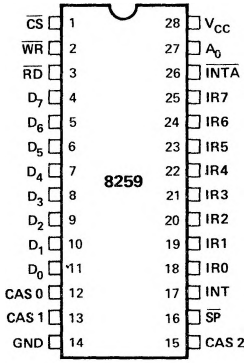
8259, 8259-5
PROGRAMMABLE INTERRUPT CONTROLLER

- **MCS-85™ Compatible 8259-5**
- **Eight Level Priority Controller**
- **Expandable to 64 Levels**
- **Programmable Interrupt Modes**
- **Individual Request Mask Capability**
- **Single +5V Supply (No Clocks)**
- **28 Pin Dual-In-Line Package**
- **Fully Compatible with Intel CPUs**

The 8259 handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts, without additional circuitry. It will be packaged in a 28-pin plastic DIP, uses nMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259 is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

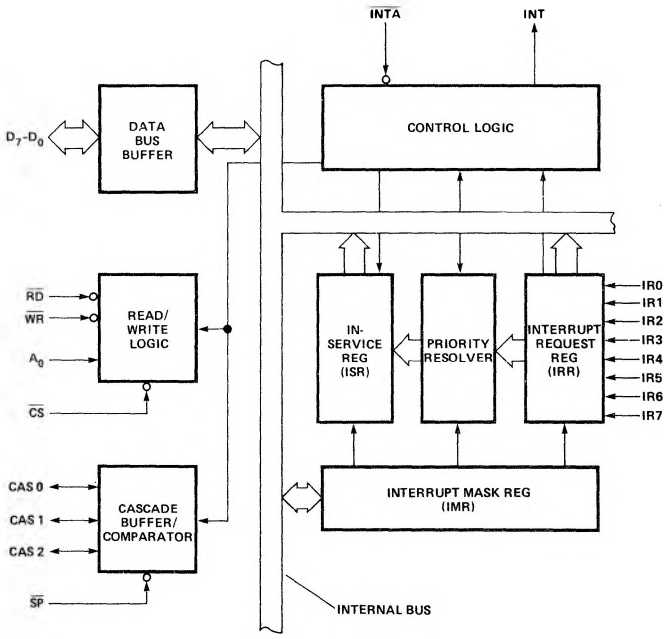
PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (BI-DIRECTIONAL)
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A_0	COMMAND SELECT ADDRESS
$\overline{CS}$	CHIP SELECT
CAS1-CAS0	CASCADE LINES
$\overline{SP}$	SLAVE PROGRAM INPUT
INT	INTERRUPT OUTPUT
$\overline{INTA}$	INTERRUPT ACKNOWLEDGE INPUT
IR0-IR7	INTERRUPT REQUEST INPUTS

BLOCK DIAGRAM



INTERRUPTS IN MICROCOMPUTER SYSTEMS

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient method so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

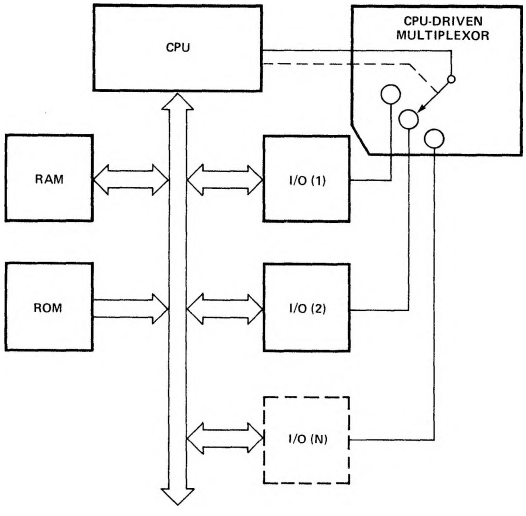
The most common method of servicing such devices is the **Polled** approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuence polling cycle and that such a method would have a serious, detrimental effect on system throughput thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desireable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete however the processor would resume exactly where it left off.

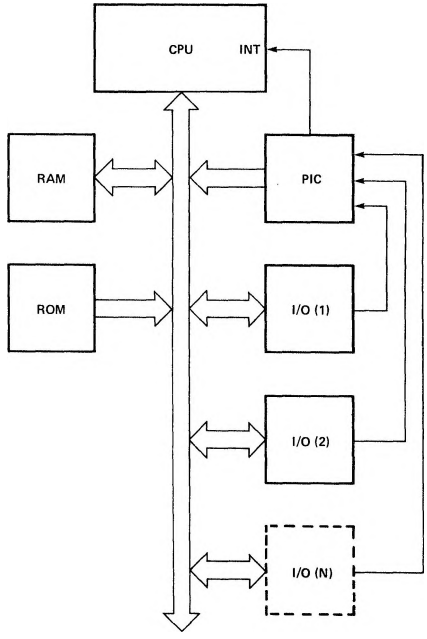
This method is called **Interrupt**. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced and issues an Interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. The PIC does this by providing the CPU with a 3-byte CALL instruction.



POLLED METHOD



INTERRUPT METHOD

8259 BASIC FUNCTIONAL DESCRIPTION

General

The 8259 is a device specifically designed for use in real time, interrupt driven, microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 8259s (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259 can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.

Interrupt Request Register (IRR) and In-Service Register (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

Priority Resolver

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INT (Interrupt)

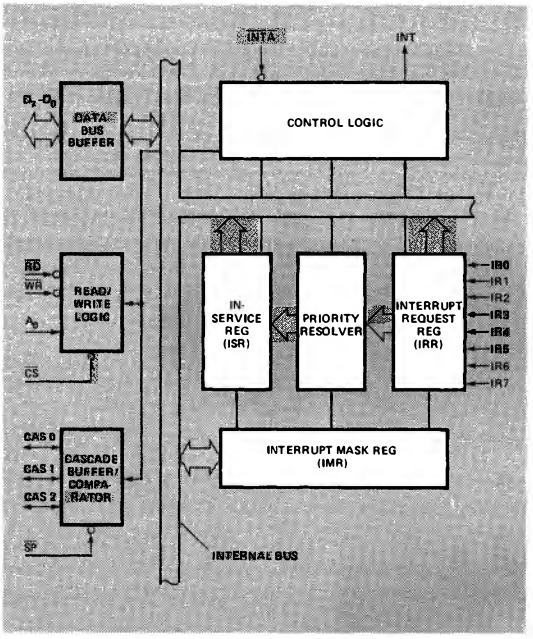
This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080 input level.

$\overline{INTA}$ (Interrupt Acknowledge)

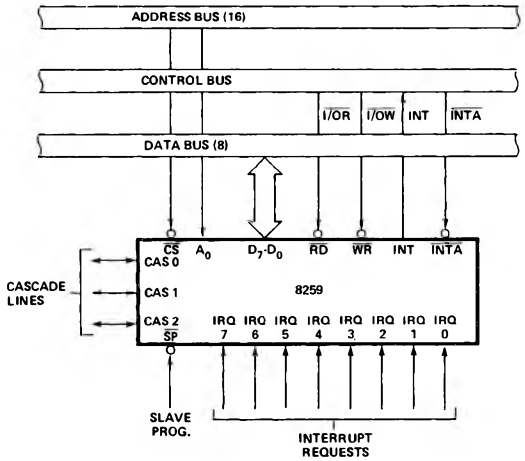
Three $\overline{INTA}$ pulses will cause the 8259 to release a 3-byte CALL instruction onto the Data Bus.

Interrupt Mask Register (IMR)

The IMR stores the bits of the interrupt lines to be masked. The IMR operates on the ISR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.



8259 BLOCK DIAGRAM



8259 INTERFACE TO STANDARD SYSTEM BUS

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8259 to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

Read/Write Control Logic

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259 to be transferred onto the Data Bus.

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8259. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (Write)

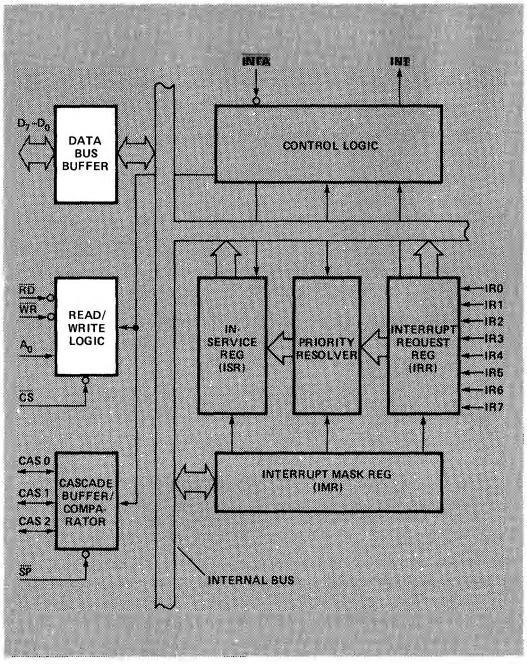
A "low" on this input enables the CPU to write control words (ICWs and OCWs) to the 8259.

$\overline{RD}$ (Read)

A "low" on this input enables the 8259 to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR) or the BCD of the Interrupt level on to the Data Bus.

A0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.



8259 BLOCK DIAGRAM

8259 BASIC OPERATION

A_0	D_4	D_3	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
						INPUT OPERATION (READ)
0			0	1	0	IRR, ISR or Interrupting Level ⇒ DATA BUS (Note 1)
1			0	1	0	IMR ⇒ DATA BUS
						OUTPUT OPERATION (WRITE)
0	0	0	1	0	0	DATA BUS ⇒ OCW2
0	0	1	1	0	0	DATA BUS ⇒ OCW3
0	1	X	1	0	0	DATA BUS ⇒ ICW1
1	X	X	1	0	0	DATA BUS ⇒ OCW1, ICW2, ICW3 (Note 2)
						DISABLE FUNCTION
X	X	X	1	1	0	DATA BUS ⇒ 3-STATE
X	X	X	X	X	1	DATA BUS ⇒ 3-STATE

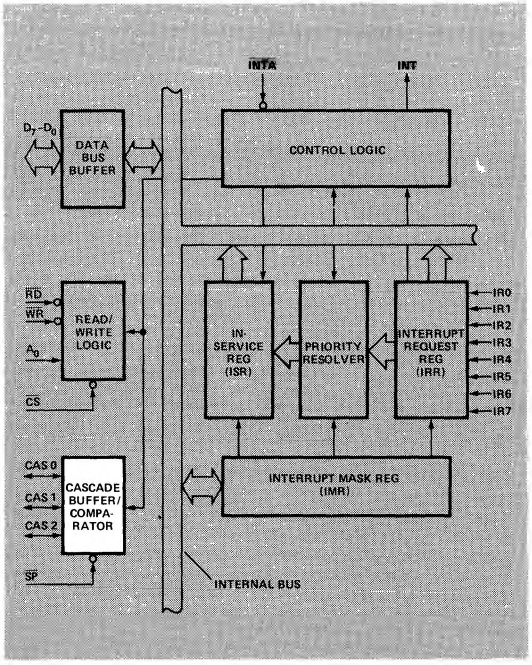
Note 1: Selection of IRR, ISR or Interrupting Level is based on the content of OCW3 written before the READ operation.
Note 2: On-chip sequencer logic queues these commands into proper sequence.

$\overline{SP}$ (Slave Program)

More than one 8259 can be used in the system to expand the priority interrupt scheme up to 64 levels. In such case, one 8259 acts as the master, and the others act as slaves. A "high" on the $\overline{SP}$ pin designates the 8259 as the master, a "low" designates it as a slave.

The Cascade Buffer/Comparator

This function block stores and compares the IDs of all 8259 used in the system. The associated three I/O pins (CAS0-2) are outputs when the 8259 is used as a master ($\overline{SP}$ = 1), and are inputs when the 8259 is used as a slave ($\overline{SP}$ = 0). As a master, the 8259 sends the ID of the interrupting slave device onto the CAS0-2 lines. The slave thus selected will send its preprogrammed subroutine addressed onto the Data Bus during next two consecutive $\overline{INTA}$ pulses. (See section "Cascading the 8259".)



8259 BLOCK DIAGRAM

8259 DETAILED OPERATIONAL SUMMARY

General

The powerful features of the 8259 in a microcomputer system are its programmability and its utilization of the CALL instruction to jump into any address in the memory map. The normal sequence of events that the 8259 interacts with the CPU is as follows:

1. One or more of the INTERRUPT REQUEST lines (IR7-0) are raised high, setting the corresponding IRR bit(s).
2. The 8259 accepts these requests, resolves the priorities, and sends an INT to the CPU.
3. The CPU acknowledges the INT and responds with an $\overline{INTA}$ pulse.
4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259 will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7-0 pins.
5. This CALL instruction will initiate two more $\overline{INTA}$ pulses to be sent to the 8259 from the CPU group.
6. These two $\overline{INTA}$ pulses allow the 8259 to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ pulse and the higher 8-bit address is released at the second $\overline{INTA}$ pulse.
7. This completes the 3-byte CALL instruction released by the 8259. ISR bit is not reset until the end of the subroutine when an EOI (End of interrupt) command is issued to the 8259.

Programming The 8259

The 8259 accepts two types of command words generated by the CPU:

1. Initialization Command Words (ICWs):
   Before normal operation can begin, each 8259 in the system must be brought to a starting point — by a sequence of 2 or 3 bytes timed by $\overline{WR}$ pulses. This sequence is described in Figure 1.
2. Operation Command Words (OCWs):
   These are the command words which command the 8259 to operate in various interrupt modes. These modes are:
   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 8259 at anytime after initialization.

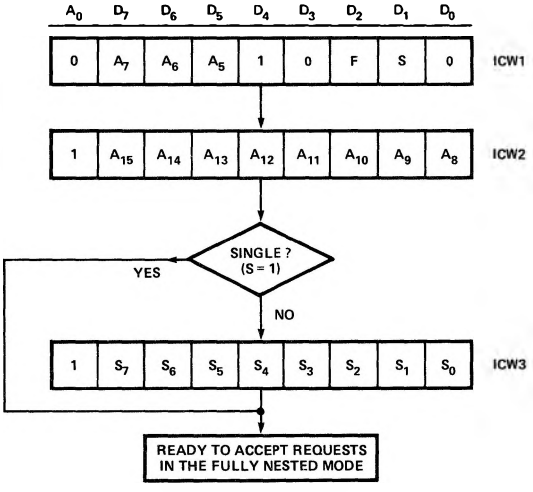


FIGURE 1. INITIALIZATION SEQUENCE

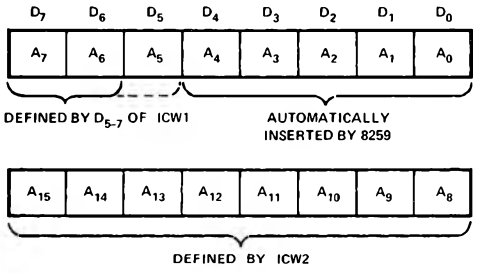
Initialization Command Words 1 and 2: (ICW1 and ICW2)

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1), and initiates the initialization sequence. During this sequence, the following occur automatically:

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low to high transition to generate an interrupt.
b. The interrupt Mask Register is cleared.
c. IR 7 input is assigned priority 7.
d. Special Mask Mode Flip-flop and status Read Flip-flop are reset.

The 8 requesting devices have 8 addresses equally spaced in memory. The addresses can be programmed at intervals of 4 or 8 bytes; the 8 routines thus occupying a page of 32 or 64 bytes respectively in memory.

The address format is:



A0-4 are automatically inserted by the 8259, while A15-6 are programmed by ICW1 and ICW2. When interval = 8, A5 is fixed by the 8259. If interval = 4, A5 is programmed in ICW1. Thus, the interrupt service routines can be located anywhere in the memory space. The 8 byte interval will maintain compatibility with current 8080 RESTART instruction software, while the 4 byte interval is best for compact jump table.

The address format inserted by the 8259 is described in Table 1.

The bits F and S are defined by ICW1 as follows:

F: Call address interval. F = 1, then interval = 4; F = 0, then interval = 8.

S: Single. S = 1 means that this is the only 8259 in the system. It avoids the necesity of programming ICW3.

	INTERVAL = 4								INTERVAL = 8							
	LOWER MEMORY ROUTINE ADDRESS															
	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
IR 7	A7	A6	A5	1	1	1	0	0	A7	A6	1	1	1	0	0	0
IR 6	A7	A6	A5	1	1	0	0	0	A7	A6	1	1	0	0	0	0
IR 5	A7	A6	A5	1	0	1	0	0	A7	A6	1	0	1	0	0	0
IR 4	A7	A6	A5	1	0	0	0	0	A7	A6	1	0	0	0	0	0
IR 3	A7	A6	A5	0	1	1	0	0	A7	A6	0	1	1	0	0	0
IR 2	A7	A6	A5	0	1	0	0	0	A7	A6	0	1	0	0	0	0
IR 1	A7	A6	A5	0	0	1	0	0	A7	A6	0	0	1	0	0	0
IR 0	A7	A6	A5	0	0	0	0	0	A7	A6	0	0	0	0	0	0

TABLE 1.

Example of Interrupt Acknowledge Sequence

Assume the 8259 is programmed with F = 1 (CALL address interval = 4), and IR5 is the interrupting level. The 3 byte sequence released by the 8259 timed by the $\overline{INTA}$ pulses is as follows:

	D7	D6	D5	D4	D3	D2	D1	D0	
1st $\overline{INTA}$	1	1	0	0	1	1	0	1	CALL CODE
2nd $\overline{INTA}$	A7	A6	A5	1	0	1	0	0	LOWER ROUTINE ADDRESS
3rd $\overline{INTA}$	A15	A14	A13	A12	A11	A10	A9	A8	HIGHER ROUTINE ADDRESS

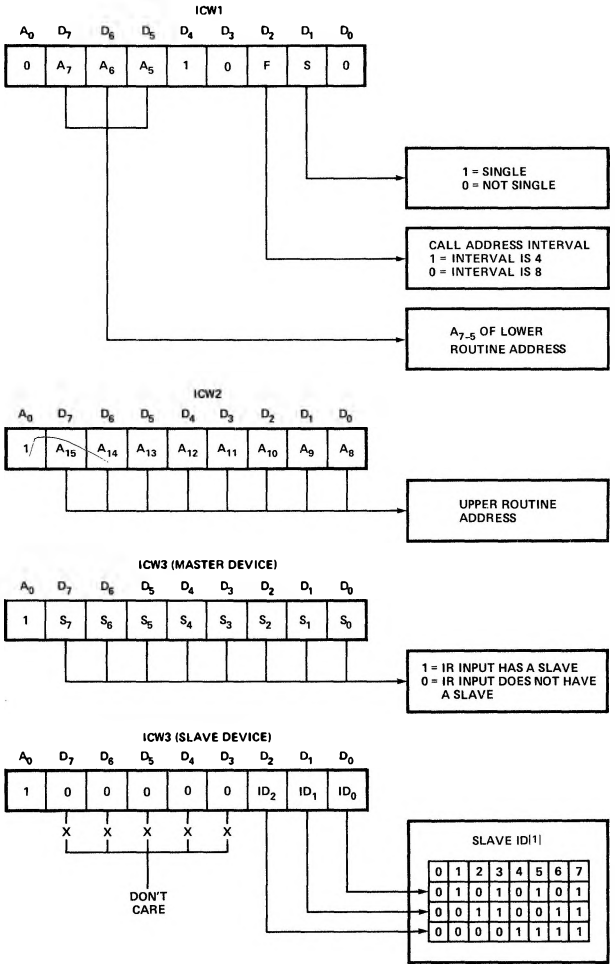
Initialization Command Word 3 (ICW3)

This will load the 8-bit slave register. The functions of this register are as follows:

a. If the 8259 is the master, a "1" is set for each slave in the system. The master then will release byte 1 of the CALL sequence and will enable the corresponding slave to release bytes 2 and 3, through the cascade lines.

b. If the 8259 is a slave, bits 2 - 0 identify the slave. The slave compares its CAS0-2 inputs (sent by the master) with these bits. If they are equal, bytes 2 and 3 of the CALL sequence are released.

If bit S is set in ICW1, there is no need to program ICW3.



NOTE 1: SLAVE ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT.

INITIALIZATION COMMAND WORD FORMAT

Operation Command Words (OCWs)

After the Initialization Command Words (ICWs) are programmed into the 8259, the chip is ready to accept interrupt requests at its input lines. However, during the 8259 operation, a selection of algorithms can command the 8259 to operate in various modes through the Operation Command Words (OCWs). These various modes and their associated OCWs are described below.

Interrupt Masks

Each Interrupt Request input can be masked individually by the Interrupt Masked Register (IMR) programmed through OCW1.

The IMR operates on the In-Service Register. Note that if an interrupt is already acknowledged by the 8259 (an $\overline{INTA}$ pulse has occurred), then the Interrupting level, although masked, will inhibit the lower priorities. To enable these lower priority interrupts, one can do one of two things: (1) Write an End of Interrupt (EOI) command (OCW2) to reset the IST bit or (2) Set the special mask mode using OCW3 (as will be explained later in the special mask mode.)

Fully Nested Mode

The 8259 will operate in the fully nested mode after the execution of the initialization sequence without any OCW being written. In this mode, the interrupt requests are ordered in priorities from 0 through 7. When an interrupt is acknowledged, the highest priority request is determined and its address vector placed on the bus. In addition, a bit of the Interrupt service register (IS 7-0) is set. This bit remains set until the CPU issues an End of Interrupt (EOI) command immediately before returning from the service routine. While the IS bit is set, all further interrupts of lower priority are inhibited, while higher levels will be able to generate an interrupt (which will only be acknowledged if the CPU has enabled its own interrupt input through software).

After the Initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained in the rotating priority mode.

Rotating Priority Commands

There are two variations of rotating priority: auto rotate and specific rotate.

1. Auto Rotate — Executing the Rotate-at-EOI (Auto) command, resets the highest priority ISR bit and assigns that input the lowest priority. Thus, a device requesting an interrupt will have to wait, in the worst case, until 7 other devices are serviced at most once each, i.e., if the priority and "in-service" status is:

BEFORE ROTATE	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
"IS" STATUS	0	1	0	1	0	0	0	0
	LOWEST PRIORITY						HIGHEST PRIORITY	
PRIORITY STATUS	7	6	5	4	3	2	1	0

AFTER ROTATE	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
"IS" STATUS	0	1	0	0	0	0	0	0
	LOWEST PRIORITY						HIGHEST PRIORITY	
PRIORITY STATUS	4	3	2	1	0	7	6	5

In this example, the In-Service FF corresponding to line 4 (the highest priority FF set) was reset and line 4 became the lowest priority, while all the other priorities rotated correspondingly.

The Rotate command is issued in OCW2, where: R = 1, EOI = 1, SEOI = 0.

2. Specific Rotate — The programmer can change priorities by programming the bottom priority, and by doing this, to fix the highest priority: i.e., if IR5 is programmed as the bottom priority device, the IR6 will have the highest one. This command can be used with or without resetting the selected ISR bit.

The Rotate command is issued in OCW2 where: R = 1, SEOI = 1. L2, L1, L0 are the BCD priority level codes of the bottom priority device. If EOI = 1 also, the ISR bit selected by L2-L0 is reset.

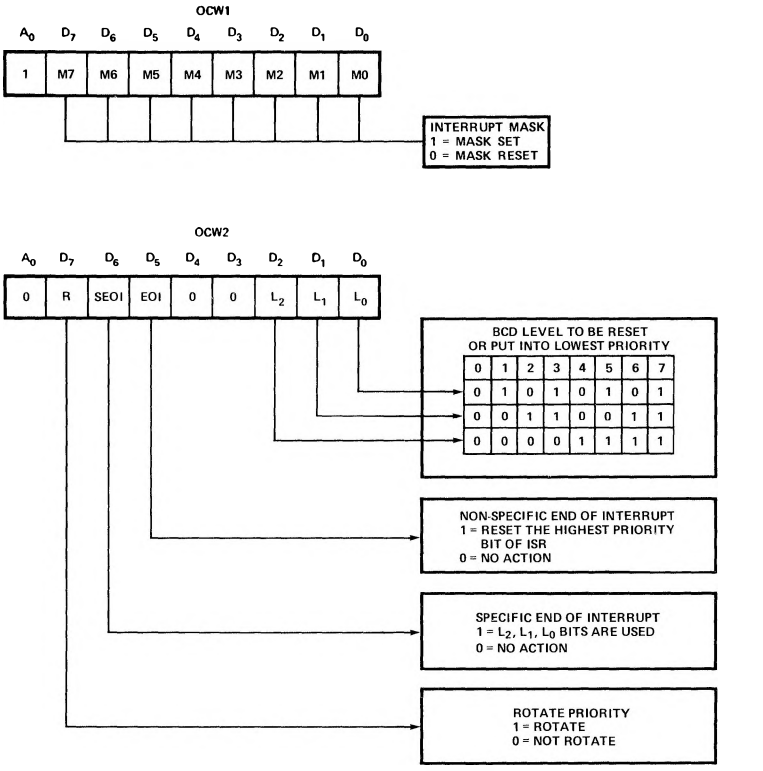
Observe that this mode is independent of the End of Interrupt Command and priority changes can be executed during EOI command or independently from the EOI command.

End of Interrupt (EOI) and Specific End of Interrupt (SEOI)

An End of Interrupt command word must be issued to the 8259 before returning from a service routine, to reset the appropriate IS bit.

There are two forms of EOI command: Specific and non-Specific. When the 8259 is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a non-Specific EOI command is issued the 8259 will automatically reset the highest IS bit of those that are set, since in the nested mode, the highest IS level was necessarily the last level acknowledged and will necessarily be the next routine level returned from.

However, when a mode is used which may disturb the fully nested structure, such as in the rotating priority case, the 8259 may no longer be able to determine the last level acknowledged. In this case, a specific EOI (SEOI) must be issued which includes the IS level to be reset as part of the command. The End of the Interrupt is issued whenever EOI = "1" in OCW2. For specific EOI, SEOI = "1", and EOI = 1. L2, L1, L0 is then the BCD level to be reset. As explained in the Rotate Mode earlier, this can also be the bottom priority code. Note that although the Rotate command can be issued during an EOI = 1, it is not necessarily tied to it.



OPERATION COMMAND WORD FORMAT

Special Mask Mode (SMM)

This mode is useful when some bit(s) are set (masked) by the Interrupt Mask Register (IMR) through OCW1. If, for some reason, we are currently in an interrupt service routine which is masked (this could happen when the subroutine intentionally mask itself off), it is still possible to enable the lower priority lines by setting the Special Mask mode. In this mode the lower priority lines are enabled until the SMM is reset. The higher priorities are not affected.

The special mask mode FF is set by OCW3 where ESMM = 1, SMM = 1, and reset where: ESSM = 1 and SMM = 0.

Polled Mode

In this mode, the CPU must disable its interrupt input. Service to device is achieved by programmer initiative by a Poll command.

The poll command is issued by setting P = "1" in OCW3 during a $\overline{WR}$ pulse.

The 8259 treats the next $\overline{RD}$ pulse as an interrupt acknowledge, sets the appropriate IS Flip-flop, if there is a request, and reads the priority level.

For polling operation, an OCW3 must be written before every read.

The word enabled onto the data bus during $\overline{RD}$ is:

D7	D6	D5	D4	D3	D2	D1	D0
I	–	–	–	–	W2	W1	W0

W0 — 2: BCD code of the highest priority level requesting service.

I: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine common to several levels — so that the $\overline{INTA}$ sequence is not needed (and this saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.

SUMMARY OF OPERATION COMMAND WORD PROGRAMMING

	A0	D4	D3		
OCW1	1			M7-M0	IMR (Interrupt Mask Register). $\overline{WR}$ will load it while status can be read with $\overline{RD}$.
OCW2	0	0	0	R SEOI EOI	
				0 0 0	No Action.
				0 0 1	Non-specific End of Interrupt.
				0 1 0	No Action.
				0 1 1	Specific End of Interrupt. L2, L1, L0 is the BCD level to be reset.
				1 0 0	No Action.
				1 0 1	Rotate priority at EOI. (Auto Mode)
				1 1 0	Rotate priority, L2, L1, L0 becomes bottom priority without Ending of Interrupt.
				1 1 1	Rotate priority at EOI (Specific Mode), L2, L1, L0 becomes bottom priority, and its corresponding IS FF is reset.
OCW3	0	0	1	ESMM SMM	
				0 0	Special Mask not Affected.
				0 1	Special Mask not Affected.
				1 0	Reset Special Mask.
				1 1	Set Special Mask.
				ERIS RIS	
				0 0	No Action.
				0 1	No Action.
				1 0	Read IR Register Status.
				1 1	Read IS Register Status.

Note: The CPU interrupt input must be disabled during:

1. Initialization sequence for all the 8259 in the system.
2. Any control command execution.

Reading 8259 Status

The input status of several internal registers can be read to update the user information on the system. The following registers can be read by issuing a suitable OCW3 and reading with $\overline{RD}$.

Interrupt Requests Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR).

In Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when prior to the $\overline{RD}$ pulse, an $\overline{WR}$ pulse is issued with OCW3, and ERIS = 1, RIS = 0.

The ISR can be read in a similar mode, when ERIS = 1, RIS = 1.

There is no need to write an OCW3 before every status read operation as long as the status read corresponds with the previous one, i.e. the 8259 "remembers" whether the IRR or ISR has been previously selected by the OCW3.

For reading the IMR, a $\overline{WR}$ pulse is not necessary to preceed the $\overline{RD}$. The output data bus will contain the IMR whenever $\overline{RD}$ is active and A0 = 1.

Polling overrides status read when P = 1, ERIS = 1 in OCW3.

Cascading

The 8259 can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

A typical system is shown in Figure 2. The master controls, through the 3 line cascade bus, which one of the slaves will release the corresponding address.

As shown in Figure 2, the slaves interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will release the 8080 CALL code during byte 1 of $\overline{INTA}$ and will enable the corresponding slave to release the device routine address during bytes 2 and 3 of $\overline{INTA}$.

The cascade bus lines are normally low and will contain the slave address code from the trailing edge of the first $\overline{INTA}$ pulse to the trailing edge of the third pulse. It is obvious that each 8259 in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select ($\overline{CS}$) input of each 8259. The slave program pin ($\overline{SP}$) must be at a "low" level for a slave (and then the cascade lines are inputs) and at a "high" level for a master (and then the cascade lines are outpus).

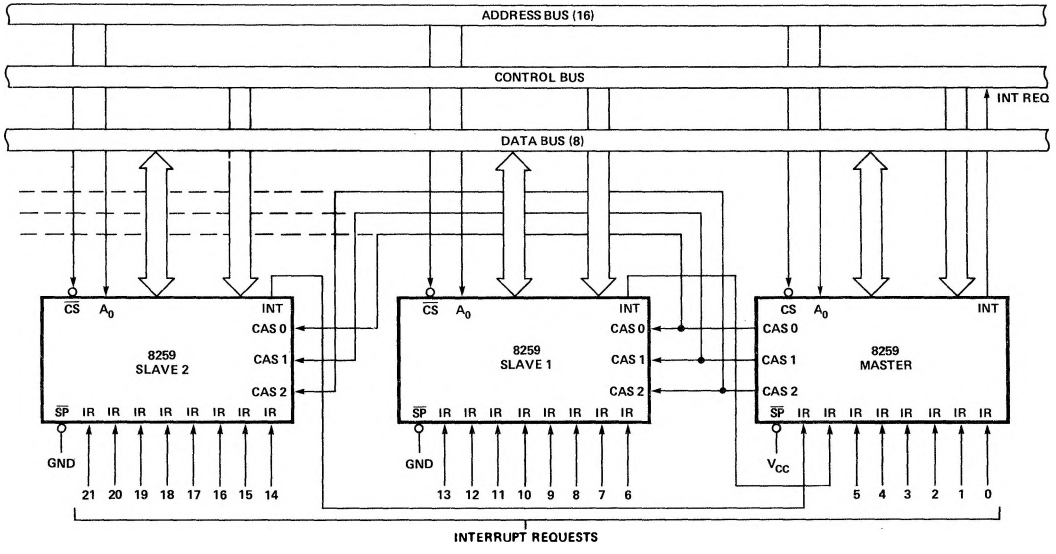


FIGURE 2. CASCADING THE 8259

8259 INSTRUCTION SET

INST. NO.		A0	D7	D6	D5	D4	D3	D2	D1	D0	OPERATION DESCRIPTION
1	ICW1 A	0	A7	A6	A5	1	0	1	1	0	Byte 1 initialization, format = 4, single.
2	ICW1 B	0	A7	A6	A5	1	0	1	0	0	Byte 1 initialization, format = 4, not single.
3	ICW1 C	0	A7	A6	A5	1	0	0	1	0	Byte 1 initialization, format = 8, single.
4	ICW1 D	0	A7	A6	A5	1	0	0	0	0	Byte 1 initialization, format = 8, not single.
5	ICW2	1	A15	A14	A13	A12	A11	A10	A9	A8	Byte 2 initialization (Address No. 2)
6	ICW3 M	1	S7	S6	S5	S4	S3	S2	S1	S0	Byte 3 initialization – master.
7	ICW3 S	1	0	0	0	0	0	S2	S1	S0	Byte 3 initialization – slave.
8	OCW1	1	M7	M6	M5	M4	M3	M2	M1	M0	Load mask reg, read mask reg.
9	OCW2 E	0	0	0	1	0	0	0	0	0	Non specific EOI.
10	OCW2 SE	0	0	1	1	0	0	L2	L1	L0	Specific EOI. L2, L1, L0 code of IS FF to be reset.
11	OCW2 RE	0	1	0	1	0	0	0	0	0	Rotate at EOI (Auto Mode).
12	OCW2 RSE	0	1	1	1	0	0	L2	L1	L0	Rotate at EOI (Specific Mode). L2, L1, L0, code of line to be reset and selected as bottom priority.
13	OCW2 RS	0	1	1	0	0	0	L2	L1	L0	L2, L1, L0 code of bottom priority line.
14	OCW3 P	0	–	0	0	0	1	1	0	0	Poll mode.
15	OCW3 RIS	0	–	0	0	0	1	0	1	1	Read IS register.
16	OCW3 RR	0	–	0	0	0	1	0	1	0	Read requests register.
17	OCW3 SM	0	–	1	1	0	1	0	0	0	Set special mask mode.
18	OCW3 RSM	0	–	1	0	0	1	0	0	0	Reset special mask mode.

Notes:
1. In the master mode $\overline{SP}$ pin = 1, in slave mode $\overline{SP}$ = 0.
2. (–) = do not care.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to +150°C
Voltage On Any Pin
With Respect to Ground ..... −0.5 V to +7 V
Power Dissipation ..... 1 Watt

*COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ±5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-.5	.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+.5V	V	
V_{OL}	Output Low Voltage		.45	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400 μA
V_{OH-INT}	Interrupt Output High Voltage	2.4 3.5		V V	I_{OH} = -400 μA I_{OH} = -50 μA
$I_{IL(IR_{0-7})}$	Input Leakage Current for IR_{0-7}		-300 10	μA μA	V_{IN} = 0V V_{IN} = V_{CC}
I_{IL}	Input Leakage Current for Other Inputs		10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = 0.45V to V_{CC}
I_{CC}	V_{CC} Supply Current		100	mA	

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
C_{IN}	Input Capacitance			10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = +5V ±5%, GND = 0V)

BUS PARAMETERS

READ

SYMBOL	PARAMETER	8259 MIN.	8259 MAX.	8259-5 MIN.	8259-5 MAX.	UNIT
t_{AR}	$\overline{CS}/A_0$ Stable Before $\overline{RD}$ or $\overline{INTA}$	50		50		ns
t_{RA}	$\overline{CS}/A_0$ Stable After $\overline{RD}$ or $\overline{INTA}$	5		30		ns
t_{RR}	$\overline{RD}$ Pulse Width	420		300		ns
t_{RD}	Data Valid From $\overline{RD}/\overline{INTA}$[1]		300		200	ns
t_{DF}	Data Float After $\overline{RD}/\overline{INTA}$	20	200	20	100	ns

WRITE

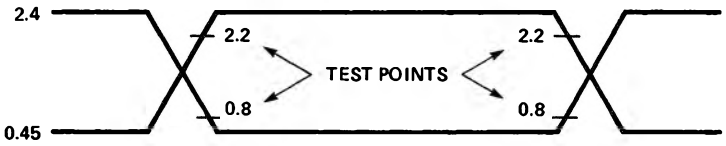
SYMBOL	PARAMETER	8259 MIN.	8259 MAX.	8259-5 MIN.	8259-5 MAX.	UNIT
t_{AW}	A_0 Stable Before $\overline{WR}$	50		50		ns
t_{WA}	A_0 Stable After $\overline{WR}$	20		30		ns
t_{WW}	$\overline{WR}$ Pulse Width	400		300		ns
t_{DW}	Data Valid to $\overline{WR}$ (T.E.)	300		250		ns
t_{WD}	Data Valid After $\overline{WR}$	40		30		ns

OTHER TIMINGS

SYMBOL	PARAMETER	8259 MIN.	8259 MAX.	8259-5 MIN.	8259-5 MAX.	UNIT
t_{IW}	Width of Interrupt Request Pulse	100		100		ns
t_{INT}	INT ↑ After IR ↑	400		350		ns
t_{IC}	Cascade Line Stable After $\overline{INTA}$ ↑	400		400		ns

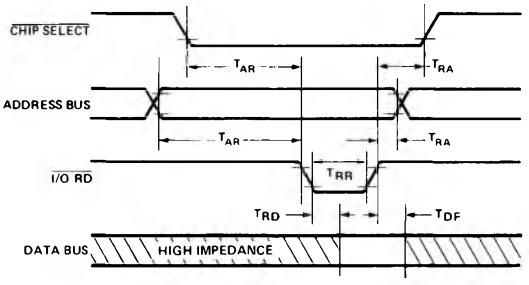
Note 1: 8259: C_L = 100pF, 8259-5: C_L = 150pF.

INPUT WAVEFORMS FOR A.C. TESTS

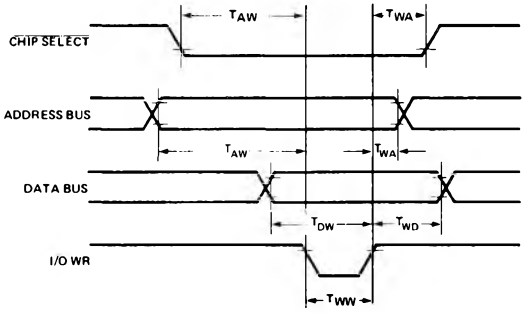


WAVEFORMS

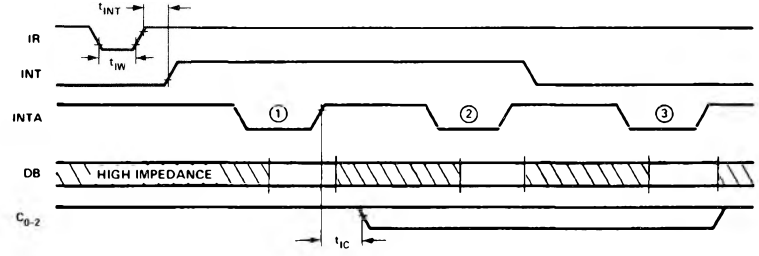
READ TIMING



WRITE TIMING

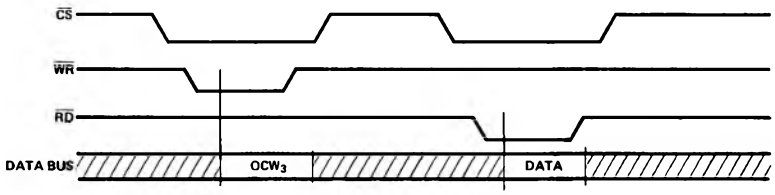


OTHER TIMING



Note: Interrupt Request must remain "HIGH" (at least) until leading edge of first INTA.

READ STATUS/POLL MODE

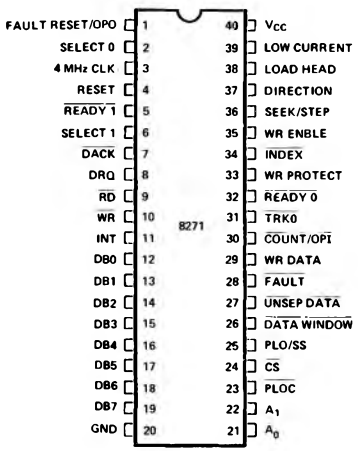


8271
PROGRAMMABLE FLOPPY DISK CONTROLLER

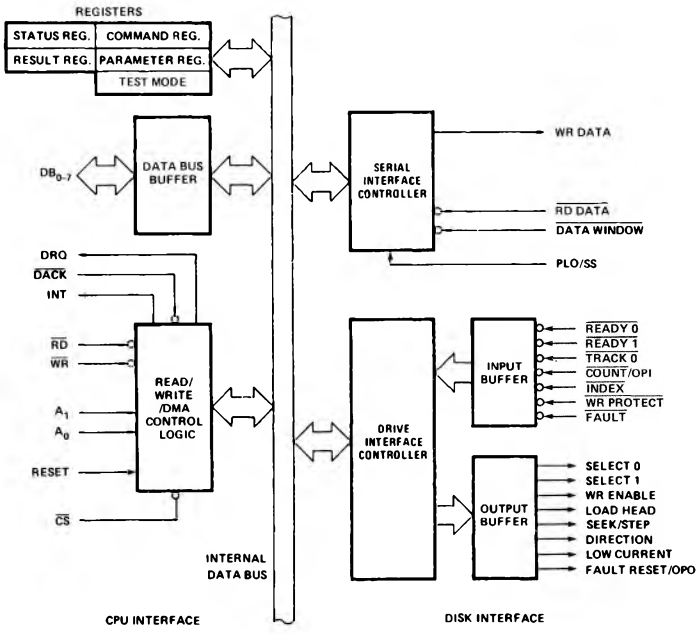
- **IBM 3740 Soft Sectored Format Compatible**
- **Programmable Record Lengths**
- **Multi-Sector Capability**
- **Maintain Dual Drives with Minimum Software Overhead Expandable to 4 Drives**
- **Automatic Read/Write Head Positioning and Verification**
- **Internal CRC Generation and Checking**
- **Programmable Step Rate, Settle-Time, Head Load Time, Head Unload Index Count**
- **Single +5Volt Supply**
- **40 Pin Package**

The 8271 Floppy Disk Controller (FDC) is an LSI Component designed to interface one to four floppy disk drives to an 8-bit microcomputer system. Its powerful control functions minimize both hardware and software overhead normally associated with floppy disk controllers.

PIN CONFIGURATION



BLOCK DIAGRAM



8271 BASIC FUNCTIONAL DESCRIPTION

General

The FDC supports a soft sectored format that is IBM 3740 compatible. This component is a high level controller that relieves the CPU (and user) of many of the control tasks associated with implementing a floppy disk interface. The FDC supports a variety of high level instructions which allow the user to store and retrieve data on a floppy disk without dealing with the low level details of the disk operation.

In addition to the standard read/write commands a scan command is supported. The scan command allows the user program to specify a data pattern and instruct the FDC to search for that pattern on a track. Any application that is required to search the disk (such as point of sale price lookup, disk directory search, etc.) for information may use the scan command to reduce the CPU overhead. Once the scan operation is initiated, no CPU intervention is required.

Hardware Description

The 8271 is packaged in a 40 pin DIP. The following is a functional description of each pin.

Pin Name	I/O	Description
V_{CC}		+5V supply
GND		Ground
4MHz Clock	I	A 4MHz square wave clock
Reset	I	A high signal on the reset input will force the 8271 to an idle state. The 8271 will remain idle until a command is issued by the CPU. The drive interface output signals are forced low.
$\overline{CS}$	I	The I/O Read and I/O Write inputs are enabled by the chip select signal.
DB_7-DB_0	I/O	The Data Bus lines are bidirectional three-state lines.
$\overline{WR}$	I	The Write signal is used to signal the control logic that a transfer of data from the data bus to the 8271 is required.
$\overline{RD}$	I	The Read signal is used to signal the control logic that a transfer of data from the 8271 to the data bus is required.
INT	O	The interrupt signal indicates that the 8271 requires service.
A_1-A_0	I	These two lines are used to select the destination of source of data to be accessed by the control logic.
DRQ	O	The DMA request signal is used to request a transfer of data between the 8271 and memory.
$\overline{DACK}$	I	The DMA ACK signal notifies the 8271 that a DMA cycle has been granted.
Select 1-Select 0	O	These lines are used to specify the selected drive.
Fault Reset/OPO	O	The fault reset line is used to reset an error condition which is latched by the drive, otherwise the pin is a user specified optional output.
Write Enable	O	This signal enables the drive write logic.
Seek/Step	O	This multi-function line is used during drive seeks.
Direction	O	The direction line specifies the seek direction.
Load Head	O	The load head line causes the drive to load the Read/Write head load pad against the diskette.
Low Current	O	This line notifies the drive that track 43 or greater is selected.
$\overline{Ready\ 1}$, $\overline{Ready\ 0}$	I	These two lines indicate that the specified drive is ready.
$\overline{Fault}$	I	This line is used by the drive to specify a file unsafe condition.
$\overline{Count/OPI}$	I	If the seek / direction / count seek mode is selected, the count pin is pulsed for each track. Otherwise this pin is user specified optional input.
$\overline{Write\ Protect}$	I	This signal is used to specify if the drive/diskette may be written.
$\overline{TRK0}$	I	This signal indicates when the R/W head is positioned over track zero.
$\overline{Index}$	I	The index signal gives an indication of the relative position of the diskette.
$\overline{PLO/SS}$	I	This pin is used to specify the type of data separator used.
Write Data	O	Composite write data.
$\overline{Unseparated\ Data}$	I	This input is the unseparated data and clocks.
$\overline{Data\ Window}$	I	This is a data window established by the single-shot or phase-locked oscillator data separator.
$\overline{PLOC}$	O	This line is low when the 8271 is searching for input data sync.

Principles of Operation

The 8271 is fully compatible with Intel microprocessors. It accepts commands from the CPU, executes these Commands and provides a Result at the end of execution.

Communication with the CPU are through the activating of $\overline{CS}$, $\overline{RD}$, $\overline{WR}$ pins. The A_1,A_0 select the appropriate registers on chip:

A_3	A_3	CS RD	CS WR
0	0	Status Reg	Command Reg
0	1	Result Reg	Parameter Reg

The FDC chip operation is composed of the following general sequence of events:

The Command Phase

During the Command Phase, the CPU issues a command byte to the 8271. The command byte provides a general description of the type of operation requested. Many operations require more detailed information about the comand. In such case, from zero to five parameters are written following the command byte to provide such information. The various commands that the 8271 can recognize are listed in the Software Operation Section.

The Execution Phase

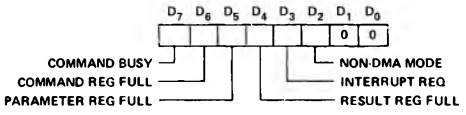
Soon as the last parameter is written into the 8271, the FDC enters the Execution Phase. During this phase there is no need for CPU involvement. The FDC may optionally interface with the 8257 (DMA controller) for high speed data transfers (See System Diagram).

The Result Phase

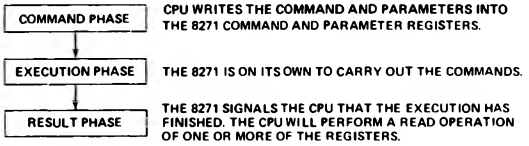
During the Result Phase, the FDC chip notified the CPU of the outcome of the command execution. This phase may be initiated by:

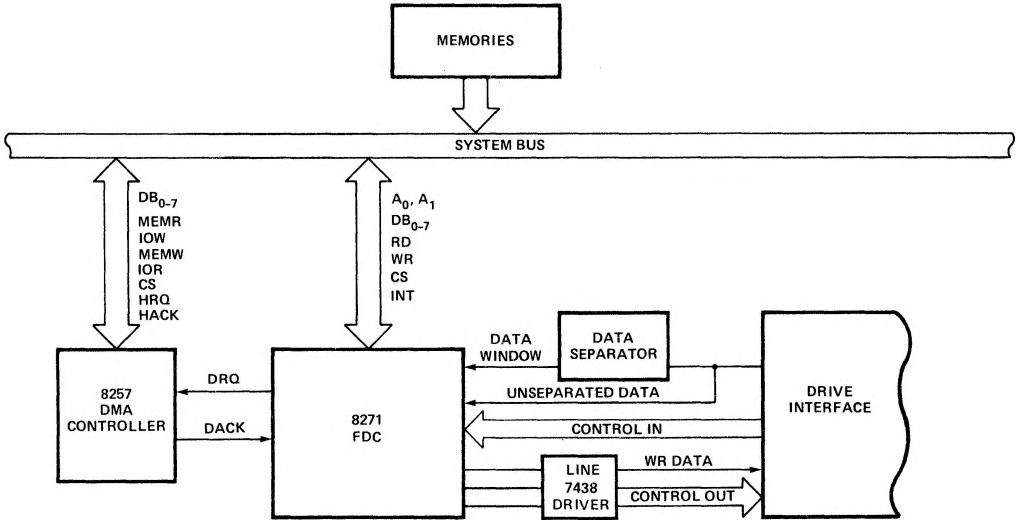
1. The successful completion of an operation.
2. An error detected during an operation.
3. An illegal command or parameter detected during the Command Phase.

In the Result Phase, the CPU Reads the Status Register which provides the following information:



After reading the Status Register, the CPU then Reads the Result Register for more information.





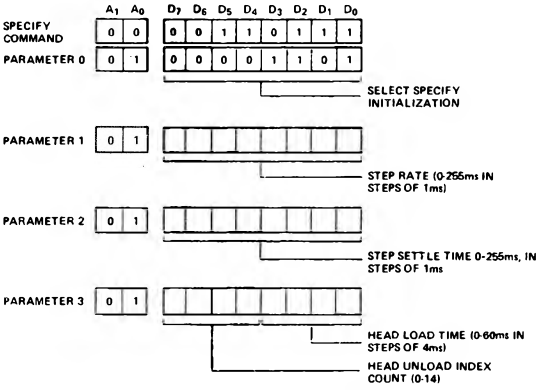
8271 SYSTEM DIAGRAM

Software Operation

The 8271 can accept many powerful commands from the CPU. The following is a list of Basic Commands (associated Parameters not shown).

SCAN DATA
SCAN DATA AND DELETED DATA
WRITE DATA
WRITE DATA AND DELETED DATA
READ DATA
READ DATA AND DELETED DATA
READ ID
VERIFY DATA AND DELETED DATA
FORMAT
SEEK
READ DRIVE STATUS
SPECIFY
RESET

As an example, the SPECIFY command is associated with 4 parameters:



EXECUTION PHASE BASIC CHARACTERISTICS

The following table summarizes the various commands with corresponding execution phase characteristics.

COMMANDS	1 Deleted Data	2 Head	3 Ready	4 Write/ Protect	5 Seek	6 Seek Check	7 Result	8 Completion Interrupt
SCAN DATA	SKIP	LOAD	√	x	YES	YES	YES	YES
SCAN DATA AND DEL DATA	XFER	LOAD	√	x	YES	YES	YES	YES
WRITE DATA	x	LOAD	√	√	YES	YES	YES	YES
WRITE DEL DATA	XFER	LOAD	√	√	YES	YES	YES	YES
READ DATA	SKIP	LOAD	√	x	YES	YES	YES	YES
READ DATA AND DEL DATA	XFER	LOAD	√	x	YES	YES	YES	YES
READ ID	x	LOAD	√	x	YES	NO	YES	YES
VERIFY DATA AND DEL DATA	XFER	LOAD	√	x	YES	YES	YES	YES
FORMAT	x	LOAD	√	√	YES	NO	YES	YES
SEEK	x	–	x	x	YES	NO	YES	YES
READ DRIVE STAT	x	–	x	x	NO	NO	YES	NO
SPECIFY	x	–	x	x	NO	NO	NO	NO
RESET	x	UNLOAD	x	x	NO	NO	NO	NO

Note: 1. "x" → DON'T CARE
2. "√" → check
3. "–" → No change