



T-49-12-05

80287 MATH COPROCESSOR

- 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 Math CoProcessor is an extension to the Intel 80286 microprocessor architecture. When combined with the 80286 microprocessor the 80287 dramatically increases the processing speed of computer application software which utilize mathematical operations. This makes an ideal computer workstation platform for applications such as financial modeling and spreadsheets, CAD/CAM, or graphics.

The 80287 Math CoProcessor adds over seventy mnemonics to the 80286 microprocessor instruction set. Specific 80287 math operations include logarithmic arithmetic, exponential, and trigonometric functions. The 80287 supports integer, extended integer, floating point and BCD data formats, and fully conforms to the ANSI/IEEE floating point standard.

The 80286/80287 is object code compatible with the 8086/8087 and 8088/8087. The 80287 is fabricated with HMOS III technology and available in a 40-pin cerdip packages. A CMOS 80C287A math coprocessor is available for higher speed or low power applications.

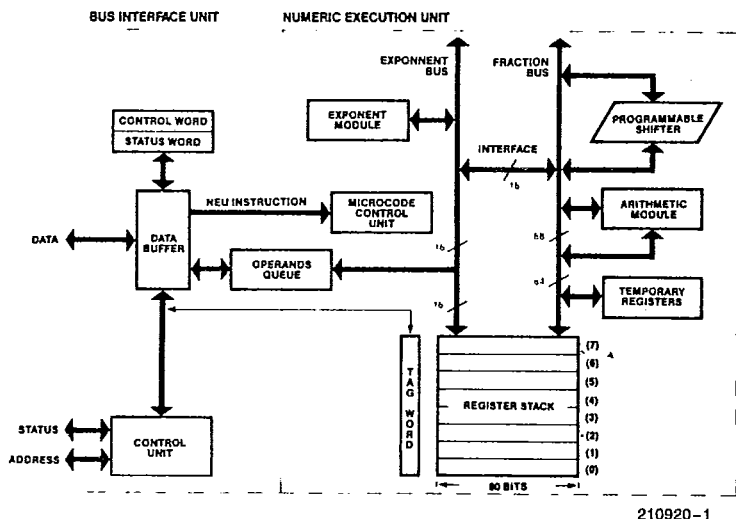
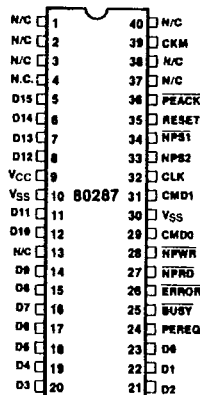


Figure 1. 80287 Block Diagram



NOTE:

N/C Pins should not be connected

Figure 2.
80287 Pin Configuration



80287

T-49-12-05

Table 1. 80287 Pin Description

Symbols	Type	Name and Function
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 μ s 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.
BUSY	O	BUSY STATUS: asserted by the 80287 to indicate that it is currently executing a command.
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 PEACK or upon actual data transfer, whichever occurs first, if no more transfers are required.
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. PEACK may be asynchronous to the 80287 clock.
NPRD	I	NUMERIC PROCESSOR READ: Enables transfer of data from the 80287. This input may be asynchronous to the 80287 clock.
NPWR	I	NUMERIC PROCESSOR READ: Enables transfer of data from the 80287. This input may be asynchronous to the 80287 clock.
NPS1, NPS2	I	NUMERIC PROCESSOR SELECTS: indicate the CPU is performing an ESCAPE instruction. Concurrent assertion of these signals (i.e., NPS1 is LOW and NPS2 is HIGH) enables the 80287 to perform floating point instructions. 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.



80287

T-49-12-05

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.

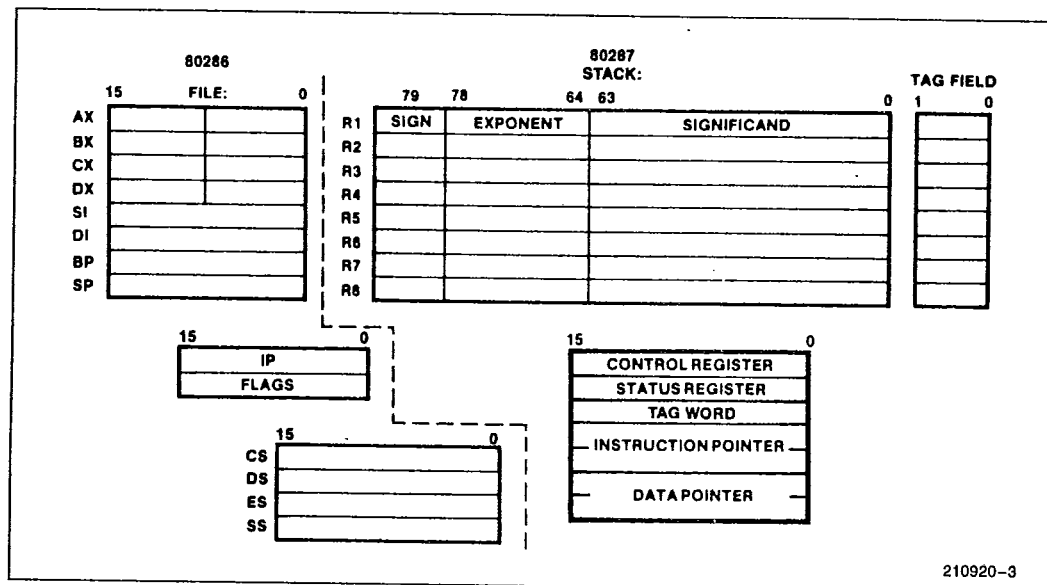


Figure 3. 80286/80287 Architecture

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.



80287

T-49-12-05

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 ($\overline{\text{PEREQ}}$, $\overline{\text{PEACK}}$), the $\overline{\text{BUSY}}$ signal and the $\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ 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 $\overline{\text{ERROR}}$ 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 ($\text{CKM} = 0$), 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 $\frac{1}{3}$ duty cycle clock on the CLK input ($\text{CKM} = 1$). 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 $\overline{\text{PEACK}}$ 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 $\overline{\text{PEREQ}}$ 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 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 $\overline{\text{M}/\overline{\text{IO}}}$. It may also be noted that the 80386 CPU automatically generates 16-bit bus cycles whenever it communicates with the 80287.

HARDWARE INTERFACE

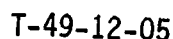
Communication of instructions and data operands between the 80286 and 80287 is handled by the $\overline{\text{CMD0}}$, $\overline{\text{CMD1}}$, $\overline{\text{NPS1}}$, $\overline{\text{NPS2}}$, $\overline{\text{NPRD}}$, and $\overline{\text{NPWR}}$ signals. I/O port addresses 00F8H, 00FAH, and 00FCH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



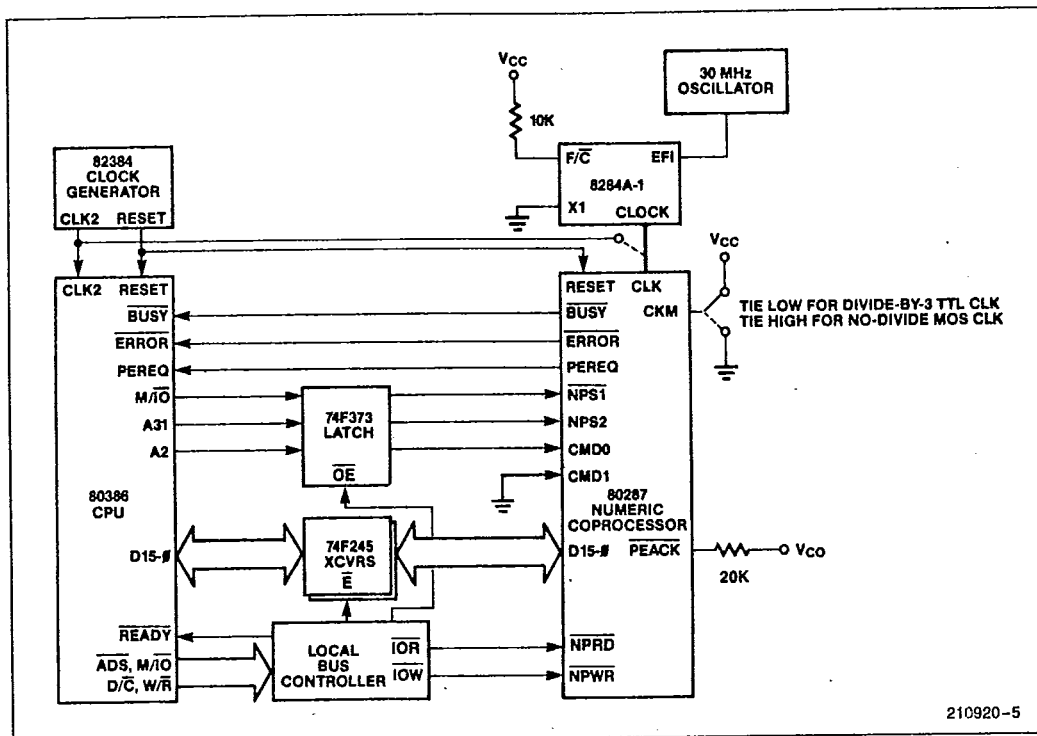
Internally the 80287 holds all numbers in the temporary real format. Load instructions automatically convert operands represented in memory as 16-, 32-, or 64-bit integers, 32- or 64-bit floating point number or

Figure 4A. 80286/80287 System Configuration

intel

80287

T-49-12-05



210920-5

Figure 4B. 80386/80287 System Configuration



80287

T-49-12-05

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
Word Integer	10^4	16 Bits														
Short Integer	10^9	32 Bits														
Long Integer	10^{19}	64 Bits														
Packed BCD	10^{18}	18 Digits														
Short Real	$10^{\pm 38}$	24 Bits														
Long Real	$10^{\pm 308}$	53 Bits														
Temporary Real	$10^{\pm 4932}$	64 Bits														

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. Δ = 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: $(-1)^s (D_{17} \dots D_0)$
8. Real: $(-1)^s (2^E \text{BIAS}) (F_0 F_1 \dots)$

210920-6

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 **BUSY** and **ERROR** signals for 80286/80287 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 <i>must</i> execute FNINIT before <i>any</i> 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 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 written or read by the NPX.

Data transfers between memory and the 80287, when needed, are controlled by the PEREQ, PEACK, NPRD, NPWR, NPST, 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.

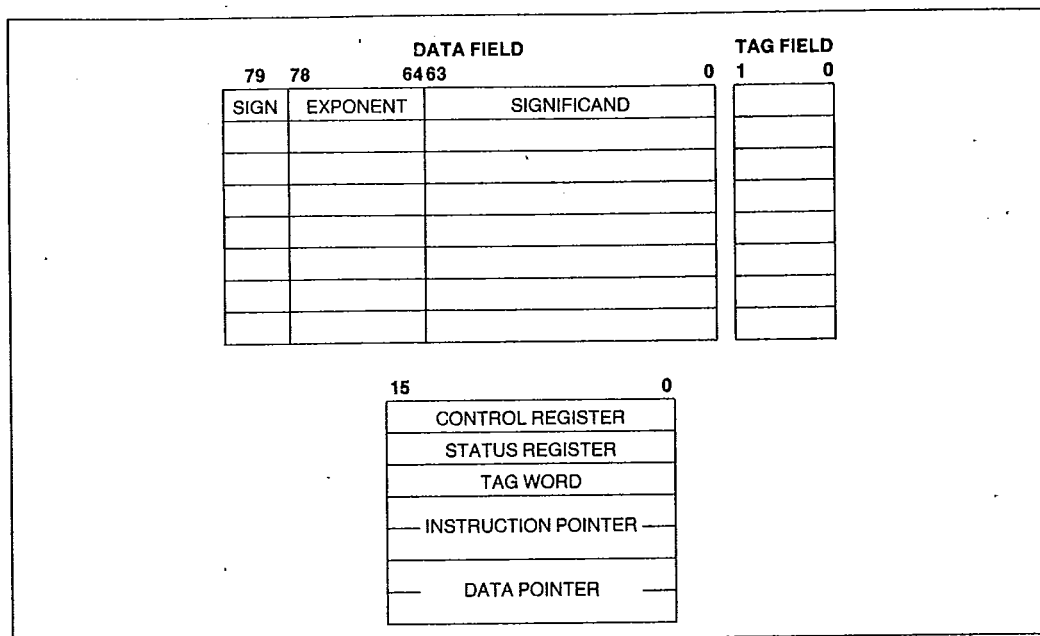
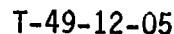
NUMERIC EXECUTION UNIT

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 significant 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 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



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 to 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 (C₀–C₃) 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.



80287

T-49-12-05

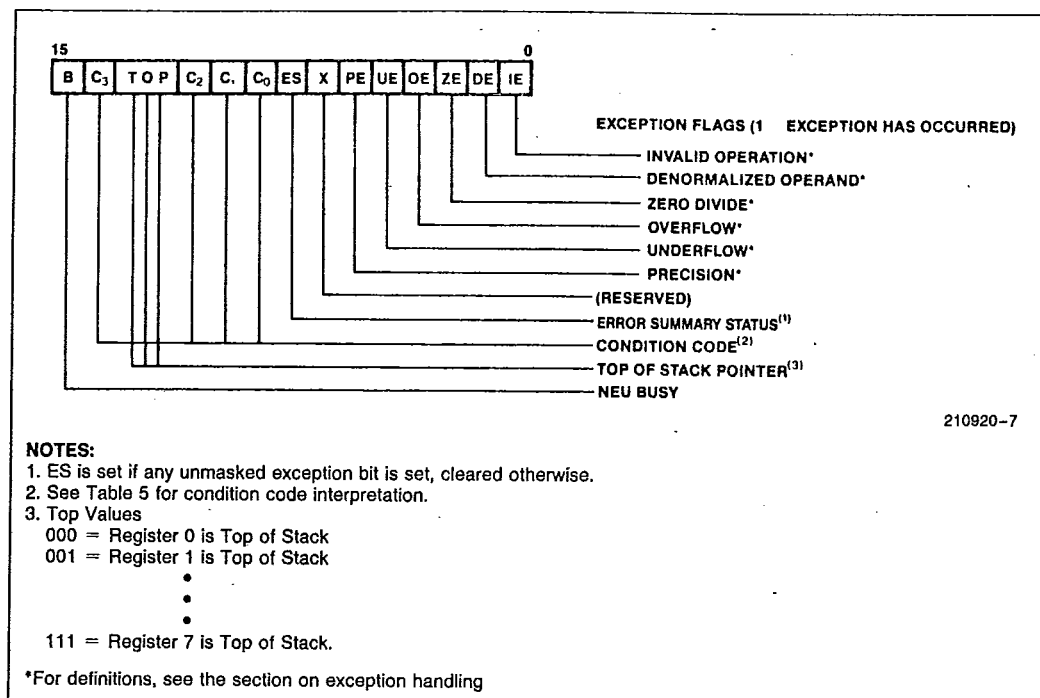


Figure 6. 80287 Status Word

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



80287

T-49-12-05

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: $\pm \infty$, or projective closure: ∞ , is treated as unsigned, may be specified).



80287

T-49-12-05

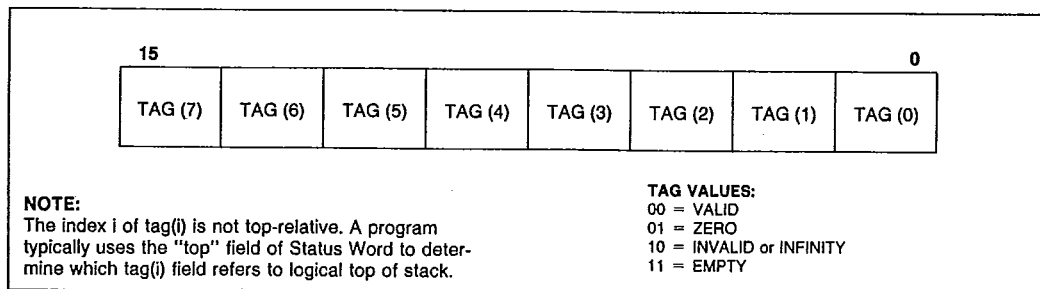


Figure 7. 80287 Tag Word

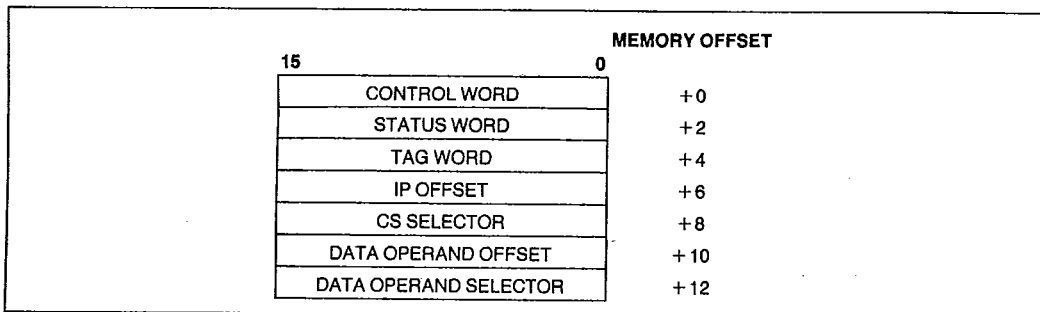


Figure 8a. Protected Mode 80287 Instruction and Data Pointer Image in Memory

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 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, ∞ , $-\infty$, 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 propagate 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 is 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.



80287

T-49-12-05

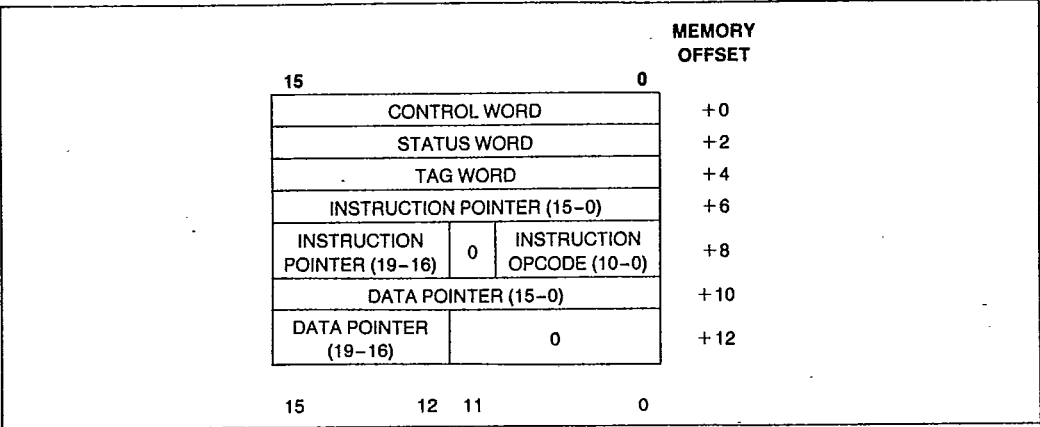


Figure 8b. Real Mode 80287 Instruction and Data Pointer Image in Memory

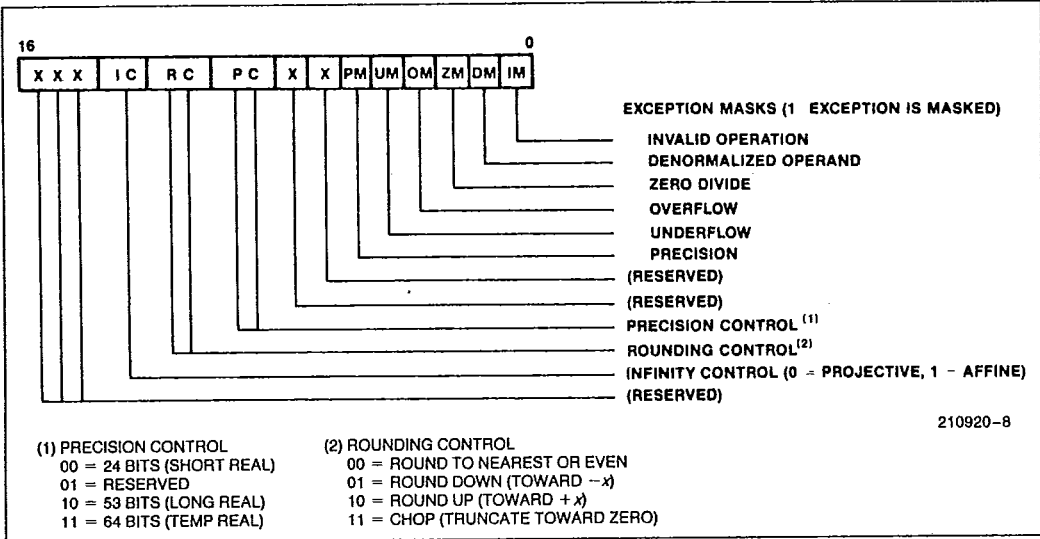


Figure 9. 80287 Control Word



80287

T-49-12-05

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 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 or 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 $T_A = 0^\circ\text{C}$ to 70°C , $T_C = 0^\circ\text{C}$ to 85°C , $V_{CC} = 5\text{V} \pm 5\%$ **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	$V_{CC} + 1$	V	
		3.8	$V_{CC} + 1$	V	
V_{ILC}	Clock Input LOW Voltage CKM = 1 CKM = 0	-0.5	0.8	V	
		-0.5	0.6	V	
V_{OL}	Output LOW Voltage		0.45	V	$I_{OL} = 3.0\text{ mA}$
V_{OH}	Output HIGH Voltage	2.4		V	$I_{OH} = -400\text{ }\mu\text{A}$
I_{LI}	Input Leakage Current	•	± 10	μA	$0\text{V} \leq V_{IN} \leq V_{CC}$
I_{LO}	Output Leakage Current	•	± 10	μA	$0.45\text{V} \leq V_{OUT} \leq V_{CC}$
I_{CC}	Power Supply Current	•	600	mA	$T_A = 0^\circ\text{C}$
			475	mA	$T_A = 25^\circ\text{C}$
			375	mA	$T_A = 70^\circ\text{C}$
C_{IN}	Input Capacitance	•	10	pF	$F_C = \text{MHz}$
C_O	Input/Output Capacitance (D0-D15)	•	20	pF	$V_C = 1\text{ MHz}$
C_{CLK}	CLK Capacitance	•	12	pF	$F_C = 1\text{ MHz}$



80287

T-49-12-05

A.C. CHARACTERISTICS $T_A = 0^\circ\text{C to } 70^\circ\text{C}$, $T_{\text{CASE}} = 0^\circ\text{C to } 85^\circ\text{C}$, $V_{\text{CC}} = 5\text{V} \pm 5\%$
TIMING REQUIREMENTS

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

Symbol	Parameter	80287-6 6 MHz		80287-8 8 MHz		80287-10 10 MHz		Units	Test Conditions
		Min	Max	Min	Max	Min	Max		
T_{CLCL}	CLK Period CKM = 1: CKM = 0:	166 62.5	500 166	125 50	500 166	100 40	500 166	ns ns	
T_{CLCH}	CLK LOW Time CKM = 1: CKM = 0:	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:	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	ns	1.0V to 3.6V if CKM = 0
T_{CL2CL1}	CLK Fall Time		10		10		10	ns	3.6V to 1.0V if CKM = 0
T_{DYWH}	Data Setup to NPWR Inactive	75		75		75		ns	
T_{WHDX}	Data Hold from NPWR Inactive	30		18		18		ns	
T_{WLWH} T_{RLRH}	NPWR NPRD Active Time	95		90		90		ns	At 0.8V
T_{AVWL} T_{AVRL}	Command Valid to NPWR or NPRD Active	0		0		0		ns	
T_{MHRL}	Minimum Delay from PEREQ Active to NPRD Active	130		130		100		ns	
T_{KLKH}	PEAK Active Time	85		85		60		ns	At 0.8V
T_{KHKL}	PEAK Inactive Time	250		250		200		ns	At 2.0V
T_{KHCH}	PEAK Inactive to NPWR, NPRD Inactive	50		40		40		ns	
T_{CHKL}	NPWR, NPRD Inactive to PEAK Active	-30		-30		-30		ns	
T_{WHAX} T_{RHAX}	Command Hold from NPWR, NPRD Inactive	30		30		22		ns	
T_{KLCL}	PEAK Active Setup to NPWR NPRD Active	50		40		40		ns	



80287

T-49-12-05

A.C. CHARACTERISTICS $T_A = 0^\circ\text{C to } 70^\circ\text{C}$, $T_{\text{CASE}} = 0^\circ\text{C to } 85^\circ\text{C}$, $V_{\text{CC}} = 5\text{V} \pm 5\%$ (Continued)

TIMING REQUIREMENTS (Continued)

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

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

TIMING RESPONSES

Symbol	Parameter	80287-6 6 MHz		80287-8 8 MHz		80287-10 10 MHz		Units	Test Conditions
		Min	Max	Min	Max	Min	Max		
T_{RHQZ}	NPRD Inactive to Data Float		37.5		35		21	ns	(Note 2)
T_{RLOV}	NPRD Active to Data Valid		60		60		60	ns	(Note 3)
T_{ILBH}	ERROR Active to BUSY Inactive	100		100		100		ns	(Note 4)
T_{WLBV}	NPWR Active to BUSY Active		100		100		100	ns	(Note 5)
T_{KLML}	PEAK Active to PEREQ Inactive		127		127		100	ns	(Note 6)
T_{CMDI}	Command Inactive Time								
	Write-to-Write	95		95		75		ns	At 2.0V
	Read-to-Read	95		95		75		ns	At 2.0V
	Write-to-Read	95		95		75		ns	At 2.0V
T_{RHQH}	Read-to-Write	95		95		75		ns	At 2.0V
	Data Hold from NPRD Inactive	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 I_{OSINF} : $X_L = 100\text{ pF}$.
4. BUSY loading: $C_L = 100\text{ pF}$.
5. BUSY loading: $C_L = 100\text{ pF}$.
6. On last data transfer on numeric instruction.
7. D0-D15 loading: $C_L = 100\text{ pF}$.

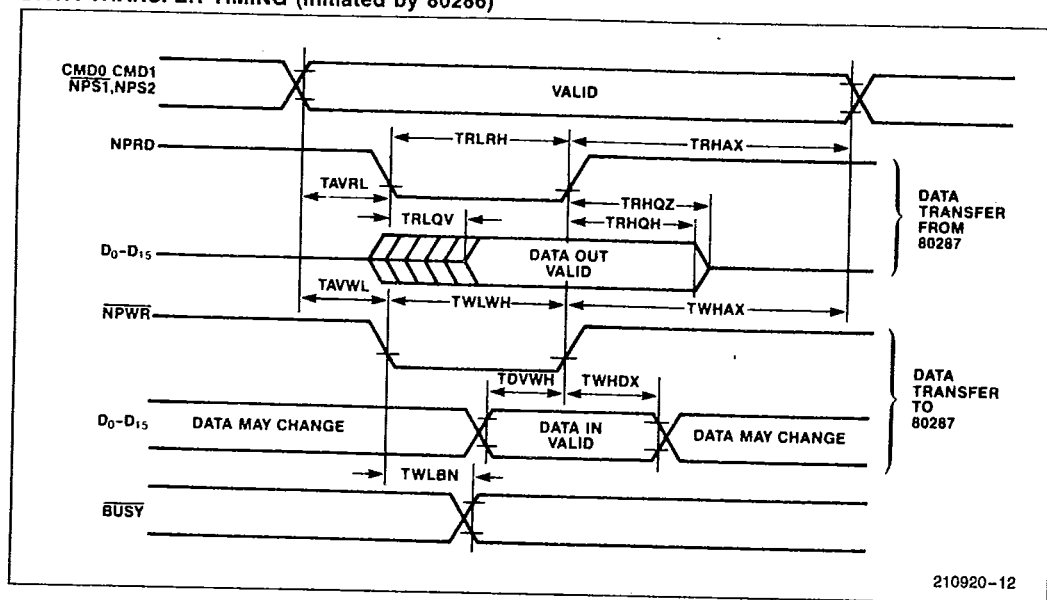


80287

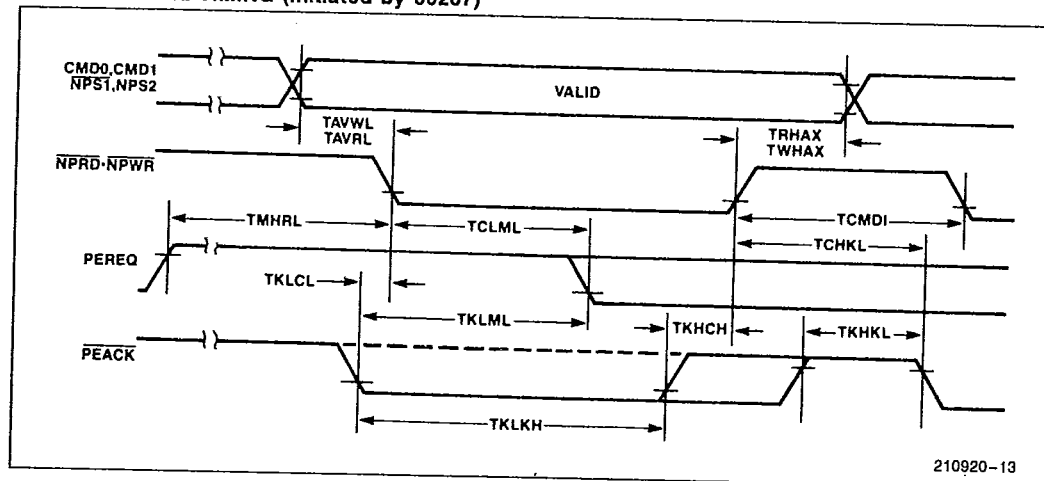
T-49-12-05

WAVEFORMS

DATA TRANSFER TIMING (Initiated by 80286)

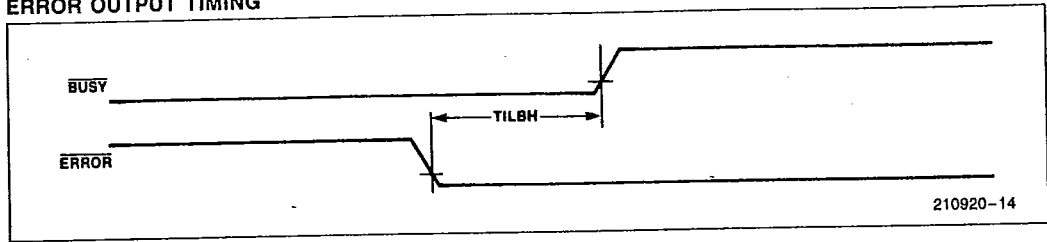


DATA CHANNEL TIMING (Initiated by 80287)

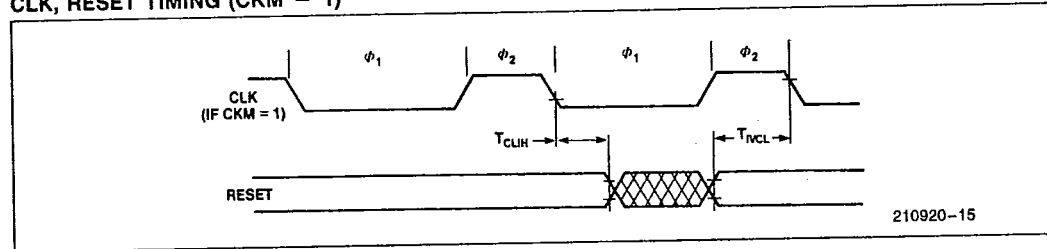


WAVEFORMS (Continued)

ERROR OUTPUT TIMING



CLK, RESET TIMING (CKM = 1)



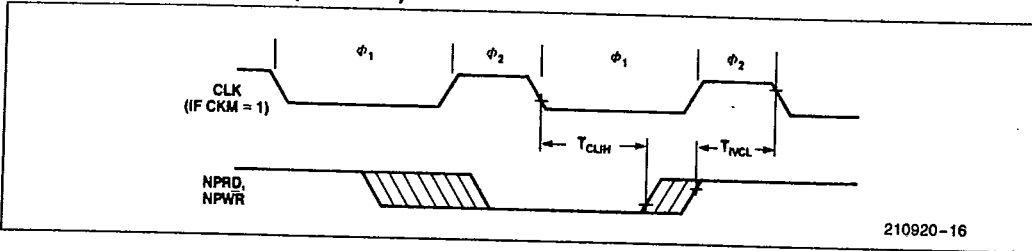
NOTE:
Reset, $\overline{\text{NPWR}}$, $\overline{\text{NPRD}}$ are inputs asynchronous to CLK. Timing requirements on this page are given for testing purposes only, to assure recognition at a specific CLK edge.



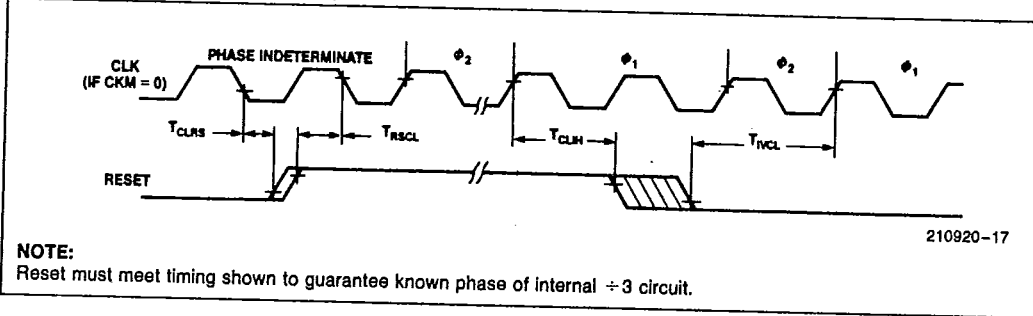
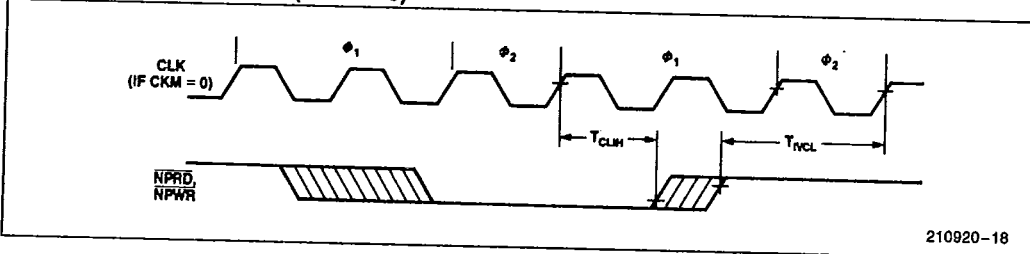
80287

T-49-12-05

WAVEFORMS (Continued)

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

CLK, RESET TIMING (CKM = 0)

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



80287

T-49-12-05

Table 6. 80287 Extensions to the 80286 Instruction Set

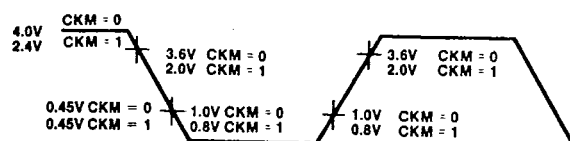
Data Transfer	Optional 8,16 Bit Displacement	Clock Count Range			
		32 Bit Real	32 Bit Integer	64 Bit Real	16 Bit Integer
FLD = LOAD	MF =	00	01	10	11
Integer/Real Memory to ST(0)	ESCAPE MF 1 MOD 0 0 0 R/M DISP	38-56	52-60	40-60	46-54
Long Integer Memory to ST(0)	ESCAPE 1 1 1 MOD 1 0 1 R/M DISP	60-68			
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



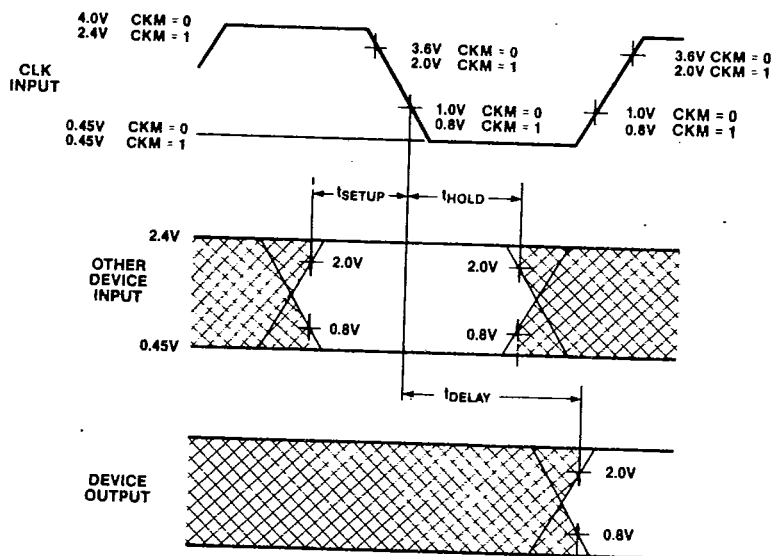
80287

T-49-12-05



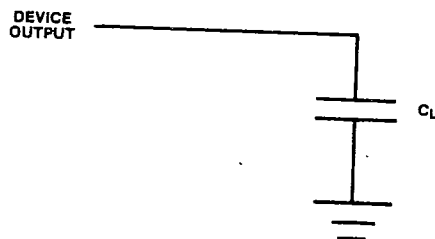
210920-9

AC Drive and Measurement Points—CLK Input



210920-10

AC Setup, Hold and Delay Time Measurement—General



210920-11

AC Test Loading on Outputs



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

Constants	MF	=	Optional 8,16 Bit Displacement	Clock Count Range			
				32 Bit Real	32 Bit Integer	64 Bit Real	16 Bit Integer
				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 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 $\log_2 2$ 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) $\div$ 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
FNCHS - 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) \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			
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:

- 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
- 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.
- MF = Memory Format
 00—32-bit Real
 01—32-bit Integer
 10—64-bit Real
 11—16-bit Integer
- ST(0) = Current stack top
 ST(i) = ith register below stack top
- d = Destination
 0—Destination is ST(0)
 1—Destination is ST(i)
- P = Pop
 0—No pop
 1—Pop ST(0)
- R = Reverse: When d = 1 reverse the sense of R
 0—Destination (op) Source
 1—Source (op) Destination
- For FSQRT: $-0 \leq \text{ST}(0) \leq +\infty$
 For FSCALE: $-2^{15} \leq \text{ST}(1) < +2^{15}$ and ST(1) integer
 For F2XM1: $0 \leq \text{ST}(0) \leq 2^{-1}$
 For FYL2X: $0 < \text{ST}(0) < \infty$
 $-\infty < \text{ST}(1) < +\infty$
 For FYL2XP1: $0 \leq |\text{ST}(0)| < (2 - \sqrt{2})/2$
 $-\infty < \text{ST}(1) < \infty$
 For FPTAN: $0 \leq \text{ST}(0) \leq \pi/4$
 For FPATAN: $0 \leq \text{ST}(0) < \text{ST}(1) < +\infty$
- ESCAPE bit pattern is 11011.