intel®

8087
MATH COPROCESSOR

- **Adds Arithmetic, Trigonometric, Exponential, and Logarithmic Instructions to the Standard 8086/8088 and 80186/80188 Instruction Set for All Data Types**
- **CPU/8087 Supports 7 Data Types: 16-, 32-, 64-Bit Integers, 32-, 64-, 80-Bit Floating Point, and 18-Digit BCD Operands**
- **Compatible with IEEE Floating Point Standard 754**

- **Available in 5 MHz (8087), 8 MHz (8087-2) and 10 MHz (8087-1): 8 MHz 80186/80188 System Operation Supported with the 8087-1**
- **Adds 8 x 80-Bit Individually Addressable Register Stack to the 8086/8088 and 80186/80188 Architecture**
- **7 Built-In Exception Handling Functions**
- **MULTIBUS System Compatible Interface**

The Intel 8087 Math CoProcessor is an extension to the Intel 8086/8088 microprocessor architecture. When combined with the 8086/8088 microprocessor, the 8087 dramatically increases the processing speed of computer applications which utilize mathematical operations such as CAM, numeric controllers, CAD or graphics.

The 8087 Math CoProcessor adds 68 mnemonics to the 8086 microprocessor instruction set. Specific 8087 math operations include logarithmic, arithmetic, exponential, and trigonometric functions. The 8087 supports integer, floating point and BCD data formats, and fully conforms to the ANSI/IEEE floating point standard.

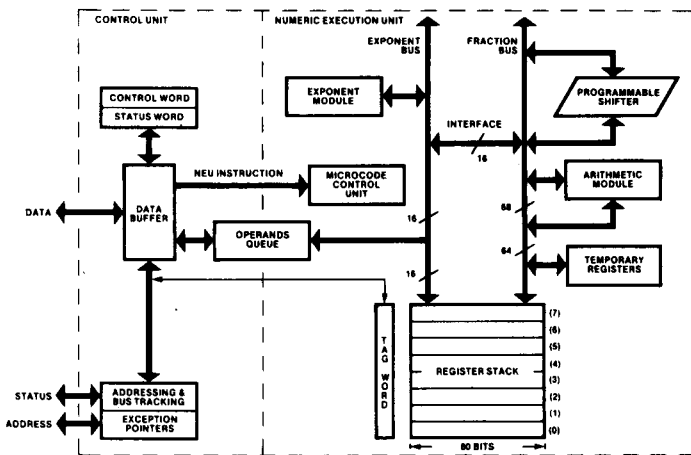The 8087 is fabricated with HMOS III technology and packaged in a 40-pin cerdip package.
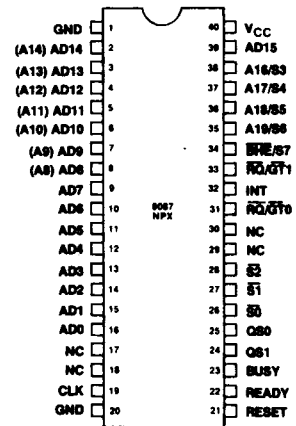

205835–1

Figure 1. 8087 Block Diagram

205835–2

Figure 2. 8087 Pin Configuration

Table 1. 8087 Pin Description

Symbol	Type	Name and Function
AD15–AD0	I/O	**ADDRESS DATA:** These lines constitute the time multiplexed memory address (T_1) and data (T_2, T_3, T_W, T_4) bus. A0 is analogous to the $\overline{BHE}$ for the lower byte of the data bus, pins D7–D0. It is LOW during T_1 when a byte is to be transferred on the lower portion of the bus in memory operations. Eight-bit oriented devices tied to the lower half of the bus would normally use A0 to condition chip select functions. These lines are active HIGH. They are input/output lines for 8087-driven bus cycles and are inputs which the 8087 monitors when the CPU is in control of the bus. A15–A8 do not require an address latch in an 8088/8087 or 80188/8087. The 8087 will supply an address for the T_1–T_4 period.
A19/S6, A18/S5, A17/S4, A16/S3	I/O	**ADDRESS MEMORY:** During T_1 these are the four most significant address lines for memory operations. During memory operations, status information is available on these lines during T_2, T_3, T_W, and T_4. For 8087-controlled bus cycles, S6, S4, and S3 are reserved and currently one (HIGH), while S5 is always LOW. These are inputs which the 8087 monitors when the CPU is in control of the bus.
BHE/S7	I/O	**BUS HIGH ENABLE:** During T_1 the bus high enable signed ($\overline{BHE}$) should be used to enable data onto the most significant half of the data bus, pins D15–D8. Eight-bit-oriented devices tied to the upper half of the bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is LOW during T_1 for read and write cycles when a byte is to be transferred on the high portion of the bus. The S7 status information is available during T_2, T_3, T_W, and T_4. The signal is active LOW. S7 is an input which the 8087 monitors during the CPU-controlled bus cycles.
$\overline{S2}$, $\overline{S1}$, $\overline{S0}$	I/O	**STATUS:** For 8087-driven, these status lines are encoded as follows: $\overline{S2}$ $\overline{S1}$ $\overline{S0}$ 0 (LOW) X X Unused 1 (HIGH) 0 0 Unused 1 0 1 Read Memory 1 1 0 Write Memory 1 1 1 Passive Status is driven active during T_4, remains valid during T_1 and T_2, and is returned to the passive state (1, 1, 1) during T_3 or during T_W when READY is HIGH. This status is used by the 8288 Bus Controller (or the 82188 Integrated Bus Controller with an 80186/80188 CPU) to generate all memory access control signals. Any change in $\overline{S2}$, $\overline{S1}$, or $\overline{S0}$ during T_4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T_3 or T_W is used to indicate the end of a bus cycle. These signals are monitored by the 8087 when the CPU is in control of the bus.
$\overline{RQ}/\overline{GT0}$	I/O	**REQUEST/GRANT:** This request/grant pin is used by the 8087 to gain control of the local bus from the CPU for operand transfers or on behalf of another bus master. It must be connected to one of the two processor request/grant pins. The request/grant sequence on this pin is as follows: 1. A pulse one clock wide is passed to the CPU to indicate a local bus request by either the 8087 or the master connected to the 8087 $\overline{RQ}/\overline{GT}$1 pin. 2. The 8087 waits for the grant pulse and when it is received will either initiate bus transfer activity in the clock cycle following the grant or pass the grant out on the $\overline{RQ}/\overline{GT}$1 pin in this clock if the initial request was for another bus master. 3. The 8087 will generate a release pulse to the CPU one clock cycle after the completion of the last 8087 bus cycle or on receipt of the release pulse from the bus master on $\overline{RQ}/\overline{GT}$1. For 80186/80188 systems the same sequence applies except $\overline{RQ}/\overline{GT}$ signals are converted to appropriate HOLD, HLDA signals by the 82188 Integrated Bus Controller. This is to conform with 80186/80188's HOLD, HLDA bus exchange protocol. Refer to the 82188 data sheet for further information.

3

Table 1. 8087 Pin Description (Continued)

Symbol	Type	Name and Function
$\overline{RQ}/\overline{GT}1$	I/O	**REQUEST/GRANT:** This request/grant pin is used by another local bus master to force the 8087 to request the local bus. If the 8087 is not in control of the bus when the request is made the request/grant sequence is passed through the 8087 on the $\overline{RQ}/\overline{GT}0$ pin one cycle later. Subsequent grant and release pulses are also passed through the 8087 with a two and one clock delay, respectively, for resynchronization. $\overline{RQ}/\overline{GT}1$ has an internal pullup resistor, and so may be left unconnected. If the 8087 has control of the bus the request/grant sequence is as follows: 1. A pulse 1 CLK wide from another local bus master indicates a local bus request to the 8087 (pulse 1). 2. During the 8087's next T_4 or T_1 a pulse 1 CLK wide from the 8087 to the requesting master (pulse 2) indicates that the 8087 has allowed the local bus to float and that it will enter the "RQ/GT acknowledge" state at the next CLK. The 8087's control unit is disconnected logically from the local bus during "RQ/GT acknowledge." 3. A pulse 1 CLK wide from the requesting master indicates to the 8087 (pulse 3) that the "RQ/GT" request is about to end and that the 8087 can reclaim the local bus at the next CLK. Each master-master exchange of the local bus is a sequence of 3 pulses. There must be one dead CLK cycle after each bus exchange. Pulses are active LOW. For 80186/80188 system, the $\overline{RQ}/\overline{GT}1$ line may be connected to the 82188 Integrated Bus Controller. In this case, a third processor with a HOLD, HLDA bus exchange system may acquire the bus from the 8087. For this configuration, $\overline{RQ}/\overline{GT}1$ will only be used if the 8087 is the bus master. Refer to 82188 data sheet for further information.
QS1, QS0	I	**QS1, QS0:** QS1 and QS0 provide the 8087 with status to allow tracking of the CPU instruction queue. **QS1** **QS0** 0 (LOW) 0 No Operation 0 1 First Byte of Op Code from Queue 1 (HIGH) 0 Empty the Queue 1 1 Subsequent Byte from Queue
INT	O	**INTERRUPT:** This line is used to indicate that an unmasked exception has occurred during numeric instruction execution when 8087 interrupts are enabled. This signal is typically routed to an 8259A for 8086/8088 systems and to INT0 for 80186/80188 systems. INT is active HIGH.
BUSY	O	**BUSY:** This signal indicates that the 8087 NEU is executing a numeric instruction. It is connected to the CPU's $\overline{TEST}$ pin to provide synchronization. In the case of an unmasked exception BUSY remains active until the exception is cleared. BUSY is active HIGH.
READY	I	**READY:** READY is the acknowledgement from the addressed memory device that it will complete the data transfer. The RDY signal from memory is synchronized by the 8284A Clock Generator to form READY for 8086 systems. For 80186/80188 systems, RDY is synchronized by the 82188 Integrated Bus Controller to form READY. This signal is active HIGH.
RESET	I	**RESET:** RESET causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. RESET is internally synchronized.
CLK	I	**CLOCK:** The clock provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
V_{CC}		**POWER:** V_{CC} is the +5V power supply pin.
GND		**GROUND:** GND are the ground pins.

NOTE:
For the pin descriptions of the 8086, 8088, 80186 and 80188 CPUs, reference the respective data sheets (8086, 8088, 80186, 80188).

APPLICATION AREAS

The 8087 provides functions meant specifically for high performance numeric processing requirements. Trigonometric, logarithmic, and exponential functions are built into the coprocessor hardware. These functions are essential in scientific, engineering, navigational, or military applications.

The 8087 also has capabilities meant for business or commercial computing. An 8087 can process Binary Coded Decimal (BCD) numbers up to 18 digits without roundoff errors. It can also perform arithmetic on integers as large as 64 bits $\pm 10^{18}$).

PROGRAMMING LANGUAGE SUPPORT

Programs for the 8087 can be written in Intel's high-level languages for 8086/8088 and 80186/80188 Systems; ASM-86 (the 8086, 8088 assembly language), PL/M-86, FORTRAN-86, and PASCAL-86.

RELATED INFORMATION

For 8086, 8088, 80186 or 80188 details, refer to the respective data sheets. For 80186 or 80188 systems, also refer to the 82188 Integrated Bus Controller data sheet.

FUNCTIONAL DESCRIPTION

The 8087 Math CoProcessor's architecture is designed for high performance numeric computing in conjunction with general purpose processing.

The 8087 is a numeric processor extension that provides arithmetic and logical instruction support for a variety of numeric data types. It also executes numerous built-in transcendental functions (e.g., tangent and log functions). The 8087 executes instructions as a coprocessor to a maximum mode CPU. It effectively extends the register and instruction set of the system and adds several new data types as well. Figure 3 presents the registers of the CPU + 8087. Table 2 shows the range of data types supported by the 8087. The 8087 is treated as an extension to the CPU, providing register, data types, control, and instruction capabilities at the hardware level. At the programmer's level the CPU and the 8087 are viewed as a single unified processor.

System Configuration

As a coprocessor to an 8086 or 8088, the 8087 is wired in parallel with the CPU as shown in Figure 4. Figure 5 shows the 80186/80188 system configuration. The CPU's status (S0–S2) and queue status lines (QS0–QS1) enable the 8087 to monitor and decode instructions in synchronization with the CPU and without any CPU overhead. For 80186/80188 systems, the queue status signals of the 80186/80188 are synchronized to 8087 requirements by the 8288 Integrated Bus Controller. Once started, the 8087 can process in parallel with, and independent of, the host CPU. For resynchronization, the 8087's BUSY signal informs the CPU that the 8087 is executing an instruction and the CPU WAIT instruction tests this signal to insure that the 8087 is ready to execute subsequent instructions. The 8087 can interrupt the CPU when it detects an error or exception. The 8087's interrupt request line is typically routed to the CPU through an 8259A Programmable Interrupt Controller for 8086, 8088 systems and INT0 for 80186/80188.
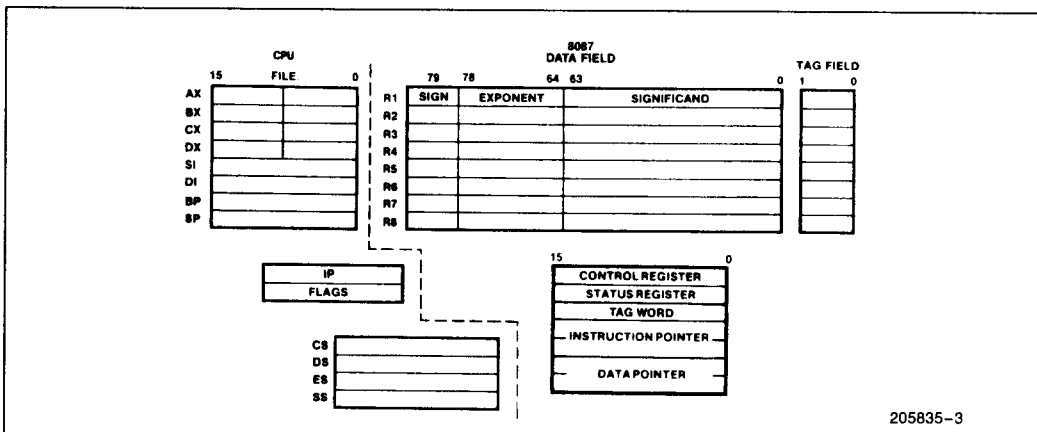

Figure 3. CPU + 8087 Architecture

The 8087 uses one of the request/grant lines of the 8086/8088 architecture (typically $\overline{RQ}/\overline{GT}0$) to obtain control of the local bus for data transfers. The other request/grant line is available for general system use (for instance by an I/O processor in LOCAL mode). A bus master can also be connected to the 8087's $\overline{RQ}/\overline{GT}1$ line. In this configuration the 8087 will pass the request/grant handshake signals between the CPU and the attached master when the 8087 is not in control of the bus and will relinquish the bus to the master directly when the 8087 is in control. In this way two additional masters can be configured in an 8086/8088 system; one will share the 8086/8088 bus with the 8087 on a first-come first-served basis, and the second will be guaranteed to be higher in priority than the 8087.

For 80186/80188 systems, $\overline{RQ}/\overline{GT}0$ and $\overline{RQ}/\overline{GT}1$ are connected to the corresponding inputs of the 82188 Integrated Bus Controller. Because the 80186/80188 has a HOLD, HLDA bus exchange protocol, an interface is needed which will translate $\overline{RQ}/\overline{GT}$ signals to corresponding HOLD, HLDA signals and vice versa. One of the functions of the 82188 IBC is to provide this translation. $\overline{RQ}/\overline{GT}0$ is translated to HOLD, HLDA signals which are then directly connected to the 80186/80188. The $\overline{RQ}/\overline{GT}1$ line is also translated into HOLD, HLDA signals (referred to as SYSHOLD, SYSHLDA signals) by the 82188 IBC. This allows a third processor (using a HOLD, HLDA bus exchange protocol) to gain control of the bus.

Unlike an 8086/8087 system, $\overline{RQ}/\overline{GT}$ is only used when the 8087 has bus control. If the third processor requests the bus when the current bus master is the 80186/80188, the 82188 IBC will directly pass the request onto the 80186/80188 without going through the 8087. The third processor has the highest bus priority in the system. If the 8087 requests the bus while the third processor has bus control, the grant pulse will not be issued until the third processor releases the bus (using SYSHOLD). In this configuration, the third processor has the highest priority, the 8087 has the next highest, and the 80186/80188 has the lowest bus priority.

Bus Operation

The 8087 bus structure, operation and timing are identical to all other processors in the 8086/8088 series (maximum mode configuration). The address is time multiplexed with the data on the first 16/8 lines of the address/data bus. A16 through A19 are time multiplexed with four status lines S3–S6. S3, S4 and S6 are always one (HIGH) for 8087-driven bus cycles while S5 is always zero (LOW). When the 8087 is monitoring CPU bus cycles (passive mode) S6 is also monitored by the 8087 to differentiate 8086/8088 activity from that of a local I/O processor or any other local bus master. (The 8086/8088 must be the only processor on the local bus to drive S6 LOW). S7 is multiplexed with and has the same value as $\overline{BHE}$ for all 8087 bus cycles.

Table 2. 8087 Data Types

Data Formats	Range	Precision	Most Significant Byte								
			7 0	7 0	7 0	7 0	7 0	7 0	7 0	7 0	7 0
Word Integer	10^4	16 Bits	I_{15}		I_0 Two's Complement						
Short Integer	10^9	32 Bits	I_{31}				I_0 Two's Complement				
Long Integer	10^{18}	64 Bits	I_{63}							I_0	Two's Complement
Packed BCD	10^{18}	18 Digits	S — $D_{17}D_{16}$								D_1 D_0
Short Real	$10^{\pm38}$	24 Bits	S E_7	E_0 F_1		F_{23} F_0 Implicit					
Long Real	$10^{\pm308}$	53 Bits	S E_{10}	E_0 F_1						F_{52} F_0 Implicit	
Temporary Real	$10^{\pm4932}$	64 Bits	S E_{14}	E_0 F_0							F_{63}

Integer: I
Packed BCD: $(-1)^S(D_{17}...D_0)$
Real: $(-1)^S(2^{E-Bias})(F_0 \bullet F_1...)$
bias = 127 for Short Real
1023 for Long Real
16383 for Temp Real

The first three status lines, $\overline{S0}$–$\overline{S2}$, are used with an 8288 bus controller or 82188 Integrated Bus Controller to determine the type of bus cycle being run:

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	
0	X	X	Unused
1	0	0	Unused
1	0	1	Memory Data Read
1	1	0	Memory Data Write
1	1	1	Passive (no bus cycle)

Programming Interface

The 8087 includes the standard 8086, 8088 instruction set for general data manipulation and program control. It also includes 68 numeric instructions for extended precision integer, floating point, trigonometric, logarithmic, and exponential functions. Sample execution times for several 8087 functions are shown in Table 3. Overall performance is up to 100 times that of an 8086 processor for numeric instructions.

Any instruction executed by the 8087 is the combined result of the CPU and 8087 activity. The CPU and the 8087 have specialized functions and registers providing fast concurrent operation. The CPU controls overall program execution while the 8087 uses the coprocessor interface to recognize and perform numeric operations.

Table 2 lists the seven data types the 8087 supports and presents the format for each type. Internally, the 8087 holds all numbers in the temporary real format. Load and store instructions automatically convert operands represented in memory as 16-, 32-, or 64-bit integers, 32- or 64-bit floating point numbers or 18-digit packed BCD numbers into temporary real format and vice versa. The 8087 also provides the capability to control round off, underflow, and overflow errors in each calculation.

Computations in the 8087 use the processor's register stack. These eight 80-bit registers provide the equivalent capacity of 20 32-bit registers. The 8087 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.

Table 5 lists the 8087's instructions by class. All appear as ESCAPE instructions to the host. Assembly language programs are written in ASM-86, the 8086, 8088 assembly language.

Table 3. Execution Times for Selected 8086/8087 Numeric Instructions and Corresponding 8086 Emulation

Floating Point Instruction	Approximate Execution Time (μs)	
	8086/8087 (8 MHz Clock)	8086 Emulation
Add/Subtract	10.6	1000
Multiply (Single Precision)	11.9	1000
Multiply (Extended Precision)	16.9	1312
Divide	24.4	2000
Compare	–5.6	812
Load (Double Precision)	–6.3	1062
Store (Double Precision)	13.1	750
Square Root	22.5	12250
Tangent	56.3	8125
Exponentiation	62.5	10687

NUMERIC PROCESSOR EXTENSION ARCHITECTURE

As shown in Figure 1, the 8087 is internally divided into two processing elements, the control unit (CU) and the numeric execution unit (NEU). The NEU executes all numeric instructions, while the CU receives and decodes instructions, reads and writes memory operands and executes 8087 control instructions. The two elements are able to operate independently of one another, allowing the CU to maintain synchronization with the CPU while the NEU is busy processing a numeric instruction.

Control Unit

The CU keeps the 8087 operating in synchronization with its host CPU. 8087 instructions are intermixed with CPU instructions in a single instruction stream. The CPU fetches all instructions from memory; by monitoring the status ($\overline{S0}$–$\overline{S2}$, S6) emitted by the CPU, the control unit determines when an instruction is being fetched. The CPU monitors the data bus in parallel with the CPU to obtain instructions that pertain to the 8087.
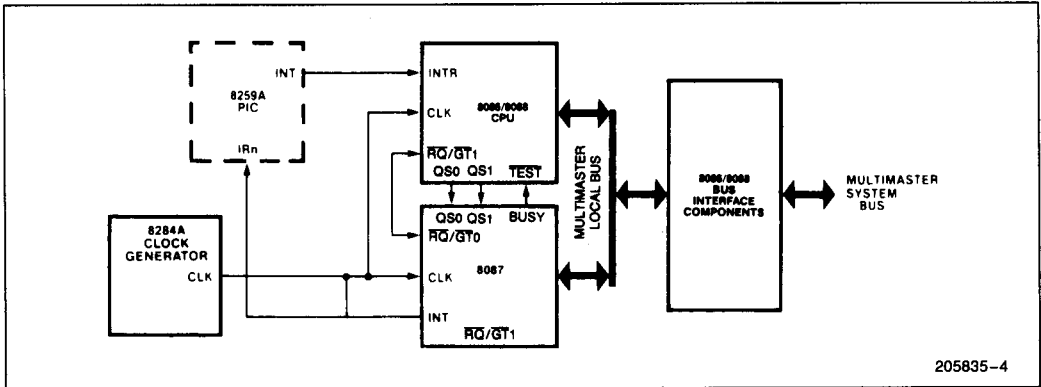
3
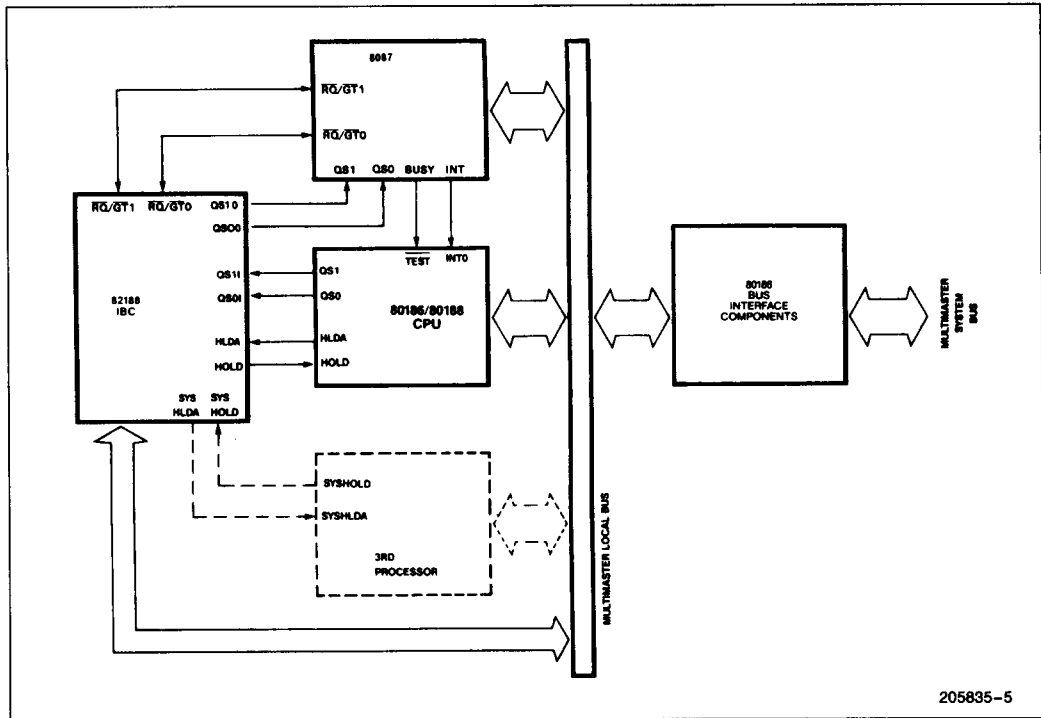
intel.

Figure 4. 8086/8087, 8088/8087 System Configuration

Figure 5. 80186/8087, 80188/8087 System Configuration

The CU maintains an instruction queue that is identical to the queue in the host CPU. The CU automatically determines if the CPU is an 8086/80186 or an 8088/80188 immediately after reset (by monitoring the BHE/S7 line) and matches its queue length accordingly. By monitoring the CPU's queue status lines (QS0, QS1), the CU obtains and decodes instructions from the queue in synchronization with the CPU.

A numeric instruction appears as an ESCAPE instruction to the CPU. Both the CPU and 8087 decode and execute the ESCAPE instruction together. The 8087 only recognizes the numeric instructions shown in Table 5. The start of a numeric operation is accomplished when the CPU executes the ESCAPE instruction. The instruction may or may not identify a memory operand.

The CPU does, however, distinguish between ESC instructions that reference memory and those that do not. If the instruction refers to a memory operand, the CPU calculates the operand's address using any one of its available addressing modes, and then performs a "dummy read" of the word at that location. (Any location within the 1M byte address space is allowed.) This is a normal read cycle except that the CPU ignores the data it receives. If the ESC instruction does not contain a memory reference (e.g. an 8087 stack operation), the CPU simply proceeds to the next instruction.

An 8087 instruction can have one of three memory reference options: (1) not reference memory; (2) load an operand word from memory into the 8087; or (3) store an operand word from the 8087 into memory. If no memory reference is required, the 8087 simply executes its instruction. If a memory reference is required, the CU uses a "dummy read" cycle initiated by the CPU to capture and save the address that the CPU places on the bus. If the instruction is a load, the CU additionally captures the data word when it becomes available on the local data bus. If data required is longer than one word, the CU immediately obtains the bus from the CPU using the request/grant protocol and reads the rest of the information in consecutive bus cycles. In a store operation, the CU captures and saves the store address as in a load, and ignores the data word that follows in the "dummy read" cycle. When the 8087 is ready to perform the store, the CU obtains the bus from the CPU and writes the operand starting at the specified address.

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 fractions 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 activates the 8087 BUSY signal. This signal can be used in conjunction with the CPU WAIT instruction to resynchronize both processors when the NEU has completed its current instruction.

Register Set

The CPU + 8087 register set is shown in Figure 3. Each of the eight data registers in the 8087's register stack is 80 bits and is divided into "fields" corresponding to the 8087's temporary real data type.

At a given point in time the TOP field in the control 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 CPU stacks in memory, the 8087 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 to by the TOP. Other instructions allow the programmer to explicitly specify the register which is to be used. Explicit register addressing is "top-relative."

Status Word

The status word shown in Figure 6 reflects the overall state of the 8087; it may be stored in memory and then inspected by CPU code. The status word is a 16-bit register divided into fields as shown in Figure 6. The busy bit (bit 15) indicates whether the NEU is either executing an instruction or has an interrupt request pending (B = 1), or is idle (B = 0). Several instructions which store and manipulate the status word are executed exclusively by the CU, and these do not set the busy bit themselves.

NOTES:
1. IR is set if any unmasked exception bit is set, cleared otherwise.
2. See Table 3 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.
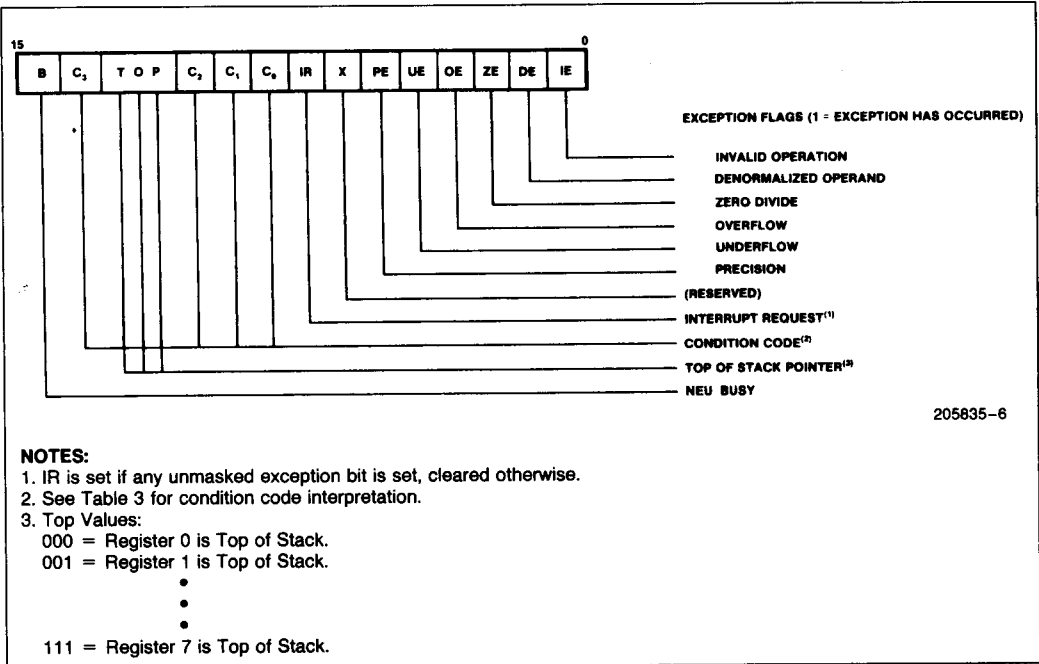
Figure 6. 8087 Status Word

The four numeric condition code bits (C_0–C_3) are similar to flags in a CPU: various instructions update these bits to reflect the outcome of the 8087 operations. The effect of these instructions on the condition code bits is summarized in Table 4.

Bits 14–12 of the status word point to the 8087 register that is the current top-of-stack (TOP) as described above.

Bit 7 is the interrupt request bit. This bit is set if any unmasked exception bit is set and cleared otherwise.

Bits 5–0 are set to indicate that the NEU has detected an exception while executing an instruction.

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 8087's performance. The tag word can be used, however, to interpret the contents of 8087 registers.

Instruction and Data Pointers

The instruction and data pointers (see Figure 8) are provided for user-written error handlers. Whenever the 8087 executes a math instruction, the CU saves the instruction address, the operand address (if present) and the instruction opcode. 8087 instructions can store this data into memory.

Figure 7. 8087 Tag Word

Table 4a. Condition Code Interpretation

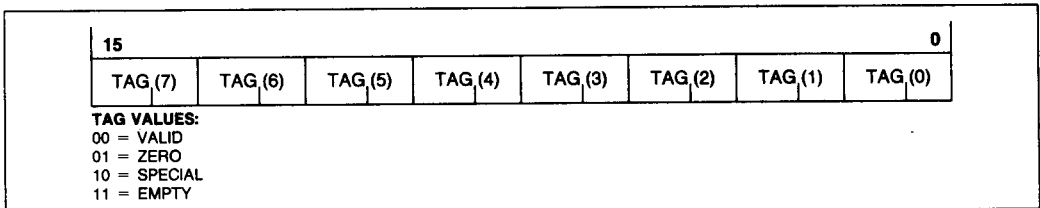| Instruction Type	C_3	C_2	C_1	C_0	Interpretation
Compare, Test	0	0	X	0	ST > Source or 0 (FTST)
	0	0	X	1	ST < Source or 0 (FTST)
	1	0	X	0	ST = Source or 0 (FTST)
	1	1	X	1	ST is not comparable
Remainder	Q_1	0	Q_0	Q_2	Complete reduction with three low bits of quotient (See Table 4b)
	U	1	U	U	Incomplete 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 Instruction As a Function of Divided Value

Dividend Range	Q_2	Q_1	Q_0
Dividend < 2 * Modulus	C_3[1]	C_1[1]	Q_0
Dividend < 4 * Modulus	C_3[1]	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.

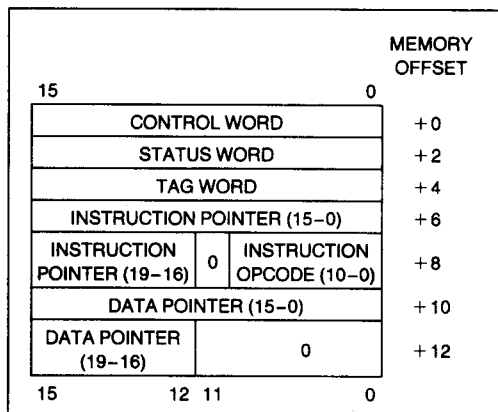|	MEMORY OFFSET
CONTROL WORD	+0
STATUS WORD	+2
TAG WORD	+4
INSTRUCTION POINTER (15–0)	+6
INSTRUCTION POINTER (19–16) / 0 / INSTRUCTION OPCODE (10–0)	+8
DATA POINTER (15–0)	+10
DATA POINTER (19–16) / 0	+12

Figure 8. 8087 Instruction and Data Pointer Image in Memory

Control Word

The 8087 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 the fields in the control word.

The low order byte of this control word configures 8087 interrupts and exception masking. Bits 5–0 of the control word contain individual masks for each of the six exceptions that the 8087 recognizes and bit 7 contains a general mask bit for all 8087 interrupts. The high order byte of the control word configures the 8087 operating mode including precision, rounding, and infinity controls. The precision control bits (bits 9–8) can be used to set the 8087 internal operating precision at less than the default of temporary real precision. This can be useful in providing compatibility with earlier generation arithmetic processors of smaller precision than the 8087. The rounding control bits (bits 11–10) provide for directed rounding and true chop as well as the unbiased round to nearest mode specified in the proposed IEEE standard. Control over closure of the number space at infinity is also provided (either affine closure, $\pm\infty$, or projective closure, ∞, is treated as unsigned, may be specified).

Exception Handling

The 8087 detects six different exception conditions that can occur during instruction execution. Any or all exceptions will cause an interrupt if unmasked and interrupts are enabled.

If interrupts are disabled the 8087 will simply continue execution regardless of whether the host clears the exception. If a specific exception class is masked and that exception occurs, however, the 8087 will post the exception in the status register and perform an on-chip default exception handling procedure, thereby allowing processing to continue. The exceptions that the 8087 detects are the following:

1. INVALID OPERATION: Stack overflow, stack underflow, indeterminate form (0/0, $\infty - \infty$, etc.) or the use of a Non-Number (NAN) as an operand. An exponent value is reserved and any bit pattern with this value in the exponent field is termed a Non-Number and causes this exception. If this exception is masked, the 8087's default response is to generate a specific NAN called IN-DEFINITE, or to propagate already existing NANs as the calculation result.

NOTES:
1. Precision Control
   00 = 24 bits
   01 = Reserved
   10 = 53 bits
   11 = 64 bits

2. Rounding Control
   00 = Round to Nearest or Even
   01 = Round Down (toward $-\infty$)
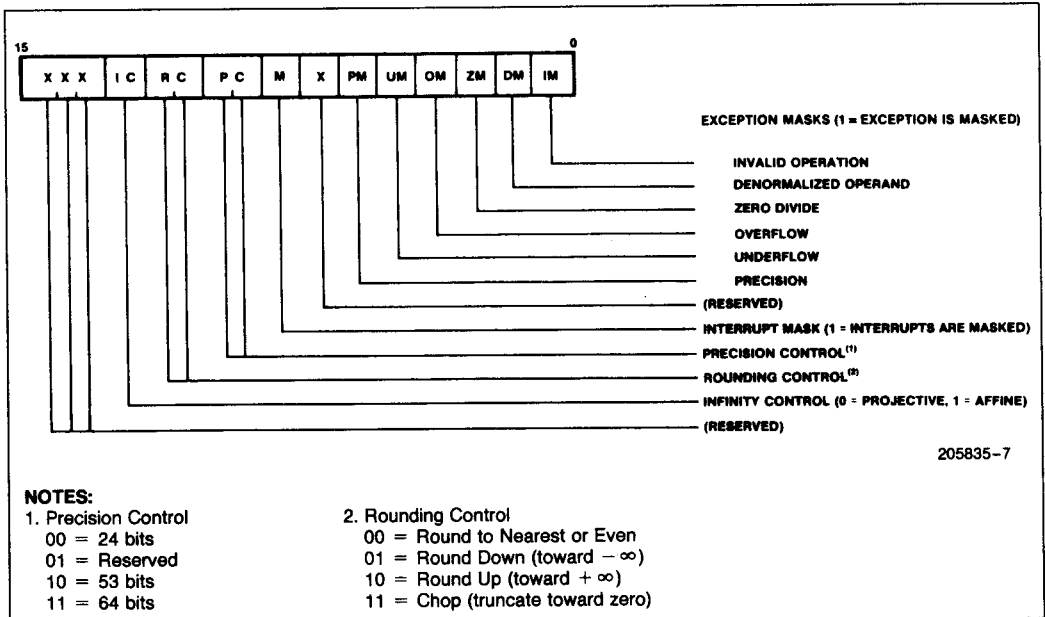   10 = Round Up (toward $+\infty$)
   11 = Chop (truncate toward zero)

Figure 9. 8087 Control Word

2. OVERFLOW: The result is too large in magnitude to fit the specified format. The 8087 will generate an encoding for infinity if this exception is masked.

3. ZERO DIVISOR: The divisor is zero while the dividend is a non-infinite, non-zero number. Again, the 8087 will generate an encoding for infinity if this exception is masked.

4. UNDERFLOW: The result is non-zero but too small in magnitude to fit in the specified format. If this exception is masked the 8087 will denormal-

ize (shift right) the fraction until the exponent is in range. This process is called gradual underflow.

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

6. INEXACT RESULT: If 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.

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 ....3.0 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

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

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage (Note 8)		0.45	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{CC}	Power Supply Current		475	mA	T_A = 25°C
I_{LI}	Input Leakage Current		±10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current		±10	μA	T_A = 25°C
V_{CL}	Clock Input Low Voltage	−0.5	0.6	V	
V_{CH}	Clock Input High Voltage	3.9	V_{CC} + 1.0	V	
C_{IN}	Capacitance of Inputs		10	pF	fc = 1 MHz
C_{IO}	Capacitance of I/O Buffer (AD0−15, A_{16}–A_{19}, BHE, S2–S0, RQ/GT) and CLK		15	pF	fc = 1 MHz
C_{OUT}	Capacitance of Outputs BUSY INT		10	pF	fc = 1 MHz

3

A.C. CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 5\%$

TIMING REQUIREMENTS

Symbol	Parameter	8087		8087-2		8087-1 (See Note 7)		Units	Test Conditions
		Min	Max	Min	Max	Min	Max		
TCLCL	CLK Cycle Period	200	500	125	500	100	500	ns	
TCLCH	CLK Low Time	118		68		53		ns	
TCHCL	CLK High Time	69		44		39		ns	
TCH1CH2	CLK Rise Time		10		10		15	ns	From 1.0V to 3.5V
TCL2CL2	CLK Fall Time		10		10		15	ns	From 3.5V to 1.0V
TDVCL	Data In Setup Time	30		20		15		ns	
TCLDX	Data In Hold Time	10		10		10		ns	
TRYHCH	READY Setup Time	118		68		53		ns	
TCHRYX	READY Hold Time	30		20		5		ns	
TRYLCL	READY Inactive to CLK (Note 6)	−8		−8		−10		ns	
TGVCH	RQ/GT Setup Time (Note 8)	30		15		15		ns	
TCHGX	RQ/GT Hold Time	40		30		20		ns	
TQVCL	QS0−1 Setup Time (Note 8)	30		30		30		ns	
TCLQX	QS0−1 Hold Time	10		10		5		ns	
TSACH	Status Active Setup Time	30		30		30		ns	
TSNCL	Status Inactive Setup Time	30		30		30		ns	
TILIH	Input Rise Time (Except CLK)		20		20		20	ns	From 0.8V to 2.0V
TIHIL	Input Fall Time (Except CLK)		12		12		15	ns	From 2.0V to 0.8V

TIMING RESPONSES

Symbol	Parameter	8087		8087-2		8087-1 (See Note 7)		Units	Test Conditions
		Min	Max	Min	Max	Min	Max		
TCLML	Command Active Delay (Notes 1, 2)	10/0	35/70	10/0	35/70	10/0	35/70	ns	C_L = 20–100 pF for all 8087 Outputs (in addition to 8087 self-load)
TCLMH	Command Inactive Delay (Notes 1, 2)	10/0	35/55	10/0	35/55	10/0	35/70	ns	
TRYHSH	Ready Active to Status Passive (Note 5)		110		65		45	ns	
TCHSV	Status Active Delay	10	110	10	60	10	45	ns	
TCLSH	Status Inactive Delay	10	130	10	70	10	55	ns	
TCLAV	Address Valid Delay	10	110	10	60	10	55	ns	
TCLAX	Address Hold Time	10		10		10		ns	

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

TIMING RESPONSES (Continued)

Symbol	Parameter	8087		8087-2		8087-1 (See Note 7)		Units	Test Conditions
		Min	Max	Min	Max	Min	Max		
TCLAZ	Address Float Delay	TCLAX	80	TCLAX	50	TCLAX	45	ns	C_L = 20–100 pF for all 8087 Outputs (in addition to 8087 self-load)
TSVLH	Status Valid to ALE High (Notes 1, 2)		15/30		15/30		15/30	ns	
TCLLH	CLK Low to ALE Valid (Notes 1, 2)		15/30		15/30		15/30	ns	
TCHLL	ALE Inactive Delay (Notes 1, 2)		15/30		15/30		15/30	ns	
TCLDV	Data Valid Delay	10	110	10	60	10	50	ns	
TCHDX	Status Hold Time	10		10		10	45	ns	
TCLDOX	Data Hold Time	10		10		10		ns	
TCVNV	Control Active Delay (Notes 1, 3)	5	45	5	45	5	45	ns	
TCVNX	Control Inactive Delay (Notes 1, 3)	10	45	10	45	10	45	ns	
TCHBV	BUSY and INT Valid Delay	10	150	10	85	10	65	ns	
TCHDTL	Direction Control Active Delay (Notes 1, 3)		50		50		50	ns	
TCHDTH	Direction Control Inactive Delay (Notes 1, 3)		30		30		30	ns	
TSVDTV	STATUS to DT/$\overline{R}$ Delay (Notes 1, 4)	0	30	0	30	0	30	ns	
TCLDTV	DT/$\overline{R}$ Active Delay (Notes 1, 4)	0	55	0	55	0	55	ns	
TCHDNV	$\overline{DEN}$ Active Delay (Notes 1, 4)	0	55	0	55	0	55	ns	
TCHDNX	$\overline{DEN}$ Inactive Delay (Notes 1, 4)	5	55	5	55	5	55	ns	
TCLGL	RQ/GT Active Delay (Note 8)	0	85	0	50	0	38	ns	C_L = 40 pF (in addition to 8087 self-load)
TCLGH	RQ/GT Inactive Delay	0	85	0	50	0	45	ns	
TOLOH	Output Rise Time		20		20		15	ns	From 0.8V to 2.0V
TOHOL	Output Fall Time		12		12		12	ns	From 2.0V to 0.8V

NOTES:
1. Signal at 8284A, 8288, or 82188 shown for reference only.
2. 8288 timing/82188 timing.
3. 8288 timing.
4. 82188 timing.
5. Applies only to T_3 and wait states.
6. Applies only to T_2 state (8 ns into T_3).
7. IMPORTANT SYSTEM CONSIDERATION: Some 8087-1 timing parameters are constrained relative to the corresponding 8086-1 specifications. Therefore, 8086-1 systems incorporating the 8087-1 should be designed with the 8087-1 specifications.
8. Changes since last revision.

intel ®

A.C. TESTING INPUT, OUTPUT WAVEFORM

205835-8

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0".

A.C. TESTING LOAD CIRCUIT

205835-9

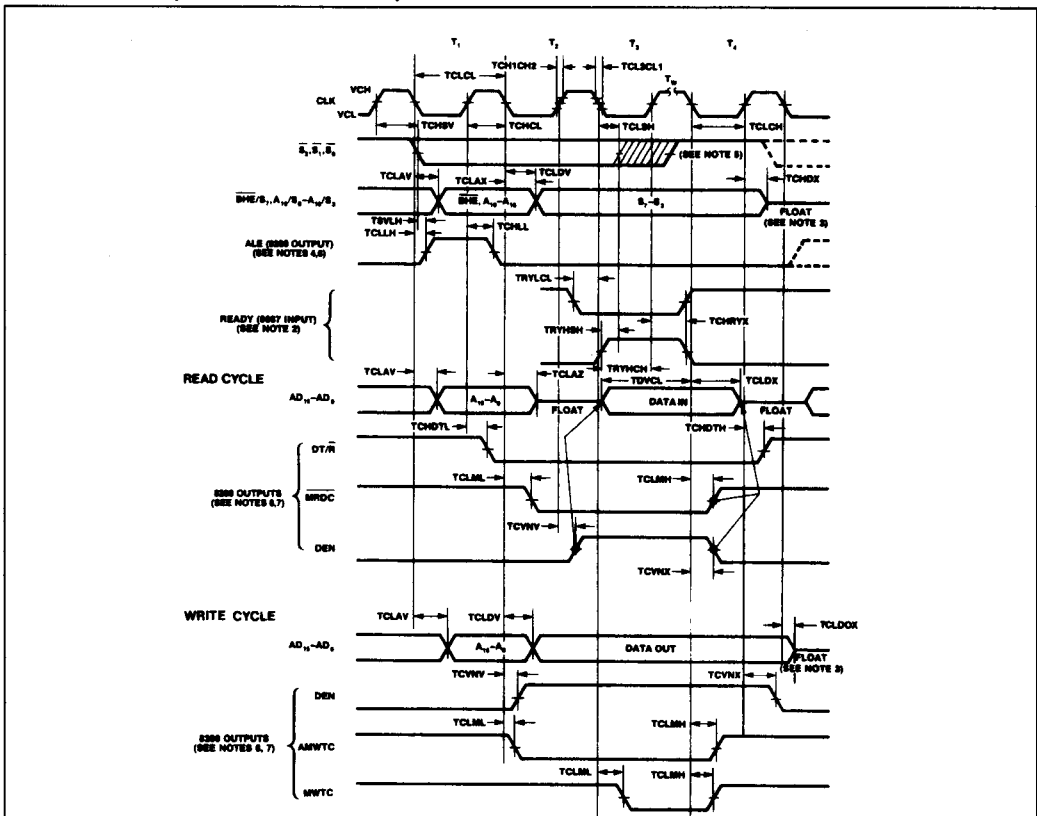C_L Includes Jig Capacitance
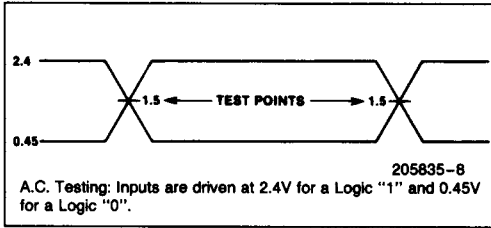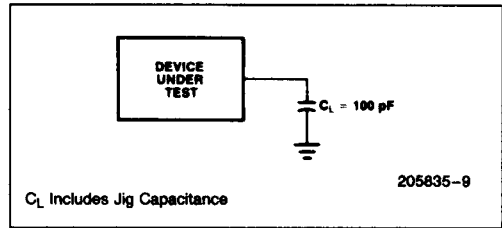
WAVEFORMS

MASTER MODE (with 8288 references)

205835-10

NOTES:
1. All signals switch between V_{OL} and V_{OH} unless otherwise specified.
2. READY is sampled near the end of T_2, T_3 and T_W to determine if T_W machine states are to be inserted.
3. The local bus floats only if the 8087 is returning control to the 8086/8088.
4. ALE rises at later of (TSVLH, TCLLH).
5. Status inactive in state just prior to T_4.
6. Signals at 8284A or 8288 are shown for reference only.
7. The issuance of 8288 command and control signals ($\overline{MRDC}$, ($\overline{MWTC}$, $\overline{AMWC}$, and DEN) lags the active high 8288 CEN.
8. All timing measurements are made at 1.5V unless otherwise noted.

intel.

WAVEFORMS (Continued)

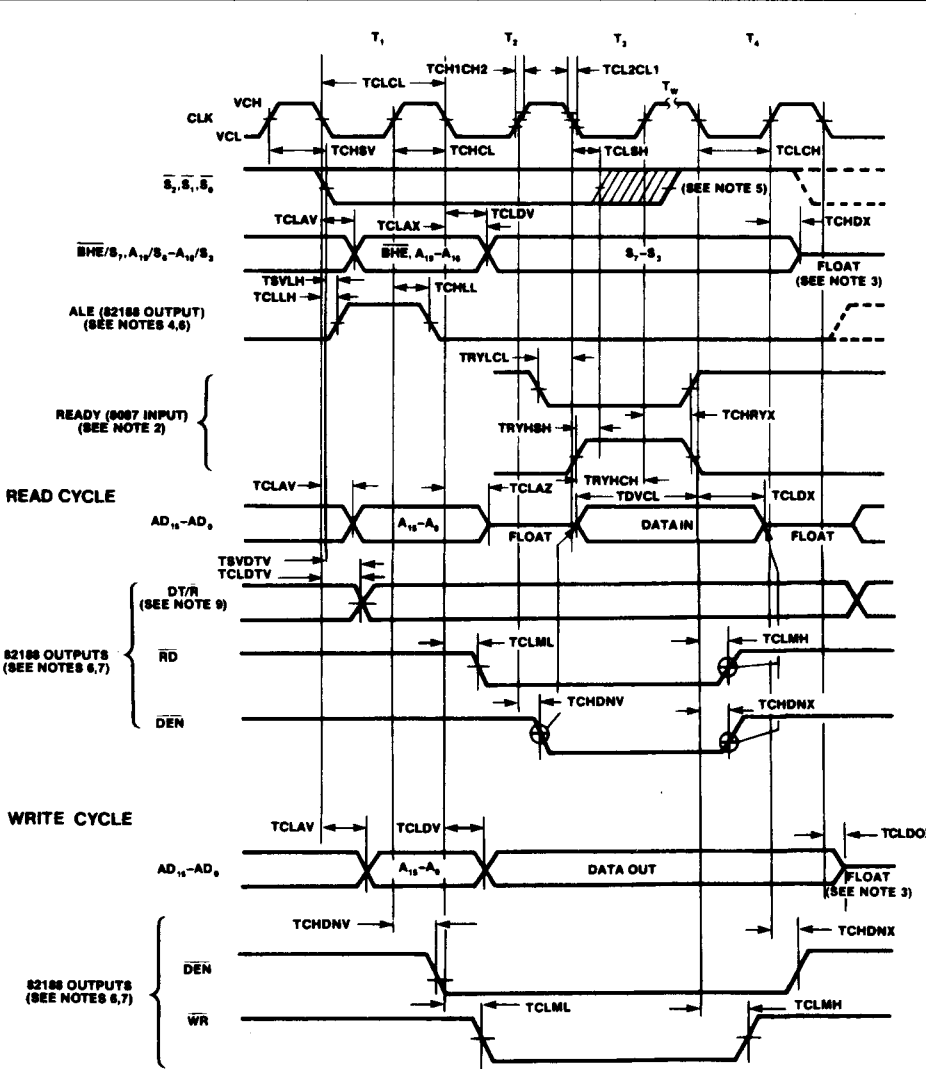### MASTER MODE (with 82188 references)

205835-11

NOTES:
1. All signals switch between V_{OL} and V_{OH} unless otherwise specified.
2. READY is sampled near the end of T_2, T_3 and T_W to determine if T_W machine states are to be inserted.
3. The local bus floats only if the 8087 is returning control to the 80186/80188.
4. ALE rises at later of (TSVLH, TCLLH).
5. Status inactive in state just prior to T_4.
6. Signals at 8284A or 82188 are shown for reference only.
7. The issuance of 8288 command and control signals (MRDC, (MWTC, AMWC, and DEN) lags the active high 8288 CEN.
8. All timing measurements are made at 1.5V unless otherwise noted.
9. DT/R becomes valid at the later of (TSVDTV, TCLDTV).

intel®

WAVEFORMS (Continued)

PASSIVE MODE

205835–12

RESET TIMING

205835–13
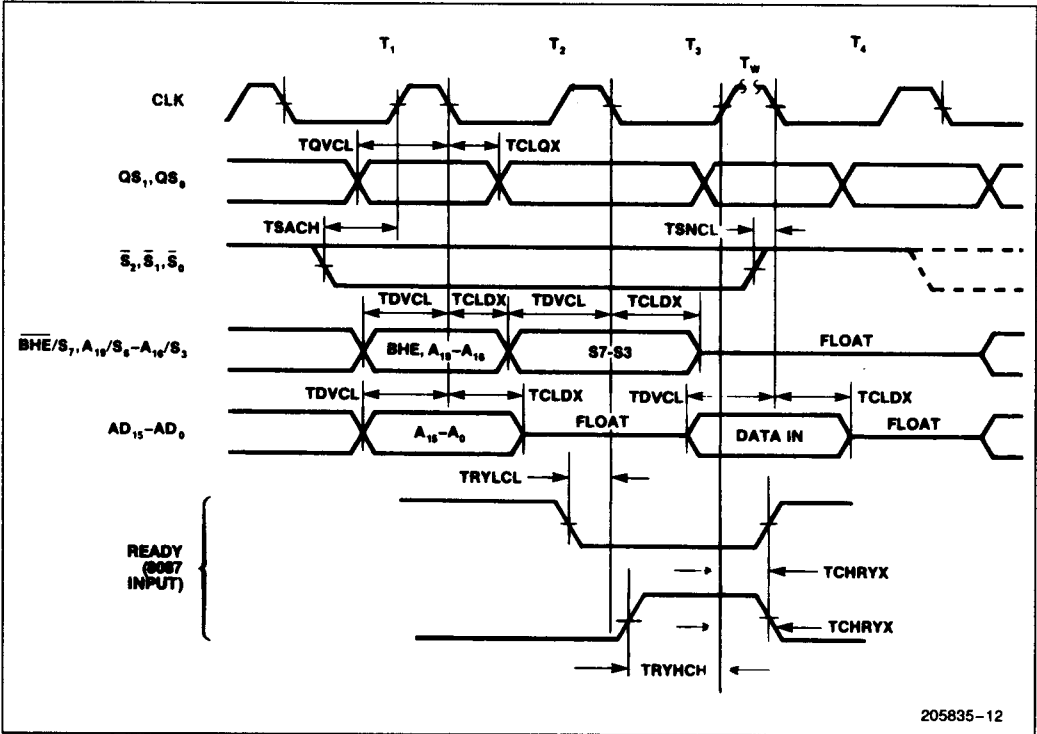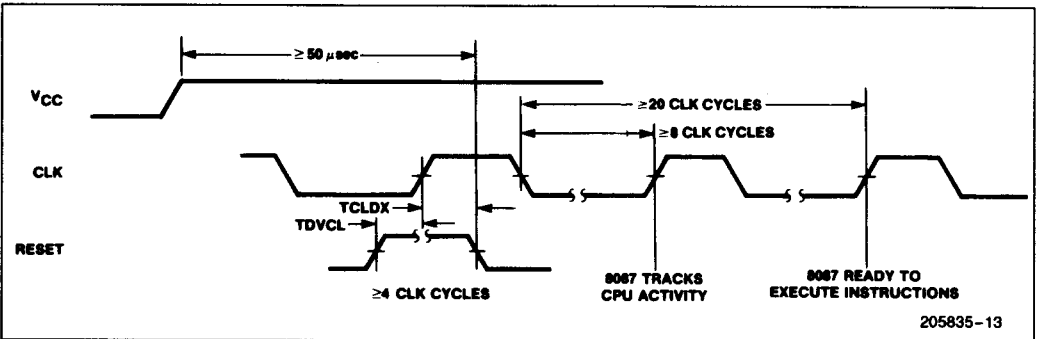
WAVEFORMS (Continued)

REQUEST/GRANT₀ TIMING

205835-14

NOTE:
The CPU provides active pullup of RQ/GT0, see TCLGH spec.

REQUEST/GRANT₁ TIMING

205835-15

NOTE:
Alternate master may not drive the buses outside of the region shown without risking bus contention.
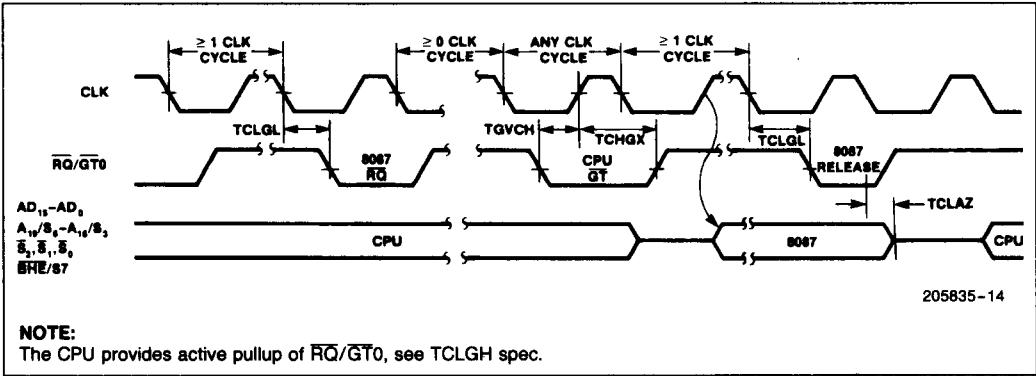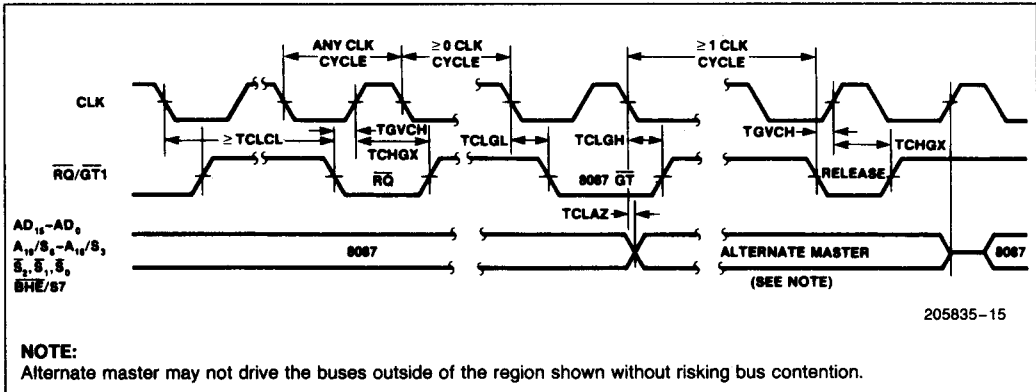
BUSY AND INTERRUPT TIMING

205835-16

Table 5. 8087 Extensions to the 86/186 Instructions Sets

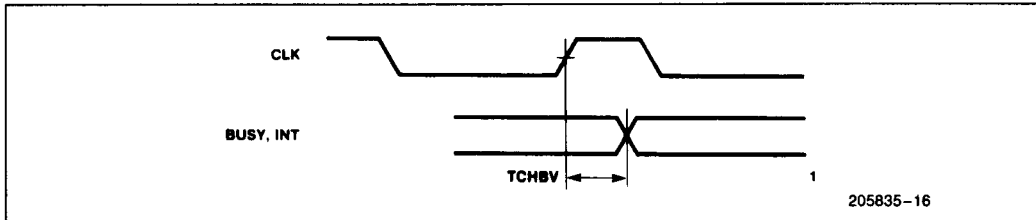| Data Transfer			Optional 8,16 Bit Displacement	Clock Count Range			
				32 Bit Real	32 Bit Integer	64 Bit Real	16 Bit Integer
FLD = LOAD	MF	=		00	01	10	11
Integer/Real Memory to ST(0)	ESCAPE MF 1	MOD 0 0 0 R/M	DISP	38–56 +EA	52–60 +EA	40–60 +EA	46–54 +EA
Long Integer Memory to ST(0)	ESCAPE 1 1 1	MOD 1 0 1 R/M	DISP	60–68 +EA			
Temporary Real Memory to ST(0)	ESCAPE 0 1 1	MOD 1 0 1 R/M	DISP	53–65 +EA			
BCD Memory to ST(0)	ESCAPE 1 1 1	MOD 1 0 0 R/M	DISP	290–310 +EA			
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 +EA	82–92 +EA	96–104 +EA	80–90 +EA
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 +EA	84–94 +EA	98–106 +EA	82–92 +EA
ST(0) to Long Integer Memory	ESCAPE 1 1 1	MOD 1 1 1 R/M	DISP	94–105 +EA			
ST(0) to Temporary Real Memory	ESCAPE 0 1 1	MOD 1 1 1 R/M	DISP	52–58 +EA			
ST(0) to BCD Memory	ESCAPE 1 1 1	MOD 1 1 0 R/M	DISP	520–540 +EA			
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

			Optional 8,16 Bit Displacement	32 Bit Real	32 Bit Integer	64 Bit Real	16 Bit Integer
FCOM = Compare							
Integer/Real Memory to ST(0)	ESCAPE MF 0	MOD 0 1 0 R/M	DISP	60–70 +EA	78–91 +EA	65–75 +EA	72–86 +EA
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 +EA	80–93 +EA	67–77 +EA	74–88 +EA
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			

205835–17

Table 5. 8087 Extensions to the 86/186 Instructions Sets (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 +EA	108-143 +EA	95-125 +EA	102-137 +EA
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 +EA	108-143 +EA	95-125 +EA	102-137 +EA
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 +EA	130-144 +EA	112-168 +EA	124-138 +EA
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 +EA	230-243 +EA	220-230 +EA	224-238 +EA
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			

205835-18

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

Table 5. 8087 Extensions to the 86/186 Instructions Sets (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

		Optional 8,16 Bit Displacement	Clock Count Range
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 = $ST(1) \cdot Log_2$ |ST(0)|	ESCAPE 0 0 1 | 1 1 1 1 0 0 0 1		900–1100
FYL2XP1 = $ST(1) \cdot Log_2$ [ST(0) +1]	ESCAPE 0 0 1 | 1 1 1 1 1 0 0 1		700–1000

Processor Control

		Optional 8,16 Bit Displacement	Clock Count Range
FINIT = Initialized 8087	ESCAPE 0 1 1 | 1 1 1 0 0 0 1 1		2–8
FENI = Enable Interrupts	ESCAPE 0 1 1 | 1 1 1 0 0 0 0 0		2–8
FDISI = Disable Interrupts	ESCAPE 0 1 1 | 1 1 1 0 0 0 0 1		2–8
FLDCW = Load Control Word	ESCAPE 0 0 1 | MOD 1 0 1 R/M	DISP	7–14 + EA
FSTCW = Store Control Word	ESCAPE 0 0 1 | MOD 1 1 1 R/M	DISP	12–18 + EA
FSTSW = Store Status Word	ESCAPE 1 0 1 | MOD 1 1 1 R/M	DISP	12–18 + EA
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 + EA
FLDENV = Load Environment	ESCAPE 0 0 1 | MOD 1 0 0 R/M	DISP	35–45 + EA
FSAVE = Save State	ESCAPE 1 0 1 | MOD 1 1 0 R/M	DISP	197–207 + EA
FRSTOR = Restore State	ESCAPE 1 0 1 | MOD 1 0 0 R/M	DISP	197–207 + EA
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

205835–19

Table 5. 8087 Extensions to the 86/186 Instructions Sets (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
FWAIT = CPU Wait for 8087	1 0 0 1 1 0 1 1	3 + 5n*
		205835–20

*n = number of times CPU examines TEST line before 8087 lowers BUSY.

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:** $0 \leq IST(0)I < (2 - \sqrt{2})/2$
                          $-\infty < ST(1) < \infty$
    For **FPTAN:** $0 \leq ST(0) \leq \pi/4$
    For **FPATAN:** $0 \leq ST(0) < ST(1) < +\infty$

3