intel

Intel287™ XL/XLT MATH COPROCESSOR

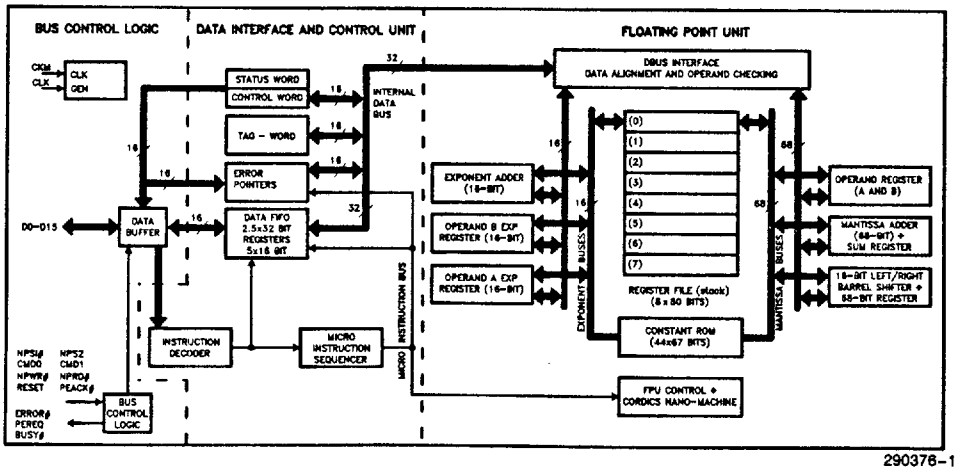
- **Interfaces with 80286 and 80C286 CPUs**
- **Operates in Any Socket Designed for Intel 80287 or Intel287™ XL MCP up to 12.5 MHz Clock Speeds**
- **Implements ANSI/IEEE Standard 754-1985 for Binary Floating-Point Arithmetic**
- **50% Higher Performance than Intel 80287**
- **Low Power CHMOS III Technology**
- **Upward Object Code Compatible from Intel 80287 and 8087**
- **Expands Data Types to Include 32-, 64-, 80-Bit Floating Point, or Integers, and 18 Digit BCD Operands**
- **Extends CPU Instruction Set to Include Tigonometric, Logarithmic, Exponential, and Arithmetic Instruction**
- **Implements Intel387™ Transcendental Operations for SINE, COSINE, TANGENT ARCTANGENT and LOGARITHM**
- **Eight 80-Bit Numeric Registers; for Stack use or Individual Access**
- **Available in 40-pin DIP as Intel287™ XL MCP and 44-pin PLCC as Intel287™ XLT MCP**

(See Packaging Outlined and Dimensions, order #231369)

The Intel287 XL Math CoProcessor is an extension of the Intel 80286 microprocessor architecture. When combined with an 80286 microprocessor, the Intel287 XL MCP dramatically increases the processing speed of computer application software which utilize floating point mathematical operations. This makes an ideal addition to a computer workstation platform for applications such as financial modeling and spreadsheets, CAD/CAM, or business graphics.

The Intel287 XL Math CoProcessor adds over seventy mnemonics to the Intel 80286 microprocessor instruction set. The Intel287 XL MCP is compatible with the Intel 80287 and 8087 Math CoProcessors. The Intel287 XL MCP increases performance by over 50% in typical floating-point tests, such as a Whetstone test, compared to the Intel 80287. The Intel287 XL MCP supports integer, floating point and BCD data formats and fully conforms to the ANSI/IEEE 754-1985 Floating Point Standard.

There are two versions of Intel287 XL MCP: the Intel287 XL MCP in a 40-pin DIP package and the Intel287 XLT MCP in a 44-pin PLCC package for small footprint applications such as portable personal computers. Each supports a clock speed up to 12.5 MHz which enables operation in any Math CoProcessor socket designed for the Intel 80287-6/8/10 or Intel 80C287A-12. Both versions are manufactured with low-power, CHMOS III technology.



290376-1

Figure 0.1. Intel287™ XL MCP Block Diagram

October 1992
Order Number: 290376-003

Intel287™ XL/XLT Math CoProcessor

CONTENTS

CONTENTS

CONTENTS

PAGE

FIGURES

CONTENTS

PAGE

TABLES

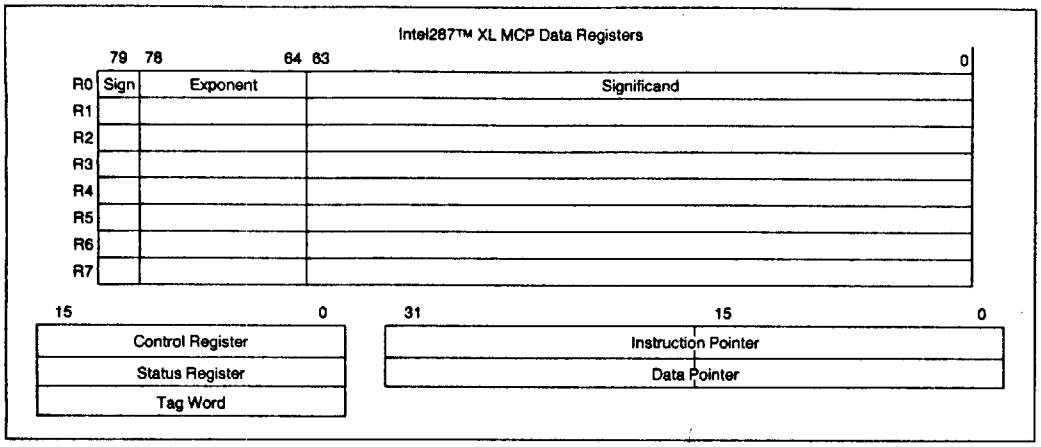


Figure 1.1. Intel287™ XL MCP Register Set

1.0 FUNCTIONAL DESCRIPTION

The Intel287 XL Math CoProcessor provides arithmetic instructions for a variety of numeric data types. It also executes numerous built-in transcendental functions (e.g. tangent, sine, cosine, and log functions). The Intel287 XL MCP effectively extends the register and instruction set of its CPU for existing data types and adds several new data types as well. Figure 1.1 shows the additional registers visible to programs in a system that includes the Intel287 XL MCP. Essentially, the Intel287 XL MCP can be treated as an additional resource or an extension to the CPU. The CPU together with an Intel287 XL Math CoProcessor can be used as a single unified system.

The Intel287 XL MCP has two operating modes. After reset, the Intel287 XL MCP is in the real-address mode. It can be placed into protected mode by executing the FSETPM instruction. It can be switched back to real-address mode by executing the FRSTPM instruction (note that this feature is useful only with CPU's that can also switch back to real-address mode). These instructions control the format of the administrative instructions FLDENV, FSTENV, FRSTOR, and FSAVE. Regardless of operating mode, all references to memory for numerics data or status information are performed by the CPU, and therefore obey the memory-management and protection rules of the CPU.

In real-address mode, a system that includes the Intel287 XL MCP is completely upward compatible with software for the 8086/8087 and for 80286/80287 or 80C287A real-address mode.

In protected mode, a system that includes the Intel287 XL MCP is completely upward compatible with software for 80286/80287 or 80C287A protected mode systems. The only differences of operation that may appear when 8086/8087 programs are ported to a protected-mode Intel287 XL MCP system are in the format of operands for the administrative instructions FLDENV, FSTENV, FRSTOR, and FSAVE. These instructions are normally used only by exception handlers and operating systems, not by applications programs.

2.0 PROGRAMMING INTERFACE

The Intel287 XL MCP adds to the CPU additional data types, registers, instructions, and interrupts specifically designed to facilitate high-speed numerics processing. To use the Intel287 XL MCP requires no special programming tools, because all new instructions and data types are directly supported by the assembler and compilers for high-level languages. All 8086/8088 development tools that support the 8087 can also be used to develop software for the 80286/Intel287 XL MCP in real-address mode. All 80286 development tools that support the 80287/80C287A can also be used to develop software for the 80286/Intel287 XL MCP and 80C286/Intel287 XL MCP. The Intel287 XL MCP supports all 80387 instructions, producing the same binary results.

All communication between the CPU and the Intel287 XL MCP is transparent to applications software. The CPU automatically controls the Intel287 XL MCP whenever a floating point instruction is executed. All physical memory and virtual memory of the CPU are available for storage of the instructions and operands of programs that use the Intel287 XL MCP. All memory addressing modes are available for addressing numerics operands.

Section 6 at the end of this data sheet lists the instructions that the Intel287 XL MCP adds to the 80286 instruction set.

2.1 Data Types

Table 2.1 lists the seven data types that the Intel287 XL MCP supports and presents the format for each type. Operands are stored in memory with the least significant digit at the lowest memory address. Programs retrieve these values by generating the lowest address. For maximum system performance, all operands should start at physical-memory addresses that correspond to the word size of the CPU; operands may begin at any other addresses, but will require extra memory cycles to access the entire operand.

Internally, the Intel287 XL MCP holds all numbers in the extended-precision real format. Instructions that load operands from memory 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 extended-precision real format. Instructions that store operands in memory perform the inverse type conversion.

2.2 Numeric Operands

A typical MCP (Math CoProcessor) instruction accepts one or two operands and produces one (or sometimes two) results. In two-operand instructions, one operand is the contents of an MCP register, while the other may be a memory location. The operands of some instructions are predefined; for example, FSQRT always takes the square root of the number in the top stack element.

2.3 Register Set

Figure 1.1 shows the Intel287 XL MCP register set. When an Intel287 XL MCP is present in a system, programmers may use these registers in addition to the registers normally available on the CPU.

2.3.1 DATA REGISTERS

Intel287 XL MCP computations use the Intel287 XL MCP's data registers. These eight 80-bit registers provide the equivalent capacity of 20 32-bit registers. Each of the eight data registers in the Intel287 XL MCP is 80 bits wide and is divided into "fields" corresponding to the MCP's extended-precision real data type.

The Intel287 XL MCP register set can be accessed either as a stack, with instructions operating on the top one or two stack elements, or as individually addressable registers. The TOP field in the status word identifies the current top-of-stack register. A "push" operation decrements TOP by one 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 one. The Intel287 XL MCP 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 at which TOP points. Other instructions allow the programmer to explicitly specify which register to use. This explicit register addressing is also relative to TOP.

2.3.2 TAG WORD

The tag word marks the content of each numeric data register, as Figure 2.1 shows. Each two-bit tag represents one of the eight data registers. The principal function of the tag word is to optimize the MCP's performance and stack handling by making it possible to distinguish between empty and nonempty register locations. It also enables exception handlers to identify special values (e.g. NaNs or denormals) in the contents of a stack location without the need to perform complex decoding of the actual data.

2.3.3 STATUS WORD

The 16-bit status word (in the status register) shown in Figure 2.2 reflects the overall state of the Intel287 XL MCP. It may be read and inspected by programs.

Bit 15, the B-bit (busy bit) is included for 8087 compatibility only. It always has the same value as the ES bit (bit 7 of the status word); it does **not** indicate the status of the BUSY# output of Intel287 XL MCP.

Bits 13–11 (TOP) point to the Intel287 XL MCP register that is the current top-of-stack.

The four numeric condition code bits (C_3–C_0) are similar to the flags in a CPU; instructions that perform arithmetic operations update these bits to reflect the outcome. The effects of these instructions on the condition code are summarized in Tables 2.2 through 2.5.

Bit 7 is the error summary (ES) status bit. This bit is set if any unmasked exception bit is set; it is clear otherwise. If this bit is set, the ERROR# signal is asserted.

Bit 6 is the stack flag (SF). This bit is used to distinguish invalid operations due to stack overflow or underflow from other kinds of invalid operations. When SF is set, bit 9 (C_1) distinguishes between stack overflow ($C_1 = 1$) and underflow ($C_1 = 0$).

Table 2.1. Intel287™ XL MCP Data Type Representation in Memory

Data Formats	Range	Precision	Most Significant Byte ... HIGHEST ADDRESSED BYTE
Word Integer	$\pm 10^4$	16 Bits	(TWO'S COMPLEMENT)
Short Integer	$\pm 10^9$	32 Bits	(TWO'S COMPLEMENT)
Long Integer	$\pm 10^{18}$	64 Bits	(TWO'S COMPLEMENT)
Packed BCD	$\pm 10^{18}$	18 Digits	S, X, MAGNITUDE
Single Precision	$\pm 10^{\pm 38}$	24 Bits	S, BIASED EXPONENT, SIGNIFICAND
Double Precision	$\pm 10^{\pm 308}$	53 Bits	S, BIASED EXPONENT, SIGNIFICAND
Extended Precision	$\pm 10^{\pm 4932}$	64 Bits	S, BIASED EXPONENT, I, SIGNIFICAND

290376-2

NOTES:
1. S = Sign bit (0 = positive, 1 = negative)
2. d_n = Decimal digit (two per byte)
3. X = Bits have no significance: Intel287 XL MCP ignores when loading, zeroes when storing
4. ▲ = Position of implicit binary point
5. I = Integer bit of significand; stored in temporary real, implicit in single and double precision
6. Exponent Bias (normalized values):
Single: 127 (7FH)
Double: 1023 (3FFH)
Extended Real: 16383 (3FFFH)
7. Packed BCD: $(-1)^S (D_{17} \ldots D_0)$
8. Real: $(-1)^S (2^{E\text{-BIAS}}) (F_0 F_1 \ldots)$

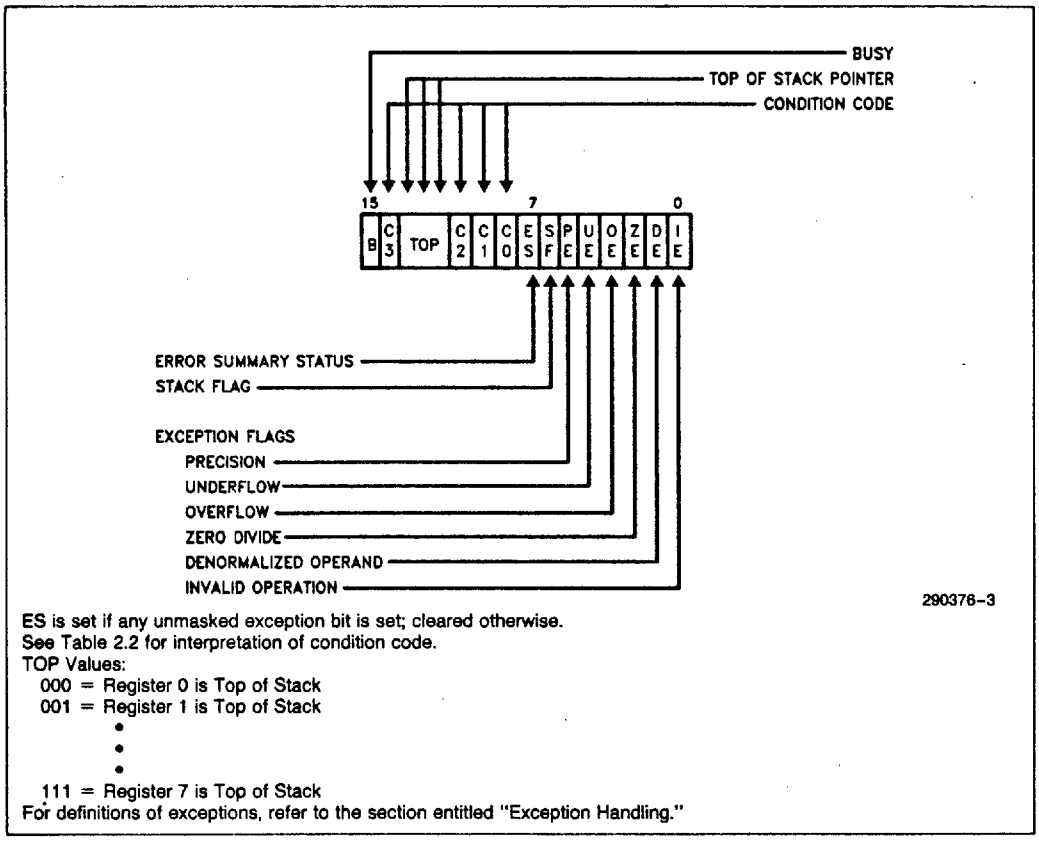
15							0
TAG (7)	TAG (6)	TAG (5)	TAG (4)	TAG (3)	TAG (2)	TAG (1)	TAG (0)

NOTE:
The index i of tag(i) is not top-relative. A program typically uses the "top" field of Status Word to determine which tag(i) field refers to logical top of stack.
TAG VALUES:
00 = Valid
01 = Zero
10 = QNaN, SNaN, Infinity, Denormal and Unsupported Formats
11 = Empty

Figure 2.1. Intel287™ XL MCP Tag Word

Figure 2.2 shows the six exception flags in bits 5–0 of the status word. Bits 5–0 are set to indicate that the Intel287 XL MCP has detected an exception while executing an instruction. A later section entitled "Exception Handling" explains how they are set and used.

Note that when a new value is loaded into the status word by the FLDENV or FRSTOR instruction, the value of ES (bit 7) and its reflection in the B-bit (bit 15) are not derived from the values loaded from memory but rather are dependent upon the values of the exception flags (bits 5–0) in the status word and their corresponding masks in the control word. If ES is set in such a case, the ERROR# output of the Intel287 XL MCP is activated immediately.



ES is set if any unmasked exception bit is set; cleared otherwise.
See Table 2.2 for interpretation of condition code.
TOP Values:
000 = Register 0 is Top of Stack
001 = Register 1 is Top of Stack
•
•
•
111 = Register 7 is Top of Stack
For definitions of exceptions, refer to the section entitled "Exception Handling."

Figure 2.2. Status Word

Table 2.2. Condition Code Interpretation

Instruction	C0 (S)	C3 (Z)	C1 (A)	C2 (C)
FPREM, FPREM1 (See Table 2.3)	Three Least Significant Bits of Quotient Q2	Q0	Q1 or O/U#	Reduction 0 = Complete 1 = Incomplete
FCOM, FCOMP, FCOMPP, FTST, FUCOM, FUCOMP, FUCOMPP, FICOM, FICOMP	Result of Comparison (See Table 2.4)		Zero or O/U#	Operand is Not Comparable (Table 2.4)
FXAM	Operand Class (See Table 2.5)		Sign or O/U#	Operand Class (Table 2.5)
FCHS, FABS, FXCH, FINCTOP, FDECTOP, Constant Loads, FXTRACT, FLD, FILD, FBLD, FSTP (Ext Real)	UNDEFINED		Zero or O/U#	UNDEFINED
FIST, FBSTP, FRNDINT, FST FSTP, FADD, FMUL, FDIV, FDIVR, FSUB, FSUBR, FSCALE, FSQRT, FPATAN, F2XM1, FYL2X, FYL2XP1	UNDEFINED		Roundup or O/U#	UNDEFINED
FPTAN, FSIN, FCOS, FSINCOS	UNDEFINED		Roundup or O/U# Undefined if C2 = 1	Reduction 0 = Complete 1 = Incomplete
FLDENV, FRSTOR	Each Bit Loaded from Memory			
FLDCW, FSTENV, FSTCW, FSTSW, FCLEX, FINIT, FSAVE	UNDEFINED			

O/U# When both IE and SF bits of status word are set, indicating a stack exception, this bit distinguishes between stack overflow (C1 = 1) and underflow (C1 = 0).

Reduction If FPREM or FPREM1 produces a remainder that is less than the modulus, reduction is complete. When reduction is incomplete the value at the top of the stack is a partial remainder, which can be used as input to further reduction. For FPTAN, FSIN, FCOS, and FSINCOS, the reduction bit is set if the operand at the top of the stack is too large. In this case the original operand remains at the top of the stack.

Roundup When the PE bit of the status word is set, this bit indicates whether one was added to the least significant bit of the result during the last rounding.

UNDEFINED Do not rely on finding any specific value in these bits.

Table 2.3. Condition Code Interpretation after FPREM and FPREM1 Instructions

Condition Code				Interpretation after FPREM and FPREM1	
C2	C3	C1	C0		
1	X	X	X	**Incomplete Reduction:** Further iteration required for complete reduction.	
	Q1	Q0	Q2	Q MOD 8	**Complete Reduction:** C0, C3, C1 contain three least significant bits of quotient.
0	0	0	0	0	
	0	1	0	1	
	1	0	0	2	
	1	1	0	3	
	0	0	1	4	
	0	1	1	5	
	1	0	1	6	
	1	1	1	7	

Table 2.4. Condition Code Resulting from Comparison

Order	C3	C2	C0
TOP > Operand	0	0	0
TOP < Operand	0	0	1
TOP = Operand	1	0	0
Unordered	1	1	1

Table 2.5. Condition Code Defining Operand Class

C3	C2	C1	C0	Value at TOP
0	0	0	0	+Unsupported
0	0	0	1	+NaN
0	0	1	0	−Unsupported
0	0	1	1	−Nan
0	1	0	0	+Normal
0	1	0	1	+Infinity
0	1	1	0	−Normal
0	1	1	1	−Infinity
1	0	0	0	+0
1	0	0	1	+Empty
1	0	1	0	−0
1	0	1	1	−Empty
1	1	0	0	+Denormal
1	1	1	0	−Denormal

2.3.4 CONTROL WORD

The MCP provides several processing options that are selected by loading a control word from memory into the control register. Figure 2.3 shows the format and encoding of fields in the control word.

The low-order byte of this control word configures exception masking. Bits 5–0 of the control word contain individual masks for each of the six exceptions that the Intel287 XL MCP recognizes.

The high-order byte of the control word configures the Intel287 XL MCP operating mode, including precision, rounding, and infinity control.

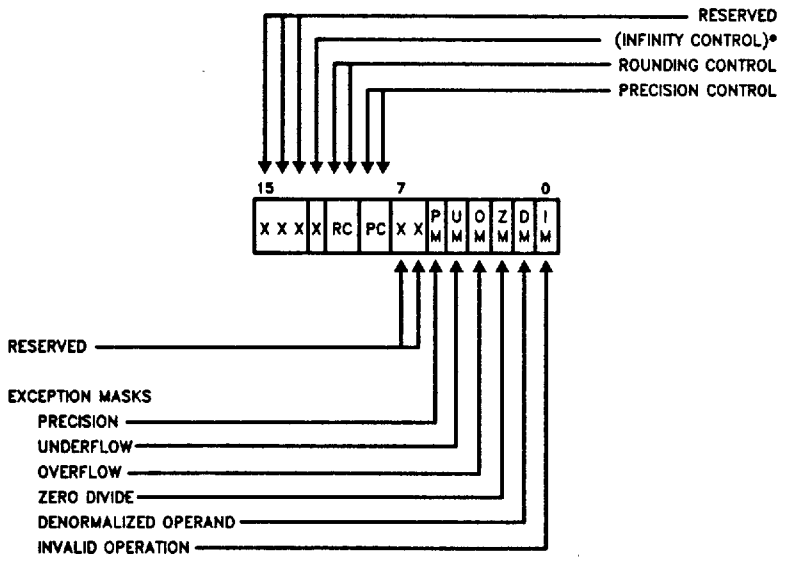
- The "infinity control bit" (bit 12) is not meaningful to the Intel287 XL MCP, and programs must ignore its value. To maintain compatibility with the 8087 and 80287, this bit can be programmed; however, regardless of its value, the Intel287 XL MCP always treats infinity in the affine sense ($-\infty < +\infty$). This bit is initialized to zero both after a hardware reset and after the FINIT instruction.
- The rounding control (RC) 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. Rounding control affects only those instructions that perform rounding at the end of the operation (and thus can generate a precision exception); namely, FST, FSTP, FIST, all arithmetic instructions (except FPREM, FPREM1, FXTRACT, FABS, and FCHS), and all transcendental instructions.
- The precision control (PC) bits (bits 9–8) can be used to set the Intel287 XL MCP internal operating precision of the significand at less than the default of 64 bits (extended precision). This can be useful in providing compatibility with early generation arithmetic processors of smaller precision. PC affects only the instructions ADD, SUB, DIV, MUL, and SQRT. For all other instructions, either the precision is determined by the opcode or extended precision is used.

2.3.5 INSTRUCTION AND DATA POINTERS

Because the MCP operates in parallel with the CPU, any exceptions detected by the MCP may be reported after the CPU has executed the ESC instruction which caused it. To allow identification of the failing numeric instruction, the Intel287 XL MCP contains registers that aid in diagnosis. These registers supply the opcode of the failing numeric instruction, the address of the instruction, and the address of its numeric memory operand (if appropriate).

The instruction and data pointers are provided for user-written exception handlers. Whenever the Intel287 XL MCP executes a new ESC instruction, it saves the address of the instruction (including any prefixes that may be present), the address of the operand (if present), and the opcode. CPUs with 32-bit internal architectures contain 32-bit versions of these registers and do not use the contents of the MCP registers. This difference is not apparent to programmers, however.

The instruction and data pointers appear in one of four formats depending on the operating mode of the system (protected mode or real-address mode) and (for CPUs with 32-bit internal architectures) depending on the operand-size attribute in effect (32-bit operand or 16-bit operand). (See Figures 2.4, 2.5, 2.6, and 2.7.) The ESC instructions FLDENV, FSTENV, FSAVE, and FRSTOR are used to transfer these values between the registers and memory. Note that the value of the data pointer is *undefined* if the prior ESC instruction did not have a memory operand.



Precision Control
- 00–24 bits (single precision)
- 01–(reserved)
- 10–53 bits (double precision)
- 11–64 bits (extended precision)

Rounding Control
- 00–Round to nearest or even
- 01–Round down (toward $-\infty$)
- 10–Round up (toward $+\infty$)
- 11–Chop (truncate toward zero)

*The "infinity control" bit is not meaningful to the Intel287 XL MCP. To maintain compatibility with the 80287, this bit can be programmed; however, regardless of its value, the Intel287 XL MCP treats infinity in the affine sense ($-\infty < +\infty$).

Figure 2.3. Control Word

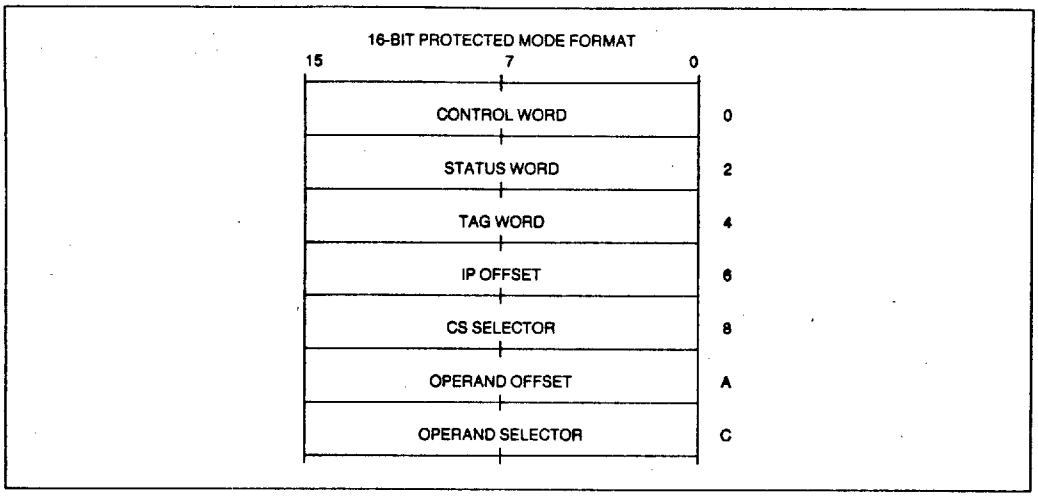


Figure 2-4. Instruction and Data Pointer Image in Memory, 16-bit Protected-Mode Format

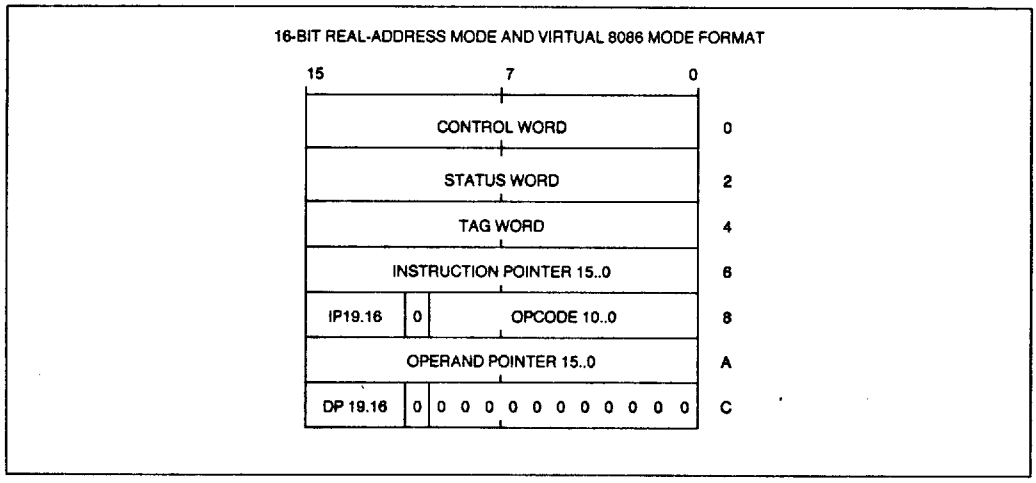


Figure 2-5. Instruction and Data Pointer Image in Memory, 16-bit Real-Mode Format

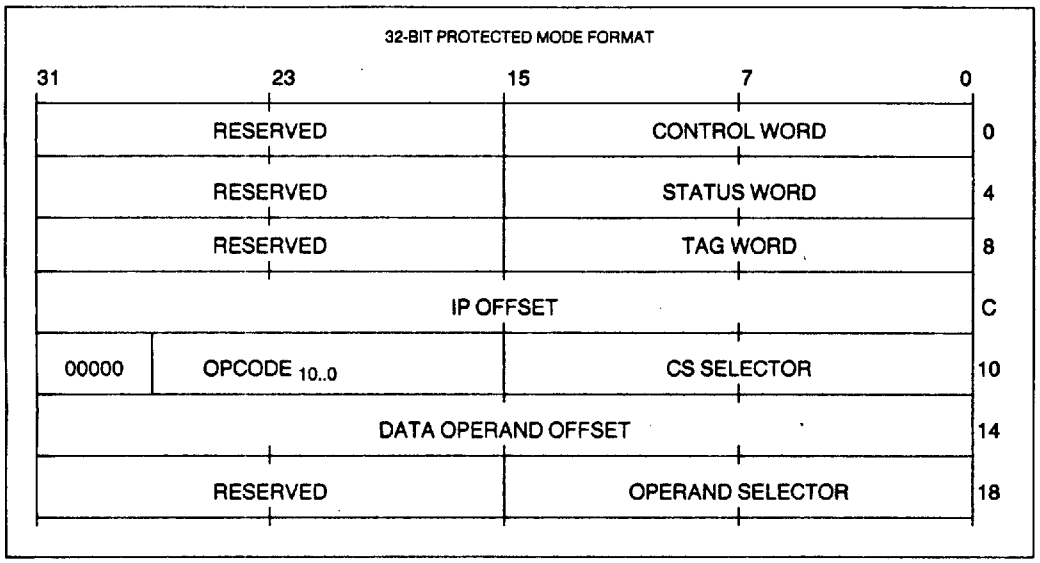


Figure 2-6. Instruction and Data Pointer Image in Memory, 32-bit Protected-Mode Format

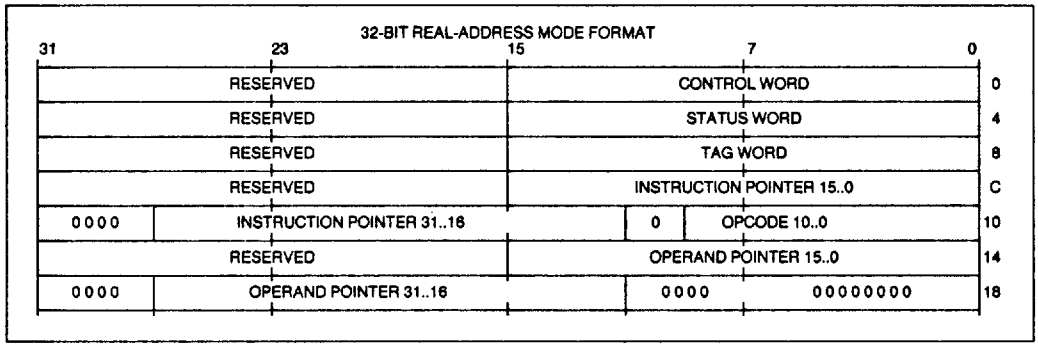


Figure 2-7. Instruction and Data Pointer Image in Memory, 32-bit Real-Mode Format

Table 2.6. CPU Interrupt Vectors Reserved for MCP

Interrupt Number	Cause of Interrupt
7	In a system with a CPU that has control registers, an ESC instruction was encountered when EM or TS of CPU control register zero (CR0) was set. EM = 1 indicates that software emulation of the instruction is required. When TS is set, either an ESC or WAIT instruction causes interrupt 7. This indicates that the current MCP context may not belong to the current task.
9	In a protected-mode system, an operand of a coprocessor instruction wrapped around an addressing limit (0FFFFH for expand-up segments, zero for expand-down segments) and spanned inaccessible addresses[1]. The failing numerics instruction is not restartable. The address of the failing numerics instruction and data operand may be lost; an FSTENV does not return reliable addresses. The segment overrun exception should be handled by executing an FNINIT instruction (i.e., an FINIT without a preceding WAIT). The exception can be avoided by never allowing numerics operands to cross the end of a segment.
13	In a protected-mode system, the first word of a numeric operand is not entirely within the limit of its segment. The return address pushed onto the stack of the exception handler points at the ESC instruction that caused the exception, including any prefixes. The Intel287 XL MCP has not executed this instruction; the instruction pointer and data pointer register refer to a previous, correctly executed instruction.
16	The previous numerics instruction caused an unmasked exception. The address of the faulty instruction and the address of its operand are stored in the instruction pointer and data pointer registers. Only ESC and WAIT instructions can cause this interrupt. The CPU return address pushed onto the stack of the exception handler points to a WAIT or ESC instruction (including prefixes). This instruction can be restarted after clearing the exception condition in the MCP. FNINIT, FNCLEX, FNSTSW, FNSTENV, and FNSAVE cannot cause this interrupt.

NOTE:
1. An operand may wrap around an addressing limit when the segment limit is near an addressing limit and the operand is near the largest valid address in the segment. Because of the wrap-around, the beginning and ending addresses of such an operand will be at opposite ends of the segment. There are two ways that such an operand may also span inaccessible addresses: 1) if the segment limit is not equal to the addressing limit (e.g. addressing limit is FFFFH and segment limit is FFFDH) the operand will span addresses that are not within the segment (e.g. an 8-byte operand that starts at valid offset FFFCH will span addresses FFFC–FFFFH and 0000–0003H; however addresses FFFEH and FFFFH are not valid, because they exceed the limit); 2) if the operand begins and ends in present and accessible segments but intermediate bytes of the operand fall in a not-present segment or page or in a segment or page to which the procedure does not have access rights.

2.4 Interrupt Description

CPU interrupts are used to report exceptional conditions while executing numeric programs in either real or protected mode. Table 2.6 shows these interrupts and their functions.

2.5 Exception Handling

The Intel287 XL MCP detects six different exception conditions that can occur during instruction execution. Table 2.7 lists the exception conditions in order of precedence, showing for each the cause and the

Table 2.7. Exceptions

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

default action taken by the Intel287 XL MCP if the exception is masked by its corresponding mask bit in the control word.

Any exception that is not masked by the control word sets the corresponding exception flag of the status word, sets the ES bit of the status word, and asserts the ERROR# signal. When the CPU attempts to execute another ESC instruction or WAIT, exception 16 occurs. The exception condition must be resolved via an interrupt service routine. The return address pushed onto the CPU stack upon entry to the service routine does not necessarily point to the failing instruction nor to the following instruction. The Intel287 XL MCP saves the address of the floating-point instruction that caused the exception and the address of any memory operand required by that instruction.

2.6 Initialization

After FNINIT or RESET, the control word contains the value 037FH (all exceptions masked, precision control 64 bits, rounding to nearest) the same values as in an 80287 after RESET. For compatibility with the 8087 and 80287, the bit that used to indicate infinity control (bit 12) is set to zero; however, regardless of its setting, infinity is treated in the affine sense. After FNINIT or RESET, the status word is initialized as follows:

- All exceptions are set to zero.
- Stack TOP is zero, so that after the first push the stack top will be register seven (111B).
- The condition code C_3–C_0 is **undefined**.
- The B-bit is zero.

The tag word contains FFFFH (all stack locations are empty).

80286/Intel287 XL MCP initialization software should execute an FNINIT instruction (i.e an FINIT without a preceding WAIT) after RESET. The FNINIT is not strictly required for either 80287, 80C287A or Intel287 XL MCP software, but Intel recommends its use to help ensure upward compatibility with other processors.

2.7 8087 and 80287 Compatibility

This section summarizes the differences between the Intel287 XL MCP and the 80287. Any migration from the 8087 directly to the Intel287 XL MCP must also take into account the differences between the 8087 and the 80287 as listed in the 80286 and 80287 Programmer's Reference Manual. There are no compatibility differences between the Intel287 XL MCP and 80C287A except the pinout configuration.

Many changes have been designed into the Intel287 XL MCP to directly support the IEEE standard in hardware. These changes result in increased performance by eliminating the need for software that supports the standard.

2.7.1 GENERAL DIFFERENCES

The Intel287 XL MCP supports only affine closure for infinity arithmetic, not projective closure.

Operands for FSCALE and FPATAN are no longer restricted in range (except for $\pm\infty$); F2XM1 and FPTAN accept a wider range of operands.

Rounding control is in effect for FLD *constant*.

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

After reset, FINIT, and incomplete FPREM, the Intel287 XL MCP resets to zero the condition code bits C_3–C_0 of the status word.

In conformance with the IEEE standard, the Intel287 XL MCP does not support the special data formats pseudozero, pseudo-NaN, pseudoinfinity, and unnormal.

The denormal exception has a different purpose on the Intel287 XL MCP. A system that uses the denormal-exception handler solely to normalize the denormal operands, would better mask the denormal exception on the Intel287 XL MCP. The Intel287 XL MCP automatically normalizes denormal operands when the denormal exception is masked.

2.7.2 EXCEPTIONS

A number of differences exist due to changes in the IEEE standard and to functional improvements to the architecture of the Intel287 XL MCP:

1. When the overflow or underflow exception is masked, the Intel287 XL MCP differs from the 80287 in rounding when overflow or underflow occurs. The Intel287 XL MCP produces results that are consistent with the rounding mode.
2. When the underflow exception is masked, the Intel287 XL MCP sets its underflow flag only if there is also a loss of accuracy during denormalization.
3. Fewer invalid-operation exceptions due to denormal operands, because the instructions FSQRT, FDIV, FPREM, and conversions to BCD or to integer normalize denormal operands before proceeding.
4. The FSQRT, FBSTP, and FPREM instructions may cause underflow, because they support denormal operands.
5. The denormal exception can occur during the transcendental instructions and the FXTRACT instruction.
6. The denormal exception no longer takes precedence over all other exceptions.
7. When the denormal exception is masked, the Intel287 XL MCP automatically normalizes denormal operands. The 8087/80287 performs unnormal arithmetic, which might produce an unnormal result.
8. When the operand is zero, the FXTRACT instruction reports a zero-divide exception and leaves $-\infty$ in ST(1).
9. The status word has a new bit (SF) that signals when invalid-operation exceptions are due to stack underflow or overflow.
10. FLD *extended precision* no longer reports denormal exceptions, because the instruction is not numeric.
11. FLD *single/double precision* when the operand is denormal converts the number to extended precision and signals the denormalized operand exception. When loading a signalling NaN, FLD *single/double precision* signals an invalid-operand exception.
12. The Intel287 XL MCP only generates quiet NaNs (as on the 80287); however, the Intel287 XL MCP distinguishes between quiet NaNs and signaling NaNs. Signaling NaNs trigger exceptions when they are used as operands; quiet NaNs do not (except for FCOM, FIST, and FBSTP which also raise IE for quiet NaNs).
13. When stack overflow occurs during FPTAN and overflow is masked, both ST(0) and ST(1) contain quiet NaNs. The 8087/80287 leaves the original operand in ST(1) intact.
14. When the scaling factor is $\pm\infty$, the FSCALE (ST(0), ST(1)) instruction behaves as follows (ST(0) and ST(1) contain the scaled and scaling operands respectively):
    - FSCALE(0,∞) generates the invalid operation exception.
    - FSCALE(finite, $-\infty$) generates zero with the same sign as the scaled operand.
    - FSCALE(finite, $+\infty$) generates $-\infty$ with the same sign as the scaled operand.

    The 8087/80287 returns zero in the first case and raises the invalid-operation exception in the other cases.

15. The Intel287 XL MCP returns signed infinity/zero as the unmasked response to massive overflow/underflow. The 8087 and 80287 support a limited range for the scaling factor; within this range either massive overflow/underflow do not occur or undefined results are produced.

3.0 HARDWARE INTERFACE

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

3.1 Signal Description

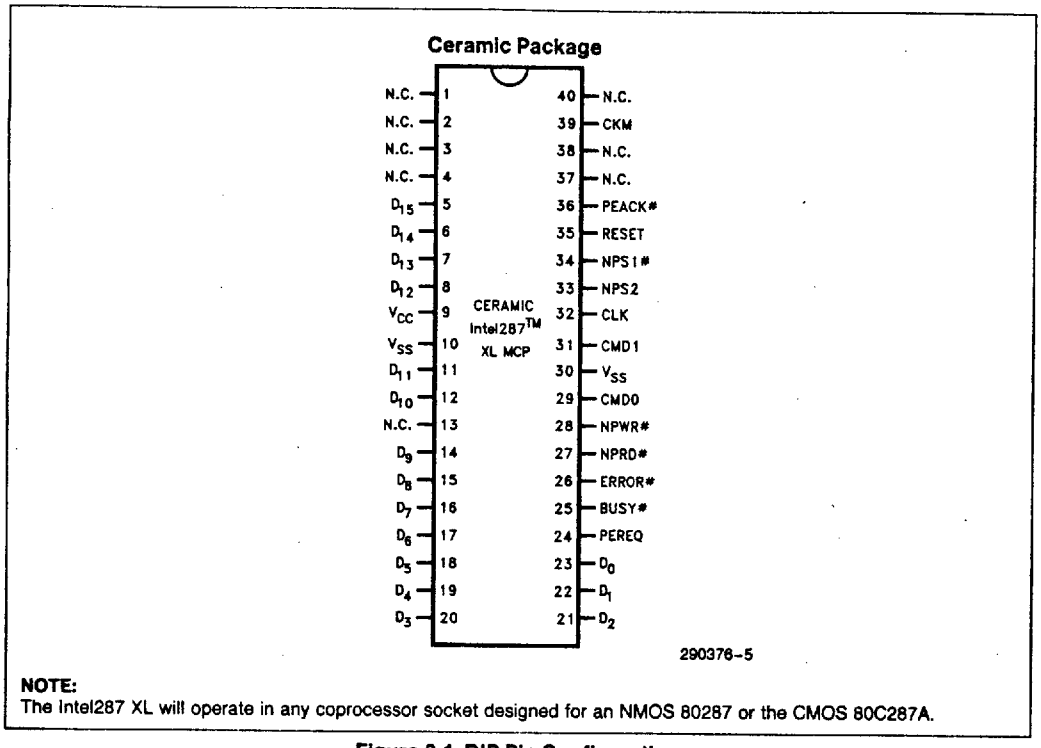
In the following signal descriptions, the Intel287 XL MCP pins are grouped by function as follows:

1. Execution control—CLK, CKM, RESET
2. MCP handshake—PEREQ, PEACK#, BUSY#, ERROR#
3. Bus interface pins—D_{15}–D_0, NPWR#, NPRD#
4. Chip/Port Select—NPS1#, NPS2, CMD0, CMD1
5. Power supplies—V_{CC}, V_{SS}

Table 3.1 lists every pin by its identifier, gives a brief description of its function, and lists some of its characteristics. Figure 3.1 shows the locations of pins on the Ceramic package, while Figure 3.2 shows the locations of pins on the PLCC package. Table 3.2 helps to locate pin identifiers in Figures 3.1 and 3.2.

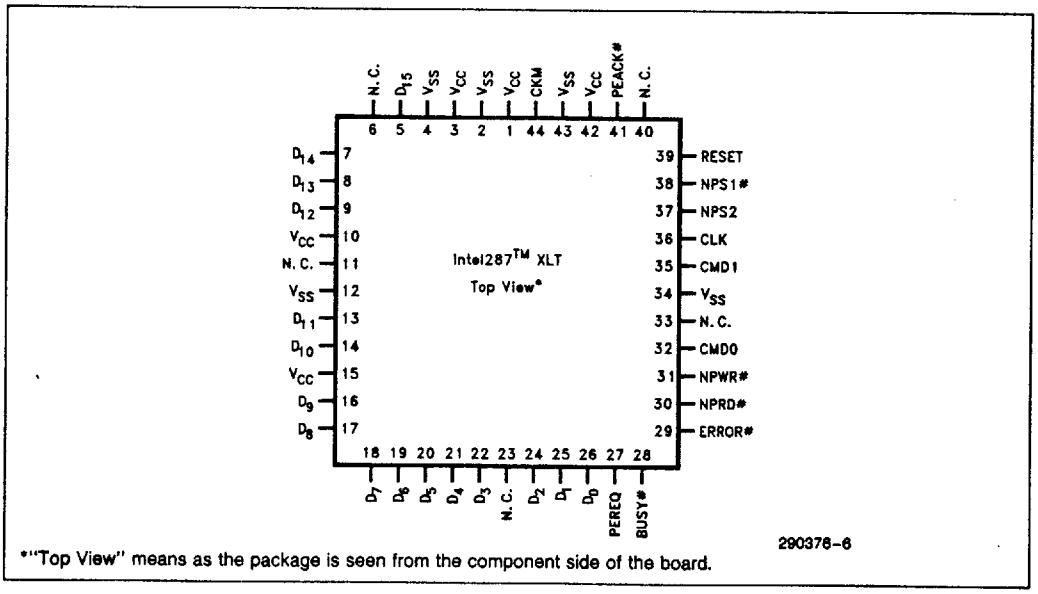
Table 3.1. Pin Summary

Pin Name	Function	Active State	Input/ Output
CLK	CLocK		I
CKM	ClocKing Mode		I
RESET	System reset	High	I
PEREQ	Processor Extension REQuest	High	O
PEACK#	Processor Extension ACKnowledge	Low	I
BUSY#	Busy status	Low	O
ERROR#	Error status	Low	O
D15–D0	Data pins	High	I/O
NPRD#	Numeric Processor ReaD	Low	I
NPWR#	Numeric Processor WRite	Low	I
NPS1#	MCP select #1	Low	I
NPS2	MCP select #2	High	I
CMD0	CoMmanD 0	High	I
CMD1	CoMmanD 1	High	I
V_{CC}	System power		I
V_{SS}	System ground		I



NOTE:
The Intel287 XL will operate in any coprocessor socket designed for an NMOS 80287 or the CMOS 80C287A.

Figure 3.1. DIP Pin Configuration



*"Top View" means as the package is seen from the component side of the board.

Figure 3.2. PLCC Pin Configuration

Table 3.2. PLCC Pin Cross-Reference

Pin Name	Ceramic Package	PLCC Package
BUSY#	25	28
CKM	39	44
CLK	32	36
CMD0	29	32
CMD1	31	35
D_0	23	26
D_1	22	25
D_2	21	24
D_3	20	22
D_4	19	21
D_5	18	20
D_6	17	19
D_7	16	18
D_8	15	17
D_9	14	16
D_{10}	12	14
D_{11}	11	13
D_{12}	8	9
D_{13}	7	8
D_{14}	6	7
D_{15}	5	5
ERROR#	26	29
No Connect	1,2,3,4,13,37,38,40	6,11,23,33,40
NPRD#	27	30
NPS1#	34	38
NPS2	33	37
NPWR#	28	31
PEACK#	36	41
PEREQ	24	27
RESET	35	39
V_{CC}	9	1,3,10,15,42
V_{SS}	10,30	2,4,12,34,43

3.1.1 CLOCK (CLK)

This input provides the basic timing for internal operation. This pin does not require MOS-level input; it will operate at either TTL or MOS levels up to the maximum allowed frequency. A minimum frequency must be provided to keep the internal logic properly functioning. Depending on the signal on CKM, the signal on CLK can be divided by two to produce the internal clock signal.

3.1.2 CLOCKING MODE (CKM)

This pin is a strapping option. When it is strapped to V_{CC} (HIGH), the CLK input is used directly; when strapped to V_{SS} (LOW), the CLK input is divided by two to produce the internal clock signal. During the RESET sequence, this input must be stable at least four internal clock cycles (i.e. CLK clocks when CKM is HIGH; 2 × CLK clocks when CKM is LOW) before RESET goes LOW.

3.1.3 SYSTEM RESET (RESET)

A LOW to HIGH transition on this pin causes the Intel287 XL MCP to terminate its present activity and to enter a dormant state. RESET must remain active (HIGH) for at least four CLK periods (i.e., the RESET signal presented to the Intel287 XL MCP must be at least four Intel287 XL MCP clocks long, regardless of the frequency of the CPU). Note that the Intel287 XL MCP is active internally for 25 clock cycles after the termination of the RESET signal (the HIGH to LOW transition of RESET); therefore, the first instruction should not be written to the Intel287 XL MCP until 25 clocks after the falling edge of RESET. Table 3.3 shows the status of the output pins during the reset sequence. After a reset, all output pins return to their inactive states.

Table 3.3. Output Pin Status during Reset

Output Pin Name	Value During Reset
BUSY#	HIGH
ERROR#	HIGH
PEREQ	LOW
D_{15}–D_0	Tristate OFF

3.1.4 PROCESSOR EXTENSION REQUEST (PEREQ)

When active, this pin signals to the CPU that the Intel287 XL MCP is ready for data transfer to/from its data FIFO. With 80286 and 80C286 CPUs, PEREQ can be deactivated after assertion of PEACK#. These CPUs rely on the MCP to deassert PEREQ when all operands have been transfered. When there are more than five data transfers, PEREQ is deactiviated after the first three transfers and subsequently after every four transfers. This signal always goes inactive before BUSY# goes inactive.

3.1.5 BUSY STATUS (BUSY#)

When active, this pin signals to the CPU that the Intel287 XL MCP is currently executing an instruction. It should be connected to the CPU's BUSY# pin. During the RESET sequence this pin is HIGH.

3.1.6 ERROR STATUS (ERROR#)

This pin reflects the ES bit of the status register. When active, it indicates that an unmasked exception has occurred. This signal can be changed to inactive state only by the following instructions (without a preceding WAIT): FNINIT, FNCLEX, FNSTENV, FNSAVE, FLDCW, FLDENV, and FRSTOR. This pin should be connected to the ERROR# pin of the CPU. ERROR# can change state only when BUSY# is active.

3.1.7 PROCESSOR EXTENSION ACKNOWLEDGE (PEACK#)

During execution of escape instructions, an 80286 or 80C286 CPU asserts PEACK# to acknowledge that the request signal (PEREQ) has been recognized and that data transfer is in progress. The 80286/80C286 also drives this signal HIGH during RESET.

This input may be asynchronous with respect to the Intel287 XL MCP clock except during a RESET sequence, when it must satisfy setup and hold requirements relative to RESET.

3.1.8 DATA PINS (D_{15}-D_0)

These bidirectional pins are used to transfer data and opcodes between the CPU and Intel287 XL MCP. They are normally connected directly to the corresponding CPU data pins. Other buffers/drivers driving the local data bus must be disabled when the CPU reads from the MCP. HIGH state indicates a value of one. D_0 is the least significant data bit.

3.1.9 NUMERIC PROCESSOR WRITE (NPWR#)

A signal on this pin enables transfers of data from the CPU to the MCP. This input is valid only when NPS1# and NPS2 are both active.

3.1.10 NUMERIC PROCESSOR READ (NPRD#)

A signal on this pin enables transfers of data from the MCP to the CPU. This input is valid only when NPS1# and NPS2 are both active.

3.1.11 NUMERIC PROCESSOR SELECTS (NPS1# and NPS2)

Concurrent assertion of these signals indicates that the CPU is performing an escape instruction and enables the Intel287 XL MCP to execute that instruction. No data transfer involving the Intel287 XL MCP occurs unless the device is selected by these lines.

3.1.12 COMMAND SELECTS (CMD0 AND CMD1)

These pins along with the select pins allow the CPU to direct the operation of the Intel287 XL MCP.

3.1.13 SYSTEM POWER (V_{CC})

System power provides the +5V±10% DC supply input. All V_{CC} pins should be tied together on the circuit board and local decoupling capacitors should be used between V_{CC} and V_{SS}.

3.1.14 SYSTEM GROUND (V_{SS})

All V_{SS} pins should be tied together on the circuit board and local decoupling capacitors should be used between V_{CC} and V_{SS}.

3.2 Processor Architecture

As shown by the block diagram on the front page, the Intel287 XL MCP is internally divided into three sections: the bus control logic (BCL), the data interface and control unit, and the floating point unit (FPU). The FPU (with the support of the control unit which contains the sequencer and other support units) executes all numerics instructions. The data interface and control unit is responsible for the data flow to and from the FPU and the control registers, for receiving the instructions, decoding them, and sequencing the microinstructions, and for handling some of the administrative instructions. The BCL is responsible for CPU bus tracking and interface.

3.2.1 BUS CONTROL LOGIC

The BCL communicates solely with the CPU using I/O bus cycles. The BCL appears to the CPU as a special peripheral device. It is special in two respects: the CPU initiates I/O automatically when it encounters ESC instructions, and the CPU uses reserved I/O addresses to communicate with the BCL. The BCL does not communicate directly with memory. The CPU performs all memory access, transferring input operands from memory to the Intel287 XL MCP and transferring outputs from the Intel287 XL MCP to memory. A dedicated communication protocol makes possible high-speed transfer of opcodes and operands between the CPU and Intel287 XL MCP.

3.2.2 DATA INTERFACE AND CONTROL UNIT

The data interface and control unit latches the data and, subject to BCL control, directs the data to the FIFO or the instruction decoder. The instruction de-

Table 3.4. Bus Cycles Definition

NPS1#	NPS2	CMD0	CMD1	NPRD#	NPWR#	Bus Cycle Type
x	0	x	x	x	x	Intel287 XL MCP not selected
1	x	x	x	x	x	Intel287 XL MCP not selected
0	1	0	0	1	0	Opcode write to Intel287 XL MCP
0	1	0	0	0	1	CW or SW read from Intel287 XL MCP
0	1	1	0	0	1	Read data from Intel287 XL MCP
0	1	1	0	1	0	Write data to Intel287 XL MCP
0	1	0	1	1	0	Write exception pointers
0	1	0	1	0	1	Reserved
0	1	1	1	0	1	Reserved
0	1	1	1	1	0	Reserved

coder decodes the ESC instructions sent to it by the CPU and generates controls that direct the data flow in the FIFO. It also triggers the microinstruction sequencer that controls execution of each instruction. If the ESC instruction is FINIT, FCLEX, FSTSW, FSTSW AX, FSTCW, FSETPM, or FRSTPM, the control executes it independent ly of the FPU and the sequencer. The data interface and control unit is the one that generates the BUSY#, PEREQ, and ERROR# signals that synchronize Intel287 XL activities with the CPU.

3.2.3 FLOATING-POINT UNIT

The FPU executes all instructions that involve the register stack, including arithmetic, logical, transcendental, constant, and data transfer instructions. The data path in the FPU 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.

3.3 Bus Cycles

The pins NPS1#, NPS2, CMD0, CMD1, NPRD#, and NPWR# identify bus cycles for the MCP. Table 3.4 defines the types of Intel287 XL MCP bus cycles.

3.3.1 Intel287™ XL MCP ADDRESSING

The NPS1#, NPS2, CMD0, and CMD1 signals allow the MCP to identify which bus cycles are intended for the MCP. The MCP responds to I/O cycles when the I/O address is 00F8H, 00FAH, 00FCH. The correspondence between I/O addresses and control signals is defined by Table 3.5. To guarantee correct operation of the MCP, programs must not perform any I/O operations to these reserved port addresses.

Table 3.5. I/O Address Decoding

I/O Address (Hexadecimal)	Intel287 XL MCP Select and Command Inputs			
	NPS2	NPS1#	CMD1	CMD0
00F8	1	0	0	0
00FA	1	0	0	1
00FC	1	0	1	0

3.3.2 CPU/MCP SYNCHRONIZATION

The pins BUSY#, PEREQ, and ERROR# are used for various aspects of synchronization between the CPU and the MCP.

BUSY# is used to synchronize instruction transfer from the CPU to the Intel287 XL MCP. When the Intel287 XL MCP recognizes an ESC instruction, it asserts BUSY#. For most ESC instructions, the CPU waits for the Intel287 XL MCP to deassert BUSY# before sending the new opcode.

The MCP uses the PEREQ pin of the CPU to signal that the MCP is ready for data transfer to or from its data FIFO. The MCP does not directly access memory; rather, the CPU provides memory access services for the MCP. Thus, memory access on behalf of the MCP always obeys the rules applicable to the mode of the CPU, whether the CPU be in real-address mode or protected mode.

Once the CPU initiates an Intel287 XL MCP instruction that has operands, the CPU waits for PEREQ signals that indicate when the Intel287 XL MCP is ready for operand transfer. Once all operands have

been transferred (or if the instruction has no operands) the CPU continues program execution while the Intel287 XL MCP executes the ESC instruction.

In 8086/8087 systems, WAIT instructions may be required to achieve synchronization of both commands and operands. In Intel287 XL MCP systems, however, WAIT instructions are required only for operand synchronization; namely, after MCP stores to memory (except FSTSW and FSTCW) or load from memory. (In 80286/Intel287 XL MCP systems, WAIT is required before FLDENV and FRSTOR; with other CPU's, WAIT is not required in these cases.) Used this way, WAIT ensures that the value has already been written or read by the MCP before the CPU reads or changes the value.

Once it has started to execute a numerics instruction and has transferred the operands from the CPU, the Intel287 XL MCP can process the instruction in parallel with and independent of the host CPU. When the MCP detects an exception, it asserts the ERROR# signal, which causes a CPU interrupt.

3.4 Bus Operation

With respect to bus interface, the Intel287 XL MCP is fully asynchronous with the CPU, even when it operates from the same clock source as the CPU. The CPU initiates a bus cycle for the MCP by activating both NPS1# and NPS2, the MCP select signals. During the CLK period in which NPS1# and NPS2 are activated, the Intel287 XL MCP also examines the NPRD# and NPWR# input signals to determine whether the cycle is a read or a write cycle and examines the CMD0 and CMD1 inputs to determine whether an opcode, operand, or control/status register transfer is to occur. The Intel287 XL MCP activates its BUSY# output some time after the leading edge of the NPRD# or NPWR# signal. Input and output data are referenced to the trailing edges of the NPRD# and NPWR# signals.

The Intel287 XL MCP activates the PEREQ signal when it is ready for data transfer. In 80286/80C286 systems, the CPU activates PEACK# when no more data transfers are required, which causes the Intel287 XL MCP to deactivate PEREQ, halting the data transfer.

3.5 80286/Intel287™ XL MCP, 80C286/Intel287™ XL MCP Interface and Socket Compatibilty

The ceramic Intel287 XL MCP device can fit into existing 80287 sockets since the pin configuration is identical.

The CERDIP 80C287A utilizes a different pin configuration with extra power and ground pins. However, the Intel287 XL MCP operates in 80C287A sockets also. The extra power and ground pins are not connected inside the Intel287 XL MCP and not used. Refer to 80C287A data sheet (Order #240347).

Note that when the clock selection is CKM = 0, the Intel287 XL MCP divides the clock input by two, not by three as on the 80287. In this case, the Intel287 XL MCP will operate faster.

The interface between the Intel287 XL MCP and the 80286/80C286 CPU (illustrated in Figure 3.3) has these characteristics:

- The Intel287 XL MCP resides on the local data bus of the CPU.
- The CPU and Intel287 XL MCP share the same RESET signals. They may also share the same clock input; however, for greatest performance, an external oscillator may be needed.
- The corresponding BUSY#, ERROR#, PEREQ, and PEACK# pins are connected together.
- NPS2 is tied HIGH permanently, while NPS1#, CMD1, and CMD0 come from the latched address pins. The 80286 generates I/O addresses 00F8H, 00FAH, and 00FCH during MCP bus cycles. Address 00FEH is reserved.
- The Intel287 XL MCP NPRD# and NPWR# inputs are connected to I/O read and write signals from local bus control logic.

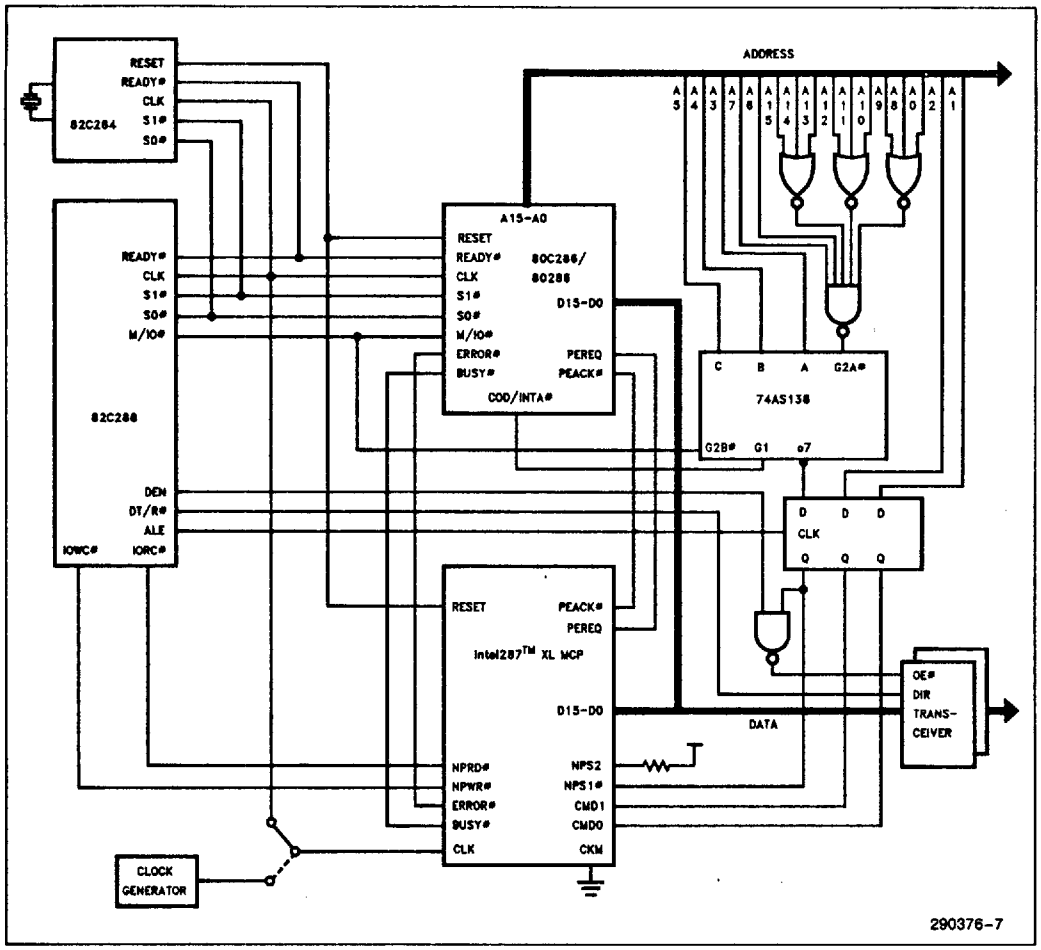


Figure 3.3. 80286/Intel287™ XL System Configuration

4.0 ELECTRICAL DATA

4.1 Absolute Maximum Ratings

NOTE

Stresses above those listed 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.

Case temperature T_C under bias .....0°C to 85°C

Storage temperature .....−65°C to +150°C

Voltage on any pin with respect to ground .....−0.5 to V_{CC}+0.5V

Power dissipation .....1.5 Watt

4.2 Power and Frequency Requirements

The typical relationship between I_{CC} and the frequency of operation F is as follows:

$$I_{CCtyp} = 55 + 5*F \text{ mA}, \text{ where F is in MHz.}$$

When the frequency is reduced below the minimum operating frequency specified in the AC Characteristics table, the internal states of the Intel287 XL MCP may become indeterminate. The Intel287 XL MCP clock cannot be stopped; otherwise, I_{CC} would increase significantly beyond what the equation above indicates. Power dissipation decreases with frequency for frequencies ≥ 4 MHz.

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.*

4.3 D.C. Characteristics

Table 4.1. D.C. Specifications T_C = 0 to 85 deg C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input LOW Voltage	−0.5	+0.8	V	
V_{IH}	Input HIGH Voltage	2.0	V_{CC}+0.5	V	
V_{ICL}	Clock Input LOW Voltage	−0.5	+0.8	V	
V_{ICH}	Clock Input HIGH Voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output LOW Voltage		0.45	V	I_{OL} = 3.0 mA
V_{OH}	Output HIGH Voltage	2.4		V	I_{OH} = −0.4 mA
I_{CC}	Power Supply Current		135	mA	Note 1
I_{LI}	Input Leakage Current		±10	μA	Note 2
I_{LO}	I/O Leakage Current		±10	μA	Note 3
C_{IN}	Input Capacitance		10	pF	Note 4
C_O	I/O or Output Capacitance		12	pF	Note 4
C_{CLK}	Clock Capacitance		20	pF	Note 4

NOTES:
1. 12.5 MHz operation, output load = 100 pF
2. 0V ≤ V_{IN} ≤ V_{CC}
3. 0.45V ≤ V_{OUT} ≤ V_{CC} −0.45
4. F_C = 1MHz

4.4 A.C. Characteristics

Table 4.2. Timing Requirements T_C = 0 to 85 deg C, V_{CC} = 5V ±10%
All timings are measured at 1.5V unless otherwise specified

Symbol	Parameter	12.5 MHz		Test Conditions
		Min (ns)	Max (ns)	
Tdvwh (t6) Twhdx (t7)	Data setup to NPWR# Data hold from NPWR#	43 14		
Trlrh (t8) Twlwh (t9)	NPRD# active time NPWR# active time	59 59		
Tavwl (t10) Tavrl (t11)	Command valid to NPWR# Command valid to NPRD#	0 0		
Tmhrl (t12)	Min delay from PEREQ active to NPRD# active	40		
Tklkh (t33) Tkhkl (t34) Tkhch (t35) Tklcl (t36)	PEACK# active time PEACK# inactive time PEACK# inactive to NPRD#, NPWR# inactive PEACK# active setup to NPRD#, NPWR# active	55 60 30 30		
Tchkl (t37)	NPRD#, NPWR# inactive to PEACK# active	−30		
Twhax (t18) Trhax (t19)	Command hold from NPWR# Command hold from NPRD#	12 12		
Tivcl (t20) Tclih (t21) Tpaksu (t38) Tpakhd (t39) Trscl (t24) Tclrs (t25)	NPRD#, NPWR#, RESET to CLK setup time NPRD#, NPWR#, RESET from CLK hold time PEACK# setup to RESET falling edge PEACK# hold from RESET falling edge RESET to CLK setup RESET from CLK hold	46 26 80 80 21 14		Note 1 Note 1 Note 1 Note 1
Tcmdi (t26)	Command inactive time Write to write Read to read Read to write Write to read	 69 69 69 69		

NOTE:
1. This is an asynchronous input. This specification is given for testing purposes only, to assure recognition at a specific CLK edge (not tested).

Table 4.3. Timing Responses

Symbol	Parameter	12.5 MHz		Test Conditions
		Min (ns)	Max (ns)	
Trhqz (t27)	NPRD# inactive to data float*		18	Note 2
Trlqv (t28)	NPRD# active to data valid		50	Note 3
Tilbh (t29)	ERROR# active to BUSY# inactive	104		Note 4
Twlbv (t30)	NPWR# active to BUSY# active		80	Note 4
Tklml (t31)	NPRD#, NPWR# or PEACK# active to PEREQ inactive		80	Note 5
Trhqh (t32)	Data hold from NPRD# inactive	2		Note 3

NOTES:
* The data float delay is not tested.
2. The float condition occurs when the measured output current is less than I_{OL} on D_{15}–D_0.
3. D_{15}–D_0 loading: C_L = 100pf.
4. BUSY# loading: C_L = 100pf.
5. On last data transfer of numeric instruction.

Table 4.4. Clock Timings

Symbol	Parameter		12.5 MHz		Test Conditions
			Min (ns)	Max (ns)	
Tclcl (t1a)	CLK period	CKM = 1	80	250	
(t1b)		CKM = 0	40	125	
Tclch (t2a)	CLK low time	CKM = 1	35		
(t2b)		CKM = 0	9		Note 6, 10
Tchcl (t3a)	CLK high time	CKM = 1	35		V_{CC} = ±10%
		CKM = 1	28		V_{CC} = ±5%, Note 11
(t3b)		CKM = 0	13		Note 7, 10
Tch1ch2 (t4)				10	Note 8
Tch2ch1 (t5)				10	Note 9

NOTES:
6. At 0.8V.
7. At 2.0V.
8. CKM = 1: 3.5V to 1.0V
9. CKM = 1: 1.0V to 3.5V
10. Proper operation can also be achieved by meeting the CPU specification
11. Provides compatibility for sockets designed for Intel 80287-6/8/10 MHz Math CoProcessors.

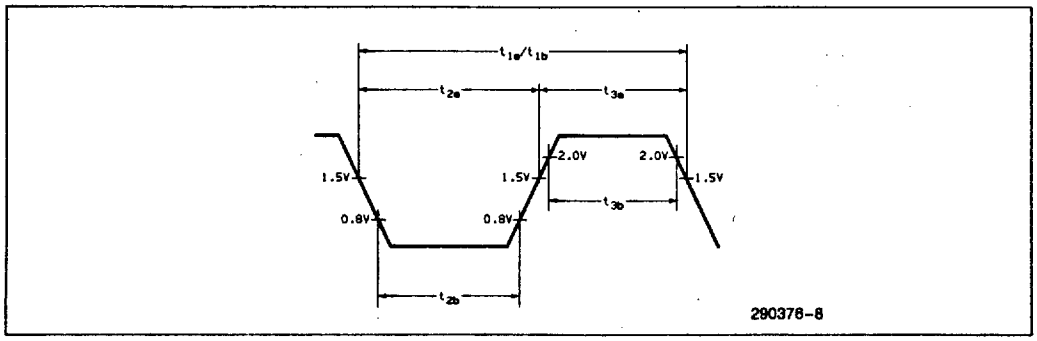


Figure 4.1. AC Drive and Measurement Points—CLK Input

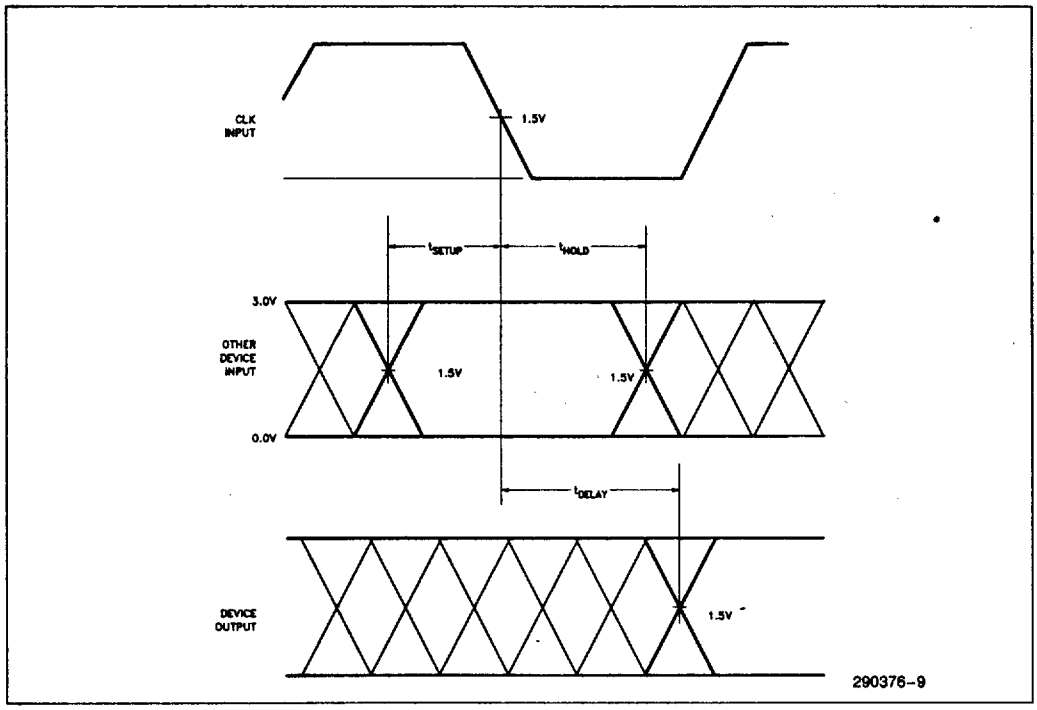


Figure 4.2. AC Setup, Hold, and Delay Time Measurements—General

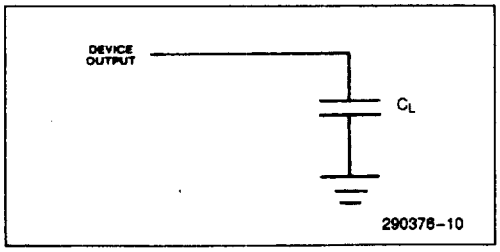


Figure 4.3. AC Test Loading on Outputs

RESET, NPWR#, NPRD# inputs are asynchronous to CLK. Timing requirements in Figures 4.7 through 4.10 are given for testing purposes only, to assure recognition at a specific CLK edge.

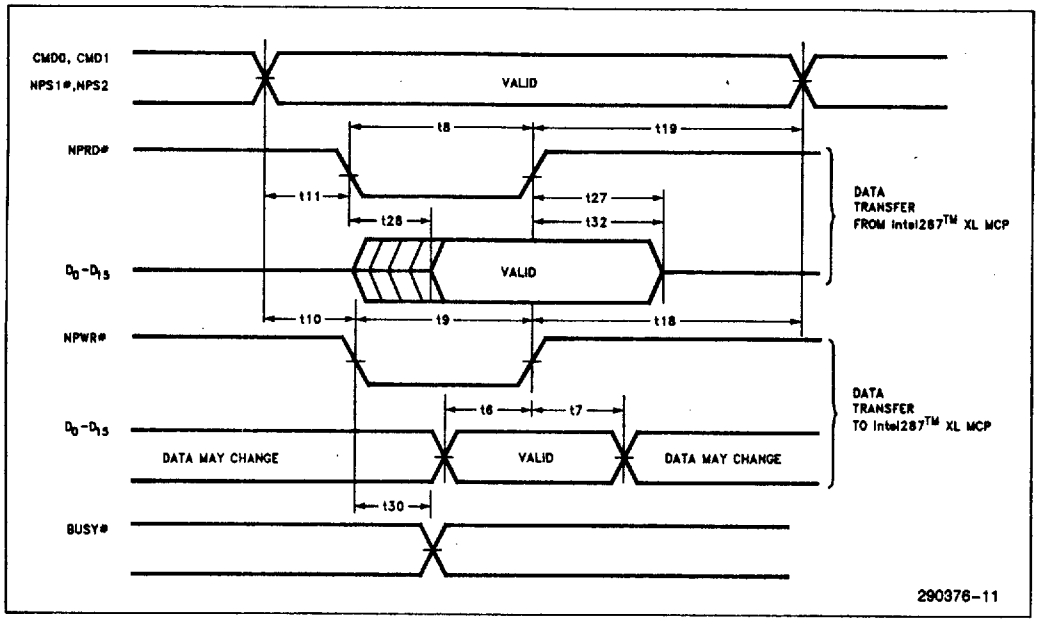


Figure 4.4. Data Transfer Timing (Initiated by CPU)

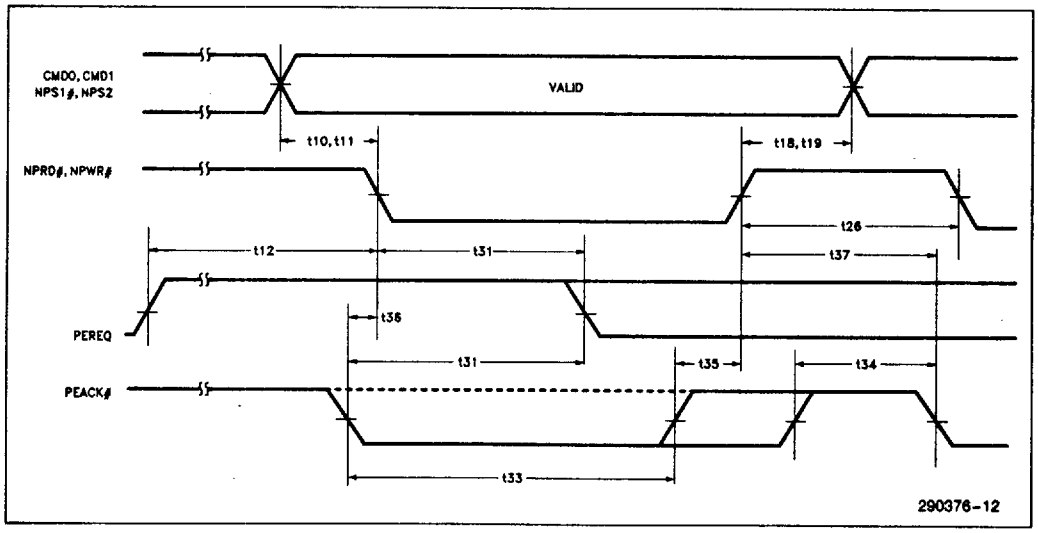


Figure 4.5. Data Channel Timing (Initiated by Intel287™ XL)

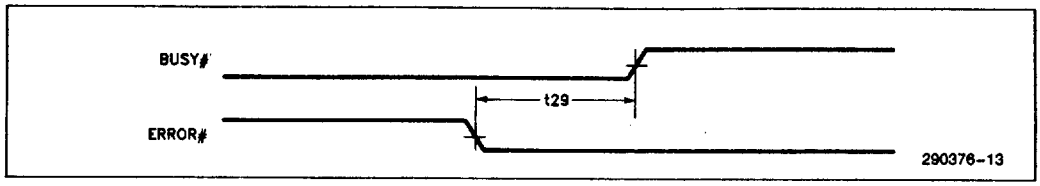


Figure 4.6. ERROR# Output Timing

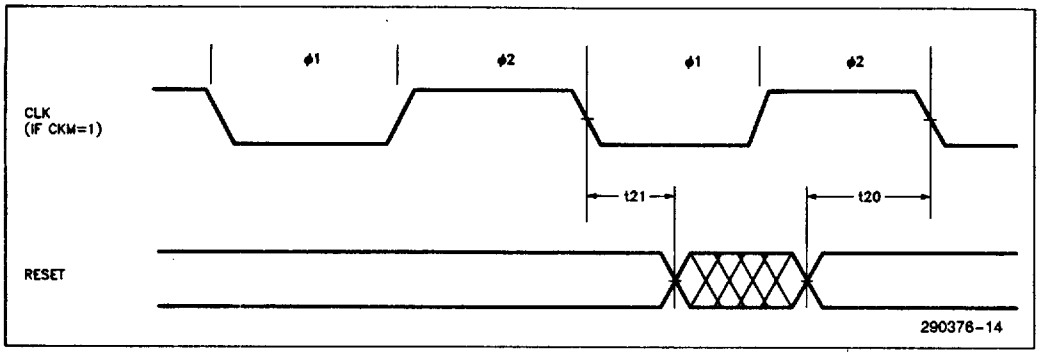


Figure 4.7. CLK, RESET Timing (CKM = 1)

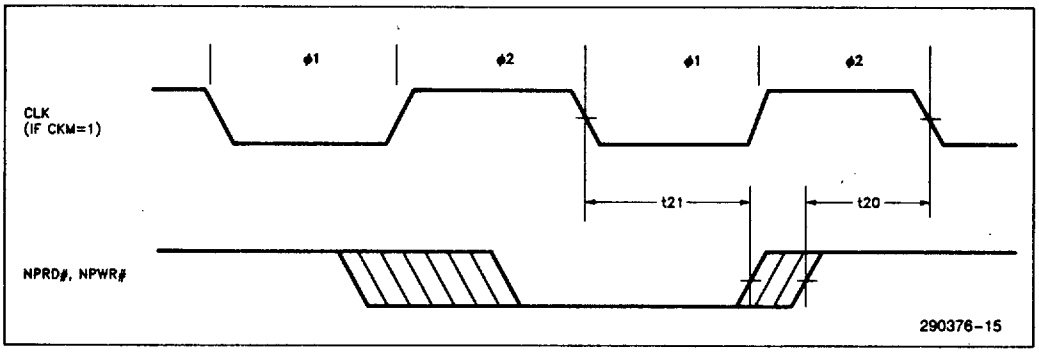
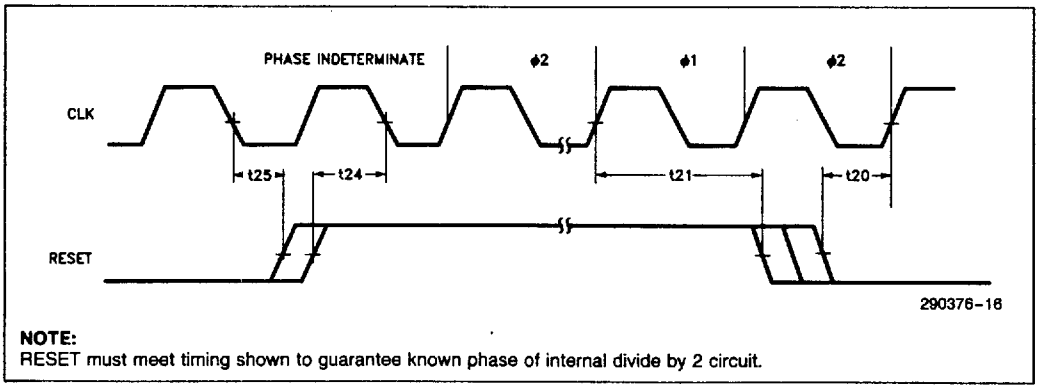


Figure 4.8. CLK, NPRD#, NPWR# Timing (CKM = 1)



NOTE:
RESET must meet timing shown to guarantee known phase of internal divide by 2 circuit.

Figure 4.9. CLK, RESET Timing (CKM = 0)

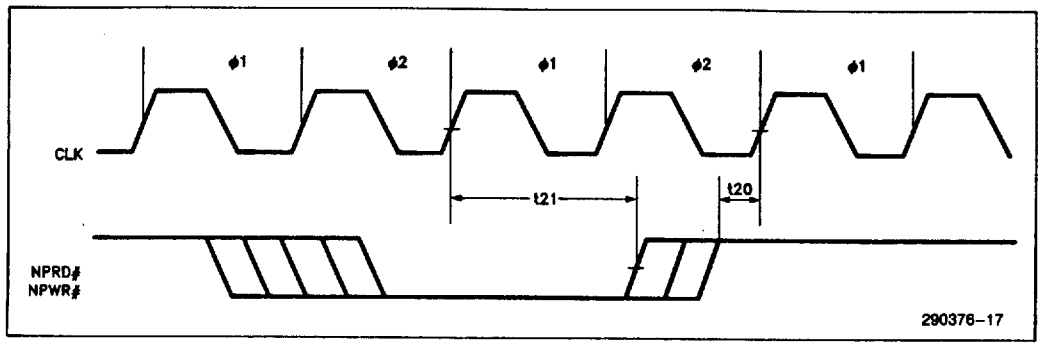


Figure 4.10. CLK, NPRD#, NPWR# Timing (CKM = 0)

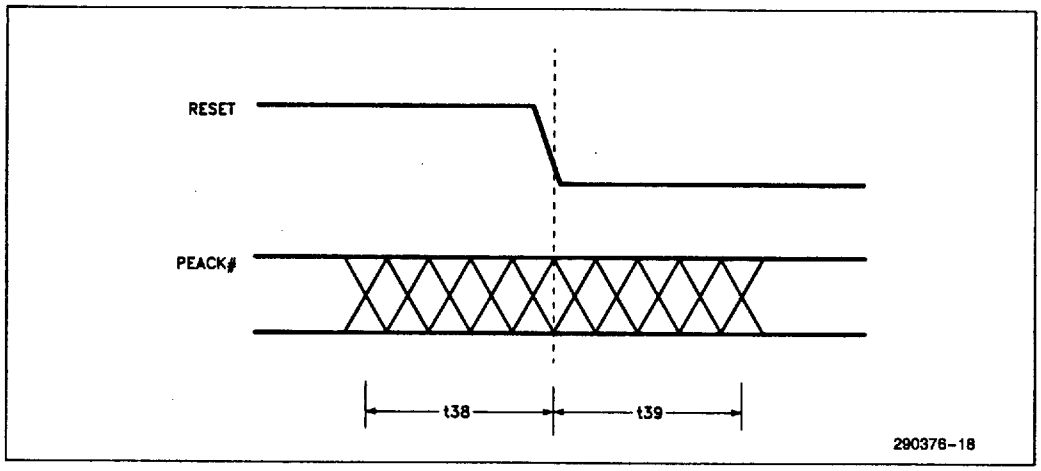


Figure 4.11. RESET, PEACK# Setup and Hold Timing

5.0 Intel287™ XL MCP EXTENSIONS TO THE CPU'S INSTRUCTION SET

Instructions for the Intel287 XL MCP assume one of the five forms shown in Table 5-1. In all cases, instructions are at least two bytes long and begin with the bit pattern 11011B, which identifies the ESCAPE class of instruction. Instructions that refer to memory operands specify addresses using the CPU's addressing modes.

MOD (Mode field) and R/M (Register/Memory specifier) have the same interpretation as the corresponding fields of CPU instructions (refer to Programmer's Reference Manual for the CPU). The DISP (displacement) is optionally present in instructions that have MOD and R/M fields. Its presence depends on the values of MOD and R/M, as for instructions of the CPU.

The instruction summaries that follow assume that the instruction has been prefetched, decoded, and is ready for execution; that bus cycles do not require wait states; that there are no local bus HOLD requests delaying processor access to the bus; and that no exceptions are detected during instruction execution. Timings are given in internal Intel287 XL MCP clocks and include the time for opcode and data transfer between the CPU and the MCP. If the instruction has MOD and R/M fields that call for both base and index registers, add one clock.

Table 5.1. Instruction Formats

	Instruction										Optional Field
	First Byte				Second Byte						
1	11011	OPA		1	MOD		1	OPB	R/M		DISP
2	11011	MF		OPA	MOD		OPB*		R/M		DISP
3	11011	d	P	OPA	1	1	OPB*		ST(i)		
4	11011	0	0	1	1	1	1	OP			
5	11011	0	1	1	1	1	1	OP			
	15–11	10	9	8	7	6	5	4	3	2 1 0	

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

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

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

R XOR d = 0-Destination (Op) Source
R XOR d = 1-Source (Op) Destination

*In FSUB and FDIV, the low-order bit of the OPB is the R (reversed) bit

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

ESC = 11011

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

Intel287™ XL MCP Extension to the CPU's Instruction Set

Instruction	Encoding			Clock Count Range			
	Byte 0	Byte 1	Optional Bytes 2–3	32-Bit Real	32-Bit Integer	64-Bit Real	16-Bit Integer
DATA TRANSFER							
FLD = Load[a]							
Integer/real memory to ST(0)	ESC MF 1	MOD 000 R/M	SIB/DISP	36	61–68	45	61–65
Long integer memory to ST(0)	ESC 111	MOD 101 R/M	SIB/DISP		76–87		
Extended real memory to ST(0)	ESC 011	MOD 101 R/M	SIB/DISP		48		
BCD memory to ST(0)	ESC 111	MOD 100 R/M	SIB/DISP		270–279		
ST(i) to ST(0)	ESC 001	11000 ST(i)			21		
FST = Store							
ST(0) to integer/real memory	ESC MF 1	MOD 010 R/M	SIB/DISP	51	86–100	56	88–101
ST(0) to ST(i)	ESC 101	11010 ST(i)			18		
FSTP = Store and Pop							
ST(0) to integer/real memory	ESC MF 1	MOD 011 R/M	SIB/DISP	51	86–100	56	88–101
ST(0) to long integer memory	ESC 111	MOD 111 R/M	SIB/DISP		91–108		
ST(0) to extended real	ESC 011	MOD 111 R/M	SIB/DISP		61		
ST(0) to BCD memory	ESC 111	MOD 110 R/M	SIB/DISP		520–542		
ST(0) to ST(i)	ESC 101	11001 ST (i)			19		
FXCH = Exchange							
ST(i) and ST(0)	ESC 001	11001 ST(i)			25		
COMPARISON							
FCOM = Compare							
Integer/real memory to ST(0)	ESC MF 0	MOD 010 R/M	SIB/DISP	42	72–79	51	71–75
ST(i) to ST(0)	ESC 000	11010 ST(i)			31		
FCOMP = Compare and pop							
Integer/real memory to ST	ESC MF 0	MOD 011 R/M	SIB/DISP	42	72–79	51	71–77
ST(i) to ST(0)	ESC 000	11011 ST(i)			33		
FCOMPP = Compare and pop twice							
ST(1) to ST(0)	ESC 110	1101 1001			33		
FTST = Test ST(0)	ESC 001	1110 0100			35		
FUCOM = Unordered compare	ESC 101	11100 ST(i)			31		
FUCOMP = Unordered compare and pop	ESC 101	11101 ST(i)			33		
FUCOMPP = Unordered compare and pop twice	ESC 010	1110 1001			33		
FXAM = Examine ST(0)	ESC 001	11100101			37–45		
CONSTANTS							
FLDZ = Load +0.0 into ST(0)	ESC 001	1110 1110			27		
FLD1 = Load +1.0 into ST(0)	ESC 001	1110 1000			31		
FLDPI = Load pi into ST(0)	ESC 001	1110 1011			47		
FLDL2T = Load $\log_2(10)$ into ST(0)	ESC 001	1110 1001			47		

Shaded areas indicate instructions not available in 8087/80287, but available on 80C287A and Intel287 XL MCP.

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

Intel287™ XL MCP Extension to the CPU's Instruction Set (Continued)

Instruction	Encoding			Clock Count Range			
	Byte 0	Byte 1	Optional Bytes 2–3	32-Bit Real	32-Bit Integer	64-Bit Real	16-Bit Integer
CONSTANTS (Continued)							
FLDL2E = Load $\log_2(e)$ into ST(0)	ESC 001	1110 1010		47			
FLDLG2 = Load $\log_{10}(2)$ into ST(0)	ESC 001	1110 1100		48			
FLDLN2 = Load $\log_e(2)$ into ST(0)	ESC 001	1110 1101		48			
ARITHMETIC							
FADD = Add							
Integer/real memory with ST(0)	ESC MF 0	MOD 000 R/M	SIB/DISP	40–48	73–78	49–79	71–85
ST(i) and ST(0)	ESC d P 0	11000 ST(i)		30–38[b]			
FSUB = Subtract							
Integer/real memory with ST(0)	ESC MF 0	MOD 10 R R/M	SIB/DISP	40–48	73–98	49–77	71–83[c]
ST(i) and ST(0)	ESC d P 0	1110 R R/M		33–41[d]			
FMUL = Multiply							
Integer/real memory with ST(0)	ESC MF 0	MOD 001 R/M	SIB/DISP	43–51	77–88	52–77	76–87
ST(i) and ST(0)	ESC d P 0	1100 1 R/M		25–53[e]			
FDIV = Divide							
Integer/real memory with ST(0)	ESC MF 0	MOD 11 R R/M	SIB/DISP	105	136–143[f]	114	136–140[g]
ST(i) and ST(0)	ESC d P 0	1111 R R/M		95[h]			
FSQRT[i] = Square root	ESC 001	1111 1010		129–136			
FSCALE = Scale ST(0) by ST(1)	ESC 001	1111 1101		74–93			
FPREM = Partial remainder of ST(0) ÷ ST(1)	ESC 001	1111 1000		81–162			
FPREM1 = Partial remainder (IEEE)	ESC 001	1111 0101		108–192			
FRNDINT = Round ST(0) to integer	ESC 001	1111 1100		73–87			
FXTRACT = Extract components of ST(0)	ESC 001	1111 0100		75–83			
FABS = Absolute value of ST(0)	ESC 001	1110 0001		29			
FCHS = Change sign of ST(0)	ESC 001	1110 0000		31–37			

Shaded areas indicate instructions not available in 8087/80287, but available on Intel287 XL MCP and 80C287A.

NOTES:
b. Add 3 clocks to the range when d = 1.
c. Add 1 clock to **each** range when R = 1.
d. Add 3 clocks to the range when d = 0.
e. Typical = 48 (When d = 0, 42–50, typical = 45).
f. Add 1 clock to the range when R = 1.
g. 135–141 when R = 1.
h. Add 3 clocks to the range when d = 1.
i. $-0 \leq ST(0) \leq +\infty$.

Intel287™ XL MCP Extension to the CPU's Instruction Set (Continued)

Instruction	Encoding			Clock Count Range
	Byte 0	Byte 1	Optional Bytes 2-3	
TRANSCENDENTAL				
FCOS = Cosine of ST(0)	ESC 001	1111 1111		130-779[j]
FPTAN[k] = Partial tangent of ST(0)	ESC 001	1111 0010		198-504[j]
FPATAN = Partial arctangent	ESC 001	1111 0011		321-494
FSIN = Sine of ST(0)	ESC 001	1111 1110		129-778[j]
FSINCOS = Sine and cosine of ST(0)	ESC 001	1111 1011		201-816[j]
F2XM1[l] = $2^{ST(0)} - 1$	ESC 001	1111 0000		215-483
FYL2X[m] = ST(1) * $\log_2(ST(0))$	ESC 001	1111 0001		127-545
FYL2XP1[n] = ST(1) * $\log_2(ST(0) + 1.0)$	ESC 001	1111 1001		264-554
PROCESSOR CONTROL				
FINIT = Initialize MCP	ESC 011	1110 0011		25
FSETPM = Set protected mode	ESC 011	1110 0100		12
FRSTPM = Reset protected mode	ESC 011	1111 0100		12
FSTSW AX = Store status word	ESC 111	1110 0000		18
FLDCW = Load control word	ESC 001	MOD 101 R/M	SIB/DISP	33
FSTCW = Store control word	ESC 101	MOD 111 R/M	SIB/DISP	18
FSTSW = Store status word	ESC 101	MOD 111 R/M	SIB/DISP	18
FCLEX = Clear exceptions	ESC 011	1110 0010		8
FSTENV = Store environment	ESC 001	MOD 110 R/M	SIB/DISP	192-193
FLDENV = Load environment	ESC 001	MOD 100 R/M	SIB/DISP	85
FSAVE = Save state	ESC 101	MOD 110 R/M	SIB/DISP	521-522
FRSTOR = Restore state	ESC 101	MOD 100 R/M	SIB/DISP	396
FINCSTP = Increment stack pointer	ESC 001	1111 0111		28
FDECSTP = Decrement stack pointer	ESC 001	1111 0110		29
FFREE = Free ST(i)	ESC 101	1100 0 ST(i)		25
FNOP = No operations	ESC 001	1101 0000		19

Shaded areas indicate instructions not available in 8087/80287, but available on Intel287 XL MCP and 80C287A.

NOTES:
j. These timings hold for operands in the range $|x| < \pi/4$. For operands not in this range, up to 78 additional clocks may be needed to reduce the operand.
k. $0 \leq |ST(0)| < 2^{63}$.
l. $-1.0 \leq ST(0) \leq 1.0$.
m. $0 \leq ST(0) < \infty$, $-\infty < ST(1) < +\infty$.
n. $0 \leq |ST(0)| < (2 - SQRT(2))/2$, $-\infty < ST(1) < +\infty$.