary for cascade operations. It also generates Buffer-Enable signals. As stated earlier, the 8259 can be cascaded with other 8259s in order to expand the interrupt handling capacity to sixty-four levels. In such a case, the former is called a **master**, and the latter are called **slaves**. The 8259 can be set up as a master or a slave by the $\overline{SP}/\overline{EN}$ pin.

$CAS_0 - CAS_2$

For a master 8259, the CAS_0 - CAS_2 pins are output pins, and for slave 8259s, these are input pins. When the 8259 is a master (that is, when it accepts interrupt requests from other 8259s), the CALL opcode is generated by the Master in response to the first INTA. The vector address must be released by the slave 8259. The master sends an identification code of three-bits to select one out of the eight possible slave 8259s on the CAS_0 - CAS_2 lines. The slave 8259s accept these three signals as inputs (on their CAS_0 - CAS_2 pins) and compare the code sent by the master with the codes assigned to them during initialisation. The slave thus selected (which had originally placed an interrupt request to the master 8259) then puts the address of the interrupt service routine during the second and third INTA pulses from the MPU.

SP / EN (Slave Program /Enable Buffer)

The $\overline{\text{SP}}/\overline{\text{EN}}$ signal is tied high for the master. However it is grounded for the slave.

In large systems where buffers are used to drive the data bus, the data sent by the 8259 in response to INTA cannot be accessed by the MPU (due to the data bus buffer being disabled). If an 8259 is used in the buffered mode (buffered or non-buffered modes of operation can be specified at the time of initialising the 8259), the $\overline{\text{SP}}/\overline{\text{EN}}$ pin is used as an output which can be used to enable the system data bus buffer whenever the data bus outputs of 8259 are enabled (i.e. when it is ready to send data).

Thus, in non-buffered mode, the $\overline{\text{SP}}/\overline{\text{EN}}$ pin of an 8259 is used to specify whether the 8259 is to operate as a master or as a slave, and in the buffered mode, the $\overline{\text{SP}}/\overline{\text{EN}}$ pin is used as an output to enable the data bus buffer of the system.

10.3 Interrupt Sequence

The events occur as follows in an 8085 system :

1. One or more of the INTERRUPT REQUEST lines (IR_0 - IR_7) are raised high, setting the corresponding IRR bit(s).
2. The priority resolver checks three registers : The IRR for interrupt requests, the IMR for masking bits, and the ISR for the interrupt request being served. It resolves the priority and sets the INT high when it is appropriate to do so.
3. In response to the INTR signal, 8085 completes current instruction cycle and executes interrupt acknowledge cycle, thus giving an INTA pulse.
4. Upon receiving an INTA from the 8085, the highest priority ISR bit is set and the corresponding IRR bit is reset. Then 8259A places the opcode for CALL instruction on the data bus.
5. This CALL instruction initiates two more interrupt acknowledge cycles.
6. These two interrupt acknowledge cycles allow the 8259 to release preprogrammed subroutine address onto the data bus. In response to second interrupt acknowledge pulse, 8259 places a lower byte of interrupt subroutine address and in response to

third interrupt acknowledge pulse 8259 places a higher byte of the subroutine address.

7. This completes the interrupt cycle. In the AEOI (Automatic End of Interrupt) mode the ISR bit is reset at the end of the second INTA pulse. Otherwise, the ISR bit remains set until the issue of an appropriate EOI command at the end of the interrupt subroutine.

10.4 Priority Modes and Other Features

The various modes of operation of the 8259 are : 1. Fully Nested Mode, 2. Special Fully Nested Mode (SFNM) 3. Rotating Priority Mode, 4. Special Masked Mode, and 5. Polled Mode.

1. Fully Nested Mode (FNM) :

After initialization, the 8259A operates in fully nested mode so it is called **default mode**. The 8259 continues to operate in the Fully Nested Mode until the mode is changed through Operation Command Words. In this mode, IR₀ has highest priority and IR₇ has lowest priority. When the interrupt is acknowledged, it sets the corresponding bit in ISR. This bit will inhibit all interrupts of the **same or lower level**, however it will accept higher priority interrupt requests. The vector address corresponding to this interrupt is then sent. The bit in the ISR will remain set until an EOI command is issued by the microprocessor at the end of interrupt service routine. But if AEOI (Automatic End Of Interrupt) bit is set, the bit in the ISR resets at the trailing edge of the last INTA .

End of Interrupt (EOI)

1. The ISR bit can be reset by an End of Interrupt command issued by the MPU, usually just before exiting from the interrupt routine.
2. In the Fully Nested Mode, the highest level in the ISR would necessarily correspond to the last interrupt acknowledged and serviced. In such a case, a non-specific EOI command may be issued by the MPU.
3. However, if the FNM is not used, the 8259 may not be able to determine the last interrupt acknowledged. In such a case, a specific EOI command will have to be issued by the MPU.
4. It should be noted that in the cascade mode, the EOI command must be issued twice, once for the master and once for the slave.

Automatic End of Interrupt (AEOI)

If the AEOI mode is set, the 8259 will perform a **non-specific EOI** on its own on the trailing edge of the third INTA pulse. The AEOI mode can only be used for a **master 8259** and not for a slave.

2. Special Fully Nested Mode (SFNM)

In the FNM, on the acknowledgement of an interrupt, further interrupts from the same level are disabled. Consider a large system which uses cascaded 8259s and where the interrupt levels within each slave have to be considered. An interrupt request input to a slave, in turn causes the slave to place an interrupt request to the master on one of the master's inputs. Further interrupts to the slave will cause the slave to place requests to the master on the same input to the master, but these will not be recognised because further interrupts on the same input level are disabled by the master.

The Special Fully Nested Mode (SFNM) is used to avoid this problem. The SFCM is set up by ICW4 during initialisation. It is similar to the FNM except for the following differences:

1. When an interrupt request from a slave is being serviced, the slave is allowed to place further requests if these requests are of a higher priority than the request currently being serviced. These interrupts are recognised by the master and it initiates interrupt requests to the MPU.
2. Before exiting from the interrupt service routine, a non-specific EOI must be sent to the slave and its ISR must be read to determine if it was the only interrupt to the slave. If the ISR is empty, a non-specific EOI command can be sent to the master. If it is not empty, it implies that the same IR level input to the master is to be serviced again due to more than one interrupts being presented to the slave, and an EOI must not be sent to the master.

3. Rotating Priority Mode

The Rotating Priority mode can be set in 1. Automatic Rotation, and 2. Specific Rotation.

1. Automatic Rotation

In this mode, a device, after being serviced, receives the lowest priority. The device just been serviced, will receive the seventh priority. Here IR_3 has just been serviced.

IR_0	IR_1	IR_2	IR_3	IR_4	IR_5	IR_6	IR_7
4	5	6	7	0	1	2	3

2. Specific Rotation

In the Automatic Rotation mode, the interrupt request last serviced is assigned the lowest priority, whereas in the Specific Rotation mode, the lowest priority can be assigned to any interrupt input (IR_0 to IR_7) thus fixes all other priorities.

For example if the lowest priority is assigned to IR_2 , other priorities are as shown below.

IR ₀	IR ₁	IR ₂	IR ₃	IR ₄	IR ₅	IR ₆	IR ₇
5	6	7	0	1	2	3	4

4. Special Mask Mode

If any interrupt is in service, then the corresponding bit is set in ISR and the lower priority interrupts are inhibited. Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion. In these cases, we have to go for special mask mode. In the special mask mode it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked. Thus any interrupt may be selectively enabled by loading the mask register.

5. Poll Mode

In this mode the INT output is not used. The microprocessor checks the status of interrupt requests by issuing poll command. The microprocessor reads contents of 8259A after issuing poll command. During this read operation the 8259A provides polled word and sets ISR bit of highest priority active interrupt request FORMAT.

I	X	X	X	X	W ₂	W ₁	W ₀
---	---	---	---	---	----------------	----------------	----------------

I = 1 → One or more interrupt requests activated.

I = 0 → No interrupt request activated.

W₂ W₁ W₀ → Binary code of highest priority active interrupt request.