Cat. No. 25-1517

Tandy 1000 TL

Technical Reference Manual

TANDY®

Tandy 1000 TL Technical Reference
Manual

Reproduction or use of any portion of this manual, without express written permission from Tandy Corporation and/or its licensor, is prohibited. While reasonable efforts have been made in the preparation of this manual to assure its accuracy, Tandy Corporation assumes no liability resulting from any errors in or omissions from this manual, or from the use of the information obtained herein.

The Technical Reference Manual for the Tandy 1000 TL describes the computer hardware components and their relationships to one another, as well as the BIOS (Basic Input Output Services).

The information in this manual is intended for hardware and software designers, engineers, programmers, and anyone who requires an understanding of the design and operation of the computer.

Timing diagrams for devices used in the system architecture, Schematics, specifications, switch settings and jumpers, and a theory of operation are provided for the following hardware sections:

Main Logic Board
Devices
Power Supplies
Keyboards
Disk Drives

The Software section contains the following:

A Quick Reference list of software interrupts
(for all device, I/O, and system status services)
Keyboard ASCII and scan codes
An MS-DOS memory map

The information in this manual is a supplement to and based on a working knowledge of the following literature:

The 1000 TL Installation and Operation Guide (Packaged with the computer)

The Intel 80286 Programing Reference Manual. Intel order number 210498-005

The Intel 80286 Hardware Reference Manual. Intel order number 210760-002

80286 Data Sheet. Intel order number 210253-012

This Intel literature may be ordered directly from Intel at the following number: 1-800-549-4725

Tandy 1000 TL
Page Insertion Guide

Important Customer Note:

A gray stripe has been printed along the right edge of the title page of each of the sections to facilitate your finding the beginning of the section.

Also, a tabbed divider for each section has been provided for insertion at this point.

- Exploded view: Insert at the end of the Assembly/Disassembly section

- Foldout schematic pages: Insert at the end of the Main Logic Board section

Schematics	C8000300 - Rev A Sheets 1 of 10 thru 10 of 10

- Foldout PCB art: Insert after the Main Logic Board schematics

Silkscreen	1700376 - Rev A
Layer 1 Component Side	
Layer 2 GND Plane	
Layer 3 + 5V Plane	
Layer 4 Solder Side	

- Foldout schematic page: Insert at the end of the 67 Watt Single Input Power Supply section

Schematic	Model No. 8790085

- Foldout schematic page: Insert at the end of the 67 Watt Dual Input Power Supply section

Schematic	Model No. 8790084

- Foldout keyboard art pages: Insert after the Fujitsu Keyboard information in the Keyboard section

Keyboard Unit Assembly	N860-4703-U001
Block Diagram	4700
Circuit Specification	N86C-4700-0001
Circuit Specification	N86C-4700-0101

- Foldout schematic page: Insert after the Fujitsu Custom IC Pin Signal sheet 2 of 3 in the Keyboard section

Schematic	Fujitsu Custom IC Pin Signals & Function Sheet 3 of 3

- Exploded view of Parts Assembly location: Insert after the Section 4 Part Replacement portion of the Disk Drive section

- Foldout schematic page: Insert after the Overall Diagram in the Disk Drive section

Table Of Contents

Introduction

Introduction

General Description

The Tandy® 1000 TL is modular in design to allow maximum flexibility in system configuration. The computer consists of a main unit, and a detachable keyboard with coiled cable. The main unit is supplied with one internal 3½-inch 720K floppy disk drive. The standard types of monitors used with the Tandy 1000 TL are the monochrome and the color RGB monitor. Since these units are modular, you can place them on top of the main unit or at any convenient location.

The Tandy 1000 TL comes standard with 640K of system RAM. An optional 128K RAM can be added on the system board to expand the memory to a full 768K bytes, the maximum RAM allowed by the system memory map.

Other features include a parallel printer port, a serial port, two built-in joystick interfaces, a real-time clock, a speaker for audio feedback, a headphone jack with volume control, and microphone/line audio in jack for sound input.

Specifications Summary

- 80286 CPU running at 8 MHz, 1 wait state, switchable to 4 MHz
- Socket for 80287 numerical coprocessor
- 640K bytes DRAM upgradeable to 768K bytes (16-bit data bus)
- 4 Mbit BIOS ROM with MS-DOS® and Deskmate® (16-bit data bus)
- Tandy 1000 TL video controller that supports:
    - 128K bytes DRAM (used as system and video memory)
    - alphanumeric mode
    - graphics modes including:
        - 160 X 200 16-color
        - 320 X 200 4-color
        - 320 X 200 16-color
        - 640 X 200 2-color
        - 640 X 200 4-color
- 8237-5 DMA controller that supports:
    - 3 DMA channels
    - 8-bit transfers
    - 4 MHz clock speed
- 8259A interrupt controller for 8 interrupts
- 8254 interval timer that supports:
    - system interrupt timing
    - sound timing
    - refresh timing
- Custom keyboard interface controller
- 101-key Enhanced keyboard
- Custom parallel printer port
- Serial port (RS-232-C)
- Real-time clock w/battery
- Audio Out interface circuit that supports:
    - internal 8-Ohm speaker
    - headphone jack with user accessible volume control
- Audio In interface circuit that supports:
    - microphone in
    - line audio in
    - sound digitizing and recording
- Joystick interface for two joysticks
- Custom floppy disk controller circuit that supports:
    - 5¼-inch 360K floppy disk drives
    - 3½-inch 720K floppy disk drives
- On-board Intelligent Hard Disk interface
- One 3½-inch 720K floppy disk drive
- Five 8-bit expansion slots
- Reset button and support logic
- 67-Watt power supply

Optional Features

- 80287 numerical math coprocessor
- 128K DRAM upgrade (16-bit data bus memory)
- 5¼-inch 360K floppy disk drive
- 3½-inch 720K floppy disk drive
- Hard disk card (20/40 meg)
- Display adapter boards that support mono, EGA, or other special video modes
- 300, 1200, or 2400 baud modem boards

Physical Specifications

(Computer and Keyboard)

Processor:	Intel® 80286
Dimensions:	Computer - 16½-inch x 13¼-inch x 5½-inch
	Keyboard - 16½-inch x 8-inch x 1¼-inch
Weight:	Computer - 18 lbs. (with 2 floppy disk drives)
	Keyboard - 3 lbs. 4 oz.

Power Requirements:

Range:	105 VAC to 135 VAC
Nominal:	120 VAC, 60 Hz, 3 Amp maximum

With 1 Floppy Disk Drive, 640K Memory:

AC Current: 350 - 400 mA with floppy doing R/W tests.

Disk Drive:

	+5 VDC	+12 VDC
R/W	560 mA (Min.)	340 mA (Max.)
Main Logic Board:	1700 mA	450 mA

Operating Environment:

Temperature:	55° to 85° F (13° to 30° C)
Humidity:	40% to 80% non-condensing

Non-Operating Environment:

Temperature:	-40° to +160° F (-40° to 71° C)
Humidity:	20% to 90% non-condensing

Disk Drive Specifications

Power:

Supply Voltage	+5 VDC Input	+12 VDC Input
Ripple		
0 to 50 kHz	100 mV	100 mV
Tolerance		
Including Ripple	+/-5%	+/-5%
Standby Current		
Nominal	190 mA	160 mA
Worst Case	220 mA	190 mA

Operating Current

Nominal	260 mA	600 mA
Worst Case	300 mA	1000 mA

Assembly/Disassembly

System Assembly/Disassembly (Including Exploded Views)

The following instructions explain how the major subassemblies are removed from the Tandy 1000 TL. Re-assembly of major subassemblies is accomplished by reversing the order of the removal procedures.

1. Top Cover Removal

a. Remove the (2) screws from the side of the computer at the rear.

b. Slide the cover forward enough to clear the power button, volume knob, and disk drive eject button and then lift the cover straight up and off.

2. 3½-inch Floppy Disk Removal

a. Remove the top cover.

b. Unplug the cable from the disk drive.

c. Remove the (3) screws attaching the drive to the drive mounting tower.

d. Slide the drive out of the drive mounting tower.

3. Power Supply Removal

a. Remove the top cover.

b. Remove the rear panel by slightly bending the hooks on each side near the bottom and rotating enough to clear the sheet metal and then lift up.

c. Remove all cables from the main logic board and disk drives.

d. Remove the arm attached to the power supply switch.

e. Remove the (3) screws from the rear of the computer and (1) screw from the side that secure the power supply to the rear of the machine.

f. Slide the power supply up and out.

4. Main Logic Board Removal

a. Remove the top cover.

b. Unplug all cables and remove all of the adapter boards from the system.

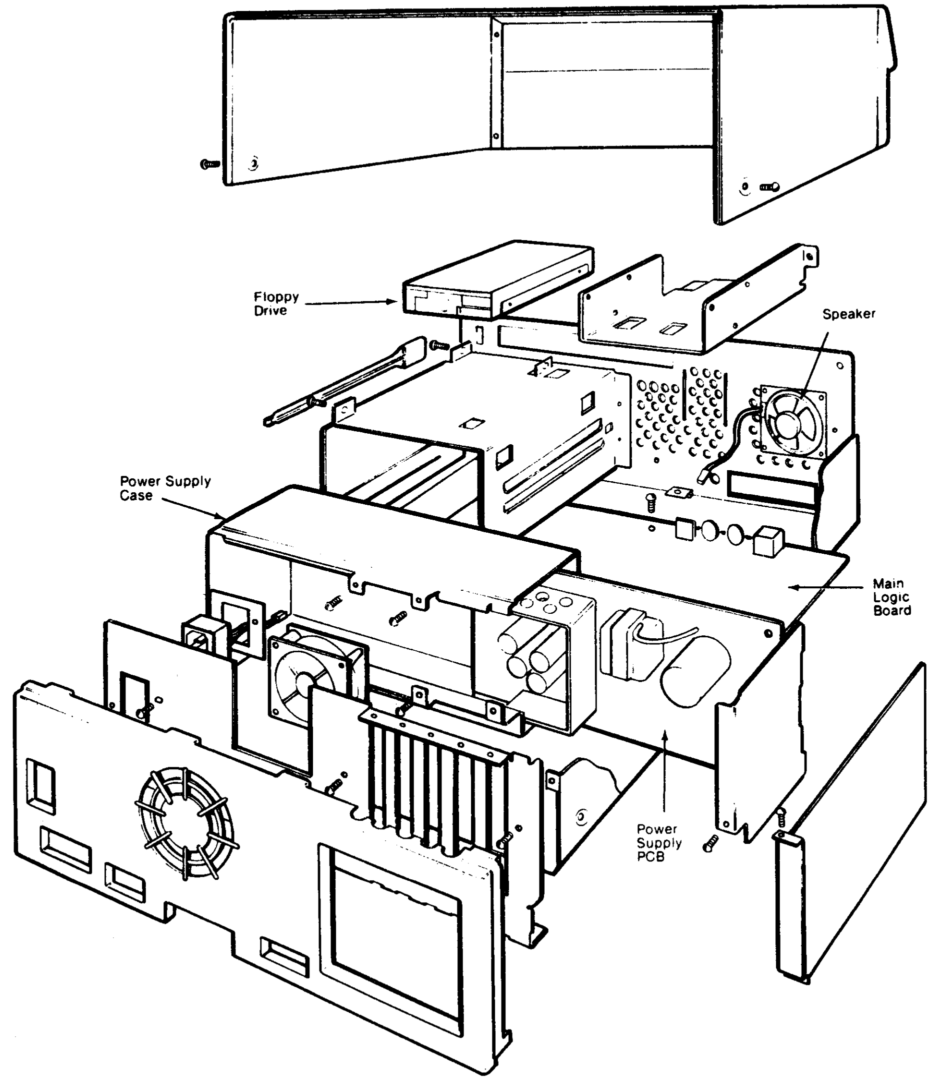
c. Remove the power supply.

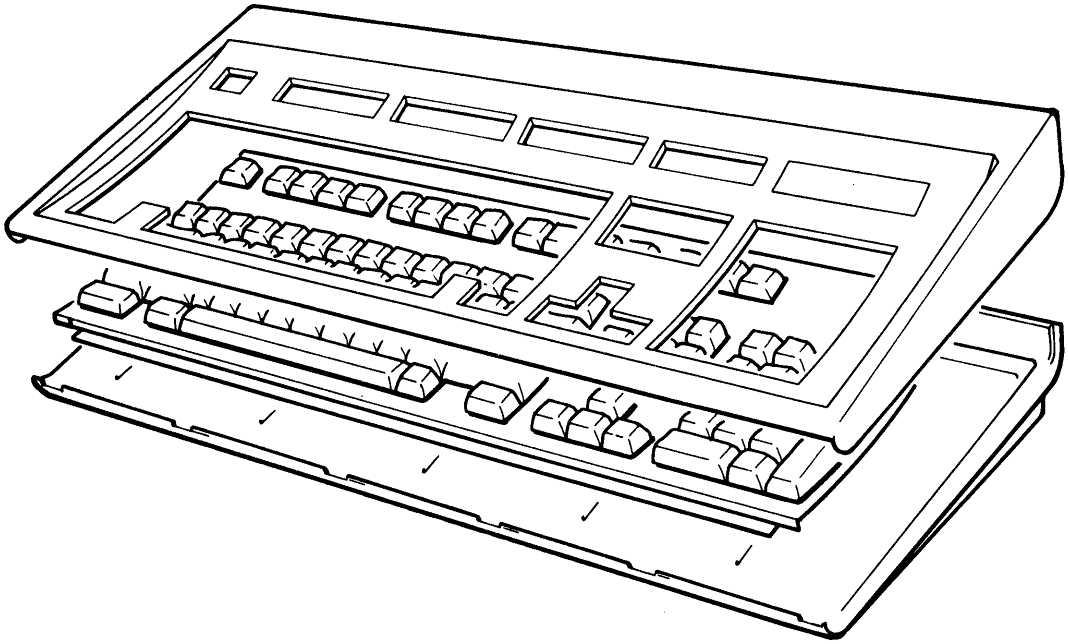
d. Remove the back of the chassis by removing (1) screw at the rear of the computer and pulling the back of the chassis to the rear and down to clear the (3) hooks in the bottom of the chassis. (also must remove 4 nuts on back)

e. Remove the (11) screws holding the main logic board in place. (also must remove drive B: bracket) (also must remove speaker)

f. Remove the main logic board by carefully pulling straight back from under the drive support and out of the chassis. (also must remove volume control clip & knob)

NOTE: WHEN REPLACING THE MAIN LOGIC BOARD, BE SURE THAT THE VOLUME CONTROL KNOB POST SLIDES INTO THE VOLUME CONTROL POT CORRECTLY.


Floppy Drive
Speaker
Power Supply Case
Main Logic Board
Power Supply PCB



Power supply cable

back

red → purple

front

Speaker connector

front

black – grey

back

MECHANICAL BILL OF MATERIAL - TANDY 1000 TL

TANDY 1000 TL FINAL ASSEMBLY

QTY.	DESCRIPTION	PART NUMBER
1	CHASSIS - WELDMENT	8729710
5	PANEL - OPTION SLOT Rev D	8729562
5	SCREW - #4-40 X 3/16 OPTION SLOT PANEL	8569333
4	FOOT	8590179
4	RIVET - #1661-0512	8565014
1	PC BOARD SUBASSY - TANDY 1000 TL MAIN LOGIC A	8859022
1	BATTERY - LITHIUM CR2032	8491012
1	SHIELD - GROUND, EARPHONE & MICROPHONE	8729720
10	SCREW - #6-32 X 1/4 MAIN BOARD	8569326
4	JACKNUT - #4-40 X 3/16 9 PIN CONNECTORS	8569341
4	SHIELDING STRIP FLITE-WAY ENGR	8729658
1	KNOB - VOLUME CONTROL	8719624
1	CLIP - HAIRPIN	8559080
1	CHASSIS - POWER SUPPLY Rev B	8729690
1	POWER SUPPLY - 67 WATT INT'L	8790084
	DOM.	8790085
	DOM.	8790091
4	SCREW - #6-32 X 5/16 POWER SUPPLY	8569339
1	DC HARNESS	8709857
1	SWITCH - POWER	8489111
1	SWITCH - POWER INT'L	8489112
2	SCREW - M3 x 5PPH	8569293
3	SCREW - #6-32 X 5/16	8569339
1	ACTUATOR - POWER SWITCH	8719620
1	BUTTON - POWER	8719625
1	RECEPTACLE - AC	8519246
1	HARNESS - AC DOMESTIC	8709868
1	HARNESS - AC Rev B INT'L	8709873
2	CAPACITOR - 1000 PFD, 400V	8352106
1	TORROID - CORE FAIRRITE	8419030
2	NUT - KEPS, #6-32	8579004
1	FAN - 80 MM; 12 VDC	8790424
4	SCREW - #10 TAPIT THREAD	8569301
1	ENDPLATE - POWER SUPPLY	8729691
1	CHASSIS - REAR	8729693
2	SCREW - #6-32 X 5/16	8569339
1	BRACKET - 3 1/2 DISK DRIVE	8729687
1	BRACKET - HARD CARD	8729704
2	SCREW - #6-32 X 5/16	8569339

MECHANICAL BILL OF MATERIAL - TANDY 1000 TL

TANDY 1000 TL FINAL ASSEMBLY

QTY.	DESCRIPTION	PART NUMBER
1	DISK DRIVE - 3 1/2" 720KB SONY MP-F11W-70D	8790144
1	BUTTON - DISK EJECT	8719596
3	SCREW - M3 X 5 PPH	8569293
1	CABLE - SIGNAL	8709854
2	RAIL - 5 1/4" DRIVE	8719603
2	CLIP - GROUNDING, DRIVE	8529064
2	SCREW - #6 - 32 X 1/4 PHILLIPS PAN HD	8569098
1	SUPPORT - 3 1/2" DRIVE BRACKET	8729722
1	SPEAKER W/CABLE	8490013
4	SCREW - #6-32 X 5/16	8569339
1	PANEL - REAR	8719602
1	BEZEL - FRONT	8719604
2	LENS	8719560
2	PIN - GUIDE	8739038
1	CASE - TOP	8729686
1	PANEL - DRIVE COVER	8719621
5	SCREW - #6 X 3/8	8569294
2	SCREW - #10-24 UNC 3/8" PHILLIPS OVAL HEAD MACHINE SCREW	8569354
1	BUTTON - RESET, FRONT	8719440
1	BUTTON - RESET, REAR	8719441
1	SPRING - RESET BUTTON	8739018
1	CORD - POWER 18/3 60/C	8709057
1	NAMEPLATE	8719613
1	LABEL - SERIAL UL/FCC	87891642
1	LABEL - SERIAL CSA	87891640
1	LABEL - SERVICE ADVISEMENT (6 LANG)	87891571
1	LABEL - CAUTION (6 LANG)	87891572
1	LABEL - SERIAL INT'L	87891641
1	LABEL - BATTERY WARNING	87891570
1	LABEL - EARTH GROUND INT'L -	87891253
1	LABEL - VIDEO, MONOCHROME COMMAND	87891648
1	KEYBOARD ASSEMBLY	

Main Logic Board

Main Logic Board

Introduction

The main unit is the heart of the Tandy 1000 TL. It houses the main logic assembly, system power supply, and floppy disk drive.

The main logic assembly is a large board mounted to the bottom of the main unit and interconnected to the keyboard, power supply, and disk drive by a series of cables.

The power supply is a 67W switching regulator type, designed to provide adequate power capacity for a fully configured system that has all the option slots in use.

The floppy disk drive uses 3½-inch double-sided, double-density diskettes to read, write, or store data. These are soft sector diskettes. The disk drive assembly comes installed in the main unit. The floppy diskette stores approximately 720K bytes (formatted) of data. All system programs, with the exception of the system startup sequence, are stored on diskette.

Switch Settings and Jumper Pin Configurations

Main Logic Board

Jumper	Function	Default
E1-E2	Select Video Interrupt on IRQ5	E2-E3
E2-E3	Normal Video Interrupt	

Jumper	Function	Default
E6-E7	Select Direct Line Audio Input	E7-E8
E7-E8	Select Mic Audio Input	

This is inaccurate. See circuit diagram, Sheet 9 of 10, which agrees with the motherboard labelling. Layer Top Silkscreen agrees with the above and is also in error. Owner's manual, page 15 is correct. The correct settings are:

Jumper	Function	Default
E6-E7	Select MIC Audio Input	E6-E7
E7-E8	Select Direct Line Audio Input	

Connecting unattenuated direct-line audio with jumper at E6-E7 will fry the sound circuitry.

Jumper set to E7-E8, 4/1/94.

Theory of Operation

80286 Microprocessor

The 80286 (U31) is an advanced, high-performance, 16-bit microprocessor with special capabilities for multi-tasking and multi-user systems. Two modes of operation are available in the 80286, the Real Address mode, and the Protected Virtual Address mode. In the Real Address mode, the 80286 is compatible with existing 8086 and 8088 software and allows addressing of one megabyte of memory space. The Tandy 1000 TL does **NOT** support the Protected Virtual Address mode.

80287 Numerical Math Coprocessor

The 80287 (U60) performs high-speed arithmetic and logarithmic functions and trigonometric operations that increase the performance of an 80286 system. Performance increases are obtained by the 80287's ability to perform math calculations faster than the 80286, and also by executing math instructions in parallel with the 80286.

Clock Generation (Night Blue)

All clocks required by the system are generated by the custom CPU Controller (U17). There are two independent clock circuits supplied by a Dual Oscillator Clock (Y2) from which all other clocks are derived.

The 16 MHz Clock is routed into the CPU Controller, which generates the output signals PRCLK, DMACLK, and SCLK. The Clock Switch circuitry required to toggle the 80286 Microprocessor between 8 MHz and 4 MHz mode, as well as the logic to prevent any short cycling during a clock switch cycle, are implemented in the CPU Controller IC. If the signal XD3 is asserted high during an I/O write to port 062 (hex), then the output signal PRCLK is 16 MHz, which operates the 80286 in 8 MHz mode. If the signal XD3 is asserted low during an I/O write to port 062 (hex), the output signal PRCLK is at 8 MHz, operating the 80286 in the 4 MHz mode. When Reset is generated, the signal RES- is asserted low and defaults the Tandy 1000 TL to the 8 MHz mode.

The CPU Controller Chip also controls wait states to insert the proper number of wait states required for a two clock mode of operation. When the PRCLK signal is 16 MHz (8 MHz Mode), then four wait states are inserted in all 8-bit Memory and I/O cycles. When the signal PRCLK is 8 MHz (4 MHz mode) then two wait states are inserted during all 8-bit Memory and I/O cycles. During all 16-bit memory cycles, only one wait state is inserted in both the 8 MHz and 4 MHz modes.

PRCLK0 is then routed through a damping resistor to produce the signal PRCLK for the 80286, PRCLKA for the 80287 math coprocessor, and PRCLKB for the DRAM/DMA control logic.

DMACLK and SCLK are output signals for system use. The DMACLK output frequency is ¼ of the PRCLK signal, and the SCLK output frequency is ½ of the PRCLK signal. Both are synchronized by Reset to the PRCLK output signal. After a Reset, DMACLK and SCLK are held low until the 80286 asserts status S1 = 0. SCLK and DMACLK make the first transition on the falling edge of PRCLK, following with a Ts state that synchronizes them to PRCLK.

SCLK is buffered and filtered, then routed to the Expansion Bus for option board use. DMACLK is filtered and then routed to the DMA Controller.

Table 2 shows all the clocks generated from 16 MHz in both modes:

	8 MHz Mode	4 MHz Mode
PRCLK	16 MHz	8 MHz
SCLK	8 MHz	4 MHz
DMACLK	4 MHz	2 MHz

Table 2. Clocks Generated From 16MHz.

Command and Control Signal Generation

The command and control signals required for the Tandy 1000 TL operation are generated by the CPU Controller (U17). The command signals are decoded from the CPU status signals S0- through S1- and M/(IO-) during the Ts cycle. The decoded signals indicate the type of cycle that is to be executed (MEMR, MEMW, IOR-, IOW-, INTA-). The control signals (ALE, DT/R, DSDEN0-, DSDEN1-, MEMCYC) control the latching of addresses, determine the direction and enabling of the data bus buffers, and start a memory cycle. Table 3 indicates the decoding of the CPU status signals.

M/(IO-)	S1-	S0-	Type of Bus Cycle
0	0	0	Interrupt Acknowledge
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	None: Idle
1	0	0	Halt or Shutdown
1	0	1	Memory Read
1	1	0	Memory Write
1	1	1	None: Idle

Table 3. CPU Status Signal Decoding.

A0 and BHE- are decoded to determine the data transfer width to and from the CPU. Table 4 shows the data transfer width depending on the state of A0 and BHE-.

BHE-	A0	Width of Data Transfer
0	0	Word Transfer
0	1	Byte Transfer D8 - D15
1	0	Byte Transfer D0 - D7
1	1	Not Used

Table 4. Data Transfer Width Decode.

Command Buffer

Some of the command signals generated by the CPU Controller require buffering to the system. This is accomplished by ½ of an ALS244 (U25) and ½ of another ALS 244 (U9). These ICs buffer the command signals to the system bus for the expansion bus slots, the peripheral devices, and also the Video Controller.

DRAM Control

The CPU address decode for the Dynamic Random Access Memory (DRAM) array is generated by the Custom DRAM/DMA Control IC (U22). These signals are latched by ALE internally to the DRAM/DMA Control IC and held for the complete cycle. The address decode signals are RAS0-, RAS1-, CASL-, and CASH-. Memory configurations supported by the Tandy 1000 TL are 640K bytes or 768K bytes (which includes 128K of video memory). Table 5 shows the different options available on the DRAM/DMA Control IC.

Memory Option	MC1	MC0	System Memory	Total System Memory*	Control	Bank	Address Range
1	0	1	512K	640K	RAS0	512K	000000-07FFFF
2	1	1	640K	768K	RAS0 RAS1	512K 128K	000000-07FFFF 080000-09FFFF

* Note: Total system memory includes 128K of video memory.

Table 5. Memory Configurations.

Port FFEA hex, Bits 6 & 7, control the Memory Configuration Options. See the I/O Map later in this manual for details. MEMCYC triggers the control signals for the DRAM array. The MEMCYC signal (generated by the CPU Controller) indicates that a DRAM bus cycle is in progress. MEMCYC enables RAS0-, RAS1-, CASH-, and CASL-, depending on the address of the bus cycle. The selected RAS(x)- lines become active at the next falling edge of PRCLK.

After ½ PRCLK cycle at the rising edge of the clock, MUX is generated and switches the DRAM address (MA0-MA8) from Row Address to Column Address. After another PRCLK cycle at the next rising edge of the clock, CASL- and/or CASH- are asserted. Two CAS signals are generated internally to the DRAM/DMA Control IC to provide the ability to access word or byte cycles in the DRAM array. Table 6 shows the state of each control signal during each type of bus cycle.

Address Range	Bus Width	RAS0-	RAS1-	CASL-	CASH-
000000-07FFFFH Even	Byte	0	1	0	1
000000-07FFFFH Odd	Byte	0	1	1	0
000000-07FFFFH Even	Word	0	1	0	0
080000-09FFFFH Even	Byte	1	0	0	1
080000-09FFFFH Odd	Byte	1	0	1	0
080000-09FFFFH Even	Word	1	0	0	0

Table 6. Signal State Control Signals.

The signals WE0- and WE1- provide read and write control. Both are asserted at the same time and are controlled by MEMW- (memory write). If WE0- and WE1- are asserted high, it is a read cycle; if they are asserted low, it is a write cycle.

Refresh Control

The REFREQ pin of the KFIT custom IC (U13) generates an active high pulse every 15 usec. The rising edge of the REFREQ signal clocks the 8237 DMA controller. This input to the DMA controller is actually Data Transfer Request 0, (DREQ0), which requests the DMA to perform a DMA cycle. The DMA controller channel 0 has been programmed to perform a single transfer from memory to an I/O device, such as a floppy drive. This causes a memory read at a certain address to be performed. Each time the REFREQ signal is generated, the DMA controller increments the address and performs another memory read. This causes all memory rows to be read every 4ms to keep data in the DRAMs stable. Refer to the section on the DMA controller for more information on DMA cycles.

BIOS ROM Control

The DRAM/DMA IC (U22) provides the CPU address decode used for the ROM select. The signal generated is called ROMCS- (ROM Chip Select) and is used as part of the decode used by PLS173 IFL U44. The PLS173 IFL then generates the ROM Page Selects (RPCS-) and Chip Enable for the BIOS ROMs U54, U55, U56, and U57. This output is asserted whenever any of three addressed ranges is detected, CPHLDA is inactive, and ALE is asserted. The three address ranges are 0E0000-0FFFFFH, EE0000-EFFFFFH, and FE0000-FFFFFFH. The address lines SA1-SA15 are provided to the BIOS ROMs for lower address control. The data is buffered onto the MD0-MD15 data bus, controlled by the 82C205 IC.

Reset Circuit

The CPU Control IC (U17) controls the system reset required either to initialize the complete system after power-up or to reboot. Two reset output signals, RESET and RESCPU, are active high and generated when a power-up condition is detected or when the reset button on the front of the computer is pressed.

The RESET signal is used as a general system reset, while the RESCPU signal resets the 80286 Processor. The RES- signal is the input to (U17), which signifies a reset condition. The RES- signal is generated either from an RC network during power up or from the reset switch. During a power-up, RES- is held low for the time period generated by the RC time constant of R41 and C62. This is the time it takes C62 to change to an active high. Also, if the reset switch is pressed, it applies a ground to C62 and discharges it. This asserts RES- low until C62 charges again to a logic high.

During power-up, the RES- signal is generated twice to provide a proper reset to the CPU control IC. A second RC time constant is generated by R27 and C83 to the input of a schmidt trigger inverter (U14). This holds the input pin of U26 (½ of a 74LS74) high for approximately 150ms. When RESCPU is negated after the first reset, the CPU issues the first command to the CPU control IC by driving SI- low. This generates the first rising edge of DMACLK, which latches into U26. The Q output of U26 is then routed to an open collector inverter, (U6), which discharges C62, again asserting RES- low and generating the second reset. CR1 provides a reset to U26 when C62 is discharged to at least .7v. After U26 is reset, RES- is released, and C62 is allowed to charge, negating RES- and finishing the second reset pulse. When the CPU issues the first command again, which starts DMACLK, U26 latches a low. This is because the D input of U26 has transitioned to a low by the end of the second reset.

The CPU Control IC (U17) also internally controls the RESCPU signal to meet the requirements of the 80286 during a detected shutdown condition.

Wait State and Ready Logic

Wait state control is implemented internally to the CPU Control IC. The function of the wait state control logic is to match the speed of the various devices in the Tandy 1000 TL to the speed of the 80286 CPU. Two circuits assert wait states within memory and I/O cycles. One method is controlled by the device being accessed, using the IOCHRDY signal input to the CPU Control IC. If a device requires additional wait states within the bus cycle, the device should negate IOCHRDY low until it can service the bus cycle. After the required number of wait states have been inserted, the device should assert IOCHRDY, causing the READY- output of the CPU Control IC to be asserted low, which tells the CPU to terminate the cycle.

The second method (internal to the CPU Control IC) is several default wait states during the various bus cycles. During a 16-bit memory cycle (which is determined by the assertion of AF16-), one wait state is automatically inserted. The default of an 8-bit memory or I/O cycle is four wait states. This can be overridden by driving IOCHRDY low as mentioned above. As long as IOCHRDY is at a logic low level, wait states are inserted indefinitely.

> Note: IOCHRDY should not be held low for longer than 15 usec because it will stop DRAM refresh cycles.

NMI- Logic

In the Tandy 1000 TL, the Non-Maskable Interrupt (NMI- indicates an I/O error condition and Numerical Math Coprocessor 80287 condition signal (NMI-). Both error conditions are being enabled by the NMI- signal, which is generated from the System data bus bit SD7, PAL CU1, and I/O write (IOW-. The INT287- signal (from the CPU Control IC) becomes active when the ERROR- signal is asserted by the Numerical Math Coprocessor.

80287 Control Logic

Incorporated into the CPU Control IC is the logic required to interface the 80287 Math Coprocessor to the 80286 CPU. This logic decodes the signals that select and reset the 80287 and also handles the Busy/(Error-) signals from the 80287 to the CPU.

The input signal 287CS- is a user I/O address decoded signal used by the CPU Control IC to generate the control signals to the 80287 IC. The 287CS- signal is asserted during I/O address 0F0h - 0FFh. Further decoding is provided by the CPU Control IC, which generates RES287, NPCS-, and BUSY287- signals. Table 7 defines the internal decode.

Hex Address	Description
0F0	Clear Math Coprocessor Busy
0F1	Reset Math Coprocessor Busy
0F8-0FF	Math Coprocessor Chip Select

Table 7. 287CS- Decode.

Given the command to perform a task, the 80287 coprocessor issues a BUSY- signal to the CPU Control IC. With the assertion of the BUSY287- output, this signal is passed to the CPU. Normally the BUSY- input is passed through to the BUSY287- output. Deassertion of BUSY- results in deassertion of BUSY287-. The BUSY287- output is latched, and the INT287- output pin is forced HIGH during this busy period if the ERROR- input becomes active (signaling a Numerical Processor error). Until cleared by an I/O write cycle to address 0F0h or 0F1h, both signals then remain active. Both the interrupt latch for INT287- and the busy latch for BUSY287- are cleared after a system reset.

The CPU Control IC's RES287 output pin handles resetting of the 80287 coprocessor. This can be activated by a system reset or an I/O write to address 0F1h. This signal is active only for the period of time that the source signal is active, as it is not latched internally.

CPU Address Buffers

(U30), (U43), and (U27) 74ALS573 implement buffering of the address lines to the system. SA01-SA19 are buffered and latched for the expansion bus slots and I/O peripherals. ALE is used to latch SA01-SA19 and held for the complete bus cycle. SA01-SA19 are also used to address the BIOS ROMS and DRAM/DMA Control. A17-A23 are routed directly to DRAM/DMA control to generate memory address decoded signals. The multiplexed address lines MA0-MA7 are also generated and buffered to the DRAM memory by the DRAM/DMA Control IC. To meet address requirements for the DRAMs, the MUX signal multiplexes SA1-SA16 internally to the DRAM/DMA Control IC.

During a DMA cycle, a 74LS245 (U43), is used to buffer S0-S7 directly from the DMA controller. S8-S16 are buffered internally to the DRAM/DMA controller. S17-S19 are buffered by AU7 (½ of a 74ALS244).

Data Buffers and Conversion Logic

The 82A205 IC (U61) provides the data buses, buffers, and drivers for D0-D15 to the system. Three data buses are generated, SD0-SD7 for the expansion bus slots, MD0-MD15 for memory access from ROM and DRAM, and D0-D15, which is routed to the 80286 CPU and 80287 Coprocessor data bus. The direction and control of the data buffers are provided by the input signals to the 82A205 IC (DT/R, DSDEN0-, DSDEN1-, SBHE-, and SA0).

DT/R controls the direction of the data path during a read or write. The DSDEN0- and DSDEN1- signals control the word and byte data transfers, while SBHE- and SA0 determine the buffers to be enabled during a byte access.

Conversion logic is also implemented in the 82A205 IC, controlled by ENHLB- and DIRHLB. This conversion logic allows data to be transferred from the lower to upper or upper to lower data byte to meet the requirements of the CPU or receiving device.

I/O Decode

Two PLS173 IFLs accomplish the I/O Address decoding. These two ICs provide all the necessary chip select signals to the system. The A, B, and C output signals of the PLS173 IFL (U42) are encoded device select lines that are fed directly to the PLS173 IFL (U44), in which I/O address decoding is generated. The second PLS173 (U44) decodes the system address to generate the other I/O address decode select signals. Refer to the Tandy 1000 TL I/O Map for details.

Floppy Disk Controller

The on-board Floppy Disk Controller (FDC) and KFIT custom IC interface the system to the Floppy Disk Drive (FDD). Up to three floppy disk drives can be supported.

The FDC circuit can be organized into the following subsections:

- uPD765A FDC Chip
- System Interface
- Clock Generation
- Precompensation
- Data Separator
- Disk Drive Interface

uPD765A Chip. The uPD765A FDC chip (U23) integrates most of the control logic necessary to:

- interface the Serial bit stream to or from the FDD to the parallel bus of the system
- implement the commands necessary to operate the FDD
- maintain information about the status of the FDD

During a read or write data operation to the FDD, the FDC chip generates a DMA request for a byte transfer to or from memory. The FDC chip continues to generate DMA requests until the preprogrammed amount of data is transferred as signified by generation of a Termination Count (TC) Signal. After the TC is reached, the FDC chip generates an interrupt to the system through INT so that status and result data can be serviced.

Refer to the device data sheet for complete descriptions of the available commands and the command and status registers.

System Interface. Various ICs, along with the KFIT custom IC, latch and buffer data to and from the system. A DOR Write (Digital Output Register) is generated on an I/O write to port 3F2 (hex). This signal latches the data byte that is bit defined as the Drive Select, DS0-, DS1-, and DS2-, Motor On, MTRON-, DMA, (FDCDRQ), Interrupt Request (FDCINT), and a reset signal (FDCRST-) to the FDC controller U23.

Clock Generation. The FDC Support IC (U15) generates all clocks required by the Floppy Disk circuit. These clocks are derived from a 16 MHz input signal. FDCCLK, required by the FDC Controller (U23), is derived by dividing the 16 MHz clock by 4. The resulting 4 MHz clock is also used as a delay counter for the DMA request signal DRQ as well as a reference clock for the write precompensation circuit. The 4 MHz clock also generates a 250 nanosecond pulse at a frequency of 500 KHz. The 500 KHz signal is used as a write clock for the FDC Controller.

Precompensation. The precompensation circuit is implemented internally to the FDC Support IC (U15). The write data bit can be shifted either early or late in the serial bit stream, depending on the requirements of the Floppy Disk Drive. This function is programmable and controlled by the FDC IC signals PS0 and PS1.

Data Separator. The FDC Support IC (U15) also contains the data separator circuit. The data separator recovers the clock and data signals from the serial bit stream of the Floppy Disk Drive. The FDC Support IC supports only MFM or Double-density mode.

Disk Drive Interface. All FDC outputs to the FDD are driven by high current open collector buffers inside the KFIT custom IC. All FDC inputs from the FDD are buffered by 74HCT14 SCHMIDT triggered inverters. The inputs are pulled up on-board by 1K ohm terminating resistors. All outputs should be terminated on the last FDD by 1K ohm resistors.

Interrupt Controller

The Interrupt Controller is contained in the KFIT custom IC (U13) and supplies the maskable interrupt input to the CPU. The KFIT custom IC has eight interrupt inputs controlled through software commands. It can mask (disable) and prioritize (arrange priority) to generate the interrupt input to the CPU. The eight interrupts are assigned as follows:

#0	Timer Channel 0	Software Timer
#1	Keyboard	Keyboard Code Received
#2	Interrupt on the Bus	Optional Bus Interrupt
#3	Interrupt on the Bus	Modem (COM2)
#4	Interrupt on the Bus	RS-232 (COM1)
#5	Interrupt on the Bus	Hard Disk Controller/ Vertical Sync
#6	Floppy Disk Controller	Optional Bus Interrupt
#7	Printer	Optional Bus Interrupt

Interrupts 0 and 1 are connected to system board functions as indicated in the chart. Interrupts 2-7 are connected directly to the Expansion Bus, with the normal assigned functions listed in the chart.

Video Controller

The next major block of the Tandy 1000 TL is the video interface circuitry. This custom part contains all the logic necessary to generate an IBM-compatible color video display. The video interface logic consists of the 100-pin custom video circuit (U19), four 64K X 4 DRAMS (U32, U33, U34, and U35), a 74LS244 buffer (U8), and associated logic for generating composite and RGBI video.

The Tandy 1000 TL video interface circuitry controls 128K of memory. This DRAM is shared by the CPU and the video. Normally, the video requires only 16K or 64K for the video screen, and the remainder of the 128K is available for system memory use.

The Tandy 1000 TL video interface custom circuit is composed of a 6845 equivalent design, dynamic RAM address generation/timing, and video attribute controller logic.

Normal function of the video interface custom circuit is as follows. After the 6845 is programmed with a correct set of operating values, a 6:1 multiplexer generates the address inputs to the dynamic RAMs. This MUX switches between video (6845) address and CPU address as well as between row and column address. Also, the video interface chip provides the RAM timing signals and generates a wait signal, VIDWT-, to the CPU for proper synchronization with the video RAM access cycles.

The outputs from the RAM chips are connected only to the video interface custom circuit, so all CPU read/write operations are buffered by this part. During a normal display cycle, video data from the RAM chips is first latched in the Video Attribute latch and the Video Character latch. The video interface requires a memory organization of 64K X 16 and latches 16 bits of memory during each access to RAM. From the output of the two latches, the data is supplied to the character ROM for the alpha modes or to the shift registers for the graphics modes. A final 2:1 MUX switches between foreground or background in the alpha modes.

From the 2:1 MUX, the RGBI data is combined with the PC color select data and latched in the Pre-Palette latch. This latch synchronizes the RGBI data before it is used to address the Palette. The Palette mask MUX switches between incoming RGBI data and the Palette address register. During a CPU write to the Palette, this address register selects one of the 16 Palette locations. Also, the Palette mask MUX allows any of the input RGBI bits to be set to zero.

The Palette allows the 16 colors to be remapped in any desired organization. Normally, the Palette is set for a 1:1 mapping (red = red, blue = blue, and so on) for PC compatibility. However, instantly changing the on-screen colors is a powerful tool for animation or graphics programs.

After the Palette, the RGBI data is resynchronized in the Post Palette register. The final logic before the RGBI data is buffered off the chip is the Border MUX. This MUX allows the Border to be replaced with any color selected by the border color latch. This latch is normally disabled in PC modes, but it is used in all PC jr modes.

Timer

The final Tandy 1000 TL function, other than I/O, is the timer found in the KFIT custom IC (U13). This part is composed of three independent programmable counters. The clock for all three counters is 1.1925 MHz, which is derived from 14 MHz/12. Counters 0 and 1 are permanently enabled. Counter 2 is controlled by port Hex 0061, Bit 0. Counter 0 is connected to system interrupt 0 and is used for software timing functions. Counter 1 is used for refresh function timing. Counter 2 is connected to the sound circuit and also to port Hex 0062, Bit 5.

Joystick Interface

The joystick interface contained in the PSSJ custom IC (U11) converts positional information from hand-held joysticks (1 or 2) into CPU data. Each joystick provides one or two push-buttons and X,Y position for a total of four bits each. You can use two joysticks.

The joystick handle is connected to two potentiometers mounted perpendicular to each other; one for X position, one for Y position. Through the cable, the main logic board applies +5 VDC to one side and ground to the other of the pots. The pot wiper is the position signal: a voltage between 0 and +5 VDC. This signal is applied to one input of a comparator HU2. The other comparator input is the reference signal (a ramp between 0.0 to +5.0 volts).

When the position signal is equal to or less than the reference signal, the comparator output goes true. This comparator output is the X or Y position data bit. The ramp is reset to 0.0 VDC whenever an I/O Write is made at Port 200/201 Hex. The joystick information is "read" by the CPU at Port 200/201 Hex through U11. See the Joystick Block Diagram.

Keyboard Interface

The next I/O function of the Tandy 1000 TL is the Keyboard interface custom circuit, part of the KFIT custom IC. The heart of this custom part is several read/write registers that are used to control the keyboard interface logic. For the interface to the keyboard connector, a 164- type shift register is used to load the serial data and allow the CPU to read it as 8 parallel bits.

Sound Out Circuit

The sound circuit is one of the five I/O functions of the Tandy 1000 TL. The circuit provides sound output for the internal speaker as well as for an external sound circuit.

The main source of sound in the Tandy 1000 TL is the PSSJ custom IC (U11). It contains the equivalent of a 76496 complex sound generator. This device has three tone generators and one white noise generator. Each tone generator can be programmed for frequency and attenuation. Also, this device has an audio input pin connected to the gated output of timer channel 2. This audio input signal is mixed with the sound generator signal and supplied to the audio output pin.

From the output of the 76496, the sound signal is connected to a dual analog multiplexer. The multiplexer is switched by Port 61, Bit 4, and turns off the audio signal to the speaker, headphone jacks, and external audio output. The output of the multiplexer is routed to audio amplifiers U71 for the internal speaker and headphone jacks. The volume of the internal speaker can be adjusted by a user-accessible volume control (R59). When the headphone jack is used, the internal speaker is disabled.

Sound In Circuit

An additional feature of the Tandy 1000 TL sound circuitry is a Digital to Analog Converter (DAC). The DAC is controlled by read/write Ports C4-C7. The DAC can be used to convert pre-recorded digital sound, voice, or music into analog audio output. A microphone jack and audio input circuitry are provided for recording analog sound, voice or music, and converting it to digital data. A set of jumpers is provided with the sound circuit to allow selection between microphone and direct line input. Jumpers E11 to E12 select microphone input, and jumpers E12 to E13 select direct line input. The audio input signal is fed through an amplifier, U72, and increased by a multiple of 100, before being sent to the AUDIOIN pin of the PSSJ custom IC. When direct line input is selected, the audio input signal is reduced in amplitude by resistors R57A and R56A before being sent to amplifier U72. Bit programming data for these ports is available in the data sheets on the 8079021 Custom IC located in the "Devices" section of this manual.

Real-time Clock

The Real-time Clock in the Tandy 1000 TL system is the Dallas Semiconductor® DS1215. Its input is derived from a 32.768 KHz crystal CY1. When system power is removed, operation of the DS1215 is maintained by a 3v battery.

The DS1215 is capable of operating in both 12-Hour mode, with an AM/PM indicator, or 24-Hour mode. The real-time clock is initialized by reading a consecutive 64-bit stream with SA2 high and the ROMCS- signal asserted. Then another 64-bit read or write may be done to set, read, or update the clock. The 64-bit pattern is programmed into 8 8-bit registers in the DS1215. All 8 registers must be programmed at the same time, and must be sent from Register 0 to Register 7. The DS1215 is automatically disabled after the 64-bit stream is read. Register and bit definition for the DS1215 can be found in the data sheets on the DS1215 in the "Devices" section of this manual.

DMA Controller

The major components of the Direct Memory Access (DMA) circuit consists of an 8237A-5 DMA controller (U11), the DRAM/DMA Control (U53), and a bi-directional address buffer 74ALS245 (U3).

Initialization--A DMA Operation. When a DMA operation is requested by software or by a peripheral through a DREQ line, the 8237A-5 DMA controller initiates a Bus Hold Request to the 80286 CPU through the CPU Controller IC. The CPU Controller arbitrates the CPU Hold Request from the DMA controller to the CPU.

When the CPU acknowledges the Hold request, the CPU control, address, and data lines are tri-stated. The CPU Controller controls the direction and enables the memory or peripheral address and data buses that correspond to the requested DMA operation.

During the DMA operation, the 8237A-5 acts as the bus master and, along with the CPU Controller IC, generates all bus control signals and address and data signals. The DMA transfers continue for the number of counts and to the destination address that was previously programmed into the DMA registers. See the device data sheet and the IO map for complete descriptions of the registers, their locations, and their functions.

DMA Bus Cycles. During the data bus cycle, the 8237A-5 first outputs the upper address (A8-A15) on its data outputs (XD0-XD7), to be latched in the buffer internally to the DRAM/DMA Control by the Address Strobe signal (AS) from the 8237A-5. Next, the lower address (A0-A7) is put directly on the S address bus by the 8237A-5.

The DMA request acknowledge signals, DACK2 and DACK3-, are used along with RFRSH- and ACK* to enable the page register to be output as the upper address (SA16 and A17 through A23), which are buffered by U22 to the system expansion slots.

A DMA bus cycle can be extended by the RDY input of the DMA controller. The DMA memory read DMAMR is routed to the CPU Controller IC for extending the DMA bus cycle by inserting on DMA clock period as a wait state. The CPU Controller inserts the wait state by controlling the DMARDY input of the DMA controller.

I/O devices can extend the DMA bus cycle by controlling the IOCHRDY signal of the expansion bus. Setup times must be observed for IOCHRDY to be recognized.

RS-232 Serial Port Interface

The RS-232 Port is a single-channel, asynchronous communications port. The heart of the serial port is the PSSJ custom IC that functions as a serial data input/output interface. It performs serial-to-parallel conversion on data characters received from a peripheral device or modem and parallel-to-serial conversion on data characters received from the CPU.

Status information reported includes the type and condition of the ACE's transfer operations as well as any error conditions detected during serial data operations. The PSSJ custom IC includes a programmable Baud Rate Generator that allows operation from 50 to 9600 Baud. The PSSJ custom IC is supplied with a clock of 14 MHz from oscillator (Y2). The PSSJ can be tailored to the user's requirements by being able to remove start bits, stop bits, and parity bits. It supports 5, 6, 7, or 8 data bit characters with 1, 1½, or 2 stop bits. Diagnostic capabilities provide loopback functions of transmit/receive and input/output signals.

The PSSJ serial port is programmed by selecting the I/O address 3F8 - 3FE hex for primary and 2FB-2FE hex for secondary and writing data out to the port. Address bits A0, A1, and A2 are used to define the modes of operation by selecting the different registers to be programmed or read.

One interrupt is provided to the system from IRQ4 for primary operation and IRQ3 for secondary operation. This interrupt is active high. Bit 3 of the modem control register must be set high in order to send interrupts to the system. When this bit is high, any interrupts allowed by the interrupt enable register cause an interrupt.

Parallel Printer Port Interface

The final I/O interface of the Tandy 1000 TL is the Printer Interface contained in the PSSJ custom IC (U11). This part supplies all the signals required to interface to a typical parallel printer. These signals are 8 data out lines, plus various handshake control signals. Also, the printer interface generates an interrupt to the CPU if enabled.

Expansion Ports

System Expansion Bus

This section identifies the I/O interface requirements for the 8-bit, PC-compatible option cards. Each of the five slots has a 62-pin connector socket.

The following connector pin assignment is used on the PC option slots; this connector socket has 62 pins.

Pin	Signal Name	I/O	Pin	Signal Name	I/O
A1	NMI-				
A2	SD7	I/O	B2	BRESET	O
A3	SD6	I/O	B3	+5V	POWER
A4	SD5	I/O	B4	IRQ2	I
A5	SD4	I/O	B5	-5V	POWER
A6	SD3	I/O	B6	FDCDRQ	I
A7	SD2	I/O	B7	-12V	POWER
A8	SD1	I/O	B8	N/C	
A9	SD0	I/O	B9	+12V	POWER
A10	IOCHRDY	I	B10	GND	GROUND
A11	AEN	O	B11	SMEMW-	O
A12	SA19	O	B12	SMEMR-	O
A13	SA18	O	B13	IOW-	O
A14	SA17	O	B14	IOR-	O
A15	SA16	O	B15	DACK3-	O
A16	SA15	O	B16	DRQ3	I
A17	SA14	O	B17	DACK1-	O
A18	SA13	O	B18	DRQ1	I
A19	SA12	O	B19	RFRSH-	O
A20	SA11	O	B20	SCLK	O
A21	SA10	O	B21	IRQ7	I
A22	SA9	O	B22	IRQ6	I
A23	SA8	O	B23	IRQ5	I
A24	SA7	O	B24	IRQ4	I
A25	SA6	O	B25	IRQ3	I
A26	SA5	O	B26	FDCDACK-	O
A27	SA4	O	B27	T/C	O
A28	SA3	O	B28	BALE	O
A29	SA2	O	B29	+5V	POWER
A30	SA1	O	B30	OSC	O
A31	SAO	O	B31	GND	GROUND

Expansion Bus Signal Description

The following signal descriptions for the System I/O Bus are for PC bus-compatible option cards. Note that all signal lines are TTL compatible levels and that I/O adapters should be designed with a maximum of two low power Shottky (LS) loads per line.

SCLK (B20). SCLK is the System clock and has a period of 125ns in 8 MHz mode, or 250ns in 4 MHz mode. It has a 50% duty cycle and is used only for synchronization with the CPU. It is not intended for uses requiring a fixed frequency.

SA0 through SA19 (A12-A31). These lines are 20 address bits used to address memory and I/O devices within the Tandy 1000 TL. They are gated on the system bus when the BALE signal is high and are latched on the falling edge of the BALE signal. Generation of these signals is accomplished by the CPU or a DMA controller. SA0-SA19 are active high.

BALE (B28). BALE is a Buffered Address Latch Enable generated by the CPU Control IC. It is used to latch valid addresses from the CPU, and can be used by an I/O board to indicate a valid CPU address, in conjunction with AEN. BALE is pulled to a high state during DMA cycles, which include Refresh cycles. BALE is active high.

AEN (A11). AEN is an Address Enable signal used to remove the CPU and other devices from the bus to allow DMA transfers to take place. During AEN active, the DMA controller has control of the address bus, the data bus, the READ command lines, and the WRITE command lines. AEN is active high.

SDO through SD7 (A2-A9). These signals are the data bus Bits 0 through 7 from the CPU to memory and I/O devices on the bus. SDO is the least significant bit (lsb), and SD7 is the most significant bit (msb).

BRESET (B2). BRESET is used to reset or initialize the expansion logic during power-up time, line voltage outage, or when the Reset switch on the front panel is pressed. BRESET is active high.

NMI- (A1). This signal indicates an uncorrectable system error when active. The NMI- signal provides the system board with parity information about memory or devices on the bus. NMI- is active low.

IOCHRDY (A10). This signal is used to lengthen I/O or memory cycles when driven low by the active device. (This signal should not be held low more than 15 microseconds.) Any slow device using this line should drive it low immediately upon detecting its valid address and a READ or WRITE command. See the timing diagram for setup times. IOCHRDY is active high (Ready condition).

IRQ2 through IRQ7 (B4, B21-B25). These signals are used to tell the CPU that an I/O device needs attention. The Interrupt Requests are prioritized with IRQ2 having the highest priority and IRQ7 the lowest. An Interrupt Request is generated when any IRQ signal is driven high and held high until the CPU acknowledges the interrupt.

IOR- (B14). IOR- is a read signal that instructs an I/O device to drive its data onto the data bus (SD0-SD7). This line can be driven by the CPU Control IC or by the DMA controller. IOR- is active low.

IOW- (B13). IOW- is a write signal that instructs an I/O device to read, or latch, the data from the data bus (SD0-SD7). This line can be driven by the CPU Control IC or by the DMA controller. IOW- is active low.

SMEMR- (B12). SMEMR- is a read signal that instructs a memory device to drive its data onto the data bus (SD0-SD7). This line can be driven by the CPU Control IC or by the DMA controller through the CPU Control IC. SMEMR- is active only when the memory address is within the first 1 megabyte range (000000-0FFFFFH). SMEMR- is active low.

SMEMW- (B11). SMEMW- is a write signal that instructs a memory device to read, or latch, the data from the data bus (SD0-SD7). This line can be driven by the CPU Control IC or by the DMA controller through the CPU Control IC. SMEMW- is active only when the memory address is within the first 1 megabyte range (000000-0FFFFFH). SMEMW- is active low.

DRQ1, FDCDRQ, and DRQ3 (B18, B6, B16). These lines are asynchronous DMA requests by peripheral devices to gain DMA service. They are prioritized with DRQ1 having the highest priority, FDCDRQ next, and DRQ3 lowest. A DMA request is generated by driving a DRQ line active high and holding it until the corresponding DACK (DMA acknowledge) signal goes active. DRQ1, FDCDRQ, and DRQ3 perform only 8-bit transfers. All DRQ lines are active high.

DACK1-, FDCACK-, and DACK3-, B26, B15). These lines are DMA acknowledge signals used to acknowledge DMA requests DRQ1, FDCDRQ, and DRQ3. All DACK signals are active low.

RFRSH- (B19). This signal is used to indicate a refresh cycle that can be used by a memory board to refresh Dynamic memory. RFRSH- is active low and generated every 15 usec.

T/C (B27). T/C is a signal that provides a pulse when the terminal count for any DMA channel is reached. T/C is active high.

OSC (B30). OSC is an oscillator signal that is a high-speed clock with a 70 nanosecond period (14.31818 megahertz). It has a 50% duty cycle.

Memory Map

Address	Name	Allocated Function
000000-07FFFF	512K System RAM	System Memory
080000-09FFFF	128K System/Video RAM or 128K Expansion Memory	System Memory and Video Display Memory or System Memory
0A0000-0BFFFF	128K Video RAM	Reserved for Graphics Display Memory
0E0000-0FFFFF, EE0000-EFFFFF, or FE0000-FFFFFF	16K BIOS ROM Memory	Reserved for BIOS ROM Memory

I/O Port Map of System

I/O Port Map Summary

Block	Usage	Function
0000-001F	0000-000F	DMA Function
0020-003F	0020-0027	Interrupt Controller
0040-005F	0040-0047	Timer
0060-007F	0060-0067	PIO Function
0080-009F	0080-008F	DMA Page Register
00A0-00BF	00A0-00A7	NMI- Mask Register
00C0-00DF	00C0-00C7	Sound Generator
00E0-00FF	00F0-00FF	Numerical Coprocessor
0100-01FF		Reserved
0200-020F	0200-0207	Joystick Interface
0210-02F7		Reserved
02F8-02FF	02F8-02FF	Serial Port Secondary (COM2 optional)
0300-031F		Reserved
0320-032F		Hard Disk Controller (optional)
0330-036F		Reserved
0370-0377	0370-0377	Floppy Disk Controller 2 (optional)
0378-037F	0378-037F	Printer
0380-03CF		Reserved
03D0-03DF	03D0-03DF	System Video
03E0-03EF		Reserved
03F0-03F7	03F0-03F7	Floppy Disk Controller 1
03F8-03FF	03F8-03FF	Serial Port Primary (COM1)
0400-FFE7		Not Used
FFE8-FFEF	FFE8-FFEF	System Control Registers
FFF0-FFFF		Reserved

Must do JMP $+2 when programming DMA controller

Address **Description**

0000 DMA Controller
IOW- = 0: Channel 0 Base and Current Address
Internal Flip/Flop = 0: Write A0-A7
Internal Flip/Flop = 1: Write A8-A15
IOR- = 0: Channel 0 Current Address
Internal Flip/Flop = 0: Read A0-A7
Internal Flip/Flop = 1: Read A8-A15

0001 DMA Controller
IOW- = 0: Channel 0 Base and Current Word Count
Internal Flip/Flop = 0: Write W0-W7
Internal Flip/Flop = 1: Write AW-W15
IOR- = 0: Channel 0 Current Word Count
Internal Flip/Flop = 0: Read W0-W7
Internal Flip/Flop = 1: Read W8-W15

0002 DMA Controller
IOW- = 0: Channel 1 Base and Current Address
Internal Flip/Flop = 0: Write A0-A7
Internal Flip/Flop = 1: Write A8-A15
IOR- = 0: Channel 1 Current Address
Internal Flip/Flop = 0: Read A0-A7
Internal Flip/Flop = 1: Read A8-A15

0003 DMA Controller
IOW- = 0: Channel 1 Base and Current Word Count
Internal Flip/Flop = 0: Write W0-W7
Internal Flip/Flop = 1: Write AW-W15
IOR- = 0: Channel 1 Current Word Count
Internal Flip/Flop = 0: Read W0-W7
Internal Flip/Flop = 1: Read W8-W15

0004 DMA Controller
IOW- = 0: Channel 2 Base and Current Address
Internal Flip/Flop = 0: Write A0-A7
Internal Flip/Flop = 1: Write A8-A15
IOR- = 0: Channel 2 Current Address
Internal Flip/Flop = 0: Read A0-A7
Internal Flip/Flop = 1: Read A8-A15

0005 DMA Controller
IOW- = 0: Channel 2 Base and Current Word Count
Internal Flip/Flop = 0: Write W0-W7
Internal Flip/Flop = 1: Write AW-W15
IOR- = 0: Channel 2 Current Word Count
Internal Flip/Flop = 0: Read W0-W7
Internal Flip/Flop = 1: Read W8-W15

Address **Description**

0006 DMA Controller
IOW- = 0: Channel 3 Base and Current Address
Internal Flip/Flop = 0: Write A0-A7
Internal Flip/Flop = 1: Write A8-A15
IOR- = 0: Channel 3 Current Address
Internal Flip/Flop = 0: Read A0-A7
Internal Flip/Flop = 1: Read A8-A15

0007 DMA Contrcller
IOW- = 0: Channel 3 Base and Current Word Count
Internal Flip/Flop = 0: Write W0-W7
Internal Flip/Flop = 1: Write AW-W15
IOR- = 0: Channel 3 Current Word Count
Internal Flip/Flop = 0: Read W0-W7
Internal Flip/Flop = 1: Read W8-W15

0008 DMA Controller
IOW- = 0, Write Command Register

Bit	Description
0	0 = Memory to Memory Disable 1 = Memory to Memory Enable
1	0 = Channel 0 Address Hold Disable 1 = Channel 0 Address Hold Enable X If Bit 0 = 0
2	0 = Controller Enable 1 = Controller Disable
3	0 = Normal Timing 1 = Compressed Timing X If Bit 0 = 1
4	0 = Fixed Priority 1 = Rotating Priority
5	0 = Late Write Selection 1 = Extended Write Selection X If Bit 3 = 1
6	0 = DREQ Sense Active High 1 = DREQ Sense Active Low
7	0 = DACK Sense Active Low 1 = DACK Sense Active High

Address **Description**

0008 DMA Controller
IOR- = 0, Read Status Register

Bit	Description
0	1 = Channel 0 Has Reached TC
1	1 = Channel 1 Has Reached TC
2	1 = Channel 2 Has Reached TC
3	1 = Channel 3 Has Reached TC
4	1 = Channel 0 Request
5	1 = Channel 1 Request
6	1 = Channel 2 Request
7	1 = Channel 3 Request

0009 DMA Controller

IOW- = 0, Write Request Register

Bit	Description		
0-1	Bit 1	Bit 0	
	0	0	Select Channel 0
	0	1	Select Channel 1
	1	0	Select Channel 2
	1	1	Select Channel 3
2	0		Reset Request Bit
	1		Set Request Bit
3-7			Don't Care

IOR- = 0, Illegal

000A DMA Controller

IOW- = 0, Write Single Mask Register

Bit	Description		
0-1	Bit 1	Bit 0	
	0	0	Select Channel 0 Mask Bit
	0	1	Select Channel 1 Mask Bit
	1	0	Select Channel 2 Mask Bit
	1	1	Select Channel 3 Mask Bit
2	0		Clear Mask Bit (Enable Channel)
	1		Set Mask Bit (Disable Channel)
3-7			Don't Care

IOR- = 0, Illegal

Address **Description**

000B DMA Controller

IOW- = 0, Write Mode Register

Bit	Description		
0-1	Bit 1	Bit 0	
	0	0	Channel 0 Select
	0	1	Channel 1 Select
	1	0	Channel 2 Select
	1	1	Channel 3 Select
2-3	Bit 3	Bit 2	
	0	0	Verify Transfer
	0	1	Write Transfer To Memory
	1	0	Read Transfer To Memory
	1	1	Illegal
X	If Bits 6 and 7 = 11		
4	0		Autoinitialization Enable
	1		Autoinitialization Disable
5	0		Address Increment Select
	1		Address Decrement Select
6-7	Bit 7	Bit 6	
	0	0	Demand Mode Select
	0	1	Single Mode Select
	1	0	Block Mode Select
	1	1	Cascade Mode Select

Probably wrong. See 8237A spec.

IOR- = 0, Illegal

000C DMA Controller

IOW- = 0, Clear Byte Pointer Flip/Flop
IOR- = 0, Illegal

must do before R/W ports 0-7

000D DMA Controller

IOW- = 0, Master Clear
IOR- = 0, Read Temporary Register

000E DMA Controller

IOW- = 0, Clear Mask Register
IOR- = 0, Illegal

Address **Description**

000F DMA Controller

IOW- = 0, Write All Mask Register Bits

Bit	Description
0	0 = Clear Channel 0 Mask Bit (Enable) 1 = Set Channel 0 Mask Bit (Disable
1	0 = Clear Channel 1 Mask Bit (Enable) 1 = Set Channel 1 Mask Bit (Disable
2	0 = Clear Channel 2 Mask Bit (Enable) 1 = Set Channel 2 Mask Bit (Disable
3	0 = Clear Channel 3 Mask Bit (Enable) 1 = Set Channel 3 Mask Bit (Disable
4-7	Don't Care

IOR- = 0, Illegal

0010-001F Reserved

0020 8259A Interrupt Controller

Note: Initialization Words are set up by the operating system and are generally not to be changed. Writing an initialization word might cancel pending interrupts.

Bit 4 = 1 Initialization Command Word 1

Bit 0 = 0 ICW4 Needed
= 1 ICW4 Not Needed

Bit 1 = 0 Cascade Mode
= 1 Single Mode

Bit 2 Not Used

Bit 3 = 0 Edge Triggered Mode
= 1 Level Triggered Mode

Bit 5-7 Not Used

Bit 4 = 0 & Operation Control Word 2

Address **Description**

0020 8259A Interrupt Controller

Bit 3 = 0 Bits 0-2: Determine the Interrupt Level Acted on when the SL Bit is Active

Interrupt Level =	0	1	2	3	4	5	6	7
Bit 0 (L0):	0	1	0	1	0	1	0	1
Bit 1 (L1):	0	0	1	1	0	0	1	1
Bit 2 (L2):	0	0	0	0	1	1	1	1

Bits 5-7: Control Rotate and End of Interrupt Modes

B7	B6	B5	
0	0	1	Non-Specific EOI Command End of Interrupt
0	1	1	Specific EOI Command End of Interrupt
1	0	1	Rotate on Non-Specific EOI Auto Rotation
1	0	0	Rotate in Automatic EOI Mode (Set) Auto Rotation
0	0	0	Rotate in Automatic EOI Mode (Clear) Auto Rotation
1	1	1	*Rotate on Specific EOI Command Specific Rotation
1	1	0	*Set Priority Command Specific Rotation
0	1	0	No Operation

(*L0 - L2 Are Used)

Bit 4 = 0 & Operation Control Word 3

Bit 3 = 1 Bits 0-1:

Bit 1	Bit 0	- Read Register Command
0	0	No Action
0	1	No Action
1	0	Read IR Register on next IOR- Pulse
1	1	Read IS Register on next IOR- Pulse

Bit 2 = 0: No Poll Command
= 1: Poll Command

Bits 5-6

Bit 5	Bit 6	- Special Mask Mode
0	0	No Action
0	1	No Action
1	0	Reset Special Mask
1	1	Set Special Mask

Bit 7 = 0

Address **Description**

0021 8259A Interrupt Controller

Initialization Control Word 2

Bits 0-7: Not Used

Bits 3-7: T3-T7 Of Interrupt Vector Address
(8086/8088/80286 Mode)

Initialization Control Word 3 (Master Device)

Bits 0-7: = 1 Indicated IR Input has a Slave
= 0 Indicated IR Input does not have a Slave

Initialization Control Word 3 (Slave Device)

Bits 0-2: = ID0-2

Bit 0	Bit 1	Bit 2 -	Slave ID #
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Bits 3-7: = 0 (Not Used)

Initialization Control Word 4
Bit 0: Type of Processor

=0 MCS-80/85 Mode
=1 8086/8088/80286 Mode

Bit 1: Type of End of Interrupt

= 0 Normal EOI
= 1 Auto EOI

Bits 2-3: Buffering Mode

Bit 3	Bit 2	
0	X	Non-Buffered Mode
1	0	Buffered Mode/Slave
1	1	Buffered Mode/Master

Address	Description
	Initialization Control Word 4 Bit 4: Nesting Mode =0 Not Special Fully Nested Mode =1 Special Fully Nested Mode Bits 5-7: =0 (Not Used)
	Operation Control Word 1 (IOR-) Bits 0-7: Interrupt Mask for IRQ0-IRQ7 =0 Mask Reset (Enable) =1 Mask Set (Disable)
	Note: Peripherals requesting an interrupt service must generate a low to high edge and then remain at a logic high level until service is acknowledged. Failure to do so results in a Default Service for IRQ7.
0022-0027	Same as 0020-0021
0028-003F	Not Used

0040	8254-2 Timer
	IOW- = 0: Load Counter No. 0
	IOR- = 0: Read Counter No. 0

0041	8254-2 Timer
	IOW- = 0: Load Counter No. 1
	IOR- = 0: Read Counter No. 1

0042	8254-2 Timer
	IOW- = 0: Load Counter No. 2
	IOR- = 0: Read Counter No. 2

Address **Description**

0043 8254-2 Timer

IOW- = 0: Write Mode Word

Control Word Format

Bit 0: BCD
=0: BCD Counter (4 Decades)
=1: Binary Counter 16 Bits

Wrong. See 8254 data sheet.

Bits 1-3: Mode Selection

Bit 3	Bit 2	Bit 1	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

Bits 4-5: Read/Load

Bit 5	Bit 4	
0	0	Counter Latching Operation
0	1	Read/Load LSB Only
1	0	Read/Load MSB Only
1	1	Read/Load LSB First, Then MSB

Bits 6-7: Select Counter

Bit 7	Bit 6	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Illegal

IOR- = 0: No-Operation 3-State

0044-0047 Same as 0040-0043

0048-005F Not Used

Address	Description
0060	Port A / Keyboard Interface Control Ports (Read Only)

Bit	Description
0	Keyboard Bit 0-LSB
1	Keyboard Bit 1
2	Keyboard Bit 2
3	Keyboard Bit 3
4	Keyboard Bit 4
5	Keyboard Bit 5
6	Keyboard Bit 6
7	Keyboard Bit 7-MSB

0061	Port B Read or Write

Bit	Description
0	8253 Gate 2 Enable 1 = Enable 0 = Disable
1	Speaker Data Out Enable 1 = Enable 0 = Disable
2	Not Used
3	Not Used
4	Internal Speaker Enable 1 = Disable 0 = Enable
5	Not Used
6	HOLDCK (if IBM PC Keyboard Mode) 1 = Tristate Keyboard Clock Line 0 = Pull Keyboard Clock Line Low
7	Keyboard Clear 1 = Clear Buffer and Reset Keyboard Interrupt 0 = Release Clear and Reset

Address **Description**

0062 Port C Read/Write: Bits 0-3; Read Only Bits: 4-7

Bit	Description
0	Not Used (Read/Write)
1	Not Used (Read/Write)
2	Not Used (Read/Write)
3	(Output) CPU Clock Rate (Read/Write) 0 = 4.00 MHz (PC Compatible Rate) 1 = 8.00 MHz (Default By Boot ROM)
4	EEPROM Data Input (Read Only)
5	8253 Out #2 (Read Only)
6	Monochrome Mode 0 = Color Monitor 1 = 350 Line Monitor, Mono
7	Reserved

0063-0064 Reserved

0065 Planar Control Register (Read/Write)

not really. Always reads Ø8h or ØØh, reflects bit 3 only.

1 = Enable
0 = Disable

Bit	Description
0	Hard Disk Select Enable
1	Parallel Port Select Enable
2	Video Port Select Enable
3	Floppy Disk Port Select Enable
4	Serial Port Select Enable
5	Reserved
6	Reserved
7	Parallel Port Output Enable

Disabling video ports continues to display, but disables writes to the buffer, reenabling reenables writes. Buffer contents unchanged.

0066-0067 Reserved

0068-007F Reserved

0080 DMA Page Register (Reserved for Diagnostics)
Write Only

<table>
<tr><th>Address</th><th>Description</th></tr>
<tr><td>0081</td><td>DMA Channel 2 Page Register (Write Only)
<table>
<tr><th>Address</th><th>Description</th></tr>
<tr><td>Bit 0</td><td>Address A16</td></tr>
<tr><td>Bit 1</td><td>Address A17</td></tr>
<tr><td>Bit 2</td><td>Address A18</td></tr>
<tr><td>Bit 3</td><td>Address A19</td></tr>
<tr><td>Bit 4</td><td>Address A20</td></tr>
<tr><td>Bit 5</td><td>Address A21</td></tr>
<tr><td>Bit 6</td><td>Address A22</td></tr>
<tr><td>Bit 7</td><td>Address A23</td></tr>
</table></td></tr>
<tr><td>0082</td><td>DMA Channel 3 Page Register (Write Only)
<table>
<tr><th>Address</th><th>Description</th></tr>
<tr><td>Bit 0</td><td>Address A16</td></tr>
<tr><td>Bit 1</td><td>Address A17</td></tr>
<tr><td>Bit 2</td><td>Address A18</td></tr>
<tr><td>Bit 3</td><td>Address A19</td></tr>
<tr><td>Bit 4</td><td>Address A20</td></tr>
<tr><td>Bit 5</td><td>Address A21</td></tr>
<tr><td>Bit 6</td><td>Address A22</td></tr>
<tr><td>Bit 7</td><td>Address A23</td></tr>
</table></td></tr>
<tr><td>0083</td><td>DMA Channel 0-1 Page Register (Write Only)
<table>
<tr><th>Address</th><th>Description</th></tr>
<tr><td>Bit 0</td><td>Address A16</td></tr>
<tr><td>Bit 1</td><td>Address A17</td></tr>
<tr><td>Bit 2</td><td>Address A18</td></tr>
<tr><td>Bit 3</td><td>Address A19</td></tr>
<tr><td>Bit 4</td><td>Address A20</td></tr>
<tr><td>Bit 5</td><td>Address A21</td></tr>
<tr><td>Bit 6</td><td>Address A22</td></tr>
<tr><td>Bit 7</td><td>Address A23</td></tr>
</table></td></tr>
<tr><td>0084-008F</td><td>Same as 0080-0083</td></tr>
</table>

Address	Description
00A0	NMI- Mask Register, Write Only

Bit	Description
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Not Used
6	Not Used
7	Non Maskable Interrupt (NMI) Enable 0 = Disabled 1 = Enabled

00A1-00A7	Reserved
00A8-00AF	Not Used
00C0-00C3	Sound SN76496

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0	
= 1	0	0	0	F6	F7	F8	F9	Update Tone Frequency 1
= 0	X	F0	F1	F2	F3	F4	F5	Additional Frequency Data
= 1	0	0	1	A0	A1	A2	A3	Update Tone Attenuation 1
= 1	0	1	0	F6	F7	F8	F9	Update Tone Frequency 2
= 0	X	F0	F1	F2	F3	F4	F5	Additional Frequency Data
= 1	0	1	1	A0	A1	A2	A3	Update Tone Attenuation 2
= 1	1	0	0	F6	F7	F8	F9	Update Tone Frequency 3
= 0	X	F0	F1	F2	F3	F4	F5	Additional Frequency Data
= 1	1	0	1	A0	A1	A2	A3	Update Tone Attenuation 3
= 1	1	1	0	X	FB	NF0	NFI	Update Noise Control
= 1	1	1	1	A0	A1	A2	A3	Update Noise Attenuation

Address **Description**

00C4-00C7 DAC Function

00C4 DAC Mode Register (Write Commands)

Bits 0-1

Bit 0	Bit 1	Dac Function Selected
0	0	Joystick
0	1	Successive Approximation
1	0	Sound Channel
1	1	Direct Write to DAC

This is wrong. See section on PSSJ for correct info.

Bit 2 DMA Enable (for SA, Direct R/W)

0 DMA Disabled
1 DMA Enabled for SA, DA

Bit 3 DMA Interrupt Clear

0 DMA Interrupt Held Clear
1 DMA Interrupt Allowed

Bit 4 DMA Interrupt Enable

0 DMA EOP Interrupt Disabled
1 DMA EOP Interrupt Enabled

Bit 5 Sound Divider Sync Enable

0 Synchronization Disabled
1 Synchronization Enabled (Write to, 00C6 or 00C7 reloads all dividers)

Bit 6 Sound Chip Extra Divide Enable

0 Extra Divide Disabled
1 Extra Divide Enabled

Bit 7 Reserved

00C4 DAC Mode Register (Read Commands)

Bit 3 DMA Interrupt Flag. A DMA Interrupt has occurred. To clear the interrupt flag, Bit 3 must be brought low and then high again.

Bit 7 Successive Approximation Done. Useful when polling instead of using DMA.

Address **Description**

00C5 Waveshape Mode Select (Write Commands)

Bits 0-2

Bit 0	Bit 1	Bit 2	Duty Cycle
0	0	0	6.25%
0	0	1	12.5%
0	1	0	18.75%
0	1	1	25.0%
1	0	0	31.25%
1	0	1	37.5%
1	1	0	43.75%
1	1	1	50.0%

Bit 3 Reserved

Bit 4 Reserved

Bit 5 Reserved

Bits 6-7 Waveshape Select

Bit 7	Bit 6	Waveshape Selected
0	0	Pulse
0	1	Ramp
1	0	Triangle
1	1	Reserved

00C5 Read DAC Registers (Read Commands)

Direct Read of DAC when 00C4 Bits 0-1 = 1X
Direct Read of Control Register when 00C4 Bits 0-1 = 01

00C6 R/W Frequency lsb for DAC sound channel

Bit 0	F0
Bit 1	F1
Bit 2	F2
Bit 3	F3
Bit 4	F4
Bit 5	F5
Bit 6	F6
Bit 7	F7

Address	Description
00C7	R/W Amplitude/frequency msb for DAC sound channel

Bit 0	F8
Bit 1	F9
Bit 2	F10
Bit 3	F11
Bit 4	Reserved
Bit 5	AMP 1
Bit 6	AMP 2
Bit 7	AMP 3

00C8-00CF	Reserved
00E0-00EF	Reserved
00F0	Clear Numerical Coprocessor Busy
00F1	Reset Numerical Coprocessor to Real Mode
00F2	Same as 00F0
00F3	Same as 00F1
00F4	Same as 00F0
00F5	Same as 00F1
00F6	Same as 00F0
00F7	Same as 00F1
00F8-00FF	Math Coprocessor Chip Select
0100-01FF	Reserved
0200-0207	Joystick
	Clear (Resets Integrator to 0)
0201	Read R = Right Joystick, L = Left Joystick

Bit	Description
0	R - X Horizontal Position
1	R - Y Vertical Position
2	L - X Horizontal Position
3	L - Y Vertical Position
4	R Button #1 (Logic 0 = Button Pressed)
5	R Button #2 (Logic 0 = Button Pressed)
6	L Button #1 (Logic 0 = Button Pressed)
7	L Button #2 (Logic 0 = Button Pressed)

0208-020F	Not Used
0210-02F7	Reserved

Address	Description
2F8-2FF	Serial Port Secondary (COM2 Optional)
0300-036F	Reserved
0370-0377	Floppy Disk Controller 2 (optional)
0378	Printer - Data Latch

Bit	Description
0	Bit 0 - LSB
1	Bit 1
2	Bit 2
3	Bit 3
4	Bit 4
5	Bit 5
6	Bit 6
7	Bit 7 - MSB

0379	Printer - Read Status

Bit	Description
0	Not Used
1	Not Used
2	Not Used
3	"0" = Error (Fault)
4	"1" = Printer Select In
5	"0" = Out of Paper (Paper Empty)
6	"0" = Acknowledge
7	"0" = Busy

037A	Printer - Control Latch

Bit	Description
0	"0" = Strobe
1	"0" = Auto FD XT
2	"0" = Initialize
3	"0" = Select Printer Out
4	"1" = Enable Interrupt
5	"0" = Enable Output Data
6	Not Used
7	Not Used

037B	Not Used

Address	Description
037C	EEPROM Control Register (Write Only) Bit Description 0 NOVDO - EEPROM Data Output 1 NOVCE - EEPROM Chip Enable 0 = Enabled or Selected 1 = Disabled or Deselected 2 NOVCLK - EEPROM Clock 0 = Toggle Clock Low 1 = Toggle Clock High
037D-037F	Reserved
0378-03CF	Not Used ξ some 3Bx used for mono video (q.v.)
03D0-03D3	Reserved
03D4	6845 Address Register
03D5	6845 Data Register
03D6	Not Used
03D7	Not Used
03D8	Mode Select Register Bit0 High Resolution Clock = 0 Selects 40 X 25 Alphanumeric Mode = 1 Selects 80 X 25 Alphanumeric Mode Bit1 Graphics Select = 0 Selects Alphanumeric Mode = 1 Selects 320 X 200 Graphics Mode Bit2 Black and White = 0 Selects Color Mode = 1 Selects Black and White Mode Bit3 Video Enable = 0 Disables Video Signal = 1 Enables Video Signal Bit4 640 Dot Graphics = 0 Disables 640 X 200 B&W Graphics Mode = 1 Enables 640 X 200 B&W Graphics Mode Bit5 Blink Enable = 0 Disables Blinking = 1 Enables Blinking

Address Description

03D9 Color Select Register

Bit 0	Background Blue
Bit 1	Background Green
Bit 2	Background Red
Bit 3	Background Intensity
Bit 4	Foreground Intensity
Bit 5	Color Select

03DA-03DE Write Video Array Address and Read Status (03DA)
Write Video Array Data (03DE)

Read	(03DA)		Write (03DE)
00	Bit 0	Display Inactive	Not Used
00	Bit 1	Light Pen Set	Not Used
00	Bit 2	Light Switch Status	Not Used
00	Bit 3	Vertical Retrace	Not Used
00	Bit 4	Not Used	Not Used
01	Bit 0		Palette Mask 0
01	Bit 1		Palette Mask 1
01	Bit 2		Palette Mask 2
01	Bit 3		Palette Mask 3
02	Bit 0		Border Blue
02	Bit 1		Border Green
02	Bit 2		Border Red
02	Bit 3		Border Intensity
02	Bit 5		Reserved = 0
03	Bit 0		Mono Enable = 1
03	Bit 1		Reserved = 0
03	Bit 2		Border Enable
03	Bit 3		4-Color High Resolution
03	Bit 4		16-Color Mode
03	Bit 5		Extra Video Mode

03DB Clear Light Pen Latch

03DC Preset Light Pen Latch

03DD Extended RAM Page Register - CPU Relative

Bit	Description
0	Extended Addressing Modes
1	Not Used
2	Not Used
3	CRT Video Page Address "17"
4	CRT Video Page Address "18"
5	CPU Page Address "17"
6	CPU Page Address "18"
7	Select 64K Or 256K RAM

Address **Description**

03DF CRT Processor Page Register - Video Memory Relative

Bit 0	A14	CRT Page 0
Bit 1	A15	CRT Page 1
Bit 2	A16	CRT Page 2
Bit 3	A14	Processor Page 0
Bit 4	A15	Processor Page 1
Bit 5	A16	Processor Page 2
Bit 6		Video Address Mode 0
Bit 7		Video Address Mode 1

Video	D0	D7	D6
Descriptions	3DDH	3DFH	3DFH
8p 1 - 16K	0	0	0
8p 2 - 8K	0	0	1
4p 2 - 16K	0	1	0
4p 4 - 8K	0	1	1
4p 1 - 32K	1	0	0
2p 2 - 32K	1	0	1

03E0-03EF Reserved

03F0 Not Used

03F1 Drive Select Switch

Bit 0	Not Used
Bit 1	"1" DS0 = DS0
	"0" DS0 = DS1
Bit 2	Not Used
Bit 3	Not Used
Bit 4	Not Used
Bit 5	Not Used
Bit 6	Not Used
Bit 7	Not Used

writing FF to Ø3F1
swaps A: and B:

03F2 DOR Register (Write Only)

Bits 0-1 Drive Select

Bit 1	Bit 0	
0	0	Drive Select A*
0	1	Drive Select B*

Bit 2	0 = FDC Reset
Bit 3	1 = Enable DMA Request/Interrupt
Bit 4	1 = Drive A Motor On
Bit 5	1 = Drive B Motor On
Bit 6	1 = FDC Terminal Count
Bit 7	Not Used

03F3 Not Used

03F4 FDC - Status (Read Only). See FDC Specification

03F5 FDC - Data (R/W). See FDC Specification

Address **Description**

03F6 Reserved

03F7 FDC Data Rate Selection

Bit	Description
0	Not Used
1	Write - Data Rate 0 = 500K bits per second 1 = 250K bits per second
2	Not Used
3	Not Used
4	Not Used
5	Not Used
6	Not Used
7	0=Disk Change

03F8-03FF Serial Port Primary (COM1)

03F8 Write Transmitter Holding Register (Character to Send)

Bit	Description
0	Bit 0 - LSB (First Bit sent Serially)
1	Bit 1
2	Bit 2
3	Bit 3
4	Bit 4
5	Bit 5
6	Bit 6
7	Bit 7 - MSB

03F8 Read Receiver Buffer Register (Character Received)

Bit	Description
0	Bit 0 - LSB (First Bit Received Serially)
1	Bit 1
2	Bit 2
3	Bit 3
4	Bit 4
5	Bit 5
6	Bit 6
7	Bit 7 - MSB

Address **Description**

03F8 Divisor Latch LSB (Divisor Latch Access Bit DLAB = "1")

Bit	Description
0	Bit 0
1	Bit 1
2	Bit 2
3	Bit 3
4	Bit 4
5	Bit 5
6	Bit 6
7	Bit 7

03F9 Divisor Latch MSB (Divisor Latch Access Bit DLAB = "1")

Bit	Description
0	Bit 0
1	Bit 1
2	Bit 2
3	Bit 3
4	Bit 4
5	Bit 5
6	Bit 6
7	Bit 7

03F9 Interrupt Enable Register

Bit	Description
0	"1" = Enables the Received Data Available Int
1	"1" = Enables the Transmitter Holding Register Int
2	"1" = Enables Receive Line Status Interrupt
3	"1" = Enables the Modem Status Interrupt
4-7	Always Logical "0"

03FA Interrupt Identification Register

Bit	Description		
0	"0" = Interrupt Pending		
1-2	Bit 2	Bit 1	
	"0"	"0"	Fourth Level Priority
	"0"	"1"	Third Level Priority
	"1"	"0"	Second Level Priority
	"1"	"1"	Highest Level Priority
3-7	Always Logical "0"		

Address **Description**

03FB Line Control Register

Bit Description

0-1 Bit 1 Bit 0

Bit 1	Bit 0	
"0"	"0"	Five Bit Word Length
"0"	"1"	Six Bit Word Length
"1"	"0"	Seven Bit Word Length
"1"	"1"	Eight Bit Word Length

2 "0" = One Stop Bit
"1" = 1½ Stop Bits when Five Bit Length Selected
Two Stop Bits with Six, Seven, or Eight Bit

3 "1" = Parity Enable

4 "0" = Odd Parity Select
"1" = Even Parity Select

5 Stick Parity Bit

6 "1" = Set Break Enable

7 "1" = Divisor Latch Access Bit Enable

03FC Modem Control Register

Bit Description

0 "1" = Data Terminal Ready Set (DTR)
"0" = Data Terminal Ready Reset (DTR)

1 Request To Send (RTS)

2 Out 1

3 Out 2

4 Loop

5-7 Always Logical "0"

03FD Line Status Register

Bit Description

0 Data Ready (DR)

1 Overrun Error (OR)

2 "1" = Detect Parity Error (PE)

3 "1" = Detect Framing Error (FE)

4 "1" = Break Interrupt (BI)

5 Transmitter Holding Register
"1" = Character Transferred from Holding to Shift Register
"0" = Loading Transmitter Holding Register

6 Transmitter Shift Register Empty
"1" = Shift Register Idle
"0" = Data Transfer from Holding Register

7 Always Logical "0"

Address	Description
03FE	Modem Status Register

Bit	Description
0	Delta Clear to Send (DCTS)
1	Delta Data Set Ready (DDSR)
2	Trailing Edge Ring Indicator "1" = On "0" = Off
3	Delta Received Line Signal Detect (If Bit 0, 1, 2, or 3 is set to a "1" modem status interrupt is generated
4	"0" = Clear to Send (CTS)
5	"0" = Data Set Ready (DSR)
6	"0" = Ring Indicator (RI)
7	"0" = Received Line Signal Detect (RLSD)

Address	Description
03FF	Reserved
0400-FFE7	Not Used
FFE8-FFEB	System Control Registers
FFE8	Video Configuration Register (write-only)

Bit	Description
0	Reserved
1	Memory Configuration 1 } must be 111b w/128k upgrade;
2	Memory Configuration 2 }
3	Memory Configuration 3 } probably 100b w/o
4	Reserved
5	16-Bit CPU Memory = 1 (must be Ø)
6	Reserved
7	Reserved

Address	Description
FFE9	Reserved

Address	Description
FFEA	WRITE/READ

Bit	Description
Bit 0:	ROM PAGING 0
Bit 1:	ROM PAGING 1
Bit 2:	ROM PAGING 2
Bit 3:	ROM PAGING 3
Bit 4:	ROM PAGING 4
Bit 5:	System Type (Reserved - See Note)
6-7	Bit 7 Bit 6

Bit 7	Bit 6	
0	0	512K System Memory
0	1	512K System Memory
1	0	512K System Memory
1	1	640K System Memory

NOTE: When reading Port FFEA, Bit 4 will be inverted from what was written, (i.e. when a 0 is written, a 1 will be read; when a 1 is written, a 0 will be read.) FFEA Bit 4 can be used to determine the system type. If Bit 4 is read back inverted, the system is identified as a Tandy 1000 SL. If Bit 4 is NOT inverted, the system is identified as a Tandy 1000 TL.

Writing 1 to bit 4 disables access to the ROM segment at EØØØØ.

Writing Ø to bit 6 or 7, with the 128k video RAM upgrade installed, disables access to memory between 8ØØØØ-9FFFF.

ROM Paging Definition:

4 1 Meg X 8 ROMs	ADDRESS				ROM PAGES					ROM CS		SELECT			64K Page	
	19	18	17	16	4	3	2	1	0	#0	#1	2	1	0	ROM 0	ROM 1
F0000-FFFFF	1	1	1	1	x	x	x	x	x	0	1	x	1	1	1	
E0000-EFFFF	1	1	1	0	1	x	1	1	1	1	1	x	x	x		
E0000-EFFFF	1	1	1	0	1	x	1	1	0	0	1	x	1	0	2	
E0000-EFFFF	1	1	1	0	1	x	1	0	1	0	1	x	x	1	3	
E0000-EFFFF	1	1	1	0	1	x	1	0	0	0	1	x	0	0	4	
E0000-EFFFF	1	1	1	0	1	x	0	1	1	1	0	x	1	1		1
E0000-EFFFF	1	1	1	0	1	x	0	1	0	1	0	x	1	0		2
E0000-EFFFF	1	1	1	0	1	x	0	0	1	1	0	x	0	1		3
E0000-EFFFF	1	1	1	0	1	x	0	0	0	1	0	x	0	0		4

2 2 Meg X 8 ROMs	19	18	17	16	4	3	2	1	0	#0	#1	2	1	0	ROM 0	ROM 1
F0000-FFFFF	1	1	1	1	x	x	x	x	x	0	1	1	1	1	1	
E0000-EFFFF	1	1	1	0	0	1	1	1	1	1	1	x	x	x		
E0000-EFFFF	1	1	1	0	0	1	1	1	0	0	1	1	1	0	2	
E0000-EFFFF	1	1	1	0	0	1	1	0	1	0	1	1	0	1	3	
E0000-EFFFF	1	1	1	0	0	1	1	0	0	0	1	1	0	0	4	
E0000-EFFFF	1	1	1	0	0	1	0	1	1	0	1	0	1	1	5	
E0000-EFFFF	1	1	1	0	0	1	0	1	0	0	1	0	1	0	6	
E0000-EFFFF	1	1	1	0	0	1	0	0	1	0	1	0	0	1	7	
E0000-EFFFF	1	1	1	0	0	1	0	0	0	0	1	0	0	0	8	
E0000-EFFFF	1	1	1	0	0	0	1	1	1	1	0	1	1	1		1
E0000-EFFFF	1	1	1	0	0	0	1	1	0	1	0	1	1	0		2
E0000-EFFFF	1	1	1	0	0	0	1	0	1	1	0	1	0	1		3
E0000-EFFFF	1	1	1	0	0	0	1	0	0	1	0	1	0	0		4
E0000-EFFFF	1	1	1	0	0	0	0	1	1	1	0	0	1	1		5
E0000-EFFFF	1	1	1	0	0	0	0	1	0	1	0	0	1	0		6
E0000-EFFFF	1	1	1	0	0	0	0	0	1	1	0	0	0	1		7
E0000-EFFFF	1	1	1	0	0	0	0	0	0	1	0	0	0	0		8

have this: 512 k ROM on 2 256k chips = 8 64k pages.

Address	Description		
FFEB	UART Clock, Joystick, and Sound Enable		
		Bit	Description
		0	0 = Clock Divided by 13 1 = Clock Divided by 1
		1	0 = Disable Joystick 1 = Enable Joystick
		2	0 = Disable Sound Chip 1 = Enable Sound Chip
		3	Reserved
		4	Reserved
		5	Keyboard Interrupt (Read Only) 1 = Keyboard Interrupt Active 0 = Keyboard Interrupt Inactive
		6	Reserved
		7	Keyboard Type (Read Only) 1 = 101 Key Enhanced Keyboard 0 = Original Tandy 1000 Keyboard
FFEC-FFEF	Reserved		
FFF0-FFFF	Reserved		

ELECTRICAL BILL OF MATERIAL - TANDY 1000 TL

MAIN LOGIC ASSY. TANDY 1000 TL

QTY.	DESCRIPTION	DESIGNATOR	VENDOR	PART NUMBER
1	CPU PCB TANDY 1000 TL REV. A			8709842
8	STAKING PINS	E1-8	AMP#1-87022-0	8529014
1	SOCKET 8-PIN DIP	U12	AMP#640460-3	8509011
17	SOCKET 16-PIN DIP	U3,45-52,62-69	AMP#2-644100-3	8509036
8	SOCKET 18-PIN DIP	U32-39	AMP#2-383060-3	8509037
3	SOCKET 24-PIN DIP	U15,42,44	AMP#640962-3	8509029
1	SOCKET 28-PIN DIP	U24	AMP#2-641605-3	8509007
2	SOCKET 32-PIN DIP	U54,56 OR 55,57	AMP#2-644018-3	8505048
2	SOCKET 32-PIN DIP	U58,59	AMP#2-644018-3	8505048
3	SOCKET 40-PIN DIP	U23,28,60	AMP#2-641606-3	8509002
6	SOCKET 68-PIN, PLCC	U11,13,17,22, 31,61	AMP#821574-1 BURNDY#68P410T	8509020
1	SWITCH, RESET	S1	ALPS#KHC15901	8489065
1	STEREO HEAD. JACK	J14	HOSIDEN#HSJ0942-01-1020	8519322
1	MICROPHONE JACK	J16	HOSIDEN#HSJ0942-01-1060	8519355
1	BATTERY HOLDER	J13	SPECIALTY ELEC. 2S2032-0	8491013
1	CONNECTOR, 2-PIN	J15 (SPKR)	MOLEX#22-29-2021	8519193
2	CONNECTOR, 6-PIN	J3,4 (JOYSTICK)	HOSIDEN#TCS5040-16-1911	8519318
1	CONNECTOR, 7-PIN	J2 (KEYBOARD)	HOSIDEN#TCS5040-17-4071	8519358
1	CONNECTOR, 9-PIN	J12 (POWER)	MOLEX#26-48-1095	8519191
1	CONNECTOR, 9-PIN FEMALE "D" SUB	U1 (VIDEO)	HOLMBERG#H4S09RA28CM42 AMP#745988-3 MOLEX#82009-2052	8519279
1	CONNECTOR, 9-PIN "D" SUB	J5 (SERIAL)	AMP#747840-3	8519269
1	CONNECTOR, 17-PIN HEADER (SHROUDED)	J11 (FLOPPY)	MOLEX# 70246-3402	8519324
5	CONNECTOR, 31-PIN CARD EDGE	J6-10	TEKA#021-31014-200 VIKING#3KT31/2JFF12 JAE#PB21-62T1-C1-S AMP#6-530843-5 HOLMBERG#A8D31DS27C2 BURNDY#PWBH31DDS1B	8519236
1	RESISTOR, VAR. 10K	R63	PIHER PT15NH510KA	8270510
1	RESISTOR, VAR. TBD	R27		
1	RESISTOR 10 OHM 1/4 WATT 5%	R48	GENERIC	8207010
15	RESISTOR 33 OHM 1/4 WATT 5%	R6-9,17,21,22, 31,32,34-39	"	8207033
1	RESISTOR 300 OHM 1/4 WATT 5%	R50,49	"	8207130

ELECTRICAL BILL OF MATERIAL - TANDY 1000 TL

MAIN LOGIC ASSY. TANDY 1000 TL

QTY.	DESCRIPTION	DESIGNATOR	VENDOR	PART NUMBER
1	RESISTOR 510 OHM 1/4 WATT 5%	R24	"	8207151
2	RESISTOR 1K OHM 1/4 WATT 5%	R5,29	GENERIC	8207210
3	RESISTOR 1.2K OHM 1/4 WATT 1%	R54,55,57	"	8207212
1	RESISTOR 1.2K OHM 1/4 WATT 1%	R26		
1	RESISTOR 1.3K OHM 1/4 WATT 5%	R58	"	8207213
1	RESISTOR 2.2K OHM 1/4 WATT 5%	R51	"	8207222
1	RESISTOR 2.4K OHM 1/4 WATT 5%	R60	"	8207244
1	RESISTOR 2.7K OHM 1/4 WATT 5%	R62	"	8207227
1	RESISTOR 3.3K OHM 1/4 WATT 5%	R52	"	8207233
4	RESISTOR 4.7K OHM 1/4 WATT 5%	R33,41,42,61	"	8207247
9	RESISTOR 10K OHM 1/4 WATT 5%	R1-4,30,40,45-47	"	8207310
2	RESISTOR 13K OHM 1/4 WATT 5%	R53,56	"	8207313
5	RESISTOR 27K OHM 1/4 WATT 5%	R12,13,20,23,25	"	8207327
3	RESISTOR 47K OHM 1/4 WATT 5%	R14,19,43	"	8207347
1	RESISTOR 68K OHM 1/4 WATT 5%	R28	"	8207368
1	RESISTOR 91K OHM 1/4 WATT 5%	R59		8207391
2	RESISTOR 100K OHM 1/4 WATT 5%	R15,44	"	8207410
4	RESISTOR 1 MEG OHM 1/4 WATT 5%	R10,11,16,18	"	8207510
6	RESISTOR PAK 33 OHM 16-PIN DIP	RP1-4,12,13	"	8290044
5	RESISTOR PAK 33 OHM 8-PIN SIP	RP5,14,15,18,19	"	8295033
2	RESISTOR PAK 1K OHM 6-PIN SIP	RP7,10	"	8290210

ELECTRICAL BILL OF MATERIAL - TANDY 1000 TL

MAIN LOGIC ASSY. TANDY 1000 TL

QTY.	DESCRIPTION	DESIGNATOR	VENDOR	PART NUMBER
2	RES. PAK 4.7K OHM 10-PIN SIP	RP6,17	GENERIC	8294247
3	RES.PAK 10K OHM 6-PIN SIP	RP9,11,20	"	8290032
2	RES.PAK 10K OHM 8-PIN SIP	RP8,16	"	8292310
2	RES. PAK 10K OHM 10-PIN SIP	RP21,22	"	8290010
45	CAPACITOR 220 PFD 25V 10% SMD	C3,4,5,8,9, 10,11,12,13, 14,15,17,19, 20,21,22,23, 24,25,26,27, 28,29,30,31, 32,33,34,38, 39,40,41,42, 43,44,45,46, 47,48,49,50, 51,52,176,180	"	X30122243
7	CAPACITOR 330 PFD 25V 10% SMD	C6,7,18,56, 57,58,59	"	X30133244
9	CAPACITOR 47 PFD 50V 5% NPO CD	C70,75,76,78, 83,90,94,96, 98	"	8300475
2	CAP 1000 PFD 50V 10% Z5P CD	C169,171	"	8302104
5	CAP .022 MFD 50V 25C +80/-20% CD	C65,69,95,77,111	"	8303224
1	CAP .047 MFD 50V 10% 25P CD OR EQUIV.	C163	"	8373474
1	CAP .047 MFD 50V 10% X7R MONO.AX.	C131	"	8373484

ELECTRICAL BILL OF MATERIAL - TANDY 1000 TL

MAIN LOGIC ASSY. TANDY 1000 TL

QTY.	DESCRIPTION	DESIGNATOR	VENDOR	PART NUMBER
50	CAP 0.1 MFD 50V +80/-20% Z5U MONO. AX.	C1,16,35,53, 62,66,72,73, 74,82,87,88, 89,91,93,97, 99,101,103, 107,108,109, 110,111,113, 114,116,117, 126,127,128, 129,130,140, 141,144,145, 154,155,157, 158,159,160, 161,165,166, 175,177,178, 179	GENERIC	8374104
24	CAP .33 MFD 50V +80/-20% Z5U MONO. AX.	C118,119,120, 121,122,123, 124,125,132, 133,134,135, 136,137,138, 139,146,147, 148,149,150, 151,152,153	"	8374334
3	CAP .47 MFD 50V CD	C64,182,183	"	8384475
13	CAP RFI TBD	C2,36,37,81, 92,100,102, 104,105,142, 143,164,112	"	
3	CAP 3.3 MFD	C168,170,172	"	8335322
9	CAP 10 MFD 16V ELEC.RAD. 20%	C60,61,63,68, 79,80,173,174, 181	"	8326101
8	CAP 22 MFD 16V ELEC.RAD. 20%	C54,55,84,85, 86,115,156,162	"	8326221
1	CAP 47 MFD 16V ELEC.RAD. 20%	C71	"	8326474
1	CAP 100 MFD 16V ELE.RAD.	C167	"	8327101
5	EMI FILTER .022 UF W/FER. BEAD	CF1,3,4,5,6	MURATA #DST310 55D-223S	8418013
1	EMI CAP. .022 (NO BEAD)	CF2	MURATA #DS310 55D-223S	8418014
8	FERRITE BEAD	FB2-5,7-10	FARRITE #2743002121	8419013
4	FERRITE BEAD (801)	FB1,6,11,12	FARRITE #2643000801	8419098

ELECTRICAL BILL OF MATERIAL - TANDY 1000 TL

MAIN LOGIC ASSY. TANDY 1000 TL

QTY.	DESCRIPTION	DESIGNATOR	VENDOR	PART NUMBER
1	FUSE - 1 AMP (CANADA ONLY)	F1		8749045
2	DIODE 1N4148	CR1,2		8150148
2	REGULATOR 78L05	VR2,3	MOTOROLA MC78L05 FAIRCHILD uA78L05 TEXAS INST. uA78L05C	8052805
1	REGULATOR 79M0A5CT	VR1	MOTOROLA MC79M05CT FAIRCHILD uA7905 TEXAS INST. uA79M05CKC	8190005
1	DUAL OSC. 28.63636/32.514 MHZ	Y3	DIAWA,MF	8409076
1	DUAL OSC. 24/16 MHZ	Y2	DIAWA,MF	8409075
1	CRYSTAL 32.768 KHZ	Y1		8404033
1	IC 7416	U6	GENERIC	8000016
1	IC 74LS04	U16	"	8020004
1	IC 74LS32	U29	"	8020032
1	IC 74LS74	U26	■	8020074
1	IC 74LS244	U8	MOTOROLA	8020244
2	IC 74ALS244	U9,U25	GENERIC	8025244
1	IC 74ALS245	U40	"	8025245
3	IC 74ALS573	U27,30,43	"	8025573
1	IC 74F08	U20	"	8015008
1	IC 74HCT00*	U18	"	8026000
1	IC 74HCT14*	U14	"	8026014
1	IC 74HCT32*	U41	"	8026032
1	IC 74HCT244*	U53	"	8026244
2	IC 74HCT273*	U21,70	"	8026273
1	IC LM339	U7	"	8050339
1	IC LM386	U71	"	8050386
1	IC MC1458	U72		8051458
1	IC MC1458S	U10		8052458
1	IC MC1488	U2	MOTOROLA	8050188
2	IC MC1489	U4,5	MOTOROLA	8050189
1	IC TANVID 2 100 PIN QFP	U19	NCR	X07900100

* CAN SUBSTITUTE LS FOR ALL HCT PARTS

MAIN LOGIC TANDY 1000 TL SUB ASSY.

QTY.	DESCRIPTION	DESIGNATOR	VENDOR	PART NUMBER
1	CPU TANDY 1000 TL SUB ASSY. (REV. A)			8859022
2	JUMPER PLUG	E2-E3,E6-E7		8519098
1	IC 80286 8 MHZ CPU C,E STEP	U31	INTEL, AMD	8041286
1	IC DATA BUFFER *	U61	MOTOROLA, FUJITSU	8079022
1	IC CPU CONTROLLER (NT. BLUE)	U17	VLSI, RCA	8040810
1	IC DRAM/DMA CONTROLLER	U22	MOTOROLA, RCA	8040142
1	IC PSSJ	U11	NCR	8079021
1	IC KFIT	U13	NCR	8079019
1	IC 8237A-5	U28	INTEL, AMD, NEC	8040237
1	IC 8272A FDC CONT. *	U23	INTEL, ROCKWELL, ZILOG	8040272
1	IC FDC SUPPORT	U15	MOTOROLA, NCR	8041401
1	IC DS1215 REAL TIME CLK	U3	DALLAS SEMI.	8079023
1	IC EEPROM 64 X 16 9346	U12	NATIONAL, HYUNDAI, AMI/GOULD	8040346
16	IC 256K X 1 DRAM 150 NS	U45-52,62-69	FUJITSU, HITACHI, NEC, TI, SAMSUNG, MICRON	8049008
4	IC 64K X 4 DRAM 120 NS	U32-35	FUJITSU, HITACHI, NEC, TI, SAMSUNG, SHARP, MOTOROLA	8040464
1	IC 256K X 8 BIOS ROM EVEN 200NS (1)	U55	SHARP	8076323
1	IC 256K X 8 BIOS ROM ODD 200NS (1)	U57	SHARP	8075323
1	IC 256K X 8 BIOS ROM EVEN 200NS (1)	U54	HITACHI	8079025
1	IC 256K X 8 BIOS ROM ODD 200NS (1)	U56	HITACHI	8079026
1	IC 16K X 8 CHARA.GEN. ROM 200NS	U24	HITACHI, SHARP, NEC	8079027
1	IC PLS173 I/O DECODE	U42	SIGNETICS	8077173
1	IC PLS173 ROM CNTL.	U44	SIGNETICS	8076173

ELECTRICAL BILL OF MATERIAL - PROJECT 878

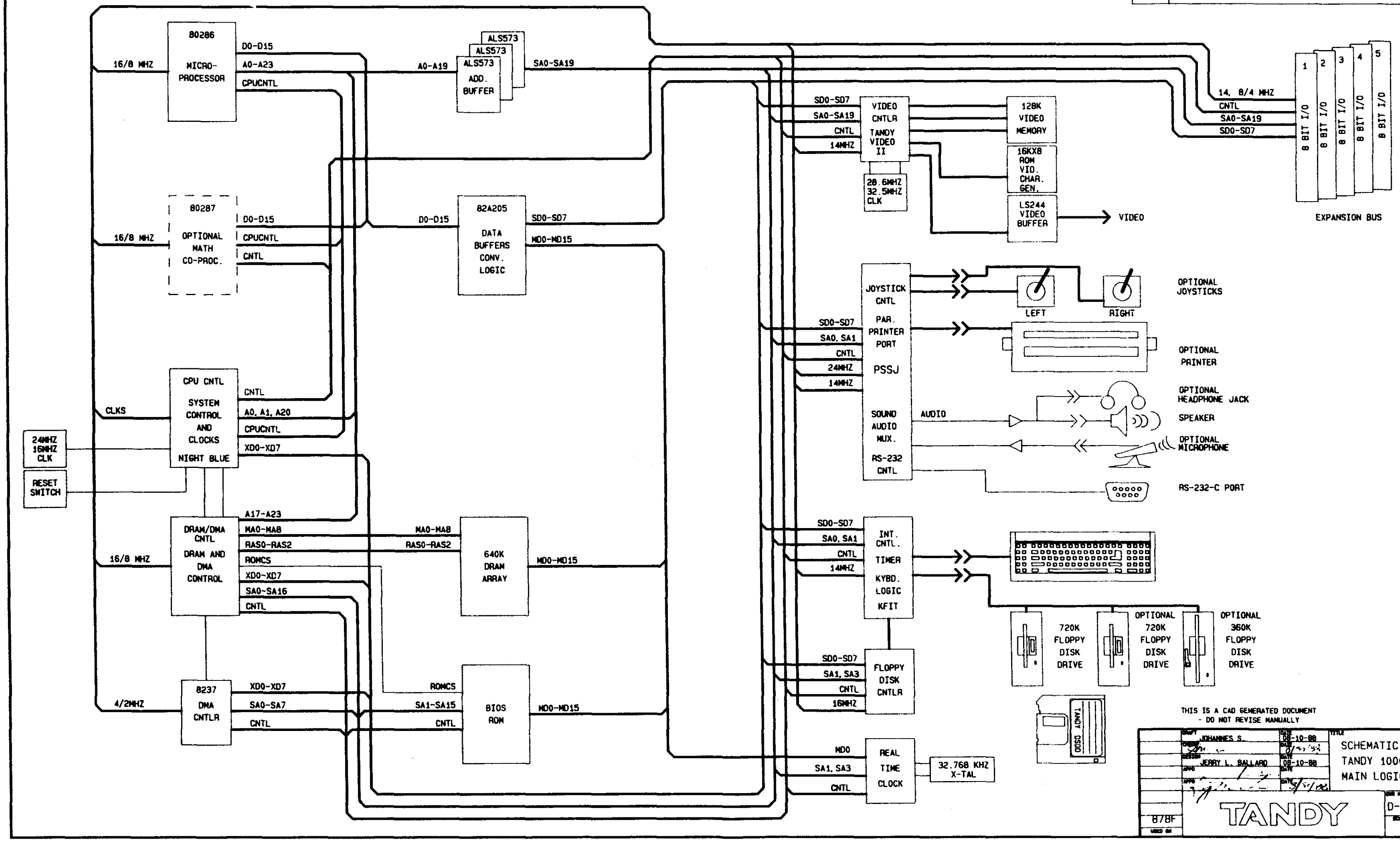
MAIN LOGIC ASSY. TANDY 1000 TL

QTY.	DESCRIPTION	DESIGNATOR	VENDOR	PART NUMBER
*THE FOLLOWING PARTS MAY BE SUBSTITUTED IF NECESSARY:				
1	IC UPD765	U23	NEC	8041765
1	IC 82A205 DATA BUFFER	U61	CHIPS	8075205

MAIN LOGIC TANDY 1000 TL SUB ASSY.

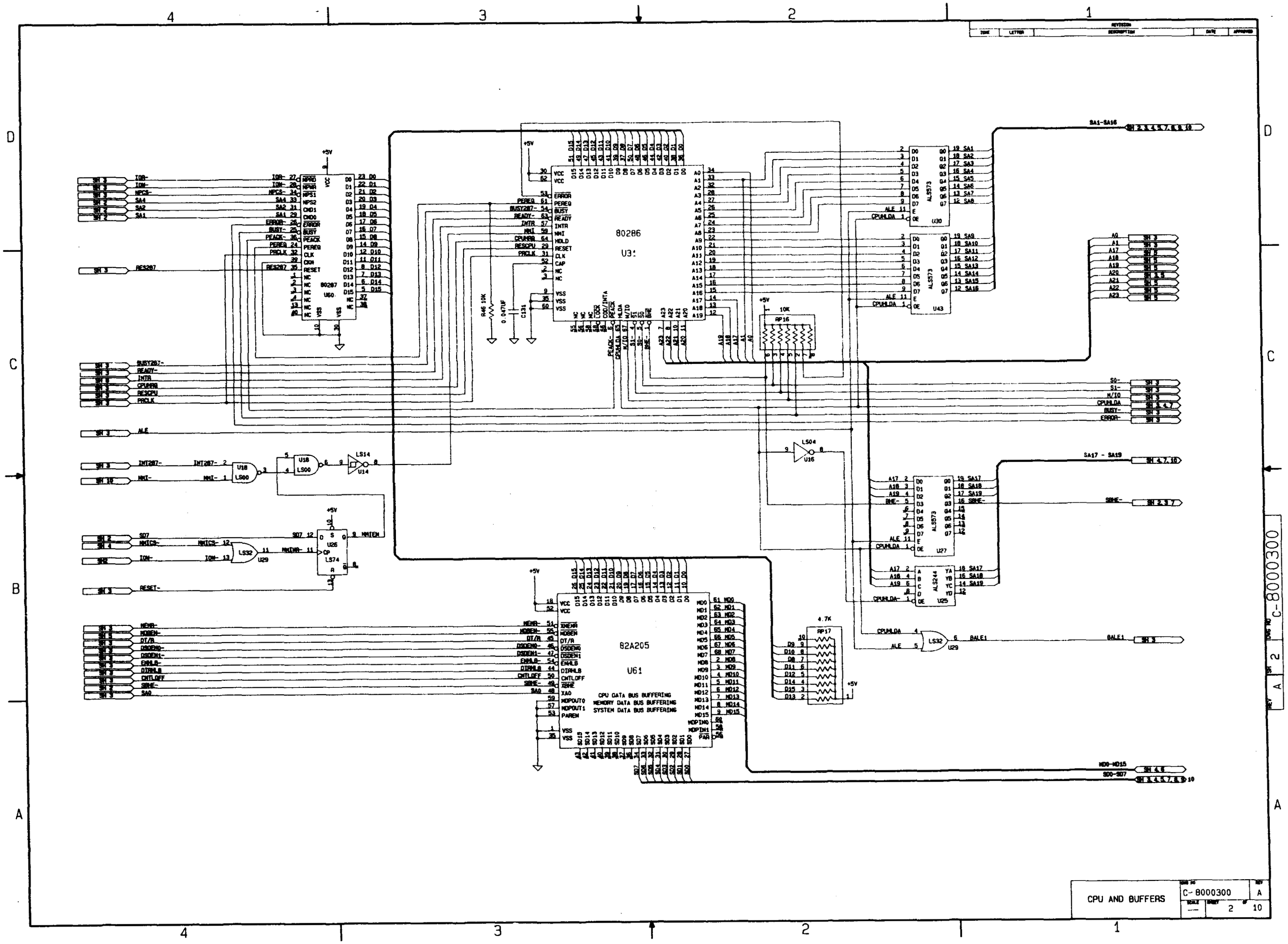
QTY.	DESCRIPTION	DESIGNATOR	VENDOR	PART NUMBER
OPTION KITS:				
1	IC 80287-6 OR 8	U60	INTEL	
4	IC 64K X 4 DRAM 150 OR 120NS	U36-39	FUJITSU, HITACHI, NEC, TI, SAMSUNG, MICRON, SHARP, MOTOROLA	8040464

REVISION			
LETTER	DESCRIPTION	DATE	APPROVED
A	RELEASED FOR PRODUCTION	08-16-88	
A-1	CHANGE PER TCO # 13671	08-24-88	



THIS IS A CAD GENERATED DOCUMENT
- DO NOT REVISE MANUALLY

	DRAFT JOHANNES S.	DATE 08-10-88	TITLE SCHEMATIC - TANDY 1000TL MAIN LOGIC BOARD	
	CHECK	DATE		
	DESIGN JERRY L. BALLARD	DATE 08-10-88		
	APPD	DATE		
	APPD	DATE		
878F	TANDY		DWG NO D- 8000300	REV A
USED ON			SCALE	SHEET 1 OF 10

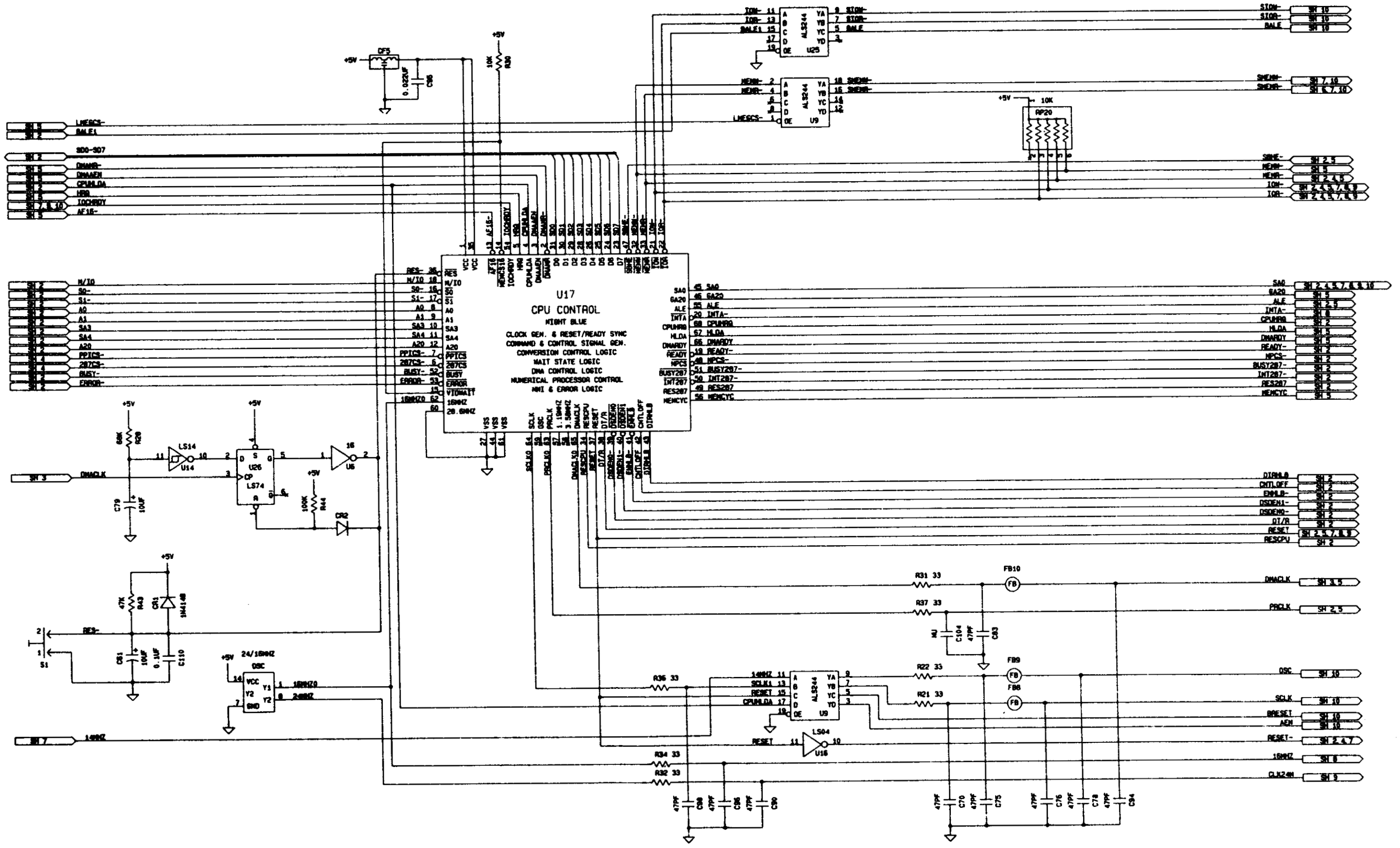


CPU AND BUFFERS

C-8000300

A

2 10



CONTROL AND CLOCKS

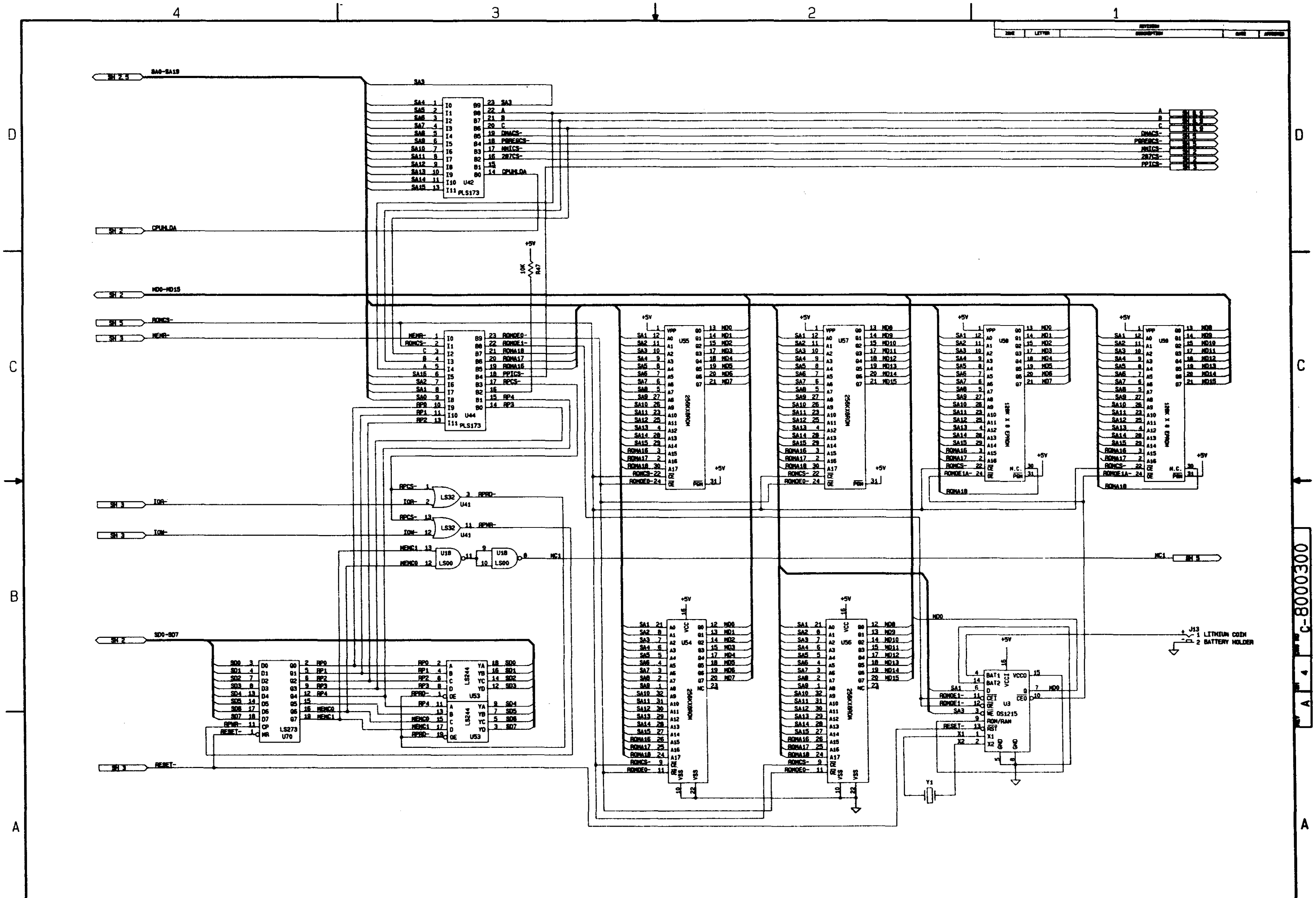
C-8000300

A

3

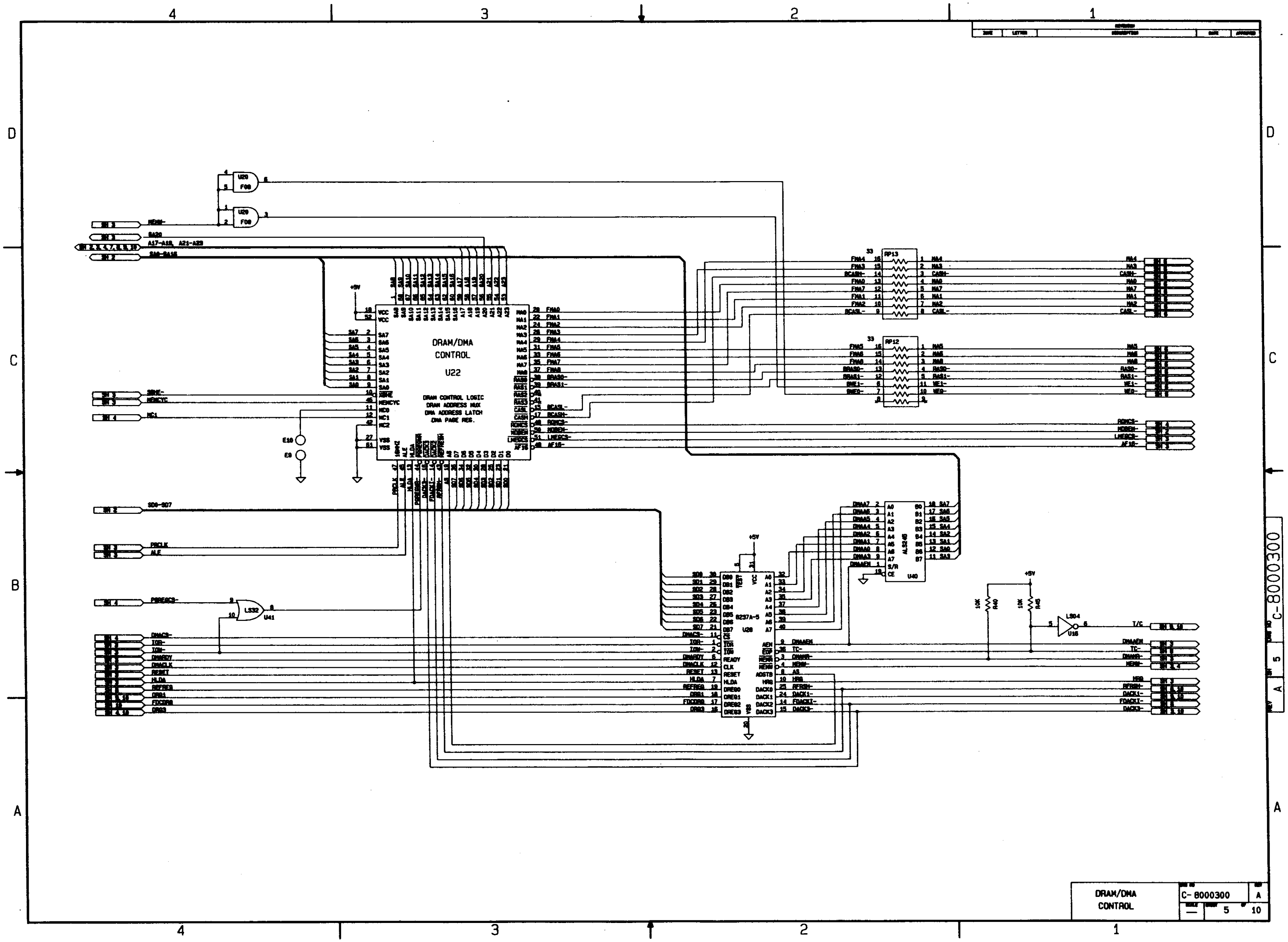
10

C-8000300 3 A



I/O DECODE, ROM,
AND REAL TIME CLOCK

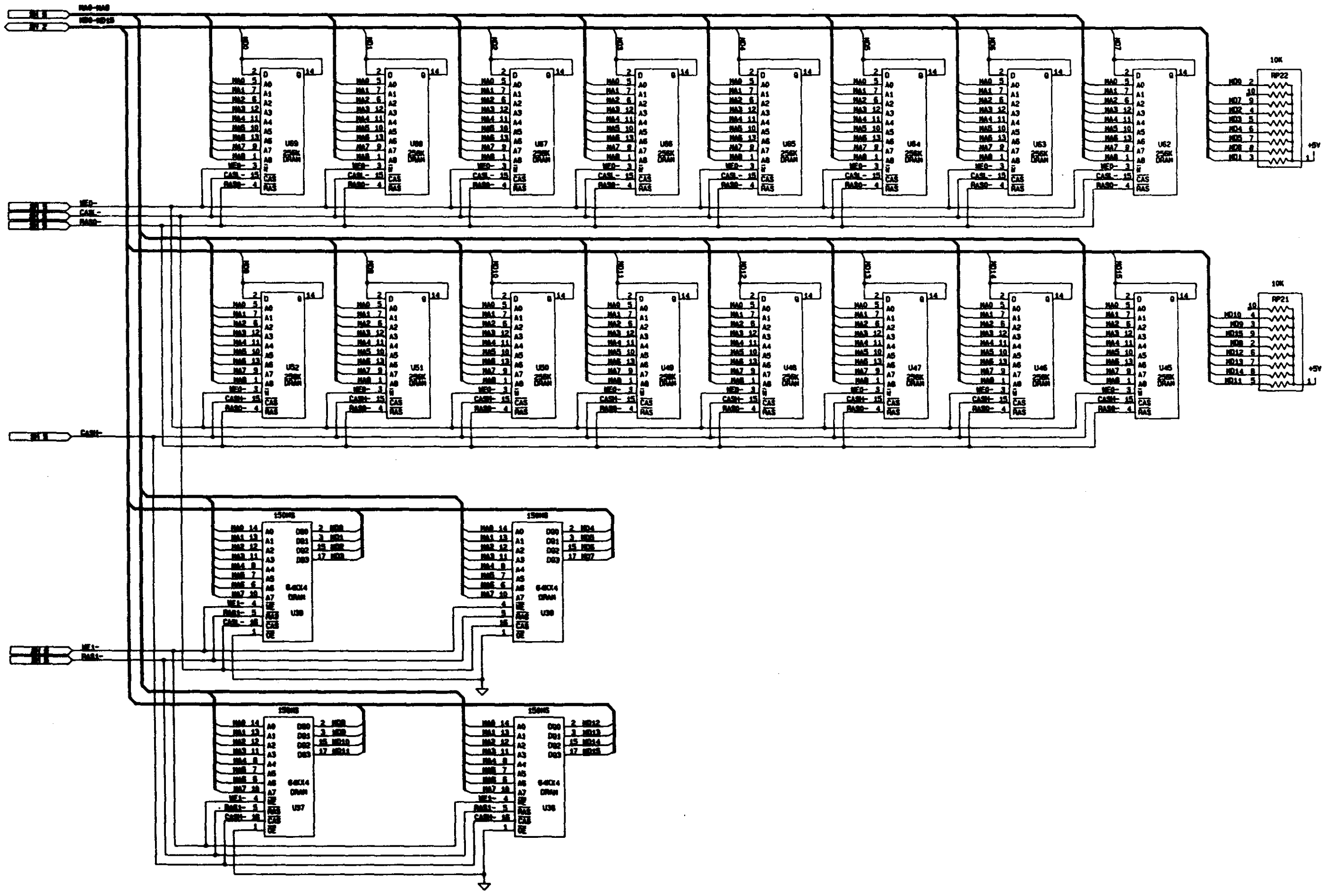
C- 8000300	A
— 4	10



DRAM/DMA CONTROL

C-8000300 A

5 10

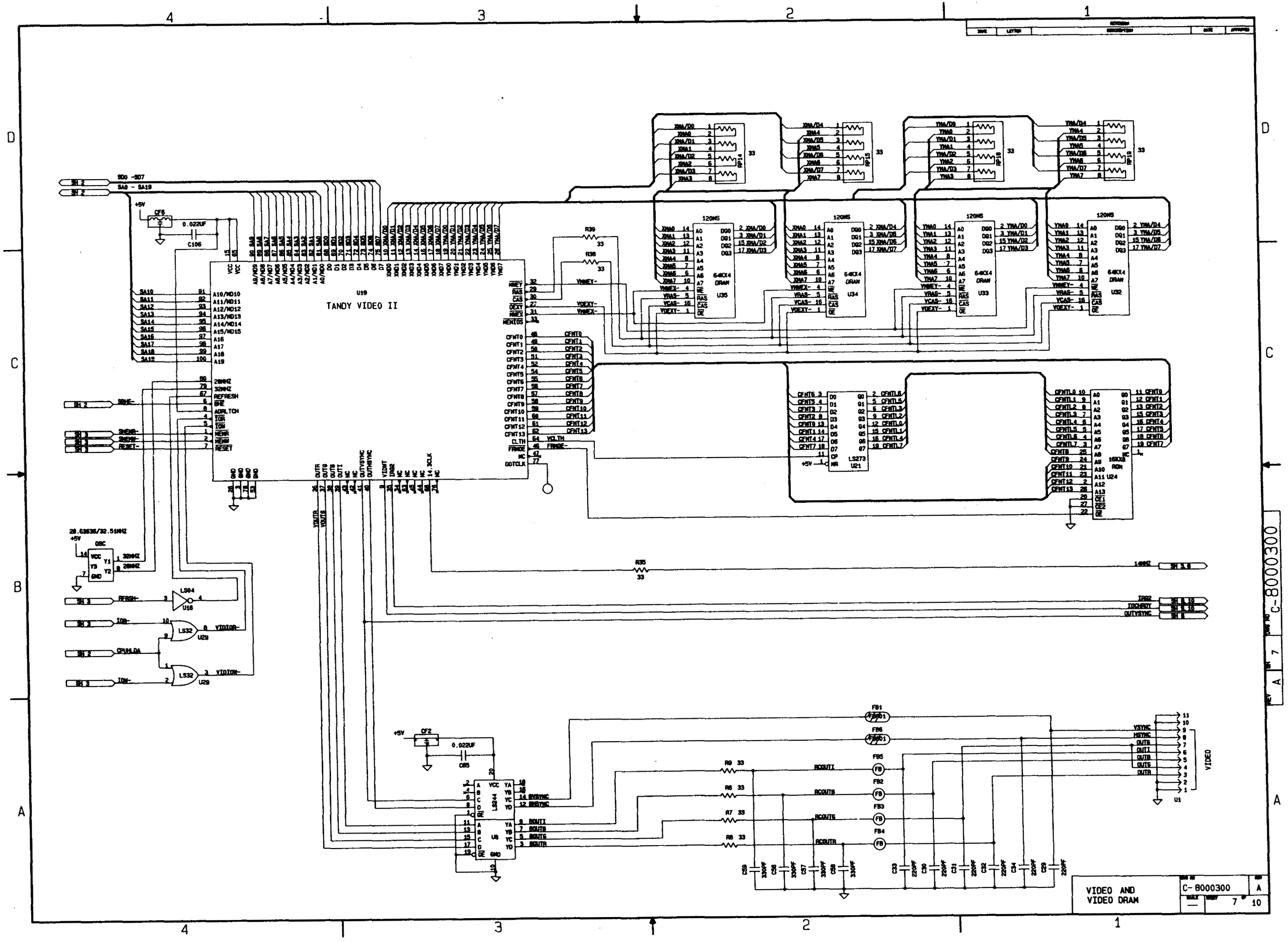


DRAM ARRAY

C- 8000300

A

6 10

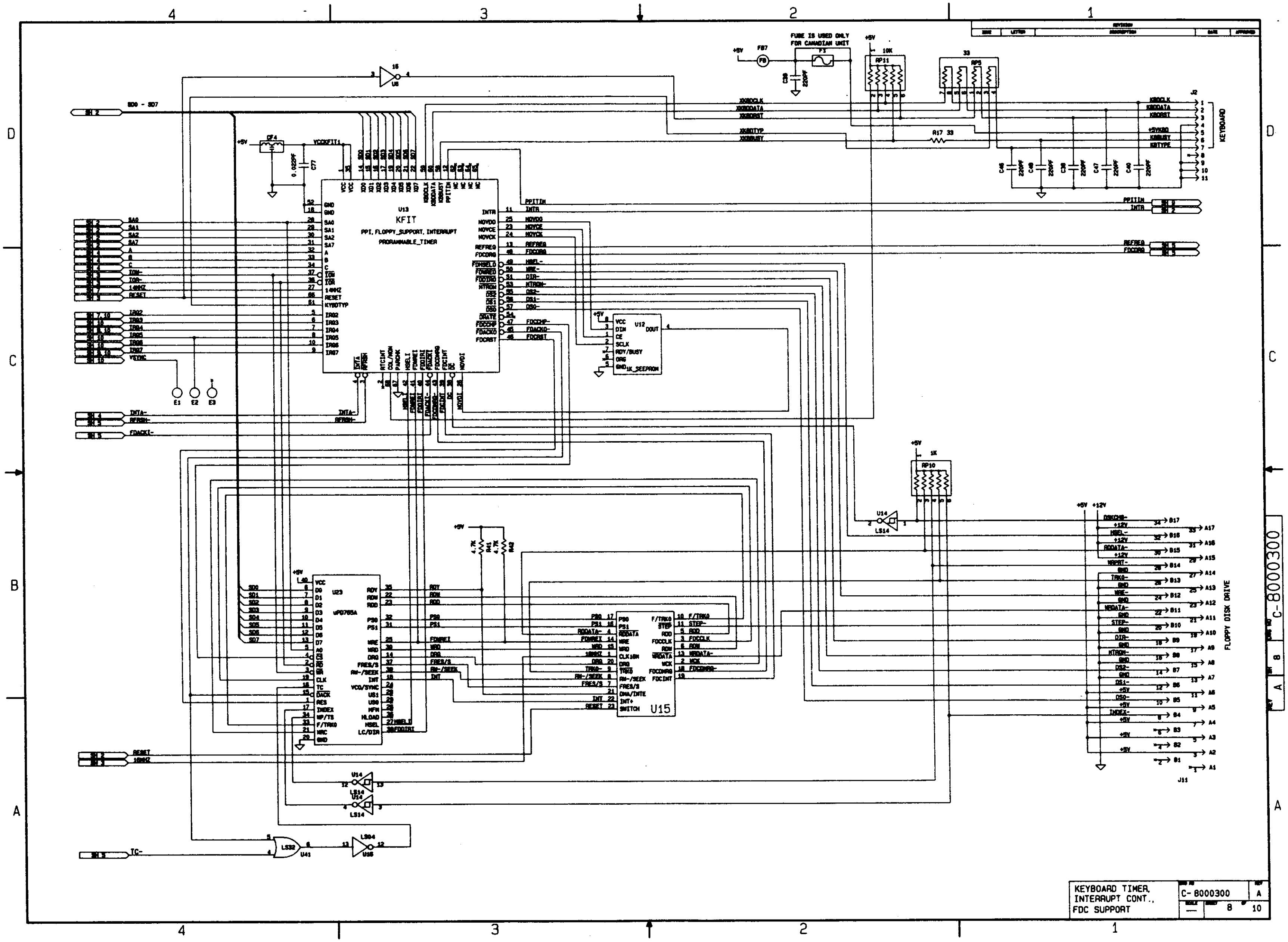


VIDEO AND
VIDEO DRAM

C-8000300

A

7 OF 10



KEYBOARD TIMER,
INTERRUPT CONT.,
FDC SUPPORT

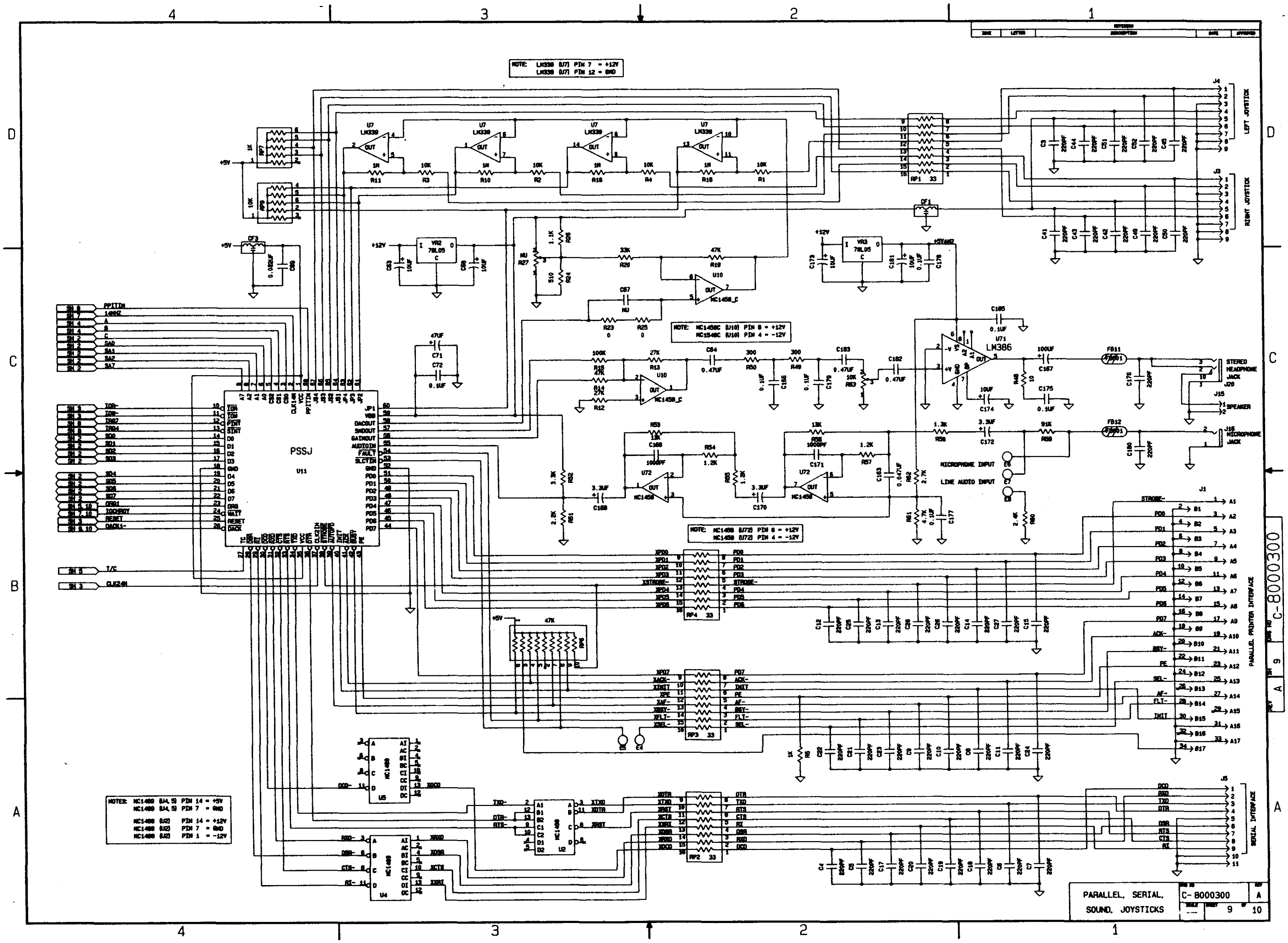
C-8000300

A

—

8

10



NOTE: LM339 (U7) PIN 7 = +12V
LM339 (U7) PIN 12 = GND

NOTE: MC1458C (U10) PIN 8 = +12V
MC1458C (U10) PIN 4 = -12V

NOTE: MC1458 (U72) PIN 8 = +12V
MC1458 (U72) PIN 4 = -12V

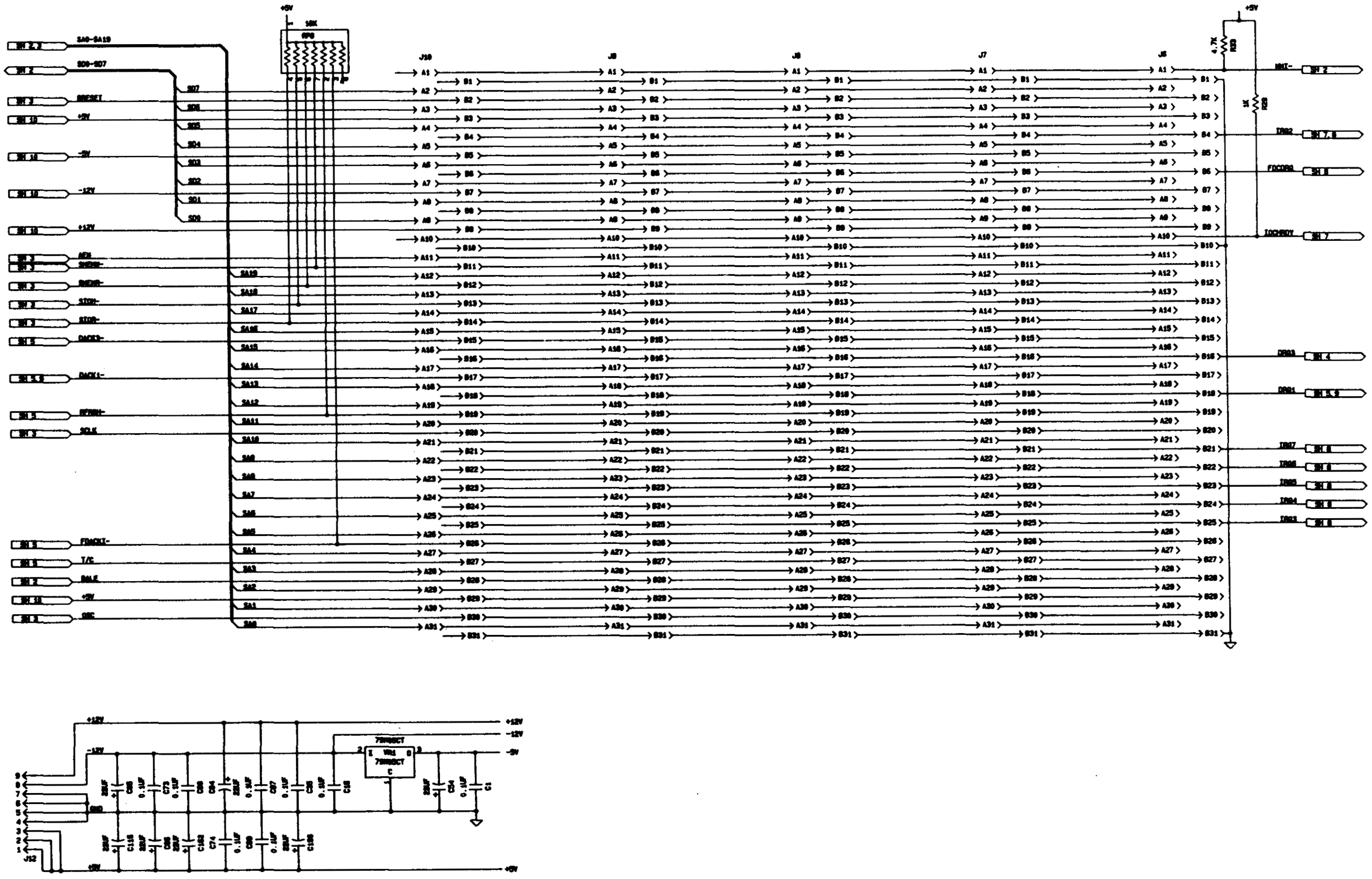
NOTES: MC1489 (U4, 5) PIN 14 = +5V
MC1489 (U4, 5) PIN 7 = GND

MC1488 (U2) PIN 14 = +12V
MC1488 (U2) PIN 7 = GND
MC1488 (U2) PIN 1 = -12V

PARALLEL, SERIAL,
SOUND, JOYSTICKS

C-8000300 A

--- 9 of 10



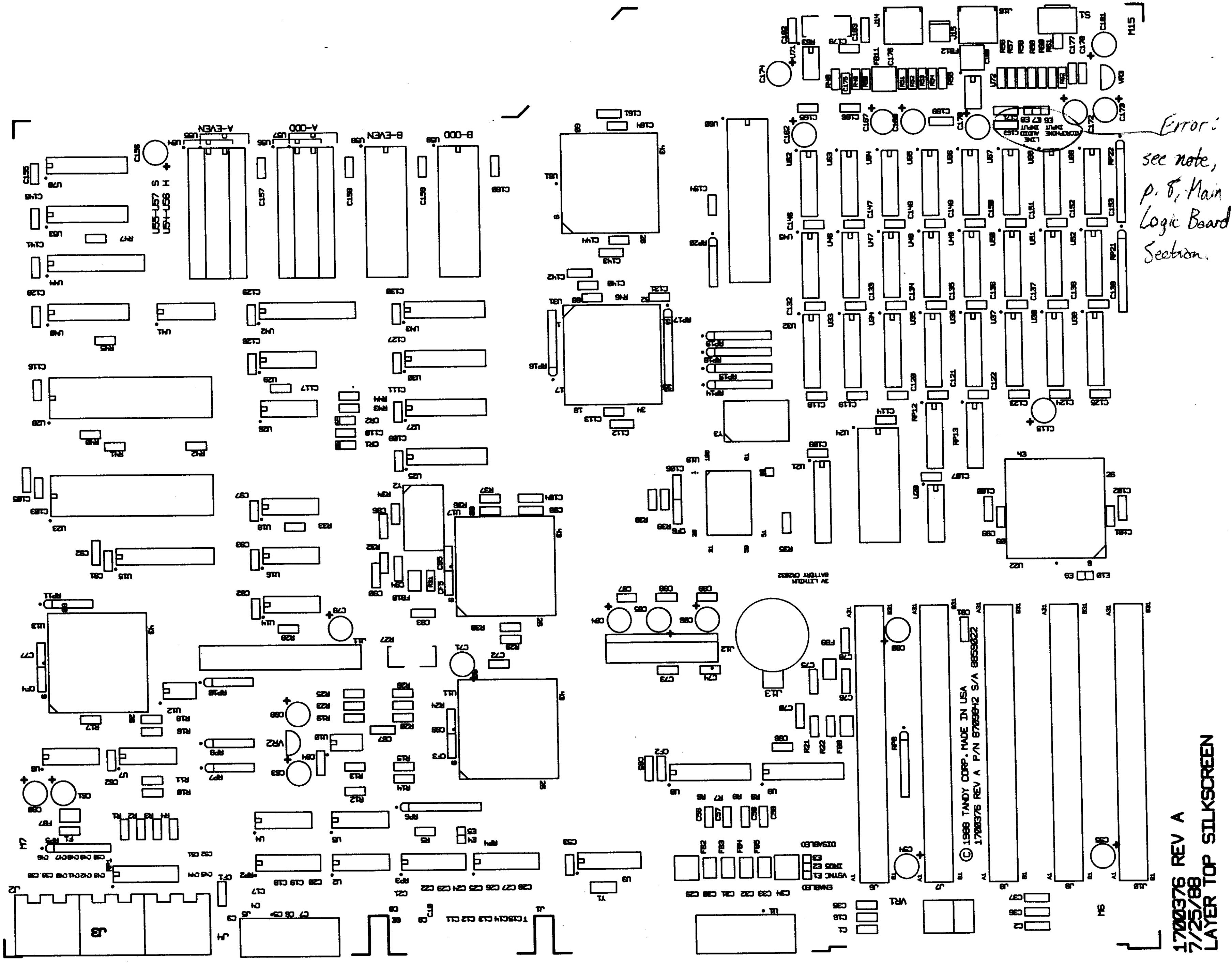
EXPANSION BUS,
POWER CONNECTOR

C- 8000300

A

10

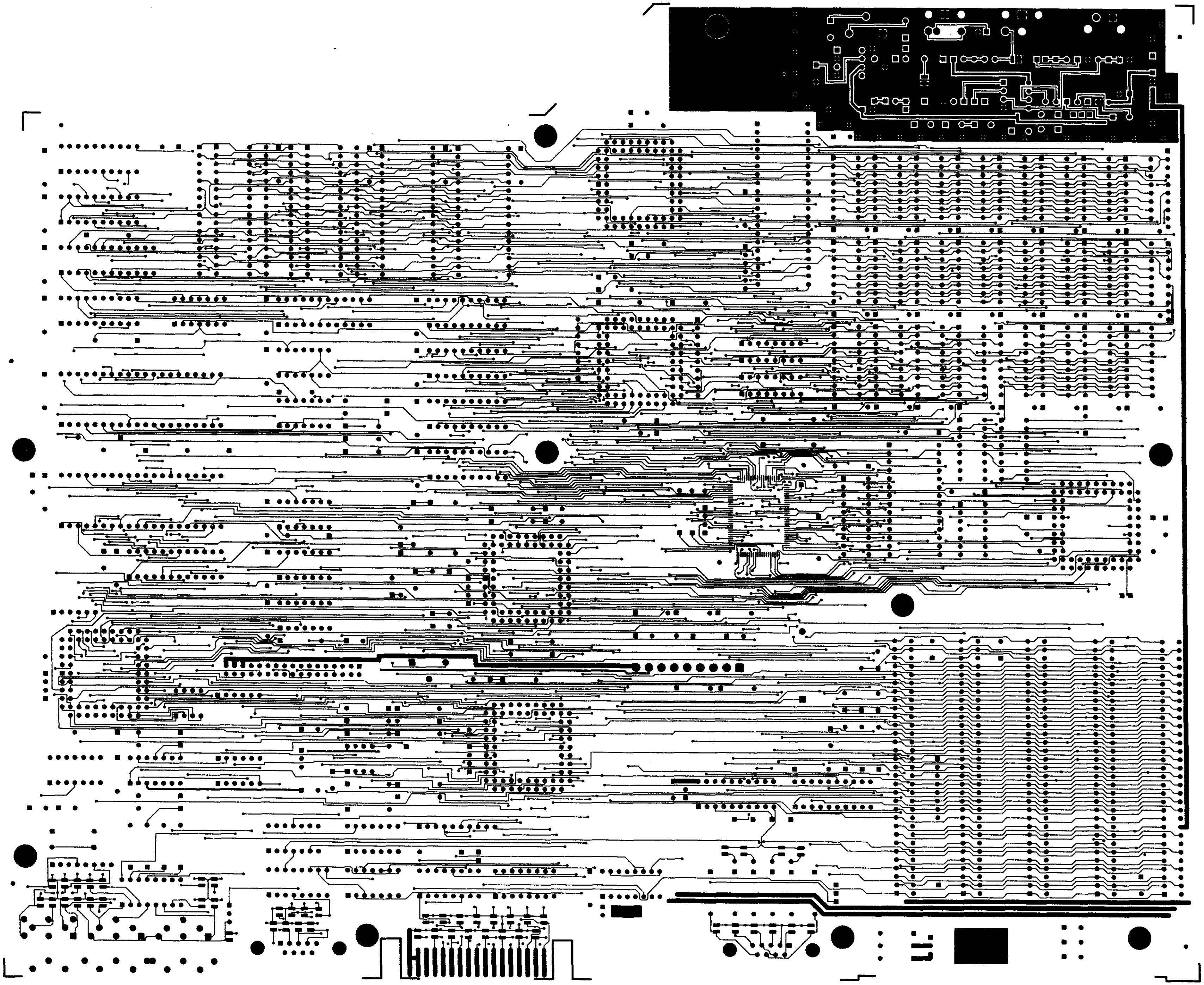
10



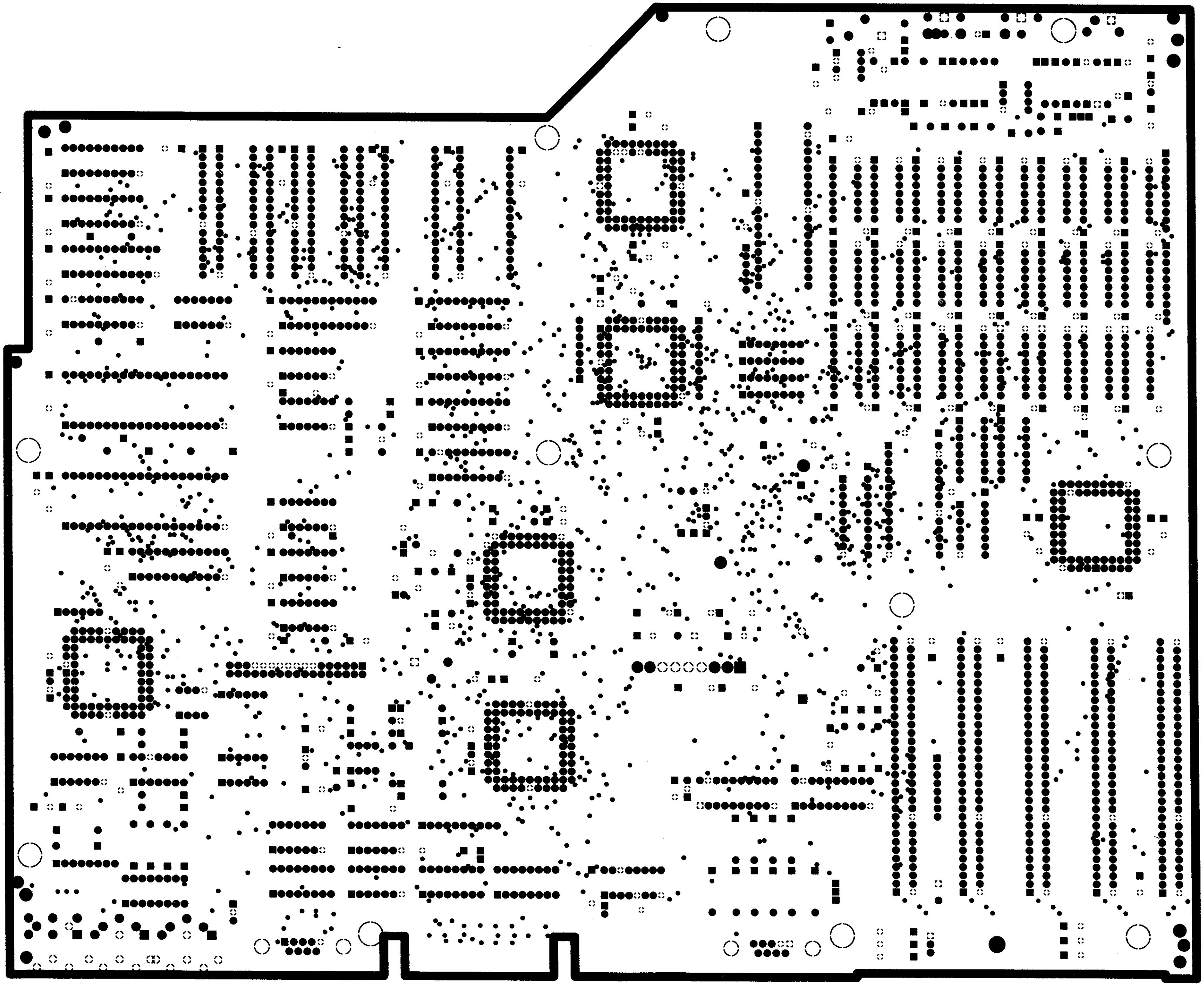
© 1988 TANDY CORP. MADE IN USA
1700376 REV A P/N 8709842 S/A 8859022

1700376 REV A
7/25/88
LAYER TOP SILKSCREEN

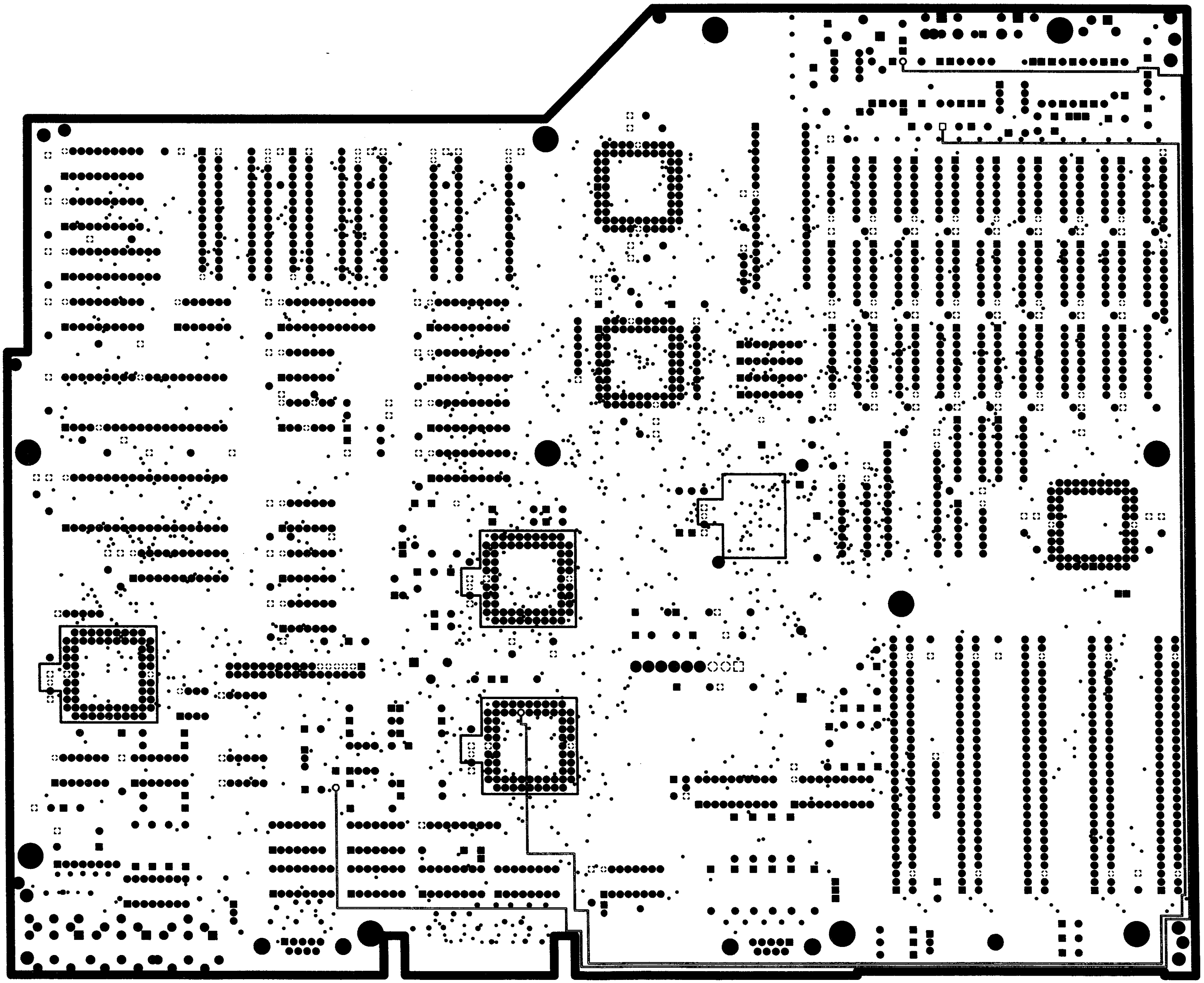
Error: see note, p. 8, Main Logic Board Section.

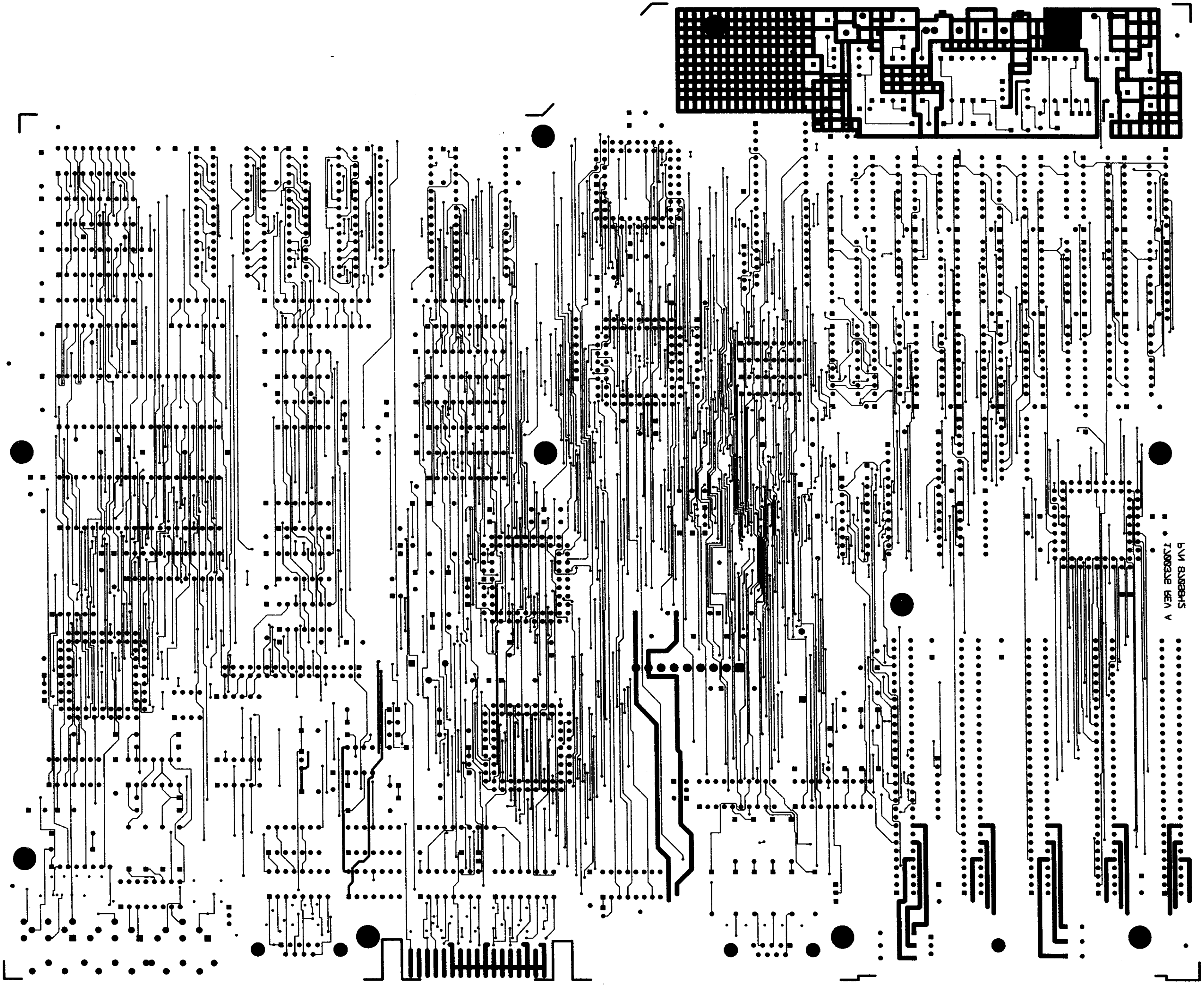


1700376 REV A
7/25/88
LAYER 1 TOP SIDE



1700376 REV A
7/25/88
LAYER 2 GND PLANE


1700376 REV A
7/25/88
LAYER 3 +5V PLANE



1700376 REV A
7/25/88
LAYER 4 BOTTOM SIDE

Devices

80286
High Performance Microprocessor with Memory Management and Protection
(80286-12, 80286-10, 80286-8)

- **High Performance Processor (Up to six times 8086)**
- **Large Address Space:**
  - **— 16 Megabytes Physical**
  - **— 1 Gigabyte Virtual per Task**
- **Integrated Memory Management, Four-Level Memory Protection and Support for Virtual Memory and Operating Systems**
- **High Bandwidth Bus Interface (12.5 Megabyte/Sec)**
- **Industry Standard O.S. Support:**
  - **— IRMX®**
  - **— XENIX***
  - **— UNIX***
  - **— MS-DOS***
- **Optional Processor Extension:**
  - **— 80287 High Performance 80-bit Numeric Data Processor**
- **Two 8086 Upward Compatible Operating Modes:**
  - **— 8086 Real Address Mode**
  - **— Protected Virtual Address Mode**
- **Range of Clock Rates**
  - **— 12.5 MHz for 80286-12**
  - **— 10 MHz for 80286-10**
  - **— 8 MHz for 80826-8**
  - **— 6 MHz for 80286-6**
- **Complete System Development Support:**
  - **— Development Software: Assembler, PL/M, Pascal, FORTRAN, and System Utilities**
  - **— In-Circuit-Emulator (ICE™-286)**
- **Available in 68 Pin Ceramic LCC (Leadless Chip Carrier), PGA (Pin Grid Array), and PLCC (Plastic Leaded Chip Carrier) Packages**

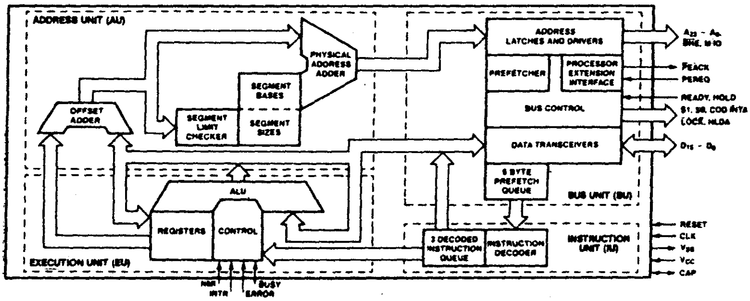
  (See Packaging Spec., Order #231369)

The 80286 is an advanced, high-performance microprocessor with specially optimized capabilities for multiple user and multi-tasking systems. The 80286 has built-in memory protection that supports operating system and task isolation as well as program and data privacy within tasks. A 12 MHz 80286 provides six times or more throughput than the standard 5 MHz 8086. The 80286 includes memory management capabilities that map 2^{30} (one gigabyte) of virtual address space per task into 2^{24} bytes (16 megabytes) of physical memory.

The 80286 is upward compatible with 8086 and 88 software. Using 8086 real address mode, the 80286 is object code compatible with existing 8086, 88 software. In protected virtual address mode, the 80286 is source code compatible with 8086, 88 software and may require upgrading to use virtual addresses supported by the 80286's integrated memory management and protection mechanism. Both modes operate at full 80286 performance and execute a superset of the 8086 and 88 instructions.

The 80286 provides special operations to support the efficient implementation and execution of operating systems. For example, one instruction can end execution of one task, save its state, switch to a new task, load its state, and start execution of the new task. The 80286 also supports virtual memory systems by providing a segment-not-present exception and restartable instructions.

*XENIX and MS-DOS are trademarks of Microsoft Corp.
*UNIX is a trademark of Bell Labs or AT&T

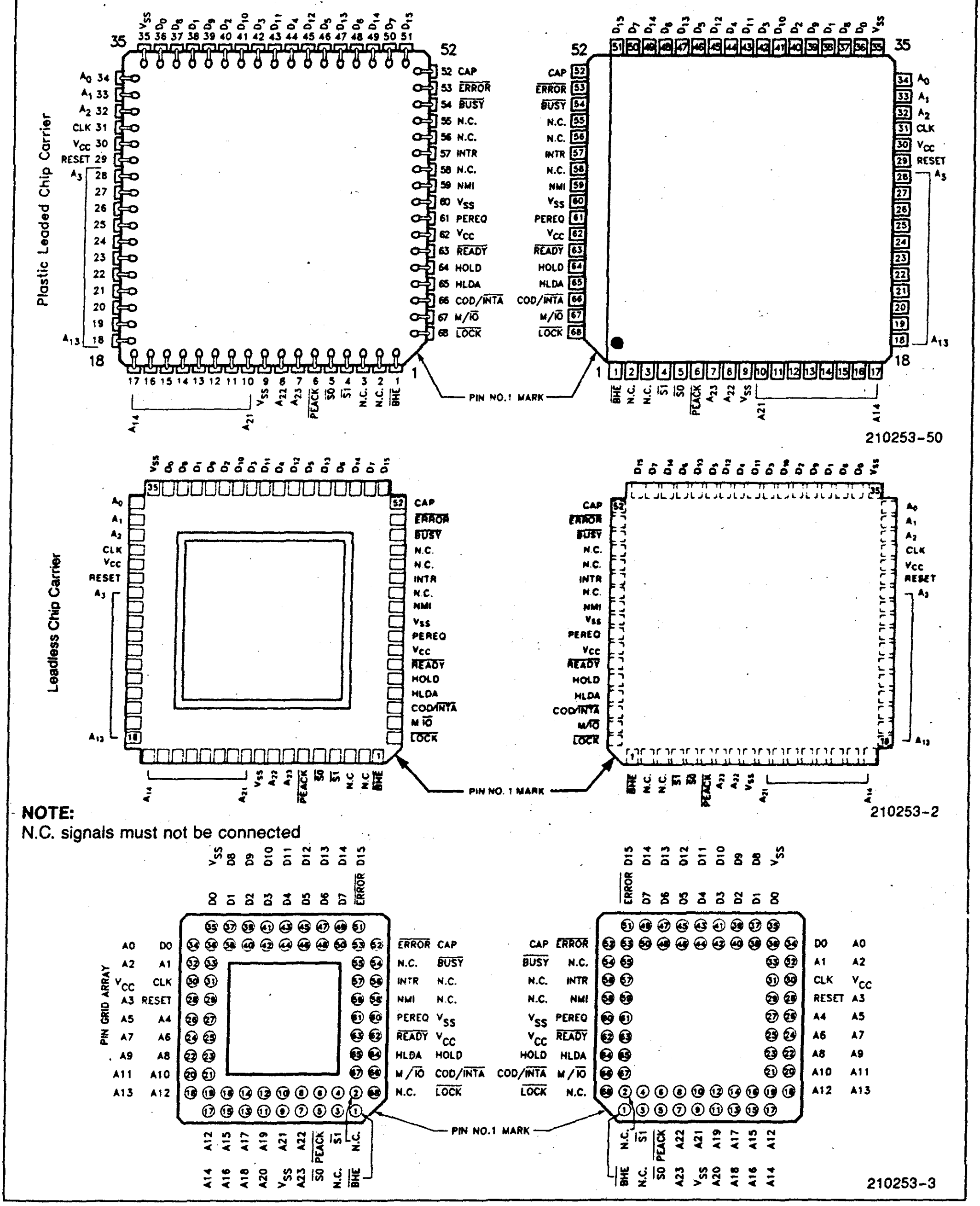


210253-1

Figure 1. 80286 Internal Block Diagram

Component Pad Views—*As viewed from underside of component when mounted on the board.*

P.C. Board Views—*As viewed from the component side of the P.C. board.*



NOTE:
N.C. signals must not be connected

Figure 2. 80286 Pin Configuration

Table 1. Pin Description

The following pin function descriptions are for the 80286 microprocessor:

<table>
<tr><th>Symbol</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>CLK</td><td>I</td><td><b>SYSTEM CLOCK</b> provides the fundamental timing for 80286 systems. It is divided by two inside the 80286 to generate the processor clock. The internal divide-by-two circuitry can be synchronized to an external clock generator by a LOW to HIGH transition on the RESET input.</td></tr>
<tr><td>$D_{15}-D_0$</td><td>I/O</td><td><b>DATA BUS</b> inputs data during memory, I/O, and interrupt acknowledge read cycles; outputs data during memory and I/O write cycles. The data bus is active HIGH and floats to 3-state OFF during bus hold acknowledge.</td></tr>
<tr><td>$A_{23}-A_0$</td><td>O</td><td><b>ADDRESS BUS</b> outputs physical memory and I/O port addresses. A0 is LOW when data is to be transferred on pins D_{7-0}. $A_{23}-A_{16}$ are LOW during I/O transfers. The address bus is active HIGH and floats to 3-state OFF during bus hold acknowledge.</td></tr>
<tr><td>$\overline{BHE}$</td><td>O</td><td><b>BUS HIGH ENABLE</b> indicates transfer or data on the upper byte of the data bus. D_{15-8}. Eight-bit oriented devices assigned to the upper byte of the data bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is active LOW and floats to 3-state OFF during bus hold acknowledge.
<table>
<tr><th colspan="3">$\overline{BHE}$ and A0 Encodings</th></tr>
<tr><th>$\overline{BHE}$ Value</th><th>A0 Value</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Word transfer</td></tr>
<tr><td>0</td><td>1</td><td>Byte transfer on upper half of data bus ($D_{15}-D_8$)</td></tr>
<tr><td>1</td><td>0</td><td>Byte transfer on lower half of data bus (D_{7-0})</td></tr>
<tr><td>1</td><td>1</td><td>Will never occur</td></tr>
</table></td></tr>
<tr><td>$\overline{S1}$, $\overline{S0}$</td><td>O</td><td><b>BUS CYCLE STATUS</b> indicates initiation of a bus cycle and, along with M/$\overline{IO}$ and COD/$\overline{INTA}$, defines the type of bus cycle. The bus is in a T_s state whenever one or both are LOW, $\overline{S1}$ and $\overline{S0}$ are active LOW and float to 3-state OFF during bus hold acknowledge.
<table>
<tr><th colspan="5">80286 Bus Cycle Status Definition</th></tr>
<tr><th>COD/$\overline{INTA}$</th><th>M/$\overline{IO}$</th><th>$\overline{S1}$</th><th>$\overline{S0}$</th><th>Bus Cycle Initiated</th></tr>
<tr><td>0 (LOW)</td><td>0</td><td>0</td><td>0</td><td>Interrupt acknowledge</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Will not occur</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Will not occur</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>None; not a status cycle</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>IF A1 = 1 then halt; else shutdown</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>Memory data read</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>Memory data write</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>None; not a status cycle</td></tr>
<tr><td>1 (HIGH)</td><td>0</td><td>0</td><td>0</td><td>Will not occur</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>I/O read</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>I/O write</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>None; not a status cycle</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Will not occur</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>Memory instruction read</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>Will not occur</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>None; not a status cycle</td></tr>
</table></td></tr>
<tr><td>M/$\overline{IO}$</td><td>O</td><td><b>MEMORY I/O SELECT</b> distinguishes memory access from I/O access. If HIGH during T_s, a memory cycle or a halt/shutdown cycle is in progress. If LOW, an I/O cycle or an interrupt acknowledge cycle is in progress. M/$\overline{IO}$ floats to 3-state OFF during bus hold acknowledge.</td></tr>
<tr><td>COD/$\overline{INTA}$</td><td>O</td><td><b>CODE/INTERRUPT ACKNOWLEDGE</b> distinguishes instruction fetch cycles from memory data read cycles. Also distinguishes interrupt acknowledge cycles from I/O cycles. COD/$\overline{INTA}$ floats to 3-state OFF during bus hold acknowledge. Its timing is the same as M/$\overline{IO}$.</td></tr>
<tr><td>$\overline{LOCK}$</td><td>O</td><td><b>BUS LOCK</b> indicates that other system bus masters are not to gain control of the system bus for the current and the following bus cycle. The $\overline{LOCK}$ signal may be activated explicitly by the "LOCK" instruction prefix or automatically by 80286 hardware during memory XCHG instructions, interrupt acknowledge, or descriptor table access. $\overline{LOCK}$ is active LOW and floats to 3-state OFF during bus hold acknowledge.</td></tr>
<tr><td>$\overline{READY}$</td><td>I</td><td><b>BUS READY</b> terminates a bus cycle. Bus cycles are extended without limit until terminated by $\overline{READY}$ LOW. $\overline{READY}$ is an active LOW synchronous input requiring setup and hold times relative to the system clock be met for correct operation. $\overline{READY}$ is ignored during bus hold acknowledge.</td></tr>
</table>

Table 1. Pin Description *(Continued)*

Symbol	Type	Name and Function
HOLD HLDA	I O	**BUS HOLD REQUEST AND HOLD ACKNOWLEDGE** control ownership of the 80286 local bus. The HOLD input allows another local bus master to request control of the local bus. When control is granted, the 80286 will float its bus drivers to 3-state OFF and then activate HLDA, thus entering the bus hold acknowledge condition. The local bus will remain granted to the requesting master until HOLD becomes inactive which results in the 80286 deactivating HLDA and regaining control of the local bus. This terminates the bus hold acknowledge condition. HOLD may be asynchronous to the system clock. These signals are active HIGH.
INTR	I	**INTERRUPT REQUEST** requests the 80286 to suspend its current program execution and service a pending external request. Interrupt requests are masked whenever the interrupt enable bit in the flag word is cleared. When the 80286 responds to an interrupt request, it performs two interrupt acknowledge bus cycles to read an 8-bit interrupt vector that identifies the source of the interrupt. To assure program interruption, INTR must remain active until the first interrupt acknowledge cycle is completed. INTR is sampled at the beginning of each processor cycle and must be active HIGH at least two processor cycles before the current instruction ends in order to interrupt before the next instruction. INTR is level sensitive, active HIGH, and may be asynchronous to the system clock.
NMI	I	**NON-MASKABLE INTERRUPT REQUEST** interrupts the 80286 with an internally supplied vector value of 2. No interrupt acknowledge cycles are performed. The interrupt enable bit in the 80286 flag word does not affect this input. The NMI input is active HIGH, may be asynchronous to the system clock, and is edge triggered after internal synchronization. For proper recognition, the input must have been previously LOW for at least four system clock cycles and remain HIGH for at least four system clock cycles.
PEREQ $\overline{\text{PEACK}}$	I O	**PROCESSOR EXTENSION OPERAND REQUEST AND ACKNOWLEDGE** extend the memory management and protection capabilities of the 80286 to processor extensions. The PEREQ input requests the 80286 to perform a data operand transfer for a processor extension. The $\overline{\text{PEACK}}$ output signals the processor extension when the requested operand is being transferred. PEREQ is active HIGH and floats to 3-state OFF during bus hold acknowledge. $\overline{\text{PEACK}}$ may be asynchronous to the system clock. $\overline{\text{PEACK}}$ is active LOW.
$\overline{\text{BUSY}}$ $\overline{\text{ERROR}}$	I I	**PROCESSOR EXTENSION BUSY AND ERROR** indicate the operating condition of a processor extension to the 80286. An active $\overline{\text{BUSY}}$ input stops 80286 program execution on WAIT and some ESC instructions until $\overline{\text{BUSY}}$ becomes inactive (HIGH). The 80286 may be interrupted while waiting for $\overline{\text{BUSY}}$ to become inactive. An active $\overline{\text{ERROR}}$ input causes the 80286 to perform a processor extension interrupt when executing WAIT or some ESC instructions. These inputs are active LOW and may be asynchronous to the system clock. These inputs have internal pull-up resistors.

Table 1. Pin Description (Continued)

<table>
<tr><th>Symbol</th><th>Type</th><th colspan="2">Name and Function</th></tr>
<tr><td rowspan="5">RESET</td><td rowspan="5">I</td><td colspan="2">**SYSTEM RESET** clears the internal logic of the 80286 and is active HIGH. The 80286 may be reinitialized at any time with a LOW to HIGH transition on RESET which remains active for more than 16 system clock cycles. During RESET active, the output pins of the 80286 enter the state shown below:</td></tr>
<tr><td colspan="2">**80286 Pin State During Reset**</td></tr>
<tr><td>**Pin Value**</td><td>**Pin Names**</td></tr>
<tr><td>1 (HIGH)
0 (LOW)
3-state OFF</td><td>$\overline{\text{S0}}$, $\overline{\text{S1}}$, $\overline{\text{PEACK}}$, A23–A0, $\overline{\text{BHE}}$, $\overline{\text{LOCK}}$
M/$\overline{\text{IO}}$, COD/$\overline{\text{INTA}}$, HLDA (Note 1)
D_{15}–D_0</td></tr>
<tr><td colspan="2">Operation of the 80286 begins after a HIGH to LOW transition on RESET. The HIGH to LOW transition of RESET must be synchronous to the system clock. Approximately 38 CLK cycles from the trailing edge of RESET are required by the 80286 for internal initialization before the first bus cycle, to fetch code from the power-on execution address, occurs.
A LOW to HIGH transition of RESET synchronous to the system clock will end a processor cycle at the second HIGH to LOW transition of the system clock. The LOW to HIGH transition of RESET may be asynchronous to the system clock; however, in this case it cannot be predetermined which phase of the processor clock will occur during the next system clock period. Synchronous LOW to HIGH transitions of RESET are required only for systems where the processor clock must be phase synchronous to another clock.</td></tr>
<tr><td>V_{SS}</td><td>I</td><td colspan="2">**SYSTEM GROUND:** 0 Volts.</td></tr>
<tr><td>V_{CC}</td><td>I</td><td colspan="2">**SYSTEM POWER:** +5 Volt Power Supply.</td></tr>
<tr><td>CAP</td><td>I</td><td colspan="2">**SUBSTRATE FILTER CAPACITOR:** a 0.047 μF ± 20% 12V capacitor must be connected between this pin and ground. This capacitor filters the output of the internal substrate bias generator. A maximum DC leakage current of 1 μA is allowed through the capacitor.
For correct operation of the 80286, the substrate bias generator must charge this capacitor to its operating voltage. The capacitor chargeup time is 5 milliseconds (max.) after V_{CC} and CLK reach their specified AC and DC parameters. RESET may be applied to prevent spurious activity by the CPU during this time. After this time, the 80286 processor clock can be synchronized to another clock by pulsing RESET LOW synchronous to the system clock.</td></tr>
</table>

NOTE:
1. HLDA is only Low if HOLD is inactive (Low).

FUNCTIONAL DESCRIPTION

Introduction

The 80286 is an advanced, high-performance microprocessor with specially optimized capabilities for multiple user and multi-tasking systems. Depending on the application, a 12 MHz 80286's performance is up to six times faster than the standard 5 MHz 8086's, while providing complete upward software compatibility with Intel's 8086, 88, and 186 family of CPU's.

The 80286 operates in two modes: 8086 real address mode and protected virtual address mode. Both modes execute a superset of the 8086 and 88 instruction set.

In 8086 real address mode programs use real addresses with up to one megabyte of address space. Programs use virtual addresses in protected virtual address mode, also called protected mode. In protected mode, the 80286 CPU automatically maps 1 gigabyte of virtual addresses per task into a 16 megabyte real address space. This mode also provides memory protection to isolate the operating system and ensure privacy of each tasks' programs and data. Both modes provide the same base instruction set, registers, and addressing modes.

The following Functional Description describes first, the base 80286 architecture common to both modes, second, 8086 real address mode, and third, protected mode.

80286 BASE ARCHITECTURE

The 8086, 88, 186, and 286 CPU family all contain the same basic set of registers, instructions, and addressing modes. The 80286 processor is upward compatible with the 8086, 8088, and 80186 CPU's.

Register Set

The 80286 base architecture has fifteen registers as shown in Figure 3. These registers are grouped into the following four categories:

General Registers: Eight 16-bit general purpose registers used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used either in their entirety as 16-bit words or split into pairs of separate 8-bit registers.

Segment Registers: Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers: Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode determines the specific registers used for operand address calculations.

Status and Control Registers: The 3 16-bit special purpose registers in figure 3A record or control certain aspects of the 80286 processor state including the Instruction Pointer, which contains the offset address of the next sequential instruction to be executed.

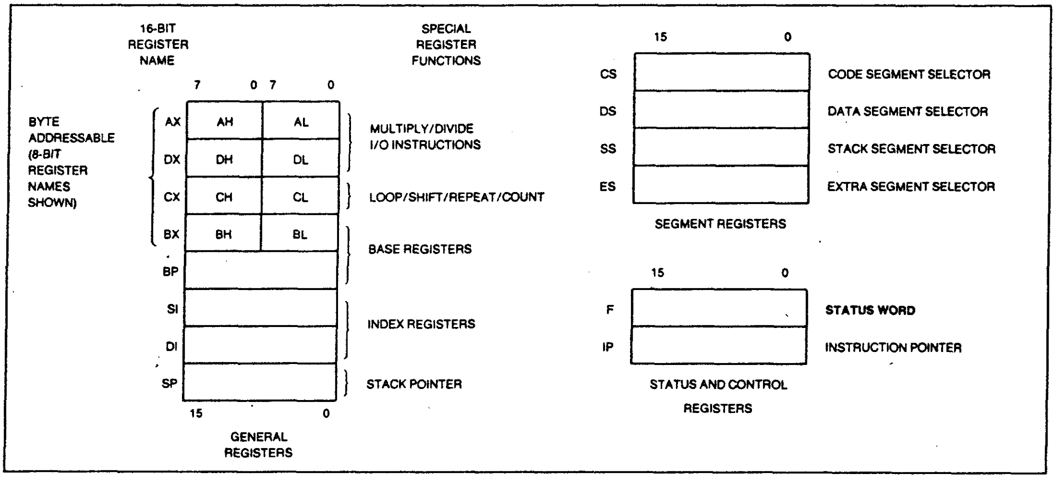
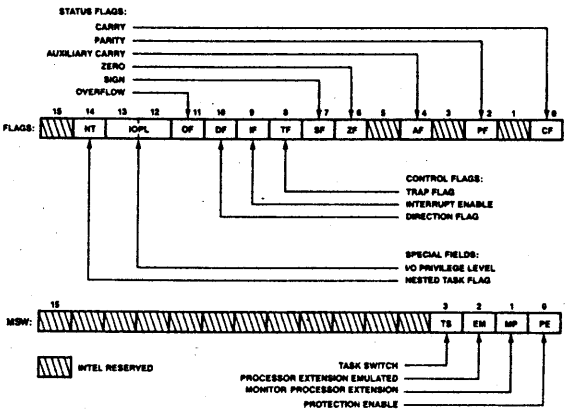


Figure 3. Register Set



210253-4

Figure 3a. Status and Control Register Bit Functions

Flags Word Description

The Flags word (Flags) records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80286 within a given operating mode (bits 8 and 9). Flags is a 16-bit register. The function of the flag bits is given in Table 2.

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, control transfer, high level instructions, and processor control. These categories are summarized in Figure 4.

An 80286 instruction can reference zero, one, or two operands; where an operand resides in a register, in the instruction itself, or in memory. Zero-operand instructions (e.g. NOP and HLT) are usually one byte long. One-operand instructions (e.g. INC and DEC) are usually two bytes long but some are encoded in only one byte. One-operand instructions may reference a register or memory location. Two-operand instructions permit the following six types of instruction operations:

—Register to Register

—Memory to Register

—Immediate to Register

—Memory to Memory

—Register to Memory

—Immediate to Memory

Table 2. Flags Word Bit Functions

Bit Position	Name	Function
0	CF	Carry Flag—Set on high-order bit carry or borrow; cleared otherwise
2	PF	Parity Flag—Set if low-order 8 bits of result contain an even number of 1-bits; cleared otherwise
4	AF	Set on carry from or borrow to the low order four bits of AL; cleared otherwise
6	ZF	Zero Flag—Set if result is zero; cleared otherwise
7	SF	Sign Flag—Set equal to high-order bit of result (0 if positive, 1 if negative)
11	OF	Overflow Flag—Set if result is a too-large positive number or a too-small negative number (excluding sign-bit) to fit in destination operand; cleared otherwise
8	TF	Single Step Flag—Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt.
9	IF	Interrupt-enable Flag—When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location.
10	DF	Direction Flag—Causes string instructions to auto decrement the appropriate index registers when set. Clearing DF causes auto increment.

Two-operand instructions (e.g. MOV and ADD) are usually three to six bytes long. Memory to memory operations are provided by a special class of string instructions requiring one to three bytes. For detailed instruction formats and encodings refer to the instruction set summary at the end of this document.

For detailed operation and usage of each instruction, see Appendix of 80286 Programmer's Reference Manual (Order No. 210498)

GENERAL PURPOSE	
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
INPUT/OUTPUT	
IN	Input byte or word
OUT	Output byte or word
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
FLAG TRANSFER	
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

Figure 4a. Data Transfer Instructions

MOVS	Move byte or word string
INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/REPZ	Repeat while equal/zero
REPNE/REPNZ	Repeat while not equal/not zero

Figure 4c. String Instructions

ADDITION	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP.	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
MULTIPLICATION	
MUL	Multiple byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
DIVISION	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to doubleword

Figure 4b. Arithmetic Instructions

LOGICALS	
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
SHIFTS	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
ROTATES	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

Figure 4d. Shift/Rotate Logical Instructions

CONDITIONAL TRANSFERS	
JA/JNBE	Jump if above/not below nor equal
JAE/JNB	Jump if above or equal/not below
JB/JNAE	Jump if below/not above nor equal
JBE/JNA	Jump if below or equal/not above
JC	Jump if carry
JE/JZ	Jump if equal/zero
JG/JNLE	Jump if greater/not less nor equal
JGE/JNL	Jump if greater or equal/not less
JL/JNGE	Jump if less/not greater nor equal
JLE/JNG	Jump if less or equal/not greater
JNC	Jump if not carry
JNE/JNZ	Jump if not equal/not zero
JNO	Jump if not overflow
JNP/JPO	Jump if not parity/parity odd
JNS	Jump if not sign
JO	Jump if overflow
JP/JPE	Jump if parity/parity even
JS	Jump if sign

UNCONDITIONAL TRANSFERS	
CALL	Call procedure
RET	Return from procedure
JMP	Jump
ITERATION CONTROLS	
LOOP	Loop
LOOPE/LOOPZ	Loop if equal/zero
LOOPNE/LOOPNZ	Loop if not equal/not zero
JCXZ	Jump if register CX = 0
INTERRUPTS	
INT	Interrupt
INTO	Interrupt if overflow
IRET	Interrupt return

Figure 4e. Program Transfer Instructions

FLAG OPERATIONS	
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
EXTERNAL SYNCHRONIZATION	
HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{\text{BUSY}}$ not active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
NO OPERATION	
NOP	No operation
EXECUTION ENVIRONMENT CONTROL	
LMSW	Load machine status word
SMSW	Store machine status word

Figure 4f. Processor Control Instructions

ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

Figure 4g. High Level Instructions

Memory Organization

Memory is organized as sets of variable length segments. Each segment is a linear contiguous sequence of up to 64K (2^{16}) 8-bit bytes. Memory is addressed using a two component address (a pointer) that consists of a 16-bit segment selector, and a 16-bit offset. The segment selector indicates the desired segment in memory. The offset component indicates the desired byte address within the segment.

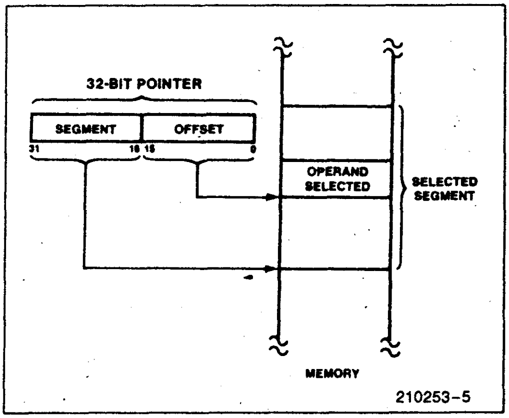


Figure 5. Two Component Address

Table 3. Segment Register Selection Rules

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Automatic with instruction prefetch
Stack	Stack (SS)	All stack pushes and pops. Any memory reference which uses BP as a base register.
Local Data	Data (DS)	All data references except when relative to stack or string destination
External (Global) Data	Extra (ES)	Alternate data segment and destination of string operation

All instructions that address operands in memory must specify the segment and the offset. For speed and compact instruction encoding, segment selectors are usually stored in the high speed segment registers. An instruction need specify only the desired segment register and an offset in order to address a memory operand.

Most instructions need not explicitly specify which segment register is used. The correct segment register is automatically chosen according to the rules of Table 3. These rules follow the way programs are written (see Figure 6) as independent modules that require areas for code and data, a stack, and access to external data areas.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs. To access operands not residing in one of the four immediately available segments, a full 32-bit pointer or a new segment selector must be loaded.

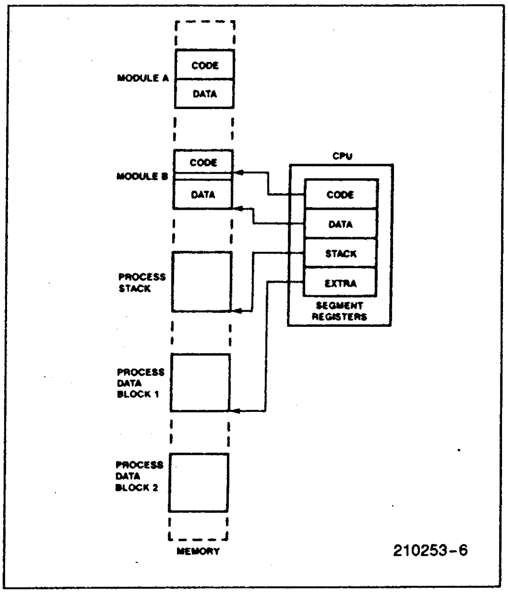


Figure 6. Segmented Memory Helps Structure Software

Addressing Modes

The 80286 provides a total of eight addressing modes for instructions to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

Register Operand Mode: The operand is located in one of the 8 or 16-bit general registers.

Immediate Operand Mode: The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: segment selector and offset. The segment selector is supplied by a segment register either implicitly chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset is calculated by summing any combination of the following three address elements:

the **displacement** (an 8 or 16-bit immediate value contained in the instruction)

the **base** (contents of either the BX or BP base registers)

the **index** (contents of either the SI or DI index registers)

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes, described below.

Direct Mode: The operand's offset is contained in the instruction as an 8 or 16-bit displacement element.

Register Indirect Mode: The operand's offset is in one of the registers SI, DI, BX, or BP.

Based Mode: The operand's offset is the sum of an 8 or 16-bit displacement and the contents of a base register (BX or BP).

Indexed Mode: The operand's offset is the sum of an 8 or 16-bit displacement and the contents of an index register (SI or DI).

Based Indexed Mode: The operand's offset is the sum of the contents of a base register and an index register.

Based Indexed Mode with Displacement: The operand's offset is the sum of a base register's contents, an index register's contents, and an 8 or 16-bit displacement.

Data Types

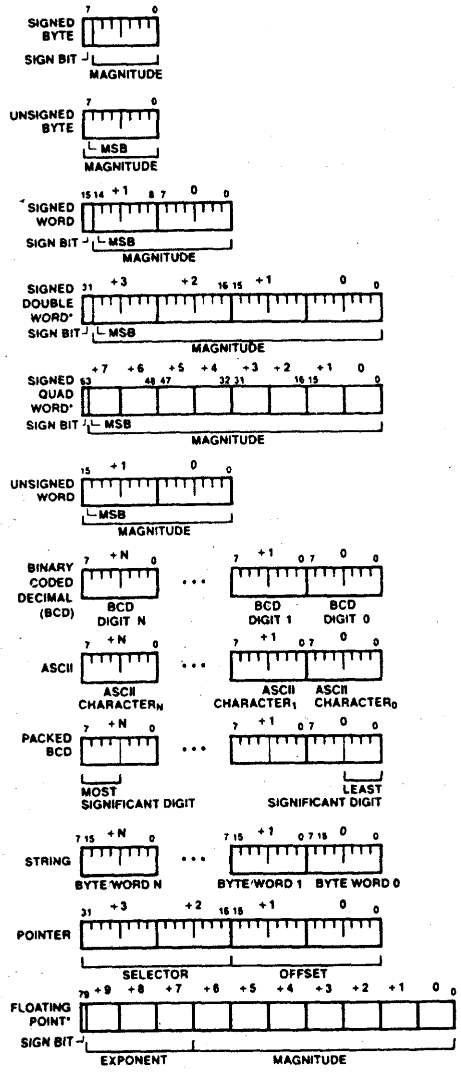
The 80286 directly supports the following data types:

Integer:	A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation. Signed 32 and 64-bit integers are supported using the Numeric Data Processor, the 80287.
Ordinal:	An unsigned binary numeric value contained in an 8-bit byte or 16-bit word.
Pointer:	A 32-bit quantity, composed of a segment selector component and an offset component. Each component is a 16-bit word.
String:	A contiguous sequence of bytes or words. A string may contain from 1 byte to 64K bytes.
ASCII:	A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
BCD:	A byte (unpacked) representation of the decimal digits 0–9.
Packed BCD:	A byte (packed) representation of two decimal digits 0–9 storing one digit in each nibble of the byte.
Floating Point:	A signed 32, 64, or 80-bit real number representation. (Floating point operands are supported using the 80287 Numeric Processor).

Figure 7 graphically represents the data types supported by the 80286.

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. I/O instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register. 8-bit port addresses are zero extended such that A_{15}–A_8 are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.



210253-7

*Supported by 80287 Numeric Data Processor

Figure 7. 80286 Supported Data Types

Table 4. Interrupt Vector Assignments

Function	Interrupt Number	Related Instructions	Does Return Address Point to Instruction Causing Exception?
Divide error exception	0	DIV, IDIV	Yes
Single step interrupt	1	All	
NMI interrupt	2	INT 2 or NMI pin	
Breakpoint interrupt	3	INT 3	
INTO detected overflow exception	4	INTO	No
BOUND range exceeded exception	5	BOUND	Yes
Invalid opcode exception	6	Any undefined opcode	Yes
Processor extension not available exception	7	ESC or WAIT	Yes
Intel reserved-do not use	8-15		
Processor extension error interrupt	16	ESC or WAIT	
Intel reserved-do not use	17-31		
User defined	32-255		

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Flags) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable. Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. The return address from an exception will always point at the instruction causing the exception and include any leading instruction prefixes.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0–31, some of which are used for instruction exceptions, are reserved. For each interrupt, an 8-bit vector must be supplied to the 80286 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

MASKABLE INTERRUPT (INTR)

The 80286 provides a maskable hardware interrupt request pin, INTR. Software enables this input by setting the interrupt flag bit (IF) in the flag word. All 224 user-defined interrupt sources can share this input, yet they can retain separate interrupt handlers. An 8-bit vector read by the CPU during the interrupt acknowledge sequence (discussed in System Interface section) identifies the source of the interrupt.

Further maskable interrupts are disabled while servicing an interrupt by resetting the IF bit as part of the response to an interrupt or exception. The saved flag word will reflect the enable status of the processor prior to the interrupt. Until the flag word is restored to the flag register, the interrupt flag will be zero unless specifically set. The interrupt return instruction includes restoring the flag word, thereby restoring the original status of IF.

NON-MASKABLE INTERRUPT REQUEST (NMI)

A non-maskable interrupt input (NMI) is also provided. NMI has higher priority than INTR. A typical use of NMI would be to activate a power failure routine. The activation of this input causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed.

While executing the NMI servicing procedure, the 80286 will service neither further NMI requests, INTR requests, nor the processor extension segment overrun interrupt until an interrupt return (IRET) instruction is executed or the CPU is reset. If NMI occurs while currently servicing an NMI, its presence will be saved for servicing after executing the first IRET instruction. IF is cleared at the beginning of an NMI interrupt to inhibit INTR interrupts.

SINGLE STEP INTERRUPT

The 80286 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single step interrupt and is controlled by the single step flag bit (TF) in the flag word. Once this bit is set, an internal single step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single stepped.

Interrupt Priorities

When simultaneous interrupt requests occur, they are processed in a fixed order as shown in Table 5. Interrupt processing involves saving the flags, return address, and setting CS:IP to point at the first instruction of the interrupt handler. If other interrupts remain enabled they are processed before the first instruction of the current interrupt handler is executed. The last interrupt processed is therefore the first one serviced.

Table 5. Interrupt Processing Order

Order	Interrupt
1	Instruction exception
2	Single step
3	NMI
4	Processor extension segment overrun
5	INTR
6	INT instruction

Initialization and Processor Reset

Processor initialization or start up is accomplished by driving the RESET input pin HIGH. RESET forces the 80286 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as RESET is active. After RESET becomes inactive and an internal processing interval elapses, the 80286 begins execution in real address mode with the instruction at physical location FFFFF0(H). RESET also sets some registers to predefined values as shown in Table 6.

Table 6. 80286 Initial Register State after RESET

Flag word	0002(H)
Machine Status Word	FFF0(H)
Instruction pointer	FFF0(H)
Code segment	F000(H)
Data segment	0000(H)
Extra segment	0000(H)
Stack segment	0000(H)

HOLD must not be active during the time from the leading edge of RESET to 34 CLKs after the trailing edge of RESET.

Machine Status Word Description

The machine status word (MSW) records when a task switch takes place and controls the operating mode of the 80286. It is a 16-bit register of which the lower four bits are used. One bit places the CPU into protected mode, while the other three bits, as shown in Table 7, control the processor extension interface. After RESET, this register contains FFF0(H) which places the 80286 in 8086 real address mode.

Table 7. MSW Bit Functions

Bit Position	Name	Function
0	PE	Protected mode enable places the 80286 into protected mode and cannot be cleared except by RESET.
1	MP	Monitor processor extension allows WAIT instructions to cause a processor extension not present exception (number 7).
2	EM	Emulate processor extension causes a processor extension not present exception (number 7) on ESC instructions to allow emulating a processor extension.
3	TS	Task switched indicates the next instruction using a processor extension will cause exception 7, allowing software to test whether the current processor extension context belongs to the current task.

The LMSW and SMSW instructions can load and store the MSW in real address mode. The recommended use of TS, EM, and MP is shown in Table 8.

Table 8. Recommended MSW Encodings For Processor Extension Control

TS	MP	EM	Recommended Use	Instructions Causing Exception 7
0	0	0	Initial encoding after RESET. 80286 operation is identical to 8086, 88.	None
0	0	1	No processor extension is available. Software will emulate its function.	ESC
1	0	1	No processor extension is available. Software will emulate its function. The current processor extension context may belong to another task.	ESC
0	1	0	A processor extension exists.	None
1	1	0	A processor extension exists. The current processor extension context may belong to another task. The Exception 7 on WAIT allows software to test for an error pending from a previous processor extension operation.	ESC or WAIT

Halt

The HLT instruction stops program execution and prevents the CPU from using the local bus until restarted. Either NMI, INTR with IF = 1, or RESET will force the 80286 out of halt. If interrupted, the saved CS:IP will point to the next instruction after the HLT.

8086 REAL ADDRESS MODE

The 80286 executes a fully upward-compatible superset of the 8086 instruction set in real address mode. In real address mode the 80286 is object code compatible with 8086 and 8088 software. The real address mode architecture (registers and addressing modes) is exactly as described in the 80286 Base Architecture section of this Functional Description.

Memory Size

Physical memory is a contiguous array of up to 1,048,576 bytes (one megabyte) addressed by pins A_0 through A_{19} and $\overline{BHE}$. A_{20} through A_{23} should be ignored.

Memory Addressing

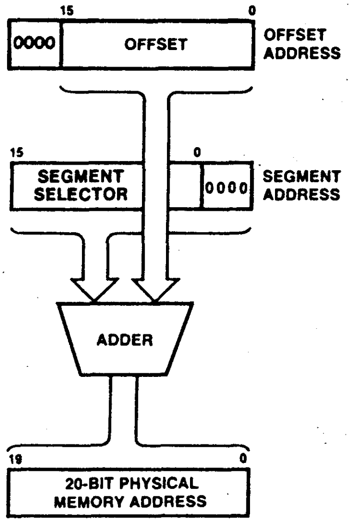
In real address mode physical memory is a contiguous array of up to 1,048,576 bytes (one megabyte) addressed by pins A_0 through A_{19} and $\overline{BHE}$. Address bits A_{20}–A_{23} may not always be zero in real mode. A_{20}–A_{23} should not be used by the system while the 80286 is operating in Real Mode.

The selector portion of a pointer is interpreted as the upper 16 bits of a 20-bit segment address. The lower four bits of the 20-bit segment address are always zero. Segment addresses, therefore, begin on multiples of 16 bytes. See Figure 8 for a graphic representation of address information.

All segments in real address mode are 64K bytes in size and may be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment (e.g. a word with its low order byte at offset FFFF(H) and its high order byte at offset 0000(H). If, in real address mode, the information contained in a segment does not use the full 64K bytes, the unused end of the segment may be overlayed by another segment to reduce physical memory requirements.

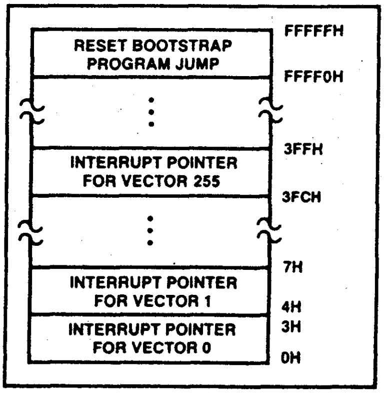
Reserved Memory Locations

The 80286 reserves two fixed areas of memory in real address mode (see Figure 9); system initialization area and interrupt table area. Locations from addresses FFFF0(H) through FFFFF(H) are reserved for system initialization. Initial execution begins at location FFFF0(H). Locations 00000(H) through 003FF(H) are reserved for interrupt vectors.



210253-8

Figure 8. 8086 Real Address Mode Address Calculation


RESET BOOTSTRAP PROGRAM JUMP
FFFFFH
FFFF0H
3FFH
INTERRUPT POINTER FOR VECTOR 255
3FCH
7H
INTERRUPT POINTER FOR VECTOR 1
4H
3H
INTERRUPT POINTER FOR VECTOR 0
0H

INITIAL CS:IP VALUE IS F000:FFF0.

210253-9

Figure 9. 8086 Real Address Mode Initially Reserved Memory Locations

Table 9. Real Address Mode Addressing Interrupts

Function	Interrupt Number	Related Instructions	Return Address Before Instruction?
Interrupt table limit too small exception	8	INT vector is not within table limit	Yes
Processor extension segment overrun interrupt	9	ESC with memory operand extending beyond offset FFFF(H)	No
Segment overrun exception	13	Word memory reference with offset = FFFF(H) or an attempt to execute past the end of a segment	Yes

Interrupts

Table 9 shows the interrupt vectors reserved for exceptions and interrupts which indicate an addressing error. The exceptions leave the CPU in the state existing before attempting to execute the failing instruction (except for PUSH, POP, PUSHA, or POPA). Refer to the next section on protected mode initialization for a discussion on exception 8.

Protected Mode Initialization

To prepare the 80286 for protected mode, the LIDT instruction is used to load the 24-bit interrupt table base and 16-bit limit for the protected mode interrupt table. This instruction can also set a base and limit for the interrupt vector table in real address mode. After reset, the interrupt table base is initialized to 000000(H) and its size set to 03FF(H). These values are compatible with 8086, 88 software. LIDT should only be executed in preparation for protected mode.

Shutdown

Shutdown occurs when a severe error is detected that prevents further instruction processing by the CPU. Shutdown and halt are externally signalled via a halt bus operation. They can be distinguished by A_1 HIGH for halt and A_1 LOW for shutdown. In real address mode, shutdown can occur under two conditions:

- Exceptions 8 or 13 happen and the IDT limit does not include the interrupt vector.
- A CALL INT or PUSH instruction attempts to wrap around the stack segment when SP is not even.

An NMI input can bring the CPU out of shutdown if the IDT limit is at least 000F(H) and SP is greater than 0005(H), otherwise shutdown can only be exited via the RESET input.

PROTECTED VIRTUAL ADDRESS MODE

The 80286 executes a fully upward-compatible superset of the 8086 instruction set in protected virtual address mode (protected mode). Protected mode also provides memory management and protection mechanisms and associated instructions.

The 80286 enters protected virtual address mode from real address mode by setting the PE (Protection Enable) bit of the machine status word with the Load Machine Status Word (LMSW) instruction. Protected mode offers extended physical and virtual memory address space, memory protection mechanisms, and new operations to support operating systems and virtual memory.

All registers, instructions, and addressing modes described in the 80286 Base Architecture section of this Functional Description remain the same. Programs for the 8086, 88, 186, and real address mode 80286 can be run in protected mode; however, embedded constants for segment selectors are different.

Memory Size

The protected mode 80286 provides a 1 gigabyte virtual address space per task mapped into a 16 megabyte physical address space defined by the address pin A_{23}-A_0 and $\overline{BHE}$. The virtual address space may be larger than the physical address space since any use of an address that does not map to a physical memory location will cause a restartable exception.

Memory Addressing

As in real address mode, protected mode uses 32-bit pointers, consisting of 16-bit selector and offset components. The selector, however, specifies an index into a memory resident table rather than the upper 16-bits of a real memory address. The 24-bit base address of the desired segment is obtained from the tables in memory. The 16-bit offset is added to the segment base address to form the physical address as shown in Figure 10. The tables are automatically referenced by the CPU whenever a segment register is loaded with a selector. All 80286 instructions which load a segment register will reference the memory based tables without additional software. The memory based tables contain 8 byte values called descriptors.

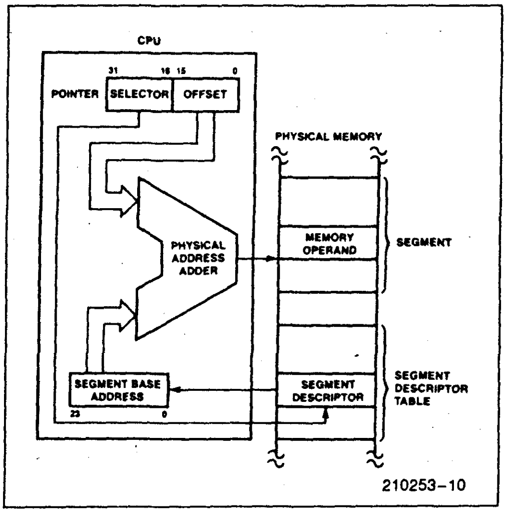


Figure 10. Protected Mode Memory Addressing

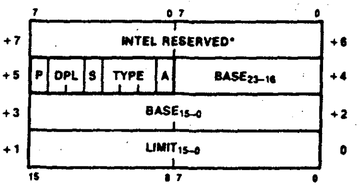
DESCRIPTORS

Descriptors define the use of memory. Special types of descriptors also define new functions for transfer of control and task switching. The 80286 has segment descriptors for code, stack and data segments, and system control descriptors for special system data segments and control transfer operations. Descriptor accesses are performed as locked bus operations to assure descriptor integrity in multi-processor systems.

CODE AND DATA SEGMENT DESCRIPTORS (S = 1)

Besides segment base addresses, code and data descriptors contain other segment attributes including segment size (1 to 64K bytes), access rights (read only, read/write, execute only, and execute/read), and presence in memory (for virtual memory systems) (See Figure 11). Any segment usage violating a segment attribute indicated by the segment descriptor will prevent the memory cycle and cause an exception or interrupt.

Code or Data Segment Descriptor



*Must be set to 0 for compatibility with 80386.

Access Rights Byte Definition

	Bit Position	Name	Function		
	7	Present (P)	P = 1	Segment is mapped into physical memory.	
			P = 0	No mapping to physical memory exits, base and limit are not used.	
	6–5	Descriptor Privilege Level (DPL)		Segment privilege attribute used in privilege tests.	
	4	Segment Descriptor (S)	S = 1	Code or Data (includes stacks) segment descriptor	
			S = 0	System Segment Descriptor or Gate Descriptor	
Type Field Definition	3	Executable (E)	E = 0	Data segment descriptor type is:	If Data Segment (S = 1, E = 0)
	2	Expansion Direction (ED)	ED = 0	Expand up segment, offsets must be ≤ limit.	
			ED = 1	Expand down segment, offsets must be > limit.	
	1	Writeable (W)	W = 0	Data segment may not be written into.	
			W = 1	Data segment may be written into.	
	3	Executable (E)	E = 1	Code Segment Descriptor type is:	If Code Segment (S = 1, E = 1)
	2	Conforming (C)	C = 1	Code segment may only be executed when CPL ≥ DPL and CPL remains unchanged.	
	1	Readable (R)	R = 0	Code segment may not be read	
			R = 1	Code segment may be read.	
	0	Accessed (A)	A = 0	Segment has not been accessed.	
			A = 1	Segment selector has been loaded into segment register or used by selector test instructions.	

Figure 11. Code and Data Segment Descriptor Formats

Code and data (including stack data) are stored in two types of segments: code segments and data segments. Both types are identified and defined by segment descriptors (S = 1). Code segments are identified by the executable (E) bit set to 1 in the descriptor access rights byte. The access rights byte of both code and data segment descriptor types have three fields in common: present (P) bit, Descriptor Privilege Level (DPL), and accessed (A) bit. If P = 0, any attempted use of this segment will cause a not-present exception. DPL specifies the privilege level of the segment descriptor. DPL controls when the descriptor may be used by a task (refer to privilege discussion below). The A bit shows whether the segment has been previously accessed for usage profiling, a necessity for virtual memory systems. The CPU will always set this bit when accessing the descriptor.

Data segments (S = 1, E = 0) may be either read-only or read-write as controlled by the W bit of the access rights byte. Read-only (W = 0) data segments may not be written into. Data segments may grow in two directions, as determined by the Expansion Direction (ED) bit: upwards (ED = 0) for data segments, and downwards (ED = 1) for a segment containing a stack. The limit field for a data segment descriptor is interpreted differently depending on the ED bit (see Figure 11).

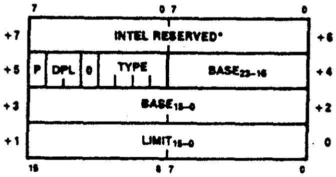
A code segment (S = 1, E = 1) may be execute-only or execute/read as determined by the Readable (R) bit. Code segments may never be written into and execute-only code segments (R = 0) may not be read. A code segment may also have an attribute called conforming (C). A conforming code segment may be shared by programs that execute at different privilege levels. The DPL of a conforming code segment defines the range of privilege levels at which the segment may be executed (refer to privilege discussion below). The limit field identifies the last byte of a code segment.

SYSTEM SEGMENT DESCRIPTORS (S = 0, TYPE = 1–3)

In addition to code and data segment descriptors, the protected mode 80286 defines System Segment Descriptors. These descriptors define special system data segments which contain a table of descriptors (Local Descriptor Table Descriptor) or segments which contain the execution state of a task (Task State Segment Descriptor).

Figure 12 gives the formats for the special system data segment descriptors. The descriptors contain a 24-bit base address of the segment and a 16-bit limit. The access byte defines the type of descriptor, its state and privilege level. The descriptor contents are valid and the segment is in physical memory if P = 1. If P = 0, the segment is not valid. The DPL field is only used in Task State Segment descriptors and indicates the privilege level at which the descriptor may be used (see Privilege). Since the Local Descriptor Table descriptor may only be used by a special privileged instruction, the DPL field is not used. Bit 4 of the access byte is 0 to indicate that it is a system control descriptor. The type field specifies the descriptor type as indicated in Figure 12.

System Segment Descriptor



210253–12

*Must be set to 0 for compatibility with 80386.

System Segment Descriptor Fields

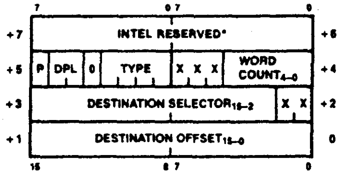
Name	Value	Description
TYPE	1 2 3	Available Task State Segment (TSS) Local Descriptor Table Busy Task State Segment (TSS)
P	0 1	Descriptor contents are not valid Descriptor contents are valid
DPL	0–3	Descriptor Privilege Level
BASE	24-bit number	Base Address of special system data segment in real memory
LIMIT	16-bit number	Offset of last byte in segment

Figure 12. System Segment Descriptor Format

GATE DESCRIPTORS (S = 0, TYPE = 4–7)

Gates are used to control access to entry points within the target code segment. The gate descriptors are call gates, task gates, interrupt gates and trap gates. Gates provide a level of indirection between the source and destination of the control transfer. This indirection allows the CPU to automatically perform protection checks and control entry point of the destination. Call gates are used to change privilege levels (see Privilege), task gates are used to perform a task switch, and interrupt and trap gates are used to specify interrupt service routines. The interrupt gate disables interrupts (resets IF) while the trap gate does not.

Gate Descriptor



210253–13

*Must be set to 0 for compatibility with 80386 (X is don't care)

Gate Descriptor Fields

Name	Value	Description
TYPE	4 5 6 7	–Call Gate –Task Gate –Interrupt Gate –Trap Gate
P	0 1	–Descriptor Contents are not valid –Descriptor Contents are valid
DPL	0–3	Descriptor Privilege Level
WORD COUNT	0–31	Number of words to copy from callers stack to called procedures stack. Only used with call gate.
DESTINATION SELECTOR	16-bit selector	Selector to the target code segment (Call, Interrupt or Trap Gate) Selector to the target task state segment (Task Gate)
DESTINATION OFFSET	16-bit offset	Entry point within the target code segment

Figure 13. Gate Descriptor Format

Figure 13 shows the format of the gate descriptors. The descriptor contains a destination pointer that points to the descriptor of the target segment and the entry point offset. The destination selector in an interrupt gate, trap gate, and call gate must refer to a code segment descriptor. These gate descriptors contain the entry point to prevent a program from constructing and using an illegal entry point. Task gates may only refer to a task state segment. Since task gates invoke a task switch, the destination offset is not used in the task gate.

Exception 13 is generated when the gate is used if a destination selector does not refer to the correct descriptor type. The word count field is used in the call gate descriptor to indicate the number of parameters (0–31 words) to be automatically copied from the caller's stack to the stack of the called routine when a control transfer changes privilege levels. The word count field is not used by any other gate descriptor.

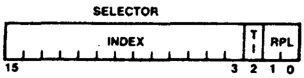
The access byte format is the same for all gate descriptors. P = 1 indicates that the gate contents are valid. P = 0 indicates the contents are not valid and causes exception 11 if referenced. DPL is the descriptor privilege level and specifies when this descriptor may be used by a task (refer to privilege discussion below). Bit 4 must equal 0 to indicate a system control descriptor. The type field specifies the descriptor type as indicated in Figure 13.

SEGMENT DESCRIPTOR CACHE REGISTERS

A segment descriptor cache register is assigned to each of the four segment registers (CS, SS, DS, ES). Segment descriptors are automatically loaded (cached) into a segment descriptor cache register (Figure 14) whenever the associated segment register is loaded with a selector. Only segment descriptors may be loaded into segment descriptor cache registers. Once loaded, all references to that segment of memory use the cached descriptor information instead of reaccessing the descriptor. The descriptor cache registers are not visible to programs. No instructions exist to store their contents. They only change when a segment register is loaded.

SELECTOR FIELDS

A protected mode selector has three fields: descriptor entry index, local or global descriptor table indicator (TI), and selector privilege (RPL) as shown in Figure 15. These fields select one of two memory based tables of descriptors, select the appropriate table entry and allow highspeed testing of the selector's privilege attribute (refer to privilege discussion below).



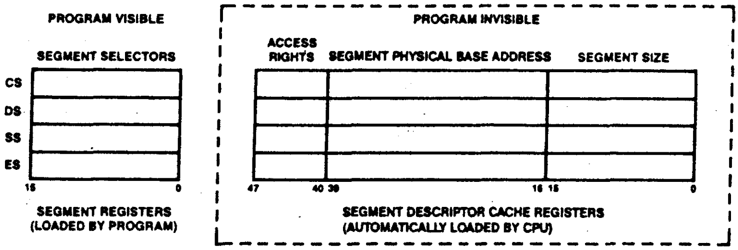
BITS	NAME	FUNCTION
1–0	REQUESTED PRIVILEGE LEVEL (RPL)	INDICATES SELECTOR PRIVILEGE LEVEL DESIRED

2	TABLE INDICATOR (TI)	TI = 0 USE GLOBAL DESCRIPTOR TABLE (GDT)
		TI = 1 USE LOCAL DESCRIPTOR TABLE (LDT)

15–3	INDEX	SELECT DESCRIPTOR ENTRY IN TABLE

210253–15

Figure 15. Selector Fields


PROGRAM VISIBLE
SEGMENT SELECTORS
CS
DS
SS
ES
15
0
SEGMENT REGISTERS (LOADED BY PROGRAM)
PROGRAM INVISIBLE
ACCESS RIGHTS
SEGMENT PHYSICAL BASE ADDRESS
SEGMENT SIZE
47
40 39
16 15
0
SEGMENT DESCRIPTOR CACHE REGISTERS (AUTOMATICALLY LOADED BY CPU)

210253–14

Figure 14. Descriptor Cache Registers

LOCAL AND GLOBAL DESCRIPTOR TABLES

Two tables of descriptors, called descriptor tables, contain all descriptors accessible by a task at any given time. A descriptor table is a linear array of up to 8192 descriptors. The upper 13 bits of the selector value are an index into a descriptor table. Each table has a 24-bit base register to locate the descriptor table in physical memory and a 16-bit limit register that confine descriptor access to the defined limits of the table as shown in Figure 16. A restartable exception (13) will occur if an attempt is made to reference a descriptor outside the table limits.

One table, called the Global Descriptor table (GDT), contains descriptors available to all tasks. The other table, called the Local Descriptor Table (LDT), contains descriptors that can be private to a task. Each task may have its own private LDT. The GDT may contain all descriptor types except interrupt and trap descriptors. The LDT may contain only segment, task gate, and call gate descriptors. A segment cannot be accessed by a task if its segment descriptor does not exist in either descriptor table at the time of access.

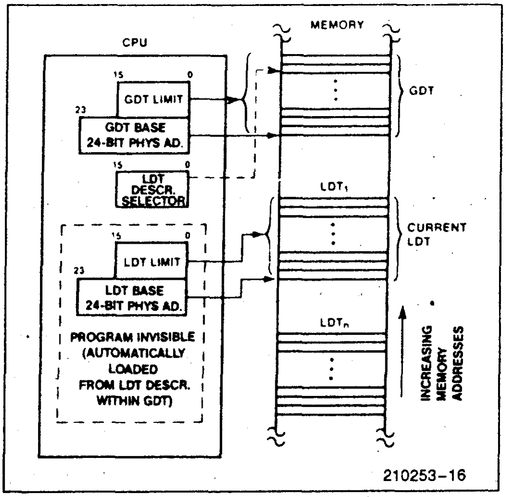
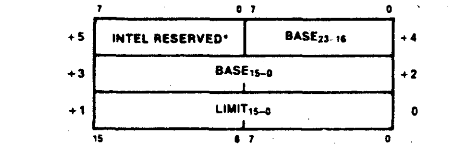


Figure 16. Local and Global Descriptor Table Definition

The LGDT and LLDT instructions load the base and limit of the global and local descriptor tables. LGDT and LLDT are privileged, i.e. they may only be executed by trusted programs operating at level 0. The LGDT instruction loads a six byte field containing the 16-bit table limit and 24-bit physical base address of the Global Descriptor Table as shown in Figure 17. The LLDT instruction loads a selector which refers to a Local Descriptor Table descriptor containing the base address and limit for an LDT, as shown in Figure 12.



*Must be set to 0 for compatibility with 80386.

Figure 17. Global Descriptor Table and Interrupt Descriptor Table Data Type

INTERRUPT DESCRIPTOR TABLE

The protected mode 80286 has a third descriptor table, called the Interrupt Descriptor Table (IDT) (see Figure 18), used to define up to 256 interrupts. It may contain only task gates, interrupt gates and trap gates. The IDT (Interrupt Descriptor Table) has a 24-bit physical base and 16-bit limit register in the CPU. The privileged LIDT instruction loads these registers with a six byte value of identical form to that of the LGDT instruction (see Figure 17 and Protected Mode Initialization).

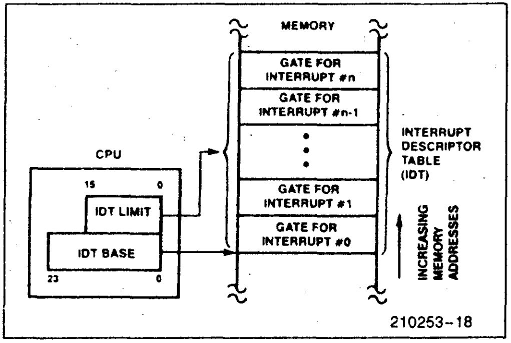
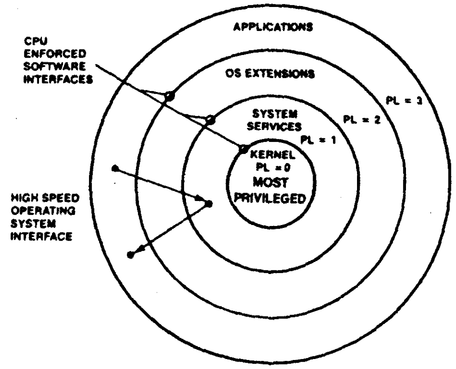


Figure 18. Interrupt Descriptor Table Definition

References to IDT entries are made via INT instructions, external interrupt vectors, or exceptions. The IDT must be at least 256 bytes in size to allocate space for all reserved interrupts.

Privilege

The 80286 has a four-level hierarchical privilege system which controls the use of privileged instructions and access to descriptors (and their associated segments) within a task. Four-level privilege, as shown in Figure 19, is an extension of the user/supervisor mode commonly found in minicomputers. The privilege levels are numbered 0 through 3. Level 0 is the


CPU ENFORCED SOFTWARE INTERFACES
APPLICATIONS
OS EXTENSIONS
SYSTEM SERVICES
KERNEL PL = 0 MOST PRIVILEGED
PL = 3
PL = 2
PL = 1
HIGH SPEED OPERATING SYSTEM INTERFACE

NOTE: PL BECOMES NUMERICALLY LOWER AS PRIVILEGE LEVEL INCREASES

210253-19

most privileged level. Privilege levels provide protection within a task. (Tasks are isolated by providing private LDT's for each task.) Operating system routines, interrupt handlers, and other system software can be included and protected within the virtual address space of each task using the four levels of privilege. Each task in the system has a separate stack for each of its privilege levels.

Tasks, descriptors, and selectors have a privilege level attribute that determines whether the descriptor may be used. Task privilege effects the use of instructions and descriptors. Descriptor and selector privilege only effect access to the descriptor.

TASK PRIVILEGE

A task always executes at one of the four privilege levels. The task privilege level at any specific instant is called the Current Privilege Level (CPL) and is defined by the lower two bits of the CS register. CPL cannot change during execution in a single code segment. A task's CPL may only be changed by control transfers through gate descriptors to a new code segment (See Control Transfer). Tasks begin executing at the CPL value specified by the code segment selector within TSS when the task is initiated via a task switch operation (See Figure 20). A task executing at Level 0 can access all data segments defined in the GDT and the task's LDT and is considered the most trusted level. A task executing a Level 3 has the most restricted access to data and is considered the least trusted level.

DESCRIPTOR PRIVILEGE

Descriptor privilege is specified by the Descriptor Privilege Level (DPL) field of the descriptor access byte. DPL specifies the least trusted task privilege level (CPL) at which a task may access the descriptor. Descriptors with DPL = 0 are the most protected. Only tasks executing at privilege level 0 (CPL = 0) may access them. Descriptors with DPL = 3 are the least protected (i.e. have the least restricted access) since tasks can access them when CPL = 0, 1, 2, or 3. This rule applies to all descriptors, except LDT descriptors.

SELECTOR PRIVILEGE

Selector privilege is specified by the Requested Privilege Level (RPL) field in the least significant two bits of a selector. Selector RPL may establish a less trusted privilege level than the current privilege level for the use of a selector. This level is called the task's effective privilege level (EPL). RPL can only reduce the scope of a task's access to data with this selector. A task's effective privilege is the numeric maximum of RPL and CPL. A selector with RPL = 0 imposes no additional restriction on its use while a selector with RPL = 3 can only refer to segments at privilege Level 3 regardless of the task's CPL. RPL is generally used to verify that pointer parameters passed to a more trusted procedure are not allowed to use data at a more privileged level than the caller (refer to pointer testing instructions).

Descriptor Access and Privilege Validation

Determining the ability of a task to access a segment involves the type of segment to be accessed, the instruction used, the type of descriptor used and CPL, RPL, and DPL. The two basic types of segment accesses are control transfer (selectors loaded into CS) and data (selectors loaded into DS, ES or SS).

DATA SEGMENT ACCESS

Instructions that load selectors into DS and ES must refer to a data segment descriptor or readable code segment descriptor. The CPL of the task and the RPL of the selector must be the same as or more privileged (numerically equal to or lower than) than the descriptor DPL. In general, a task can only access data segments at the same or less privileged levels than the CPL or RPL (whichever is numerically higher) to prevent a program from accessing data it cannot be trusted to use.

An exception to the rule is a readable conforming code segment. This type of code segment can be read from any privilege level.

If the privilege checks fail (e.g. DPL is numerically less than the maximum of CPL and RPL) or an incorrect type of descriptor is referenced (e.g. gate descriptor or execute only code segment) exception 13 occurs. If the segment is not present, exception 11 is generated.

Instructions that load selectors into SS must refer to data segment descriptors for writable data segments. The descriptor privilege (DPL) and RPL must equal CPL. All other descriptor types or a privilege level violation will cause exception 13. A not present fault causes exception 12.

CONTROL TRANSFER

Four types of control transfer can occur when a selector is loaded into CS by a control transfer operation (see Table 10). Each transfer type can only occur if the operation which loaded the selector references the correct descriptor type. Any violation of these descriptor usage rules (e.g. JMP through a call gate or RET to a Task State Segment) will cause exception 13.

The ability to reference a descriptor for control transfer is also subject to rules of privilege. A CALL or JUMP instruction may only reference a code segment descriptor with DPL equal to the task CPL or a conforming segment with DPL of equal or greater privilege than CPL. The RPL of the selector used to reference the code descriptor must have as much privilege as CPL.

RET and IRET instructions may only reference code segment descriptors with descriptor privilege equal to or less privileged than the task CPL. The selector loaded into CS is the return address from the stack. After the return, the selector RPL is the task's new CPL. If CPL changes, the old stack pointer is popped after the return address.

When a JMP or CALL references a Task State Segment descriptor, the descriptor DPL must be the same or less privileged than the task's CPL. Reference to a valid Task State Segment descriptor causes a task switch (see Task Switch Operation). Reference to a Task State Segment descriptor at a more privileged level than the task's CPL generates exception 13.

When an instruction or interrupt references a gate descriptor, the gate DPL must have the same or less privilege than the task CPL. If DPL is at a more privileged level than CPL, exeception 13 occurs. If the destination selector contained in the gate references a code segment descriptor, the code segment descriptor DPL must be the same or more privileged than the task CPL. If not, Exception 13 is issued. After the control transfer, the code segment descriptors DPL is the task's new CPL. If the destination selector in the gate references a task state segment, a task switch is automatically performed (see Task Switch Operation).

The privilege rules on control transfer require:

— JMP or CALL direct to a code segment (code segment descriptor) can only be to a conforming segment with DPL of equal or greater privilege than CPL or a non-conforming segment at the same privilege level.

— interrupts within the task or calls that may change privilege levels, can only transfer control through a gate at the same or a less privileged level than CPL to a code segment at the same or more privileged level than CPL.

— return instructions that don't switch tasks can only return control to a code segment at the same or less privileged level.

— task switch can be performed by a call, jump or interrupt which references either a task gate or task state segment at the same or less privileged level.

Table 10. Descriptor Types Used for Control Transfer

Control Transfer Types	Operation Types	Descriptor Referenced	Descriptor Table
Intersegment within the same privilege level	JMP, CALL, RET, IRET*	Code Segment	GDT/LDT
Intersegment to the same or higher privilege level Interrupt within task may change CPL.	CALL	Call Gate	GDT/LDT
	Interrupt Instruction, Exception, External Interrupt	Trap or Interrupt Gate	IDT
Intersegment to a lower privilege level (changes task CPL)	RET, IRET*	Code Segment	GDT/LDT
	CALL, JMP	Task State Segment	GDT
Task Switch	CALL, JMP	Task Gate	GDT/LDT
	IRET** Interrupt Instruction, Exception, External Interrupt	Task Gate	IDT

*NT (Nested Task bit of flag word) = 0
**NT (Nested Task bit of flag word) = 1

PRIVILEGE LEVEL CHANGES

Any control transfer that changes CPL within the task, causes a change of stacks as part of the operation. Initial values of SS:SP for privilege levels 0, 1, and 2 are kept in the task state segment (refer to Task Switch Operation). During a JMP or CALL control transfer, the new stack pointer is loaded into the SS and SP registers and the previous stack pointer is pushed onto the new stack.

When returning to the original privilege level, its stack is restored as part of the RET or IRET instruction operation. For subroutine calls that pass parameters on the stack and cross privilege levels, a fixed number of words, as specified in the gate, are copied from the previous stack to the current stack. The inter-segment RET instruction with a stack adjustment value will correctly restore the previous stack pointer upon return.

Protection

The 80286 includes mechanisms to protect critical instructions that affect the CPU execution state (e.g. HLT) and code or data segments from improper usage. These protection mechanisms are grouped into three forms:

Restricted *usage* of segments (e.g. no write allowed to read-only data segments). The only segments available for use are defined by descriptors in the Local Descriptor Table (LDT) and Global Descriptor Table (GDT).

Restricted *access* to segments via the rules of privilege and descriptor usage.

Privileged instructions or operations that may only be executed at certain privilege levels as determined by the CPL and I/O Privilege Level (IOPL). The IOPL is defined by bits 14 and 13 of the flag word.

These checks are performed for all instructions and can be split into three categories: segment load checks (Table 11), operand reference checks (Table 12), and privileged instruction checks (Table 13). Any violation of the rules shown will result in an exception. A not-present exception related to the stack segment causes exception 12.

The IRET and POPF instructions do not perform some of their defined functions if CPL is not of sufficient privilege (numerically small enough). Precisely these are:

- The IF bit is not changed if CPL > IOPL.
- The IOPL field of the flag word is not changed if CPL > 0.

No exceptions or other indication are given when these conditions occur.

Table 11
Segment Register Load Checks

Error Description	Exception Number
Descriptor table limit exceeded	13
Segment descriptor not-present	11 or 12
Privilege rules violated	13
Invalid descriptor/segment type segment register load: —Read only data segment load to SS —Special Control descriptor load to DS, ES, SS —Execute only segment load to DS, ES, SS —Data segment load to CS —Read/Execute code segment load to SS	13

Table 12. Operand Reference Checks

Error Description	Exception Number
Write into code segment	13
Read from execute-only code segment	13
Write to read-only data segment	13
Segment limit exceeded[1]	12 or 13

NOTE:
Carry out in offset calculations is ignored.

Table 13. Privileged Instruction Checks

Error Description	Exception Number
CPL ≠ 0 when executing the following instructions: LIDT, LLDT, LGDT, LTR, LMSW, CTS, HLT	13
CPL > IOPL when executing the following instructions: INS, IN, OUTS, OUT, STI, CLI, LOCK	13

EXCEPTIONS

The 80286 detects several types of exceptions and interrupts, in protected mode (see Table 14). Most are restartable after the exceptional condition is removed. Interrupt handlers for most exceptions can read an error code, pushed on the stack after the return address, that identifies the selector involved (0 if none). The return address normally points to the failing instruction, including all leading prefixes. For a processor extension segment overrun exception, the return address will not point at the ESC instruction that caused the exception; however, the processor extension registers may contain the address of the failing instruction.

Table 14. Protected Mode Exceptions

Interrupt Vector	Function	Return Address At Falling Instruction?	Always Restart-able?	Error Code on Stack?
8	Double exception detected	Yes	No[2]	Yes
9	Processor extension segment overrun	No	No[2]	No
10	Invalid task state segment	Yes	Yes	Yes
11	Segment not present	Yes	Yes	Yes
12	Stack segment overrun or stack segment not present	Yes	Yes[1]	Yes
13	General protection	Yes	No[2]	Yes

NOTE:
1. When a PUSHA or POPA instruction attempts to wrap around the stack segment, the machine state after the exception will not be restartable because stack segment wrap around is not permitted. This condition is identified by the value of the saved SP being either 0000(H), 0001(H), FFFE(H), or FFFF(H).
2. These exceptions indicate a violation to privilege rules or usage rules has occurred. Restart is generally not attempted under those conditions.

These exceptions indicate a violation to privilege rules or usage rules has occurred. Restart is generally not attempted under those conditions.

All these checks are performed for all instructions and can be split into three categories: segment load checks (Table 11), operand reference checks (Table 12), and privileged instruction checks (Table 13). Any violation of the rules shown will result in an exception. A not-present exception causes exception 11 or 12 and is restartable.

Special Operations

TASK SWITCH OPERATION

The 80286 provides a built-in task switch operation which saves the entire 80286 execution state (registers, address space, and a link to the previous task), loads a new execution state, and commences execution in the new task. Like gates, the task switch operation is invoked by executing an inter-segment JMP or CALL instruction which refers to a Task State Segment (TSS) or task gate descriptor in the GDT or LDT. An INT n instruction, exception, or external interrupt may also invoke the task switch operation by selecting a task gate descriptor in the associated IDT descriptor entry.

The TSS descriptor points at a segment (see Figure 20) containing the entire 80286 execution state while a task gate descriptor contains a TSS selector. The limit field of the descriptor must be >002B(H).

Each task must have a TSS associated with it. The current TSS is identified by a special register in the 80286 called the Task Register (TR). This register contains a selector referring to the task state segment descriptor that defines the current TSS. A hidden base and limit register associated with TR are loaded whenever TR is loaded with a new selector.

The IRET instruction is used to return control to the task that called the current task or was interrupted. Bit 14 in the flag register is called the Nested Task (NT) bit. It controls the function of the IRET instruction. If NT = 0, the IRET instruction performs the regular current task by popping values off the stack; when NT = 1, IRET performs a task switch operation back to the previous task.

When a CALL, JMP, or INT instruction initiates a task switch, the old (except for case of JMP) and new TSS will be marked busy and the back link field of the new TSS set to the old TSS selector. The NT bit of the new task is set by CALL or INT initiated task switches. An interrupt that does not cause a task switch will clear NT. NT may also be set or cleared by POPF or IRET instructions.

The task state segment is marked busy by changing the descriptor type field from Type 1 to Type 3. Use of a selector that references a busy task state segment causes Exception 13.

PROCESSOR EXTENSION CONTEXT SWITCHING

The context of a processor extension (such as the 80287 numerics processor) is not changed by the task switch operation. A processor extension context need only be changed when a different task attempts to use the processor extension (which still contains the context of a previous task). The 80286 detects the first use of a processor extension after a task switch by causing the processor extension not present exception (7). The interrupt handler may then decide whether a context change is necessary.

Whenever the 80286 switches tasks, it sets the Task Switched (TS) bit of the MSW. TS indicates that a processor extension context may belong to a different task than the current one. The processor extension not present exception (7) will occur when attempting to execute an ESC or WAIT instruction if TS = 1 and a processor extension is present (MP = 1 in MSW).

POINTER TESTING INSTRUCTIONS

The 80286 provides several instructions to speed pointer testing and consistency checks for maintaining system integrity (see Table 15). These instructions use the memory management hardware to verify that a selector value refers to an appropriate segment without risking an exception. A condition flag (ZF) indicates whether use of the selector or segment will cause an exception.

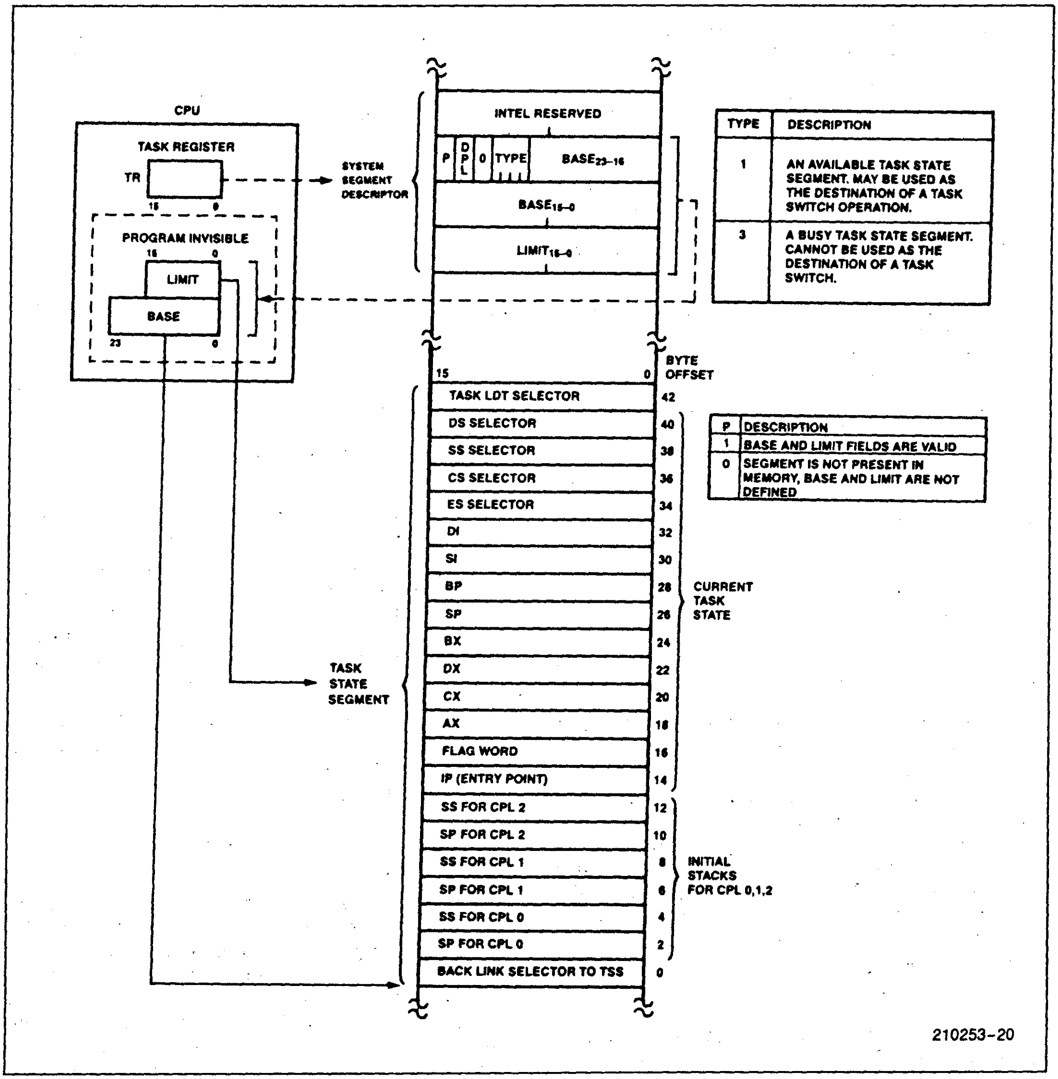


Figure 20. Task State Segment and TSS Registers

Table 15. 80286 Pointer Test Instructions

Instruction	Operands	Function
ARPL	Selector, Register	Adjust Requested Privilege Level: adjusts the RPL of the selector to the numeric maximum of current selector RPL value and the RPL value in the register. Set zero flag if selector RPL was changed by ARPL.
VERR	Selector	VERify for Read: sets the zero flag if the segment referred to by the selector can be read.
VERW	Selector	VERify for Write: sets the zero flag if the segment referred to by the selector can be written.
LSL	Register, Selector	Load Segment Limit: reads the segment limit into the register if privilege rules and descriptor type allow. Set zero flag if successful.
LAR	Register, Selector	Load Access Rights: reads the descriptor access rights byte into the register if privilege rules allow. Set zero flag if successful.

DOUBLE FAULT AND SHUTDOWN

If two separate exceptions are detected during a single instruction execution, the 80286 performs the double fault exception (8). If an execution occurs during processing of the double fault exception, the 80286 will enter shutdown. During shutdown no further instructions or exceptions are processed. Either NMI (CPU remains in protected mode) or RESET (CPU exits protected mode) can force the 80286 out of shutdown. Shutdown is externally signalled via a HALT bus operation with A_1 LOW.

PROTECTED MODE INITIALIZATION

The 80286 initially executes in real address mode after RESET. To allow initialization code to be placed at the top of physical memory, A_{23}–A_{20} will be HIGH when the 80286 performs memory references relative to the CS register until CS is changed. A_{23}–A_{20} will be zero for references to the DS, ES, or SS segments. Changing CS in real address mode will force A_{23}–A_{20} LOW whenever CS is used again. The initial CS:IP value of F000:FFF0 provides 64K bytes of code space for initialization code without changing CS.

Protected mode operation requires several registers to be initialized. The GDT and IDT base registers must refer to a valid GDT and IDT. After executing the LMSW instruction to set PE, the 80286 must immediately execute an intra-segment JMP instruction to clear the instruction queue of instructions decoded in real address mode.

To force the 80286 CPU registers to match the initial protected mode state assumed by software, execute a JMP instruction with a selector referring to the initial TSS used in the system. This will load the task register, local descriptor table register, segment registers and initial general register state. The TR should point at a valid TSS since any task switch operation involves saving the current task state.

SYSTEM INTERFACE

The 80286 system interface appears in two forms: a local bus and a system bus. The local bus consists of address, data, status, and control signals at the pins of the CPU. A system bus is any buffered version of the local bus. A system bus may also differ from the local bus in terms of coding of status and control lines and/or timing and loading of signals. The 80286 family includes several devices to generate standard system buses such as the IEEE 796 standard MULTIBUS.

Bus Interface Signals and Timing

The 80286 microsystem local bus interfaces the 80286 to local memory and I/O components. The interface has 24 address lines, 16 data lines, and 8 status and control signals.

The 80286 CPU, 82C284 clock generator, 82288 bus controller, 82289 bus arbiter, tranceivers, and latches provide a buffered and decoded system bus interface. The 82C284 generates the system clock and synchronizes $\overline{\text{READY}}$ and RESET. The 82288 converts bus operation status encoded by the 80286 into command and bus control signals. The 82289 bus arbiter generates Multibus bus arbitration signals. These components can provide the timing and electrical power drive levels required for most system bus interfaces including the Multibus.

Physical Memory and I/O Interface

A maximum of 16 megabytes of physical memory can be addressed in protected mode. One megabyte can be addressed in real address mode. Memory is accessible as bytes or words. Words consist of any two consecutive bytes addressed with the least significant byte stored in the lowest address.

Byte transfers occur on either half of the 16-bit local data bus. Even bytes are accessed over D_{7-0} while odd bytes are transferred over D_{15-8}. Even-addressed words are transferred over D_{15-0} in one bus cycle, while odd-addressed word require *two* bus operations. The first transfers data on D_{15-8}, and the second transfers data on D_{7-0}. Both byte data transfers occur automatically, transparent to software.

Two bus signals, A_0 and $\overline{BHE}$, control transfers over the lower and upper halves of the data bus. Even address byte transfers are indicated by A_0 LOW and $\overline{BHE}$ HIGH. Odd address byte transfers are indicated by A_0 HIGH and $\overline{BHE}$ LOW. Both A_0 and $\overline{BHE}$ are LOW for even address word transfers.

The I/O address space contains 64K addresses in both modes. The I/O space is accessible as either bytes or words, as is memory. Byte wide peripheral devices may be attached to either the upper or lower byte of the data bus. Byte-wide I/O devices attached to the upper data byte (D_{15-8}) are accessed with odd I/O addresses. Devices on the lower data byte are accessed with even I/O addresses. An interrupt controller such as Intel's 8259A must be connected to the lower data byte (D_{7-0}) for proper return of the interrupt vector.

Bus Operation

The 80286 uses a double frequency system clock (CLK input) to control bus timing. All signals on the local bus are measured relative to the system CLK input. The CPU divides the system clock by 2 to produce the internal processor clock, which determines bus state. Each processor clock is composed of two system clock cycles named phase 1 and phase 2. The 82C284 clock generator output (PCLK) identifies the next phase of the processor clock. (See Figure 21.)

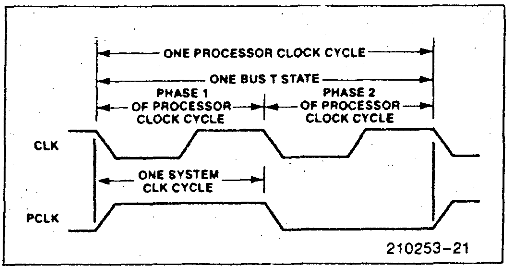

ONE PROCESSOR CLOCK CYCLE
ONE BUS T STATE
PHASE 1 OF PROCESSOR CLOCK CYCLE
PHASE 2 OF PROCESSOR CLOCK CYCLE
CLK
ONE SYSTEM CLK CYCLE
PCLK
210253-21

Figure 21. System and Processor Clock Relationships

Six types of bus operations are supported; memory read, memory write, I/O read, I/O write, interrupt acknowledge, and halt/shutdown. Data can be transferred at a maximum rate of one word per two processor clock cycles.

The 80286 bus has three basic states: idle (T_i), send status (T_S), and perform command (T_C). The 80286 CPU also has a fourth local bus state called hold (T_h). T_h indicates that the 80286 has surrendered control of the local bus to another bus master in response to a HOLD request.

Each bus state is one processor clock long. Figure 22 shows the four 80286 local bus states and allowed transitions.

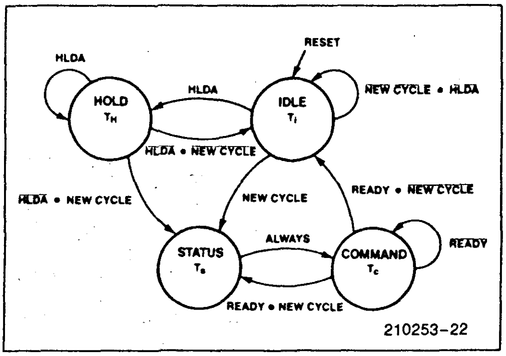

RESET
HLDA
HOLD T_H
HLDA
IDLE T_I
$\overline{\text{NEW CYCLE}}$ • $\overline{\text{HLDA}}$
$\overline{\text{HLDA}}$ • $\overline{\text{NEW CYCLE}}$
$\overline{\text{HLDA}}$ • NEW CYCLE
NEW CYCLE
READY • $\overline{\text{NEW CYCLE}}$
ALWAYS
STATUS T_S
COMMAND T_C
$\overline{\text{READY}}$
READY • NEW CYCLE
210253-22

Figure 22. 80286 Bus States

Bus States

The idle (T_i) state indicates that no data transfers are in progress or requested. The first active state T_S is signaled by status line $\overline{S1}$ or $\overline{S0}$ going LOW and identifying phase 1 of the processor clock. During T_S, the command encoding, the address, and data (for a write operation) are available on the 80286 output pins. The 82288 bus controller decodes the status signals and generates Multibus compatible read/write command and local transceiver control signals.

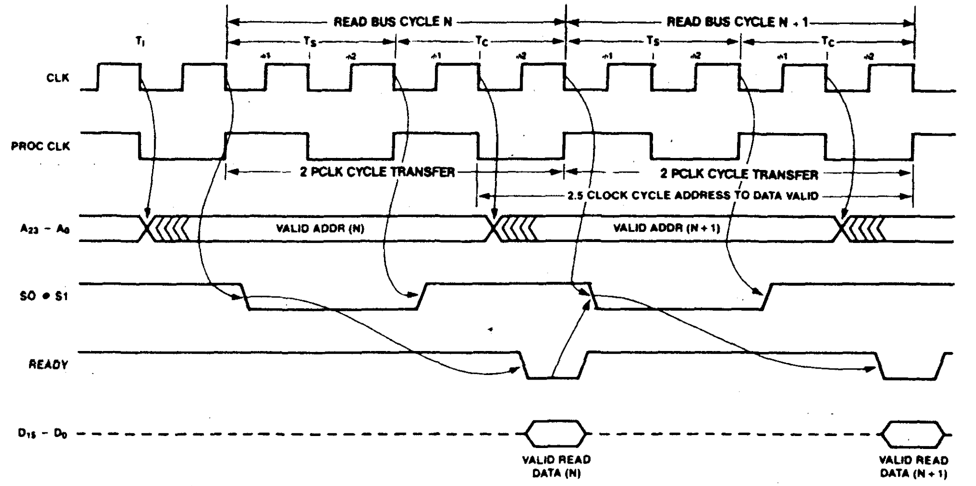
After T_S, the perform command (T_C) state is entered. Memory or I/O devices respond to the bus operation during T_C, either transferring read data to the CPU or accepting write data. T_C states may be repeated as often as necessary to assure sufficient time for the memory or I/O device to respond. The $\overline{READY}$ signal determines whether T_C is repeated. A repeated T_C state is called a wait state.

During hold (T_h), the 80286 will float all address, data, and status output pins enabling another bus master to use the local bus. The 80286 HOLD input signal is used to place the 80286 into the T_h state. The 80286 HLDA output signal indicates that the CPU has entered T_h.

Pipelined Addressing

The 80286 uses a local bus interface with pipelined timing to allow as much time as possible for data access. Pipelined timing allows a new bus operation to be initiated every two processor cycles, while allowing each individual bus operation to last for three processor cycles.

The timing of the address outputs is pipelined such that the address of the next bus operation becomes available during the current bus operation. Or in other words, the first clock of the next bus operation is overlapped with the last clock of the current bus operation. Therefore, address decode and routing logic can operate in advance of the next bus operation.



PIPELINING: VALID ADDRESS (N + 1) AVAILABLE IN LAST PHASE OF BUS CYCLE (N).

210253-23

Figure 23. Basic Bus Cycle

External address latches may hold the address stable for the entire bus operation, and provide additional AC and DC buffering.

The 80286 does not maintain the address of the current bus operation during all T_C states. Instead, the address for the next bus operation may be emitted during phase 2 of any T_C. The address remains valid during phase 1 of the first T_C to guarantee hold time, relative to ALE, for the address latch inputs.

Bus Control Signals

The 82288 bus controller provides control signals; address latch enable (ALE), Read/Write commands, data transmit/receive (DT/$\overline{R}$), and data enable (DEN) that control the address latches, data transceivers, write enable, and output enable for memory and I/O systems.

The Address Latch Enable (ALE) output determines when the address may be latched. ALE provides at least one system CLK period of address hold time from the end of the previous bus operation until the address for the next bus operation appears at the latch outputs. This address hold time is required to support MULTIBUS and common memory systems.

The data bus transceivers are controlled by 82288 outputs Data Enable (DEN) and Data Transmit/Receive (DT/$\overline{R}$). DEN enables the data transceivers; while DT/$\overline{R}$ controls tranceiver direction. DEN and DT/$\overline{R}$ are timed to prevent bus contention between the bus master, data bus transceivers, and system data bus transceivers.

Command Timing Controls

Two system timing customization options, command extension and command delay, are provided on the 80286 local bus.

Command extension allows additional time for external devices to respond to a command and is analogous to inserting wait states on the 8086. External logic can control the duration of any bus operation such that the operation is only as long as necessary. The $\overline{\text{READY}}$ input signal can extend any bus operation for as long as necessary.

Command delay allows an increase of address or write data setup time to system bus command active for any bus operation by delaying when the system bus command becomes active. Command delay is controlled by the 82288 CMDLY input. After T_S, the bus controller samples CMDLY at each failing edge of CLK. If CMDLY is HIGH, the 82288 will not activate the command signal. When CMDLY is LOW, the 82288 will activate the command signal. After the command becomes active, the CMDLY input is not sampled.

When a command is delayed, the available response time from command active to return read data or accept write data is less. To customize system bus timing, an address decoder can determine which bus operations require delaying the command. The CMDLY input does not affect the timing of ALE, DEN, or DT/$\overline{R}$.

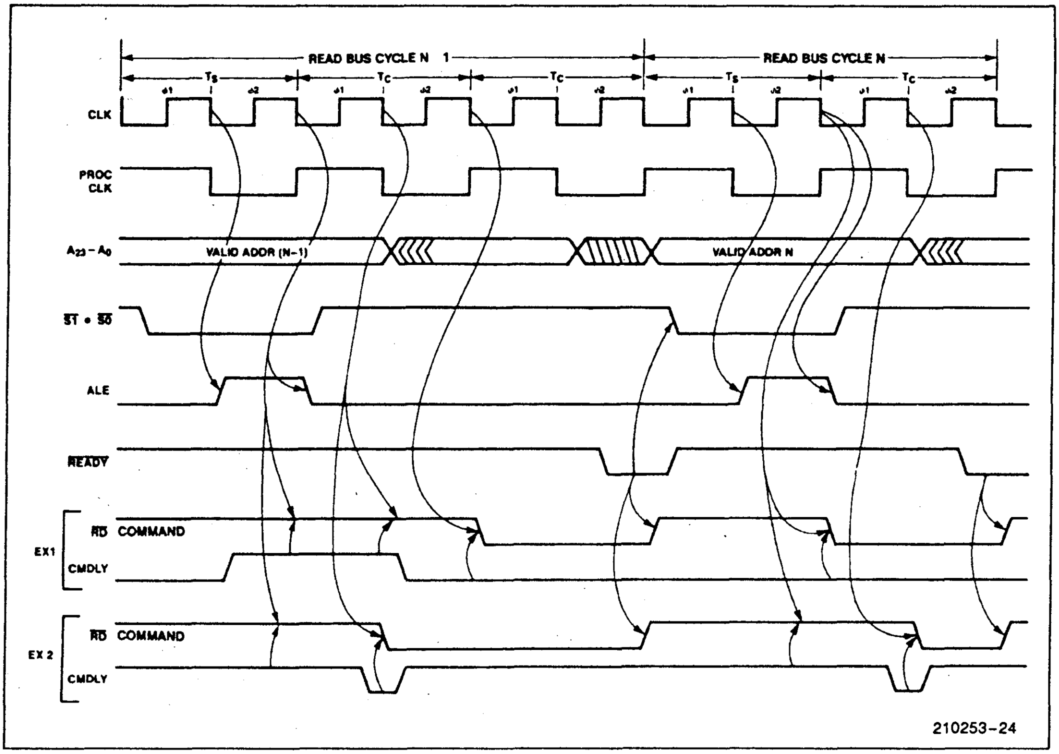


Figure 24. CMDLY Controls the Leading Edge of Command Signal

Figure 24 illustrates four uses of CMDLY. Example 1 shows delaying the read command two system CLKs for cycle N-1 and no delay for cycle N, and example 2 shows delaying the read command one system CLK for cycle N-1 and one system CLK delay for cycle N.

Bus Cycle Termination

At maximum transfer rates, the 80286 bus alternates between the status and command states. The bus status signals become inactive after T_S so that they may correctly signal the start of the next bus operation after the completion of the current cycle. No external indication of T_C exists on the 80286 local bus. The bus master and bus controller enter T_C directly after T_S and continue executing T_C cycles until terminated by READY.

READY Operation

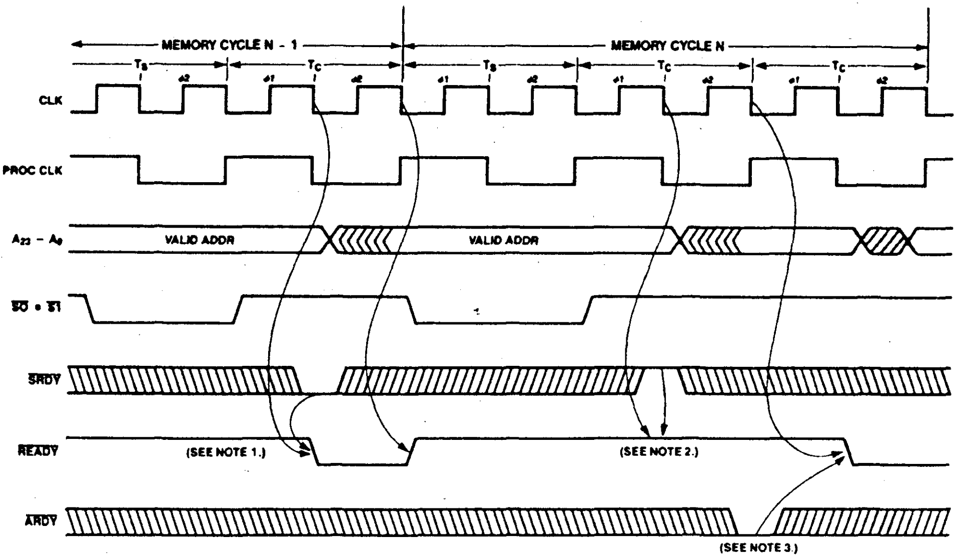
The current bus master and 82288 bus controller terminate each bus operation simultaneously to achieve maximum bus operation bandwidth. Both are informed in advance by READY active (open-collector output from 82C284) which identifies the last T_C cycle of the current bus operation. The bus master and bus controller must see the same sense of the READY signal, thereby requiring READY be synchronous to the system clock.

Synchronous Ready

The 82C284 clock generator provides READY synchronization from both synchronous and asynchronous sources (see Figure 25). The synchronous ready input (SRDY) of the clock generator is sampled with the falling edge of CLK at the end of phase 1 of each T_C. The state of SRDY is then broadcast to the bus master and bus controller via the READY output line.

Asynchronous Ready

Many systems have devices or subsystems that are asynchronous to the system clock. As a result, their ready outputs cannot be guaranteed to meet the 82C284 SRDY setup and hold time requirements. But the 82C284 asynchronous ready input (ARDY) is designed to accept such signals. The ARDY input is sampled at the beginning of each T_C cycle by 82C284 synchronization logic. This provides one system CLK cycle time to resolve its value before broadcasting it to the bus master and bus controller.



210253-25

NOTES:

1. $\overline{\text{SRDYEN}}$ is active low.
2. If $\overline{\text{SRDYEN}}$ is high, the state of $\overline{\text{SRDY}}$ will no affect $\overline{\text{READY}}$.
3. $\overline{\text{ARDYEN}}$ is active low.

Figure 25. Synchronous and Asynchronous Ready

$\overline{\text{ARDY}}$ or $\overline{\text{ARDYEN}}$ must be HIGH at the end of T_S. $\overline{\text{ARDY}}$ cannot be used to terminate bus cycle with no wait states.

Each ready input of the 82C284 has an enable pin ($\overline{\text{SRDYEN}}$ and $\overline{\text{ARDYEN}}$) to select whether the current bus operation will be terminated by the synchronous or asynchronous ready. Either of the ready inputs may terminate a bus operation. These enable inputs are active low and have the same timing as their respective ready inputs. Address decode logic usually selects whether the current bus operation should be terminated by $\overline{\text{ARDY}}$ or $\overline{\text{SRDY}}$.

Data Bus Control

Figures 26, 27, and 28 show how the DT/$\overline{\text{R}}$, DEN, data bus, and address signals operate for different combinations of read, write, and idle bus operations. DT/$\overline{\text{R}}$ goes active (LOW) for a read operation. DT/$\overline{\text{R}}$ remains HIGH before, during, and between write operations.

The data bus is driven with write data during the second phase of T_S. The delay in write data timing allows the read data drivers, from a previous read cycle, sufficient time to enter 3-state OFF before the 80286 CPU begins driving the local data bus for write operations. Write data will always remain valid for one system clock past the last T_C to provide sufficient hold time for Multibus or other similar memory or I/O systems. During write-read or write-idle sequences the data bus enters 3-state OFF during the second phase of the processor cycle after the last T_C. In a write-write sequence the data bus does not enter 3-state OFF between T_C and T_S.

Bus Usage

The 80286 local bus may be used for several functions: instruction data transfers, data transfers by other bus masters, instruction fetching, processor extension data transfers, interrupt acknowledge, and halt/shutdown. This section describes local bus activities which have special signals or requirements.

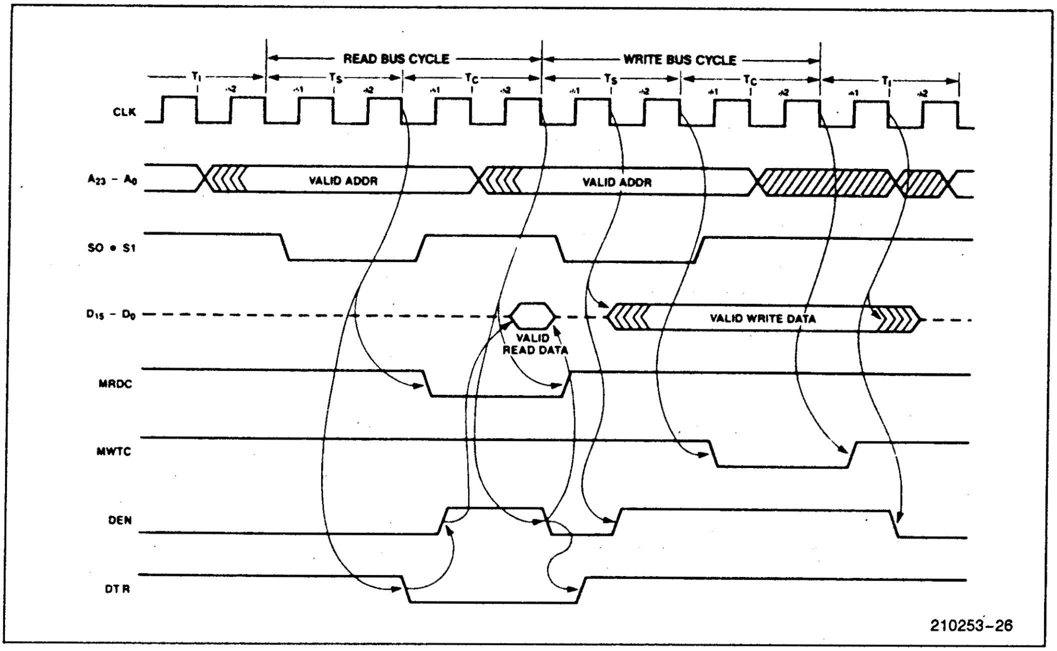


Figure 26. Back to Back Read-Write Cycles

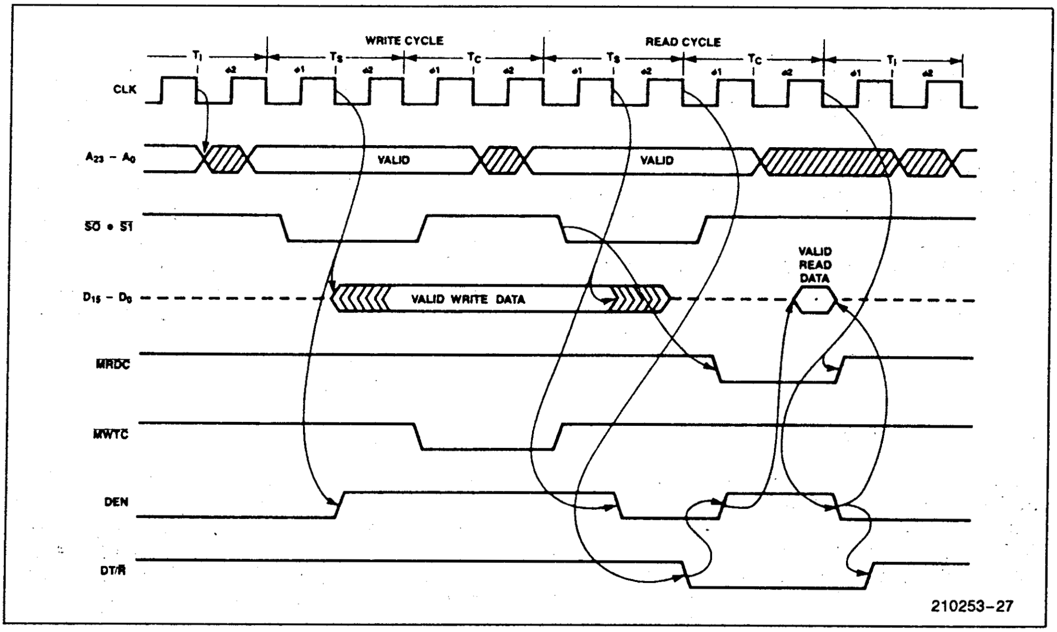


Figure 27. Back to Back Write-Read Cycles

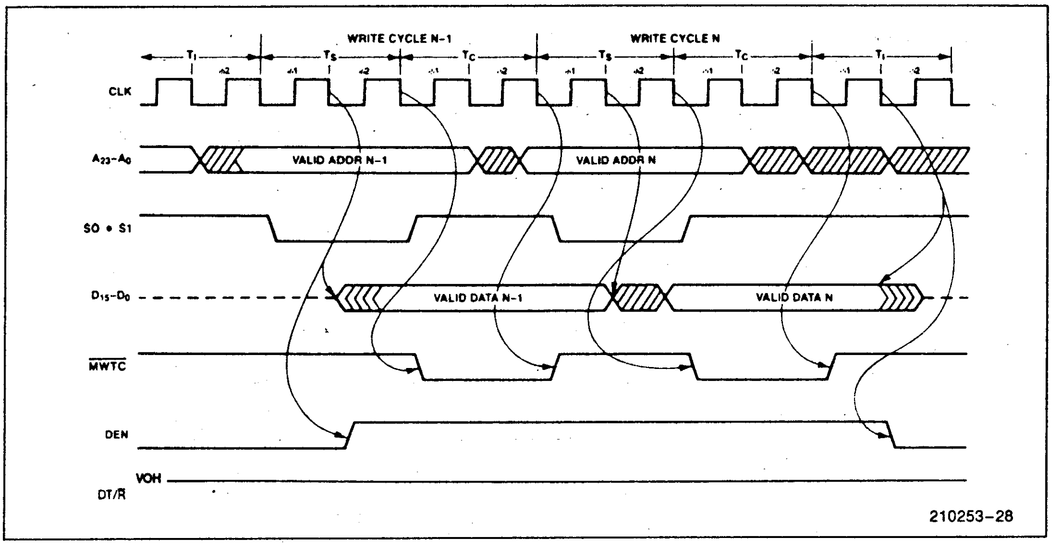


Figure 28. Back to Back Write-Write Cycles

HOLD and HLDA

HOLD AND HLDA allow another bus master to gain control of the local bus by placing the 80286 bus into the T_h state. The sequence of events required to pass control between the 80286 and another local bus master are shown in Figure 29.

In this example, the 80286 is initially in the T_h state as signaled by HLDA being active. Upon leaving T_h, as signaled by HLDA going inactive, a write operation is started. During the write operation another local bus master requests the local bus from the 80286 as shown by the HOLD signal. After completing the write operation, the 80286 performs one T_i bus cycle, to guarantee write data hold time, then enters T_h as signaled by HLDA going active.

The CMDLY signal and $\overline{\text{ARDY}}$ ready are used to start and stop the write bus command, respectively. Note that $\overline{\text{SRDY}}$ must be inactive or disabled by $\overline{\text{SRDYEN}}$ to guarantee $\overline{\text{ARDY}}$ will terminate the cycle.

HOLD must not be active during the time from the leading edge of RESET until 34 CLKs following the trailing edge of RESET.

Lock

The CPU asserts an active lock signal during Interrupt-Acknowledge cycles, the XCHG instruction, and during some descriptor accesses. Lock is also asserted when the LOCK prefix is used. The LOCK prefix may be used with the following ASM-286 assembly instructions; MOVS, INS, and OUTS. For bus cycles other than Interrupt-Acknowledge cycles, Lock will be active for the first and subsequent cycles of a series of cycles to be locked. Lock will not be shown active during the last cycle to be locked. For the next-to-last cycle, Lock will become inactive at the end of the first T_c regardless of the number of wait-states inserted. For Interrupt-Acknowledge cycles, Lock will be active for each cycle, and will become inactive at the end of the first T_c for each cycle regardless of the number of wait-states inserted.

Instruction Fetching

The 80286 Bus Unit (BU) will fetch instructions ahead of the current instruction being executed. This activity is called prefetching. It occurs when the local bus would otherwise be idle and obeys the following rules:

A prefetch bus operation starts when at least two bytes of the 6-byte prefetch queue are empty.

The prefetcher normally performs word prefetches independent of the byte alignment of the code segment base in physical memory.

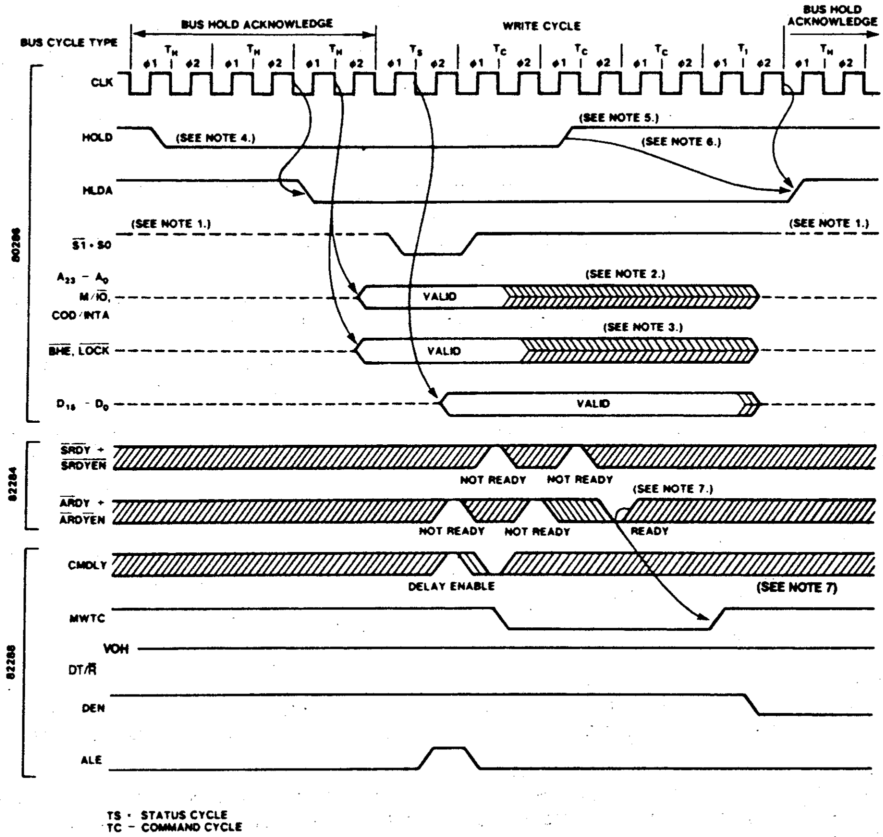
The prefetcher will perform only a byte code fetch operation for control transfers to an instruction beginning on a numerically odd physical address.

Prefetching stops whenever a control transfer or HLT instruction is decoded by the IU and placed into the instruction queue.

In real address mode, the prefetcher may fetch up to 6 bytes beyond the last control transfer or HLT instruction in a code segment.

In protected mode, the prefetcher will never cause a segment overrun exception. The prefetcher stops at the last physical memory word of the code segment. Exception 13 will occur if the program attempts to execute beyond the last full instruction in the code segment.

If the last byte of a code segment appears on an even physical memory address, the prefetcher will read the next physical byte of memory (perform a word code fetch). The value of this byte is ignored and any attempt to execute it causes exception 13.



210253-29

NOTES:
1. Status lines are not driven by 80286, yet remain high due to pullup resistors in 82288 and 82289 during HOLD state.
2. Address, M/$\overline{IO}$ and COD/$\overline{INTA}$ may start floating during any T_C depending on when internal 80286 bus arbiter decides to release bus to external HOLD. The float starts in φ2 of T_C.
3. $\overline{BHE}$ and $\overline{LOCK}$ may start floating after the end of any T_C depending on when internal 80286 bus arbiter decides to release bus to external HOLD. The float starts in φ1 of T_C.
4. The minimum HOLD to HLDA time is shown. Maximum is one T_H longer.
5. The earliest HOLD time is shown. It will always allow a subsequent memory cycle if pending is shown.
6. The minimum HOLD to HLDA time is shown. Maximum is a function of the instruction, type of bus cycle and other machine state (i.e., Interrupts, Waits, Lock, etc.).
7. Asynchronous ready allows termination of the cycle. Synchronous ready does not signal ready in this example. Synchronous ready state is ignored after ready is signaled via the asynchronous input.

Figure 29. MULTIBUS® Write Terminated by Asynchronous Ready with Bus Hold

Processor Extension Transfers

The processor extension interface uses I/O port addresses 00F8(H), 00FA(H), and 00FC(H) which are part of the I/O port address range reserved by Intel. An ESC instruction with Machine Status Word bits EM = 0 and TS = 0 will perform I/O bus operations to one or more of these I/O port addresses independent of the value of IOPL and CPL.

ESC instructions with memory references enable the CPU to accept PEREQ inputs for processor extension operand transfers. The CPU will determine the operand starting address and read/write status of the instruction. For each operand transfer, two or three bus operations are performed, one word transfer with I/O port address 00FA(H) and one or two bus operations with memory. Three bus operations are required for each word operand aligned on an odd byte address.

NOTE:

Odd-aligned numerics operands should be avoided when using an 80286 system running six or more memory-write wait states. The 80286 can generate an incorrect numerics address if all the following conditions are met:

- Two floating point (FP) instructions are fetched and in the 80286 queue.
- The first FP instruction is any floating point store except FSTSW AX.
- The second FP instruction accesses memory.
- The operand of the first instruction is aligned on an odd memory address.
- Six or more wait states are inserted during either of the last two memory write (odd aligned operands are transferred as two bytes) transfers of the first instruction.

The second FP operand's address will be incremented by one if these conditions are met. These conditions are most likely to occur in a multi-master system. For a hardware solution, contact your local Intel representative.

Commands to the numerics coprocessor should not be delayed by nine or more T-states. Excessive (nine or more) command-delays can cause the 80286 and 80287 to lose synchronization.

Interrupt Acknowledge Sequence

Figure 30 illustrates an interrupt acknowledge sequence performed by the 80286 in response to an INTR input. An interrupt acknowledge sequence consists of two INTA bus operations. The first allows a master 8259A Programmable Interrupt Controller (PIC) to determine which if any of its slaves should return the interrupt vector. An eight bit vector is read on D0–D7 of the 80286 during the second INTA bus operation to select an interrupt handler routine from the interrupt table.

The Master Cascade Enable (MCE) signal of the 82288 is used to enable the cascade address drivers, during INTA bus operations (See Figure 30), onto the local address bus for distribution to slave interrupt controllers via the system address bus. The 80286 emits the $\overline{\text{LOCK}}$ signal (active LOW) during T_S of the first INTA bus operation. A local bus "hold" request will not be honored until the end of the second INTA bus operation.

Three idle processor clocks are provided by the 80286 between INTA bus operations to allow for the minimum INTA to INTA time and CAS (cascade address) out delay of the 8259A. The second INTA bus operation must always have at least one extra T_C state added via logic controlling $\overline{\text{READY}}$. This is needed to meet the 8259A minimum INTA pulse width.

Local Bus Usage Priorities

The 80286 local bus is shared among several internal units and external HOLD requests. In case of simultaneous requests, their relative priorities are:

(Highest) Any transfers which assert $\overline{\text{LOCK}}$ either explicitly (via the LOCK instruction prefix) or implicitly (i.e. some segment descriptor accesses, interrupt acknowledge sequence, or an XCHG with memory).

The second of the two byte bus operations required for an odd aligned word operand.

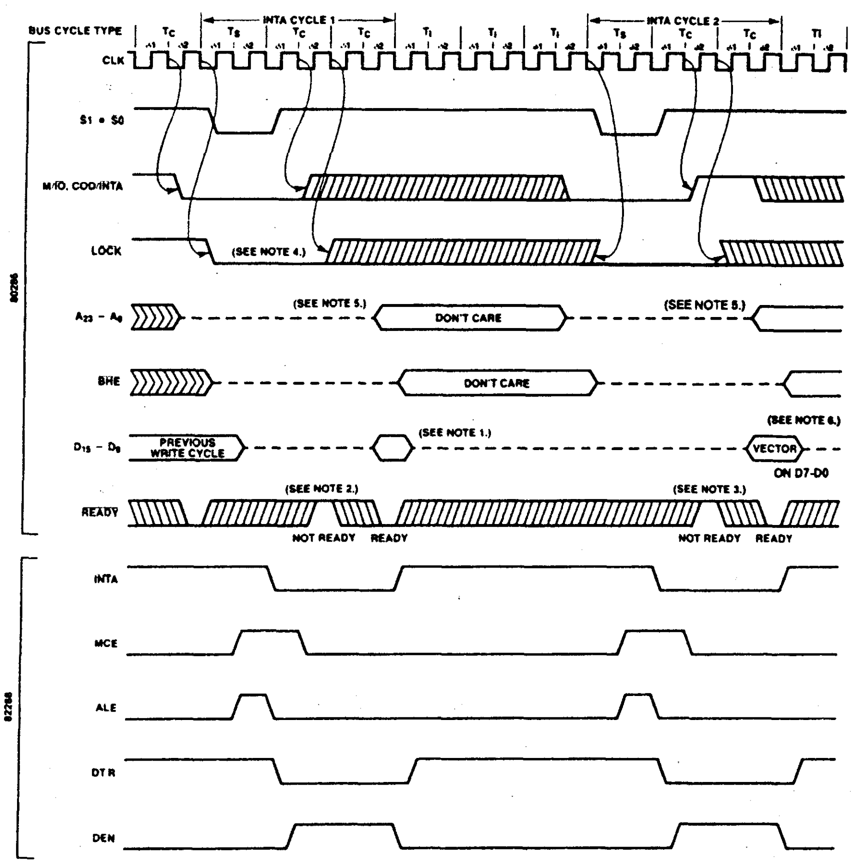
The second or third cycle of a processor extension data transfer.

Local bus request via HOLD input.

Processor extension data operand transfer via PEREQ input.

Data transfer performed by EU as part of an instruction.

(Lowest) An instruction prefetch request from BU. The EU will inhibit prefetching two processor clocks in advance of any data transfers to minimize waiting by EU for a prefetch to finish.



210253-31

NOTES:

1. Data is ignored, upper data bus, D_8–D_{15}, should not change state during this time.
2. First INTA cycle should have at least one wait state inserted to meet 8259A minimum INTA pulse width.
3. Second INTA cycle should have at least one wait state inserted to meet 8259A minimum INTA pulse width.
4. $\overline{\text{LOCK}}$ is active for the first INTA cycle to prevent the 82289 from releasing the bus between INTA cycles in a multi-master system. $\overline{\text{LOCK}}$ is also active for the second INTA cycle.
5. A_{23}–A_0 exits 3-state OFF during $\phi 2$ of the second T_C in the INTA cycle.
6. Upper data bus should not change state during this time.

Figure 30. Interrupt Acknowledge Sequence

Halt or Shutdown Cycles

The 80286 externally indicates halt or shutdown conditions as a bus operation. These conditions occur due to a HLT instruction or multiple protection exceptions while attempting to execute one instruction. A halt or shutdown bus operation is signalled when $\overline{\text{S1}}$, $\overline{\text{S0}}$ and COD/$\overline{\text{INTA}}$ are LOW and M/$\overline{\text{IO}}$ is HIGH. A_1 HIGH indicates halt, and A_1 LOW indicates shutdown. The 82288 bus controller does not issue ALE, nor is $\overline{\text{READY}}$ required to terminate a halt or shutdown bus operation.

During halt or shutdown, the 80286 may service PEREQ or HOLD requests. A processor extension segment overrun exception during shutdown will inhibit further service of PEREQ. Either NMI or RESET will force the 80286 out of either halt or shutdown. An INTR, if interrupts are enabled, or a processor extension segment overrun exception will also force the 80286 out of halt.

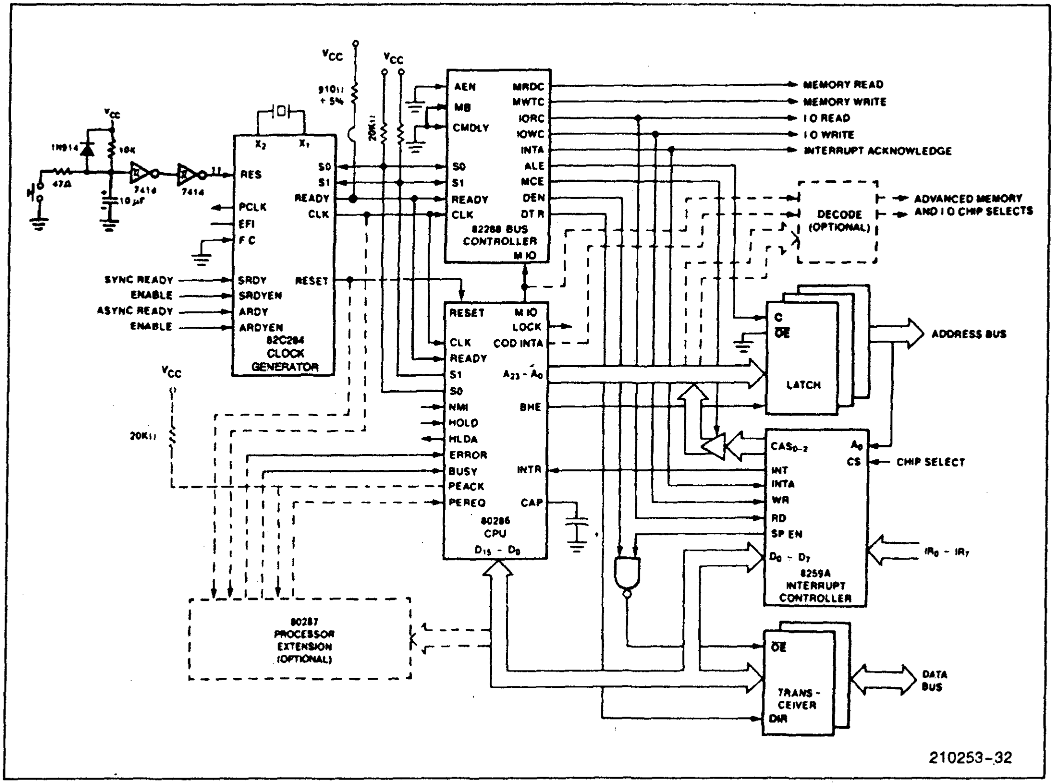


Figure 31. Basic 80286 System Configuration

SYSTEM CONFIGURATIONS

The versatile bus structure of the 80286 microsystem, with a full complement of support chips, allows flexible configuration of a wide range of systems. The basic configuration, shown in Figure 31, is similar to an 8086 maximum mode system. It includes the CPU plus an 8259A interrupt controller, 82C284 clock generator, and the 82288 Bus Controller.

As indicated by the dashed lines in Figure 31, the ability to add processor extensions is an integral feature of 80286 microsystems. The processor extension interface allows external hardware to perform special functions and transfer data concurrent with CPU execution of other instructions. Full system integrity is maintained because the 80286 supervises all data transfers and instruction execution for the processor extension.

The 80287 has all the instructions and data types of an 8087. The 80287 NPX can perform numeric calculations and data transfers concurrently with CPU program execution. Numerics code and data have the same integrity as all other information protected by the 80286 protection mechanism.

The 80286 can overlap chip select decoding and address propagation during the data transfer for the previous bus operation. This information is latched by ALE during the middle of a T_S cycle. The latched chip select and address information remains stable during the bus operation while the next cycle's address is being decoded and propagated into the system. Decode logic can be implemented with a high speed bipolar PROM.

The optional decode logic shown in Figure 31 takes advantage of the overlap between address and data of the 80286 bus cycle to generate advanced memory and IO-select signals. This minimizes system

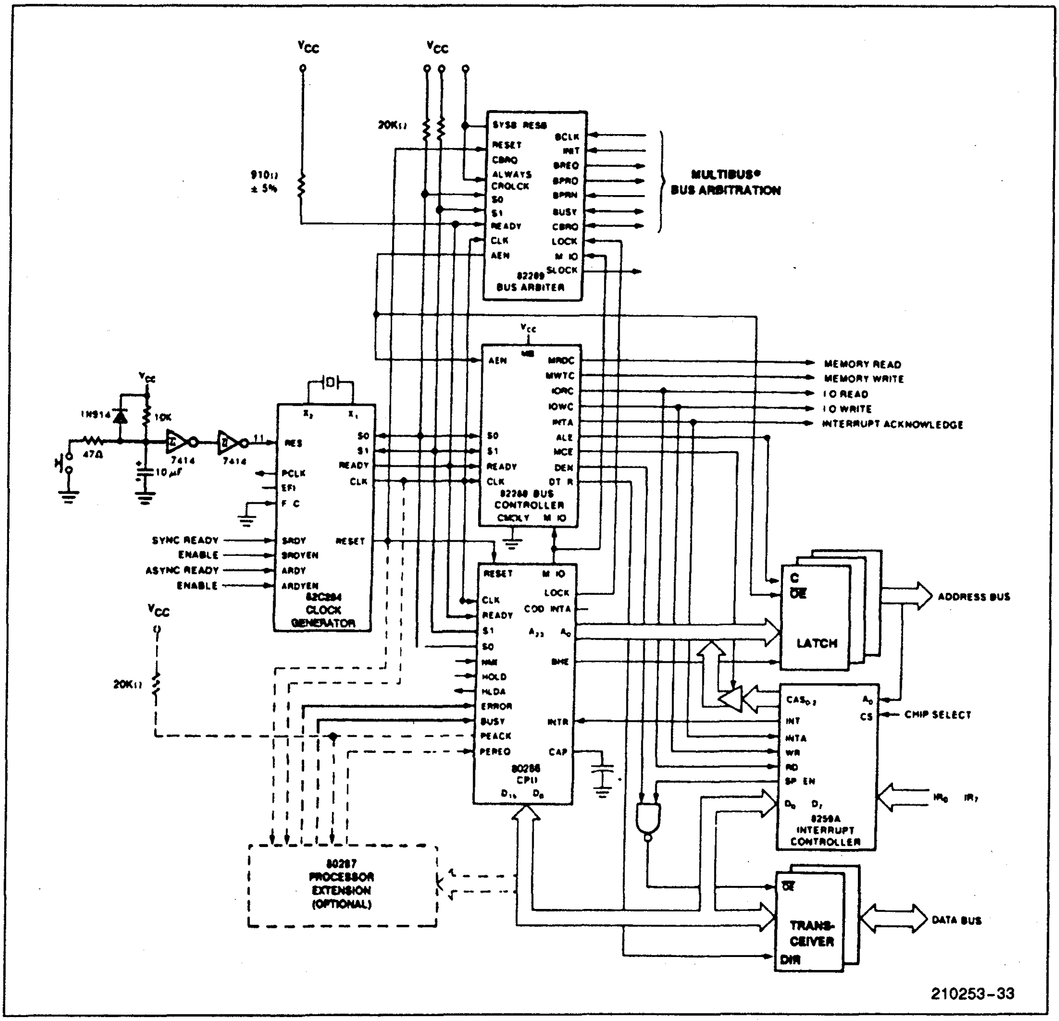
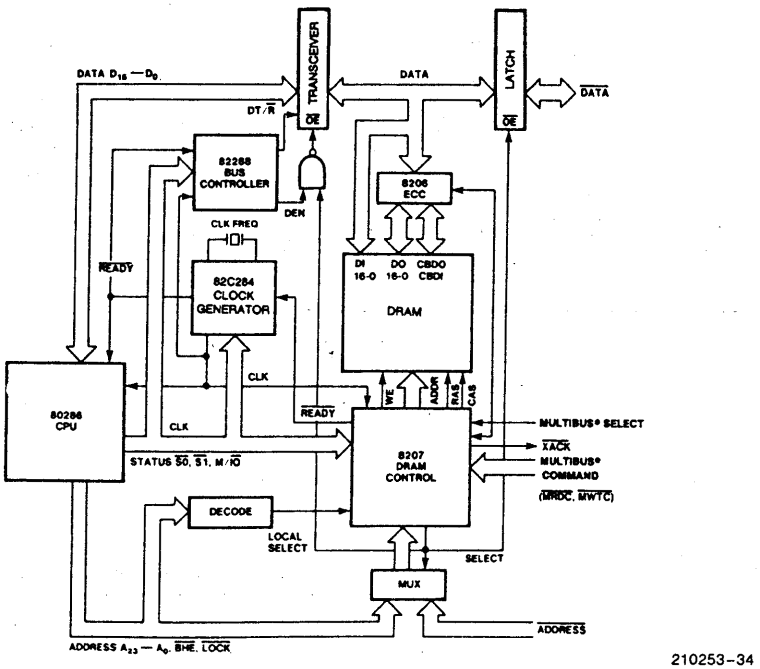


Figure 32. MULTIBUS® System Bus Interface

performance degradation caused by address propagation and decode delays. In addition to selecting memory and I/O, the advanced selects may be used with configurations supporting local and system buses to enable the appropriate bus interface for each bus cycle. The COD/$\overline{\text{INTA}}$ and M/$\overline{\text{IO}}$ signals are applied to the decode logic to distinguish between interrupt, I/O, code and data bus cycles.

By adding the 82289 bus arbiter chip, the 80286 provides a MULTIBUS system bus interface as shown in Figure 32. The ALE output of the 82288 for the MULTIBUS bus is connected to its CMDLY input to delay the start of commands one system CLK as required to meet MULTIBUS address and write data setup times. This arrangement will add at least one extra T_C state to each bus operation which uses the MULTIBUS.

A second 82288 bus controller and additional latches and transceivers could be added to the local bus of Figure 32. This configuration allows the 80286 to support an on-board bus for local memory and peripherals, and the MULTIBUS for system bus interfacing.



210253-34

Figure 33. 80286 System Configuration with Dual-Ported Memory

Figure 33 shows the addition of dual ported dynamic memory between the MULTIBUS system bus and the 80286 local bus. The dual port interface is provided by the 8207 Dual Port DRAM Controller. The 8207 runs synchronously with the CPU to maximize throughput for local memory references. It also arbitrates between requests from the local and system buses and performs functions such as refresh, initialization of RAM, and read/modify/write cycles. The 8207 combined with the 8206 Error Checking and Correction memory controller provide for single bit error correction. The dual-ported memory can be combined with a standard MULTIBUS system bus interface to maximize performance and protection in multiprocessor system configurations.

Table 16. 80286 Systems Recommended Pull Up Resistor Values

80286 Pin and Name	Pullup Value	Purpose
4—$\overline{S1}$	20 KΩ ±10%	Pull $\overline{S0}$, $\overline{S1}$, and $\overline{PEACK}$ inactive during 80286 hold periods[1]
5—$\overline{S0}$		
6—$\overline{PEACK}$		
63—$\overline{READY}$	910Ω ±5%	Pull $\overline{READY}$ inactive within required minimum time (C_L = 150 pF, $I_R \leq 7$ mA)

NOTE:
1. Pull-up resistors are not required on $\overline{S0}$ and $\overline{S1}$ when the corresponding pins of the 82C284 are connected to $\overline{S0}$ and $\overline{S1}$.

I²ICE™-286 System Design Considerations

One of the advantages of using the 80286 is that full in-circuit emulation debugging support is provided through the I²ICE system 80286 probe. To utilize this powerful tool it is necessary that the system designer be aware of a few minor parametric and functional differences between the 80286 and I²ICE system 80286 probe. The I²ICE data sheet (I²ICE Integrated Instrumentation and In-Circuit Emulation System, order #210469) contains a detailed description of these design considerations. It is recommended that this document be reviewed by the 80286 system designer to determine whether or not these differences affect his design.

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	−1.0V to +7V
Power Dissipation	3.3W

**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} = 5V ±5%, T_{CASE} = 0°C to +85°C)*

Symbol	Parameter	Min	Max	Unit	Test Condition
I_{CC}	Supply Current (0°C Turn On)		600	mA	(Note 1)
C_{CLK}	CLK Input Capacitance		20	pF	(Note 2)
C_{IN}	Other Input Capacitance		10	pF	(Note 2)
C_O	Input/Output Capacitance		20	pF	(Note 2)

NOTES:

1. Tested at worst case load and maximum frequency.
2. These are not tested. They are guaranteed by design characterization.

D.C. CHARACTERISTICS

(V_{CC} = 5V ±5%, T_{CASE} = 0°C to +85°C)* Tested at the minimum operating frequency of the part.

Symbol	Parameter	Min	Max	Unit	Test Condition
V_{IL}	Input LOW Voltage	−0.5	0.8	V	
V_{IH}	Input HIGH Voltage	2.0	V_{CC} +0.5	V	
V_{ILC}	CLK Input LOW Voltage	−0.5	0.6	V	
V_{IHC}	CLK Input HIGH Voltage	3.8	V_{CC} +0.5	V	
V_{OL}	Output LOW Voltge		0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output HIGH Voltage	2.4		V	I_{OH} = −400.0 μA
I_{LI}	Input Leakage Current		±10	μA	$0V \leq V_{IN} \leq V_{CC}$
I_{LCR}	Input CLK, RESET Leakage Current		±10	μA	$0.45 \leq V_{IN} \leq V_{CC}$
I_{LCR}	Input CLK, RESET Leakage Current		±1	mA	$0 \leq V_{IN} < 0.45$
I_{IL}	Input Sustaining Current on $\overline{\text{BUSY}}$ and $\overline{\text{ERROR}}$ Pins	30	500	μA	V_{IN} = 0V
I_{LO}	Output Leakage Current		±10	μA	$0.45 \leq V_{OUT} \leq V_{CC}$
I_{LO}	Output Leakage Current		±1	mA	$0 \leq V_{OUT} < 0.45$

*T_A is guaranteed from 0°C to +55°C as long as T_{CASE} is not exceeded.

A.C. CHARACTERISTICS (V_{CC} = 5V ±5%, T_{CASE} = 0°C to +85°C)*

AC timings are referenced to 0.8V and 2.0V points of signals as illustrated in datasheet waveforms, unless otherwise noted.

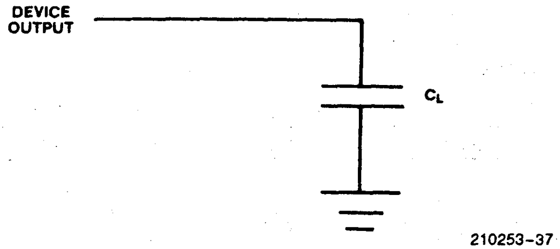
Symbol	Parameter	8 MHz		10 MHz		12.5 MHz (Preliminary)		Unit	Test Condition
		-8 Min	-8 Max	-10 Min	-10 Max	-12 Min	-12 Max		
1	System Clock (CLK) Period	62	250	50	250	40	250	ns	
2	System Clock (CLK) LOW Time	15	225	12	232	11	237	ns	at 1.0V
3	System Clock (CLK) HIGH Time	25	235	16	239	13	239	ns	at 3.6V
17	System Clock (CLK) Rise Time		10		8	—	8	ns	1.0V to 3.6V, (Note 7)
18	System Clock (CLK) Fall Time		10		8	—	8	ns	3.6V to 1.0V, (Note 7)
4	Asynch. Inputs Setup Time	20		20		15		ns	(Note 1)
5	Asynch. Inputs Hold Time	20		20		15		ns	(Note 1)
6	RESET Setup Time	28		23		18		ns	
7	RESET Hold Time	5		5		5		ns	
8	Read Data Setup Time	10		8		5		ns	
9	Read Data Hold Time	8		8		6		ns	
10	$\overline{\text{READY}}$ Setup Time	38		26		22		ns	
11	$\overline{\text{READY}}$ Hold Time	25		25		20		ns	
12	Status/$\overline{\text{PEACK}}$ Valid Delay	1	40	—	—	—	—	ns	(Notes 2, 3)
12a1	Status Active Delay	—	—	1	22	3	18	ns	(Notes 2, 3)
12a2	$\overline{\text{PEACK}}$ Active Delay	—	—	1	22	3	20	ns	(Notes 2, 3)
12b	Status/$\overline{\text{PEACK}}$ Inactive Delay	—	—	1	30	3	22	ns	(Notes 2, 3)
13	Address Valid Delay	1	60	1	35	1	32	ns	(Notes 2, 3)
14	Write Data Valid Delay	0	50	0	30	0	30	ns	(Notes 2, 3)
15	Address/Status/Data Float Delay	0	50	0	47	0	32	ns	(Notes 2, 4, 7)
16	HLDA Valid Delay	0	50	0	47	0	27	ns	(Notes 2, 3)
19	Address Valid To Status Valid Setup Time	38		27		22		ns	(Notes 3, 5, 6)

*T_A is guaranteed from 0°C to +55°C as long as T_{CASE} is not exceeded.

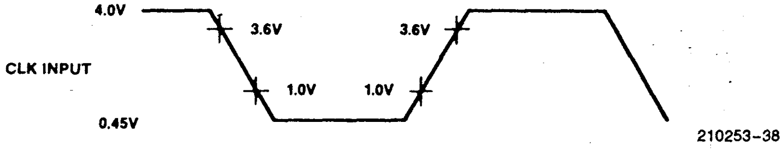
NOTES:

1. Asynchronous inputs are INTR, NMI, HOLD, PEREQ, $\overline{\text{ERROR}}$, and $\overline{\text{BUSY}}$. This specification is given only for testing purposes, to assure recognition at a specific CLK edge.
2. Delay from 1.0V on the CLK, to 0.8V or 2.0V or float on the output as appropriate for valid or floating condition.
3. Output load: C_L = 100 pF.
4. Float condition occurs when output current is less than I_{LO} in magnitude.
5. Delay measured from address either reaching 0.8V or 2.0V (valid) to status going active reaching 2.0V or status going inactive reaching 0.8V.
6. For load capacitance of 10 pF or more on STATUS/$\overline{\text{PEACK}}$ lines, subtract typically 7 ns for 8 MHz, 10 MHz and 12.5 MHz spec.
7. These are not tested. They are guaranteed by design characterization.

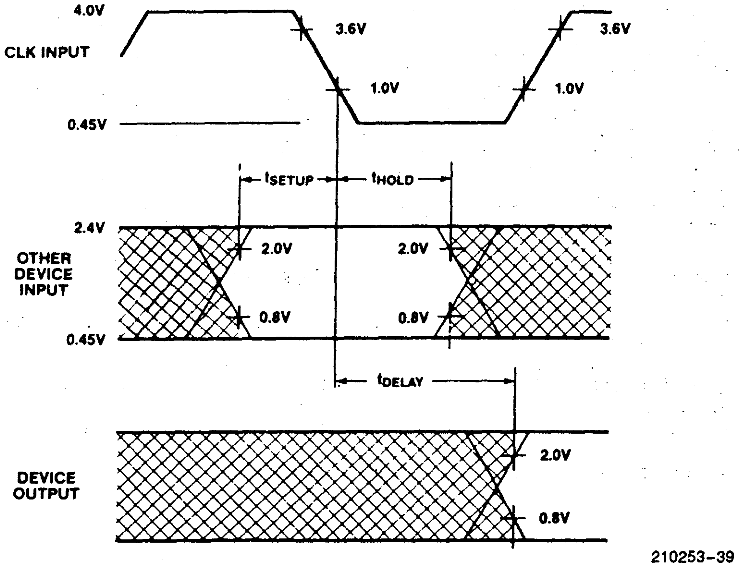
A.C. CHARACTERISTICS (Continued)



NOTE 8:
AC Test Loading on Outputs



NOTE 9:
AC Drive and Measurement Points—CLK Input



NOTE 10:
AC Setup, Hold and Delay Time Measurement—General

A.C. CHARACTERISTICS (Continued)

82C284 Timing Requirements

Symbol	Parameter	82C284-8		82C284-10		82C284-12		Units	Test Conditions
		Min	Max	Min	Max	Min	Max		
11	$\overline{\text{SRDY}}$/$\overline{\text{SRDYEN}}$ Setup Time	17		15		15		ns	
12	$\overline{\text{SRDY}}$/$\overline{\text{SRDYEN}}$ Hold Time	0		2		2		ns	
13	$\overline{\text{ARDY}}$/$\overline{\text{ARDYEN}}$ Setup Time	0		0		0		ns	(Note 1)
14	$\overline{\text{ARDY}}$/$\overline{\text{ARDYEN}}$ Hold Time	30		30		25		ns	(Note 1)
19	PCLK Delay	0	45	0	35	0	23	ns	C_L = 75 pF I_{OL} = 5 mA I_{OH} = −1 mA

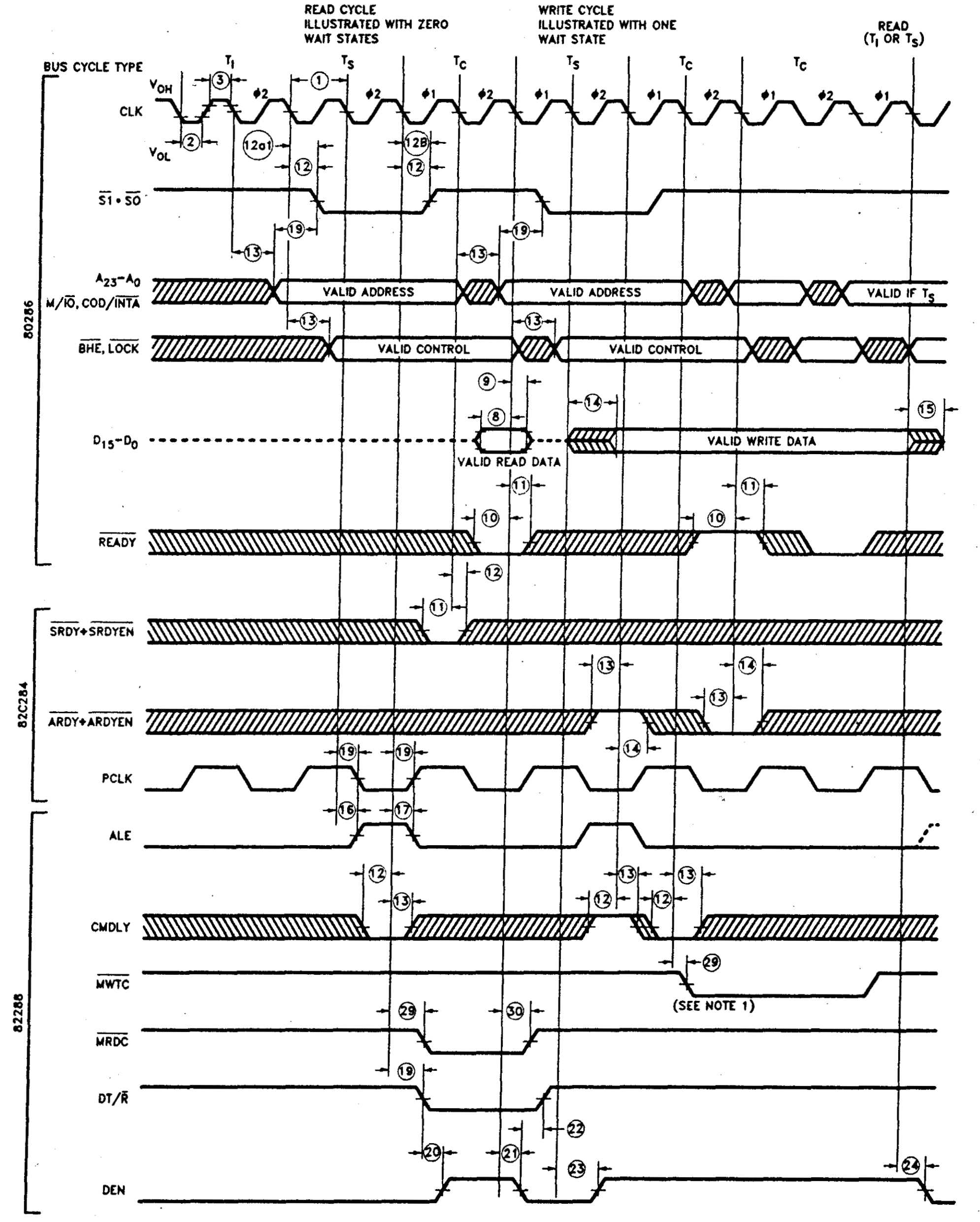
NOTE 1:
These times are given for testing purposes to assure a predetermined action.

82288 Timing Requirements

Symbol	Parameter		82288-8		82288-10		82288-12		Units	Test Conditions
			Min	Max	Min	Max	Min	Max		
12	CMDLY Setup Time		20		15		15		ns	
13	CMDLY Hold Time		1		1		1		ns	
30	Command Delay from CLK	Command Inactive	5	20	5	20	5	20	ns	C_L = 300 pF max I_{OL} = 32 mA max I_{OH} = −5 mA max
29		Command Active	3	25	3	21	3	21		
16	ALE Active Delay		3	20	3	16	3	16	ns	C_L = 150 pF I_{OL} = 16 mA max I_{OH} = −1 mA max
17	ALE Inactive Delay			25		19		19	ns	
19	DT/$\overline{\text{R}}$ Read Active Delay			25		23		23	ns	
22	DT/$\overline{\text{R}}$ Read Inactive Delay		5	35	5	20	5	18	ns	
20	DEN Read Active Delay		5	35	5	21	5	21	ns	
21	DEN Read Inactive Delay		3	35	3	21	3	19	ns	
23	DEN Write Active Delay			30		23		23	ns	
24	DEN Write Inactive Delay		3	30	3	19	3	19	ns	

WAVEFORMS

MAJOR CYCLE TIMING

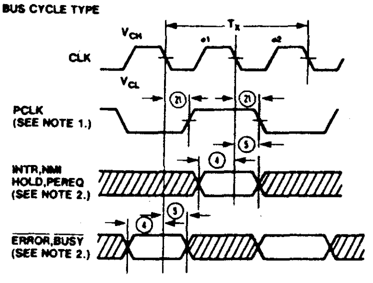


210253-40

NOTE:
1. The modified timing is due to the $\overline{CMDLY}$ signal being active.

WAVEFORMS (Continued)

80286 ASYNCHRONOUS INPUT SIGNAL TIMING

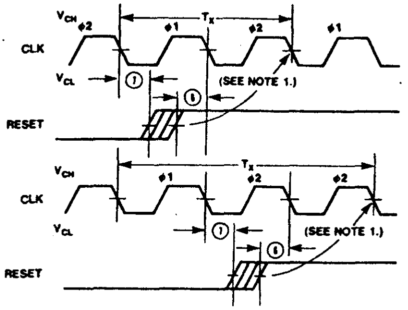


210253-41

NOTES:

1. PCLK indicates which processor cycle phase will occur on the next CLK. PCLK may not indicate the correct phase until the first bus cycle is performed.
2. These inputs are asynchronous. The setup and hold times shown assure recognition for testing purposes.

80286 RESET INPUT TIMING AND SUBSEQUENT PROCESSOR CYCLE PHASE

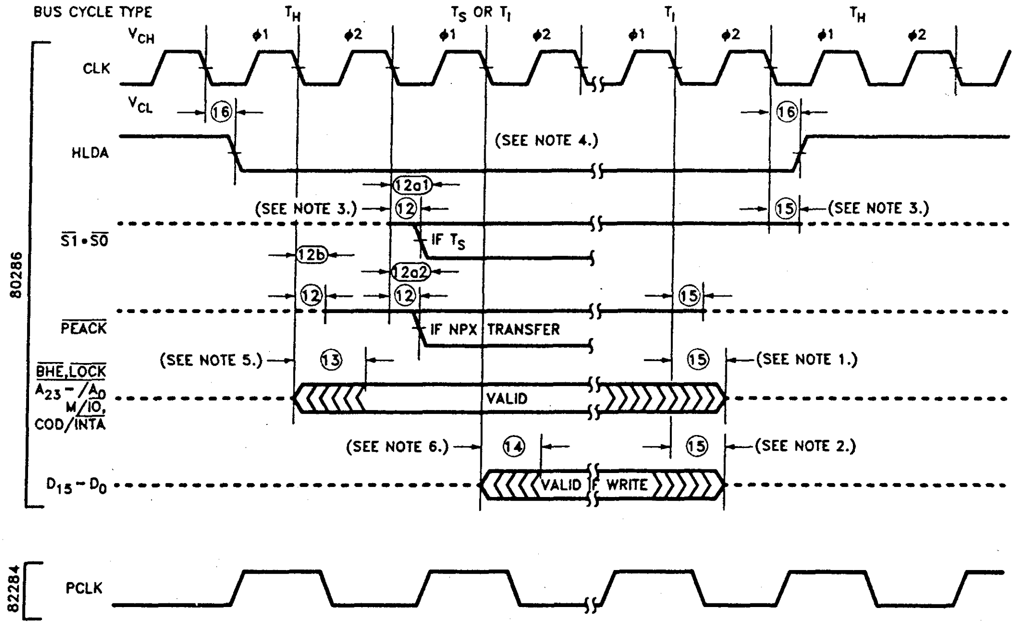


210253-42

NOTE:

When RESET meets the setup time shown, the next CLK will start or repeat $\phi2$ of a processor cycle.

EXITING AND ENTERING HOLD



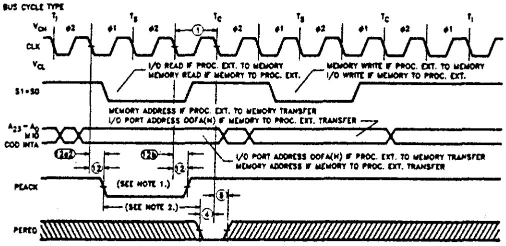
210253-43

NOTES:

1. These signals may not be driven by the 80286 during the time shown. The worst case in terms of latest float time is shown.
2. The data bus will be driven as shown if the last cycle before T_I in the diagram was a write T_C.
3. The 80286 floats its status pins during T_H. External 20 KΩ resistors keep these signals high (see Table 16).
4. For HOLD request set up to HLDA, refer to Figure 29.
5. $\overline{BHE}$ and $\overline{LOCK}$ are driven at this time but will not become valid until T_S.
6. The data bus will remain in 3-state OFF if a read cycle is performed.

WAVEFORMS (Continued)

80286 PEREQ/$\overline{\text{PEACK}}$ TIMING FOR ONE TRANSFER ONLY

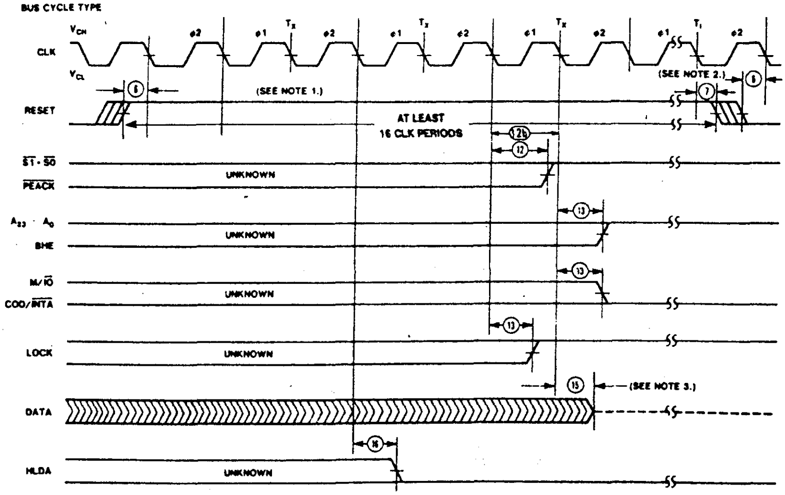

BUS CYCLE TYPE
T_I
T_S
T_C
T_S
T_C
T_I
V_{CH}
CLK
V_{CL}
φ2
φ1
φ2
1
φ2
φ1
φ2
φ1
φ2
φ1
I/O READ IF PROC. EXT. TO MEMORY
MEMORY READ IF MEMORY TO PROC. EXT.
MEMORY WRITE IF PROC. EXT. TO MEMORY
I/O WRITE IF MEMORY TO PROC. EXT.
S1•S0
MEMORY ADDRESS IF PROC. EXT. TO MEMORY TRANSFER
I/O PORT ADDRESS OOFA(H) IF MEMORY TO PROC. EXT. TRANSFER
$A_{23}-A_0$
M IO
COD INTA
12a2
12b
12
12
I/O PORT ADDRESS OOFA(H) IF PROC. EXT. TO MEMORY TRANSFER
MEMORY ADDRESS IF MEMORY TO PROC. EXT. TRANSFER
PEACK
(SEE NOTE 1.)
(SEE NOTE 2.)
5
4
PEREQ

210253-44

NOTES:

1. $\overline{\text{PEACK}}$ always goes active during the first bus operation of a processor extension data operand transfer sequence. The first bus operation will be either a memory read at operand address or I/O read at port address OOFA(H).
2. To prevent a second processor extension data operand transfer, the worst case maximum time (Shown above) is: 3× ①−12$a_{2max.}$−④$_{min.}$. The actual, configuration dependent, maximum time is: 3× ①−12$a_{2max.}$−④$_{min.}$ + A×2× ①.
A is the number of extra T_C states added to either the first or second bus operation of the processor extension data operand transfer sequence.

INITIAL 80286 PIN STATE DURING RESET



210253-45

NOTES:

1. Setup time for RESET ↑ may be violated with the consideration that φ1 of the processor clock may begin one system CLK period later.
2. Setup and hold times for RESET ↓ must be met for proper operation, but RESET ↓ may occur during φ1 or φ2.
3. The data bus is only guaranteed to be in 3-state OFF at the time shown.

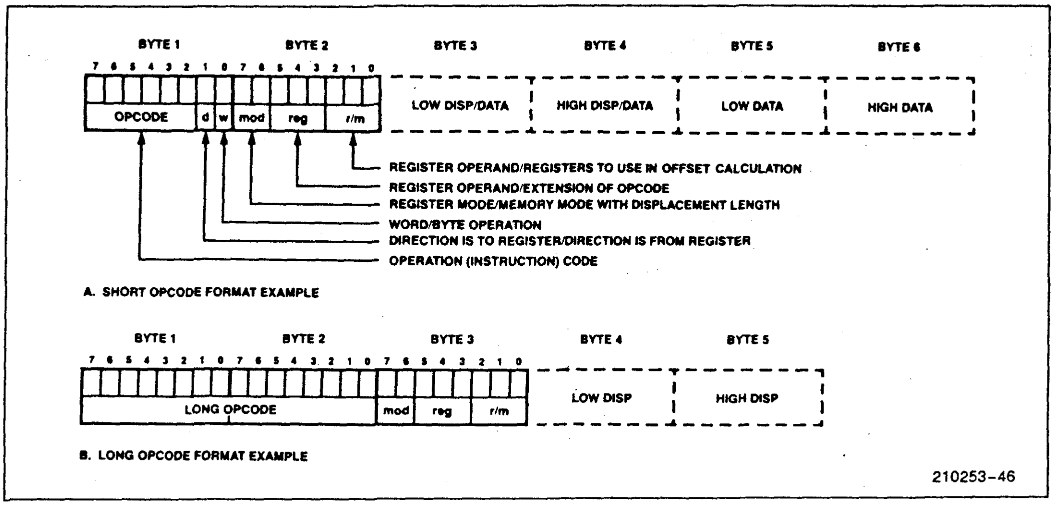

BYTE 1
BYTE 2
BYTE 3
BYTE 4
BYTE 5
BYTE 6
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
OPCODE
d
w
mod
reg
r/m
LOW DISP/DATA
HIGH DISP/DATA
LOW DATA
HIGH DATA
REGISTER OPERAND/REGISTERS TO USE IN OFFSET CALCULATION
REGISTER OPERAND/EXTENSION OF OPCODE
REGISTER MODE/MEMORY MODE WITH DISPLACEMENT LENGTH
WORD/BYTE OPERATION
DIRECTION IS TO REGISTER/DIRECTION IS FROM REGISTER
OPERATION (INSTRUCTION) CODE
A. SHORT OPCODE FORMAT EXAMPLE
BYTE 1
BYTE 2
BYTE 3
BYTE 4
BYTE 5
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
LONG OPCODE
mod
reg
r/m
LOW DISP
HIGH DISP
B. LONG OPCODE FORMAT EXAMPLE
210253-46

Figure 35. 80286 Instruction Format Examples

80286 INSTRUCTION SET SUMMARY

Instruction Timing Notes

The instruction clock counts listed below establish the maximum execution rate of the 80286. With no delays in bus cycles, the actual clock count of an 80286 program will average 5% more than the calculated clock count, due to instruction sequences which execute faster than they can be fetched from memory.

To calculate elapsed times for instruction sequences, multiply the sum of all instruction clock counts, as listed in the table below, by the processor clock period. An 8 MHz processor clock has a clock period of 125 nanoseconds and requires an 80286 system clock (CLK input) of 16 MHz.

Instruction Clock Count Assumptions

1. The instruction has been prefetched, decoded, and is ready for execution. Control transfer instruction clock counts include all time required to fetch, decode, and prepare the next instruction for execution.
2. Bus cycles do not require wait states.
3. There are no processor extension data transfer or local bus HOLD requests.
4. No exceptions occur during instruction execution.

Instruction Set Summary Notes

Addressing displacements selected by the MOD field are not shown. If necessary they appear after the instruction fields shown.

Above/below refers to unsigned value

Greater refers to positive signed value

Less refers to less positive (more negative) signed values

if d = 1 then to register; if d = 0 then from register

if w = 1 then word instruction; if w = 0 then byte instruction

if s = 0 then 16-bit immediate data form the operand

if s = 1 then an immediate data byte is sign-extended to form the 16-bit operand

x don't care

z used for string primitives for comparison with ZF FLAG

If two clock counts are given, the smaller refers to a register operand and the larger refers to a memory operand

* = add one clock if offset calculation requires summing 3 elements

n = number of times repeated

m = number of bytes of code in next instruction

Level (L)—Lexical nesting level of the procedure

The following comments describe possible exceptions, side effects, and allowed usage for instructions in both operating modes of the 80286.

REAL ADDRESS MODE ONLY

1. This is a protected mode instruction. Attempted execution in real address mode will result in an undefined opcode exception (6).
2. A segment overrun exception (13) will occur if a word operand reference at offset FFFF(H) is attempted.
3. This instruction may be executed in real address mode to initialize the CPU for protected mode.
4. The IOPL and NT fields will remain 0.
5. Processor extension segment overrun interrupt (9) will occur if the operand exceeds the segment limit.

EITHER MODE

6. An exception may occur, depending on the value of the operand.
7. $\overline{\text{LOCK}}$ is automatically asserted regardless of the presence or absence of the LOCK instruction prefix.
8. $\overline{\text{LOCK}}$ does not remain active between all operand transfers.

PROTECTED VIRTUAL ADDRESS MODE ONLY

9. A general protection exception (13) will occur if the memory operand cannot be used due to either a segment limit or access rights violation. If a stack segment limit is violated, a stack segment overrun exception (12) occurs.
10. For segment load operations, the CPL, RPL, and DPL must agree with privilege rules to avoid an exception. The segment must be present to avoid a not-present exception (11). If the SS register is the destination, and a segment not-present violation occurs, a stack exception (12) occurs.
11. All segment descriptor accesses in the GDT or LDT made by this instruction will automatically assert $\overline{\text{LOCK}}$ to maintain descriptor integrity in multiprocessor systems.
12. JMP, CALL, INT, RET, IRET instructions referring to another code segment will cause a general protection exception (13) if any privilege rule is violated.
13. A general protection exception (13) occurs if CPL $\neq$ 0.
14. A general protection exception (13) occurs if CPL > IOPL.
15. The IF field of the flag word is not updated if CPL > IOPL. The IOPL field is updated only if CPL = 0.
16. Any violation of privilege rules as applied to the selector operand do not cause a protection exception; rather, the instruction does not return a result and the zero flag is cleared.
17. If the starting address of the memory operand violates a segment limit, or an invalid access is attempted, a general protection exception (13) will occur before the ESC instruction is executed. A stack segment overrun exception (12) will occur if the stack limit is violated by the operand's starting address. If a segment limit is violated during an attempted data transfer then a processor extension segment overrun exception (9) occurs.
18. The destination of an INT, JMP, CALL, RET or IRET instruction must be in the defined limit of a code segment or a general protection exception (13) will occur.

80286 INSTRUCTION SET SUMMARY

FUNCTION	FORMAT					CLOCK COUNT		COMMENTS	
						Real Address Mode	Protected Virtual Address Mode	Real Address Mode	Protected Virtual Address Mode
DATA TRANSFER									
MOV = Move:									
Register to Register/Memory	1000100w	mod reg r/m				2,3*	2,3*	2	9
Register/memory to register	1000101w	mod reg r/m				2,5*	2,5*	2	9
Immediate to register/memory	1100011w	mod 000 r/m	data	data if w = 1		2,3*	2,3*	2	9
Immediate to register	1011w reg	data	data if w=1			2	2		
Memory to accumulator	1010000w	addr-low	addr-high			5	5	2	9
Accumulator to memory	1010001w	addr-low	addr-high			3	3	2	9
Register/memory to segment register	10001110	mod 0 reg r/m				2,5*	17,19*	2	9,10,11
Segment register to register/memory	10001100	mod 0 reg r/m				2,3*	2,3*	2	9
PUSH = Push:									
Memory	11111111	mod 110 r/m				5*	5*	2	9
Register	01010 reg					3	3	2	9
Segment register	000 reg 110					3	3	2	9
Immediate	011010s0	data	data if s=0			3	3	2	9
PUSHA = Push All	01100000					17	17	2	9
POP = Pop:									
Memory	10001111	mod 000 r/m				5*	5*	2	9
Register	01011 reg					5	5	2	9
Segment register	000 reg 111	(reg≠01)				5	20	2	9,10,11
POPA = Pop All	01100001					19	19	2	9
XCHG = Exhcange:									
Register/memory with register	1000011w	mod reg r/m				3,5*	3,5*	2,7	7,9
Register with accumulator	10010 reg					3	3		
IN = Input from:									
Fixed port	1110010w	port				5	5		14
Variable port	1110110w					5	5		14
OUT = Output to:									
Fixed port	1110011w	port				3	3		14
Variable port	1110111w					3	3		14
XLAT = Translate byte to AL	11010111					5	5		9
LEA = Load EA to register	10001101	mod reg r/m				3*	3*		
LDS = Load pointer to DS	11000101	mod reg r/m	(mod≠11)			7*	21*	2	9,10,11
LES = Load pointer to ES	11000100	mod reg r/m	(mod≠1)			7*	21*	2	9,10,11

Shaded areas indicate instructions not available in 8086, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT				CLOCK COUNT		COMMENTS	
					Real Address Mode	Protected Virtual Address Mode	Real Address Mode	Protected Virtual Address Mode
DATA TRANSFER (Continued)								
LAHF Load AH with flags	10011111				2	2		
SAHF = Store AH into flags	10011110				2	2		
PUSHF = Push flags	10011100				3	3	2	9
POPF = Pop flags	10011101				5	5	2,4	9,15
ARITHMETIC								
ADD = Add:								
Reg/memory with register to either	000000dw	mod reg r/m			2,7*	2,7*	2	9
Immediate to register/memory	100000sw	mod 000 r/m	data	data if s w = 01	3,7*	3,7*	2	9
Immediate to accumulator	0000010w	data	data if w = 1		3	3		
ADC = Add with carry:								
Reg/memory with register to either	000100dw	mod reg r/m			2,7*	2,7*	2	9
Immediate to register/memory	100000sw	mod 010 r/m	data	data if s w = 01	3,7*	3,7*	2	9
Immediate to accumulator	0001010w	data	data if w = 1		3	3		
INC = Increment:								
Register/memory	1111111w	mod 000 r/m			2,7*	2,7*	2	9
Register	01000 reg				2	2		
SUB = Subtract:								
Reg/memory and register to either	001010dw	mod reg r/m			2,7*	2,7*	2	9
Immediate from register/memory	100000sw	mod 101 r/m	data	data if s w = 01	3,7*	3,7*	2	9
Immediate from accumulator	0010110w	data	data if w = 1		3	3		
SBB = Subtract with borrow:								
Reg/memory and register to either	000110dw	mod reg r/m			2,7*	2,7*	2	9
Immediate from register/memory	100000sw	mod 011 r/m	data	data if s w = 01	3,7*	3,7*	2	9
Immediate from accumulator	0001110w	data	data if w = 1		3	3		
DEC = Decrement								
Register/memory	1111111w	mod 001 r/m			2,7*	2,7*	2	9
Register	01001 reg				2	2		
CMP = Compare								
Register/memory with register	0011101w	mod reg r/m			2,6*	2,6*	2	9
Register with register/memory	0011100w	mod reg r/m			2,7*	2,7*	2	9
Immediate with register/memory	100000sw	mod 111 r/m	data	data if s w = 01	3,6*	3,6*	2	9
Immediate with accumulator	0011110w	data	data if w = 1		3	3		
NEG = Change sign	1111011w	mod 011 r/m			2	7*	2	9
AAA = ASCII adjust for add	00110111				3	3		
DAA = Decimal adjust for add	00100111				3	3		

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT		COMMENTS	
		Real Address Mode	Protected Virtual Address Mode	Real Address Mode	Protected Virtual Address Mode
ARITHMETIC (Continued)					
AAS = ASCII adjust for subtract	00111111	3	3		
DAS = Decimal adjust for subtract	00101111	3	3		
MUL = Multiply (unsigned):	1111011w · mod 100 r/m				
Register-Byte		13	13		
Register-Word		21	21		
Memory-Byte		16*	16*	2	9
Memory-Word		24*	24*	2	9
IMUL = Integer multiply (signed):	1111011w · mod 101 r/m				
Register-Byte		13	13		
Register-Word		21	21		
Memory-Byte		16*	16*	2	9
Memory-Word		24*	24*	2	9
IMUL = Integer immediate multiply (signed)	011010s1 · mod reg r/m · data · data if s = 0	21,24*	21,24*	2	9
DIV = Divide (unsigned)	1111011w · mod 110 r/m				
Register-Byte		14	14	6	6
Register-Word		22	22	6	6
Memory-Byte		17*	17*	2,6	6,9
Memory-Word		25*	25*	2,6	6,9
IDIV = Integer divide (signed)	1111011w · mod 111 r/m				
Register-Byte		17	17	6	6
Register-Word		25	25	6	6
Memory-Byte		20*	20*	2,6	6,9
Memory-Word		28*	28*	2,6	6,9
AAM = ASCII adjust for multiply	11010100 · 00001010	16	16		
AAD = ASCII adjust for divide	11010101 · 00001010	14	14		
CBW = Convert byte to word	10011000	2	2		
CWD = Convert word to double word	10011001	2	2		
LOGIC					
Shift/Rotate Instructions:					
Register/Memory by 1	1101000w · mod TTT r/m	2,7*	2,7*	2	9
Register/Memory by CL	1101001w · mod TTT r/m	5+n,8+n*	5+n,8+n*	2	9
Register/Memory by Count	1100000w · mod TTT r/m · count	5+n,8+n*	5+n,8+n*	2	9

TTT	Instruction
000	ROL
001	ROR
010	RCL
011	RCR
100	SHL/SAL
101	SHR
111	SAR

Shaded areas indicate instructions not available in 8086, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT		COMMENTS	
		Real Address Mode	Protected Virtual Address Mode	Real Address Mode	Protected Virtual Address Mode
ARITHMETIC (Continued)					
AND = And:					
Reg/memory and register to either	001000dw | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1000000w | mod 100 r/m | data | data if w=1	3,7*	3,7*	2	9
Immediate to accumulator	0010010w | data | data if w=1	3	3		
TEST = And function to flags, no result:					
Register/memory and register	1000010w | mod reg r/m	2,6*	2,6*	2	9
Immediate data and register/memory	1111011w | mod 000 r/m | data | data if w=1	3,6*	3,6*	2	9
Immediate data and accumulator	1010100w | data | data if w=1	3	3		
OR = Or:					
Reg/memory and register to either	000010dw | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1000000w | mod 001 r/m | data | data if w=1	3,7*	3,7*	2	9
Immediate to accumulator	0000110w | data | data if w=1	3	3		
XOR = Exclusive or:					
Reg/memory and register to either	001100dw | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1000000w | mod 110 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0011010w | data | data if w = 1	3	3		
NOT = Invert register/memory	1111011w | mod 010 r/m	2,7*	2,7*	2	9
STRING MANIPULATION:					
MOVS = Move byte/word	1010010w	5	5	2	9
CMPS = Compare byte/word	1010011w	8	8	2	9
SCAS = Scan byte/word	1010111w	7	7	2	9
LODS = Load byte/wd to AL/AX	1010110w	5	5	2	9
STOS = Stor byte/wd from AL/A	1010101w	3	3	2	9
INS = Input byte/wd from DX port	0110110w	5	5	2	9,14
OUTS = Output byte/wd to DX port	0110111w	5	5	2	9,14
Repeated by count in CX					
MOVS = Move string	11110011 | 1010010w	5+4n	5+4n	2	9
CMPS = Compare string	1111001z | 1010011w	5+9n	5+9n	2,8	8,9
SCAS = Scan string	1111001z | 1010111w	5+8n	5+8n	2,8	8,9
LODS = Load string	11110011 | 1010110w	5+4n	5+4n	2,8	8,9
STOS = Store string	11110011 | 1010101w	4+3n	4+3n	2,8	8,9
INS = Input string	11110011 | 0110110w	5+4n	5+4n	2	9,14
OUTS = Output string	11110011 | 0110111w	5+4n	5+4n	2	9,14

Shaded areas indicate instructions not available in 8086, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

					CLOCK COUNT		COMMENTS	
FUNCTION	FORMAT				Real Address Mode	Protected Virtual Address Mode	Real Address Mode	Protected Virtual Address Mode
CONTROL TRANSFER								
CALL = Call:								
Direct within segment	11101000	disp-low	disp-high		7+m	7+m	2	18
Register/memory indirect within segment	11111111	mod 010 r/m			7+m, 11+m*	7+m, 11+m*	2,8	8,9,18
Direct intersegment	10011010	segment offset			13+m	26+m	2	11,12,18
Protected Mode Only (Direct intersegment):		segment selector						
Via call gate to same privilege level						41+m		8,11,12,18
Via call gate to different privilege level, no parameters						82+m		8,11,12,18
Via call gate to different privilege level, x parameters						86 +4x+m		8,11,12,18
Via TSS						177+m		8,11,12,18
Via task gate						182+m		8,11,12,18
Indirect intersegment	11111111	mod 011 r/m		(mod≠11)	16+m	29+m*	2	8,9,11,12,18
Protected Mode Only (Indirect intersegment):								
Via call gate to same privilege level						44+m*		8,9,11,12,18
Via call gate to different privilege level, no parameters						83 +m*		8,9,11,12,18
Via call gate to different privilege level, x parameters						90+4x +m*		8,9,11,12,18
Via TSS						180+m*		8,9,11,12,18
Via task gate						185+m*		8,9,11,12,18
JMP = Unconditional jump:								
Short/long	11101011	disp-low			7+m	7+m		18
Direct within segment	11101001	disp-low	disp-high		7+m	7+ m		18
Register/memory indirect within segment	11111111	mod 100 r/m			7 +m, 11+m*	7+m, 11+m*	2	9,18
Direct intersegment	11101010	segment offset			11+m	23+m		11,12,18
Protected Mode Only (Direct intersegment):		segment selector						
Via call gate to same privilege level						38+m		8,11,12,18
Via TSS						175+m		8,11,12,18
Via task gate						180+m		8,11,12,18
Indirect intersegment	11111111	mod 101 r/m		(mod≠11)	15+m*	26+m*	2	8,9,11,12,18
Protected Mode Only (Indirect intersegment):								
Via call gate to same privilege level						41+m*		8,9,11,12,18
Via TSS						178+m*		8,9,11,12,18
Via task gate						183+m*		8,9,11,12,18
RET = Return from CALL:								
Within segment	11000011				11+m	11+m	2	8,9,18
Within seg adding immed to SP	11000010	data-low	data-high		11+m	11+m	2	8,9,18
Intersegment	11001011				15+m	25+m	2	8,9,11,12,18
Intersegment adding immediate to SP	11001010	data-low	data-high		15+m		2	8,9,11,12,18
Protected Mode Only (RET):								
To different privilege level						55+m		9,11,12,18

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT				CLOCK COUNT		COMMENTS	
					Real Address Mode	Protected Virtual Address Mode	Real Address Mode	Protected Virtual Address Mode
CONTROL TRANSFER (Continued)								
JE/JZ = Jump on equal zero	01110100	disp			7 + m or 3	7 + m or 3		18
JL/JNGE = Jump on less/not greater or equal	01111100	disp			7 + m or 3	7 + m or 3		18
JLE/JNG = Jump on less or equal/not greater	01111110	disp			7 + m or 3	7 + m or 3		18
JB/JNAE = Jump on below/not above or equal	01110010	disp			7 + m or 3	7 + m or 3		18
JBE/JNA = Jump on below or equal/not above	01110110	disp			7 + m or 3	7 + m or 3		18
JP/JPE = Jump on parity/parity even	01111010	disp			7 + m or 3	7 + m or 3		18
JO = Jump on overflow	01110000	disp			7 + m or 3	7 + m or 3		18
JS = Jump on sign	01111000	disp			7 + m or 3	7 + m or 3		18
JNE/JNZ = Jump on not equal/not zero	01110101	disp			7 + m or 3	7 + m or 3		18
JNL/JGE = Jump on not less/greater or equal	01111101	disp			7 + m or 3	7 + m or 3		18
JNLE/JG = Jump on not less or equal/greater	01111111	disp			7 + m or 3	7 + m or 3		18
JNB/JAE = Jump on not below/above or equal	01110011	disp			7 + m or 3	7 + m or 3		18
JNBE/JA = Jump on not below or equal/above	01110111	disp			7 + m or 3	7 + m or 3		18
JNP/JPO = Jump on not par/par odd	01111011	disp			7 + m or 3	7 + m or 3		18
JNO = Jump on not overflow	01110001	disp			7 + m or 3	7 + m or 3		18
JNS = Jump on not sign	01111001	disp			7 + m or 3	7 + m or 3		18
LOOP = Loop CX times	11100010	disp			8 + m or 4	8 + m or 4		18
LOOPZ/LOOPE = Loop while zero/equal	11100001	disp			8 + m or 4	8 + m or 4		18
LOOPNZ/LOOPNE = Loop while not zero/equal	11100000	disp			8 + m or 4	8 + m or 4		18
JCXZ = Jump on CX zero	11100011	disp			8 + m or 4	8 + m or 4		18
ENTER = Enter Procedure	11001000	data-low	data-high	L			2,8	8,9
L = 0					11	11	2,8	8,9
L = 1					15	15	2,8	8,9
L > 1					16 + 4(L − 1)	16 + 4(L − 1)	2,8	8,9
LEAVE = Leave Procedure	11001001				5	5		
INT = Interrupt:								
Type specified	11001101	type			23 + m		2,7,8	
Type 3	11001100				23 + m		2,7,8	
INTO = Interrupt on overflow	11001110				24 + m or 3 (3 if no interrupt)	(3 if no interrupt)	2,6,8	

Shaded areas indicate instructions not available in 8086, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT		COMMENTS	
		Real Address Mode	Protected Virtual Address Mode	Real Address Mode	Protected Virtual Address Mode
CONTROL TRANSFER (Continued)					
Protected Mode Only:					
Via interrupt or trap gate to same privilege level			40 + m		7,8,11,12,18
Via interrupt or trap gate to fit different privilege level			78 + m		7,8,11,12,18
Via Task Gate			167 + m		7,8,11,12,18
IRET = Interrupt return	11001111	17 + m	31 + m	2,4	8,9,11,12,15,18
Protected Mode Only:					
To different privilege level			55 + m		8,9,11,12,15,18
To different task (NT = 1)			169 + m		8,9,11,12,18
BOUND = Detect value out of range	01100010 mod reg r/m	13*	13* (Use INT clock count if exception 5)	2,6	6,8,9,11,12,18
PROCESSOR CONTROL					
CLC = Clear carry	11111000	2	2		
CMC = Complement carry	11110101	2	2		
STC = Set carry	11111001	2	2		
CLD = Clear direction	11111100	2	2		
STD = Set direction	11111101	2	2		
CLI = Clear interrupt	11111010	3	3		14
STI = Set interrupt	11111011	2	2		14
HLT = Halt	11110100	2	2		13
WAIT = Wait	10011011	3	3		
LOCK = Bus lock prefix	11110000	0	0		14
CTS = Clear task switched flag	00001111 00000110	2	2	3	13
ESC = Processor Extension Escape	11011TTT mod LLL r/m (TTT LLL are opcode to processor extension)	9-20*	9-20*	5,8	8,17
SEG = Segment Override Prefix	001 reg 110	0	0		
PROTECTION CONTROL					
LGDT = Load global descriptor table register	00001111 00000001 mod 010 r/m	11*	11*	2,3	9,13
SGDT = Store global descriptor table register	00001111 00000001 mod 000 r/m	11*	11*	2,3	9
LIDT = Load interrupt descriptor table register	00001111 00000001 mod 011 r/m	12*	12*	2,3	9,13
SIDT = Store interrupt descriptor table register	00001111 00000001 mod 001 r/m	12*	12*	2,3	9
LLDT = Load local descriptor table register from register memory	00001111 00000000 mod 010 r/m		17,19*	1	9,11,13
SLDT = Store local descriptor table register to register/memory	00001111 00000000 mod 000 r/m		2,3*	1	9

Shaded areas indicate instructions not available in 8086, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT			CLOCK COUNT		COMMENTS	
				Real Address Mode	Protected Virtual Address Mode	Real Address Mode	Protected Virtual Address Mode
PROTECTION CONTROL (Continued)							
LTR = Local task register from register/memory	00001111	00000000	mod 0 1 1 r/m		17,19*	1	9,11,13
STR = Store task register to register memory	00001111	00000000	mod 0 0 1 r/m		2,3*	1	9
LMSW = Load machine status word from register/memory	00001111	00000001	mod 1 1 0 r/m	3,6*	3,6*	2,3	9,13
SMSW = Store machine status word	00001111	00000001	mod 1 0 0 r/m	2,3*	2,3*	2,3	9
LAR = Load access rights from register/memory	00001111	00000010	mod reg r/m		14,16*	1	9,11,16
LSL = Load segment limit from register/memory	00001111	00000011	mod reg r/m		14,16*	1	9,11,16
ARPL = Adjust requested privilege level: from register/memory		01100011	mod reg r/m		10*,11*	2	8,9
VERR = Verify read access: register/memory	00001111	00000000	mod 1 0 0 r/m		14,16*	1	9,11,16
VERR = Verify write access:	00001111	00000000	mod 1 0 1 r/m		14,16*	1	9,11,16

Shaded areas indicate instructions not available in 8086, 88 microsystems.

Footnotes

The Effective Address (EA) of the memory operand is computed according to the mod and r/m fields:

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16 bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EQ = disp-high: disp-low.

SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

reg is assigned according to the following:

reg	Segment Register
00	ES
01	CS
10	SS
11	DC

REG is assigned according to the following table:

16-Bit (w = 1)		8-Bit (w = 0)	
000	AX	000	AL
001	CX	001	CL
010	DX	010	DL
011	BX	011	BL
100	SP	100	AH
101	BP	101	CH
101	SI	110	DH
111	DI	111	BH

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those addressed by the DI register) are computed using the ES segment, which may not be overridden.

DATA SHEET REVISION REVIEW

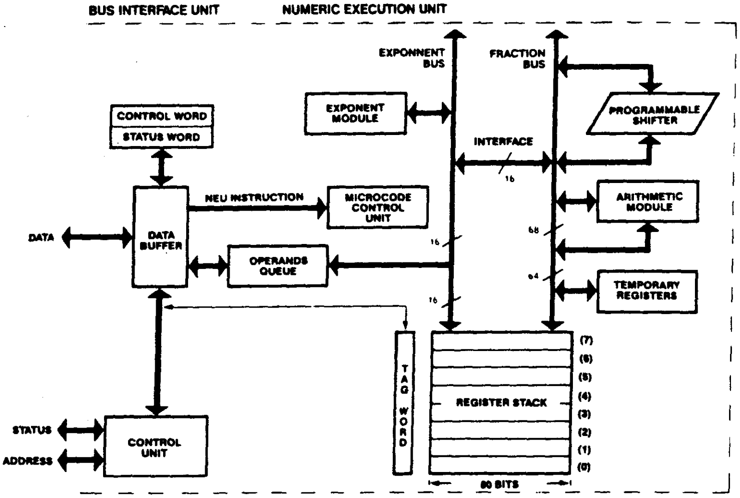
The following list represents key differences between this and the -012 data sheet. Please review this summary carefully.

1. Specifications for the 6 MHz version of the part have been deleted. Intel no longer manufactures an 80286-6.
2. The system diagrams (Figures 31 and 32) have been modified. The circuit which drives the RES input of the 82C284 has been modified in order to allow the 82C284 to correctly generate a system reset signal. See the 82C284 data sheet (Order No. 210453) for further information.

80287
80-BIT HMOS
NUMERIC PROCESSOR EXTENSION
(80287-3, 80287-6, 80287-8, 80287-10)

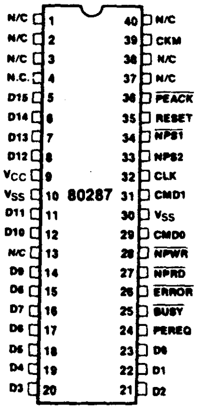
- **High Performance 80-Bit Internal Architecture**
- **Implements Proposed IEEE Floating Point Standard 754**
- **Expands 80286 Data types to Include 32-, 64-, 80-Bit Floating Point, 32-, 64-Bit Integers and 18-Digit BCD Operands**
- **Object Code Compatible with 8087**
- **Built-in Exception Handling**
- **Operates in Both Real and Protected Mode 80286 Systems**
- **8x80-Bit, Individually Addressable, Numeric Register Stack**
- **Protected Mode Operation Completely Conforms to the 80286 Memory Management and Protection Mechanisms**
- **Directly Extends 80286 Instruction Set to Trigonometric, Logarithmic, Exponential and Arithmetic Instructions for All Data types**
- **Operates with 80386 CPU without Software Modification**
- **Available in EXPRESS—Standard Temperature Range**
- **Available in 40 pin-CERDIP package** (see Packaging Spec: Order #231369)

The Intel 80287 is a high performance numerics processor extension that extends the 80286 architecture with floating point, extended integer and BCD data types. The 80286/80287 computing system fully conforms to the proposed IEEE Floating Point Standard. Using a numerics oriented architecture, the 80287 adds over fifty mnemonics to the 80286/80287 instruction set, making the 80286/80287 a complete solution for high performance numeric processing. The 80287 is implemented in N-channel, depletion load, silicon gate technology (HMOS) and packaged in a 40-pin cerdip package. The 80286/80287 is object code compatible with the 8086/8087 and 8088/8087.



210920-1

Figure 1. 80287 Block Diagram



210920-2

NOTE:
N/C Pins should not be connected

Figure 2. 80287 Pin Configuration

September 1987
Order Number: 210920-007

Table 1. 80287 Pin Description

Symbols	Type	Name and Functon
CLK	I	**CLOCK INPUT:** this clock provides the basic timing for internal 80287 operations. Special MOS level inputs are required. The 82284 or 8284A CLK outputs are compatible to this input.
CKM	I	**CLOCK MODE SIGNAL:** indicates whether CLK input is to be divided by 3 or used directly. A HIGH input will cause CLK to be used directly. This input must be connected to V_{CC} or V_{SS} as appropriate. This input must be either HIGH or LOW 20 CLK cycles before RESET goes LOW.
RESET	I	**SYSTEM RESET:** causes the 80287 to immediately terminate its present activity and enter a dormant state. RESET is required to be HIGH for more than 4 80287 CLK cycles. For proper initialization the HIGH-LOW transition must occur no sooner than 50 μs after V_{CC} and CLK meet their D.C. and A.C. specifications.
D15–D0	I/O	**DATA:** 1-bit bidirectional data bus. Inputs to these pins may be applied asynchronous to the 80287 clock.
$\overline{\text{BUSY}}$	O	**BUSY STATUS:** asserted by the 80287 to indicate that it is currently executing a command.
$\overline{\text{ERROR}}$	O	**ERROR STATUS:** reflects the ES bit of the status word. This signal indicates that an unmasked error condition exists.
PEREQ	O	**PROCESSOR EXTENSION DATA CHANNEL OPERAND TRANSFER REQUEST:** a HIGH on this output indicates that the 80287 is ready to transfer data. PEREQ will be disabled upon assertion of $\overline{\text{PEACK}}$ or upon actual data transfer, whichever occurs first, if no more transfers are required.
$\overline{\text{PEACK}}$	I	**PROCESSOR EXTENSION DATA CHANNEL OPERAND tRANSFER ACKNOWLEDGE:** acknowledges that the request signal (PEREQ) has been recognized. Will cause the request (PEREQ) to be withdrawn in case there are no more transfers required. $\overline{\text{PEACK}}$ may be asynchronous to the 80287 clock.
$\overline{\text{NPRD}}$	I	**NUMERIC PROCESSOR READ:** Enables transfer of data from the 80287. This input may be asynchronous to the 80287 clock.
$\overline{\text{NPWR}}$	I	**NUMERIC PROCESSOR READ:** Enables transfer of data from the 80287. This input may be asynchronous to the 80287 clock.
$\overline{\text{NPS1}}$, NPS2	I	**NUMERIC PROCESSOR SELECTS:** indicate the CPU is performing an ESCAPE instruction. Concurrent assertion of these signals (i.e., $\overline{\text{NPS1}}$ is LOW and NPS2 is HIGH) enables the 80287 to perform floating point instrucctions. No data transfers involving the 80287 will occur unless the device is selected via these lines. These inputs may be asynchronous to the 80287 clock.
CMD1, CMD0	I	**COMMAND LINES:** These, along with select inputs, allow the CPU to direct the operation of the 80287. These inputs may be asynchronous to the 80287 clock.

Table 1. 80187 Pin Description (Continued)

Symbols	Type	Name and Function
V_{SS}	I	System ground, both pins must be connected to ground.
V_{CC}	I	+5V supply

FUNCTIONAL DESCRIPTION

The 80287 Numeric Processor Extension (NPX) provides arithmetic instructions for a variety of numeric data types in 80286/80287 systems. It also executes numerous built-in transcendental functions (e.g., tangent and log functions). The 80287 executes instructions in parallel with an 80286. It effectively extends the register and instruction set of an 80286 system for existing 80286 data types and adds several new data types as well. Figure 3 presents the program visible register model of the 80286/80287. Essentially, the 80287 can be treated as an additional resource or an extension to the 80286 that can be used as a single unified system, the 80286/80287.

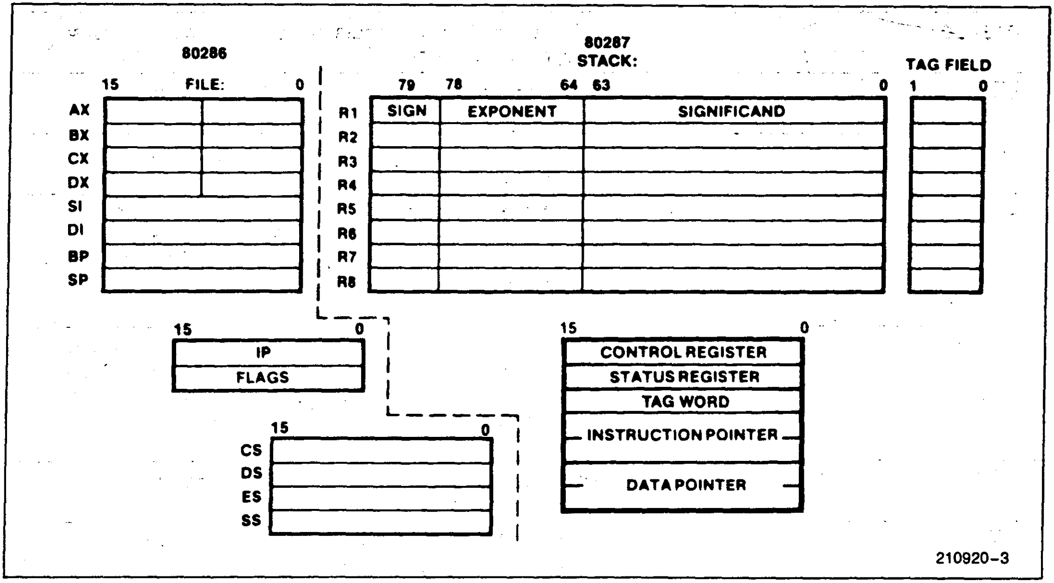


Figure 3. 80286/80287 Architecture

The 80287 has two operating modes similar to the two modes of the 80286. When reset, 80287 is in the real address mode. It can be placed in the protected virtual address mode by executing the SETPM ESC instruction. The 80287 cannot be switched back to the real address mode except by reset. In the real address mode, the 80286/80287 is completely software compatible with 8086/8087 and 8088/8087.

Once in protected mode, all references to memory for numerics data or status information, obey the 80286 memory management and protection rules giving a fully protected extension of the 80286 CPU. In the protected mode, 80286/80287 numerics software is also completely compatible with 8086/8087 and 8088/8087.

SYSTEM CONFIGURATION WITH 80286

As a processor extension to an 80286, the 80287 can be connected to the CPU as shown in Figure 4A. The data channel control signals (PEREQ, PEACK), the BUSY signal and the NPRD, NPWR signals, allow the NPX to receive instructions and data from the CPU. When in the protected mode, all information received by the NPX is validated by the 80286 memory management and protection unit. Once started, the 80287 can process in parallel with and independent of the host CPU. When the NPX detects an error or exception, it will indicate this to the CPU by asserting the ERROR signal.

The NPX uses the processor extension request and acknowledge pins of the 80286 CPU to implement data transfers with memory under the protection model of the CPU. The full virtual and physical address space of the 80286 is available. Data for the 80287 in memory is addressed and represented in the same manner as for an 8087.

The 80287 can operate either directly from the CPU clock or with a dedicated clock. For operation with the CPU clock (CKM = 0), the 80287 works at one-third the frequency of the system clock (i.e., for an 8 MHz 80286, the 16 MHz system clock is divided down to 5.3 MHz). The 80287 provides a capability to internally divide the CPU clock by three to produce the required internal clock (33% duty cycle). To use a higher performance 80287 (8 MHz), an 8284A clock driver and appropriate crystal may be used to directly drive the 80287 with a ⅓ duty cycle clock on the CLK input (CKM = 1). The following table describes the relationship between the clock speed and the 287 speed version needed as a function of the CKM state.

287 Speed Version	CLK Speed	
	CKM = 0	CKM = 1
5 MHz	12 MHz	5 MHz
6 MHz	16 MHz	6 MHz
8 MHz	20 MHz	8 MHz
10 MHz	25 MHz	10 MHz

SYSTEM CONFIGURATION WITH 80386

The 80287 can also be connected as a processor extension to the 80386 CPU as shown in Figure 4b. All software written for 8086/8087 and 80286/80287 is object code compatible with 80386/80287 and can benefit from the increased speed of the 80386 CPU.

Note that the PEACK input pin is pulled high. This is because the 80287 is not required to keep track of the number of words transferred during an operand transfer when it is connected to the 80386 CPU. Unlike the 80286 CPU, the 80386 CPU knows the exact length of the operand being transferred to/from the 80287. After an ESC instruction has been sent to the 80287, the 80386 processor extension data channel will initiate the data transfer as soon as it receives the PEREQ signal from the 80287. The transfer is automatically terminated by the 80386 CPU as soon as all the words of the operand have been transferred.

Because of the very high speed local local bus of the 80386 CPU, the 80287 cannot reside directly on the CPU local bus. A local bus controller logic is used to generate the necessary read and write cycle timings as well as the chip select timings for the 80287. The 80386 CPU uses I/O addresses 800000F8 through 800000FF to communicate with the 80287. This is beyond the normal I/O address space of the CPU and makes it easier to generate the chip select signals using A31 and M/IO. It may also be noted that the 80386 CPU automatically generates 16-bit bus cycles whenever it communicates with the 80287.

HARDWARE INTERFACE

Communication of instructions and data operands between the 80286 and 80287 is handled by the CMD0, CMD1, NPS1, NPS2, NPRD, and NPWR signals. I/O port addresses 00F8H, 00FAH, and 00FCH are used by the 80286 for this communication. When any of these addresses are used, the NPS1 input must be LOW and NPS2 input HIGH. The IORC and IOWC outputs of the 82288 identify I/O space transfers (see Figure 4A). CMD0 should be connected to latched 80286 A1 and CMD1 should be connected to latched 80286 A2.

I/O ports 00F8H to 00FFH are reserved for the 80286/80287 interface. To guarantee correct operation of the 80287, programs must not perform any I/O operations to these ports.

The PEREQ, PEACK, BUSY, and ERROR signals of the 80287 are connected to the same-named 80286 input. The data pins of the 80287 should be directly connected to the 80286 data bus. Note that all bus drivers connected to the 80286 local bus must be inhibited when the 80286 reads from the 80287. The use of M/IO in the decoder prevents INTA bus cycles from disabling the data transceivers.

PROGRAMMING INTERFACE

Table 2 lists the seven data types the 80287 supports and presents the format for each type. These values are stored in memory with the least significant digits at the lowest memory address. Programs retrieve these values by generating the lowest address. All values should start at even addresses for maximum system performance.

Internally the 80287 holds all numbers in the temporary real format. Load instructions automatically convert operands represented in memory as 16-, 32-, or 64-bit integers, 32- or 64-bit floating point number or 18-digit packed BCD numbers into temporary real format. Store instructions perform the reverse type conversion.

80287 computations use the processor's register stack. These eight 80-bit registers provide the equivalent capacity of 40 16-bit registers. The 80287 register set can be accessed as a stack, with instructions operating on the top one or two stack elements, or as a fixed register set, with instructions operating on explicitly designated registers.

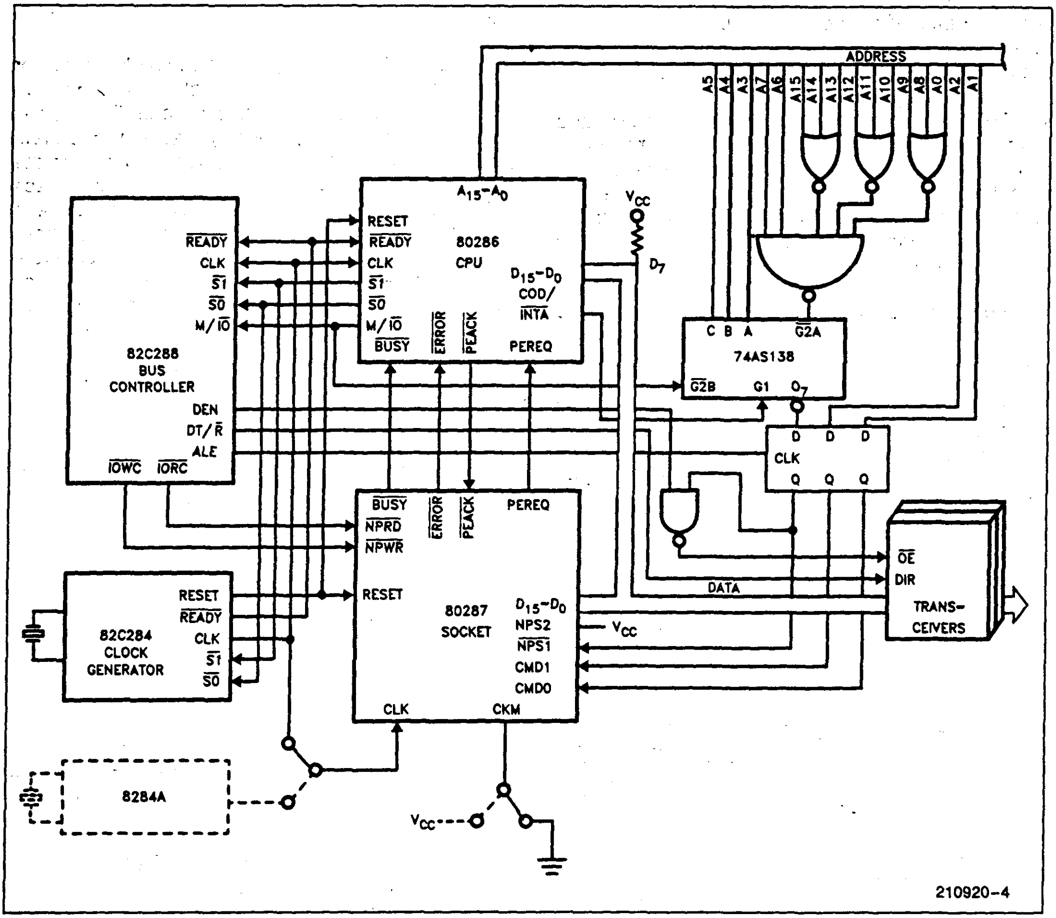


Figure 4A. 80286/80287 System Configuration

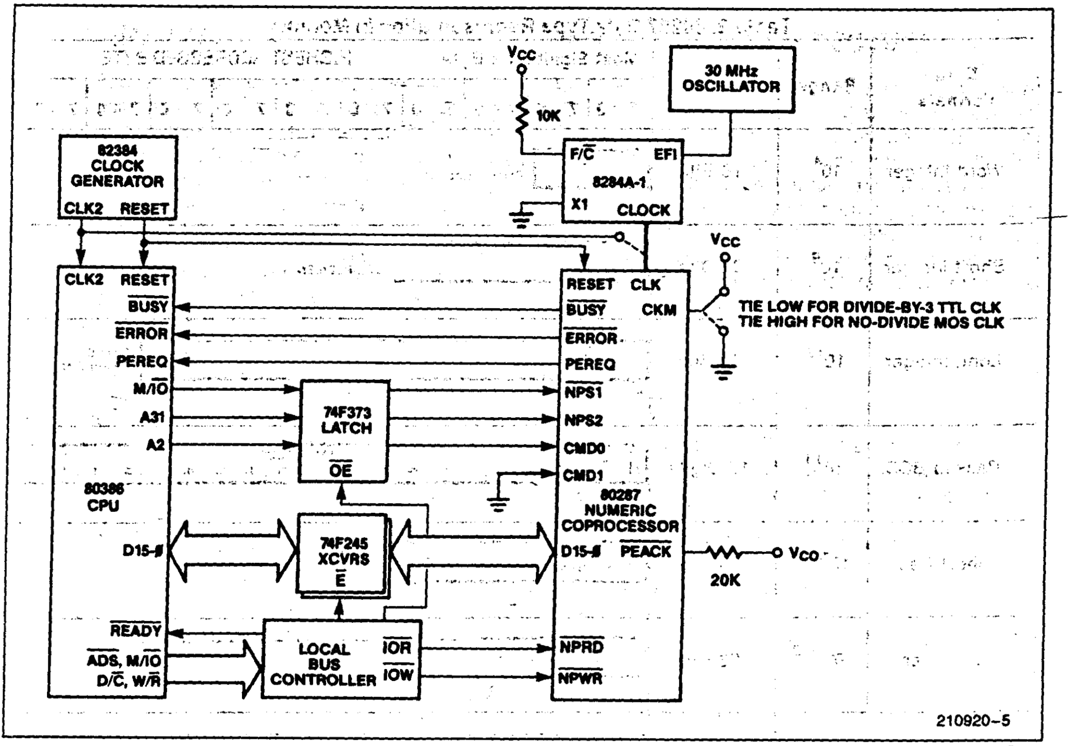
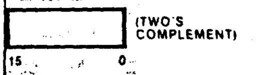
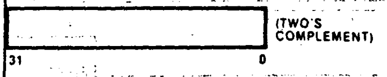
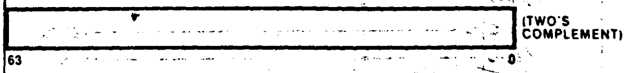
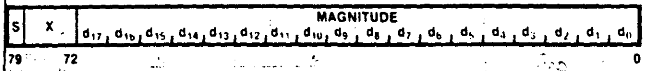
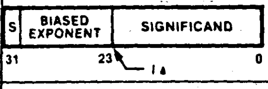
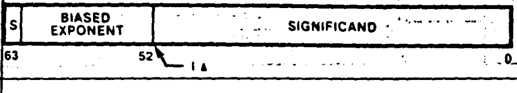
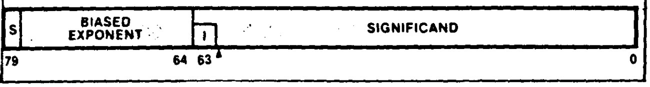


Figure 4B. 80386/80287 System Configuration

Table 2. 80287 Data Type Representation in Memory

Data Formats	Range	Precision	Most Significant Byte HIGHEST ADDRESSED BYTE 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0
Word Integer	10^4	16 Bits	
Short Integer	10^9	32 Bits	
Long Integer	10^{19}	64 Bits	
Packed BCD	10^{18}	18 Digits	 S X MAGNITUDE d17 d16 d15 d14 d13 d12 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0 79 72 0
Short Real	$10^{\pm 38}$	24 Bits	
Long Real	$10^{\pm 308}$	53 Bits	
Temporary Real	$10^{\pm 4932}$	64 Bits	 S BIASED EXPONENT I SIGNIFICAND 79 64 63 0

210920-6

NOTES:

1. S = Sign bit (0 = positive, 1 = negative)
2. d_n = Decimal digit (two per byte)
3. X = Bits have no significance; 8087 ignores when loading, zeros when storing.
4. ▲ = Position of implicit binary point
5. I = Integer bit of significant; stored in temporary real, implicit in short and long real.
6. Exponent Bias (normalized values):
   Short Real: 127 (7FH)
   Long Real: 1023 (3FFH)
   Temporary Real: 16383 (3FFFH)
7. Packed BCD: $(-1)^s (D_{17} \ldots D_0)$
8. Real: $(-1)^s (2^{E\text{-}BIAS})(F_0F_1 \ldots)$

Table 6 lists the 80287's instructions by class. No special programming tools are necessary to use the 80287 since all new instructions and data types are directly supported by the 80286 assembler and appropriate high level languages. All 8086/8088 development tools which support the 8087 can also be used to develop software for the 80286/80287 in real address mode.

SOFTWARE INTERFACE

The 80286/80287 is programmed as a single processor. All communication between the 80286 and the 80287 is transparent to software. The CPU automatically controls the 80287 whenever a numeric instruction is executed. All memory addressing modes, physical memory, and virtual memory of the CPU are available for use by the NPX.

Since the NPX operates in parallel with the CPU, any errors detected by the NPX may be reported after the CPU has executed the ESCAPE instruction which caused it. To allow identification of the failing numeric instruction, the NPX contains two pointer registers which identify the address of the failing numeric instruction and the numeric memory operand if appropriate for the instruction encountering this error.

INTERRUPT DESCRIPTION

Several interrupts of the 80286 are used to report exceptional conditions while executing numeric programs in either real or protected mode. The interrupts and their functions are shown in Table 3.

PROCESSOR ARCHITECTURE

As shown in Figure 1, the NPX is internally divided into two processing elements, the bus interface unit (BIU) and the numeric execution unit (NEU). The NEU executes all numeric instructions, while the BIU receives and decodes instructions, requests operand transfers to and from memory and executes processor control instructions. The two units are able to operate independently of one another allowing the BIU to maintain asynchronous communication with the CPU while the NEU is busy processing a numeric instruction.

BUS INTERFACE UNIT

The BIU decodes the ESC instruction executed by the CPU. If the ESC code defines a math instruction, the BIU transmits the formatted instruction to the NEU. If the ESC code defines an administrative instruction, the BIU executes it independently of the NEU. The parallel operation of the NPX with the CPU is normally transparent to the user. The BIU generates the $\overline{\text{BUSY}}$ and $\overline{\text{ERROR}}$ signals for 80826/80287 processor synchronization and error notification, respectively.

The 80287 executes a single numeric instruction at a time. When executing most ESC instructions, the

Table 3. 80286 Interrupt Vectors Reserved for NPX

Interrupt Number	Interrupt Function
7	An ESC instruction was encountered when EM or TS of the 80286 MSW was set. EM = 1 indicates that software emulation of the instruction is required. When TS is set, either an ESC or WAIT instruction will cause interrupt 7. This indicates that the current NPX context may not belong to the current task.
9	The second or subsequent words of a numeric operand in memory exceeded a segment's limit. This interrupt occurs after executing an ESC instruction. The saved return address will not point at the numeric instruction causing this interrupt. After processing the addressing error, the 80286 program can be restarted at the return address with IRET. The address of the failing numeric instruction and numeric operand and saved in the 80287. An interrupt handler for this interrupt *must* execute FNINIT before *any* other ESC or WAIT instruction.
13	The starting address of a numeric operand is not in the segment's limit. The return address will point at the ESC instruction, including prefixes, causing this error. The 80287 has not executed this instruction. The instruction and data address is 80287 refer to a previous, correctly executed, instruction.
16	The previous numeric instruction caused an unmasked numeric error. The address of the faulty numeric instruction or numeric data operand is stored in the 80287. Only ESC or WAIT instructions can cause this interrupt. The 80286 return address will point at a WAIT or ESC instruction, including prefixes, which may be restarted after clearing the error condition in the NPX.

80286 tests the $\overline{\text{BUSY}}$ pin and waits until the 80287 indicates that it is not busy before initiating the command. Once initiated, the 80286 continues program execution while the 80287 executes the ESC instruction. In 8086/8087 systems, this synchronization is achieved by placing a WAIT instruction before an ESC instruction. For most ESC instructions, the 80287 does not require a WAIT instruction before the ESC opcode. However, the 80287 will operate correctly with these WAIT instruction. In all cases, a WAIT or ESC instruction should be inserted after any 80287 store to memory (except FSTSW and FSTCW) or load from memory (except FLDENV or FRSTOR) before the 80286 reads or changes the value to be sure the numeric value has already been wrtten or read by the NPX.

Data transfers between memory and the 80287, when needed, are controlled by the PEREQ $\overline{\text{PEACK}}$, $\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$, $\overline{\text{NPS1}}$, NPS2 signals. The 80286 does the actual data transfer with memory through its processor extension data channel. Numeric data transfers with memory performed by the 80286 use the same timing as any other bus cycle. Control signal for the 80287 are generated by the 80826 as shown in Figure 4a, and meet the timing requirements shown in the AC requirements section.

NUMERIC EXECUTION UNIT

The NEU executes all instructions that involve the register stack; these include arithmetic, logical, transcendental, constant and data transfer instructions. The data path in the NEU is 84 bits wide (68 significand bits, 15 exponent bits and a sign bit) which allows internal operand transfers to be performed at very high speeds.

When the NEU begins executing an instruction, it activated the BIU $\overline{\text{BUSY}}$ signal. This signal is used in conjunction with the CPU WAIT instruction or automatically with most of the ESC instructions to synchronize both processors.

REGISTER SET

The 80287 register set is shown in Figure 5. Each of the eight data registers in the 8087's register stack

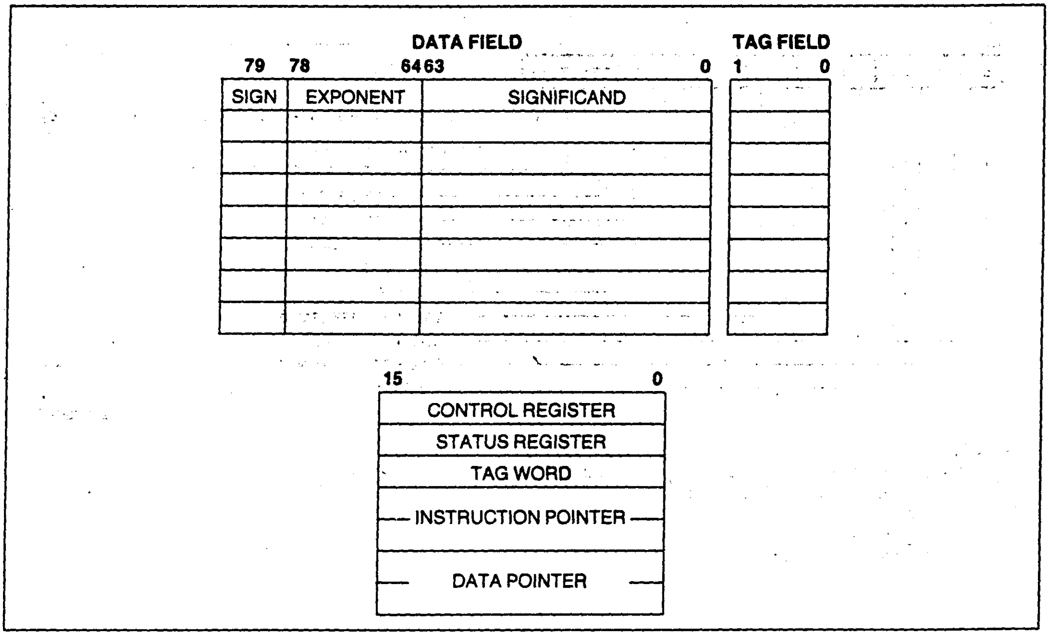


Figure 5. 80287 Register Set

is 80 bits wide and is divided into "fields" corresponding to the NPX's temporary real data type.

At a given point in time the TOP field in the status word identifies the current top-of-stack register. A "push" operation decrements TOP by 1 and loads a value into the new top register. A "pop" operation stores the value from the current top register and then increments TOP by 1. Like 80286 stacks in memory, the 80287 register stack grows "down" toward lower-addressed registers.

Instructions may address the data registers either implicitly or explicitly. Many instructions operate on the register at the TOP of the stack. These instructions implicitly address the register pointed by the TOP. Other instructions allow the programmer to explicitly specify the register which is to be used. This explicit register addressing is also "top-relative."

STATUS WORD

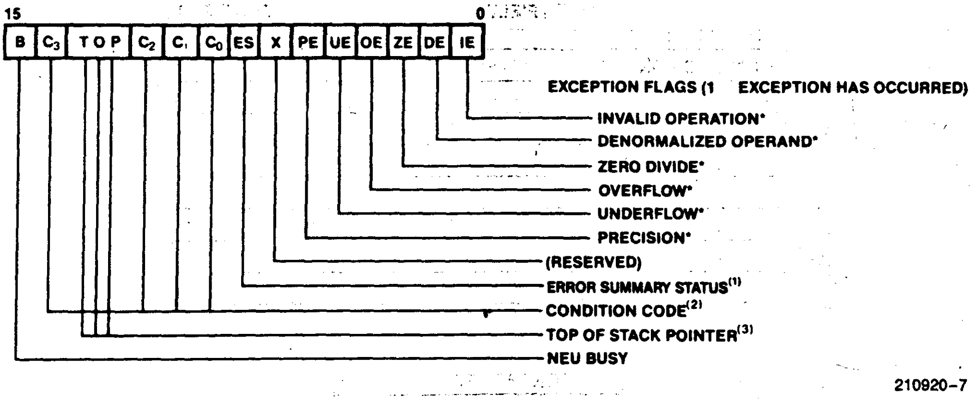
The 16-bit status word (in the status register) shown in Figure 6 reflects the overall state of the 80287. It may be read and inspected by CPU code. The busy bit (bit 15) indicates whether the NEU is executing an instruction (B = 1) or is idle (B = 0).

The instructions FSTSW, FSTSW AX, FSTENV, and FSAVE which store the status word are executed exclusively by the BIU and do not set the busy bit themselves or require the Busy bit be cleared in order to be executed.

The four numeric condition code bits (C_0–C_3) are similar to the flags in a CPU: instructions that perform arithmetic operations update these bits to reflect the outcome of NPX operations. The effect of these instructions on the condition code is summarized in Tables 4a and 4b.

Bits 14–12 of the status word point to the 80287 register that is the current top-of-stack (TOP) as described above. Figure 6 shows the six error flags in bits 5–0 of the status word. Bits 5–0 are set to indicate that the NEU has detected an exception while executing an instruction. The section on exception handling explains how they are set and used.

Bit 7 is the error summary status bit. This bit is set if any unmasked exception bit is set and cleared otherwise. If this bit is set, the ERROR signal is asserted.



NOTES:
1. ES is set if any unmasked exception bit is set, cleared otherwise.
2. See Table 5 for condition code interpretation.
3. Top Values
000 = Register 0 is Top of Stack
001 = Register 1 is Top of Stack
•
•
•
111 = Register 7 is Top of Stack.

*For definitions, see the section on exception handling

Figure 6. 80287 Status Word

TAG WORD

The tag word marks the content of each register as shown in Figure 7. The principal function of the tag word is to optimize the NPX's performance. The eight two-bit tags in the tag word can be used, however, to interpret the contents of 80287 registers.

INSTRUCTION AND DATA POINTERS

The instruction and data pointers (See Figures 8a and 8b) are provided for user-written error handlers. Whenever the 80287 executes a new instruction, the BIU saves the instruction address, the operand address (if present) and the instruction opcode. 80287 instructions can store this data into memory.

The instruction and data pointers appear in one of two formats depending on the operating mode of the 80287. In real mode, these values are the 20-bit physical address and 11-bit opcode formatted like the 8087. In protection mode, these values are the 32-bit virtual address used by the program which executed an ESC instruction. The same FLDENV/FSTENV/FSAVE/FRSTOR instructions as those of the 8087 are used to transfer these values between the 80287 registers and memory.

The saved instruction address in the 80287 will point at any prefixes which preceded the instruction. This is different than in the 8087 which only pointed at the ESCAPE instruction opcode.

CONTROL WORD

The NPX provides several processing options which are selected by loading a word from memory into the control word. Figure 9 shows the format and encoding of fields in the control word.

The low order byte of this control word configures the 80287 error and exception masking. Bits 5–0 of the control word contain individual masks for each of the six exceptions that the 80287 recognizes. The high order byte of the control word configures the

Table 4a. Condition Code Interpretation

Instruction Type	C_3	C_2	C_1	C_0	Interpretation
Compare, Test	0	0	X	0	ST > Source or 0 (FTST)
	0	0	X	1	ST < Source or 0 (FTST)
	1	0	X	0	ST = Source or 0 (FTST)
	1	1	X	1	ST is not comparable
Remainder	Q_1	0	Q_0	Q_2	Complete reduction with three low bits of quotient (See Table 5b)
	U	1	U	U	Incomplete Reduction
Examine	0	0	0	0	Valid, positive unnormalized
	0	0	0	1	Invalid, positive, exponent = 0
	0	0	1	0	Valid, negative, unnormalized
	0	0	1	1	Invalid, negative, exponent = 0
	0	1	0	0	Valid, positive, normalized
	0	1	0	1	Infinity, positive
	0	1	1	0	Valid, negative, normalized
	0	1	1	1	Infinity, negative
	1	0	0	0	Zero, positive
	1	0	0	1	Empty
	1	0	1	0	Zero, Negative
	1	0	1	1	Empty
	1	1	0	0	Invalid, positive, exponent = 0
	1	1	0	1	Empty
	1	1	1	0	Invalid, negative, exponent = 0
	1	1	1	1	Empty

NOTES:

1. ST = Top of Stack
2. X = value is not affected by instruction
3. U = value is undefined following instruction
4. Q_n = Quotient bit n

Table 4b. Condition Code Interpretation after FPREM (See Note 1) Instruction as a Function of Dividend Value

Dividend Range	Q_2	Q_1	Q_0
Dividend < 2 * Modulus	C_3	C_1	Q_0
Dividend < 4 * Modulus	C_3	Q_1	Q_0
Dividend ≥ 4 * Modulus	Q_2	Q_1	Q_0

NOTE:

1. Previous value of indicated bit, not affected by FPREM instruction execution.

80287 operating mode including precision, rounding, and infinity control. The precision control bits (bits 9–8) can be used to set the 80287 internal operating precision at less than the default of temporary real (80-bit) precision. This can be useful in providing compatibility with the early generation arithmetic processors of smaller precision than the 80287. The rounding control bits (bits 11–10) provide for directed rounding and true chop as well as the unbiased round to nearest even mode specified in the IEEE standard. Control over closure of the number space at infinity is also provided (either affine closure: ± ∞, or projective closure: ∞, is treated as unsigned, may be specified).

15							0
TAG (7)	TAG (6)	TAG (5)	TAG (4)	TAG (3)	TAG (2)	TAG (1)	TAG (0)

NOTE:
The index i of tag(i) is not top-relative. A program typically uses the "top" field of Status Word to determine which tag(i) field refers to logical top of stack.

TAG VALUES:
00 = VALID
01 = ZERO
10 = INVALID or INFINITY
11 = EMPTY

Figure 7. 80287 Tag Word

15 0	MEMORY OFFSET
CONTROL WORD	+0
STATUS WORD	+2
TAG WORD	+4
IP OFFSET	+6
CS SELECTOR	+8
DATA OPERAND OFFSET	+10
DATA OPERAND SELECTOR	+12

Figure 8a. Protected Mode 80287 Instruction and Data Pointer Image in Memory

EXCEPTION HANDLING

The 80287 detects six different exception conditions that can occur during instruction execution. Any or all exceptions will cause the assertion of external $\overline{\text{ERROR}}$ signal and ES bit of the Status Word if the appropriate exception masks are not set.

The exceptions that the 80287 detects and the 'default' procedures that will be carried out if the exception is masked, are as follows:

Invalid Operation: Stack overflow, stack underflow, indeterminate form (0/0, ∞, − ∞, etc) or the use of a Non-Number (NAN) as an operand. An exponent value of all ones and non-zero significand is reserved to identify NANs. If this exception is masked, the 80287 default response is to generate a specific NAN called INDEFINITE, or to propogate already existing NANs as the calculation result.

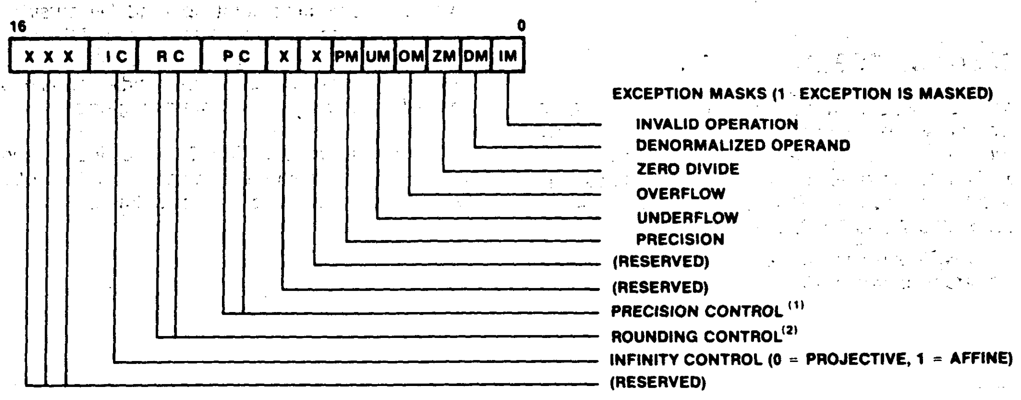
Overflow: The result is too large in magnitude to fit the specified format. The 80287 will generate an encoding for infinity if this exception is masked.

Zero Divisor: The divisor is zero while the dividend is a non-infinite, non-zero number. Again, the 80287 will generate an encoding for infinity if this exception is masked.

Underflow: The result in non-zero but too small in magnitude to fit in the specified format. If this exception is masked the 80287 will denormalize (shift right) the fraction until the exponent is in range. The process is called gradual underflow.

15		0	MEMORY OFFSET
CONTROL WORD			+0
STATUS WORD			+2
TAG WORD			+4
INSTRUCTION POINTER (15–0)			+6
INSTRUCTION POINTER (19–16)	0	INSTRUCTION OPCODE (10–0)	+8
DATA POINTER (15–0)			+10
DATA POINTER (19–16)	0		+12
15 12	11	0	

Figure 8b. Real Mode 80287 Instruction and Data Pointer Image in Memory



(1) PRECISION CONTROL
00 = 24 BITS (SHORT REAL)
01 = RESERVED
10 = 53 BITS (LONG REAL)
11 = 64 BITS (TEMP REAL)

(2) ROUNDING CONTROL
00 = ROUND TO NEAREST OR EVEN
01 = ROUND DOWN (TOWARD −∞)
10 = ROUND UP (TOWARD +∞)
11 = CHOP (TRUNCATE TOWARD ZERO)

Figure 9. 80287 Control Word

Denormalized Operand: At least one of the operands is denormalized; it has the smallest exponent but a non-zero significand. Normal processing continues if this exception is masked off.

Inexact Result: The true result is not exactly representable in the specified format, the result is rounded according to the rounding mode, and this flag is set. If this exception is masked, processing will simply continue.

If the error is not masked, the corresponding error bit and the error status bit (ES) in the control word will be set, and the $\overline{\text{ERROR}}$ output signal will be asserted. If the CPU attempts to execute another ESC or WAIT instruction, exception 7 will occur.

The error condition must be resolved via an interrupt service routine. The 80287 saves the address of the floating point instruction causing the error as well as the address of the lowest memory location of any memory operand required by that instruction.

8086/8087 COMPATIBILITY:

The 80286/80287 supports portability of 8086/8087 programs when it is in the real address mode. However, because of differences in the numeric error handling techniques, error handling routines *may* need to be changed. The differences between an 80286/80287 and 8086/8087 are:

1. The NPX error signal does not pass through an interrupt controller (8087 INT signal does). Therefore, any interrupt controller oriented instructions for the 8086/8087 may have to be deleted.
2. Interrupt vector 16 must point at the numeric error handler routine.
3. The saved floating point instruction address in the 80287 includes any leading prefixes before the ESCAPE opcode. The corresponding saved address of the 8087 does not include leading prefixes.
4. In protected mode, the format of the saved instruction and operand pointers is different than for the 8087. The instruction opcode is not saved—it must be read from memory if needed.
5. Interrupt 7 will occur when executing ESC instructions with either TS or EM or MSW = 1. If TS of MSW = 1 then WAIT will also cause interrupt 7. An interrupt handler should be added to handle this situation.
6. Interrupt 9 will occur if the second or subsequent words of a floating point operand fall outside a segment's size. Interrupt 13 will occur if the starting address of a numeric operand falls outside a segment's size. An interrupt handler should be added to report these programming errors.

In the protected mode, 8086/8087 application code can be directly ported via recompilation if the 80286 memory protection rules are not violated.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ...... 0°C to 70°C
Storage Temperature .... −65°C to +150°C
Case Temperature .... 0°C to 85°C
Voltage on any Pin with
Respect to Ground .... −1.0 to +7V
Power Dissipation .... 3.0 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 T_A = 0°C to 70°C, T_C = 0°C to 85°C, V_{CC} = 5V ±5%

ALL SPEEDS SELECTIONS

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_{IHC}	Clock Input HIGH Voltage CKM = 1: CKM = 0:	 2.0 3.8	 V_{CC} +1 V_{CC} +1	 V V	
V_{ILC}	Clock Input LOW Voltage CKM = 1 CKM = 0	 −0.5 −0.5	 0.8 0.6	 V V	
V_{OL}	Output LOW Voltage		0.45	V	I_{OL} = 3.0 mA
V_{OH}	Output HIGH Voltage	2.4		V	I_{OH} = −400 μA
I_{LI}	Input Leakage Current	•	±10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current	•	±10	μA	0.45V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	Power Supply Current	•	600 475 375	mA mA mA	T_A = 0°C T_A = 25°C T_A = 70°C
C_{IN}	Input Capacitance	•	10	pF	F_C = MHz
C_O	Input/Output Capacitance (D0–D15)	•	20	pF	V_C = 1 MHz
C_{CLK}	CLK Capacitance	•	12	pF	F_C = 1 MHz

A.C. CHARACTERISTICS T_A = 0°C to 70°C, T_{CASE} = 0°C to 85°C, V_{CC} = 5V ±5%

TIMING REQUIREMENTS

A.C. timings are referenced to 0.8V and 2.0V points on signals unless otherwise noted.

Symbol	Parameter	80287-3 5 MHz		80287-6 6 MHz		80287-8 8 MHz		80287-10 10 MHz Preliminary		Units	Test Conditions
		Min	Max	Min	Max	Min	Max	Min	Max		
T_{CLCL}	CLK Period CKM = 1: CKM = 0:	 200 62.5	 500 250	 166 62.5	 500 166	 125 50	 500 166	 100 40	 500 166	 ns ns	
T_{CLCH}	CLK LOW Time CKM = 1: CKM = 0:	 118 15	 230	 100 15	 343 146	 68 15	 343 146	 62 11	 343 146	 ns ns	 At 0.8V At 0.6V
T_{CHCL}	CLK HIGH Time CKM = 1: CKM = 0:	 69 20	 235	 50 20	 230 151	 43 20	 230 151	 28 18	 230 151	 ns ns	 At 2.0V At 3.6V
T_{CH1CH2}	CLK Rise Time		10		10		10		10	ns	1.0V to 3.6V if CKM = 0
T_{CL2CL1}	CLK Fall Time		10		10		10		10	ns	3.6V to 1.0V if CKM = 0
T_{DYWH}	Data Setup to $\overline{\text{NPWR}}$ Inactive	75		75		75		75		ns	
T_{WHDX}	Data Hold from $\overline{\text{NPWR}}$ Inactive	30		30		18		18		ns	
T_{WLWH} T_{RLRH}	$\overline{\text{NPWR}}$ $\overline{\text{NPRD}}$ Active Time	95		95		90		90		ns	At 0.8V
T_{AVWL} T_{AVRL}	Command Valid to $\overline{\text{NPWR}}$ or $\overline{\text{NPRD}}$Active	0		0		0		0		ns	
T_{MHRL}	Minimum Delay from PEREQ Active to $\overline{\text{NPRD}}$ Active	130		130		130		100		ns	
T_{KLKH}	$\overline{\text{PEAK}}$ Active Time	85		85		85		60		ns	At 0.8V
T_{KHKL}	$\overline{\text{PEAK}}$ Inactive Time	250		250		250		200		ns	At 2.0V
T_{KHCH}	$\overline{\text{PEAK}}$ Inactive to $\overline{\text{NPWR}}$, $\overline{\text{NPRD}}$ Inactive	50		50		40		40		ns	
T_{CHKL}	$\overline{\text{NPWR}}$, $\overline{\text{NPRD}}$ Inactive to $\overline{\text{PEAK}}$ Active	−30		−30		−30		−30		ns	
T_{WHAX} T_{RHAX}	Command Hold from $\overline{\text{NPWR}}$, $\overline{\text{NPRD}}$ Inactive	30		30		30		22		ns	
T_{KLCL}	$\overline{\text{PEAK}}$ Active Setup to $\overline{\text{NPWR}}$ $\overline{\text{NPRD}}$ Active	50		50		40		40		ns	

A.C. CHARACTERISTICS T_A = 0°C to 70°C, T_{CASE} = 0°C to 85°C, V_{CC} = 5V ±5% (Continued)

TIMING REQUIREMENTS (Continued)

A.C. timings are referenced to 0.8V and 2.0V points on signals unless otherwise noted.

Symbol	Parameter	80287-3 5 MHz		80287-6 6 MHz		80287-8 8 MHz		80287-10 10 MHz Preliminary		Units	Test Conditions
		Min	Max	Min	Max	Min	Max	Min	Max		
T_{IVCL}	$\overline{NPWR}$, $\overline{NPRD}$ to CLK Setup Time	70		70		70		53		ns	(Note 1)
T_{CLIH}	$\overline{NPWR}$, $\overline{NPRD}$ from CLK Hold Time	45		45		45		37		ns	(Note 1)
T_{RSCL}	RESET to CLK Setup Time	20		20		20		20		ns	(Note 1)
T_{CLRS}	RESET from CLK Hold Time	20		20		20		20		ns	(Note 1)

TIMING RESPONSES

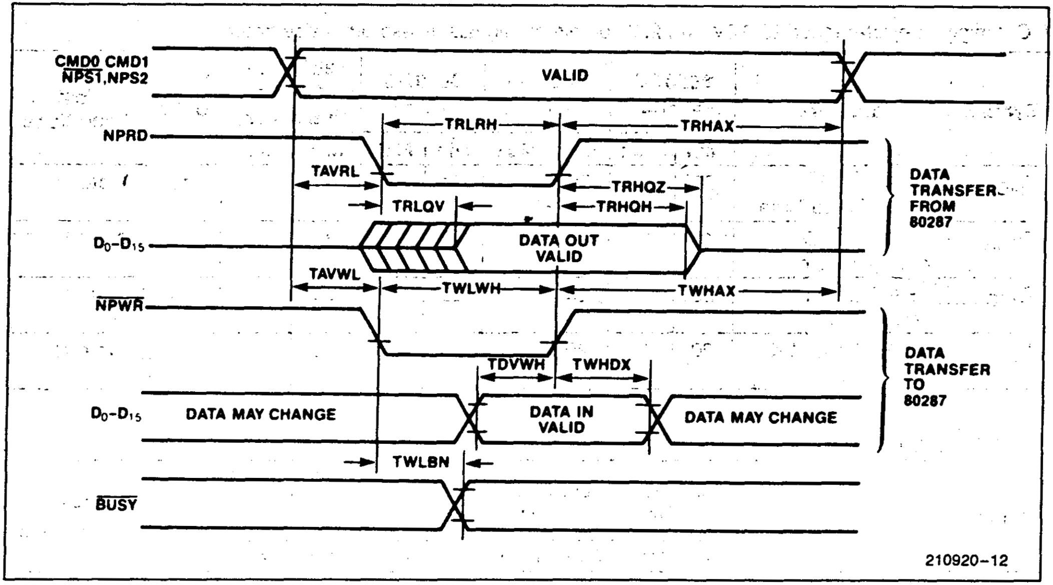
Symbol	Parameter	80287-3 5 MHz		80287-6 6 MHz		80287-8 8 MHz		80287-10 10 MHz Preliminary		Units	Test Conditions
		Min	Max	Min	Max	Min	Max	Min	Max		
T_{RHQZ}	$\overline{NPRD}$ Inactive to Data Float		37.5		37.5		35		21	ns	(Note 2)
T_{RLOV}	$\overline{NPRD}$ Active to Data Valid		60		60		60		60	ns	(Note 3)
T_{ILBH}	$\overline{ERROR}$ Active to $\overline{BUSY}$ Inactive	100		100		100		100		ns	(Note 4)
T_{WLBV}	$\overline{NPWR}$ Active to $\overline{BUSY}$ Active		100		100		100		100	ns	(Note 5)
T_{KLML}	$\overline{PEAK}$ Active to $\overline{PEREQ}$ Inactive		127		127		127		100	ns	(Note 6)
T_{CMDI}	Command Inactive Time Write-to-Write Read-to-Read Write-to-Read Read-to-Write	 95 250 105 95		 95 95 95 95		 95 95 95 95		 75 75 75 75		 ns ns ns ns	 At 2.0V At 2.0V At 2.0V At 2.0V
T_{RHQH}	Data Hold from $\overline{NPRD}$ Inactive	5		3		3		3		ns	(Note 7)

NOTES:

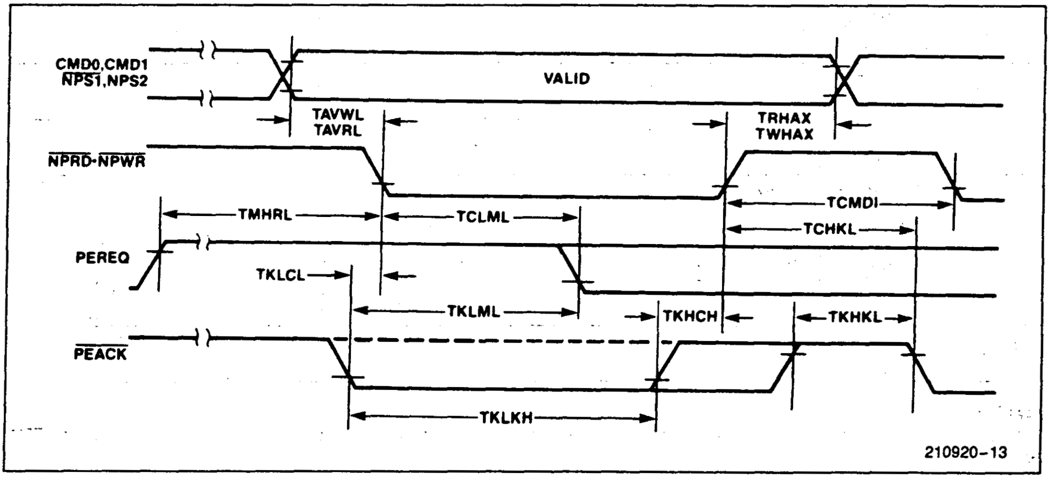
1. This is an asynchronous input. This specification is given for testing purposes only, to assure recognition at a specific CLK edge.
2. Float condition occurs when output current is less than I_{LO} on D0–D15.
3. D0–D15 loSINF¢: XL = 100 pF.
4. $\overline{BUSY}$ loading: CL = 100 pF.
5. $\overline{BUSY}$ loading: CL = 100 pF.
6. On last data transfer on numeric instruction.
7. D0–D15 loading: CL = 100 pF.

WAVEFORMS

DATA TRANSFER TIMING (Initiated by 80286)

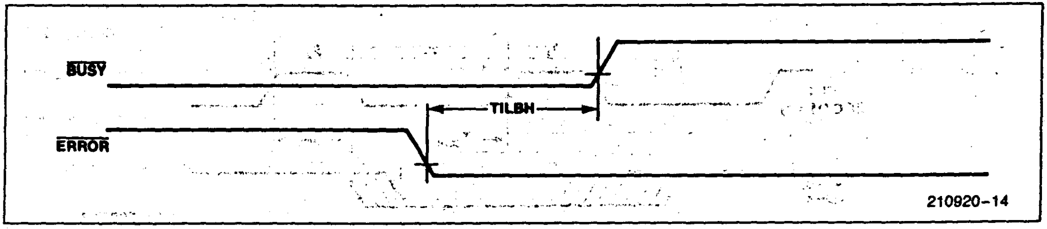


DATA CHANNEL TIMING (Initiated by 80287)

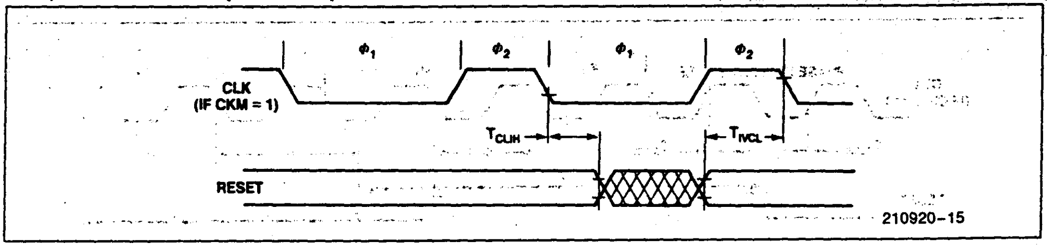


WAVEFORMS (Continued)

ERROR OUTPUT TIMING



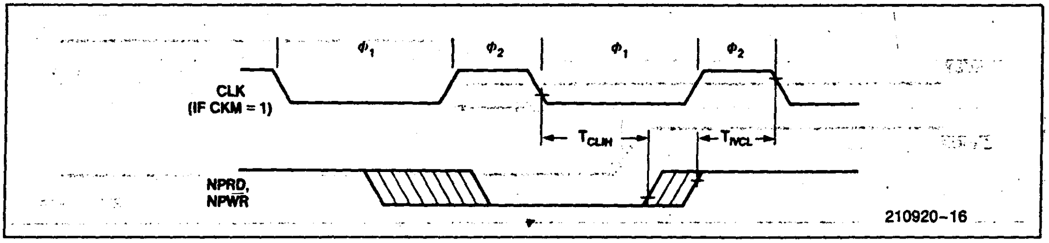
CLK, RESET TIMING (CKM = 1)



NOTE:
Reset, NPWR, NPRD are inputs asynchronous to CLK. Timing requirements on this page are given for testing purposes only, to assure recognition at a specific CLK edge.

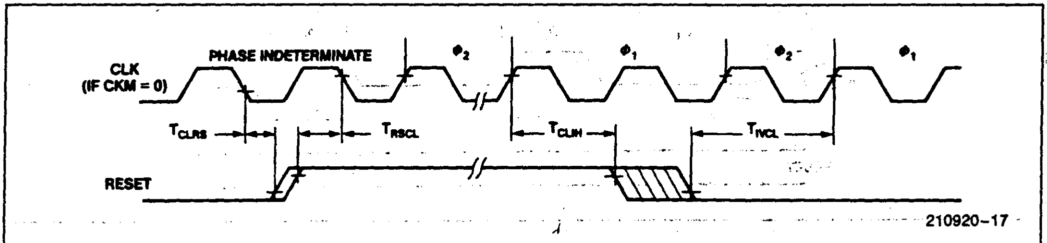
WAVEFORMS (Continued)

CLK, $\overline{NPRD}$, $\overline{NPWR}$ TIMING (CKM = 1)



210920-16

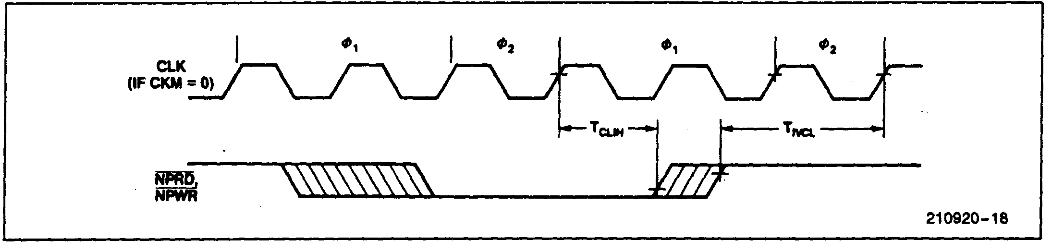
CLK, RESET TIMING (CKM = 0)



210920-17

NOTE:
Reset must meet timing shown to guarantee known phase of internal ÷ 3 circuit.

CLK, $\overline{NPRD}$, $\overline{NPWR}$ TIMING (CKM = 0)

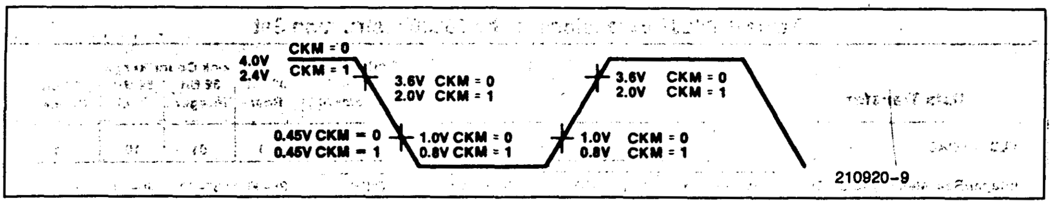


210920-18

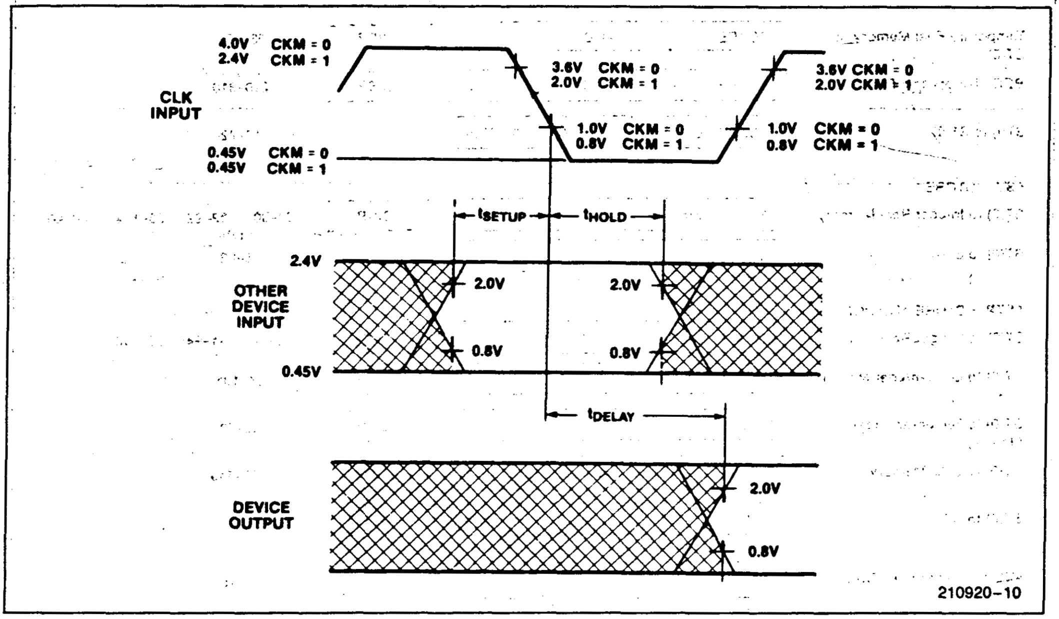
Table 6. 80287 Extensions to the 80286 Instruction Set

Data Transfer			Optional 8,16 Bit Displacement	Clock Count Range			
				32 Bit Real	32 Bit Integer	64 Bit Real	16 Bit Integer
FLD = LOAD	MF =			00	01	10	11
Integer/Real Memory to ST(0)	ESCAPE MF 1	MOD 0 0 0 R/M	DISP	38–56	52–60	40–60	46–54
Long Integer Memory to ST(0)	ESCAPE 1 1 1	MOD 1 0 1 R/M	DISP	60–68			
Temporary Real Memory to ST(0)	ESCAPE 0 1 1	MOD 1 0 1 R/M	DISP	53–65			
BCD Memory to ST(0)	ESCAPE 1 1 1	MOD 1 0 0 R/M	DISP	290–310			
ST(i) to ST(0)	ESCAPE 0 0 1	1 1 0 0 0 ST(i)		17–22			
FST = STORE							
ST(0) to Integer/Real Memory	ESCAPE MF 1	MOD 0 1 0 R/M	DISP	84–90	82–92	96–104	80–90
ST(0) to ST(i)	ESCAPE 1 0 1	1 1 0 1 0 ST(i)		15–22			
FSTP = STORE AND POP							
ST(0) to Integer/Real Memory	ESCAPE MF 1	MOD 0 1 1 R/M	DISP	86–92	84–94	98–106	82–92
ST(0) to Long Integer Memory	ESCAPE 1 1 1	MOD 1 1 1 R/M	DISP	94–105			
ST(0) to Temporary Real Memory	ESCAPE 0 1 1	MOD 1 1 1 R/M	DISP	52–58			
ST(0) to BCD Memory	ESCAPE 1 1 1	MOD 1 1 0 R/M	DISP	520–540			
ST(0) to ST(i)	ESCAPE 1 0 1	1 1 0 1 1 ST(i)		17–24			
FXCH = Exchange ST(i) and ST(0)	ESCAPE 0 0 1	1 1 0 0 1 ST(i)		10–15			
Comparison							
FCOM = Compare							
Integer/Real Memory to ST(0)	ESCAPE MF 0	MOD 0 1 0 R/M	DISP	60–70	78–91	65–75	72–86
ST(i) to ST (0)	ESCAPE 0 0 0	1 1 0 1 0 ST(i)		40–50			
FCOMP = Compare and Pop							
Integer/Real Memory to ST(0)	ESCAPE MF 0	MOD 0 1 1 R/M	DISP	63–73	80–93	67–77	74–88
ST(i) to ST(0)	ESCAPE 0 0 0	1 1 0 1 1 ST(i)		45–52			
FCOMPP = Compare ST(1) to ST(0) and Pop Twice	ESCAPE 1 1 0	1 1 0 1 1 0 0 1		45–55			
FTST = Test ST(0)	ESCAPE 0 0 1	1 1 1 0 0 1 0 0		38–48			
FXAM = Examine ST(0)	ESCAPE 0 0 1	1 1 1 0 0 1 0 1		12–23			

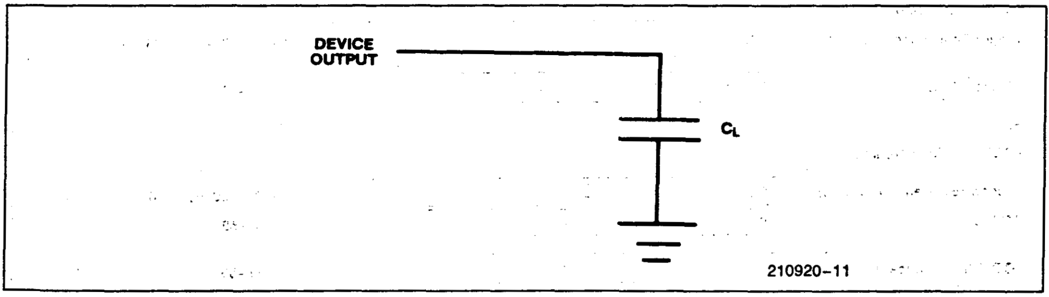
210920-19



AC Drive and Measurement Points—CLK Input



AC Setup, Hold and Delay Time Measurement—General



AC Test Loading on Outputs

Table 6. 80287 Extensions to the 80286 Instruction Set (Continued)

Constants			Optional 8,16 Bit Displacement	Clock Count Range 32 Bit Real	 32 Bit Integer	 64 Bit Real	 16 Bit Integer
	MF	=		00	01	10	11
FLDZ = LOAD + 0.0 into ST(0)	ESCAPE 0 0 1	1 1 1 0 1 1 1 0		11–17			
FLD1 = LOAD + 1.0 into ST(0)	ESCAPE 0 0 1	1 1 1 0 1 0 0 0		15–21			
FLDPI = LOAD π into ST(0)	ESCAPE 0 0 1	1 1 1 0 1 0 1 1		16–22			
FLDL2T = LOAD $\log_2 10$ into ST(0)	ESCAPE 0 0 1	1 1 1 0 1 0 0 1		16–22			
FLDL2E = LOAD $\log_2 e$ into ST(0)	ESCAPE 0 0 1	1 1 1 0 1 0 1 0		15–21			
FLDLG2 = LOAD $\log_{10} 2$ into ST(0)	ESCAPE 0 0 1	1 1 1 0 1 1 0 0		18–24			
FLDLN2 = LOAD $\log_e 2$ into ST(0)	ESCAPE 0 0 1	1 1 1 0 1 1 0 1		17–23			
Arithmetic							
FADD = Addition							
Integer/Real Memory with ST(0)	ESCAPE MF 0	MOD 0 0 0 R/M	DISP	90–120	108–143	95–125	102–137
ST(i) and ST(0)	ESCAPE d P 0	1 1 0 0 0 ST(i)		70–100 (Note 1)			
FSUB = Subtraction							
Integer/Real Memory with ST(0)	ESCAPE MF 0	MOD 1 0 R R/M	DISP	90–120	108–143	95–125	102–137
ST(i) and ST(0)	ESCAPE d P 0	1 1 1 0 R R/M		70–100 (Note 1)			
FMUL = Multiplication							
Integer/Real Memory with ST(0)	ESCAPE MF 0	MOD 0 0 1 R/M	DISP	110–125	130–144	112–168	124–138
ST(i) and ST(0)	ESCAPE d P 0	1 1 0 0 1 R/M		90–145 (Note 1)			
FDIV = Division							
Integer/Real Memory with ST(0)	ESCAPE MF 0	MOD 1 1 R R/M	DISP	215–225	230–243	220–230	224–238
ST(i) and ST(0)	ESCAPE d P 0	1 1 1 1 R R/M		193–203 (Note 1)			
FSQRT = Square Root of ST(0)	ESCAPE 0 0 1	1 1 1 1 1 0 1 0		180–186			
FSCALE = Scale ST(0) by ST(1)	ESCAPE 0 0 1	1 1 1 1 1 1 0 1		32–38			
FPREM = Partial Remainder of ST(0) ÷ST(1)	ESCAPE 0 0 1	1 1 1 1 1 0 0 0		15–190			
FRNDINT = Round ST(0) to Integer	ESCAPE 0 0 1	1 1 1 1 1 1 0 0		16–50			

210920–20

NOTE:
1. If P = 1 then add 5 clocks.

Table 6. 80287 Extensions to the 80286 Instruction Set (Continued)

			Optional 8,16 Bit Displacement	Clock Count Range
FXTRACT = Extract Components of St(0)	ESCAPE 0 0 1	1 1 1 1 0 1 0 0		27–55
FABS = Absolute Value of ST(0)	ESCAPE 0 0 1	1 1 1 0 0 0 0 1		10–17
FCHS = Change Sign of ST(0)	ESCAPE 0 0 1	1 1 1 0 0 0 0 0		10–17
Transcendental				
FPTAN = Partial Tangent of ST(0)	ESCAPE 0 0 1	1 1 1 1 0 0 1 0		30–540
FPATAN = Partial Arctangent of ST(0) ÷ST(1)	ESCAPE 0 0 1	1 1 1 1 0 0 1 1		250–800
F2XM1 = $2^{ST(0)}-1$	ESCAPE 0 0 1	1 1 1 1 0 0 0 0		310–630
FYL2X = ST(1) • Log_2 |ST(0)|	ESCAPE 0 0 1	1 1 1 1 0 0 0 1		900–1100
FYL2XP1 = ST(1) • Log_2 |ST(0) +1|	ESCAPE 0 0 1	1 1 1 1 1 0 0 1		700–1000
Processor Control				
FINIT = Initialize NPX	ESCAPE 0 1 1	1 1 1 0 0 0 1 1		2–8
FSETPM = Enter Protected Mode	ESCAPE 0 1 1	1 1 1 0 0 1 0 0		2–8
FSTSW AX = Store Control Word	ESCAPE 1 1 1	1 1 1 0 0 0 0 0		10–16
FLDCW = Load Control Word	ESCAPE 0 0 1	MOD 1 0 1 R/M	DISP	7–14
FSTCW = Store Control Word	ESCAPE 0 0 1	MOD 1 1 1 R/M	DISP	12–18
FSTSW = Store Status Word	ESCAPE 1 0 1	MOD 1 1 1 R/M	DISP	12–18
FCLEX = Clear Exceptions	ESCAPE 0 1 1	1 1 1 0 0 0 1 0		2–8
FSTENV = Store Environment	ESCAPE 0 0 1	MOD 1 1 0 R/M	DISP	40–50
FLDENV = Load Environment	ESCAPE 0 0 1	MOD 1 0 0 R/M	DISP	35–45
FSAVE = Save State	ESCAPE 1 0 1	MOD 1 1 0 R/M	DISP	205–215
FRSTOR = Restore State	ESCAPE 1 0 1	MOD 1 0 0 R/M	DISP	205–215
FINCSTP = Increment Stack Pointer	ESCAPE 0 0 1	1 1 1 1 0 1 1 1		6–12
FDECSTP = Decrement Stack Pointer	ESCAPE 0 0 1	1 1 1 1 0 1 1 0		6–12

210920–21

Table 6. 80287 Extensions to the 80286 Instruction Set (Continued)

			Clock Count Range
FFREE = Free ST(i)	ESCAPE 1 0 1	1 1 0 0 0 ST(i)	9–16
FNOP = No Operation	ESCAPE 0 0 1	1 1 0 1 0 0 0 0	10–16

210920–22

NOTES:

1. if mod = 00 then DISP = 0*, disp-low and disp-high are absent
   if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
   if mod = 10 then DISP = disp-high; disp-low
   if mod = 11 then r/m is treated as an ST(i) field
2. if r/m = 000 then EA = (BX) + (SI) +DISP
   if r/m = 001 then EA = (BX) + (DI) +DISP
   if r/m = 010 then EA = (BP) + (SI) +DISP
   if r/m = 011 then EA = (BP) + (DI) +DISP
   if r/m = 100 then EA = (SI) + DISP
   if r/m = 101 then EA = (DI) + DISP
   if r/m = 110 then EA = (BP) + DISP
   if r/m = 111 then EA = (BX) + DISP
   *except if mod = 000 and r/m = 110 then EA = disp-high; disp-low.
3. MF = Memory Format
   00—32-bit Real
   01—32-bit Integer
   10—64-bit Real
   11—16-bit Integer
4. ST(0) = Current stack top
   ST(i) = i^{th} register below stack top
5. d = Destination
   0—Destination is ST(0)
   1—Destination is ST(i)
6. P = Pop
   0—No pop
   1—Pop ST(0)
7. R = Reverse: When d = 1 reverse the sense of R
   0—Destination (op) Source
   1—Source (op) Destination
8. For **FSQRT**: $-0 \leq ST(0) \leq +\infty$
   For **FSCALE**: $-2^{15} \leq ST(1) < +2^{15}$ and ST(1) integer
   For **F2XM1**: $0 \leq ST(0) \leq 2^{-1}$
   For **FYL2X**: $0 < ST(0) < \infty$
   $-\infty < ST(1) < +\infty$
   For **FYL2XP1**: $0 \leq |ST(0)| < (2 - \sqrt{2})/2$
   $-\infty < ST(1) < \infty$
   For **FPTAN**: $0 \leq ST(0) \leq \pi/4$
   For **FPATAN**: $0 \leq ST(0) < ST(1) < +\infty$
9. ESCAPE bit pattern is 11011.

DATA SHEET REVISION REVIEW

The following list represents the key differences between this and the -006 80287 Data Sheet. Please review the summary carefully.

1. The CLK speed table in the section entitled "SYSTEM CONFIGURATION WITH 80286" was modified to show the required CLK frequencies in the divide-by-3 mode (CKM = 0) for the 287 speeds tabulated.
2. Obsolete components were replaced with readily available components in Figure 4A.
3. In the AC TIMING REQUIREMENTS table, the timing symbols, T_{AVRL} and T_{AVWL} were reversed in order to match the parameter description.

2.7 8087 and 80287 Compatibility

This section summarizes the differences between the 80387 and the 80287. Any migration from the 8087 directly to the 80387 must also take into account the differences between the 8087 and the 80287 as listed in Appendix A.

Many changes have been designed into the 80387 to directly support the IEEE standard in hardware. These changes result in increased performance by eliminating the need for software that supports the standard.

2.7.1 GENERAL DIFFERENCES

The 80387 supports only affine closure for infinity arithmetic, not projective closure. Bit 12 of the Control Word (CW) no longer defines infinity control. It is a reserved bit; but it is initialized to zero after RESET or FINIT and is changeable upon loading the CW. Programs must ignore this bit.

Operands for FSCALE and FPATAN are no longer restricted in range (except for ±∞); F2XM1 and FPTAN accept a wider range of operands.

The results of transcendental operations may be slightly different from those computed by 80287.

In the case of FPTAN, the 80387 supplies a true tangent result in ST(1), and (always) a floating point 1 in ST.

Rounding control is in effect for FLD *constant*.

Software cannot change entries of the tag word to values (other than empty) that do not reflect the actual register contents.

After reset, FINIT, and incomplete FPREM, the 80387 resets to zero the condition code bits C_3-C_0 of the status word.

In conformance with the IEEE standard, the 80387 does not support the special data formats: pseudozero, pseudo-NaN, pseudoinfinity, and unnormal.

Table 2.7. Exceptions

Exception	Cause	Default Action (If exception is masked)
Invalid Operation	Operation on a signaling NaN, unsupported format, indeterminate form (0*∞, 0/0, (+∞) + (−∞), etc.), or stack overflow/underflow (SF is also set).	Result is a quiet NaN, integer indefinite, or BCD indefinite
Denormalized Operand	At least one of the operands is denormalized, i.e. it has the smallest exponent but a nonzero significand.	Normal processing continues
Zero Divisor	The divisor is zero while the dividend is a noninfinite, nonzero number.	Result is ∞
Overflow	The result is too large in magnitude to fit in the specified format.	Result is largest finite value or ∞
Underflow	The true result is nonzero but too small to be represented in the specified format, and, if underflow exception is masked, denormalization causes loss of accuracy.	Result is denormalized or zero
Inexact Result (Precision)	The true result is not exactly representable in the specified format (e.g. 1/3); the result is rounded according to the rounding mode.	Normal processing continues

Note: the 80287XL and most 287-compatible chips use the 387 instruction set.

2.7.2 EXCEPTIONS

A number of differences exist due to changes in the IEEE standard and to functional improvements to the architecture of the 80387:

1. When the overflow or underflow exception is masked, the 80387 differs from the 80287 in rounding when overflow or underflow occurs. The 80387 produces results that are consistent with the rounding mode.
2. When the underflow exception is masked, the 80387 sets its underflow flag only if there is also a loss of accuracy during denormalization.
3. Fewer invalid-operation exceptions due to denormal operands, because the instructions FSQRT, FDIV, FPREM, and conversions to BCD or to integer normalize denormal operands before proceeding.
4. The FSQRT, FBSTP, and FPREM instructions may cause underflow, because they support denormal operands.
5. The denormal exception can occur during the transcendental instructions and the FXTRACT instruction.
6. The denormal exception no longer takes precedence over all other exceptions.
7. When the denormal exception is masked, the 80387 automatically normalizes denormal operands. The 8087/80287 performs unnormal arithmetic, which might produce an unnormal result.
8. When the operand is zero, the FXTRACT instruction reports a zero-divide exception and leaves $-\infty$ in ST(1).
9. The status word has a new bit (SF) that signals when invalid-operation exceptions are due to stack underflow or overflow.
10. FLD *extended precision* no longer reports denormal exceptions, because the instruction is not numeric.
11. FLD *single/double precision* when the operand is denormal converts the number to extended precision and signals the denormalized operand exception. When loading a signaling NaN, FLD *single/double precision* signals an invalid-operand exception.
12. The 80387 only generates quiet NaNs (as on the 80287); however, the 80387 distinguishes between quiet NaNs and signaling NaNs. Signaling NaNs trigger exceptions when they are used as operands; quiet NaNs do not (except for FCOM, FIST, and FBSTP which also raise IE for quiet NaNs).
13. When stack overflow occurs during FPTAN and overflow is masked, both ST(0) and ST(1) contain quiet NaNs. The 80287/8087 leaves the original operand in ST(1) intact.
14. When the scaling factor is $\pm\infty$, the FSCALE (ST(0), ST(1)) instruction behaves as follows (ST(0) and ST(1) contain the scaled and scaling operands respectively):
    - FSCALE(0, ∞) generates the invalid operation exception.
    - FSCALE(finite, $-\infty$) generates zero with the same sign as the scaled operand.
    - FSCALE(finite, $+\infty$) generates ∞ with the same sign as the scaled operand.

    The 8087/80287 returns zero in the first case and raises the invalid-operation exception in the other cases.
15. The 80387 returns signed infinity/zero as the unmasked response to massive overflow/underflow. The 8087 and 80287 support a limited range for the scaling factor; within this range either massive overflow/underflow do not occur or undefined results are produced.

3.0 HARDWARE INTERFACE

In the following description of hardware interface, the # symbol at the end of a signal name indicates that the active or asserted state occurs when the signal is at a low voltage. When no # is present after the signal name, the signal is asserted when at the high voltage level.

3.1 Signal Description

In the following signal descriptions, the 80387 pins are grouped by function as follows:

1. Execution control—CPUCLK2, NUMCLK2, CKM, RESETIN
2. NPX handshake—PEREQ, BUSY#, ERROR#
3. Bus interface pins—D31-D0, W/R#, ADS#, READY#, READYO#
4. Chip/Port Select—STEN, NPS1#, NPS2, CMD0#
5. Power supplies—V_{CC}, V_{SS}

Table 3.1 lists every pin by its identifier, gives a brief description of its function, and lists some of its characteristics. All output signals are tristate; they leave floating state only when STEN is active. The output buffers of the bidirectional data pins D31-D0 are also tristate; they leave floating state only in read cycles when the 80387 is selected (i.e. when STEN, NPS1#, and NPS2 are all active).

Figure 3.1 and Table 3.2 together show the location of every pin in the pin grid array.

<table>
<tr><th></th><th colspan="9">Instruction</th><th colspan="2" rowspan="2">Optional Fields</th></tr>
<tr><th></th><th colspan="4">First Byte</th><th colspan="5">Second Byte</th></tr>
<tr><td>1</td><td>11011</td><td colspan="2">OPA</td><td>1</td><td colspan="2">MOD</td><td>1</td><td>OPB</td><td>R/M</td><td>SIB</td><td>DISP</td></tr>
<tr><td>2</td><td>11011</td><td colspan="2">MF</td><td>OPA</td><td colspan="2">MOD</td><td colspan="2">OPB</td><td>R/M</td><td>SIB</td><td>DISP</td></tr>
<tr><td>3</td><td>11011</td><td>d</td><td>P</td><td>OPA</td><td>1</td><td>1</td><td colspan="2">OPB</td><td>ST(i)</td><td></td><td></td></tr>
<tr><td>4</td><td>11011</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td colspan="2">OP</td><td></td><td></td></tr>
<tr><td>5</td><td>11011</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td colspan="2">OP</td><td></td><td></td></tr>
<tr><td></td><td>15-11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4 3</td><td>2 1 0</td><td></td><td></td></tr>
</table>

6.0 80387 EXTENSIONS TO THE 386™ CPU INSTRUCTION SET

Instructions for the 80387 assume one of the five forms shown in the following table. In all cases, instructions are at least two bytes long and begin with the bit pattern 11011B, which identifies the ESCAPE class of instruction. Instructions that refer to memory operands specify addresses using the 386 CPU addressing modes.

OP = Instruction opcode, possible split into two fields OPA and OPB

MF = Memory Format
- 00—32-bit real
- 01—32-bit integer
- 10—64-bit real
- 11—16-bit integer

P = Pop
- 0—Do not pop stack
- 1—Pop stack after operation

ESC = 11011

d = Destination
- 0—Destination is ST(0)
- 1—Destination is ST(i)

R XOR d = 0—Destination (op) Source
R XOR d = 1—Source (op) Destination

ST(i) = Register stack element *i*
- 000 = Stack top
- 001 = Second stack element
- •
- •
- •
- 111 = Eighth stack element

MOD (Mode field) and R/M (Register/Memory specifier) have the same interpretation as the corresponding fields of the 386 Microprocessor instructions (refer to *386™ Microprocessor Programmer's Reference Manual*).

SIB (Scale Index Base) byte and DISP (displacement) are optionally present in instructions that have MOD and R/M fields. Their presence depends on the values of MOD and R/M, as for 386 Microprocessor instructions.

The instruction summaries that follow assume that the instruction has been prefetched, decoded, and is ready for execution; that bus cycles do not require wait states; that there are no local bus HOLD request delaying processor access to the bus; and that no exceptions are detected during instruction execution. If the instruction has MOD and R/M fields that call for both base and index registers, add one clock.

80387 Extensions to the 386™ CPU Instruction Set

Instruction	Encoding			Clock Count Range			
	Byte 0	Byte 1	Optional Bytes 2–6	32-Bit Real	32-Bit Integer	64-Bit Real	16-Bit Integer
DATA TRANSFER							
FLD = Load[a]							
Integer/real memory to ST(0)	ESC MF 1	MOD 000 R/M	SIB/DISP	20	45–52	25	61–65
Long integer memory to ST(0)	ESC 111	MOD 101 R/M	SIB/DISP	56–67			
Extended real memory to ST(0)	ESC 011	MOD 101 R/M	SIB/DISP	44			
BCD memory to ST(0)	ESC 111	MOD 100 R/M	SIB/DISP	266–275			
ST(i) to ST(0)	ESC 001	11000 ST(i)		14			
FST = Store							
ST(0) to integer/real memory	ESC MF 1	MOD 010 R/M	SIB/DISP	44	79–93	45	82–95
ST(0) to ST(i)	ESC 101	11010 ST(i)		11			
FSTP = Store and Pop							
ST(0) to integer/real memory	ESC MF 1	MOD 011 R/M	SIB/DISP	44	79–93	45	82–95
ST(0) to long integer memory	ESC 111	MOD 111 R/M	SIB/DISP	80–97			
ST(0) to extended real	ESC 011	MOD 111 R/M	SIB/DISP	53			
ST(0) to BCD memory	ESC 111	MOD 110 R/M	SIB/DISP	512–534			
ST(0) to ST(i)	ESC 101	11001 ST (i)		12			
FXCH = Exchange							
ST(i) and ST(0)	ESC 001	11001 ST(i)		18			
COMPARISON							
FCOM = Compare							
Integer/real memory to ST(0)	ESC MF 0	MOD 010 R/M	SIB/DISP	26	56–63	31	71–75
ST(i) to ST(0)	ESC 000	11010 ST(i)		24			
FCOMP = Compare and pop							
Integer/real memory to ST	ESC MF 0	MOD 011 R/M	SIB/DISP	26	56–63	31	71–75
ST(i) to ST(0)	ESC 000	11011 ST(i)		26			
FCOMPP = Compare and pop twice							
ST(1) to ST(0)	ESC 110	1101 1001		26			
FTST = Test ST(0)	ESC 001	1110 0100		28			
FUCOM = Unordered compare	ESC 101	11100 ST(i)		24			
FUCOMP = Unordered compare and pop	ESC 101	11101 ST(i)		26			
FUCOMPP = Unordered compare and pop twice	ESC 010	1110 1001		26			
FXAM = Examine ST(0)	ESC 001	11100101		30-38			
CONSTANTS							
FLDZ = Load +0.0 into ST(0)	ESC 001	1110 1110		20			
FLD1 = Load +1.0 into ST(0)	ESC 001	1110 1000		24			
FLDPI = Load pi into ST(0)	ESC 001	1110 1011		40			
FLDL2T = Load $\log_2(10)$ into ST(0)	ESC 001	1110 1001		40			

Shaded areas indicate instructions not available in 8087/80287.

NOTE:
a. When loading single- or double-precision zero from memory, add 5 clocks.

80387 Extensions to the 386™ CPU Instruction Set (Continued)

Instruction	Encoding			Clock Count Range			
	Byte 0	Byte 1	Optional Bytes 2–6	32-Bit Real	32-Bit Integer	64-Bit Real	16-Bit Integer
CONSTANTS (Continued)							
FLDL2E = Load log$_2$(e) into ST(0)	ESC 001	1110 1010		40			
FLDLG2 = Load log$_{10}$(2) into ST(0)	ESC 001	1110 1100		41			
FLDLN2 = Load log$_e$(2) into ST(0)	ESC 001	1110 1101		41			
ARITHMETIC							
FADD = Add							
Integer/real memory with ST(0)	ESC MF 0	MOD 000 R/M	SIB/DISP	24–32	57–72	29–37	71–85
ST(i) and ST(0)	ESC d P 0	11000 ST(i)		23–31[b]			
FSUB = Subtract							
Integer/real memory with ST(0)	ESC MF 0	MOD 10 R R/M	SIB/DISP	24–32	57–82	28–36	71–83[c]
ST(i) and ST(0)	ESC d P 0	1110 R R/M		26–34[d]			
FMUL = Multiply							
Integer/real memory with ST(0)	ESC MF 0	MOD 001 R/M	SIB/DISP	27–35	61–82	32–57	76–87
ST(i) and ST(0)	ESC d P 0	1100 1 R/M		29–57[e]			
FDIV = Divide							
Integer/real memory with ST(0)	ESC MF 0	MOD 11 R R/M	SIB/DISP	89	120–127[f]	94	136–140[g]
ST(i) and ST(0)	ESC d P 0	1111 R R/M		88[h]			
FSQRT[i] = Square root	ESC 001	1111 1010		122–129			
FSCALE = Scale ST(0) by ST(1)	ESC 001	1111 1101		67–86			
FPREM = Partial remainder	ESC 001	1111 1000		74–155			
FPREM1 = Partial remainder (IEEE)	ESC 001	1111 0101		95–185			
FRNDINT = Round ST(0) to integer	ESC 001	1111 1100		66–80			
FXTRACT = Extract components of ST(0)	ESC 001	1111 0100		70–76			
FABS = Absolute value of ST(0)	ESC 001	1110 0001		22			
FCHS = Change sign of ST(0)	ESC 001	1110 0000		24–25			

Shaded areas indicate instructions not available in 8087/80287.

NOTES:
b. Add 3 clocks to the range when d = 1.
c. Add 1 clock to **each** range when R = 1.
d. Add 3 clocks to the range when d = 0.
e. typical = 52 (When d = 0, 46–54, typical = 49).
f. Add 1 clock to the range when R = 1.
g. 135–141 when R = 1.
h. Add 3 clocks to the range when d = 1.
i. $-0 \leq ST(0) \leq +\infty$.

80387 Extensions to the 386™ CPU Instruction Set (Continued)

Instruction	Encoding			Clock Count Range
	Byte 0	Byte 1	Optional Bytes 2-6	
TRANSCENDENTAL				
FCOS[k] = Cosine of ST(0)	ESC 001	1111 1111		123–772[j]
FPTAN[k] = Partial tangent of ST(0)	ESC 001	1111 0010		191–497[j]
FPATAN = Partial arctangent	ESC 001	1111 0011		314–487
FSIN[k] = Sine of ST(0)	ESC 001	1111 1110		122–771[j]
FSINCOS[k] = Sine and cosine of ST(0)	ESC 001	1111 1011		194–809[j]
F2XM1[l] = $2^{ST(0)} - 1$	ESC 001	1111 0000		211–476
FYL2X[m] = ST(1) * $\log_2$(ST(0))	ESC 001	1111 0001		120–538
FYL2XP1[n] = ST(1) * $\log_2$(ST(0) + 1.0)	ESC 001	1111 1001		257–547
PROCESSOR CONTROL				
FINIT = Initialize NPX	ESC 011	1110 0011		33
FSTSW AX = Store status word	ESC 111	1110 0000		13
FLDCW = Load control word	ESC 001	MOD 101 R/M	SIB/DISP	19
FSTCW = Store control word	ESC 101	MOD 111 R/M	SIB/DISP	15
FSTSW = Store status word	ESC 101	MOD 111 R/M	SIB/DISP	15
FCLEX = Clear exceptions	ESC 011	1110 0010		11
FSTENV = Store environment	ESC 001	MOD 110 R/M	SIB/DISP	103–104
FLDENV = Load environment	ESC 001	MOD 100 R/M	SIB/DISP	71
FSAVE = Save state	ESC 101	MOD 110 R/M	SIB/DISP	375–376
FRSTOR = Restore state	ESC 101	MOD 100 R/M	SIB/DISP	308
FINCSTP = Increment stack pointer	ESC 001	1111 0111		21
FDECSTP = Decrement stack pointer	ESC 001	1111 0110		22
FFREE = Free ST(i)	ESC 101	1100 0 ST(i)		18
FNOP = No operations	ESC 001	1101 0000		12

Shaded areas indicate instructions not available in 8087/80287.

NOTES:

j. These timings hold for operands in the range $|x| < \pi/4$. For operands not in this range, up to 76 additional clocks may be needed to reduce the operand.

k. $0 \leq |ST(0)| < 2^{63}$.

l. $-1.0 \leq ST(0) \leq 1.0$.

m. $0 \leq ST(0) < \infty, -\infty < ST(1) < +\infty$.

n. $0 \leq |ST(0)| < (2 - SQRT(2))/2, -\infty < ST(1) < +\infty$.

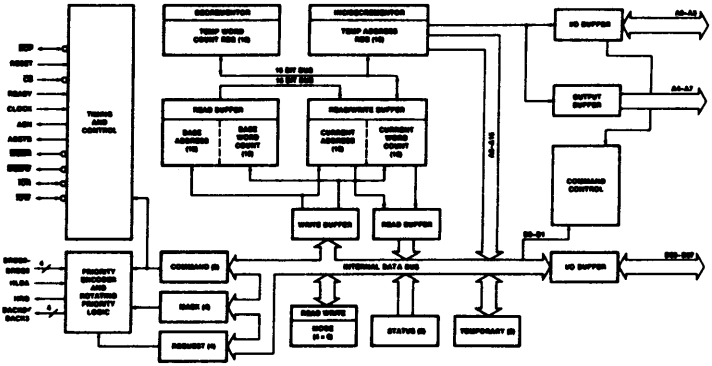
8237A
HIGH PERFORMANCE PROGRAMMABLE DMA CONTROLLER (8237A, 8237A-4, 8237A-5)

- **Enable/Disable Control of Individual DMA Requests**
- **Four Independent DMA Channels**
- **Independent Autoinitialization of All Channels**
- **Memory-to-Memory Transfers**
- **Memory Block Initialization**
- **Address Increment or Decrement**
- **High Performance: Transfers up to 1.6M Bytes/Second with 5 MHz 8237A-5**
- **Directly Expandable to Any Number of Channels**
- **End of Process Input for Terminating Transfers**
- **Software DMA Requests**
- **Independent Polarity Control for DREQ and DACK Signals**
- **Available in EXPRESS —Standard Temperature Range**
- **Available in 40-Lead Cerdip and Plastic Packages**
  (See Packaging Spec, Order #231369)

The 8237A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information from the system memory. Memory-to-memory transfer capability is also provided. The 8237A offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

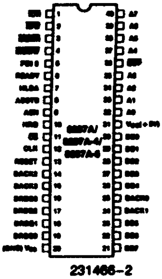
The 8237A is designed to be used in conjunction with an external 8-bit address latch. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips. The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process (EOP). Each channel has a full 64K address and word count capability.

The 8273A-4 and 8237A-5 are 4 MHz and 5 MHz versions of the standard 3 MHz 8237A respectively.



231466-1

Figure 1. Block Diagram



231466-2

Figure 2. Pin Configuration

Table 1. Pin Description

Symbol	Type	Name and Function
V_{CC}		**POWER:** +5V supply.
V_{SS}		**GROUND:** Ground.
CLK	I	**CLOCK INPUT:** Clock Input controls the internal operations of the 8237A and its rate of data transfers. The input may be driven at up to 3 MHz for the standard 8237A and up to 5 MHz for the 8237A-5.
$\overline{CS}$	I	**CHIP SELECT:** Chip Select is an active low input used to select the 8237A as an I/O device during the Idle cycle. This allows CPU communication on the data bus.
RESET	I	**RESET:** Reset is an active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip/flop and sets the Mask register. Following a Reset the device is in the Idle cycle.
READY	I	**READY:** Ready is an input used to extend the memory read and write pulses from the 8237A to accommodate slow memories or I/O peripheral devices. Ready must not make transitions during its specified setup/hold time.
HLDA	I	**HOLD ACKNOWLEDGE:** The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system busses.
DREQ0–DREQ3	I	**DMA REQUEST:** The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active.
DB0–DB7	I/O	**DATA BUS:** The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the 8237A control registers. During DMA cycles the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the 8237A on the data bus during the read-from-memory transfer. In the write-to-memory transfer, the data bus outputs place the data into the new memory location.
$\overline{IOR}$	I/O	**I/O READ:** I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the 8237A to access data from a peripheral during a DMA Write transfer.
$\overline{IOW}$	I/O	**I/O WRITE:** I/O Write is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to load information into the 8237A. In the Active cycle, it is an output control signal used by the 8237A to load data to the peripheral during a DMA Read transfer.

Table 1. Pin Description (Continued)

Symbol	Type	Name and Function
$\overline{EOP}$	I/O	**END OF PROCESS:** End of Process is an active low bidirectional signal. Information concerning the completion of DMA services is available at the bidirectional $\overline{EOP}$ pin. The 8237A allows an external signal to terminate an active DMA service. This is accomplished by pulling the $\overline{EOP}$ input low with an external $\overline{EOP}$ signal. The 8237A also generates a pulse when the terminal count (TC) for any channel is reached. This generates an $\overline{EOP}$ signal which is output through the $\overline{EOP}$ line. The reception of $\overline{EOP}$, either internal or external, will cause the 8237A to terminate the service, reset the request, and, if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by $\overline{EOP}$ unless the channel is programmed for Autoinitialize. In that case, the mask bit remains unchanged. During memory-to-memory transfers, $\overline{EOP}$ will be output when the TC for channel 1 occurs. $\overline{EOP}$ should be tied high with a pull-up resistor if it is not used to prevent erroneous end of process inputs.
A0–A3	I/O	**ADDRESS:** The four least significant address lines are bidirectional three-state signals. In the Idle cycle they are inputs and are used by the CPU to address the register to be loaded or read. In the Active cycle they are outputs and provide the lower 4 bits of the output address.
A4–A7	O	**ADDRESS:** The four most significant address lines are three-state outputs and provide 4 bits of address. These lines are enabled only during the DMA service.
HRQ	O	**HOLD REQUEST:** This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes 8237A to issue the HRQ.
DACK0–DACK3	O	**DMA ACKNOWLEDGE:** DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The sense of these lines is programmable. Reset initializes them to active low.
AEN	O	**ADDRESS ENABLE:** Address Enable enables the 8-bit latch containing the upper 8 address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers. AEN is active HIGH.
ADSTB	O	**ADDRESS STROBE:** The active high, Address Strobe is used to strobe the upper address byte into an external latch.
$\overline{MEMR}$	O	**MEMORY READ:** The Memory Read signal is an active low three-state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.
$\overline{MEMW}$	O	**MEMORY WRITE:** The Memory Write is an active low three-state output used to write data to the selected memory location during a DMA Write or a memory-to-memory transfer.
PIN5	I	**PIN5:** This pin should always be at a logic HIGH level. An internal pull-up resistor will establish a logic high when the pin is left floating. It is recommended however, that PIN5 be connected to V_{CC}.

FUNCTIONAL DESCRIPTION

The 8237A block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The 8237A contains 344 bits of internal memory in the form of registers. Figure 3 lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Figure 3. 8237A Internal Registers

The 8237A contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the 8237A. The Program Command Control block decodes the various commands given to the 8237A by the microprocessor prior to servicing a DMA Request. It also decodes the Mode Control word used to select the type of DMA during the servicing. The Priority Encoder block resolves priority contention between DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input. In 8237A systems, this input will usually be the φ2 TTL clock from an 8224 or CLK from an 8085AH or 8284A. 33% duty cycle clock generators, however, may not meet the clock high time requirement of the 8237A of the same frequency. For example, 82C84A-5 CLK output violates the clock high time requirement of 8237A-5. In this case 82C84A CLK can simply be inverted to meet 8237A-5 clock high and low time requirements. For 8085AH-2 systems above 3.9 MHz, the 8085 CLK(OUT) does not satisfy 8237A-5 clock LOW and HIGH time requirements. In this case, an external clock should be used to drive the 8237A-5.

DMA OPERATION

The 8237A is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The 8237A can assume seven separate states, each composed of one full clock period. State I (SI) is the inactive state. It is entered when the 8237A has no valid DMA requests pending. While in SI, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State S0 (S0) is the first state of a DMA service. The 8237A has requested a hold but the processor has not yet returned an acknowledge. The 8237A may still be programmed until it receives HLDA from the CPU. An acknowledge from the CPU will signal that DMA transfers may begin. S1, S2, S3 and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted between S2 or S3 and S4 by the use of the Ready line on the 8237A. Note that the data is transferred directly from the I/O device to memory (or vice versa) with $\overline{\text{IOR}}$ and $\overline{\text{MEMW}}$ (or $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$) being active at the same time. The data is not read into or driven out of the 8237A in I/O-to-memory or memory-to-I/O DMA transfers.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for a single transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23, S24) for the write-to-memory half of the transfer.

IDLE CYCLE

When no channel is requesting service, the 8237A will enter the Idle cycle and perform "SI" states. In this cycle the 8237A will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample $\overline{\text{CS}}$, looking for an attempt by the microprocessor to write or read the internal registers of the 8237A. When $\overline{\text{CS}}$ is low and HLDA is low, the 8237A enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0–A3 are inputs to the device and select which registers will be read or written. The $\overline{\text{IOR}}$ and $\overline{\text{IOW}}$ lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip-flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip-flop is reset by Master Clear or Reset. A separate software command can also reset this flip-flop.

Special software commands can be executed by the 8237A in the Program Condition. These commands are decoded as sets of addresses with the $\overline{\text{CS}}$ and $\overline{\text{IOW}}$. The commands do not make use of the data bus. Instructions include Clear First/Last Flip-Flop and Master Clear.

ACTIVE CYCLE

When the 8237A is in the Idle cycle and a non-masked channel requests a DMA service, the device will output an HRQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place, in one of four modes:

Single Transfer Mode—In Single Transfer mode the device is programmed to make one transfer only. The word count will be decremented and the address decremented or incremented following each transfer. When the word count "rolls over" from zero to FFFFH, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the single transfer, HRQ will go inactive and release the bus to the system. It will again go active and, upon receipt of a new HLDA, another single transfer will be performed. In 8080A, 8085AH, 8088, or 8086 system, this will ensure one full machine cycle execution between DMA transfers. Details of timing between the 8237A and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode—In Block Transfer mode the device is activated by DREQ to continue making transfers during the service until a TC, caused by word count going to FFFFH, or an external End of Process ($\overline{\text{EOP}}$) is encountered. DREQ need only be held active until DACK becomes active. Again, an Autoinitialization will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode—In Demand Transfer mode the device is programmed to continue making transfers until a TC or external $\overline{\text{EOP}}$ is encountered or until DREQ goes inactive. Thus transfers may continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is re-established by means of a DREQ. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count are stored in the 8237A Current Address and Current Word Count registers. Only an $\overline{\text{EOP}}$ can cause an Autoinitialize at the end of the service. $\overline{\text{EOP}}$ is generated either by TC or by an external signal. DREQ has to be low before S4 to prevent another Transfer.

Cascade Mode—This mode is used to cascade more than one 8237A together for simple system expansion. The HRQ and HLDA signals from the additional 8237A are connected to the DREQ and DACK signals of a channel of the initial 8237A. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel of the initial 8237A is used only for prioritizing the additional device, it does not output any address or control

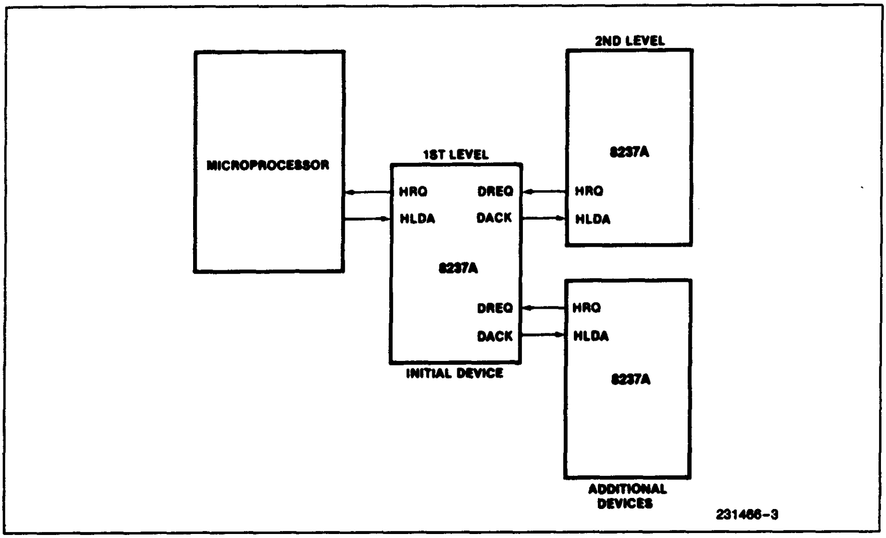


Figure 4. Cascaded 8237As

signals of its own. These could conflict with the outputs of the active channel in the added device. The 8237A will respond to DREQ and DACK but all other outputs except HRQ will be disabled. The ready input is ignored.

Figure 4 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More 8237As could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level device, forming a third level.

TRANSFER TYPES

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from an I/O device to the memory by activating $\overline{\text{MEMW}}$ and $\overline{\text{IOR}}$. Read transfers move data from memory to an I/O device by activating $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$. Verify transfers are pseudo transfers. The 8237A operates as in Read or Write transfers generating addresses, and responding to EOP, etc. However, the memory and I/O control lines all remain inactive. The ready input is ignored in verify mode.

Memory-to-Memory—To perform block moves of data from one memory address space to another with a minimum of program effort and time, the 8237A includes a memory-to-memory transfer feature. Programming a bit in the Command register selects channels 0 and 1 to operate as memory-to-memory transfer channels. The transfer is initiated by setting the software DREQ for channel 0. The 8237A requests a DMA service in the normal manner. After HLDA is true, the device, using four state transfers in Block Transfer mode, reads data from the memory. The channel 0 Current Address register is the source for the address used and is decremented or incremented in the normal manner. The data byte read from the memory is stored in the 8237A internal Temporary register. Channel 1 then performs a four-state transfer of the data from the Temporary register to memory using the address in its Current Address register and incrementing or decrementing it in the normal manner. The channel 1 current Word Count is decremented. When the word count of channel 1 goes to FFFFH, a TC is generated causing an $\overline{\text{EOP}}$ output terminating the service.

Channel 0 may be programmed to retain the same address for all transfers. This allows a single word to be written to a block of memory.

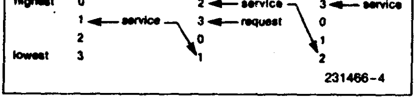
The 8237A will respond to external $\overline{\text{EOP}}$ signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers is found in Figure 12. Memory-to-memory operations can be detected as an active AEN with no DACK outputs.

Autoinitialize—By programming a bit in the Mode register, a channel may be set up as an Autoinitialize channel. During Autoinitialize initialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word count registers of that channel following $\overline{\text{EOP}}$. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not altered when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to perform another DMA service, without CPU intervention, as soon as a valid DREQ is detected. In order to Autoinitialize both channels in a memory-to-memory transfer, both word counts should be programmed identically. If interrupted externally, $\overline{\text{EOP}}$ pulses should be applied in both bus cycles.

Priority—The 8237A has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is completed.

After completion of a service, HRQ will go inactive and the 8237A will wait for HLDA to go low before activating HRQ to service another channel.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly.



With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.

Compressed Timing—In order to achieve even greater throughput where system characteristics permit, the 8237A can compress the transfer time to two clock cycles. From Figure 11 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3, the read pulse width is made equal to the write pulse width and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur when A8–A15 need updating (see Address Generation). Timing for compressed transfers is found in Figure 14.

Address Generation—In order to reduce pin count, the 8237A multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a three-state enable. The lower order address bits are output by the 8237A directly. Lines A0–A7 should be connected to the address bus. Figure 11 shows the time relationships between CLK, AEN, ADSTB, DB0–DB7 and A0–A7.

During Block and Demand Transfer mode services, which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the 8237A executes S1 states only when updating of A8–A15 in the latch is necessary. This means for long services, S1 states and Address Strobes may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

REGISTER DESCRIPTION

Current Address Register—Each channel has a 16-bit Current Address register. This register holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize takes place only after an $\overline{\text{EOP}}$.

Current Word Register—Each channel has a 16-bit Current Word Count register. This register determines the number of transfers to be performed. The actual number of transfers will be one more than the number programmed in the Current Word Count register (i.e., programming a count of 100 will result in 101 transfers). The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes from zero to FFFFH, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service it may also be reinitialized by an Autoinitialization back to its original value. Autoinitialize can occur only when an $\overline{\text{EOP}}$ occurs. If it is not Autoinitialized, this register will have a count of FFFFH after TC.

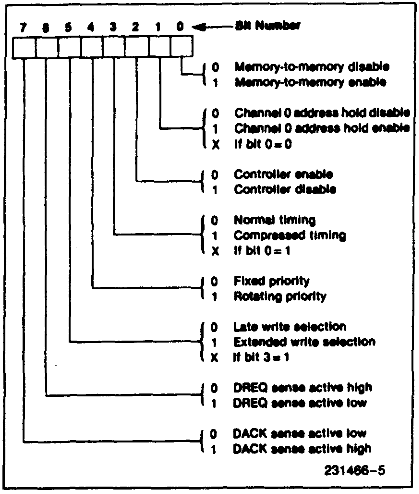
Base Address and Base Word Count Registers—Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original value of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes in the Program Condition by the microprocessor. These registers cannot be read by the microprocessor.

Command Register—This 8-bit register controls the operation of the 8237A. It is programmed by the microprocessor in the Program Condition and is cleared by Reset or a Master Clear instruction. The following table lists the function of the command bits. See Figure 6 for address coding.

Mode Register—Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written.

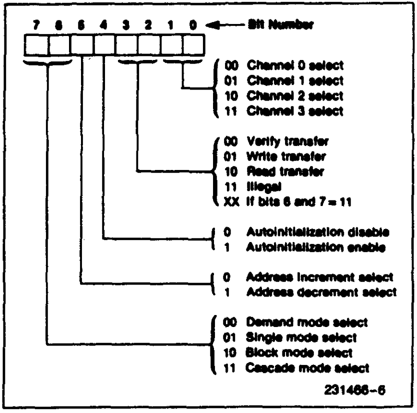
Request Register—The 8237A can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are non-maskable and subject to prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external $\overline{\text{EOP}}$. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 5 for register address coding. In order to make a software request, the channel must be in Block Mode.

Command Register



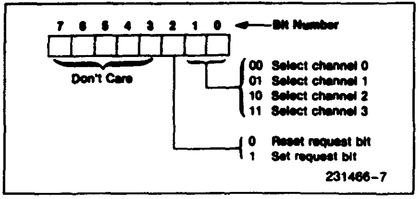
231466-5

Mode Register



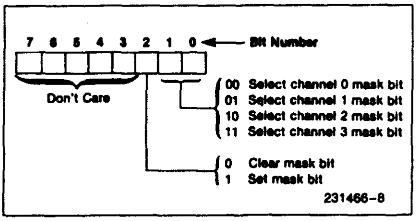
231466-6

Request Register



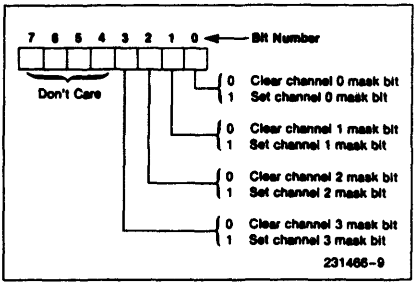
231466-7

Mask Register—Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an $\overline{EOP}$ if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 5 for instruction addressing.


7 6 5 4 3 2 1 0 ◄— Bit Number
Don't Care
00 Select channel 0 mask bit
01 Select channel 1 mask bit
10 Select channel 2 mask bit
11 Select channel 3 mask bit
0 Clear mask bit
1 Set mask bit

231466-8

All four bits of the Mask register may also be written with a single command.

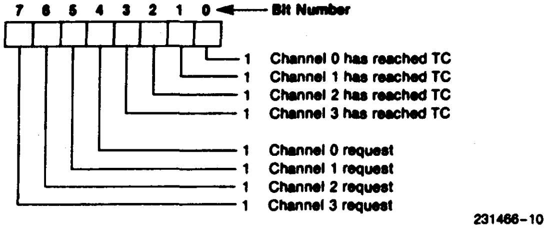

7 6 5 4 3 2 1 0 ◄— Bit Number
Don't Care
0 Clear channel 0 mask bit
1 Set channel 0 mask bit
0 Clear channel 1 mask bit
1 Set channel 1 mask bit
0 Clear channel 2 mask bit
1 Set channel 2 mask bit
0 Clear channel 3 mask bit
1 Set channel 3 mask bit

231466-9

Register	Operation	Signals						
		$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0
Command	Write	0	1	0	1	0	0	0
Mode	Write	0	1	0	1	0	1	1
Request	Write	0	1	0	1	0	0	1
Mask	Set/Reset	0	1	0	1	0	1	0
Mask	Write	0	1	0	1	1	1	1
Temporary	Read	0	0	1	1	1	0	1
Status	Read	0	0	1	1	0	0	0

Figure 5. Definition of Register Codes

Status Register—The Status register is available to be read out of the 8237A by the microprocessor. It contains information about the status of the devices at this point. This information includes which channels have reached a terminal count and which chan-


7 6 5 4 3 2 1 0 ← Bit Number
1 Channel 0 has reached TC
1 Channel 1 has reached TC
1 Channel 2 has reached TC
1 Channel 3 has reached TC
1 Channel 0 request
1 Channel 1 request
1 Channel 2 request
1 Channel 3 request
231466-10

nels have pending DMA requests. Bits 0–3 are set every time a TC is reached by that channel or an external $\overline{\text{EOP}}$ is applied. These bits are cleared upon Reset and on each Status Read. Bits 4–7 are set whenever their corresponding channel is requesting service.

Temporary Register—The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands—These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The three software commands are:

Clear First/Last Flip-Flop: This command must be executed prior to writing or reading new address or word count information to the 8237A. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The 8237A will enter the Idle cycle.

Clear Mask Register: This command clears the mask bits of all four channels, enabling them to accept DMA requests.

Figure 6 lists the address codes for the software commands.

Signals						Operation
A3	A2	A1	A0	$\overline{\text{IOR}}$	$\overline{\text{IOW}}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 6. Software Command Codes

Channel	Register	Operation	Signals							Internal	Data Bus
			$\overline{\text{CS}}$	$\overline{\text{IOR}}$	$\overline{\text{IOW}}$	A3	A2	A1	A0	Flip-Flop	DB0–DB7
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0–A7
			0	1	0	0	0	0	0	1	A8–A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0–A7
			0	0	1	0	0	0	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	0	0	1	0	W0–W7
			0	1	0	0	0	0	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0–W7
			0	0	1	0	0	0	1	1	W8–W15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0–A7
			0	1	0	0	0	1	0	1	A8–A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0–A7
			0	0	1	0	0	1	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	0	1	1	0	W0–W7
			0	1	0	0	0	1	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0–W7
			0	0	1	0	0	1	1	1	W8–W15
2	Base and Current Address	Write	0	1	0	0	1	0	0	0	A0–A7
			0	1	0	0	1	0	0	1	A8–A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0–A7
			0	0	1	0	1	0	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	1	0	1	0	W0–W7
			0	1	0	0	1	0	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0–W7
			0	0	1	0	1	0	1	1	W8–W15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0–A7
			0	1	0	0	1	1	0	1	A8–A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0–A7
			0	0	1	0	1	1	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	1	1	1	0	W0–W7
			0	1	0	0	1	1	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0–W7
			0	0	1	0	1	1	1	1	W8–W15

Figure 7. Word Count and Address Register Command Codes

PROGRAMMING

The 8237A will accept programming from the host processor any time that HLDA is inactive; this is true even if HRQ is active. The responsibility of the host is to assure that programming and HLDA are mutually exclusive. Note that a problem can occur if a DMA request occurs, on an unmasked channel while the 8237A is being programmed. For instance, the CPU may be starting to reprogram the two byte Address register of channel 1 when channel 1 receives a DMA request. If the 8237A is enabled (bit 2 in the command register is 0) and channel 1 is unmasked, a DMA service will occur after only one byte of the Address register has been reprogrammed. This can be avoided by disabling the controller (setting bit 2 in the command register) or masking the channel before programming any other registers. Once the programming is complete, the controller can be enabled/unmasked.

After power-up it is suggested that all internal locations, especially the Mode registers, be loaded with some valid value. This should be done even if some channels are unused. An invalid mode may force all control signals to go active at the same time.

APPLICATION INFORMATION (Note 1)

Figure 8 shows a convenient method for configuring a DMA system with the 8237A controller and an 8080A/8085AH microprocessor system. The multimode DMA controller issues a HRQ to the processor whenever there is at least one valid DMA request from a peripheral device. When the processor replies with a HLDA signal, the 8237A takes control of the address bus, the data bus and the control bus. The address for the first transfer operation comes out in two bytes—the least significant 8 bits on the eight address outputs and the most significant 8 bits on the data bus. The contents of the data bus are then latched into an 8-bit latch to complete the full 16 bits of the address bus. The 8282 is a high speed, 8-bit, three-state latch in a 20-pin package. After the initial transfer takes place, the latch is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one 8237A is used.

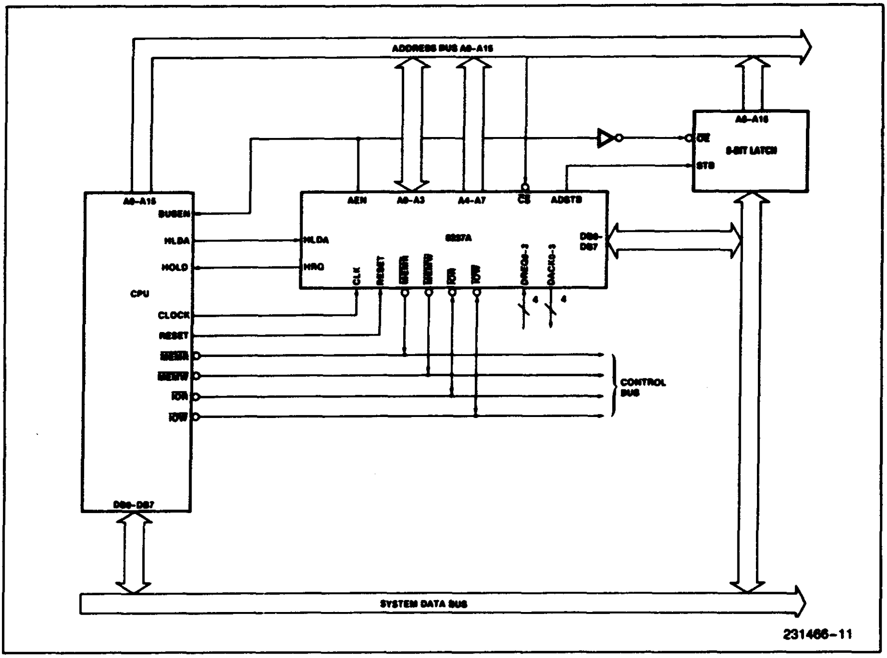


Figure 8. 8237A System Interface

NOTE:
1. See Application Note AP-67 for 8086 design information.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias ......0°C to 70°C
Case Temperature ....0°C to +75°C
Storage Temperature .... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground.............. −0.5V to +7V
Power Dissipation.......................1.5 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

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5.0V ±5%, GND = 0V

Symbol	Parameter	Min	Typ (Note 1)	Max	Unit	Test Conditions
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −200 μA
		3.3			V	I_{OH} = −100 μA (HRQ Only)
V_{OL}	Output LOW Voltage			0.40	V	I_{OL} = 3.2 mA
V_{IH}	Input HIGH Voltage	2.0		V_{CC} + 0.5	V	
V_{IL}	Input LOW Voltage	−0.5		0.8	V	
I_{LI}	Input Load Current			±10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current			±10	μA	0.45V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current		110	130	mA	T_A = +25°C
			130	150	mA	T_A = 0°C
C_O	Output Capacitance		4	8	pF	fc = 1.0 MHz, Inputs = 0V
C_I	Input Capacitance		8	15	pF	
C_{IO}	I/O Capacitance		10	18	pF	

NOTE:
1. Typical values are for T_A = 25°C, nominal supply voltage and nominal processing parameters.

A.C. CHARACTERISTICS—DMA (MASTER) MODE

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5V ±5%, GND = 0V

Symbol	Parameter	8237A		8237A-4		8237A-5		Unit
		Min	Max	Min	Max	Min	Max	
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		300		225		200	ns
TAET	AEN LOW from CLK HIGH (SI) Delay Time		200		150		130	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		150		120		90	ns
TAFC	$\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ Float from CLK HIGH		150		120		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		250		190		170	ns
TAHR	ADR from $\overline{\text{READ}}$ HIGH Hold Time	TCY-100		TCY-100		TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	40		40		30		ns
TAHW	ADR from $\overline{\text{WRITE}}$ HIGH Hold Time	TCY-50		TCY-50		TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time (Note 1)		250		220		170	ns
	$\overline{\text{EOP}}$ HIGH from CLK HIGH Delay Time (Note 2)		250		190		170	ns
	$\overline{\text{EOP}}$ LOW from CLK HIGH Delay Time		250		190		170	ns
TASM	ADR Stable from CLK HIGH		250		190		170	ns
TASS	DB to ADSTB LOW Setup Time	100		100		100		ns
TCH	Clock High Time (Transitions ≤ 10 ns)	120		100		80		ns
TCL	Clock LOW Time (Transitions ≤ 10 ns)	150		110		68		ns
TCY	CLK Cycle Time	320		250		200		ns
TDCL	CLK HIGH to $\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ LOW Delay (Note 3)		270		200		190	ns
TDCTR	$\overline{\text{READ}}$ HIGH from CLK HIGH (S4) Delay Time (Note 3)		270		210		190	ns
TDCTW	$\overline{\text{WRITE}}$ HIGH from CLK HIGH (S4) Delay Time (Note 3)		200		150		130	ns
TDQ1	HRQ Valid from CLK HIGH Delay Time (Note 4)		160		120		120	ns
TDQ2			250		190		120	ns
TEPS	$\overline{\text{EOP}}$ LOW from CLK LOW Setup Time	60		45		40		ns
TEPW	$\overline{\text{EOP}}$ Pulse Width	300		225		220		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		250		190		170	ns
TFAC	$\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ Active from CLK HIGH		200		150		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		300		225		200	ns
THS	HLDA Valid to CLK HIGH Setup Time	100		75		75		ns
TIDH	Input Data from $\overline{\text{MEMR}}$ HIGH Hold Time	0		0		0		ns
TIDS	Input Data to $\overline{\text{MEMR}}$ HIGH Setup Time	250		190		170		ns
TODH	Output Data from $\overline{\text{MEMW}}$ HIGH Hold Time	20		20		10		ns
TODV	Output Data Valid to $\overline{\text{MEMW}}$ HIGH	200		125		125		ns
TQS	DREQ to CLK LOW (SI, S4) Setup Time (Note 1)	0		0		0		ns
TRH	CLK to READY LOW Hold Time	20		20		20		ns
TRS	READY to CLK LOW Setup Time	100		60		60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		200		150		130	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		140		110		90	ns

A.C. CHARACTERISTICS—PERIPHERAL (SLAVE) MODE

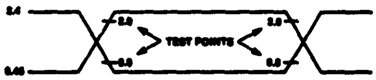
T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5V ±5%, GND = 0V

Symbol	Parameter	8237A		8237A-4		8237A-5		Unit
		Min	Max	Min	Max	Min	Max	
TAR	ADR Valid or $\overline{\text{CS}}$ LOW to $\overline{\text{READ}}$ LOW	50		50		50		ns
TAW	ADR Valid to $\overline{\text{WRITE}}$ HIGH Setup Time	200		150		130		ns
TCW	CS LOW to $\overline{\text{WRITE}}$ HIGH Setup Time	200		150		130		ns
TDW	Data Valid to $\overline{\text{WRITE}}$ HIGH Setup Time	200		150		130		ns
TRA	ADR or CS Hold from $\overline{\text{READ}}$ HIGH	0		0		0		ns
TRDE	Data Access from $\overline{\text{READ}}$ LOW (Note 5)		200		200		140	ns
TRDF	DB Float Delay from $\overline{\text{READ}}$ HIGH	20	100	20	100	0	70	ns
TRSTD	Power Supply HIGH to RESET LOW Setup Time	500		500		500		ns
TRSTS	RESET to First $\overline{\text{IOWR}}$	2TCY		2TCY		2TCY		ns
TRSTW	RESET Pulse Width	300		300		300		ns
TRW	$\overline{\text{READ}}$ Width	300		250		200		ns
TWA	ADR from $\overline{\text{WRITE}}$ HIGH Hold Time	20		20		20		ns
TWC	CS HIGH from $\overline{\text{WRITE}}$ HIGH Hold Time	20		20		20		ns
TWD	Data from $\overline{\text{WRITE}}$ HIGH Hold Time	30		30		30		ns
TWWS	Write Width	200		200		160		ns
TWR	End of Write to End of Read in DMA Transfer	0		0		0		ns

NOTES:

1. DREQ and DACK signals may be active high or active low. Timing diagrams assume the active high mode.
2. $\overline{\text{EOP}}$ is an open collector output. This parameter assumes the presence of a 2.2K pullup to V_{CC}.
3. The net $\overline{\text{IOW}}$ or $\overline{\text{MEMW}}$ Pulse width for normal write will be TCY − 100 ns and for extended write will be 2TCY − 100 ns. The net $\overline{\text{IOR}}$ or $\overline{\text{MEMR}}$ pulse width for normal read will be 2TCY − 50 ns and for compressed read will be TCY − 50 ns.
4. TDQ is specified for two different output HIGH levels. TDQ1 is measured at 2.0V. TDQ2 is measured at 3.3V. The value for TDQ2 assumes an external 3.3 KΩ pull-up resistor connected from HRQ to V_{CC}.
5. Output Loading on the Data Bus is 1 TTL Gate plus 100 pF capacitance.

A.C. TESTING INPUT/OUTPUT WAVEFORM

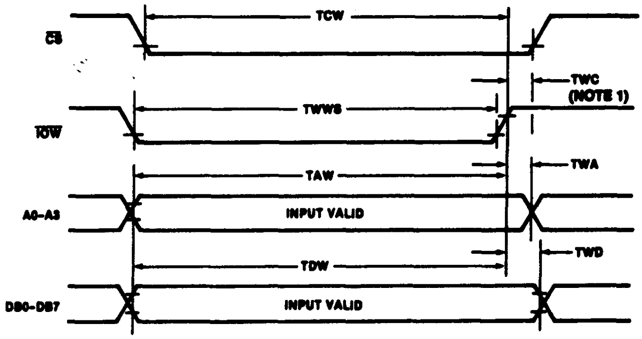


231466–12

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0." Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0." Input timing parameters assume transition times of 20 ns or less. Waveform measurement points for both input and output signals are 2.0V for HIGH and 0.8V for LOW, unless otherwise noted.

WAVEFORMS

SLAVE MODE WRITE TIMING

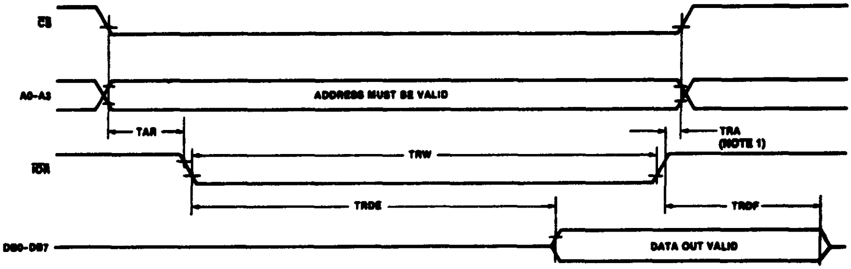


231466-13

NOTE:
1. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600 ns for the 8237A, at least 500 ns for the 8237A-4, and at least 400 ns for the 8237A-5 as recovery time between active read or write pulses. The same recovery time is needed between an active read or write pulse followed by a DMA transfer.

Figure 9. Slave Mode Write

SLAVE MODE READ TIMING



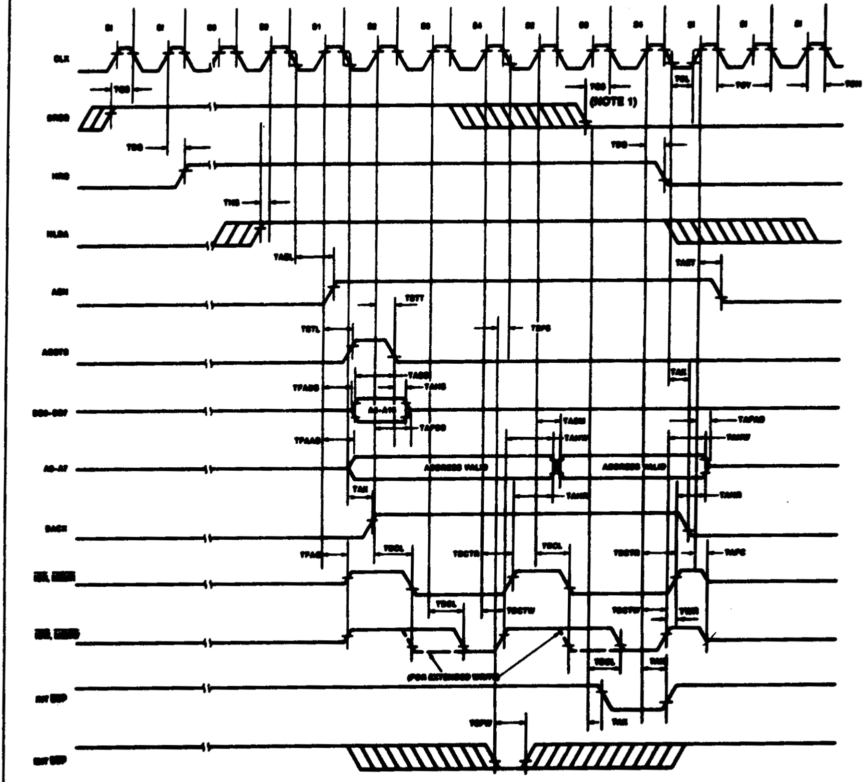
231466-14

NOTE:
1. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600 ns for the 8237A, at least 500 ns for the 8237A-4, and at least 400 ns for the 8237A-5 as recovery time between active read or write pulses. The same recovery time is needed between an active read or write pulse followed by a DMA transfer.

Figure 10. Slave Mode Read

WAVEFORMS (Continued)

DMA TRANSFER TIMING



231466-15

NOTE:
1. DREQ should be held active until DACK is returned.

Figure 11. DMA Transfer

WAVEFORMS (Continued)

MEMORY-TO-MEMORY TRANSFER TIMING

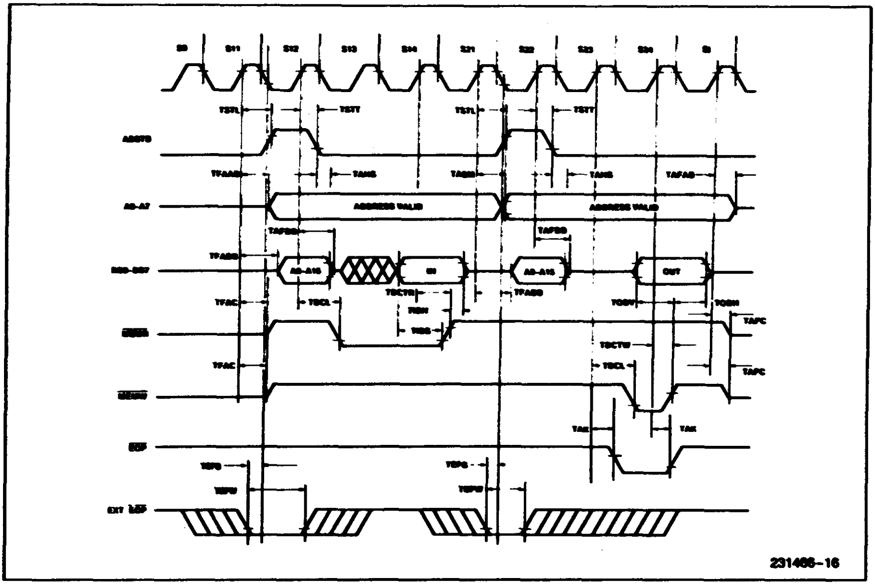


Figure 12. Memory-to-Memory Transfer

READY TIMING

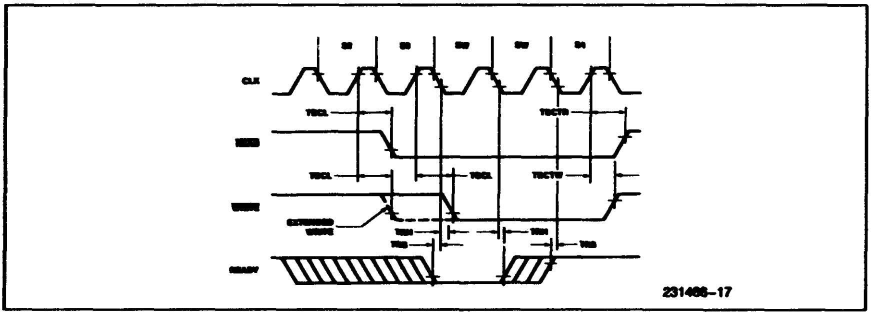


Figure 13. Ready

WAVEFORMS (Continued)

COMPRESSED TRANSFER TIMING

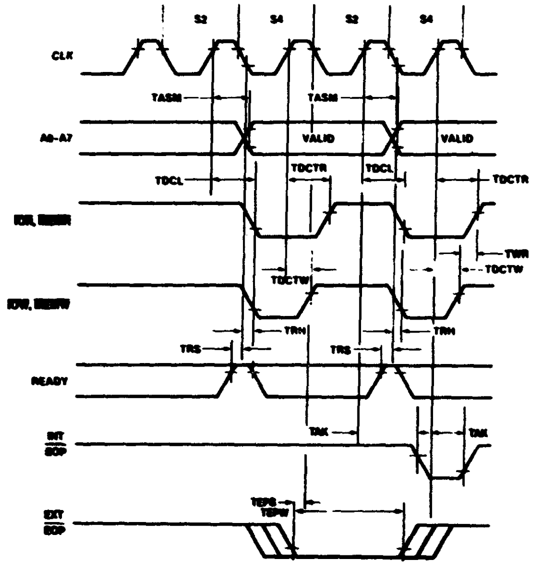


Figure 14. Compressed Transfer

RESET TIMING

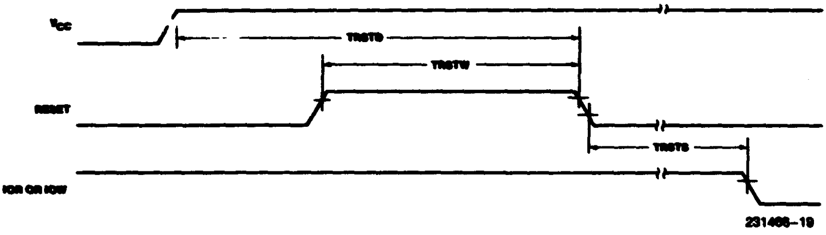


Figure 15. Reset

DESIGN CONSIDERATIONS

1. ***Cascading from channel zero.*** When using multiple 8237s, always start cascading with channel zero. Channel zero of the 8237 will operate incorrectly if one or more of channels 1, 2, or 3 are used in the cascade mode while channel zero is used in a mode other than cascade.
2. ***Do not treat the DREQ signal as an asynchronous input while the channel is in the "demand" or "cascade" modes.*** If DREQ becomes inactive at any time during state S4, an illegal state may occur causing the 8237 to operate improperly.
3. ***HRQ must remain active until HLDA becomes active.*** If HRQ goes inactive before HLDA is received the 8237 can enter an illegal state causing it to operate improperly.
4. ***Make sure the MEMR# line has 50 pF loading capacitance on it.*** When doing memory to memory transfers, the 8237 requires at least 50 pF loading capacitance on the MEMR# signal for proper operation. In most cases board capacitance is sufficient.
5. ***Treat the READY input as a synchronous input.*** If a transition occurs during the setup/hold window, erratic operation may result.

DATA SHEET REVISION REVIEW

The following list represents key differences between this and the -002 data sheet. Please review this summary carefully.

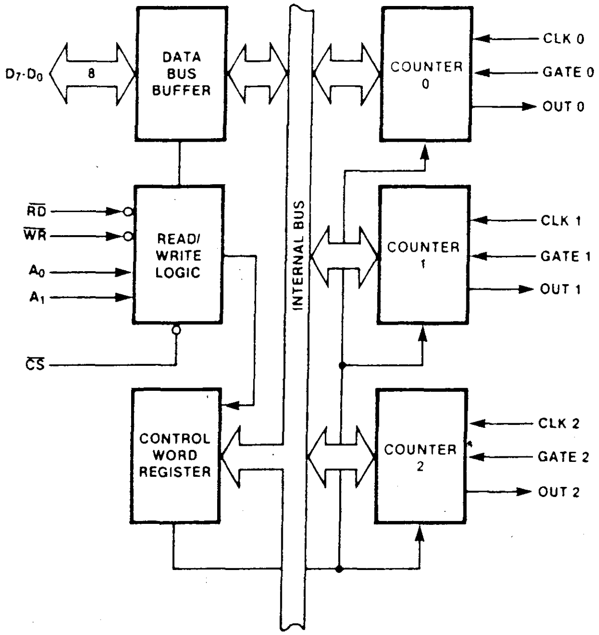
1. Major cleanup on the "NOTE" sections of this data sheet.
   a. Pin 5 no longer references a note. It is now included in the pin description area under the name "PIN5".
   b. The note placed in the "typical" section of the D.C. Characteristics table is now referenced to a note section included with that table.
   c. Notes in the A.C. Characteristics table have been renumbered and are included in a notes section for the A.C. Characteristics.
   d. The note that was previously referenced in the A.C. TESTING INPUT/OUTPUT WAVEFORM diagram has been replaced with the actual note.
   e. The note that was previously referenced in the SLAVE MODE WRITE TIMING diagram has been included in a "NOTE" section with the diagram.
   f. The note that was previously referenced in the SLAVE MODE READ TIMING diagram has been included in a "NOTE" section with the diagram.
   g. The note that was previously referenced in the DMA TRANSFER TIMING diagram has been included in a "NOTE" section with the diagram.
2. A "Design Considerations" section was added to alert designers to certain design aspects of the 8237.
3. The timing parameters TAR for the 8237A-4 and 8237A-5 have been changed from 50 ns to 0 ns.

8254
PROGRAMMABLE INTERVAL TIMER

- Compatible with All Intel and Most Other Microprocessors
- Handles Inputs from DC to 10 MHz
  - — 5 MHz 8254-5
  - — 8 MHz 8254
  - — 10 MHz 8254-2
- Status Read-Back Command
- Six Programmable Counter Modes
- Three Independent 16-Bit Counters
- Binary or BCD Counting
- Single +5V Supply
- Available in EXPRESS
  - —Standard Temperature Range

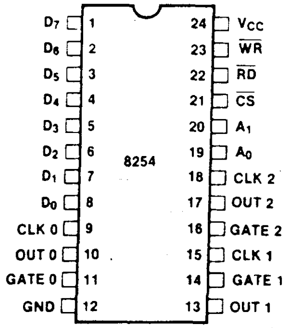
The Intel® 8254 is a counter/timer device designed to solve the common timing control problems in microcomputer system design. It provides three independent 16-bit counters, each capable of handiing cicck inputs up to 10 MHz. All modes are software programmable. The 8254 is a superset of the 8253.

The 8254 uses HMOS technology and comes in a 24-pin plastic or CERDIP package.



231164-1

Figure 1. 8254 Block Diagram



231164-2

Figure 2. Pin Configuration

Table 1. Pin Description

<table>
<tr><th>Symbol</th><th>Pin No.</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>D_7–D_0</td><td>1–8</td><td>I/O</td><td><b>DATA:</b> Bi-directional three state data bus lines, connected to system data bus.</td></tr>
<tr><td>CLK 0</td><td>9</td><td>I</td><td><b>CLOCK 0:</b> Clock input of Counter 0.</td></tr>
<tr><td>OUT 0</td><td>10</td><td>O</td><td><b>OUTPUT 0:</b> Output of Counter 0.</td></tr>
<tr><td>GATE 0</td><td>11</td><td>I</td><td><b>GATE 0:</b> Gate input of Counter 0.</td></tr>
<tr><td>GND</td><td>12</td><td></td><td><b>GROUND:</b> Power supply connection.</td></tr>
<tr><td>V_{CC}</td><td>24</td><td></td><td><b>POWER:</b> +5V power supply connection.</td></tr>
<tr><td>$\overline{WR}$</td><td>23</td><td>I</td><td><b>WRITE CONTROL:</b> This input is low during CPU write operations.</td></tr>
<tr><td>$\overline{RD}$</td><td>22</td><td>I</td><td><b>READ CONTROL:</b> This input is low during CPU read operations.</td></tr>
<tr><td>$\overline{CS}$</td><td>21</td><td>I</td><td><b>CHIP SELECT:</b> A low on this input enables the 8254 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.</td></tr>
<tr><td>A_1, A_0</td><td>20–19</td><td>I</td><td><b>ADDRESS:</b> Used to select one of the three Counters or the Control Word Register for read or write operations. Normally connected to the system address bus.
<table>
<tr><th>A_1</th><th>A_0</th><th>Selects</th></tr>
<tr><td>0</td><td>0</td><td>Counter 0</td></tr>
<tr><td>0</td><td>1</td><td>Counter 1</td></tr>
<tr><td>1</td><td>0</td><td>Counter 2</td></tr>
<tr><td>1</td><td>1</td><td>Control Word Register</td></tr>
</table></td></tr>
<tr><td>CLK 2</td><td>18</td><td>I</td><td><b>CLOCK 2:</b> Clock input of Counter 2.</td></tr>
<tr><td>OUT 2</td><td>17</td><td>O</td><td><b>OUT 2:</b> Output of Counter 2.</td></tr>
<tr><td>GATE 2</td><td>16</td><td>I</td><td><b>GATE 2:</b> Gate input of Counter 2.</td></tr>
<tr><td>CLK 1</td><td>15</td><td>I</td><td><b>CLOCK 1:</b> Clock input of Counter 1.</td></tr>
<tr><td>GATE 1</td><td>14</td><td>I</td><td><b>GATE 1:</b> Gate input of Counter 1.</td></tr>
<tr><td>OUT 1</td><td>13</td><td>O</td><td><b>OUT 1:</b> Output of Counter 1.</td></tr>
</table>

FUNCTIONAL DESCRIPTION

General

The 8254 is a programmable interval timer/counter designed for use with Intel microcomputer systems. It is a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8254 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 8254 to match his requirements and programs one of the counters for the desired delay. After the desired delay, the 8254 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other counter/timer functions common to microcomputers which can be implemented with the 8254 are:

- Real time clock
- Event-counter
- Digital one-shot
- Programmable rate generator
- Square wave generator
- Binary rate multiplier
- Complex waveform generator
- Complex motor controller

Block Diagram

DATA BUS BUFFER

This 3-state, bi-directional, 8-bit buffer is used to interface the 8254 to the system bus (see Figure 3).

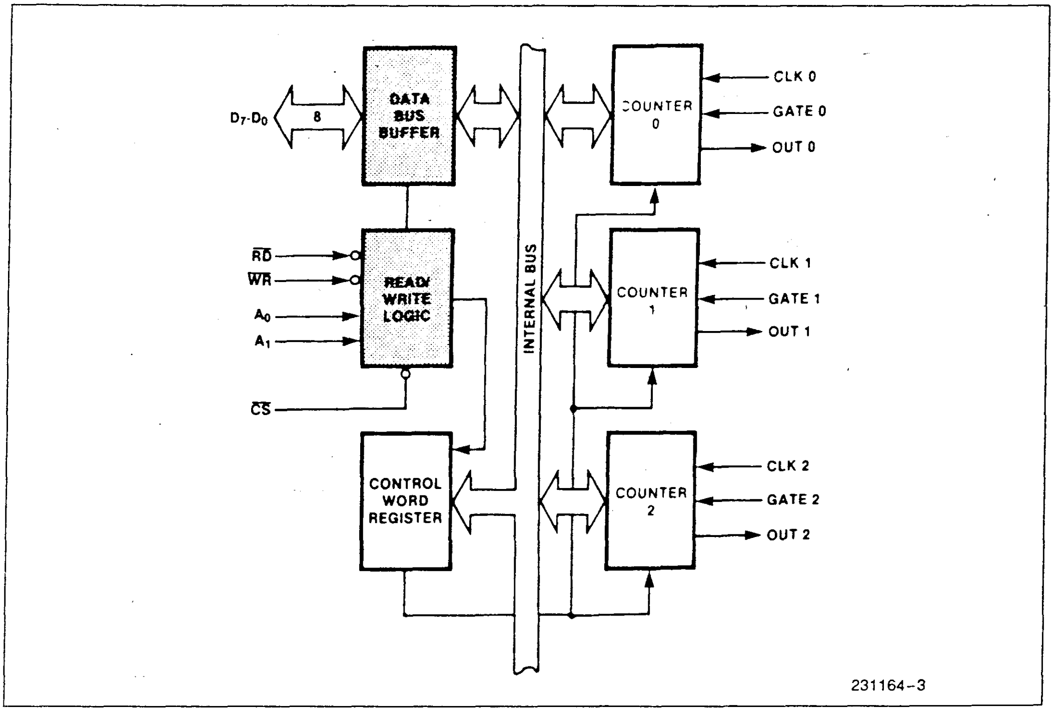


Figure 3. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

READ/WRITE LOGIC

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 8254. A_1 and A_0 select one of the three counters or the Control Word Register to be read from/written into. A "low" on the $\overline{RD}$ input tells the 8254 that the CPU is reading one of the counters. A "low" on the $\overline{WR}$ input tells the 8254 that the CPU is writing either a Control Word or an initial count. Both $\overline{RD}$ and $\overline{WR}$ are qualified by $\overline{CS}$; $\overline{RD}$ and $\overline{WR}$ are ignored unless the 8254 has been selected by holding $\overline{CS}$ low.

CONTROL WORD REGISTER

The Control Word Register (see Figure 4) is selected by the Read/Write Logic when $A_1,A_0 = 11$. If the CPU then does a write operation to the 8254, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

COUNTER 0, COUNTER 1, COUNTER 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a single counter is shown in Figure 5.

The Counters are fully independent. Each Counter may operate in a different Mode.

The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates.

The status register, shown in Figure 5, when latched, contains the current contents of the Control Word Register and status of the output and null count flag. (See detailed explanation of the Read-Back command.)

The actual counter is labelled CE (for "Counting Element"). It is a 16-bit presettable synchronous down counter.

OL_M and OL_L are two 8-bit latches. OL stands for "Output Latch"; the subscripts M and L stand for "Most significant byte" and "Least significant byte"

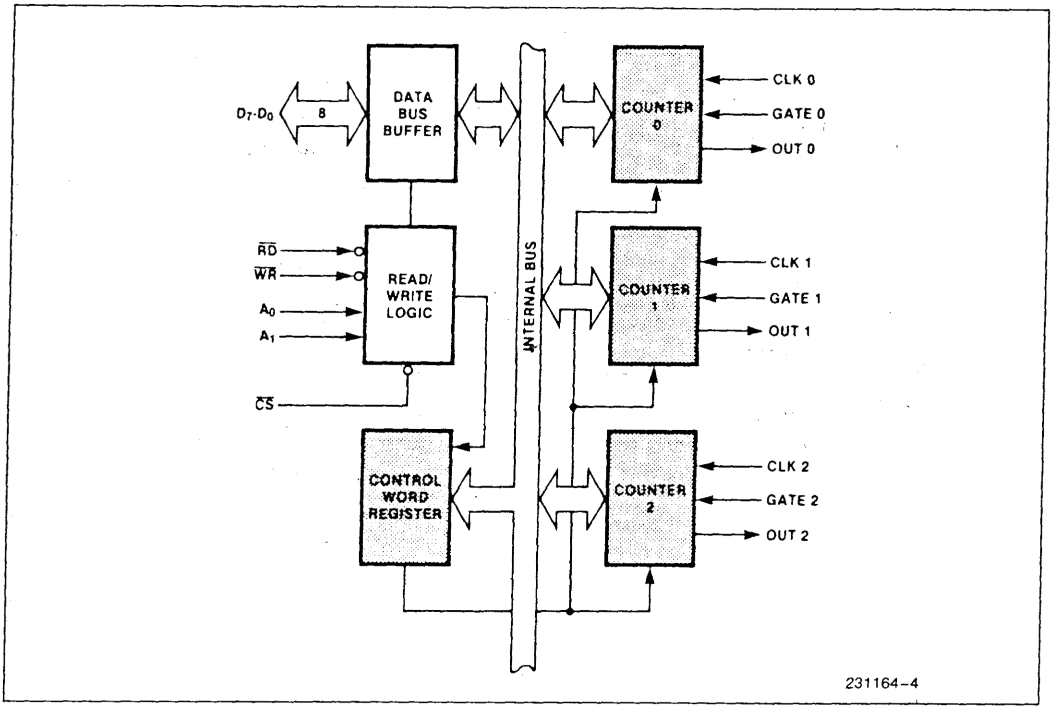


Figure 4. Block Diagram Showing Control Word Register and Counter Functions

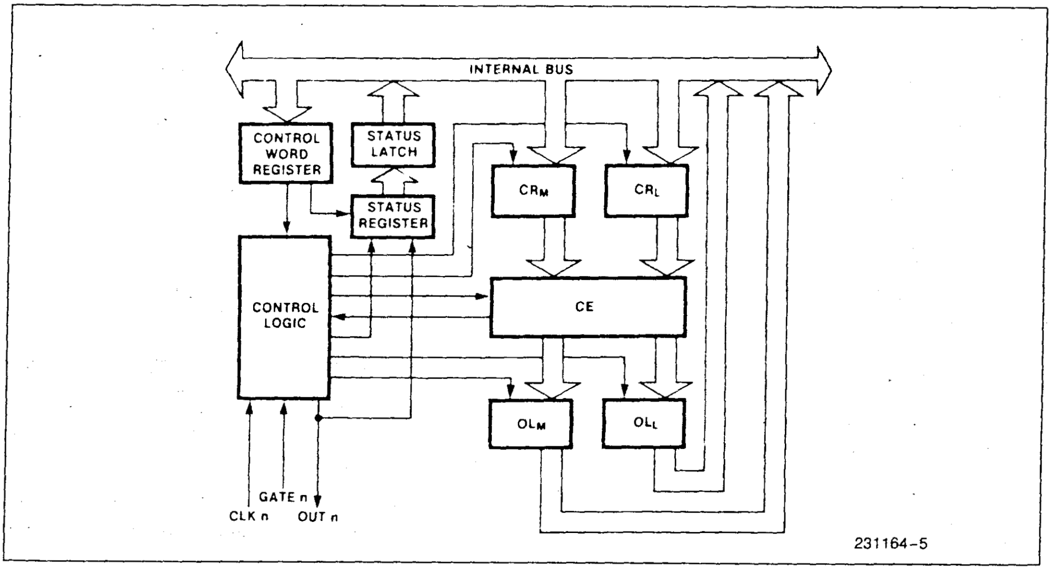


Figure 5. Internal Block Diagram of a Counter

respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter Latch Command is sent to the 8254, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CR_M and CR_L (for "Count Register"). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CR_M and CR_L are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

8254 SYSTEM INTERFACE

The 8254 is a component of the Intel Microcomputer Systems and interfaces in the same manner as all other peripherals of the family. It is treated by the system's software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A_0,A_1 connect to the A_0, A_1 address bus signals of the CPU. The CS can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel 8205 for larger systems.

OPERATIONAL DESCRIPTION

General

After power-up, the state of the 8254 is undefined. The Mode, count value, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused counters need not be programmed.

Programming the 8254

Counters are programmed by writing a Control Word and then an initial count.

The Control Words are written into the Control Word Register, which is selected when A_1,A_0 = 11. The Control Word itself specifies which Counter is being programmed.

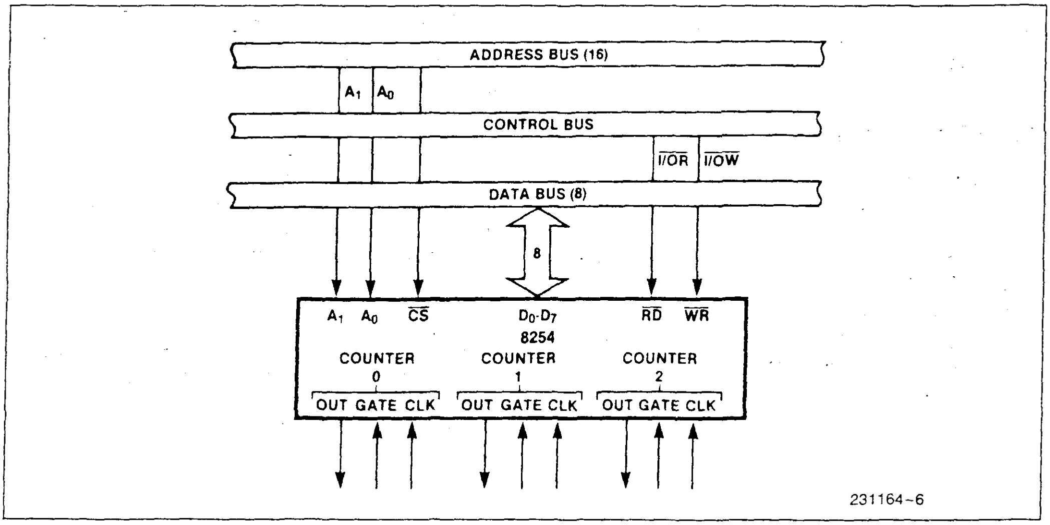


Figure 6. 8254 System Interface

Control Word Format

$A_1,A_0 = 11$ $\overline{CS} = 0$ $\overline{RD} = 1$ $\overline{WR} = 0$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC—Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (see Read Operations)

M—Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW—Read/Write

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first, then most significant byte

BCD

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 7. Control Word Format

By contrast, initial counts are written into the Counters, not the Control Word Register. The A_1,A_0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

Write Operations

The programming procedure for the 8254 is very flexible. Only two conventions need to be remembered:

1) For each Counter, the Control Word must be written before the initial count is written.
2) The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A_1,A_0 inputs), and each Control Word specifies the Counter it applies to (SC0,SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions in Figure 7 is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be affected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

	A_1	A_0
Control Word—Counter 0	1	1
LSB of count—Counter 0	0	0
MSB of count—Counter 0	0	0
Control Word—Counter 1	1	1
LSB of count—Counter 1	0	1
MSB of count—Counter 1	0	1
Control Word—Counter 2	1	1
LSB of count—Counter 2	1	0
MSB of count—Counter 2	1	0

	A_1	A_0
Control Word—Counter 2	1	1
Control Word—Counter 1	1	1
Control Word—Counter 0	1	1
LSB of count—Counter 2	1	0
MSB of count—Counter 2	1	0
LSB of count—Counter 1	0	1
MSB of count—Counter 1	0	1
LSB of count—Counter 0	0	0
MSB of count—Counter 0	0	0

	A_1	A_0
Control Word—Counter 0	1	1
Control Word—Counter 1	1	1
Control Word—Counter 2	1	1
LSB of count—Counter 2	1	0
LSB of count—Counter 1	0	1
LSB of count—Counter 0	0	0
MSB of count—Counter 0	0	0
MSB of count—Counter 1	0	1
MSB of count—Counter 2	1	0

	A_1	A_0
Control Word—Counter 1	1	1
Control Word—Counter 0	1	1
LSB of count—Counter 1	0	1
Control Word—Counter 2	1	1
LSB of count—Counter 0	0	0
MSB of count—Counter 1	0	1
LSB of count—Counter 2	1	0
MSB of count—Counter 0	0	0
MSB of count—Counter 2	1	0

NOTE:
In all four examples, all Counters are programmed to read/write two-byte counts. These are only four of many possible programming sequences.

Figure 8. A Few Possible Programming Sequences

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the 8254.

There are three possible methods for reading the counters: a simple read operation, the Counter Latch Command, and the Read-Back Command. Each is explained below. The first method is to perform a simple read operation. To read the Counter, which is selected with the A1, A0 inputs, the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in the process of changing when it is read, giving an undefined result.

COUNTER LATCH COMMAND

The second method uses the "Counter Latch Command". Like a Control Word, this command is written to the Control Word Register, which is selected when A_1,A_0 = 11. Also like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D5 and D4, distinguish this command from a Control Word.

A_1,A_0 = 11; CS = 0; RD = 1; WR = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	0	0	X	X	X	X

SC1,SC0—specify counter to be latched

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Read-Back Command

D5,D4—00 designates Counter Latch Command

X—don't care

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 9. Counter Latching Command Format

The selected Counter's output latch (OL) latches the count at the time the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE). This allows reading the contents of the Counters "on the fly" without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until it is read. Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read or write or programming operations of other Counters may be inserted between them.

Another feature of the 8254 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two byte counts, the following sequence is valid.

1) Read least significant byte.
2) Write new least significant byte.
3) Read most significant byte.
4) Write new most significant byte.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

READ-BACK COMMAND

The third method uses the Read-Back Command. This command allows the user to check the count value, programmed Mode, and current states of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in Figure 10. The command applies to the counters selected by setting their corresponding bits D3, D2, D1 = 1.

A0, A1 = 11 $\overline{\text{CS}}$ = 0 $\overline{\text{RD}}$ = 1 $\overline{\text{WR}}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{\text{COUNT}}$	$\overline{\text{STATUS}}$	CNT 2	CNT 1	CNT 0	0

D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counters(s)
D_3: 1 = Select Counter 2
D_2: 1 = Select Counter 1
D_1: 1 = Select Counter 0
D_0: Reserved for future expansion; Must be 0

Figure 10. Read-Back Command Format

The read-back command may be used to latch multiple counter output latches (OL) by setting the $\overline{\text{COUNT}}$ bit D5 = 0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). The counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting $\overline{\text{STATUS}}$ bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.

The counter status format is shown in Figure 11. Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUTPUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Output	Null Count	RW1	RW0	M2	M1	M0	BCD

D_7 1 = OUT Pin is 1
0 = OUT Pin is 0

D_6 1 = Null Count
0 = Count available for reading

D_5–D_0 Counter programmed mode (see Figure 7)

Figure 11. Status Byte

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the Mode of the counter and is described in the Mode Definitions, but until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown in Figure 12.

This Action	Causes
A. Write to the control word register;[1]	Null Count = 1
B. Write to the count register (CR);[2]	Null Count = 1
C. New Count is loaded into CE (CR → CE);	Null Count = 0

NOTE:

1. Only the counter specified by the control word will have its Null Count set to 1. Null count bits of other counters are unaffected.
2. If the counter is programmed for two-byte counts (least significant byte then most significant byte) Null Count goes to 1 when the second byte is written.

Figure 12. Null Count Operation

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is the status of the counter at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both $\overline{\text{COUNT}}$ and $\overline{\text{STATUS}}$ bits D5,D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored. This is illustrated in Figure 13.

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

$\overline{\text{CS}}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	A_1	A_0	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3-State)
1	X	X	X	X	No-Operation (3-State)
0	1	1	X	X	No-Operation (3-State)

Figure 14. Read/Write Operations Summary

Command								Description	Result
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	1	0	0	0	0	1	0	Read back count and status of Counter 0	Count and status latched for Counter 0
1	1	1	0	0	1	0	0	Read back status of Counter 1	Status latched for Counter 1
1	1	1	0	1	1	0	0	Read back status of Counters 2, 1	Status latched for Counter 2, but not Counter 1
1	1	0	1	1	0	0	0	Read back count of Counter 2	Count latched for Counter 2
1	1	0	0	0	1	0	0	Read back count and status of Counter 1	Count latched for Counter 1, but not status
1	1	1	0	0	0	1	0	Read back status of Counter 1	Command ignored, status already latched for Counter 1

Figure 13. Read-Back Command Example

Mode Definitions

The following are defined for use in describing the operation of the 8254.

CLK Pulse: a rising edge, then a falling edge, in that order, of a Counter's CLK input.

Trigger: a rising edge of a Counter's GATE input.

Counter loading: the transfer of a count from the CR to the CE (refer to the "Functional Description")

MODE 0: INTERRUPT ON TERMINAL COUNT

Mode 0 is typically used for event counting. After the Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the Counter.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not go high until N + 1 CLK pulses after the initial count is written.

If a new count is written to the Counter, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte disables counting. OUT is set low immediately (no clock pulse required)
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the counting sequence to be synchronized by software. Again, OUT does not go high until N+1 CLK pulses after the new count of N is written.

If an initial count is written while GATE = 0, it will still be loaded on the next CLK pulse. When GATE goes high, OUT will go high N CLK pulses later; no CLK pulse is needed to load the Counter as this has already been done.

MODE 1: HARDWARE RETRIGGERABLE ONE-SHOT

OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the counter is retriggered. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

MODE 2: RATE GENERATOR

This Mode functions like a divide-by-N counter. It is typically used to generate a Real Time Clock interrupt. OUT will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

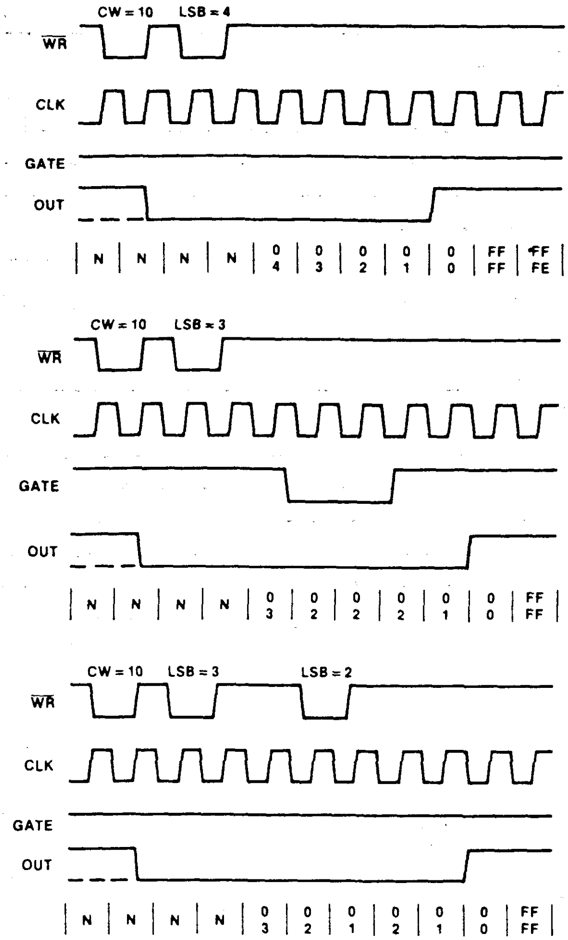
GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. OUT goes low N CLK Pulses after the initial count is written. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current counting cycle. In mode 2, a COUNT of 1 is illegal.

MODE 3: SQUARE WAVE MODE

Mode 3 is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the


CW = 10
LSB = 4
WR
CLK
GATE
OUT
N N N N 0 4 0 3 0 2 0 1 0 0 FF FF FF FE
CW = 10
LSB = 3
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 2 0 2 0 1 0 0 FF FF
CW = 10
LSB = 3
LSB = 2
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 1 0 2 0 1 0 0 FF FF

231164-7

NOTE:

The following conventions apply to all mode timing diagrams:

1. Counters are programmed for binary (not BCD) counting and for reading/writing least significant byte (LSB) only.
2. The counter is always selected ($\overline{CS}$ always low).
3. CW stands for "Control Word"; CW = 10 means a control word of 10 HEX is written to the counter.
4. LSB stands for "Least Significant Byte" of count.
5. Numbers below diagrams are count values. The lower number is the least significant byte. The upper number is the most significant byte. Since the counter is programmed to read/write LSB only, the most significant byte cannot be read.
   N stands for an undefined count.
   Vertical lines show transitions between count values.

Figure 15. Mode 0

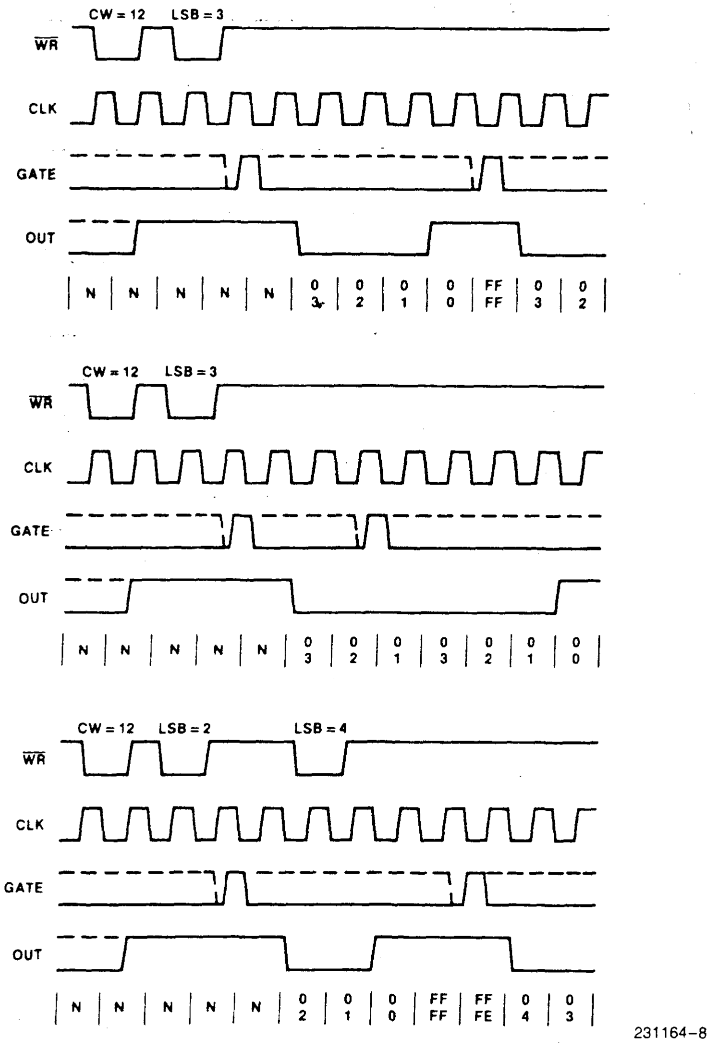


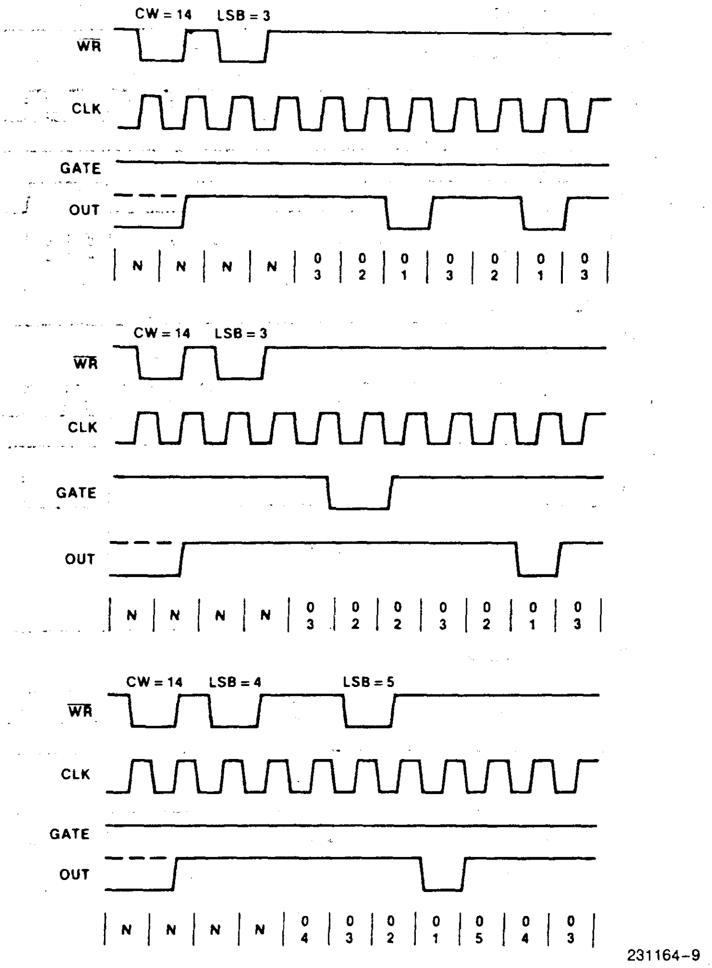
Figure 16. Mode 1

initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the


CW = 14
LSB = 3
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 1 0 3 0 2 0 1 0 3
CW = 14
LSB = 3
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 2 0 3 0 2 0 1 0 3
CW = 14
LSB = 4
LSB = 5
WR
CLK
GATE
OUT
N N N N 0 4 0 3 0 2 0 1 0 5 0 4 0 3
231164–9

NOTE:
A GATE transition should not occur one clock prior to terminal count.

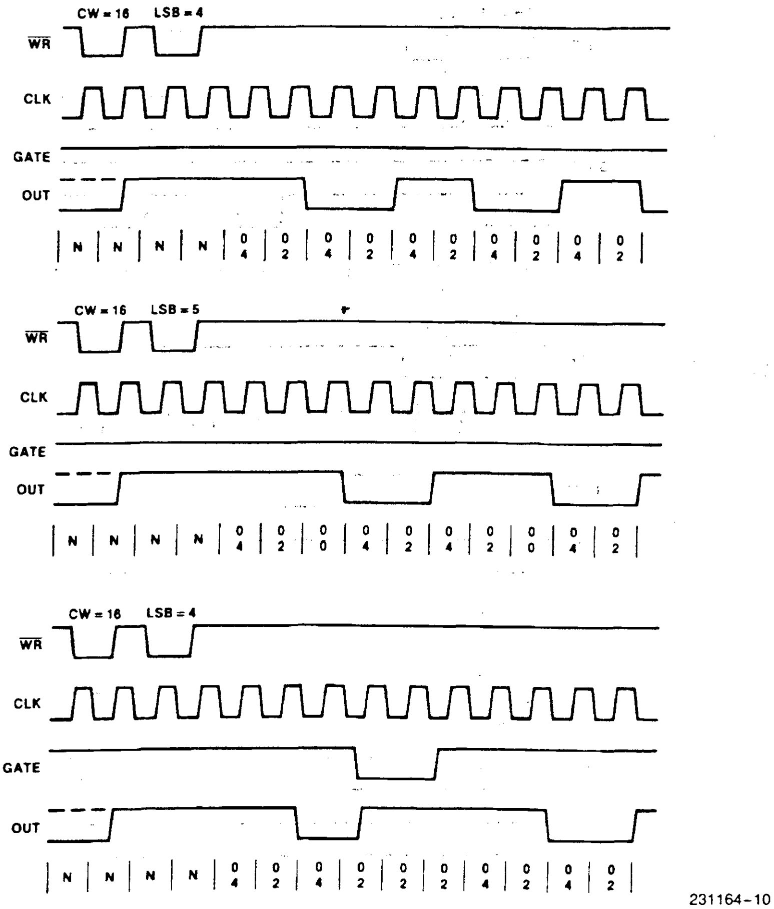
Figure 17. Mode 2

new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

Mode 3 is implemented as follows:

Even counts: OUT is initially high. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires OUT changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

Odd counts: OUT is initially high. The initial count minus one (an even number) is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. One CLK pulse *after* the count expires, OUT goes low and the Counter is reloaded with the initial count minus one. Succeeding CLK pulses decrement the count by two. When the count expires, OUT goes high again and the Counter is reloaded with the initial count minus one. The above process is repeated indefinitely. So for odd counts, OUT will be high for (N + 1)/2 counts and low for (N − 1)/2 counts.


CW = 16 LSB = 4
WR
CLK
GATE
OUT
N N N N 0 4 0 2 0 4 0 2 0 4 0 2 0 4 0 2 0 4 0 2
CW = 16 LSB = 5
WR
CLK
GATE
OUT
N N N N 0 4 0 2 0 0 0 4 0 2 0 4 0 2 0 0 0 4 0 2
CW = 16 LSB = 4
WR
CLK
GATE
OUT
N N N N 0 4 0 2 0 4 0 2 0 2 0 2 0 4 0 2 0 4 0 2
231164-10

NOTE:
A GATE transition should not occur one clock prior to terminal count.

Figure 18. Mode 3

MODE 4: SOFTWARE TRIGGERED STROBE

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse and then go high again. The counting sequence is "triggered" by writing the initial count.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte has no effect on counting.
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. OUT strobes low N + 1 CLK pulses after the new count of N is written.

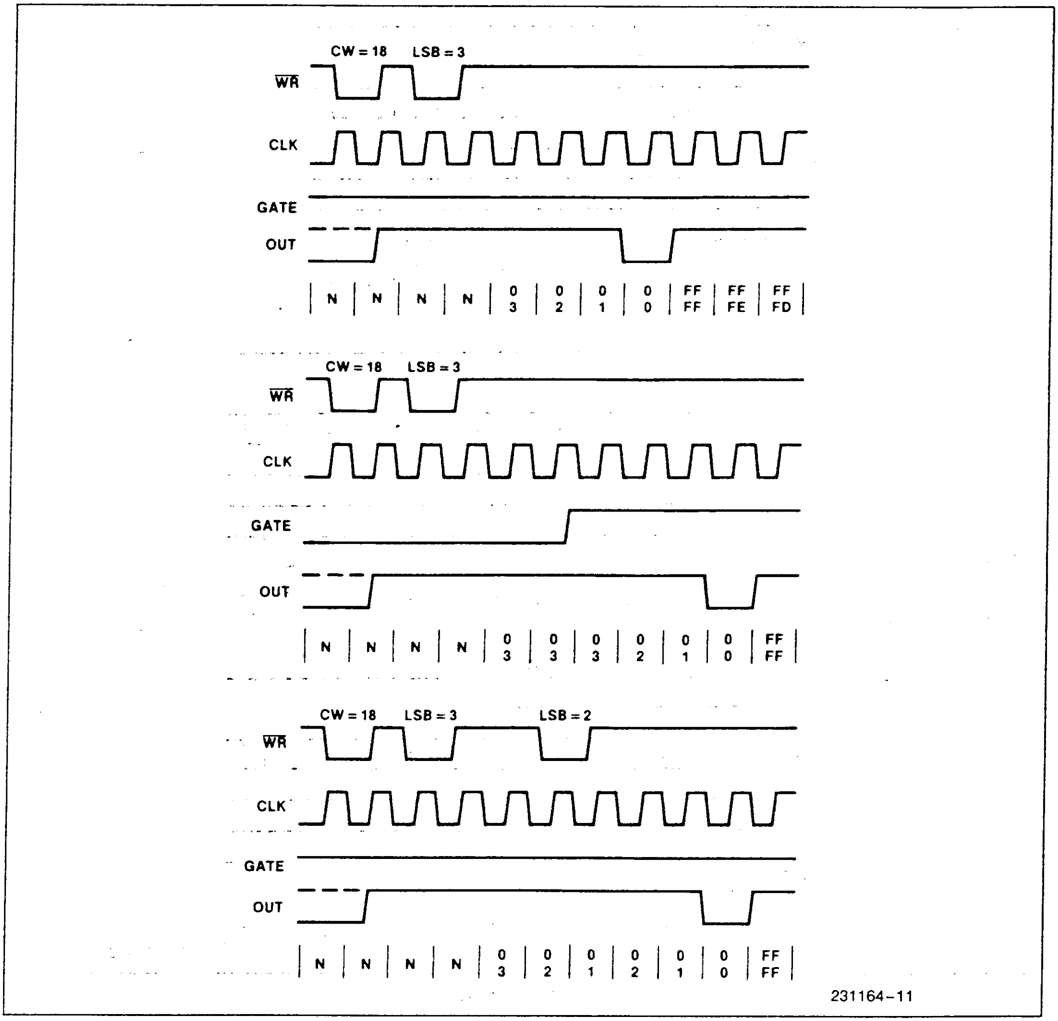

CW = 18
LSB = 3
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 1 0 0 FF FF FF FE FF FD
CW = 18
LSB = 3
WR
CLK
GATE
OUT
N N N N 0 3 0 3 0 3 0 2 0 1 0 0 FF FF
CW = 18
LSB = 3
LSB = 2
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 1 0 2 0 1 0 0 FF FF
231164-11

Figure 19. Mode 4

MODE 5: HARDWARE TRIGGERED STROBE (RETRIGGERABLE)

OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again.

After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after a trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. OUT will not strobe low for N + 1 CLK pulses after any trigger. GATE has no effect on OUT.

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

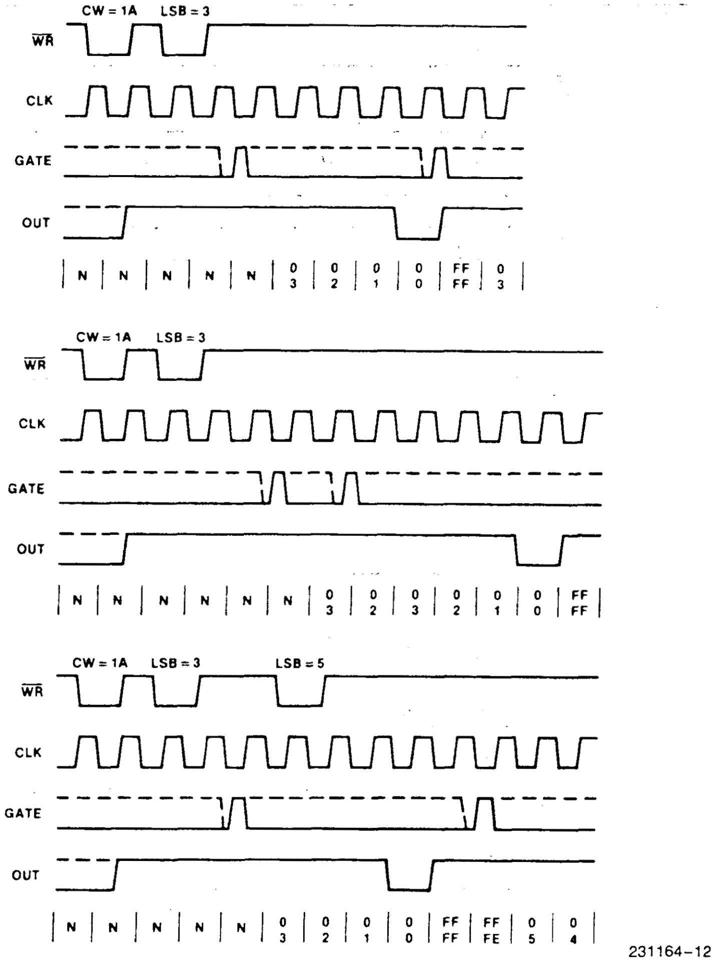

CW = 1A
LSB = 3
WR
CLK
GATE
OUT
N N N N N 0 3 0 2 0 1 0 0 FF FF 0 3
CW = 1A
LSB = 3
WR
CLK
GATE
OUT
N N N N N N 0 3 0 2 0 3 0 2 0 1 0 0 FF FF
CW = 1A
LSB = 3
LSB = 5
WR
CLK
GATE
OUT
N N N N N 0 3 0 2 0 1 0 0 FF FF FF FE 0 5 0 4
231164-12

Figure 20. Mode 5

Signal Status Modes	Low Or Going Low	Rising	High
0	Disables Counting	— —	Enables Counting
1	— —	1) Initiates Counting 2) Resets Output after Next Clock	— —
2	1) Disables Counting 2) Sets Output Immediately High	Initiates Counting	Enables Counting
3	1) Disables Counting 2) Sets Output Immediately High	Initiates Counting	Enables Counting
4	Disables Counting	— —	Enables Counting
5	— —	Initiates Counting	— —

Figure 21. Gate Pin Operations Summary

Mode	Min Count	Max Count
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0
5	1	0

NOTE:
0 is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

Figure 22. Minimum and Maximum Initial Counts

Operation Common to All Modes

PROGRAMMING

When a Control Word is written to a Counter, all Control Logic is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

GATE

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3, and 4 the GATE input is level sensitive, and the logic level is sampled on the rising edge of CLK. In Modes 1, 2, 3, and 5 the GATE input is rising-edge sensitive. In these Modes, a rising edge of GATE (trigger) sets an edge-sensitive flip-flop in the Counter. This flip-flop is then sampled on the next rising edge of CLK; the flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs—a high logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive. In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following $\overline{WR}$ of a new count value.

COUNTER

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The Counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5 the Counter "wraps around" to the highest count, either FFFF hex for binary counting or 9999 for BCD counting, and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.

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..............−0.5V to +7V

Power Dissipation 1W

**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 T_A = 0°C to 70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5V	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0.45V
I_{CC}	V_{CC} Supply Current		170	mA	
C_{IN}	Input Capacitance		10	pF	f_c = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to V_{SS}[4]

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND = 0V

Bus Parameters[1]

READ CYCLE

Symbol	Parameter	8254-5		8254		8254-2		Unit
		Min	Max	Min	Max	Min	Max	
t_{AR}	Address Stable Before $\overline{RD}$ ↓	45		45		30		ns
t_{SR}	$\overline{CS}$ Stable Before $\overline{RD}$ ↓	0		0		0		ns
t_{RA}	Address Hold Time After $\overline{RD}$ ↑	0		0		0		ns
t_{RR}	$\overline{RD}$ Pulse Width	150		150		95		ns
t_{RD}	Data Delay from $\overline{RD}$ ↓		120		120		85	ns
t_{AD}	Data Delay from Address		220		220		185	ns
t_{DF}	$\overline{RD}$ ↑ to Data Floating	5	90	5	90	5	65	ns
t_{RV}	Command Recovery Time	200		200		165		ns

NOTE:

[1] AC timings measured at V_{OH} = 2.0V, V_{OL} = 0.8V.

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND = 0V (Continued)

WRITE CYCLE

Symbol	Parameter	8254-5		8254		8254-2		Unit
		Min	Max	Min	Max	Min	Max	
t_{AW}	Address Stable Before $\overline{WR}$ ↓	0		0		0		ns
t_{SW}	$\overline{CS}$ Stable Before $\overline{WR}$ ↓	0		0		0		ns
$t_{\overline{WA}}$	Address Hold Time After $\overline{WR}$ ↓	0		0		0		ns
t_{WW}	$\overline{WR}$ Pulse Width	150		150		95		ns
t_{DW}	Data Setup Time Before $\overline{WR}$ ↑	120		120		95		ns
t_{WD}	Data Hold Time After $\overline{WR}$ ↑	0		0		0		ns
t_{RV}	Command Recovery Time	200		200		165		ns

CLOCK AND GATE

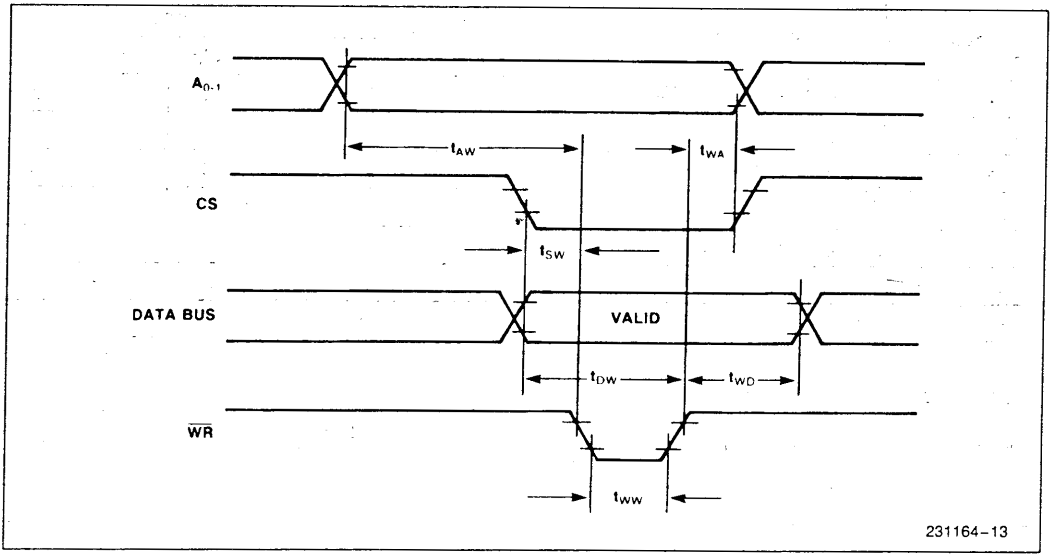
Symbol	Parameter	8254-5		8254		8254-2		Unit
		Min	Max	Min	Max	Min	Max	
t_{CLK}	Clock Period	200	DC	125	DC	100	DC	ns
t_{PWH}	High Pulse Width	60[3]		60[3]		30[3]		ns
t_{PWL}	Low Pulse Width	60[3]		60[3]		50[3]		ns
t_R	Clock Rise Time		25		25		25	ns
t_F	Clock Fall Time		25		25		25	ns
t_{GW}	Gate Width High	50		50		50		ns
t_{GL}	Gate Width Low	50		50		50		ns
t_{GS}	Gate Setup Time to CLK ↑	50		50		40		ns
t_{GH}	Gate Setup Time After CLK ↑	50[2]		50[2]		50[2]		ns
t_{OD}	Output Delay from CLK ↓		150		150		100	ns
t_{ODG}	Output Delay from Gate ↓		120		120		100	ns
t_{WC}	CLK Delay for Loading ↓	0	55	0	55	0	55	ns
t_{WG}	Gate Delay for Sampling	−5	50	−5	50	−5	40	ns
t_{WO}	OUT Delay from Mode Write		260		260		240	ns
t_{CL}	CLK Set Up for Count Latch	−40	45	−40	45	−40	40	ns

NOTES:

2. In Modes 1 and 5 triggers are sampled on each rising clock edge. A second trigger within 120 ns (70 ns for the 8254-2) of the rising clock edge may not be detected.
3. Low-going glitches that violate t_{PWH}, t_{PWL} may cause errors requiring counter reprogramming.
4. Sampled, not 100% tested. T_A = 25°C.
5. If CLK present at TWC min then Count equals N+2 CLK pulses, TWC max equals Count N+1 CLK pulse. TWC min to TWC max, count will be either N+1 or N+2 CLK pulses.
6. In Modes 1 and 5, if GATE is present when writing a new Count value, at TWG min Counter will not be triggered, at TWG max Counter will be triggered.
7. If CLK present when writing a Counter Latch or ReadBack Command, at TCL min CLK will be reflected in count value latched, at TCL max CLK will not be reflected in the count value latched.

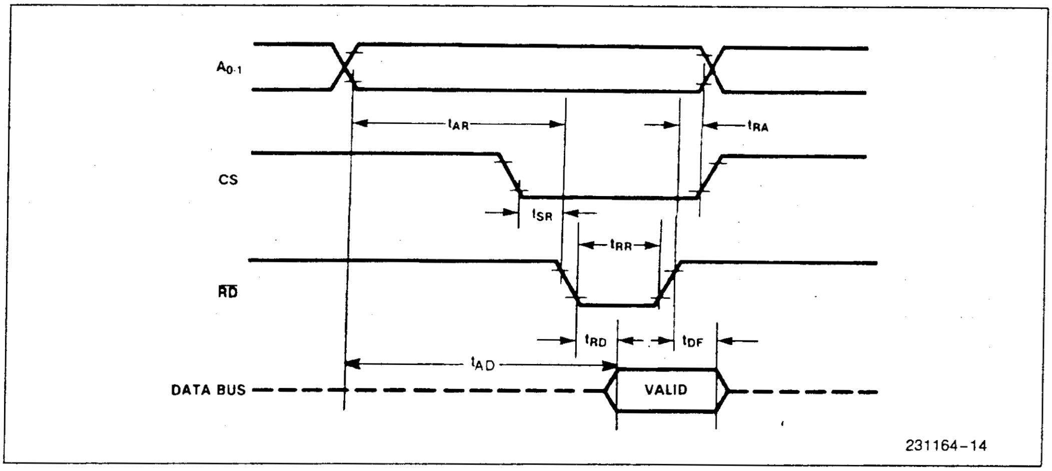
WAVEFORMS

WRITE



231164–13

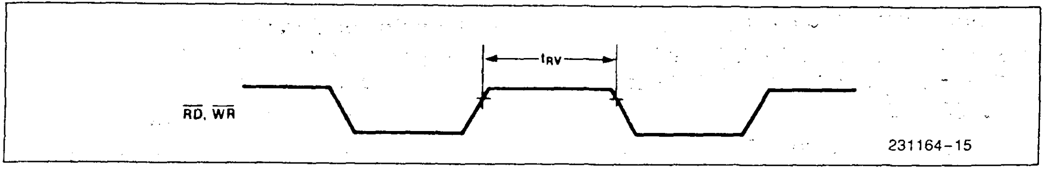
READ



231164–14

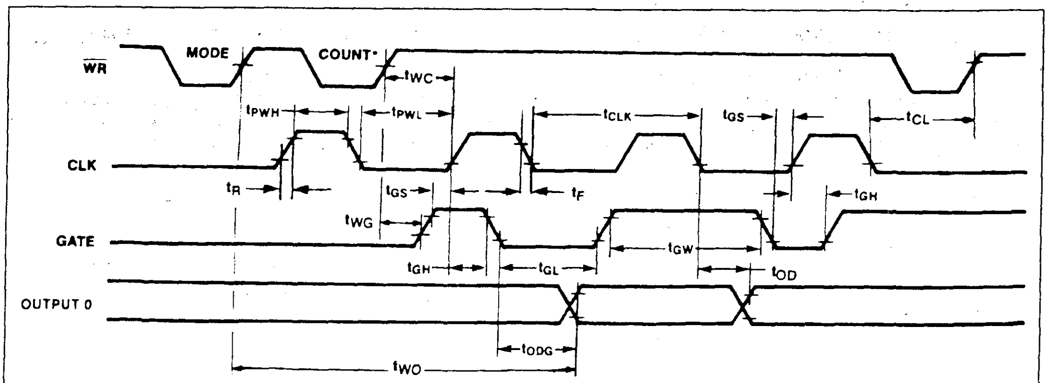
WAVEFORMS (Continued)

RECOVERY



231164-15

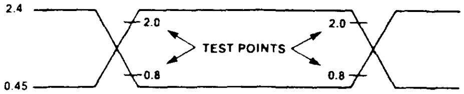
CLOCK AND GATE



231164-16

*Last byte of count being written.

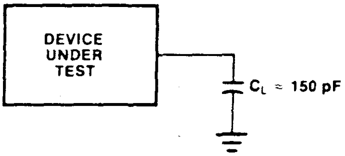
A.C. TESTING INPUT, OUTPUT WAVEFORM



231164-17

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0." Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT



231164-18

C_L = 150 pF
C_L Includes Jig Capacitance

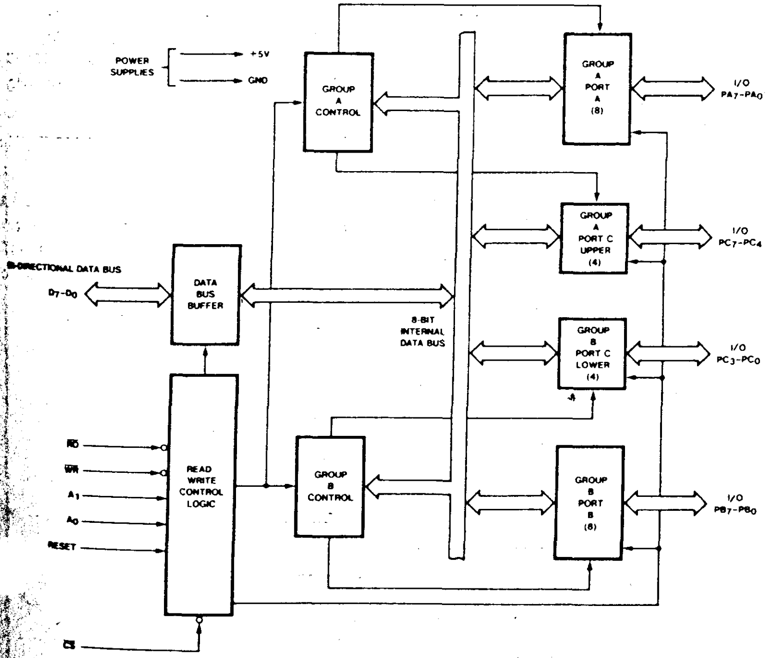
8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5

  24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability
- Available in EXPRESS
  - —Standard Temperature Range
  - —Extended Temperature Range
- 40 Pin DIP Package or 44 Lead PLCC

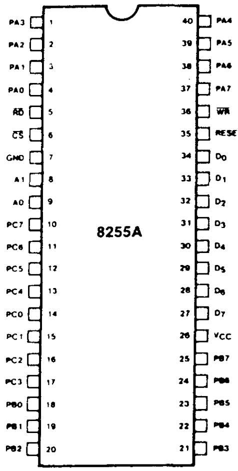
  (See Intel Packaging: Order Number: 231369)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.



231308-1

Figure 1. 8255A Block Diagram



231308-2

Figure 2. Pin Configuration

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

($\overline{RD}$)

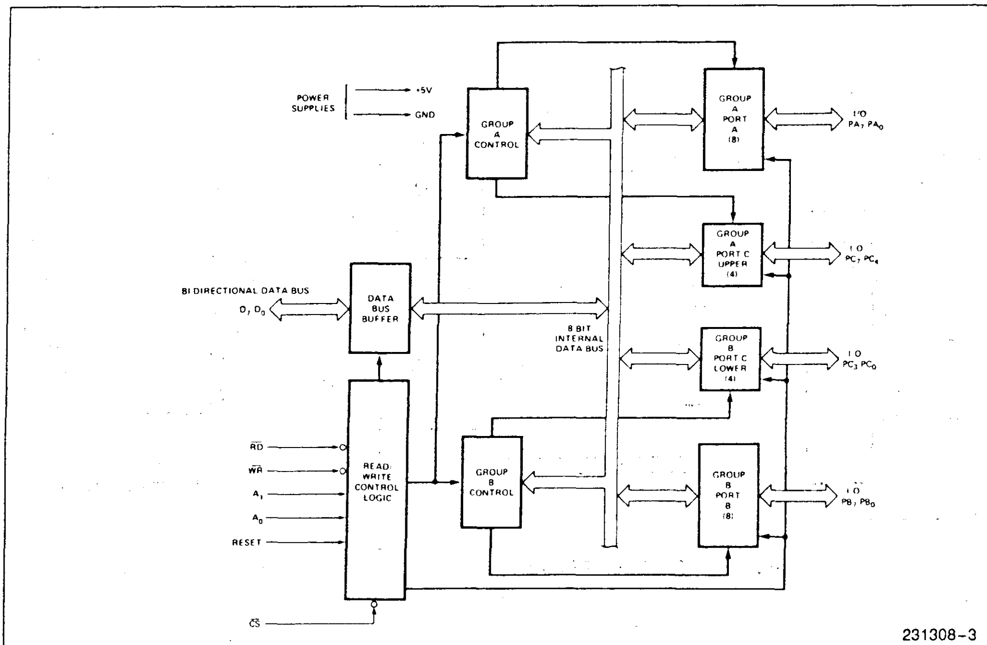
Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).



231308-3

Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	**Input Operation (READ)**
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
					Output Operation (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
X	X	X	X	1	Data Bus → 3-State
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Data Bus → 3-State

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A—Port A and Port C upper (C7–C4)

Control Group B—Port B and Port C lower (C3–C0)

The Control Word Register can **Only** be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

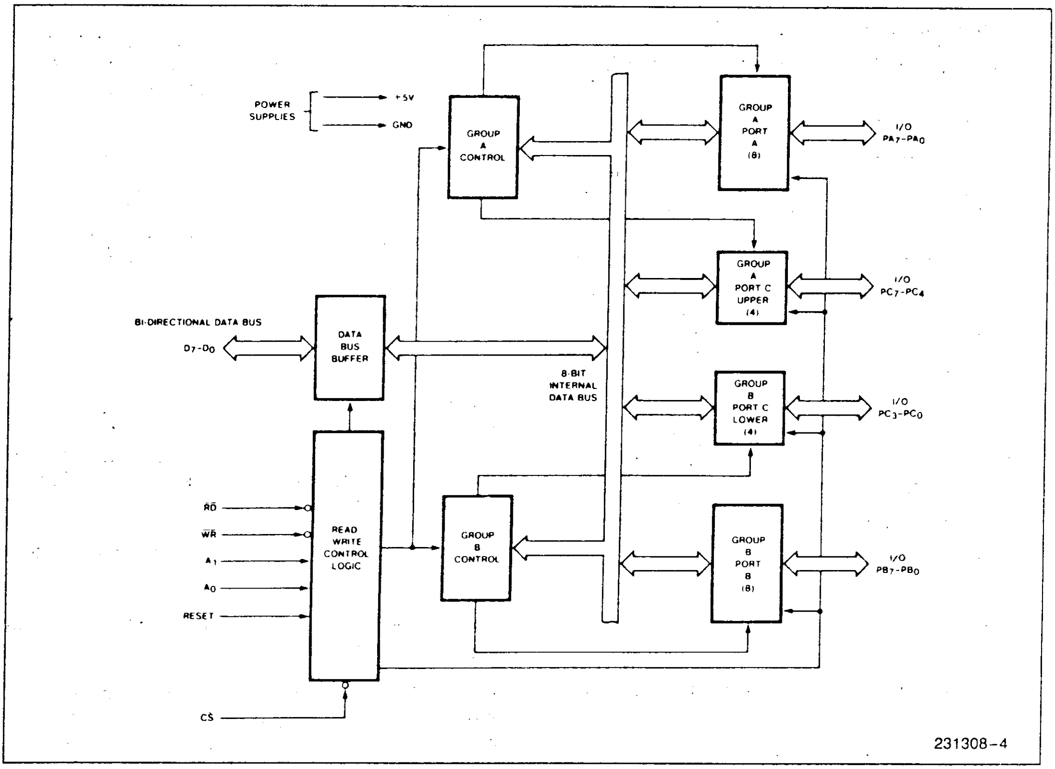
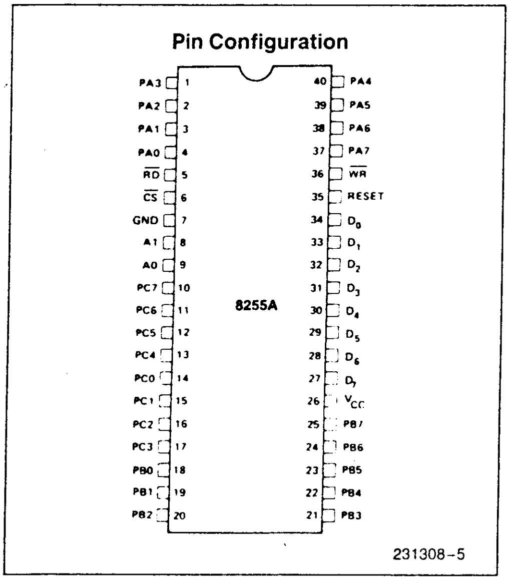


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions



Pin Configuration

Pin Names

D_7–D_0	Data Bus (Bi-Directional)
RESET	Reset Input
$\overline{CS}$	Chip Select
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A0, A1	Port Address
PA7–PA0	Port A (BIT)
PB7–PB0	Port B (BIT)
PC7–PC0	Port C (BIT)
V_{CC}	+5 Volts
GND	0 Volts

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0—Basic Input/Output

Mode 1—Strobed Input/Output

Mode 2—Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

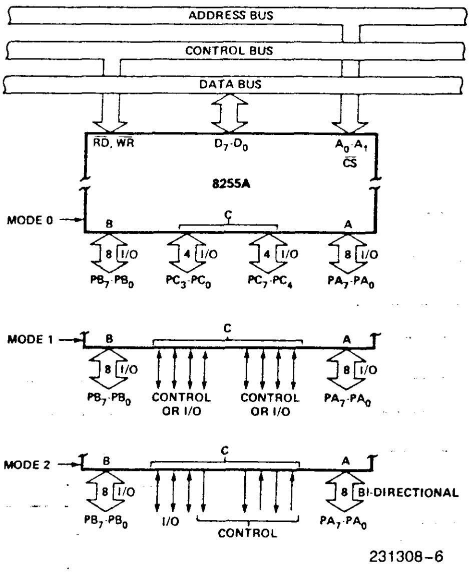


Figure 5. Basic Mode Definitions and Bus Interface

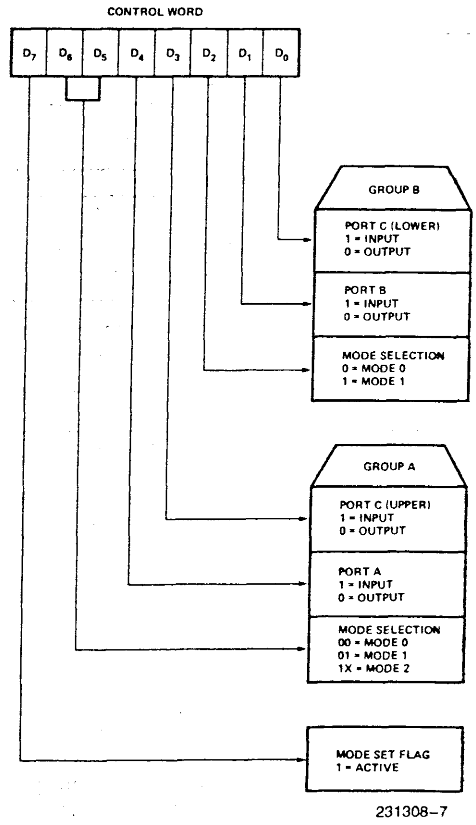


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

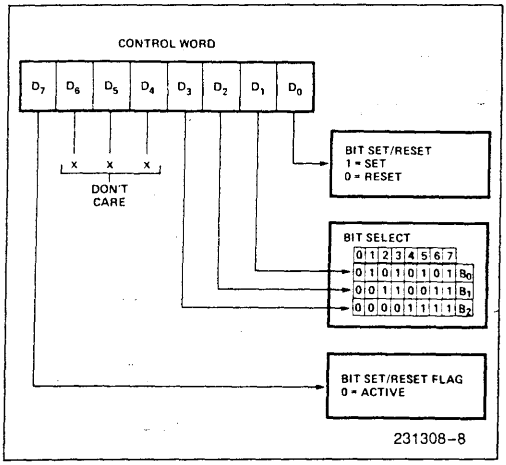


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is set—Interrupt enable

(BIT-RESET)—INTE is RESET—Interrupt disable

NOTE:
All Mask flip-flops are automatically reset during mode selection and device Reset.

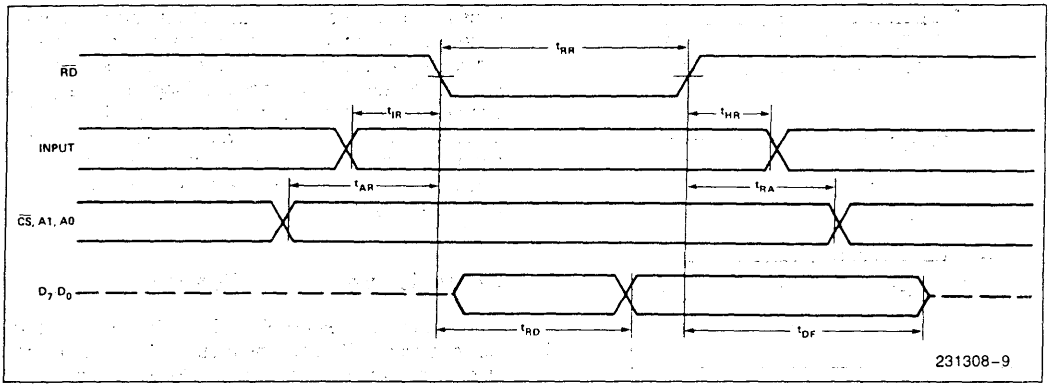
Operating Modes

MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

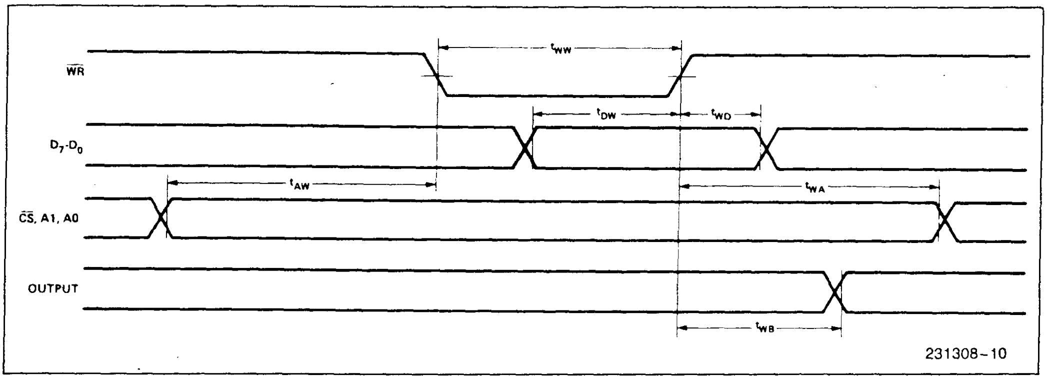
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

MODE 0 (BASIC INPUT)



MODE 0 (BASIC OUTPUT)



231308-10

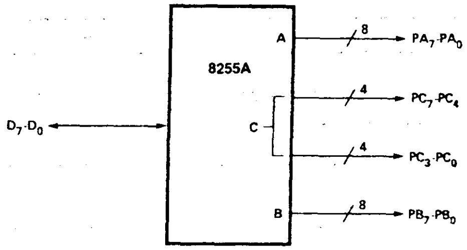
MODE 0 PORT DEFINITION

A		B		Group A			Group B	
D_4	D_3	D_1	D_0	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE CONFIGURATIONS

CONTROL WORD #0

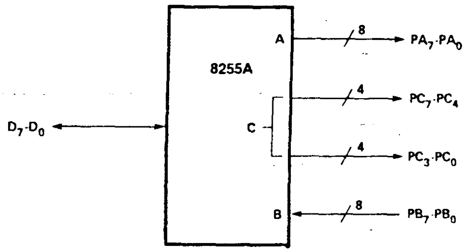
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	0



231308–11

CONTROL WORD #2

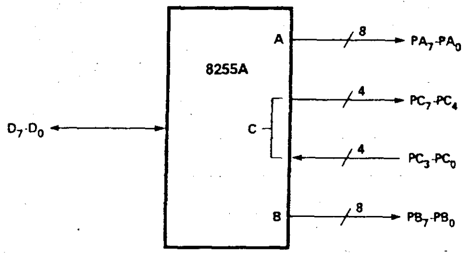
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	0



231308–12

CONTROL WORD #1

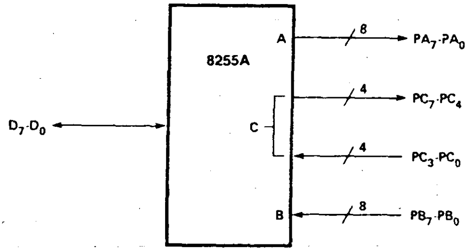
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	1



231308–13

CONTROL WORD #3

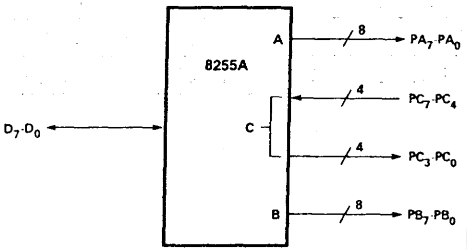
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	1



231308–14

CONTROL WORD #4

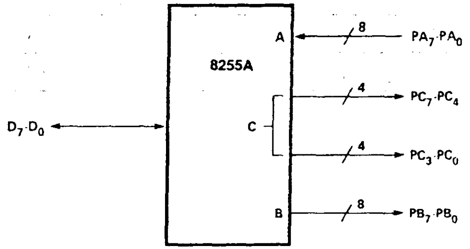
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



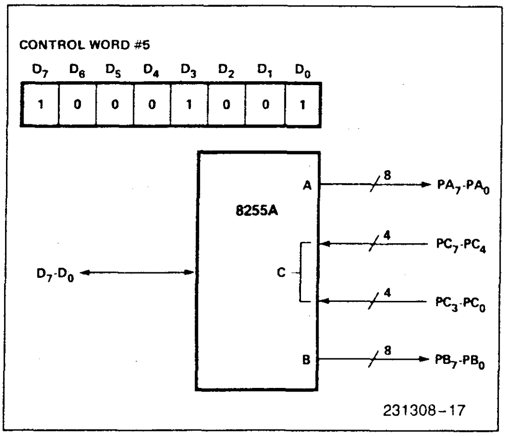
231308–15

CONTROL WORD #8

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0

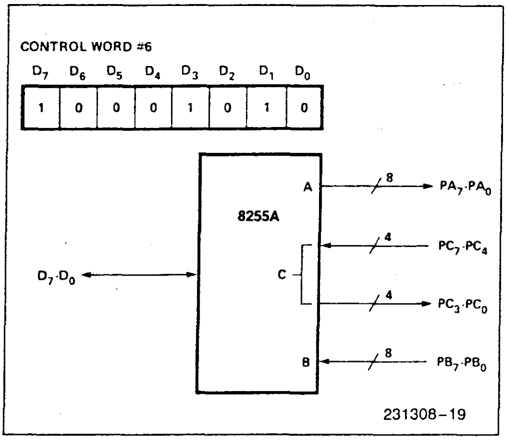


231308–16


CONTROL WORD #5

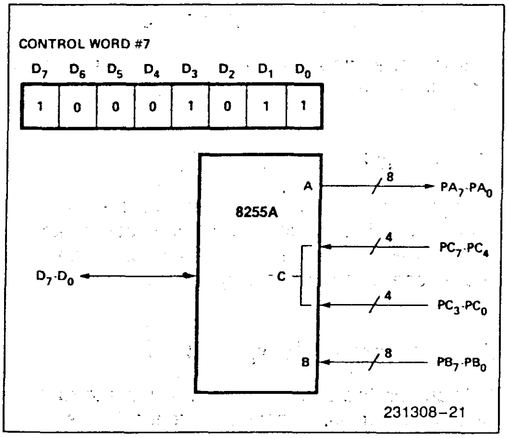
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	1

231308-17


CONTROL WORD #6

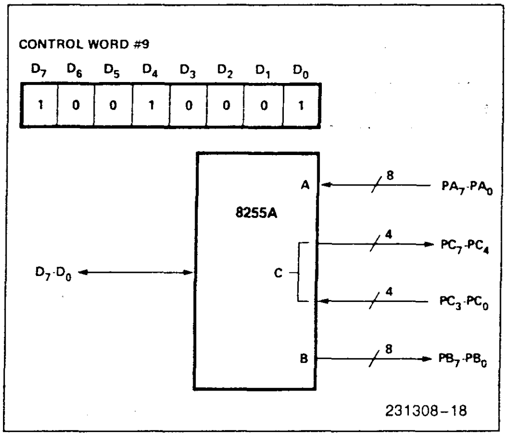
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	0

231308-19


CONTROL WORD #7

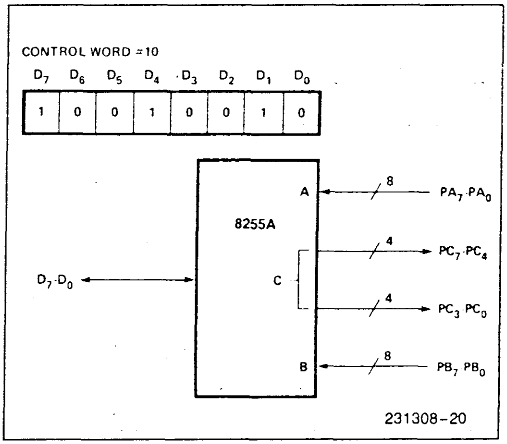
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	1

231308-21


CONTROL WORD #9

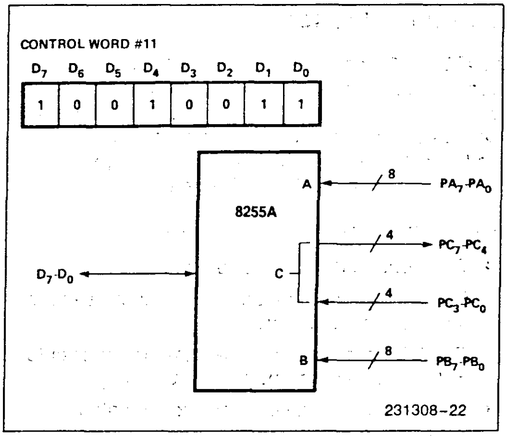
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	1

231308-18


CONTROL WORD =10

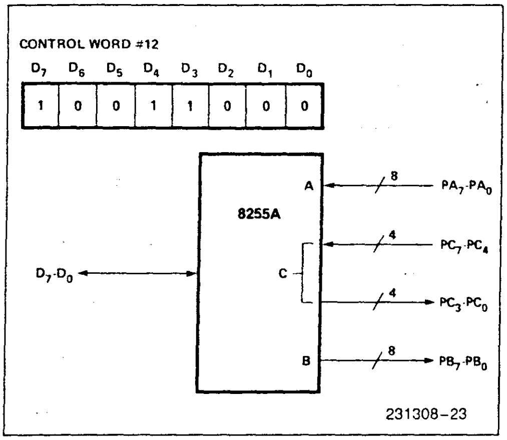
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0

231308-20

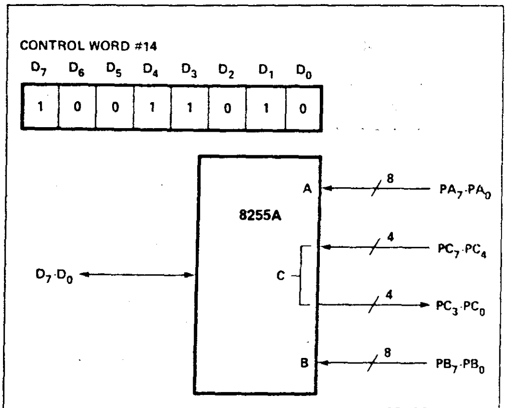

CONTROL WORD #11

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	1

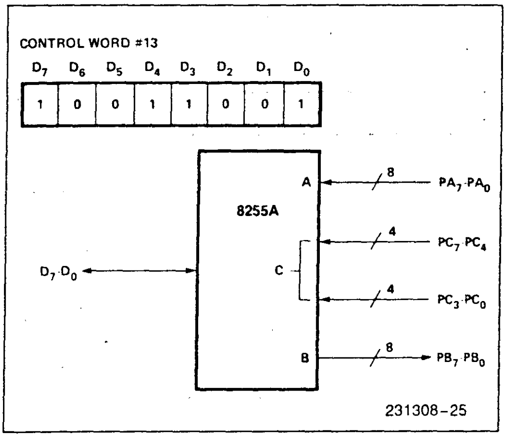
231308-22



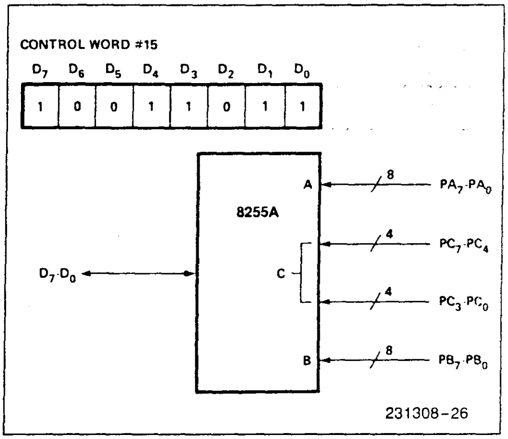
231308-23



231308-24



231308-25



231308-26

Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

$\overline{\text{STB}}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{\text{STB}}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{\text{RD}}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

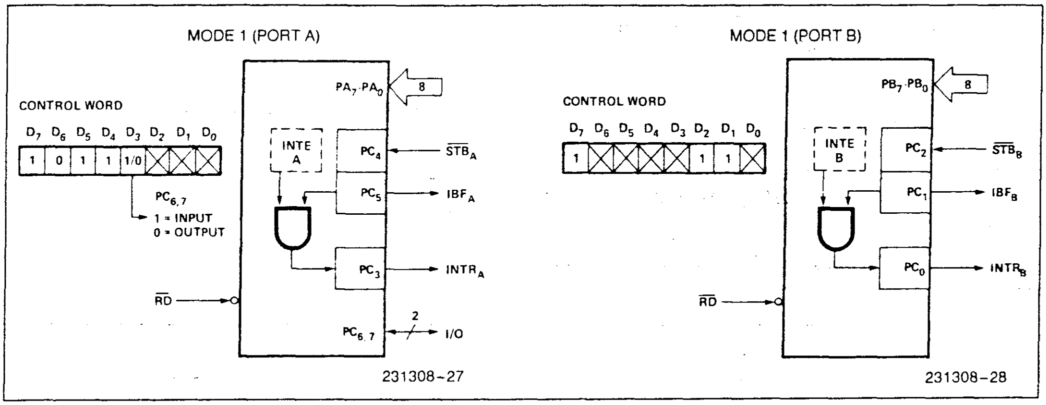


Figure 8. MODE 1 Input

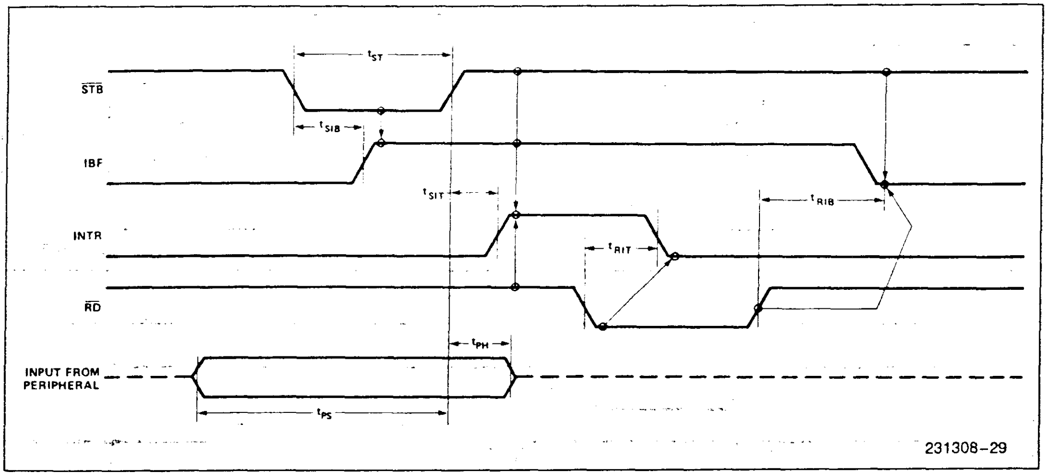


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{\text{OBF}}$ (Output Buffer Full F/F). The $\overline{\text{OBF}}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{\text{OBF}}$ F/F will be set by the rising edge of the $\overline{\text{WR}}$ input and reset by $\overline{\text{ACK}}$ input being low.

$\overline{\text{ACK}}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{\text{ACK}}$ is a "one", $\overline{\text{OBF}}$ is a "one", and INTE is a "one". It is reset by the falling edge of $\overline{\text{WR}}$.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

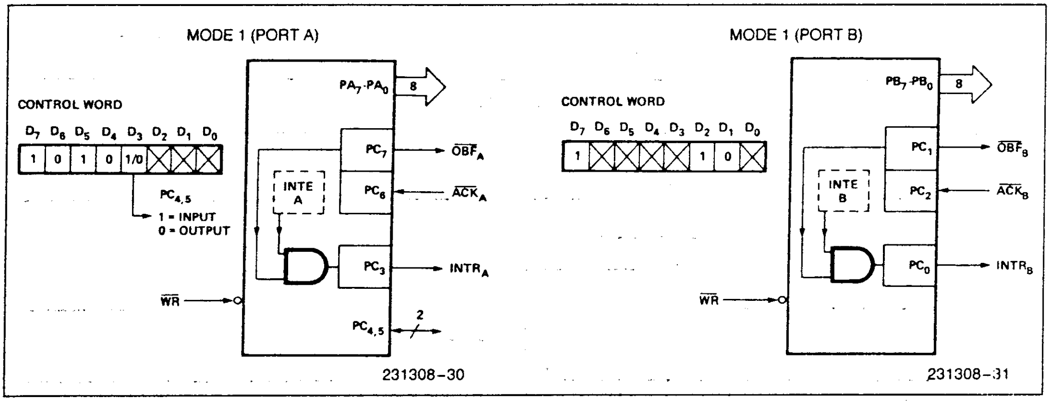


Figure 10. MODE 1 Output

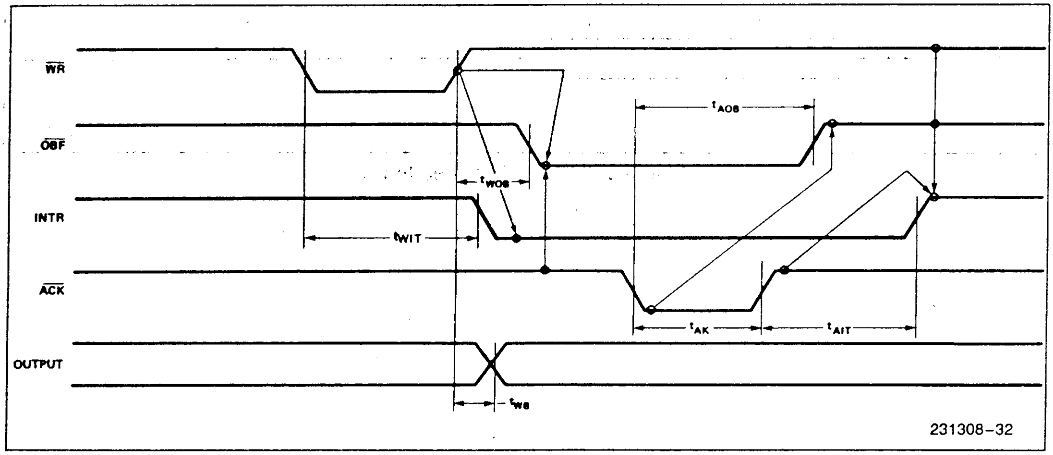
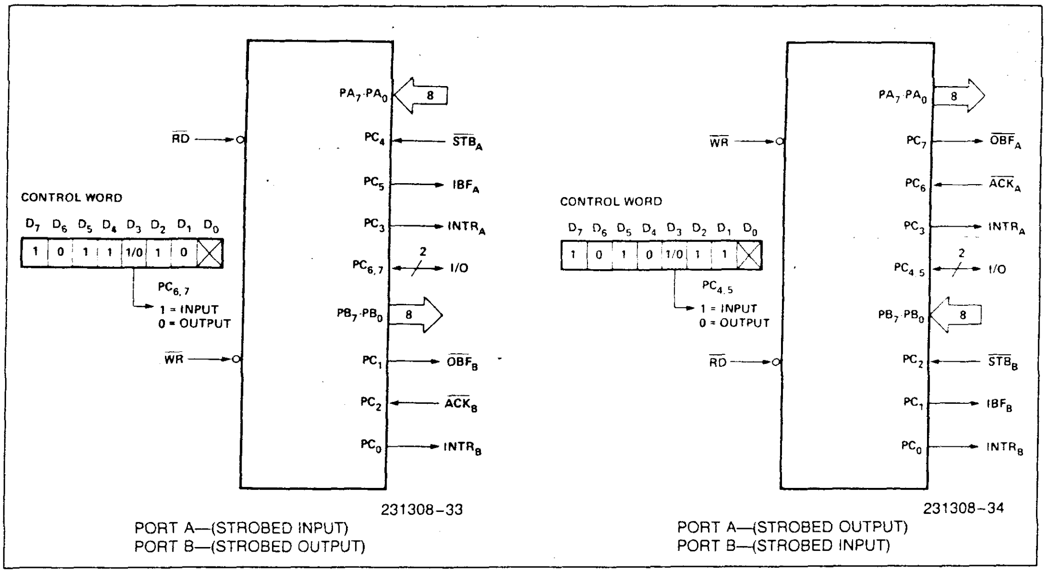


Figure 11. MODE 1 (Strobed Output)



PORT A—(STROBED INPUT)
PORT B—(STROBED OUTPUT)

PORT A—(STROBED OUTPUT)
PORT B—(STROBED INPUT)

Figure 12. Combinations of MODE 1

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in MODE 1 to support a wide variety of strobed I/O applications.

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A **only**.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{\text{OBF}}$ (Output Buffer Full). The $\overline{\text{OBF}}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{\text{ACK}}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{\text{STB}}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

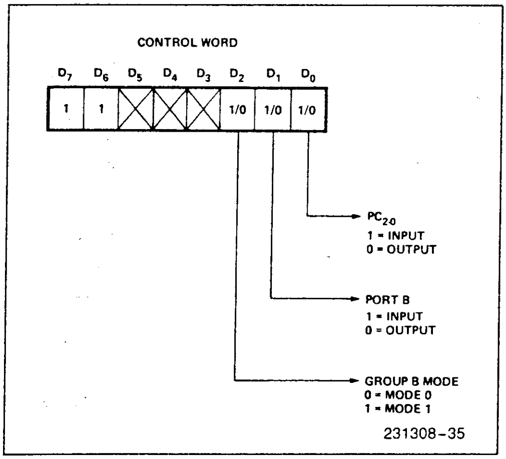


Figure 13. MODE Control Word

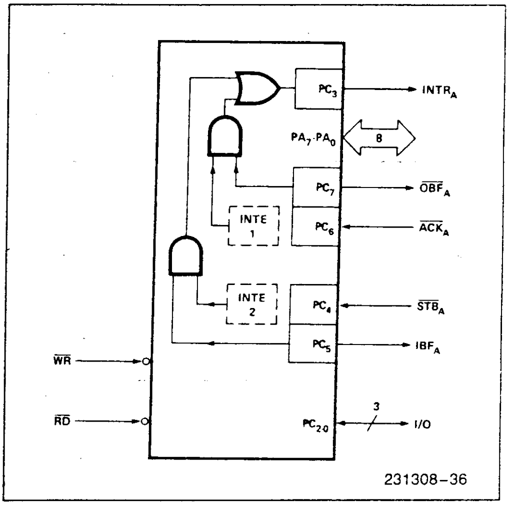
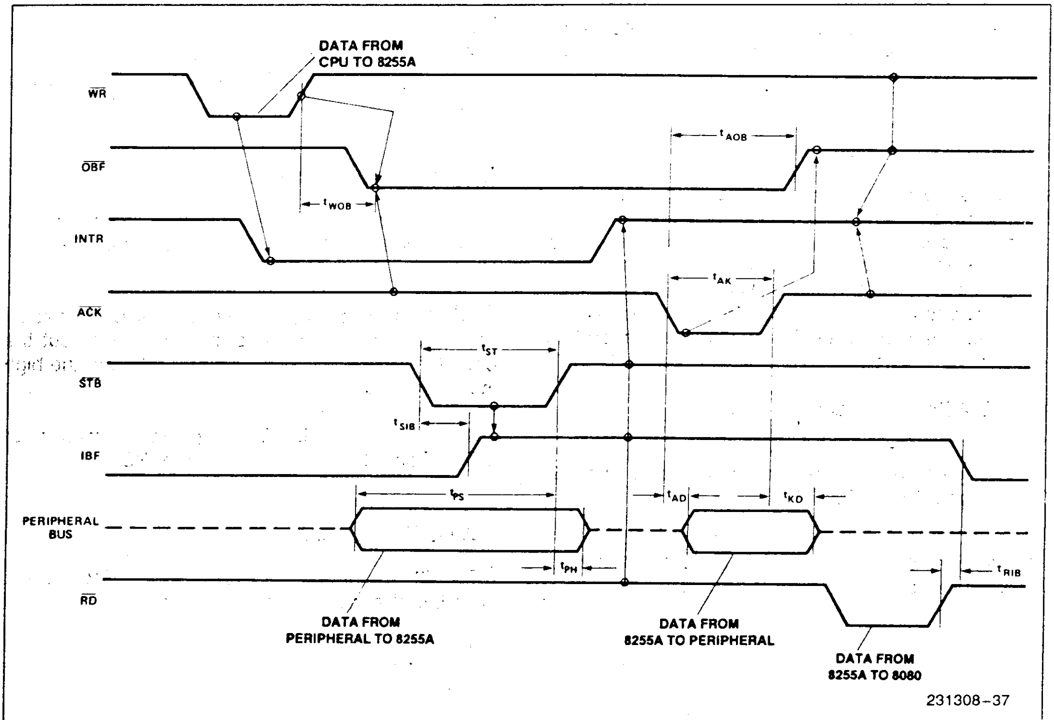


Figure 14. MODE 2



NOTE:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

Figure 15. MODE 2 (Bidirectional)

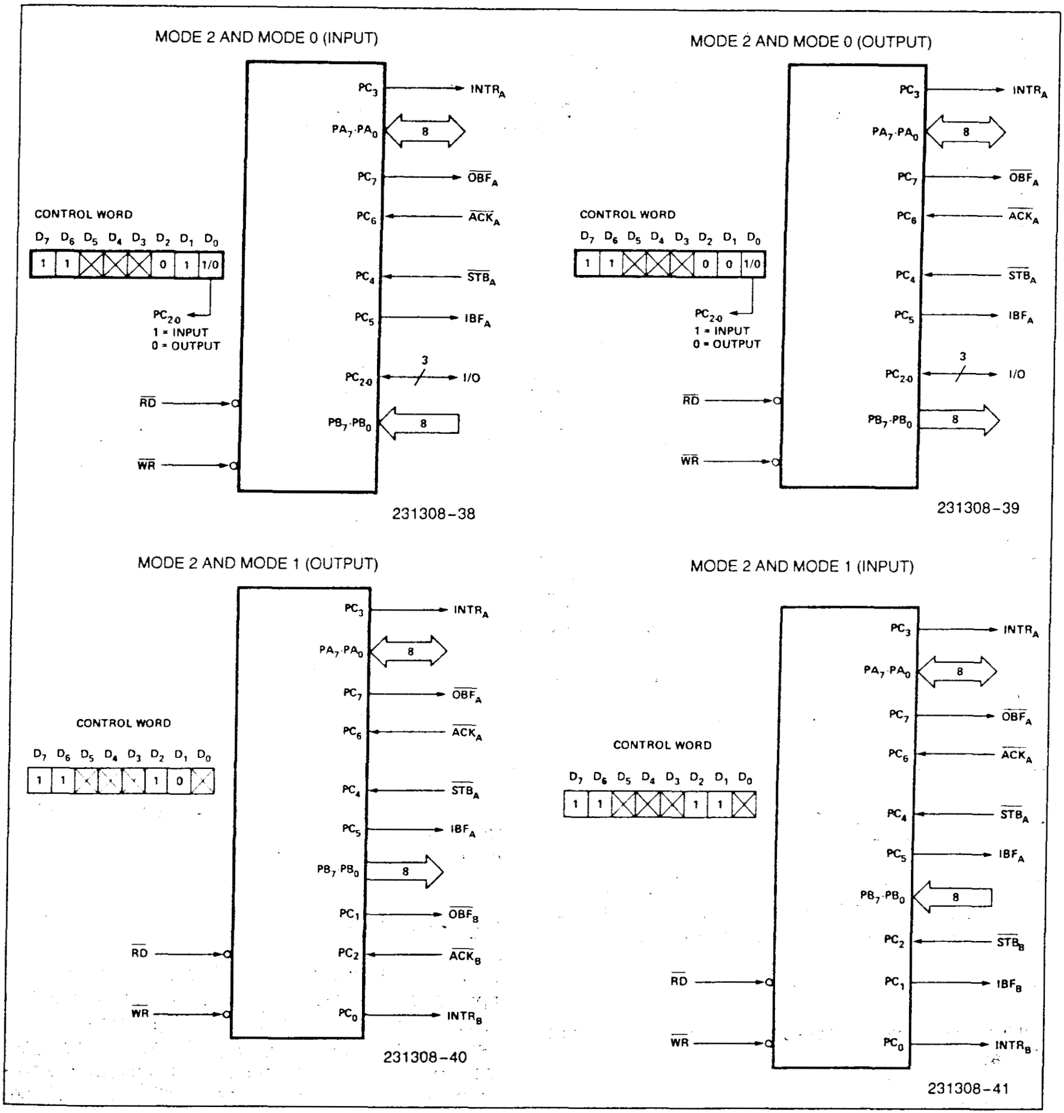


Figure 16. MODE 1/4 Combinations

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	⟷	
PA_1	IN	OUT	IN	OUT	⟷	
PA_2	IN	OUT	IN	OUT	⟷	
PA_3	IN	OUT	IN	OUT	⟷	
PA_4	IN	OUT	IN	OUT	⟷	
PA_5	IN	OUT	IN	OUT	⟷	
PA_6	IN	OUT	IN	OUT	⟷	
PA_7	IN	OUT	IN	OUT	⟷	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs—

All input lines can be accessed during a normal Port C read.

If Programmed as Outputs—

Bits in C upper (PC_7–PC_4) must be individually accessed using the bit set/reset function.

Bits in C lower (PC_3–PC_0) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

Source Current Capability on Port B and Port C

Any set of **eight** output buffers, selected randomly from Ports B and C can source 1 mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

INPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
GROUP A					GROUP B		

OUTPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
$\overline{OBF_A}$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	$\overline{OBF_B}$	$INTR_B$
GROUP A					GROUP B		

231308-59

Figure 17. MODE 1 Status Word Format

D7	D6	D5	D4	D3	D2	D1	D0
$\overline{OBF_A}$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$	☒	☒	☒
GROUP A					GROUP B		

(DEFINED BY MODE 0 OR MODE 1 SELECTION)

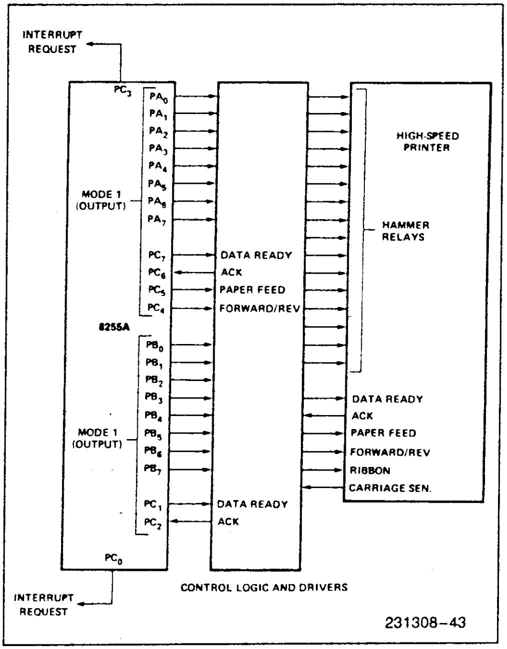
231308-42

Figure 18. MODE 2 Status Word Format

APPLICATIONS OF THE 8255A

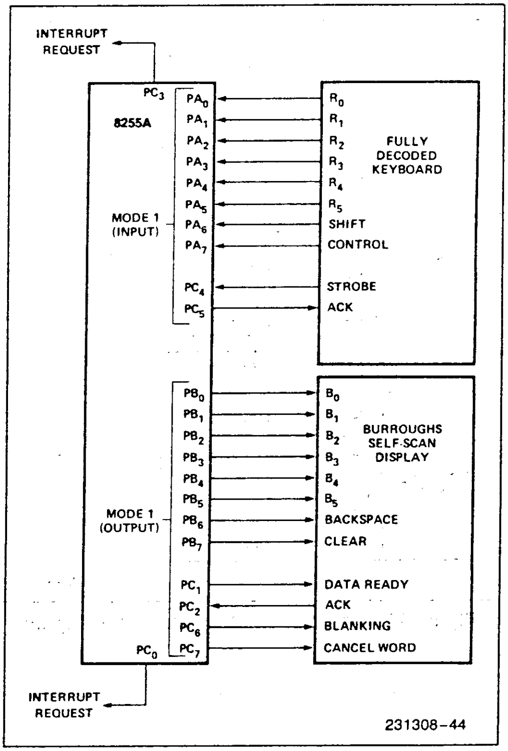
The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 19 through 25 represent a few examples of typical applications of the 8255A.



231308-43

Figure 19. Printer Interface



231308-44

Figure 20. Keyboard and Display Interface

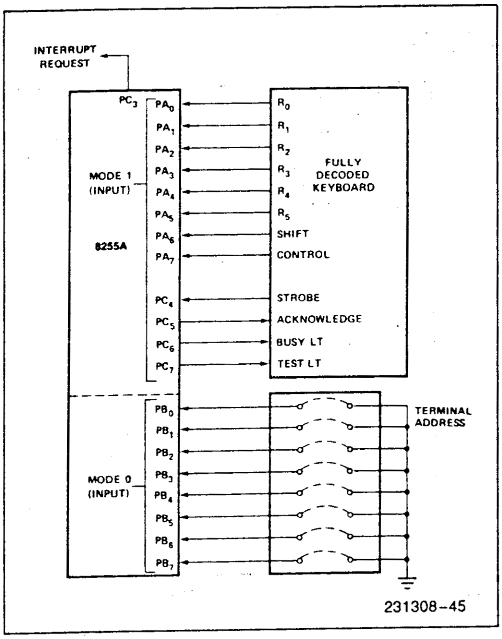

INTERRUPT REQUEST
PC_3
MODE 1 (INPUT)
8255A
PA_0 PA_1 PA_2 PA_3 PA_4 PA_5 PA_6 PA_7
PC_4 PC_5 PC_6 PC_7
R_0 R_1 R_2 R_3 R_4 R_5
SHIFT
CONTROL
FULLY DECODED KEYBOARD
STROBE
ACKNOWLEDGE
BUSY LT
TEST LT
MODE 0 (INPUT)
PB_0 PB_1 PB_2 PB_3 PB_4 PB_5 PB_6 PB_7
TERMINAL ADDRESS
231308–45

Figure 21. Keyboard and Terminal Address Interface

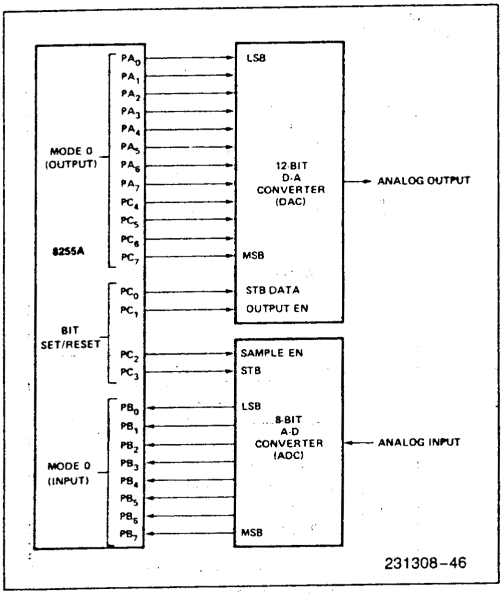

MODE 0 (OUTPUT)
8255A
PA_0 PA_1 PA_2 PA_3 PA_4 PA_5 PA_6 PA_7
PC_4 PC_5 PC_6 PC_7
LSB
12-BIT D-A CONVERTER (DAC)
MSB
ANALOG OUTPUT
BIT SET/RESET
PC_0 PC_1 PC_2 PC_3
STB DATA
OUTPUT EN
SAMPLE EN
STB
MODE 0 (INPUT)
PB_0 PB_1 PB_2 PB_3 PB_4 PB_5 PB_6 PB_7
LSB
8-BIT A-D CONVERTER (ADC)
MSB
ANALOG INPUT
231308–46

Figure 22. Digital to Analog, Analog to Digital

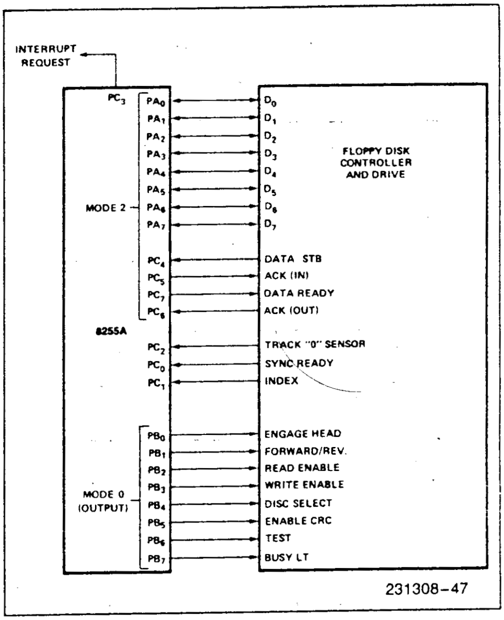


Figure 23. Basic Floppy Disk Interface

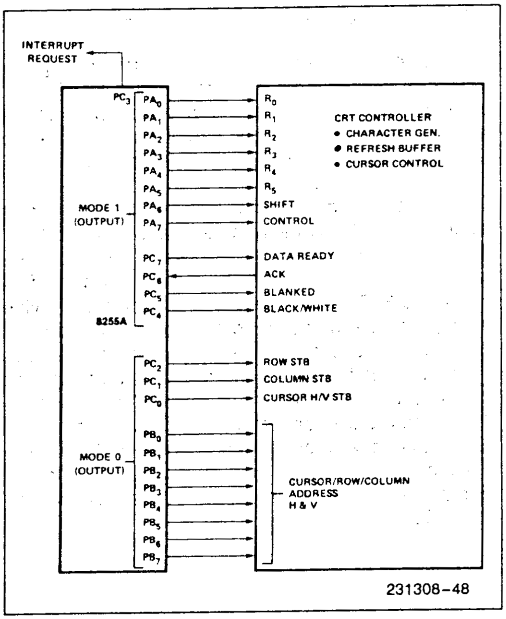


Figure 24. Basic CRT Controller Interface

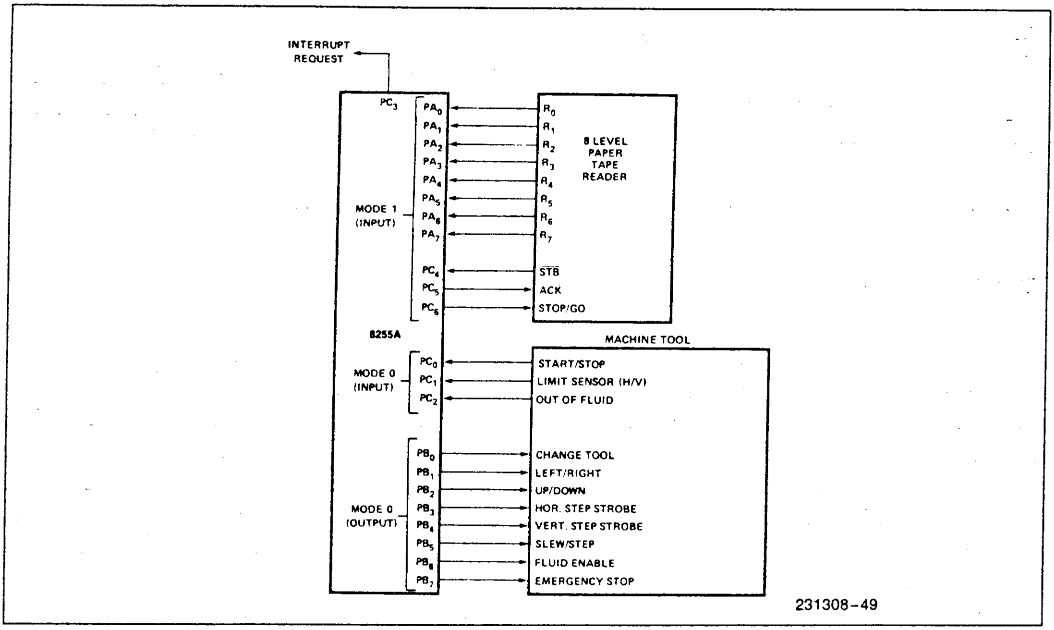


Figure 25. Machine Tool Controller Interface

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..........−0.5V 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 T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V*

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}	V	
V_{OL} (DB)	Output Low Voltage (Data Bus)		0.45*	V	I_{OL} = 2.5 mA
V_{OL} (PER)	Output Low Voltage (Peripheral Port)		0.45*	V	I_{OL} = 1.7 mA
V_{OH} (DB)	Output High Voltage (Data Bus)	2.4		V	I_{OH} = −400 μA
V_{OH} (PER)	Output High Voltage (Peripheral Port)	2.4		V	I_{OH} = −200 μA
I_{DAR}[1]	Darlington Drive Current	−1.0	−4.0	mA	R_{EXT} = 750Ω; V_{EXT} = 1.5V
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0.45V

NOTE:
1. Available on any 8 pins from Port B and C.

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

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	$f_c = 1$ MHz(4)
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND(4)

A.C. CHARACTERISTICS $T_A = 0°C$ to $70°C, V_{CC} = +5V \pm 10\%, GND = 0V$*

Bus Parameters

READ

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{AR}	Address Stable before READ	0		0		ns
t_{RA}	Address Stable after READ	0		0		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid from READ(1)		250		200	ns
t_{DF}	Data Float after READ	10	150	10	100	ns
t_{RV}	Time between READs and/or WRITEs	850		850		ns

WRITE

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{AW}	Address Stable before WRITE	0		0		ns
t_{WA}	Address Stable after WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid after WRITE	30		30		ns

OTHER TIMINGS

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{WB}	WR = 1 to Output(1)		350		350	ns
t_{IR}	Peripheral Data before RD	0		0		ns
t_{HR}	Peripheral Data after RD	0		0		ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data before T.E. of STB	0		0		ns
t_{PH}	Per. Data after T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output(1)		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns

A.C. CHARACTERISTICS (Continued)

OTHER TIMINGS (Continued)

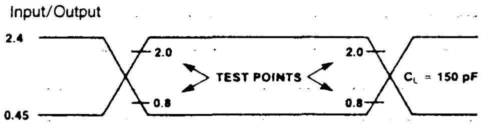
Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{WOB}	WR = 1 to OBF = 0(1)		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1(1)		350		350	ns
t_{SIB}	STB = 0 to IBF = 1(1)		300		300	ns
t_{RIB}	RD = 1 to IBF = 0(1)		300		300	ns
t_{RIT}	RD = 0 to INTR = 0(1)		400		400	ns
t_{SIT}	STB = 1 to INTR = 1(1)		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1(1)		350		350	ns
t_{WIT}	WR = 0 to INTR = 0(1, 3)		850		850	ns

NOTES:

1. Test Conditions: C_L = 150 pF.
2. Period of Reset pulse must be at least 50 μs during or after power on. Subsequent Reset pulse can be 500 ns min.
3. INTR ↑ may occur as early as $\overline{WR}$ ↓.
4. Sampled, not 100% tested.

*For Extended Temperature EXPRESS, use M8255A electrical parameters.

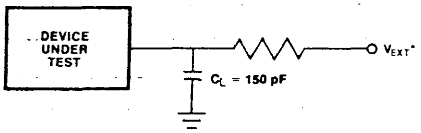
A.C. TESTING INPUT, OUTPUT WAVEFORM



231308-50

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT

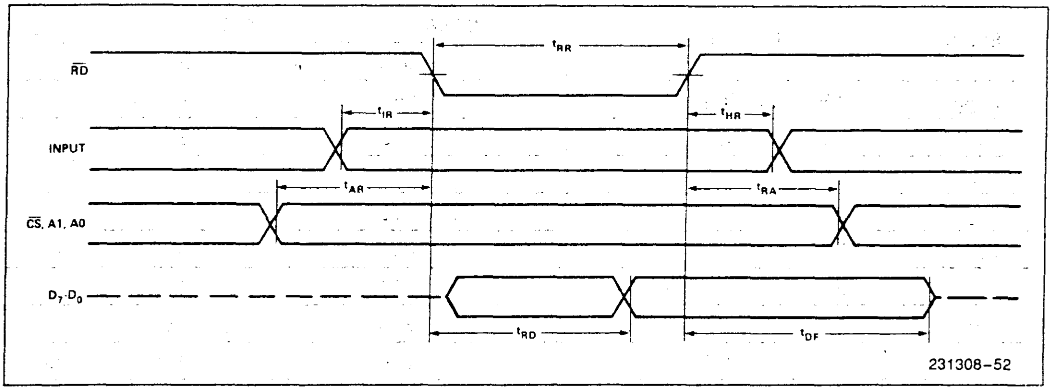


231308-51

*V_{EXT} is set at various voltages during testing to guarantee the specification. C_L includes jig capacitance.

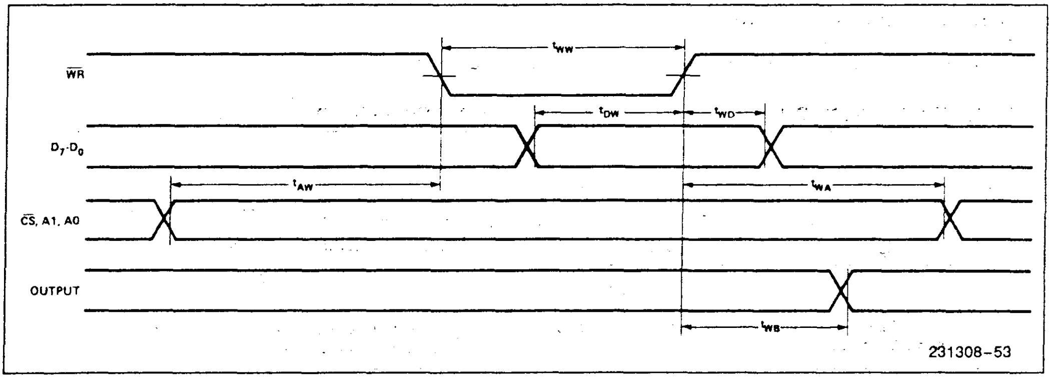
WAVEFORMS

MODE 0 (BASIC INPUT)



231308-52

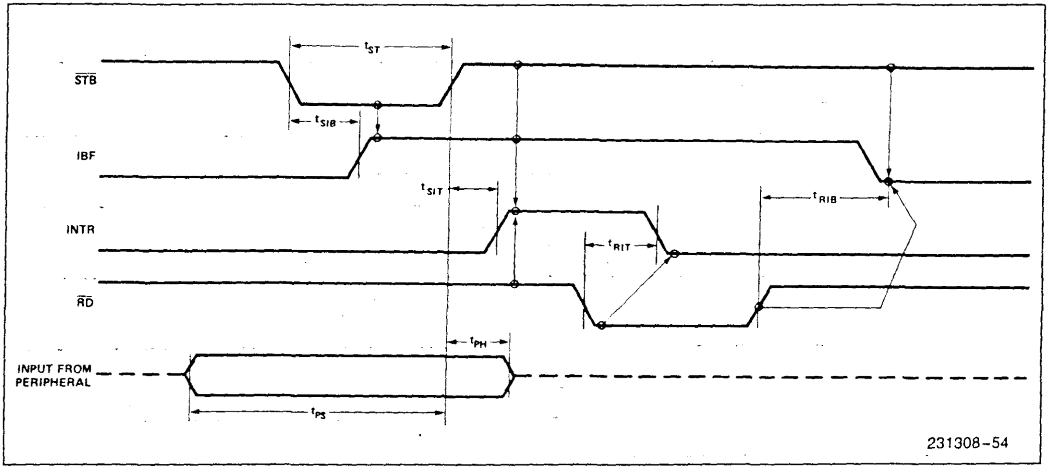
MODE 0 (BASIC OUTPUT)



231308-53

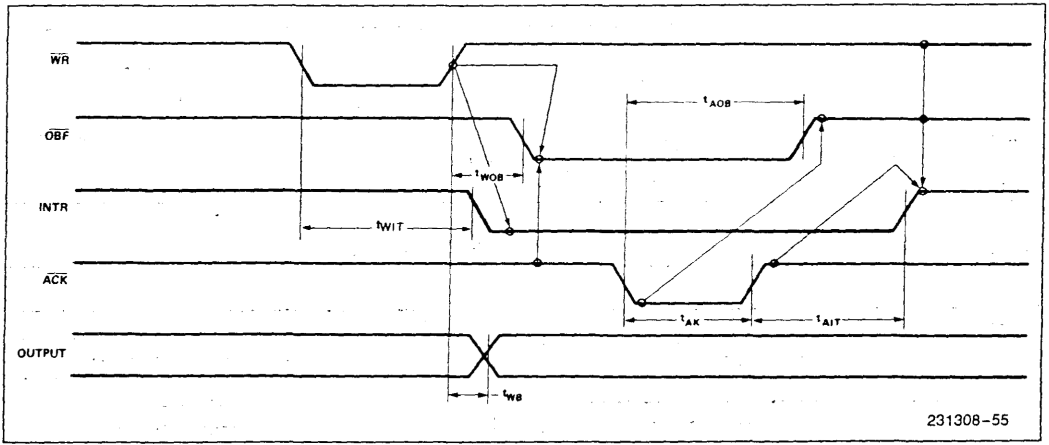
WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)



231308-54

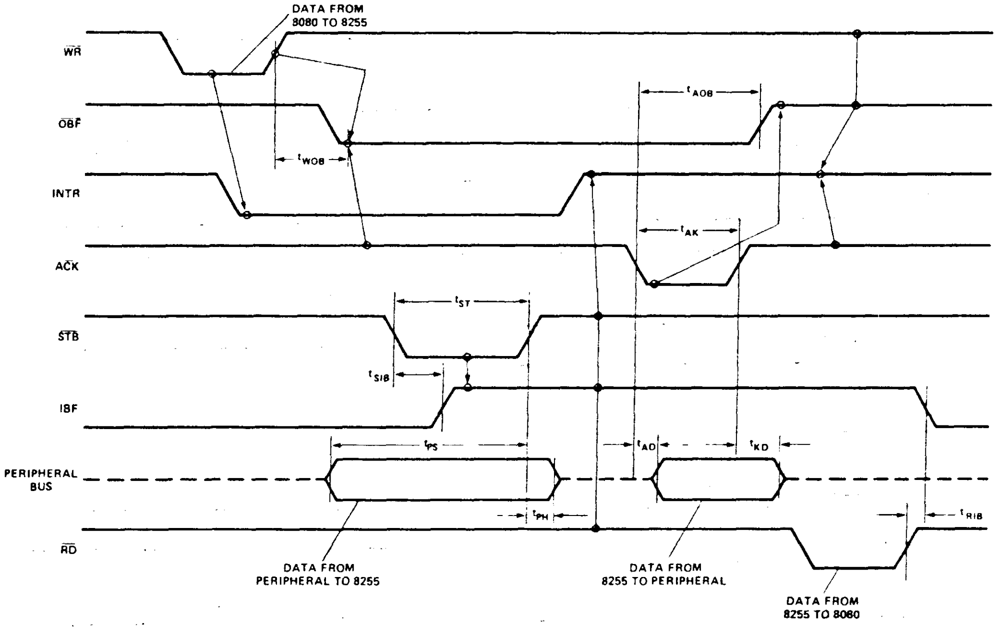
MODE 1 (STROBED OUTPUT)



231308-55

WAVEFORMS (Continued)

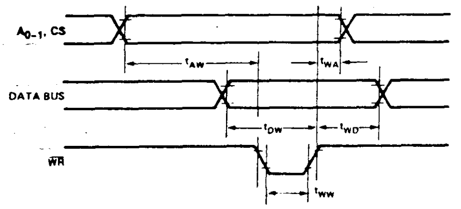
MODE 2 (BIDIRECTIONAL)



231308–56

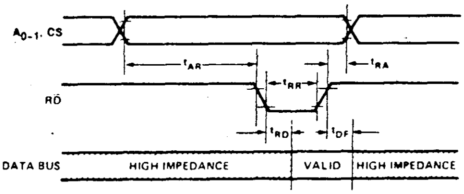
NOTE:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

WRITE TIMING



231308–57

READ TIMING



231308–58

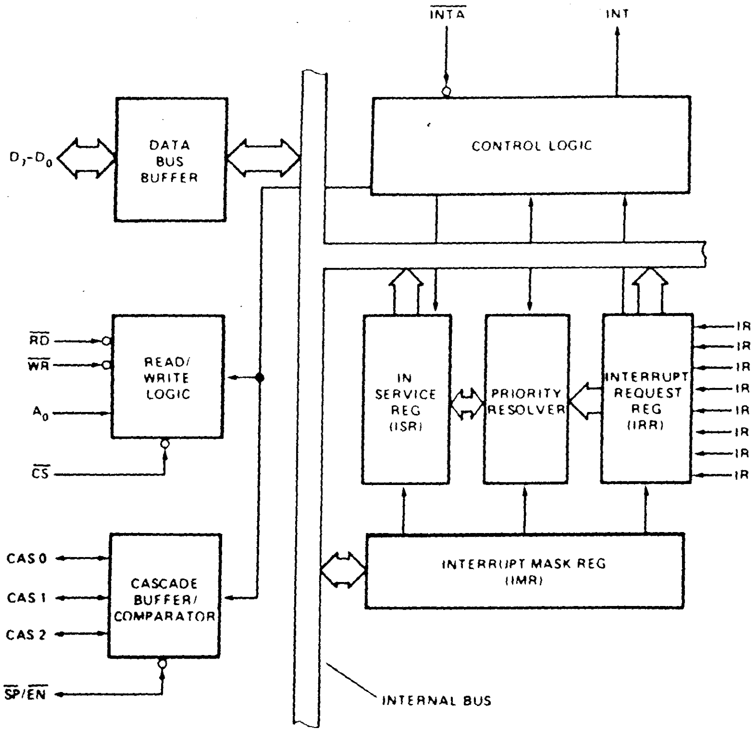
8259A
PROGRAMMABLE INTERRUPT CONTROLLER (8259A/8259A-2)

- **8086, 8088 Compatible**
- **MCS-80®, MCS-85® Compatible**
- **Eight-Level Priority Controller**
- **Expandable to 64 Levels**
- **Programmable Interrupt Modes**
- **Individual Request Mask Capability**
- **Single +5V Supply (No Clocks)**
- **Available in 28-Pin DIP and 28-Lead PLCC Package**
  (See Packaging Spec., Order #231369)
- **Available in EXPRESS**
  —Standard Temperature Range
  —Extended Temperature Range

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

The 8259A 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.

The 8259A is fully upward compatible with the Intel 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes (MCS-80/85, Non-Buffered, Edge Triggered).



231468-1

Figure 1. Block Diagram

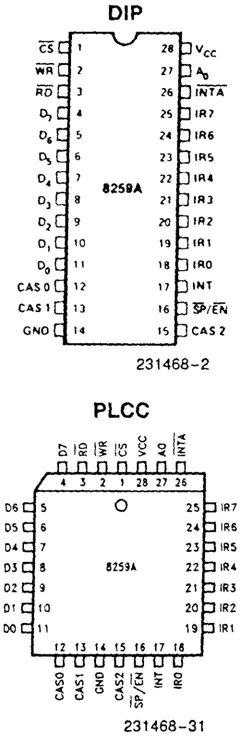


Figure 2. Pin Configurations

October 1988

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
V_{CC}	28	I	**SUPPLY:** + 5V Supply.
GND	14	I	**GROUND**
$\overline{CS}$	1	I	**CHIP SELECT:** A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the 8259A. INTA functions are independent of CS.
$\overline{WR}$	2	I	**WRITE:** A low on this pin when CS is low enables the 8259A to accept command words from the CPU.
$\overline{RD}$	3	I	**READ:** A low on this pin when CS is low enables the 8259A to release status onto the data bus for the CPU.
D_7-D_0	4-11	I/O	**BIDIRECTIONAL DATA BUS:** Control, status and interrupt-vector information is transferred via this bus.
CAS_0-CAS_2	12, 13, 15	I/O	**CASCADE LINES:** The CAS lines form a private 8259A bus to control a multiple 8259A structure. These pins are outputs for a master 8259A and inputs for a slave 8259A.
$\overline{SP}/\overline{EN}$	16	I/O	**SLAVE PROGRAM/ENABLE BUFFER:** This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP = 1) or slave (SP = 0).
INT	17	O	**INTERRUPT:** This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR_0-IR_7	18-25	I	**INTERRUPT REQUESTS:** Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
$\overline{INTA}$	26	I	**INTERRUPT ACKNOWLEDGE:** This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A_0	27	I	**AO ADDRESS LINE:** This pin acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. It is used by the 8259A to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A0 address line (A1 for 8086, 8088).

FUNCTIONAL DESCRIPTION

Interrupts in Microcomputer Systems

Microcomputer system design requires that I.O devices such as keyboards, displays, sensors and other components receive servicing in a an efficient manner 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 continuous 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 desirable 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. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

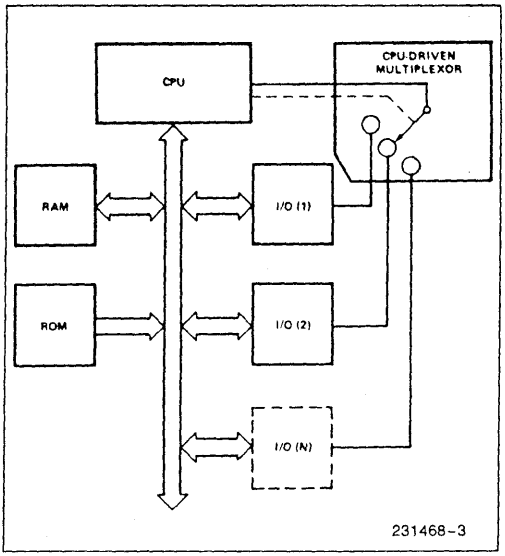


Figure 3a. Polled Method

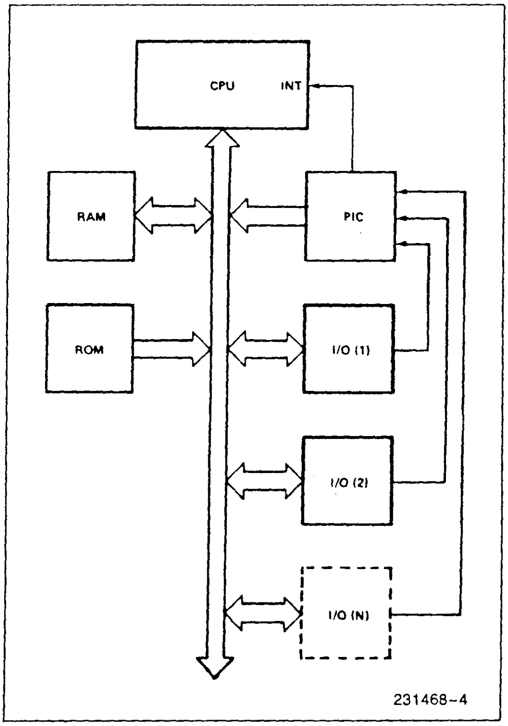


Figure 3b. Interrupt Method

The 8259A 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 8259A's (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 8259A 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 (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 priorites of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{\text{INTA}}$ pulse.

INTERRUPT MASK REGISTER (IMR)

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

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 8080A, 8085A and 8086 input levels.

$\overline{\text{INTA}}$ (INTERRUPT ACKNOWLEDGE)

$\overline{\text{INTA}}$ pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the 8259A.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A 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 8259A to be transferred onto the Data Bus.

$\overline{\text{CS}}$ (CHIP SELECT)

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

$\overline{\text{WR}}$ (WRITE)

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

$\overline{\text{RD}}$ (READ)

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

A_0

This input signal is used in conjunction with $\overline{\text{WR}}$ and $\overline{\text{RD}}$ 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.

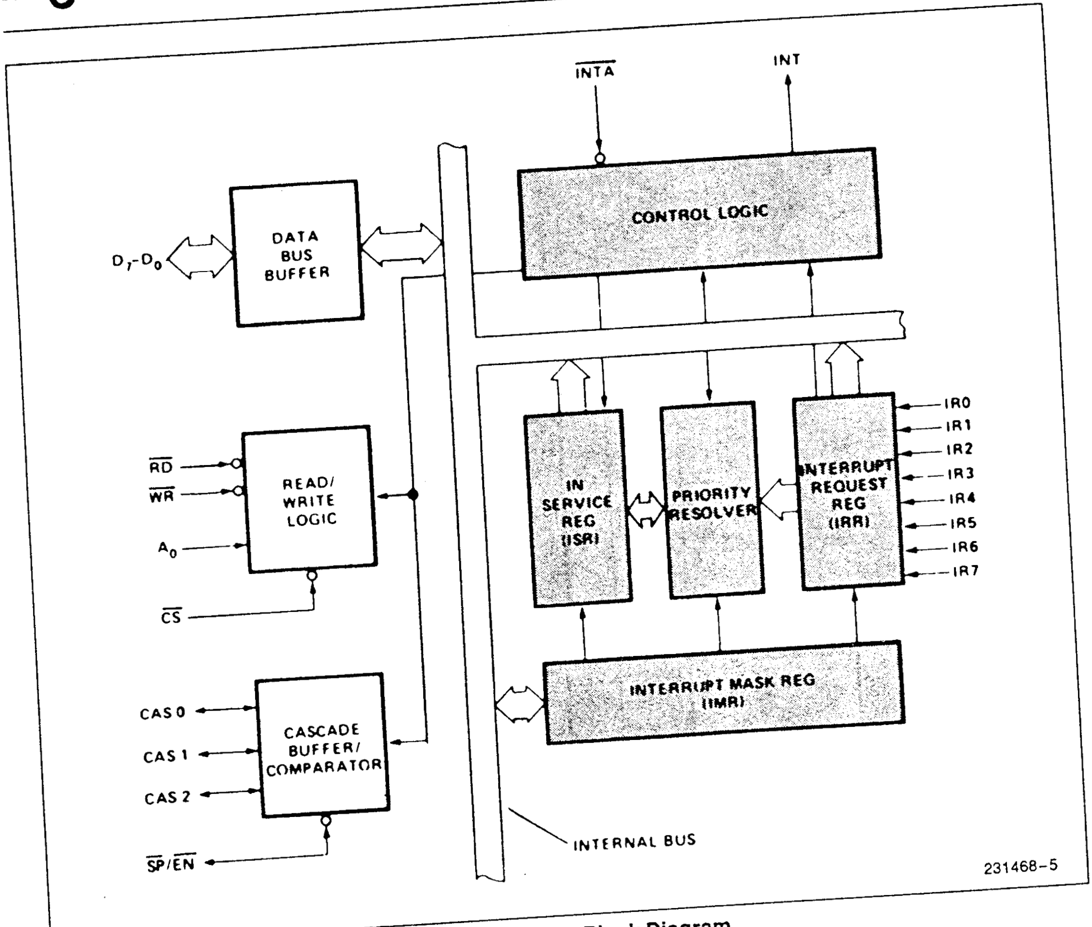


Figure 4a. 8259A Block Diagram

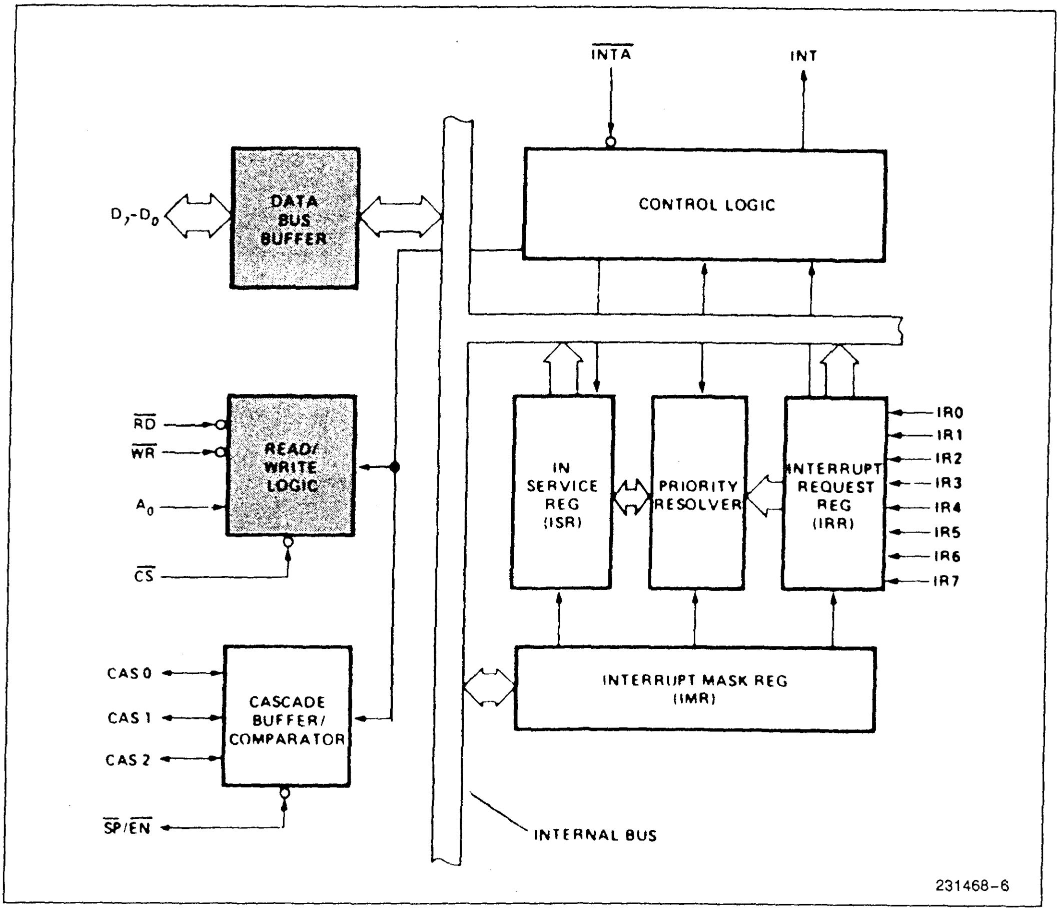


Figure 4b. 8259A Block Diagram

THE CASCADE BUFFER/COMPARATOR

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

INTERRUPT SEQUENCE

The powerful features of the 8259A in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST lines (IR7–0) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{\text{INTA}}$ pulse.
4. Upon receiving an $\overline{\text{INTA}}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259A 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 $\overline{\text{INTA}}$ pulses to be sent to the 8259A from the CPU group.
6. These two $\overline{\text{INTA}}$ pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{\text{INTA}}$ pulse and the higher 8-bit address is released at the second $\overline{\text{INTA}}$ pulse.
7. This completes the 3-byte CALL instruction released by the 8259A. In the AEOI mode the ISR bit is reset at the end of the third $\overline{\text{INTA}}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

The events occuring in an 8086 system are the same until step 4.

4. Upon receiving an $\overline{\text{INTA}}$ from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.
5. The 8086 will initiate a second $\overline{\text{INTA}}$ pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second $\overline{\text{INTA}}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step 4 of either sequence (i.e., the request was too short in duration) the 8259A will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.

When the 8259A PIC receives an interrupt, INT becomes active and an interrupt acknowledge cycle is started. If a higher priority interrupt occurs between the two INTA pulses, the INT line goes inactive immediately after the second INTA pulse. After an unspecified amount of time the INT line is activated again to signify the higher priority interrupt waiting for service. This inactive time is not specified and can vary between parts. The designer should be aware of this consideration when designing a system which uses the 8259A. It is recommended that proper asynchronous design techniques be followed.

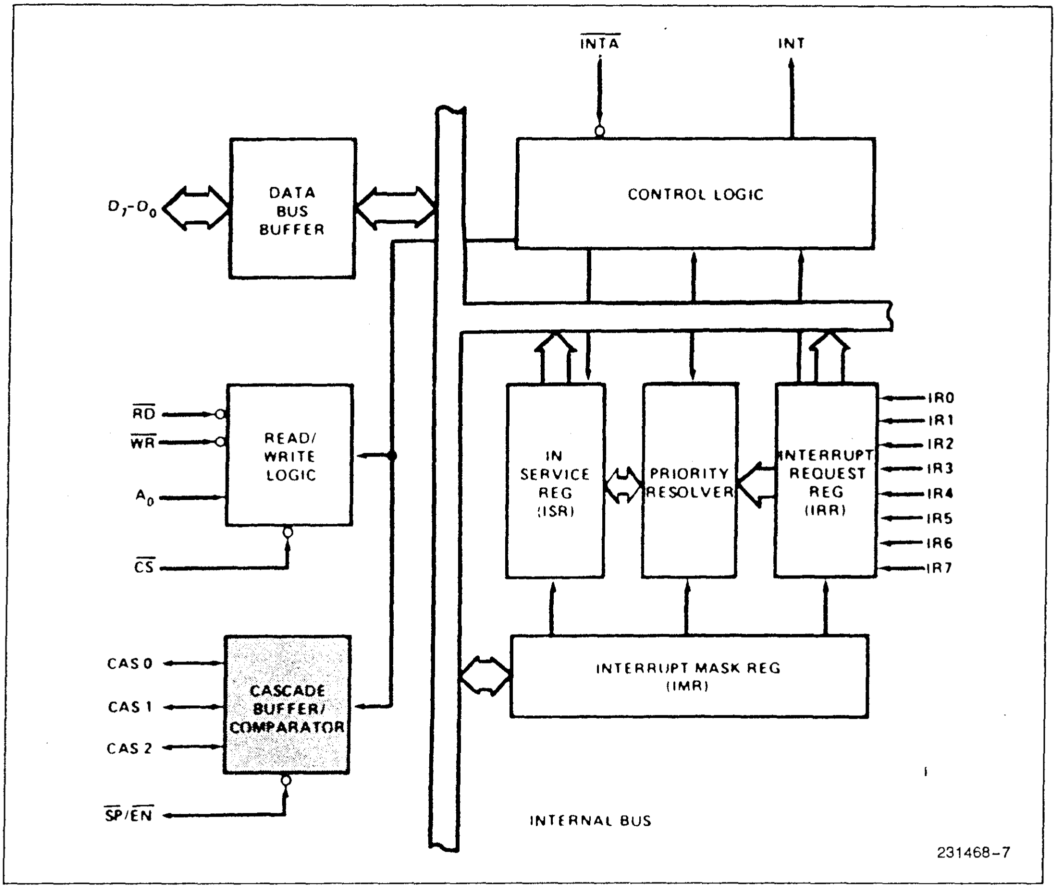


Figure 4c. 8259A Block Diagram

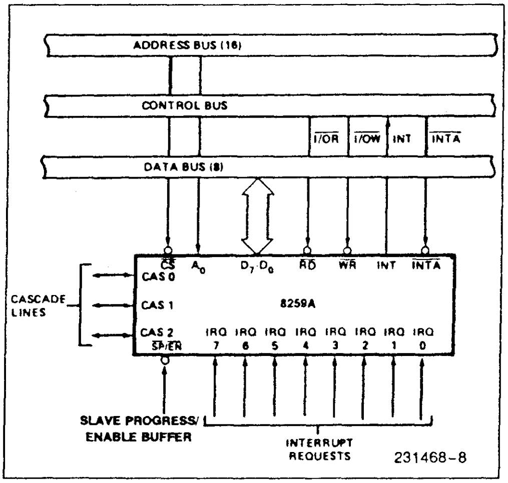


Figure 5. 8259A Interface to Standard System Bus

INTERRUPT SEQUENCE OUTPUTS

MCS-80®, MCS-85®

This sequence is timed by three $\overline{INTA}$ pulses. During the first $\overline{INTA}$ pulse the CALL opcode is enabled onto the data bus.

Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second $\overline{INTA}$ pulse the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits A_5–A_7 are programmed, while A_0–A_4 are automatically inserted by the 8259A. When Interval = 8 only A_6 and A_7 are programmed, while A_0–A_5 are automatically inserted.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third $\overline{\text{INTA}}$ pulse the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence (A_8–A_{15}), is enabled onto the bus.

Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

8086, 8088

8086 mode is similar to MCS-80 mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80, 85 systems in that the 8259A uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the $\overline{\text{INTA}}$ pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in 8086 mode the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and A_5–A_{11} are unused in 8086 mode):

Content of Interrupt Vector Byte for 8086 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

PROGRAMMING THE 8259A

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

1. *Initialization Command Words (ICWs):* Before normal operation can begin, each 8259A in the system must be brought to a starting point—by a sequence of 2 to 4 bytes timed by $\overline{\text{WR}}$ pulses.
2. *Operation Command Words (OCWs):* These are the command words which command the 8259A 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 8259A anytime after initialization.

INITIALIZATION COMMAND WORDS (ICWS)

General

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the intiitalization sequence during which the following automatically occur.

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low-to-high transistion to generate an interrupt.

b. The Interrupt Mask Register is cleared.

c. IR7 input is assigned priority 7.

d. The slave mode address is set to 7.

e. Special Mask Mode is cleared and Status Read is set to IRR.

f. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80, 85 system).

***NOTE:**
Master/Slave in ICW4 is only used in the buffered mode.

Initialization Command Words 1 and 2 (ICW1, ICW2)

A_5–A_{15}: *Page starting address of service routines.* In an MCS 80/85 system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long (A_0–A_{15}). When the routine interval is 4, A_0–A_4 are automatically inserted by the 8259A, while A_5–A_{15} are programmed externally. When the routine interval is 8, A_0–A_5 are automatically inserted by the 8259A, while A_6–A_{15} are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an 8086 system A_{15}–A_{11} are inserted in the five most significant bits of the vectoring byte and the 8259A sets the three least significant bits according to the interrupt level. A_{10}–A_5 are ignored and ADI (Address interval) has no effect.

LTIM: If LTIM = 1, then the 8259A will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. ADI = 1 then interval = 4; ADI = 0 then interval = 8.

SNGL: Single. Means that this is the only 8259A in the system. If SNGL = 1 no ICW3 will be issued.

IC4: If this bit is set—ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

Initialization Command Word 3 (ICW3)

This word is read only when there is more than one 8259A in the system and cascading is used, in which case SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

a. In the master mode (either when SP = 1, or in buffered mode when M/S = 1 in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS-80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for 8086 only byte 2) through the cascade lines.

b. In the slave mode (either when $\overline{SP}$ = 0, or if BUF = 1 and M/S = 0 in ICW4) bits 2–0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for 8086) are released by it on the Data Bus.

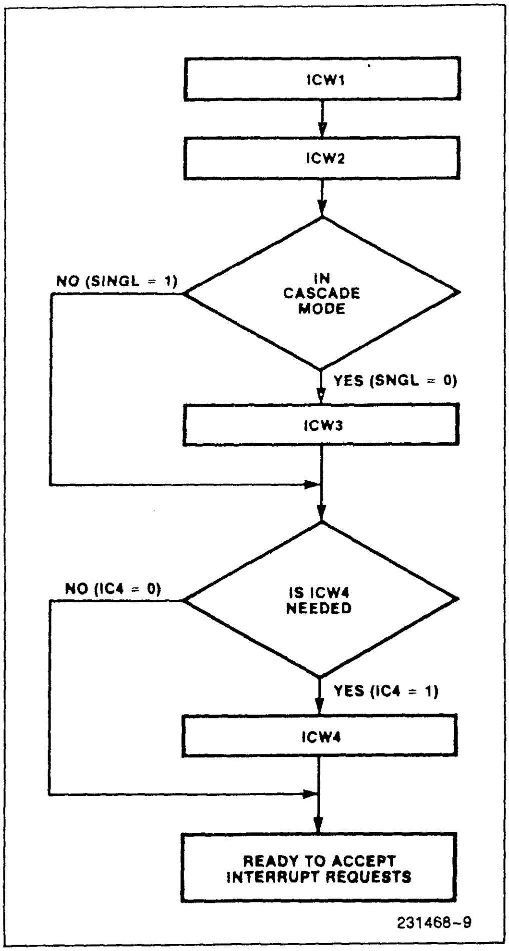


Figure 6. Initialization Sequence

Initialization Command Word 4 (ICW4)

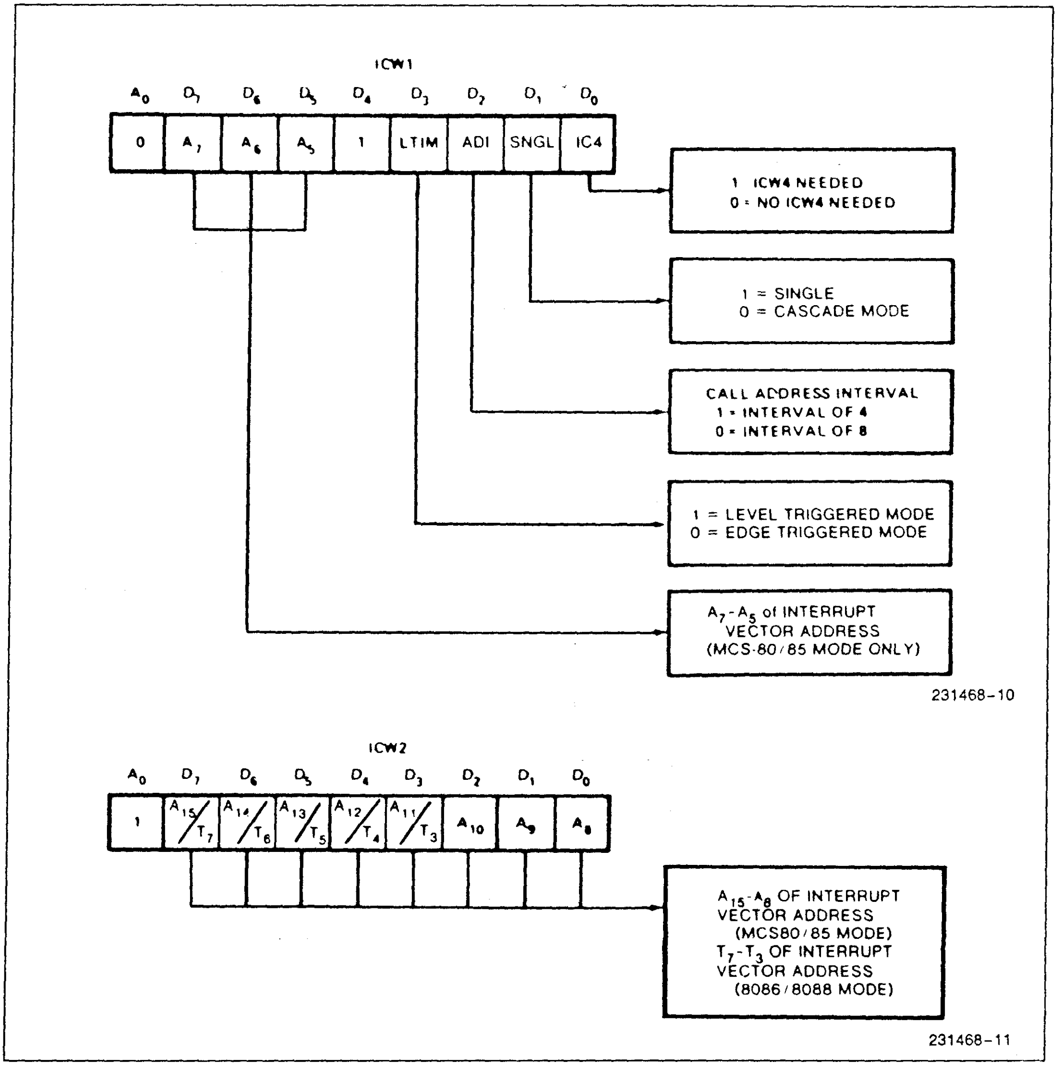
SFNM: If SFNM = 1 the special fully nested mode is programmed.

BUF: If BUF = 1 the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S.

M/S: If buffered mode is selected: M/S = 1 means the 8259A is programmed to be a master, M/S = 0 means the 8259A is programmed to be a slave. If BUF = 0, M/S has no function.

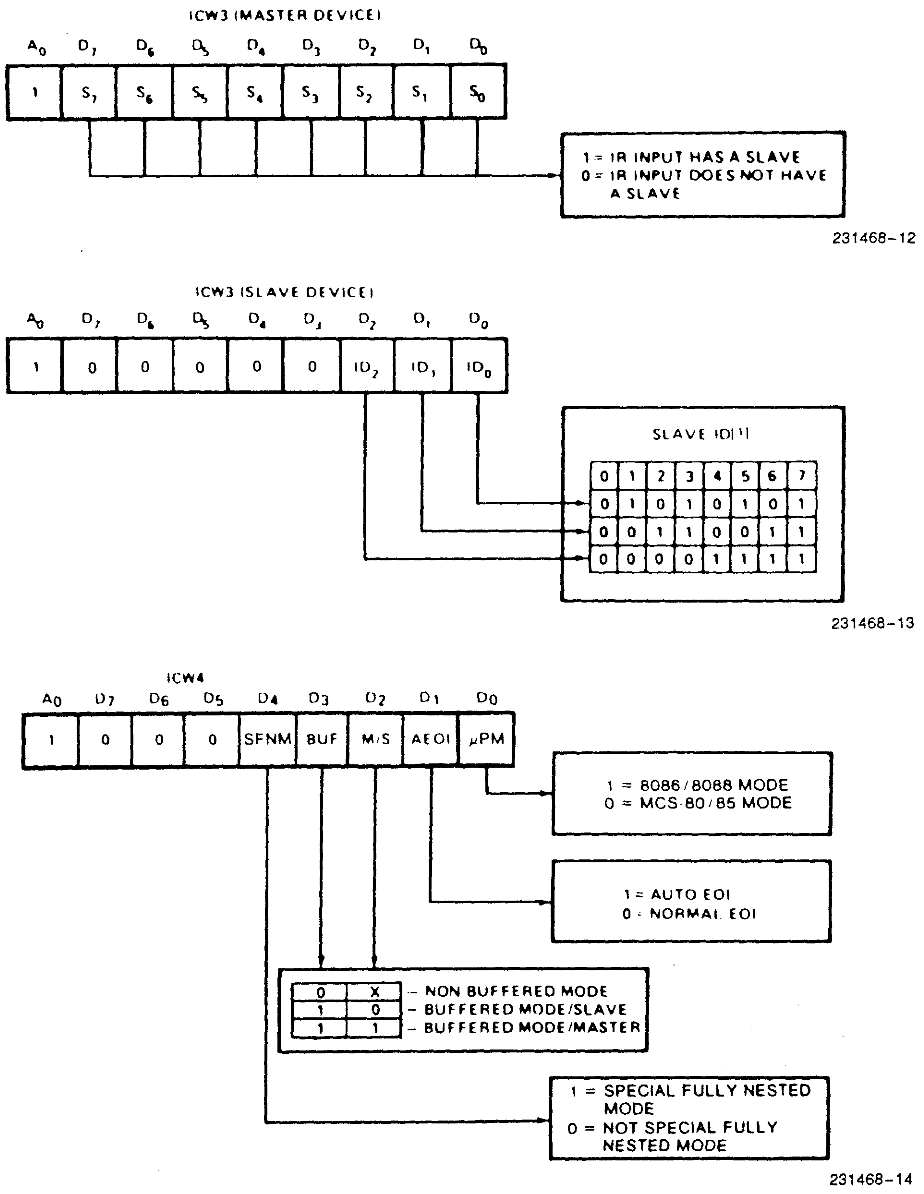
AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 8259A for MCS-80, 85 system operation, μPM = 1 sets the 8259A for 8086 system operation.



ICW1
A_0 D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
0 A_7 A_6 A_5 1 LTIM ADI SNGL IC4
1 ICW4 NEEDED
0 = NO ICW4 NEEDED
1 = SINGLE
0 = CASCADE MODE
CALL ADDRESS INTERVAL
1 = INTERVAL OF 4
0 = INTERVAL OF 8
1 = LEVEL TRIGGERED MODE
0 = EDGE TRIGGERED MODE
A_7-A_5 of INTERRUPT VECTOR ADDRESS (MCS-80/85 MODE ONLY)
231468-10
ICW2
A_0 D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
1 A_{15}/T_7 A_{14}/T_6 A_{13}/T_5 A_{12}/T_4 A_{11}/T_3 A_{10} A_9 A_8
A_{15}-A_8 OF INTERRUPT VECTOR ADDRESS (MCS80/85 MODE)
T_7-T_3 OF INTERRUPT VECTOR ADDRESS (8086/8088 MODE)
231468-11

Figure 7. Initialization Command Word Format


ICW3 (MASTER DEVICE)
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 S7 S6 S5 S4 S3 S2 S1 S0
1 = IR INPUT HAS A SLAVE
0 = IR INPUT DOES NOT HAVE A SLAVE
231468-12
ICW3 (SLAVE DEVICE)
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 0 0 0 ID2 ID1 ID0
SLAVE ID(1)
231468-13
ICW4
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 0 SFNM BUF M/S AEOI μPM
1 = 8086/8088 MODE
0 = MCS-80/85 MODE
1 = AUTO EOI
0 = NORMAL EOI
1 = SPECIAL FULLY NESTED MODE
0 = NOT SPECIAL FULLY NESTED MODE
231468-14

SLAVE ID(1)

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

0	X	– NON BUFFERED MODE
1	0	– BUFFERED MODE/SLAVE
1	1	– BUFFERED MODE/MASTER

NOTE:
Slave ID is equal to the corresponding master IR input.

Figure 7. Initialization Command Word Format (Continued)

OPERATION COMMAND WORDS (OCWS)

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

Operation Control Words (OCWs)

OCW1

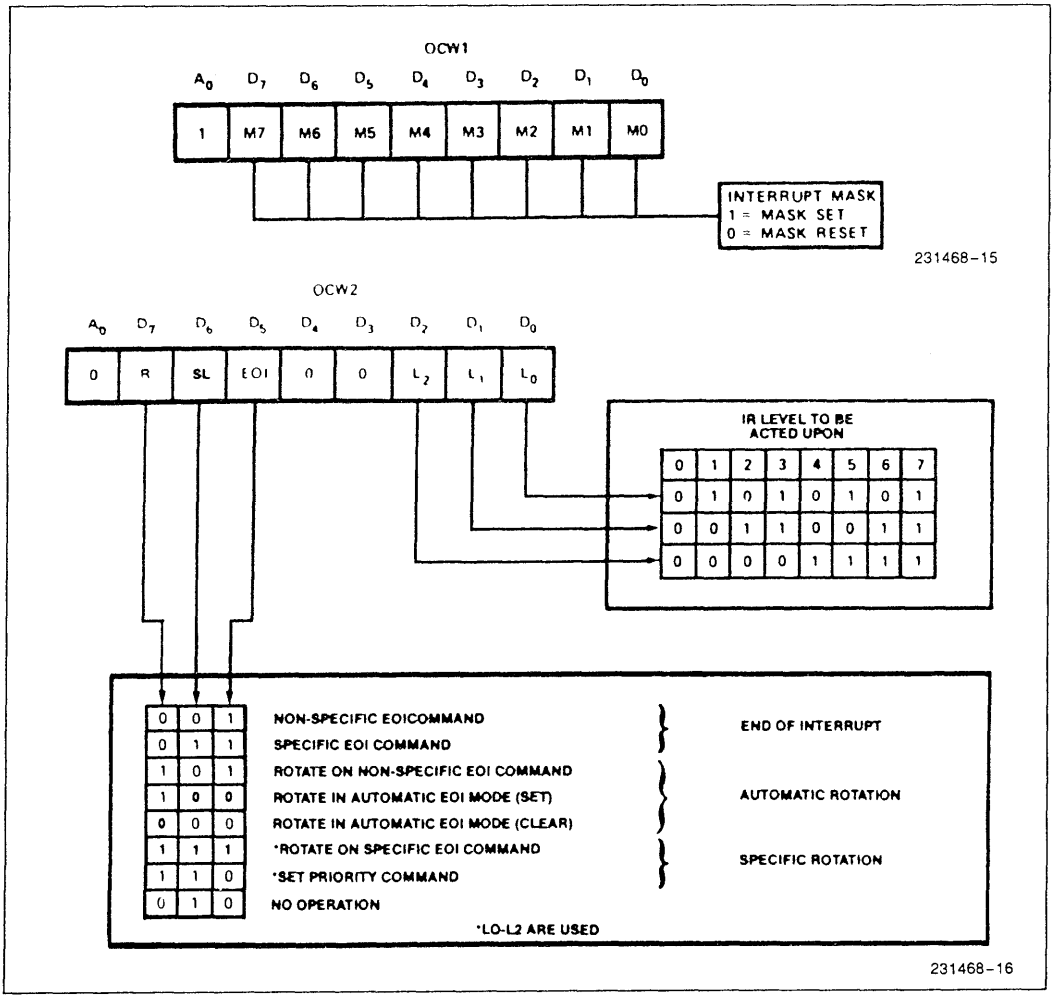
A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

OCW2

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	R	SL	EOI	0	0	L2	L1	L0

OCW3

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	ESMM	SMM	0	1	P	RR	RIS


OCW1
A_0 D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
1 M7 M6 M5 M4 M3 M2 M1 M0
INTERRUPT MASK
1 = MASK SET
0 = MASK RESET
231468-15
OCW2
A_0 D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
0 R SL EOI 0 0 L_2 L_1 L_0
IR LEVEL TO BE ACTED UPON
231468-16

IR LEVEL TO BE ACTED UPON

	0	1	2	3	4	5	6	7
L_0	0	1	0	1	0	1	0	1
L_1	0	0	1	1	0	0	1	1
L_2	0	0	0	0	1	1	1	1

R	SL	EOI		
0	0	1	NON-SPECIFIC EOICOMMAND	END OF INTERRUPT
0	1	1	SPECIFIC EOI COMMAND	
1	0	1	ROTATE ON NON-SPECIFIC EOI COMMAND	AUTOMATIC ROTATION
1	0	0	ROTATE IN AUTOMATIC EOI MODE (SET)	
0	0	0	ROTATE IN AUTOMATIC EOI MODE (CLEAR)	
1	1	1	*ROTATE ON SPECIFIC EOI COMMAND	SPECIFIC ROTATION
1	1	0	*SET PRIORITY COMMAND	
0	1	0	NO OPERATION	

*LO-L2 ARE USED

Figure 8. Operation Command Word Format

Operation Control Word 1 (OCW1)

OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). M_7–M_0 represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled.

Operation Control Word 2 (OCW2)

R, SL, EOI—These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L_2, L_1, L_0—These bits determine the interrupt level acted upon when the SL bit is active.

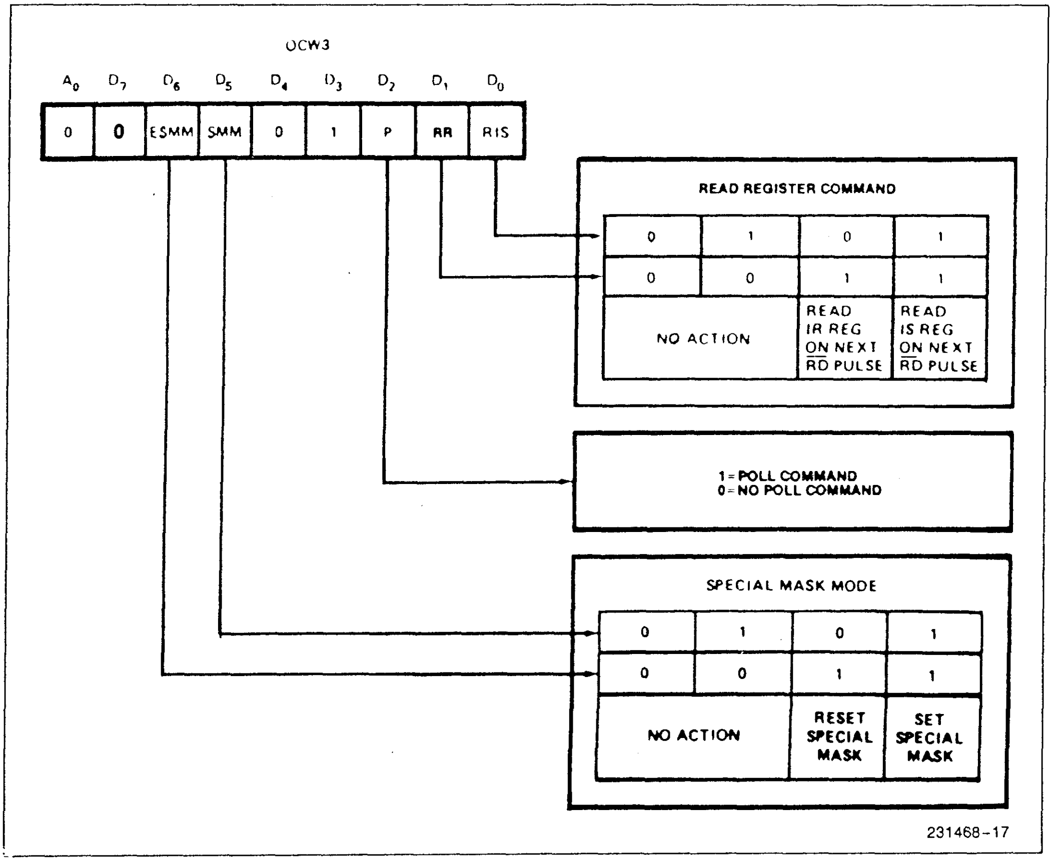


Figure 8. Operation Command Word Format (Continued)

Operation Control Word 3 (OCW3)

ESMM—Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM—Special Mask Mode. If ESMM = 1 and SMM = 1 the 8259A will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 8259A will revert to normal mask mode. When ESMM = 0, SMM has no effect.

Fully Nested Mode

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority from 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISO-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interupt enable flip-flop has been re-enabled through software).

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

End of Interrupt (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence $\overline{\text{INTA}}$ pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the 8259A before returning from a service routine (EOI command). An EOI command must be issued twice if in the Cascade mode, once for the master and once for the corresponding slave.

There are two forms of EOI command: Specific and Non-Specific. When the 8259A is operated in modes which perserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the 8259A will automatically reset the highest IS bit of those that are set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the 8259A may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0–L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 8259A is in the Special Mask Mode.

Automatic End of Interrupt (AEOI) Mode

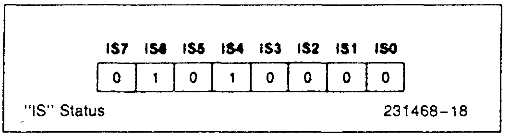
If AEOI = 1 in ICW4, then the 8259A will operate in AEOI mode continuously until reprogrammed by ICW4. in this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85, second in 8086). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 8259A.

The AEOI mode can only be used in a master 8259A and not a slave. 8259As with a copyright date of 1985 or later will operate in the AEOI mode as a master or a slave.

Automatic Rotation (Equal Priority Devices)

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most *once*. For example, if the priority and "in service" status is:

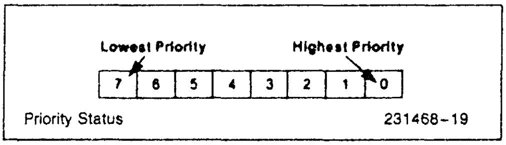
Before Rotate (IR4 the highest prioirity requiring service)



IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	1	0	0	0	0

"IS" Status

231468–18

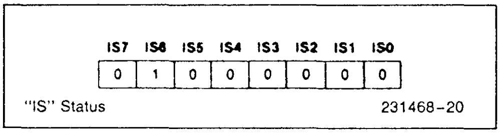


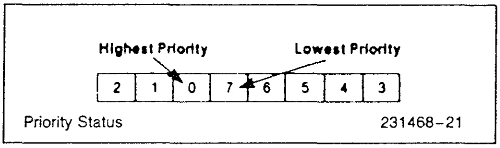
7	6	5	4	3	2	1	0

Priority Status

231468–19

After Rotate (IR4 was serviced, all other priorities rotated correspondingly)





There are two ways to accomplish Automatic Rotation using OCW2, the Rotation on Non-Specific EOI Command (R = 1, SL = 0, EOI = 1) and the Rotate in Automatic EOI Mode which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

Specific Rotation (Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Set Priority command is issued in OCW2 where: R = 1, SL = 1, L0–L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R = 1, SL = 1, EOI = 1 and LO–L2 = IR level to receive bottom priority).

Interrupt Masks

Each Interrupt Request input can bem masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 8259A would have inhibited all lower priority requests with no easy way for the routine to enable them.

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the mask register.

The special Mask Mode is set by OWC3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

Poll Command

In Poll mode the INT output functions as it normally does. The microprocessor should ignore this output. This can be accomplished either by not connecting the INT output or by masking interrupts within the microprocessor, thereby disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting P = '1" in OCW3. The 8259A treats the next $\overline{RD}$ pulse to the 8259A (i.e., $\overline{RD}$ = 0, $\overline{CS}$ = 0) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from $\overline{WR}$ to $\overline{RD}$.

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

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

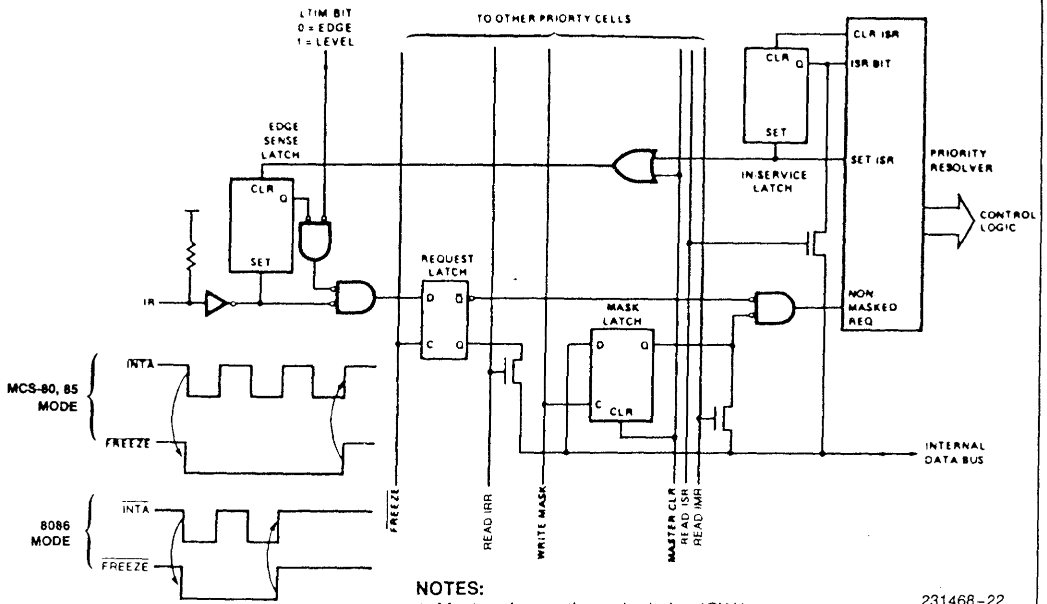
W0–W2: Binary code of the highest priority level requesting service.

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

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

Reading the 8259A Status

The input status of several internal registers can be read to update the user information on the system.



NOTES:
1. Master clear active only during ICW1.
2. FREEZE is active during INTA and poll sequences only.
3. Truth Table for a D-Latch.

C	D	Q	Operation
1	Di	Di	Follow
0	X	Qn-1	Hold

231468-22

Figure 9. Priority Cell—Simplified Logic Diagram

The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

Interrupt Request 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 RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0.)

The ISR can be read, when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 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 8259A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the 8259A is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever RD is active and A0 = 1 (OCW1).

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

Edge and Level Triggered Modes

This mode is programmed using bit 3 in ICW1.

If LTIM = '0', an interrupt request will be recognized by a low to high transition on an IR input. The IR input can remain high without generating another interrupt.

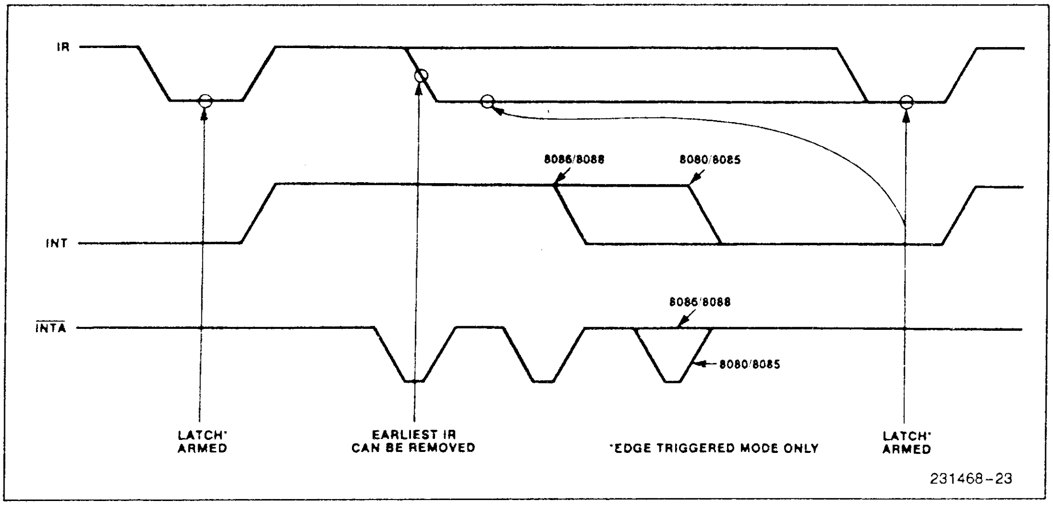


Figure 10. IR Triggering Timing Requirements

If LTIM = '1', an interrupt request will be recognized by a 'high' level on IR Input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupts is enabled to prevent a second interrupt from occurring.

The priority cell diagram shows a conceptual circuit of the level sensitive and edge sensitive input circuitry of the 8259A. Be sure to note that the request latch is a transparent D type latch.

In both the edge and level triggered modes the IR inputs must remain high until after the falling edge of the first $\overline{\text{INTA}}$. If the IR input goes low before this time a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IR inputs. To implement this feature the IR7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IR7 is needed for other purposes a default IR7 can still be detected by reading the ISR. A normal IR7 interrupt will set the corresponding ISR bit, a default IR7 won't. If a default IR7 routine occurs during a normal IR7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IR7 routine was previously entered. If another IR7 occurs it is a default.

The Special Fully Nest Mode

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a. When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

Buffered Mode

When the 8259A is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 8259A to send an enable signal on $\overline{\text{SP}}/\overline{\text{EN}}$ to enable the buffers. In this mode, whenever the 8259A's data bus outputs are enabled, the $\overline{\text{SP}}/\overline{\text{EN}}$ output becomes active.

This modification forces the use of software programming to determine whether the 8259A is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave.

CASCADE MODE

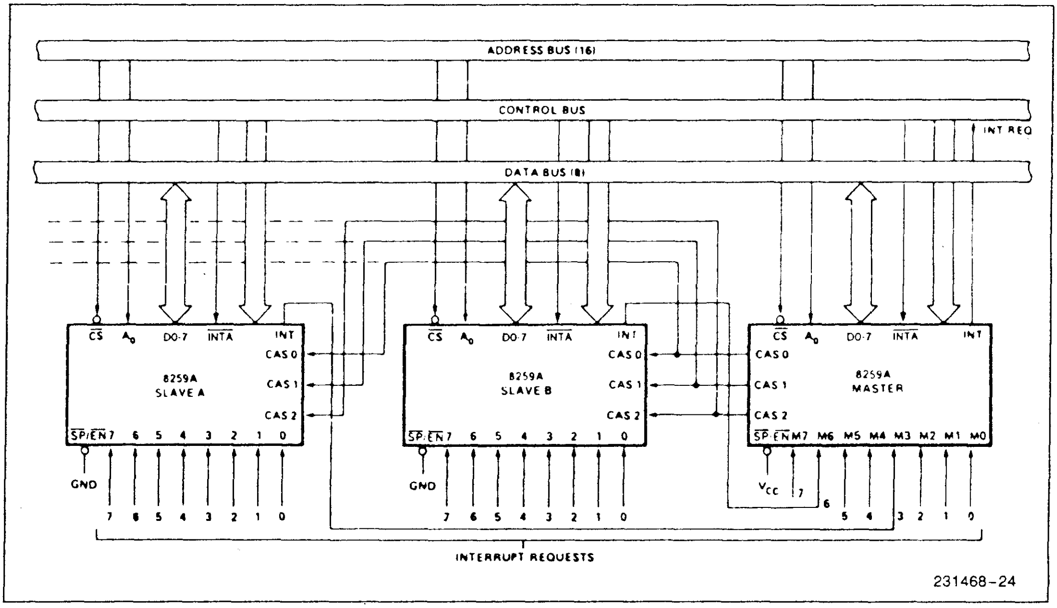
The 8259A can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

The master controls the slaves through the 3 line cascade bus. The cascade bus acts like chip selects to the slaves during the $\overline{\text{INTA}}$ sequence.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of $\overline{\text{INTA}}$. (Byte 2 only for 8086/8088).

The cascade bus lines are normally low and will contain the slave address code from the trailing edge of the first $\overline{\text{INTA}}$ pulse to the trailing edge of the third pulse. Each 8259A 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 (CS) input of each 8259A.

The cascade lines of the Master 8259A are activated only for slave inputs, non-slave inputs leave the cascade line inactive (low).



231468-24

Figure 11. Cascading the 8259A

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.......... −0.5V to +7V

Power Dissipation 1W

**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 T_A = 0°C to 70°C, V_{CC} = 5V ± 10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0*	V_{CC} + 0.5V	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
$V_{OH(INT)}$	Interrupt Output High Voltage	3.5		V	I_{OH} = −100 μA
		2.4		V	I_{OH} = −400 μA
I_{LI}	Input Load Current	−10	+10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LOL}	Output Leakage Current	−10	+10	μA	0.45V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current		85	mA	
I_{LIR}	IR Input Load Current		−300	μA	V_{IN} = 0
			10	μA	V_{IN} = V_{CC}

***NOTE:**
For Extended Temperature EXPRESS V_{IH} = 2.3V.

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 ±10%

TIMING REQUIREMENTS

Symbol	Parameter	8259A		8259A-2		Units	Test Conditions
		Min	Max	Min	Max		
TAHRL	AO/$\overline{\text{CS}}$ Setup to $\overline{\text{RD}}$/$\overline{\text{INTA}}$ ↓	0		0		ns	
TRHAX	AO/$\overline{\text{CS}}$ Hold after $\overline{\text{RD}}$/$\overline{\text{INTA}}$ ↑	0		0		ns	
TRLRH	$\overline{\text{RD}}$ Pulse Width	235		160		ns	
TAHWL	AO/$\overline{\text{CS}}$ Setup to $\overline{\text{WR}}$ ↓	0		0		ns	
TWHAX	AO/$\overline{\text{CS}}$ Hold after $\overline{\text{WR}}$ ↑	0		0		ns	
TWLWH	$\overline{\text{WR}}$ Pulse Width	290		190		ns	
TDVWH	Data Setup to $\overline{\text{WR}}$ ↑	240		160		ns	
TWHDX	Data Hold after $\overline{\text{WR}}$ ↑	0		0		ns	
TJLJH	Interrupt Request Width (Low)	100		100		ns	See Note 1
TCVIAL	Cascade Setup to Second or Third $\overline{\text{INTA}}$ ↓ (Slave Only)	55		40		ns	
TRHRL	End of $\overline{\text{RD}}$ to Next $\overline{\text{RD}}$ End of $\overline{\text{INTA}}$ to Next $\overline{\text{INTA}}$ within an $\overline{\text{INTA}}$ Sequence Only	160		100		ns	
TWHWL	End of $\overline{\text{WR}}$ to Next $\overline{\text{WR}}$	190		100		ns	
*TCHCL	End of Command to Next Command (Not Same Command Type)	500		150		ns	
	End of $\overline{\text{INTA}}$ Sequence to Next $\overline{\text{INTA}}$ Sequence.	500		300			

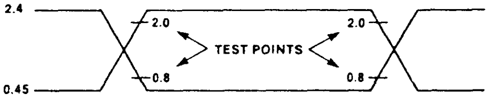
*Worst case timing for TCHCL in an actual microprocessor system is typically much greater than 500 ns (i.e. 8085A = 1.6 μs, 8085A-2 = 1 μs, 8086 = 1 μs, 8086-2 = 625 ns)

NOTE:
This is the low time required to clear the input latch in the edge triggered mode.

TIMING RESPONSES

Symbol	Parameter	8259A		8259A-2		Units	Test Conditions
		Min	Max	Min	Max		
TRLDV	Data Valid from $\overline{\text{RD}}$/$\overline{\text{INTA}}$ ↓		200		120	ns	C of Data Bus = 100 pF
TRHDZ	Data Float after $\overline{\text{RD}}$/$\overline{\text{INTA}}$ ↑	10	100	10	85	ns	C of Data Bus Max Test C = 100 pF Min Test C = 15 pF
TJHIH	Interrupt Output Delay		350		300	ns	
TIALCV	Cascade Valid from First $\overline{\text{INTA}}$ ↓ (Master Only)		565		360	ns	C_{INT} = 100 pF
TRLEL	Enable Active from $\overline{\text{RD}}$ ↓ or $\overline{\text{INTA}}$ ↓		125		100	ns	$C_{CASCADE}$ = 100 pF
TRHEH	Enable Inactive from $\overline{\text{RD}}$ ↑ or $\overline{\text{INTA}}$ ↑		150		150	ns	
TAHDV	Data Valid from Stable Address		200		200	ns	
TCVDV	Cascade Valid to Valid Data		300		200	ns	

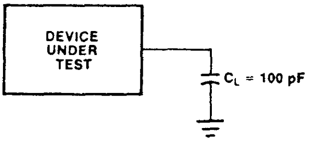
A.C. TESTING INPUT/OUTPUT WAVEFORM



231468-25

A.C. Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".

A.C. TESTING LOAD CIRCUIT

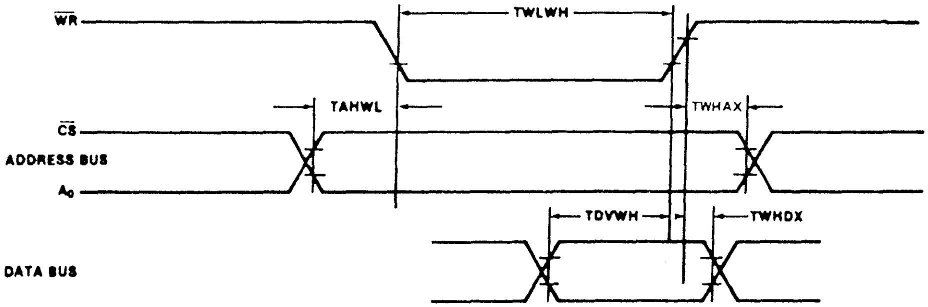


231468-26

C_L = 100 pF
C_L Includes Jig Capacitance

WAVEFORMS

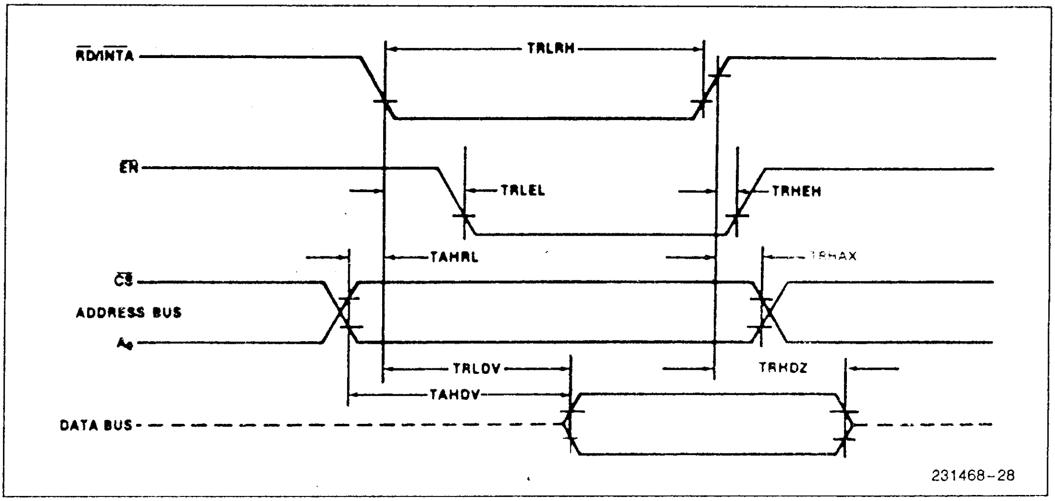
WRITE



231468-27

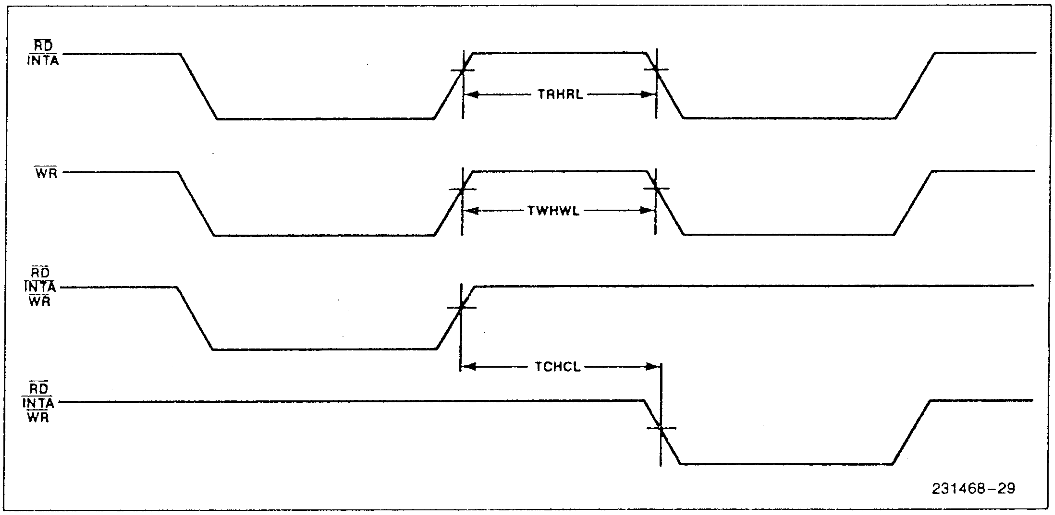
WAVEFORMS (Continued)

READ/INTA



231468-28

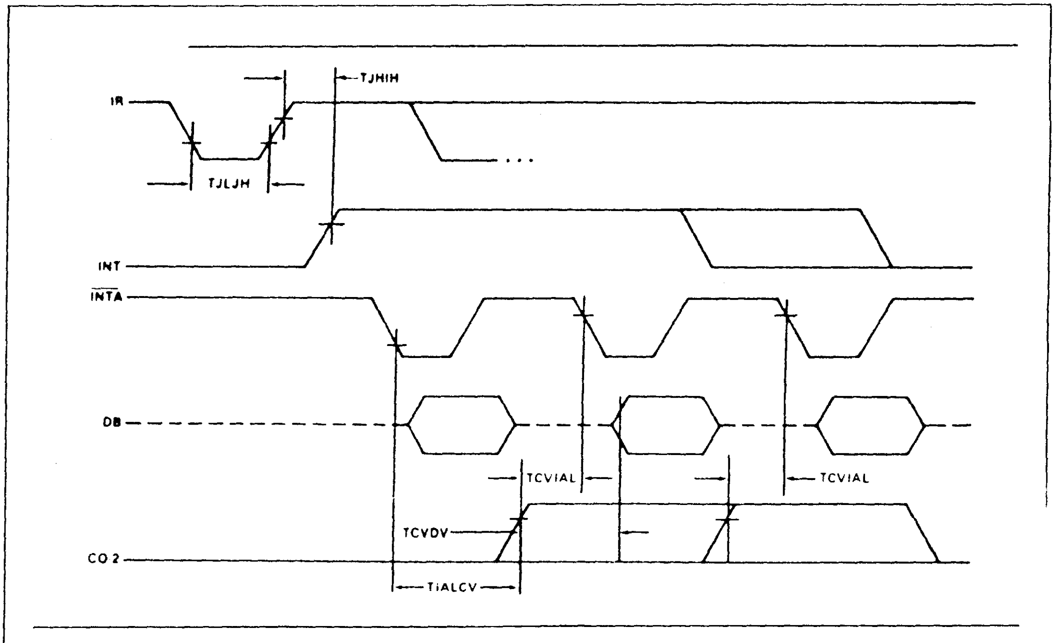
OTHER TIMING



231468-29

WAVEFORMS (Continued)

$\overline{\text{INTA}}$ SEQUENCE



231468-30

NOTES:

Interrupt output must remain HIGH at least until leading edge of first $\overline{\text{INTA}}$.

1. Cycle 1 in 8086, 8088 systems, the Data Bus is not active.

Data Sheet Revision Review

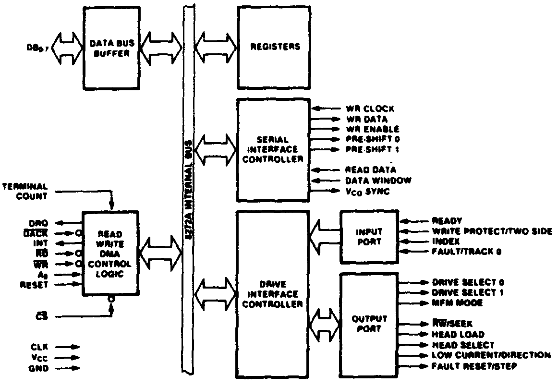
The following changes have been made since revision 2 of the 8259A data sheet.

1. The first paragraph of the Poll Command section was rewritten to clarify the status of the INT pin.
2. A paragraph was added to the Interrupt Sequence section to indicate the status of the INT pin during multiple interrupts.
3. A reference to PLCC packaging was added.
4. All references to the 8259A-8 have been deleted.

8272A
SINGLE/DOUBLE DENSITY FLOPPY DISK CONTROLLER

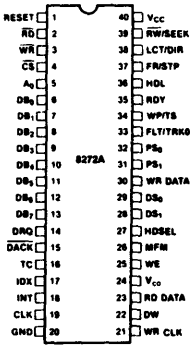
- **IBM Compatible in Both Single and Double Density Recording Formats**
- **Programmable Data Record Lengths: 128, 256, 512, or 1024 Bytes/Sector**
- **Multi-Sector and Multi-Track Transfer Capability**
- **Drives Up to 4 Floppy or Mini-Floppy Disks**
- **Controls 8", 5¼" and 3½" Floppy Disk Drives**
- **Data Transfers in DMA or Non-DMA Mode**
- **Parallel Seek Operations on Up to Four Drives**
- **Compatible with all Intel and Most Other Microprocessors**
- **Single-Phase 8 MHz Clock**
- **Single +5V Power Supply (±10%)**
- **Plastic 40 Pin DIP or 40 Pin CERDIP Packages**

The 8272A is an LSI Floppy Disk Controller (FDC) Chip, which contains the circuitry and control functions for interfacing a processor to 4 Floppy Disk Drives. It is capable of supporting either IBM 3740 single density format (FM), or IBM System 34 Double Density format (MFM) including double sided recording. The 8272A provides control signals which simplify the design of an external phase locked loop and write precompensation circuitry. The FDC simplifies and handles most of the burdens associated with implementing a Floppy Disk Drive Interface. The 8272A is a pin-compatible upgrade to the 8272.



210606-1

Figure 1. 8272A Internal Block Diagram



210606-2

Figure 2. Pin Configuration

Table 1. Pin Description

Symbol	Pin No.	Type	Connection To	Name and Function
RESET	1	I	µP	**RESET:** Places FDC in idle state. Resets output lines to FDD to "0" (low). Does not clear the last specify command.
$\overline{RD}$	2	I[1]	µP	**READ:** Control signal for transfer of data from FDC to Data Bus, when "0" (low).
$\overline{WR}$	3	I[1]	µP	**WRITE:** Control signal for transfer of data to FDC via Data Bus, when "0" (low).
$\overline{CS}$	4	I	µP	**CHIP SELECT:** IC selected when "0" (low) allowing $\overline{RD}$ and $\overline{WR}$ to be enabled.
A_0	5	I[1]	µP	**DATA/STATUS REGISTER SELECT:** Selects Data Reg (A_0 = 1) or Status Reg (A_0 = 0) contents to be sent to Data Bus.
DB_0–DB_7	6–13	I/O[1]	µP	**DATA BUS:** Bidirectional 8-Bit Data Bus.
DRQ	14	O	DMA	**DATA DMA REQUEST:** DMA Request is being made by FDC when DRQ "1".[3]
$\overline{DACK}$	15	I	DMA	**DMA ACKNOWLEDGE:** DMA cycle is active when "0" (low) and Controller is performing DMA transfer.
TC	16	I	DMA	**TERMINAL COUNT:** Indicates the termination of a DMA transfer when "1" (high)[2].
IDX	17	I	FDD	**INDEX:** Indicates the beginning of a disk track.
INT	18	O	µP	**INTERRUPT:** Interrupt Request Generated by FDC.
CLK	19	I		**CLOCK:** Single Phase 8 MHz (4 MHz for mini floppies) Squarewave Clock.
GND	20			**GROUND:** D.C. Power Return.
V_{CC}	40			**D.C. POWER:** +5V
$\overline{RW}$/SEEK	39	O	FDD	**READ WRITE/SEEK:** When "1" (high) Seek mode selected and when "0" (low) Read/Write mode selected.
LCT/DIR	38	O	FDD	**LOW CURRENT/DIRECTION:** Lowers Write current on inner tracks in Read/Write mode, determines direction head will step in Seek mode.
FR/STP	37	O	FDD	**FAULT RESET/STEP:** Resets fault FF in FDD in Read/Write mode, provides step pulses to move head to another cylinder in Seek mode.
HDL	36	O	FDD	**HEAD LOAD:** Command which causes Read/Write head in FDD to contact diskette.
RDY	35	I	FDD	**READY:** Indicates FDD is ready to send or receive data. Must be tied high (gated by the index pulse) for mini floppies which do not normally have a Ready line.
WP/TS	34	I	FDD	**WRITE PROTECT/TWO-SIDE:** Senses Write Protect status in Read/Write mode, and Two Side Media in Seek mode.
FLT/TRK0	33	I	FDD	**FAULT/TRACK 0:** Senses FDD fault condition in Read/Write mode and Track 0 condition in Seek mode.
PS_1, PS_0	31, 32	O	FDD	**PRECOMPENSATION (PRE-SHIFT):** Write precompensation status during MFM mode. Determines early, late, and normal times.
WR DATA	30	O	FDD	**WRITE DATA:** Serial clock and data bits to FDD.
DS_1, DS_0	28, 29	O	FDD	**DRIVE SELECT:** Selects FDD unit.

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Connection To	Name and Function
HDSEL	27	O	FDD	**HEAD SELECT:** Head 1 selected when "1" (high) Head 0 selected when "0" (low).
MFM	26	O	PLL	**MFM MODE:** MFM mode when "1," FM mode when "0".
WE	25	O	FDD	**WRITE ENABLE:** Enables write data into FDD.
VCO	24	O	PLL	**VCO SYNC:** Inhibits VCO in PLL when "0" (low), enables VCO when "1."
RD DATA	23	I	FDD	**READ DATA:** Read data from FDD, containing clock and data bits.
DW	22	I	PLL	**DATA WINDOW:** Generated by PLL, and used to sample data from FDD.
WR CLK	21	I		**WRITE CLOCK:** Write data rate to FDD FM = 500 kHz, MFM = 1 MHz, with a pulse width of 250 ns for both FM and MFM. Must be enabled for all operations, both Read and Write.

NOTES:
1. Disabled when $\overline{CS}$ = 1.
2. TC must be activated to terminate the Execution Phase of any command.
3. DRQ is also an input for certain test modes. It should have a 5 kΩ pull-up resistor to prevent activation.

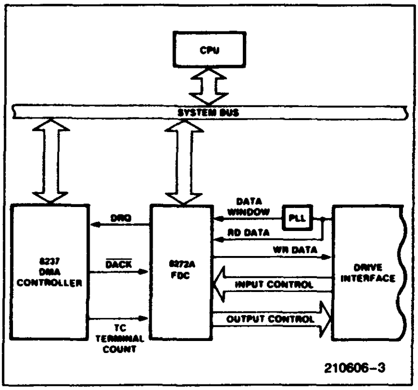


Figure 3. 8272A System Block Diagram

DESCRIPTION

Hand-shaking signals are provided in the 8272A which make DMA operation easy to incorporate with the aid of an external DMA Controller chip, such as the 8237A. The FDC will operate in either DMA or Non-DMA mode. In the Non-DMA mode, the FDC generates interrupts to the processor for every transfer of a data byte between the CPU and the 8272A. In the DMA mode, the processor need only load a command into the FDC and all data transfers occur under control of the 8272A and DMA controller.

There are 15 separate commands which the 8272A will execute. Each of these commands require multiple 8-bit bytes to fully specify the operation which the processor wishes the FDC to perform. The following commands are available.

Read Data
Read ID
Read Deleted Data
Read a Track
Scan Equal
Scan High or Equal
Scan Low or Equal
Specify

Write Data
Format a Track
Write Deleted Data
Seek
Recalibrate (Restore to Track 0)
Sense Interrupt Status
Sense Drive Status

For more information see the Intel Application Notes AP-116 and AP-121.

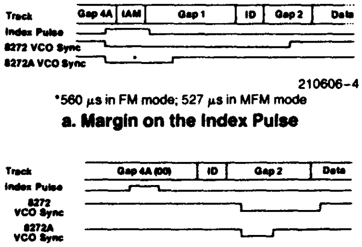
FEATURES

Address mark detection circuitry is internal to the FDC which simplifies the phase locked loop and read electronics. The track stepping rate, head load time, and head unload time may be programmed by the user. The 8272A offers many additional features such as multiple sector transfers in both read and write modes with a single command, and full IBM compatibility in both single (FM) and double density (MFM) modes.

8272A ENHANCEMENTS

On the 8272A, after detecting the Index Pulse, the VCO Sync output stays low for a shorter period of time. See Figure 4a.

On the 8272 there can be a problem reading data when Gap 4A is 00 and there is no IAM. This occurs on some older floppy formats. The 8272A cures this problem by adjusting the VCO Sync timing so that it is not low during the data field. See Figure 4b.



*560 µs in FM mode; 527 µs in MFM mode

a. Margin on the Index Pulse

b. Ability to Read Data When Gap 4A Contains 00

Figure 4. 8272A Enhancements over the 8272

8272A REGISTERS—CPU INTERFACE

The 8272A contains two registers which may be accessed by the main system processor; a Status Register and a Data Register. The 8-bit Main Status Register contains the status information of the FDC, and may be accessed at any time. The 8-bit Data Register (actually consists of several registers in a stack with only one register presented to the data bus at a time), stores data, commands, parameters, and FDD status information. Data bytes are read out of, or written into, the Data Register in order to program or obtain the results after execution of a command. The Status Register may only be read and is used to facilitate the transfer of data between the processor and 8272A.

The relationship between the Status/Down registers and the signals $\overline{RD}$, $\overline{WR}$, and A_0 is shown in Table 2.

Table 2. A_0, $\overline{RD}$, $\overline{WR}$ Decoding for the Selection of Status/Data Register Functions.

A_0	$\overline{RD}$	$\overline{WR}$	Function
0	0	1	Read Main Status Register
0	1	0	Illegal(1)
0	0	0	Illegal(1)
1	0	0	Illegal(1)
1	0	1	Read from Data Register
1	1	0	Write into Data Register

NOTE:
1. Design must guarantee that the 8272A is not subjected to illegal inputs.

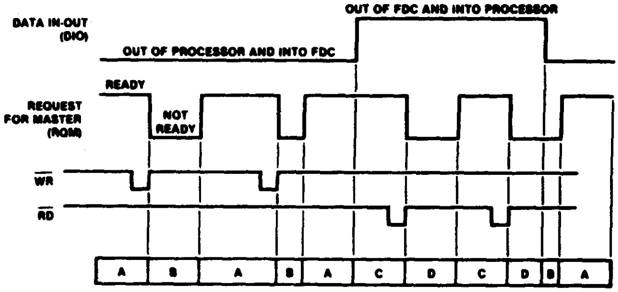
The Main Status Register bits are defined in Table 3.

Table 3. Main Status Register Bit Description

Bit Number	Name	Symbol	Description
D_0	FDD 0 Busy	D_0B	FDD number 0 is in the Seek mode.
D_1	FDD 1 Busy	D_1B	FDD number 1 is in the Seek mode.
D_2	FDD 2 Busy	D_2B	FDD number 2 is in the Seek mode.
D_3	FDD 3 Busy	D_3B	FDD number 3 is in the Seek mode.
D_4	FDC Busy	CB	A read or write command is in process.
D_5	Non-DMA Mode	NDM	The FDC is in the non-DMA mode. This bit is set only during the execution phase in non-DMA mode. Transition to "0" state indicates execution phase has ended.
D_6	Data Input/Output	DIO	Indicates direction of data transfer between FDC and Data Register. If DIO = "1" then transfer is from Data Register to the Processor. If DIO = "0", then transfer is from the Processor to Data Register.
D_7	Request for Master	RQM	Indicates Data Register is ready to send or receive data to or from the Processor. Both bits DIO and RQM should be used to perform the handshaking functions of "ready" and "direction" to the processor.

The DIO and RQM bits in the Status Register indicate when Data is ready and in which direction data will be transferred on the Data Bus.

NOTE:
There is a 12 μs or 24μs RQM flag delay when using an 8 or 4 MHz clock respectively.



210606-6

NOTES:
A—Data register ready to be written into by processor
B—Data register not ready to be written into by processor
C—Data register ready for next data byte to be read by the processor
D—Data register not ready for next data byte to be read by processor

Figure 5. Status Register Timing

The 8272A is capable of executing 15 different commands. Each command is initiated by a multi-byte transfer from the processor, and the result after execution of the command may also be a multi-byte transfer back to the processor. Because of this multi-byte interchange of information between the 8272A and the processor, it is convenient to consider each command as consisting of three phases:

Command Phase: The FDC receives all information required to perform a particular operation from the processor.

Execution Phase: The FDC performs the operation it was instructed to do.

Result Phase: After completion of the operation, status and other housekeeping information are made available to the processor.

During Command or Result Phases the Main Status Register (described in Table 3) must be read by the processor before each byte of information is written into or read from the Data Register. Bits D6 and D7 in the Main Status Register must be in a 0 and 1 state, respectively, before each byte of the command word may be written into the 8272A. Many of the commands require multiple bytes, and as a result the Main Status Register must be read prior to each byte transfer to the 8272A. On the other hand, during the Result Phase, D6 and D7 in the Main Status Register must both be 1's (D6 = 1 and D7 = 1) before reading each byte from the Data Register.

NOTE:
This reading of the Main Status Register before each byte transfer to the 8272A is required in only the Command and Result Phases, and NOT during the Execution Phase.

During the Execution Phase, the Main Status Register need not be read. If the 8272A is in the non-DMA Mode, then the receipt of each data byte (if 8272A is reading data from FDD) is indicated by an interrupt signal on pin 18 (INT = 1). The generation of a Read signal (RD = 0) will reset the interrupt as well as output the Data onto the Data Bus. For example, if the processor cannot handle Interrupts fast enough (every 13 μs for MFM mode) then it may poll the Main Status Register and then bit D7 (RQM) functions just like the Interrupt signal. If a Write Command is in process, then the WR signal performs the reset to the Interrupt signal.

The 8272A always operates in a multi-sector transfer mode. It continues to transfer data until the TC input is active. In Non-DMA Mode, the system must supply the TC input.

If the 8272A is in the DMA Mode, no Interrupts are generated during the Execution Phase. The 8272A generates DRQ's (DMA Requests) when each byte of data is available. The DMA Controller responds to this request with both a $\overline{\text{DACK}}$ = 0 (DMA Acknowledge) and a $\overline{\text{RD}}$ = 0 (Read signal). When the DMA Acknowledge signal goes low ($\overline{\text{DACK}}$ = 0) then the DMA Request is reset (DRQ = 0). If a Write Command has been programmed then a $\overline{\text{WR}}$ signal will appear instead of $\overline{\text{RD}}$. After the Execution Phase has been completed (Terminal Count has occurred) then an Interrupt will occur (INT = 1). This signifies the beginning of the Result Phase. When the first byte of data is read during the Result Phase, the Interrupt is automatically reset (INT = 0).

It is important to note that during the Result Phase all bytes shown in the Command Table must be read. The Read Data Command, for example has seven bytes of data in the Result Phase. All seven bytes must be read in order to successfully complete the Read Data Command. The 8272A will not accept a new command until all seven bytes have been read. Other commands may require fewer bytes to be read during the Result Phase.

The 8272A contains five Status Registers. The Main Status Register mentioned above may be read by the processor at any time. The other four Status Registers (ST0, ST1, ST2, and ST3) are only available during the Result Phase, and may be read only after successfully completing a command. The particular command which has been executed determines how many of the Status Registers will be read.

The bytes of data which are sent to the 8272A to form the Command Phase, and are read out of the 8272A in the Result Phase, must occur in the order shown in the Table 4. That is, the Command Code must be sent first and the other bytes sent in the prescribed sequence. No foreshortening of the Command or Result Phases are allowed. After the last byte of data in the Command Phase is sent to the 8272A, the Execution Phase automatically starts. In a similar fashion, when the last byte of

Table 4. 8272A Command Set

Phase	R/W	Data Bus								Remarks
		D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$	
READ DATA										
Command	W	MT	MFM	SK	0	0	1	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W		———		C			———		Sector ID Information
	W		———		H			———		Prior to Command
	W		———		R			———		Execution
	W		———		N			———		
	W		———		EOT			———		
	W		———		GPL			———		
	W		———		DTL			———		
Execution										Data Transfer
										Between the FDD
										and Main-System
Result	R		———		ST 0			———		Status Information
	R		———		ST 1			———		After Command
	R		———		ST 2			———		Execution
	R		———		C			———		
	R		———		H			———		Sector ID Information
	R		———		R			———		After Command
	R		———		N			———		Execution
READ DELETED DATA										
Command	W	MT	MFM	SK	0	1	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W		———		C			———		Sector ID Information
	W		———		H			———		Prior to Command
	W		———		R			———		Execution
	W		———		N			———		
	W		———		EOT			———		
	W		———		GPL			———		
	W		———		DTL			———		
Execution										Data Transfer
										Between the FDD
										and Main-System
Result	R		———		ST 0			———		Status Information
	R		———		ST 1			———		After Command
	R		———		ST 2			———		Execution
	R		———		C			———		
	R		———		H			———		Sector ID Information
	R		———		R			———		After Command
	R		———		N			———		Execution

Table 4. 8272A Command Set (Continued)

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
WRITE DATA										
Command	W	MT	MFM	0	0	0	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	——— C ———								Sector ID Information Prior to Command Execution
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— DTL ———								
Execution										Data Transfer Between the Main-System and FDD
Result	R	——— ST 0 ———								Status Information After Command Execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								
	R	——— H ———								Sector ID Information After Command Execution
	R	——— R ———								
	R	——— N ———								
WRITE DELETED DATA										
Command	W	MT	MFM	0	0	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	——— C ———								Sector ID Information Prior to Command Execution
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— DTL ———								
Execution										Data Transfer Between the FDD and Main-System
Result	R	——— ST 0 ———								Status Information After Command Execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								
	R	——— H ———								Sector ID Information After Command Execution
	R	——— R ———								
	R	——— N ———								

Table 4. 8272A Command Set (Continued)

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
READ A TRACK										
Command	W	0	MFM	SK	0	0	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID Information Prior to Command Execution
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data Transfer Between the FDD and Main-System. FDC Reads all of Cylinders Contents from Index Hole to EOT
Result	R	ST 0								Status Information After Command Execution
	R	ST 1								
	R	ST 2								
	R	C								
	R	H								Sector ID Information After Command Execution
	R	R								
	R	N								
READ ID										
Command	W	0	MFM	0	0	1	0	1	0	Commands
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution										The First Correct ID Information on the Cylinder is Stored in Data Register
Result	R	ST 0								Status Information After Command Execution
	R	ST 1								
	R	ST 2								
	R	C								
	R	H								Sector ID Information During Execution Phase
	R	R								
	R	N								

Table 4. 8272A Command Set (Continued)

Phase	R/W	Data Bus								Remarks
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
FORMAT A TRACK										
Command	W	0	MFM	0	0	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W		———		N		———			Bytes/Sector
	W		———		SC		———			Sectors/Cylinder
	W		———		GPL		———			Gap 3
	W		———		D		———			Filler Byte
Execution										FDC Formats an Entire Cylinder
Result	R		———		ST 0		———			Status Information
	R		———		ST 1		———			After Command
	R		———		ST 2		———			Execution
	R		———		C		———			
	R		———		H		———			In This Case, the ID
	R		———		R		———			Information has no
	R		———		N		———			Meaning

Table 4. 8272A Command Set (Continued)

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
SCAN EQUAL										
Command	W	MT	MFM	SK	1	0	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W		———		C		———			Sector ID Information Prior to Command Execution
	W		———		H		———			
	W		———		R		———			
	W		———		N		———			
	W		———		EOT		———			
	W		———		GPL		———			
	W		———		STP		———			
Execution										Data Compared Between the FDD and Main-System
Result	R		———		ST 0		———			Status Information After Command Execution
	R		———		ST 1		———			
	R		———		ST 2		———			
	R		———		C		———			
	R		———		H		———			Sector ID Information After Command Execution
	R		———		R		———			
	R		———		N		———			
SCAN LOW OR EQUAL										
Command	W	MT	MFM	SK	1	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W		———		C		———			Sector ID Information Prior to Command Execution
	W		———		H		———			
	W		———		R		———			
	W		———		N		———			
	W		———		EOT		———			
	W		———		GPL		———			
	W		———		STP		———			
Execution										Data Compared Between the FDD and Main-System
Result	R		———		ST 0		———			Status Information After Command Execution
	R		———		ST 1		———			
	R		———		ST 2		———			
	R		———		C		———			
	R		———		H		———			Sector ID Information After Command Execution
	R		———		R		———			
	R		———		N		———			

Table 4. 8272A Command Set (Continued)

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
SCAN HIGH OR EQUAL										
Command	W	MT	MFM	SK	1	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W		———		C		———			Sector ID Information
	W		———		H		———			Prior to Command
	W		———		R		———			Execution
	W		———		N		———			
	W		———		EOT		———			
	W		———		GPL		———			
	W		———		STP		———			
Execution										Data Compared Between the FDD and Main-System
Result	R		———		ST 0		———			Status Information
	R		———		ST 1		———			After Command
	R		———		ST 2		———			Execution
	R		———		C		———			
	R		———		H		———			Sector ID Information
	R		———		R		———			After Command
	R		———		N		———			Execution
RECALIBRATE										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS1	DS0	
Execution										Head Retracted to Track 0
SENSE INTERRUPT STATUS										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R		———		ST 0		———			Status Information at
	R		———		PCN		———			the End of Each Seek Operation About the FDC

Table 4. 8272A Command Set (Continued)

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
SPECIFY										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	_	SRT	———>		<———		HUT	_	
	W	_	HLT	——————————>					ND	
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R		————		ST 3		————			Status Information about FDD
SEEK										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W		————		NCN		————			
Execution										Head is Positioned Over Proper Cylinder on Diskette
INVALID										
Command	W	————			Invalid Codes			————		Invalid Command Codes (NoOp—FDC Goes Into Standby State)
Result	R		————		ST 0		————			ST 0 = 80 (16)

Table 5. Command Mnemonics

Symbol	Name	Description
A_0	Address Line 0	A_0 controls selection of Main Status Register (A_0 = 0) or Data Register (A_0 = 1).
C	Cylinder Number	C stands for the current selected Cylinder track number 0 through 76 of the medium.
D	Data	D stands for the data pattern which is going to be written into a Sector.
D_7–D_0	Data Bus	8-bit Data bus where D_7 is the most significant bit, and D_0 is the least significant bit.
DS0, DS1	Drive Select	DS stands for a selected drive number 0 or 1.
DTL	Data Length	When N is defined as 00, DTL stands for the data length which users are going to read out or write into the Sector.
EOT	End of Track	EOT stands for the final Sector number of a Cylinder.
GPL	Gap Length	GPL stands for the length of Gap 3 (spacing between Sectors excluding VCO Sync Field).
H	Head Address	H stands for head number 0 or 1, as specified in ID field.
HDS	Head Select	HDS stands for a selected head number 0 or 1 (H = HDS in all command words).
HLT	Head Load Time	HLT stands for the head load time in the FDD (2 to 254 ms in 2 ms increments).
HUT	Head Unload Time	HUT stands for the head unload time after a read or write operation has occurred (16 to 240 ms in 16 ms increments).
MFM	FM or MFM Mode	If MF is low, FM mode is selected and if it is high, MFM mode is selected.
MT	Multi-Track	If MT is high, a multi-track operation is to be peformed (a cylinder under both HD0 and HD1 will be read or written).
N	Number	N stands for the number of data bytes written in a Sector.
NCN	New Cylinder Number	NCN stands for a new Cylinder number, which is going to be reached as a result of the Seek operation. Desired position of Head.
ND	Non-DMA Mode	ND stands for operation in the Non-DMA Mode.
PCN	Present Cylinder Number	PCN stands for the Cylinder number at the completion of SENSE INTERRUPT STATUS Command. Position of Head at present time.
R	Record	R stands for the Sector number, which will be read or written.
R/W	Read/Write	R/W stands for either Read (R) or Write (W) signal.
SC	Sector	SC indicates the number of Sectors per Cylinder.
SK	Skip	SK stands for Skip Deleted Data Address Mark.
SRT	Step Rate Time	SRT stands for the Stepping Rate for the FDD (1 to 16 ms in 1 ms increments). The same Stepping Rate applies to all drives (F = 1 ms, E = 2 ms, etc.).

Table 5. Command Mnemonics (Continued)

Symbol	Name	Description
ST 0 ST 1 ST 2 ST 3	Status 0 Status 1 Status 2 Status 3	ST 0–3 stand for one of four registers which store the status information after a command has been executed. This information is available during the result phase after command execution. These registers should not be confused with the main status register (selected by $A_0 = 0$). ST 0–3 may be read only after a command has been executed and contain information relevant to that particular command.
STP		During a Scan operation, if STP = 1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA), and if STP = 2, then alternate sectors are read and compared.

data is read out in the Result Phase, the command is automatically ended and the 8272A is ready for a new command. A command may be aborted by simply sending a Terminal Count signal to pin 16 (TC = 1). This is a convenient means of ensuring that the processor may always get the 8272A's attention even if the disk system hangs up in an abnormal manner.

POLLING FEATURE OF THE 8272A

After power-up RESET, the Drive Select Lines DS0 and DS1 will automatically go into a polling mode. In between commands (and between step pulses in the SEEK command) the 8272A polls all four FDDs looking for a change in the Ready line from any of the drives. If the Ready line changes state (usually due to a door opening or closing) then the 8272A will generate an interrupt. When Status Register 0 (ST0) is read (after Sense Interrupt Status is issued), Not Ready (NR) will be indicated. The polling of the Ready line by the 8272A occurs continuously between instructions, thus notifying the processor which drives are on or off line. Approximate scan timing is shown in Table 6.

Table 6. Scan Timing

DS1	DS0	Approximate Scan Timing
0	0	220 μs
0	1	220 μs
1	0	220 μs
1	1	440 μs

COMMAND DESCRIPTIONS

During the Command Phase, the Main Status Register must be polled by the CPU before each byte is written into the Data Register. The DIO (DB6) and RQM (BD7) bits in the Main Status Register must be in the "0" and "1" states respectively, before each byte of the command may be written into the 8272A. The beginning of the execution phase for any of these commands will cause DIO and RQM to switch to "1" and "0" states respectively.

READ DATA

A set of nine (9) byte words are required to place the FDC into the Read Data Mode. After the Read Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the Specify Command), and begins reading ID Address Marks and ID fields. When the current sector number ("R") stored in the ID Register (IDR) compares with the sector number read off the diskette, then the FDC outputs data (from the data field) byte-by-byte to the main system via the data bus.

After completion of the read operation from the current sector, the Sector Number is incremented by one, and the data from the next sector is read and output on the data bus. This continuous function is called a "Multi-Sector Read Operation." The Read Data Command must be terminated by the receipt of a Terminal Count signal. Upon receipt of this signal, the FDC stops outputting data to the processor, but will continue to read data from the current sector, check CRC (Cyclic Redundancy Count) bytes, and then at the end of the sector terminate the Read Data Command.

The amount of data which can be handled with a single command to the FDC depends upon MT (multi-track), MFM (MFM/FM), and N (Number of Bytes/Sector). Table 7 on the next page shows the Transfer Capacity.

Table 7. Transfer Capacity

Multi-Track MT	MFM/FM MFM	Bytes/Sector N	Maximum Transfer Capacity (Bytes/Sector)(Number of Sectors)	Final Sector Read from Diskette
0 0	0 1	00 01	(128) (26) = 3,328 (256) (26) = 6,656	26 at Side 0 or 26 at Side 1
1 1	0 1	00 01	(128) (52) = 6,656 (256) (52) = 13,312	26 at Side 1
0 0	0 1	01 02	(256) (15) = 3,840 (512) (15) = 7,680	15 at Side 0 or 15 at Side 1
1 1	0 1	01 02	(256) (30) = 7,680 (512) (30) = 15,360	15 at Side 1
0 0	0 1	02 03	(512) (8) = 4,096 (1024) (8) = 8,192	8 at Side 0 or 8 at Side 1
1 1	0 1	02 03	(512) (16) = 8,192 (1024) (16) = 16,384	8 at Side 1

The "multi-track" function (MT) allows the FDC to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector 1, Side 0 and completing at Sector L, Side 1 (Sector L = last sector on the side).

NOTE:
This function pertains to only one cylinder (the same track) on each side of the diskette.

When N = 0, then DTL defines the data length which the FDC must treat as a sector. If DTL is smaller than the actual data length in a Sector, the data beyond DTL in the Sector is not sent to the Data Bus. The FDC reads (internally) the complete Sector performing the CRC check, and depending upon the manner of command termination, may perform a Multi-Sector Read Operation. When N is non-zero, then DTL has no meaning and should be set to 0FFH.

At the completion of the Read Data Command, the head is not unloaded until after Head Unload Time Interval (specified in the Specify Command) has elapsed. If the processor issues another command before the head unloads then the head settling time may be saved between subsequent reads. This time out is particularly valuable when a diskette is copied from one drive to another.

If the FDC detects the Index Hole twice without finding the right sector, (indicated in "R"), then the FDC sets the ND (No Data) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

After reading the ID and Data Fields in each sector, the FDC checks the CRC bytes. If a read error is detected (incorrect CRC in ID field), the FDC sets the DE (Data Error) flag in Status Register 1 to a 1 (high), and if a CRC error occurs in the Data Field the FDC also sets the DD (Data Error in Data Field) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

If the FDC reads a Deleted Data Address Mark off the diskette, and the SK bit (bit D5 in the first Command Word) is not set (SK = 0), then the FDC sets the CM (Control Mark) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command, after reading all the data in the Sector. If SK = 1, the FDC skips the sector with the Deleted Data Address Mark and reads the next sector.

During disk data transfers between the FDC and the processor, via the data bus, the FDC must be serviced by the processor every 27 μs in the FM Mode, and every 13 μs in the MFM Mode, or the FDC sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command.

If the processor terminates a read (or write) operation in the FDC, then the ID Information in the Result Phase is dependent upon the state of the MT bit and EOT byte. Table 5 shows the values for C, H, R, and N, when the processor terminates the Command.

Table 8. ID Information When Processor Terminates Command

MT	EOT	Final Sector Transferred to Processor	ID Information at Result Phase			
			C	H	R	N
0	1A 0F 08	Sector 1 to 25 at Side 0 Sector 1 to 14 at Side 0 Sector 1 to 7 at Side 0	NC	NC	R + 1	NC
	1A 0F 08	Sector 26 at Side 0 Sector 15 at Side 0 Sector 8 at Side 0	C + 1	NC	R = 01	NC
	1A 0F 08	Sector 1 to 25 at Side 1 Sector 1 to 14 at Side 1 Sector 1 to 7 at Side 1	NC	NC	R + 1	NC
	1A 0F 08	Sector 26 at Side 1 Sector 15 at Side 1 Sector 8 at Side 1	C + 1	NC	R = 01	NC
1	1A 0F 08	Sector 1 to 25 at Side 0 Sector 1 to 14 at Side 0 Sector 1 to 7 at Side 0	NC	NC	R + 1	NC
	1A 0F 08	Sector 26 at Side 0 Sector 15 at Side 0 Sector 8 at Side 0	NC	LSB	R = 01	NC
	1A 0F 08	Sector 1 to 25 at Side 1 Sector 1 to 14 at Side 1 Sector 1 to 7 at Side 1	NC	NC	R + 1	NC
	1A 0F 08	Sector 26 at Side 1 Sector 15 at Side 1 Sector 8 at Side 1	C + 1	LSB	R = 01	NC

NOTES:
1. NC (No Change): The same value as the one at the beginning of command execution.
2. LSB (Least Significant Bit): The least significant bit of H is complemented.

WRITE DATA

A set of nine (9) bytes are required to set the FDC into the Write Data mode. After the Write Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the Specify Command), and begins reading ID Fields. When the current sector number ("R"), stored in the ID Register (IDR) compares with the sector number read off the diskette, then the FDC takes data from the processor byte-by-byte via the data bus, and outputs it to the FDD.

After writing data into the current sector, the Sector Number stored in "R" is incremented by one, and the next data field is written into. The FDC continues this "Multi-Sector Write Operation" until the issuance of a Terminal Count signal. If a Terminal Count signal is sent to the FDC it continues writing into the current sector to complete the data field. If the Terminal Count signal is received while a data field is being written then the remainder of the data field is filled with 00 (zeros).

The FDC reads the ID field of each sector and checks the CRC bytes. If the FDC detects a read error (incorrect CRC) in one of the ID Fields, it sets the DE (Data Error) flag of Status Register 1 to a 1 (high), and terminates the Write Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

The Write Command operates in much the same manner as the Read Command. The following items are the same; refer to the Read Data Command for details:

- Transfer Capacity
- EN (End of Cylinder) Flag
- ND (No Data) Flag
- Head Unload Time Interval
- ID Information when the processor terminates command (see Table 2)
- Definition of DTL when N = 0 and when N ≠ 0

In the Write Data mode, data transfers between the processor and FDC must occur every 31 μs in the FM mode, and every 15 μs in the MFM mode. If the time interval between data transfers is longer than this then the FDC sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Write Data Command.

For mini-floppies, multiple track writes are usually not permitted. This is because of the turn-off time of the erase head coils—the head switches tracks before the erase head turns off. Therefore the system should typically wait 1.3 ms before attempting to step or change sides.

WRITE DELETED DATA

This command is the same as the Write Data Command except a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark.

READ DELETED DATA

This command is the same as the Read Data Command except that when the FDC detects a Data Address Mark at the beginning of a Data Field (and SK = 0 (low)), it will read all the data in the sector and set the CM flag in Status Register 2 to a 1 (high), and then terminate the command. If SK = 1, then the FDC skips the sector with the Data Address Mark and reads the next sector.

READ A TRACK

This command is similar to READ DATA Command except that the entire data field is read continuously from each of the sectors of a track. Immediately after encountering the INDEX HOLE, the FDC starts reading all data fields on the track as continuous blocks of data. If the FDC finds an error in the ID or DATA CRC check bytes, it continues to read data from the track. The FDC compares the ID information read from each sector with the value stored in the IDR, and sets the ND flag of Status Register 1 to a 1 (high) if there is no comparison. Multi-track or skip operations are not allowed with this command.

This command terminates when EOT number of sectors have been read. If the FDC does not find an ID Address Mark on the diskette after it encounters the INDEX HOLE for the second time, then it sets the MA (missing address mark) flag in Status Register 1 to a 1 (high), and terminates the command. (Status Register 0 has bits 7 and 6 set to 0 and 1 respectively.)

READ ID

The READ ID Command is used to give the present position of the recording head. The FDC stores the values from the first ID Field it is able to read. If no proper ID Address Mark is found on the diskette, before the INDEX HOLE is encountered for the second time then the MA (Missing Address Mark) flag in Status Register 1 is set to a 1 (high), and if no data is found then the ND (No Data) flag is also set in Status Register 1 to a 1 (high) and the command is terminated.

FORMAT A TRACK

The Format Command allows an entire track to be formatted. After the INDEX HOLE is detected, Data is written on the Diskette: Gaps, Address Marks, ID Fields and Data Fields, all per the IBM System 34 (Double Density) or System 3740 (Single Density) Format are recorded. The particular format which will be written is controlled by the values programmed into N (number of bytes/sector), SC (sectors/cylinder), GPL (Gap Length), and D (Data Pattern) which are supplied by the processor during the Command Phase. The Data Field is filled with the Byte of data stored in D. The ID Field for each sector is supplied by the processor; that is, four data requests per sector are made by the FDC for C (Cylinder Number), H(Head Number), R(Sector Number) and N(Number of Bytes/Sector). This allows the diskette to be formatted with nonsequential sector numbers, if desired.

After formatting each sector, the processor must send new values for C, H, R, and N to the 8272A for each sector on the track. The contents of the R Register is incremented by one after each sector is formatted, thus, the R register contains a value of R + 1 when it is read during the Result Phase. This incrementing and formatting continues for the whole track until the FDC encounters the INDEX HOLE for the second time, whereupon it terminates the command.

If a FAULT signal is received from the FDD at the end of a write operation, then the FDC sets the EC flag of Status Register 0 to a 1 (high), and terminates the command after setting bits 7 and 6 of Status Register 0 to 0 and 1 respectively. Also the loss of a READY signal at the beginning of a command execution phase causes command termination.

Table 9 shows the relationship between N, SC, and GPL for various sector sizes:

Table 9. Sector Size Relationships

Format	Bytes/ Sector	8" Floppy				Bytes/ Sector	5¼" Floppy				Bytes/ Sector	3½" Mini Floppy			
		N	SC	GPL(1)	GPL(2)		N	SC	GPL(1)	GPL(2)		N	SC	GPL(1)	GPL(2)
FM Mode	128	00	1A	07	1B	128	00	12	07	09	128	0	0F	07	1B
	256	01	0F	0E	2A	128	00	10	10	19	—	—	—	—	—
	512	02	08	1B	3A	256	01	08	18	30	256	1	09	0F	2A
	1024	03	04	47	8A	512	02	04	46	87	512	2	05	1B	3A
	2048	04	02	C8	FF	1024	03	02	C8	FF	—	—	—	—	—
	4096	05	01	C8	FF	2048	04	01	C8	FF	—	—	—	—	—
MPM Mode	256	01	1A	0E	36	256	01	12	0A	0C	256	1	0F	CE	36
	512	02	OF	1B	54	256	01	10	20	32	—	—	—	—	—
	1024	03	08	35	74	512	02	08	2A	50	512	2	09	1B	54
	2048	04	04	99	FF	1024	03	04	80	F0	1024	3	05	35	74
	4096	05	02	C8	FF	2048	04	02	C8	FF	—	—	—	—	—
	8192	06	01	C8	FF	4096	05	01	C8	FF	—	—	—	—	—

NOTES:
1. Suggested values of GPL in Read or Write Commands to avoid splice point between data field and ID field of contiguous sections.
2. Suggested values of GPL in format command.

SCAN COMMANDS

The SCAN Commands allow data which is being read from the diskette to be compared against data which is being supplied from the main system (Processor in NON-DMA mode, and DMA Controller in DMA mode). The FDC compares the data on a byte-by-byte basis, and looks for a sector of data which meets the conditions of $D_{FDD} = D_{Processor}$, $D_{FDD} \leq D_{Processor}$, or $D_{FDD} \geq D_{Processor}$. Ones complement arithmetic is used for comparison (FF = largest number, 00 = smallest number). After a whole sector of data is compared, if the conditions are not met, the sector number is incremented (R + STP → R), and the scan operation is continued. The scan operation continues until one of the following conditions occur; the conditions for scan are met (equal, low, or high), the last sector on the track is reached (EOT), or the terminal count signal is received.

If the conditions for scan are met then the FDC sets the SH (Scan Hit) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. If the

Table 10. Scan Status Codes

Command	Status Register 2		Comments
	Bit 2 = SN	Bit 3 = SH	
Scan Equal	0	1	$D_{FDD} = D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan Low or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} < D_{Processor}$
	1	0	$D_{FDD} \ngtr D_{Processor}$
Scan High or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} > D_{Processor}$
	1	0	$D_{FDD} \nless D_{Processor}$

conditions for scan are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), then the FDC sets the SN (Scan Not Satisfied) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. The receipt of a TERMINAL COUNT signal from the Processor or DMA Controller during the scan operation will cause the FDC to complete the comparison of the particular byte which is in process, and then to terminate the command. Table 10 shows the status of bits SH and SN under various conditions of SCAN.

If the FDC encounters a Deleted Data Address Mark on one of the sectors (and SK = 0), then it regards the sector as the last sector on the cylinder, sets CM (Control Mark) flag of Status Register 2 to a 1 (high) and terminates the command. If SK = 1, the FDC skips the sector with the Deleted Address Mark, and reads the next sector. In the second case (SK = 1), the FDC sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) in order to show that a Deleted Sector had been encountered.

When either the STP (contiguous sectors STP = 01, or alternate sectors STP = 02 sectors are read) or the MT (Multi-Track) are programmed, it is necessary to remember that the last sector on the track must be read. For example, if STP = 02, MT = 0, the sectors are numbered sequentially 1 through 26, and we start the Scan Command at sector 21; the following will happen. Sectors 21, 23, and 25 will be read, then the next sector (26) will be skipped and the Index Hole will be encountered before the EOT value of 26 can be read. This will result in an abnormal termination of the command. If the EOT had been set at 25 or the scanning started at sector 20, then the Scan Command would be completed in a normal manner.

During the Scan Command data is supplied by either the processor or DMA Controller for comparison against the data read from the diskette. In order to avoid having the OR (Over Run) flag set in Status Register 1, it is necessary to have the data available in less than 27 μs (FM Mode) or 13 μs (MFM Mode). If an Overrun occurs the FDC terminates the command.

SEEK

The read/write within the FDD is moved from cylinder to cylinder under control of the Seek Command. The FDC compares the PCN (Present Cylinder Number) which is the current head position with the NCN (New Cylinder Number), and performs the following operation if there is a difference:

PCN < NCN: Direction signal to FDD set to a 1 (high), and Step Pulses are issued. (Step In.)

PCN > NCN: Direction signal to FDD set to a 0 (low), and Step Pulses are issued. (Step Out.)

The rate at which Step Pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY Command. After each Step Pulse is issued NCN is compared against PCN, and when NCN = PCN, then the SE (Seek End) flag is set in Status Register 0 to a 1 (high), and the command is terminated.

During the Command Phase of the Seek operation the FDC is in the FDC BUSY state, but during the Execution Phase it is in the NON BUSY state. While the FDC is in the NON BUSY state, another Seek Command may be issued, and in this manner parallel seek operations may be done on up to 4 Drives at once.

If an FDD is in a NOT READY state at the beginning of the command execution phase or during the seek operation, then the NR (NOT READY) flag is set in Status Register 0 to a 1 (high), and the command is terminated.

Note that the 8272A Read and Write Commands do not have implied Seeks. Any R/W command should be preceded by: 1) Seek Command; 2) Sense Interrupt Status; and 3) Read ID.

RECALIBRATE

This command causes the read/write head within the FDD to retract to the Track 0 position. The FDC clears the contents of the PCN counter, and checks the status of the Track 0 signal from the FDD. As long as the Track 0 signal is low, the Direction signal remains 1 (high) and Step Pulses are issued. When the Track 0 signal goes high, the SE (SEEK END) flag in Status Register 0 is set to a 1 (high) and the command is terminated. If the Track 0 signal is still low after 77 Step Pulses have been issued, the FDC sets the SE (SEEK END) and EC (EQUIPMENT CHECK) flags of Status Register 0 to both 1s (highs), and terminates the command.

The ability to overlap RECALIBRATE Commands to multiple FDDs, and the loss of the READY signal, as described in the SEEK Command, also applies to the RECALIBRATE Command.

SENSE INTERRUPT STATUS

An Interrupt signal is generated by the FDC for one of the following reasons:

1) Upon entering the Result Phase of:
   a) Read Data Command
   b) Read a Track Command
   c) Read ID Command
   d) Read Deleted Data Command
   e) Write Data Command
   f) Format a Cylinder Command
   g) Write Deleted Data Command
   h) Scan Commands
2) Ready Line of FDD changes state
3) End of Seek or Recalibrate Command
4) During Execution Phase in the NON-DMA Mode

Interrupts caused by reasons 1 and 4 above occur during normal command operations and are easily discernible by the processor. However, interrupts caused by reasons 2 and 3 above may be uniquely identified with the aid of the Sense Interrupt Status Command. This command when issued resets the interrupt signal and via bits 5, 6, and 7 of Status Register 0 identifies the cause of the interrupt.

Neither the Seek or Recalibrate Command have a Result Phase. Therefore, it is mandatory to use the Sense Interrupt Status Command after these commands to effectively terminate them and to provide verification of the head position (PCN).

Table 11. Seek, Interrupt Codes

Seek End Bit 5	Interrupt Code		Cause
	Bit 6	Bit 7	
0	1	1	Ready Line Changed State, Either Polarity
1	0	0	Normal Termination of Seek or Recalibrate Command
1	1	0	Abnormal Termination of Seek or Recalibrate Command

SPECIFY

The Specify Command sets the intitial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the Execution Phase of one of the Read/Write Commands to the head unload state. This timer is programmable from 16 to 240 ms in increments of 16 ms (01 = 16 ms, 02 = 32 ms . . . . OF = 240 ms). The SRT (Step Rate Time) defines the time interval between adjacent step pulses. This timer is programmable from 1 to 16 ms in increments of 1 ms (F = 1 ms, E = 2 ms, D = 3 ms, etc.). The HLT (Head Load Time) defines the time between when the Head Load signal goes high and when the Read/Write operation starts. This timer is programmable from 2 to 254 ms in increments of 2 ms (01 = 2 ms, 02 = 4 ms, 03 = 6 ms . . . . FE = 254 ms).

The step rate should be programmed 1 ms longer than the minimum time required by the drive.

The time intervals mentioned above are a direct function of the clock (CLK on pin 19). Times indicated above are for an 8 MHz clock, if the clock was reduced to 4 MHz (mini-floppy application) then all time intervals are increased by a factor of 2.

The choice of DMA or NON-DMA operation is made by the ND (NON-DMA) bit. When this bit is high (ND = 1) the NON-DMA mode is selected, and when ND = 0 the DMA mode is selected.

SENSE DRIVE STATUS

This command may be used by the processor whenever it wishes to obtain the status of the FDDs. Status Register 3 contains the Drive Status information.

INVALID

If an invalid command is sent to the FDC (a command not defined above), then the FDC will terminate the command. No interrupt is generated by the 8272A during this condition. Bit 6 and bit 7 (DIO and RQM) in the Main Status Register are both high ("1") indicating to the processor that the 8272A is in the Result Phase and the contents of Status Register 0 (STO) must be read. When the processor reads Status Register 0 it will find an 80H indicating an invalid command was received.

A Sense Interrupt Status Command must be sent after a Seek or Recalibrate interrupt, otherwise the FDC will consider the next command to be an Invalid Command.

In some applications the user may wish to use this command as a No-Op command, to place the FDC in a standby or no operation state.

Table 12. Status Registers

Bit			Description
No.	Name	Symbol	
STATUS REGISTER 0			
D_7	Interrupt Code	IC	$D_7 = 0$ and $D_6 = 0$ Normal Termination of Command, (NT). Command was completed and properly executed.
D_6			$D_7 = 0$ and $D_6 = 1$ Abnormal Termination of Command, (AT). Execution of Command was started, but was not successfully completed.
			$D_7 = 1$ and $D_6 = 0$ Invalid Command issue, (IC). Command which was issued was never started.
			$D_7 = 1$ and $D_6 = 1$ Abnormal Termination because during command execution the ready signal from FDD changed state.
D_5	Seek End	SE	When the FDC completes the SEEK Command, this flag is set to 1 (high).
D_4	Equipment Check	EC	If a fault Signal is received from the FDD, or if the Track 0 Signal fails to occur after 77 Step Pulses (Recalibrate Command) then this flag is set.
D_3	Not Ready	NR	When the FDD is in the not-ready state and a read or write command is issued, this flag is set. If a read or write command is issued to Side 1 of a single sided drive, then this flag is set.
D_2	Head Address	HD	This flag is used to indicate the state of the head at Interrupt.
D_1	Unit Select 1	US 1	These flags are used to indicate a Drive Unit Number at Interrupt.
D_0	Unit Select 0	US 0	
STATUS REGISTER 1			
D_7	End of Cylinder	EN	When the FDC tries to access a Sector beyond the final Sector of a Cylinder, this flag is set.
D_6			Not used. This bit is always 0 (low).
D_5	Data Error	DE	When the FDC detects a CRC error in either the ID field or the data field, this flag is set.
D_4	Over Run	OR	If the FDC is not serviced by the main-systems during data transfers, within a certain time interval, this flag is set.
D_3			Not used. This bit always 0 (low).
D_2	No Data	ND	During execution of READ DATA, WRITE DELETED DATA or SCAN Command, if the FDC cannot find the Sector specified in the IDR Register, this flag is set.
			During executing the READ ID Command, if the FDC cannot read the ID field without an error, then this flag is set.
			During the execution of the READ A Cylinder Command, if the starting sector cannot be found, then this flag is set.

Table 12. Status Register (Continued)

Bit			Description
No.	Name	Symbol	
STATUS REGISTER 1 (Continued)			
D_1	Not Writable	NW	During execution of WRITE DATA, WRITE DELETED DATA or Format A Cylinder Command, if the FDC detects a write protect signal from the FDD, then this flag is set.
D_0	Missing Address Mark	MA	If the FDC cannot detect the ID Address Mark after encountering the index hole twice, then this flag is set. If the FDC cannot detect the Data Address Mark or Deleted Data Address Mark, this flag is set. Also at the same time, the MD (Missing Address Mark in Data Field) of Status Register 2 is set.
STATUS REGISTER 2			
D_7			Not used. This bit is always 0 (low).
D_6	Control Mark	CM	During executing the READ DATA or SCAN Command, if the FDC encounters a Sector which contains a Deleted Data Address Mark, this flag is set.
D_5	Data Error in Data Field	DD	If the FDC detects a CRC error in the data field then this flag is set.
D_4	Wrong Cylinder	WC	This bit is related with the ND bit, and when the contents of C on the medium is different from that stored in the IDR, this flag is set.
D_3	Scan Equal Hit	SH	During execution, the SCAN Command, if the condition of "equal" is satisfied, this flag is set.
D_2	Scan Not Satisfied	SN	During executing the SCAN Command, if the FDC cannot find a Sector on the cylinder which meets the condition, then this flag is set.
D_1	Bad Cylinder	BC	This bit is related with the ND bit, and when the content of C on the medium is different from that stored in the IDR and the content of C is FF, then this flag is set.
D_0	Missing Address Mark in Data Field	MD	When data is read from the medium, if the FDC cannot find a Data Address Mark or Deleted Data Address Mark, then this flag is set.
STATUS REGISTER 3			
D_7	Fault	FT	This bit is used to indicate the status of the Fault signal from the FDD.
D_6	Write Protected	WP	This bit is used to indicate the status of the Write Protected signal from the FDD.
D_5	Ready	RDY	This bit is used to indicate the status of the Ready signal from the FDD.
D_4	Track 0	T0	This bit is used to indicate the status of the Track 0 signal from the FDD.
D_3	Two Side	TS	This bit is used to indicate the status of the Two Side signal from the FDD.
D_2	Head Address	HD	This bit is used to indicate the status of Side Select signal to the FDD.
D_1	Unit Select 1	US 1	This bit is used to indicate the status of the Unit Select 1 signal to the FDD.
D_0	Unit Select 0	US 0	This bit is used to indicate the status of the Unit Select 0 signal to the FDD.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature	0°C to +70°C
Storage Temperature	−40°C to +125°C
All Output Voltages	−0.5 to +7V
All Input Voltages	−0.5 to +7V
Supply Voltage V_{CC}	−0.5 to +7V
Power Dissipation	1 Watt

*T_A = 25°C

**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 T_A = 0°C to +70°C, V_{CC} = +5V ±10%

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4	V_{CC}	V	I_{OH} = −400 µA
I_{CC}	V_{CC} Supply Current		120	mA	
I_{IL}	Input Load Current (All Input Pins)		10 −10	µA µA	V_{IN} = V_{CC} V_{IN} = 0V
I_{LOH}	High Level Output Leakage Current		10	µA	V_{OUT} = V_{CC}
I_{OFL}	Output Float Leakage Current		±10	µA	0.45C ≤ V_{OUT} ≤ V_{CC}

CAPACITANCE T_A = 25°C, f_c = 1 MHz, V_{CC} = 0V

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
$C_{IN(\phi)}$	Clock Input Capacitance		20	pF	All Pins Except Pin Under Test Tied to AC Ground
C_{IN}	Input Capacitance		10	pF	
$C_{I/O}$	Input/Output Capacitance		20	pF	

A.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5.0V \pm 10\%$

Symbol	Parameter	Typ(1)	Min	Max	Unit	Notes
CLOCK TIMING						
t_{CY}	Clock Period		120	500	ns	(Note 5)
t_{CH}	Clock High Period		40		ns	(Note 4, 5)
t_{RST}	Reset Width		14		t_{CY}	
READ CYCLE						
t_{AR}	Select Setup to $\overline{RD}$ ↓		0		ns	
t_{RA}	Select Hold from $\overline{RD}$ ↑		0		ns	
t_{RR}	$\overline{RD}$ Pulse Width		250		ns	
t_{RD}	Data Delay from $\overline{RD}$ ↓			200	ns	
t_{DF}	Output Float Delay		20	100	ns	
WRITE CYCLE						
t_{AW}	Select Setup to $\overline{WR}$ ↓		0		ns	
t_{WA}	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 $\overline{WR}$ ↑		5		ns	
INTERRUPTS						
t_{RI}	INT Delay from $\overline{RD}$ ↑			500	ns	(Note 6)
t_{WI}	INT Delay from $\overline{WR}$ ↑			500	ns	(Note 6)
DMA						
t_{RQCY}	DRQ Cycle Period		13		µs	(Note 6)
t_{AKRQ}	$\overline{DACK}$ ↓ to DRQ ↓			200	ns	
t_{RQR}	DRQ ↑ to $\overline{RD}$ ↓		800		ns	(Note 6)
t_{RQW}	DRQ ↑ to $\overline{WR}$ ↓		250		ns	(Note 6)
t_{RQRW}	DRQ ↑ to $\overline{RD}$ ↑ or $\overline{WR}$ ↑			12	µs	(Note 6)

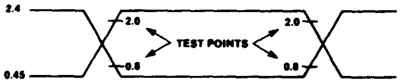
A.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = +5.0V ±10% (Continued)

Symbol	Parameter	Typ(1)	Min	Max	Unit	Notes
FDD INTERFACE						
t_{WCY}	WCK Cycle Time	2 or 4 1 or 2			μs	MFM = 0 MFM = 1 (Note 2)
t_{WCH}	WCK High Time	250	80	350	ns	
t_{CP}	Pre-Shift Delay from WCK ↑		20	100	ns	
t_{CD}	WDA Delay from WCK ↑		20	100	ns	
t_{WDD}	Write Data Width		t_{WCH} − 50		ns	
t_{WE}	WE ↑ to WCK ↑ or WE ↓ to WCK ↓ Delay		20	100	ns	
t_{WWCY}	Window Cycle Time	2 1			μs	MM = 0 MFM = 1
t_{WRD}	Window Setup to RDD ↑		15		ns	
t_{RDW}	Window Hold from RDD ↓		15		ns	
t_{RDD}	RDD Active Time (HIGH)		40		ns	
FDD SEEK/DIRECTION/STEP						
t_{US}	$US_{0,1}$ Setup to $\overline{RW}$/SEEK ↑		12		μs	(Note 6)
t_{SU}	$US_{0,1}$ Hold after $\overline{RW}$/SEEK ↓		15		μs	(Note 6)
t_{SD}	$\overline{RW}$/SEEK Setup to LCT/DIR		7		μs	(Note 6)
t_{DS}	$\overline{RW}$/SEEK Hold from LCT/DIR		30		μs	(Note 6)
t_{DST}	LCT/DIR Setup to FR/STEP ↑		1		μs	(Note 6)
t_{STD}	LCT/DIR Hold from FR/STEP ↓		24		μs	(Note 6)
t_{STU}	$DS_{2,1}$ Hold from FR/Step ↓		5		μs	(Note 6)
t_{STP}	STEP Active Time (High)	5			μs	(Note 6)
t_{SC}	STEP Cycle Time		33		μs	(Note 3, 6)
t_{FR}	FAULT RESET Active Time (High)		8	10	μs	(Note 6)
t_{IDX}	INDEX Pulse Width	10			t_{CY}	
t_{TC}	Terminal Count Width		1		t_{CY}	

NOTES:

1. Typical values for T_A = 25°C and nominal supply voltage.
2. The former values are used for standard floppy and the latter values are used for mini-floppies.
3. t_{SC} = 33 μs min. is for different drive units. In the case of same unit, t_{SC} can be ranged from 1 ms to 16 ms with 8 MHz clock period, and 2 ms to 32 ms with 4 MHz clock, under software control.
4. From 2.0V to +2.0V.
5. At 4 MHz, the clock duty cycle may range from 16% to 76%. Using an 8 MHz clock the duty cycle can range from 32% to 52%. Duty cycle is defined as: D.C. = 100 (t_{CH} ÷ t_{CY}) with typical rise and fall times of 5 ns.
6. The specified values listed are for an 8 MHz clock period. Multiply timings by 2 when using a 4 MHz clock period.

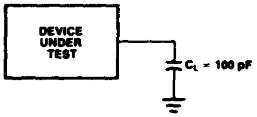
A.C. TESTING INPUT, OUTPUT WAVEFORM



210606-7

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT

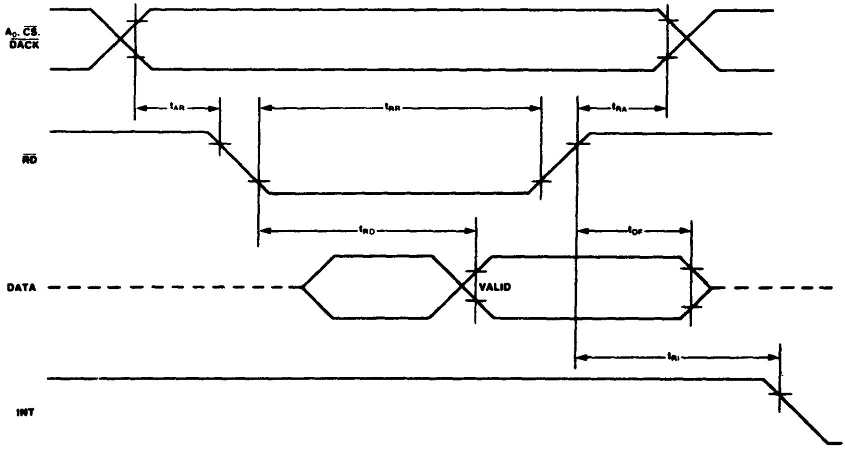


210606-8

C_L = 100 pF
C_L Includes Jig Capacitance

WAVEFORMS

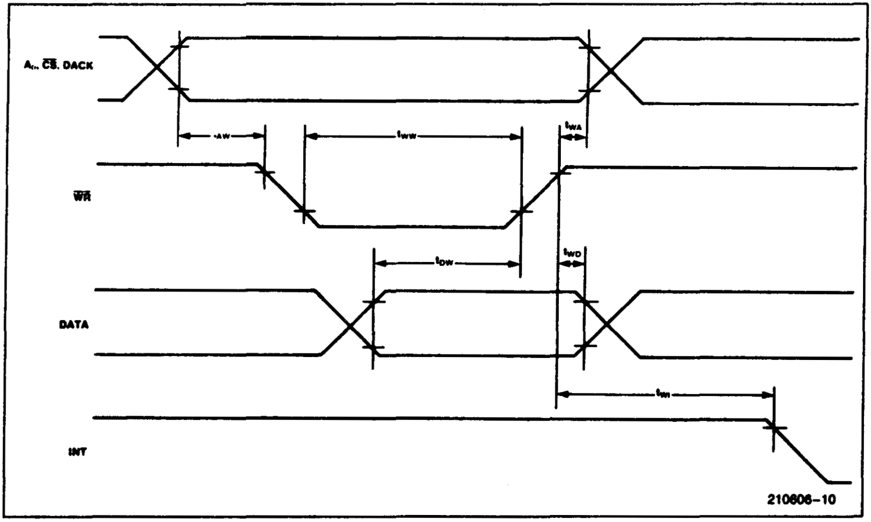
PROCESSOR READ OPERATION



210606-9

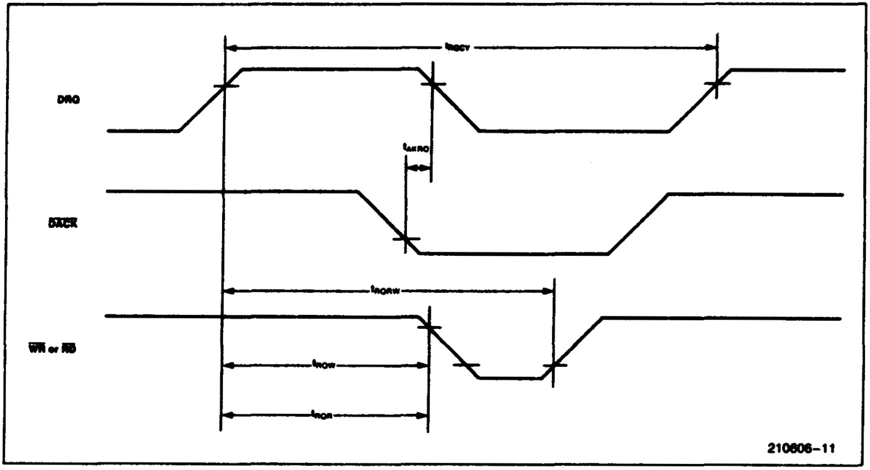
WAVEFORMS (Continued)

PROCESSOR WRITE OPERATION



210606-10

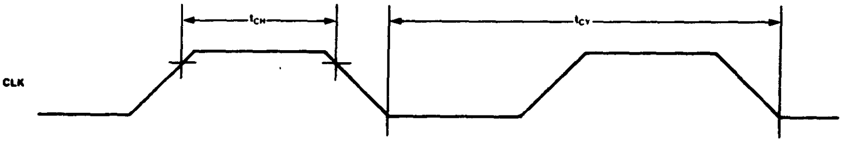
DMA OPERATION



210606-11

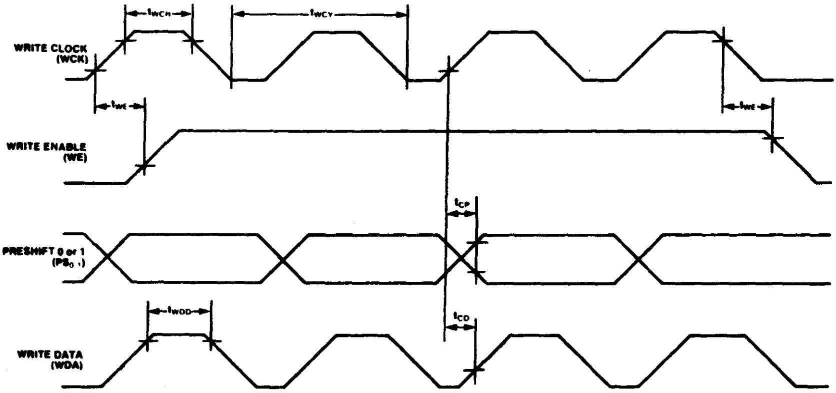
WAVEFORMS (Continued)

CLOCK TIMING



210606-12

FDD WRITE OPERATION

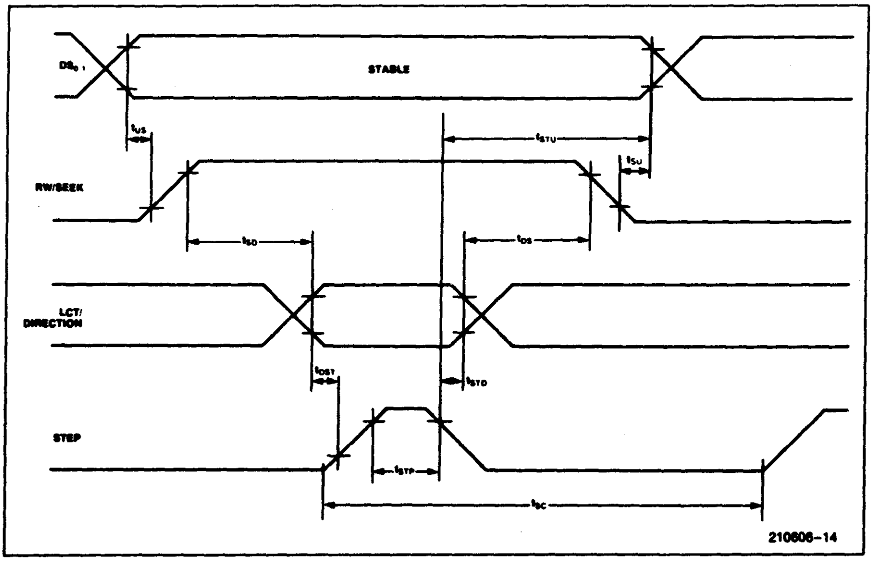


210606-13

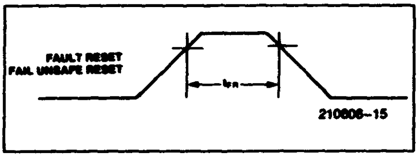
	Preshift 0	Preshift 1
Normal	0	0
Late	0	1
Early	1	0
Invalid	1	1

WAVEFORMS (Continued)

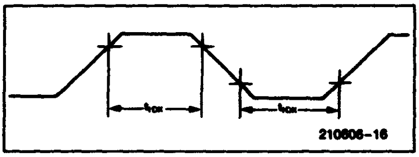
SEEK OPERATION



FLT RESET



INDEX



Tandy Video II

Custom IC

Part Number 8079020

Table Of Contents

IMPROVED TANDY 1000 VIDEO CONTROLLER

VIDEO CONTROLLER FEATURES

1) 100 % software compatible with the current Tandy 1000 video controller design.

2) Provides international support with alternate character sets and video mode control locking.

3) 128K of Video/System memory.

4) External character ROM for all 200 line and 350 line modes.

5) Supports 720 X 350 Monochrome text and 720 X 348 Hercules graphics.

Improved Tandy 1000 Video Controller Pin-Out

PIN#	NAME	DESCRIPTION	DRIVE	PIN#	NAME	DESCRIPTION	DRIVE
1	MEMR-	Mem read	IN	51	CFNT3	Font dat/adr 3	2MA
2	MEMW-	Mem write	IN	52	CFNT4	Font dat/adr 4	2MA
3	GND	GROUND	IN	53	GND	GROUND	IN
4	IOR-	I/O Read	IN	54	CFNT5	Font dat/adr 5	2MA
5	IOW-	I/O Write	IN	55	CFNT6	Font dat/adr 6	2MA
6	BHE-	CPU CTL	IN	56	CFNT7	Font dat/adr 7	2MA
7	RESET	Sys Reset	IN	57	CFNT8	Cfont add	2MA
8	ADRLTCH	Add latch	IN	58	CFNT9	Cfont add	2MA
9	VIDWT	Video wait	8MA	59	CFNT10	Cfont add	2MA
10	XMD0	Memory A/D	4MA	60	CFNT11	Cfont add	2MA
11	XMD1	Memory A/D	4MA	61	CFNT12	Cfont add	2MA
12	XMD2	Memory A/D	4MA	62	CFNT13	Cfont add	2MA
13	XMD3	Memory A/D	4MA	63	N/C	NO CONNECT	2MA
14	XMD4	Memory A/D	4MA	64	CLTH	Cfont latch	2MA
15	VCC	+ 5 Volts	IN	65	VCC	+ 5 Volts	IN
16	XMD5	Memory A/D	4MA	66	14.3	SYS CLOCK	4MA
17	XMD6	Memory A/D	4MA	67	RFSH	REFRESH IN	IN
18	XMD7	Memory A/D	4MA	68	D0	CPU Data	4MA
19	YMD0	Memory A/D	4MA	69	D1	CPU Data	4MA
20	YMD1	Memory A/D	4MA	70	D2	CPU Data	4MA
21	YMD2	Memory A/D	4MA	71	D3	CPU Data	4MA
22	YMD3	Memory A/D	4MA	72	D4	CPU Data	4MA
23	YMD4	Memory A/D	4MA	73	D5	CPU Data	4MA
24	YMD5	Memory A/D	4MA	74	D6	CPU Data	4MA
25	YMD6	Memory A/D	4MA	75	D7	CPU Data	4MA
26	YMD7	Memory A/D	4MA	76	N/C	No Connect	
27	OEXY-	RAM CTL	4MA	77	DOTCLK	DOT CLOCK	4MA
28	GND	Ground	IN	78	GND	Ground	IN
29	RAS-	RAM CTL	8MA	79	25	25.175	IN
30	CAS-	RAM CTL	8MA	80	28MHZ	Clock	IN
31	MWEX-	RAM Write	4MA	81	A0/MD0	CPU A/D	4MA
32	MWEY-	RAM Write	4MA	82	A1/MD1	CPU A/D	4MA
33	MEMIOS-	Mem & I/O sel	2MA	83	A2/MD2	CPU A/D	4MA
34	N/C	NO CONNECT		84	A3/MD3	CPU A/D	4MA
35	N/C	NO CONNECT		85	A4/MD4	CPU A/D	4MA
36	OUTR	RED VIDEO	4MA	86	A5/MD5	CPU A/D	4MA
37	OUTG	GREEN/MONO	4MA	87	A6/MD6	CPU A/D	4MA
38	OUTB	BLUE VIDEO	4MA	88	A7/MD7	CPU A/D	4MA
39	OUTI	INTENSITY	4MA	89	A8/MD8	CPU A/D	4MA
40	OUTHSYNC	Horz Sync	4MA	90	A9/MD9	CPU A/D	4MA
41	OUTVSYNC	Vert Sync	4MA	91	A10/MD10	CPU A/D	4MA
42	N/C	NO CONNECT		92	A11/MD11	CPU A/D	4MA
43	N/C	NO CONNECT		93	A12/MD12	CPU A/D	4MA
44	N/C	NO CONNECT		94	A13/MD13	CPU A/D	4MA
45	N/C	NO CONNECT		95	A14/MD14	CPU A/D	4MA
46	FRMOE*	ROM Enable	2MA	96	A15/MD15	CPU A/D	4MA
47	N/C	NO CONNECT	2MA	97	A16	CPU Address	IN
48	CFNT0	Font dat/adr 0	2MA	98	A17	CPU Address	IN
49	CFNT1	Font dat/adr 1	2MA	99	A18	CPU Address	IN
50	CFNT2	Font dat/adr 2	2MA	100	A19	CPU Address	IN

General Description

The new improved Tandy 1000 video controller chip is designed to operate with one of two types of monitors, an RGBI 200 line Color monitor or a Monochrome 350 line monitor. This custom controller implements all of the video logic for the Tandy 1000. Figure 1 shows a block diagram of this custom video controller chip.

If an RGBI 200 line monitor is used, the display system supports up to 16 colors. These sixteen colors are defined by combinations of the R, G, B, and I bits as shown in the chart below.

I	R	G	B	COLOR
0	0	0	0	BLACK
0	0	0	1	BLUE
0	0	1	0	GREEN
0	0	1	1	CYAN
0	1	0	0	RED
0	1	0	1	MAGENTA
0	1	1	0	BROWN
0	1	1	1	LIGHT GRAY
1	0	0	0	DARK GRAY
1	0	0	1	LIGHT BLUE
1	0	1	0	LIGHT GREEN
1	0	1	1	LIGHT CYAN
1	1	0	0	PINK
1	1	0	1	LIGHT MAGENTA
1	1	1	0	YELLOW
1	1	1	1	WHITE

TABLE 1
AVAILABLE COLORS

OPERATING MODES

The operating modes supported by the Tandy 1000 video controller may be grouped in two categories: Alphanumeric and Graphic. A list of these modes is shown in Table 1A.

ALPHANUMERIC MODES

The Alphanumeric modes have two basic types of operation:

80 character and 40 character. In both modes the character font tables are stored in a separate character ROM and consist of one or two pages of 256 characters. Two bytes of data are used to define each character on the screen. The even address is the character code and is used to address the character font pattern stored in ROM. The odd address is the attribute byte, that defines the foreground and the background color of the character. The following chart shows how the attribute byte controls the colors.

ATTRIBUTE BYTE							
7	6	5	4	3	2	1	0
BACKGROUND				FOREGROUND			
I	R	G	B	I	R	G	B
*	R	G	B	*	R	G	B

^ Selects blink ^ Selects font page

TABLE 2
ALPHANUMERIC MODE ATTRIBUTE BYTE DEFINITION

TANDY 1000 IMPROVED VIDEO CONTROLLER MODES

BIOS			ALPHA	BUFFER	BOX			
MODE	TYPE	COLORS	FORMAT	START	SIZE	MONITOR	RESOLUTION	TYPE
0/1	ALPHA	16	40 X 25	B8000	8X9	CM-5/11	320 X 225	CGA
2/3	ALPHA	16	80 X 25	B8000	8X9	CM-5/11	640 X 225	CGA
4/5	GRAPHICS	4	40 X 25	B8000	8X8	CM-5/11	320 X 200	CGA
6	GRAPHICS	2	80 X 25	B8000	8X8	CM-5/11	640 X 200	CGA
7	ALPHA	4	80 X 25	B0000	9X14	VM-5	720 X 350	MDA
7H	GRAPHICS	2	80 X 25	B0000	9X14	VM-5	720 X 348	MDA
8	GRAPHICS	16	20 X 25	B8000	8X8	CM-5/11	160 X 200	PCjr
9	GRAPHICS	16	40 X 25	B8000	8X8	CM-5/11	320 X 200	PCjr
A	GRAPHICS	4	80 X 25	B8000	8X8	CM-5/11	640 X 200	PCjr
B	RESERVED	Used by the BIOS to load the color character font						
C	RESERVED	Used by the BIOS to load the mono character font						
E	GRAPHICS	16	80 X 25	A0000	8X8	CM-5/11	640 X 200	TDA

TABLE 1A

To take advantage of the type of monitor used, the Tandy 1000 supports two different types of character box sizes depending on the type of monitor that is used. With a standard 200 line RGBI monitor, the Tandy 1000 video controller supports an 8 X 9 character box. On the 350 lin monochrome monitor, a 9 X 14 character box is used.

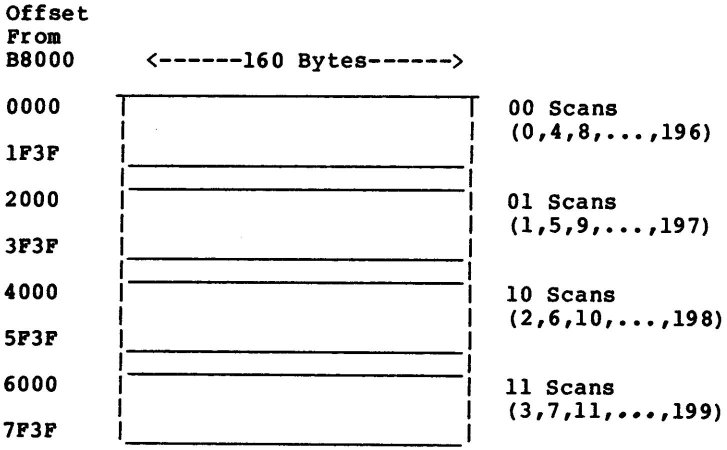
GRAPHIC MODES

The Tandy 1000 video controller supports a variety of graphics modes.

GRAPHICS MEMORY USAGE

*200 Line Low or Medium Resolution Graphics Memory uses either 2 or 4 banks of 8000 bytes. In either case, pixel information for the display's upper left corner is found a address B8000.

#### The 4 Color High Resolution 640 X 200 (A) and
#### the 16 Color Medium Resolution 320 X 200 (9)
#### use 4 banks of 8000 bytes as follows:



```
Offset
From
B8000     <------160 Bytes------>

0000                                00 Scans
                                    (0,4,8,...,196)
1F3F

2000                                01 Scans
                                    (1,5,9,...,197)
3F3F

4000                                10 Scans
                                    (2,6,10,...,198)
5F3F

6000                                11 Scans
                                    (3,7,11,...,199)
7F3F
```

```
####    The  2 Color High Resolution 640 X 200 (6) and
####    the  4 Color Medium Resolution 320 X 200 (4/5)and
####    the 16 Color Low Resolution 160 X 200 (8)
####    use only 2 banks of 8000 bytes as follows:
```

```
Offset
From  B8000    <--80 Bytes-->
                ________________
      0000     |                |     Even Scans
               |                |     (0,2,4,6,8,...,198)
      1F3F     |________________|
               |________________|
      2000     |                |     Odd Scans
               |                |     (1,3,5,7,9,...,199)
      3F3F     |________________|
               |________________|
```

*350 Line Graphics Memory uses 4 banks of 8K bytes. The pixel information for the display's upper left corner is found at address B0000.

```
####    Hercules Graphics 720 x 348 (7H)
####    uses 4 banks of 8K bytes
```

```
             <-------- 90 Bytes ------->
              __________________________
    B0000    |                          |   00 SCANS
             |                          |   (0,4,8,...,344)
    B1E96    |__________________________|
             |__________________________|
    B2000    |                          |   10 SCANS
             |                          |   (1,5,9,...,345)
    B3E96    |__________________________|
             |__________________________|
    B4000    |                          |   01 SCANS
             |                          |  (2,6,10,...,346)
    B5E96    |__________________________|
             |__________________________|
    B6000    |                          |   11 SCANS
             |                          |  (3,7,11,...,347)
    B7E96    |__________________________|
```

2 COLOR MEDIUM RESOLUTION 640 X 200 GRAPHICS MODE (6)

The 2 Color Medium Resolution 640 x 200 Graphics mode may require a high resolution monitor for proper operation. Available in the IBM PC and IBM PCjr, this mode has the following characteristics:

Contains a maximum of 200 rows of 640 PELs.
Can display 2 of 16 possible colors.
Requires 16k bytes of read/write memory.
Formats 8 PELs per byte for each byte in the following manner:

7	6	5	4	3	2	1	0
PA0	PA0	PA0	PA0	PA0	PA0	PA0	PA0
First Dsply PEL	Secnd Dsply PEL	Third Dsply PEL	Forth Dsply PEL	Fifth Dsply PEL	Sixth Dsply PEL	Svnth Dsply PEL	Eghth Dsply PEL

4 COLOR MEDIUM RESOLUTION 640 X 200 GRAPHICS MODE (A)

The 4 Color Medium Resolution 640 X 200 Graphics mode may require a high resolution monitor for proper operation. Only supported on the IBM PCjr, this mode has the following characteristics:

Contains a maximum of 200 rows of 640 PELs
Can display 4 of 16 possible colors
Each pixel selects 1 of 4 colors
Requires 32K bytes of read/write memory
Formats 8 PELs per two bytes (1 even byte and 1 odd byte) in the following manner:

EVEN BYTES

7	6	5	4	3	2	1	0
PA0	PA0	PA0	PA0	PA0	PA0	PA0	PA0
First Dsply PEL	Second Dsply PEL	Third Dsply PEL	Forth Dsply PEL	Fifth Dsply PEL	Sixth Dsply PEL	Seventh Dsply PEL	Eighth Dsply PEL
PA1	PA1	PA1	PA1	PA1	PA1	PA1	PA1
7	6	5	4	3	2	1	0

ODD BYTES

(odd byte has most significant bit)
(PA = palette address)

16 COLOR MEDIUM RESOLUTION 320 X 200 GRAPHICS MODE (9)

The 16 Color Medium Resolution 320 X 200 Graphics mode works with all types of display devices. This mode is available on the IBM PCjr only and has the following characteristics:

Contains a maximum of 200 rows of 320 PELs
Can display 16 of 16 possible colors
Each pixel selects 1 of 16 colors
Requires 32K bytes of read/write memory
Formats 2 PELs per byte in the following manner:

7	6	5	4	3	2	1	0
PA3	PA2	PA1	PA0	PA3	PA2	PA1	PA0
First Display PEL				Second Display PEL			

16 COLOR LOW RESOLUTION 160 X 200 GRAPHICS MODE (8)

The 16 Color Medium Resolution 160 X 200 Graphics mode works with all types of display devices. This mode is available on the IBM PCjr only and has the following characteristics:

Contains a maximum of 200 rows of 160 PELs
Can display 16 of 16 possible colors
Each pixel selects 1 of 16 colors
Requires 16K bytes of read/write memory
Formats 2 PELs per byte in the following manner:

7	6	5	4	3	2	1	0
PA3	PA2	PA1	PA0	PA3	PA2	PA1	PA0
First Display PEL				Second Display PEL			

4 COLOR MEDIUM RESOLUTION 160 X 200 GRAPHICS MODE (4/5)

The 16 Color Medium Resolution 320 X 200 Graphics mode works with all types of display devices. This mode is available on the IBM PC and IBM PCjr modes and has the following characteristics:

Contains a maximum of 200 rows of 320 PELs
Can display 4 of 16 possible colors
Each pixel selects 1 of 4 colors
Requires 16K bytes of read/write memory
Formats 4 PELs per byte in the following manner:

7	6	5	4	3	2	1	0
PA3	PA2	PA1	PA0	PA3	PA2	PA1	PA0
First Display PEL		Second Display PEL		Third Display PEL		Fourth Display PEL	

HERCULES GRAPHICS (7H)

2 Color Hercules graphics works with a 350 line Monochrone display. This mode has the following characteristics:

720 PELs by 348 Rows
2 color Monochrome Graphics requires 32k bytes of RAM
Formats 8 PELs per bytes in the following manner :

7	6	5	4	3	2	1	0
PA0	PA0	PA0	PA0	PA0	PA0	PA0	PA0
First Dsply PEL	Second Dsply PEL	Third Dsply PEL	Forth Dsply PEL	Fifth Dsply PEL	Sixth Dsply PEL	Seventh Dsply PEL	Eighth Dsply PEL

16 COLOR HIGH RESOLUTION 640 X 200 GRAPHICS MODE (E)

The 16 Color High Resolution 640 x 200 Graphics mode works with a 200 line high resolution RGBI monitor. This mode has the following characteristics:

Contains 200 rows of 640 PELs
Can display 16 of 256K possible colors
Requires 64k bytes of read/write memory
Formats 2 PELs per byte for each byte in the following manner:

7	6	5	4	3	2	1	0
PA3	PA2	PA1	PA0	PA3	PA2	PA1	PA0
First Display PEL				Second Display PEL			

VIDEO I/O MAP

Hex Address	Register
0065	System Chip Select Register
FFE8	Video Configuration Register
03B4	CRTC Address Register
03B5	CRTC Data Register
03B8	Monochrome Control Register
03BA	Monochrome Status Register
03BF	Monochrome Configuration Switch
03D0	Not Used
03D1	Not Used
03D2	Not Used
03D3	Not Used
03D4	CRTC Address Register
03D5	CRTC Data Register
03D6	Not Used
03D7	Not Used
03D8	Mode Select Register
03D9	Color Select Register
03DA	Video Array Address & Status
03DB	Not Used
03DC	Not Used
03DD	Not Used ← yes it is; see I/O map
03DE	Video Array Data
03DF	CRT Processor Page Register

```
+-------+------------+-----------------------------------------+
|Address|Register    |           Bit Programming               |
|       |            |---+---+---+---+---+---+---+---+-----|
|(Hex)  |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+-------+------------+---+---+---+---+---+---+---+---+-----+
| 0065  |System Chip | ^ | X | X | ^ | ^ | ^ | ^ | X |Write|
|       |Selects     | | |   |   | | | | | | | | | |   | Only|
+-------+------------+-|-+---+---+-|-+-|-+-|-+-|-+---+-----+
|       |              |           |   |   |   |           |
|       |(Parallel OE)_|           |   |   |   |           |
|       |(Serial CS)_______________|   |   |   |           |
|       |(Disk CS)_______________________|   |   |           |
| VIDCS |Video CS  1 = video on____________|   |           |
|       |(Parallel CS)_________________________|           |
|       |                                                  |
|       |                                                  |
+-------+--------------------------------------------------+
```

```
+-------+------------+-----------------------------------------+
|Address|Register    |           Bit Programming               |
|       |            |---+---+---+---+---+---+---+---+-----|
|(Hex)  |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+-------+------------+---+---+---+---+---+---+---+---+-----+
| FFE8  |Video       | X | X | ^ | X | ^ | ^ | ^ | X |Write|
|       |Config.     |   |   | | |   | | | | | | | |   |Only |
+-------+------------+---+---+-|-+---+-|-+-|-+-|-+---+-----+
|       |                      |       |   |   |           |
|16BIT  |16 Bit CPU Memory = 1_|       |   |   |           |
| MC3   |Memory Configuration 3 _______|   |   |           |
| MC2   |Memory Configuration 2 ___________|   |           |
| MC1   |Memory Configuration 1 _______________|           |
|       |                                                  |
+-------+--------------------------------------------------+
```

3B4 CRTC ADDRESS REGISTER (see 3D4)

3B5 CRTC DATA REGISTER (see 3D5)

```
+--------+------------+-------------------------------------+
|Address |Register    |            Bit Programming          |
|        |            |---+---+---+---+---+---+---+---+-----|
|(Hex)   |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+------------+---+---+---+---+---+---+---+---+-----+
| 3B8    |Monochrome  | ^ | X | ^ | X | ^ | X | ^ | ^ |Write|
|        |Control     | | |   | | |   | | |   | | | | |Only |
+--------+------------+-|-+---+-|-+---+-|-+---+-|-+-|-+-----+
|        |              |       |       |       |   |       |
|MSTART  |Display start_|       |       |       |   |       |
|        |0 = B0000 1 = B8000   |       |       |   |       |
|MNOBLK  |Disable blink = 0 ____|       |       |   |       |
|MNOVID  |Disable video = 0 ____________|       |   |       |
|HRCULES |Hercules Graphics = 1 ________________|   |       |
|MTEXT   |Monochrome text   = 1 ____________________|       |
|        |                                                  |
+--------+--------------------------------------------------+
```

```
+--------+------------+-------------------------------------+
|Address |Register    |            Bit Programming          |
|        |            |---+---+---+---+---+---+---+---+-----|
|(Hex)   |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+------------+---+---+---+---+---+---+---+---+-----+
|3BA     |Monochrome  | ^ | ^ | X | X | ^ | X | X | ^ |Read |
|        |Status      | | | | |   |   | | |   |   | | |Only |
+--------+------------+-|-+-|-+---+---+-|-+---+---+-|-+-----+
|        |              |   |           |           |       |
|CVSYNC* |Vertical Sync_|   |           |           |       |
|HRGRP   | = 1 if 3B8       |           |           |       |
|        | bit 1 and 3BF    |           |           |       |
|        | bit 0 are both 1_|           |           |       |
|OUTG    |Video ________________________|           |       |
|CHSYNC  |Horizontal Sync __________________________|       |
|        |                                                  |
+--------+--------------------------------------------------+
```

```
+--------+------------+-------------------------------------+
|Address |Register    |            Bit Programming          |
|        |            |---+---+---+---+---+---+---+---+-----|
|(Hex)   |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+------------+---+---+---+---+---+---+---+---+-----+
|3BF     |Monochrome  | X | X | X | X | X | X | ^ | ^ |Write|
|        |Config.     |   |   |   |   |   |   | | | | |Only |
+--------+------------+---+---+---+---+---+---+-|-+-|-+-----+
|        |                                      |   |       |
|MSKCLMP |Mask B8000 and lock at B0000 = 0 _____|   |       |
|HERCOK  |Disable Hercules Graphics = 0 ____________|       |
|        |                                                  |
+--------+--------------------------------------------------+
```

CRTC CONTROLLER AND MISCELLANEOUS CONTROL REGISTERS

The CRTC and Miscellaneous control registers are accessed by two I/O commands. The two I/O commands function by first writing the desired index value to address hex 3D4 (3B4), and then writing the data to address hex 3D5 (3B5).

Index Register, Hex 3D4 (3B4): This register is read and write, and points to the specific data register addressed through hex 3D5 (3B5).

Bit	Function
7	Not Used
6	Not Used
5	Index5
4	Index4
3	Index3
2	Index2
1	Index1
0	Index0

Figure 1B CRTC Controller Index Register

The following is a list of the data registers and their functions.

Index (Hex)	Register Description
00	Horizontal Total
01	Horizontal Characters Displayed
02	Start Horizontal Sync
03	Sync Pulse Width
04	Vertical Total
05	Vertical Total Adjust
06	Vertical Characters Displayed
07	Start Vertical Sync
08	Reserved
09	Scan Lines per Character
0A	Cursor Start
0B	Cursor End
0C	Start of Screen High
0D	Start of Screen Low
0E	Cursor Position High
0F	Cursor Position Low
10	Mode Control
12	Character Generator Interface and Sync Polarity, or Display Sense
13	Character Font Pointer
21	Test Mode Register 2

HORIZONTAL TOTAL REGISTER, INDEX 00:

This register contains the total number of characters in the horizontal scan interval. The number consists of the total of the displayed and non-displayed characters. minus one. This register determines the frequency of the 'horizontal sync' signal.

HORIZONTAL DISPLAYED REGISTER, INDEX 01:

This register determines the total number of characters to be displayed during a horizontal line. This register must be loaded with a value that is less than the horizontal total register.

HORIZONTAL SYNC POSITION REGISTER, INDEX 02:

This register specifies the character position count at which the 'horizontal sync' signal becomes active. The specified value -1 must be programmed.

SYNC WIDTH REGISTER, INDEX 03:

This register specifies the pulse widths of the horizontal and vertical synchronization signals. The horizontal pulse width is programmed in units of character clocks. The vertical pulse width is programmed in units of the horizontal synchronization period. This register is programmed to match the display specifications.

```
+--------+-------------+-------------------------------------+
|Address |Register     |           Bit Programming           |
|        |             |---+---+---+---+---+---+---+---+-----|
|(Hex)   |             | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+-------------+---+---+---+---+---+---+---+---+-----+
|I03     |Sync Pulse   | ^ | ^ | ^ | ^ | ^ | ^ | ^ | ^ |Write|
|        |Width Reg.   | | | | | | | | | | | | | | | | |Only |
+--------+-------------+-|-+-|-+-|-+-|-+-|-+-|-+-|-+-|-+-----+
|        |               |   |   |   |   |   |   |   |       |
|        |Width VSync3 _|   |   |   |   |   |   |   |       |
|        |   VSync2 ________|   |   |   |   |   |   |       |
|        |   VSync1 ____________|   |   |   |   |   |       |
|        |   VSync0 ________________|   |   |   |   |       |
|        | Width HSync3 _____________|   |   |   |       |
|        |   HSync2 ________________________|   |   |       |
|        |   HSync1 ____________________________|   |       |
|        |   HSync0 ________________________________|       |
|        |                                                  |
+--------+--------------------------------------------------+
```

Figure 1-31. Sync Pulse Width Register

VERTICAL TOTAL REGISTER, INDEX 04:

This register contains the 8 bits for the total number of horizontal scan lines in the vertical scan interval. The total number consists of both the displayed and non-displayed scan lines minus 1. This register and the Vertical Total Adjust register determine the frequency of the 'vertical sync' signal.

VERTICAL TOTAL ADJUST REGISTER, INDEX 05:

This register is used to adjust the total number of horizontal scan lines in the vertical scanning interval. It allows for an odd number of horizontal lines (525 for 60 Hz).

```
+--------+------------+---------------------------------------+
|Address |Register    |           Bit Programming             |
|        |            |---+---+---+---+---+---+---+---+------|
|(Hex)   |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes |
+--------+------------+---+---+---+---+---+---+---+---+------+
|I05     |Vertical Tot| X | X | ^ | ^ | ^ | ^ | ^ | ^ |Write |
|        |Adjust Reg. |   |   | | | | | | | | | | | | |Only  |
+--------+------------+---+---+-|-+-|-+-|-+-|-+-|-+-|-+------+
|        |                        |   |   |   |   |   |       |
|        |VAdjust5 _______________|   |   |   |   |   |       |
|        |VAdjust4 ___________________|   |   |   |   |       |
|        |VAdjust3 _______________________|   |   |   |       |
|        |VAdjust2 ___________________________|   |   |       |
|        |VAdjust1 _______________________________|   |       |
|        |VAdjust0 ___________________________________|       |
+--------+----------------------------------------------------+
```

Figure 1-32. Vertical Total Adjust Register

VERTICAL DISPLAYED REGISTER, INDEX 06:

This register contains the 8 least-significant bits for the number of horizontal scan lines displayed during the vertical scan interval. The ninth bit is the inversion of bit 6 of the Mode Control register.

VERTICAL SYNC POSITION REGISTER, INDEX 07:

This register contains the 8 bits for the vertical scan line count. It determines when the 'vertical sync' signal becomes active. The specified value -1 must be programmed.

SCAN LINES PER CHARACTER REGISTER, INDEX 09:

This register determines the number of horizontal scan lines in a character row.

```
+--------+------------+---------------------------------------+
|Address|Register    |           Bit Programming              |
|       |            |---+---+---+---+---+---+---+---+-------|
|(Hex)  |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Notes|
+--------+------------+---+---+---+---+---+---+---+---+-------+
|I09     |Scan Lines | X | X | X | X | ^ | ^ | ^ | ^ |Write |
|        |per Char.  |   |   |   |   | | | | | | | | | | Only |
|        | Register  |   |   |   |   | | | | | | | | | |      |
+--------+------------+---+---+---+---+-|-+-|-+-|-+-|-+-------+
|        |                                |   |   |   |        |
|        |Row Size3 ______________________|   |   |   |        |
|        |Row Size2 __________________________|   |   |        |
|        |Row Size1 ______________________________|   |        |
|        |Row Size0 __________________________________|        |
+--------+----------------------------------------------------+
```

Figure 1-33. Scan Lines per Character Register

CURSOR START REGISTER, INDEX 0A:

Bits 3 through 0 in this register determine the horizontal scan line count at which the cursor outputbecomes active.

```
+--------+------------+---------------------------------------+
|Address|Register    |           Bit Programming             |
|       |            |---+---+---+---+---+---+---+---+------+
| (Hex) |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+------------+---+---+---+---+---+---+---+---+------+
|  I0A   |Cursor Start| X | X | 0 | X | ^ | ^ | ^ | ^ |Write|
|        |Register    |   |   | | |   | | | | | | | | | |Only |
+--------+------------+---+---+-|-+---+-|-+-|-+-|-+-|-+------+
|        |Reserved 0__________|         |   |   |   |        |
|        |Cursor Start3 _________________|   |   |   |        |
|        |Cursor Start2 _____________________|   |   |        |
|        |Cursor Start1 _________________________|   |        |
|        |Cursor Start0 _____________________________|        |
+--------+----------------------------------------------------+
```

Figure 1-34. Cursor Start Register

CURSOR END REGISTER, INDEX 0B:

This register determines the horizontal scan line count when the cursor output becomes inactive.

```
+-------+------------+-------------------------------------+
|Address|Register    |           Bit Programming           |
|       |            |---+---+---+---+---+---+---+---+-----+
|(Hex)  |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+-------+------------+---+---+---+---+---+---+---+---+-----+
|IOB    |Cursor End  | X | X | X | X | ^ | ^ | ^ | ^ |Write|
|       |            |   |   |   |   | | | | | | | | |Only |
+-------+------------+---+---+---+---+-|-+-|-+-|-+-|-+-----+
|       |Cursor End3 __________________|   |   |   |       |
|       |Cursor End2 ______________________|   |   |       |
|       |Cursor End1 __________________________|   |       |
|       |Cursor End0 ______________________________|       |
+-------+--------------------------------------------------+
```

Figure 1-35. Cursor End Register

START OF SCREEN HIGH REGISTER, INDEX 0C:

This register contains the 6 most-significant bits for the starting memory address of the video display buffer. Fourteen address bits determine the starting address. This register is initialized to a value of hex 00.

START OF SCREEN LOW REGISTER, INDEX 0D:

This register, together with the Start of Screen High register, gives the starting address of the display buffer. For all modes, this register is initialized to a value of hex 00.

CURSOR POSITION HIGH REGISTER, INDEX 0E:

This register contains the 4 most-significant bits for the cursor location.

```
+-------+------------+-------------------------------------+
|Address|Register    |           Bit Programming           |
|       |            |---+---+---+---+---+---+---+---+-----+
|(Hex   |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+-------+------------+---+---+---+---+---+---+---+---+-----+
|IOE    |Cursor Pos. | X | X | X | X | ^ | ^ | ^ | ^ |Write|
|       |High Reg.   |   |   |   |   | | | | | | | | |Only |
+-------+------------+---+---+---+---+-|-+-|-+-|-+-|-+-----+
|       |Cursor PositionB _____________|   |   |   |       |
|       |Cursor PositionA _________________|   |   |       |
|       |Cursor Position9 _____________________|   |       |
|       |Cursor Position8 _________________________|       |
+-------+--------------------------------------------------+
```

Figure 1-36. Cursor Position High Register

CURSOR POSITION LOW REGISTER, INDEX 0F:

This register contains the 8 least-significant bits for the location of the cursor. A value of hex 00 in both of these registers will locate the cursor in the upper left-hand corner. The cursor is not supported in any graphics mode.

MODE CONTROL REGISTER, INDEX 10:

```
+--------+------------+-------------------------------------+
|Address|Register     |           Bit Programming            |
|        |            |---+---+---+---+---+---+---+---+-----+
|(Hex)   |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+------------+---+---+---+---+---+---+---+---+-----+
|I10     |Mode Control| ^ | ^ | X | x | ^ | X | X | X |Write|
|        |Register    | | | | |   |   | | |   |   |   |Only |
+--------+------------+-|-+-|-+---+---+-|-+---+---+---+-----+
|NOCRTCW|Inhibit Write_|   |           |                     |
|        |Reserved = 0 ____|           |                     |
|        |                             |                     |
|        |Reserved = 1 _______________|                     |
|        |                                                  |
|        |                                                  |
+--------+--------------------------------------------------+
```

Figure 1-37. Mode Control, Write

WRITE

BIT 7 When set to 1, the inhibit write bit prevents any writes to the horizontal and vertical registers.

```
+--------+------------+-------------------------------------+
|Address|Register     |           Bit Programming            |
|        |            |---+---+---+---+---+---+---+---+-----+
| (Hex)  |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+------------+---+---+---+---+---+---+---+---+-----+
| I10    |Mode Control| ^ | X | ^ | ^ | ^ | ^ | X | X |Read |
|        |Register RD | | |   | | | | | | | | |   |   |Only |
+--------+------------+-|-+---+-|-+-|-+-|-+-|-+---+---+-----+
|HRESCK  |80 X 25 _____|       |   |   |   |                 |
|HRESAD  |Clock Select _______|   |   |   |                 |
|25CLK   |Clock __________________|   |   |                 |
|ALPHAMD|Alpha Mode ____________________|   |                 |
|SPCTL1  |Double Scan ________________________|                 |
|        |                                                  |
|        |                                                  |
+--------+--------------------------------------------------+
```

Figure 1-38. Mode Control, Read

CHARACTER GENERATOR INTERFACE AND SYNC POLARITY REGISTER, INDEX 12:

The register controls the character font ROM.

```
+--------+------------+---------------------------------------+
|Address|Register    |              Bit Programming           |
|       |            |---+---+---+---+---+---+---+---+-------+
| (Hex) |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Notes|
+--------+------------+---+---+---+---+---+---+---+---+-------+
|  I12  |Character   | X | X | ^ | ^ | X | X | X | X | Write|
|       |Gen. I/F &  |   |   | | | | |   |   |   |   | Only |
|       |Sync        |   |   | | | | |   |   |   |   |      |
|       |Polarity    |   |   | | | | |   |   |   |   |      |
+--------+------------+---+---+-|-+-|-+---+---+---+---+-------+
|       |                       |   |                          |
|SWPFONT|Swap Active Font ______|   |                          |
|512CHAR|Enable 512 Characters ____|                          |
+--------+-----------------------------------------------------+
```

Figure 1-40. Character Generator Interface and Sync Polarity Register

BIT 5 This bit selects the font page that is used as the font table. When set to 1, font page 1 is selected; when clear to 0, font page 0 is selected.

BIT 4 When this bit is set to 1, 512 character codes are displayable in the text modes. Bit 3 of the attribute byte then determines the font page when displaying the character. When this bit is set to 1, only eight foreground colors are supported. When this bit is cleared to 0, only 256 character codes are displayed, and bit 5 of this register determines the active font. See font table, end of video section.

CHARACTER FONT REGISTER, INDEX 13:

Bit 4 of this register is used to select between the 200 line RGBI character set and the 350 line monochrome character set. A 0 in bit 4 selects the 200 line character set.

TEST MODE REGISTER 1, INDEX 20

This register is reserved for future test purposes and should be cleared to zero at all times.

TEST MODE REGISTER 2, INDEX 21

This register is used only during manufacturing tests and must be cleared to zero at all other times.

```
+-------+-----------+---------------------------------------+
|Address|Register   |            Bit Programming            |
|       |           |---+---+---+---+---+---+---+---+-------+
| (Hex) |           | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Notes |
+-------+-----------+---+---+---+---+---+---+---+---+-------+
|  I21  |   Test    | X | X | X | X | ^ | ^ | ^ | ^ | Write |
|       | Register  |   |   |   |   | | | | | | | | | Only  |
+-------+-----------+---+---+---+---+-|-+-|-+-|-+-|-+-------+
|       |                             |   |   |   |         |
|       |                             |   |   |   |         |
|       |1 = Clock 6845 & Blink       |   |   |   |         |
|       |    with 28 MHz and mux      |   |   |   |         |
|       |    out blink _______________|   |   |   |         |
|       |1 = Enable a test reset _________|   |   |         |
|       |1 = Force access to active __________|   |         |
|       |1 = Mux out Dotclk on OUTB ______________|         |
+-------+---------------------------------------------------+
```

VIDEO ARRAY REGISTERS

The following registers can be accessed by writing their Hex address to 3DA and their Hex Data to 3DE.

Hex Address	Video Array Register
01	Palette Mask
02	Border Color
03	Mode Control
05	Extended RAM Page Register
08	Monitor / Mode Selection
10 - 1F	Palette Registers

ARRAY PALETTE MASK REGISTER

```
+-------+------------+-------------------------------------+
|Address|Register    |           Bit Programming           |
|       |            |---+---+---+---+---+---+---+---+-----+
| (Hex) |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+-------+------------+---+---+---+---+---+---+---+---+-----+
|  I01  |Palette     | X | X | X | X | ^ | ^ | ^ | ^ |Write|
|       | Mask       |   |   |   |   | | | | | | | | |Only |
+-------+------------+---+---+---+---+-|-+-|-+-|-+-|-+-----+
|       |                              |   |   |   |       |
|MSK(3) |Palette Mask 3 _______________|   |   |   |       |
|MSK(2) |Palette Mask 2 ___________________|   |   |       |
|MSK(1) |Palette Mask 1 _______________________|   |       |
|MSK(0) |Palette Mask 0 ___________________________|       |
|       |                                                  |
+-------+--------------------------------------------------+
|When bits 0-3 are 0, they force the appropriate palette   |
|address to be 0 regardless of the incoming color          |
|information. This feature allows some of the color        |
|information in memory to hidden until it is required.     |
|                                                          |
+----------------------------------------------------------+
```

ARRAY BORDER COLOR

```
+-------+------------+-------------------------------------+
|Address|Register    |           Bit Programming           |
|       |            |---+---+---+---+---+---+---+---+-----+
| (Hex) |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+-------+------------+---+---+---+---+---+---+---+---+-----+
|  I02  |Border      | X | X | 0 | X | ^ | ^ | ^ | ^ |Write|
|       |Color       |   |   | ^ |   | | | | | | | | |Only |
+-------+------------+---+---+-|-+---+-|-+-|-+-|-+-|-+-----+
|       |Reserved = 0 _________|       |   |   |   |       |
| BORI  |I (Iten) Border Color Select__|   |   |   |       |
| BORR  |R (Red) Border Color Select_______|   |   |       |
| BORG  |G (Green) Border Color Select_________|   |       |
| BORB  |B (Blue) Border Color Select______________|       |
+-------+--------------------------------------------------+
|A combination of bits 0-3 selects the screen border as one|
|of 16 colors, as listed in Table 1 "Available Color Table"|
|at the beginning of this section.                         |
+----------------------------------------------------------+
```

ARRAY MODE CONTROL REGISTER

```
+--------+-------------+-------------------------------------+
|Address|Register      |           Bit Programming           |
|        |             |---+---+---+---+---+---+---+---+-----+
| (Hex)  |             | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+-------------+---+---+---+---+---+---+---+---+-----+
|  I03   |Mode         | X | X | X | ^ | ^ | ^ | 0 | X |Write|
|        |Control      |   |   |   | | | | | | | ^ |   |Only |
+--------+-------------+---+---+---+-|-+-|-+-|-+-|-+---+-----+
|        |                           |   |   |   |           |
|C16COL  |1 for 16 Color Modes_______|   |   |   |           |
|        |                               |   |   |           |
|C4COLHR|1 for 4 color 640x200 Mode_____|   |   |           |
|        |                                   |   |           |
|BORENB  |Enables the Border color register_|   |           |
|        |For PC compatibility, this bit should |           |
|        |be set to 0. For PCjr compatibility,  |           |
|        |this bit should be 1.                 |           |
|        |                                       |           |
|        |Reserved for future implementations.__|           |
|        |Must always be set to 0                            |
+--------+---------------------------------------------------+
```

```
+--------+-----------+---------------------------------------+
|Address|Register    |            Bit Programming            |
|        |           |---+---+---+---+---+---+---+---+-------|
| (Hex)  |           | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes  |
+--------+-----------+---+---+---+---+---+---+---+---+-------+
|  I05   |Extended   | ^ | X | X | X | X | X | X | ^ |Write  |
|        |RAM  Page  | | |   |   |   |   |   |   | | |Only   |
|        |Register   | | |   |   |   |   |   |   | | |       |
+--------+-----------+-|-+---+---+---+---+---+---+-|-+-------+
|        |             |                           |         |
|SWTCK   |Select Video_|                           |         |
|        |Clock 0 = 28.6 (CGA monitor)             |         |
|        |1 = 32.5 (Mono monitor)                  |         |
|        |                                         |         |
|EXTADR  |Extended addressing mode bit ____________|         |
|        |                                                   |
+------------------------------------------------------------+
--------+-----------+-------------------------------------+
|Address|Register   |           Bit Programming           |
|       |           |---+---+---+---+---+---+---+---+-----|
| (Hex) |           | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+-------+-----------+---+---+---+---+---+---+---+---+-----+
|  I08  |Monitor    | X | X | X | ^ | X | X | ^ | X |Write|
|       |Mode       |   |   |   | | |   |   | | |   |Only |
|       |Selection  |   |   |   | | |   |   | | |   |     |
+-------+-----------+- -+---+---+-|-+---+---+-|-+---+-----+
|       |                         |           |           |
|MONO   |1=350 line mono display__|           |           |
|MODEE  |1=Graphics mode E _______________|           |
|       |                                                 |
+---------------------------------------------------------+
```

ARRAY PALETTE REGISTERS

There are sixteen 4 bit wide palette registers implemented by a 16x4 bit RAM. These registers are Read/Write. Their address in the video array are from 10 - 1F Hex, and can be used to re-define any color.

To load the palette, write the hex address to the Video Array Register at 3DA hex. Then, the new palette color is written to 3DE hex.

Palette address 10 hex is accessed whenever the color code from memory is a hex 0, and address 11 hes is accessed whenever the color code from memory is a hex 1, and so forth. A description of the color codes can be found in the "Available Colors Table" at the beginning of this section.

Note: **The palette address can be 'masked' by using the palette mask register.**

The following is a description of the register's bit functions:

Bit Number	Function
0	Blue
1	Green
2	Red
3	Intensity

When loading the palette, the video is 'disabled' and the color viewed on the screen is the data contained in the register being addressed by the processor.

When the program has completed loading the palette, it must change the hex address to some address less than 10 hex for video to be 'enabled, again.

If a programmer does not wish a user to see the adverse effects of loading the palette, the palette should be loaded during the vertical retrace time. The program must modify the palette and change the address to less than 10 hex within the vertical retrace time. A vertical retrace interrupt and a status bit are provided to facilitate this procedure.

In two color modes, the palette is defined by using one bit, PA0, with the following logic:

PA0	Function
0	Palette Register 0
1	Palette Register 1

In four color modes, the palette is defined by using two bits, PA1 and PA0, with the following logic:

PA1	PA0	Function
0	0	Palette Register 0
0	1	Palette Register 1
1	0	Palette Register 2
1	1	Palette Register 3

In sixteen color modes, the palette is defined by using four bits, PA3, PA2, PA1 and PA0, with the following logic:

PA3	PA2	PA1	PA0	Function
0	0	0	0	Palette Register 0
0	0	0	1	Palette Register 1
0	0	1	0	Palette Register 2
0	0	1	1	Palette Register 3
0	1	0	0	Palette Register 4
0	1	0	1	Palette Register 5
0	1	1	0	Palette Register 6
0	1	1	1	Palette Register 7
1	0	0	0	Palette Register 8
1	0	0	1	Palette Register 9
1	0	1	0	Palette Register 10
1	0	1	1	Palette Register 11
1	1	0	0	Palette Register 12
1	1	0	1	Palette Register 13
1	1	1	0	Palette Register 14
1	1	1	1	Palette Register 15

```
+-------+------------+-------------------------------------+
|Address|Register    |           Bit Programming           |
|       |            |---+---+---+---+---+---+---+---+-----+
| (Hex) |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+-------+------------+---+---+---+---+---+---+---+---+-----+
|  3D8  |Mode        | X | X | ^ | ^ | ^ | ^ | 0 | ^ |Write|
|       |Control     |   |   | | | | | | | | | ^ | | |Only |
+-------+------------+---+---+-|-+-|-+-|-+-|-+-|-+-|-+-----+
|ENA    |                      |   |   |   |   |   |       |
|BLINK  |Set to 1 for _________|   |   |   |   |   |       |
|       |blink if attribute bit    |   |   |   |   |       |
|       |7 is  set.                |   |   |   |   |       |
|       |Set to 0 = 16 Color       |   |   |   |   |       |
|       |background. A 1 selects 8 |   |   |   |   |       |
|       |color background          |   |   |   |   |       |
|       |                          |   |   |   |   |       |
|HRESAD |Set to 1 640X200 2 or 4___|   |   |   |   |       |
|       |color graphics                |   |   |   |       |
|       |                              |   |   |   |       |
|VIDEN  |                              |   |   |   |       |
|       |1 = enable video display______|   |   |   |       |
|       |                                  |   |   |       |
|BW     |Black and White select. A ________|   |   |       |
|       |different color palette is selected by|   |       |
|       |this bit in 320 X 200 4 color mode.   |   |       |
|       |                                      |   |       |
|GRPH   |Set to 1 for Graphics ________________|   |       |
|       |and 0 for Alpha Numeric Mode              |       |
|       |                                          |       |
|HRESEK |High Resolution Dot Clock ________________|       |
|     C |A 0 selects lower speed for 40 character text     |
|       |or low resolution graphics. A 1 selects high speed|
|       |for for 80 character text or high resolution      |
|       |graphics                                          |
+-------+--------------------------------------------------+
```

shd be Color Select (see p. 11)

```
+--------+------------+---------------------------------------+
|Address |Register    |            Bit Programming            |
|        |            |---+---+---+---+---+---+---+---+-----+
| (Hex)  |            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+------------+---+---+---+---+---+---+---+---+-----+
|  3D9   |Mode        | X | X | ^ | ^ | ^ | ^ | 0 |   |Write|
|        |Control     |   |   | | | | | | | | | ^ | ^ |Only |
+--------+------------+---+---+-|-+-|-+-|-+-|-+-|-+-|-+-----+
|COLSEL  |320X200 4 color blue _|   |   |   |   |   |         |
|        |control                   |   |   |   |   |         |
|        |                          |   |   |   |   |         |
|BACK    |                          |   |   |   |   |         |
|GROUNDI |Alpha background _________|   |   |   |   |         |
|        |/320 Graphics Foreground      |   |   |   |         |
|        |intensity. When blink is      |   |   |   |         |
|        |enabled in Alpha mode, this   |   |   |   |         |
|        |bit is used to select         |   |   |   |         |
|        |intensity.                    |   |   |   |         |
|        |In the 320 X 200 4 color mode |   |   |   |         |
|        |it  selects the intensity of  |   |   |   |         |
|        |the  foreground               |   |   |   |         |
|        |                              |   |   |   |         |
|OVER    |                              |   |   |   |         |
|SCANI   |In Alpha Mode Screen Border __|   |   |   |         |
|        |intensity                         |   |   |         |
|        |In 320 X 200 4 color mode sets    |   |   |         |
|        |Background intensity if PA0       |   |   |         |
|        |and PA1 = 0                       |   |   |         |
|        |In 640 X 200 2 color selects      |   |   |         |
|        |Foreground Intensity              |   |   |         |
|        |                                  |   |   |         |
|OVER    |                                  |   |   |         |
|SCANR   |In Alpha Mode Screen Border Red___|   |   |         |
|        |In 320 X 200 4 color mode sets        |   |         |
|        |Background Red if PA0 and PA1 = 0     |   |         |
|        |In 640X200 2 color selects Foreground |   |         |
|        |Red                                   |   |         |
|        |                                      |   |         |
|        |                                      |   |         |
|OVER    |                                      |   |         |
|SCANG   |In Alpha Mode Screen Border Green_____|   |         |
|        |In 320 X 200 4 color mode sets            |         |
|        |Background Green if PA0 and PA1 = 0       |         |
|        |In 640X200 2 color selects Foreground     |         |
|        |Green                                     |         |
|        |                                          |         |
|        |                                          |         |
|OVER    |                                          |         |
|SCANB   |In Alpha Mode Screen Border Blue__________|         |
|        |In 320 X 200 4 color mode sets                      |
|        |Background Blue if PA0 and PA1 = 0                  |
|        |In 640X200 2 color selects Foreground               |
|        |Blue                                                |
+--------+----------------------------------------------------+
+-------+-----------+--------------------------------------+
|Address|Register   |             Bit Programming          |
|       |           |---+---+---+---+---+---+---+---+------|
| (Hex) |           | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes |
+-------+-----------+---+---+---+---+---+---+---+---+------+
|  3DA  |Video      | X | X | ^ | X | ^ | X | X | ^ |Read  |
|       |Status     |   |   | | |   | | |   |   | | |Only  |
+-------+-----------+---+---+-|-+---+-|-+---+---+-|-+------+
|       |                     |       |           |        |
|CHSYNV |1 during horiz. sync_|       |           |        |
|       |                     |       |           |        |
|CVSYNV |Equal 1 during vertical sync_|           |        |
|       |                                         |        |
|DISPENB|Equal 0 when the display is active_______|        |
|       |                                                  |
+----------------------------------------------------------+
+--------+-------------+-----------------------------------------+
|Address|Register      |            Bit Programming              |
|        |             |---+---+---+---+---+---+---+---+-----+
| (Hex)  |             | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |Notes|
+--------+-------------+---+---+---+---+---+---+---+---+-----+
|  3DF   |CRT/Proc.    | ^ | ^ | ^ | ^ | ^ | ^ | ^ | ^ |Write|
|        |Page Reg.    | | | | | | | | | | | | | | | | |Only |
+--------+-------------+-|-+-|-+-|-+-|-+-|-+-|-+-|-+-|-+-----+
|        |               |   |   |   |   |   |   |   |       |
|ADRM1   |               |   |   |   |   |   |   |   |       |
|(**)    |Video Address_|   |   |   |   |   |   |   |       |
|        |Mode 1 with Reg.   |   |   |   |   |   |   |       |
|        |3DE index 5 bit 0  |   |   |   |   |   |   |       |
|        |selects Video      |   |   |   |   |   |   |       |
|        |Address            |   |   |   |   |   |   |       |
|        |supplied to RAM    |   |   |   |   |   |   |       |
|        |                   |   |   |   |   |   |   |       |
|ADRM0   |                   |   |   |   |   |   |   |       |
|(**)    |Video Address______|   |   |   |   |   |   |       |
|        |Mode 0 with            |   |   |   |   |   |       |
|        |Register 3DE index     |   |   |   |   |   |       |
|        |5 bit 0 selects Video  |   |   |   |   |   |       |
|        |Address supplied to    |   |   |   |   |   |       |
|        |RAM                    |   |   |   |   |   |       |
|        |                       |   |   |   |   |   |       |
|CPUPG2  |Processor Page 2_______|   |   |   |   |   |       |
|CPUPG1  |Processor Page 1___________|   |   |   |   |       |
|CPUPG0  |Processor Page 0_______________|   |   |   |       |
|        |                                   |   |   |       |
|CRTPG2  |CRT Page 2_________________________|   |   |       |
|CRTPG1  |CRT Page 1_____________________________|   |       |
|CRTPG0  |CRT Page 0_________________________________|       |
+--------+---------------------------------------------------+
|The processor page bits are combined with the CPU address |
|to select the 32K segment of memory accessed at B8000     |
|hex. If an odd pagenumber is selected, the window is      |
|reduced to 16K.                                           |
|                                                          |
|The CRT page bits select the 16K Page used by the Video.  |
|In 32K modes bit 0 is ignored.                            |
|                                                          |
|Note (**): These bits are used in conjunction with        |
|register 3DE index 5, bit 0 to select the video addresses |
|to RAM. See the Video Addressing Modes Table. Also the    |
|Graphics control bit ,3D8 bit 1, (GRPH) will force the    |
|same condition as ADRM0.                                  |
+----------------------------------------------------------+
```

CONTROL BIT PROGRAMMING CHART

MODE	MONO	GRPH	HRCULES	HERCOK	MTEXT	HRESAD	HRESCK	C4COLHR	C16COL	MODEE
0	0	0	X	X	X	0	0	X	X	X
2	0	0	X	X	X	0	1	X	X	X
4	0	1	X	X	X	0	0	0	0	0
6	0	1	X	X	X	1	0	0	0	0
7	1	X	X	X	1	X	X	X	X	X
7H	1	X	1	1	0	X	X	X	X	0
8	0	1	X	X	X	0	0	0	1	0
9	0	1	X	X	X	0	1	0	1	0
A	0	1	X	X	X	1	1	1	0	0
E	0	1	X	X	X	1	1	0	1	1
	3DE I8	3D8	3B8	3BF	3B8	3D8	3D8	3DE I3	3DE I3	3DE I8

CRTC PROGRAMMING TABLE

CRTC INDEX VALUE	CRTC REGISTER	VIDEO MODES						
		0	2	4 5 6 8	9 A	7	7H	E
00	HORIZONTAL TOTAL	38	71	38	71	61	35	71
01	HORIZONTAL DISPLAYED	28	50	28	50	50	2D	50
02	HORIZONTAL SYNC POS	2D	5A	2D	5A	52	2E	5A
03	HORIZONTAL SYNC WIDTH	F8	FE	F8	FE	FF	F7	FE
04	VERTICAL TOTAL	1C	1C	7F	3F	19	5B	FF
05	VERTICAL ADJUST	01	01	06	06	06	02	06
06	VERTICAL DISPLAYED	19	19	64	32	19	57	C8
07	VERTICAL SYNC POS	1A	1A	70	38	19	57	E2
09	SCAN LINES /CHARACTER	08	08	~~02~~ Ø1	03	0D	03	00
0A	CURSOR START	06	06	XX	XX	0B	XX	XX
0B	CURSOR END	07	07	XX	XX	0C	XX	XX
0C	SCREEN HIGH	00	00	00	00	00	00	00
0D	SCREEN LOW	00	00	00	00	00	00	00
0E	CURSOR HIGH	00	00	XX	XX	00	XX	XX
0F	CURSOR LOW	00	00	XX	XX	00	XX	XX
10	MODE CONTROL	18	18	18	18	18	18	18
12	CHAR GEN	07	07	07	07	07	07	07
13	CHAR FNT PTR	00	00	XX	XX	10	XX	XX

Mode 4, 5, 6, 8 actual BIOS value is Ø1, as indicated (from video initialization table).

VIDEO/SYSTEM MEMORY ADDRESS MAP

FFE8 BITS 3 2 1	VIDEO/SYSTEM MEMORY START ADDRESS	VIDEO/SYSTEM MEMORY ADDRESS RANGE
0 0 0	00000	00000 - 1FFFF
0 0 1	20000	20000 - 3FFFF
0 1 0	40000	40000 - 5FFFF
0 1 1	60000	60000 - 7FFFF
1 0 0	80000	80000 - 9FFFF
1 1 1	-	No System Memory

VIDEO MEMORY ADDRESSING MODES

03DAI5 BIT 0 EXTADR	03DF BIT 7 ADRM1	03DF BIT 6 ADRM0	VIDEO MEMORY ORGANIZATION
0	0	0	1 - 16K segment of memory (8 pages)
0	0	1	2 - 8K segments of memory (8 pages) Switched on RA0
0	1	0	2 - 16K segments of memry (4 pages) Switched on RA0
0	1	1	4 - 8K segments of memory (4 pages) Switched on RA0 & RA1
1	0	0	1 - 32K (64K)segment of memory (4/2 pages)

Mode E uses the larger memory size (64K).

For 8 Page Modes the Video Pages are selected by CRTPG[2:0].
For 4 Page Modes the Video Pages are selected by CRTPG[1:0].
For 2 Page Modes the Video Pages are selected by CRTPG2.

Notes on video memory:

A write to port 65h with bit 2 clear disables access to video memory through addresses A0000-BFFFFh. It does not disable access through lower addresses if that has been enabled at port FFE8h. Ports 3B4-3DFh are disabled, but port FFE8h is not.

If bit 6 or 7 at port FFEAh is cleared, access to main memory in the range 80000-9FFFFh is disabled. These addresses correspond to the 128k video memory upgrade.

Bits 1-3 at port FFE8h determine whether video memory is addressable at locations below A0000h, and if so, at which address. If these bits are 100b, video memory is addressed at 80000-9FFFFh, as well as at A0000-BFFFFh. If 111b, video memory is not addressable below A0000h.

ASCII code (decimal)	IBM character (font Ø)	Tandy character (font 1)
Ø	nul	nul
1	☺	☺
2	☺ reversed	☺ reversed
3	♥	♥
4	♦	♦
5	♣	♣
6	♠	♠
7	•	•
8	• reversed	• reversed
9	○ (circle)	○ (circle)
1Ø	○ reversed	○ reversed
11	♂	♂
12	♀	♀
13	♪	♪
14	♫	♫
15	☼	☼
16	►	►
17	◄	◄
18	↕	↕
19	‼	‼
2Ø	¶	¶
21	§	§
22	▬	▬
23	↨	↨
24	↑	↑
25	↓	↓
26	→	→
27	←	←
28	∟	∟
29	↔	↔
3Ø	▲	▲
31	▼	▼
32	space	space
33	!	!
34	"	"
35	#	#
36	$	$
37	%	%
38	&	&
39	' (single quote)	' (single quote)
4Ø	(	(
41	)	)
42	*	*
43	+ (plus)	+ (plus)
44	, (comma)	, (comma)
45	- (hyphen)	- (hyphen)
46	. (period)	. (period)
47	/	/
48	Ø (zero)	Ø (zero)
49	1	1
5Ø	2	2
51	3	3
52	4	4
53	5	5
54	6	6
55	7	7
56	8	8
57	9	9
58	:	:
59	;	;
6Ø	<	<
61	= (equals)	= (equals)
62	>	>
63	?	?
64	@	@

See register 305 index 12

ASCII code (decimal)	IBM character (font Ø)	Tandy character (font 1)
65	A	A
66	B	B
67	C	C
68	D	D
69	E	E
7Ø	F	F
71	G	G
72	H	H
73	I	I
74	J	J
75	K	K
76	L	L
77	M	M
78	N	N
79	O (letter O)	O (letter O)
8Ø	P	P
81	Q	Q
82	R	R
83	S	S
84	T	T
85	U	U
86	V	V
87	W	W
88	X	X
89	Y	Y
9Ø	Z	Z
91	[	[
92	\	\
93	]	]
94	^	^
95	_ (underscore)	_ (underscore)
96	`	`
97	a	a

ASCII code	IBM character	Tandy character
98	b	b
99	c	c
1ØØ	d	d
1Ø1	e	e
1Ø2	f	f
1Ø3	g	g
1Ø4	h	h
1Ø5	i	i
1Ø6	j	j
1Ø7	k	k
1Ø8	l (letter l)	l (letter l)
1Ø9	m	m
11Ø	n	n
111	o (letter o)	o (letter o)
112	p	p
113	q	q
114	r	r
115	s	s
116	t	t
117	u	u
118	v	v
119	w	w
12Ø	x	x
121	y	y
122	z	z
123	{	{
124	¦ (pipe mark)	¦ (pipe mark)
125	}	}
126	~	~
127	⌂	⌂
128	Ç (upper case)	Ç (upper case)
129	ü (lower case)	ü (lower case)

ASCII code (decimal)	IBM character (font Ø)	Tandy character (font 1)
13Ø	é	é
131	â	â
132	ä	ä
133	à	à
134	å	å
135	ç (lower case)	ç (lower case)
136	ê	ê
137	ë	ë
138	è	è
139	ï (lower case)	ï (lower case)
14Ø	î	î
141	ì	ì
142	Ä	Ä
143	Å	Å
144	É	É
145	æ	æ
146	Æ	Æ
147	ô (lower case)	ô (lower case)
148	ö (lower case)	ö (lower case)
149	ò (lower case)	ò (lower case)
15Ø	û (lower case)	û (lower case)
151	ù (lower case)	ù (lower case)
152	ÿ	ÿ
153	Ö (upper case)	Ö (upper case)
154	Ü (upper case)	Ü (upper case)
· 155	¢ (cent)	ø (lower case)
156	£	£
· 157	¥ (yen)	Ø (upper case)
· 158	₧	x (one scan line higher than letter x)
· 159	ƒ (forte)	ƒ (forte, one scan line shorter, no descender)

ASCII code	IBM charater	Tandy character
16Ø	á	á
161	í	í
162	ó (lower case)	ó (lower case)
163	ú (lower case)	ú (lower case)
164	ñ	ñ
165	Ñ	Ñ
166	ª	ª
167	º	º
168	¿	¿
· 169	⌐	® (registered trademark)
17Ø	¬ (logical not)	¬ (logical not)
171	½	½
172	¼	¼
173	¡	¡
174	«	«
175	»	»
176	░	░
177	▒	▒
178	▓	▓
179	│ (vertical bar)	│ (vertical bar)
18Ø	┤	┤
· 181	╡	Á
· 182	╢	Â
· 183	╖	À
· 184	╕	© (copyright)
185	╣	╣
186	║	║
187	╗	╗
188	╝	╝
· 189	╜	¢ (one scan line taller than 155, font Ø)

ASCII code (decimal)	IBM character (font Ø)	Tandy character (font 1)
·19Ø	╛	¥ (yen)
191	┐ (box corner)	┐ (box corner)
192	└ (box corner)	└ (box corner)
193	┴	┴
194	┬	┬
195	├	├
196	─ (horiz. line)	─ (horiz. line)
197	┼	┼
·198	╞	ã
·199	╟	Ã
2ØØ	╚	╚
2Ø1	╔	╔
2Ø2	╩	╩
2Ø3	╦	╦
2Ø4	╠	╠
2Ø5	═ (double horiz. line)	═ (double horiz. line)
2Ø6	╬	╬
·2Ø7	╧	¤
·2Ø8	╨	ð
·2Ø9	╤	Đ
·21Ø	╥ (box part)	Ê
·211	╙	Ë
·212	╘	È
·213	╒ (box corner)	ı (lower case i, no dot)
·214	╓	Í (slightly different from 161)
·215	╫	Î
·216	╪	Ï
ASCII code	IBM character	Tandy character
217	┘	┘
218	┌	┌
219	█	█
22Ø	▄ (lower half)	▄ (lower half)
·221	▌ (left half)	¦ (pipe mark)
·222	▐ (right half)	Ì
223	▀ (upper half)	▀ (upper half)
·224	α (alpha)	Ó (upper case)
225	β	β
·226	Γ (gamma)	Ô (upper case)
·227	π (pi)	Ò (upper case)
·228	Σ (sigma)	õ (lower case)
·229	σ (sigma)	Õ (upper case)
·23Ø	µ (mu)	µ (mu, one scan line lower)
·231	τ (tau)	Þ (upper case)
·232	Φ (phi)	þ (lowercase)
·233	Θ (theta)	Ú (uppercase)
·234	Ω	Û (upper case)
·235	δ	Ù (upper case)
·236	∞	ý
·237	φ (null set, different from 155 & 157 font 1)	Ý
·238	∈	¯ (line above)
·239	∩	´ (acute accent)
·24Ø	≡	- (hyphen)
241	±	±
·242	≥	‗ (double line below)

ASCII code (decimal)	IBM character (font Ø)	Tandy character (font 1)
·243	≤	3/4
·244	⌠ (top half of integral sign)	¶ (same as 2Ø)
·245	⌡ (lower half of integral sign)	§ (same as 21)
246	÷	÷
·247	≈ (approx. equal)	¸ (cedilla)
248	° (degree)	° (degree)
·249	• (center dot, large)	¨ (dierésis)
·25Ø	· (center dot, small)	· (center dot, small, one scan line higher)
·251	√ (radical)	^{1} (1 superscript)
·252	^{n} (n superscript ←note tables say η (eta) should be here)	^{3} (3 superscript)
253	^{2} (2 superscript)	^{2} (2 superscript)
254	■	■
255	nul	nul

IBM chars are Code Page 437

Tandy chars are Code Page 850

JACKSBORO SPECIFICATION
jmp 05-26-88

Table of Contents

1.0 GENERAL

1.1 Functional Description

The PSSJ Tandy ASIC is contained in a 68 pin PLCC package, and comprises the Printer port, a Serial (RS232) port, the Sound function, and the Joystick function of the Tandy 1000 computers.

2.0 PIN LIST

PIN NAME	PIN NO.	DRIVE	DESCRIPTION
VCC	1,35	--	Power inputs
VBB	59	--	Analog Power input
GND	18,52	--	Grounds
RST	25	TTL in	System reset signal, active high.
CLK14M	2	TTL in	Clock signal input, 14.31313 MHz, 50% duty cycle.
CLK2IN	37	TTL in	Clock signal input, either 24 MHz or 1.8432 MHz, 50% duty cycle.
IOD0 - IOD7	14,15,16,17 19,20,21,22	DS1218, 8 mA TS	Eight bit peripheral data bus intended to drive 5 XT type I/O slots, as well as all on board peripherals.
IOR-	10	TTL in	CPU/DMA I/O Read signal, active low. System control line.
IOW-	11	DS1218	CPU/DMA I/O Write signal, active low. System control line.
A0 - A2, A7	6,7,8,9	TTL in	System address lines.
CS0 - CS2	3,4,5	TTL in	Address decode inputs.
PINT	12	2 mA TS	Printer Interrupt, tristate.

PIN NAME	PIN NO.	DRIVE	DESCRIPTION
SINT	13	2 mA TS	Serial Interrupt, tristate.
PPITIM	68	TTL in	Low frequency sound input.
AUDIO_IN	55	An in.	Analog audio input, 1 V p-p.
SND_OUT	57	An out	Analog audio output, 2 V p-p.
GAIN_OUT	56	An out	Analog audio output, 2 V p-p
DRQ	23	2 mA TS	Data request for DMA operations, tristate.
TC	27	TTL in	Terminal Count input.
DACK1	26	TTL in	Data acknowledge for DMA ops.
WAIT-	24	2 mA OD	Sound chip wait output, open drain.
JPOS1 - JPOS4	60,61,62,63	DS1218	Digital joystick position input.
JSW1 - JSW4	64,65,66,67	DS1218	Digital joystick switch inputs.
DAC_OUT	58	An out	Analog DAC output for external integration, comparison with joystick voltages.
PD0 - PD7	51,50,49,48 47,46,45,44	DS1218 4 mA TS	Printer data inputs/outputs.
INIT	40	4 mA OD	Printer initialization output.
AFXT-	39	4 mA OD	Printer auto feed output.
STROBE-	38	4 mA OD	Printer strobe output.
ACK-	41	TTL in	Printer acknowledge input.

PIN NAME	PIN NO.	DRIVE	DESCRIPTION
PE	43	TTL in	Printer paper empty input.
SLCTIN-	53	TTL in	Printer select input.
BUSY-	42	TTL in	Printer busy input.
FAULT-	54	TTL in	Printer fault input.
DTR-	36	2 mA	RS232 data terminal ready output.
RTS-	33	2 mA	RS232 request to send output.
TXD-	34	2 mA	RS232 transmit data output.
RI-	29	TTL in	RS232 ring indicator input.
DCD-	30	TTL in	RS232 carrier detect input.
DSR-	28	TTL in	RS232 data set ready input.
CTS-	32	TTL in	RS232 clear to send input.
RXD-	31	TTL in	RS232 receive data input.

3.0 ABSOLUTE MAXIMUM RATINGS

	Min	Typ	Max	Units
Storage Temperature:	-65		150	degrees C
Operating Temperature:	0	25	55	degrees C
All output pins	-0.5		7.0	volts DC
All input pins	-0.5		7.0	volts DC
Power Supply (Vcc)	-0.5		7.0	volts DC
Power dissipation			700	milliwatts

4.0 D. C. ELECTRICAL CHARACTERISTICS

4.1 Inputs

Leakage current	Min	Typ	Max	Units
			+/-10	uA
Vih (TTL in)	2.0		Vcc+.5	volts DC
Vih (DS1218)	2.1		Vcc+.5	volts DC
Vil	-0.5		0.8	volts DC
Input capacitance			10	pF

4.2 PD0 - PD7, INIT, /AFXT, /STROBE

	Min	Typ	Max	Units
Iol	4			mA
Vol			0.4	volts DC
Ioh	1			mA
Voh	2.4			volts DC
Capacitive load	100			pF

4.3 /WAIT

	Min	Typ	Max	Units
Iol	4			mA
Vol			0.4	volts DC
Capacitive load	100			pF

4.4 DRQ, /TXD, /DTR, /RTS, PINT, SINT

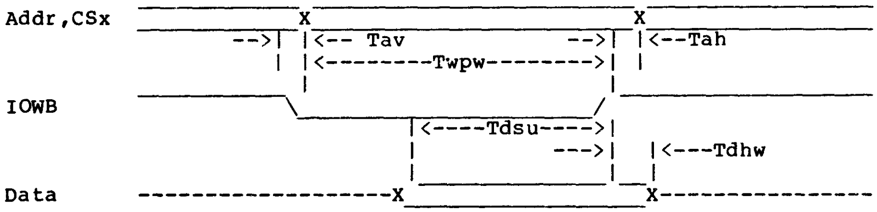
	Min	Typ	Max	Units
Iol	2			mA
Vol			0.4	volts DC
Ioh	1			mA
Voh	2.4			volts DC
Capacitive load	40			pF

4.5 IOD0 - IOD7

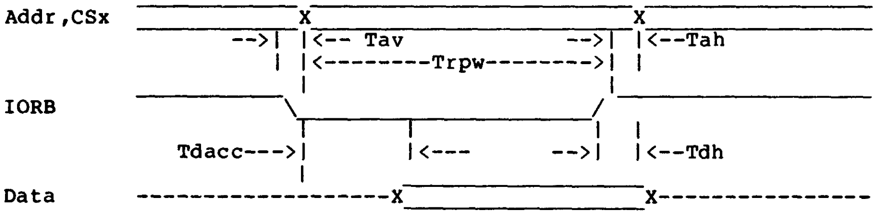
	Min	Typ	Max	Units
Iol	8			mA
Vol			0.4	volts DC
Ioh	2			mA
Voh	2.4			volts DC
Capacitive load	100			pF

5.0 AC CHARACTERISTICS

Parameter	Min	Typ	Max	Units
Tav (Address Valid)	-15			nSec
Tah (Address Hold)	30			nSec
Trpw (Read Pulse Width)	120			nSec
Twpw (Write Pulse Width)	125			nSec
Tdsu (Data Setup (Write))	65			nSec
Tdacc (Data Access (Read))			100	nSec
Tdhr (Data Hold (Read))	10		30	nSec
Tdhw (Data Hold (Write))	25			nSec



I/O Write Cycle



I/O Read Cycle

6.0 Modifications to the 76496

6.1 Extra Bit of Division by each channel.

When clocked by a 3.579545 MHz signal, the lowest frequency generated by the 76496 (with its 10 bit dividers) is 109.24 Hz. It is desired to be able to generate lower frequencies. An extra bit of division will allow frequencies down to 54.62 Hz, or an octave lower than the lowest note currently available. Since there is an extra bit in the frequency update register (second byte), it makes sense to implement this feature here. However, to maintain backwards compatibility, since it is not known what is programmed in this bit, there needs to be a way of defeating the extra bit of division. Therefore, there is a signal (SEDE), which enables the extra bit for all three channels. This bit defaults to a logic zero (low) on reset. When it is set, by writing to port C4 with bit 6 high, the extra divider will be enabled.

6.2 Synchronization of frequency dividers.

The current 76496 design loads each divider when initially written to, with no provision for synchronization of the dividers. This is a handicap when programming frequencies of low integer relationships to each other, because it is not possible to guarantee the phase of the signals. Therefore, if synchronization is desired, it is enabled by writing to port C4, with bit 5 set (which defaults to reset). When this bit is high, any write to a frequency register of the new sound channel will not only load its divider, but reload the dividers in the present 76496.

6.3 Minimum Wait State Generation

The 32 wait states generated by the 76496 need to be reduced. The chip must be guaranteed to latch the data written in the same time allotted for the 8250A megacell. Any wait states generated should only apply to a successive write (not the first in a series). All write timing should be referenced to the rising edge of the IOW- strobe.

Note: "76496" = Texas Instruments SN76496 3-voice tone and 1-voice noise generator chip, used in the IBM PC-Jr. The PSSJ is register-compatible with the 76496.

7.0 Software Specification

Port C0 - C3 Write

Access 76496 megacell

Port	R/W	7	6	5	4	3	2	1	0
C4	W	(res)	SEDE	SDSE	DIEN	DICL-	DMAEN	DF1	DF0

Where:

DF1	DF0	=	Dac Function Select
0	0	=	Joystick
0	1	=	Sound Channel
1	0	=	Successive Approximation
1	1	=	Direct write to DAC
DMAEN		=	DMA Enable (for SA, direct R/W)
0		=	DMA Disabled
1		=	DMA Enabled for SA, DA
DICL-		=	DMA interrupt clear
0		=	DMA interrupt held clear
1		=	DMA interrupt allowed
DIEN		=	DMA Interrupt enable
0		=	DMA EOP interrupt disabled
1		=	DMA EOP interrupt enabled
SDSE		=	Sound Divider Sync Enable
0		=	Synchronization Disabled
1		=	Sync Enabled: Write to C6 or C7 reloads all dividers
SEDE		=	Sound Chip Extra Divide Enable
0		=	Extra Divide disabled
1		=	Extra Divide enabled
(res)		=	reserved

ˋort C4   Read

Readback all bits except bit 3. In addition:]

bit 7	=	SAD-	=	Successive Approximation done. Useful when polling instead of DMA for successive approximation.
bit 3	=	DIO	=	DMA interrupt has occurred. To clear the interrupt it is necessary bring DICL low, then back high.

Port C5   Write

Direct write to DAC   (DF1,0 = 11 bin).
Pulse width and waveshape   (DF1,0 = 01 bin).

7	6	5	4	3	2	1	0
				=			
WS1	WS0	(res)	(res)	(res)	PW2	PW1	PW0

Where:

WS1	WS0		
		=	Waveshape select bits
0	0	=	Pulse
0	1	=	Ramp
1	0	=	Triangle
1	1	=	Reserved

PW2	PW1	PW0		
0	0	0	=	6.25% duty cycle
0	0	1	=	12.5% duty cycle
0	1	0	=	18.75% duty cycle
0	1	1	=	25.0% duty cycle
1	0	0	=	31.25% duty cycle
1	0	1	=	37.5% duty cycle
1	1	0	=	43.75% duty cycle
1	1	1	=	50% duty cycle

Port C5   Read

Direct read of DAC (Succ. Approx.) (DF1,0 = 1X bin).
Direct read of Snd Control register (DF1,0 = 01 bin).

duty cycle = width of pulse?

sampling output:

FØ - F11 control samples/sec

SAMP1-3 control volume

Port C6   R/W

Frequency LSB for DAC sound channel.

7	6	5	4	3	2	1	0
F7	F6	F5	F4	F3	F2	F1	F0

Port C7   R/W

Amplitude/frequency MSN for DAC sound channel.

7	6	5	4	3	2	1	0
SAMP3	SAMP2	SAMP1	res	F11	F10	F9	F8

The amplitude will be programmable in 7 levels, with approximately 3 dB per level. The maximum level ('111') will closely approximate that in the existing sound chip. A value of '000' will result in no output. This level control also applies to the raw DAC output when outputting digitized sound.

The ramp will count up the five MSB's of the DAC. The triangle will count up the four MSB's of the DAC for the first half of the wave, then count them back down for the second half. The frequency range of the DAC as a sound channel will have the same upper limit and a lower limit of one octave lower than the new frequency range of the sound chip (down to 27.3 Hz.). Obviously, the bit programming order of the frequency is different. The actual frequency will be 111.86 KHz divided by the number programmed into the sound frequendy register(s).

Port 200 - 207 WR   --   Clear Joystick DAC counter

A write to port 20X, where X = 0 to 7, will clear a free-running counter, and load a value of 16 into the 12-bit divider. The eight bit free-running counter will be clocked by the 3.58 MHz signal divided by 24, or 149.1 KHz. The output of the eight bit counter will drive the DAC to produce a stairstep wave, which simulates a ramp for use by the joystick comparators. When the counter reaches a count of 255, it will stop until port 20X is written to again.

The elapsed time for the complete ramp will be approximately 1.7 milliseconds, closely approximating the elapsed time of the current Tandy 1000 Joystick circuitry.

Bit 4 at port C7 is used to determine the DAC version. If able to set and clear the bit, old version — else new, See POST.

Port 200 - 207 RD -- Joystick Status

The data read at port 20X, where X = 0 to 7, will be the outputs of the joystick position comparators and the states of the joystick pushbuttons, in the same manner as the current Tandy 1000 Joystick circuitry.

Planar Control

Port 65 contains three bits which are used to enable the printer interface (bit 1), the printer output (bit 7), and the serial port (bit 4). These bits are all enabled (set high) on reset, and must be cleared by software to disable the appropriate function. The printer output enable function is logically "or-ed" with the current Tandy 1000 printer output enable bit, so that either one will enable the printer output buffer.

Additional control is available at port FFEB. Bit D0 selects whether the serial clock is divided by 13 or 1. Bit D1 must be high to enable the joystick function, and bit D2 must be high to enable the sound chip functions. Bits D1 and D2 default to high on power up.

I/O Map Summary

The following ports are utilized in the PSSJ part:

PORT	R/W	BITS	FUNCTION
0061	W	4	Sound Chip Enable
0065	R/W	1,4,7	Planar Control
00C0-00C3	W	all	Sound Chip Data
00C4-00C7	R/W	all	DAC Functions
0200-0207	R/W	all	Joystick Function
0378-037A	R/W	all	Printer Interface
03F8-03FF	R/W	all	Serial Interface
FFEB	R/W	0,1,2	UART clock select,JSE,DSE

Floppy Disk Support Chip Specification

Floppy Disk Support Chip Specification
Contents

Floppy Disk Support Logic
Tandy Part #8041404
January 29, 1987

1.0 General Description

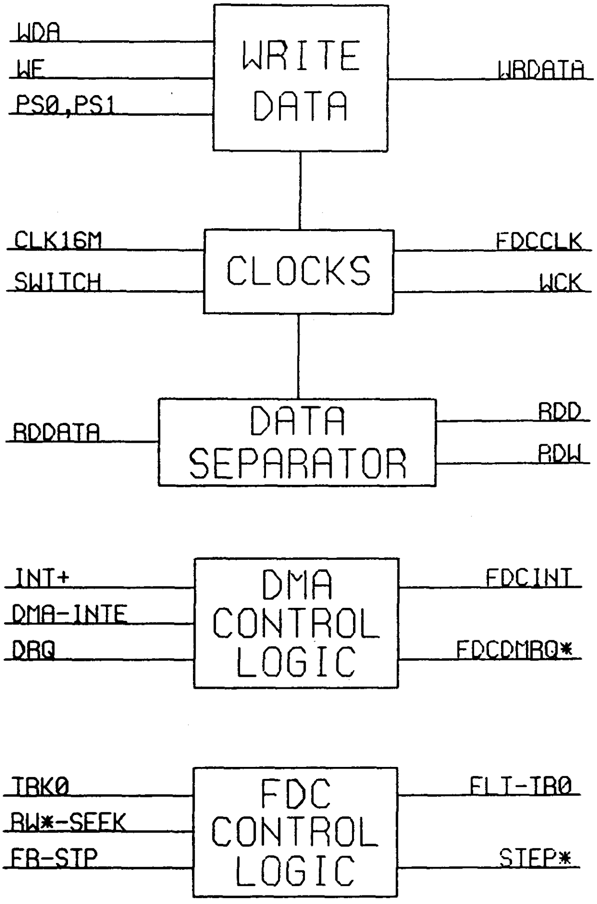
1.1 The Tandy Part #8041404 - Floppy Disk Support Logic:
-Generates the clock to the 765 Floppy Disk Controller.
-Generates the write clock to the Floppy Disk.
-Generates step pulses, track 0 indicator, DMA request, and FDC interrupt signals.

1--	CLK16M	+5V	--24
2--	WCK	SWITCH	--23
3--	FDCCLK	INT+	--22
4--	RDDATA*	DMA/INTE	--21
5--	RDD	DRQ	--20
6--	RDW	FDCINT	--19
7--	FRES/S	FDCDMRQ*	--18
8--	RW*/SEEK	PSO	--17
9--	TRKO*	PS1	--16
10--	F/TRKO	WRD	--15
11--	STEP*	WRE	--14
12--	GND	WRDATA*	--13

Figure 1. Pin Assignment

1.2 DESCRIPTION OF PINS:

PIN #	PIN NAME	TYPE	DESCRIPTION
1	CLK16M	INPUT	Frequency = 16.0000 Tolerance = 100pmm
2	WCK	OUTPUT	If SWITCH = 0, period = 2 us, 250 ns pulse If SWITCH = 1, period = 1 us, 250 ns pulse
3	FDCCLK	OUTPUT	If SWITCH = 0, then CLK16M/4 If SWITCH = 1, then CLK16M/2
4	RDDATA	INPUT	Serial data from FDD
5	RDD	OUTPUT	Serial data from FDC
6	RDW	OUTPUT	Read Data Window
7	FRES/S	INPUT	Step pulses to move head to another cylinder
8	RW*/SEEK	INPUT	Specifies seek mode when high
9	TRK0*	INPUT	From FDD, indicating head is on track 0
10	F/TRK0	OUTPUT	To FDC, indicating head is on track 0
11	STEP*	OUTPUT	Moves head of FDD
12	GND		Ground
13	WRDATA*	OUTPUT	Serial Data to FDD
14	WRE	INPUT	Write Enable
15	WRD	INPUT	Serial Data from FDC
16	PS1	INPUT	Write precompensation status
17	PS0	INPUT	Write precompensation status
18	FDCDMRQ*	OUTPUT	DRQ delayed by 1.0 usec.
19	FDCINT	OUTPUT	Interrupt request
20	DRQ	INPUT	FDC DMA Request
21	DMA/INTE	INPUT	DMA request and FDC interrupt enable
22	INT+	INPUT	Interrupt request generated by FDC
23	SWITCH	INPUT	0 = low density drive 1 = high density drive
24	+5V		+5 Volts



BLOCK DIAGRAM

2.0 ENVIROMENTAL SPECIFICATIONS

2.1 Storage temperature: -65°C min., +150°C max.
2.2 Operating temperature: 0°C min., +25°C typ, +70°C max.

3.0 DC ELECTRICAL SPECIFICATIONS

3.1 Absolute Maximum Rating:
Voltage on any pin
w.r.t. Ground: -0.5 min., 7.0 max. volts

3.2 Operating Electrical Specifications:

	Min.	Typ.	Max.	Units
3.2.1 Operating Ambient:				
Air Temperatue Range	0	25	70	°C
3.2.2 Power Supplies:				
VCC	4.5	5.0	5.5	volts
VSS	0	0	0	volts
ICC				milliamps
Total Power				milliwatts
3.2.3 Leakage Current, All Inputs:				
Vin = 0.0 v			-10	microamps
Vin = 5.0 v			+10	microamps
3.2.4 Input voltages:				
3.2.4.1 Except RDDATA*, TRK*				
Logic "0"			.8	volts
Logic "1"	2.0			volts
3.2.4.2 RDDATA*, TRK*				
Positive going threshold		1.8		volts
Negative going threshold		1.2		volts
Hysteresis voltage	220			millivolts
3.2.5 Output Voltages:				
3.2.5.1 Except WRDATA*, STEP*				
Logic "0" @ 4.0 mA load			.4	volts
Logic "1" @ 4.0 mA load	2.4			volts
3.2.5.2 WRDATA*, STEP*				
Logic "0" @ 48 mA			.5	volts
3.2.6 Input Capacitance (0.0 < Vin < 5.0)				
All inputs			10	pf
3.2.7 Output Capacitance				
All loads			50	pf

4.0 AC CHARACTERISTICS

4.1 FDCCLK Timing

Parameter	Min.	Typ.	Max.	Units
t_H	90	120	130	nSec
t_R, t_F		5	10	nSec
t_L	100	120	160	nSec
t_{CY}	245	250	255	nSec

4.2 WCK Timing

t_H	100	250	250	nSec
t_R, t_F		5	10	nSec
t_L		$t_{CY}-(t_H+t_R+t_F)$		
t_{CY}		2.0		μSec

4.3 WRDATA* Timing

WCK_H-WE_H	20			nSec
WCK_L-WE_L	20			nSec
PSD	20		100	nSec
WDD	20		100	nSec
WDA_W			WCK_H-50	nSec
WRD_W	115	125	135	nSec
WDD_H-WRD_L early	150		250	nSec
WDD_H-WRD_L nominal	275		375	nSec
WDD_H-WRD_L late	400		500	nSec

4.4 DMA/INTERRUPT Timing

I_H-FI_H			30	nSec
I_L-FI_L			30	nSec
DI_L-FI_L			30	nSec
WCK_H-DRQ_H	0			nSec
WCK_L-DRQ_H		-20		nSec
DRQ_H-$FDRQ_H$	750		1050	nSec
DRQ_L-$FDRQ_L$			30	nSec
DI_L-$FDRQ_L$			30	nSec
FCK_H-$FDRQ_H$			30	nSec

4.5 CONTROL Timing

Parameter	Min.	Typ.	Max.	Units
T_L-FT_H			30	nSec
T_H-FT_L			30	nSec
RS_L-FT_L			30	nSec
F_H-S_L			30	nSec
F_L-S_H			30	nSec
RS_L-S_H			30	nSec

4.6 DATA SEPARATOR Timing

Parameter	Min.	Typ.	Max.	Units
RDA_W	200	350	550	nSec
RDA_L-RDD_H	188		313	nSec
RDD_W	240	250	260	nSec
RDD_H-RDW_C	850	875	900	nSec
RDW(ND)$_W$		2.0		µSec

"A"

Parameter	Min.	Typ.	Max.	Units
RDA_S	3062			nSec
RDW_C-RDD_H	15			nSec

"B"

Parameter	Min.	Typ.	Max.	Units
RDA_S	4812			nSec
RDW_C-RDD_H			1938	nSec

"C"

Parameter	Min.	Typ.	Max.	Units
RDA_S	5062			nSec
RDW_C-RDD_H	15			nSec

FDSL AC TIMING

FIG. 1 FDCCLK

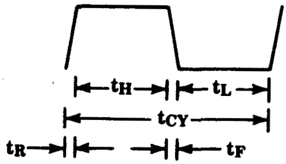
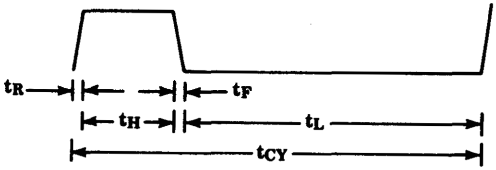


FIG. 2 WCK



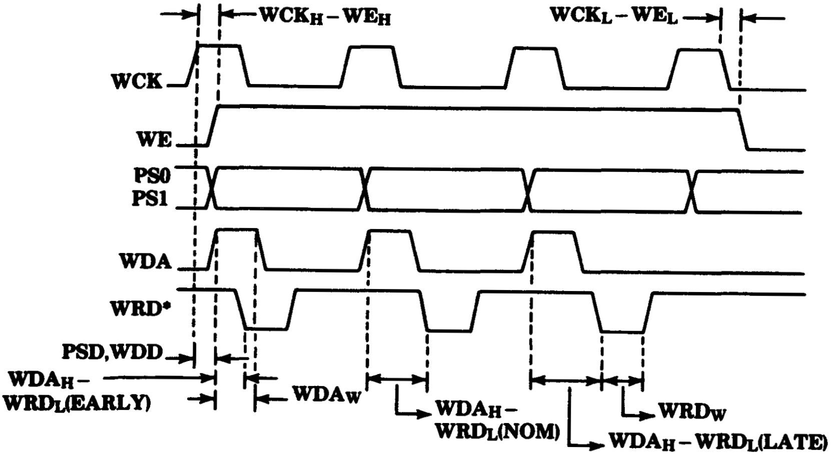


FIG. 3 WRITE DATA TIMING.

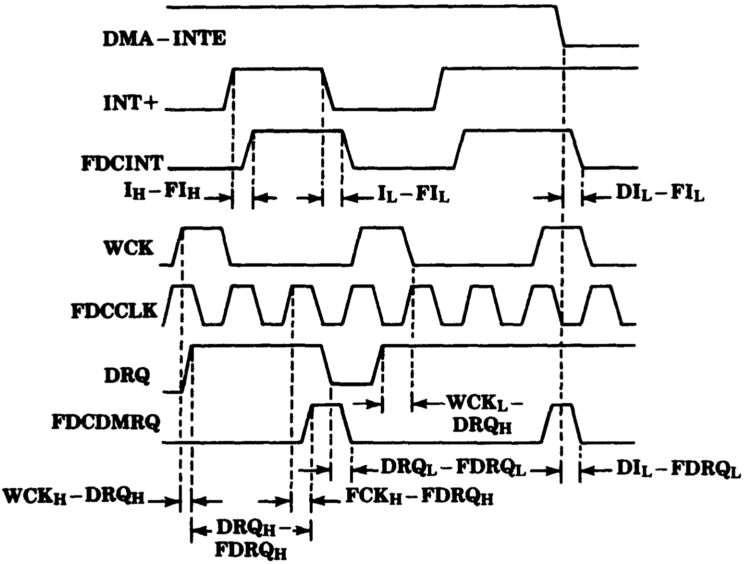


FIG. 4 DMA/INTERRUPT TIMING.

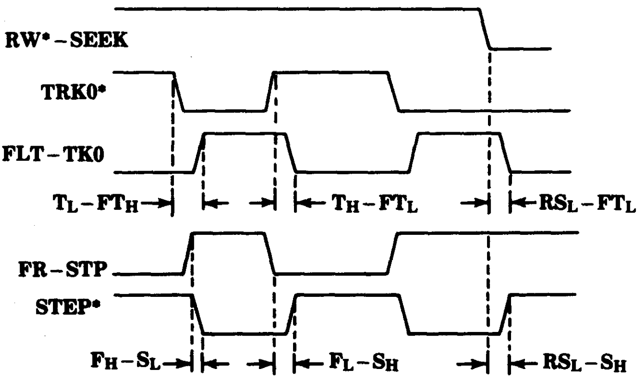


FIG. 5 CONTROL LOGIC TIMING.

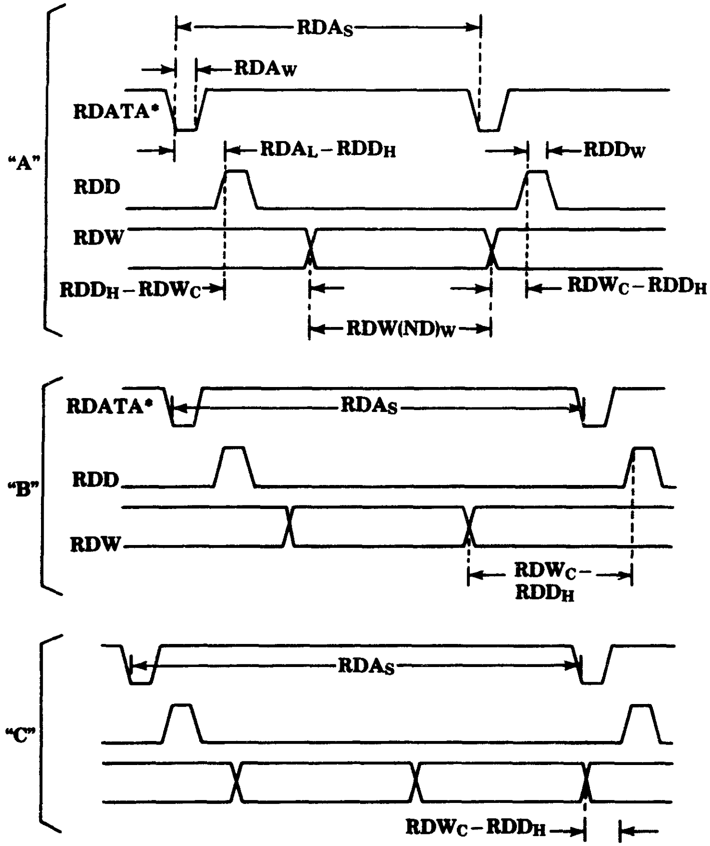


FIG. 6 DATA SEPARATOR TIMING.

KFIT CUSTOM CHIP

(KEYBOARD, FLOPPY SUPPORT, INTERRUPT, TIMER)

Tandy Corporation
1000 Two Tandy Center
Fort Worth, Tx 76102

TANDY PART #: 8079019

Table Of Contents

FUNCTIONAL DESCRIPTIONS

This Tandy KFIT custom IC consists of the following functional blocks:

- Programmable Peripheral Interface (PPI)
- Keyboard Interface Logic
- Floppy Disk Interface Logic
- Programmable Interrupt (equivalent to Intel 8259A) and sharing interrupt logic
- Programmable Timer (equivalent to Intel 8254-2 and Clock Divider
- Address Decoding Logic

Programmable Peripheral Interface

This section of the KFIT custom integrated circuit replaces the Intel 8255A that was used on the original Tandy 1000 computer. On the block diagram for this section of logic, the 8255A is represented by three 74LS244 buffers addressed by read A (0060), read B (0061), read C (0062). Also the two latches addressed by write B (0061), write C (0062) which are part of the original 8255A logic.

Keyboard Interface Logic

This section of the KFIT custom integrated circuit is design to support Tandy 1000 keyboard or Tandy 101 enhanced keyboard. KYBDTYP signal is used to select Tandy 1000 keyboard when is LOW or Tandy 101 enhanced keyboard when is HIGH. The KYBDTYP is being read in to port FFEB(hex) bit 7. The KBDDATA - keyboard data is serial data bit stream and then is converted to 8 bits parallel data by 74LS322. The serial data is entered in the LOW to HIGH transition of the KBCLK.

Floppy Disk Interface Logic

This section of the KFIT custom integrated circuit is design to support Floppy Disk Digital Output Register (DOR) function. This register is mapped in address 03F2 hex - data bit 0 to 7 (write only) to generate drive select DS0B,DS1B,DS2B; FDCRST (FDC reset) DMA/I and MTRONB (motor ON) signal. The DMA/I signal is used to disable FDCINTI, FDCDMRQ, and FDACKI signals for allowing the used of external FDC controller.

Programmable Interval Timer

This section of the KFIT custom integrated circuit is equivalent to an Intel 8254-5 and is designed to use with the Tandy 1000 TX. It is organized as three independent 16-bit counters, each with a clock of 1.19 MHZ. The 1.19 MHZ clock is generated from 14MHZ divided by 12. All modes of operation are software programmable.

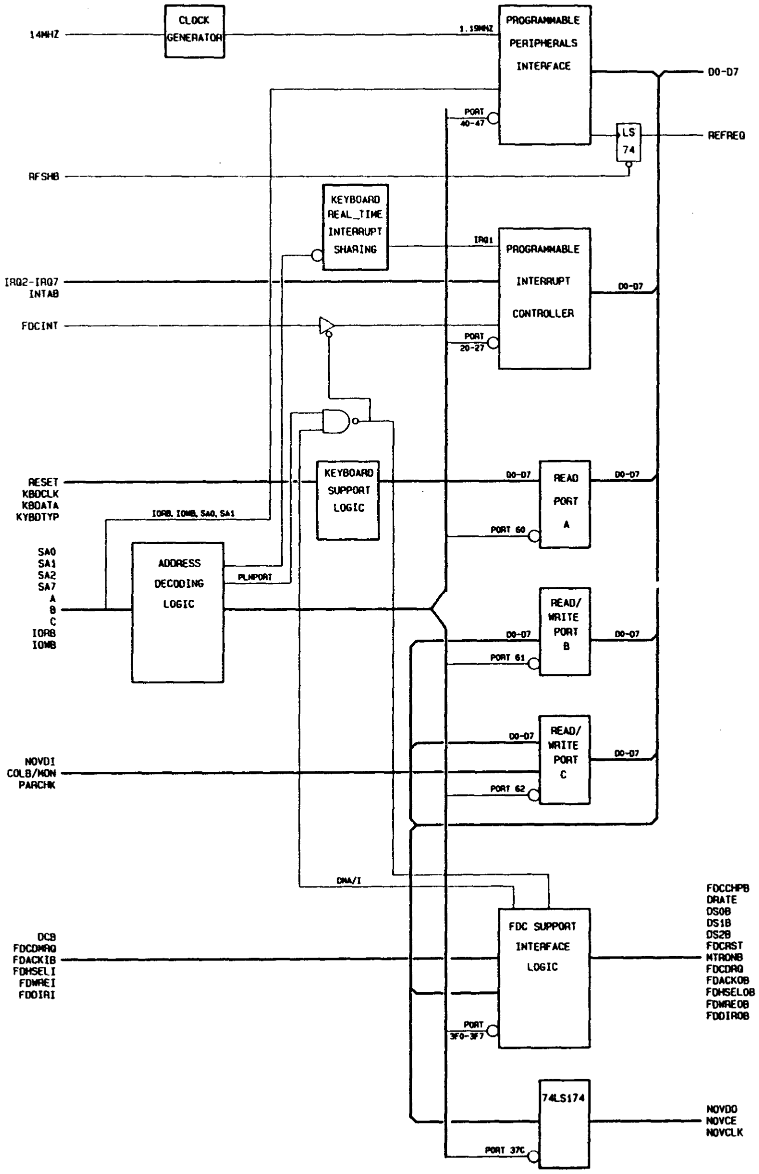
Programmable Interrupt

This section of the KTIF custom integrated circuit is equivalent to an Intel 8259A that capable of handling eight-vector priority interrupt, individual request mask and programmable interrupt modes. This circuit generates INTR output signal for the CPU. In addition, the sharing interrupt logics are implemented in the design for IRQ1 (between keyboard and real time clock interrupt)

Address Decoding Logic

This section contains 3 to 8 address decode to generate Programmable Interrupt Chip select, Programmable Interval Timer chip select, Floppy Disk chip select (FDCCHP*) and Programmable Peripheral Interface address of three decoded address A, B and C. (see IO signal definition). The FDC port is enabled by Planar register-port 0065hex bit 3 when bit 3 is HIGH.

BLOCK DIAGRAM



INPUT/OUTPUT PIN DESCRIPTIONS

#	Signal	Output Current	Pin Number	Type	Descriptions
1	XD0 (S.T input)	8ma	14	I/O	Data bus 0
2	XD1 (S.T input)	8ma	15	I/O	Data bus 1
3	XD2 (S.T input)	8ma	16	I/O	Data bus 2
4	XD3 (S.T input)	8ma	17	I/O	Data bus 3
5	XD4 (S.T input)	8ma	19	I/O	Data bus 4
6	XD5 (S.T input)	8ma	20	I/O	Data bus 5
7	XD6 (S.T input)	8ma	21	I/O	Data bus 6
8	XD7 (S.T input)	8ma	22	I/O	Data bus 07
9	SA0		28	I	System address 0
10	SA1		29	I	System address 1
11	SA2		30	I	System address 2
12	SA7		31	I	System address 7
13	A		32	I	CPU I/O address decode LSB
14	B		33	I	CPU I/O address decode
15	C		34	I	CPU I/O address decode MSB
16	IOWB		37	I	Active LOW. CPU I/O write signal
17	IORB		36	I	Active LOW. CPU I/O read signal

#	Signal	Output Current	Pin Number	Type	Descriptions
18	14MHZ		27	I	Clock signal 14.318 MHZ
19	BUSY (O.C., Pull_up)*	8ma	58	O	Keyboard busy When High
20	KYBDTYP (Pull-up)		61	I	Keyboard type select. When High, selects IBM PC keyboard. When Low selects Tandy keyboard
21	PPITM	2ma	12	O	Programmable Peripheral Interface Timer output signal for sound generator.
22	KBDDATA (3-state)	8ma	60	I/O	Input data signal from keyboard. In the IBM PC keyboard this pin is used as an output to hold the data Low.
23	KBDCLK (3-state)	8ma	59	I/O	Input clock signal from keyboard. In the IBM PC keyboard this pin is used as an output to hold the clock LOW.
24	DS0B (O.C. Pull_up*)	8ma	57	O	Drive select signal When Low.
25	DS1B (O.C. Pull_up*)	8ma	56	O	Drive select signal when is LOW.
26	DS2B (O.C. Pull_up*)	8ma	55	O	Drive select signal when is LOW.
27	DCB (Pull_up)		38	I	Disk change signal when is LOW

#	Signal	Output Current	Pin Number	Type	Descriptions
28	DRATE (O.C.)	16ma	54	O	Data rate select signal. When is LOW, 500 kbps is selected. When is HIGH 250kbps is selected.
29	FDCRST	4ma	46	O	FDC reset signal to the FDC controller when is HIGH.
30	RESET		66	I	System reset input signal when is HIGH.
31	MTRONB (O.C.)	16ma	53	O	Floppy disk motor ON output signal when is LOW.
32	FDHSELI		42	I	Head select input signal from floppy disk controller.
33	FDHSELOB (O.C.)	16ma	49	O	Head select input signal for floppy drives when is LOW.
34	FDWREI		41	I	Write enable input signal from floppy disk controller.
35	FDWREOB (O.C.)	16ma	50	O	Write enable output signal for floppy drives when is LOW
36	FDDIRI		40	I	Head travel direction input signal from FDC controller.
37	FDDIROB (O.C.)	16ma	51	O	Head travel direction for floppy drive.
38	FDCCHPB	4ma	47	O	FDC chip select output signal or FDC controller when is LOW.

#	Signal	Output Current	Pin Number	Type	Descriptions
39	FDACKIB		44	I	FDC controller acknowledge output signal when is LOW.
40	FDACKOB	2ma	45	O	FDC controller acknowledge output signal when is LOW.
41	FDCINT		39	I	Floppy disk interrupt input signal when is HIGH.
42	FDCDMRQ		43	I	Floppy disk service request input signal to DMA when is LOW
43	FDCDRQ	4ma	48	O	Floppy disk service request output signal to the DMA when is LOW
44	IRQ2 (S.T. Pull_up)		5	I	Interrupt request 2 input signal
45	IRQ3 (S.T. Pull_up)		6	I	Interrupt request 3 input signal
46	IRQ4 (S.T. Pull_up)		7	I	Interrupt request 4 input signal
47	IRQ5 (S.T. Pull_up)		8	I	Interrupt request 5 input signal
48	IRQ6 (3-state, Pull_up)		10	I	Interrupt request 6 input/output signal
49	IRQ7 (S.T. Pull_up)		9	I	Interrupt request 7 input signal

#	Signal	Output Current	Pin Number	Type	Descriptions
50	INTAB		4	I	Interrupt acknowledge signal. This signal is used to enable interrupt vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU
51	INTR	2ma	11	O	Interrupt request signal. This signal is used to interrupt the CPU when HIGH
52	RTCINTB (S.T. Pull_up)		2	I	Real time clock interrupt signal from the Real Time Clock device when LOW.
53	NOVDI		26	I	NOV_RAM data in signal
54	NOVCE	2ma	23	O	NOV_RAM chip enable signal
55	NOVDO	2ma	25	O	NOV_RAM data out signal
56	NOVCK	2ma	24	O	NOV_RAM clock
57	RFRSHB		3	I	DMA acknowledge signal from 8237. This signal is active HIGH
58	REFREQ	2ma	13	O	DMA Request signal for 8237. This signal is active HIGH
59	COL/MON		68	I	Input configuration control signal
60	PARCHK		67	I	Parity Check input signal

#	Signal	Output	Pin	Type	Descriptions
		Current	Number		
61	VCC		1		Power supply +5V
62	VCC		35		Power supply +5V
63	GND		52		Ground
64	GND		18		Ground
65	not used		62		
66	not used		63		
67	not used		64		
68	not used		65		

Notes:- O.C. = Open_Collector
- 3-State = Tri_State
- S.T. = Schmitt Trigger
- * = Max.=1.6ma, Min.=0.4ma sinking current.
  These signals must have external termination.

I/O MAPS

I/O Signal Definition:

C	B	A	Address Range Hex	Function
0	0	0	0020 - 0027	Interrupts
0	0	1	0040 - 0047	Timer
			00C0 - 00C7	Sound
0	1	0	0060 - 0067	PPI
			0065	Planar Register
0	1	1	03F0 - 03F7	Floppy
1	0	0	0200 - 0207	Joystick
1	0	1	0378 - 037F	Printer
			037C	NOVRAM
			03F8 - 03FF	Serial
1	1	0	FFE8 - FFEF	Non IBM compatible
1	1	1	----------	Inactive

Register Definition:

Address Range Hex	Bit	Description
Interrupt		
0020	~	Initialization Command Word 1
0021	~	Initialization Command Word 2
0022 - 0027		Not used
Timer		
0040/0044	~	Timer
0041/0045	~	Timer
0042/0047	~	Timer

Note: ~ = refers to system I/O maps.

PPI/Keyboard

Address Range Hex	Bit	Description
0060 - Port A		Keyboard Read Data Input
	0	Read only Keyboard bit 0 LSB
	1	Read only Keyboard bit 1
	2	Read only Keyboard bit 2
	3	Read only Keyboard bit 3
	4	Read only Keyboard bit 4
	5	Read only Keyboard bit 5
	6	Read only Keyboard bit 6
	7	Read only Keyboard bit 7 MSB
0061 - Port B		Read/Write
	0	R/W Timer gate #2 enable
	1	R/W Speaker data out enable
	2	R/W not used
	3	R/w not used
	4	R/W 1=disable internal speaker
	5	R/W not used
	6	R/W HOLDCK
	7	R/W 1=keyboard clear

Address Range Hex	Bit	Description
0062 - Port C		Read/Write
	0	R/W not used
	1	R/W not used
	2	R/W not used
	3	R/W 0=slow speed
	4	Read NOVDI
	5	Read output Timer #2
	6	Read 0=color
	7	Read 1=Parity check
0063-0064		Port not used

Planar Control

Address Range Hex	Bit	Description
0065		Planar Register Read/Write
	0	Reserved
	1	Reserved
	2	Reserved
	3	1=FDC chip select enable
	4	Reserved
	5	Reserved
	6	Reserved
	7	Reserved
0066		Not Used
0067		Port D not used

Non Volatile Memory Access

Address Range Hex	Bit	Description
037C		Non-volatile memory write only
	0	NOVDO
	1	NOVCE
	2	NOVCLK
	3	Reserved
	4	Reserved
	5	Reserved
	6	Reserved
	7	Reserved

Floppy Disk Control

Address Range Hex	Bit	Description			
03F0		Not used			
03F1		FDC Mode Control			
	0	Not used			
	1	Write - Drive Select switch 0 = 0-0 1-1 1 = 0-1 1-0			
	2	Not used			
	3	Not used			
	4	Not used			
	5	Not used			
	6	Not used			
	7	Not used			
03F2		FDC Digital Output Register (DOR) Write Only			
			DS0	DS1	DS2
	0	Write -	0	1	0
	1	Write -	0	0	1
	2	Write - FDC reset			
	3	Write - Enable DMA Req/Int.			
	4	Write - Drive 0 Motor ON			
	5	Write - Drive 1 Motor ON			
	6	Not used			
	7	Not used			
03F3		Not used			
03F4		FDC chip select			
03F5		FDC chip select			
03F6		Not used			
03F7		FDC Data Rate Selection			
	0	Not used			
	1	Write - Data Rate 0 = 500K bits per second 1 = 250K bits per second			
	2	Not used			
	3	Not used			
	4	Not used			
	5	Not used			
	6	Not used			
	7	0=Disk Change			

System Configuration Register

FFEB		Non IBM Compatible Read/Write
	0	Reserved
	1	Reserved
	2	Reserved
	3	Reserved
	4	Reserved
	5	Read 1=Keyboard Interrupt
	6	Read 1=Real Time Clock Interrupt Write 1=Enable Real Time clock Interrupt
	7	Read Keyboard Select 0=Tandy Keyboard 1=101 Enhanced Keyboard

Summary on the active/float data bits. (READ ONLY)

Address	Net Name	Active Bits	Float Bits
0065	CSEN	XD3	XD0-XD2, XD4-XD7
03F7	FDMDRDB	XD7	XD0-XD6
FFEB	CDENRDB	XD5 XD6 XD7	XD0-XD4

ELECTRICAL SPECIFICATIONS

Absolute Maximum Rating

Parameter	Min.	Typ.	Max.	Unit	Condition
Voltage, any pin	-0.5		7.0	V	W.R.T gnd
Power Dissipation					

D.C. Electrical Characteristics at Ta= 0 to 70 degree Celsius

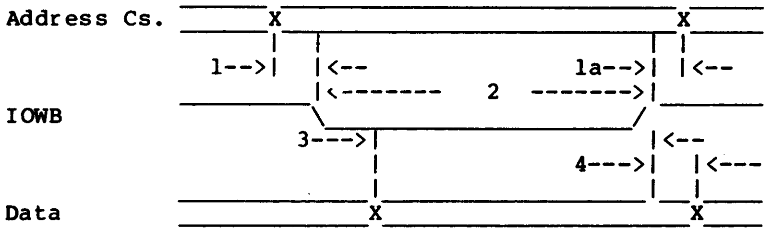
SYM.	Parameter	Min.	Typ.	Max.	Unit	Condition
Vdd	Supply Voltage	4.5		5.5	V	
Vil	Input Low Voltage			0.8	V	TTL input
Vih	Input High Voltage	2.0			V	TTL input
Iin	Input Leakage Current	-10		10	UA	
Cin	Input Capacitance			10	PF	
Vol	Output Low Voltage Unless otherwise specified in I/O Pin Descriptions			0.4	V	2MA
Voh	Output High Voltage Unless otherwise specified in I/O Pin Descriptions	2.4			V	-2MA
Ioz	High Impedance leak	-10		10	UA	
Voh(INTR)	Output High Voltage for INTR	3.5 2.4			V V	@ -100ua @ -400ua
Zo	Output Capacitance XD0 - XD7	100			PF	Note A

Notes: A. 50 PF is used in manufacture test.

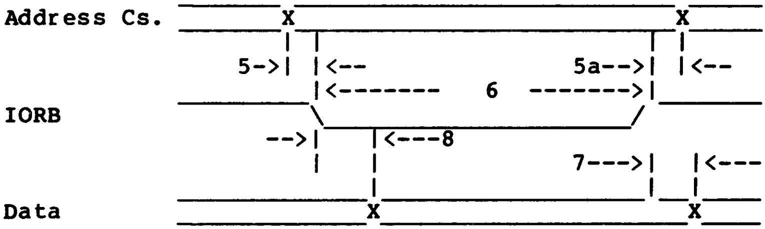
KEYBOARD TIMING SPECIFICATIONS

NUM.	Parameter	Min.	Typ.	Max	Unit	
1	Address valid to IOWB active	15			ns	
1a	Address hold from IOWB Inactive	20			ns	
2	IOWB pulse width	125			ns	
3	Data setup from IOWB Inactive	65			ns	Write
4	Data hold from IOWB Inactive	30			ns	Write
5	Address valid to IORB active	15			ns	
5a	Address hold from IORB inactive	30			ns	Read
6	IORB pulse width	120			ns	
7	Data hold/release from IORB inactive	5		55	ns	Read
8	Data access time			100	ns	Read

I/O Write Cycle



I/O Read Cycle



FLOPPY DISK TIMING SPECIFICATIONS:

NUM.	Parameter	Min.	Typ.	Max	Unit	
1	Address setup from IOWB active	15			ns	
1a	Address hold from IOWB Inactive	20			ns	
2	IOWB pulse width	125			ns	
3	Data setup from IOWB inactive	65			ns	
4	Data hold from IOWB inactive	30			ns	
5	DS0B-DS2B,FDCRST, MTRONB inactive delay from IOWB inactive			43	ns	
6	DS0B-DS2B,FDCRST, MTRONB active delay from IOWB inactive			41	ns	

I/O Write Cycle

```
            _______ _____________________________ ______
Address Cs. _______X_____________________________X______
                   | |                      |    |
               1->|  |<--              1a-->|    |<--
                     |<-------   2   ------->|
            _________                         __________
IOWB                 \_______________________/
                 3--->|                              |<--
                      |               4--->|         |<---
            ____________|___________________|_________|_____
Data        ____________X___________________________X_____
                                    5-->|     |<--
                                    6-->|  |<--
                                           |_ |
DS0B-DS2B  Low to High___________________/ ‾  \_ _ _ _ _ _
                                                High to LOW
```

PROGRAMMABLE INTERRUPT TIMING AND DESCRIPTIONS
Must meet Intel 8259A. Any differences must be specified.

PROGRAMMABLE TIMER TIMING AND DESCRIPTIONS
Must meet Intel 8254-5.

Any differences must be specified.

ADDRESS PORT EQUATIONS

```
/**************************************************************/
/*                                                            */
/*      KEYBOARD, TIMER CONTROL, INTERRUPT CONTROL, FDC-DOR   */
/*      AND DECODE LOGIC                                      */
/**************************************************************/
/*  Allowable Target Device Types:    F153                    */
/**************************************************************/

/**  Inputs  **/

PIN     1    =  sa01     ;    /* System address 1               */
PIN     2    =  sa00     ;    /* System address 0               */
PIN     3    =  sa02     ;    /* System address 2               */
PIN     4    =  !iow     ;    /* I/O Write                      */
PIN     5    =  !ior     ;    /* I/O Read                       */
PIN     6    =  !fdcport ;    /* FDC Port 03F0-03F8 hex         */
PIN     7    =  !keyport ;    /* Keyboard Port 0060 - 0067      */

/**  Outputs  **/

PIN     9    =  !fdmdrd  ;    /* Read FDC Port 03F7 hex         */
PIN     11   =  !fdmdwt  ;    /* Write FDC Port 03F7 hex        */
PIN     12   =  !fdcchp  ;    /* FDC Chip Select 03F4 - 03F5    */
PIN     13   =  dorltch  ;    /* Write DORLTCH Port 03F2        */
PIN     14   =  drvsck   ;    /* Write Port 03F1 DriveSwitch    */
PIN     15   =  writdp   ;    /* Write Keyboard Port 0067 or D*/
PIN     16   =  cp       ;    /* Port 0062 or C                 */
PIN     17   =  bp       ;    /* Port 0061 or B                 */
PIN     18   =  readap   ;    /* Read Port 0060                 */
PIN     19   =  csen     ;    /* Chip Select Enable Port 0065 */

/**  Logic Equations  **/

fdmdrd = fdcport & sa02 & sa01 & sa00 & ior;

fdmdwt = fdcport & sa02 & sa01 & sa00 & iow;

fdcchp = fdcport & sa02 & !sa01;

!dorltch = fdcport & iow & !sa02 & sa01 & !sa00;

!drvsck = fdcport & iow & !sa02 & !sa01 & sa00;

!readap = keyport & ior & !sa02 & !sa01 & !sa00;

bp = keyport & !sa02 & !sa01 & sa00;

cp = keyport & !sa02 & sa01 & !sa00;

csen = keyport & sa02 & !sa01 & sa00;
```

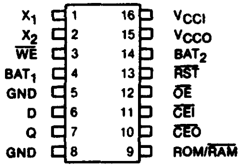
Dallas Semiconductor
TimeChip

DS1215

FEATURES

- TimeChip keeps track of hundredths of seconds, seconds, minutes, hours, days, date of the month, months, and years
- Adjusts for months with fewer than 31 days
- Leap year automatically corrected
- No address space required
- Provides nonvolatile controller functions for battery backing up RAM
- Supports redundant batteries for high-rel applications
- Uses a 32.768 KHz watch crystal
- Full 10% operating range
- Operating temperature range 0°C to 70°C
- Space saving 16-pin DIP package

PIN CONNECTIONS



PIN NAMES

Pins 1 & 2	X_1, X_2	32.768 KHz Crystal Connections
Pin 3	$\overline{WE}$	Write Enable
Pin 4	BAT_1	Battery 1 Input
Pins 5 & 8	GND	Ground
Pin 6	D	Data In
Pin 7	Q	Data Out
Pin 9	ROM/$\overline{RAM}$	ROM-RAM Select
Pin 10	$\overline{CEO}$	Chip Enable Out
Pin 11	$\overline{CEI}$	Chip Enable Input
Pin 12	$\overline{OE}$	Output Enable
Pin 13	$\overline{RST}$	Reset
Pin 14	BAT_2	Battery 2 Input
Pin 15	V_{CCO}	+5V Output
Pin 16	V_{CCI}	+5V DC Input

DESCRIPTION

The DS1215 is a combination of a CMOS timekeeper and a nonvolatile memory controller. In the absence of power, an external battery maintains the timekeeping operation and provides power for a CMOS static RAM. The watch provides hundredths of seconds, seconds, minutes, hours, day, date, month, and year information, while the nonvolatile controller supplies all the necessary support circuitry to convert a CMOS RAM to a nonvolatile memory. The DS1215 can be interfaced with either RAM or ROM without leaving gaps in memory.

The last date of the month is automatically adjusted for months with less than 31 days, including correction for leap year every four years. The watch operates in one of two formats: a 12-hour mode with an AM/PM indicator, or a 24-hour mode.

The nonvolatile memory controller portion of the circuit is designed to handle power fail detection, memory write protection, and battery redundancy. In short, the controller changes standard CMOS memories into nonvolatile memories, and provides continuous power to the TimeChip. Alternatively the TimeChip can be used with ROM memory by controlling the Chip Enable Output signal ($\overline{\text{CEO}}$) while the TimeChip is being accessed.

OPERATION

The block diagram of Figure 3 illustrates the main elements of the TimeChip. Communication with the TimeChip is established by pattern recognition of a serial bit stream of 64 bits which must be matched by executing 64 consecutive write cycles containing the proper data on Data In (D). All accesses which occur prior to recognition of the 64-bit pattern are directed to memory via the Chip Enable Output pin ($\overline{\text{CEO}}$).

After recognition is established, the next 64 read or write cycles either extract or update data in the TimeChip and Chip Enable Output remains high during this time, disabling the connected memory.

Data transfer to and from the timekeeping function is accomplished with a serial bit stream under control of chip enable ($\overline{\text{CEI}}$), output enable ($\overline{\text{OE}}$), and write enable ($\overline{\text{WE}}$). Initially, a read cycle using the $\overline{\text{CEI}}$ and $\overline{\text{OE}}$ control of the TimeChip starts the pattern recognition sequence by moving a pointer to the first bit of the 64-bit comparison register. Next, 64 consecutive write cycles are executed using the $\overline{\text{CEI}}$ and $\overline{\text{WE}}$ control of the TimeChip. These 64 write cycles are used only to gain access to the TimeChip.

When the first write cycle is executed, it is compared to bit 1 of the 64-bit comparison register. If a match is found, the pointer increments to the next location of the comparison register and awaits the next write cycle. If a match is *not* found, the pointer does not advance and all subsequent write cycles are ignored. If a read cycle occurs at any time during pattern recognition, the present sequence is aborted and the comparison register pointer is reset. Pattern recognition continues for a total of 64 write cycles as described above until all the bits in the comparison register have been matched. (This bit pattern is shown in Figure 1). With a correct match for 64 bits, the TimeChip is enabled and data transfer to or from the timekeeping registers may proceed. The next 64 cycles will cause the TimeChip to either receive data on D, or transmit data on Q, depending on the level of $\overline{\text{OE}}$ pin or the $\overline{\text{WE}}$ pin. Cycles to other locations outside the memory block can be interleaved with $\overline{\text{CEI}}$ cycles without interrupting the pattern recognition sequence or data transfer sequence to the TimeChip.

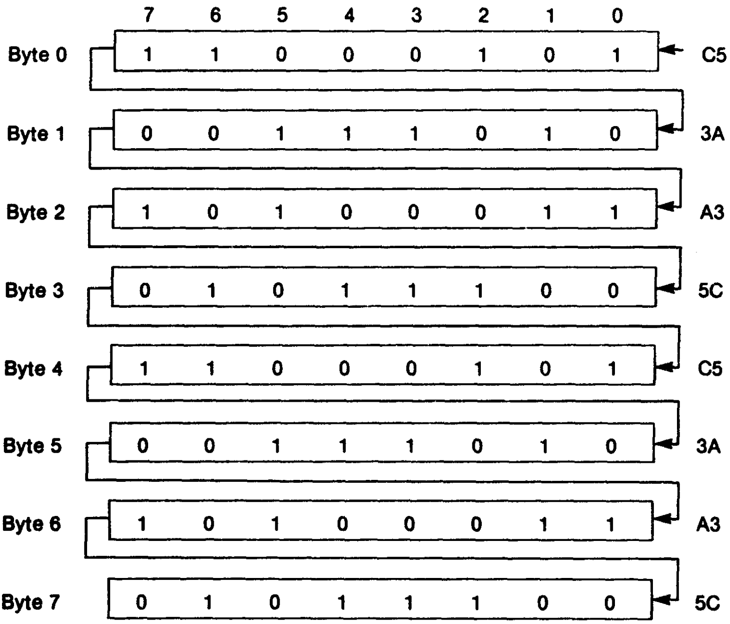
A 32,786 Hz quartz crystal, Seiko part no. DS-VT-200 or equivalent, can be directly connected to the DS1215 via pins 1 and 2 (X_1, X_2). The crystal selected for use should have a specified load capacitance (C_L) of 6 pF.

NONVOLATILE CONTROLLER OPERATION

The operation of the nonvolatile controller circuits within the TimeChip is determined by the level of the ROM/$\overline{\text{RAM}}$ select pin. When ROM/$\overline{\text{RAM}}$ is connected to ground, the controller is set in the RAM mode and performs the circuit functions required to make static CMOS RAM and the timekeeping function nonvolatile. First a switch is provided to direct power from the battery inputs or V_{CCI} to V_{CCO} with a maximum voltage drop of 0.2 volts. The V_{CCO} output pin is used to supply uninterrupted power to CMOS static RAM. The DS1215 also performs redundant battery control for high reliability. On power fail the battery with the highest voltage is automatically switched to V_{CCO}. If only one battery is used in the system, the unused battery input should be connected to ground. The DS1215 provides the function of safeguarding the TimeChip and RAM data by power fail detection and write protection. Power fail detection occurs when V_{CCI} falls below VTP which is equal to 1.26 × V_{BAT}. The DS1215 constantly monitors the V_{CCI} supply pin. When V_{CCI} is less than VTP, a comparator outputs a power fail signal to the control logic. The power fail signal forces the chip enable output ($\overline{\text{CEO}}$) to V_{CCI} or $V_{BAT} - 0.2$ volts for external RAM write protection. During nominal supply conditions, $\overline{\text{CEO}}$ will track $\overline{\text{CEI}}$ with a maximum propagation delay of 20 ns. Internally, the DS1215 aborts any data transfer in progress without changing any of the TimeChip registers and prevents future access until V_{CCI} exceeds VTP. A typical RAM/TimeChip interface is illustrated in Figure 4.

When the ROM/$\overline{\text{RAM}}$ pin is connected to V_{CCO}, the controller is set in the ROM mode. Since ROM is a read-only device which retains data in the absence of power, battery backup and write protection is not required. As a result, the chip enable logic will not force $\overline{\text{CEO}}$ high when power fails. However, the TimeChip does retain the same internal nonvolatility and write protection as described in the RAM mode. In addition, the chip enable output is set at a low level on power fail as V_{CCI} falls below the level of V_{BAT}. A typical ROM/TimeChip interface is illustrated in Figure 5.

TIMECHIP COMPARISON REGISTER DEFINITION Figure 1



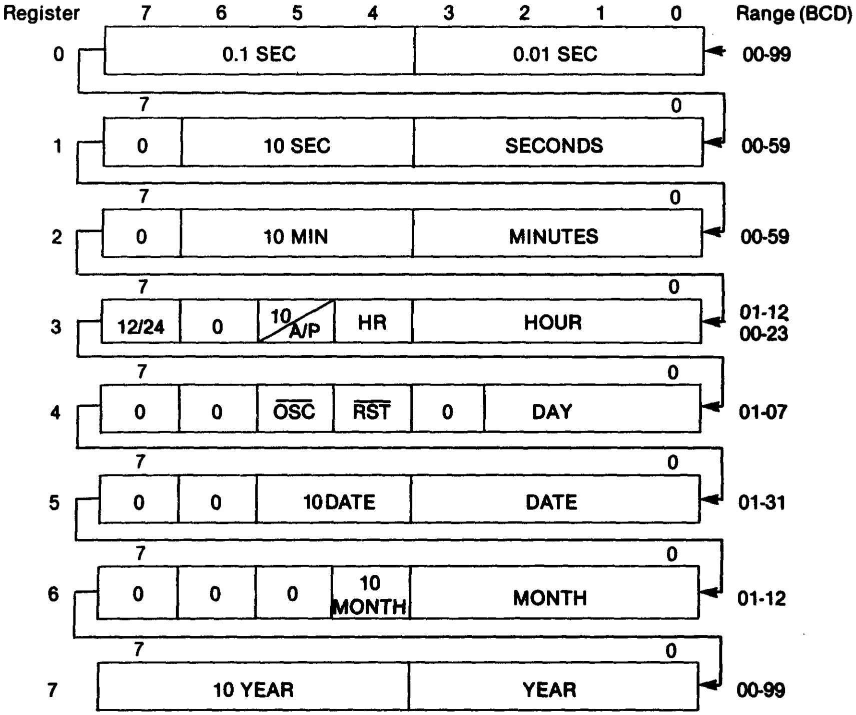
	7	6	5	4	3	2	1	0	
Byte 0	1	1	0	0	0	1	0	1	C5
Byte 1	0	0	1	1	1	0	1	0	3A
Byte 2	1	0	1	0	0	0	1	1	A3
Byte 3	0	1	0	1	1	1	0	0	5C
Byte 4	1	1	0	0	0	1	0	1	C5
Byte 5	0	0	1	1	1	0	1	0	3A
Byte 6	1	0	1	0	0	0	1	1	A3
Byte 7	0	1	0	1	1	1	0	0	5C

Note:
The pattern recognition in Hex is C5, 3A, A3, 5C, C5, 3A, A3, 5C. The odds of this pattern being accidentally duplicated and causing inadvertent entry to the TimeChip is less than 1 in 10^{19}.

TIMECHIP REGISTER INFORMATION
The TimeChip information is contained in 8 registers of 8 bits each which are sequentially accessed one bit at a time after the 64-bit pattern recognition sequence has been completed. When updating the TimeChip registers, each must be handled in groups of 8 bits. Writing and reading individual bits within a register could produce erroneous results. These read/write registers are defined in Figure 2.

Data contained in the TimeChip registers are not binary coded decimal format (BCD) in 12-hour mode. Reading and writing the registers is always accomplished by stepping through all 8 registers, starting with bit 0 of register 0 and ending with bit 7 of register 7.

TIMECHIP REGISTER DEFINITION Figure 2



Register	7	6	5	4	3	2	1	0	Range (BCD)
0	0.1 SEC				0.01 SEC				00-99
1	0	10 SEC			SECONDS				00-59
2	0	10 MIN			MINUTES				00-59
3	12/24	0	10/A/P	HR	HOUR				01-12 00-23
4	0	0	$\overline{\text{OSC}}$	$\overline{\text{RST}}$	0	DAY			01-07
5	0	0	10DATE		DATE				01-31
6	0	0	0	10 MONTH	MONTH				01-12
7	10 YEAR				YEAR				00-99

AM-PM/12/24 MODE

Bit 7 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20-23 hours).

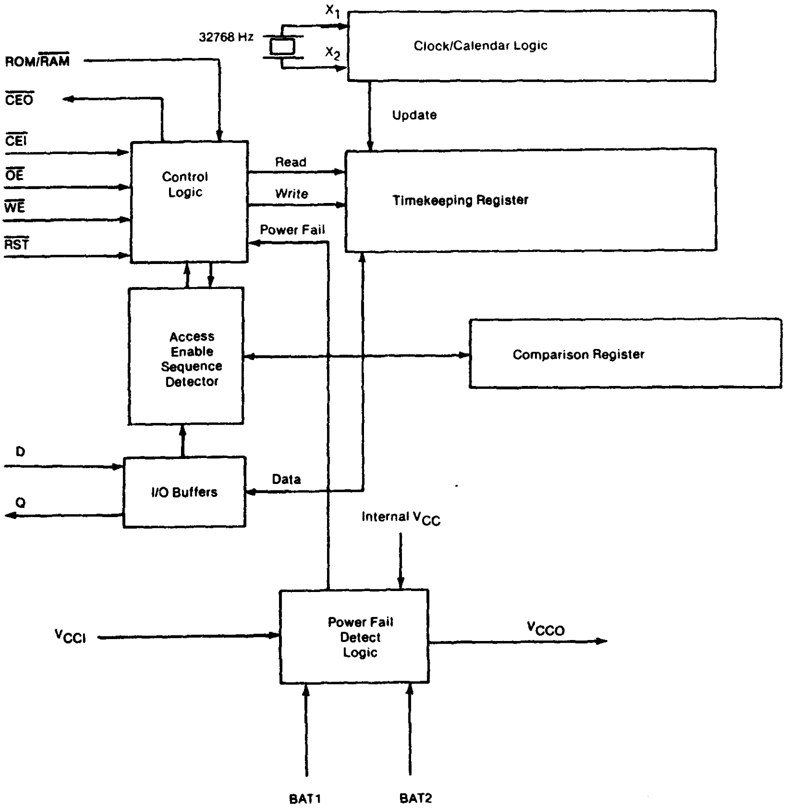
OSCILLATOR AND RESET BITS

Bits 4 and 5 of the day register are used to control the reset and oscillator functions. Bit 4 controls the reset pin (Pin 13). When the reset bit is set to logical 1, the reset input pin is ignored. When the reset bit is set to logical 0, a low input on the reset pin will cause the TimeChip to abort data transfer without changing data in the timekeeping registers. Reset operates independently of all other inputs. Bit 5 controls the oscillator. When set to Logic 0 the oscillator turns on and the watch becomes operational.

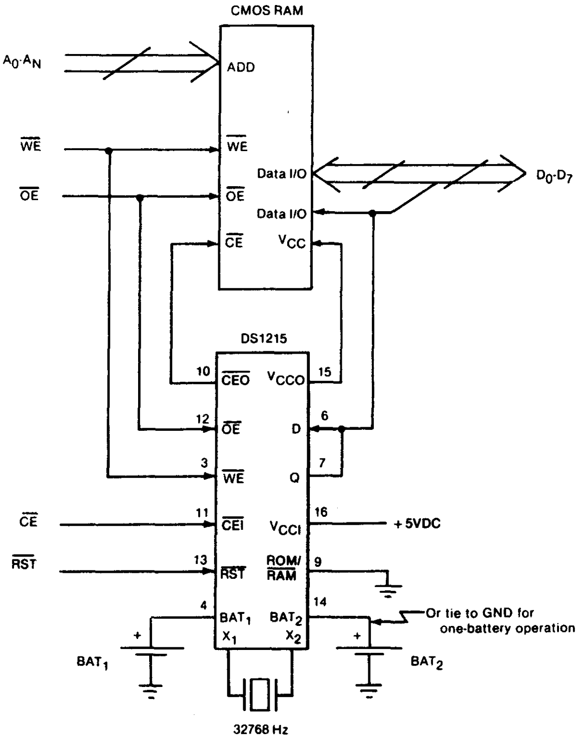
ZERO BITS

Registers 1, 2, 3, 4, 5, and 6 contain one or more bits which will always read logical 0. When writing these locations, either a logical 1 or 0 is acceptable.

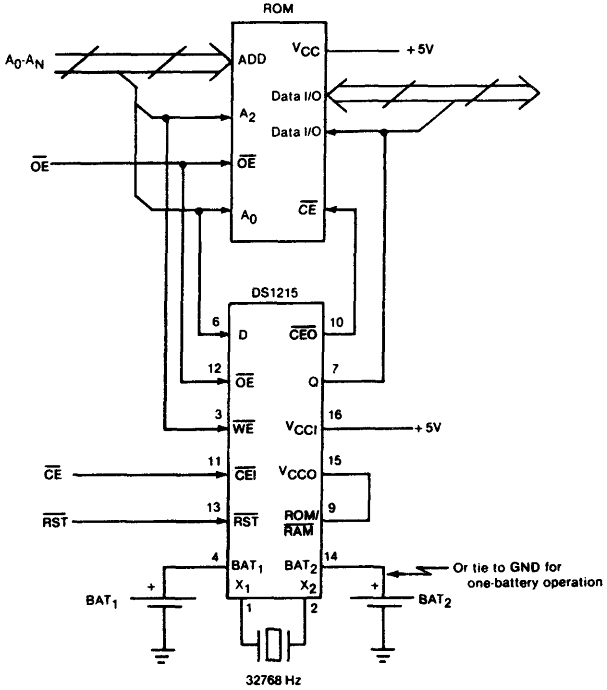
TIMECHIP BLOCK DIAGRAM Figure 3



RAM/TIMECHIP INTERFACE Figure 4



ROM/TIMECHIP INTERFACE Figure 5



ABSOLUTE MAXIMUM RATINGS*

Voltage on any Pin Relative to Ground -1.0V to +7.0V
Operating Temperature 0°C to 70°C
Storage Temperature -55°C to 125°C
Soldering Temperature 260°C for 10 Sec

*This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

RECOMMENDED D.C. OPERATING CONDITIONS (0°C to 70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	4.5	5.0	5.5	V	1
Logic 1	V_{IH}	2.2		$V_{CC}+0.3$	V	1
Logic 0	V_{IL}	−0.3		+0.8	V	1
V_{BAT1} or V_{BAT2} Battery Voltage	V_{BAT}	2.5		3.7	V	7

D.C. ELECTRICAL CHARACTERISTICS (0°C to 70°C, V_{CC} = 4.5 to 5.5V)

Supply Current	I_{CCI}			5	mA	6
Supply Current $V_{CCO}=V_{CCI}-0.2$	I_{CCO1}			80	mA	8
Input Leakage	I_{IL}	−1.0		+1.0	μA	
Output Leakage	I_{LO}	−1.0		+1.0	μA	
Output @2.4V	I_{OH}	−1.0			mA	2
Output @0.4V	I_{OL}			4.0	mA	2

(0°C to 70°C, V_{CC} 4.5V)

$\overline{CEO}$ Output	V_{OH1}	V_{CCI} or $V_{BAT}-0.2$			V	9
V_{BAT1} or V_{BAT2} Battery Current	I_{BAT}			1	μA	6
Battery Backup Current @$V_{CCO}=V_{BAT}-0.2V$	I_{CCO2}			10	μA	10

CAPACITANCE (t_A = 25°C)

PARAMETER	SYMBOL	MIN	UNITS	NOTES
Input Capacitance	C_{IN}	5	pF	
Output Capacitance	C_{OUT}	7	pF	

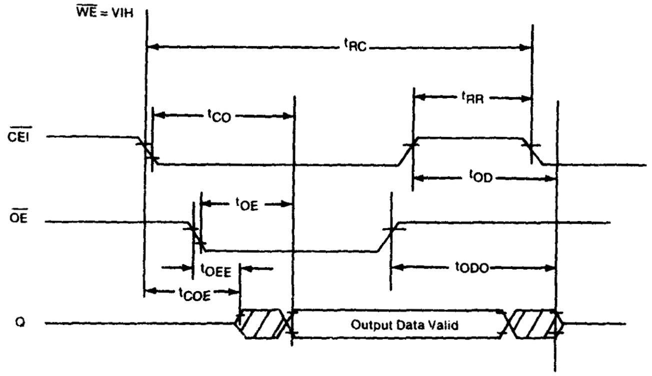
A.C. ELECTRICAL CHARACTERISTICS ROM/$\overline{RAM}$ = GND (0°C to 70°C, V_{CC} = 4.5 to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Read Cycle Time	t_{RC}	250			ns	
$\overline{CEI}$ Access Time	t_{CO}			200	ns	
$\overline{OE}$ Access Time	t_{OE}			100	ns	
$\overline{CEI}$ To Output Low Z	t_{COE}	10			ns	
$\overline{OE}$ To Output Low Z	t_{OEE}	10			ns	
$\overline{CEI}$ To Output High Z	t_{OD}			100	ns	
$\overline{OE}$ To Output High Z	t_{ODO}			100	ns	
Read Recovery	t_{RR}	50			ns	
Write Cycle	t_{WC}	250			ns	
Write Pulse Width	t_{WP}	170			ns	
Write Recovery	t_{WR}	50			ns	4
Data Set Up	t_{DS}	100			ns	5
Data Hold Time	t_{DH}	10			ns	5
$\overline{CEI}$ Pulse Width	t_{CW}	170			ns	
$\overline{RST}$ Pulse Width	t_{RST}	200			ns	
$\overline{CEI}$ Propagation Delay	t_{PD}	5	10	20	ns	2, 3
$\overline{CEI}$ High to Power Fail	t_{PF}			0	ns	

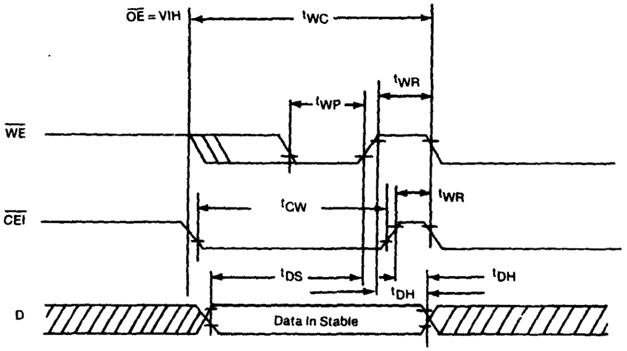
(0°C to 70°C, V_{CC}<4.5V)

Recovery at Power Up	t_{REC}			2	ms	
V_{CC} Slew Rate 4.5 - 3.0V	t_F	0			ms	

TIMING DIAGRAM—READ CYCLE TO TIMECHIP ROM/$\overline{\text{RAM}}$ = GND



TIMING DIAGRAM—WRITE CYCLE TO TIMECHIP ROM/$\overline{\text{RAM}}$ = GND



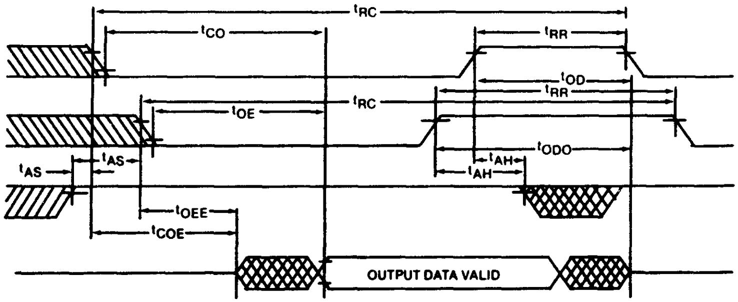
A.C. ELECTRICAL CHARACTERISTICS ROM/$\overline{\text{RAM}}$ = V_{CCO} (0 °C to 70 °C, V_{CC} = 5V ± 10%)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Read Cycle Time	t_{RC}	250			ns	
$\overline{\text{CEI}}$ Access Time	t_{CO}			200	ns	
$\overline{\text{OE}}$ Access Time	t_{OE}			200	ns	
$\overline{\text{CEI}}$ to Output in Low Z	t_{COE}	10			ns	
$\overline{\text{OE}}$ to Output in Low Z	t_{OEE}	10			ns	
$\overline{\text{CEI}}$ to Output in High Z	t_{OD}			100	ns	
$\overline{\text{OE}}$ to Output in High Z	t_{ODO}			100	ns	
Address Set Up Time	t_{AS}	20			ns	
Address Hold Time	t_{AH}			10	ns	
Read Recovery	t_{RR}	50			ns	
Write Cycle Time	t_{WC}	250			ns	
$\overline{\text{CEI}}$ Pulse Width	t_{CW}	170			ns	
$\overline{\text{OE}}$ Pulse Width	t_{OW}	170			ns	
Write Recovery	t_{WR}	50			ns	4
Data Set Up Time	t_{DS}	100			ns	5
Data Hold Time	t_{DH}	10			ns	5
$\overline{\text{RST}}$ Pulse Width	t_{RST}	200			ns	
$\overline{\text{CEI}}$ Propagation Delay	t_{PD}	5	10	20	ns	2,3
$\overline{\text{CEI}}$ High to Power Fail	t_{PF}			0	ns	

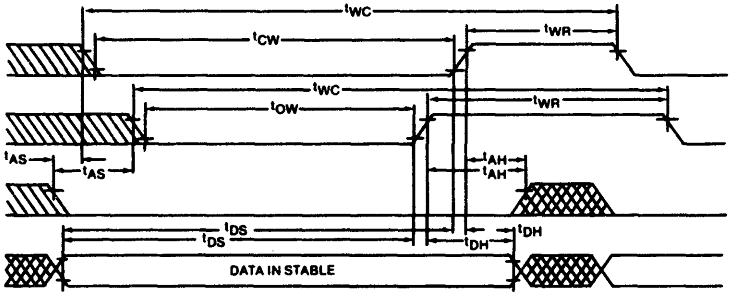
(0 °C to 70 °C, $V_{CC} < 4.5V$)

Recovery at Power Up	t_{REC}			2	ms	
V_{CC} Slew Rate 4.5 -3V	t_F	0			ms	

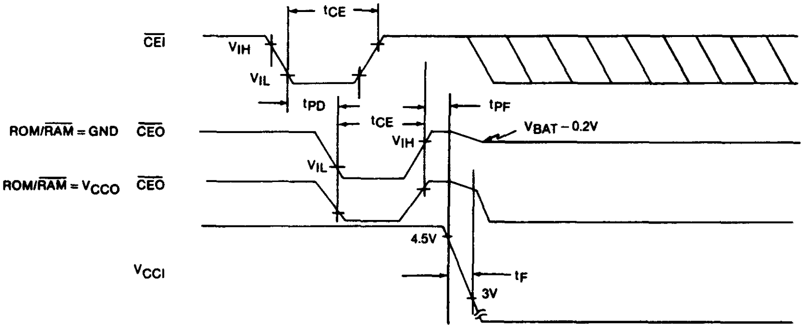
TIMING DIAGRAM—READ CYCLE ROM/$\overline{\text{RAM}}$ = V_{CCO}



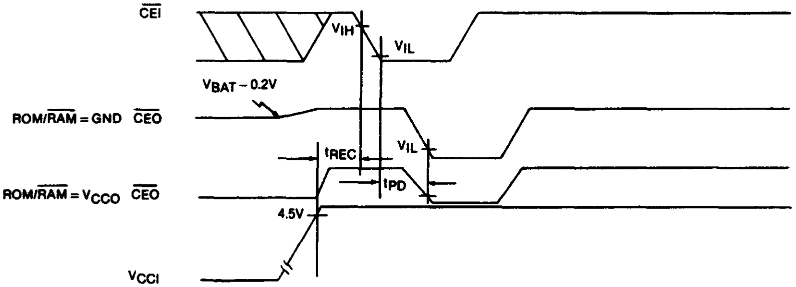
TIMING DIAGRAM—WRITE CYCLE ROM/$\overline{\text{RAM}}$ = V_{CCO}



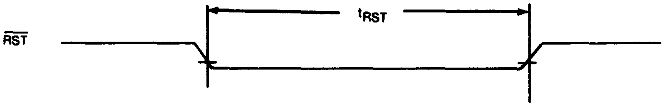
TIMING DIAGRAM—POWER DOWN



TIMING DIAGRAM—POWER UP



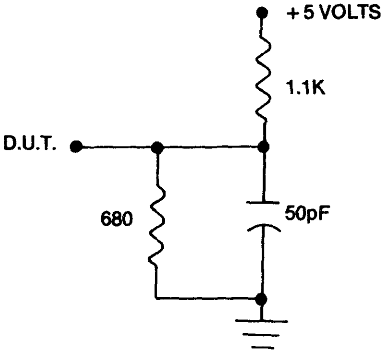
TIMING DIAGRAM—RESET FOR TIMECHIP



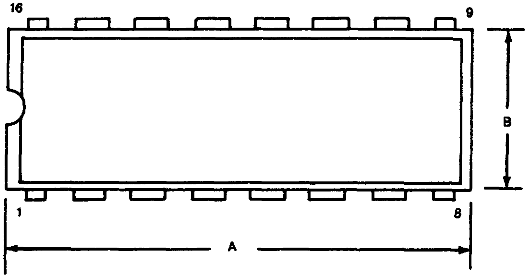
NOTES

1. All voltages are referenced to ground.
2. Measured with load shown in Figure 6.
3. Input pulse rise and fall times equal 10 ns.
4. t_{WR} is a function of the latter occurring edge of $\overline{WE}$ or $\overline{CE}$ in RAM mode or $\overline{OE}$ or $\overline{CE}$ in ROM mode.
5. t_{DH} and t_{DS} are functions of the first occurring edge of $\overline{WE}$ or $\overline{CE}$ in RAM mode or $\overline{OE}$ or $\overline{CE}$ in ROM mode.
6. Measured without RAM connected.
7. Trip point voltage for power fail detect. $V_{TP} = 1.26 \times V_{BAT}$ For 10% operation $V_{BAT} = 3.5V$ max.; for 5% operation $V_{BAT} = 3.7V$ max.
8. I_{CCO1} is the maximum average load current the DS1215 can supply to memory.
9. Applies to $\overline{CEO}$ with the ROM/$\overline{RAM}$ pin grounded. When the ROM/$\overline{RAM}$ pin is connected to V_{CCO}, $\overline{CEO}$ will go to a low level as V_{CCI} falls below V_{BAT}.
10. I_{CCO2} is the maximum average load current which the DS1215 can supply to memory in the battery backup mode.
11. Applies to all input pins except $\overline{RST}$. $\overline{RST}$ is pulled internally to V_{CCI}.

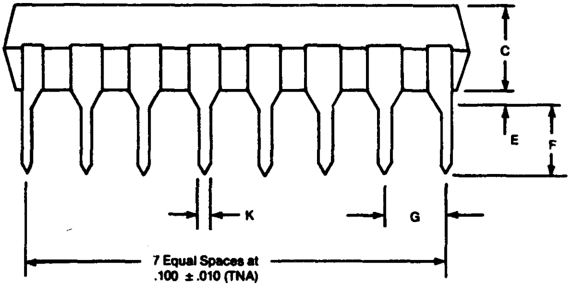
OUTPUT LOAD Figure 6

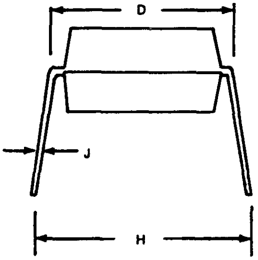


DS1215
TimeChip



DIM.	INCHES	
	MIN.	MAX.
A	.740	.780
B	.240	.260
C	.120	.140
D	.290	.310
E	.020	.030
F	.110	.130
G	.090	.110
H	.300	.350
J	.008	.012
K	.015	.021





DRAM/DMA CONTROL CHIP

TABLE OF CONTENTS

Tandy 1000TL DRAM/DMA Control IC consists of the following functional blocks:

- ROM/DRAM Decode Latch and Control signals
- Page Registers
- Memory Address Multiplexer

ROM/DRAM Decode Latch and Control signal

The ROM/DRAM Decode Latch block contains the circuitry to decode the CPU's Address bus A17 - A23 and to provide the necessary latched signals for controlling ROMS and DRAMS. CPU's Address A17 - A23 together with MC0 - MC2 determine which segment (bank) of memory is being selected based on one of six possible memory configurations. (see memory map figure 1.).The memory configurations ranging from 256 kilo bytes to 1.5 mega bytes.

Additional support for memory refresh is also provided in this block. During a Refresh Cycle, the assertion of REFRESH* will cause the circuitry to ignore the current address inputs and activate RAS0*, RAS1*, RAS2*, RAS3* and LMEGCS* outputs. Also MEMCYC and PRCLK signals are used for controlling the start of a memory access and timing for all RASx*s, MA0 - MA8 and CASx*s signals. (see Timing Diagram figure 2.).

ROMCS* is decoded from A17 - A23 and latched by ALE when HLDA is active. The ROM address ranges from 0E0000h to 0FFFFFh for the low address, EE0000h to EFFFFFh and FE0000h to FFFFFFh for high address. This output signal is asserted when any one of the three address ranges is detected and REFRESH* is inactive.

The two outputs LMEGCS* and MDBEN* are intended to be used as memory buffer enable signal. LMEGCS* is active whenever any memory access is made to an address below 010000h or when REFRESH* is active. MDBEN* is memory data bus buffer and becomes active whenever CASx*s or ROMCS* is active.

The ROM/DRAM Decode's internal latch is controlled by two input signals - HLDA and ALE. During a CPU Memory Cycle, ALE will enable the RAM Decode latch and allow the input from decoder to be transferred to the output pins. When ALE goes inactive, the decoder outputs is latched for the remainder of the cycle. Asserting HLDA will enable the decoder outputs to the output pins and force ROMCS* inactive. This block has one output signal that is unlatched. This output, AF16*, is intended to be used by external circuitry as an indication that a 16-bit memory transfer is taking place.

Page Register

At the time the 82C37A-5 - DMA Controller takes control of the address bus, the first operation comes in two bytes. The first byte is a lower address bus (SA0 through SA7) that is put directly on the S address bus by DMA controller. The second byte is a upper address (SA8 through SA15) that is on its data outputs, to be latched in the 74ALS373 internally by Address Strobe (AS) signal from DMA Controller.

Two 4 by 4 registers are used to perform Page Register function. During DMA Bus Cycle, the read function is controlled by the DMA Request Acknowledge signals - DACK2* and DACK3* in conjunction with HLDA* enables the Page Register to be output as the upper address (SA16 and A17 through A23). The write function is controlled by SA0 to SA3 in conjunction with PGREGWR* to latch data bits - XD0 to XD7 into Page Register.

Address Multiplexer

The Memory Address Multiplexer is used to provide the Row Address or Column Address and refresh counter that is required by Dynamic Rams. Additionally, it provides the drive and buffering capability for memory address bus MA0 to MA8. MA7 is generated from SA0 or SA8 which is multiplexed by REFRESH* signal. The addresses for the memory are multiplexed as shown below.

Equation For Multiplexed Memory Addresses

	Row Address (First)	Column Address (Second)
MA0 -----	SA1	SA9
MA1 -----	SA2	SA10
MA2 -----	SA3	SA11
MA3 -----	SA4	SA12
MA4 -----	SA5	SA13
MA5 -----	SA6	SA14
MA6 -----	SA7	SA15
MA7 -----	SA8/SA0	SA16
MA8 -----	SA17	SA18

2. MEMORY CONFIGURATION

OPTION	MC2	MC1	MC0	BANK0	BANK1	BANK2	CONTROL	ADDRESS RANGE
1	0	0	0		128K		Ras1	(000000-01FFFF)
						128K	Ras2	(020000-03FFFF)
2	0	0	1	512K			Ras0	(000000-07FFFF)
3	0	1	1	512K			Ras0	(000000-07FFFF)
					128K		Ras1	(080000-09FFFF)
4	0	1	0	512K			Ras0	(000000-07FFFF)
					128K		Ras1	(08FFFF-09FFFF)
						128K	Ras2	(100000-17FFFF)
5	1	1	1	512K			Ras0	(000000-07FFFF)
					128K		Ras1	(080000-09FFFF)
						512K	Ras2	(100000-17FFFF)
6	1	1	0	512K			Ras0	(000000-07FFFF)
					128K		Ras1	(080000-09FFFF)
						512K	Ras2	(100000-17FFFF)
						512K	Ras3	(180000-1FFFFF)
7	1	0	1	NOT DEFINED				
8	1	0	0	NOT DEFINED				

FIGURE 1. MEMORY CONFIGURATION.

3. DMA Control Logic Equations

```
/Bras0 = /la23 & /la22 & /la21 & /la20 & /la19 & /mc2 & mc0
         +/la23 & /la22 & /la21 & /la20 & /la19 & mc1
         +refresh;

/Bras1 = /la23 & /la22 & /la21 & /la20 & /la19 & /la18 & /la17 & /mc0
          & /mc1 & /mc2
         +/la23 & /la22 & /la21 & /la20 & la19 & /la18 & /la17 & mc1
         +refresh;

/Bras2 = /la23 & /la22 & /la21 & /la20 & /la19 & /la18 & la17 & /mc0
          &  /mc1 & /mc2
         +/la23 & /la22 & /la21 & la20 & /la19 & /la18 & /la17 & /mc0 & mc1
          & /mc2
         +/la23 & /la22 & /la21 & la20 & /la19 & mc1 & mc2
         +refresh;

/Bras3 = /la23 & /la22 & /la21 & la20 & la19 & /mc0 & mc1 & mc2
         +refresh;

/Bcas  = /la23 & /la22 & /la21 & /la20 & /la19 & /la18 & /mc2 & /mc1
          & /mc0 & /refresh
         +/la23 & /la22 & /la21 & /la20 & /la19 & /mc2 & mc0 & /refresh
         +/la23 & /la22 & /la21 & /la20 & /la19 & mc1 & /refresh
         +/la23 & /la22 & /la21 & /la20 & la19 & /la18 & /la17 & mc1
          & /refresh
         +/la23 & /la22 & /la21 & la20 & /la19 & /la18 & /la17 & /mc2 & mc1
          & /mc0 & /refresh
         +/la23 & /la22 & /la21 & la20 & /la19 & mc2  & mc1  & /refresh
         +/la23 & /la22 & /la21 & la20 & mc2 & mc1 & /mc0 & /refresh;

/B1meg  = /la23 & /la22 & /la21 & /la20+refresh;

/Bromcs =/la23 & /la22 & /la21 & /la20 & la19 & la18 & la17 & /refresh
         +la23 & la22 & la21 & la20 & la19 & la18 & la17 & /refresh
         +la23 & la22 & la21 & /la20 & la19 & la18 & la17 & /refresh;
```

4. Block Diagram

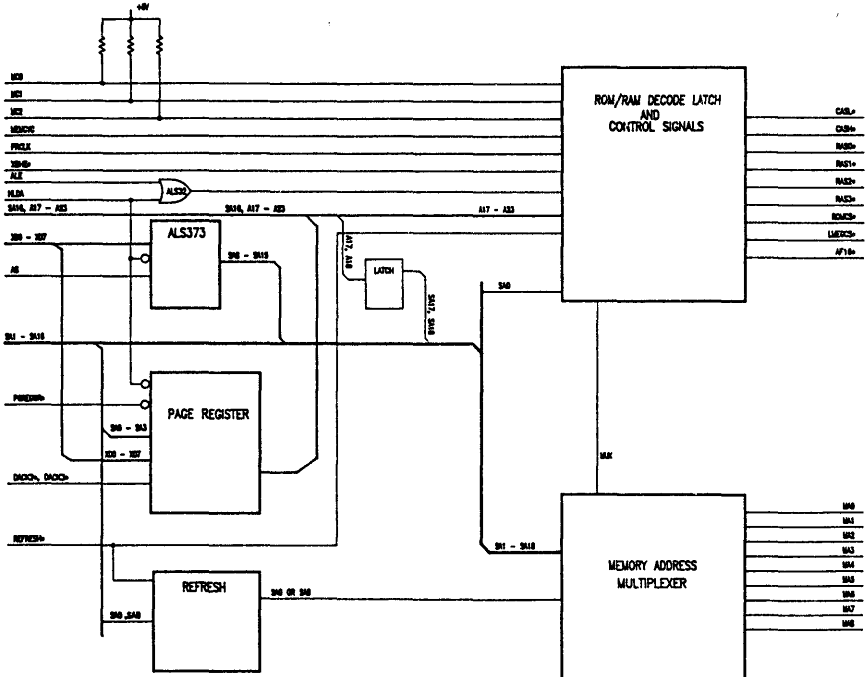


Figure 2. Block Diagram

SUBTITLE BLOCK DIAGRAM DRAM/DMA CONTROL CHIP	DWG NO		REV
	SCALE	SHEET	OF

5. PIN CONFIGURATION

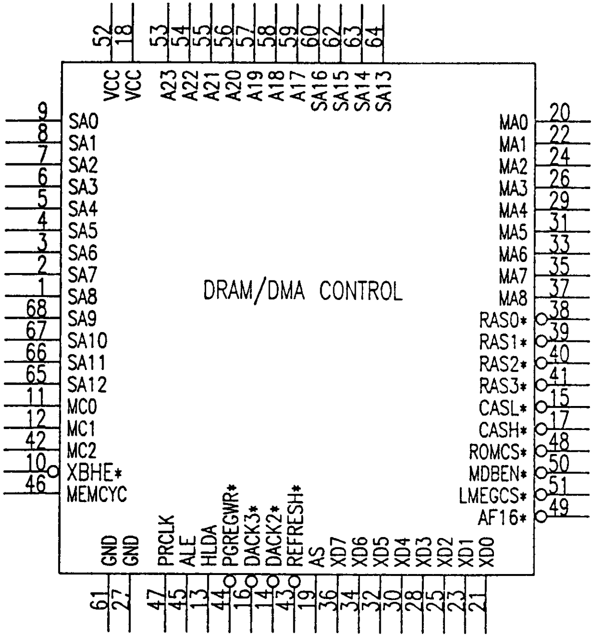


FIGURE 3. PIN CONFIGURATION

SUBTITLE PIN CONFIGURATION	DWG NO	REV
	SCALE SHEET	OF

6. INPUT/OUTPUT PINS FUNCTION AND DESCRIPTION

PIN#	PIN NAME	TYPE	DESCRIPTION
9	SA0	INPUT	CPU ADDRESS LINE
8	SA1	INPUT	CPU ADDRESS LINE
7	SA2	INPUT	CPU ADDRESS LINE
6	SA3	INPUT	CPU ADDRESS LINE
5	SA4	INPUT	CPU ADDRESS LINE
4	SA5	INPUT	CPU ADDRESS LINE
3	SA6	INPUT	CPU ADDRESS LINE
2	SA7	INPUT	CPU ADDRESS LINE
1	SA8	INPUT/OUTPUT	CPU ADDRESS LINE
68	SA9	INPUT/OUTPUT	CPU ADDRESS LINE
67	SA10	INPUT/OUTPUT	CPU ADDRESS LINE
66	SA11	INPUT/OUTPUT	CPU ADDRESS LINE
65	SA12	INPUT/OUTPUT	CPU ADDRESS LINE
64	SA13	INPUT/OUTPUT	CPU ADDRESS LINE
63	SA14	INPUT/OUTPUT	CPU ADDRESS LINE
62	SA15	INPUT/OUTPUT	CPU ADDRESS LINE
60	SA16	INPUT/OUTPUT	CPU ADDRESS LINE
59	A17	INPUT/OUTPUT	CPU ADDRESS LINE
58	A18	INPUT/OUTPUT	CPU ADDRESS LINE
57	A19	INPUT/OUTPUT	CPU ADDRESS LINE
56	A20	INPUT/OUTPUT	CPU ADDRESS LINE
55	A21	INPUT/OUTPUT	CPU ADDRESS LINE
54	A22	INPUT/OUTPUT	CPU ADDRESS LINE
53	A23	INPUT/OUTPUT	CPU ADDRESS LINE
45	ALE	INPUT	ADDRESS LATCH ENABLE Active HIGH - to latch generated RAS/CAS/LMEG.
43	REFRESH*	INPUT	REFRESH - Active LOW to initiate a refresh cycle for dynamic RAMs.
14	DACK2*	INPUT	8237 Channel 2 DMA ACKNOWLEDGE
16	DACK3*	INPUT	8237 Channel 3 DMA ACKNOWLEDGE
13	HLDA	INPUT	HOLD ACKNOWLEDGE Active HIGH - it indicates that the DMA has the system bus.
44	PGREGWR*	INPUT	PAGE REGISTER WRITE- active LOW to perform I/O WRITE cycle to the DMA Page Register.
46	MEMCYC	INPUT	MEMORY CYCLE - Active HIGH to initiate RAS/CAS/MA0-MA8 outputs.
19	AS	INPUT	ADDRESS STROBE - Active HIGH it latches the addres lines SA8-SA15 D0-D7 into external latch for DMA cycle

PIN#	PIN NAME	TYPE	DESCRIPTION
10	XBHE*	INPUT	BYTE HIGH ENABLE To enable the high memory data bytes D8-D15
47	PRCLK	INPUT	16MHZ CLOCK.
11	MC0	INPUT	MEMORY CONFIGURATION SELECT LSB.
12	MC1	INPUT	MEMORY CONFIGURATION SELECT.
42	MC2	INPUT	MEMORY CONFIG. SELECT MSB.
20	MA0	OUTPUT	MULTIPLEXED MEMORY ADDRESS - addressing required for DRAM memory.
22	MA1	OUTPUT	MULTIPLEXED MEMORY ADDRESS - addressing required for DRAM memory.
24	MA2	OUTPUT	MULTIPLEXED MEMORY ADDRESS - addressing required for DRAM memory.
26	MA3	OUTPUT	MULTIPLEXED MEMORY ADDRESS - addressing required for DRAM memory.
29	MA4	OUTPUT	MULTIPLEXED MEMORY ADDRESS - addressing required for DRAM memory.
31	MA5	OUTPUT	MULTIPLEXED MEMORY ADDRESS - addressing required for DRAM memory.
33	MA6	OUTPUT	MULTIPLEXED MEMORY ADDRESS - addressing required for DRAM memory.
35	MA7	OUTPUT	MULTIPLEXED MEMORY ADDRESS - addressing required for DRAM memory.
37	MA8	OUTPUT	MULTIPLEXED MEMORY ADDRESS - addressing required for DRAM memory.
17	CASH*	OUTPUT	COLUMN ADDRESS STROBE HIGH - Active LOW, it's used to select the high data byte MD8-MD15 for a DRAM access cycle.

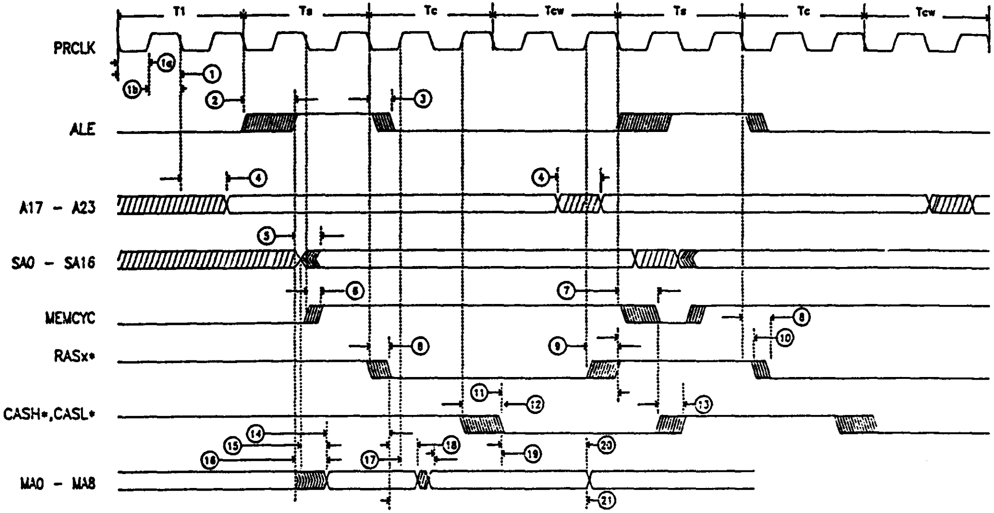
PIN#	PIN NAME	TYPE	DESCRIPTION
15	CASL*	OUTPUT	COLUMN ADDRESS STROBE LOW - Active LOW, it's used to select low data byte MD0-MD7 for a DRAM access cycle.
38	RAS0*	OUTPUT	ROW ADDRESS STROBE 0 - Active LOW, it's used for selecting DRAM Bank0.
39	RAS1*	OUTPUT	ROW ADDRESS STROBE 1 - Active LOW, it's used for selecting DRAM Bank1.
40	RAS2*	OUTPUT	ROW ADDRESS STROBE 2 - Active LOW, it's used for selecting DRAM Bank2.
41	RAS3*	OUTPUT	ROW ADDRESS STROBE 3 - Active LOW, it's used for selecting DRAM Bank2.
51	LMEGCS*	OUTPUT	LOWER MEGABYTE CHIP SELECT - Active LOW,it indicates that bellow 1 megabyte
50	MDBEN*	OUTPUT	MEMORY DATA BUS ENABLE Active LOW it is used to enable the data bus buffer.
49	AF16*	OUTPUT	AF16 Active LOW - it signals the control logic (CPU CNTL) that the memory cycle is a 16 bit 1 wait state cycle.
48	ROMCS*	OUTPUT	ROM CHIP SELECT Active LOW
21	XD0	INPUT	DATA BUS 0 for the peripheral bus.
23	XD1	INPUT	DATA BUS 1 for the peripheral bus.
25	XD2	INPUT	DATA BUS 2 for the peripheral bus.
28	XD3	INPUT	DATA BUS 3 for the peripheral bus.
30	XD4	INPUT	DATA BUS 4 for the peripheral bus.
32	XD5	INPUT	DATA BUS 5 for the peripheral bus.
34	XD6	INPUT	DATA BUS 6 for the peripheral bus.
36	XD7	INPUT	DATA BUS 7 for the peripheral bus.
18,52	VCC		+5V POWER SUPPLY
27,61	GND		GROUND

7. TIMING SPECIFICATIONS

SYM	PARAMETER	MIN	MAX	UNITS	REMARKS
T1	PRCLK Period	62.5	250	nsec	
T1a	PRCLK Low Time	25	125	nsec	
T1b	PRCLK High time	25	125	nsec	
T2	ALE Active Delav	6	53	nsec	
T3	ALE Inactive Delav	5	25	nsec	
T4	Address Delay	1	60	nsec	
T5	Latched Address From ALE Active	-	23	nsec	
T6	MEMCYC Active Delay From ↓PRCLK	5	25	nsec	
T7	MEMCYC Inactive Delav From ↓PRCLK	5	35	nsec	
T8	RASx* Active Delav From ↓PRCLK	5	45	nsec	at 100pf
T9	RASx* Inactive Delav From ↑PRCLK	5	30	nsec	
T10	RASx* Precharge Time	110		nsec	
T11	RASx* Hold Time	30		nsec	
T12	CASx* Active Delav Time From ↑PRCLK	5	40	nsec	at 165pf
T13	CASx* Inactive Delav Time From ↓ MEMCYC	5	30	nsec	
T14	Row Address Setup Time	5		nsec	
T15	Memory Address Delay Time From SA1 to SA16	0	50	nsec	at 200pf
T16	Memory Address Delav From ALE	8	50	nsec	
T17	Memorv Address Stable Delav From ↑ PRCLK		50	nsec	at 200pf
T18	Row Address Hold Time	20		nsec	
T19	Column Address Setup Time	5		nsec	
T20	Column Address Hold Time	30		nsec	
T21	Column Address Hold Time Referenced To RASx*	120		nsec	
T22	ALE ↑ to MDBEN ↑↓A/I		50	nsec	
T23	ALE ↑ to LMEGCS* ↓↑ A/I		50	nsec	
T24	ALE ↑ to ROMCS* ↓↑ A/I		50	nsec	CPUHLDA HIGH

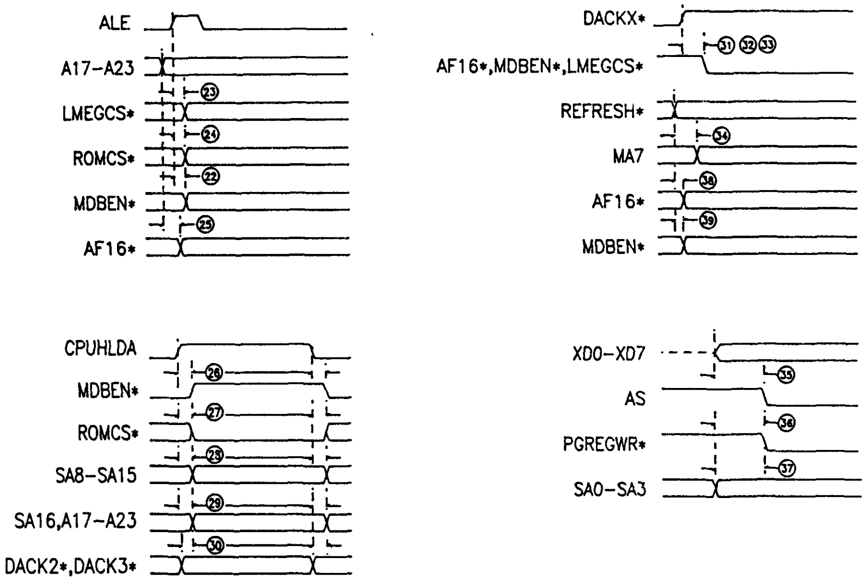
SYM	PARAMETER	MIN	MAX	UNITS	REMARKS
T25	A17-A23 ↑↓ to AF16* ↓↑ A/I		50	nsec	
T26	CPUHLDA↑↓ to MDBEN ↑I		50	nsec	
T27	CPUHLDA↑↓ to ROMCS* ↓↑ A/I		50	nsec	
T28	CPUHLDA↑↓ to SA8-SA15 ↓↑		50	nsec	
T29	CPUHLDA↑↓ to SA16,A17-A23 ↑↓		50	nsec	
T30	DACK2*, DACK3* ↑↓ to SA16, ↑↓ A17-A23		50	nsec	
T31 T32 T33	DACKx* to AF16*, MDBEN* LMEGCS*		80	nsec	
T34	REFRESH* ↑↓ to MA7↑↓A/I		50	nsec	at 200pf
T35	XD0-XD7 Setup to AS↓	100		nsec	
T36	XD0-XD7 Setup to PGREGWR*	100		nsec	
T37	SA0-SA3 Setup to PGREGWR*	100		nsec	
T38	REFRESH* ↓↑ to AF16*		50	nsec	
T39	REFRESH* ↓↑ to MDBEN*		50	nsec	

8. Timing Diagram.



SUBTITLE
TIMING DIAGRAM
CPU CYCLE

Timing Diagram Continue



TIMING DIAGRAM 2 2

9. ELECTRICAL SPECIFICATIONS

ABSOLUTE MAX RATINGS (NON-OPERATING, VSS=0.0V)

	MIN	MAX	UNITS
STORAGE TEMPERATURE	-65	+150	Degrees C.
VOLTAGE ON ANY PIN W.R.T GROUND	-0.5	7.0	Volts

OPERATING ELECTRICAL SPECIFICATIONS:

OPERATING AMBIENT	MIN	TYP	MAX	UNITS
AIR TEMP. RANGE	0	25	70	Degrees C
POWER SUPPLIES				
VCC	4.5	5.0	5.5	Volts
VSS	0	0	0	Volts

	MIN	TYP	MAX	UNITS
LEAKAGE CURRENT				
Vin = 0.0 v		20		Microamps
Vin = 5.0 v	-20			Microamps
INPUT VOLTAGES				
LOGIC "0" (Vil)		0.8		volts
LOGIC "1" (Vih)	2.0			volts

OUTPUT VOLTAGES CURRENT LOADING

	MIN	TYP	MAX	UNITS
LOGIC "0" (vol)			0.4	volts

MA[0]-MA[8] @ 8ma(min)
SXA8-SXA16,A17-A23 @ 4ma(min)
RASx* @ 4ma(min)
CASX* @ 8 ma(min)
Others must be able
to SINK minimum @ 2ma

	MIN	TYP	MAX	UNITS
LOGIC "1" (Voh)	2.4			volts

MA[0]-MA[8],
SXA8-SXA16,A17-A23 @ 8ma
CASx*,RASx* @ 4ma
CASX* @ 8 ma
Others must be able
to DRIVE minimum @ 2ma

INPUT CAPACITANCE

All inputs 0.0 < Vin < 5.0	10	picofarads

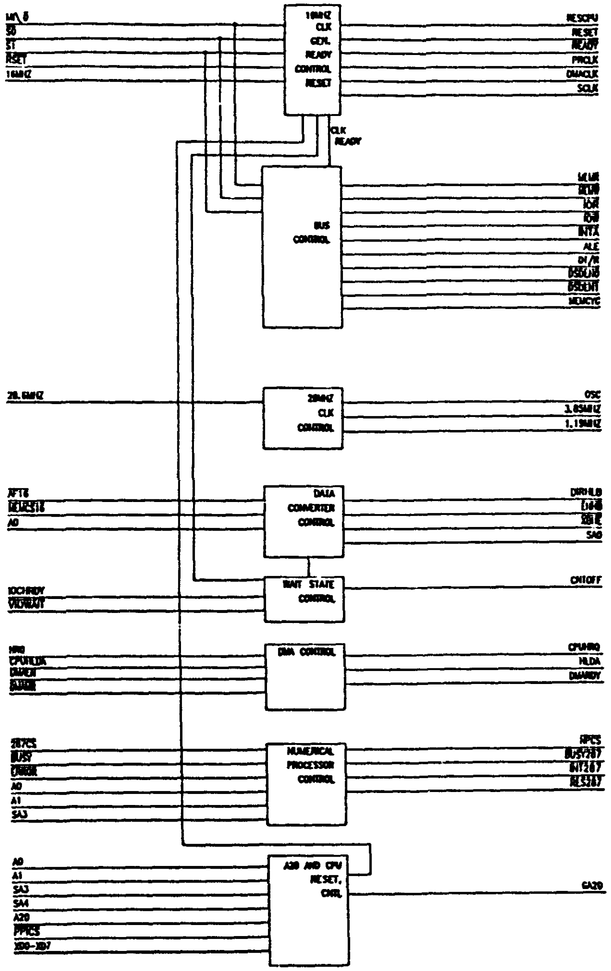
OUTPUT CAPACITANCE

MA[0]-MA[8]	200	picofarads
CAS	165	picofarads
RAS	100	picofarads
SA8-SA16	150	picofarads
All other outputs	50	picofarads

CPU Control Chip

Contents

Block Diagram



SUBTITLE
CPU CONTROL
BLOK DIAGRAM

DWG NO

REV

SCALE

SHEET

OF

Functional Description

The Tandy 1000 TL CPU Control IC consists of the following function blocks:

. - Clock Generation, Ready and Reset Control Logic
. - Bus Control Logic
. - Data Conversion Logic
. - Wait State Control Logic
. - DMA Arbitration Logic
. - Numerical Co-processor Control Logic
. - A20Gate and CPU Reset Logic

Clock Generation, Ready and Reset Control Logic

The clock generation logic includes the clock inputs used to derive all of the system clocks. The 16 MHz clock input is used to generate the CPU clock (PRCLK), and is twice the CPU operation frequency of 8 MHz. The 28.6 MHz clock input is used to generate the 14.31818 MHz clock (OSC) and other clocks required by the system.

Three clocks are generated from the 28.6 MHz clock input. These are OSC, 3.58 MHz, and the 1.19 MHz clock outputs. The 28.6 MHz signal input is divided by 2 to generate the OSC output which is 14.31818 MHz. The 28.6 MHz clock is divided by 8 to generate the 3.58 MHz clock which is used by the sound generator circuit. The 28.6 MHz clock is also divided by 24 to generate the 1.19 MHz clock that is used by the Interval Timer 8254.

All other system clocks are generated from the 16 MHz clock input. PRCLK is used to drive the CPU, the Co-processor, and the DRAM/DMA Control IC. PRCLK is output as 16 MHz in the 8 MHz (FAST) mode and is divided by 2 to 8 MHz in the 4 MHz (SLOW) mode. PRCLK output buffer has sufficient drive capability to meet the 3.8 volt minimum Vih requirement of the 80286/80287 clock inputs.

SCLK and DMACLK are system clocks generated from 16 MHz. SCLK is generated by dividing PRCLK by 2 and DMACLK is generated by dividing PRCLK by 4. Both SCLK and DMACLK are held in a LOW state immediately following a reset and will not start operation until the CPU issues the first bus cycle by asserting S1* LOW. Immediately following S1* asserted LOW, SCLK and DMACLK will make their first LOW to HIGH transition at the falling edge of PRCLK during the start of Tc. This will synchronize SCLK and DMACLK with PRCLK. Refer to timing diagrams for an illustration of the startup of SCLK and DMACLK.

A ready signal (READY*) is generated by the CPU Control IC to allow the CPU to operated with slower devices such as peripherals and slow memory. READY* is synchronized with PRCLK in this control block by the control logic of READY* located in the Wait State Control Logic.

Two reset output signals are generated by the CPU Control IC to provide a reset to the main system and a separate reset to the CPU. RESET which is active HIGH is a synchronized reset and is used for reseting the main system. RESCPU is dedicated to the CPU for proper reseting of the 80286. Both RESET and RESCPU are generated when RES* input is asserted LOW to indicate a power-on reset or when RES* is asserted LOW by a reset switch. RESCPU is also generated when a Shutdown condition of the CPU is detected. When a Shutdown condition is detected, RESCPU is asserted HIGH for 16 PRCLK cycles and then negated to assure proper CPU operation. RESCPU can also be generated by doing an I/O write to Port 068 with bit 2 = 0. This will generated RESCPU to reset the CPU and can be used to jump from protected mode to real time mode of the CPU.

The RES* input is buffered internal to the CPU Control IC with a Schmitt trigger input buffer. This signal should be held in a LOW state during power-up for at least 10 msec or until all operating voltages have reached their specified range of operation.

Bus Control Logic

The Bus Control Logic of the CPU Control IC generates several Memory and I/O command and control signals that are issued by the 80286. There are two types of control signals that are generated as output signals. The command outputs are the first type of control signals generated which are decoded from the CPU status inputs MI/O*, S1*, S0*. The generated output command signals are MEMR*, MEMW*, IOR*, IOW*, INTA* which determine which type of cycle is to be performed. The second type of signals are the control signals which is ALE, DT/R*, DSDEN0*, DSDEN1*, and MEMCYC. These signals latch the address from the CPU, control the direction and enabling of the data bus buffers, and determine the start of an on board memory cycle. Table 1. contains a list of the command output signals that are generated from the CPU status line inputs.

MI/O*	S1*	S0*	Type of Cycle
0	0	0	Interrupt Acknowledge
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	None; Idle
1	0	0	Halt or Shutdown
1	0	1	Memory Read
1	1	0	Memory Write
1	1	1	None; Idle

Table 1. CPU Control Signal Generation

The state machine that controls the output timing of the command and control signals has three bus states, which are the Idle state (Ti), the Status state (Ts), and the command state (Tc). The Idle state (Ti) is generated when the CPU is not actively issuing a bus cycle. During an Idle state, all command and control output signals are in an inactive condition. The beginning of a bus cycle is detected when the CPU asserts S1* or S0*. The state machine will start a Status state (Ts) and will assert ALE to an active HIGH state until the end of Ts. At the end of the Status state, the state machine enters the command state (Tc) and will assert the decoded command issued by the CPU. The command signal may be delayed by one half bus cycle or one PRCLK clock cycle, if the conditions exist to produce a command delay. A Command delay will be generated on all I/O cycles, all Memory cycles in which both AF16* and MEMCS16* are negated are inactive HIGH, and on all INTA* cycles.

For the control of the data bus buffer, three control signals are generated which are DT/R*, DSDEN0*, and DSDEN1*. DT/R* is used to control the direction of the data bus to and from the CPU. DSDEN0* and DSDEN1* are enable signals for the data bus buffers. DSDEN0* is qualified with A0 to control the lower 8 bits of the data bus (D0-D7). DSDEN1* is qualified with BHE* to control the upper 8 data bits (D8-D15). Control of the upper and lower data bits independently is required in the AT type architecture to control the 8 to 16 bit data conversions.

ALE is a control signal used to latch the address line from the CPU so that the address will remain stable throughout the complete command cycle. ALE as generated when either S0* or S1* is asserted from the CPU to start a bus cycle. This allows the address latches to be enabled as early as possible to provide sufficient address setup time to the on board memory array. ALE is negated at the start of the command state (Tc) to latch the address while they are guaranteed from the CPU.

The last control signal generated is MEMCYC which is used to control the start of a memory access to the on board memory array. MEMCYC is asserted HIGH at the middle of the Status state (Ts) and is held active during the complete bus cycle. MEMCYC is negated at the end of a bus cycle or the end of the command state (Tc).

Bus Conversion Logic

This block of the CPU Control IC provides the logic to control the 16 bit transfers to and from an 8 bit device or memory. The conversion logic detects when a conversion is required, asserts a wait to the CPU by asserting the READY* line to a HIGH or inactive state, and then generates the required control signals to the data buffers to perform the conversion. A conversion is required during all 16 bit transfers to all I/O devices, the Interrupt Controller, and 8 bit memory. An 8 bit memory cycle is detected by sampling the state of AF16* and/or MEMCS16* at the beginning of a memory cycle. If both are inactive HIGH then the conversion logic acknowledges it as an 8 bit memory cycle. The control signals that are generated are DIRHLB (Direction High Low Byte), ENHLB (Enable High Low Byte), and CNTLOFF (Control Off). ENHLB enables the conversion buffer during the conversion cycle and DIRHLB determines the direction of the buffer. CNTLOFF is used to latch the lower 8 bits (D0-D7) during a 16 bit read from an 8 bit device. SA0 is also controlled and generated by the conversion logic so the even byte is read first then SA0 is toggled to a HIGH state so the odd byte will be read.

Wait State Control Logic

The Wait State Control Logic is used to allow slow memory and peripheral devices to be used with a faster CPU. Wait states are controlled by the CPU Control IC by negating the READY* line to the CPU to inactive state. The CPU will not end the cycle in progress until the READY* line is reasserted to an active LOW state. The READY* line is negated at the middle of the Status state (Ts) to an inactive HIGH to guarantee recognition of a wait state by the CPU. The READY* line is reasserted at the middle of the Command state or Wait State to end the existing cycle.

There are several default wait state cycles that are generated by the CPU Control IC for control of all bus cycles. During a 16 bit memory cycle which is determined by the assertion of AF16* or MEMCS16*, a default of one wait state is automatically inserted. During a 8 bit memory cycle which is determined when both AF16* and MEMCS16* are negated, a default of four wait states are automatically inserted. All I/O cycles have a default of four wait states inserted to guarantee timing compatibility with existing I/O channel add in boards. All of the default wait state are the same in the 8 MHz (FAST) mode and the 4 MHz (SLOW) mode except for the I/O bus cycles. During an I/O cycles in the 4 MHz (SLOW) mode, a default of only two wait states are inserted. This allows only the memory cycles to be effected to slow down the program speed without degrading the performance of I/O accesses.

All bus cycles can be extended above the default number of wait states by negating the IOCHRDY (I/O Channel Ready) to inactive LOW. The CPU Control IC will continue to insert wait states to the CPU until IOCHRDY is released allowing it to be asserted. IOCHRDY should not be held LOW for more that 15 usec. because Refresh to the DRAMs will be inhibited.

DMA Arbitration Logic

The DMA Arbitration Logic in the CPU Control IC control the arbitration of the bus between the CPU and the DMA Controller. The arbitration logic detects when the DMA is requesting the bus when HRQ (Hold Request) is asserted HIGH. After HRQ is recognized, the arbitration logic will assert CPUHRQ to the CPU to request release of the bus. The CPU will respond to CPUHRQ after finishing the cycle in progress by suspending operation and asserting CPUHLDA to the CPU Control IC. The arbitration Logic will then tri-state the output command signals and generate HLDA to the DMA, allowing it to take control over the bus.

All DMA cycles have a default of one wait state automatically inserted to maintain compatibility with existing I/O channel boards and peripheral devices. The DMA cycle can be extended by and I/O device by applying a LOW state to the IOCHRDY signal. The DMA Controller will be held in a wait state until IOCHRDY is returned to a HIGH state. Refer to Timing Diagrams for Timing specifications of a DMA cycle and DMA Arbitration operation.

The CPU Control IC also controls two functions required in the 80286 architecture during a DMA cycle. In order to guarantee the integrity of an Interrupt Acknowledge (INTA) cycle, which requires two bus cycles to complete, a DMA arbitration inhibit is included in the arbitration logic. After an INTA cycle has started, a DMA cycle will be inhibited until the CPU performs a memory write cycle. The memory write cycle will normally be executed after the INTA cycle due the stack operation of the CPU required to service an interrupt routine. The inhibit will guarantee the a DMA cycle can not be allowed until both bus cycles of an INTA cycle is complete.

Numerical Co-processor Control Logic

The Numerical Co-processor Control Logic provides an interface for the 80286 CPU to control an 80287 Co-processor. This logic controls the decoding required to select and reset the 80287, control of the BUSY* and ERROR* signals from the 80287 to the CPU, and generating the interrupt signal during an error condition.

The input signal 287CS* is an I/O decode at 0F0-0FFh I/O address. 287CS* is used by the CPU Control IC to generate the Numerical Co-processor Chip Select (NPCS*), and the reset signal (RES287*) to the 80287. Refer to the I/O decode table for details about the further internal decode done by the CPU Control IC.

When the 80287 receives a command to perform a task, it will output the BUSY* signal which is input to the CPU Control IC. The CPU Control IC will then assert BUSY287* to the CPU. During normal operation of the 80287, when it finishes the task, it will negate BUSY* which will in turn negate BUSY287* to the CPU. If the ERROR input from the 80287 is asserted during this busy time, which indicates a 80287 error, BUSY287* output is latched and INT287* is asserted LOW to generate an interrupt to the CPU. Both BUSY287* and INT287* will remain latched in an active LOW state until they are cleared by writing to the I/O address 0F0h or 0Flh. These signals are cleared after a system reset.

RES287* is generated to reset the 80287. It is generated by writing to I/O address 0Flh or when a system reset is issued.

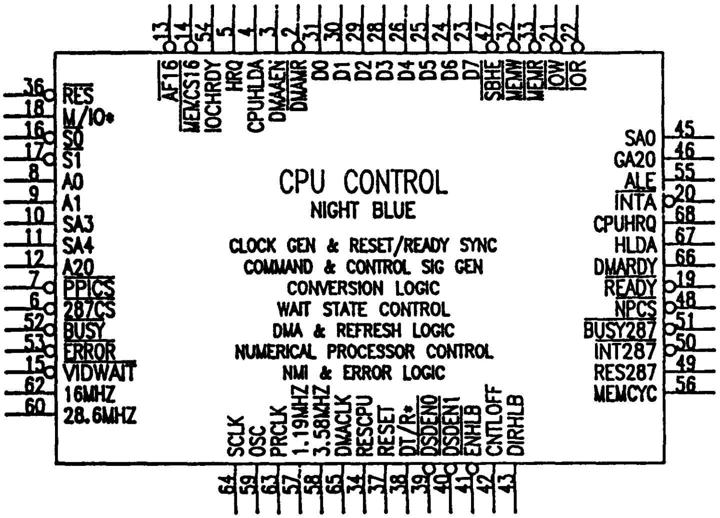
A20 and CPU Reset Control Logic

The CPU Control IC incorporates the logic required to control the Address line A20 and to reset the CPU. In the PC architecture, A20 must be held in a LOW state so the some programs that do wrapping will function correctly. After a reset, the output signal GA20 is help in a LOW state. If a program needs to address above the 1 MEG limit, OUT to I/O address 068h with data bit 1 to a HIGH or "1" state will allow A20 to be muxed to GA20.

The CPU can reset itself by doing an I/O wtrie to address 068h with bit 2 at a LOW or "0" state. After a reset has occurred, by Reading I/O address 068, it can be determined it was a power up reset or if the CPU reset itself. Refer to the I/O address map for further details.

```
Address                 Description

062                 Port C (Write Only)

Bit
7 6 5 4 3 2 1 0
¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶
¶ ¶ ¶ ¶ ¶ ¶ ¶ +---  Reserved
¶ ¶ ¶ ¶ ¶ ¶ +-----  Reserved
¶ ¶ ¶ ¶ ¶ +-------  Reserved
¶ ¶ ¶ ¶ +---------  0 = 4 MHz (SLOW) Mode
¶ ¶ ¶ ¶             1 = 8 MHz (FAST) Mode
¶ ¶ ¶ ¶                 Default after Reset
¶ ¶ ¶ +-----------  Reserved
¶ ¶ +-------------  Reserved
¶ +---------------  Reserved
+-----------------  Reserved


068                 Port I (Read and Write)


Bit
7 6 5 4 3 2 1 0
¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶
¶ ¶ ¶ ¶ ¶ ¶ ¶ +---  Not Used
¶ ¶ ¶ ¶ ¶ ¶ +-----  0 = GA20 always 0
¶ ¶ ¶ ¶ ¶ ¶             Default after Reset
¶ ¶ ¶ ¶ ¶ ¶         1 = A20 Muxed to GA20
¶ ¶ ¶ ¶ ¶ +-------  Write Mode  0 = Reset CPU
¶ ¶ ¶ ¶ ¶                       1 = No Effect
¶ ¶ ¶ ¶ ¶           Read Mode   0 = Power On Reset
¶ ¶ ¶ ¶ ¶                       1 = CPU Reset
¶ ¶ ¶ ¶ +---------  Not Used
¶ ¶ ¶ +-----------  Not Used
¶ ¶ +-------------  Not Used
¶ +---------------  Not Used
+-----------------  Not Used

0F0                 Clear Numerical Co-processor Busy
0F1                 Reset Numerical Co-processor
0F8-0FF             Numerical Co-processor Chip Select
```

Pin Configuration



Functional Pin Description

Pin#	Pin Name	Type	Description
62	16MHZ	I	16 MHZ Clock Input
63	PRCLK	O	80286 Processor Clock - 16:8 MHZ
64	SCLK	O	System Clock - 8:4 MHZ
65	DMACLK	O	DMA Controller Clock - 4:2 MHZ
60	28.6MHZ	I	28.63636 MHZ Clock Input
59	OSC	O	14.31818 MHZ Clock
58	3.58MHZ	O	3.5795 MHZ Clock for sound chip
57	1.19MHZ	O	1.19 MHZ Clock for Interval Timer chip 8254.
18	MI/O	I	Memory Input/Output from the CPU. Indicates a memory access when HIGH and a I/O access when LOW. Used to generate the memory and I/O command signals for the system.
17	S1*	I	Status Line 1 from the CPU
16	S0*	I	Status Line 0 from the CPU
19	READY*	O	Ready signal to the 80286 Processor Indicates the current bus be completed when LOW.
68	CPUHRQ	O	CPU hold request signal for the CPU. Indicates DMA transfers by the DMA controller when HIGH. It is also active during refresh cycles.
4	CPUHLDA	I	Hold Acknowledge signal from CPU. Indicates the CPU granting a DMA cycle to the DMA controller when HIGH. And causes all command signals to be tri-stated provided the CNTLOFF output is LOW
33	MEMR*	O	Memory Read Command output signal instructs a memory device to place data on the bus when LOW. It is also active during refresh.

32	MEMW*	I/O	Memory Write Command Input/Output signal. Instructs a memory device to read the data on the bus when LOW.
22	IOR*	I/O	Input/Output Read signal. Indicates a Read cycle is performed with an I/O device or port when LOW.
21	IOW*	I/O	Input/Output Write signal. Indicates a Read cycle is performed with an I/O device or port when LOW.
20	INTA*	O	Interrupt Acknowledge for the Interrupt Controller. It is used by the Interrupt controller to output the interrupt vector onto the data bus when LOW.
55	ALE	O	Address Latch Enable. It is used to hold the address during bus cycle when HIGH.
56	MEMCYC	O	Memory cycle signal. It is used to generate memory control signal, RAS, MUX and CAS when HIGH.
36	RES*	I	Reset signal. It is connected to the power good and used to reset the system when LOW.
37	RESET	O	Reset output signal. It is a synchronized reset signal for general system reset when HIGH.
34	RESCPU	O	CPU Reset output signal. It is used to reset CPU when HIGH.
47	XBHE* (SBHE*)	I/O	Bus High Enable Input/Output signal. It is used to enable the high byte data bus signals when LOW
45	SA0	O	System Address 0.
8	A0	I	Address 0 input signal from CPU. It is used to generate the enable signal for the data bus.

9	A1	I	Address 1 input signal from CPU. It is used to detect the SHUT DOWN condition of the CPU.
10	SA3	I	System Address 3 input signal. It is used to generate the chip select and reset signal for 80287.
11	SA4	I	System Address 4 input signal. It is used to generate the chip select.
12	A20	I	Address 20 from the CPU. It is used to generate GA20 signal when CPUHLDA LOW.
46	GA20	O	Gated Address 20. Address 20 is being negated by XD1 and PORTI (internal). When XD1 is LOW, Gated A20 on the CPU address bus is forced LOW. When XD1 is HIGH, GA20 is transmited as Address 20.
7	PPICS*	I	Programable Peripheral Interface Chip Select input signal. It is used to generate Chip Select Signal signal for the peripheral interface device when is LOW.
6	287CS*	I	287 Chip Select input signal. It is used for generating the Numerical Processor Select NPCS for 80287 when is HIGH.
5	HRQ	I	Hold Request input signal from DMA controller. It is used to generate the CPU Hold Request Signal when is HIGH.
67	HLDA	O	Hold Acknowledge output signal for DMA Controller. It is used to provide Hold Acknowledge for DMA controller when is HIGH.
3	DMAAEN	I	DMA Address Enable input Signal from DMA Controller. It is used to provide enable signal for any I/O device during DMA Access to the system memory when is HIGH.

2	DMAMR*	I	DMA Memory Read signal from DMA controller. It is used to generate Memory Read - MEMR* signal when is LOW.
66	DMARDY	O	DMA Ready output signal for DMA Controller. It is used to extend memory Read and Write cycles from the DMA Controller for slower memory or I/O device when is HIGH.
52	BUSY*	I	Busy input signal from 80287. It indicates that 80287 is currently executing a command when is LOW.
53	ERROR*	I	Error input signal from 80287. It indicates an unmasked error condition exists.
51	BUSY287*	O	Busy 80287 output signal for the CPU. It indicates to the processor the operating condition of the 80287 when is LOW.
50	INT287*	O	Interrupt 80287 output signal for the Interrupt controller. It is the interrupt request from 80287
49	RES287	O	Reset 80287 output signal for the 80287. It is used to reset to the 80287 when is HIGH.
48	NPCS*	O	Numerical Processor Chip Select output signal for the 80287. It is used to select the 80287 device when is LOW.
38	DT/R*	O	Data Transmit/Received output signal. It is used to determine the data direction to and from local data bus. It is a write bus cycle when is HIGH and read bus cycle when is LOW.
39	DSDEN0*	O	Data Strobe Data Enable 0 output. It is used to enable the data transceivers connected to the low byte (D0-D7) when is LOW.
40	DSDEN1*	O	Data Strobe Data Enable 1 output signal. It is used to enable the data transceivers connected to the high byte (D8-D15) data bus when is LOW.

41	ENHLB*	O	Enable High To Low byte output signal. It is used to perform the high to low conversion when is LOW.
42	CNTLOFF	O	Control Off output signal. It is used to enable the low data bus latch during byte accesses when is LOW.
43	DIRHLD	O	Direction High to Low byte output signal. It is used to perform high to low byte conversion when is HIGH.
13	AF16*	I	AF16* output signal. It is used to control the 16 bit memory accesses and to inhibit the command delays for memory accesses by I/O device when is LOW.
14	MEMCS16*	I	Memory Chip Select input signal. It is used to inhibit command delays when 16 bit memory accesses are made when is LOW.
54	IOCHRDY	I	I/O Channel Ready input signal from I/O device. It is used generate wait states in I/O or memory accesses by I/O device when is HIGH.
15	VIDWAIT*	I	Video Wait input signal from video controller. It is used to generate Video delay ready signal when is HIGH.
31	D0	I/O	Data Bus Bit 0 of the peripheral data bus.
30	D1	I/O	Data Bus Bit 1 of the peripheral data bus.
29	D2	I/O	Data Bus Bit 2 of the peripheral data bus.
28	D3	I/O	Data Bus Bit 3 of the peripheral data bus.
26	D4	I/O	Data Bus Bit 4 of the peripheral data bus.

25	D5	I/O	Data Bus Bit 5 of the peripheral data bus.
24	D6	I/O	Data Bus Bit 6 of the peripheral data bus.
23	D7	I/O	Data Bus Bit 7 of the peripheral data bus.
1,35	VCC	PWR	Power Supply.
27,44 61	VSS	GND	Ground.

ELECTRICAL SPECIFICATIONS

ELECTRICAL PARAMETERS

ABSOLUTE MAX RATINGS (NON-OPERATING, VSS=0.0V)

	MIN	MAX	UNITS
STORAGE TEMPERATURE	-65	+150	Degrees C.
VOLTGAE ON ANY PIN W.R.T GROUND	-0.5	7.0	Volts

OPERATING ELECTRICAL SPECIFICATIONS:

OPERATING AMBIENT	MIN	TYP	MAX	UNITS
AIR TEMP. RANGE	0	25	70	Degrees C
POWER SUPPLIES				
VCC	4.5	5.0	5.5	Volts
VSS	0	0	0	Volts

	MIN	TYP	MAX	UNITS
LEAKAGE CURRENT				
Vin = 0.0 v		20		Microamps
Vin = 5.0 v	-20			Microamps

DC ELECTRICAL CHARACTERISTICS

<u>INPUT VOLTAGES AND CURRENTS</u>	MIN	TYP	MAX	UNITS
LOGIC "0" (Vil) @ 8ma			0.8	Volts
LOGIC "1" (Vih) @ 8ma	2.0			Volts
INPUT CURRENT			+10	Microamps
All inputs 0.0 ¼ Vin ¼ 5.0	10			Picofarads

<u>OUTPUT VOLTAGES CURRENT LOADING</u>	MIN	TYP	MAX	UNITS
LOGIC "0" (Vol) @ 8ma			0.45	Volts
LOGIC "1" (Voh) @ 8ma	2.4			Volts

CURRENT LOADING AND CAPACITANCE OF EACH OUTPUT

At Vol = 0.45 volts, Voh = 2.4 volts

PIN NAME	MIN	UNITS	CAPACITANCE	NOTES
OUTPUT				
RESCPU, $\overline{\text{NPCS}}$, $\overline{\text{BUSY287, INT287}}$ RES287,CPUHRQ, HLDA, $\overline{\text{DIRHLB}}$ CNTLOFF, $\overline{\text{ENHLB}}$, DMARDY	2	ma	20pf	
RESET, GA20	2	ma	50pf	
READY, MEMCYC, DT/$\overline{\text{R}}$, $\overline{\text{DSDEN0}}$ DSDEN1	4	ma	20pf	
SCLK, DMACLK, OSC, 1.19MHZ 3.58MHZ	4	ma	50pf	
PRCLK, $\overline{\text{INTA}}$	8	ma	50pf	
ALE	8	ma	50pf	
$\overline{\text{MEMR}}$	8	ma	80pf	
INPUT/OUTPUT				
XD0 - XD7	2	ma	80pf	
$\overline{\text{XBHE}}$	4	ma	50pf	
$\overline{\text{MEMW}}$	8	ma	80pf	
$\overline{\text{IOR}}$, $\overline{\text{IOW}}$	8	ma	115pf	
SA0	4	ma	120pf	

Timing Diagram Specifications

SYM	DESCRIPTION	MIN	MAX	UNITS	NOTES
1	PRCLK Period	62	250	ns	
2	PRCLK Low Time	25	125	ns	
3	PRCLK High Time	25	125	ns	
4	$\overline{S0}$,$\overline{S1}$,M/$\overline{IO}$ Setup time to PRCLK ↓	20		ns	
5	$\overline{S0}$,$\overline{S1}$,M/$\overline{IO}$ Hold time to PRCLK ↓	1		ns	
6	ALE active from $\overline{S0}$,$\overline{S1}$ ↓	5	25	ns	
7	ALE inactive from PRCLK ↓	5	25	ns	
7A	OSC delay from 28.6 MHZ		30	ns	
8	MEMCYC active from PRCLK ↓	5	25	ns	
8A	3.85 MHZ delay from OSC		15	ns	
9	MEMCYC inactive delay from PRCLK ↓	8	35	ns	
9A	1.19 MHZ delay from OSC		15	ns	
10	Command active delay from PRCLK ↓	5	45	ns	
11	Command inactive delay from PRCLK ↓	5	45	ns	
12	DT/$\overline{R}$ active delay from PRCLK ↓	5	50	ns	Read
13	DT/$\overline{R}$ inactive delay from PRCLK ↓	10	40	ns	Read
14	$\overline{DSDEN0,1}$ active delay from PRCLK ↓	10	50	ns	Read
15	$\overline{DSDEN0,1}$ inactive delay from DT/$\overline{R}$	3		ns	Read
16	$\overline{DSDEN0,1}$ inactive delay from PRCLK ↓	8	50	ns	Read
17	DT/$\overline{R}$ inactive delay from $\overline{DSDEN0,1}$	3		ns	Read
18	$\overline{DSDEN0,1}$ active delay from PRCLK ↓	8	60	ns	Write
19	$\overline{DSDEN0,1}$ inactive delay from PRCLK ↓	8	50	ns	Write

SYM	DESCRIPTION	MIN	MAX	UNITS	NOTES
20	$\overline{AF16}$,$\overline{MEMCS16}$ setup time to ALE↓	35		ns	
21	$\overline{AF16}$,$\overline{MEMCS16}$ hold time from ALE↓	1		ns	
22	PRCLK delay from 16MHZ		50	ns	
23	$\overline{RSET}$ hold time from PRCLK↓	10		ns	
24	$\overline{RSET}$ setup time to PRCLK↓	25		ns	
25	RESET A/I delay from PRCLK↓	5	35	ns	
26	SCLK delay from PRCLK		25	ns	
27	DMACLK delay from SCLK		15	ns	
28	RESCPU inactive delay from PRCLK↓	5	50	ns	
29	RESCPU active delay from SCLK↑	5	20	ns	
30	HRQ setup to DMACLK↑	15		ns	1
31	HRQ holdtime from DMACLK↑	1		ns	1
32	CPUHRQ A/I delay from CPUHLDA↑		30	ns	
33	HLDA active delay from SCLK		25	ns	
34	HLDA A/I delay from CPUHLDA		30	ns	
35	HLDA inactive delay from DMACLK↑		30	ns	
36	$\overline{DMAMR}$ setup to DMACLK↑	15		ns	
37	$\overline{CMD}$ tristate elay from CPUHLDA		30	ns	2
38	$\overline{MEMR}$ active delay from DMACLK↑ (Due to DMAMR)		25	ns	
39	$\overline{MEMR}$ inactive delay from $\overline{DMAMR}$↑		25	ns	
40	$\overline{MEMR}$ tri-state delay from DMACLK↑ (Due to DMAMR)		25	ns	
41	$\overline{CMD}$ active delay from CPUHLDA		30	ns	
42	DMARDY inactive delay from $\overline{DMAMR}$,$\overline{IOR}$		30	ns	
43	DMARDY active delay from DMACLK↓		30	ns	

SYM	DESCRIPTION	MIN	MAX	UNITS	NOTES
44	MEMCYC active delay from $\overline{\text{MEMW}}$,$\overline{\text{MEMR}}$		30	ns	
45	MEMCYC inactive delay from from $\overline{\text{MEMW}}$,$\overline{\text{MEMR}}$		30	ns	
46	A0 setup time to PRCLK ↓	0		ns	
47	A0 hold time from PRCLK↓	5		ns	
48	SA0 delay from PRCLK ↓		50	ns	
49	$\overline{\text{XBHE}}$ setup time to ALE↓	0		ns	
50	$\overline{\text{READY}}$ inactive delay from PRCLK↓ (middle of Ts cycle)		25	ns	
51	$\overline{\text{READY}}$ active delay from PRCLK↓		24	ns	
52	CNTLOFF A/I delay from PRCLK		30	ns	
53	DIRHLB,$\overline{\text{ENHLB}}$ active delay from from $\overline{\text{IOR}}$,$\overline{\text{IOW}}$		50	ns	
54	$\overline{\text{DIRHLB}}$ inactive delay from PRCLK ($\overline{\text{IOW}}$)		50	ns	
55	$\overline{\text{ENHLB}}$ active delay from PRCLK ($\overline{\text{IOW}}$ cycle)		50	ns	
56	$\overline{\text{ENHLB}}$ inactive delay from $\overline{\text{IOR}}$		50	ns	
57	$\overline{\text{ERROR}}$ holdtime from $\overline{\text{BUSY}}$ active	0		ns	
58	$\overline{\text{BUSY}}$ active pulse width	15		ns	
59	$\overline{\text{ERROR}}$ setup to $\overline{\text{BUSY}}$	10		ns	
60	$\overline{\text{BUSY287}}$ A/I delay from $\overline{\text{BUSY}}$ A/I		30	ns	
61	$\overline{\text{ERROR}}$ active pulse width	10		ns	
62	INT287 active delay from BUSY,ERROR		30	ns	
63	INT287 inactive delay from $\overline{\text{ERROR}}$		30	ns	
64	$\overline{\text{BUSY287}}$ inactive delay from $\overline{\text{IOW}}$		40	ns	

SYM	DESCRIPTION	MIN	MAX	UNITS	NOTES
65	SMIO,SA0,SA3,$\overline{\text{287CS}}$,$\overline{\text{INTA}}$ setup time to $\overline{\text{IOW}}$	10		ns	
66	SMIO,SA0,SA3,$\overline{\text{287CS}}$,$\overline{\text{INTA}}$ hold time from $\overline{\text{IOW}}$	1		ns	
67	RES287 active delay from $\overline{\text{IOW}}$		40	ns	
68	RES387 inactive delay from $\overline{\text{IOW}}$		40	ns	
69	$\overline{\text{NPCS}}$ active delay from SMIO,SA3, 287CS, INTA		35	ns	
70	NPCS inactive delay from SMIO,SA3, 287CS,INTA		35	ns	
71	Data Setup to $\overline{\text{IOW}}$ ↑	30		ns	
72	Data Output from $\overline{\text{IOR}}$ ↓		80	ns	
73	Data float from $\overline{\text{IOR}}$ ↑		50	ns	
74	U46 Q outputs from $\overline{\text{IOW}}$ ↑		80	ns	
75	U46 Q outputs clear from $\overline{\text{RESET}}$ ↓		80	ns	
76	A20 GATE delay from $\overline{\text{IOW}}$ ↑		80	ns	
77	A20 GATE delay from $\overline{\text{RESET}}$ ↓		80	ns	
78	GA20 delay from A20 GATE	5	50	ns	
79	GA20 delay from A20	5	50	ns	

1. Setup and Hold times are required to guarantee recognition of signa' at clock edge.
2. CMD = MEMR,MEMW,IOR and IOW.
   A/I = Active and Inactive

Power Supply

1000TL POWER SUPPLIES

1000TL 67 WATT SINGLE INPUT POWER SUPPLY

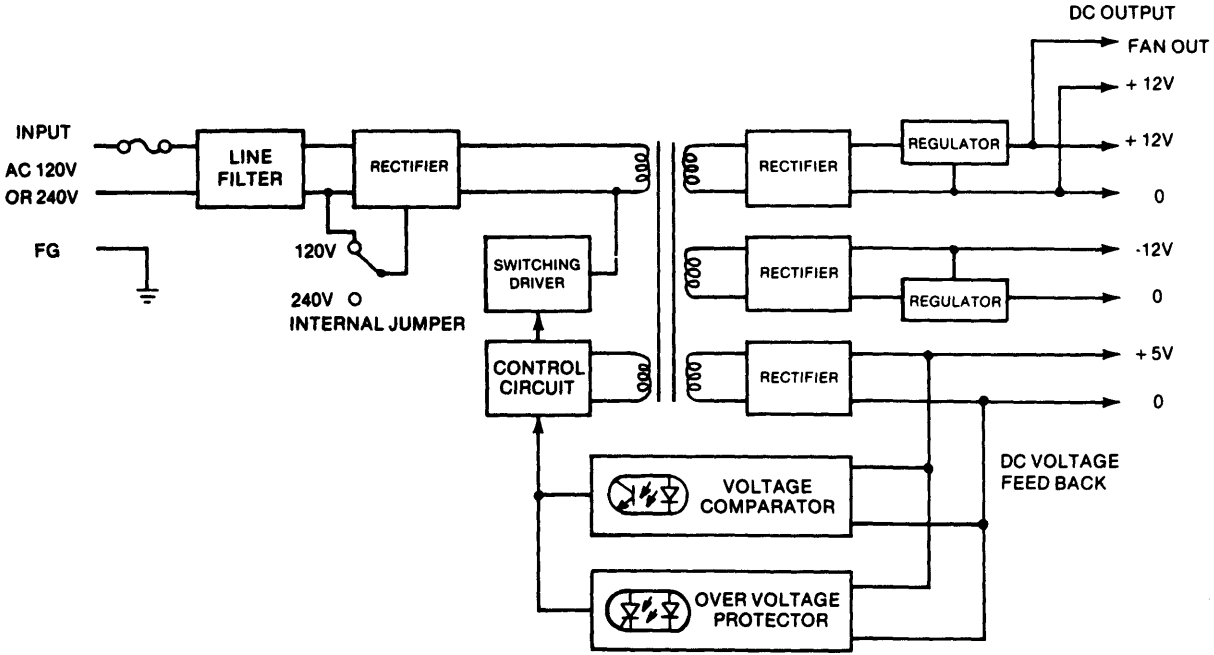
1000TL 67 Watt Single Input Power Supply

Contents

OPERATING CHARACTERISTICS

	MINIMUM	TYPICAL	MAXIMUM	UNITS
Operating Voltage Range	90	120	135	VAC
Line Frequency	47	50/60	63	Hz
Output Voltages				
Vo1	4.85	5.00	5.15	V
Vo2	11.40	12.00	12.60	V
Vo3	-13.20	-12.00	-10.80	V
Output Loads				
Io1	1.25	-	7.0	A
Io2	0.15	-	2.4	A
Io3	0	-	0.25	A
Over Current Protection				
Current Limit ICL1	-	-	14.0	A
ICL2	-	-	4.8	A
ICL3	-	-	1.0	A
Over Voltage Protection				
Crowbar	5.8	-	6.8	V
Output Noise				
Vo1	-	-	50	mV P-P
Vo2	-	-	100	mV P-P
Vo3	-	-	150	mV P-P
Efficiency	63	65	-	%
Holdup Time				
Full Load at Nominal Line	16	-	-	mSec.
Insulation Resistance				
Input to Output	7	1000	-	M ohms
Input to Ground	7	1000	-	M ohms
Isolation				
Input to Ground	1.7	-	-	KVDC

Power Supply Block Diagram



Theory of Operation

AC Input Circuit

This circuit is composed of an AC Power Switch, a fuse, a line filter, and an inrush current limiting circuit and rectifying smoothing circuit. The inrush current limiting circuit controls the charging current to electrolytic capacitors when power is ON. The line filter reduces noise that leaks from the power source to the AC line or that returns from the unit to the power souce; it satisfies the specifications of noise regulations.

Control Circuit & Power Converter Circuit

This circuit is a self oscillation switching system, generally called an R.C.C. (Ringing Choke Converter). The R.C.C. circuit does not fix the oscillating frequency. Whenever input voltage is high or the load becomes light, the oscillating frequency will be high.

The current through R2 supplies transistor Q1's base, then Q1 turns ON. When transistor Q1 is On, the Q1 current excites the transformer T1 and voltage rises in the bias coil of T1(5-6) which leads transistor Q1 positive bias, then transistor Q1 turns ON.

When transistor Q1 turns ON, collector current charges the energy to primary inductance of transformer T1 (1-3). Increasing the collector current of transistor Q1 to the point of:

$$I_C \geqq I_B .hfe$$

Then, transistor Q1 immediately turns OFF. In a moment, transformer T1 will have negative voltage which will be supplied to the secondary circuit through a rectifier. A Short Circuit Protector is provided to protect transistor Q1 from excess amounts of current when the secondary circuit becomes shorted. When transistor Q2 detects the voltage drop at R12, the collector of Q2 shorts the base and emitter of Q1. Then Q1 stops working so that the circuit protects Q1 from over current.

The over current protector in the -12V line is provided by the three terminal positive voltage regulator IC1 (built-in current fold back protection), which protects Q1 against excessive current from the -12V line.

5V Output Voltage Detecting Circuit

The circuit detects the change of output load current compared with the output voltage and AC line input voltage, which feeds back to the control circuit through a photo coupler PHC1 to keep the output voltage stable. The Photo coupler isolates the primary and secondary circuits.

Over-Voltage Protection

When the +5 output voltage rises, between 5.8V to 6.8V, a control signal turns on the photo coupler PHC2 (Photo Thyristor) with the current of zener diode (D11) and stops oscillation by turning on Q3, which turns off Q1 in the switching circuit.

In the case of stopped oscillation, correct the cause of the failure, and reinput the power. The overvoltage protection circuit will reset automatically under good conditions.

The Photo Thyristor isolates the primary and secondary circuits.

1000TL 67 WATT DUAL INPUT POWER SUPPLY

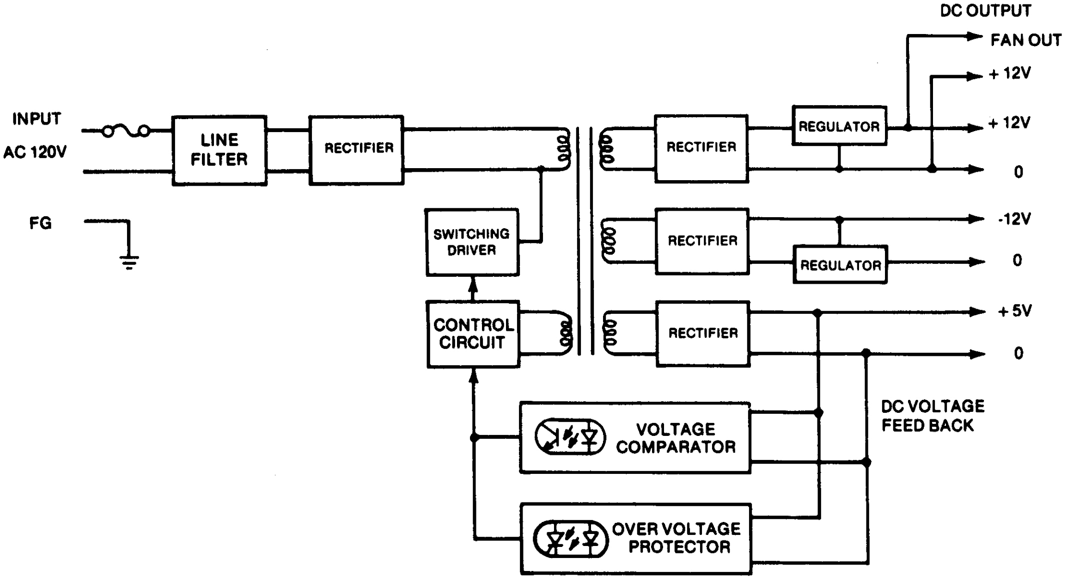
1000TL 67 Watt Dual Input Power Supply

Contents

OPERATING CHARACTERISTICS

		MINIMUM	TYPICAL	MAXIMUM	UNITS
Operating Voltage Range		90 198	120 240	135 264	VAC
Line Frequency		47	50/60	63	Hz
Output Voltages					
Vo1		4.85	5.00	5.15	V
Vo2		11.40	12.00	12.60	V
Vo3		-13.20	-12.00	-10.80	V
Output Loads					
Io1		1.25	-	7.0	A
Io2		0.15	-	2.4	A
Io3		0	-	0.25	A
Over Current Protection					
Current Limit	ICL1	-	-	14.0	A
	ICL2	-	-	4.8	A
	ICL3	-	-	1.0	A
Over Voltage Protection					
Crowbar		5.8	-	6.8	V
Output Noise					
Vo1		-	-	50	mV P-P
Vo2		-	-	100	mV P-P
Vo3		-	-	150	mV P-P
Efficiency		63	65	-	%
Holdup Time					
Full Load at Nominal Line		16	-	-	mSec.
Insulation Resistance					
Input to Output		7	1000	-	M ohms
Input to Ground		7	1000	-	M ohms
Isolation					
Input to Ground		1.25	-	-	KVAC
Input to Output		3.75	-	-	KVAC

Power Supply Block Diagram



Theory of Operation

AC Input Circuit

This circuit is composed of an AC Power Switch, a fuse, a line filter, and an inrush current limiting circuit and rectifying smoothing circuit. The inrush current limiting circuit controls the charging current to electrolytic capacitors when power is ON. The line filter reduces noise that leaks from the power source to the AC line or that returns from the unit to the power souce; it satisfies the specifications of noise regulations.

Control Circuit & Power Converter Circuit

This circuit is a self oscillation switching system, generally called an R.C.C. (Ringing Choke Converter). The R.C.C. circuit does not fix the oscillating frequency. Whenever input voltage is high or the load becomes light, the oscillating frequency will be high.

The current through R4 and R5 supplies transistor Q1's base, then Q1 turns ON. When transistor Q1 is On, the Q1 current excites the transformer T1 and voltage rises in the bias coil of T1(2-3) which leads transistor Q1 positive bias, then transistor Q1 turns ON.

When transistor Q1 turns ON, collector current charges the energy to primary inductance of transformer T1 (4-6). Increasing the collector current of transistor Q1 to the point of:

$$I_C \geq I_B .hfe$$

Then, transistor Q1 immediately turns OFF. In a moment, transformer T1 will have negative voltage which will be supplied to the secondary circuit through a rectifier. A Short Circuit Protector is provided to protect transistor Q1 from excess amounts of current when the secondary circuit becomes shorted. When transistor Q2 detects the voltage drop at R13, the collector of Q2 shorts the base and emitter of Q1. Then Q1 stops working so that the circuit protects Q1 from over current.

The over current protector in the -12V line is provided by the three terminal positive voltage regulators IC2, IC3 (built-in current fold back protection), which protects Q1 against excessive current from the -12V line.

5V Output Voltage Detecting Circuit

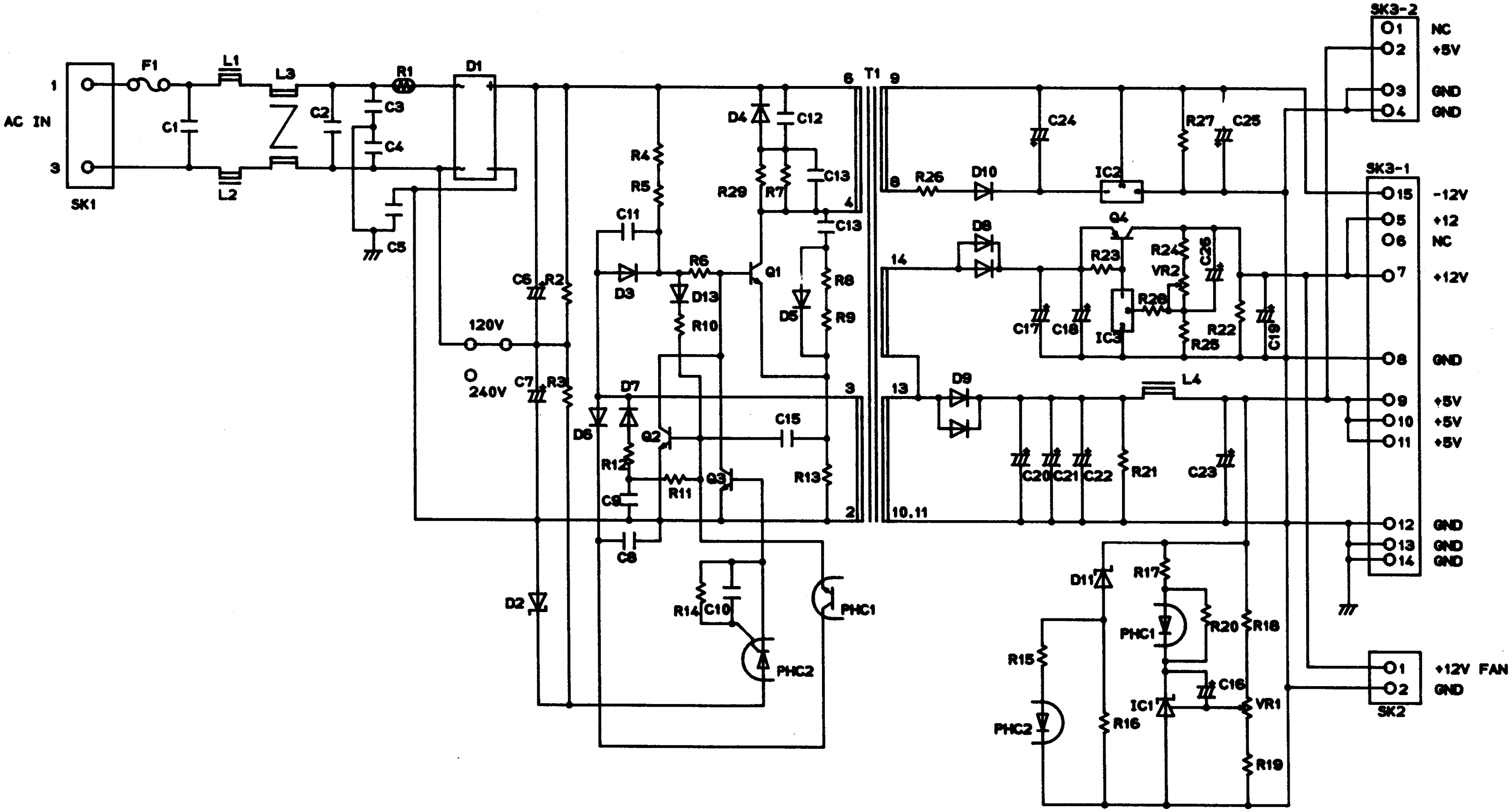
The circuit detects the change of output load current compared with the output voltage and AC line input voltage, which feeds back to the control circuit through a photo coupler PHCl to keep the output voltage stable. The Photo coupler isolates the primary and secondary circuits.

Over-Voltage Protection

When the +5 output voltage rises, between 5.8V to 6.8V, a control signal turns on the photo coupler PHC2 (Photo Thyristor) with the current of zener diode (Dll) and stops oscillation by turning on Q3, which turns off Ql in the switching circuit.

In the case of stopped oscillation, correct the cause of the failure, and reinput the power. The overvoltage protection circuit will reset automatically under good conditions.

The Photo Thyristor isolates the primary and secondary circuits.



model no. 8790084

Keyboard

FUJITSU KEYBOARD
ASSEMBLY # N860-4703-T

1.0 GENERAL

The keyboard is a direct, plug-compatible replacement for the Enhanced Keyboard for the IBM PC, XT, and AT personal computer. No software modification or special interface is needed by the user.

2.0 SCOPE

This specification describes the functional, mechanical, electrical, environmental, and reliability characteristics of the FUJITSU N860-4703-T Keyboard assembly.

The keyboard is encoded in such a way as to produce a unique output code for each key that is pressed and/or released. The communication with the host computer is a synchronous serial link. The Key Layout, Switch Encoding and Serial Communication are all compatible with the IBM PC, XT and AT.

3.0 MECHANICAL SPECIFICATION

3.1 Key Layout, Legends, and Colors

Figure 1 shows keytop layout, appropriate legends and keytop colors. The keys are numbered from left to right starting with the spacebar row (Row A) and ending with the function key row (Row F).

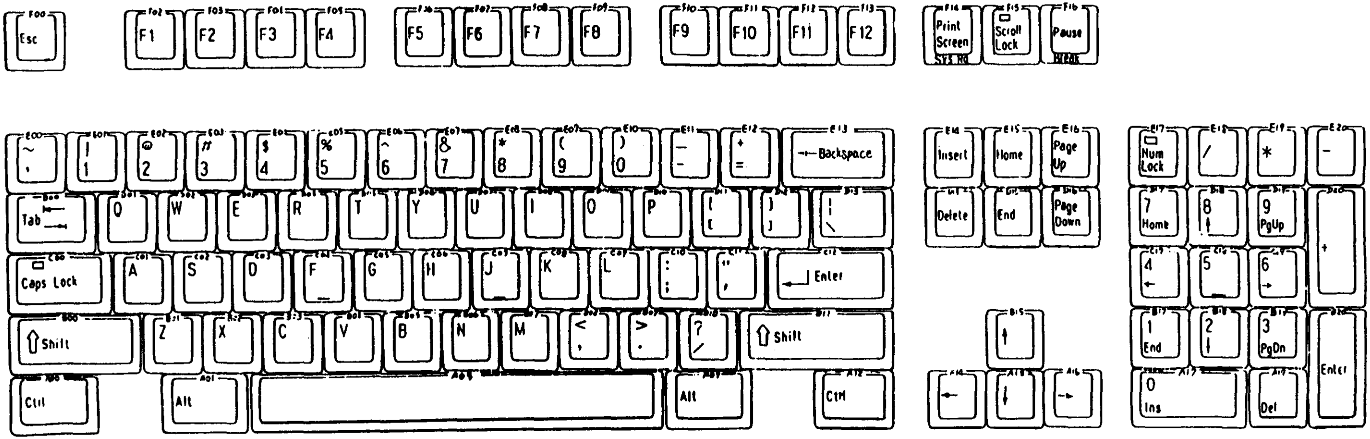

Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Print Screen SYS RQ
Scroll Lock
Pause Break
~ `
! 1
@ 2
# 3
$ 4
% 5
^ 6
& 7
* 8
(9
) 0
_ -
+ =
Backspace
Insert
Home
Page Up
Num Lock
/
*
-
Tab
Q
W
E
R
T
Y
U
I
O
P
{ [
}]
| \
Delete
End
Page Down
7 Home
8
9 PgUp
+
Caps Lock
A
S
D
F
G
H
J
K
L
: ;
" '
Enter
4
5
6
Shift
Z
X
C
V
B
N
M
< ,
> .
? /
Shift
1 End
2
3 PgDn
Enter
Ctrl
Alt
Alt
Ctrl
0 Ins
Del

FIGURE 1

3.2 KEYSWITCH

3.2.1 Total Travel: 0.150", +/- 0.020" (3.8, +/-0.5mm)

3.2.2 FORCE: All keyswitches shall utilize a 2.0 ounce (+/- 0.9 oz) operating force. This is accomplished using both a rubber keyswitch membrane and springs.

3.2.3 BREAKOVER FEEDBACK: The keyswitches utilize a tactile feedback to assure the operator that the key has been fully pressed.

4.0 FUNCTIONAL REQUIREMENTS

4.1 SCAN CODES

The keyboard generates a unique Hex scan code for each keyswitch that is pressed (make code) and released (break code). For the AT Mode, the break code is the same as the make code preceded by "F0" Hex. Example: The make code for the "ESC" key is 76 Hex and the break code is two bytes, F0 76 Hex. For the XT Mode, the break code is 80 Hex, plus the make code. Example: The make code for the "ESC" key is 01 Hex and the break code is 81 Hex. The keyswitch-to-scan-code (make and break codes) assignments, Standard ASCII Codes, and Extended ASCII Codes are listed on the following pages.

KEYBOARD SCAN CODES

Key	Key	AT Mode		XT Mode		Standard ASCII				Extended ASCII			
#	Descript.	Make	Break	Make	Break	(Scancode/ASCII code)				(Scancode/ASCII code)			
		Code	Code	Code	Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
1	Esc	76	F076	01	81	011B	011B	011B	-----	011B	011B	011B	0100
2	F1	05	F005	3B	BB	3B00	5400	5E00	6800	3B00	5400	5E00	6800
3	F2	06	F006	3C	BC	3C00	5500	5F00	6900	3C00	5500	5F00	6900
4	F3	04	F004	3D	BD	3D00	5600	6000	6A00	3D00	5600	6000	6A00
5	F4	0C	F00C	3E	BE	3E00	5700	6100	6B00	3E00	5700	6100	6B00
6	F5	03	F003	3F	BF	3F00	5800	6200	6C00	3F00	5800	6200	6C00
7	F6	0B	F00B	40	C0	4000	5900	6300	6D00	4000	5900	6300	6D00
8	F7	83	F083	41	C1	4100	5A00	6400	6E00	4100	5A00	6400	6E00
9	F8	0A	F00A	42	C2	4200	5B00	6500	6F00	4200	5B00	6500	6F00
10	F9	01	F001	43	C3	4300	5C00	6600	7000	4300	5C00	6600	7000
11	F10	09	F009	44	C4	4400	5D00	6700	7100	4400	5D00	6700	7100
12	F11	78	F078	57	D7	-----	-----	-----	-----	8500	8700	8900	8B00
13	F12	07	F007	58	D8	-----	-----	-----	-----	8600	8800	8A00	8C00
14	Print Scrn	E07C	E0F07C	E02AE037	E0B7E0AA	Note[1]	Note[1]	7200	-----	Note[1]	Note[1]	7200	-----
15	Scroll Lock	7E	F07E	46	C6	Note[2]	Note[2]	------	Note[2]	Note[2]	Note[2]	-----	Note[2]
16	Pause Break	E11477	E1F014F077	E11D45	E19DC5	Note[3]	Note[3]	Note[4]	Note[3]	Note[3]	Note[3]	Note[4]	Note[3]
17	~ or \	0E	F00E	2B	AB	2960	297E	-----	-----	6000	7E00	-----	2900
18	! or 1	16	F016	02	82	0231	0221	-----	7800	0231	0221	-----	7800

Table 1

KEYBOARD SCAN CODES

Key	Key	AT Mode		XT Mode		Standard ASCII				Extended ASCII			
#	Descript.	Make	Break	Make	Break	(Scancode/ASCII code)				(Scancode/ASCII code)			
		Code	Code	Code	Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
19	@ or 2	1E	F01E	03	83	0332	0340	0300	7900	0332	0340	0300	7900
20	# or 3	26	F026	04	84	0433	0423	-----	7A00	0433	0423	-----	7A00
21	$ or 4	25	F025	05	85	0534	0524	-----	7B00	0534	0524	-----	7B00
22	% or 5	2E	F02E	06	86	0635	0625	-----	7C00	0635	0625	-----	7C00
23	^ or 6	36	F036	07	87	0736	075E	071E	7D00	0736	075E	071E	7D00
24	& or 7	3D	F03D	08	88	0837	0826	-----	7E00	0837	0826	-----	7E00
25	* or 8	3E	F03E	09	89	0938	092A	-----	7F00	0938	092A	-----	7F00
26	(or 9	46	F046	0A	8A	0A39	0A28	-----	8000	0A39	0A28	-----	8000
27	) or 0	45	F045	0B	8B	0B34	0B29	-----	8100	0B34	0B29	-----	8100
28	_ or -	4E	F04E	0C	8C	0C2D	0C5F	0C1F	8200	0C2D	0C5F	0C1F	8200
29	+ or =	55	F055	0D	8D	0D3D	0D2B	-----	8300	0D3D	0D2B	-----	8300
30	Backspace	66	F066	0E	8E	0E08	0E08	0E7F	-----	0E08	0E08	0E7F	0E00
31	Insert	E070	E0F070	E02AE052	E0D2E0AA	5200	5200	-----	-----	52E0	52E0	92E0	A200
32	Home	E06C	E0F06C	E02AE047	E0C7E0AA	4700	4700	7700	-----	47E0	47E0	77E0	9700
33	Pg Up	E07D	E0F07D	E02AE049	E0C9E0AA	4900	4900	8400	-----	49E0	49E0	84E0	9900
34	Num Lock	77	F077	45	C5	Note[5]	Note[5]	-----	Note[5]	Note[5]	Note[5]	----	Note[5]
35	/	E04A	E0F04A	E035	E0B5	352F	352F	-----	-----	E02F	E02F	9500	A400
36	•	7C	F07C	37	B7	372A	372A	-----	-----	372A	372A	9600	3700
37	-	7B	F07B	4A	CA	4A2D	4A2D	-----	-----	4A2D	4A2D	8E00	4A00

KEYBOARD SCAN CODES

Key	Key	AT Mode		XTMode		Standard ASCII				Extended ASCII			
#	Descript.	Make	Break	Make	Break	(Scancode/ASCII code)				(Scancode/ASCII code)			
		Code	Code	Code	Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
38	Tab	0D	F00D	0F	8F	0F90	0F00	-----	-----	0F09	0F00	9400	A500
39	Q or q	15	F015	10	90	1071	1051	1011	1000	1071	1051	1011	1000
40	W or w	1D	F01D	11	91	1177	1157	1117	1100	1177	1157	1117	1100
41	E or e	24	F024	12	92	1265	1245	1205	1200	1265	1245	1205	1200
42	R or r	2D	F02D	13	93	1372	1352	1312	1300	1372	1352	1312	1300
43	T or t	2C	F02C	14	94	1474	1454	1414	1400	1474	1454	1414	1400
44	Y or y	35	F035	15	95	1579	1559	1519	1500	1579	1559	1519	1500
45	U or u	3C	F03C	16	96	1675	1655	1615	1600	1675	1655	1615	1600
46	I or i	43	F043	17	97	1769	1749	1709	1700	1769	1749	1709	1700
47	O or o	44	F044	18	98	186F	184F	180F	1800	186F	184F	180F	1800
48	P or p	4D	F04D	19	99	1970	1950	1910	1900	1970	1950	1910	1900
49	{ or [	54	D054	1A	9A	1A5B	1A7B	1A1B	-----	1A5B	1A7B	1A1B	1A00
50	} or]	5B	F05B	1B	9B	1B5D	1B7D	1B1D	-----	1B5D	1B7D	1B1D	1B00
51	| or \	5D	F05D	2B	AB	2B5C	2B7C	2B1C	-----	2B5C	2B7C	2B1C	2B00
52	Delete	E071	E0F071	E02AE053	E0D3E0AA	5300	5300	-----	-----	53E0	53E0	93E0	A300
53	End	E069	E0F069	E02AE04F	E0CFE0AA	4F00	4F00	7500	-----	4FE0	4FE0	75E0	9F00
54	Page Down	E07A	E0F07A	E02AE051	E0D1E0AA	5100	5100	7600	-----	51E0	51E0	76E0	A100
55	7 or Home	6C	F06C	47	C7	4700	4737	7700	Note[6]	4700	4737	7700	Note[6]
56	8	75	F075	48	C8	4800	4838	----	Note[6]	4800	4838	8D00	Note[6]
57	9 or Page Up	7D	F07D	49	C9	4900	4939	8400	Note[6]	4900	4939	8400	Note[6]

KEYBOARD SCAN CODES

Key	Key	AT Mode		XT Mode		Standard ASCII				Extended ASCII			
#	Descript.	Make	Break	Make	Break	(Scancode/ASCII code)				(Scancode/ASCII code)			
		Code	Code	Code	Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
58	+	79	F079	4E	CE	4E2B	4E2B	----	----	4E2B	4E2B	9000	4E00
59	Caps Lock	58	F058	3A	BA	Note[7]	Note[7]	----	Note[7]	Note[7]	Note[7]	----	Note[7]
60	A or a	1C	F01C	1E	9E	1E61	1E41	1E01	1E00	1E61	1E41	1E01	1E00
61	S or s	1B	F01B	1F	9F	1F73	1F53	1F13	1F00	1F73	1F53	1F13	1F00
62	D or d	23	F023	20	A0	2064	2044	2004	2000	2064	2044	2004	2000
63	F or f	2B	F02B	21	A1	2166	2146	2106	2100	2166	2146	2106	2100
64	G or g	34	F034	22	A2	2267	2247	2207	2200	2267	2247	2207	2200
65	H or h	33	F033	23	A3	2368	2348	2308	2300	2368	2348	2308	2300
66	J or j	3B	F03B	24	A4	246A	244A	240A	2400	246A	244A	240A	2400
67	K or k	42	F042	25	A5	256B	254B	250B	2500	256B	254B	250B	2500
68	L or l	4B	F04B	26	A6	266C	264C	260C	2600	266C	264C	260C	2600
69	: or ;	4C	F04C	27	A7	273B	273A	----	----	273B	273A	----	2700
70	" or '	52	F052	28	A8	2827	2822	----	----	2827	2822	----	2800
71	Enter	5A	F05A	1C	9C	1C0D	1C0D	1C0A	----	1C0D	1C0D	1C0A	1C00
72	4	6B	F06B	4B	CB	4B00	4B34	7300	Note[6]	4B00	4B34	7300	Note[6]
73	5	73	F073	4C	CC	-----	4C35	---	Note[6]	4C00	4C35	8F00	Note[6]
74	6	74	F074	4D	CD	4D00	4D36	7400	Note[6]	4D00	4D36	7400	Note[6]

KEYBOARD SCAN CODES

Key	Key	AT Mode		XT Mode		Standard ASCII (Scancode/ASCII code)				Extended ASCII (Scancode/ASCII code)			
#	Descript.	Make Code	Break Code	Make Code	Break Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
76	Left Shift	12	F012	2A	AA	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]
77	Z or z	1A	F01A	2C	AC	2C7A	2C5A	2C1A	2C00	2C7A	2C5A	2C1A	2C00
78	X or x	22	F022	2D	AD	2D78	2D58	2D18	2000	2D78	2D58	2D18	2000
79	C or c	21	F021	2E	AE	2E63	2E43	2E03	2E00	2E63	2E43	2E03	2E00
80	V or v	2A	F02A	2F	AF	2F76	2F56	2F16	2F00	2F76	2F56	2F16	2F00
81	B or b	32	F032	30	B0	3062	3042	3002	3000	3062	3042	3002	3000
82	N or n	31	F031	31	B1	316E	314E	310E	3100	316E	314E	310E	3100
83	M or m	3A	F03A	32	B2	326D	324D	320D	3200	326D	324D	320D	3200
84	< or ,	41	F041	33	B3	332C	333C	-----	-----	332C	333C	-----	3300
85	> or .	49	F049	34	B4	342E	343E	-----	-----	342E	343E	-----	3400
86	? or /	4A	F04A	35	B5	352F	353F	-----	-----	352F	353F	-----	3500
87	Right Shift	59	F059	36	B6	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]
88	Up Arrow	E075	E0F075	E02AE048	E0C8E0AA	4800	4800	-----	----	48E0	48E0	8DE0	9800
89	1 or End	69	F069	4F	CF	4F00	4F31	7500	Note[6]	4F00	4F31	7500	Note[6]
90	2	72	F072	50	D0	5000	5032	----	Note[6]	5000	5032	9100	Note[6]
91	3 or Pg Dn	7A	F07A	51	D1	5100	5133	7600	Note[6]	5100	5133	7600	Note[6]
92	Left Ctrl	14	F014	1D	9D	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]
93	Left Alt	11	F011	38	B8	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]
94	Space	29	F029	39	B9	3920	3920	3920	3920	3920	3920	3920	3920

KEYBOARD SCAN CODES

Key	Key	AT Mode		XT Mode		Standard ASCII (Scancode/ASCII code)				Extended ASCII (Scancode/ASCII code)			
#	Descript.	Make Code	Break Code	Make Code	Break Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
95	Right Alt	E011	E0F011	E038	E0B8	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]
96	Right Ctrl	E014	E0F014	E0ID	E09D	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]
97	Left Arrow	E06B	E0F06B	E02AE04B	E0CBE0AA	4B00	4B00	7300	-----	4BE0	4BE0	73E0	9B00
98	Down Arrow	E072	E0F072	E02AE050	E0D0E0AA	5000	5000	-----	-----	50E0	50E0	91E0	A000
99	Right Arrow	E074	E0F074	E024E04D	E0CDE0AA	4D00	4000	7400	----	4DE0	4DE0	74E0	9D00
100	0 or Ins	70	F070	52	D2	5200	5230	----	Note[6]	5200	523H	9200	Note[6]
101	. or Del	71	F071	53	D3	5300	532E	----	----	5300	532E	9300	-----
102	Enter	E05A	E0F05A	E01C	E09C	1C0D	1C0D	1C0A	----	E00D	E00D	E00A	A600

NOTES

Note1 —INT O5H is invoked and a screen dump is performed
Note2 —the scroll lock active bit is toggled
Note3 —the pause state is initiated
Note4 —INT 1BH is invoked
Note5 —the numlock active bit is toggled
Note6 —ALT num pad generates raw ascii code of typed number
Note7 —the caps lock active bit is toggled
Note8 —hold shift lock active until key is released
Note9 —hold control shift active until key is released
Note10—hold alternate shift active until key is released

Probably have this. :(

5.0 PROTOCOL

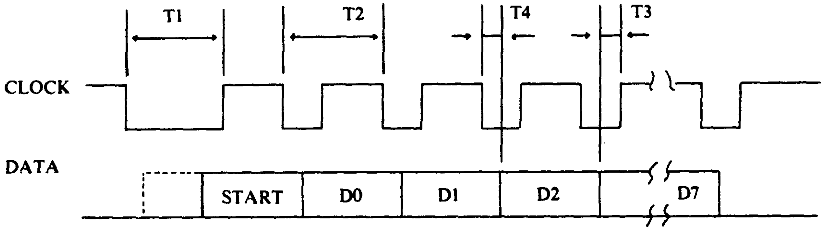
5.0.1 COMMUNICATION MODE 1 (PC/XT Mode)

The keyboard communicates with the host computer using a synchronous serial protocol at approximately 9600BPS. One start bit, eight data bits, no stop bit, and no parity are used to make up the nine bit data word. When no communications are in progress, the keyboard holds the data line low and the clock line high.

Before starting a transmission, the keyboard lowers the clock line as a Request To Send (RTS). The state of the data line is then checked. If the host system is holding the data line low, transmissions are disabled. The keyboard will retain the keycode for the pressed key in its buffer until the clock and data lines return to the idle state. The keyboard then resumes scanning the keyboard matrix until transmissions are enabled.

When transmissions are enabled, 100 to 250 microseconds after the keyboard drives the clock line low, the keyboard transmits its data in the previously described format. Data is valid during the time the clock line is high and for a minimum of 10 microseconds after the falling edge of the clock line. See Figure 2 for a timing diagram.

< KEYBOARD DATA OUTPUT-XT MODE >



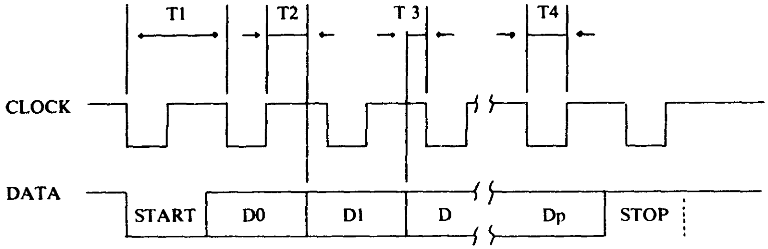
T1 ; 100~250µs (CHECKING TIME OF DATA OUTPUT READY OR PROHIBITION)
T2 ; 104µs±20%
T3 ; 10µs MIN.
T4 ; 10µs MIN.

FIGURE 2

5.0.2 COMMUNICATION MODE 2 (AT MODE)

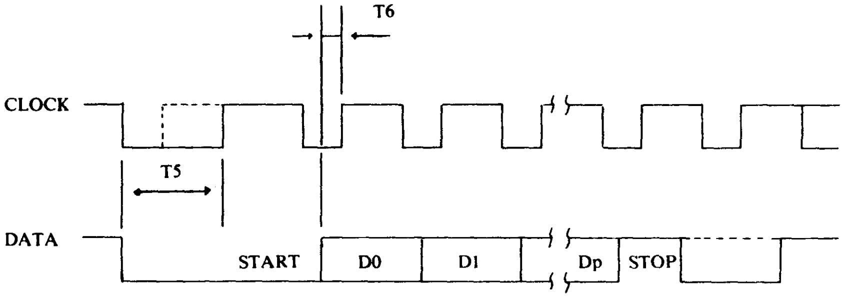
The keyboard communicates with the host computer using a synchronous serial protocol at approximately 9600BPS. One start bit, eight data bits, odd parity, and one stop bit form the eleven bit data word. This communication is bi-directional, with the keyboard clocking all data transfers. When no communications are in progress, the data and clock lines are high, indicating an idle state.

< KEYBOARD DATA OUTPUT-AT MODE > Figure 3A



T1 ; 104µs ±20%
T2 ; 20µs MIN.
T3 ; 20µs MIN.
T4 ; 35µs ±20%

<KEYBOARD DATA INPUT-AT MODE > Figure 3B



T5 ; 60µs MIN. (WAITING TIME FOR HOST DATA OUTPUT)
T6 ; 5µs MIN.

FIGURE 3

Before starting a transmission, the keyboard checks the status of the clock and data lines. Transmissions are disabled if the clock line is low and the code for the pressed key is held in the keyboard buffer until transmissions are enabled. If the clock line is high and the data line is low, the host system is sending a Request To Send (RTS), and the code for the pressed key is stored in the buffer.

When transmissions are enabled and no RTS is detected from the host system, both clock and data line will be high. The keyboard starts a transmission by sending a low start bit, followed by the rest of the data word. Data is valid 20 microseconds minimum prior to the falling edge of the clock. See Figure 3 for timing diagram.

During the transmission of a data word, the keyboard periodically checks the state of the clock line. If the clock line is low during these checks prior to the rising edge of the parity bit, a data collision occurs. When a data collision occurs, the keyboard stops transmitting, returns the code for the pressed key to the keyboard buffer, and prepares to respond to actions requested by the host system.

5.1 COMMANDS FROM THE KEYBOARD TO THE HOST SYSTEM

Keyboard Buffer Overrun — AT and XT Modes

This buffer can store up to sixteen codes for pressed keys. When this buffer is full and the seventeenth code is received, this code is replaced by an Overrun Code. The following chart shows the Overrun Codes for each mode.

XT Mode	FF Hex
AT Mode	00 Hex

Codes received after this code are lost until the keyboard clears additional space in the keyboard buffer.

Self Test Passed — AA Hex AT and XT Modes

The keyboard issues this command upon successful completion of the keyboard self test. The self test consists of the following.

XT MODE

1. Memory is cleared.
2. Keyboard buffer is cleared.
3. ROM checksum is read and compared.
4. RAM is tested.
5. Self test completion code is output.
6. AA Hex is output to indicate successful self test.
7. FC Hex is output to indicate a defect in the self test.

AT MODE

1. ROM checksum is read and compared.
2. RAM is tested.
3. AA Hex is output to indicate successful self test.
4. FC Hex is output to indicate a defect in the self test.

This self test is initiated by the host system reset (Ctrl, Alt, Delete) or by Power-on Reset. Upon successful completion of the Self Test during Power-on Reset, the keyboard is set to XT mode if the keyboard detects a low level on the data line for more than 10 microseconds after 5 microseconds from the falling edge of the clock line. If this condition is not met, the keyboard is placed in the AT Mode, and Typematic Rate and Delay are set to the following defaults:

Typematic Rate 10.9 cps
Delay 500 milliseconds

ECHO — EE Hex AT and XT Modes

The Echo Command (EE Hex) is sent in response to an Echo Command from the host system instead of the normal Acknowledge (ACK) for diagnostic purposes.

Acknowledge — FA Hex AT Mode

The keyboard sends an Acknowledge (FA Hex) in response to a valid command from the host system, with the exceptions of the Resend and Echo commands.

Resend -- FE Hex AT and XT Modes

The keyboard issues a Resend (FE Hex) in response to inputs which have parity errors, framing errors, or invalid data received from the host system.

KEYBOARD BUFFER OVERRUN -- AT and XT Modes

When the 16-character keyboard buffer receives the 17th character, an overflow condition occurs. This condition is communicated to the host system by transmitting the Keyboard buffer Overrun (FF Hex for XT Mode, 00 Hex for AT Mode) to the host system.

5.2 COMMANDS FROM THE HOST SYSTEM TO THE KEYBOARD

Prior to sending commands to the keyboard, the host system must first check to see if the keyboard is sending data. If the keyboard is transmitting, and the data is past the parity bit, the host system must accept the data prior to initiating its own transmission.

If the keyboard's data has not yet reached the tenth clock pulse (Parity Bit), or is not transmitting data, the host system assumes control by lowering the clock line for a minimum of 60 microseconds, then releasing the clock line after clamping the data line low to indicate a start bit. The keyboard will respond with an RTS within 5 microseconds by clocking the start bit into the keyboard. The keyboard continues to clock data as shown in the timing diagram (Figure 3B). The host system must ensure that the data is valid prior to the rising edge and after the falling edge of the keyboard clock pulse.

After the parity bit, the host system should raise the data line to indicate a stop bit. The keyboard checks for a logical high stop bit, then clamps the data line low prior to clock in the stop bit. This signals the host system that the keyboard received the data correctly (Acknowledge). If the host system has not raised the data line to indicate a stop bit, a framing error results and the keyboard continues to clock data until the data line is raised by the host system. Upon receiving either a framing or parity error, the keyboard issues a RESEND to the host system.

All commands from the host system require a response from the keyboard. The keyboard will respond to these commands within 20 microseconds.

The following commands may be sent to the keyboard at any time, following the protocol described for the AT Mode. These commands are valid only in the AT Mode. During the reset command, the keyboard will not respond within 20 microseconds as described above.

RESET — FF Hex

Upon receiving this command, the keyboard transmits an Acknowledge to the host system. The keyboard then waits for the host system to accept the Acknowledge response. The host system will accept the Acknowledge by raising the clock and data lines for a minimum of 500 microseconds.

The keyboard then executes the self test routine similar to the Power-On Reset, and is placed in its default state.

RESEND — FE Hex

Upon receiving this command, the keyboard will transmit the last byte of data sent to the host system.

SET DEFAULT — F6 Hex

This command resets the keyboard to the Power-Up default state. The keyboard responds with an Acknowledge, clears the output buffer,· sets the scanset to AT Mode, sets the default typematic rate and delay, and continues to scan the matrix.

DEFAULT DISABLE — F5 Hex

This command is similar to the Set Default command, except the keyboard stops scanning the matrix and waits for further instructions to be sent by the host system.

ENABLE — F4 Hex

Upon receipt of this command the keyboard responds with an Acknowledge, clears the output buffer, and starts scanning the matrix.

SET TYPEMATIC RATE/DELAY -- F3 Hex

This command consists of one command byte and one parameter byte. The keyboard Acknowledges the command byte, stops scanning the matrix, and waits for the parameter byte. Upon receiving the parameter byte, the keyboard sends an Acknowledge, sets the typematic rate and delay as indicated in Table 2, and continues scanning the matrix.

If another command is received instead of the parameter byte, the set typematic rate/delay function ends with no change to the existing rate or delay parameters. The new command is processed and the keyboard continues scanning the matrix.

The parameter byte consists of an eight-bit word with bit 7 (most significant bit) always being set. Bits 0-4 set the typematic rate and bits 5-6 set the delay.

Example

1	0	1	0	1	0	1	1	Parameter Byte
7	6	5	4	3	2	1	0	Bit Number

The above example shows the default 10.9 cps typematic rate and 500 microsecond delay. Bits 0-4 (010111) correspond to the rate of 10.9 cps shown on Table 2 ii and bits 5-6 (01) correspond to the 500 microsecond delay shown in Table 2 i.

See Table 2 for Typematic Rate/Delay values other than the default settings.

i) Delay

Bit 6 5	Delay ms
0 0	250
0 1	500
1 0	750
1 0	1000

ii) Rate

		Bit			Rate
4	3	2	1	0	(CPS)
0	0	0	0	0	30.0
0	0	0	0	1	26.7
0	0	0	1	0	24.0
0	0	0	1	1	21.8
0	0	1	0	0	20.0
0	0	1	0	1	18.5
0	0	1	1	0	17.1
0	0	1	1	1	16.0
0	1	0	0	0	15.0
0	1	0	0	1	13.3
0	1	0	1	0	12.0
0	1	0	1	1	10.9
0	1	1	0	0	10.0
0	1	1	0	1	9.2
0	1	1	1	0	8.6
0	1	1	1	1	8.0

		Bit			Rate
4	3	2	1	0	(CPS)
1	0	0	0	0	7.5
1	0	0	0	1	6.7
1	0	0	1	0	6.0
1	0	0	1	1	5.5
1	0	1	0	0	5.0
1	0	1	0	1	4.6
1	0	1	1	0	4.3
1	0	1	1	1	4.0
1	1	0	0	0	3.7
1	1	0	0	1	3.3
1	1	0	1	0	3.0
1	1	0	1	1	2.7
1	1	1	0	0	2.5
1	1	1	0	1	2.3
1	1	1	1	0	2.1
1	1	1	1	1	2.0

TABLE 2

ECHO — FE Hex

This command is provided for diagnostic purposes. The keyboard shall respond with EE Hex, instead of Acknowledge, and continue scanning the matrix.

SET/RESET STATUS INDICATORS — ED Hex

This command consists of one command byte and one parameter byte. The keyboard Acknowledges the command byte, stops scanning the matrix, and waits for the parameter byte. Upon receiving the parameter byte, the keyboard sends an Acknowledge, sets the status indicators, and starts scanning the matrix.

If another command is received instead of the parameter byte, the keyboard disregards the Set/Reset Status Indicators command without changing the present status of the indicators, processes the new command, and starts scanning the matrix.

The parameter byte is an eight-bit word with bits 3-7 always set to low. Bit 0 is the Scroll Lock Indicator, bit 1 is the Num Lock Indicator, and bit 2 is the Caps Lock Indicator. A high in each individual bit indicates that Indicator is active and the indicator lamp should be on.

Example

0	0	0	0	0	0	1	0	PARAMETER BYTE
7	6	5	4	3	2	1	0	BIT NUMBER

The above example shows the power-on default of the Tandy 3000 NL. Bit 1 is high indicating Num Lock is active and the Num Lock indicator lamp is on.

READ KEYBOARD ID — F2 Hex

This command causes the keyboard to return two identification bytes AB83 Hex. The keyboard responds with an Acknowledge to the command and stops scanning the matrix. The keyboard then transmits the keyboard ID AB83 Hex and resumes scanning the matrix.

SET/READ SCAN SET — F0 Hex

This command is used to select one of three Scan Sets or to tell the host system which Scan Set is currently being used. This command consists of a command byte and a parameter byte. Upon receiving this command, the keyboard sends an Acknowledge to the host system and waits for the parameter byte. When the keyboard receives the parameter byte, it responds with an Acknowledge.

A parameter byte of 00 Hex will cause the keyboard to transmit the Hex value for the Scan Set currently in use. A parameter byte of 01 Hex selects the Scan Set 1 (XT Scan Codes), 02 Hex selects Scan Set 2 (Default AT Scan Codes), and 03 Hex selects Scan Set 3 (Special AT Scan Codes — See Table 3 for Scan Codes and Default Key State Information). The keyboard resumes scanning the matrix.

TABLE 3

KEY DESCRIPTION	MAKE CODE	BREAK CODE	DEFAULT KEY STATE
Esc	08	F0 08	Make Only
F1	07	F0 07	Make Only
F2	0F	F0 0F	Make Only
F3	17	F0 17	Make Only
F4	1F	F0 1F	Make Only
F5	27	F0 27	Make Only
F6	2F	F0 2F	Make Only
F7	37	F0 37	Make Only
F8	3F	F0 3F	Make Only
F9	47	F0 47	Make Only
F10	4F	F0 4F	Make Only
F11	56	F0 56	Make Only
F12	5E	F0 5E	Make Only
Print Scrn	57	F0 57	Make Only
Scroll Lock	5F	F0 5F	Make Only
Pause or Break	62	F0 62	Make Only
`	0E	F0 0E	Typematic
1	16	F0 16	Typematic
2	1E	F0 1E	Typematic
3	26	F0 26	Typematic
4	25	F0 25	Typematic
5	2E	F0 2E	Typematic
6	36	F0 36	Typematic
7	3D	F0 3D	Typematic
8	3E	F0 3E	Typematic
9	46	F0 46	Typematic
0	45	F0 45	Typematic
-	4E	F0 4E	Typematic
=	55	F0 55	Typematic
Backspace	66	F0 66	Make Only
Insert cursor pad	67	F0 67	Make Only
Home cursor pad	6E	F0 6E	Make Only
Page Up cursor pad	6F	F0 6F	Make Only
Num Lock	76	F0 76	Make Only
/ number pad	77	F0 77	Make Only
* number pad	7E	F0 7E	Make Only
- number pad	84	F0 84	Make Only
Tab	0D	F0 0D	Typematic
q	15	F0 15	Typematic
w	1D	F0 1D	Typematic
e	24	F0 24	Typematic
r	2D	F0 2D	Typematic
t	2C	F0 2C	Typematic
y	35	F0 35	Typematic
u	3C	F0 3C	Typematic
i	43	F0 43	Typematic
o	44	F0 44	Typematic
p	4D	F0 4D	Typematic
[	54	F0 54	Typematic
]	5B	F0 5B	Typematic

\	5C	F0 5C	Typematic
Delete cursor pad	64	F0 64	Typematic
Home cursor pad	65	F0 65	Make Only
Page Up cursor pad	6D	F0 6D	Make Only
7 number pad	6C	F0 6C	Make Only
8 number pad	75	F0 75	Make Only
9 number pad	7D	F0 7D	Make Only
+ number pad	7C	F0 7C	Make Only
Caps Lock	14	F0 14	Make/Break
a	1C	F0 1C	Typematic
s	1B	F0 1B	Typematic
d	23	F0 23	Typematic
f	2B	F0 2B	Typematic
g	34	F0 34	Typematic
h	33	F0 33	Typematic
j	3B	F0 3B	Typematic
k	42	F0 42	Typematic
l	4B	F0 4B	Typematic
;	4C	F0 4C	Typematic
'	52	F0 52	Typematic
Enter	5A	F0 5A	Typematic
4 number pad	6B	F0 6B	Make Only
5 number pad	73	F0 73	Make Only
6 number pad	74	F0 74	Make Only
Left Shift	12	F0 12	Make/Break
z	1A	F0 1A	Typematic
x	22	F0 22	Typematic
c	21	F0 21	Typematic
v	2A	F0 2A	Typematic
b	32	F0 32	Typematic
n	31	F0 31	Typematic
m	3A	F0 3A	Typematic
,	41	F0 41	Typematic
.	49	F0 49	Typematic
Right Shift	59	F0 59	Make/Break
Up Arrow cursor pad	63	F0 63	Typematic
1 number pad	69	F0 69	Make Only
2 number pad	72	F0 72	Make Only
3 number pad	7A	F0 7A	Make Only
Enter number pad	79	F0 79	Make Only
Left Ctrl	11	F0 11	Make/Break
Left Alt	19	F0 19	Make/Break
Spacebar	29	F0 29	Typematic
Right Alt	39	F0 39	Make Only
Right Ctrl	58	F0 58	Make Only
Left Arrow cursor	61	F0 61	Typematic
Down Arrow	60	F0 60	Typematic
Right Arrow	6A	F0 6A	Typematic
Ins number pad	70	F0 70	Make Only
Del number pad	71	F0 71	Make Only

SET ALL KEYS — TYPEMATIC — F7 Hex

SET ALL KEYS — MAKE/BREAK — F8 Hex

SET ALL KEYS — MAKE ONLY — F9 Hex

SET ALL KEYS — TYPEMATIC/MAKE/BREAK — FA Hex

These commands affect Scan Set 3 only, but may be sent using any Scan Set. The keyboard responds with an Acknowledge, clears the output buffer, sets all keys to the function requested by the command, and continues scanning the matrix if it was previously enabled.

SET SINGLE KEY -- TYPEMATIC — FB Hex

SET SINGLE KEY -- MAKE/BREAK — FC Hex

SET SINGLE KEY -- MAKE ONLY -- FD Hex

These commands consist of a command byte and a parameter byte. The keyboard responds to the command byte with an Acknowledge and waits for the parameter byte. The parameter byte is the scan code from Scan Set 3 for the key to be changed. Upon receiving the parameter byte, the keyboard sets the selected key to the function selected by the command byte, and continues to scan the matrix if it was previously enabled. These commands affect only Scan Set 3 operation, but may be sent using any Scan Set.

5.3 KEY ROLLOVER

The keyboard incorporates N-Key Rollover in software to avoid loss of keystroke data during high speed entry. N-Key Rollover is defined as all keys pressed and released will be output in the proper sequence. However, when the keyboard detects more than four keys pressed during a scan of the matrix, the keyboard does not output the keycodes until one or more of the keys are released. If the released key was not properly detected as a pressed key, an error condition occurs and the keyboard issues an buffer overrun code to the host system.

5.4 AUTOREPEAT

The power-on default condition will cause the last key pressed to repeat at 10.9 characters-per-second after a 500 millisecond delay. This may be changed by the system when the keyboard is using the AT Communications Mode.

5.5 BUFFERING

The keyboard is capable of storing 16 scan codes in a first in/ first out (FIFO) circular buffer. When the buffer overflows, the last code is replaced by a Hex 00, in AT Mode and a Hex FF, in XT Mode.

5.6 STATUS INDICATORS

Three LED Status Indicators are provided: Num Lock, Caps Lock, and Scroll Lock.

These indicators are located in the keytop of each respective key. The keyboard will power up with all indicators OFF, except when the host system (such as the Tandy 3000 NL) sets them to ON.

6.0 ELECTRICAL REQUIREMENTS

The interface consists of two bi-directional lines, clock and data, which are controlled by 74LS125 equivalent buffers. The keyboard side is terminated by 2200 Ohm resistors. All voltage levels are TTL compatible and the keyboard drivers are capable of sinking 20 mA minimum including the current sourced by the pullup resistors on the keyboard.

6.1 CONNECTOR

The connector is a 5-pin DIN connector. Connections are shown in the following table.

Table 4

PIN #	SIGNAL
1	CLOCK
2	DATA
3	NO CONNECTION
4	LOGIC GROUND
5	+5 VOLTS DC

6.2 CHASSIS GROUND

Chassis ground is isolated from logic ground.

6.3 POWER REQUIREMENTS

The keyboard requires 5 Volts DC, +/-5%, at 500 milliamps (max).

7.0 ENVIRONMENTAL REQUIREMENTS

7.1 TEMPERATURE

OPERATING................0 to 50 degrees C
NON-OPERATING............-20 to 60 degrees C

7.2 RELATIVE HUMIDITY

20% to 90% non-condensing

7.3 SHOCK

Operating and non-operating...........10G 11 mS duration

7.4 VIBRATION

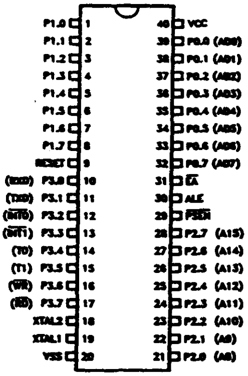
Operating and non-operating...........0.3 mm amplitude 10 to 55 Hz

8.0 RELIABILITY

8.1 SWITCH LIFE

Switch life of the keyboard is a minimum of 20 million cycles.

Pin Connections



DOCUMENT CONTROL SECTION

DATE

CUSTOM IC PIN SIGNAL AND FUNCTION

FUJITSU LIMITED

1 / 3

PIN DESCRIPTION

V_{SS}

Circuit ground potential.

V_{CC}

+5V power supply during operaticn, programming and verification.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port.

It is also the multiplexed low-order address and data bus when using external memory. It is used for data input and output during programming and verification. Port 0 can sink/source two TTL loads.

Port 1

Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during programming and verification. Port 1 can sink/source one TTL load.

Port 2

Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order 8 bits of address when accessing external memory. It is used for the high-order address and the control signals during programming and verification. Port 2 can sink/source one TTL load.

Port 3

Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a special function must be programmed to a one (1) for that function to operate. Port 3 can sink/source one TTL load. The special functions are assigned to the pins of Port 3, as follows:

- —RXD/data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous).
- —TXD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous).
- —$\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0.
- —$\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1.
- —T0 (P3.4). Input to counter 0.
- —T1 (P3.5). Input to counter 1.
- —$\overline{WR}$ (P3.6). The write control signal latches the data byte from Port 0 into the External Data Memory.
- —$\overline{RD}$ (P3.7). The read control signal enables External Data Memory to Port 0.

RST/V_{PD}

A low to high transition on this pin (at approximately 3V) resets the 8051. If V_{PD} is held within its spec (approximately +5V), while V_{CC} drops below spec, V_{PD} will provide standby power to the RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC}. A small internal resistor permits power-on reset using only a capacitor connected to V_{CC}.

ALE/$\overline{PROG}$

Provides Address Latch Enable output used for latching the address into external memory during normal operation. Receives the program pulse input during EPROM programming.

$\overline{PSEN}$

The Program Store Enable output is a control signal that enaoles the external Program Memory to the bus during normal fetch operations.

$\overline{EA}$/V_{DD}

When held at a TTL high level, the 8051 executes instructions from the internal ROM/EPROM when the PC is less than 4096. When held at a TTL low level, the 8051 fetches all instuctions from external Program Memory. The pin also receives the 21V EPROM programming supply voltage.

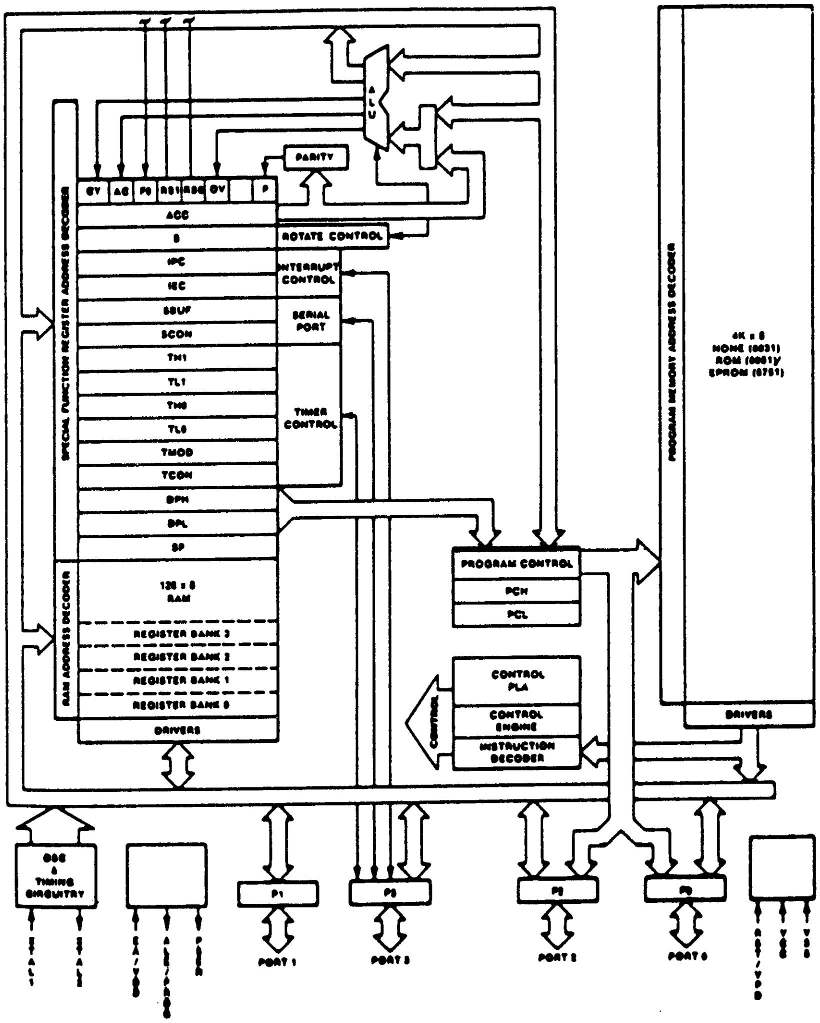
XTAL1

Input to the oscillator's high gain amplifier. A crystal or external source can be used.

XTAL2

Output from the oscillator's amplifier. Required when a crystal is used.

Block Diagram



CUSTOM IC PIN SIGNAL AND FUNCTION

EDIT.	DATE	DESIG.	CHECK	DESCRIPTION	
DESIG.		CHECK		APPR.	

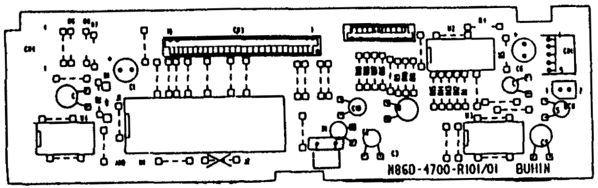
DRAW. NO.

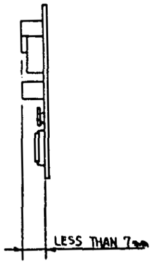
CUST.

FUJITSU LIMITED

3/3

DATE	DOCUMENT CONTROL SECTION
	→





NOTE

1. CAPACITORS AND RESONATOR (X1) TO BE BENT AS ABOVE DRAWING AFTER MOUNTED.
   CAPACITOR (C6) TO BE NOT BENT.
2. CONNECTORS (CN1.CN2.CN3) TO BE NOT FLOATED.
3. JUMPER WIRE J2 TO BE NOT MOUNTED.
4. JUMPER WIRE IS 30 POSITIONS (CONTAIN J1)
5. COMPONENTS HEIGHT IS LESS THAN 7mm.

Feb. 25 89

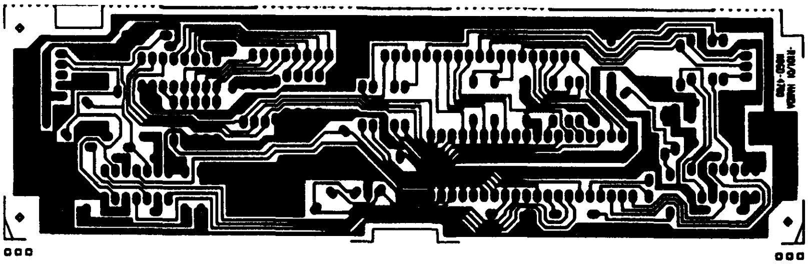
TITLE PCB ASSEMBLE

DRAW. NO. N86D-4703-P101 ⁄01

FUJITSU LIMITED

6

61.3. FDXA-4902-5



N86D-4700-R101/01 HANDA

P.C.B ARTWORK

N86D-4700-Q101/01

6

FUJITSU LIMITED

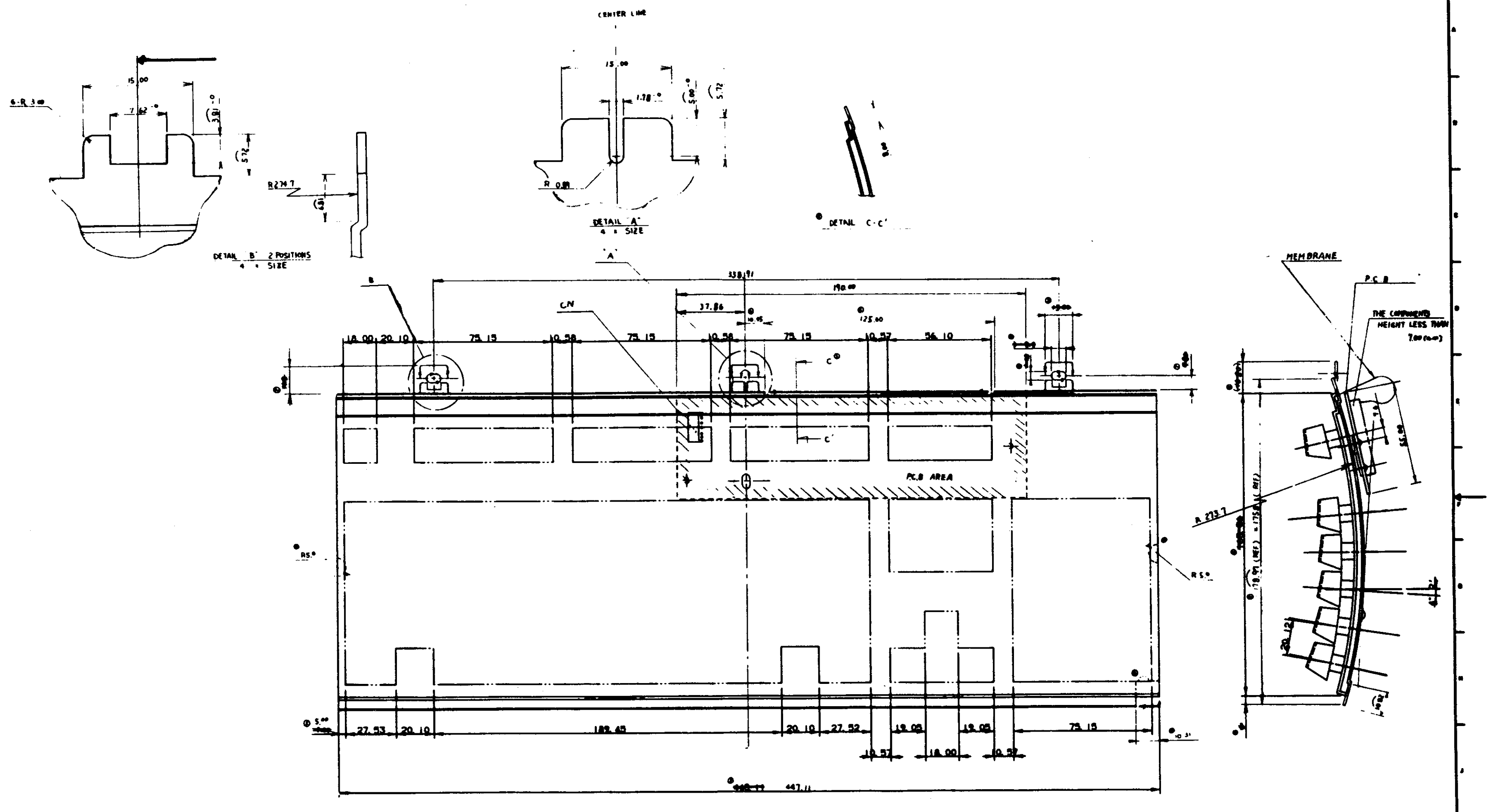
Jan. 18 98 H. Okada

S. Wada

61.3. FDXA-4902-5

DATE

DOCUMENT CONTROL SECTION



KEYBOARD UNIT ASSY

<PROPOSAL>

NOTE

1. TOLERANCE IS ± 0.5 mm UNLESS OTHERWISE SPECIFIED.
2. MEASUREMENTS WITH "⌒" MARK SHOW THE ARC.

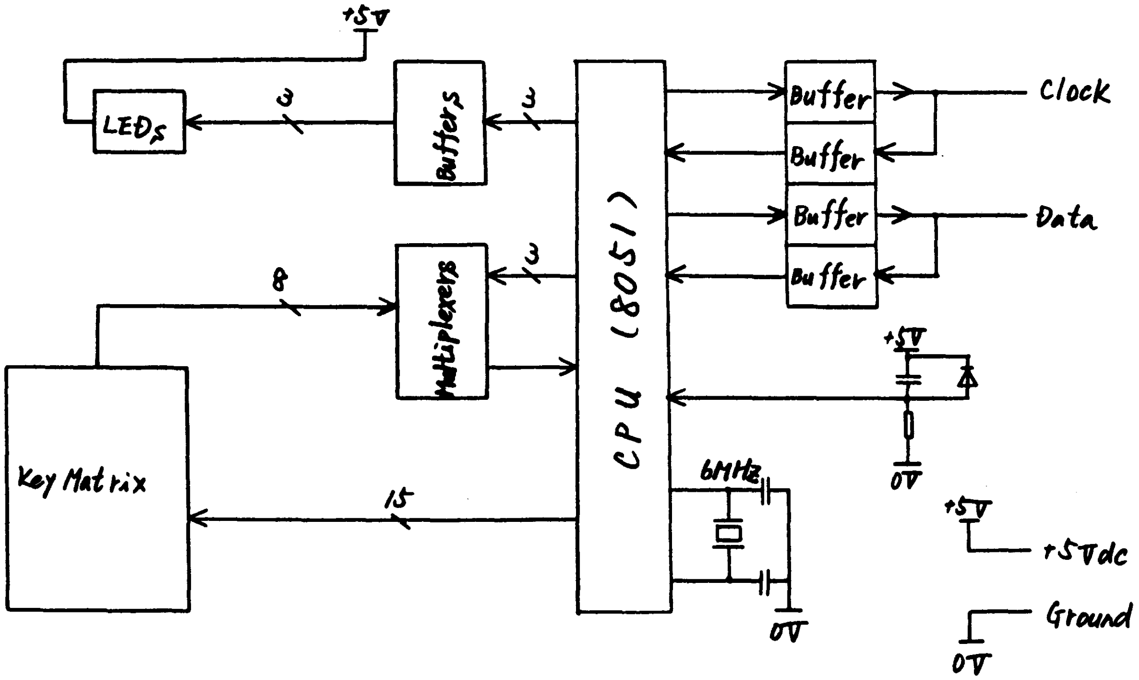
ARC

KEYBOARD
UNIT ASSY

N860-4703-U001 | 6

FUJITSU LIMITED | 1

日付	原図管理元
	→



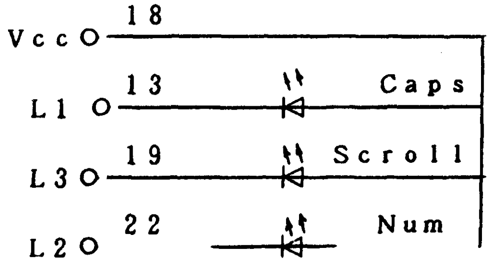
一般公差	機械・板金加工					
FN111-12	項目 ＼ 寸法区分	0.5 ~ 30	~ 120	~ 315	~ 1000	~ 2000
	中心距離（金属）	±0.1	±0.15	±0.2	±0.3	±0.5
	中心距離（樹脂） 一般寸法（金属）	±0.15	±0.25	±0.4	±0.5	±0.7
	一般寸法（樹脂）、組立寸法、 曲げ寸法（長手方向315以下）	±0.2	±0.3	±0.5	±0.8	±1.2

FN111-10	（　　）級
FN111-11	金属機械、アルミニウム合金鋳物（金・砂型）、 ガラス加工、鋳鉄、銅合金鋳物、 ゴム成形・加工、スポンジ成形・加工
FN111-13	プラスチックモールド
FN111-14	ダイカスト

材料		処理				
版	年月日	設計	調査	変更内容		
設計	87.04.08 K.Yasuda	調査		承認		
名称	BLOCK DIAGRAM	尺度	:			
図番	4700			提出先		
富士通株式会社		ページ	/			

61.2 FDXA-4006-5

		Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
		1	2	3	4	5	6	7	8
X14 ○	9	A00					A12		
X2 ○	10			B00	B11				
X13 ○	11	A01	C12	B00					
X6 ○	12	B01	B02	B03	B04	B05	B06	B07	B08
X8 ○	14	F00	F02	F03	F04	F05	F06	F07	F08
X10 ○	15	E00	E01	E02	E03	E04	E05	E06	E07
X12 ○	16	D00	D01	D02	D03	D04	D05	D06	D07
X5 ○	17	C01	C02	C03	C04	C05	C06	C07	C08
X9 ○	20	D10	F09	F10	F11	F12	F13	F15	F16
X11 ○	21	E08	E09	E10	E11	E12	E13	D08	D09
X1 ○	23	E15	E16	F14	C00	A09	E17	B20	E18
X3 ○	24	D11	D12	D13	D17	D18	D19	E19	E20
X4 ○	25	C09	C10	C11	C12	C17	C18	C19	D20
X7 ○	26	B09	B10	A05	A17	A19	B17	B18	B19
X0 ○	27	A14	A15	A16	B15	D14	D15	D16	E14



原図管理元 ↑

日付

版	年月日	設計	調査	変更内容
設計	May.30.88 Y.Yatsuda	調査		承認

名称 CIRCUIT SPECIFICATION

図番 N86C-4700-0001

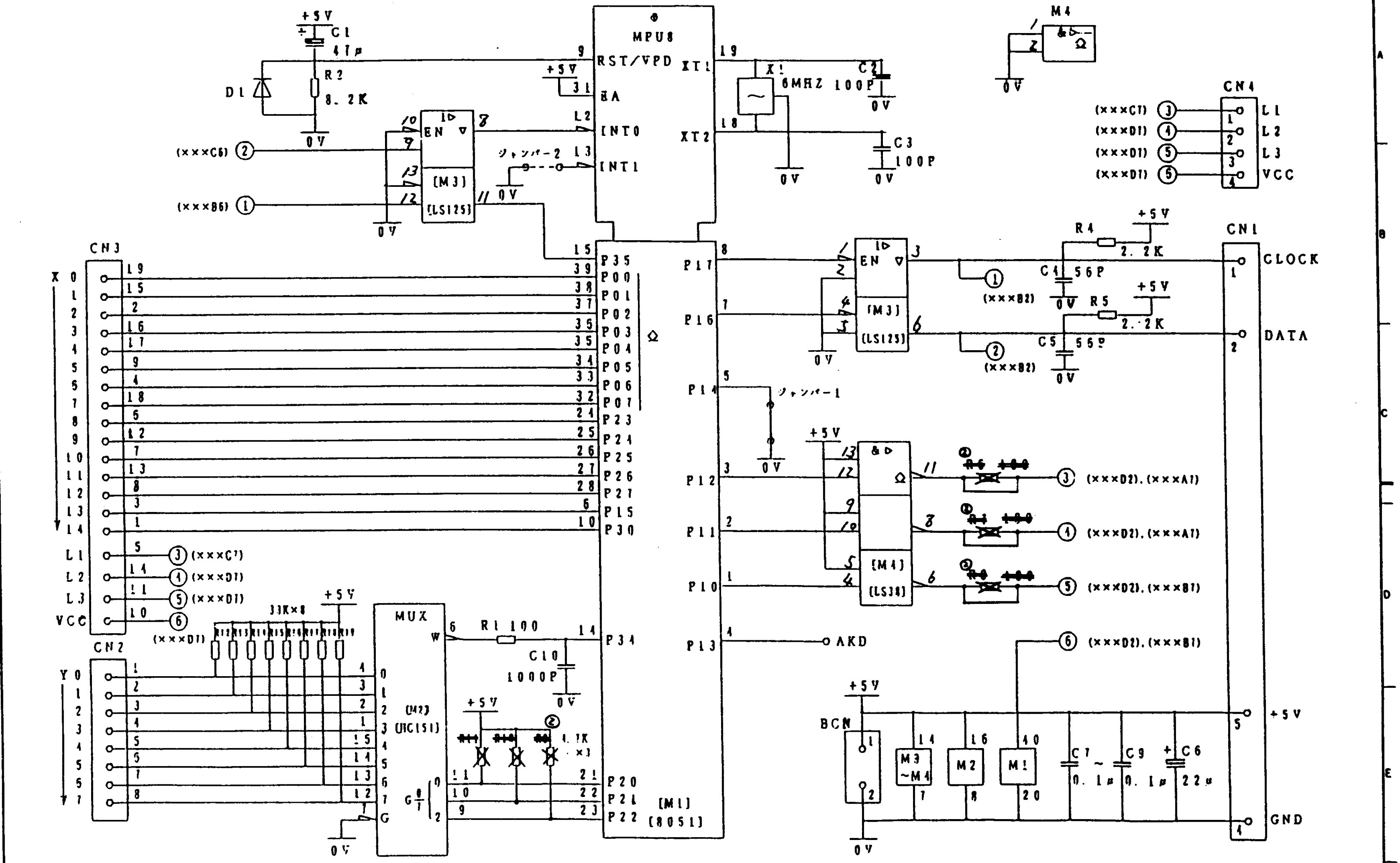
提出先 6

富士通株式会社

ページ /

3

61.3. FDXA-4003-5



CIRCUIT SPECIFICATION

N86C-4700-0101 | 6

FUJITSU LIMITED | /

02 | 88.04.28 | T. Takeda | S. Wakai | ② R6~R11 deleted | H. Ishizawa

57.08.28 | H. Tsukuda | S. Wakai

Disk Drive

SONY

MECHATRONIC PRODUCTS GROUP

PART NO. : 00-0065

REVISION : 12-88

ISSUE DATE : May 10, 88

Product Specifications

MODEL NAME :

8790144 (MP-F11W-70D)

*** Table of Contents ***

1. Description

This document describes the specifications of the SONY 3.5" Micro Floppy Disk Drive, the MP-F11W-70D that is designed for general applications, even for lap-top computers. It features a low profile, a low power cunsumption, single power voltage, a light weight, ruggedness, high reliability and easiness to use.

2. Specifications

2.1 Configuration

The drive consists of Read/Write heads, head positioning mechanism, disk motor, interface logic circuit and Read/Write circuit.

2.2 Physical Dimensions

The detailed physical dimensions are shown in Figure 2.1.
The main dimensions are:

1) Height : 25.4 mm (1.0 inch)
2) Width : 101.6 mm (4.0 inches)
3) Depth : 150.0 mm (5.9 inches)
(excluding a front panel thickness)

2.3 Weight

Weight : 425 g (0.94 pound)

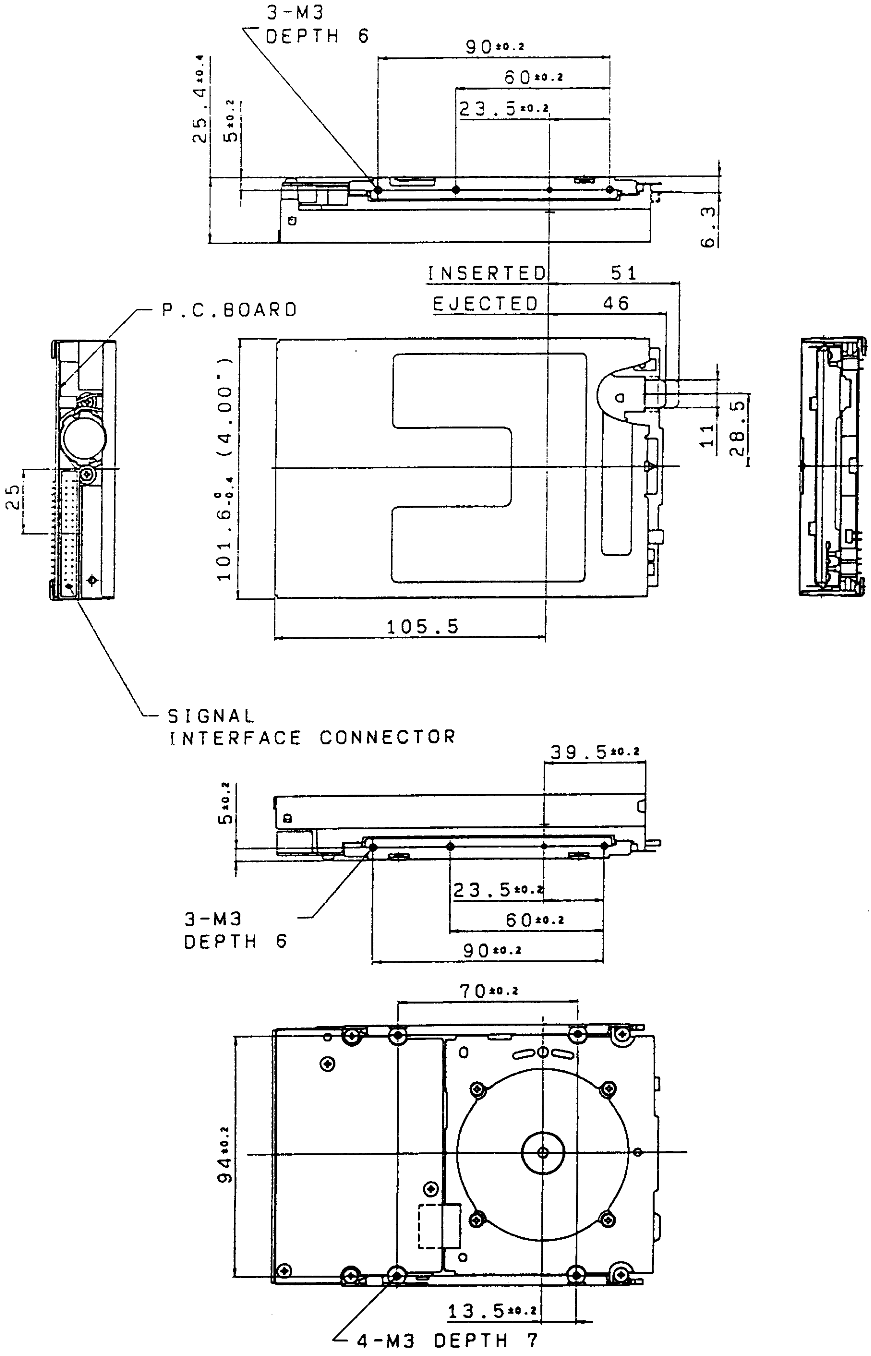
2.4 Performance

2.4.1 Recording Capacity (unformatted, MFM)

1.0 Mbytes/disk 0.5 Mbytes/surface

2.4.2 Transfer Rate

Burst transfer rate : 250 Kbits/sec for MFM



MP-F11W / F17W-70D OUTLINE DIMENSIONS

2.4.3 Access Time

a. Track to Track Slew Rate : 3 msec min.

b. Head Settling Time : 15 msec max.

c. Motor Start Time : 500 msec max.

2.4.4 Functional

a. Rotation Speed : 300 rpm
The continuous speed variation is within ±1.5%.
The instantanuous speed variation is within ±1.5%.

b. Recording Density : 8717 BPI
(Side 1, Track 79)

c. Track Density : 135 TPI

d. Number of Cylinders : 80

e. Number of Tracks : 160

f. R/W Heads : 2

2.4.5 Reliability

a. Mean Time Between Failures (MTBF) : 30,000 POH

b. Mean Time to Repair (MTTR) : 30 minutes

c. Preventive Maintenance (PM) : Not Required

d. Components life : 5 years or 15,000 POH

e. Error Rate :

1. Soft Read Error : Less than 1 per 10^9 bits read

2. Hard Read Error : Less than 1 per 10^{12} bits read

3. Seek Error : Less than 1 per 10^6 seeks

2.5 Input Power Requirements

2.5.1 Power Consumption

Standby	0.1 W max.
Operation (read/write mode)	1.1 W typ.

2.5.2 Supply Voltages

Voltage	Max. Ripple	Current
+5.0V ±10%	0.1Vpp	20 mA max. (Standby)
		220 mA Typ. (Read/Write)
		680 mA max. (Motor starts)
		890 mA max. (Step during **motor rotates**)

2.5.3 Current Profile

See Figure 2.2

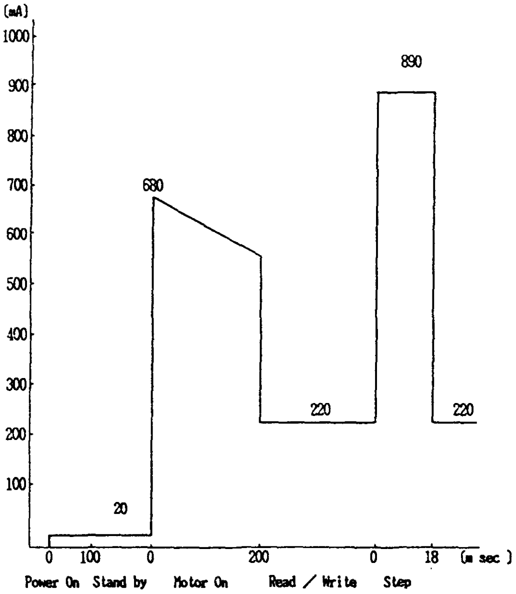


Figure 2.2. Current Profile

2.6 Environmental Limits

2.6.1 Temperature Range

Operating	:	$5^{o}C$ to $50^{o}C$ ambient	($40^{o}F$ to $122^{o}F$)
Transportation	:	$-40^{o}C$ to $60^{o}C$	($-40^{o}F$ to $140^{o}F$)
Storage	:	$-20^{o}C$ to $60^{o}C$	($-20^{o}F$ to $140^{o}F$)

2.6.2 Humidity Range

Operating : 8% to 80% relative humidity with a wet bulb temperature of $29^{o}C$ (85F) and no condensation.

Transportation and Storage : 5% to 95% relative humidity with a wet bulb temperature of $29^{o}C$ (85F) and no condensation.

2.6.3 Vibration

Operating : The unit can perform Read/Write operations without errors at continuous vibration from 10 to 500 Hz at a maximum of 0.5G along each of the two mutually perpendicular axes.

Transportation and Storage : The unit can withstand continuous vibration from 10 to 500 Hz with a maximum level of 2.0G along each of the three mutually perpendicular axes without any degradation of any characteristics below the performance specifications.

2.6.4 Shock

Operating : The unit can withstand a shock of 5.0G shock for 11 msec with a 1/2 sine wave shape in each of the two mutually perpendicular axis while performing normal read/write functions without damage or any loss of data.

Transportation and Storage : The unit when unpacked can withstand an 11 msec with a 1/2 sine wave shock of 70G on any of the three mutually perpendicular axis.

2.6.5 Orientation

The drive can be set horizontal or vertical including a top-loading. The detail orientations are shown in Figure 2.3.

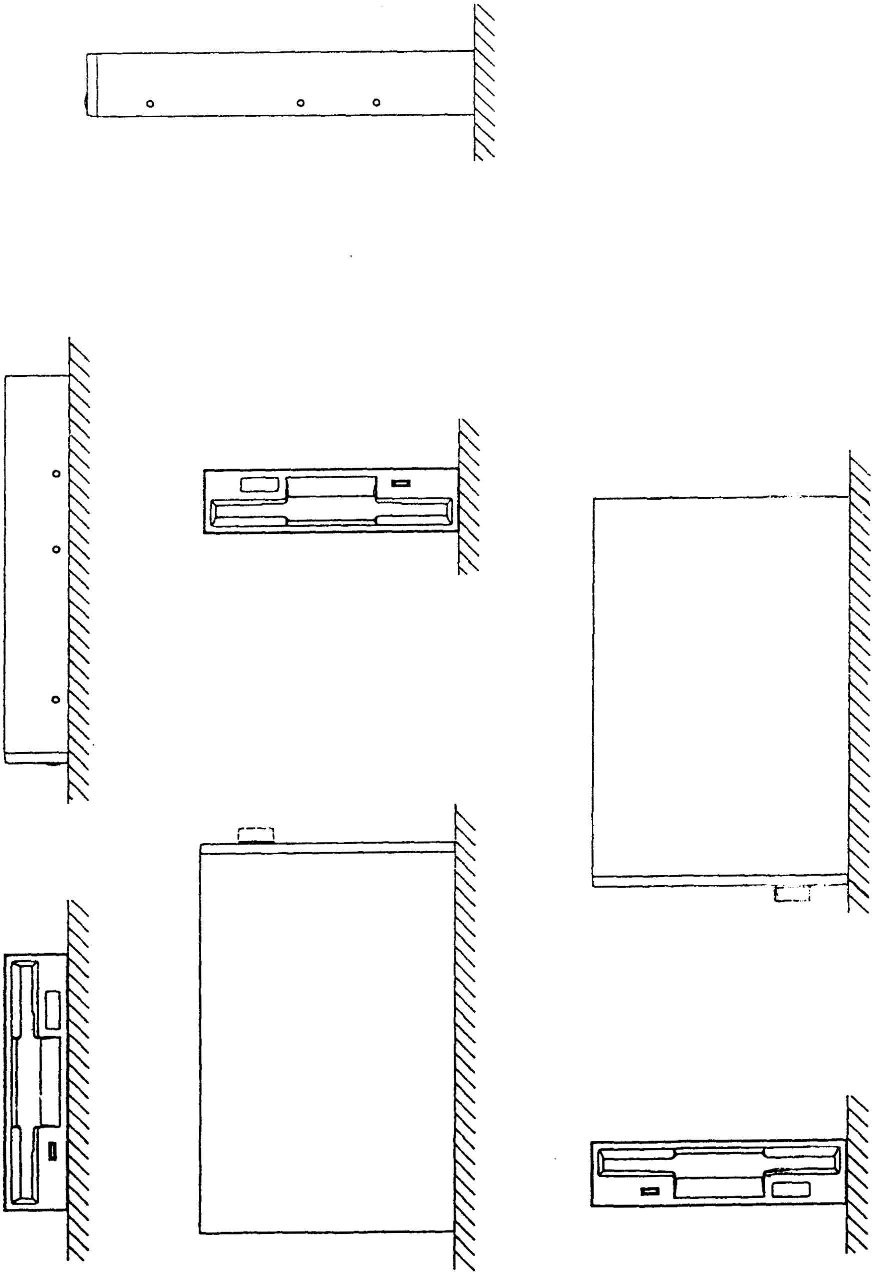


Figure 2.3. Orientations

3. Signal Interface

3.1 Connector and Pin Assignments

3.1.1 Signal connector

Receptacle	:	3M 3414-6500xx or equivalent
Cable	:	3M 3365/34 or equivalent

3.1.2 Signal Connector Pin Assignment

PIN	SIGNAL DESCRIPTION	PIN	SIGNAL DESCRIPTION
1	N.C.	2	N.C.
3	N.C.	4	N.C.
5	+ 5V	6	DRIVE SELECT 3
7	+ 5V	8	INDEX
9	+ 5V	10	DRIVE SELECT 0
11	+ 5V	12	DRIVE SELECT 1
13	RETURN	14	DRIVE SELECT 2
15	RETURN	16	MOTOR ON
17	RETURN	18	DIRECTION
19	RETURN	20	STEP
21	RETURN	22	WRITE DATA
23	RETURN	24	WRITE GATE
25	RETURN	26	TRACK 00
27	RETURN	28	WRITE PROTECT
29	N.C.	30	READ DATA
31	N.C.	32	HEAD SELECT
33	N.C.	34	DISK CHANGE

3.2 DC Characteristics of Interface Signals

3.2.1 Output Signal from Drive

C-MOS Open Drain Driver is used for the Output.

Output voltage (VOH)	:	Open
Output voltage (VOL)	:	0.4 V max.
Output current (IOL)	:	48 mA max.

3.2.2 Input Signal to Drive

Input voltage (VIH)	:	2.2 V min.
Input voltage (VIL)	:	0.8 V max.

3.2.3 Recommended Circuit for Signal Interface

The detail interface circuits are shown in Figure 3.1.

A 1K ohm pull-up resister can be put on each output line from the drive. The recommended cable length is 1.5 m (4.92ft.) or less.

Recommended driver ICs : 7406, 7438 or equivalent.

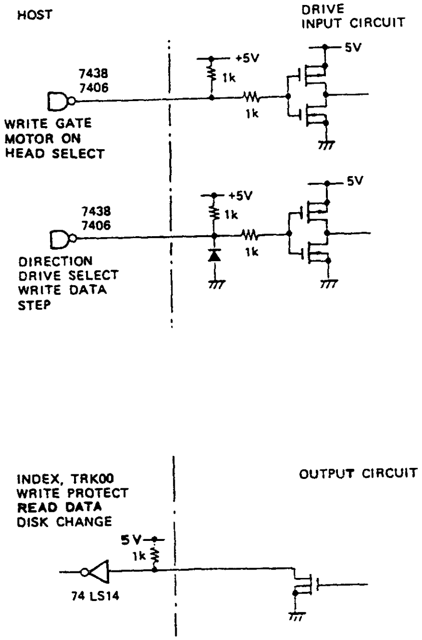


Figure 3.1. Interface Circuits

3.3 Signal Definitions

3.3.1 DRIVE SELECT 0,1,2,3

The SELECT lines are used to enable or disable all other interface lines except a MOTOR ON line. When the SELECT line is true (low), the drive is enabled and is considered active. When the SELECT line is false (high), all controlled inputs except the MOTOR ON line are ignored and all output lines are disabled.

N.B. IN USE (LED) lamp

When a drive is selected, the IN USE lamp on the selected drive is turned on, and when a drive is not selected, it is turned off.

3.3.2 MOTOR ON

When this input is true (low), the spindle motor will start to run. When this line is made false (high), the spindle motor will decelerate and stop. The spindle motor will not rotate until a disk is inserted even if the MOTOR ON signal is low (true). If the MOTOR ON signal becomes false during either a write or erase operation, the disk motor will not stop rotating until both the ERASE GATE signal and the WRITE GATE signal become high (false).

3.3.3 STEP

When a drive is selected, a true (low) pulse on this line will cause the read/write heads to move to the adjacent track. The direction of the head movement is determined by the DIRECTION input at the trailing edge of the pulse.

The step operation can be performed even if there is no disk inserted in the drive.

3.3.4 DIRECTION

When a drive is selected, a false (high) level on this input will cause a STEP input to move the read/write head away from the disk spindle. A true (low) level will cause a STEP input to move the read/write head toward the drive spindle.

3.3.5 HEAD SELECT

When a drive is selected, a true (low) level on this input will cause Head 1 (upper) to be selected. A false (high) level on this input will cause Head 0 (lower) to be selected. If the HEAD SELECT signal changes during either write or erase operation, the head will not be changed until both ERASE GATE and WRITE GATE signal become high (false).

3.3.6 WRITE GATE

When a drive is selected and this line is made true (low), the write current circuit is enabled and information in the WRITE DATA input may be written.

3.3.7 WRITE DATA

If a drive is selected, and the WRITE GATE is true (low), a true pulse (low) on the WRITE DATA line will cause a bit to be written on the disk. Pulses on this line will be neglected when WRITE GATE is false (high). No pre-compensation is required.

3.3.8 INDEX

When the drive is selected, a true (low) pulse is generated on this line by each revolution of the spindle.

3.3.9 TRACK 00

This line is true (low) when the drive is selected and the Read/Write head is positioned on track 00.

3.3.10 WRITE PROTECT

If a write-protect disk or no disk is inserted while a drive is selected, this line will be true (low) and the drive will not be able to write data.

3.3.11 READ DATA

When the drive is selected, a true (low) pulse is generated on this line every time a bit is detected.

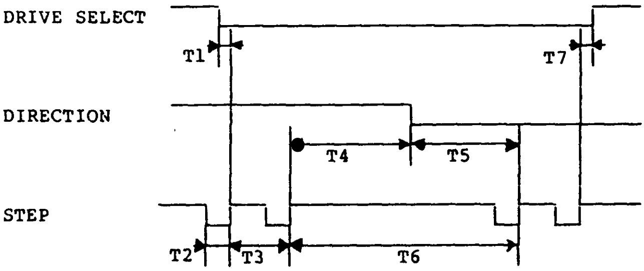
3.3.12 DISK CHANGE

This line is true (low) whenever a disk is removed from the drive. The line will remain true (low) until both the following conditions have been met:

1. A disk is inserted.
2. A STEP pulse has been received when the drive is selected.

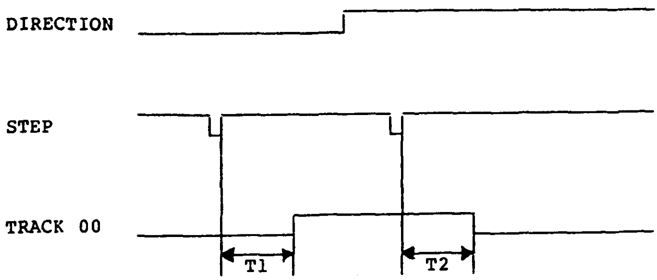
3.4 Timing Requirements

3.4.1 Head Access



T1	:	0.5 usec min.
T2	:	0.5 usec min.
T3	:	3.0 msec min.
T4	:	0.5 usec min.
T5	:	0.5 usec min.
T6	:	18 msec min.
T7	:	0.5 usec min.

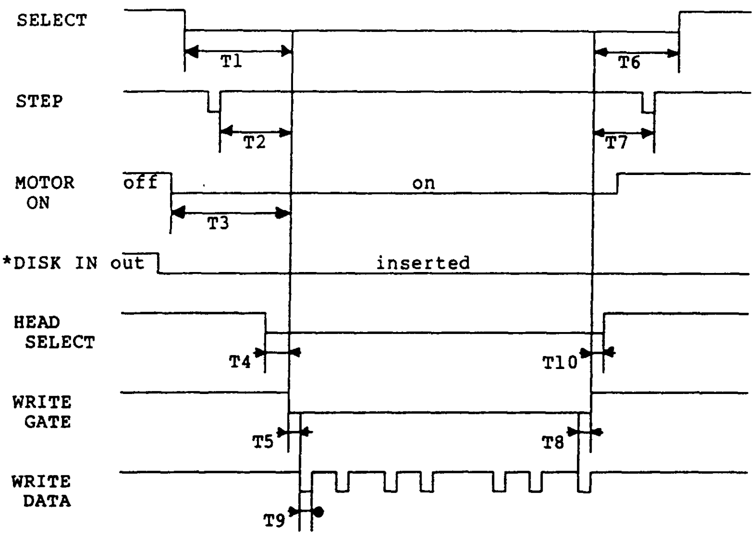
3.4.2 TRACK 00 Signal



T1 : 2.9 msec max.

T2 : 2.9 msec max.

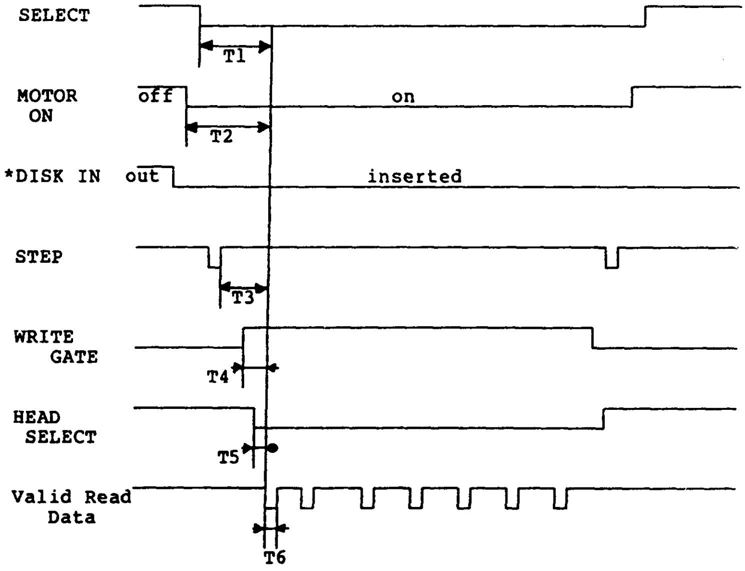
3.4.3 Write Data Timing



T1	:	0.5 usec min.	T6	:	0.5 usec min.
T2	:	18 msec min.	T7	:	905 usec min.
T3	:	500 msec min.	T8	:	8 usec max.
T4	:	100 usec min.	T9	:	150 ns min., 2000 ns max.
T5	:	8 usec max.	T10	:	See 3.3.5

*DISK IN, the disk-in sensor signal inside the drive, is low when a disk is inserted in the drive.

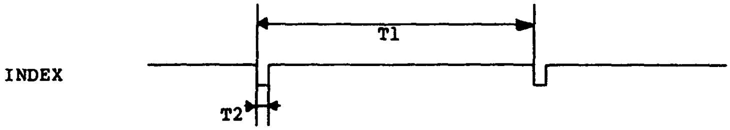
3.4.4 Read Data Timing



T1	:	0.5 usec max.	T4	:	935 usec max
T2	:	500 msec max.	T5	:	100 usec max
T3	:	18 msec max.	T6	:	500 nsec min., 1200 nsec max.

*DISK IN, the disk-in sensor signal inside the drive, is low when a disk is inserted in the drive.

3.4.5 Index Pulse

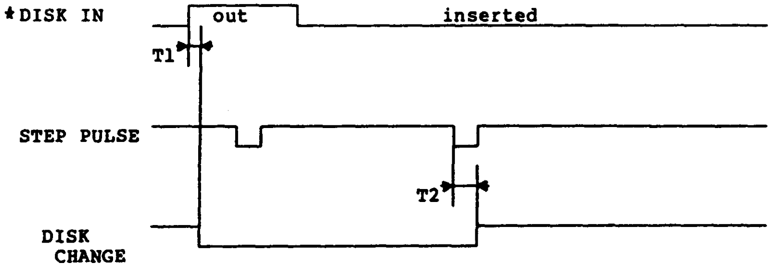


T1* : 197 msec min., 203 msec max.

T2 : 1.5 msec min., 1.7 msec max.

*When the disk motor rotation is at the steady state.

3.4.6 Disk Change



T1 : 0.5 usec max. T2 : 1.0 usec max.

*DISK IN, the disk-in sensor signal inside the drive, is low when a disk is inserted in the drive.

3.5 Power On and Power off Requirements

3.5.1 Data Protection

Turning power on or off will not cause any damage to recorded data on the disk as long as the drive is not in the midst of writing when the power is shut off or supplied.

3.5.2 Power Supply Sequencing

When the power is turned on, no special power supply sequencing is required. When the power is turned off, the power supply must fall monotonically to zero volt.

3.6 Power-On Reset Timing

A drive will be ready in 100 msec after power on. It takes up to 100 msec to reset the control IC of the drive.

4. Safety

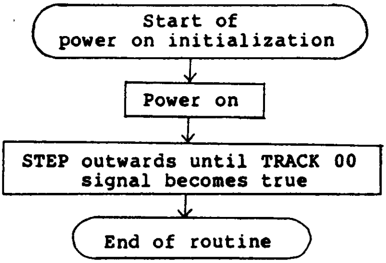
The drive will meet the following product safety regulations:

U.L. 478
C.S.A. C.22.2, No.154
U.L. 94V-0 for Front Bezel

5. Power On Initialization

In order to reduce the peak current requirement when drives are used in a daisy chain, the MP-F11W-70D has been designed not to seek track 00 automatically. If all the drives connected in the daisy chain sought trach 00 simultaneously, this would place a significant power drain on the host system.
Thus, the host system must perform the following routine just after power on in order to reset the track counter inside the drive.

Power on initialization



SONY

MECHATRONIC PRODUCTS GROUP

PART NO. : 061

REVISION : 11-88

ISSUE DATE : Mar. 4, 88

OEM MANUAL

MODEL NAME :

MP-F11W / MP-F17W

*** Table of Contents ***

1. INTRODUCTION

1-1. Purpose

This material provides the information necessary to interface the MP-F11W/MP-F17W Micro Floppydisk drive to a floppy disk controller.

1-2. General description

The SONY Micro Floppydisk Drive represents a technological break through offering extreme lightweight, compactness, (101.6mm wide by 25.4mm high by 150.0mm deep, weighting 425g) that provides a versatile data storage component for the systems designer.

SONY's leadership in high-density recording techniques, perfected in video technology, enabled SONY engineers to develop the 3.5" Micro Floppydisk standard. Yet an unformatted, storage capacity of 1MBytes (MP-F11W, double density), 2MBytes (MP-F17W high density) in 135 tracks per inch provides that of conventional 5.25" disks or more.

A semi-rigid protective shell provides protection unique to the Micro Floppydisk. When the disk is inserted into or taken out of the drive, its shutter automatically opens and closes to protect the disk from dust, dirt, fingerprints and other foreign objects that might degrade performance. A metal centering hub allows positioning with greater ease and more positive accuracy in over 30,000 disk interchanges.

The SONY proprietary read/write and tunnel erase head, developed using video techniques, is positioned by a precision stepper motor assembly, providing fast access while maintaining high positioning accuracy. High coercivity media accomplishes high data integrity with the SONY high density head.

The SONY Micro Floppydisk drive is interface signal-compatible with conventional 5" floppydisk drives. Accordingly, popular FDC chips such as Western Digital FD1WDs, NEC µPD765 series can be used.

Whether your application is small business systems, test instruments, personal computers or any related application, you will find that the Micro Floppydisk drive will offer a whole new range of possibilities.

The Micro Floppydisk drive offers the following features :
* 3.5" floppy disk media with automatic shutter mechanism
* Large capacity .... 1MB(MP-F11W) / 2MB(MP-F17W)
* Small footprint
* Light weight
* Low power consumption
* High track density .... 135 TPI
* High reliability .... MTBF 30,000POHs
* High transfer rate .... 250Kbps(1MB), 500Kbps(2MB)

2. MECHANISM

The mechanism mounted on the chassis consists of a cassette compartment mechanism, a disk chucking and disk rotation mechanism, magnetic heads and carriage, a head positioning mechanism, some detectors, and so on.

2-1. Chassis

The chassis made of aluminum diecast is strong, highly durable, and precision-manufactured. This is the base structure for mounting most of mechanisms and printed circuit board.

2-2. Cassette compartment mechanism

The cassette compartment mechanism precisely positions the cassette on chassis by one touch operation. The mechanism is designed that the cassette is ejected easily as well.

2-3. Disk chucking and disk rotation mechanism

The disk chucking mechanism is mounted on the top of the thin brushless direct drive DC motor. The disk chucking mechanism precisely positions a disk and drives it.

The disk rotation mechanism directly mouted in a chassis consists of a printed circuit board for drive control, a rotor magnet, a rotor yoke, a stator yoke, and so on.

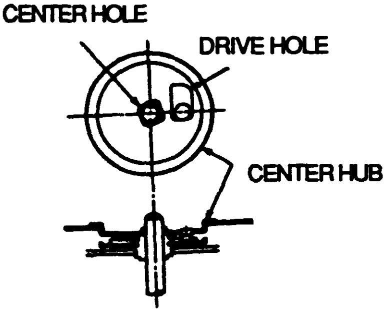


Fig. 2-1 Chucking Mechanism

2-4. Magnetic heads and head carriage

The tunnel erase type head with narrow track width is developed to attain a 135 TPI track density. The head is designed for higher Read signal quality. The head surface is treated for less the disk wear and the head wear. The head comprises a read/write gap to read and write data and two erase gaps to erase the recorded track edge immediately after recording. The heads and the carriage are very impotant parts in the MFDD, so they are especially precision-assembled.

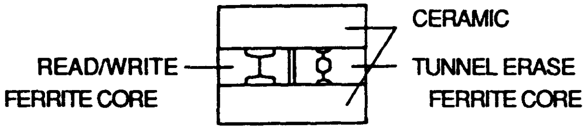


Fig. 2-2 Magnetic Head

2-5. Head positioning mechanism

The head positioning mechanism consists of a stepping motor, a lead screw, a bearing, and a guide shaft. The head carriage is held with the guide shaft and the lead screw with stepping motor. The 4-phase stepping motor rotates 2 micro steps (18°X 2) per track.

2-6. Detection mechanism

a) Write protect

The micro switch is mounted on the printed circuit board to detect the position of the write protect tab on the disk. If a write-protected disk is inserted and the micro switch is not pushed down, the recording/erasing power is not supplied and the data is protected from an erroneous writing command.

b) Track 00

The mechanism consists of a photo-interrupter to detect the outermost track position (Track 00). The head carriage is not moved more outer track, after the MFDD detects the Track 00. So, the head and head positioning mechanism do not get damaged.

c) Index

Since the MFDD is designed to make the spindle motor and the disk be fixed in the same corresponding position, the Index pluse is generated by the spindle motor circuit.

d) Disk-in

If the cassette pushes the micro switch on the printed circuit board, it is detected that the cassette is inserted.

e) Density select (Only for MP-F17Ws)

The micro switch is mounted on the printed circuit board to detect the hole of the high density disk. If a high density cassette is inserted and the micro switch is not pushed, the MFDD works in high density mode. If a nomal density disk is inserted and the micro switch is pushed down, the MFDD works in normal density mode.

3. ELECTRONICS

The electronics of the MFDD consists of a read/write circuit, a control circuit and some drivers, and so forth. The electronic circuits are mounted on two printed circuit boards (Motor and Logic boards).

3-1. Read/write circuit

The read/write circuit consists of a read circuit, a write circuit, and a power on detection circuit. These circuits are mostly included in a read/write amplifier LSI(bipolar) on the Logic board.

3-2. Control circuit

The control circuit consists of an input and output interface circuit, a spindle motor control circuit, a stepping motor control circuit, a sensor circuit, and a function control circuit. These circuits are mostly included in a control LSI(C-MOS) on the Logic board.

3-3. Spindle motor driver circuit

This circuit consists of a three phase spindle motor driver, a rotation speed detector, a rotation phase detector, and a current limiter. These circuits are mostly included in a spindle motor driver IC(bipolar) on the Motor board.

3-4. Stepping motor driver circuit

This circuit consists of two bridge driver circuits. The IC is made by the bipolar process, which is mounted in the Logic board.

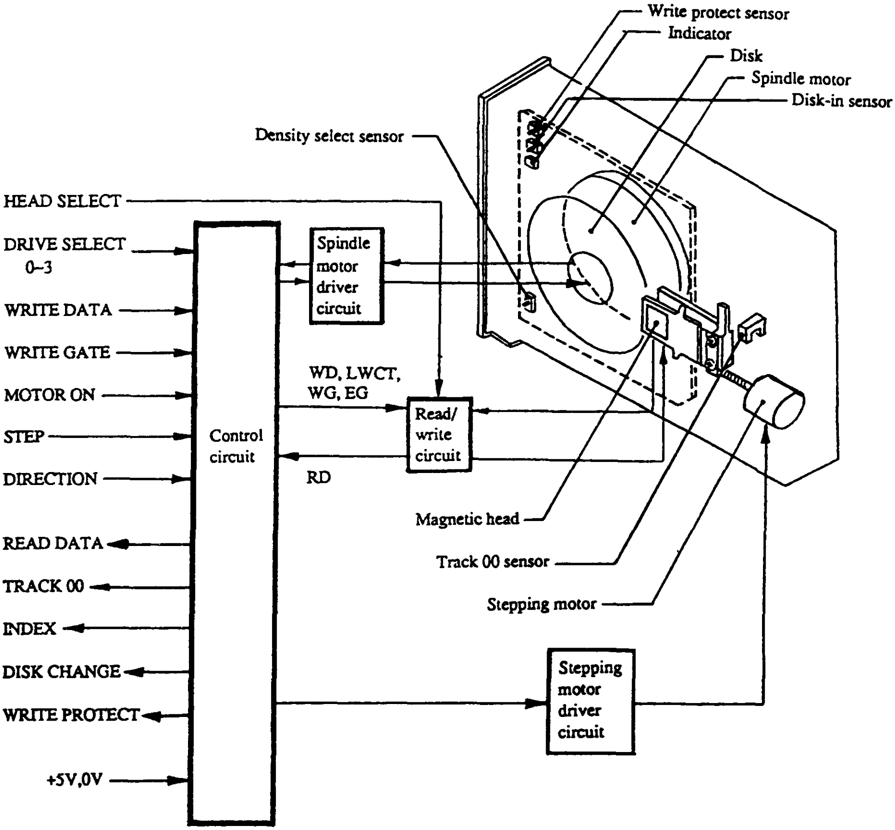


Fig. 3-1 General Block Diagram

4. SPECIFICATIONS

4-1. Drive performance

Table 4-1 lists performance specifications for the SONY Micro Floppydisk drive.

TABLE 4-1 Performance Specifications

Specification	MP-F11W	MP-F17W (2MB mode)
CAPACITY		
Unformatted per disk	1MB for MFM	2MB for MFM
Unformatted per track	6.25KB for MFM	12.5KB for MFM
Recording density	8,717BPI	17,434BPI
Burst transfer rate	250Kbits/sec	500Kbits/sec
ACCESS TIME		
Track to track	3msec	
Settling Time	15msec	
FUNCTIONAL		
Rotational speed	300rpm	
Track density	135TPI	
Cylinders	80	
Tracks	160	
R/W heads	2	
Encoding method	MFM or FM	
POWER CONSUMPTION		
Read/Write mode	220mA	
Standby mode	20mA(TTL I/F), 6mA(C-MOS I/F)	

4-2. Dimensional data

Table 4-2 lists the dimensional data for the Micro Floppydisk drive.

TABLE 4-2 Dimensional Data

Physical Dimension	Value
Height	25.4 mm
Width	101.6 mm
Depth	150.0 mm
Weight	425 g

5. INTERFACE DESCRIPTION

5-1. Host system interface

The SONY Micro Floppydisk drive is compatible with conventional floppy disk controllers. The interface consists of 15 signal lines for data, control and hand-shaking (see Figure 5-1, details will appear in a product specification of each mode).

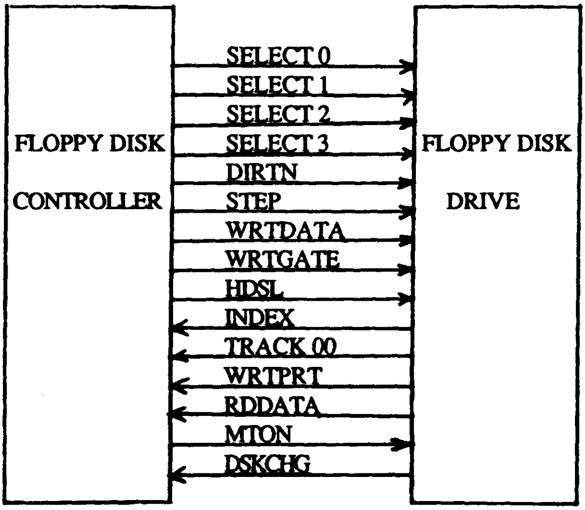


Fig. 5-1 Interface Signal Diagram

5-2. Signal interface

The drive has 15 interface signals. Ten signals are input to the drive, and six are output.

Table 5-2 MNEMONIC

SIGNAL DESCRIPTION	MNEMONIC
DRIVE SELECT 0	SELECT 0
DRIVE SELECT 1	SELECT 1
DRIVE SELECT 2	SELECT 2
DRIVE SELECT 3	SELECT 3
DIRECTION SELECT	DIRTN
STEP	STEP
WRITE DATA	WRTDATA
WRITE GATE	WRTGATE
HEAD SELECT	HDSL
INDEX	INDEX
TRACK00	TRK00
WRITE PROTECT	WRTPRT
READ DATA	RDDATA
MOTOR ON	MTON
DISK CHANGE	DSKCHG

6. EJECT BUTTON, INDICATOR AND SELECT SWITCH

6-1. Eject button

The eject button (see Fig. 6-1) is used to remove a disk cartridge from the unit. Depression of the eject button causes the disk cartridge in the unit to be ejected.

6-2. Indicator

The activity indicator (see Fig. 6-1) indicates that a drive is selected.

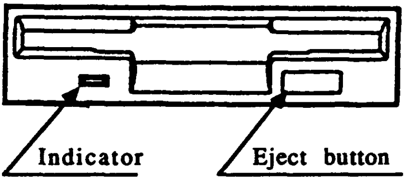


Fig. 6-1 Front Bezel

6-3. Drive select switch

Drive Select switch located next to a power connector on the rear side is used to designate drive0 up to drive3 in a daisy chain application. Usually, this switch is set to drive 1 in shipping from the drive factory.

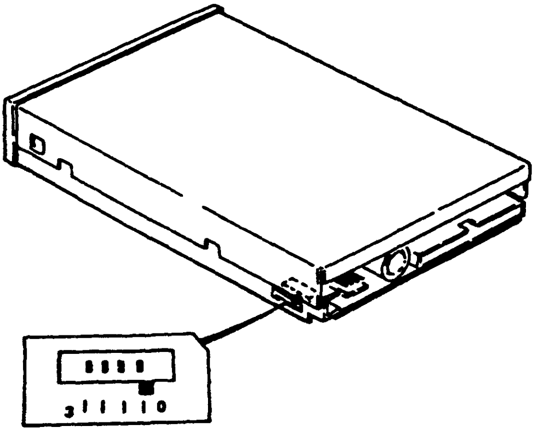


Fig. 6-2 Drive Select Switch

SECTION 4
PART REPLACEMENT

4-1 COVER ASS'Y REPLACEMENT

4-1-1 Removal

a. Insert your finger into the rear right side portion of the Cover Ass'y as shown in Fig. 4-1, and dislocate the Cover Ass'y from the hook while applying some force to the direction marked with arrow. The whole Cover Ass'y can be taken out.

4-1-2 Installation

a. Match the projection located in the front side of the Cover Ass'y with square hole of the front panel ass'y and then put the rear of the Cover Ass'y into the proper position. (Refer to Fig. 4-1)

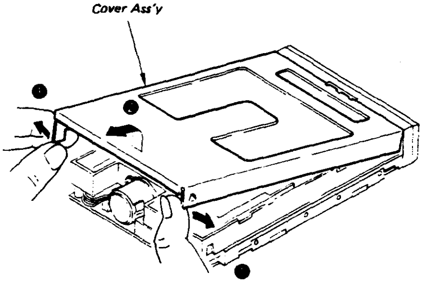


Fig. 4-1 Cover Ass'y Replacement

4-2 FRONT PANEL ASS'Y AND EJECT BUTTON REPLACEMENT

4-2-1 Removal

a. Remove the cover ass'y. (Refer to 4-1)

b. Push slightly the plastic hinges located on each of right and left sides with (-) shaped screw driver through the square hole of upper cover, and pull it slowly toward you. (Refer to Fig. 4-2)

c. Dislocate the Eject Button from the hook of the slide plate.

4-2-2 Installation

a. Hang the Eject Button to the hook of slide plate as shown Fig. 4-2.

b. Install the Front Panel Ass'y into the metal frame by sliding in the hook located on each side of the Front Panel Ass'y.

c. Make sure that protuberances of LED and the Eject Button are properly located in the recess and square hole of the Front Panel Ass'y.

d. Install the cover ass'y. (Refer to 4-1)

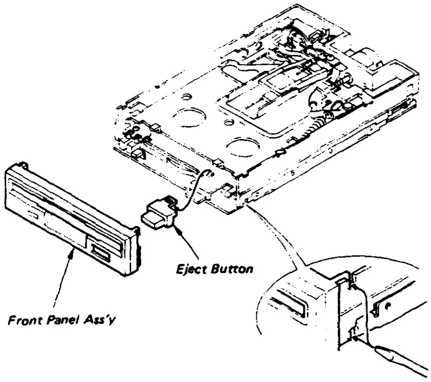


Fig. 4-2 Front Panel Ass'y Replacement

4-3 LG-2 MOUNTED BOARD REPLACEMENT

4-3-1 Removal

a. Remove the cover ass'y. (Refer to 4-1)

b. Disconnect the all connectors. (CN102 for stepping motor, CN104 for 00 sensor, CN105 and CN106 for head carriage ass'y)

c. Desolder the jumper cable (CN1) on MT-2

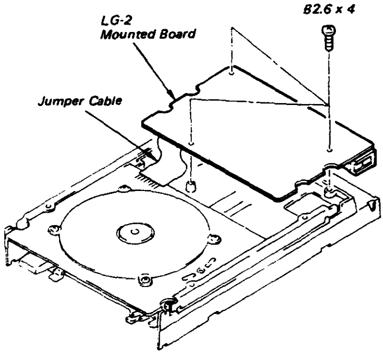


Fig. 4-3 LG-2 Mounted Board Replacement

d. Remove the three screws (B2.6x4) securing the LG-2 Mounted Board and then remove the LG-2 Mounted Board. (Refer to Fig. 4-3)

4-3-2 Installation

a. Solder the jumper cable to the CN1 of MT-2 mounted board.

b. Install the LG-2 Mounted Board with three screws (B2.6x4).

c. Connect the all connectors.

d. Perform the Radial Alignment and TRK00 sensor Adjustment. (Refer to 5-1)

e. Install the cover ass'y. (Refer to 4-1)

4-4 MT-2 MOUNTED BOARD REPLACEMENT

4-4-1 Removal

a. Desolder the jumper cable on MT-2 Mounted Board with soldering iron.

b. Remove the four screws (P2.6x4) securing the Stator Yoke Ass'y and then remove the Stator Yoke Ass'y. (Refer to Fig. 4-4)

c. Disconnect the connenctor CN2 (hall IC ass'y) and then remove the MT-2 Mounted Board.

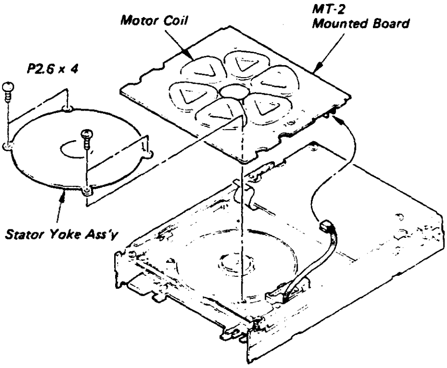


Fig. 4-4 MT-2 Mounted Board Replacement

4-4-2 Installation

a. Connect the connector CN2 (hall IC ass'y) onto the MT-2 Mounted Board.

b. Solder the Jumper Cable to CN1 of MT-2 Mounted Board.

c. Match the hole of the MT-2 Mounted Board with the emboss of the chassis ass'y.

d. Install the Stator Yoke Ass'y on MT-2 Mounted Board with four screws (P2.6x4), not to damage to six motor coils for disk motor. (Refer to Fig. 4-4)

4-5 HALL IC ASS'Y REPLACEMENT

4-5-1 Removal

a. Remove the four screws (P2.6x4) securing the stator yoke ass'y and then remove the stator yoke ass'y. (Refer to Fig. 4-4)

b. Disconnect the connector CN2 (Hall IC Ass'y).

c. Remove the screw (PSW2x5) securing the Hall IC Ass'y and then remove the Hall IC Ass'y. (Refer to Fig. 4-5)

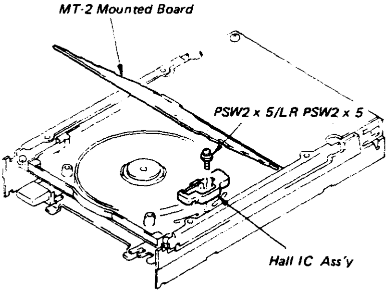


Fig. 4-5 Hall IC Ass'y Replacement

4-5-2 Installation

a. Fasten loosely the Hall IC Ass'y with a screw (PSW2x5). But this screw must not be tightened for later adjustment.

b. Connect the connector CN2 (Hall IC Ass'y) onto the MT-2 mounted board.

c. Match the hole of the MT-2 mounted board with the emboss of the chassis ass'y.

d. Install the stator yoke ass'y on MT-2 mounted board with four screws (P2.6x4), not to damage to six motor coils for disk motor. (Refer to Fig. 4-4)

e. Perform the Index phase adjustment. (Refer to 5-3)

4-6 CASSETTE HOLDER ASS'Y REPLACEMENT

4-6-1 Removal

a. Connect the drive to the MFD Function Checker (Refer to Fig. 2-14), move the head to the TRK00 and then disconnect the drive from MFD Function Checker.

b. Remove the cover ass'y. (Refer to 4-1)

c. Remove the front panel ass'y and eject button. (Refer to 4-2)

d. Be careful not to apply the excessive force to the head carriage ass'y. While pushing the eject lever, take the Cassette Holder Ass'y. (Refer to Fig. 4-6)

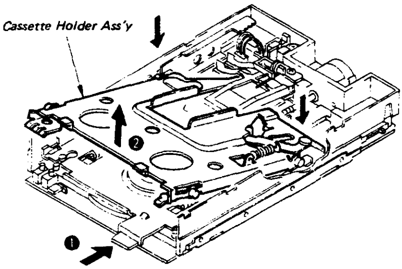


Fig. 4-6 Cassette Holder Ass'y Replacement

4-6-2 Installation

a. Insert carefully the Cassette Holder Ass'y underneath the arm of the head carriage ass'y, then set the holder into the location shown by arrow as shown in Fig. 4-6, while pushing the eject lever.

b. Install the front panel ass'y and eject button. (Refer to 4-2)

c. Install the cover ass'y. (Refer to 4-1)

4-7 SLIDE PLATE ASS'Y REPLACEMENT

4-7-1 Removal

a. Remove the cover ass'y. (Refer to 4-1)

b. Remove the front panel ass'y and eject button. (Refer to 4-2)

c. Remove the cassette holder ass'y. (Refer to 4-6)

d. Slide the trigger arm to set the Disk In-mode as shown in Fig. 4-7.

e. Remove the one end of Tension Springs on chassis ass'y and then take the Slide Plate Ass'y. (Refer to Fig. 4-7)

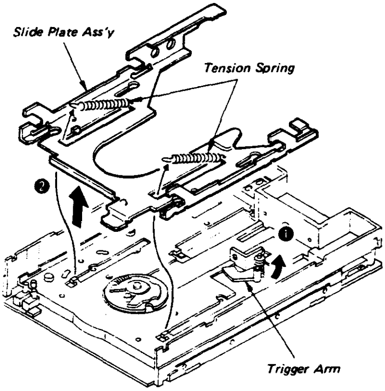


Fig. 4-7 Slide Plate Ass'y Replacement

4-7-2 Installation

a. While pushing the trigger arm, set the Slide Plate Ass'y into the location shown in Fig. 4-7.

b. Hang the Tension Springs on hook of chassis ass'y.

c. Push the Slide Plate Ass'y, while pushing the trigger arm back side. (This stays "disk-in mode")

d. Install the cassette holder ass'y. (Refer to 4-6)

e. Install the front panel ass'y and eject button. (Refer to 4-2)

f. Install the cover ass'y. (Refer to 4-1)

g. Make the head clean. (Refer to 5-4)

4-8 00 SENSOR REPLACEMENT

4-8-1 Removal

a. Connect the drive to the MFD Function Checker as shown in Fig. 2-14, move the head to TRK79 and then disconnect the drive from the MFD Function Checker.

b Remove the cover ass'y. (Refer to 4-1)

c. Disconnect the connector CN104 (00 Sensor) from the LG-2 mounted board.

d. Remove the screw (PSW2.6x5) securing the 00 Sensor and remove 00 Sensor. (Refer to Fig. 4-8)

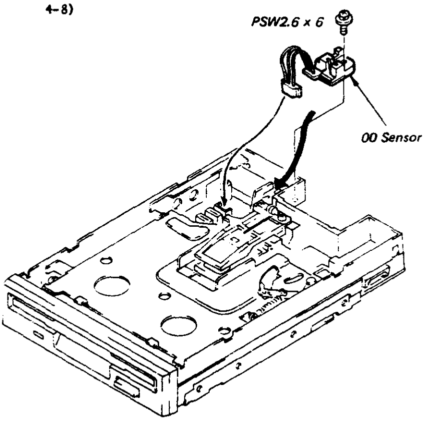


Fig. 4-8 00 Sensor Replacement

4-8-2 Installation

a. Install the 00 Sensor with a screw (PSW2.6x5).

b. Connect the CN104 connector onto the LG-2 mounted board.

c. Perform the radial alignment and TRK00 sensor adjustment. (Refer to 5-1)

d. Install the cover ass'y. (Refer to 4-1)

4-9 HEAD CARRIAGE ASS'Y REPLACEMENT

4-9-1 Removal

a. Remove the cover ass'y. (Refer to 4-1)

b. Remove the front panel ass'y and eject button. (Refer to 4-2)

c. Remove the cassette holder ass'y. (Refer to 4-6)

d. Disconnect the connectors CN105 and CN106 (flexible boards) from the LG-2 mounted board.

e. Remove the screw (P2.6x4) securing the Guide Retainer and then remove the Guide Retainer, Head Carriage Ass'y and Slide Guide Shaft. (Refer to Fig. 4-9)

4-9-2 Installation

Note: Apply Sony oil to the Guide Shaft before installation. Apply Sony oil to the openings of Head Carriage Ass'y using the bamboo stick.

a. Pass the Guide Shaft through the opening of Head Carriage Ass'y.

Note: The spring of Head Carriage Ass'y, that is located in around shaft hole, should be installed inside so that the Guide Shaft is forced outwardly by the spring force, as shown in Fig. 4-9.

b. Put the Head Carriage Ass'y and the Slide Guide Shaft in place, and install the guide retainer with a screw (P2.6x4). (Refer to Fig. 4-9)

c. Connect the flexible boards to CN105 and CN106 on the LG-2 mounted board.

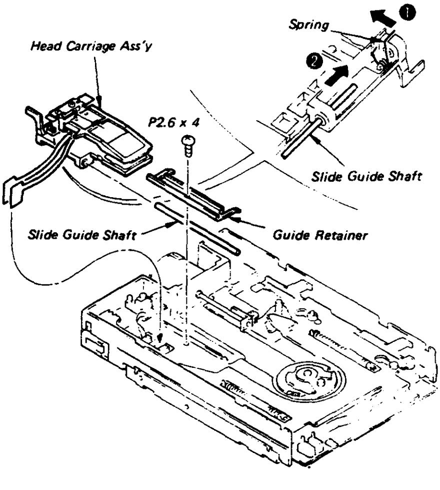


Fig. 4-9 Head Carriage Ass'y Replacement

d. Install the cassette holder ass'y. (Refer to 4-6)

e. Install the front panel ass'y and eject button. (Refer to 4-2)

f. Perform the radial alignment and TRK00 sensor adjustment. (Refer to 5-1)

g. Perform the head compliance. (Refer to 5-2)

h. Install the cover ass'y. (Refer to 4-1)

i. Make the head clean. (Refer to 5-4)

4-10 STEPPING MOTOR ASS'Y (ROTOR ASS'Y AND STATOR ASS'Y) REPLACEMENT

4-10-1 Removal

a. Remove the cover ass'y. (Refer to 4-1)

b. Remove the front panel ass'y and eject button. (Refer to 4-2)

c. Remove the cassette holder ass'y. (Refer to 4-6)

d. Disconnect the connector CN102 (Stator Ass'y) from the LG-2 mounted board.

e. Remove the three screws (B2.6x4) securing LG-2 mounted board so that removal of Stator Ass'y can be easily performed.

f. Remove the two screws (PSW2.6x5) securing the Stator Ass'y and then remove the Stator Ass'y and steel ball. (Refer to Fig. 4-10)

g. Wipe away the grease applied around lead screw with soft cloth before removal of Stator Ass'y and Rotor Ass'y, not to leave the grease in the chass'y hole during the removal.

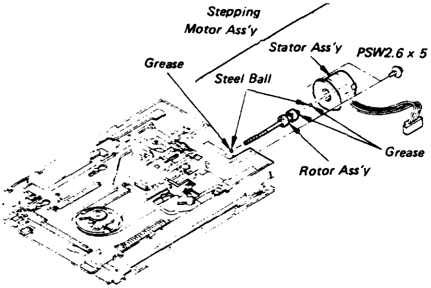


Fig. 4-10 Stepping Motor Ass'y (Stator Ass'y and Rotor Ass'y) Replacement

h. While twisting Rotor Ass'y, separate the Rotor Ass'y from the needle pin of head carriage ass'y.

i. Take the steel ball from the hole of chassis ass'y.

4-10-2 Installation

Note: The stepping motor must be replaced with the whole ass'y, since the wrong combination (Rotor Ass'y and Stator Ass'y) in the ass'y causes the mulfunction.

Note: Apply Molykote Grease (EM10L) (same quantity of match tip) on whole area of lead screw and two steel balls before the installation.

a. Insert the steel ball into hole of the chassis ass'y.

b. While lifting the head carriage ass'y a little, insert the lead screw of the Rotor Ass'y between the needle and plate spring of head carriage ass'y.

c. Insert the steel ball into hole of Rotor Ass'y.

d. Fasten loosely the Stator Ass'y with two screws (PSW2.6x5). But these screws must not be tightened for later adjustment.

e. Connect the connector CN102 (Rotor Ass'y) onto the LG-2 mounted board.

f. Install the cassette holder ass'y. (Refer to 4-6)

g. Install the front panel ass'y and eject button. (Refer to 4-2)

h. Perform the radial alignment and TRK00 sensor adjustment. (Refer to 5-1)

i. Install the cover ass'y. (Refer to 4-1)

j. Make the head clean. (Refer to 5-4)

6-6-2 Chip parts replacement procedure

This unit uses chip components such as carbon resistor, ceramic capacitor, transistor and diode in some circuits. It also uses IC's of flat-pack type. As the appearance of carbon resistor and ceramic capacitor are identical, destinguishment of each can be possible by visual check of reference address of silk-screen print on the printed circuit board. As the shape of transistor and diode are same, they also are distinguished by the reference address of silk-screen print.

Tools:

Soldering iron: 20W

(If possible, use soldering tip with heat-controller of 270±10°C)

Desoldering metal braid ("SOLDER TAUL" or equivalent)

Solder (of 0.6mm dia. is recommended.)

Tweezers

Soldering Conditions:

Tip temperature; 270±10°C

Solder within 2sec. per an electrode

Higher temperature or longer tip application than specified may be damaged to the chip component.

(1) Resistor and capacitor

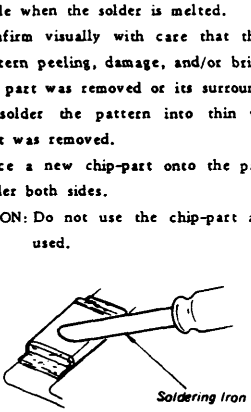
a. Add heat onto the chip-part by the top of soldering iron tip and slide the chip-part aside when the solder is melted.

b. Confirm visually with care that there is no pattern peeling, damage, and/or bridge where the part was removed or its surrounding.

c. Presolder the pattern into thin where the part was removed.

d. Place a new chip-part onto the pattern and solder both sides.

CAUTION: Do not use the chip-part again once used.



(2) Transistor and diode

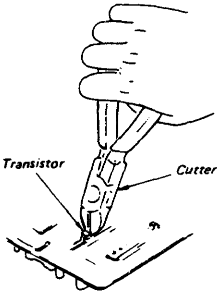
a. Cut the leads of the semiconductor part to be removed with a cutter.

b. Remove the each pin of semiconductor from the pattern by tweezers while heating the pin by soldering iron.

c. Confirm visually with care that there is no pattern peeling, damage, and/or bridge where the part was removed or its surrounding.

d. Presolder the pattern into thin where the part was removed.

e. Place a new semiconductor onto the pattern and solder the leads.

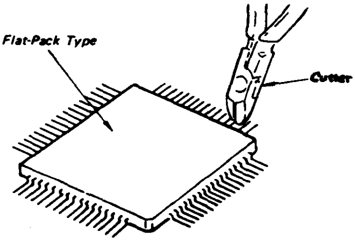


(3) IC (Flat-pack type)

a. Cut the leads of the IC to be removed with a cutter.

b. Remove the each pin of IC from the pattern by tweezers while heating the pin by soldering iron.

c. Confirm visually with care that there is no pattern peeling, damage, and/or bridge where the part was removed or its surrounding.



d. Presolder the pattern into thin where the part was removed.

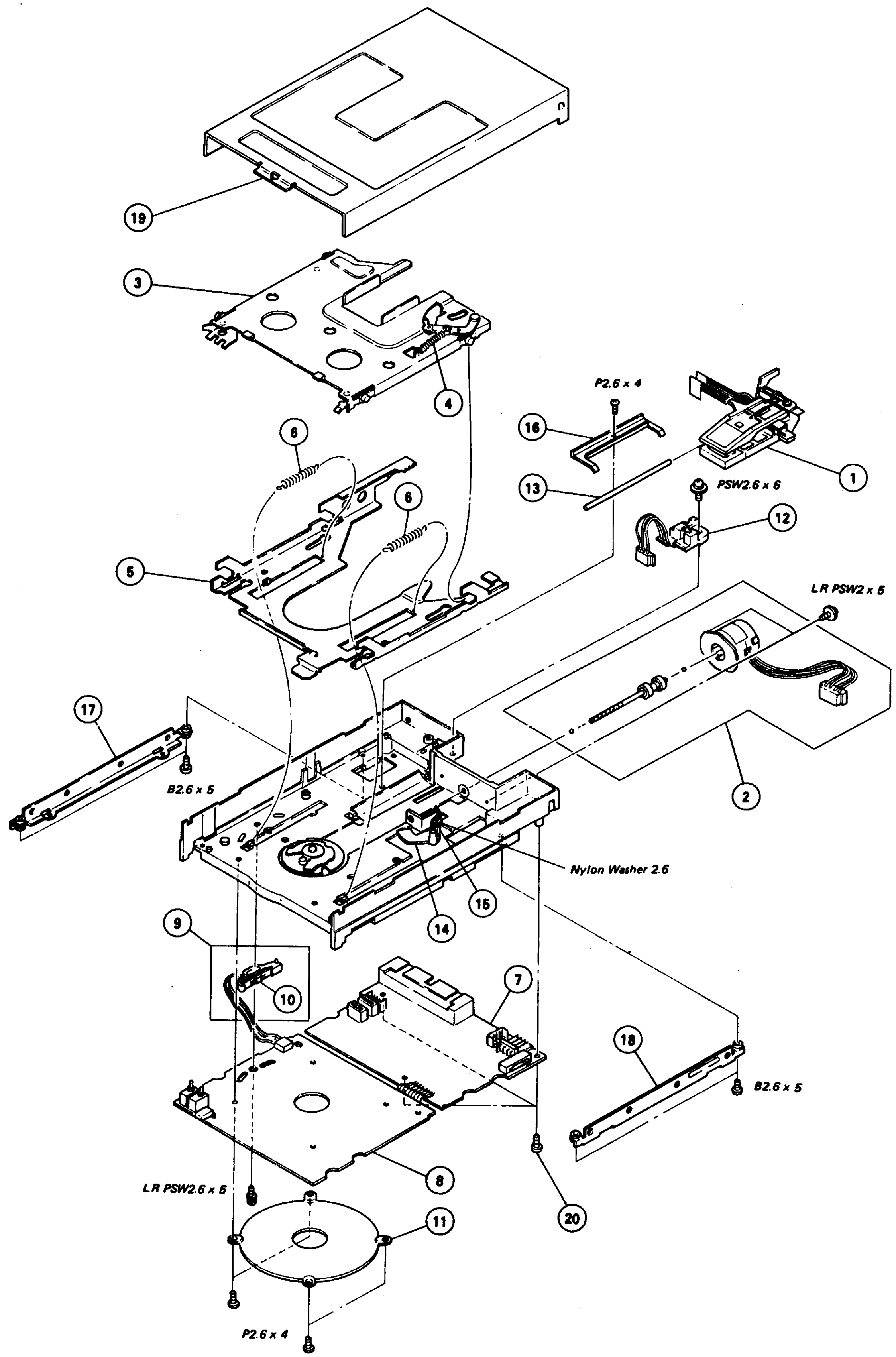
e. Place a new IC onto the pattern and solder it.

f. Confirm by a tester that each conduction between IC's terminal and copper pattern is surely made.

g. If not, resolder the portion.

SECTION 6
PARTS LOCATION AND LIST

6-1. PARTS ASS'Y LOCATION



6-2. MECHANICAL PARTS LIST

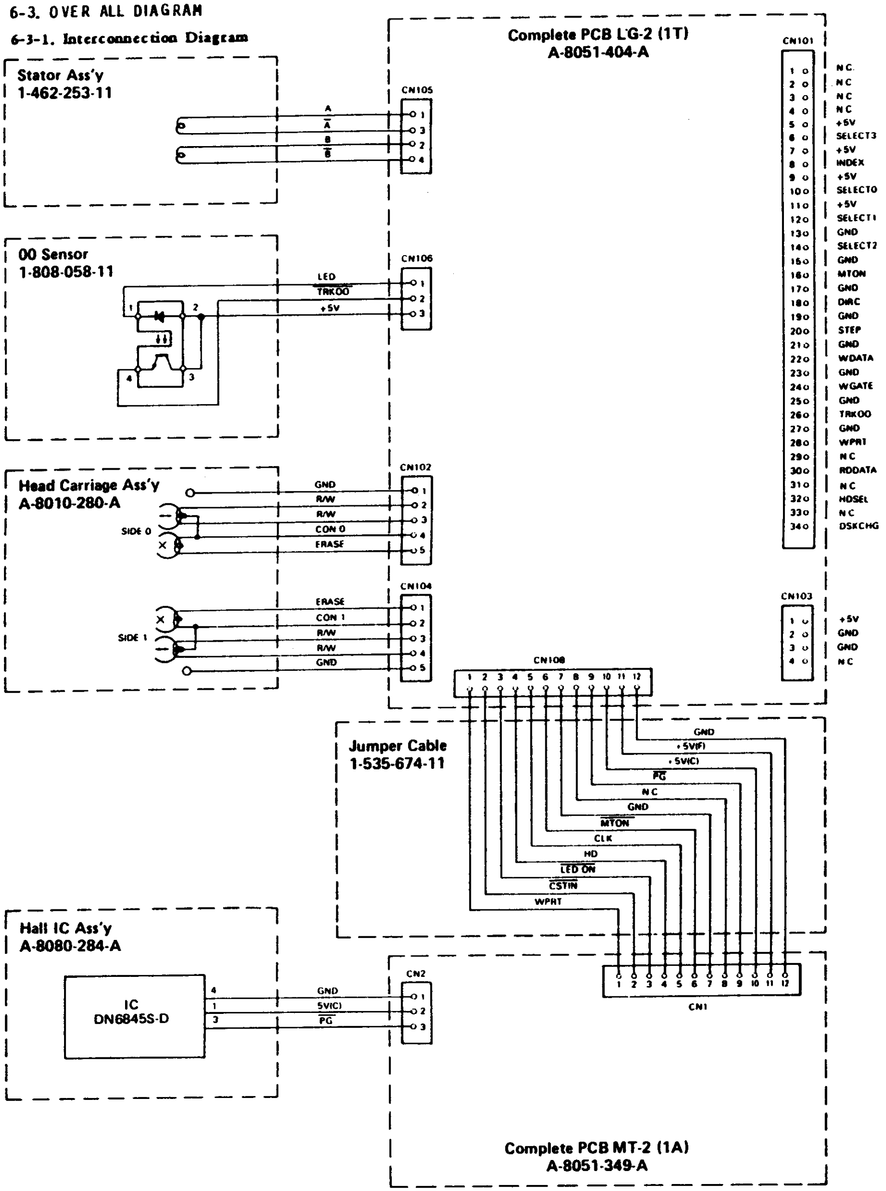
Note: 1. Parts printed in Bold-Face type are normally stocked for replacement purposes. The remaining parts shown in this list are not normally required for routine service work. Orders for parts not shown in Bold-Face type will be processed, but allow for additional delivery time.

2. The screws and washers may be supplied with the substitution that is similar to ones listed because of part-standardization program in SONY.

No.	Parts No.	Description
1	A-8010-280-A	Head Carriage Ass'y (OH)
2	A-8010-253-A	Stepping Motor Ass'y
3	A-8010-283-A	Cassette Holder Ass'y
4	4-613-114-01	Tension Spring
5	A-8010-234-A	Slide Plate Ass'y
6	4-613-132-01	Tension Spring
7	A-8051-404-A	LG-2 (1T) Mounted Board
8	A-8051-349-A	MT-2 (1A) Mounted Board
9	A-8080-284-A	Hall IC Ass'y
10	8-759-404-25	DM6845S-D
11	X-4613-102-1	Stator Yoke Ass'y
12	1-808-058-11	00 Sensor
13	4-606-001-11	Slide Guide Shaft
14	4-613-103-01	Trigger Arm
15	4-613-104-01	Spring
16	4-613-105-01	Guide Retainer
17	X-4613-105-1	Bracket Ass'y (Left)
18	X-4613-106-1	Bracket Ass'y (Riglt)
19	4-613-145-02	Cover
20	3-701-428-01	Special Screw +B 2.6x4
	4-613-717-01	Screw LR PSW 2x5
	4-614-322-01	Screw LR PSW 2.6x5
	7-621-259-25	Screw +P 2.6x4
	7-621-775-20	Screw +B 2.6x5
	7-621-759-45	Screw +PSW 2.6x6
	7-623-923-11	Nylone Washer 2.6

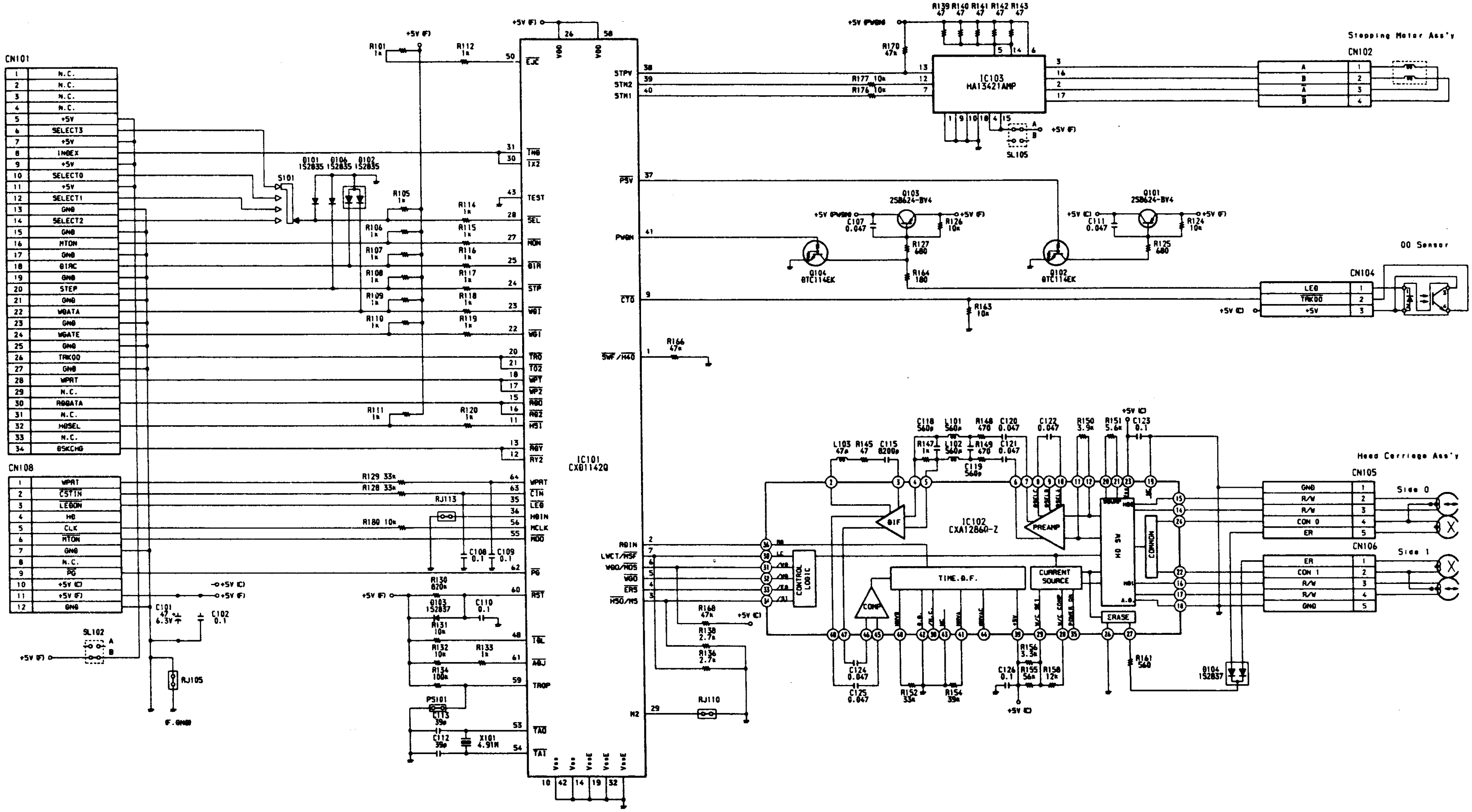
6-3. OVER ALL DIAGRAM

6-3-1. Interconnection Diagram

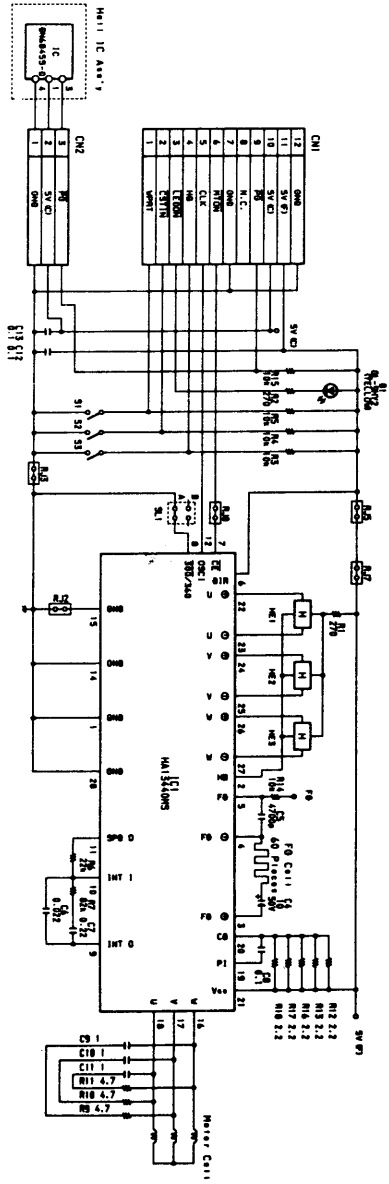


6-4. CIRCUIT DIAGRAM

6-4-1. Circuit Diagram on LG-2 (1T)



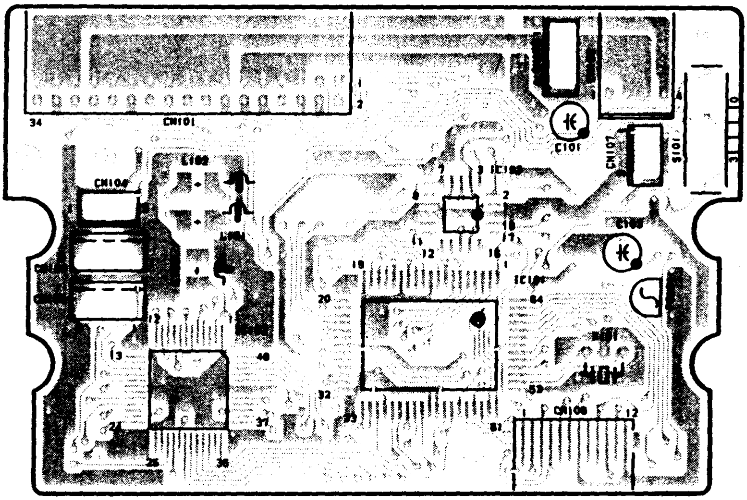
6-4-2. Circuit Diagram on MT-2 (7A)



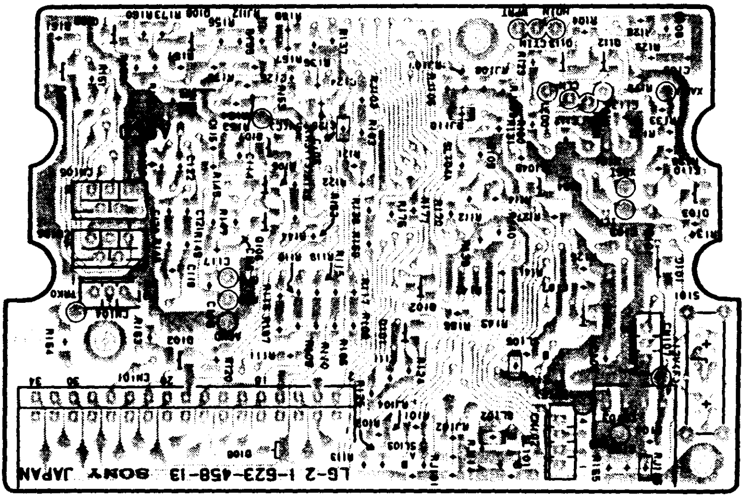
6-5. PARTS LAYOUT

6-5-1. Parts Layout on LG-2 (1T)

- Component Side -

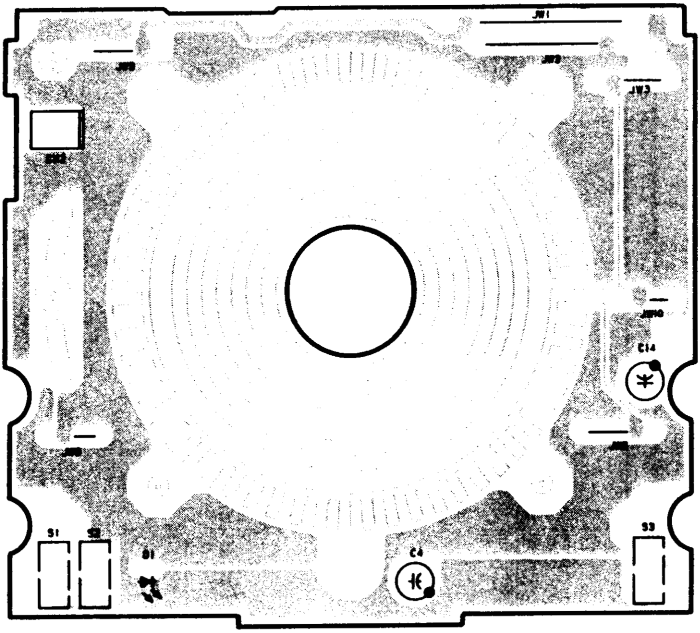


- Pattern Side -

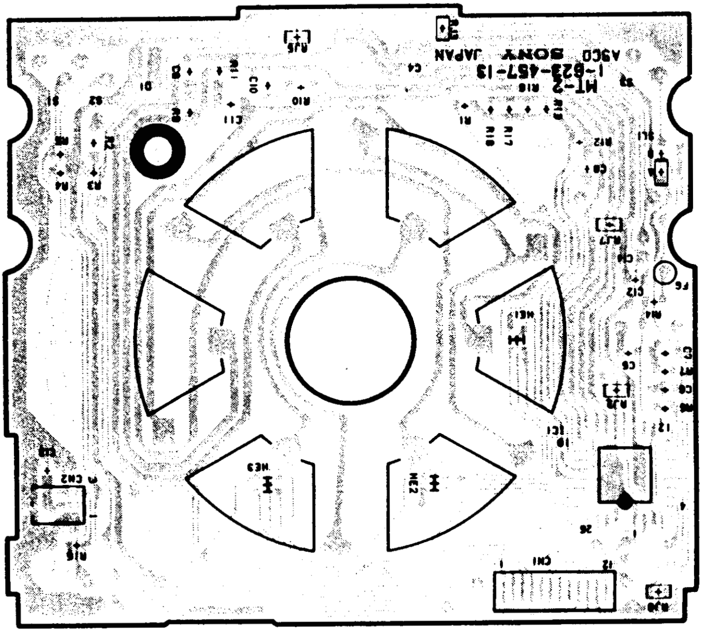


6-5-2. Parts Layout on MT-2 (1A)

- Component Side -



- Pattern Side -



6-6. ELECTRIC PARTS

6-6-1. ELECTRIC PARTS LIST

Note: 1. All capacitors are in micro farads unless otherwise specified.
2. All inductors are in micro henries unless otherwise specified.
3. All resistors are in ohms.
4. "CHIP" stands for chip component.

LG-2 (1T) MOUNTED BOARD

CAPACITORS

Ref. No.	Parts No.	Description			
C101	1-126-154-11	ELECT	47	20%	6.3V
C102	1-163-038-00	CERAMIC CHIP	0.1		25V
C107	1-163-809-11	CERAMIC CHIP	0.047	10%	25V
C108	1-163-038-00	CERAMIC CHIP	0.1		25V
C109	1-163-038-00	CERAMIC CHIP	0.1		25V
C110	1-163-038-00	CERAMIC CHIP	0.1		25V
C111	1-163-809-11	CERAMIC CHIP	0.047	10%	25V
C112	1-163-241-11	CERAMIC CHIP	39PF	5%	50V
C113	1-163-241-11	CERAMIC CHIP	39PF	5%	50V
C115	1-163-020-00	CERAMIC CHIP	0.0082	10%	50V
C118	1-163-135-00	CERAMIC CHIP	560PF	5%	50V
C119	1-163-135-00	CERAMIC CHIP	560PF	5%	50V
C120	1-163-809-11	CERAMIC CHIP	0.047	10%	25V
C121	1-163-809-11	CERAMIC CHIP	0.047	10%	25V
C122	1-163-809-11	CERAMIC CHIP	0.047	10%	25V
C123	1-163-038-00	CERAMIC CHIP	0.1		25V
C124	1-163-809-11	CERAMIC CHIP	0.047	10%	25V
C125	1-163-809-11	CERAMIC CHIP	0.047	10%	25V
C126	1-163-038-00	CERAMIC CHIP	0.1		25V

CONNECTORS

Ref. No.	Parts No.	Description
CN101	1-564-941-11	HEADER, CONNECTOR 34P
CN102	1-564-003-00	PIN, CONNECTOR 4P
CN104	1-564-002-00	PIN, CONNECTOR 3P
CN105	1-562-787-21	CONNECTOR, FLEXIBLE 5P
CN106	1-562-787-21	CONNECTOR, FLEXIBLE 5P
CN108	1-535-674-11	JUMPER CABLE

DIODES

Ref. No.	Parts No.	Description
D101	8-719-100-03	1S2835 (CHIP)
D102	8-719-100-03	1S2835 (CHIP)
D103	8-719-100-05	1S2837 (CHIP)
D104	8-719-100-05	1S2837 (CHIP)
D106	8-719-100-03	1S2835 (CHIP)

ICS

Ref. No.	Parts No.	Description
IC101	8-752-325-50	CXD1142Q
IC102	8-752-033-09	CXA1286Q-Z
IC103	8-759-305-19	HA13421AMP

COILS

Ref. No.	Parts No.	Description
L101	1-410-941-21	INDUCTOR CHIP 560
L102	1-410-941-21	INDUCTOR CHIP 560
L103	1-408-785-21	INDUCTOR CHIP 47

TRANSISTORS

Ref. No.	Parts No.	Description
Q101	8-729-162-44	2SB624-BV4 (CHIP)
Q102	8-729-900-53	DTC114EK (CHIP)
Q103	8-729-162-44	2SB624-BV4 (CHIP)
Q104	8-729-900-53	DTC114EK (CHIP)

RESISTORS

Ref. No.	Parts No.	Description			
R101	1-216-049-00	METAL CHIP	1K	5%	1/10W
R105	1-216-049-00	METAL CHIP	1K	5%	1/10W
R106	1-216-049-00	METAL CHIP	1K	5%	1/10W
R107	1-216-049-00	METAL CHIP	1K	5%	1/10W
R108	1-216-049-00	METAL CHIP	1K	5%	1/10W
R109	1-216-049-00	METAL CHIP	1K	5%	1/10W
R110	1-216-049-00	METAL CHIP	1K	5%	1/10W
R111	1-216-049-00	METAL CHIP	1K	5%	1/10W
R112	1-216-049-00	METAL CHIP	1K	5%	1/10W
R114	1-216-049-00	METAL CHIP	1K	5%	1/10W
R115	1-216-049-00	METAL CHIP	1K	5%	1/10W
R116	1-216-049-00	METAL CHIP	1K	5%	1/10W
R117	1-216-049-00	METAL CHIP	1K	5%	1/10W
R118	1-216-049-00	METAL CHIP	1K	5%	1/10W
R119	1-216-049-00	METAL CHIP	1K	5%	1/10W
R120	1-216-049-00	METAL CHIP	1K	5%	1/10W
R124	1-216-073-00	METAL CHIP	10K	5%	1/10W
R125	1-216-045-00	METAL CHIP	680	5%	1/10W
R126	1-216-073-00	METAL CHIP	10K	5%	1/10W
R127	1-216-045-00	METAL CHIP	680	5%	1/10W
R128	1-216-085-00	METAL CHIP	33K	5%	1/10W
R129	1-216-085-00	METAL CHIP	33K	5%	1/10W
R130	1-216-119-00	METAL CHIP	820K	5%	1/10W
R131	1-216-073-00	METAL CHIP	10K	5%	1/10W
R132	1-216-073-00	METAL CHIP	10K	5%	1/10W
R133	1-216-049-00	METAL CHIP	1K	5%	1/10W
R134	1-216-097-00	METAL CHIP	100K	5%	1/10W
R136	1-216-059-00	METAL CHIP	2.7K	5%	1/10W
R138	1-216-059-00	METAL CHIP	2.7K	5%	1/10W
R139	1-216-166-00	METAL CHIP	47	5%	1/8W
R140	1-216-166-00	METAL CHIP	47	5%	1/8W
R141	1-216-166-00	METAL CHIP	47	5%	1/8W
R142	1-216-166-00	METAL CHIP	47	5%	1/8W
R143	1-216-166-00	METAL CHIP	47	5%	1/8W
R145	1-216-017-00	METAL CHIP	47	5%	1/10W
R147	1-216-049-00	METAL CHIP	1K	5%	1/10W
R148	1-216-041-00	METAL CHIP	470	5%	1/10W
R149	1-216-041-00	METAL CHIP	470	5%	1/10W
R150	1-216-063-00	METAL CHIP	3.9K	5%	1/10W
R151	1-216-067-00	METAL CHIP	5.6K	5%	1/10W
R152	1-216-085-00	METAL CHIP	33K	5%	1/10W
R154	1-216-748-11	METAL CHIP	39K	1%	1/10W
R155	1-216-091-00	METAL CHIP	56K	5%	1/10W
R156	1-216-061-00	METAL CHIP	3.3K	5%	1/10W
R158	1-216-075-00	METAL CHIP	12K	5%	1/10W
R161	1-216-043-00	METAL CHIP	560	5%	1/10W
R163	1-216-073-00	METAL CHIP	10K	5%	1/10W
R164	1-216-180-00	METAL CHIP	180	5%	1/8W
R166	1-216-089-00	METAL CHIP	47K	5%	1/10W
R168	1-216-089-00	METAL CHIP	47K	5%	1/10W
R170	1-216-089-00	METAL CHIP	47K	5%	1/10W

Ref. No.	Parts No.	Description			
R176	1-216-073-00	METAL CHIP	10K	5%	1/10W
R177	1-216-073-00	METAL CHIP	10K	5%	1/10W
R180	1-216-073-00	METAL CHIP	10K	5%	1/10W
RJ105	1-216-295-00	METAL CHIP	0	5%	1/10W
RJ110	1-216-295-00	METAL CHIP	0	5%	1/10W
RJ113	1-216-295-00	METAL CHIP	0	5%	1/10W
SLA105	1-216-295-00	METAL CHIP	0	5%	1/10W
SLB102	1-216-295-00	METAL CHIP	0	5%	1/10W

SWITCH

S101	1-554-644-00	SWITCH, SLIDE

OSCILLATOR

X101	1-567-912-11	OSCILLATOR, CERAMIC (4.91M)

IC LINK

PS101	1-532-727-11	IC LINK

MT-2 (1A) MOUNTED BOARD

CAPACITORS

C4	1-124-261-00	ELECT	10	20%	50V
C5	1-163-017-00	CERAMIC CHIP	0.0047	10%	50V
C6	1-163-037-11	CERAMIC CHIP	0.022	10%	25V
C7	1-163-081-00	CERAMIC CHIP	0.22		25V
C8	1-163-038-00	CERAMIC CHIP	0.1		25V
C9	1-162-638-11	CERAMIC CHIP	1		16V
C10	1-162-638-11	CERAMIC CHIP	1		16V
C11	1-162-638-11	CERAMIC CHIP	1		16V
C12	1-163-038-00	CERAMIC CHIP	0.1		25V
C13	1-163-038-00	CERAMIC CHIP	0.1		25V

CONNECTOR

CN2	1-566-427-11	PIN, CONNECTOR 3P

DIODE

D1	8-719-904-92	GL-9HY2 (Yellow)

IC

IC1	8-759-305-25	HA13440MS

RESISTORS

R1	1-216-035-00	METAL CHIP	270	5%	1/10W
R2	1-216-184-00	METAL CHIP	270	5%	1/8W
R4	1-216-073-00	METAL CHIP	10K	5%	1/10W
R5	1-216-073-00	METAL CHIP	10K	5%	1/10W
R6	1-216-081-00	METAL CHIP	22K	5%	1/10W
R7	1-216-095-00	METAL CHIP	82K	5%	1/10W
R9	1-216-142-00	METAL CHIP	4.7	5%	1/8W
R10	1-216-142-00	METAL CHIP	4.7	5%	1/8W
R11	1-216-142-00	METAL CHIP	4.7	5%	1/8W
R12	1-216-298-00	METAL CHIP	2.2	5%	1/10W
R13	1-216-298-00	METAL CHIP	2.2	5%	1/10W
R14	1-216-073-00	METAL CHIP	10K	5%	1/10W
R15	1-216-073-00	METAL CHIP	10K	5%	1/10W
R16	1-216-298-00	METAL CHIP	2.2	5%	1/10W
R17	1-216-298-00	METAL CHIP	2.2	5%	1/10W
R18	1-216-298-00	METAL CHIP	2.2	5%	1/10W
RJ2	1-216-296-00	METAL CHIP	0	5%	1/8W

Ref. No.	Parts No.	Description			
RJ3	1-216-295-00	METAL CHIP	0	5%	1/10W
RJ5	1-216-295-00	METAL CHIP	0	5%	1/10W
RJ7	1-216-296-00	METAL CHIP	0	5%	1/8W
RJ8	1-216-295-00	METAL CHIP	0	5%	1/10W
SLA1	1-216-295-00	METAL CHIP	0	5%	1/10W

SWITCHES

S1	1-571-237-11	SWITCH, MICRO
S2	1-571-237-11	SWITCH, MICRO

Software

Software

Software Contents

BIOS Services

Device I/O Services
Introduction

The BIOS (Basic Input/Output System) is the lowest-level interface between other software (application programs and the operating system itself) and the hardware. The BIOS routines provide various device input/output services as well as bootstrap and print screen and other services. Some of the services that BIOS provides are not available through the operating system, such as the graphics routines.

All calls to the BIOS are made through software interrupts (that is, by means of assembly language "INT *x*" instructions). Each I/O device is provided with a software interrupt, which transfers execution to the routine.

Entry parameters to BIOS routines are normally passed in CPU registers. Similarly, exit parameters are generally returned from these routines to the caller in CPU registers. To insure BIOS compatibility with other machines, the register usage and conventions are, for the most part, identical.

The following pages describe the entry and exit requirements for each BIOS routine. To execute a BIOS call, load the registers as indicated under the "Entry Conditions" banner. (Register AH will contain the function number in cases where a single interrupt can perform more than one operation.) Then issue the interrupt given for the call. The following example can be used to read a character from the keyboard:

```
MOV AH,0
INT 16H
```

Upon return, AL contains the ASCII character and AH the keyboard scan code.

> **Note:** All registers except those used to return parameters to the caller are saved and restored by the BIOS routines.

Following is a quick reference list of software interrupts for all device I/O and system status services.

Service	**Software Interrupts**
Video Display	10 hex (16 dec)
Equipment	11 hex (17 dec)
Memory Size	12 hex (18 dec)
Diskette	13 hex (19 dec)
Serial Communications	14 hex (20 dec)
System Services	15 hex (21dec)
Keyboard	16 hex (22 dec)
Line Printer	17 hex (23 dec)
Bootstrap Loader	19 hex (25 dec)
System Clock	1Ahex (26 dec)

Keyboard

16 hex (22 dec)

Function Summary

AH	= 0:	Read keyboard (destructive with wait)
AH	= 1:	Scan keyboard (nondestructive, no wait)
AH	= 2:	Get current shift status
AH	= 5:	Store ASCII character and scan code in keyboard buffer
AH	= 10H:	Extended keyboard read
AH	= 11H:	Extended ASCII status
AH	= 12H:	Extended shift status read

Function Descriptions

Read Keyboard

Read the next character typed at the keyboard. Return the ASCII value of the character and the keyboard scan code, removing the entry from the keyboard buffer (destructive read).

Entry Conditions

AH = 0

Exit Conditions

AL = *ASCII value of character*

AH = *keyboard scan code*

Scan Keyboard

Set up the zero flag (Z flag) to indicate whether a character is available to read from the keyboard or not. If a character is available, return the ASCII value of the character and the keyboard scan code. The entry remains in the keyboard buffer (non-destructive read).

Entry Conditions

AH	=	1

Exit Conditions

Z	=	no character available
NZ	=	a character is available, in which case:
AL	=	*ASCII value of character*
AH	=	*keyboard scan code*

Get Shift Status

Return the current shift status.

Entry Conditions

AH	=	2

Exit Conditions

AL = *current shift status* (bit settings: set = true, reset = false)

Bit 0	=	RIGHT SHIFT key depressed
Bit 1	=	LEFT SHIFT key depressed
Bit 2	=	CTRL (control) key depressed
Bit 3	=	ALT (alternate mode) key depressed
Bit 4	=	SCROLL state active
Bit 5	=	NUMBER lock engaged
Bit 6	=	CAPS lock engaged
Bit 7	=	INSERT state active

Store ASCII Character

Entry Conditions

AH	=	5
CL	=	ASCII character
CH	=	Scan Code

Exit Conditions

AL	=	00: Successful
AL	=	01: Buffer full
[C]	=	Operation failed

Extended Keyboard Read

Entry Conditions

AH	=	10H

Exit Conditions

AL	=	*ASCII value of character*
AH	=	*keyboard scan code*

Extended ASCII Status

Entry Conditions

AH	=	11H

Exit Conditions

Z	=	No character is available
NZ	=	A character is available, in which case:
AL	=	*ASCII value of character*
AH	=	*keyboard scan code*

Extended Shift Status Read

Entry Conditions

AH = 12H

Exit Conditions

AL = *shift status* (bit settings: set = true, reset = false)

- Bit 7 = INSERT active
- Bit 6 = CAPS LOCK active
- Bit 5 = NUM LOCK active
- Bit 4 = SCROLL LOCK active
- Bit 3 = ALT pressed
- Bit 2 = CTRL pressed
- Bit 1 = LEFT SHIFT pressed
- Bit 0 = RIGHT SHIFT pressed

AH = *extended shift status* (bit settings: set = true, reset = false)

- Bit 7 = SYS REQ pressed
- Bit 6 = CAPS LOCK active
- Bit 5 = NUM LOCK active
- Bit 4 = SCROLL LOCK active
- Bit 3 = RIGHT ALT active
- Bit 2 = RIGHT CTRL active
- Bit 1 = LEFT ALT active
- Bit 0 = LEFT CTRL active

Video Display

These routines provide an interface for the video display - the output half of the console (CON) device. MS-DOS considers the video display to be the default standard output (STDOUT) device.

Software Interrupts

10 hex (16 dec)

Function Summary Table
Supported Video BIOS Calls

INT 10H

AH = 00	Set Video Mode
AH = 01	Set Cursor Type
AH = 02	Set Cursor Position
AH = 03	Read Cursor Position
AH = 05	Select Active Display Page
AH = 06	Scroll Active Page Up
AH = 07	Scroll Active Page Down
AH = 08	Read Attribute/Character at Current Cursor Position
AH = 09	Write Attribute/Character at Current Cursor Position
AH = 0A	Write Character Only at Current Cursor Position
AH = 0B	Set Color Palette
AH = 0C	Write Dot
AH = 0D	Read Dot
AH = 0E	Write TTY to Active Display
AH = 0F	Current Video State

INT 10H

AH=10	Color Palette Interface
AL=00	Set Individual Register
AL=01	Set Border Color
AL=02	Set All Palette Registers and Border
AH=13	Write String
AL=00	Write Character String
AL=01	Write Character String and Move Cursor
AL=02	Write Character and Attribute Strings
AL=03	Write Character and Attribute Strings and Move Cursor

Function Descriptions
Set CRT Mode

Entry Conditions

AH = 0

AL = *mode value*, as follows:

Alpha Modes

AL = 0: 40x25 black and white
AL = 1: 40x25 color
AL = 2: 80x25 black and white
AL = 3: 80x25 color

Graphics Modes

AL = 4: 320x200 color graphics
AL = 5: 320x200 black and white graphics with 4 shades
AL = 6: 640x200 black and white graphics with 2 shades
AL = 7: monochrome text

Additional Modes

AL = 8: 160x200 color graphics with 16 colors
AL = 9: 320x200 color graphics with 16 colors
AL = A: 640x200 color graphics with 4 colors

Note: If the high order bit of the AL register is 1, then the video buffer is not cleared.

Set Cursor Type

Set the cursor type and attribute.

Entry Conditions

AH = 1

CH = *bit values:*

Bits 5-6 = an invisible or erratically blinking cursor

Bits 5-6 = 0: produces a visible, blinking cursor

Bits 4-0 = *start line for cursor within character cell*

CL = *bit values:*

Bits 4-0 = *end line for cursor within character cell*

Set Cursor Position

Write (set) cursor position.

Entry Conditions

AH = 2

BH = *page number* (must be 0 for graphics modes)

DH = *row* (0 = top row)

DL = *column* (0 = leftmost column)

Get Cursor Position

Read (get) cursor position.

Entry Conditions

AH = 3

BH = *page number* (must be 0 for graphics modes)

Exit Conditions

DH = *row of current cursor position* (0 = top row)

DL = *column of current cursor position*
(0 = leftmost column)

CX = *cursor type currently set* [1]:

See previous "Set Cursor Type" (AH = 1).

Select Active Page

Select active display page (valid in alpha mode only).

Entry Conditions

AH = 5

AL = 0 through 7: *new page value for modes 0, 1*

AL = 0 through 3: *new page values for modes 2, 3*

AL = 80H: read CRT/CPU page registers

AL = 81H: set CPU page register to value in BL

AL = 82H: set CRT page register to value in BH

AL = 83H: set CRT and CPU page registers in BH and BL

Exit Conditions

If Bit 7 of AL = 1 upon entry, then:

BH = *contents of CRT page register*

BL = *contents of CPU page register*

Scroll Up

Scroll active page up.

Entry Conditions

AH	=	6
AL	=	*numbers of lines to scroll*. The number of lines that will be left blank at the bottom of the window. (0 = blank entire window)
CH	=	*row of upper left corner of scroll window*
CL	=	*column of upper left corner of scroll window*
DH	=	*row of lower right corner of scroll window*
DL	=	*column of lower right corner of scroll window*
BH	=	*attribute* (alpha modes) *or color* (graphics modes) *to be used on blank line*

Attributes

Color modes.

Foreground color:

Bit 0	=	blue
Bit 1	=	green
Bit 2	=	red
Bit 3	=	intensity
All bits off	=	black

Background color:

Bit 4	=	blue
Bit 5	=	green
Bit 6	=	red
Bit 7	=	blink
All bits off	=	white

Scroll Down

Scroll active page down.

Entry Conditions

AH = 7

AL = *number of lines to scroll* (0 = blank entire window)

CH = *row of upper left corner of scroll window*

CL = *column of upper left comer of scroll window*

DH = *row of lower right corner of scroll window*

DL = *column of lower right corner of scroll window*

BH = *attribute* (alpha modes) *of color* (graphics modes) *to be used on blank line.* See "Scroll Up" (AH = 6) for attribute values and "Set Color Palette" (AH = 11) for color values.

Read Attribute or Color/Character

Read a character and its attribute or color at the current cursor position.

Entry Conditions

AH = 8

BH = *display page number* (not used in graphics modes)

Exit Conditions

AL = *character read*

AH = *attribute of character* (alpha modes only)

Write Attribute or Color/Character

Write a character and its attribute or color at the current cursor position.

Entry Conditions

AH = 9

BH = *display page number* (not used in graphics modes)

CX = *number of characters to write*

AL = *character to write*

BL = *attribute of character* (for alpha modes) *or color of character* (for graphics modes. If Bit 7 of BL is set, the color of the character is XOR'ed with the color value). See “Scroll Up” (AH = 6) for attribute values and “Set Color Palette” (AH = 0BH) for color values.

Write Character Only

Write character only at current cursor position.

Entry Conditions

AH = 0AH

BH = *display page number* (valid for alpha modes only)

CX = *number of characters to write*

AL = *character to write*

BL = *color of character* (graphics mode)

Set Color Palette

Select the color palette.

Entry Conditions

AH = 0BH

BH = 0: Set background color (0-15) to color value in BL.

BL = *color value*:

1 = blue
2 = green
3 = cyan
4 = red
5 = magenta
6 = yellow
7 = light grey
8 = dark grey
9 = light blue
10 = light green
11 = light cyan
12 = light red
13 = light magenta
14 = yellow
15 = white

or

BH = 1: Set default palette to the number (0 or 1) in BL.

In black and white modes:

BL = 0: 1 for white

BL = 1: 1 for black

In 4 color graphics modes:

BL = 0: (1 = green, 2 = red, 3 = yellow)

BL = 1: (1 = cyan, 2 = magenta, 3 = white)

In 16 color graphics modes:

1 = blue
2 = green
3 = cyan
4 = red
5 = magenta
6 = yellow
7 = light grey
8 = dark grey
9 = light blue
10 = light green
11 = light cyan
12 = light red
13 = light magenta
14 = yellow
15 = white

Note: For alpha modes, Palette Entry 0 indicates the border color. For graphics modes, Palette Entry 0 indicates the border and the background color.

Write Dot

Write a pixel (dot).

Entry Conditions

AH = 0CH

DX = *row number*

CX = *column number*

AL = *color value* (When Bit 7 of AL is set, the resultant color value of the dot is the exclusive OR of the current dot color value and the value in AL.)

Read Dot

Read a pixel (dot).

Entry Conditions

AH = 0DH

DX = *row number*

CX = *column number*

Exit Conditions

AL = *color value of dot read*

Write TTY

Write a character in teletype fashion. (Control characters are interpreted in the normal manner.)

Entry Conditions

AH	=	0EH
AL	=	*character to write*
BL	=	*foreground color* (graphics mode)
BH	=	*display page* (alpha modes)

Get CRT Mode

Get the current video mode.

Entry Conditions

AH	=	0FH

Exit Conditions

AL	=	*current video mode*. See the previous "Set CTR Mode" (AH = 0) for values
AH	=	*number of columns on screen*
BH	=	*current active display page*

Set Palette Registers

Sets palette registers.

Entry Conditions

AH = 10H

AL = 0: Set Palette register

BL = *number of palette registe*r (0 -15) to set
BH = *color value to store*

AL = 1: Set border color register

BH = *color value to store*

AL = 2: Set palette color value to store and
border registers

ES:DX points to a 17-byte list.

Bytes 0-15 = *values for palette registers 0-15*
Byte 16 = *value for border register*

Write String

Display a string of characters on screen.

Entry Conditions

AH = 13H

ES:BP = *pointer to start of string*

CX = *length of string* (attributes do not count)

DX = *starting cursor position* (DH = row, DL = column)

BH = *page number* (for text modes)

BL = *attribute for characters* (graphics modes)

AL = 00: Characters only string, cursor not updated

= 01: Characters only string, cursor updated

= 02: Character, attribute alternating string, cursor not updated

= 03: Character, attribute alternating string, cursor updated

Control characters interpreted as w/function 0Eh.

Serial Communications

These routines provide asynchronous byte stream I/O from and to the RS-232C serial communications port. This device is labeled the auxiliary (AUX) I/O device in the device list maintained by MS-DOS.

Software Interrupts

14 hex (20 dec)

Function Summary

AH	= 0:	Reset Comm port
AH	= 1:	Transmit character
AH	= 2:	Receive character
AH	= 3:	Get current Comm status
DX	=	*communication port number* (0 or 1)

Function Descriptions

Reset Comm Port

Reset (or initialize) the communication port according to the parameters in AL, DL, and DH.

Entry Conditions

AH = 0

AL = *RS-232C parameters,* as follows:

DX = *port number* (0 or 1)

7 6 5	4 3	2	1 0
Baud Rate	*Parity*	*Stop Bits*	*Word Length*

000	= 110 baud	00	= none	0	= 1 bit	10	= 7 bits
001	= 150 baud	01	= odd	1	= 2 bits	11	= 8 bits
010	= 300 baud	11	= even				
011	= 600 baud						
100	= 1200 baud						
101	= 2400 baud						
110	= 4800 baud						
111	= 9600 baud						

Exit Conditions

AX = *RS-232C status*; See the following"Get Current Comm Status" (AH = 3)

Transmit Character

Transmit (output) the character in AL (which is preserved).

Entry Conditions

AH = 1

AL = *character to transmit*

DX = *port number* (0 or 1)

Exit Conditions

AH = *RS-232C status;* See the following "Get Current Comm Status" (AH = 3). If Bit 7 is set, the routine was unable to transmit the character because of a timeout error.)

AL is preserved

Receive Character

Receive (input) a character in AL (wait for a character, if necessary). On exit, AH will contain the RS-232 status, except that only the error bits (1, 2, 3, 4, 7) can be set; the timeout bit (7), if set, indicates that data set ready was not received and the bits in AH are not meaningful. Thus, AH is non-zero only when an error occurred.

Entry Conditions

AH = 2

DX = *port number* (0 or 1)

Exit Conditions

AL = *character received*

AH = *RS-232C status;* See the following "Get Current Comm Status" (AH = 3)

Get Current Comm Status

Read the communication status into AX.

Entry Conditions

AH = 3

DX = *port number* (0 or 1)

Exit Conditions

AH = *RS-232C status*, as follows (set = true):

Bit 0 = data ready
Bit 1 = overrun error
Bit 2 = parity error
Bit 3 = framing error
Bit 4 = break detect
Bit 5 = transmitter holding register empty
Bit 6 = transmitter shift register empty
Bit 7 = timeout occurred

AL = *modem status,* as follows (set = true):

Bit 0 = delta clear to send
Bit 1 = delta data set ready
Bit 2 = trailing edge ring detector
Bit 3 = delta receive line signal detect
Bit 4 = clear to send
Bit 5 = data set ready
Bit 6 = ring indicator
Bit 7 = receive line signal detect

Line Printer

These routines provide an interface to the parallel line printer. This device is labeled "PRN" in the device list maintained by the operating system.

Software Interrupts

17 hex (23 dec)

Function Summary

AH	= 0:	Print character
AH	= 1:	Reset printer port
AH	= 2:	Get current printer status

Function Descriptions
Print a Character

Entry Conditions

AH	= 0
AL	= *character to print*
DX	= *printer to be used* (0-2)

Exit Conditions

0AH = *printer status*. See the following "Get Current Printer Status" (AH = 2)

(If Bit 0 is set, the character could not be printed because of a timeout error.)

Reset Printer Port

Reset (or initialize) the printer port.

Entry Conditions

AH = 1

DX = *printer to be used* (0-2)

Exit Conditions

AH = *printer status;* See the following "Get Current Printer Status" (AH = 2)

Get Current Printer Status

Read the printer status into AH.

Entry Conditions

AH = 2

Exit Conditions

DX = *printer to be used* (0-2)

AH = *printer status as follows* (set = true):

Bit 0 = timeout occurred

Bit 1 = [unused]

Bit 2 = [unused]

Bit 3 = I/O error

Bit 4 = selected

Bit 5 = out of paper

Bit 6 = acknowledge

Bit 7 = not busy

System Clock

These routines provide methods of reading and setting the clock maintained by the system. This device is labeled CLOCK in the device list of the operating system. An interface for setting the multiplexer for audio source is also provided.

Software Interrupts

1A hex (26 dec)

Function Summary

AH	= 0:	Get time of day
AH	= 1:	Set time of day
AH	= 2:	Read real-time clock
AH	= 3:	Set real-time clock
AH	= 4:	Read date from real-time clock
AH	= 5:	Set the date in the real-time clock
AH	= 80H:	Set up sound multiplexer

The clock runs at the rate of 1,193,180/65,536 per second (about 18.2 times per second).

Function Descriptions

Get Time of Day

Get (read) the time of day in binary format.

Entry Conditions

AH = 0

Exit Conditions

CX = *high* (most significant) *portion of the clock count*

DX = *low* (least significant) *portion of the clock count*

AL = 0 of the clock was read or written (via AH = 0,1) within the current 24-hour period; otherwise, AL = 0

Set Time of Day

Set (write) the time of day using binary format.

Entry Conditions

AH = 1

CX = *high* (most significant) *portion of clock count*

DX = *low* (least significant) *portion of clock count*

Read Clock Time of Day

Read the time of day kept in the clock.

Entry Conditions

AH	=	2

Exit Conditions

CH	=	hours in BCD
CL	=	minutes in BCD
DH	=	seconds in BCD

Set Clock Time of Day

Set the time of day kept in the clock.

Entry Conditions

AH	=	3
CH	=	hours in BCD
CL	=	minutes in BCD
DH	=	seconds in BCD

Read Clock Date

Read the date kept in the clock.

Entry Conditions

AH	=	4

Exit Conditions

CH	=	century in BCD
CL	=	year in BCD
DH	=	month in BCD
DL	=	day in BCD

Set Clock Date

Set the date kept in the clock.

Entry Conditions

AH	=	5
CH	=	century in BCD
CL	=	year in BCD
DH	=	month in BCD
DL	=	day in BCD

Sound Multiplexer

Sets the multiplexer for audio source.

Entry Conditions

AH	=	80
AL	=	*source of sound*
00	=	8253 channel 2
02	=	audio in
03	=	complex sound generator chip

Disk I/O Support for Diskette Only

System Configuration

Software Interrupt

13 hex (19 dec)

Function Summary

AH	= 0:	Reset diskette
AH	= 1:	Return status of last diskette operation
AH	= 2:	Read sector(s) from diskette
AH	= 3:	Write sector(s) to diskette
AH	= 4:	Verify sector(s) on diskette
AH	= 5:	Format track on diskette
AH	= 08H:	Read drive parameters
AH	= 15H:	Read DASD type
AH	= 16H:	Diskette change line status

Function Descriptions

Reset Diskette

Reset the diskette system. Resets associated hardware and recalibrates all diskette drives.

Entry Conditions

AH = 0

Exit Conditions

See the following "Exits From All Calls."

Return Status of Last Diskette Operation

Returns the diskette status of the last operation in AH.

Entry Conditions

AH = 1

Exit Conditions

AL = *status of the last operation*. For values, see the following "Exits From All Calls."

Read Sector(s) from Diskette

Read the desired sector(s) from the diskette into RAM.

Entry Conditions

AH = 2

DL = *drive number* (0-2 if Tandy 1000 TL; 0-1 if Tandy 1000 SL)

DH = *head number* (0-1)

CH = *track number* (0-79)

CL = *sector number* (1-9)

AL = *sector count* (1-9)

ES:BX = *pointer to disk buffer*

Exit Conditions

See the following "Exits from all Calls."

AL = *number of sectors read*

Write Sector(s) to Diskette

Write the desired sector(s) from RAM to disk.

Entry Conditions

AH = 3

DL = *drive number* (0-2 if Tandy 1000 TL; 0-1 if Tandy 1000 SL)

DH = *head number* (0-1)

CH = *track number* (0-79)

CL = *sector number* (1-9)

AL = *sector count* (1-9)

ES:BX = *pointer to disk buffer*

Exit Conditions

See the following "Exits From All Calls."

AL = *number of sectors written*

Verify Sector(s) on Diskette

Verify the desired sector(s) are readable.

Entry Conditions

AH = 4
DL = *drive number* (0-2 if Tandy 1000 TL; 0-1 if Tandy 1000 SL)
DH = *head number* (0-1)
CH = *track number* (0-79)
CL = *sector number* (1-9)
AL = *sector count* (1-9)

Exit Conditions

See the following "Exits From All Calls."

AL = *number of sectors verified*

Format on Diskette

Format the desired track.

Entry Conditions

AH = 5
AL = *sector count* (1-9)
DL = *drive number* (0-2 if Tandy 1000 TL; 0-1 if Tandy 1000 SL)
DH = *head number* (0-1)
CH = *track number* (0-79)
CL = *sector number* (1-9)

ES:BX = *pointer to a group of address fields for each track.* Each address field is made up of 4 bytes. These are C, H, R, and N, where:

C = *track number*

H = *head number*

R = *sector number*

N = *the number of bytes per sector*
(00 = 128, 01 = 256, 02 = 512, 03 = 1024)

There is one entry for every sector on a given track.

Exit Conditions

See the following "Exits From All Calls."

Read Drive Parameters

Return the drive parameters.

Entry Conditions

AH = 08H

DL = *drive number* (0-2 if Tandy 1000 TL; 0-1 if Tandy 1000 SL)

Exit Conditions

AX = 0
BH = 0
CH = Maximum usable track number
CL = Maximum usable sector number
DH = Maximum usable head number
DL = Number of diskette drives installed (0-2 if Tandy 1000TL; 0-1 if Tandy 1000 SL)
ES:Dl = Pointer to diskette drive parameter table for the maximum media type supported on the specified drive

CF = 0: No error

CF = 1: Illegal parameter

Read DASD Type

Return the change line status.

Entry Conditions

AH = 15H

DL = *drive numbe*r (0-1 if Tandy 10000 TL; 0-1 if Tandy 1000 SL)

Exit Conditions

CF = 1: Operation was not successful. Previous versions of the Tandy 1000 will return CF = 1.

AH = 1: Invalid command.

CF = 0: Operation was successful

AH = 0: Drive not present

= 1: Diskette, no change line available

= 2: Diskette, change line available

Diskette Change Line Status

Return the status of the diskette change line.

Entry Conditions

AH = 16H

DL = *drive number* (0-2 if Tandy 1000 TL; 0-1 if Tandy 1000 SL)

Exit Conditions

CF = 0: If AH = 0
CF = 1: If AH is non 0
AH = 0: Diskette change signal not active
= 1: Invalid diskette parameter
= 6: Diskette change signal active
= 80: Diskette drive not ready (drive door is open)

Exits From All Calls

AH = *Status of operation*, where set = true

Error Code	Condition
01H	Illegal Function
02H	Address Mark Not Found
03H	Write Protect Error
04H	Sector Not Found
06H	Diskette Change Line Active
08H	DMA Overrun
09H	Attempt to DMA Across a 64K Boundary
10H	Bad CRC on Disk Read
20H	Controller Failure
40H	Seek Failure
80H	Device Timeout, Device Failed to Respond

[NC] = operation successful (AH = 0)
[C] = operation failed (AH = error status)

Equipment

This service returns the "equipment flag" (hardware configuration of the computer system) in the AX register.

Software Interrupts

11 hex (17 dec)

The "equipment flag" returned in the AX register has the following meanings for each bit:

Reset		=	the indicated equipment is not in the system
Set		=	the indicated equipment is in the system
Bit 0		=	diskette installed
Bit 1		=	math coprocessor
Bits 2,3	always = 11		
Bits 4,5	*initial video mode*		
	01		40x25 Color
	10		80x25 Color
	11		80x25 Monochrome
Bits 6,7	*number of diskette drives* (only if Bit 0 = 1)		
	00		1
	01		2
	10		3 (Tandy 1000 TL ONLY)
Bit 8	0	=	DMA present (**always present**)
	1	=	no DMA present
Bits 9, 10, 11			*number of RS232 cards*
Bit 12			game I/O adapter present (joystick)
Bit 13			not used
Bits 14,15			*number of printers*

Memory Size

This service returns the total number of kilobytes of RAM in the computer system (contiguous starting from Address 0) in the AX register. The maximum value returned is 640.

Software Interrupts

12 hex (18 dec)

Bootstrap Loader

Track 0, Sector 1 is read into Segment 0, Offset 7C00.

Control is then transferred as follows: (CS) = 0000H

(IP) = 7C00H

(DL) – drive where bootstrap sector was read

Software Interrupts

19 hex (25 dec)

System Services

Software Interrupts

15 hex (21dec)

Function Summary

AH = C0H: Machine identification

AH = 15H: Read and write EEPROM data

should be 7ØH, see below

Function Descriptions
Machine Identification

The machine identification algorithm is the same as all previous Tandy 1000's. As well, the Tandy 1000 SL and Tandy 1000 TL computers have a new BIOS call to further identify the machine.

All current and previous Tandy 1000 computers have the following machine identification:

Byte at address FFFF:E = FF hex (compatible with IBM PC)

Byte at address FC000:0 = 21 hex (Tandy 1000 unique)

Entry Conditions

AH	=	C0H

Exit Conditions

If CF	=	0
ES:BX	=	*pointer to machine identification data in ROM*
DW	0003	*Byte count of data that follows* (always 3)
DB	*xx*	*Model ID*
DB	*xx*	*Submodel ID*
DB	*xx*	*BIOS revision level*

IF CF = 1, the call is not supported (all previous versions of the Tandy 1000)

	Tandy 1000 SL	**Tandy 1000 TL**
Model ID	FF	FF
Submodel ID	00	01
BIOS revision level	*xx*	*xx*

Function Descriptions
Read From EEPROM

Read the 16-bit value from the indicated EEPROM word.

Entry Conditions

AH = 70H

AL = 0

BL = *word number to read* (0-63)

Exit Conditions

DX = *word value*

Carry flag set indicates EEPROM call not supported.

Write to EEPROM

Write a 16-bit value to the indicated EEPROM word.

Entry Conditions

AH = 70H

AL = 1

BL = *word number to write* (0-63)

DX = *word value to write*

Exit Conditions

Carry Flag set indicates EEPROM call not supported.

Get ROM Page

Get ROM page addressed at E000:0.

Entry Conditions:

AH = 70H

AL = 2

Exit Conditions:

AL = current ROM page (0-6)

Set ROM Page

Set ROM page addressed at E000:0.

Entry Conditions:

AH = 70H

AL = 3

DL = ROM page (0-6)

Exit Conditions:

ROM page set if carry clear. Carry set if invalid ROM page specified.

Tandy 1000 SL and Tandy 1000 TL BIOS Sound Support

The BIOS in these computers has the same support for sound as all previous Tandy 1000 computers, as well as support for additional sound features. The API for this new BIOS support is defined in the following information.

Software Interrupts

1A hex (26 dec)

Function Summary

AH	=	81H:	Get sound status
AH	=	82H:	Input sound (from the microphone)
AH	=	83H:	Output sound (to the speaker)
AH	=	84H:	Stop sound input and output

Function Descriptions

Get Sound Status

Gets sound status.

Entry Conditions

AH = 81H

Exit Conditions

Not Busy:

AX = 00C4H

CF = 0

Busy:

AX = 00C4H

CF = 1

Input Sound

Inputs sound from the microphone.

Entry Conditions

AH = 82H

ES:BX = *buffer address*

CX = *buffer length*

DX = *transfer rate* (1-4095, where 1 is the fastest transfer rate)

Exit Conditions

Not Busy:

AH = 0

CF = 0

Busy:

AH = 0

CF = 1

Output Sound

Outputs sound to the speaker.

Entry Conditions

AH = 83H: Output sound (to the speaker)

ES:BX = *buffer address*

CX = *buffer length*

DX = *transfer rate* (1-4095, where 1 is the fastest transfer rate)

AL = *volume* (0-7, where 0 = no sound)

Exit Conditions

Not Busy:

AH = 0

CF = 0

Busy:

AH = 0

CF = 1

For sample rate s samples/sec, the DX value is:

$$DX = \frac{3.57 \cdot 10^6}{s}$$

Ex: 8000=s ⇒ DX=1BEh

Stop Sound Input and Output

Stops sound input and output.

Entry Conditions

AH = 84H

Notes: The transfer rate values in register DX are not the same for calls AH = 82H and AH = 83H. To input a buffer of data with the AH = 82H call with a given DX value, then play it back with the AH = 83H call so that it sounds the same, set the DX value for output approximately 11.5 times as large as the DX value for input when run on a Tandy 1000 SL and approximately 10.0 times faster on a Tandy 1000 TL.

This BIOS call uses the DMA hardware to input and output the sound buffer. When functions AH = 82H and AH = 83H are called, the BIOS initiates the I/O and returns to the calling program immediately. When the DMA transfer is complete, the BIOS will receive a hardware interrupt and will execute a software INT 15H with AH = 91H and AL = FBH. If an application program needs to know when the data transfer is complete, it has to hook INT 15H and watch for this event.

The BIOS call masks the hardware restriction of not being able to DMA across a 64 kilobyte memory address boundary from the calling program.

Keyboard ASCII and Scan Codes
Function Keys, Cursor Keypad, Numeric Keypad

SCAN	NORM CASE		UPPER CASE		CTRL CASEALT CASE			
CODE	ASCII CODE		ASCII CODE		ASCII CODE		ASCII CODE	
3B	F1	x3B	F11	x54	F21	x5E	F31	x68
3C	F2	x3C	F12	x55	F22	x5F	F32	x69
3D	F3	x3D	F13	x56	F23	x60	F33	x6A
3E	F4	x3E	F14	x57	F24	x61	F34	x6B
3F	F5	x3F	F15	x58	F25	x62	F35	x6C
40	F6	x40	F16	x59	F26	x63	F36	x6D
41	F7	x41	F17	x5A	F27	x64	F37	x6E
42	F8	x42	F18	x5B	F28	x65	F38	x6F
43	F9	x43	F19	x5C	F29	x66	F39	x70
44	F10	x44	F20	x5D	F30	x67	F40	x71
57	F11	e8500		e8700		e8900		e8B00
58	F12	e8600		e8800		e8A00		e8C00
E037	PrintScrn*		PrintScrn*		CPrSc	x72	SysRq*	--
46	Scr Lock	-	Scr Lock-		--	--	Scr Lock	--
E145	Pause*	--	Pause*	--			Pause*	--
E046					Break*	x00		
E052	Insert		x52	e52E0	--	e92E0	--	eA200
E047	Home		x47	e47E0	x77	e77E0	--	e9700
E049	Page Up		x49	e47E0	x84	e84E0	--	e9900
E053	Delete		x53	e53E0	--	e93E0	--	eA300
E04F	End		x4F	e4FE0	x75	e75E0	--	e9F00
E051	Page Down		x51	e51E0	x76	e76E0	--	eA100
E048	Up		x48	e48E0	--	e8DE0	--	e9800
E04B	Left		x4B	e4BE0	x73	e73E0	--	e9B00
E050	Down		x50	e50E0	--	e91E0	--	eA000
E04D	Right		x4d	e4DE0	x74	e74E0	--	e9D00
45	Num Lock--		Num Lock	--	--	--	Num Lock	--
E035	/	2F	/	2F		e9500		eA400
37	*	2A	*	2A		e9600		e37F0
47	Home	x47	7	37	ClrSc	x77	¥	--
48	UP	x48	8	38		e8D00	¥	--
49	Page Up	x49	9	39	TOS	x84	¥	--
4A	-	2D	-	2D		e8E00		e4AF0
4B	Left	x4B	4	34	LWord	x73	¥	--
4C		e4CF0	5	35		e8F00	¥	--
4D	RIGHT	x4D	6	36	RWord	x74	¥	--
4E	+	2B	+	2B		e9000		e4EF0
4F	End	x4F	1	31	ErEOL	x75	¥	--
50	DOWN	x50	2	32		e9100	¥	--
51	Pg Dn	x51	3	33	ErEOS	x76	¥	--
52	Ins	x52	0	30		e9200	¥	--
53	Del	x53	.	2E		e9300	--	--
E01C	Enter	0D	Enter	0D	LF	0A		eA600
01	ESC	1B	ESC	1B	ESC	1B		e01F0
02	1	31	!	21	--	--	ALT1	x78
03	2	32	@	40	NULL	00	ALT2	x79
04	3	33	#	23	--	--	ALT3	x7A
05	4	34	$	24	--	--	ALT4	x7B
06	5	35	%	25	--	--	ALT5	x7C
07	6	36	^	5E	RS	1E	ALT6	x7D
08	7	37	&	26	--	--	ALT7	x7E
09	8	38	*	2A	--	--	ALT8	x7F

SCAN	NORM CASE		UPPER CASE		CTRL CASEALT CASE			
CODE	ASCII CODE		ASCII CODE		ASCII CODE		ASCII CODE	
0A	9	39	(	28	--	--	ALT9	x80
OB	-	30	)	29	--	--	ALT0	x81
0C	-	2D	_	5F	US	1F	ALT-	x82
0D	=	3D	+	2B	--	--	ALT =	x83
0E	BS	08	BS	08	DEL	7F		e0EF0
0F	HT	09	BTab	x0F		e9400		eA500
10	q	71	Q	51	DC1	11	ALTQ	x10
11	w	77	W	57	ETB	17	ALTW	x11
12	e	65	E	45	ENQ	05	ALTE	x12
13	r	72	R	52	DC2	12	ALTR	x13
14	t	74	T	54	DC4	14	ALTT	x14
15	y	79	Y	59	EM	19	ALTY	x15
16	u	75	U	55	NAK	15	ALTU	x16
17	i	69	I	49	HT	09	ALTI	x17
18	o	6F	O	4F	SI	0F	ALTO	x18
19	p	70	P	50	DLE	10	ALTP	x19
1A	[	5B	{	7B	ESC	1B		e1AF0
1B	]	5D	}	7D	GS	1D		e1BF0
1C	Enter	0D	Enter	0D	LF	0A		e1CF0
1D	Ctrl	--	Ctrl	--	Ctrl	--	Ctrl	--
E01D	Ctrl	--	Ctrl	--	Ctrl	--	Ctrl	--
1E	a	61	A	41	SOH	01	ALTA	x1E
1F	s	73	S	53	DC3	13	ALTS	x1F
20	d	64	D	44	EOT	04	ALTD	x20
21	f	66	F	46	ACK	06	ALTF	x21
22	g	67	G	47	BEL	07	ALTG	x22
23	h	68	H	48	BS	08	ALTH	x23
24	j	6A	J	4A	LF	0A	ALTJ	x24
25	k	6B	K	4B	VT	0B	ALTK	x25
26	l	6C	L	4C	FF	0C	ALTL	x26
27	;	3B	:	3A	--	--		e27FO
28	'	27	"	22	--	--		e28F0
29	‘	60	~	7E	--	--		e29F0
2A	LShift	--	LShift	--			LShift	--
2B	\	5C	|	7C	FS	1C		e2BF0
2C	z	7A	Z	5A	SUB	1A	ALTZ	x2C
2D	x	78	X	58	CAN	18	ALTX	x2D
2E	c	63	C	43	ETX	03	ALTC	x2E
2F	v	76	V	56	SYN	16	ALTV	x2F
30	b	62	B	42	STX	02	ALTB	x30
31	n	6E	N	4E	S0	0E	ALTN	x31
32	m	6D	M	4D	CR	0D	ALTM	x32
33	,	2C	<	3C	--	--		e33F0
34	.	2E	>	3E	--	--		e34F0
35	/	2F	?	3F	--	--		e35F0
36	RShift	--	RShift	--	RShift	--	RShift	--
38	Alt	--	Alt	--	Alt	--	Alt	--
EO38	Alt	--	Alt	--	Alt	--	Alt	--
39	SPACE	20	SPACE	20	SPACE	20	SPACE	20
3A	CapsLock--		CapsLock	--	--	--	CapsLock	--
56+	\	5C	|	7C	--	--	--	--

Keyboard Tables

These symbols have special meanings in the following tables:

-- Indicates that no ASCII code is generated for the key combination.

x Values preceded by x are extended ASCII codes. The keyboard driver returns a NULL ASCII code and the number in the table for the scan code.

e Values preceded by e are produced when you are using an enhanced BIOS. When using the BIOS Read Key function, these keys are either discarded or translated to a value compatible with older computers. When using the Enhanced Read Key function, AH = 10H, INT 16H, these keys are returned to your program.

+ A + in the scan code field denotes the extra key on the international version of the enhanced keyboard. This key is not available on the standard USA enhanced keyboard.

¥ The ALT key provides a way to generate the ASCII code of the decimal numbers in the range 1 to 255. Hold down the ALT key while typing a number in that range on the numeric keypad. When you release the ALT key, the character of the ASCII code you typed is generated and displayed.

BREAK Empties the keyboard queue and executes the keyboard break interrupt (INT 1BH). Places a NULL ASCII scan code in the keyboard queue.

PAUSE Delays system activity until you press another key.

PrtSC or Print Scrn Invokes the BIOS print screen function (INT 5H).

CPrSc Tells MS-DOS to direct console output to both the printer and the console. A second CPrSc halts printer output.

SysRq Interrupts the current process and allows another program to take control, if supported. When the SysRq key is pressed, INT 15H is invoked with AX=8500H. When the key is released, INT 15H is invoked with AX=8501H.

Reset Restarts your computer.

MS-DOS Memory Map

Hexadecimal Starting Address (Segment:Offset)	**Description**
000:00	BIOS Interrupt Vectors
000:80	Available Interrupt Vectors
0040:00[1]	ROM BIOS Data Area
0050:00	MSDOS and BASIC Data Area
0070:00	I/O.SYS Drivers
0190:00[2]	MS-DOS
05B0:00[2]	Available to user
X800:00[3]	Video RAM in 32K video modes
XC00:00[3]	Video RAM in 16K video mode
B800:00[4]	Video RAM Window (32K)
E000:00	ROM Drive
F000:00	Reserved for system ROM
FC00:00	System BIOS ROM

Notes:

[1] Detailed description in following pages.

[2] Approximate address; subject to change.

[3] X is defined as follows:

Memory Size	**X Value**
128K	1
256K	3
384K	5
512K	7
640K	9
768K	B

[4] Video memory accessed through the B800:0 window for all video modes.

ROM BIOS Data Area

The following table gives the starting offset, and length of each BIOS device driver. This area is located at segment 40:00.

Comm card address	0000	8 (1 word per card)
Printer addresses	0008	8 (1 word per printer)
Devices installed	0010	2 (16 bits)
Not used	0012	1
Memory size	0013	2 (1 word)
I/O channel RAM size	0015	2 (1 word)
KBD data area	0017	39
Disk data area	003E	11
Video data area	0049	30
Not used	0067	5
Clock data area	006C	5
KBD Break & Reset flags	0071	3
Not used	0074	4
Printer timeout counter	0078	4 (1 byte per printer)
Comm timeout counter	007C	4 (1 byte per card)
KBD extra data area	0080	4 (2 words)

The structure and usage of the Video driver RAM data area is as follows:

HEX Offset **From Segment** **0040:0000**	**Length and** **Intended Use**	
49H	1 byte -	current CRT mode (0-7)
4AH	1 word -	screen column width
4CH	1 word -	byte length of screen
4EH	1 word -	address/offset of beginning of current display page
50H	8 words-	row/col coordinates of the cursor for each of up to 8 display pages
60H	1 word -	current cursor type (See "Set Cursor Type" for correct encoding) typo, insert "63H" here
62H	1 byte -	current display page 1 word - base address + 4 of the CRT controller card
65H	1 byte -	copy of value written to the Mode Select Register
66H	1 byte -	current color palette setting

The equipment check BIOS call (INT 11H) and memory size BIOS call (INT 12H) return information from the following data areas:

HEX Offset **From Segment** **0040:0000**	**Length and** **Intended Use**
10H	Devices installed word
13H	Memory installed word

The structure and usage of the diskette driver RAM data area is as follows:

HEX Offset **From Segment** **0040:0000**	**Length and** **Intended Use**	
3EH	1 byte -	drive recalibration status - bit 3-0, if 0 then drive 3-0 needs recalibration before next Seek. Bit 7 indicates interrupt occurrence
3FH	1 byte -	motor status - Bit 3-0 drive 3-0 motor is on/off. Bit 7 - current operation is write, requires delay
40H	1 byte -	motor turn off timeout counter (see Timer ISR)
41H	1 byte -	disk status - codes are defined as in this section
42H	7 bytes -	7 bytes of status returned by the controller during result phase of operation

Value	Error Condition
01H	Illegal Function
02H	Address Mark Not Found
03H	Write Protect Error
04H	Sector Not Found
06H	Diskette Change Line Active
08H	DMA Overrun
09H	Attempt to DMA Across a 64K Boundary
10H	Bad CRC on Disk Read
20H	Controller Failure
40H	Seek Failure
80H	Device Timeout, Device Failed to Respond

The structure and usage of the RS232 driver RAM data area is as follows:

HEX Offset From Segment 0040:0000	Length and Intended Use	
00H	4 words -	Base address of each one of 4 possible comm cards
7CH	4 words -	1 word timeout count for each of 4 possible comm cards

The structure and usage of the Keyboard driver RAM data area is as follows:

HEX Offset		
From Segment	Length and	
0040:0010	Intended Use	
17	1 byte-	Keyboard shift state flag returned by function 02
	Bits 7-	INSERT state active, 6 - CAPS LOCK on/off, 5 - NUM LOCK on/off, 4 - SCROLL LOCK on/off, 3 - ALT key pressed 2 - CTRL key pressed 1 - Left SHIFT key pressed, 0 - Right SHIFT key pressed,
18	1byte-	Secondary shift state flag,
	Bits	INSERT key pressed, 6 - CAPS LOCK pressed, 5 - NUM LOCK pressed, 4 - SCROLL LOCK NUM LOCK pressed, 4 - SCROLL pressed, 4 - SCROLL LOCK pressed, 3 - Pause on/off, pressed, 3 - Pause on/off, 2,1,0 - not used
19	1 byte-	Used to store ALT keypad entry
1A	1 word-	Pointer to beginning of the keyboard buffer
1C	1 word-	Pointer to end of the keyboard buffer
1E	16-	Keyboard buffer (enough for words)
	15-	Type ahead entries

The structure and usage of the clock service routine is as follows:

HEX Offset From Segment 0040:0000	**Length and Intended Use**	
6CH	1 word -	Least significant 16 bits of clock count
6EH	1 word -	Most significant 16 bits of clock count
70H	1 byte -	Twentyfour hour rollover flag

Additional Data Area

HEX Offset
From Segment
0040:0000

B0H	2 words international support				
B4H	1 byte		0	=	No monochrome monitor
			FFH	=	Monochrome monitor
B5H	1 byte	Bit 0:	0	=	Drive A is 5-1/4"
			1	=	Drive A is 3-1/2"
		Bit 1:	0	=	Drive B is 5-1/4"
			1	=	Drive B is 3-1/2"
		Bit 2:	0	=	Tandy 1000 keyboard layout
			1	=	IBM keyboard layout
		Bit 3:	0	=	Slow CPU speed mode
			1	=	Fast CPU speed mode
		Bit 4:	0	=	Internal color video support enabled
			1	=	Internal color video support disabled, external color video enabled

HEX Offset **From Segment** **0040:0000**				
		Bit 5:	0	= No external monochrome video installed
			1	= External monochrome video installed
B6H	1 byte	Bit 0:	0	= Drive C is 5-1/4"
			1	= Drive C is 3-1/2"
40:C2	1 byte		01	= ROM drive is A:
			02	= ROM drive is B:
			03	= ROM drive is C:
			04	= ROM drive is D:

this column returned by INT 15H function 4FH

KEYBOARD SCAN CODES

Key	Key	Hardware Kybrd		Kybrd Interrupt		Standard ASCII (Scancode/ASCII code)				Extended ASCII (Scancode/ASCII code)			
#	Descript.	Make Code	Break Code	Make Code	Break Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
1	Esc	76	F076	01	81	011B	011B	011B	-----	011B	011B	011B	0100
2	F1	05	F005	3B	BB	3B00	5400	5E00	6800	3B00	5400	5E00	6800
3	F2	06	F006	3C	BC	3C00	5500	5F00	6900	3C00	5500	5F00	6900
4	F3	04	F004	3D	BD	3D00	5600	6000	6A00	3D00	5600	6000	6A00
5	F4	0C	F00C	3E	BE	3E00	5700	6100	6B00	3E00	5700	6100	6B00
6	F5	03	F003	3F	BF	3F00	5800	6200	6C00	3F00	5800	6200	6C00
7	F6	0B	F00B	40	C0	4000	5900	6300	6D00	4000	5900	6300	6D00
8	F7	83	F083	41	C1	4100	5A00	6400	6E00	4100	5A00	6400	6E00
9	F8	0A	F00A	42	C2	4200	5B00	6500	6F00	4200	5B00	6500	6F00
10	F9	01	F001	43	C3	4300	5C00	6600	7000	4300	5C00	6600	7000
11	F10	09	F009	44	C4	4400	5D00	6700	7100	4400	5D00	6700	7100
12	F11	78	F078	57	D7	-----	-----	-----	-----	8500	8700	8900	8B00
13	F12	07	F007	58	D8	-----	-----	-----	-----	8600	8800	8A00	8C00
14	Print Scrn	E07C	E0F07C	E02AE037	E0B7E0AA	Note[1]	Note[1]	7200	-----	Note[1]	Note[1]	7200	-----
15	Scroll Lock	7E	F07E	46	C6	Note[2]	Note[2]	------	Note[2]	Note[2]	Note[2]	-----	Note[2]
16	Pause Break	E11477	E1F014F077	E11D45	E19DC5	Note[3]	Note[3]	Note[4]	Note[3]	Note[3]	Note[3]	Note[4]	Note[3]
17	~ or \	0E	F00E	2B	AB	2960	297E	-----	-----	6000	7E00	-----	2900
18	! or 1	16	F016	02	82	0231	0221	-----	7800	0231	0221	-----	7800

When >1 shift key down: <alt> overrides <cntrl>, <cntrl> overrides <shift>, <alt> overrides <shift>.

this column returned by INT 16H function 00H

this column returned by INT 16H function 10H

KEYBOARD SCAN CODES

Key	Key	Hardware Kybrd		Kybrd Interrupt		Standard ASCII				Extended ASCII			
						(Scancode/ASCII code)				(Scancode/ASCII code)			
#	Descript.	Make Code	Break Code	Make Code	Break Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
19	@ or 2	1E	F01E	03	83	0332	0340	0300	7900	0332	0340	0300	7900
20	# or 3	26	F026	04	84	0433	0423	-----	7A00	0433	0423	-----	7A00
21	$ or 4	25	F025	05	85	0534	0524	-----	7B00	0534	0524	-----	7B00
22	% or 5	2E	F02E	06	86	0635	0625	-----	7C00	0635	0625	-----	7C00
23	^ or 6	36	F036	07	87	0736	075E	071E	7D00	0736	075E	071E	7D00
24	& or 7	3D	F03D	08	88	0837	0826	-----	7E00	0837	0826	-----	7E00
25	* or 8	3E	F03E	09	89	0938	092A	-----	7F00	0938	092A	-----	7F00
26	(or 9	46	F046	0A	8A	0A39	0A28	-----	8000	0A39	0A28	-----	8000
27	) or 0	45	F045	0B	8B	0B34	0B29	-----	8100	0B34	0B29	-----	8100
28	_ or -	4E	F04E	0C	8C	0C2D	0C5F	0C1F	8200	0C2D	0C5F	0C1F	8200
29	+ or =	55	F055	0D	8D	0D3D	0D2B	-----	8300	0D3D	0D2B	-----	8300
30	Backspace	66	F066	0E	8E	0E08	0E08	0E7F	-----	0E08	0E08	0E7F	0E00
31	Insert	E070	E0F070	E02AE052	E0D2E0AA	5200	5200	-----	-----	52E0	52E0	92E0	A200
32	Home	E06C	E0F06C	E02AE047	E0C7E0AA	4700	4700	7700	-----	47E0	47E0	77E0	9700
33	Pg Up	E07D	E0F07D	E02AE049	E0C9E0AA	4900	4900	8400	-----	49E0	49E0	84E0	9900
34	Num Lock	77	F077	45	C5	Note[5]	Note[5]	-----	Note[5]	Note[5]	Note[5]	----	Note[5]
35	/	E04A	E0F04A	E035	E0B5	352F	352F	-----	-----	E02F	E02F	9500	A400 3 nonstandard?
36	*	7C	F07C	37	B7	372A	372A	-----	-----	372A	372A	9600	3700
37	-	7B	F07B	4A	CA	4A2D	4A2D	-----	-----	4A2D	4A2D	8E00	4A00

KEYBOARD SCAN CODES

Key	Key	Hardware Kybrd		Kybrd Interrupt		Standard ASCII				Extended ASCII			
#	Descript.	Make	Break	Make	Break	(Scancode/ASCII code)				(Scancode/ASCII code)			
		Code	Code	Code	Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
38	Tab	0D	F00D	0F	8F	0F90	0F00	-----	-----	0F09	0F00	9400	A500
39	Q or q	15	F015	10	90	1071	1051	1011	1000	1071	1051	1011	1000
40	W or w	1D	F01D	11	91	1177	1157	1117	1100	1177	1157	1117	1100
41	E or e	24	F024	12	92	1265	1245	1205	1200	1265	1245	1205	1200
42	R or r	2D	F02D	13	93	1372	1352	1312	1300	1372	1352	1312	1300
43	T or t	2C	F02C	14	94	1474	1454	1414	1400	1474	1454	1414	1400
44	Y or y	35	F035	15	95	1579	1559	1519	1500	1579	1559	1519	1500
45	U or u	3C	F03C	16	96	1675	1655	1615	1600	1675	1655	1615	1600
46	I or i	43	F043	17	97	1769	1749	1709	1700	1769	1749	1709	1700
47	O or o	44	F044	18	98	186F	184F	180F	1800	186F	184F	180F	1800
48	P or p	4D	F04D	19	99	1970	1950	1910	1900	1970	1950	1910	1900
49	{ or [	54	D054	1A	9A	1A5B	1A7B	1A1B	-----	1A5B	1A7B	1A1B	1A00
50	} or]	5B	F05B	1B	9B	1B5D	1B7D	1B1D	-----	1B5D	1B7D	1B1D	1B00
51	| or \	5D	F05D	2B	AB	2B5C	2B7C	2B1C	-----	2B5C	2B7C	2B1C	2B00
52	Delete	E071	E0F071	E02AE053	E0D3E0AA	5300	5300	-----	-----	53E0	53E0	93E0	A300
53	End	E069	E0F069	E02AE04F	E0CFE0AA	4F00	4F00	7500	-----	4FE0	4FE0	75E0	9F00
54	Page Down	E07A	E0F07A	E02AE051	E0D1E0AA	5100	5100	7600	-----	51E0	51E0	76E0	A100
55	7 or Home	6C	F06C	47	C7	4700	4737	7700	Note[6]	4700	4737	7700	Note[6]
56	8	75	F075	48	C8	4800	4838	----	Note[6]	4800	4838	8D00	Note[6]
57	9 or Page Up	7D	F07D	49	C9	4900	4939	8400	Note[6]	4900	4939	8400	Note[6]

KEYBOARD SCAN CODES

Key	Key	Hardware Kybrd		Kybrd Interrupt		Standard ASCII				Extended ASCII			
#	Descript.	Make	Break	Make	Break	(Scancode/ASCII code)				(Scancode/ASCII code)			
		Code	Code	Code	Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
58	+	79	F079	4E	CE	4E2B	4E2B	----	----	4E2B	4E2B	9000	4E00
59	Caps Lock	58	F058	3A	BA	Note[7]	Note[7]	----	Note[7]	Note[7]	Note[7]	----	Note[7]
60	A or a	1C	F01C	1E	9E	1E61	1E41	1E01	1E00	1E61	1E41	1E01	1E00
61	S or s	1B	F01B	1F	9F	1F73	1F53	1F13	1F00	1F73	1F53	1F13	1F00
62	D or d	23	F023	20	A0	2064	2044	2004	2000	2064	2044	2004	2000
63	F or f	2B	F02B	21	A1	2166	2146	2106	2100	2166	2146	2106	2100
64	G or g	34	F034	22	A2	2267	2247	2207	2200	2267	2247	2207	2200
65	H or h	33	F033	23	A3	2368	2348	2308	2300	2368	2348	2308	2300
66	J or j	3B	F03B	24	A4	246A	244A	240A	2400	246A	244A	240A	2400
67	K or k	42	F042	25	A5	256B	254B	250B	2500	256B	254B	250B	2500
68	L or l	4B	F04B	26	A6	266C	264C	260C	2600	266C	264C	260C	2600
69	: or ;	4C	F04C	27	A7	273B	273A	----	----	273B	273A	----	2700
70	" or '	52	F052	28	A8	2827	2822	----	----	2827	2822	----	2800
71	Enter	5A	F05A	1C	9C	1C0D	1C0D	1C0A	----	1C0D	1C0D	1C0A	1C00
72	4	6B	F06B	4B	CB	4B00	4B34	7300	Note[6]	4B00	4B34	7300	Note[6]
73	5	73	F073	4C	CC	-----	4C35	---	Note[6]	4C00	4C35	8F00	Note[6]
74	6	74	F074	4D	CD	4D00	4D36	7400	Note[6]	4D00	4D36	7400	Note[6]

KEYBOARD SCAN CODES

shd be 2D00

Key	Key	Hardware Kybrd		Kybrd Interrupt		Standard ASCII				Extended ASCII			
#	Descript.	Make	Break	Make	Break	(Scancode/ASCII code)				(Scancode/ASCII code)			
		Code	Code	Code	Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
76	Left Shift	12	F012	2A	AA	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]
77	Z or z	1A	F01A	2C	AC	2C7A	2C5A	2C1A	2C00	2C7A	2C5A	2C1A	2C00
78	X or x	22	F022	2D	AD	2D78	2D58	2D18	2000	2D78	2D58	2D18	2000
79	C or c	21	F021	2E	AE	2E63	2E43	2E03	2E00	2E63	2E43	2E03	2E00
80	V or v	2A	F02A	2F	AF	2F76	2F56	2F16	2F00	2F76	2F56	2F16	2F00
81	B or b	32	F032	30	B0	3062	3042	3002	3000	3062	3042	3002	3000
82	N or n	31	F031	31	B1	316E	314E	310E	3100	316E	314E	310E	3100
83	M or m	3A	F03A	32	B2	326D	324D	320D	3200	326D	324D	320D	3200
84	< or ,	41	F041	33	B3	332C	333C	-----	-----	332C	333C	-----	3300
85	> or .	49	F049	34	B4	342E	343E	-----	-----	342E	343E	-----	3400
86	? or /	4A	F04A	35	B5	352F	353F	-----	-----	352F	353F	-----	3500
87	Right Shift	59	F059	36	B6	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]	Note[8]
88	Up Arrow	E075	E0F075	E02AE048	E0C8E0AA	4800	4800	-----	----	48E0	48E0	8DE0	9800
89	1 or End	69	F069	4F	CF	4F00	4F31	7500	Note[6]	4F00	4F31	7500	Note[6]
90	2	72	F072	50	D0	5000	5032	----	Note[6]	5000	5032	9100	Note[6]
91	3 or Pg Dn	7A	F07A	51	D1	5100	5133	7600	Note[6]	5100	5133	7600	Note[6]
92	Left Ctrl	14	F014	1D	9D	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]
93	Left Alt	11	F011	38	B8	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]
94	Space	29	F029	39	B9	3920	3920	3920	3920	3920	3920	3920	3920

? shld be 2D00

KEYBOARD SCAN CODES

Key	Key	Hardware Kybrd		Kybrd Interrupt		Standard ASCII				Extended ASCII			
#	Descript.	Make	Break	Make	Break	(Scancode/ASCII code)				(Scancode/ASCII code)			
		Code	Code	Code	Code	Norm	Shift	Ctrl	Alt	Norm	Shift	Ctrl	Alt
95	Right Alt	E011	E0F011	E038	E0B8	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]	Note[10]
96	Right Ctrl	E014	E0F014	E01D	E09D	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]	Note[9]
97	Left Arrow	E06B	E0F06B	E02AE04B	E0CBE0AA	4B00	4B00	7300	-----	4BE0	4BE0	73E0	9B00
98	Down Arrow	E072	E0F072	E02AE050	E0D0E0AA	5000	5000	-----	-----	50E0	50E0	91E0	A000
99	Right Arrow	E074	E0F074	E024E04D	E0CDE0AA	4D00	4000	7400	----	4DE0	4DE0	74E0	9D00
100	0 or Ins	70	F070	52	D2	5200	5230	----	Note[6]	5200	523H	9200	Note[6]
101	. or Del	71	F071	53	D3	5300	532E	----	----	5300	532E	9300	-----
102	Enter	E05A	E0F05A	E01C	E09C	1C0D	1C0D	1C0A	----	E00D	E00D	E00A	A600

NOTES

Note1 —INT O5H is invoked and a screen dump is performed

Note2 —the scroll lock active bit is toggled

Note3 —the pause state is initiated

Note4 —INT 1BH is invoked

Note5 —the numlock active bit is toggled

Note6 —ALT num pad generates raw ascii code of typed number

Note7 —the caps lock active bit is toggled

Note8 —hold shift lock active until key is released

Note9 —hold control shift active until key is released

Note10—hold alternate shift active until key is released

RADIO SHACK
A Division of Tandy Corporation
Fort Worth, Texas 76102

11/88-SWCG 874-8086 Printed in U.S.A.