



# 8271/8271-6 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
- Fully MCS-80™ and MCS-85™ Compatible
- Single +5V Supply
- 40-Pin Package

The Intel® 8271 Programmable Floppy Disk Controller (FDC) is an LSI component designed to interface one to 4 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.

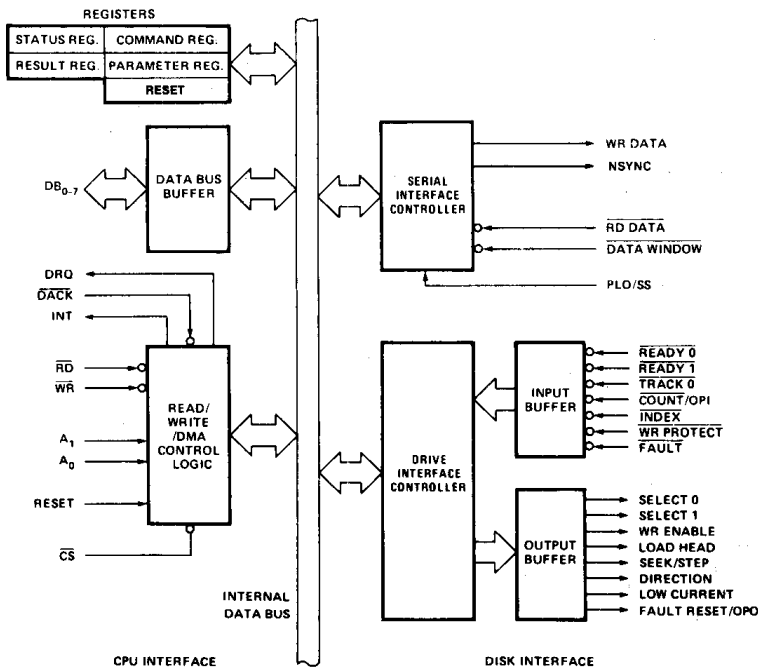


Figure 1. Block Diagram

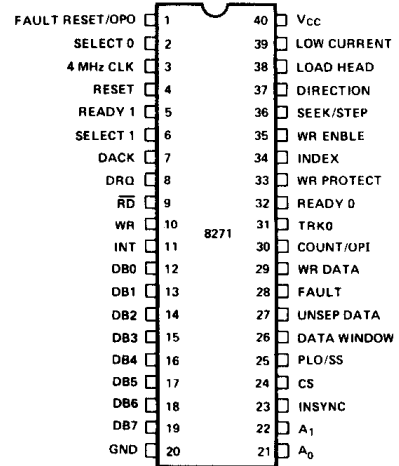


Figure 2. Pin Configuration

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
V <sub>CC</sub>	40		+5V Supply.
GND	20		Ground.
Clock	3	I	<b>Clock:</b> A square wave clock.
Reset	4	I	<b>Reset:</b> A high signal on the reset input forces the 8271 to an idle state. The 8271 remains idle until a command is issued by the CPU. The output signals of the drive interface are forced inactive (LOW). Reset must be active for 10 or more clock cycles.
$\overline{CS}$	24	I	<b>Chip Select:</b> The I/O Read and I/O Write inputs are enabled by the chip select signal.
DB <sub>7</sub> -DB <sub>0</sub>	19-12	I/O	<b>Data Bus:</b> The Data Bus lines are bidirectional, three-state lines (8080 data bus compatible).
$\overline{WR}$	10	I	<b>Write:</b> The Write signal is used to signal the control logic that a transfer of data from the data bus to the 8271 is required.
$\overline{RD}$	9	I	<b>Read:</b> 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	11	O	<b>Interrupt:</b> The interrupt signal indicates that the 8271 requires service.
A <sub>1</sub> -A <sub>0</sub>	22-21	I	<b>Address Line:</b> These two lines are CPU Interface Register select lines.
DRQ	8	O	<b>Data Request:</b> The DMA request signal is used to request a transfer of data between the 8271 and memory.
$\overline{DACK}$	7	I	<b>Data Acknowledge:</b> The DMA acknowledge signal notifies the 8271 that a DMA cycle has been granted. For non-DMA transfers, this signal should be driven in the manner of a "Chip Select."
Select 1- Select 0	6 2	O	<b>Selected Drive:</b> These lines are used to specify the selected drive. These lines are set by the command byte.

Symbol	Pin No.	Type	Name and Function
Fault Reset/ OPO	1	O	<b>Fault Reset:</b> The optional fault reset output line is used to reset an error condition which is latched by the drive. If this line is not used for a fault reset it can be used as an optional output line. This line is set with the write special register command.
Write Enable	35	O	<b>Write Enable:</b> This signal enables the drive write logic.
Seek/Step	36	O	<b>Seek/Step:</b> This multi-function line is used during drive seeks.
Direction	37	O	<b>Direction:</b> The direction line specifies the seek direction. A high level on this pin steps the R/W head toward the spindle (step-in), a low level steps the head away from the spindle (step-out).
Load Head	38	O	<b>Load Head:</b> The load head line causes the drive to load the Read/Write head against the diskette.
Low Current	39	O	<b>Low Current:</b> This line notifies the drive that track 43 or greater is selected.
Ready 1, Ready 0	5 32	I	<b>Ready 1:</b> These two lines indicate that the specified drive is ready.
Fault	28	I	<b>Fault:</b> This line is used by the drive to specify a file unsafe condition.
$\overline{Count/OPI}$	30	I	<b>Count/OPI:</b> If the optional seek/direction/count seek mode is selected, the count pin receives pulses to step the R/W head to the desired track. Otherwise, this line can be used as an optional input.
$\overline{Write Protect}$	33	I	<b>Write Protect:</b> This signal specifies that the diskette inserted is write protected.
TRK0	31	I	<b>Track Zero:</b> This signal indicates when the R/W head is positioned over track zero.
$\overline{Index}$	34	I	<b>Index:</b> The index signal gives an indication of the relative position of the diskette.

**Table 1. Pin Description (Continued)**

Symbol	Pin No.	Type	Name and Function
PLO/SS	25	I	<b>Phase-Locked Oscillator/Single Shot:</b> This pin is used to specify the type of data separator used.
Write Data	29	O	<b>Write Data:</b> Composite write data.
Unseparated Data	27	I	<b>Unseparated Data:</b> This input is the unseparated data and clocks.
Data Window	26	I	<b>Data Window:</b> This is a data window established by a single-shot or phase-locked oscillator data separator.
INSYNC	23	O	<b>Input Synchronization:</b> This line is high when 8271 has attained input data synchronization, by detecting 2 bytes of zeros followed by an expected Address Mark. It will stay high until the end of the ID or data field.

**FUNCTIONAL DESCRIPTION**

**General**

The 8271 Floppy Disk Controller (FDC) interfaces either two single or one dual floppy drive to an eight bit microprocessor and is fully compatible with Intel's new high performance MCS-85 microcomputer system. With minimum external circuitry, this innovative controller supports most standard, commonly-available flexible disk drives including the mini-floppy.

The 8271 FDC supports a comprehensive soft sectored format which is IBM 3740 compatible and includes provision for the designating and handling of bad tracks. It 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 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 instructs the FDC to search for that pattern on a track. Any application that is required to search the disk for information (such as point of sale price lookup, disk directory search, etc.), may use the scan command to reduce the CPU overhead. Once the scan operation is initiated, no CPU intervention is required.

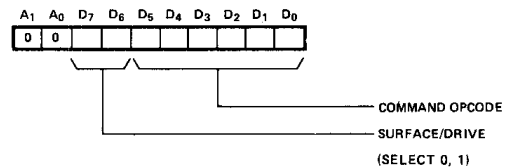
**CPU Interface Description**

This interface minimizes CPU involvement by supporting a set of high level commands and both DMA and non-DMA type data transfers and by providing hierarchical status information regarding the result of command execution.

The CPU utilizes the control interface (see the Block diagram) to specify the FDC commands and to determine the result of an executed command. This interface is supported by five Registers which are addressed by the CPU via the A<sub>1</sub>, A<sub>0</sub>,  $\overline{RD}$  and  $\overline{WR}$  signals. If an 8080 based system is used, the  $\overline{RD}$  and  $\overline{WR}$  signals can be driven by the 8228's  $\overline{I/OR}$  and  $\overline{I/OW}$  signals. The registers are defined as follows:

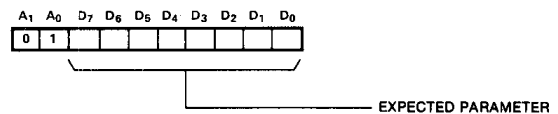
**Command Register**

The CPU loads an appropriate command into the Command Register which has the following format:



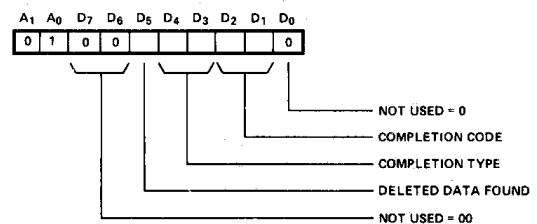
**Parameter Register**

Accepts parameters of commands that require further description; up to five parameters may be required, example:



**Result Register**

The Result Register is used to supply the outcome of FDC command execution (such as a good/bad completion) to the CPU. The standard Result byte format is:



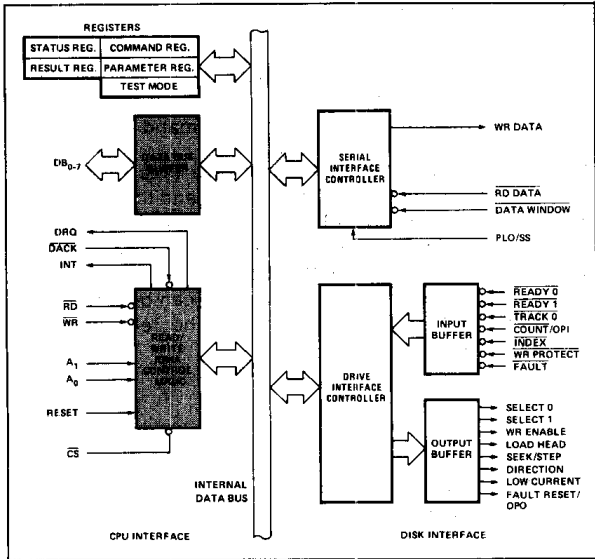
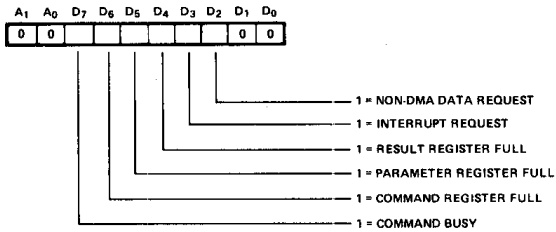


Figure 3. 8271 Block Diagram Showing CPU Interface Functions

**Status Register**

Reflects the state of the FDC.



**Reset Register**

Allows the 8271 to be reset by the program. Reset must be active for 11 or more chip clocks.

**INT (Interrupt Line)**

Another element of the control interface is the Interrupt line (INT). This line is used to signal the CPU that an FDC operation has been completed. It remains active until the result register is read.

**DMA Operation**

The 8271 can transfer data in either DMA or non DMA mode. The data transfer rate of a floppy disk drive is high enough (one byte every 32 usec) to justify DMA transfer. In DMA mode the elements of the DMA interface are:

**DRQ: DMA Request:**

The DMA request signal is used to request a transfer of data between the 8271 and memory.

**DACK: DMA Acknowledge:**

The DMA acknowledge signal notifies the 8271 that a DMA cycle has been granted.

**RD, WR: Read, Write**

The read and write signals are used to specify the direction of the data transfer.

DMA transfers require the use of a DMA controller such as the Intel®8257. The function of the DMA controller is to provide sequential addresses and timing for the transfer at a starting address determined by the CPU. Counting of data block lengths is performed by the FDC.

To request a DMA transfer, the FDC raises DRQ. DACK and RD enable DMA data onto the bus (independently of CHIP SELECT). DACK and WR transfer DMA data to the FDC. If a data transfer request (read or write) is not serviced within 31 μsec, the command is cancelled, a late DMA status is set, and an interrupt is generated. In DMA mode, an interrupt is generated at the completion of the data block transfer.

When configured to transfer data in non-DMA mode, the CPU must pass data to the FDC in response to the non-DMA data requests indicated by the status word. The data is passed to and from the chip by asserting the DACK and the RD or WR signals. Chip select should be inactive (HIGH).

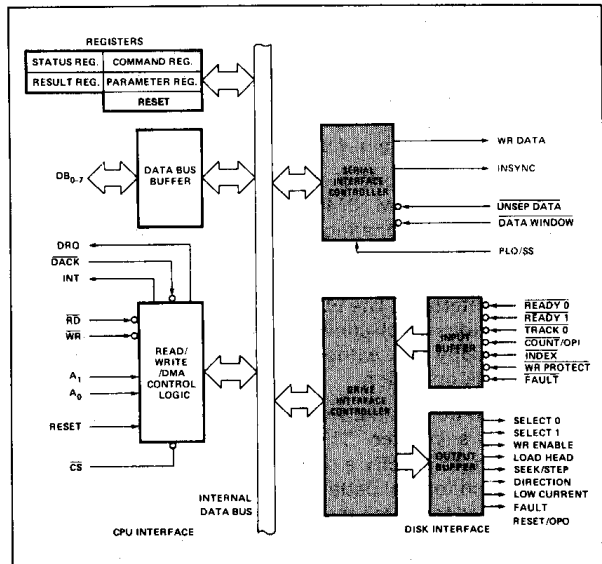


Figure 4. 8271 Block Diagram Showing Disk Interface Functions

**Disk Drive Interface**

The 8271 disk drive interface supports the high level command structure described in the Command Description section. The 8271 maintains the location of bad tracks and the current track location for two drives. However, with minor software support, this interface can support four drives by expanding the two drive select lines (select 0, select 1) with the addition of minimal support hardware.

The FDC Disk Drive Interface has the following major functions.

**READ FUNCTIONS**

Utilize the user supplied data window to obtain the clock and data patterns from the unseparated read data.

Establish byte synchronization.

Compute and verify the ID and data field CRCs.

**WRITE FUNCTIONS**

Encode composite write data.

Compute the ID and data field CRCs and append them to their respective fields.

**CONTROL FUNCTIONS**

Generate the programmed step rate, head load time, head settling time, head unload delay, and monitor drive functions.

**Data Separation**

The 8271 needs only a data window to separate the data from the composite read data as well as to detect missing clocks in the Address Marks.

The window generation logic may be implemented using either a single-shot separator or a phase-locked oscillator.

**Single-Shot Separator**

The single-shot separator approach is the lowest cost solution.

The FDC samples the value of Data Window on the leading edge of Unseparated Data and determines whether the delay from the previous pulse was a half or full bit-cell (high input = full bit-cell, low input = half bit-cell). PLO/SS should be tied to Ground.

**Insync Pin**

This pin gives an indication of whether the 8271 is synchronized with the serial data stream during read operations. This pin can be used with a phase-locked oscillator for soft and hard locking.

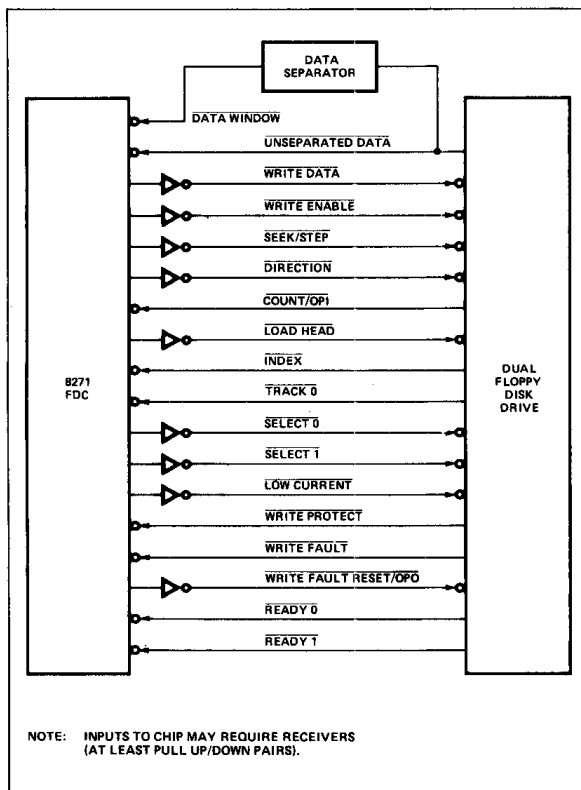
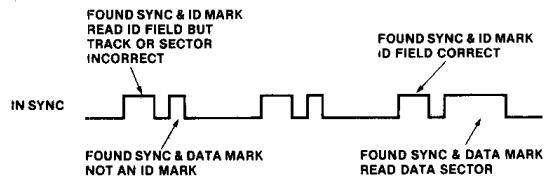


Figure 5. 8271 Disk Drive Interface



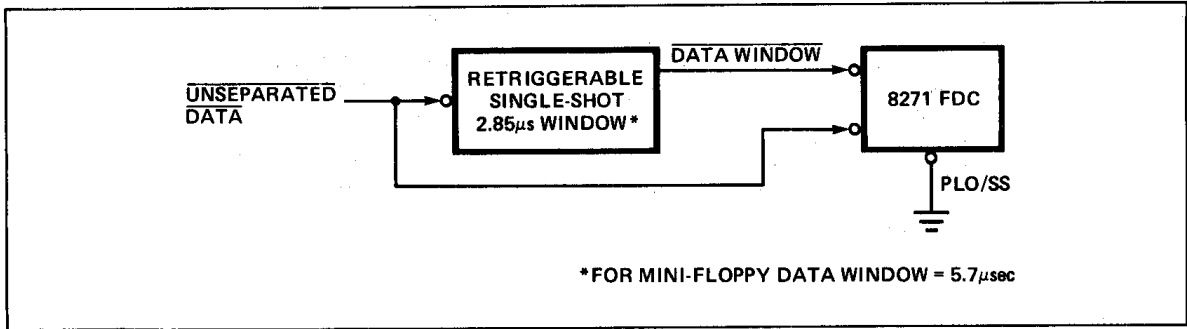


Figure 6. Single-Shot Data Separator Block Diagram

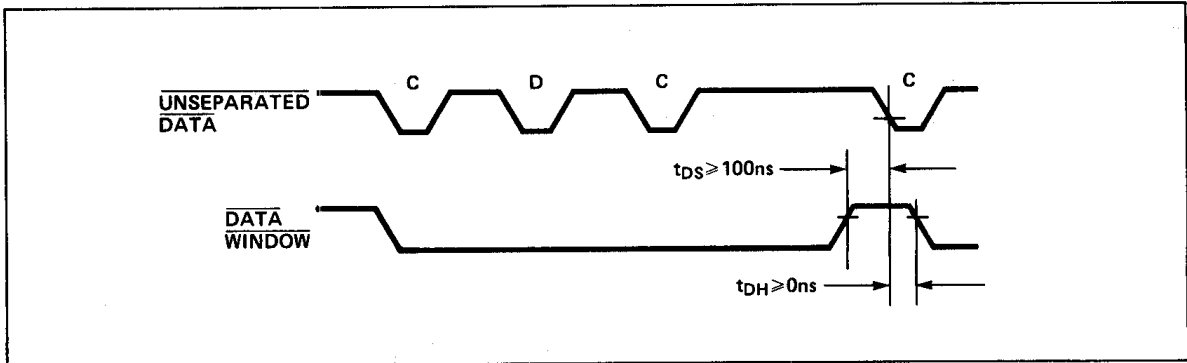


Figure 7. Single-Shot Data Window Timing

**Phase-Locked Oscillator Separator**

The FDC samples the value of Data Window on the leading edge of Unseparated Data and determines whether the pulse represents a Clock or Data Pulse.

Insync may be used to provide soft and hard locking control for the phase-locked oscillator.

PLO/SS should be tied to Vcc (+5V).

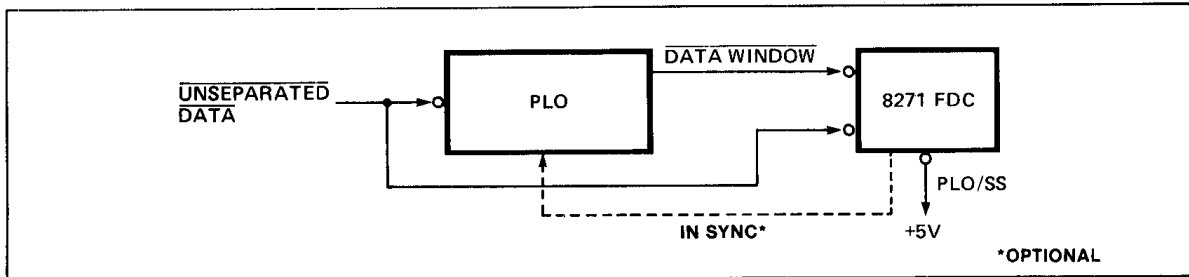


Figure 8. PLO Data Separator Block Diagram

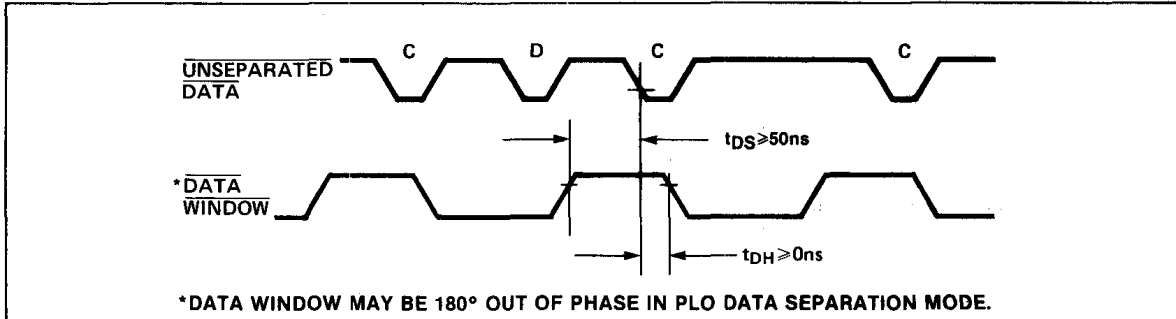


Figure 9. PLO Data Window Timing

**Disk Drive Control Interface**

The disk drive control interface performs the high level and programmable flexible disk drive operations. It custom tailors many varied drive performance parameters such as the step rate, settling time, head load time, and head unload index count. The following is the description of the control interface.

**Write Enable**

The Write Enable controls the read and write functions of a flexible disk drive. When Write Enable is a logical one, it enables the drive write electronics to pass current through the Read/Write head. When Write Enable is a logical zero, the drive Write circuitry is disabled and the Read/Write head detects the magnetic flux transitions recorded on a diskette. The write current turn-on is as follows.

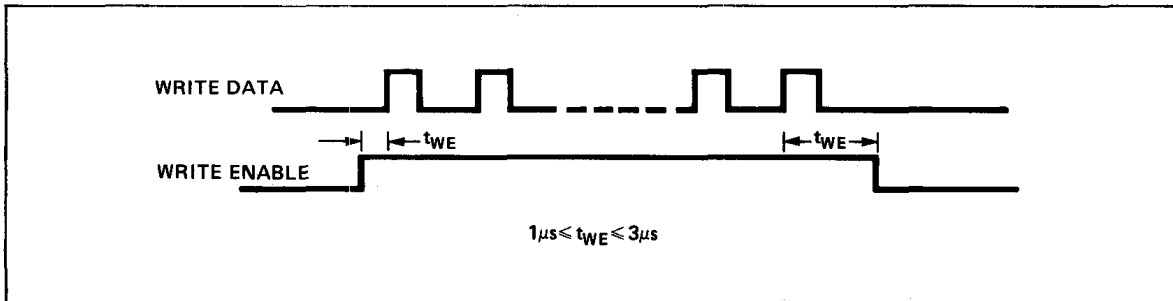


Figure 10. Write Enable Timing

**Seek Control**

Seek Control is accomplished by Seek/Step, Direction, and Count pins and can be implemented two ways to provide maximum flexibility in the subsystem design. One instance is when the programmed step rate is not equal to zero. In this case, the 8271 uses the Seek/Step and Direction pins (the Seek/Step pin becomes a Step pin). Programmable Step timing parameters are shown.

The Direction pin is a control level indicating the direction in which the R/W head is stepped. A logic high level on this line moves the head toward the spindle (step-in). A logic low level moves the head away from the spindle (step-out).

Another instance is when the programmable step rate is equal to zero, in which case the 8271 holds the seek line high until the appropriate number of user-supplied step pulses have been counted on the count input pin.

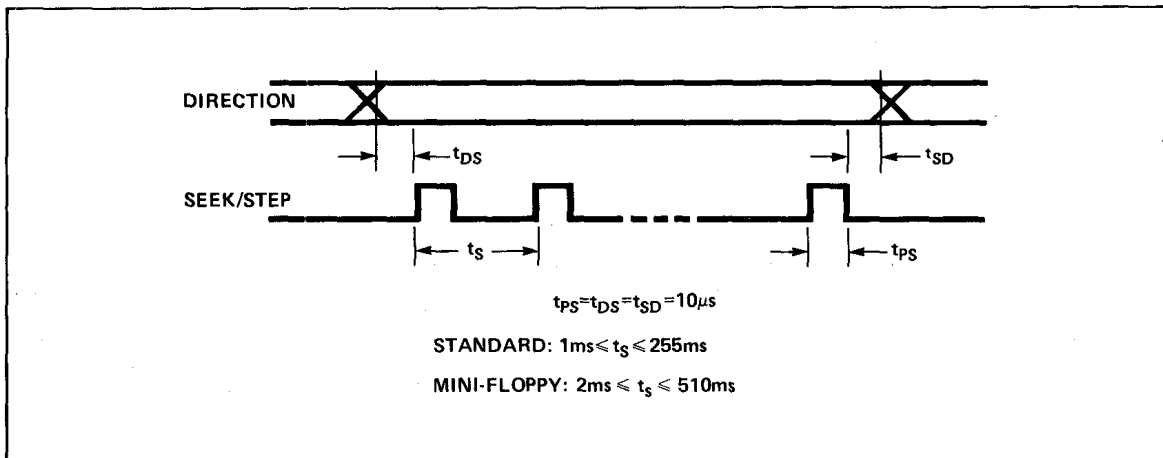


Figure 11. Seek Timing

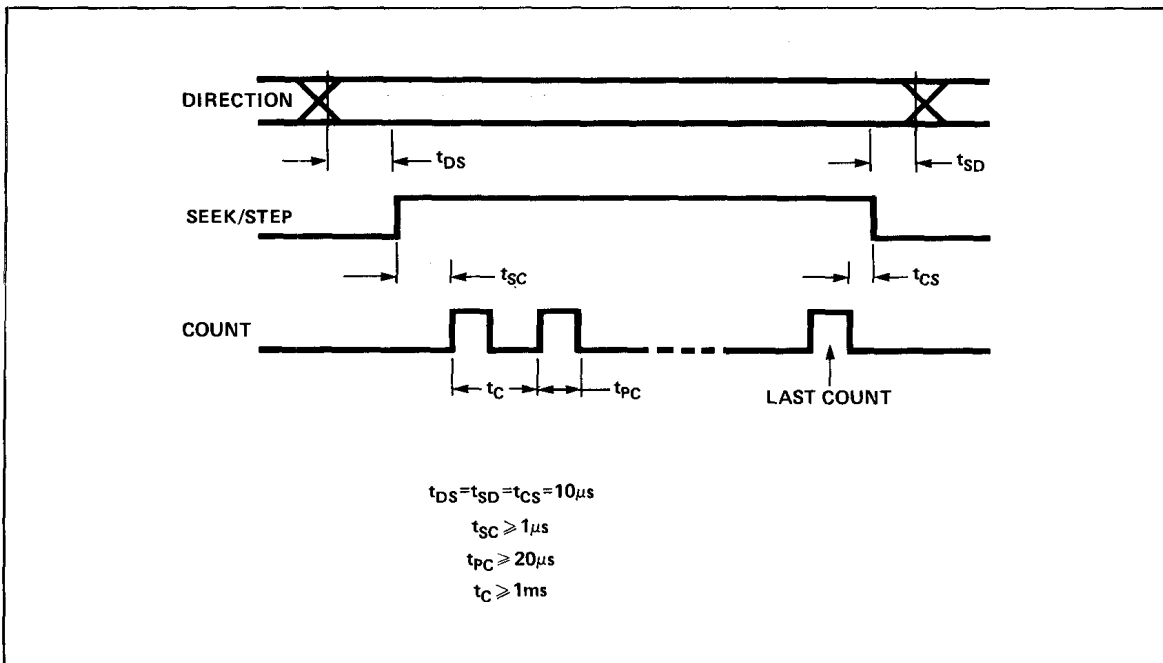


Figure 12. Seek/Step/Count Timing



**Head Seek Settling Time**

The 8271 allows the head settling time to be programmed from 0 to 255ms, in increments of 1ms.

The head settling time is defined as the interval of time from completion of the last step to the time when reading or writing on the diskette is possible (R/W Enable). The R/W head is assumed loaded.

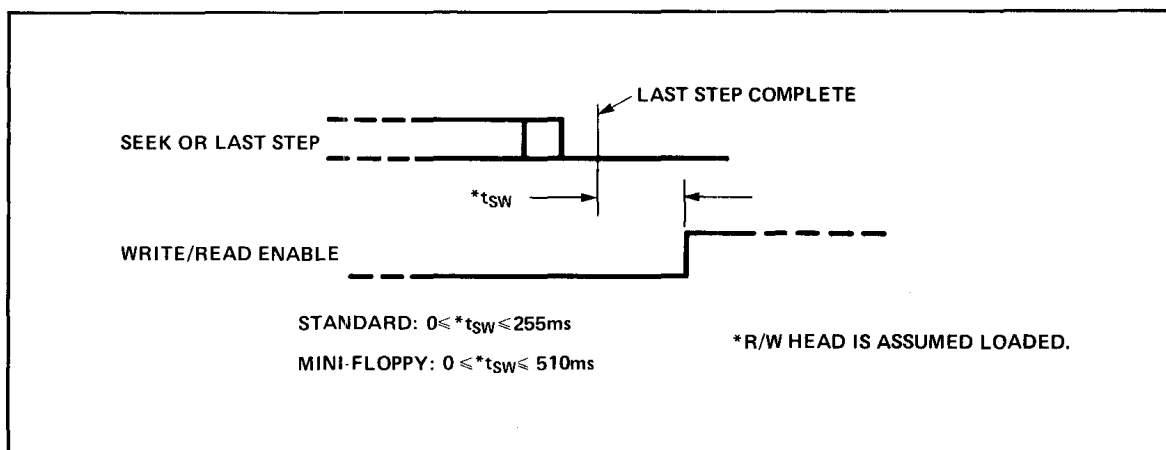


Figure 13. Head Load Settling Timing

**Load Head**

When active, load head output pin causes the drive's read/write head to be loaded on the diskette. When the head is initially loaded, there is a programmed delay (0 to 60ms in 4ms increments) prior to any read or write operation. Provision is also made to unload the head following an operation within a programmed number of diskette revolutions.

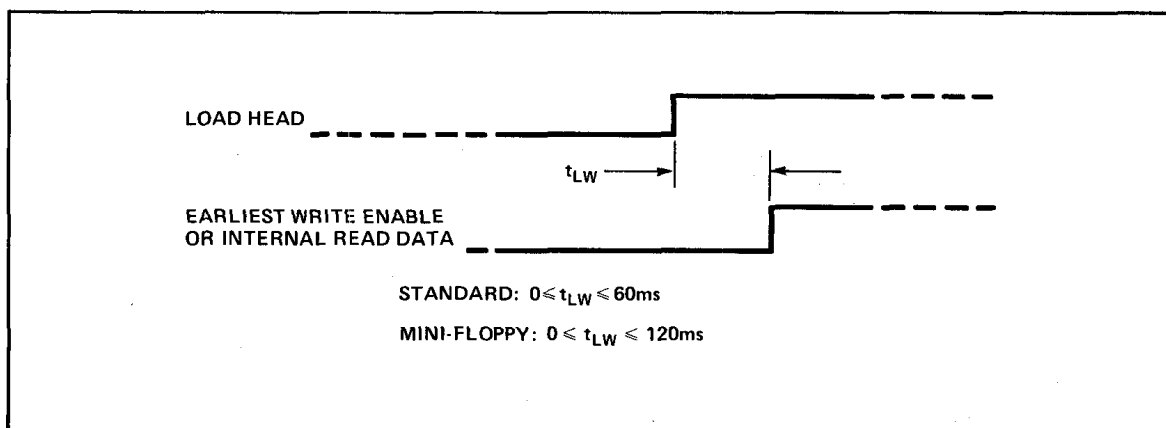
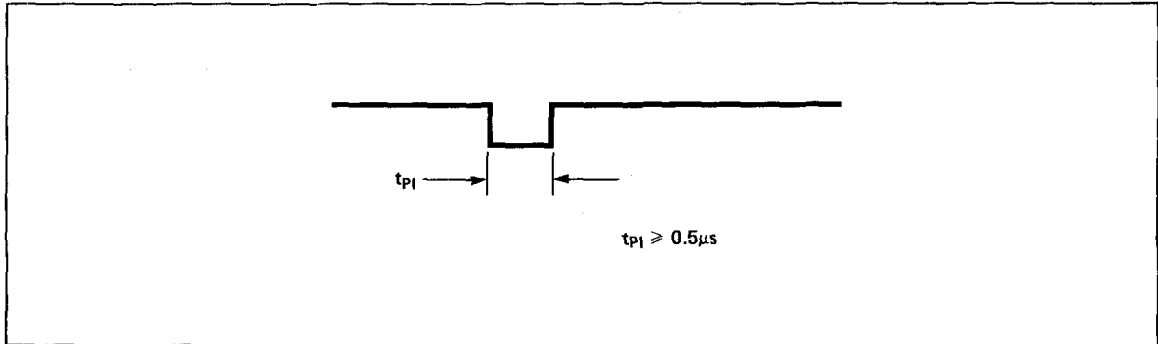


Figure 14. Head Load to Read/Write Timing

**Index**

The Index input is used to determine "Sector not found" status and to initiate format track/read ID commands and head unload Index and Count operations.



**Figure 15. Index Timing**

**Track 0**

This input pin indicates that the diskette is at track 0. During any seek operation, the stepping out of the actuator ceases when the track 0 pin becomes active.

**Select 1, 0**

Only one drive may be selected at a time. The Input/Output pins that must be externally qualified with Select 0 and Select 1 are:

- Unseparated Data
- Data Window
- Write Enable
- Seek/Step
- Count/Optional Input
- Load Head
- Track 0
- Low Current
- Write Protect
- Write Fault
- Fault Reset/Optional Output
- Index

When a new set of select bits is specified by a new command or the FDC finishes the index count before head unload, the following pins will be set to the 0 state:

- Write Enable (35)
- Seek/Step (36)
- Direction (37)
- Load Head (38)
- Low Head Current (39)

The select pins will be set to the state specified by the command or both are set to zero following the index count before head unload.

**Low Current**

This output pin is active whenever the physical track location of the selected drive is greater than 43. Generally

this signal is used to enable compensation for the lower velocities encountered while recording on the inner tracks.

**Write Protect**

The 8271 will not write to a disk when this input pin is active and will interrupt the CPU if a Write attempt is made. Operations which check Write Protect are aborted if the Write Protect line is active.

This signal normally originates from a sensor which detects the presence or absence of the Write Protect hole in the diskette jacket.

**Write Fault and Write Fault Reset**

The Write Fault input is normally latched by the drive and indicates any condition which could endanger data integrity. The 8271 interrupts the CPU anytime Write Fault is detected during an operation and immediately resets the Write Enable, Seek/Step, Direction, and Low Current signals. The write fault condition can be cleared by using the write fault reset pin. If the drive being used does not support write fault, then this pin should be connected to V<sub>CC</sub> through a pull-up resistor.

**Ready 1, 0**

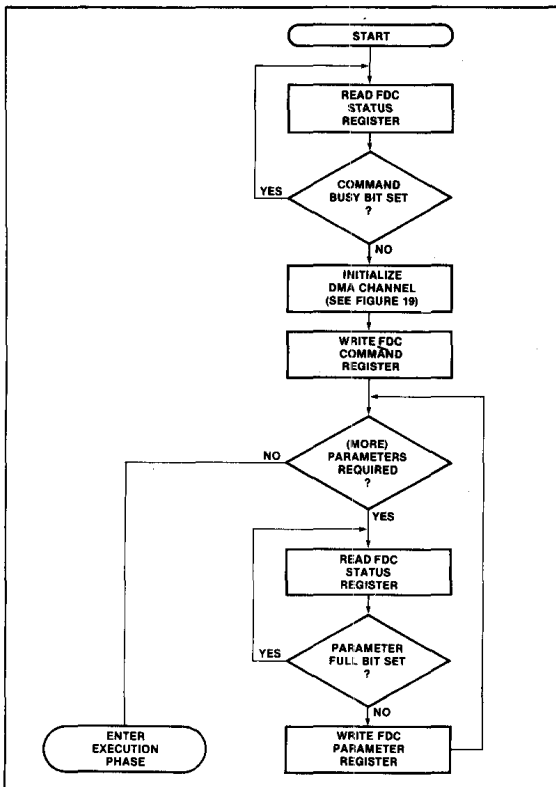
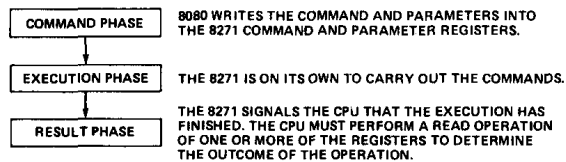
These two pins indicate the functional status of the disk drives. Whenever an operation is attempted on a drive which is not ready, an interrupt is generated. The interface continually monitors this input during an operation and if a Not Ready condition occurs, immediately terminates the operation. Note that the 8271 latches the Not Ready condition and it can only be reset by the execution of a Read Drive Status command. For drives that do not support a ready signal, either one can be derived with a one shot and the index pulse, or the ready inputs can be grounded and Ready determined through some software means.

**PRINCIPLES OF OPERATION**

As an 8080 peripheral device, the 8271 accepts commands from the CPU, executes them and provides a RESULT back to the 8080 CPU at the end of command execution. The communication with the CPU is established by the activation of CS and RD or WR. The A<sub>1</sub>, A<sub>0</sub> inputs select the appropriate registers on the chip:

DACK	CS	A <sub>1</sub>	A <sub>0</sub>	RD	WR	Operation
1	0	0	0	0	1	Read Status
1	0	0	0	1	0	Write Command
1	0	0	1	0	1	Read Result
1	0	0	1	1	0	Write Parameter
1	0	1	0	1	0	Write Reset Reg.
0	1	X	X	1	0	Write Data
0	1	X	X	0	1	Read Data
0	0	X	X	X	X	Not Allowed

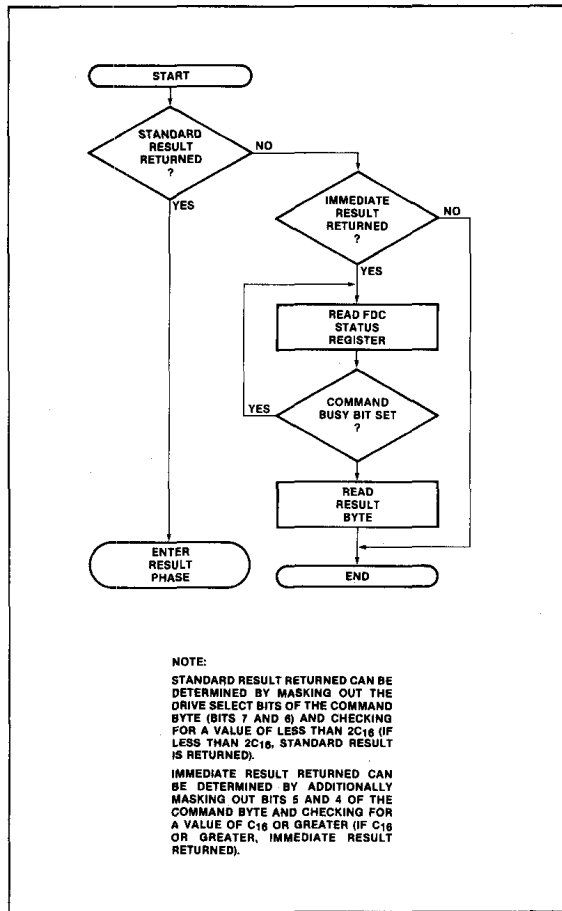
The FDC operation is composed of the following sequence of events.



**Figure 16. Passing the Command and Parameters to the 8271**

**The Command Phase**

The software writes a command to the command register. As a function of the command issued, from zero to five parameters are written to the parameter register. Refer to diagram showing a flow chart of the command phase. Note that the flow chart shows that a command may not be issued if the FDC status register indicates that the device is busy. Issuing a command while another command is in progress is illegal. The flow chart also shows a parameter buffer full check. The FDC status indicates the state of the parameter buffer. If a parameter is issued while the parameter buffer is full, the previous parameter is over written and lost.



**Figure 17. Checking for Result Type Following 8271 Command and Parameters**

**The Execution Phase**

During the execution phase the operation specified during the command phase is performed. During this phase, there is no CPU involvement if the system utilizes DMA for the data transfers. The execution phase of each command is discussed within the detailed command descriptions. The following table summarizes many of the basic execution phase characteristics.

**EXECUTION PHASE BASIC CHARACTERISTICS**

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

**Table 2. Execution Phase Basic 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	x	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 TRACK	x	LOAD	✓	✓	YES	NO	YES	YES
SEEK	x	LOAD	y	x	YES	NO	YES	YES
READ DRIVE STATUS SPECIFY	x	-	x	x	NO	NO	NOTE 5	NO
RESET	x	UNLOAD	x	x	NO	NO	NO	NO
R SP REGISTERS	x	-	x	x	NO	NO	NOTE 6	NO
W SP REGISTERS	x	-	x	x	NO	NO	NO	NO

Note: 1. "x" → DONT CARE 2. "✓" → check 3. "-" → No change 4. "y" → Check at end of operation 5. See "READ DRIVE STATUS" command.  
6. See "READ SPECIAL REGISTER" command.

Explanation of the execution phase characteristics table.

**1. Deleted Data Processing**

If deleted data is encountered during an operation that is marked skip in the table, the deleted data record is not transferred into memory, but the record is counted. For example, if the command and parameters specify a read of five records and one of the records was written with a deleted data mark, four records are transferred to memory. The deleted data flag is set in the result byte. However, if the operation is marked transfer, all data is transferred to memory regardless of the type of data mark.

**2. Head**

The Head column in the table specifies whether the Read/Write head will be loaded or not. If the table specifies load, the head is loaded after it is positioned over the track. The head loaded by a command remains loaded until the user specified number of index pulses have occurred.

**3. Ready**

The Ready column indicates if the ready line (Ready 1, Ready 0) associated with the selected drive is checked. A not ready state is latched by the 8271 until the user executes a read status command.

**4. Write Protect**

The operations that are marked check Write Protect are immediately aborted if Write Protect line is active at the beginning of an operation.

**5. Seek**

Many of the 8271 commands cause a seek to the desired track. A current track register is maintained for each drive or surface.

**6. Seek Check**

Operations that perform Seek Check verify that selected data in the ID field is correct before the 8271 accesses the data field.

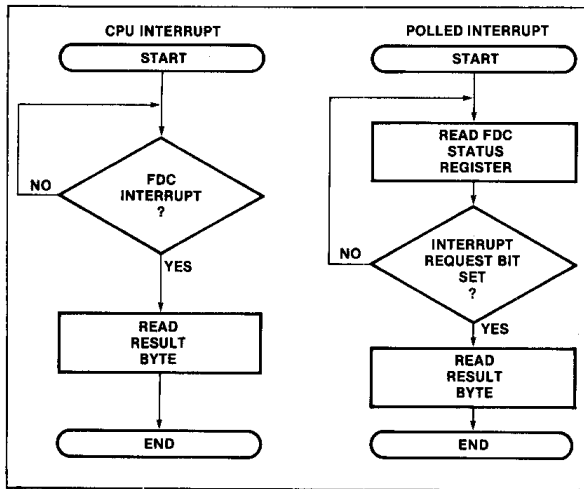


Figure 18. Getting the Result

**The Result Phase**

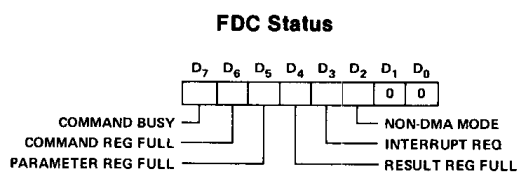
During the Result Phase, the FDC notifies the CPU of the outcome of the command execution. This phase may be initiated by:

1. The successful completion of an operation.
2. An error detected during an operation.

**PROGRAMMING**

A <sub>1</sub>	A <sub>0</sub>	CS RD	CS WR
0	0	Status Reg	Command Reg
0	1	Result Reg	Parameter Reg
1	0	—	Reset Reg
1	1	—	—

**STATUS REGISTER**



**Bit 7: Command Busy**

The command busy bit is set on writing to the command register. Whenever the FDC is busy processing a command, the command busy bit is set to a one. This bit is set to zero after the command is completed.

**Bit 6: Command Full**

The command full bit is set on writing to the command buffer and cleared when the FDC begins processing the command.

**Bit 5: Parameter Full**

This bit indicates the state of the parameter buffer. This bit is set when a parameter is written to the FDC and reset after the FDC has accepted the parameter.

**Bit 4: Result Full**

This bit indicates the state of the result buffer. It is valid only after Command Busy bit is low. This bit is set when the FDC finishes a command and is reset after the result byte is read by the CPU. The data in the result buffer is valid only after the FDC has completed a command. Reading the result buffer while a command is in progress yields no useful information.

**Bit 3: Interrupt Request**

This bit reflects the state of the FDC INT pin. It is set when FDC requests attention as a result of the completion of an operation or failure to complete an intended operation. This bit is cleared by reading the result register.

**Bit 2: Non-DMA Data Request**

When the FDC is utilized without a DMA controller, this bit is used to indicate FDC data requests. Note that in the non-DMA mode, an interrupt is generated (interrupt request bit is set) with each data byte written to or read from the diskette.

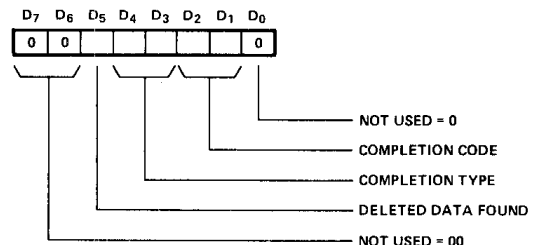
**Bits 1 and 0:**

Not used (zero returned).

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

**THE RESULT REGISTER**

This byte format facilitates the use of an address table to look up error routines and messages. The standard result byte format is:



**Bits 7 and 6:**

Not used (zero returned).

**Bit 5:**

Deleted Data Found: This bit is set when deleted data is encountered during a transaction.

**Bits 4 and 3: Completion Type**

The completion type field provides general information regarding the outcome of an operation.

The completion type field provides general information regarding the outcome of an operation.

Completion Type	Event
00	Good Completion — No Error
01	System Error — recoverable errors; operator intervention probably required for recovery.
10	Command/Drive Error — either a program error or drive hardware failure.
11	Command/Drive Error — either a program error or drive hardware failure.

**Bits 2 and 1: Completion Code**

The completion code field provides more detailed information about the completion type (See Table).

Completion Type	Completion Code	Event
00	00	Good Completion/ Scan Not Met
00	01	Scan Met Equal
00	10	Scan Met Not Equal
00	11	---
01	00	Clock Error
01	01	Late DMA
01	10	ID CRC Error
01	11	Data CRC Error
10	00	Drive Not Ready
10	01	Write Protect
10	10	Track 0 Not Found
10	11	Write Fault
11	00	Sector Not Found
11	01	---
11	10	---
11	11	---

It is important to note the hierarchical structure of the result byte. In very simple systems where only a GO-NO GO result is required, the user may simply branch on a zero result (a zero result is a good completion). The next level of complexity is at the completion type interface. The completion type supplies enough information so that the software may distinguish between fatal and non-fatal errors. If a completion type 01 occurs, ten retries should be performed before the error is considered unrecoverable.

The Completion Type/Completion Code interface supplies the greatest detail about each type of completion. This interface is used when detailed information about the transaction completion is required.

**Bit 0:**  
Not used (zero returned).

**Table 3. Completion Code Interpretation**

Definition	Interpretation
Successful Completion/ Scan Not Met	The diskette operation specified was completed without error. If scan operation was specified, the pattern scanned was not found on the track addressed.
Scan Met Equal	The data pattern specified with the scan command was found on the track addressed with the specified comparison, and the equality was met.
Scan Met Not Equal	The data pattern specified with the scan command was found with the specified comparison on the track addressed, but the equality was not met.
Clock Error	During a diskette read operation, a clock bit was missing (dropped). Note that this function is disabled when reading any of the ID address marks (which contain missing clock pulses). If this error occurs, the operation is terminated immediately and an interrupt is generated.
Late DMA	During either a diskette read or write operation, the data channel did not respond within the allotted time interval to prevent data from being overwritten or lost. This error immediately terminates the operation and generates an interrupt.
ID Field CRC Error	The CRC word (two bytes) derived from the data read in an ID field did not match the CRC word written in the ID field when the track was formatted. If this error occurs, the associated diskette operation is prevented and no data is transferred.
Data Field CRC Error	During a diskette read operation, the CRC word derived from the data field read did not match the data field CRC word previously written. If this error occurs, the data read from the sector should be considered invalid.
Drive Not Ready	The drive addressed was not ready. This indication is caused by any of the following conditions: 1. Drive not powered up 2. Diskette not loaded 3. Non-existent drive addressed 4. Drive went not ready during an operation Note that this completion code is cleared only through an FDC read drive status command.
Write Protect	A diskette write operation was specified on a write protected diskette. The intended write operation is prevented and no data is written on the diskette.
Track 00 Not Found	During a seek to track 00 operation, the drive failed to provide a track 00 indication after being stepped 255 times.
Write Fault	This error is dependent on the drive supported and indicates that the fault input to the FDC has been activated by the drive.
Sector Not Found	Either the sector addressed could not be found within one complete revolution of the diskette (two index marks encountered) or the track address specified did not match the track address contained in the ID field. Note that when the track address specified and the track address read do not match, the FDC automatically increments its track address register (stepping the drive to the next track) and again compares the track addresses. If the track addresses still do not match, the track address register is incremented a second time and another comparison is made before the sector not found completion code is set.



**INITIALIZATION**

**Reset Command**

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	0	0	0	0	0	0	0	1
PAR:	1	0	0	0	0	0	0	0	0	1
PAR:	1	0	0	0	0	0	0	0	0	0

Function: The Reset command emulates the action of the reset pin. It is issued by outputting a one followed by a zero to the Reset register.

1. The drive control signals are forced low.
2. An in-progress command is aborted.
3. The FDC status register flags are cleared.
4. The FDC enters an idle state until the next command is issued.

Reset must be active for 10 or more clock cycles.

**SPECIFY COMMAND**

Many of the interface characteristics of the FDC are specified by the systems software. Prior to initiating any drive operation command, the software must execute the three specify commands. There are two types of specify commands selectable by the first parameter issued.

**First Parameter Specify Type**

0DH	Initialization
10H	Load bad Tracks Surface '0'
18H	Load bad Tracks Surface '1'

The Specify command is used prior to performing any diskette operation (including formatting of a diskette) to define the drive's inherent operating characteristics and also is used following a formatting operation or installation of another diskette to define the locations of bad tracks. Since the Specify command only loads internal registers within the 8271 and does not involve an actual diskette operation, command processing is limited to only Command Phase. Note that once the operating characteristics and bad tracks have been specified for a given drive and diskette, redefining these values need only be done if a diskette with unique bad tracks is to be used or if the system is powered down.

**Initialization:**

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	0	0	1	1	0	1	0	1
PAR:	0	1	0	0	0	0	1	1	0	1
PAR:	0	1	STEP RATE*							
PAR:	0	1	HEAD SETTling TIME*							
PAR:	0	1	INDEX CNT BEFORE HEAD UNLOAD*				HEAD LOAD TIME*			

\*Note: Mini-floppy parameters are doubled.

- Parameter 0 — 0DH = Select Specify Initialization.
- Parameter 1 — D<sub>7</sub>-D<sub>0</sub> = Step Rate (0-255ms in 1ms steps).
- Parameter 2 — D<sub>7</sub>-D<sub>0</sub> = Head Settling Time (0-255ms in 1ms steps). {0 – 510ms in 2ms steps} () = standard, {} = mini
- Parameter 3 — D<sub>7</sub>-D<sub>4</sub> = Index Count — Specifies the number of Revolutions (0-14) which are to occur before the FDC automatically unloads the R/W head. If 15 is specified, the head remains loaded.
- D<sub>3</sub>-D<sub>0</sub> = Head Load Time (0-60ms in steps of 4ms). {0 – 120ms in 8ms steps} () = standard, {} = mini

**Load Bad Tracks**

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	0	0	1	1	0	1	0	1
PAR:	0	1	0	0	0	1	1/0	0	0	0
PAR:	0	1	BAD TRACK NO. 1							
PAR:	0	1	BAD TRACK NO. 2							
PAR:	0	1	CURRENT TRACK							

Parameter 0: 10H = Load Surface zero bad tracks  
18H = Load Surface one bad track

Parameter 1:  
Bad track address number 1 (Physical Address).

It is recommended to program both bad tracks and current track to FF<sub>H</sub> during initialization.

**SEEK COMMAND**

The seek command moves the head to the specified track without loading the head or verifying the track.

The seek operation uses the specified bad tracks to compute the physical track address. This feature insures that the seek operation positions the head over the correct track.

When a seek to track zero is specified, the FDC steps the head until the track 00 signal is detected.

If the track 00 signal is not detected within (FF)<sub>H</sub> steps, a track 0 not found error status is returned.

A seek to track zero is used to position the read/write head when the current head position is unknown (such as after a power up).

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	SEL 1	SEL 0	1	0	1	0	0	1
PAR:	0	1	TRACK ADDRESS 0 255							

Seek operations are not verified. A subsequent read or write operation must be performed to determine if the correct track is located.

**READ DRIVE STATUS COMMAND**

This command is used to interrogate the drive status. Upon completion the result register will hold the final drive status.

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	SEL 1	SEL 0	1	0	1	1	0	0

RESULT: EACH BIT INDICATES CURRENT STATE OF INPUT PINS.

A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	1	RDY 1*	WR FAULT	INDEX	WR PROT	RDY 0*	TRACK CNT	0	GPI

IF A DRIVE NOT READY RESULT IS RETURNED, THE READ STATUS MUST BE ISSUED TO CLEAR THE CONDITION.

\*Note the two ready bits are zero latching. Therefore, to clear the drive not ready condition, assuming the drive is ready, and to detect it via software, one must issue this command twice.

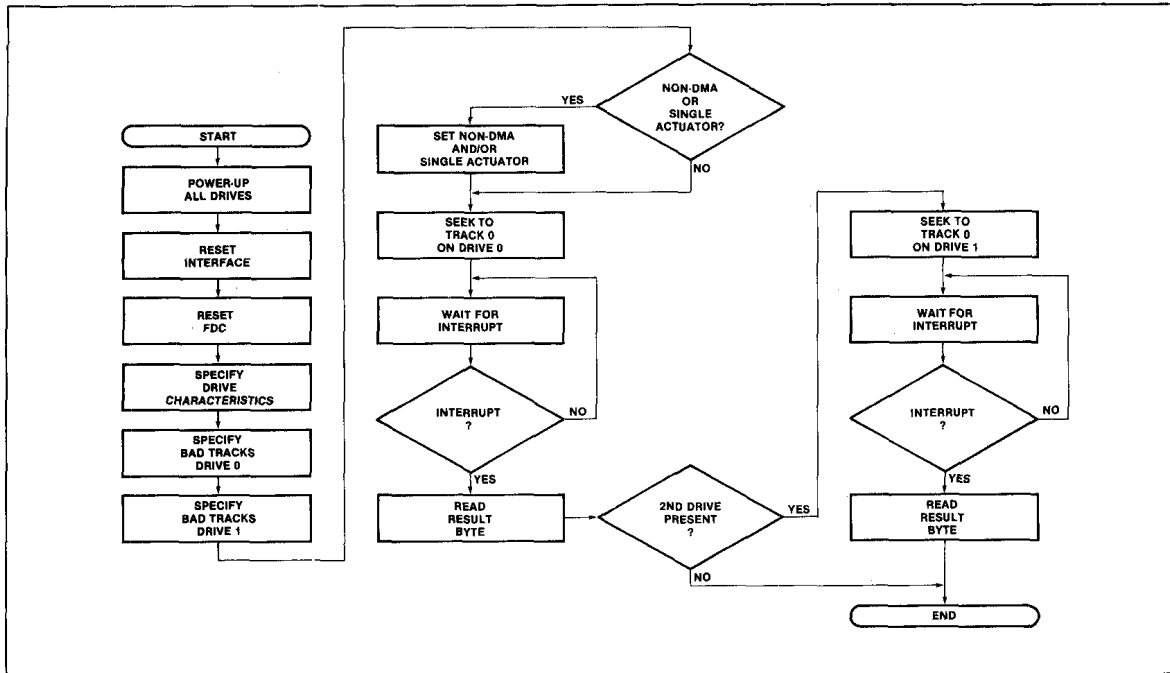


Figure 19. Initialization of the 8271 by the User

**Read/Write Special Registers**

This command is used to access special registers within the 8271.

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	SEL 1	SEL 0	COMMAND OPCODE					
PAR:	0	1	REGISTER ADDRESS							

Command code:

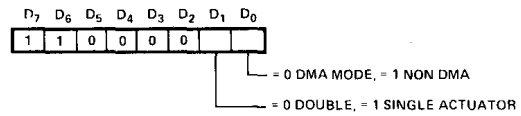
- 3DH Read Special Register
- 3AH Write Special Register

For both commands, the first parameter is the register address; for Write commands a second parameter specifies data to be written. Only the Read Special Register command supplies a result.

Table 4. Special Registers

Description	Register Address in Hex	Comment
Scan Sector Number	06	See Scan Description
Scan MSB of Count	14	See Scan Description
Scan LSB of Count	13	See Scan Description
Surface 0 Current Track	12	
Surface 1 Current Track	1A	
Mode Register	17	See Mode Register Description
Drive Control Output Port	23	See Drive Output Port Description
Drive Control Input Port	22	See Drive Input Port Description
Surface 0 Bad Track 1	10	
Surface 0 Bad Track 2	11	
Surface 1 Bad Track 1	18	
Surface 1 Bad Track 2	19	

**Mode Register Write Parameter Format**



**Bits 6 & 7**

Must be one.

**Bits 5-2**

(Not used). Must be set to zero.

**\*Bit 1**

Double/Single Actuator: Selects single or double actuator mode. If the single actuator mode is selected, the FDC assumes that the physical track location of both disks is always the same. This mode facilitates control of a drive which has a single actuator mechanism to move two heads.

**\*Bit 0**

Data Transfer Mode: This bit selects the data transfer mode. If this bit is a zero, the FDC operates in the DMA mode (DMA Request/ACK). If this bit is a one, the FDC operates in non-DMA mode. When the FDC is operating in DMA mode, interrupts are generated at the completion of commands. If the non-DMA mode is selected, the FDC generates an interrupt for every data byte transferred.

\*Bits 0 and 1 are initialized to zero.



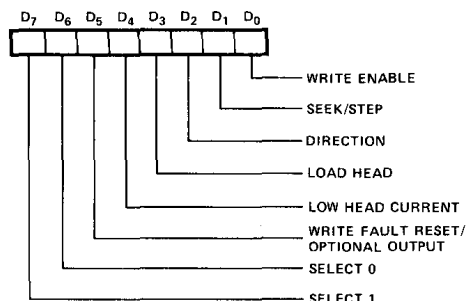
### Non-DMA Transfers in DMA Mode

If the user desires, he may retain the use of interrupts generated upon command completions. This mode is accomplished by selecting the DMA capability, but using the DMA REQ/ACK pins as effective INT and CS signals, respectively.

### Drive Control Input Port

Reading this port will give the CPU exactly the data that the FDC sees at the corresponding pins. Reading this port will update the drive not ready status, but will not clear the status. (See Read Drive Status Command for Bit locations.)

### Drive Control Output Port Format



Each of these signals correspond to the chip pin of the same name. On standard-sized drives with write fault detection logic, bit 5 is set to generate the write fault reset signal. This signal is used to clear a write fault indication within the drive. On mini-sized drives, this bit can be used to turn on or off the drive motor prior to initiating a drive operation. A time delay after turn on may be necessary for the drive to come up to speed. The register must be read prior to writing the register in order to save the states of the remaining bits. When the register is subsequently written to modify bit 5, the remaining bits must be restored to their previous states.

### IBM DISKETTE GENERAL FORMAT INFORMATION

The IBM Flexible Diskette used for data storage and retrieval is organized into concentric circular paths or TRACKS. There are 77 tracks on either one or both sides (surfaces) of the diskette. On double-sided diskettes, the corresponding top and bottom tracks are referred to as a CYLINDER. Each track is further divided into fixed length sections or SECTORS. The number of sectors per track — 26, 15 or 8 — is determined when a track is formatted and is dependent on the sector length — 128, 256 or 512 bytes respectively — specified.

All tracks on the diskette are referenced to a physical index mark (a small hole in the diskette). Each time the hole passes a photodetector cell (one revolution of the diskette), an Index pulse is generated to indicate the logical beginning of a track. This index pulse is used to initiate a track formatting operation.

### Track Format

Each Diskette Surface is divided into 77 tracks with each track divided into fixed length sectors. A sector can hold a whole record or a part of a record. If the record is shorter than the sector length, the unused bytes are filled with binary zeros. If a record is longer than the sector length, the record is written over as many sectors as its length requires. The sector size that provides the most efficient use of diskette space can be chosen depending upon the record length required.

Tracks are numbered from 00 (outer-most) to 76 (inner-most) and are used as follows:

- TRACK 00 reserved as System Label Track
- TRACKS 01 through 74 used for data
- TRACKS 75 and 76 used as alternates.

Each sector consists of an ID field (which holds a unique address for the sector) and a data field.

The ID field is seven bytes long and is written for each sector when the track is formatted. Each ID field consists of an ID field Address Mark, a Cylinder Number byte which identifies the track number, a Head Number byte which specifies the head used (top or bottom) to access the sector, a Record Number byte identifying the sector number (1 through 26 for 128 byte sectors), an N-byte specifying the byte length of the sector and two CRC (Cyclic Redundancy Check) bytes.

The Gaps separating the index mark and the ID and data fields are written on a track when it is formatted. These gaps provide both an interval for switching the drive electronics from reading or writing and compensation for rotational speed and other diskette-to-diskette and drive-to-drive manufacturing tolerances to ensure that data written on a diskette by one system can be read by another (diskette interchangeability).

### IBM Format Implementation Summary

#### Track Format

The disk has 77 tracks, numbered physically from 00 to 76, with track 00 being the outermost track. There are logically 75 data tracks and two alternate tracks. Any two tracks may be initialized as bad tracks. The data tracks are numbered logically in sequence from 00 to 74, skipping over bad tracks (alternate tracks replace bad tracks). Note: In IBM format track 00 cannot be a bad track.

#### Sector Format

Each track is divided into 26, 15, or 8 sectors of 128, 256, or 512 bytes length respectively. The first sector is numbered 01, and is physically the first sector after the physical index mark. The logical sequence of the remaining sectors may be nonsequential physically. The location of these is determined at initialization by CPU software.

Each sector consists of an ID field and a data field. All fields are separated by gaps. The beginning of each field is indicated by 6 bytes of (00)H followed by a one byte address mark.

#### Address Marks

Address Marks are unique bit patterns one byte in length which are used to identify the beginning of ID and Data fields. Address Mark bytes are unique from all other data

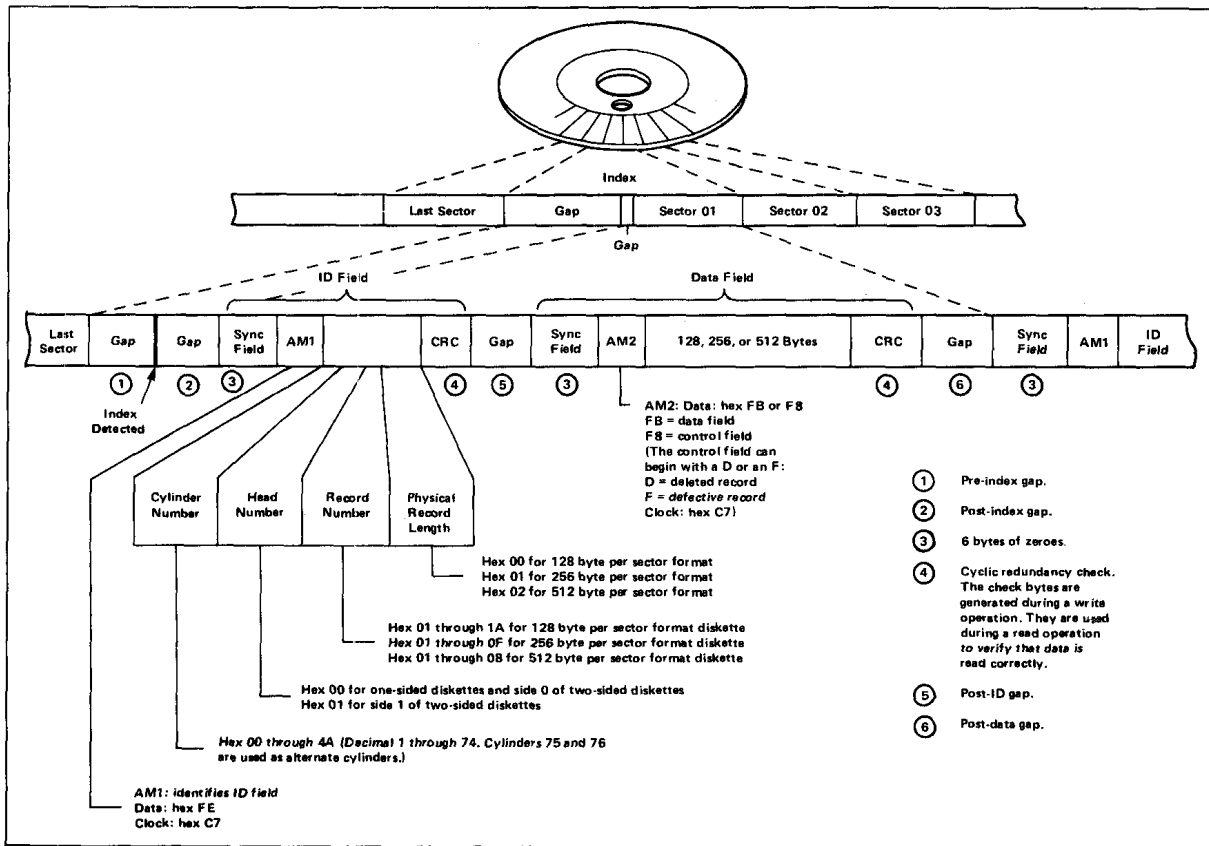


Figure 20. Track Format

bytes in that certain bit cells do not contain a clock bit (all other data bytes have clock bits in every bit cell.) There are four different types of Address Marks used. Each of these is used to identify different types of fields.

**Index Address Mark**

The Index Address Mark is located at the beginning of each track and is a fixed number of bytes in front of the first record.

**ID Address Mark**

The ID Address Mark byte is located at the beginning of each ID field on the diskette.

**Data Address Mark**

The Data Address Mark byte is located at the beginning of each non-deleted Data Field on the diskette.

**Deleted Data Address Mark**

The Deleted Data Address Mark byte is located at the beginning of each deleted Data Field on the diskette.

Address Mark Summary	Clock Pattern	Data Pattern
Index Address Mark	D7	FC
ID Address Mark	C7	FE
Data Address Mark	C7	FB
Deleted Data Address Mark	C7	F8
Bad Track ID Address Mark	C7	FE

**ID Field**

MARK	C	H	R	N	CRC	CRC
------	---	---	---	---	-----	-----

- C = Cylinder (Track) Address, 00-74
- H = Head Address
- R = Record (Sector) Address, 01-26
- N = Record (Sector) Length, 00-02
- Note: Sector Length =  $128 \times 2^N$  bytes
- CRC = 16 Bit CRC Character (See Below)

**Data Field**

MARK	DATA	CRC	CRC
------	------	-----	-----

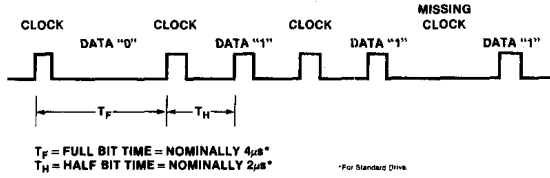
Data is 128, 256, or 512 bytes long.  
Note: All marks, data, ID characters and CRC characters are recorded and read most significant bit first.

**CRC Character**

The 16-bit CRC character is generated using the generator polynomial  $X^{16} + X^{12} + X^5 + 1$ , normally initialized to (FF)<sub>H</sub>. It is generated from all characters (except the CRC in the ID or data field), including the data (not the clocks) in the address mark. It is recorded and read most significant bit first.

**Data Format**

Data is written (general case) in the following manner:



**References**

"The IBM Diskette for Standard Data Interchange," IBM Document GA21-9182-0. "System 32," Chapter 8, IBM Document GA21-9176-0.

**Bad Track Format**

The Bad Track Format is the same as the good track format except that the bad track ID field is initialized as follows:

$$C = H = R = N = (FF)_H$$

When formatting, bad track registers should be set to  $FF_H$  for the drive during the formatting, thus specifying no bad tracks. Thus, all tracks are left available for formatting.

The track following the bad track(s) should be one higher in number than track before the bad track(s).

Upon completion of the format the bad tracks should be set up using the write special register command. The 8271 will then generate an extra step pulse to cross the bad track, locating a new track that now happens to be an extra track out.

**Format Track**

		Format Command									
		A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:		0	0	SEL 1	SEL 0	1	0	0	0	1	1
PAR:		0	1	TRACK ADDRESS							
PAR:		0	1	GAP 3 SIZE MINUS 6							
PAR:		0	1	RECORD LENGTH				NO. OF SECTORS/TRACK			
PAR:		0	1	GAP 5 SIZE MINUS 6							
PAR:		0	1	GAP 1 SIZE MINUS 6							

The format command can be used to initialize a disk track compatible with the IBM 3740 format. A Shugart "IBM Type" mini-floppy format may also be generated.

The Format command can be used to initialize a diskette, one track at a time. When format command is used, the program must supply ID fields for each sector on the track. During command execution, the supplied ID fields (track head sector addresses and the sector length) are written sequentially on the diskette. The ID address marks originate from the 8271 and are written automatically as the first byte of each ID field. The CRC character is written in the last two bytes of the ID field and is derived from the data written in the first five bytes. During the formatting operation, the data field of each sector is filled with data pattern  $(E5)_H$ . The CRC, derived from the data pattern is also appended to the last byte.

1. The parameter 2 ( $D_7 - D_5$ ) of the Format command specify record length, the bits are coded the same way as in the Read Data commands.
2. The programmable gap sizes (gap 3, gap 5, and gap 1) must be programmed such that the 6 bytes of zero (sync) are subtracted from the intended gap size i.e., if gap 1 is intended to be 16 bytes long, programmed length must be  $16 - 6 = 10$  bytes (of  $FF_H$ 's).

**Mini-Floppy Disk Format**

The mini-floppy disk format differs from the standard disk format in the following ways:

1. Gap 5 and the Index Address mark have been eliminated.
2. There are fewer sectors/tracks.

**GAPS**

The following is the gap size and description summary:

- Gap 1 Programmable
- Gap 2 17 Bytes
- Gap 3 Programmable
- Gap 4 Variable
- Gap 5 Programmable

The last six bytes of gaps 1,2,3 and 5 are  $(00)_H$ , all other bytes in the gaps are  $(FF)_H$ . The Gap 1,3 and 5 count specified by the user are the number of bytes of  $(FF)_H$ . Gap 4 is written until the leading edge of the index pulse. If a Gap 5 size of zero is specified, the Index Mark is not written.

- Gap 1: N bytes  $FF$ 's  
6 bytes 0's for sync

This gap separates the index address mark of the index pulse from the first ID mark. It is used to protect the first ID field from a write on the last physical sector of the current track.
- Gap 2: 11 bytes  $FF$ 's  
6 bytes 0's for sync

This gap separates the ID field from the data mark and field such that during a write only the data field will be changed even if the write gate turns on early, due to drive speed changes.
- Gap 3: N bytes  $FF$ 's  
6 bytes 0's for sync

This gap separates a data area from the next ID field. It is used so that the next ID mark will not be overwritten, thus causing loss of data.
- Gap 4:  $FF$ 's only

This gap fills out the rest of the disk and is used for slack during formatting. During drive speed variations this gap will shrink or grow if the disk is re-formatted.
- Gap 5: N bytes  $FF$ 's  
6 bytes 0's for sync

This gap separates the last sector from the Index Address mark and is used to assure that the index address mark is not destroyed by writing on the last physical data sector on the track.

The number of  $FF$  bytes is programmable for gaps 1, 3 and 5.

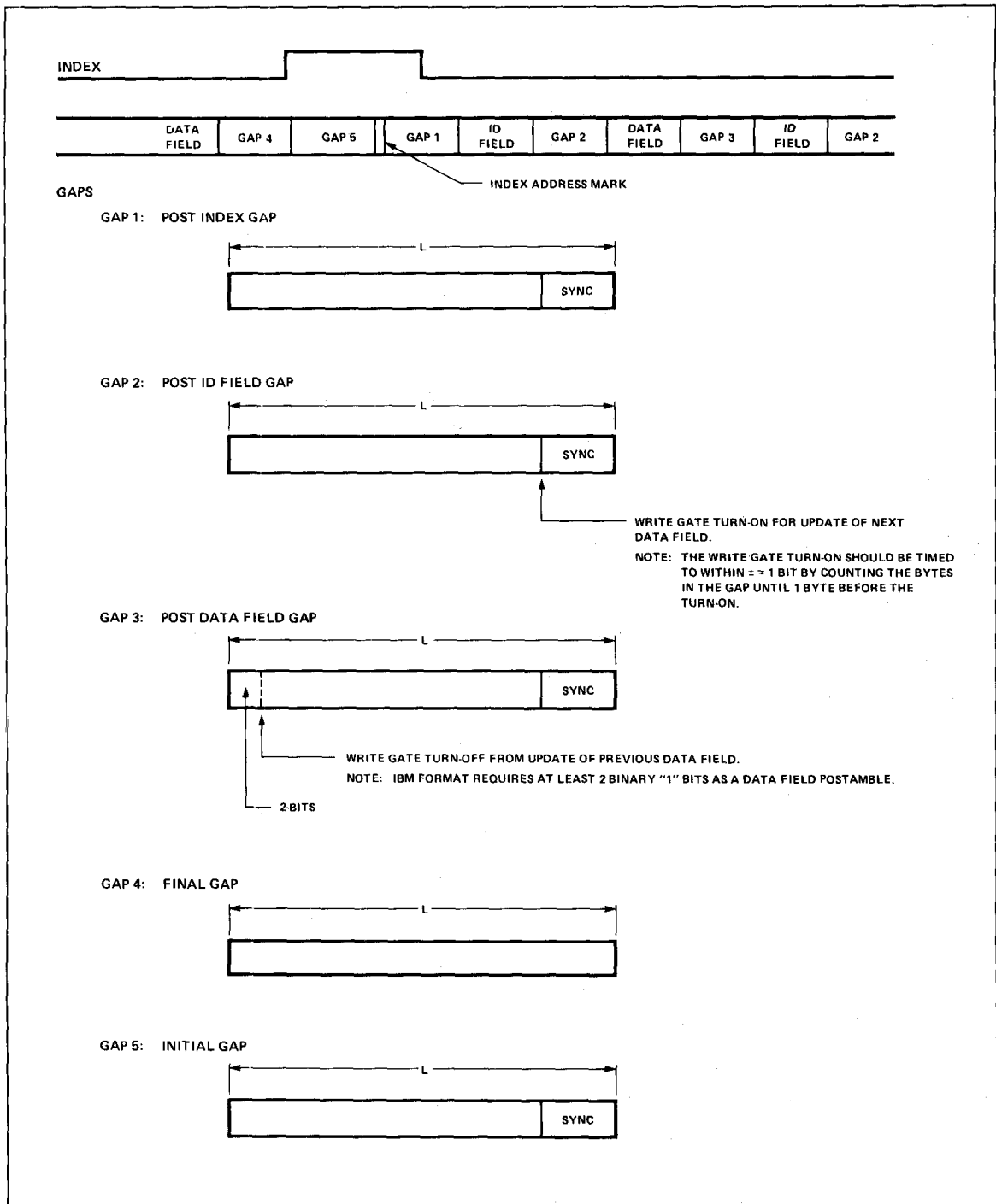


Figure 21. Track Format

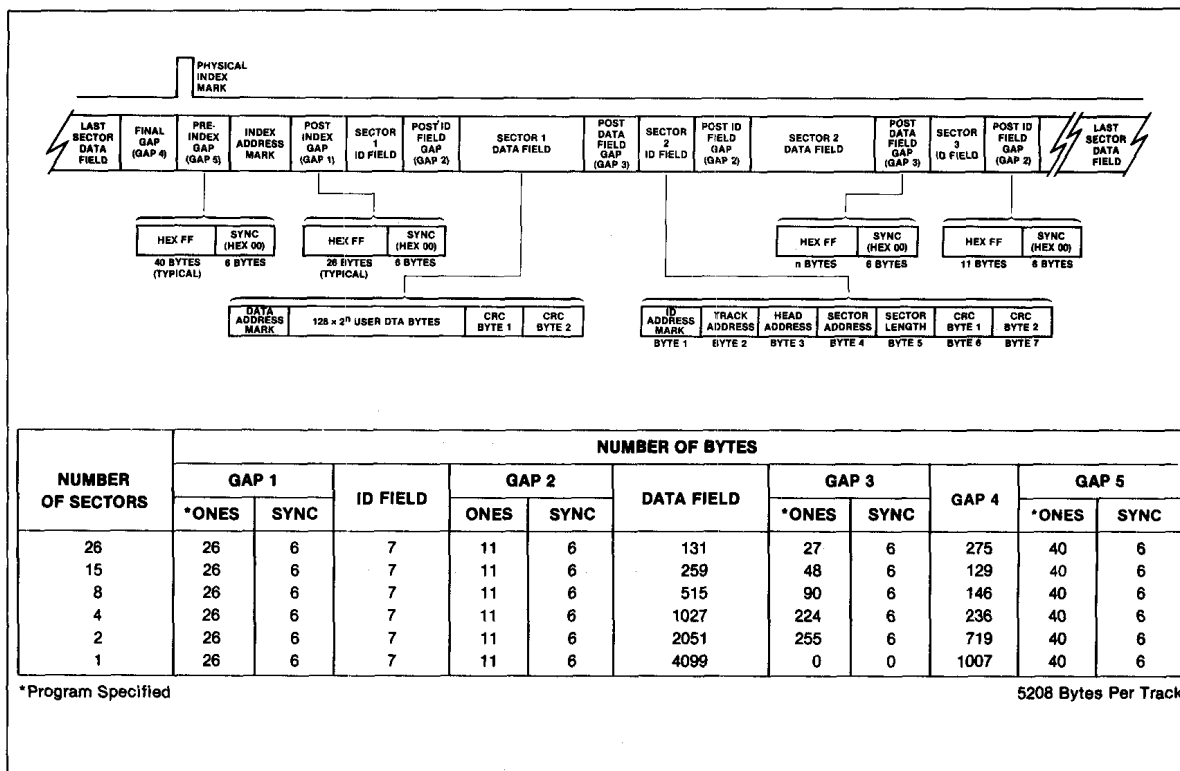


Figure 22. Standard Diskette Track Format

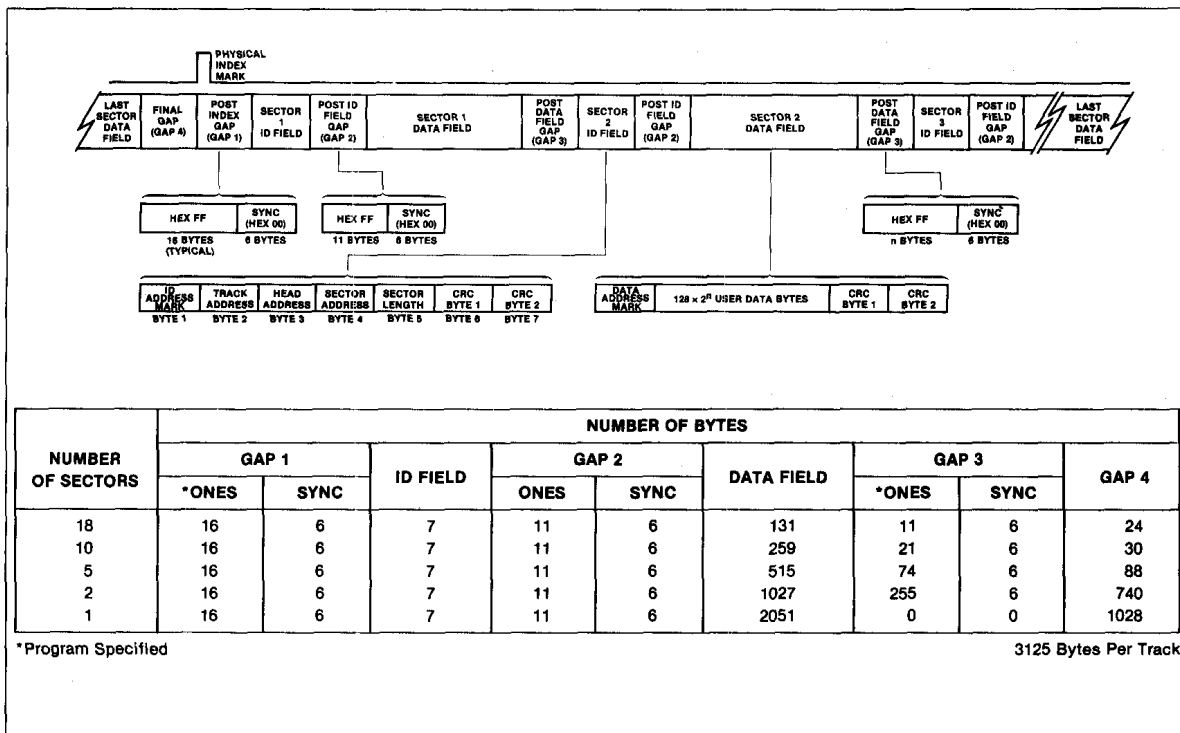


Figure 23. Mini-Diskette Track Format

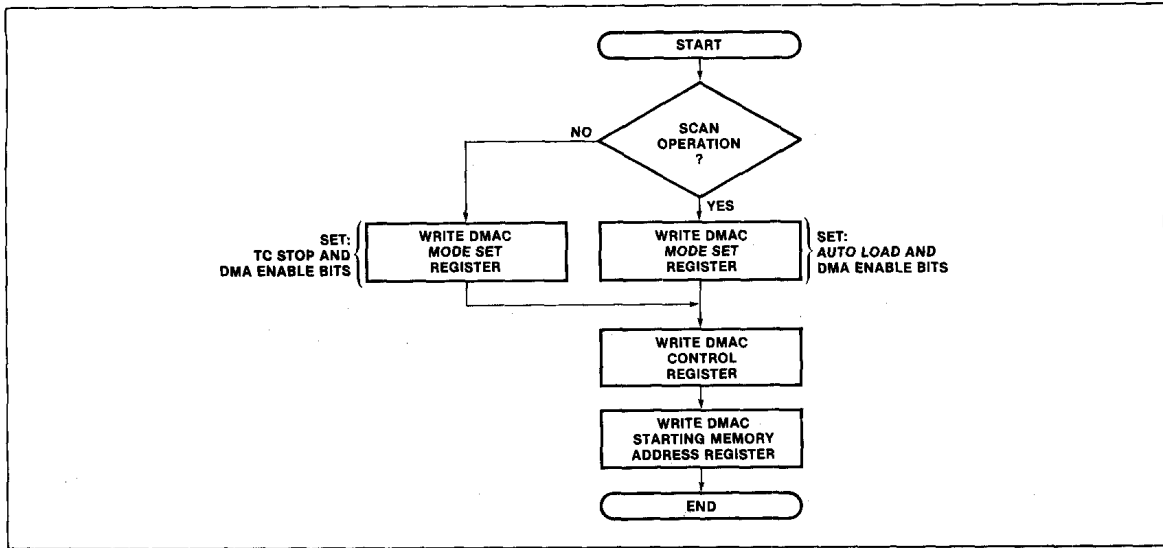


Figure 24. User DMA Channel Initialization Flowchart

**Read ID Command**

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	SEL <sub>1</sub>	SEL <sub>0</sub>	0	1	1	0	1	1
PAR:	0	1	TRACK ADDRESS							
PAR:	0	1	0	0	0	0	0	0	0	0
PAR:	0	1	NUMBER OF ID FIELDS							

The Read ID command transfers the specified number of ID fields into memory (beginning with the first ID field after Index). The CRC character is checked but not transferred.

These fields are entered into memory in the order in which they are physically located on the disk, with the first field being the one starting at the index pulse.

**Data Processing Commands**

All the routine Read/Write commands examine specific drive status lines before beginning execution, perform an implicit seek to the track address and load the drive's read/write head. Regardless of the type of command (i.e., read, write or verify), the 8271 first reads the ID field(s) to verify that the correct track has been located (see sector not found completion code) and also to locate the addressed sector. When a transfer is complete (or cannot be completed), the 8271 sets the interrupt request bit in the status register and provides an indication of the outcome of the operation in the result register.

If a CRC error is detected during a multisector transfer, processing is terminated with the sector in error. The address of the failing sector number can be determined by examining the Scan Sector Number register using the Read Special Register command.

Full power of the multisector read/write commands can be realized by doing DMA transfer using Intel® 8257 DMA Controller. For example, in a 128 byte per sector multisector write command, the entire data block (containing 128 bytes times the number of sectors) can be located in a disk memory buffer. Upon completion of the command phase, the 8271 begins execution by accessing the desired track, verifying the ID field, and locating the data field of the first record to be written. The 8271 then DMA-accesses the first sector and starts counting and writing one byte at a time until all 128 bytes are written. It then locates the data field of the next sector and repeats the procedure until all the specified sectors have been written. Upon completion of the execution phase the 8271 enters into the result phase and interrupts the CPU for availability of status and completion results. Note that all read/write commands, single or multisector are executed without CPU intervention.

Note, execution of multi-sector operations are faster if the sectors are *not* interleaved.

**128 Byte Single Record Format**

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	SEL <sub>1</sub>	SEL <sub>0</sub>	COMMAND OPCODE					
PAR:	0	1	TRACK ADDR 0-255							
PAR:	0	1	SECTOR 0-255							

**Commands**

- READ DATA 12
- READ DATA AND DELETED DATA 16
- WRITE DATA 0A
- WRITE DELETED DATA 0E
- VERIFY DATA AND DELETED DATA 1E

**Opcode**



**Variable Length/Multi-Record Format**

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	SEL 1	SEL 0	COMMAND OPCODE					
PAR:	0	1	TRACK ADDR 0-255							
PAR:	0	1	SECTOR 0-255							
PAR:	0	1	LENGTH				NO. OF SECTORS			

D<sub>7</sub>-D<sub>5</sub> of Parameter 2 determine the length of the disk record.

0 0 0	128 Bytes
0 0 1	256 Bytes
0 1 0	512 Bytes
0 1 1	1024 Bytes
1 0 0	2048 Bytes
1 0 1	4096 Bytes
1 1 0	8192 Bytes
1 1 1	16,384 Bytes

Commands	Opcode
READ DATA	13
READ DATA AND DELETED DATA	17
WRITE DATA	0B
WRITE DELETED DATA	0F
VERIFY DATA AND DELETED DATA	1F
SCAN DATA	00
SCAN DATA AND DELETED DATA	04

**Read Commands**

Read Data, Read Data and Deleted Data.

**Function**

The read command transfers data from a specified disk record or group of records to memory. The operation of this command is outlined in execution phase table.

**Write Commands**

Write Data, Write Deleted Data.

**Function**

The write command transfers data from memory to a specified disk record or group of records.

**Verify Command**

Verify Data and Deleted Data.

**Function**

The verify command is identical to the read data and deleted data command except that the data is not transferred to memory. This command is used to check that a record or a group of records has been written correctly by verifying the CRC character.

**Scan Commands**

	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CMD:	0	0	SEL 1	SEL 0	0	0	0	S.DATA S.DELO	0	0
PAR:	0	1	TRACK ADDR 0-255							
PAR:	0	1	SECTOR 0-255							
PAR:	0	1	LENGTH				NO. OF SECTORS			
PAR:	0	1	SCAN TYPE				STEP SIZE			
PAR:	0	1	FIELD LENGTH (KEY)							

Command D<sub>2</sub> = 0 Scan Data  
 D<sub>2</sub> = 1 Scan Data and Deleted Data

Scan Commands, Scan Data and Scan Data and Deleted Data, are used to search a specific data pattern or "key" from memory. The 8271 FDC operation during a scan is unique in that data is read from memory and from the diskette simultaneously.

During the scan operation, the key is compared repetitively (using the 8257 DMA Controller in auto load mode) with the data read from the diskette (e.g., an eight byte key would be compared with the first eight bytes (1-8) read from the diskette, the second eight bytes (9-16), the third eight bytes (17-24), etc.). The scan operation is concluded when the key is located or when the specified number of sectors have been searched without locating the key. When concluded, the 8271 FDC requests an interrupt. The program must then read the result register to determine if the scan was successful (if the key was located). If successful, several of the FDC's special registers can be examined (read special registers command) to determine more specific information relating to the scan (i.e., the sector number in which the key was located, and the number of bytes within the sector that were not compared when the key was located).

The 8271 does not do a sliding scan, it does a fixed block linear search. This means the key in memory is compared to an equal length block in a sector; when these blocks meet the scan conditions the scan will stop. Otherwise, the scan continues until all the sectors specified have been searched.

The following factors regarding key length must be considered when establishing a key in memory.

1. When searching multiple sectors, the length of the key must be evenly divisible into the sector length to prevent the key from being split at subsequent sector boundaries. Since the character FF<sub>H</sub> is not compared, the key in memory can be padded to the required length using this character. For example, if the actual pattern compared on the diskette is twelve characters in length, the field length should be sixteen and four bytes of FF<sub>H</sub>

would be appended to the key. Consequently, the last block of sixteen bytes compared within the first sector would end at the sector boundary and the first byte of the next sector would be compared with the first byte of the key. Splitting data over sector boundaries will not work properly since the FDC expects the start of key at each sector boundary.

- Since the first byte of the key is compared with the first byte of the sector, when the pattern does not begin with the first byte of the sector, the key must be offset using the character FF<sub>16</sub>. For example, if the first byte of a nine byte pattern begins on the fifth byte of the sector, four bytes of FF<sub>16</sub> are prefixed to the key (and three bytes of FF<sub>16</sub> are appended to the key to meet the length requirement) so that the first actual comparison begins on the fifth byte.

The Scan Commands require five parameters:

**Parameter 0, Track Address**

Specifies the track number containing the sectors to be scanned. Legal values range from 00<sub>H</sub> to 4C<sub>H</sub> (0 to 76) for a standard diskette and from 00<sub>H</sub> to 22<sub>H</sub> (0 to 34) for a mini-sized diskette.

**Parameter 1, Sector Address**

Specifies the first sector to be scanned. The number of sectors scanned is specified in parameter 2, and the order in which sectors are scanned is specified in parameter 3.

**Parameter 2, Sector Length/Number of Sectors**

The sector length field (bits 7-5) specifies the number of data bytes allocated to each sector (see parameter 2, routine read and write commands for field interpretation). The number of sectors field (bits 4-0) specifies the number of sectors to be scanned. The number specified ranges from one sector to the physical number of sectors on the track.

**Parameter 3**

- D<sub>7</sub>-D<sub>6</sub>: Indicate scan type
- 00-EQ Scan for each character within the field length (key) equal to the corresponding character within the disk sector. The scan stops after the first equal condition is met.
  - 01-GEQ Scan for each character within the disk sector greater than or equal to the corresponding character within the field length (key). The scan stops after the first greater than or equal condition is met.

- 10-LEQ Scan for each character within the disk sector less than or equal to the corresponding character within the field length (key). The scan stops after the first less than or equal condition is met.

- D<sub>5</sub>-D<sub>0</sub>: Step Size: The Step Size field specifies the offset to the next sector in a multisector scan. In this case, the next sector address is generated by adding the Step Size to the current sector address.

**Parameter 4, Field Length**

Specifies the number of bytes to be compared (length of key). While the range of legal values is from 1 to 255, the field length specified should be evenly divisible into the sector length to prevent the key from being split at sector boundaries, if the multisector scan commands are used.

**Scan Command Results**

More detailed information about the completion of Scan Commands may be obtained by executing Read Special Register commands.

**Read Special Register**

Parameter (Hex)	Results
06	The <u>sector number</u> of the sector in which the specified scan data pattern was located.
14	MSB Count — The number of 128 byte blocks remaining to be compared in the current sector when the scan data pattern was located. This register is decremented with each 128 byte block read.
13	LSB Count — The number of bytes remaining to be compared in the current sector when the scan data pattern is located. This register is initialized to 128 and is decremented with each byte compared.

Upon a scan met condition, the equation below can be used to determine the last byte in the located pattern.

$$\text{Pointer} = \text{sector length} - ((\text{Register 14H}) * 128 + (\text{Register 13H}))$$





**8271 Scan Command Example**

Assume there are only 2 records on track 0 with the following data:

Record 01: 01 02 03 04 05 06 07 08 000....00  
 Record 02: 01 02 AA 55 00 00 00 00 .....00

Command	Field Length <sup>[1]</sup>	Starting Sector #	# of Sectors	Key <sup>[2]</sup>	Completion Code <sup>[3]</sup>	Special Registers <sup>[4]</sup>			Comment
						R06	R14	R13	
* SCAN EQ	2	1	1	01,02	SME	01	0	127D	Met in first field
SCAN EQ	2	1	1	02,03	SNM	X	X	X	Not met
SCAN EQ	2	1	1	FF <sup>[5]</sup> ,05	SNM	X	X	X	Not met with don't care
* SCAN EQ	2	1	1	FF <sup>[5]</sup> ,06	SME	01	0	123D	Met with don't care
* SCAN EQ	2	1	2	AA,55	SME	02	0	125D	Met in Record 02
* SCAN EQ	2	2	1	01,02	SME	02	0	127D	Starting sector ≠ 1
* SCAN EQ	4	1	1	05,06,07,08	SME	01	0	121D	Field, Key length = 4
* SCAN GEQ	4	1	1	05,06,07,08	SME	01	0	121D	GEQ-SME
* SCAN GEQ	4	1	1	05,04,07,08	SMNE	01	0	121D	GEQ-SMNE
* SCAN GEQ	4	1	2	00,03,AA,44 <sup>[6]</sup>	SNM	X	X	X	GEQ-SNM
* SCAN LEQ	4	1	1	01,03,FF,04	SMNE	01	0	125D	LEQ-SMNE
* SCAN LEQ	4	1	1	01,02,FF,04	SME	01	0	125D	LEQ-SME

**NOTES:**

- Field Length — Each record is partitioned into a number of fields equal to the record size divided by the field length. Note that the record size should be evenly divisible by the field length to insure proper operation of multi record scan. Also, maximum field length = 256 bytes.
- Key — The key is a string of bytes located in the user system memory. The key length should equal the field length. By programming the 8257 DMA Controller into the auto load mode, the key will be recursively read in by the chip (once per field).
- Completion Code — Shows how Scan command was met or not met.  
 SNM — SCAN Not Met — 0 0 (also Good Complete)  
 SME — SCAN Met Equal — 0 1  
 SMNE — SCAN Met Not Equal — 1 0
- Special Registers  
 R06 — This register contains the record number where the scan was met.  
 R14 — This register contains the MSB count and is decremented every 128 characters.

Length (ℓ) (D7-D5 of PAR 2)	Record Size	R14 = 2ℓ - 1 (Initialize at Beginning of Record)
000	128 Bytes	0
001	256 Bytes	1
010	512 Bytes	3
011	1024 Bytes	7
•	•	•
•	•	•
•	•	•

R13 — This register contains a modulo 128 LSB count which is initialized to 128 at beginning of each record. This count is decremented after each character is compared except for the last character in a pattern match situation.

- The OFF<sub>H</sub> character in the key is treated as a don't care character position.
- The Scan comparison is done on a byte by byte basis. That is, byte 1 of each field is compared to byte 1 of the key, byte 2 of each field is compared to byte 2 of the key, etc.



**ABSOLUTE MAXIMUM RATINGS\***

Ambient Temperature Under Bias.....0°C to 70°C<sup>1</sup>  
 Storage Temperature..... - 65°C to + 150°C  
 Voltage on Any Pin with  
 Respect to Ground..... - 0.5V to + 7V  
 Power Dissipation.....1 Watt

*\*NOTICE: 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. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

**D.C. CHARACTERISTICS** ( $V_{CC} = +5.0V \pm 5\%$   
 8271 and 8271-8:  $T_A = 0^\circ C$  to  $70^\circ C$ ; 8271-6:  $T_A = 0^\circ C$  to  $50^\circ C$ )

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
$V_{IL}$	Input Low Voltage	-0.5	0.8	V	
$V_{IH}$	Input High Voltage	2.0	( $V_{CC} + 0.5$ )	V	
$V_{OLD}$	Output Low Voltage (Data Bus)		0.45	V	$I_{OL} = 2.0 \text{ mA}$
$V_{OLI}$	Output Low Voltage (Interface Pins)		0.5	V	$I_{OL} = 1.6 \text{ mA}$
$V_{OH}$	Output High Voltage	2.4		V	$I_{OH} = - 220 \mu A$
$I_{IL}$	Input Load Current		$\pm 10$	$\mu A$	$V_{IN} = V_{CC}$ to 0V
$I_{OZ}$	Off-State Output Current		$\pm 10$	$\mu A$	$V_{OUT} = V_{CC}$ to 0V
$I_{CC}$	$V_{CC}$ Supply Current		180	mA	

**CAPACITANCE** ( $T_A = 25^\circ C$ ;  $V_{CC} = GND = 0V$ )

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
$C_{IN}$	Input Capacitance			10	pF	$t_c = 1 \text{ MHz}$
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured Pins Returned to GND

**NOTE:** 1. Ambient temperature under bias for 8271-6 is  $0^\circ C$  to  $50^\circ C$ .

**A.C. CHARACTERISTICS** ( $V_{CC} = +5.0V \pm 5\%$   
 (8271 and 8271-8:  $T_A = 0^\circ C$  to  $70^\circ C$ ; 8271-6:  $T_A = 0^\circ C$  to  $50^\circ C$ )

**READ CYCLE**

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
$t_{AC}$	Select Setup to $\overline{RD}$	0		ns	Note 2
$t_{CA}$	Select Hold from $\overline{RD}$	0		ns	Note 2
$t_{RR}$	$\overline{RD}$ Pulse Width	250		ns	
$t_{AD}$	Data Delay from Address		250	ns	Note 2
$t_{RD}$	Data Delay from $\overline{RD}$		150	ns	$C_L = 150 \text{ pF}$ , Note 2
$t_{DF}$	Output Float Delay	20	100	ns	$C_L = 20 \text{ pF}$ for Minimum; 150 pF for Maximum
$t_{DC}$	DACK Setup to $\overline{RD}$	25		ns	
$t_{CD}$	DACK Hold from $\overline{RD}$	25		ns	
$t_{KD}$	Data Delay from DACK		250	ns	



**A.C. CHARACTERISTICS (Continued)**

**WRITE CYCLE**

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
$t_{AC}$	Select Setup to $\overline{WR}$	0		ns	
$t_{CA}$	Select Hold from $\overline{WR}$	0		ns	
$t_{WW}$	$\overline{WR}$ Pulse Width	250		ns	
$t_{DW}$	Data Setup to $\overline{WR}$	150		ns	
$t_{WD}$	Data Hold from $\overline{WR}$	0		ns	
$t_{DC}$	DACK Setup to $\overline{WR}$	25		ns	
$t_{CD}$	DACK Hold from $\overline{WR}$	25		ns	

**DMA**

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
$t_{CO}$	Request Hold from $\overline{WR}$ or $\overline{RD}$ (for Non-Burst Mode)		150	ns	

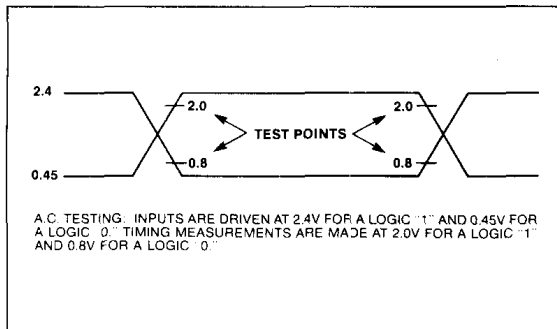
**OTHER TIMINGS**

Symbol	Parameter	8271/8271-6		Unit	Test Conditions
		Min.	Max.		
$t_{RSTW}$	Reset Pulse Width	10		$t_{CY}$	
$t_r$	Input Signal Rise Time		20	ns	
$t_f$	Input Signal Fall Time		20	ns	
$t_{RSTS}$	Reset to First IOWR	2		$t_{CY}$	
$t_{CY}$	Clock Period	250			Note 3
$t_{CL}$	Clock Low Period	110		ns	
$t_{CH}$	Clock High Period	125		ns	
$t_{DS}$	Data Window Setup to Unseparated Clock and Data	50		ns	
$t_{DH}$	Data Window Hold from Unseparated Clock and Data	0		ns	

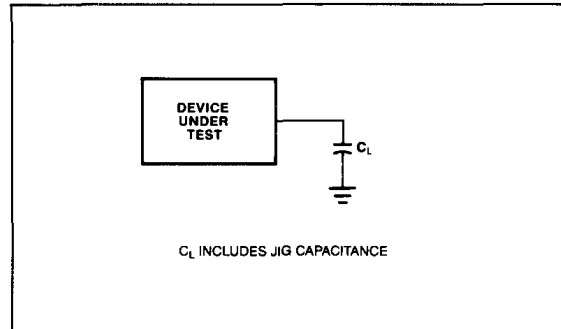
**NOTES:**

- All timing measurements are made at the reference voltages unless otherwise specified: Input "1" at 2.0V, "0" at 0.8V  
Output "1" at 2.0V, "0" at 0.8V
- $t_{AD}$ ,  $t_{RD}$ ,  $t_{AC}$ , and  $t_{CA}$  are not concurrent specs.
- Standard Floppy:  $t_{CY} = 250 \text{ ns} \pm 0.4\%$  Mini-Floppy:  $t_{CY} = 500 \text{ ns} \pm 0.4\%$

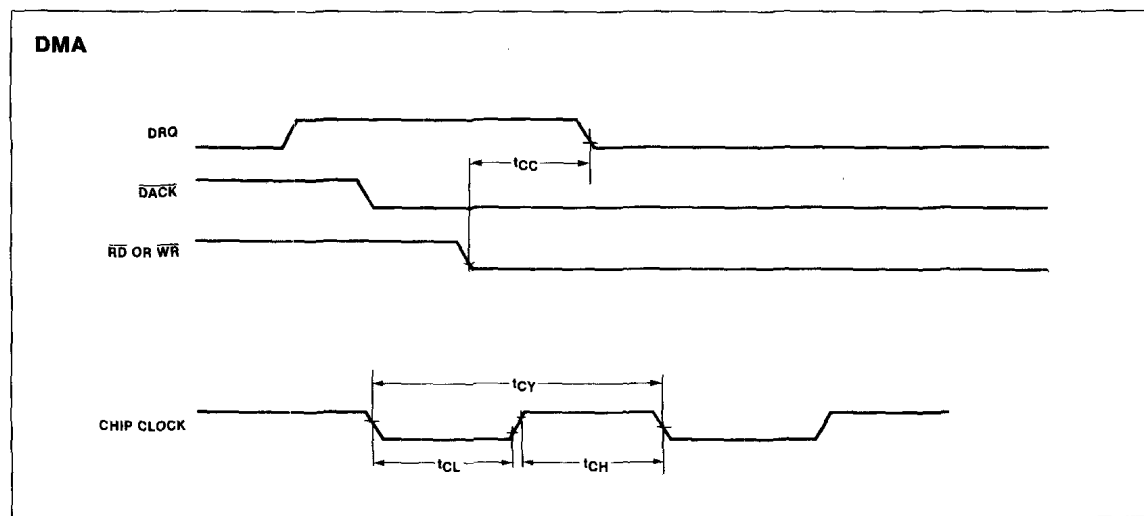
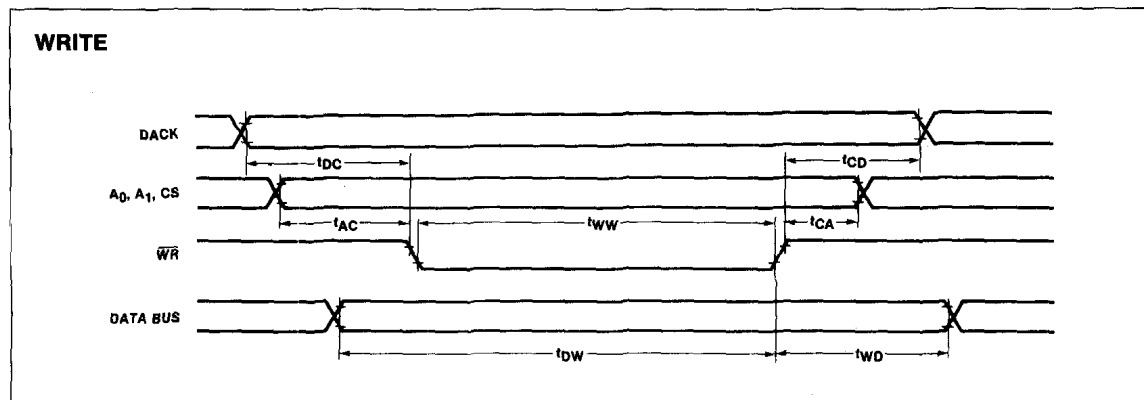
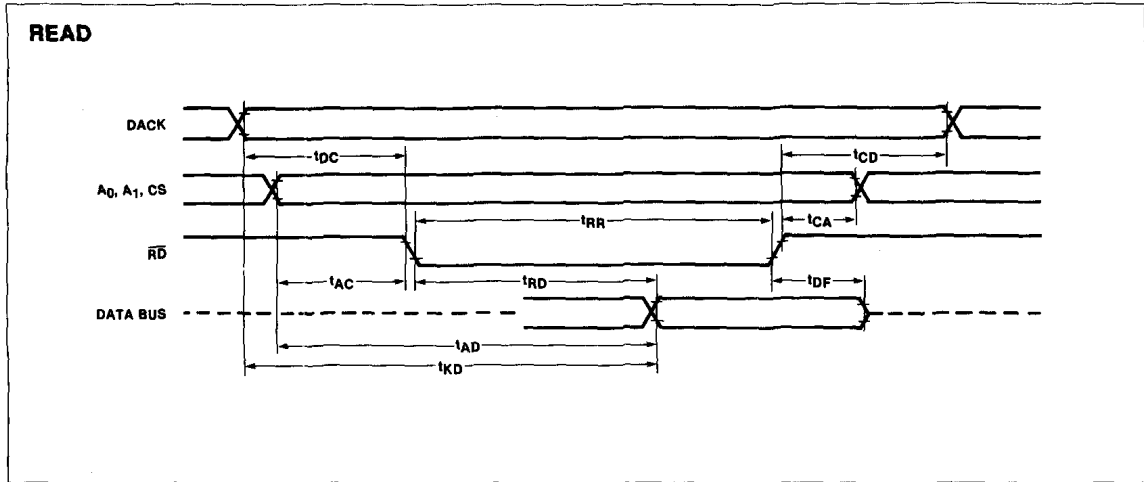
**A.C. TESTING INPUT, OUTPUT WAVEFORM**



**A.C. TESTING LOAD CIRCUIT**



WAVEFORMS



WAVEFORMS (Continued)

