Preliminary

80C188

CMOS High Integration 16-Bit Microprocessor

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

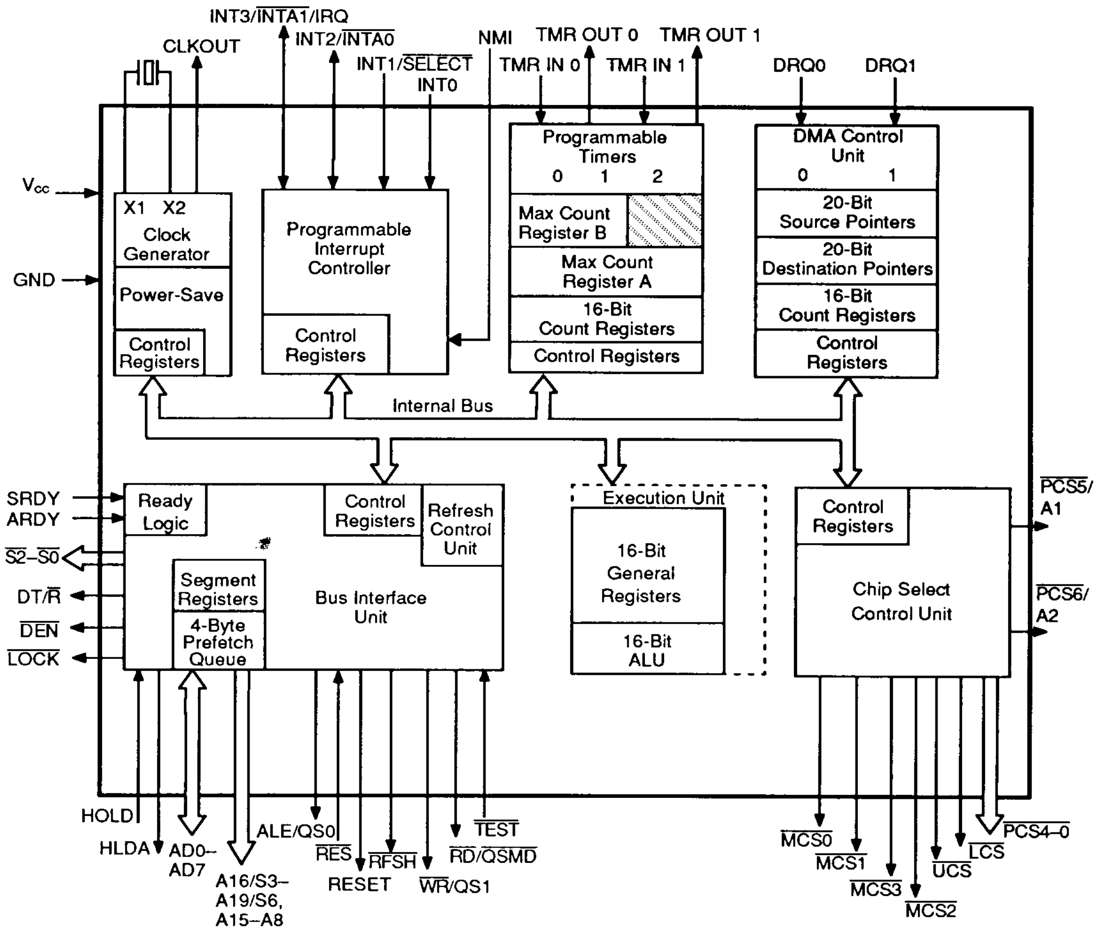
- **Operation Modes Include:**
  - –Enhanced mode which has
    - –DRAM Refresh Control Unit
    - –Power-save mode
  - –Compatible Mode
    - –NMOS 80188 pin-for-pin replacement for non-numerics applications
- **Integrated Feature Set**
  - –Enhanced 80C86/C88 CPU
  - –Clock generator
  - –Two independent DMA channels
  - –Programmable interrupt controller
  - –Three programmable 16-bit timers
  - –Dynamic RAM refresh control unit
  - –Programmable memory and peripheral chip select logic
  - –Programmable wait-state generator
  - –Local bus controller
  - –Power save mode
  - –System-level testing support (high-impedance test mode)
- **Available in 20-MHz (80C188-20), 16-MHz (80C188-16), 12.5-MHz (80C188-12), and 10-MHz (80C188) versions**
- **Direct addressing capability to 1 MByte of memory and 64-KByte I/O**
- **Fully static CMOS design**
- **Completely object code compatible with all existing 8086/8088 software and also has ten additional instructions over 8086/8088**
- **Complete system development**
  - –There are many vendors making support tools for the 80C188. Software tools for the NMOS 80188 can be used for the 80C188 as can the NMOS emulators
- **Available in:**
  - –68-Pin Plastic Leaded Chip Carrier (PLCC)
  - –80-Pin Plastic Quad Flat Pack (PQFP)
    - –In TapePak® and Trimmed/Formed Configurations

GENERAL DESCRIPTION

The 80C188 is a CMOS high-integration microprocessor. It has features which are new to the 80186 family, which include a DRAM refresh control unit, power-save mode, and a direct numerics interface. When used in "compatible" mode, the 80C188 is 100% pin-for-pin compatible with the NMOS 80188 (except for 8087 applications). The "enhanced" mode of operation allows the full feature set of the 80C188 to be used. The 80C186 is upward compatible with 8086 and 8088 software and fully compatible with 80186 and 80188 software.

Publication # 13088 Rev. A Amendment /0
Issue Date: August 1990

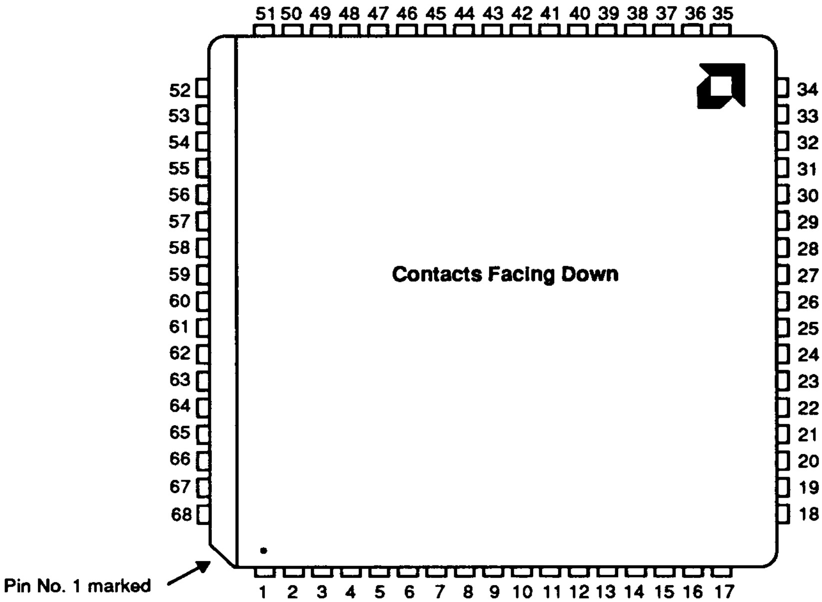
BLOCK DIAGRAM



13087D-001

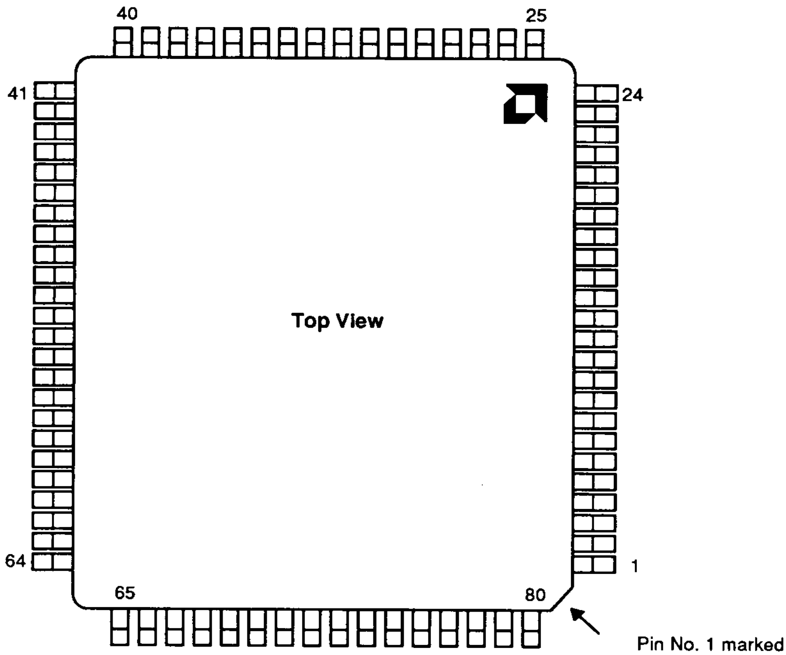
CONNECTION DIAGRAMS

68-Pin PLCC

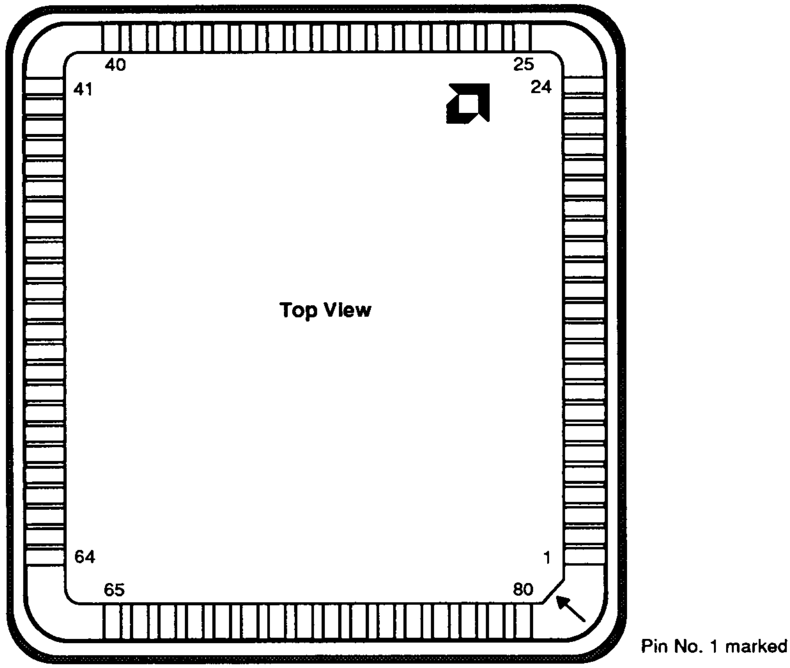


CONNECTION DIAGRAMS (continued)

80-Pin Trimmed and Formed PQFP



80-Pin TapePak PQFP



PIN DESIGNATIONS
(Sorted by Pin Name)

Symbol	PLCC Pin #	PQFP Pin #	Type
A19/S6	65	6	O
A18/S5	66	5	O
A17/S4	67	4	O
A16/S3	68	3	O
A15	1	1	O
A14	3	79	O
A13	5	77	O
A12	7	75	O
A11	10	71	O
A10	12	69	O
A9	14	67	O
A8	16	65	O
AD7	2	80	I/O
AD6	4	78	I/O
AD5	6	76	I/O
AD4	8	74	I/O
AD3	11	70	I/O
AD2	13	68	I/O
AD1	15	66	I/O
AD0	17	64	I/O
ALE/QSO	61	10	O
ARDY	55	20	I
$\overline{RFSH}$	64	7	O
CLKOUT	56	19	O
$\overline{DEN}$	39	38	O
DRQ0	18	61	I
DRQ1	19	60	I
DT/$\overline{R}$	40	37	O
HOLD	50	26	I
HLDA	51	25	O
INT0	45	31	I
INT1/$\overline{SELECT}$	44	32	I
INT2/$\overline{INTA0}$	42	35	I/O
INT3/$\overline{INTA1}$/IRQ	41	36	I/O

Symbol	PLCC Pin #	PQFP Pin #	Type
$\overline{LCS}$	33	46	I/O
$\overline{LOCK}$	48	28	O
$\overline{MCS0}$	38	39	I/O
$\overline{MCS1}$	37	40	I/O
$\overline{MCS2}$	36	41	O
$\overline{MCS3}$	35	42	O
NMI	46	30	I
$\overline{PCS5}$/A1	31	48	O
$\overline{PCS6}$/A2	32	47	O
$\overline{PCS4}$	30	49	O
$\overline{PCS3}$	29	50	O
$\overline{PCS2}$	28	51	O
$\overline{PCS1}$	27	52	O
$\overline{PCS0}$	25	54	O
$\overline{RD}$/$\overline{QSMD}$	62	9	I/O
$\overline{RES}$	24	55	I
RESET	57	18	O
$\overline{S2}$	54	21	O
$\overline{S1}$	53	22	O
$\overline{S0}$	52	23	O
SRDY	49	27	I
$\overline{TEST}$	47	29	I
TMR IN 0	20	59	I
TMR IN 1	21	58	I
TMR OUT 0	22	57	O
TMR OUT 1	23	56	O
$\overline{UCS}$	34	45	I/O
V_{CC}	9	33, 34, 72, 73	I
V_{SS}	26	12	I
V_{SS}	60	13, 53	I
$\overline{WR}$/QS1	63	8	O
X1	59	16	I
X2	58	17	O

*On the PQFP package the following pins are No Connects: 2, 11, 14, 15, 24, 43, 44, 62, and 63.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option (if applicable)
e. PQFP Option

N 80C188

e. PQFP Option
M = TapePak
Blank = Trimmed and Formed

d. SPEED OPTION
Blank = 10 MHz
–12 = 12.5 MHz
–16 = 16 MHz
–20 = 20 MHz

c. DEVICE NUMBER/DESCRIPTION
80C188
CMOS High-Integration 16-Bit Microprocessor

b. PACKAGE TYPE
N = 68-Pin Plastic Leaded Chip Carrier
S = 80-Pin Plastic Quad Flat Pack

a. TEMPERATURE RANGE
Blank = Commercial (0 to +70°C)

Valid Combinations	
N	80C188
	80C188–12
	80C188–16
	80C188–20

S TapePak	80C188M
	80C188–12M
	80C188–16M
	80C188–20M

S Trimmed and Formed	80C188
	80C188–12
	80C188–16
	80C188–20

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

TapePak is a plastic package technology that consists of a PQFP surrounded by an external plastic carrier ring. This ring holds the unformed leads together during testing and shipping. The customer does the lead trim and form operation prior to board installation at whatever standoff, lead length, or angle desired.

The TapePak is the PQFP configuration recommended by AMD because it virtually eliminates lead coplanarity problems since lead trim and form is performed immediately prior to board population. Also, the coin-stack tubes that TapePak comes in simplify incoming test and handling.

Trimmed and formed configuration has AMD doing the trim and form function to a JEDEC standard standoff and lead length. The product is shipped in trays.

PIN DESCRIPTIONS

A19/S6, A18/S5, A17/S4, A16/S3

Address Bus Outputs (Outputs)

Address Bus Outputs (19–16) and Bus Cycle Status (6–3) indicate the four most significant address bits during T1. These signals are active High.

During T2, T3, TW, and T4, the S6 pin is Low to indicate a CPU-initiated bus cycle or High to indicate a DMA-initiated bus cycle. During the same T-states, S3, S4, and S5 are always Low. These outputs are floated during bus HOLD or RESET.

A15–A8

Address-Only Bus (Outputs)

Address-Only Bus (15–8) contains valid addresses from T1–T4. The bus is active High. These outputs are floated during a bus HOLD or RESET.

AD7–AD0

Address/Data Bus (Inputs/Outputs)

Address/Data Bus (7–0) signals constitute the time multiplexed memory or I/O address (T1) and data (T2, T3, TW, and T4) bus. The bus is active High. These pins are floated during bus HOLD or RESET.

ALE/QS0

Address Latch Enable/Queue Status (Output)

Address Latch Enable/Queue Status 0 is provided by the 80C188 to latch the address. ALE is active High, with addresses guaranteed to be valid on the trailing edge.

ARDY

Asynchronous Ready (Input)

Asynchronous Ready informs the 80C188 that the addressed memory space or I/O device will complete a data transfer. The ARDY pin accepts a rising edge that is asynchronous to CLKOUT and is active High. The falling edge of ARDY must be synchronized to the 80C188 clock. Connecting ARDY High will always assert the ready condition to the CPU. If this line is unused, it should be tied Low to yield control to the SRDY pin.

CLKOUT

Clock Output (Output)

Clock Output provides the system with a 50% duty cycle waveform. All device pin timings are specified relative to CLKOUT. CLKOUT is active during RESET and bus HOLD.

$\overline{\text{DEN}}$

Data Enable (Output)

Data Enable is provided as a data bus transceiver output enable. $\overline{\text{DEN}}$ is active Low during each memory and I/O access (including 80C187 access). $\overline{\text{DEN}}$ is High whenever DT/$\overline{\text{R}}$ changes state. $\overline{\text{DEN}}$ will float during a bus HOLD or RESET.

DRQ0–DRQ1

DMA Requests (Inputs)

DMA Request is asserted High by an external device when it is ready for DMA Channel 0 or 1 to perform a transfer. These signals are level-triggered and internally synchronized.

DT/$\overline{\text{R}}$

Data Transmit/Receive (Output)

Data Transmit/Receive controls the direction of data flow through an external data bus transceiver. When Low, data is transferred to the 80C188. When High, the 80C188 places write data on the data bus. DT/$\overline{\text{R}}$ floats during a bus HOLD or RESET.

HOLD, HLDA

(Input, Output)

HOLD indicates that another bus master is requesting the local bus. The HOLD input is active High. The 80C188 generates HLDA (High) in response to a HOLD request. Simultaneous with the issuance of HLDA, the 80C188 will float the local bus and control lines. After HOLD is detected as being Low, the 80C188 will lower HLDA. When the 80C188 needs to run another bus cycle, it will again drive the local bus and control lines.

In Enhanced Mode, HLDA will go Low when a DRAM refresh cycle is pending in the 80C188 and an external bus master has control of the bus. It will be up to the external master to relinquish the bus by lowering HOLD so that the 80C188 may execute the refresh cycle.

INT0, INT1/$\overline{\text{SELECT}}$, INT2/$\overline{\text{INTA0}}$, INT3/$\overline{\text{INTA1}}$/IRQ

Maskable Interrupt Requests (Inputs, Input/Output)

Maskable Interrupt Requests can be requested by activating one of these pins. When configured as inputs, these pins are active High. Interrupt Requests are synchronized internally. INT2 and INT3 may be configured to provide active-Low interrupt-acknowledge output signals. All interrupt inputs may be configured to be either edge- or level-triggered. To ensure recognition, all interrupt requests must remain active until the interrupt is acknowledged. When slave mode is selected, the function of these pins changes (see Interrupt Controller section of this datasheet).

$\overline{\text{LCS}}$

Lower Memory Chip Select (Output/Input)

Lower Memory Chip Select is active Low whenever a memory reference is made to the defined lower portion (1K–256K) of memory. $\overline{\text{LCS}}$ does not float during bus HOLD. The address range activating $\overline{\text{LCS}}$ is software programmable.

UCS and LCS are sampled upon the rising edge of RES. If both pins are held Low, the 80C188 will enter ONCE mode. In ONCE mode all pins assume a high impedance state and remain so until a subsequent RESET. LCS has a weak internal pull-up that is active only during RESET to ensure that the 80C188 does not enter ONCE mode inadvertently.

LOCK

Lock (Output)

LOCK output indicates that other system bus masters are not to gain control of the system bus. LOCK is active Low. The LOCK signal is requested by the LOCK prefix instruction and is activated at the beginning of the first data cycle associated with the instruction immediately following the LOCK prefix. It remains active until the completion of that instruction. No instruction prefetching will occur while LOCK is asserted. LOCK floats during bus HOLD or RESET.

MCS0 MCS1, MCS2, MCS3

Mid-Range Memory Chip Select (Outputs)

Mid-Range Memory Chip Select signals are active Low when a memory reference is made to the defined mid-range portion of memory (8K–512K). These lines do not float during bus HOLD. The address ranges activating MCS3–0 are software programmable.

NMI

Non-Maskable Interrupt (Input)

The Non-Maskable Interrupt input causes a Type 2 interrupt. An NMI transition from Low to High is latched and synchronized internally, and initiates the interrupt at the next instruction boundary. NMI must be asserted for at least one CLKOUT period. The Non-Maskable Interrupt cannot be avoided by programming.

PCS5/A1

Peripheral Chip Select 5 or Latched A1 (Output)

Peripheral Chip Select 5 or Latched A1 may be programmed to provide a sixth peripheral chip select, or to provide an internally latched A1 signal. The address range activating PCS5 is software programmable. When programmed to provide latched A1 rather than PCS5, this pin will retain the previously latched value of A1 during a bus HOLD. A1 is active High. PCS5/A1 does not float during bus HOLD.

PCS6/A2

Peripheral Chip Select 6 or Latched A2 (Output)

Peripheral Chip Select 6 or Latched A2 may be programmed to provide a seventh peripheral chip select, or to provide an internally latched A2 signal. The address range activating PCS6 is software programmable. When programmed to provide latched A2 rather than PCS6, this pin will retain the previously latched value of A2 during a bus HOLD. A2 is active High. PCS6/A2 does not float during bus HOLD.

PCS4–0

Peripheral Chip Select Signals (Outputs)

Peripheral Chip Select signals 4–0 are active Low when a reference is made to the defined peripheral area (64-Kb I/O or 1-Mb memory space). These lines do not float during bus HOLD. The address ranges activating PCS4–0 are software programmable.

RD/QSMD

Read Strobe (Output/Input)

Read Strobe is an active Low signal which indicates that the 80C188 is performing a memory or I/O read cycle. It is guaranteed not to go Low before the A/D bus is floated. An internal pull-up ensures that RD/QSMD is High during RESET. Following RESET the pin is sampled to determine whether the 80C188 is to provide ALE, RD, and WR, or queue status information. To enable Queue Status Mode, RD must be connected to GND. RD will float during bus HOLD.

RES

RESET (Input)

An active RES causes the 80C188 to immediately terminate its present activity, clear the internal logic, and enter a dormant state. This signal may be asynchronous to the 80C188 clock. The 80C188 begins fetching instructions approximately 6½ clock cycles after RES is returned High. For proper initialization, V_{CC} must be within specifications and the clock signal must be stable for more than 4 clocks with RES held Low. RES is internally synchronized. This input is provided with a Schmitt-trigger to facilitate power-on RES generation via an RC network.

RESET

System Reset (Output)

Reset output indicates that the 80C188 CPU is being reset and can be used as a system reset. It is active High, synchronized with the processor clock, and lasts an integer number of clock periods corresponding to the length of the RES signal. Reset goes inactive two clockout periods after RES goes inactive. When tied to the TEST pin, RESET forces the 80C188 into enhanced mode. RESET is not floated during bus HOLD.

RFSH

Refresh (Output)

In compatible mode, RFSH is High. In enhanced mode, RFSH is asserted Low to signify a refresh bus cycle. The RFSH output pin floats during bus HOLD or RESET, regardless of operating mode.

S2–S0

Bus Cycle Status (Outputs)

Bus cycle status S2–S0 are encoded to provide bus-transaction information:

80C188 Bus Cycle Status Information

S2	S1	S0	Bus Cycle Initiated
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

The status pins float during HOLD/HLDA.

S2 may be used as a logical memory or I/O indicator, and S1 as a DT/R indicator.

SRDY

Synchronous Ready (Input)

Synchronous Ready informs the 80C188 that the addressed memory space or I/O device will complete a data transfer. The SRDY pin accepts an active-High input synchronized to CLKOUT. The use of SRDY allows a relaxed system timing over ARDY. This is accomplished by elimination of the one-half clock cycle required to internally synchronize the ARDY input signal. Connecting SRDY High will always assert the ready condition to the CPU. If this line is unused, it should be tied Low to yield control to the ARDY pin.

TEST

Test (Input)

The TEST pin is sampled during and after reset to determine whether the 80C188 is to enter Compatible or Enhanced Mode. Enhanced Mode requires TEST to be High on the rising edge of RES and Low four CLKOUT cycles later. Any other combination will place the 80C188 in Compatible Mode. A weak internal pull-up ensures a High state when the pin is not driven. This pin is examined by the WAIT instruction. If the TEST input is High when WAIT execution begins, instruction execution will suspend. TEST will be resampled every five clocks until it goes Low, at which time execution will resume. If interrupts are enabled while the 80C188 is waiting for TEST, interrupts will be serviced.

TMR IN 0, TMR IN 1

Timer Inputs (Inputs)

Timer inputs are used either as clock or control signals, depending upon the programmed timer mode. These inputs are active High (or Low-to-High transitions are counted) and internally synchronized. Timer inputs must be tied High when not being used as clock or retrigger inputs.

TMR OUT 0, TMR OUT 1

Timer Outputs (Outputs)

Timer outputs are used to provide single pulse or continuous waveform generation, depending upon the timer mode selected. These outputs are not floated during a bus HOLD.

UCS

Upper Memory Chip Select (Output/Input)

Upper Memory Chip Select is an active Low output whenever a memory reference is made to the defined upper portion (1K–256K block) of memory. UCS does not float during bus HOLD. The address range activating UCS is software programmable.

UCS and LCS are sampled upon the rising edge of RES. If both pins are held Low, the 80C188 will enter ONCE mode. In ONCE mode all pins assume a high impedance state and remain so until a subsequent RESET. UCS has a weak internal pull-up that is active during RESET to ensure that the 80C188 does not enter ONCE mode inadvertently.

V_{CC}

Power Supply (Inputs)

System power: +5-V power supply.

V_{SS}

Ground (Inputs)

System ground.

WR/QS1

Write Strobe/Queue Status 1 (Output)

Write Strobe/Queue Status 1 indicates that the data on the bus is to be written into a memory or an I/O device. It is active Low, and floats during bus HOLD or RESET. When the 80C188 is in queue status mode, the ALE/QS0 and WR/QS1 pins provide information about processor/instruction queue interaction.

QS1	QS0	Queue Operation
0	0	No queue operation
0	1	First opcode byte fetched from the queue
1	1	Subsequent byte fetched from the queue
1	0	Empty the queue

X1, X2

Crystal Inputs (Input/Output)

Crystal inputs X1 and X2 provide external connections for a fundamental mode or third overtone parallel resonant crystal for the internal oscillator. X1 can connect to an external clock instead of a crystal. In this case, minimize the capacitance on X2. The input or oscillator frequency is internally divided by two to generate the clock signal (CLKOUT).

FUNCTIONAL DESCRIPTION

Introduction

The following Functional Description describes the base architecture of the 80C188. The 80C188 is a very high integration 16-bit microprocessor. It combines 15–20 of the most common microprocessor system components onto one chip. The 80C188 is object code compatible with the 8086/8088 microprocessors and adds 10 new instruction types to the 8086/8088 instruction set.

The 80C188 has two major modes of operation, Compatible and Enhanced. In Compatible Mode the 80C188 is completely compatible with the NMOS 80188, with the exception of 8087 support. The Enhanced mode adds two new features to the system design: Power-save control and Dynamic RAM refresh.

80C188 Base Architecture

The 8086, 8088, 80186, and 80188 families all contain the same basic set of registers, instructions, and addressing modes. The 80C188 processor is upward compatible with the 8086 and 8088 CPUs.

Register Set

The 80C188 base architecture has fourteen registers, as shown in Figures 1 and 2. These registers are grouped into the following categories.

General Registers

Eight 16-bit general purpose registers may be used for arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used as 16-bit registers or split into pairs of separate 8-bit registers.

Segment Registers

Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization, page 11.)

Base and Index Registers

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

Status and Control Registers

Two 16-bit special purpose registers record or alter certain aspects of the 80C188 processor state. These are the Instruction Pointer Register, which contains the offset address of the next sequential instruction to be executed, and the Status Word Register, which contains status and control flag bits (see Figures 1 and 2).

Status Word Description

The Status Word records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80C186 within a given operating mode (bits 8, 9, and 10). The Status Word Register is 16-bits wide. The function of the Status Word bits is shown in Table 1.

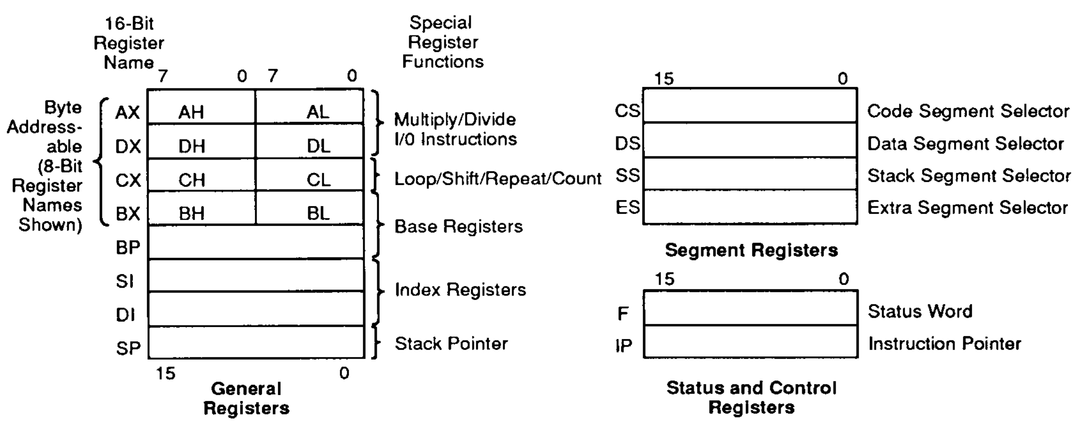


Figure 1. 80C188 Register Set

13087D-002

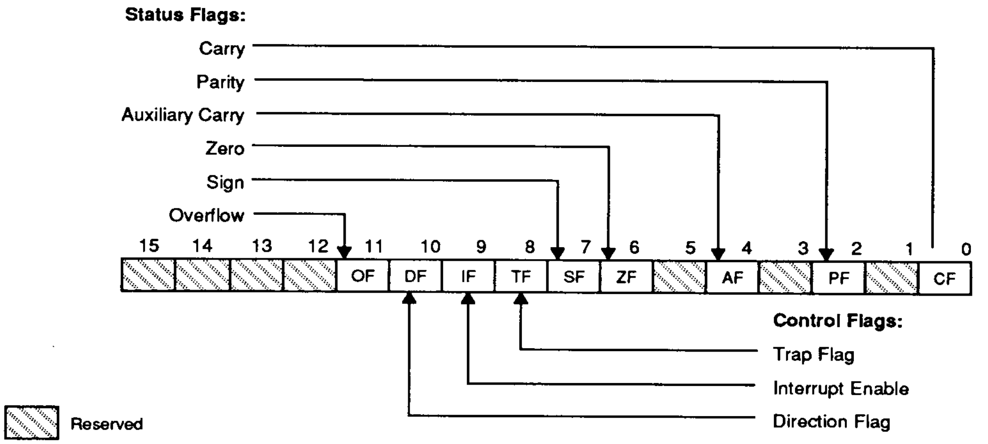


Figure 2. Status Word Format

Table 1. Status Word Bit Functions

Bit Position	Name	Function
0	CF	Carry Flag—Set on high-order bit carry or borrow; cleared otherwise.
2	PF	Parity Flag—Set if low-order 8 bits of result contain an even number of 1 bits; cleared otherwise.
4	AF	Auxiliary Carry—Set on carry from or borrow to the low order four bits of the general purpose register AL; cleared otherwise.
6	ZF	Zero Flag—Set if result is 0; cleared otherwise.
7	SF	Sign Flag—Set equal to high-order bit of result (0 if positive, 1 if negative).
8	TF	Single-Step Flag—Once set, a single-step interrupt occurs after the next instruction executes. TF is cleared by the single-step interrupt.
9	IF	Interrupt-Enable Flag—When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location.
10	DF	Direction Flag—Causes string instructions to auto decrement the appropriate index register when set. Clearing DF causes auto-increment.
11	OF	Overflow Flag—Set if the signed result cannot be expressed within the number of bits in the destination operand; cleared otherwise.

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, control transfer, high-level instructions, and processor control. These categories are summarized in 80C188 Instruction Set section, page 12.

An 80C188 instruction can reference anywhere from zero to several operands. An operand can reside in a register, in the instruction itself, or in memory. Specific operand addressing modes are discussed later in this datasheet.

Memory Organization

Memory is organized in sets of segments. Each segment is a linear contiguous sequence of up to 64K (2^{16}) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit base segment and a 16-bit offset. The 16-bit base values are contained in one of four internal segment registers (code, data, stack, extra). The physical address is calculated by shifting the base value LEFT by 4 bits and adding the 16-bit offset value to yield a 20-bit physical address (see Figure 3). This allows for a 1-Mb physical address size.

All instructions that address operands in memory must specify the base segment and the 16-bit offset value. For speed and compact instruction encoding, the segment register used for physical address generation is implied by the addressing mode used (see Table 3). These rules follow the way programs are written (see Figure 4) as independent modules that require areas for code and data, a stack, and access to external data areas.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs.

80C188 Instruction Set

All mnemonics copyright Intel Corp.

General Purpose

MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte

Input/Output

IN	Input byte or word
OUT	Output byte or word

Address Object

LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES

Flag Transfer

LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

Addition

ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition

Subtraction

SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction

Multiplication

MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply

Division

DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte or word
CWD	Convert word to doubleword
MOVS	Move byte or word string
INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/REPZ	Repeat while equal/zero
REPNE/REPNZ	Repeat while not equal/not zero

Logicals

NOT	"NOT" byte or word
AND	"AND" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word

Shifts

SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word

Rotates

ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

Flag Operations

STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt-enable flag
CLI	Clear interrupt-enable flag

External Synchronization

HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{TEST}$ pin active
LOCK	Lock bus during next instruction

No Operation

NOP	No operation

High Level Instructions

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

Conditional Transfers

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

Unconditional Transfers

CALL	Call procedure
RET	Return from procedure
JMP	Jump

Iteration Controls

LOOP	Loop
LOOPE/LOOPZ	Loop if equal/zero
LOOPNE/ LOOPNZ	Loop if not equal/not zero
JCXZ	Jump if register CX = 0

Interrupts

INT	Interrupt
INTO	Interrupt if overflow
IRET	Interrupt return

To access operands that do not reside in one of the four immediately available segments, a full 32-bit pointer can be used to reload both the base (segment) and offset values.

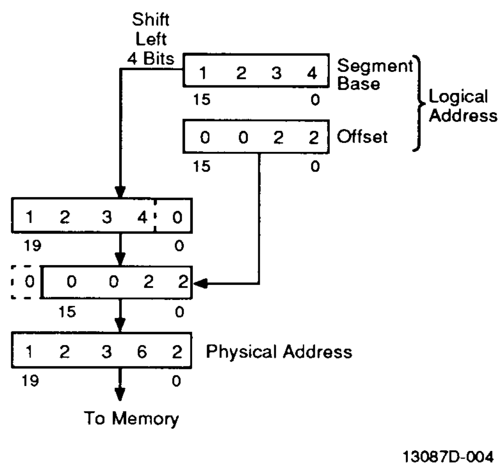


Figure 3. Two-Component Address

Table 2. Segment Register Selection Rules

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Instruction prefetch and immediate data.
Stack	Stack (SS)	All stack pushes and pops; any memory references which use BP Register as a base register.
External Data (Global)	Extra (ES)	All string instruction references which use the DI register as an index.
Local Data	Data (DS)	All other data references.

Addressing Modes

The 80C188 provides eight categories of addressing modes to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

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

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

- the *displacement* (an 8- or 16-bit immediate value contained in the instruction);
- the *base* (contents of either the BX or BP base registers); and
- the *index* (contents of either the SI or DI index registers).

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

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

- Direct Mode: The operand's offset is contained in the instruction as an 8- or 16-bit displacement element.
- Register Indirect Mode: The operand's offset is in one of the registers SI, DI, BX, or BP.
- Based Mode: The operand's offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).
- Indexed Mode: The operand's offset is the sum of an 8- or 16-bit displacement and the contents of an index register (SI or DI).
- Based Indexed Mode: The operand's offset is the sum of the contents of a base register and an index register.
- Based Index Mode with Displacement: The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

Data Types

The 80C188 directly supports the following data types:

- Integer: A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation.
- Ordinal: An unsigned binary numeric value contained in an 8-bit byte or a 16-bit word.
- Pointer: A 16- or 32-bit quantity, composed of a 16-bit offset component or a 16-bit segment base component in addition to a 16-bit offset component.
- String: A contiguous sequence of bytes or words. A string may contain from 1 to 64K bytes.
- ASCII: A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
- BCD: A byte (unpacked) representation of the decimal digits 0–9.
- Packed BCD: A byte (packed) representation of two decimal digits (0–9). One digit is stored in each nibble (4-bits) of the byte.

In general, individual data elements must fit within defined segment limits. Figure 5 graphically represents the data types supported by the 80C188.

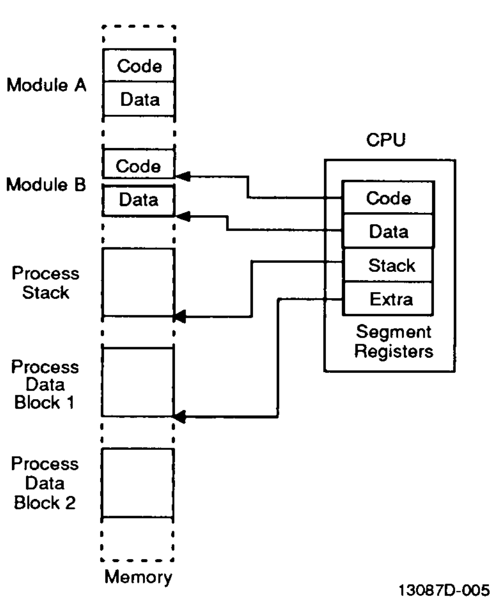


Figure 4. Segmented Memory Helps Structure Software

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. Separate instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register. Eight-bit port addresses are zero-extended such that A15–A8 are Low. I/O port addresses 00F8(H) through 00FF(H) are reserved.

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Status Word) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware-initiated interrupts occur in response to an external input and are classified as non-maskable or maskable.

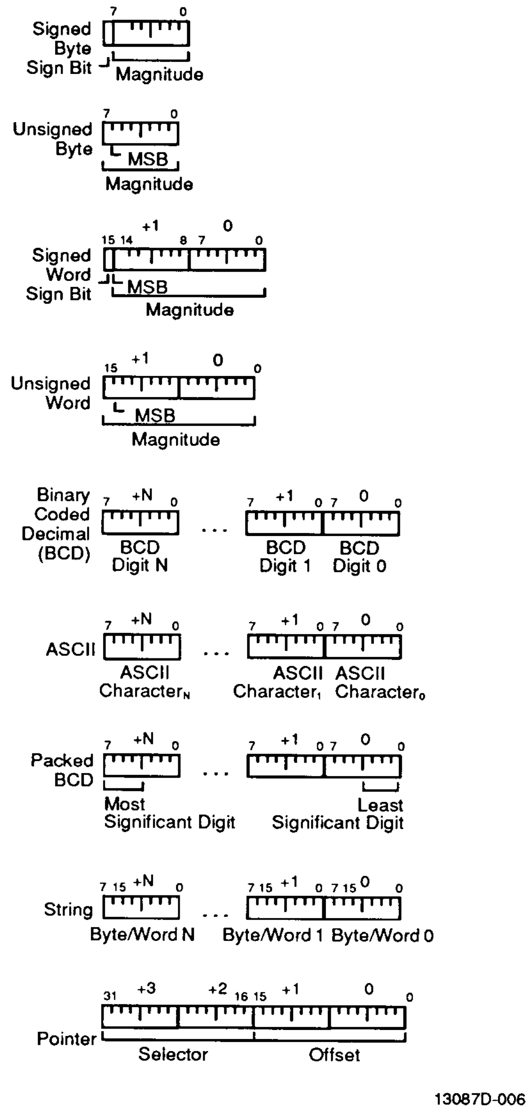


Figure 5. 80C188 Supported Data Types

Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. If the exception was caused by executing an ESC instruction with the ESC trap bit set in the relocation register, the return instruction will point to the ESC instruction, or to the segment override prefix immediately preceding the ESC instruction if the prefix was present. In all other cases, the return address from an exception will point at the instruction immediately following the instruction causing the exception.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0–31, some of which are used for instruction exceptions, are reserved. Table 3 shows the 80C188 predefined types and default priority levels. For each interrupt, an 8-bit vector must be supplied to the 80C186, which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. In addition, internal peripherals and non-cascaded external interrupts will generate their own vectors through the internal interrupt controller. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware-initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Interrupt Sources

The 80C188 can service interrupts generated by software or hardware. The software interrupts are generated by specific instructions (INT, ESC, unused OP, etc.) or the results of conditions specified by instructions (array bounds check, INTO, DIV, IDIV, etc.). All interrupt sources are serviced by an indirect call through an element of a vector table. This vector table is indexed by using the interrupt vector type (Table 3), multiplied by four. All hardware-generated interrupts are sampled at the end of each instruction. Thus, the software interrupts will begin service first. Once the service routine is entered and interrupts are enabled, any hardware source of sufficient priority can interrupt the service routine in progress.

Those pre-defined 80C188 interrupts which cannot be masked by programming are described below.

Divide Error Exception (Type 0)

Generated when a DIV or IDIV instruction quotient cannot be expressed in the number of bits in the destination.

Single-Step Interrupt (Type 1)

Generated after most instructions if the TF (single step) flag in the status word is set. This interrupt allows programs to execute one instruction at a time. Interrupts will not be generated after prefix instructions (e.g., REP), instructions which modify segment registers (e.g., POP DS), or the WAIT instruction. Vectoring to the single-step interrupt service routine clears the TF bit. An IRET instruction in the interrupt service routine restores the TF bit to logic 1 and transfers control to the next instruction to be single-stepped.

Non-Maskable Interrupt-NMI (Type 2)

An external interrupt source which is serviced regardless of the state of the IF (interrupt enable flag) bit. No external interrupt acknowledge sequence is performed. The IF bit is cleared at the beginning of a NMI interrupt to prevent maskable interrupts from being serviced. A typical use of NMI would be to activate a power failure routine.

Breakpoint Interrupt (Type 3)

A 1-byte version of the INT instructions. It uses 12 (OCH) as an index into the service routine address table (because it is a Type 3 interrupt).

INTO Detected Overflow Exception (Type 4)

Generated during an INTO instruction if the 0F bit is set.

Array BOUNDS Exception (Type 5)

Generated during a BOUND instruction if the array index is outside the array bounds. The array bounds are located in memory at a location indicated by one of the instruction operands. The other operand indicates the value of the index to be checked.

Unused Opcode Exception (Type 6)

Generated if execution is attempted on undefined opcodes.

ESCAPE Opcode Exception (Type 7)

Generated if execution is attempted of ESC opcodes (D8–DFH). The 80C188 does not check an escape opcode trap bit as does the 80C186. On the 80C188, ESC traps occur in both compatible and enhanced operating modes. The return address of this exception will point to the ESC instruction causing the exception. If a segment override prefix preceded the ESC instruction, the return address will point to the segment override prefix.

Note: Unlike the 80188, all numerics coprocessor opcodes cause a trap. The 80C188 does not support the numerics interface.

Hardware-generated interrupts are divided into two groups: maskable interrupts and non-maskable interrupts. The 80C188 provides maskable hardware interrupt request pins INT0–INT3. In addition, maskable interrupts may be generated by the 80C188 integrated DMA controller and the integrated timer unit. The vector types for these interrupts is shown in Table 3. Software enables these inputs by setting the interrupt flag bit (IF) in the Status Word. The interrupt controller is discussed in the peripheral section of this datasheet.

Further maskable interrupts are disabled while servicing an interrupt because the IF bit is reset as part of the response to an interrupt or exception. The saved Status Word will reflect the enable status of the processor prior to the interrupt. The interrupt flag will remain zero unless specifically set. The interrupt return instruction restores the Status Word, thereby restoring the original status of IF (interrupt-enable flag) bit. If the interrupt return reenables interrupts, and another interrupt is pending, the 80C188 will immediately service the highest-priority interrupt pending, i.e., no instructions of the main line program will be executed.

Initialization and Processor Reset

Processor initialization or startup is accomplished by driving the $\overline{RES}$ input pin Low. $\overline{RES}$ must be Low during power-up to ensure proper device initialization. $\overline{RES}$ forces the 80C188 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as $\overline{RES}$ is active. After $\overline{RES}$ becomes inactive and an internal processing interval elapses, the 80C188 begins execution with the instruction at physical location FFFF0(H). $\overline{RES}$ also sets some registers to predefined values as shown in Table 4.

Table 3. 80C188 Interrupt Vectors

Interrupt Name	Vector Type	Vector Address	Default Priority	Related Instructions	Applicable Notes
Divide Error Exception	0	00H	1	DIV, IDIV	1
Single-Step Interrupt	1	04H	1A	All	2
Non-Maskable Interrupt (NMI)	2	08H	1	All	
Breakpoint Interrupt	3	0CH	1	INT	1
INTO Detected Overflow Exception	4	10H	1	INTO	1
Array Bounds Exception	5	14H	1	BOUND	1
Unused Opcode Exception	6	18H	1	Undefined Opcodes	1
ESC Opcode Exception	7	1CH	1	ESC Opcodes	1, 3
Timer 0 Interrupt	8	20H	2A		4, 5
Timer 1 Interrupt	18	48H	2B		4, 5
Timer 2 Interrupt	19	4CH	2C		4, 5
Reserved	9	24H	3		
DMA 0 Interrupt	10	28H	4		5
DMA 1 Interrupt	11	2CH	5		5
INT0 Interrupt	12	30H	6		
INT1 Interrupt	13	34H	7		
INT2 Interrupt	14	38H	8		
INT3 Interrupt	15	3CH	9		
Reserved	16, 17	40H, 44H			
Reserved	20–31	50H . . . 7CH			

Notes: Default priorities for the interrupt sources are used only if the user does not program each source to a unique priority level.

1. Generated as a result of an instruction execution.
2. Performed in the same manner as 8088.
3. An ESC opcode will cause a trap if the 80C188 is in compatible mode or if the processor is in enhanced mode with the proper bit set in the peripheral control block relocation register. The 80C188 is not directly compatible with the 80188 in this respect.
4. All three timers constitute one source of request to the interrupt controller. As such, they share the same priority level with respect to other interrupt sources. However, the timers have a defined priority order among themselves (2A > 2B > 2C).
5. The vector type numbers of these sources are programmable in Slave Mode.

Table 4. 80C188 Initial Register State after RESET

Status Word	F002(H)
Instruction Pointer	0000(H)
Code Segment	FFFF(H)
Data Segment	0000(H)
Extra Segment	0000(H)
Stack Segment	0000(H)
Relocation Register	20FF(H)
UMCS	FFFB(H)

THE 80C188 COMPARED TO THE 80C186

The 80C188 is an 8-bit processor designed based on the 80C186 internal structure. Most internal functions of the 80C188 are identical to the equivalent 80C186 functions. The 80C188 handles the external bus the same way the 80C186 does with the distinction of handling only 8 bits at a time. Sixteen-bit operands are fetched or written in two consecutive bus cycles. The processors will look the same to the software engineer, with the exception of execution time. The internal register structure is identical and all instructions except numerics instructions have the same end result. Internally, there are four differences between the 80C188 and the 80C186. All changes are related to the 8-bit bus interface.

- The queue length is 4 bytes in the 80C188, whereas the 80C186 queue contains 6 bytes, or three words. The queue was shortened to prevent overuse of the bus by the BIU when prefetching instructions. This was required because of the additional time necessary to fetch instructions 8 bits at a time.
- To further optimize the queue, the prefetching algorithm was changed. The 80C188 BIU will fetch a new instruction to load into the queue each time there is a 1-byte hole (space available) in the queue. The 80C186 waits until a 2-byte space is available.
- The internal execution time of an instruction is affected by the 8-bit interface. All 16-bit fetches and writes from/to memory take an additional four-clock cycles. The CPU may also be limited by the rate of instruction fetches when a series of simple operations occur. When the more sophisticated instructions of the 80C188 are being used, the queue has more time to fill and the execution proceeds more closely to the speed at which the execution unit will allow.
- The 80C188 does not have a numerics interface, since the 80C186 numerics interface inherently requires 16-bit communication with the numerics coprocessor.

The 80C188 and 80C186 are completely software compatible (except for numerics instructions) by virtue of their identical execution units. However, software that is system dependent may not be completely transferable.

The bus interface and associated control signals vary somewhat between the two processors. The pin assignments are newly identical, with the following functional changes:

- A15–8—These pins are only address outputs on the 80C188. These address lines are latched internally and remain valid throughout the bus cycle.
- $\overline{BHE}$ has no meaning on the 80C188. However, it was necessary to designate this pin the $\overline{RFSH}$ pin in order to provide an indication of DRAM refresh bus cycles.

80C188 CLOCK GENERATOR

The 80C188 provides an on-chip clock generator for both internal and external clock generation. The clock generator features a crystal oscillator, a divide-by-two counter, synchronous and asynchronous ready inputs, and reset circuitry.

Oscillator

The oscillator circuit of the 80C188 is designed to be used either with a parallel resonant fundamental or third-overtone mode crystal, depending upon the frequency range of the application, as shown in Figure 6C. This is used as the time base for the 80C188. The crystal frequency chosen should be twice the required processor frequency. Use of an LC or RC circuit is not recommended.

The output of the oscillator is not directly available outside the 80C188. The two recommended crystal configurations are shown in Figures 6A and 6B. When used in third-overtone mode the tank circuit shown in Figure 6B is recommended for stable operation. The sum of the stray capacitances and loading capacitors should equal the values shown. It is advisable to limit stray capacitance between the X1 and X2 pins to less than 10 pF. While a fundamental-mode circuit will require approximately 1 ms for start-up, the third-overtone arrangement may require 1 ms to 3 ms to stabilize.

Alternately, the oscillator may be driven from an external source, as shown in Figure 6D. The configuration shown in Figure 6E is not recommended.

The following parameters should be used when choosing a crystal:

Temperature Range:	0 to 70°C
ESR (Equivalent Series Resistance):	40 ohms max
C0 (Shunt Capacitance of Crystal):	7.0 pF max
C1 (Load Capacitance):	20 pF ±2 pF
Drive Level:	1 mW max

Clock Generator

The 80C188 clock generator provides the 50%-duty cycle processor clock for the 80C188. It does this by dividing the oscillator output by 2, forming the symmetrical clock. If an external oscillator is used, the state of the clock generator will change on the falling edge of the oscillator signal. The CLKOUT pin provides the processor clock signal for use outside the 80C188. This may be used to drive other system components. All timings are referenced to the output clock.

READY Synchronization

The 80C188 provides both synchronous and asynchronous ready inputs. Asynchronous ready synchronization is accomplished by circuitry which samples ARDY in the middle of T2, T3, and again in the middle of each TW until ARDY is sampled High. One-half CLKOUT cycle of resolution time is used for full synchronization of a rising ARDY signal. A High-to-Low transition on ARDY may be used as an indication of the not-ready condition, but it must be performed synchronously to CLKOUT, either in the middle of T2, T3, or TW, or at the falling edge of T3 or TW.

A second ready input (SRDY) is provided to interface with externally synchronized ready signals. This input is sampled at the end of T2, T3, and again at the end of each TW until it is sampled High. By using this input rather than the asynchronous ready input, the half-clock cycle resolution time penalty is eliminated. This input must satisfy set-up and hold times to guarantee proper operation of the circuit.

In addition, the 80C188, as part of the integrated chip-select logic, has the capability to program WAIT states for memory and peripheral blocks. This is discussed in the Chip Select/ Ready Logic description.

RESET Logic

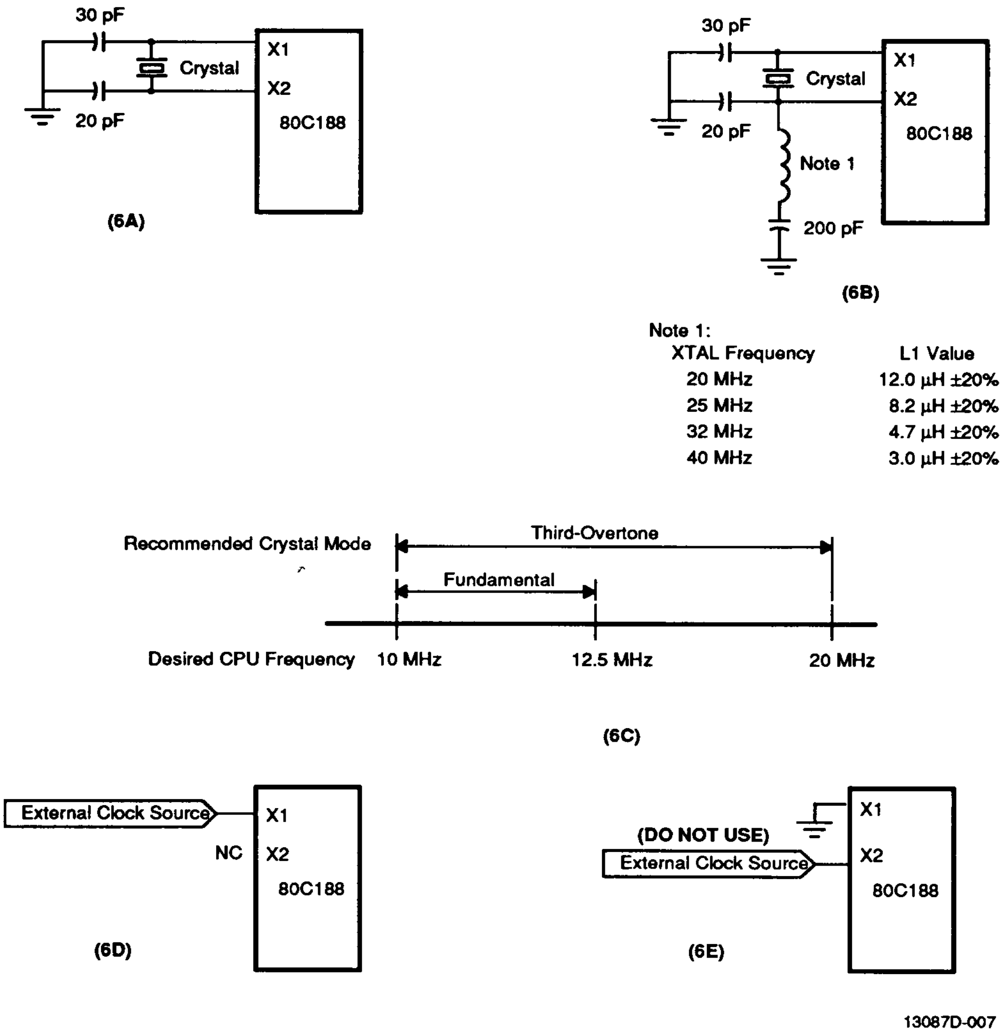
The 80C188 provides both a $\overline{RES}$ input pin and a synchronized RESET output pin for use with other system components. The $\overline{RES}$ input pin on the 80C188 is provided with hysteresis in order to facilitate power-on Reset generation via an RC network. RESET is guaranteed to remain active for at least five clocks given a $\overline{RES}$ input of at least six clocks. RESET may be delayed up to approximately two and one-half clocks behind $\overline{RES}$.

LOCAL BUS CONTROLLER

The 80C188 provides a local bus controller to generate the local bus control signals. In addition, it employs a HOLD/HLDA protocol for relinquishing the local bus to other bus masters. It also provides outputs that can be used to enable external buffers and to direct the flow of data on and off the local bus.

Memory/Peripheral Control

The 80C188 provides ALE, $\overline{RD}$, and $\overline{WR}$ bus control signals. The $\overline{RD}$ and $\overline{WR}$ signals are used to strobe data from memory or I/O to the 80C188 or to strobe data from the 80C188 to memory or I/O. The ALE line provides a strobe to latch the address when it is valid. The 80C188 local bus controller does not provide a memory/$\overline{I/O}$ signal. If this is required, use the $\overline{S2}$ signal (which will require external latching, 0 = I/O and 1 = memory), make the memory and I/O spaces non-overlapping, or use only the integrated chip-select circuitry.



Note 1:

XTAL Frequency	L1 Value
20 MHz	12.0 µH ±20%
25 MHz	8.2 µH ±20%
32 MHz	4.7 µH ±20%
40 MHz	3.0 µH ±20%

Figure 6. 80C188 Oscillator Configurations (see text)

Transceiver Control

The 80C188 generates two control signals for external transceiver chips. This capability allows the addition of transceivers for extra buffering without adding external logic. These control lines, DT/R and DEN, are generated to control the flow of data through the transceivers. The operation of these signals is shown in Table 5.

Table 5. Transceiver Control Signals Description

Pin Name	Function
DEN (Data Enable)	Enables the output drivers of the transceivers. It is active Low during memory, I/O, numeric processor extension, or INTA cycles.
DT/R (Data Transmit/ Receive)	Determines the direction of travel through the transceivers. A High level directs data away from the processor during write operations, while a Low level directs data toward the processor during a read operation.

Local Bus Arbitration

The 80C188 uses a HOLD/HLDA system of local bus exchange. This provides an asynchronous bus exchange mechanism. This means multiple masters utilizing the same bus can operate at separate clock frequencies. The 80C188 provides a single HOLD/HLDA pair through which all other bus masters may gain control of the local bus. External circuitry must arbitrate which external device will gain control of the bus when there is more than one alternate local bus master. When the 80C188 relinquishes control of the local bus, it floats DEN, RD, WR, S2–S0, LOCK, AD7–AD0, A19–A8, S7/RFSH, and DT/R to allow another master to drive these lines directly.

The 80C188 HOLD latency time, that is, the time between HOLD request and HOLD acknowledge, is a function of the activity occurring in the processor when the HOLD request is received. A HOLD request is second only to DRAM refresh requests in priority of activity requests the processor may receive. Any bus cycle in progress will be completed before the 80C188 relinquishes the bus. This implies that if a HOLD request is received just as a DMA transfer begins, the HOLD latency can be as great as 4-bus cycles. This will occur if a DMA word transfer operation is taking place from an odd address to an odd address. This is a total of 16-clock cycles or more if WAIT states are required. In addition, if locked transfers are performed, the HOLD latency time will be increased by the length of the locked transfer.

If the 80C188 has relinquished the bus and a refresh request is pending, HLDA is removed (driven Low) to signal the remote processor that the 80C188 wishes to regain control of the bus. The 80C188 will wait until HOLD is removed before taking control of the bus to run the refresh cycle.

Local Bus Controller and Reset

During RESET, the local bus controller will perform the following action:

- Drive DEN, RD, and WR High for one clock cycle, then float them.
- Drive S2–S0 to the inactive state (all High) and then float.
- Drive LOCK High and then float.
- Float AD7–AD0, A19–A8, S7/RFSH, DT/R.
- Drive ALE Low.
- Drive HLDA Low.

RD/QSMD, UCS, LCS, and TEST pins have internal pull-up devices which are active while RES is applied. Excessive loading or grounding certain of these pins causes the 80C188 to enter an alternative mode of operation:

- RD/QSMD Low results in Queue Status Mode.
- UCS and LCS Low results in ONCE Mode.
- TEST Low (and High later) results in Enhanced Mode.

INTERNAL PERIPHERAL INTERFACE

All the 80C188 integrated peripherals are controlled by 16-bit registers contained within an internal 256-byte control block. The control block may be mapped into either memory or I/O space. Internal logic will recognize control block addresses and respond to bus cycles. During bus cycles to internal registers, the bus controller will signal the operation externally (i.e., the RD, WR, status, address, data, etc., lines will be driven as in a normal bus cycle), but AD7–AD0, SRDY, and ARDY will be ignored. The base address of the control block must be on an even 256-byte boundary (i.e., the lower 8 bits of the base address are all 0s). All of the defined registers within this control block may be read or written by the 80C188 CPU at any time.

The control block base address is programmed by a 16-bit relocation register contained within the control block at offset FEH from the base address of the control block (see Figure 7). It provides the upper 12 bits of the base address of the control block. The control block is effectively an internal chip select range and must abide by all the rules concerning chip selects (the chip select circuitry is discussed later in this datasheet). Any access to the 256 bytes of the control block activates an internal chip select.

Offset: FEH	15	14	13	12	11–0
	ET	Slave/$\overline{\text{Master}}$	X	M/$\overline{\text{IO}}$	Relocation Address Bits R19–R8

ET = ESC Trap/No ESC Trap (1/0)
M/$\overline{\text{IO}}$ = Register block located in Memory/ I/O Space (1/0)
Slave/$\overline{\text{Master}}$ = Configures interrupt controller for Slave/Master Mode (1/0)

Figure 7. Relocation Register 13087D-008

Other chip selects may overlap the control block only if they are programmed to zero wait states and ignore external ready. In addition, bit 12 of this register determines whether the control block will be mapped into I/O or memory space. If this bit is 1, the control block will be located in memory space. If the bit is 0, the control block will be located in I/O space. If the control register block is mapped into I/O space, the upper 4 bits of the base address must be programmed as 0 (since I/O addresses are only 16-bits wide).

In addition to providing relocation information for the control block, the relocation register contains bits which place the interrupt controller into slave mode and cause the CPU to interrupt upon encountering ESC instructions. At RESET, the relocation register is set to 20FFH, which maps the control block to start at FF00H in I/O space. An offset map of the 256-byte control register block is shown in Figure 8.

CHIP-SELECT/READY GENERATION LOGIC

The 80C188 contains logic which provides programmable chip-select generation for both memories and peripherals. In addition, it can be programmed to provide READY (or WAIT state) generation. It can also provide latched address bits A1 and A2. The chip-select lines are active for all memory and I/O cycles in their programmed areas, whether they be generated by the CPU or by the integrated DMA unit.

Memory Chip Selects

The 80C188 provides six memory chip select outputs for three address areas: upper memory, lower memory, and mid-range memory. One each is provided for upper memory and lower memory, while four are provided for mid-range memory.

The range for each chip select is user-programmable and can be set to 2K, 4K, 8K, 16K, 32K, 64K, 128K (plus 1K and 256K for upper and lower chip selects). In addition, the beginning or base address of the mid-range memory chip select may also be selected. Only one chip select may be programmed to be active for any memory location at a time. All chip select sizes are in bytes, whereas 80C188 memory is arranged in words. This means that if, for example, 16 64K×1 memories are used, the memory block size will be 128K, not 64K.

Register	OFFSET
Relocation Register	FEH
DMA Descriptors Channel 1	DAH D0H
DMA Descriptors Channel 0	CAH C0H
Chip-Select Control Registers	ABH A0H
Time 2 Control Registers	66H 60H
Time 1 Control Registers	5EH 58H
Time 0 Control Registers	56H 50H
Interrupt Controller Registers	3EH 20H

13087D-009

Figure 8. Internal Register Map

Upper Memory $\overline{\text{CS}}$

The 80C188 provides a chip select, called $\overline{\text{UCS}}$, for the top of memory. The top of memory is usually used as the system memory because after reset the 80C188 begins executing at memory location FFFF0H.

The upper limit of memory defined by this chip select is always FFFFFH, while the lower limit is programmable. By programming the lower limit, the size of the select block is also defined. Table 6 shows the relationship between the base address selected and the size of the memory block obtained.

Table 6. UMCS Programming Values

Starting Address (Base Address)	Memory Block Size	UMCS Value (Assuming R0 = R1 = R2 = 0)
FFC00	1K	FFF8H
FF800	2K	FFB8H
FF000	4K	FF38H
FE000	8K	FE38H
FC000	16K	FC38H
F8000	32K	F838H
F0000	64K	F038H
E0000	128K	E038H
C0000	256K	C038H

The lower limit of this memory block is defined in the UMCS register (see Figure 9). This register is at offset A0H in the internal control block. The legal values for bits 13–6 and the resulting starting address and memory block sizes are given in Table 6. Any combination of bits 13–6 not shown in Table 6 will result in undefined operation. After reset, the UMCS register is programmed for a 1K area. It must be reprogrammed if a larger upper memory area is desired.

The internal generation of any 20-bit address whose upper 16 bits are equal to or greater than the UMCS value (with bits 5–0 as 0) asserts $\overline{UCS}$. UMCS bits R2–R0 specify the READY mode for the area of memory defined by the chip-select register, as explained later.

Lower Memory $\overline{CS}$

The 80C188 provides a chip select for low memory called $\overline{LCS}$. The bottom of memory contains the interrupt vector table, starting at location 00000H.

The lower limit of memory defined by this chip select is always 0H, while the upper limit is programmable. By programming the upper limit, the size of the memory block is defined. Table 7 shows the relationship between the upper address selected and the size of the memory block obtained.

Table 7. LMCS Programming Values

Upper Address	Memory Block Size	LMCS Value (Assuming R0 = R1 = R2 = 0)
003FFH	1K	0038H
007FFH	2K	0078H
00FFFH	4K	00F8H
01FFFH	8K	01F8H
03FFFH	16K	03F8H
07FFFH	32K	07F8H
0FFFFH	64K	0FF8H
1FFFFH	128K	1FF8H
3FFFFH	256K	3FF8H

The upper limit of this memory block is defined in the LMCS register (see Figure 10) at offset A2H in the internal control block. The legal values for bits 15–6 and the resulting upper address and memory block sizes are given in Table 7. Any combination of bits 15–6 not shown in Table 7 will result in undefined operation. After reset, the LMCS register value is undefined. However, the $\overline{LCS}$ chip-select line will not become active until the LMCS register is accessed.

Any internally generated 20-bit address whose upper 16 bits are less than or equal to LMCS (with bits 5–0 "1") will assert $\overline{LCS}$. LMCS register bits R2–R0 specify the READY mode for the area of memory defined by this chip-select register.

Mid-Range Memory $\overline{CS}$

The 80C188 provides four $\overline{MCS}$ lines which are active within a user-locatable memory block. This block can be located within the 80C188 1-Mb memory address space exclusive of the areas defined by $\overline{UCS}$ and $\overline{LCS}$. Both the base address and size of this memory block are programmable.

The size of the memory block defined by the mid-range select lines, as shown in Table 8, is determined by bits 14–8 of the MPCS register (see Figure 11). This register is at location A8H in the internal control block. One and only one of bits 14–8 must be set at a time. Unpredictable operation of the $\overline{MCS}$ lines will otherwise occur. Each of the four chip-select lines is active for one of the four equal contiguous divisions of the mid-range block. If the total block size is 32K, each chip select is active for 8K of memory with $\overline{MCS0}$ being active for the first range, $\overline{MCS1}$ active for the second, $\overline{MCS2}$ for the third, and $\overline{MCS3}$ being active for the last range.

The EX and MS in MPCS relate to peripheral functionality, as described in a later section.

Table 8. MPCS Programming Values

Total Block Size	Individual Select Size	MPCS Bits 14–8
8K	2K	0000001B
16K	4K	0000010B
32K	8K	0000100B
64K	16K	0001000B
128K	32K	0010000B
256K	64K	0100000B
512K	128K	1000000B

The base address of the mid-range memory block is defined by bits 15–9 of the MMCS register (see Figure 12). This register is at offset A6H in the internal control block (see Figure 8). These bits correspond to bits A19–A13 of the 20-bit memory address. Bits A12–A0 of the base address are always 0. The base address may be set at any integer multiple of the size of the total memory block selected. For example, if the mid-range block size is 32K (or the size of the block for which each MCS line is active is 8K), the block could be located at 10000H or 18000H, but not at 14000H, since the first few integer multiples of a 32K memory block are 0H, 8000H, 10000H, 18000H, etc. After reset, the contents of both registers are undefined. However, none of the MCS lines will be active until both the MMCS and MPCS registers are accessed.

MMCS bits R2–R0 specify READY mode of operation for all four mid-range chip selects.

The 512K block size for the mid-range memory chip selects is a special case. When using 512K, the base address would have to be at either locations 00000H or 80000H. If it were to be programmed at 00000H when the LCS line was programmed, there would be an internal conflict between the LCS ready generation logic and the MCS ready generation logic. Likewise, if the base address were programmed at 80000H, there would be a conflict with the UCS ready generation logic. Since the LCS chip-select line does not become active until programmed, while the UCS line is active at reset, the memory base can be set only at 00000H. If this base address is selected, however, the LCS range must not be programmed.

Peripheral Chip Selects

The 80C188 can generate chip selects for up to seven peripheral devices. These chip selects are active for seven contiguous blocks of 128 bytes above a programmable base address. The base address may be located in either memory or I/O space.

Seven CS lines called PCS6–0 are generated by the 80C188. The base address is user-programmable; however, it can only be a multiple of 1K bytes, i.e., the least significant 10 bits of the starting address are always 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset: A0H	1	1	U	U	U	U	U	U	U	U	1	1	1	R2	R1	R0
	A19								A11							

Figure 9. UMCS Register

13087D-010

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset: A2H	0	0	U	U	U	U	U	U	U	U	1	1	1	R2	R1	R0
	A19								A11							

Figure 10. LMCS Register

13087D-011

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset: A8H	1	M6	M5	M4	M3	M2	M1	M0	EX	MS	1	1	1	R2	R1	R0

Figure 11. MPCS Register

13087D-012

	15						9						3			0
Offset: A6H	U	U	U	U	U	U	U	1	1	1	1	1	1	R2	R1	R0
	A19						A13									

Figure 12. MMCS Register

13087D-013

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset: A4H	U	U	U	U	U	U	U	U	U	U	1	1	1	R2	R1	R0
	A19									A10						

Figure 13. PACS Register

13087D-014

PCS5 and PCS6 can also be programmed to provide latched address bits A1 and A2. If so programmed, they cannot be used as peripheral selects. These outputs can be connected directly to the A0 and A1 pins used for selecting internal registers of 8-bit peripheral chips. This scheme simplifies the external hardware because the peripheral registers can be located on even boundaries in I/O or memory space.

The starting address of the peripheral chip-select block is defined by the PACS register (see Figure 13). The register is located at offset A4H in the internal control block. Bits 15–6 of this register correspond to bits 19–10 of the 20-bit Programmable Base Address (PBA) of the peripheral chip-select block. Bits 9–0 of the PBA of the peripheral chip-select block are all 0s. If the chip-select block is located in I/O space, bits 15–12 must be programmed 0, since the I/O address is only 16-bits wide. Table 9 shows the address range of each peripheral chip select with respect to the PBA contained in PACS register.

The user should program bits 15–6 to correspond to the desired peripheral base location. PACS bits 2–0 are used to specify READY mode for PCS3–PCS0.

Table 9. PCS Address Ranges

PCS Line	Active between Locations
PCS0	PBA —PBA+127
PCS1	PBA+128—PBA+255
PCS2	PBA+256—PBA+383
PCS3	PBA+384—PBA+511
PCS4	PBA+512—PBA+639
PCS5	PBA+640—PBA+767
PCS6	PBA+768—PBA+895

The mode of operation of the peripheral chip selects is defined by the MPCS register (which is also used to set the size of the mid-range memory chip-select block, see Figure 11). The register is located at offset A8H in the internal control block. Bit 7 is used to select the function of PCS5 and PCS6, while bit 6 is used to select whether the peripheral chip selects are mapped into memory or I/O space. Table 10 describes the programming of these bits. After reset, the contents of both the MPCS and the PACS registers are undefined; however, none of the PCS lines will be active until both of the MPCS and PACS registers are accessed.

Table 10. MS, EX Programming Values

Bit	Description
MS	1 = Peripherals mapped into memory space. 0 = Peripherals mapped into I/O space.
EX	0 = 5 PCS lines. A1, A2 provided. 1 = 7 PCS lines. A1, A2 are not provided.

MPCS bits 2–0 specify the READY mode for PCS6–PCS4, as outlined below.

READY Generation Logic

The 80C188 can generate a READY signal internally for each of the memory or peripheral CS lines. The number of WAIT states to be inserted for each peripheral or memory is programmable to provide 3–0 WAIT states for all accesses to the area for which the chip select is active. In addition, the 80C188 may be programmed to either ignore external READY for each chip-select range individually or to factor external READY with the integrated ready generator.

READY control consists of 3 bits for each CS line or group of lines generated by the 80C188. The interpretation of the READY bits is shown in Table 11.

Table 11. READY Bits Programming

R2	R1	R0	Number of WAIT States Generated
0	0	0	0 wait states, external RDY also used.
0	0	1	1 wait state inserted, external RDY also used.
0	1	0	2 wait states inserted, external RDY also used.
0	1	1	3 wait states inserted, external RDY also used.
1	0	0	0 wait states, external RDY ignored.
1	0	1	1 wait state inserted, external RDY ignored.
1	1	0	2 wait states inserted, external RDY ignored.
1	1	1	3 wait states inserted, external RDY ignored.

The internal ready generator operates in parallel with external READY, not in series if the external READY is used (R2 = 0). For example, if the internal generator is set to insert two wait states, but activity on the external READY lines will insert four wait states, the processor will only insert four wait states, not six. This is because the two wait states generated by the internal generator overlapped the first two wait states generated by the external ready signal. Note that the external ARDY and SRDY lines are always ignored during cycles accessing internal peripherals.

R2–R0 of each control word specifies the READY mode for the corresponding block, with the exception of the peripheral chip selects: R2–R0 of PACS set the PCS3–0 READY mode, R2–R0 of MPCS set the PCS6–4 READY mode.

Chip Select/Ready Logic and Reset

Upon RESET, the Chip-Select/Ready Logic will perform the following actions:

- All chip-select outputs will be driven High.
- Upon leaving RESET, the UCS line will be programmed to provide chip selects to a 1K block with the accompanying READY control bits set at 011 to insert 3 wait states in conjunction with external Ready (i.e., UMCS resets to FFBH).

- No other chip select or READY control registers have any predefined values after RESET. They will not become active until the CPU accesses their control registers. Both the PACS and MPCS registers must be accessed before the $\overline{PCS}$ lines will become active.

DMA CHANNELS

The 80C188 DMA controller provides two independent high-speed DMA channels. Data transfers can occur between memory and I/O spaces (e.g., Memory to I/O) or within the same space (e.g., Memory to Memory or I/O to I/O). Data can be transferred either in bytes (8 bits) or in words (16 bits) to or from even or odd addresses. Each DMA channel maintains both a 20-bit source and destination pointer which can be optionally incremented or decremented after each data transfer (by one or two, depending on byte or word transfers). Each data transfer consumes 2 bus cycles (a minimum of 8 clocks), one cycle to fetch data and the other to store data.

DMA Operation

Each channel has six registers in the control block which define each channel's specific operation. The control registers consist of a 20-bit Source pointer (2 words), a 20-bit Destination Pointer (2 words), a 16-bit Transfer Count Register, and a 16-bit Control Word. The format of the DMA Control Blocks is shown in Table 12. The Transfer Count Register (TC) specifies the number of DMA transfers to be performed. Up to 64K byte or word transfers can be performed with automatic termination. The Control Word defines the channel's operation (see Figure 15). All registers may be modified or altered during any DMA activity. Any changes made to these registers will be reflected immediately in DMA operation.

Table 12. DMA Control Block Format

Register Name	Register Address Ch 0	Ch1
Control Word	CAH	DAH
Transfer Control	C8H	D8H
Destination Pointer (upper 4 bits)	C6H	D6H
Destination Pointer	C4H	D4H
Source Pointer (upper 4 bits)	C2H	D2H
Source Pointer	COH	DOH

DMA Channel Control Word Register

Each DMA Channel Control Word determines the mode of operation for the particular 80C188 DMA channel. This register specifies:

- the mode of synchronization;
- whether bytes or words will be transferred;
- whether interrupts will be generated after the last transfer;
- whether DMA activity will cease after a programmed number of DMA cycles;
- the relative priority of the DMA channel with respect to the other DMA channel;
- whether the source pointer will be incremented, decremented, or maintained constant after each transfer;
- whether the source pointer addresses memory or I/O space;
- whether the destination pointer will be incremented, decremented, or maintained constant after each transfer; and
- whether the destination pointer will address memory or I/O space.

The DMA channel control registers may be changed while the channel is operating. However, any changes made during operation will affect the current DMA transfer.

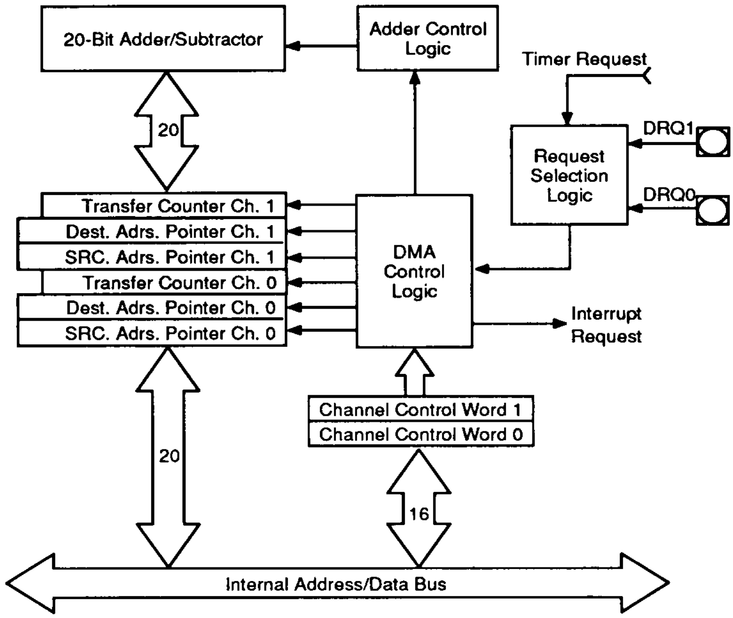


Figure 14. DMA Unit Block Diagram

13087D-015

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M/$\overline{IO}$	Destination DEC	INC	M/$\overline{IO}$	Source DEC	INC	TC	INT	SYN		P	TDRQ	X	CHG/$\overline{NOCHG}$	ST/$\overline{STOP}$	X

X = Don't Care

Figure 15. DMA Control Register

13087D-016

DMA Control Word Bit Descriptions

DEST:

M/$\overline{IO}$ Destination pointer is in memory (1) or I/O (0) space.

DEC Decrement destination pointer by 1 after each transfer.

INC Increment destination pointer by 1 after each transfer.

If both INC and DEC are specified, the pointer will remain constant after each cycle.

SOURCE:

M/$\overline{IO}$: Source pointer is in memory (1) or I/O (0) space.

DEC: Decrement source pointer by 1 after each transfer.

INC Increment source pointer by 1 after each transfer.

If both INC and DEC are specified, the pointer will remain constant after each cycle.

TC: If set, DMA will terminate when the contents of the Transfer Count register reach 0. The ST/$\overline{STOP}$ bit will also be reset at this point. If this bit is cleared, the DMA unit will decrement the transfer count register for each DMA cycle, but the DMA transfer will not stop when the contents of the TC register reach 0.

INT: Enable Interrupts to CPU upon transfer count termination.

SYN: 00 No synchronization

Note: When unsynchronized transfers are specified, the TC bit will be ignored and the ST/$\overline{\text{STOP}}$ bit will be cleared upon the transfer count reaching zero, stopping the channel.

01 Source synchronization.
10 Destination synchronization.
11 Unused.

P: Channel priority relative to other channel during simultaneous requests.

0 Low priority
1 High priority.

Channels will alternate cycles if both are set at same priority level.

TDRQ: Enable/Disable (1/0) DMA requests from Timer 2.

CHG/$\overline{\text{NOCHG}}$: Change/Do not change (1/0) ST/$\overline{\text{STOP}}$ bit. If this bit is set when writing to the control word, the ST/$\overline{\text{STOP}}$ bit will be programmed by the write to the control word. If this bit is cleared when writing the control word, the ST/$\overline{\text{STOP}}$ bit will not be altered. This bit is not stored; it will always be read as 0.

ST/$\overline{\text{STOP}}$: Start/Stop (1/0) channel.

DMA Destination and Source Pointer Registers

Each DMA channel maintains a 20-bit source and a 20-bit destination pointer. Each of these pointers takes up two full 16-bit registers in the peripheral control block. For each DMA channel to be used, all four pointer registers must be initialized. The lower four bits of the upper register contain the upper four bits of the 20-bit physical address (see Figure 16). These pointers may be individually incremented or decremented after each transfer. If word transfers are performed, the pointer is incremented or decremented by two.

Each pointer may point into either memory or I/O space. Since the upper four bits of the address are not automatically programmed to zero, the user must program them in order to address the normal 64K I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the pointer registers. Higher transfer rates can be achieved if all word transfers are performed to or from even addresses so that accesses will occur in single bus cycles.

DMA Transfer Count Register

Each DMA channel maintains a 16-bit transfer count register (TC). This register is decremented after every DMA cycle, regardless of the state of the TC bit in the DMA Control Register. If the TC bit in the DMA control word is set or if unsynchronized transfers are programmed, however, DMA activity will terminate when the transfer counter register reaches 0.

DMA Requests

Data transfers may be either source or destination synchronized, that is, either the source of the data or the destination of the data may request the data transfer. In addition, DMA transfers may be unsynchronized; that is, the transfer will take place continually until the correct number of transfers has occurred. When source or unsynchronized transfers are performed, the DMA channel may begin another transfer immediately after the end of a previous DMA transfer. This allows a complete transfer to take place every 2-bus cycles or 8-clock cycles (assuming no wait states). When destination synchronization is performed, data will not be fetched from the source address until the destination device signals that it is ready to receive it. When destination synchronized transfers are requested, the DMA controller will relinquish control of the bus after every transfer. If no other bus activity is initiated, another DMA cycle will begin after two processor clocks. This allows the destination device time to remove its request if another transfer is not desired. Since the DMA controller will relinquish the bus, the CPU can initiate a bus cycle. As a result, a complete bus cycle will often be inserted between destination-synchronized transfers. Table 13 shows the maximum DMA transfer rates.

Table 13. Maximum DMA Transfer Rates at 16 MHz

Type of Synchronization Selected	CPU Running	CPU Halted
Unsynchronized	2.0 Mb/s	2.0 Mb/s
Source Synch	2.0 Mb/s	2.0 Mb/s
Destination Synch	1.3 Mb/s	1.6 Mb/s

	15			0
Higher Register Address	XXX	XXX	XXX	A19 – A16
Lower Register Address	A15 – A12	A11 – A8	A7 – A4	A3 – A0

XXX = Don't Care

13087D-017

Figure 16. DMA Pointer Register Format

DMA Acknowledge

No explicit DMA acknowledge pulse is provided. Since both source and destination pointers are maintained, a read from a requesting source, or a write to a requesting destination, should be used as the DMA acknowledge signal. Since the chip-select lines can be programmed to be active for a given block of memory or I/O space, and the DMA pointers can be programmed to point to the same given block, a chip-select line could be used to indicate a DMA acknowledge.

DMA Priority

The DMA channels may be programmed such that one channel is always given priority over the other, or they may be programmed such as to alternate cycles when both have DMA requests pending. DMA cycles always have priority over internal CPU cycles except between locked memory accesses or word accesses to odd memory locations; however, an external bus hold takes priority over an internal DMA cycle. Because an interrupt request cannot suspend a DMA operation and the CPU cannot access memory during a DMA cycle, interrupt latency time will suffer during sequences of continuous DMA cycles. An NMI request, however, will cause all internal DMA activity to halt. This allows the CPU to quickly respond to the NMI request.

DMA Programming

DMA cycles will occur whenever the ST/$\overline{\text{STOP}}$ bit of the Control Register is set. If synchronized transfers are programmed, a DRQ must also be generated. Therefore, the source and destination transfer pointers, and the transfer count register (if used) must be programmed before the ST/$\overline{\text{STOP}}$ bit is set.

Each DMA register may be modified while the channel is operating. If the CHG/$\overline{\text{NOCHG}}$ bit is cleared when the control register is written, the ST/$\overline{\text{STOP}}$ bit of the control register will not be modified by the write. If multiple channel registers are modified, it is recommended that a LOCKED string transfer be used to prevent a DMA transfer from occurring between updates to the channel registers.

DMA Channels and Reset

Upon RESET, the state of the DMA channels will be as follows:

- The ST/$\overline{\text{STOP}}$ bit for each channel will be reset to $\overline{\text{STOP}}$.
- Any transfer in progress is aborted.
- The values of the transfer count registers, source pointers, and destination pointers are indeterminate.

TIMERS

The 80C188 provides three internal 16-bit programmable timers (see Figure 17). Two of these are highly flexible and are connected to four external pins (2 per timer). They can be used to count external events, time external events, generate non-repetitive waveforms, etc. The third timer is not connected to any external pins, and is useful for real-time coding and time delay applications. In addition, this third timer can be used as a prescaler to the other two, or as a DMA request source.

Timer Operation

The timers are controlled by eleven 16-bit registers in the peripheral control block. The configuration of these registers is shown in Table 14. The count register contains the current value of the timer. It can be read or written at any time independent of whether the timer is running or not. The value of this register will be incremented for each timer event. Each of the timers is equipped with a MAX COUNT register, which defines the maximum count the timer will reach. After reaching the MAX COUNT register value, the timer count value will reset to 0 during that same clock, that is, the maximum count value is never stored in the count register itself. Timers 0 and 1 are, in addition, equipped with a second MAX COUNT register, which enables the timers to alternate their count between two different MAX COUNT values. If a single MAX COUNT register is used, the timer output pin will switch Low for a clock, one clock after the maximum count value has been reached. In the dual MAX COUNT register mode, the output pin will indicate which MAX COUNT register is currently in use, thus allowing nearly complete freedom in selecting waveform duty cycles. For the timers with two MAX COUNT registers, the RIU bit in the control register determines which is used for the comparison.

Each timer gets serviced every fourth CPU-clock cycle, and thus can operate at speeds up to one-quarter the internal clock frequency (one-eighth the crystal rate). External clocking of the timers may be done at up to a rate of one-quarter of the internal CPU-clock rate. Due to internal synchronization and pipelining of the timer circuitry, a timer output may take up to six clocks to respond to any individual clock or gate input.

Since the count registers and the maximum count registers are all 16-bits wide, 16 bits of resolution are provided. However, any Read or Write access to the timers will add one wait state to the minimum four-clock bus cycle. This is needed to synchronize and coordinate the internal data flows between the internal timers and the internal bus.

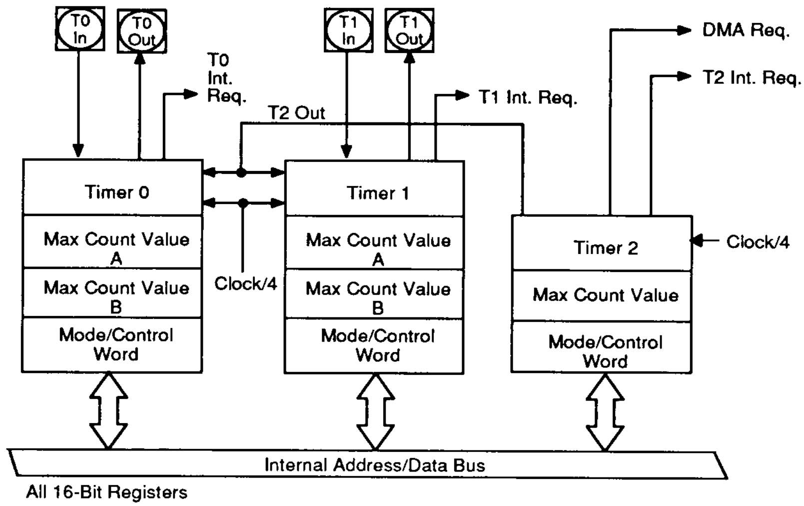


Figure 17. Timer Block Diagram

The timers have several programmable options:

- All three timers can be set to halt or continue on a terminal count.
- Timers 0 and 1 can select between internal and external clocks, alternate between MAX COUNT registers, and be set to retrigger on external events.
- The timers may be programmed to cause an interrupt on terminal count.

These options are selectable via the timer mode/control word.

Timer Mode/Control Register

The mode/control register (see Figure 18) allows the user to program the specific mode of operation or check the current programmed status for any of the three integrated timers.

Table 14. Timer Control Block Format

Register Name	Register Offset Tmr. 0	Tmr. 1	Tmr. 2
Mode/Control Word	56H	5EH	66H
Max Count B	54H	5CH	Not present
Max Count A	52H	5AH	62H
Count Register	50H	58H	60H

EN

The Enable bit provides programmer control over the timer's RUN/HALT status. When set, the timer is enabled to increment subject to the input pin constraints in the internal clock mode (discussed previously). When cleared, the timer will be inhibited from counting. All input pin transitions during the time EN is 0 will be ignored.

If CONT is 0, the EN bit is automatically cleared upon maximum count.

$\overline{\text{INH}}$

The Inhibit bit allows the selective updating of the enable (EN) bit. If $\overline{\text{INH}}$ is a 1 during the write to the mode/control word, then the state of the EN bit will be modified by the write. If $\overline{\text{INH}}$ is a 0 during the write, the EN bit will be unaffected by the operation. This bit is not stored; it will always be a 0 on a read.

INT

When set, the INT bit enables interrupts from the timer, which will be generated on every terminal count. If the timer is configured in dual MAX COUNT register mode, an interrupt will be generated each time the value in MAX COUNT register A is reached and each time the value in MAX COUNT register B is reached. If this enable bit is cleared after the interrupt request has been generated, but before a pending interrupt is serviced, the interrupt request will still be in force. (The request is latched in the Interrupt Controller.)

RIU

The Register In Use bit indicates which MAX COUNT register is currently being used for comparison to the timer count value. A 0 value indicates register A. The RIU bit cannot be written, i.e., its value is not affected when the control register is written. It is always cleared when the ALT bit is 0.

MC

The Maximum Count bit is set whenever the timer reaches its final maximum count value. If the timer is configured in dual MAX COUNT register mode, this bit will be set each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. This bit is set regardless of the timer's interrupt-enable bit. The MC bit gives the user the ability to monitor timer status through software instead of through interrupts.

Programmers' intervention is required to clear this bit.

RTG

Retrigger bit is only active for internal clocking (EXT = 0). In this case it determines the control function provided by the input pin.

If RTG = 0, the input level gates the internal clock on and off. If the input pin is High, the timer will count; if the input pin is Low, the timer will hold its value. As indicated previously, the input signal may be asynchronous with respect to the 80C188 clock.

When RTG = 1, the input pin detects Low-to-High transitions. The first such transition starts the timer running, clearing the timer value to 0 on the first clock, and then incrementing thereafter. Further transitions on the input pin will again reset the timer to 0, from which it will start counting up again. If CONT = 0, when the timer has reached maximum count, the EN bit will be cleared, inhibiting further timer activity.

P

The Prescaler bit is ignored unless internal clocking has been selected (EXT = 0). If the P bit is a 0, the timer will count at one-fourth the internal CPU clock rate. If the P bit is a 1, the output of Timer 2 will be used as a clock for the timer. Note that the user must initialize and start Timer 2 to obtain the prescaled clock.

EXT

The External bit selects between internal and external clocking for the timer. The external signal may be asynchronous with respect to the 80C188 clock.

If this bit is set, the timer will count Low-to-High transitions on the input pin. If cleared, it will count an internal clock while using the input pin for control. In this mode, the function of the external pin is defined by the RTG bit. The maximum input to output transition latency time may be as much as six clocks. However, clock inputs may be pipelined as closely together as every four clocks without losing clock pulses.

ALT

The ALT bit determines which of two MAX COUNT registers is used for count comparison. If ALT = 0, register A for that timer is always used, while if ALT = 1, the comparison will alternate between register A and register B when each maximum count is reached. This alternation allows the user to change one MAX COUNT register while the other is being used, and thus provides a method of generating non-repetitive waveforms. Square waves and pulse outputs of any duty cycle are a subset of available signals obtained by not changing the final count registers. The ALT bit also determines the function of the timer output pin. If ALT is 0, the output pin will go Low for one clock, the clock after the maximum count is reached. If ALT is 1, the output pin will reflect the current MAX COUNT register being used (0/1 for B/A).

CONT

Setting the CONT bit causes the associated timer to run continuously, while resetting it causes the timer to halt upon maximum count. If CONT = 0 and ALT = 1, the timer will count to the MAX COUNT register A value, reset, count to the register B value, reset, and halt.

Not all mode bits are provided for Timer 2. Certain bits are hardwired as indicated below:

ALT = 0, EXT = 0, P = 0, RTG = 0, RIU = 0

15	14	13	12	11		5	4	3	2	1	0
EN	$\overline{INH}$	INT	RIU	0	....	MC	RTG	P	EXT	ALT	CONT

Figure 18. Timer Mode/Control Register 13087D-019

Count Registers

Each of the three timers has a 16-bit count register. The contents of this register may be read or written by the processor at any time. If the register is written into while the timer is counting, the new value will take effect in the current count cycle.

The count registers should be programmed before attempting to use the timers, since they are not automatically initialized to zero.

Max Count Registers

Timers 0 and 1 have two MAX COUNT registers, while Timer 2 has a single MAX COUNT register. These contain the number of events the timer will count. In timers 0 and 1, the MAX COUNT register used can alternate between the two MAX COUNT values whenever the current maximum count is reached. A timer resets when the timer count register equals the MAX COUNT value being used. If the timer count register or the MAX COUNT register is changed so that the MAX COUNT is less than the timer count the timer does not immediately reset. Instead, the timer counts up to 0FFFFH, "wraps around" to zero, counts up to the MAX COUNT value, and then resets.

Timers and Reset

Upon RESET, the Timers will perform the following actions:

- All EN (Enable) bits are reset preventing timer counting.
- For Timers 0 and 1, the RIU bits are reset to zero and the ALT bits are set to one. This results in the Timer Out pins going High.
- The contents of the count registers are indeterminate.

INTERRUPT CONTROLLER

The 80C188 can receive interrupts from a number of sources, both internal and external. The internal interrupt controller serves to merge these requests on a priority basis, for individual service by the CPU.

Internal interrupt sources (Timers and DMA channels) can be disabled by their own control registers or by mask bits within the interrupt controller. The 80C188 interrupt controller has its own control register that sets the mode of operation for the controller.

The interrupt controller will resolve priority among requests that are pending simultaneously. Nesting is provided so interrupt service routines for lower priority interrupts may themselves be interrupted by higher priority interrupts. A block diagram of the interrupt controller is shown in Figure 19.

The 80C188 has a special slave mode in which the internal interrupt controller acts as a slave to an external master. The controller is programmed into this mode by setting bit 14 in the peripheral control block relocation register (see Slave Mode section).

MASTER MODE OPERATION

Interrupt Controller External Interface

Five pins are provided for external interrupt sources. One of these pins is NMI, the non-maskable interrupt. NMI is generally used for unusual events such as power-fail interrupts. The other four pins may be configured in any of the following ways:

- As four interrupt lines with internally generated interrupt vectors.
- As an interrupt line and interrupt acknowledge line pair (cascade mode) with externally generated interrupt vectors plus two interrupt input lines with internally generated vectors.
- As two pairs of interrupt/interrupt acknowledge lines (cascade mode) with externally generated interrupt vectors.

External sources in the cascade mode use externally generated interrupt vectors. When an interrupt is acknowledged, two $\overline{\text{INTA}}$ cycles are initiated and the vector is read into the 80C188 on the second cycle. The capability to interface to external 82C59A programmable interrupt controllers is provided the inputs are configured in cascade mode.

Interrupt Controller Modes of Operation

The basic modes of operation of the interrupt controller in master mode are similar to the 82C59A. The interrupt controller responds identically to internal interrupts in all three modes; the difference is only in the interpretation of function of the four external interrupt pins. The interrupt controller is set into one of these three modes by programming the correct bits in the INT0 and INT1 control registers. The modes of interrupt controller operation are as follows:

Fully Nested Mode

When in the fully nested mode four pins are used as direct interrupt requests as in Figure 20. The vectors for these four inputs are generated internally. An in-service bit is provided for every interrupt source. If a lower-priority device requests an interrupt while the in-service bit (IS) is set, no interrupt will be generated by the interrupt controller. In addition, if another interrupt request occurs from the same interrupt source while the in-service bit is set, no interrupt will be generated by the interrupt controller. This allows interrupt service routines to operate with interrupts enabled, yet be suspended only by interrupts of higher priority than the in-service interrupt.

When a service routine is completed, the proper IS bit must be reset by writing the proper pattern to the EOI register. This is required to allow subsequent interrupts from this interrupt source and to allow servicing of lower-priority interrupts. An EOI command is executed at the end of the service routine just before the return from interrupt instruction. If the fully nested structure has been upheld, the next highest-priority source with its IS bit set is then serviced.

Cascade Mode

The 80C188 has four interrupt pins and two of them have dual functions. In the fully nested mode, the four pins are used as direct interrupt inputs and the corresponding vectors are generated internally. In the cascade mode, the four pins are configured into interrupt input-dedicated acknowledge signal pairs. The interconnection is shown in Figure 21. INT0 is an interrupt input interfaced to an 82C59A, while INT2/$\overline{\text{INTA0}}$ serves as the dedicated interrupt acknowledge signal to that peripheral. The same is true for INT1 and INT3/$\overline{\text{INTA1}}$. Each pair can selectively be placed in the cascade or non-cascade mode by programming the proper value into INT0 and INT1 control registers. The use of the dedicated acknowledge signals eliminates the need for the use of external logic to generate $\overline{\text{INTA}}$ and device select signals.

The primary cascade mode allows the capability to serve up to 128 external interrupt sources through the use of external master and slave 82C59As. Three levels of priority are created, requiring priority resolution in the 80C188 interrupt controller, the master 82C59As, and the slave 82C59As. If an external interrupt is serviced, one IS bit is set at each of these levels. When the interrupt service routine is completed, up to three end-of-interrupt commands must be issued by the programmer.

Special Fully Nested Mode

This mode is entered by setting the SFNM bit in INT0 or INT1 control register. It enables complete nestability with external 82C59A masters. Normally, an interrupt request from an interrupt source will not be recognized unless the in-service bit for that source is reset. If more than one interrupt source is connected to an external interrupt controller, all of the interrupts will be funneled through the same 80C188 interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt will not be recognized by the 80C188 controller until the 80C188 in-service bit is reset. In special fully nested mode, the 80C188 interrupt controller will allow interrupts from an external pin, regardless of the state of the in-service bit for an interrupt source, in order to allow multiple interrupts from a single pin. An in-service bit will continue to be set, however, to inhibit interrupts from other lower-priority 80C188 interrupt sources.

Special procedures should be followed when resetting IS bits at the end of interrupt service routines. Software polling of the IS register in the external master 82C59A is required to determine if there is more than one bit set. If so, the IS bit in the 80C188 remains active and the next interrupt service routine is entered.

Operation in a Polled Environment

The controller may be used in a polled mode if interrupts are undesirable. When polling, the processor disables interrupts and then polls the interrupt controller whenever it is convenient. Polling the interrupt controller is accomplished by reading the Poll Word (Figure 30). Bit 15 in the poll word indicates to the processor that an interrupt of high enough priority is requesting service. Bits 4–0 indicate to the processor the type vector of the highest-priority source requesting service. Reading the Poll Word causes the in-service bit of the highest-priority source to be set.

It is desirable to be able to read the Poll Word information without guaranteeing service of any pending interrupt, that is, not set the indicated in-service bit. The 80C188 provides a Poll Status Word, in addition to the conventional Poll Word, to allow this to be done. Poll Word information is duplicated in the Poll Status Word, but reading the Poll Status Word does not set the associated in-service bit. These words are located in two adjacent memory locations in the register file.

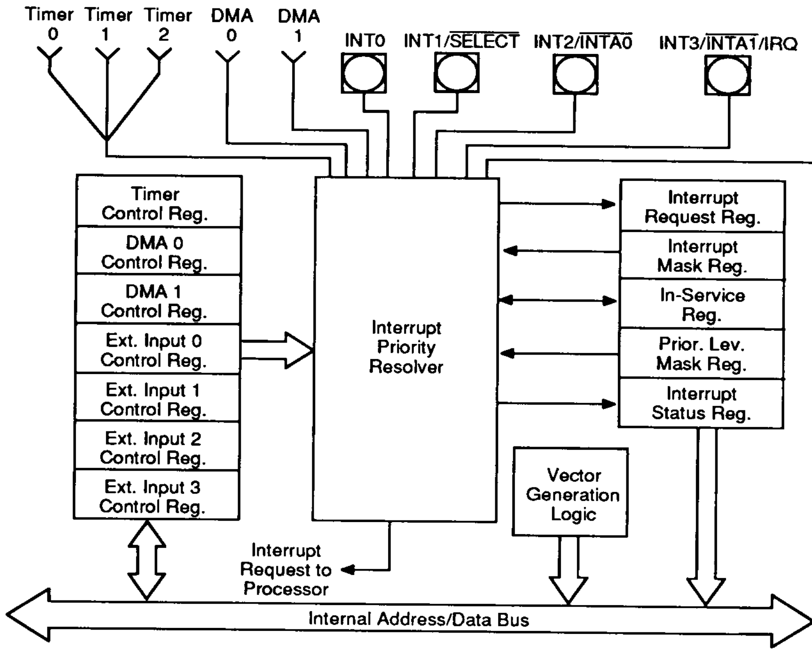


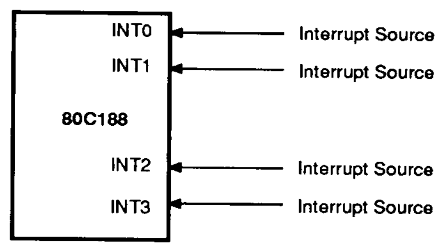
Figure 19. Interrupt Controller Block Diagram 13087D-020

Master Mode Features

Programmable Priority

The user can program the interrupt sources into any of eight different priority levels. The programming is done by placing a 3-bit priority level (0–7) in the control register of each interrupt source. (A source with a priority level of 4 has higher priority over all priority levels from 5 to 7. Priority registers containing values lower than 4 have greater priority.) All interrupt sources have preprogrammed default priority levels (see Table 3).

If two requests with the same programmed priority level are pending at once, the priority ordering scheme shown in Table 3 is used. If the serviced interrupt routine reenables interrupts, it allows other interrupt requests to be serviced.



13087D-021

Figure 20. Fully Nested (Direct) Mode Interrupt Controller Connections

End-of-Interrupt Command

The end-of-interrupt (EOI) command is used by the programmer to reset the in-service (IS) bit when an interrupt service routine is completed. The EOI command is issued by writing the proper pattern to the EOI register. There are two types of EOI commands, specific and non-specific. The non-specific command does not specify which IS bit is reset. When issued, the interrupt controller automatically resets the IS bit of the highest priority source with an active service routine. A specific EOI command requires that the programmer send the interrupt vector type to the interrupt controller indicating which source's IS bit is to be reset. This command is used when the fully nested structure has been disturbed or the highest priority IS bit that was set does not belong to the service routine in progress.

Trigger Mode

The four external interrupt pins can be programmed in either edge- or level-trigger mode. The control register for each external source has a level-trigger mode (LTM) bit. All interrupt inputs are active High. In the edge sense mode or the level-trigger mode, the interrupt request must remain active (High) until the interrupt request is acknowledged by the 80C188 CPU. In the edge-sense mode, if the level remains high after the interrupt is acknowledged, the input is disabled and no further requests will be generated. The input level must go Low for at least one clock cycle to reenable the input. In the level-trigger mode, no such provision is made; holding the interrupt input High will cause continuous interrupt requests.

Interrupt Vectoring

The 80C186 Interrupt Controller will generate interrupt vectors for the integrated DMA channels and the integrated Timers. In addition, the Interrupt Controller will generate interrupt vectors for the external interrupt lines, if they are not configured in Cascade or Special Fully Nested Mode. The interrupt vectors generated are fixed and cannot be changed (see Table 3).

Interrupt Controller Registers

The Interrupt Controller Register mode is shown in Figure 22. It contains 15 registers. All registers can either be read or written, unless specified otherwise.

In-Service Register

This register can be read from or written into. The format is shown in Figure 23 It contains the in-service bit for each of the interrupt sources. The in-service bit is set to indicate that a source's service routine is in progress. When an in-service bit is set, the interrupt controller will not generate interrupts to the CPU when it receives interrupt requests from devices with a lower programmed priority level. The TMR bit is the in-service bit for all three timers; the D0 and D1 bits are the in-service bits for the two DMA channels; the I3–I0 are the in-service bits for the external interrupt pins. The IS bit is set when the processor acknowledges an interrupt request either by an interrupt acknowledge or by reading the poll register. The IS bit is reset at the end of the interrupt service routine by an end-of-interrupt command.

Interrupt Request Register

The internal interrupt sources have interrupt request bits inside the interrupt controller. The format of this register is shown in Figure 23. A read from this register yields the status of these bits. The TMR bit is the logical OR of all timer interrupt requests. D0 and D1 are the interrupt request bits for the DMA channels.

The state of the external interrupt input pins is also indicated. The state of the external interrupt pins is not a stored condition inside the interrupt controller; therefore, the external interrupt bits cannot be written. The external interrupt request bits are set when an interrupt request is given to the interrupt controller, so if edge-triggered mode is selected, the bit in the register will be High only after an inactive-to-active transition. For internal interrupt sources, the register bits are set when a request arrives and are reset when the processor acknowledges the requests.

Writes to the interrupt request register will affect the D0 and D1 interrupt request bits. Setting either bit will cause the corresponding interrupt request while clearing either bit will remove the corresponding interrupt request. All other bits in the register are read-only.

Mask Register

This is a 16-bit register that contains a mask bit for each interrupt source. The format for this register is shown in Figure 23. A 1 in a bit position corresponding to a particular source serves to mask the source from generating interrupts. These mask bits are the exact same bits which are used in the individual control registers; programming a mask bit using the mask register will also change this bit in the individual control registers, and vice versa.

Priority Mask Register

This register is used to mask all interrupts below a particular interrupt priority level. The format of this register is shown in Figure 24. The code in the lower three bits of this register inhibits interrupts of priority lower (a higher priority number) than the code specified. For example, 100 written into this register masks interrupts of level five (101), six (110), and seven (111). The register is reset to seven (111) upon RESET so no interrupts are masked due to priority number.

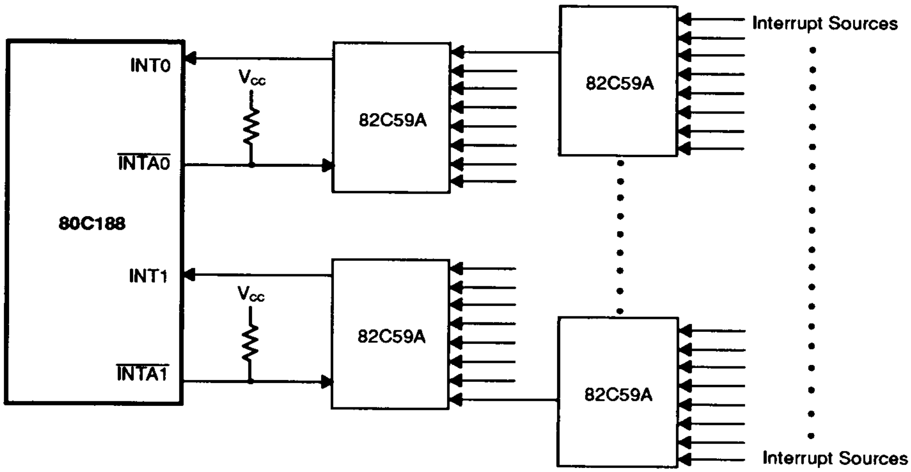


Figure 21. Cascade and Special Fully Nested Mode Interrupt Controller Connections

13087D-022

Interrupt Status Register

This register contains general interrupt controller status information. The format of this register is shown in Figure 25. The bits in the status register have the following functions:

DHLT: DMA Halt Transfer; setting this bit halts all DMA transfers. It is automatically set whenever a non-maskable interrupt occurs, and it is reset when an IRET instruction is executed. This bit allows prompt service of all non-maskable interrupts. This bit may also be set by the programmer.

IRTx: These three bits represent the individual timer interrupt request bits. These bits differentiate between timer interrupts, since the timer IR bit in the interrupt request register is the OR function of all timer interrupt requests. Note that setting any one of these three bits initiates an interrupt request to the interrupt controller.

Timer, DMA 0, 1; Control Registers

These registers are the control words for all the internal interrupt sources. The format for these registers is shown in Figure 26. The three bit positions PR0, PR1, and PR2 represent the programmable priority level of the interrupt source. The MSK bit inhibits interrupt requests from the interrupt source. The MSK bits in the individual control registers are the exact same bits as in the Mask Register; modifying them in the individual control registers will also modify them in the Mask Register, and vice versa.

	Offset
INT3 Control Register	3EH
INT2 Control Register	3CH
INT1 Control Register	3AH
INT0 Control Register	38H
DMA 1 Control Register	36H
DMA 0 Control Register	34H
Timer Control Register	32H
Interrupt Status Register	30H
Interrupt Request Register	2EH
In-Service Register	2CH
Priority Mask Register	2AH
Mask Register	28H
Poll Status Register	26H
Poll Register	24H
EOI Register	22H

13087D-023

Figure 22. Interrupt Controller Registers (Master Mode)

15	14		10	9	8	7	6	5	4	3	2	1	0
0	0	. . .	0	0	0	I3	I2	I1	I0	D1	D0	0	TMR

Figure 23. In-Service, Interrupt Request, and Mask Register Formats

13087D-024

15	14		3	2	1	0
0	0	. . . .	0	PRM2	PRM1	PRM0

Figure 24. Priority Mask Register Format

13087D-025

15	14		7	6	5	4	3	2	1	0
DHLT	0	. . . . . .	0	0	0	0	0	IRT2	IRT1	IRT0

Figure 25. Interrupt Status Register Format (Master Mode)

13087D-026

15	14		4	3	2	1	0
0	0	. . . .	0	MSK	PR2	PR1	PR0

Figure 26. Timer/DMA Control Register Formats

13087D-027

15	14		7	6	5	4	3	2	1	0
0	0	. . . . . .	0	SFNM	C	LTM	MSK	PR2	PR1	PR0

Figure 27. INT0/INT1 Control Register Formats

13087D-028

15	14		5	4	3	2	1	0
0	0	. . . .	0	LTM	MSK	PR2	PR1	PR0

Figure 28. INT2/INT3 Control Register Formats

13087D-029

INT3–INT0 Control Registers

These registers are the control words for the four external input pins. Figure 27 shows the format of the INT0 and INT1 Control registers; Figure 28 shows the format of the INT2 and INT3 Control registers. In cascade mode or special fully nested mode, the control words for INT2 and INT3 are not used.

The bits in the various control registers are encoded as follows:

PR2–0: Priority programming information. Highest priority = 000, lowest priority = 111.

LTM: Level-trigger mode bit. 1 = level-triggered; 0 = edge-triggered. Interrupt input levels are active High. In level-triggered mode, an interrupt is generated whenever the external line is High. In edge-triggered mode, an interrupt will be generated only when this level is preceded by an inactive-to-active transition on the line. In both cases, the level must remain active until the interrupt is acknowledged.

MSK: Mask bit, 1 = mask; 0 = non-mask.

C: Cascade mode bit, 1 = cascade; 0 = direct.

SFNM: Special fully nested mode bit, 1 = SFNM

EOI Register

The end of the interrupt register is a command register which can only be written into. The format of this register is shown in Figure 29. It initiates an EOI command when written to by the 80C188 CPU.

The bits in the EOI register are encoded as follows:

S_x: Encoded information that specifies an interrupt source vector type as shown in Table 3. For example, to reset the In-service bit for DMA channel 0, these bits should be set to 01010, since the vector type for DMA channel 0 is 10.

Note: To reset the single In-Service bit for any of the three timers, the vector type for Timer 0(8) should be written in this register.

NSPEC/SPEC: A bit that determines the type of EOI command. Nonspecific = 1, Specific = 0.

Poll and Poll Status Registers

These registers contain polling information. The format of these registers is shown in Figure 30. They can only be read. Reading the Poll register constitutes a software poll. This will set the IS bit of the highest priority pending interrupt. Reading the poll status register will not set the IS bit of the highest priority pending interrupt; only the status of pending interrupts will be provided.

Encoding of the Poll and Poll Status register bits are as follows:

S_x: Encoded information that indicates the vector type of the highest priority interrupting source. Valid only when INTREQ = 1.

INTREQ: This bit determines if an interrupt request is present. Interrupt Request = 1; no Interrupt Request = 0.

SLAVE MODE OPERATION

When slave mode is used, the internal 80C188 interrupt controller will be used as a slave controller to an external master interrupt controller. The internal 80C188 resources will be monitored by the internal interrupt controller, while the external controller functions as the system master interrupt controller.

Upon reset, the 80C188 will be in the master mode. To provide for slave mode operation, bit 14 of the relocation register should be set (see Figure 7).

Because of pin limitations caused by the need to interface to an external 82C59A master, the internal interrupt controller will no longer accept external inputs. There are, however, enough 80C188 interrupt controller inputs (internally) to dedicate one to each timer. In this mode, each timer interrupt source has its own mask bit, IS bit, and control word.

In slave mode each peripheral must be assigned a unique priority to ensure proper interrupt controller operation. Therefore, it is the programmer's responsibility to assign correct priorities and initialize interrupt control registers before enabling interrupts.

Slave Mode External Interface

The configuration of the 80C188 with respect to an external 82C59A master is shown in Figure 31. The INT0 (Pin 45) input is used as the 80C188 CPU interrupt input. INT3 (Pin 41) functions as an output to send the 80C188 slave-interrupt-request to one of the eight master PIC inputs.

Correct master-slave interface requires decoding of the slave addresses (CAS2–0). Slave 82C59As do this internally. Because of pin limitations, the 80C188 slave address will have to be decoded externally. $\overline{\text{INT1}}$ (Pin 44) is used as a slave-select input. Note that the slave vector address is transferred internally, but the READY input must be supplied externally.

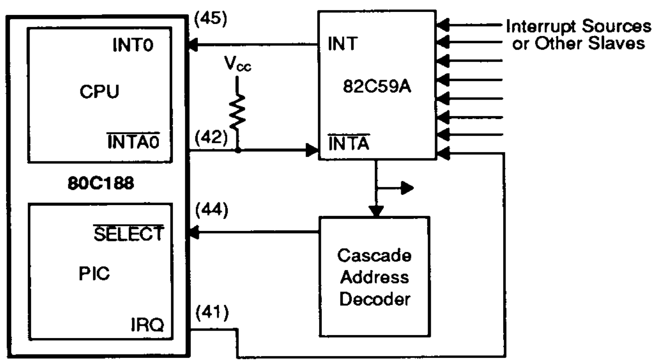
$\overline{\text{INT2}}$ (Pin 42) is used as an acknowledge output, suitable to drive the $\overline{\text{INTA}}$ input of an 82C59A.

15	14	13		5	4	3	2	1	0
SPEC/NSPEC	0	0	. . . .	0	S4	S3	S2	S1	S0

Figure 29. EOI Register Format 13087D-030

15	14	13		5	4	3	2	1	0
INT REQ	0	0	. . . .	0	S4	S3	S2	S1	S0

Figure 30. Poll Register Format 13087D-031



13087D-032

Figure 31. Slave Mode Interrupt Controller Connections

Interrupt Nesting

Slave mode operation allows nesting of interrupt requests. When an interrupt is acknowledged, the priority logic masks off all priority levels except those with equal or higher priority.

Vector Generation in the Slave Mode

Vector generation in slave mode is exactly like that of an 8259A or 82C59A slave. The interrupt controller generates an 8-bit vector type number which the CPU multiplies by four and uses as an address into the vector table. The five most significant bits of this type number are user-programmable while the three least significant bits are defined according to Figure 32. The significant five bits of the vector are programmed by writing to the Interrupt Vector register at offset 20H.

Specific End-of-Interrupt

In slave mode the specific EOI command operates to reset an in-service bit of a specific priority. The user supplies a 3-bit priority-level value that points to an in-service bit to be reset. The command is executed by writing the correct value in the Specific EOI register at offset 22H.

Interrupt Controller Registers in the Slave Mode

All control and command registers are located inside the internal peripheral control block. Figure 32 shows the offsets of these registers.

End-of-Interrupt Register

The end-of-interrupt register is a command register which can only be written. The format of this register is shown in Figure 33. It initiates an EOI command when written by the 80C188 CPU.

The bits in the EOI register are encoded as follows:

L_x: Encoded value indicating the priority of the IS bit to be reset.

In-Service Register

This register can be read from or written into. It contains the in-service bit for each of the interrupt sources. The format for this register is shown in Figure 34. Bit positions 2 and 3 correspond to the DMA channels; positions 0, 4, and 5 correspond to the integral timers. The source's IS bit is set when the processor acknowledges its interrupt request.

Interrupt Request Register

This register indicates which internal peripherals have interrupt requests pending. The format of this register is shown in Figure 34. The Interrupt request bits are set when a request arrives from an internal source, and are reset when the processor acknowledges the request. The interrupt as in master mode, D0 and D1 are read/write; all other bits are read only.

Mask Register

This register contains a mask bit for each interrupt source. The format for this register is shown in Figure 34. If the bit in this register corresponding to a particular interrupt source is set, any interrupts from that source will be masked. These mask bits are exactly the same bits which are used in the individual control registers; that is, changing the state of a mask bit in this register will also change the state of the mask bit in the individual interrupt control register corresponding to the bit.

Control Registers

These registers are the control words for all the internal interrupt sources. The format of these registers is shown in Figure 35. Each of the timers and both of the DMA channels have their own Control Register.

The bits of the Control Registers are encoded as follows:

pr_x: 3-bit encoded field indicating a priority level for the source; note that each source must be programmed at specified levels.

msk: mask bit for the priority level indicated by pr_x bits.

	Offset
Timer 2 Control Register (Vector Type XXXXX101)	3AH
Timer 1 Control Register (Vector Type XXXXX100)	38H
DMA 1 Control Register (Vector Type XXXXX011)	36H
DMA 0 Control Register (Vector Type XXXXX010)	34H
Timer 0 Control Register (Vector Type XXXXX000)	32H
Interrupt Status Register	30H
Interrupt-Request Register	2EH
In-Service Register	2CH
Priority-Level Mask Register	2AH
Mask Register	28H
Specific EOI Register	22H
Interrupt Vector Register	20H

13087D-033

Figure 32. Interrupt Controller Registers (Slave Mode)

Interrupt Vector Register

This register provides the upper five bits of the interrupt vector address. The format of this register is shown in Figure 36. The interrupt controller itself provides the lower three bits of the interrupt vector, as determined by the priority level of the interrupt request.

The format of the bits in this register is:

t_x: 5-bit field indicating the upper five bits of the vector address.

Priority-Level Mask Register

This register indicates the lowest priority-level interrupt which will be serviced.

The encoding of the bits in this register is:

m_x: 3-bit encoded field indication priority-level value. All levels of lower priority will be masked.

15	14	13		8	7	6	5	4	3	2	1	0
0	0	0	. . . . .	0	0	0	0	0	0	L2	L1	L0

13087D-034

Figure 33. Specific EOI Register Format

15	14	13		8	7	6	5	4	3	2	1	0
0	0	0	. . . . .	0	0	0	TMR2	TMR1	D1	D0	0	TMR0

13087D-035

Figure 34. In-Service, Interrupt Request, and Mask Register Format

15	14	13		8	7	6	5	4	3	2	1	0
0	0	0	. . . . .	0	0	0	0	0	MSK	PR2	PR1	PR0

13087D-036

Figure 35. Control Word Format

15	14	13		8	7	6	5	4	3	2	1	0
0	0	0	. . . . .	0	T4	T3	T2	T1	T0	0	0	0

13087D-037

Figure 36. Interrupt Vector Register Format

15	14	13		8	7	6	5	4	3	2	1	0
0	0	0	. . . . .	0	0	0	0	0	0	m2	m1	m0

13087D-038

Figure 37. Priority Level Mask Register

Interrupt Status Register

This register is defined as in master mode except that DHLT is not implemented (see Figure 25).

Interrupt Controller and Reset

Upon RESET, the interrupt controller will perform the following actions:

- All SFNM bits reset to 0, implying Fully Nested Mode.
- All PR bits in the various control registers set to 1. This places all sources at lowest priority (level 111).
- All LTM bits reset to 0, resulting in edge-sense mode.
- All Interrupt Service bits reset to 0.
- All Interrupt Request bits reset to 0.
- All MSK (Interrupt Mask) bits set to 1 (mask).
- All C (Cascade) bits reset to 0 (non-cascade).
- All PRM (Priority Mask) bits set to 1, implying no levels masked.
- Initialized to master mode.

Enhanced Mode Operation

In Compatible Mode, the 80C188 operates with all the features of the NMOS 80188, with the exception of 8087 support (i.e., no numeric coprocessing is possible in Compatible Mode). Queue-Status information is still available for design purposes other than 8087 support.

All the Enhanced Mode features are completely masked when in Compatible Mode. A write to any of the Enhanced Mode registers will have no effect, while a read will not return any valid data.

In Enhanced Mode, the 80C188 will operate with Power-Save and DRAM refresh, in addition to all the Compatible Mode features.

Entering Enhanced Mode

Enhanced Mode can be entered by tying the RESET output signal from the 80C188 to the $\overline{TEST}$ input.

Queue-Status Mode

The queue-status mode is entered by strapping the $\overline{RD}$ pin Low. $\overline{RD}$ is sampled at RESET and if Low, the 80C188 will reconfigure the ALE and $\overline{WR}$ pins to be QS0 and QS1, respectively. This mode is available on the 80C188 in both Compatible and Enhanced Modes.

DRAM Refresh Control Unit Description

The Refresh Control Unit (RCU) automatically generates DRAM refresh bus cycles. The RCU operates only in Enhanced Mode. After a programmable period of time, the RCU generates a memory read request to the BIU. If the address generated during a refresh bus cycle is within the range of a properly programmed chip select, that chip select will be activated when the BIU executes the refresh bus cycle. The ready logic and wait states programmed for that region will also be in force. If no chip select is activated, then external ready is automatically required to terminate the refresh bus cycle.

If the HLDA pin is active when a DRAM refresh request is generated (indicating a bus hold condition), then the 80C188 will deactivate the HLDA pin in order to perform a refresh cycle. The circuit external to the 80C188 must remove the HOLD signal for at least one clock in order to execute the refresh cycle. The sequence of HLDA going inactive while HOLD is being held active can be used to signal a pending refresh request.

All registers controlling DRAM refresh may be read and written in Enhanced Mode. When the processor is operating in Compatible Mode, they are deselected and are therefore inaccessible. Some fields of these registers cannot be written and are always read as 0s.

DRAM Refresh Addresses

The address generated during a refresh cycle is determined by the contents of the MDRAM register (see Figure 38) and the contents of a 9-bit counter. Figure 39 illustrates the origin of each bit.

Refresh Control Unit Programming and Operation

After programming the MDRAM and the CDRAM registers (see Figures 38 and 40), the RCU is enabled by setting the "E" bit in the EDRAM register (Figure 41). The clock counter (T0–T8 of EDRAM) will be loaded from C0–C8 of CDRAM during T3 of instruction cycle that sets the "E" bit. The clock counter is then decremented at each subsequent CLKOUT.

A refresh is requested when the value of the counter has reached 1 and the counter is reloaded from CDRAM. In order to avoid missing refresh requests, the value in the CDRAM register should always be at least 18 (12H). Clearing the "E" bit at anytime will clear the counter and stop refresh requests, but will not reset the refresh address counter.

POWER-SAVE CONTROL

Power-Save Operation

The 80C188, when in Enhanced Mode, can enter a power saving state by internally dividing the processor clock frequency by a programmable factor. This divided frequency is also available at the CLKOUT pin. The PDCON register contains the 3-bit fields for selecting the clock division factor and the enable bit.

All internal logic, including the Refresh Control Unit and the timers, will have their clocks slowed down by the division factor. To maintain a real time count or a fixed DRAM refresh rate, these peripherals must be reprogrammed when entering and leaving the power-save mode.

The power-save mode is exited whenever an interrupt is processed by automatically resetting the enable bit. If the power-save mode is to be re-entered after serving the interrupt, the enable bit will need to be set in software before returning from the interrupt routine.

The internal clocks of the 80C188 will begin to be divided during the T3 state of the instruction cycle that sets the enable bit. Clearing the enable bit will restore full speed in the T3 state of that instruction.

The AMD® 80C188 is a static design and as such has no minimum clock frequency.

MDRAM: Offset E0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M6	M5	M4	M3	M2	M1	M0	0	0	0	0	0	0	0	0	0

Bits 15–9: M6–M0 are address bits A19–A13 of the 20-bit memory refresh address. These bits should correspond to any chip select address to be activated for the DRAM partition. These bits are cleared to 0 at RESET.
Bits 8–0: Reserved, read back as 0.

13087D-039

Figure 38. Memory Partition Register

A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
M6	M5	M4	M3	M2	M1	M0	0	0	0	CA8	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0	1

M6–M0: Bits defined by MDRAM Register.
CA8–CA0: Bits defined by refresh address counter. These bits change according to a linear/feedback shift register; they do not directly follow a binary count.

Figure 39. Addresses Generated by RCU

13087D-040

CDRAM: Offset E2H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	C8	C7	C6	C5	C4	C3	C2	C1	C0

Bits 15–9: Reserved, read back as 0.
Bits 8–0: C8–C0, clock divisor register, holds the number of CLKOUT cycles between each refresh request.

13087D-041

Figure 40. Clock Pre-Scaler Register

EDRAM: Offset E4H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E	0	0	0	0	0	0	T8	T7	T6	T5	T4	T3	T2	T1	T0

Bit 15: Enable RCU, set to 0 on RESET.
Bits 14–9: Reserved, read back as 0.
Bits 8–0: T8–T0 refresh counter outputs. Read only.

Figure 41. Enable RCU Register

13087D-042

PDCON: Offset F0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E	0	0	0	0	0	0	0	0	0	0	0	0	F2	F1	F0

Bit 15: Enable Power-Save Mode. Set to 0 on RESET.
Bits 14–3: Reserved, read back as 0.
Bits 2–0: Clock Divisor Select.

F2	F1	F0	Divider Factor	F2	F1	F0	Divider Factor
0	0	0	Divide by 1	1	0	0	Divide by 32
0	0	1	Divide by 4	1	0	1	Divide by 64
0	1	0	Divide by 8	1	1	0	Divide by 128
0	1	1	Divide by 16	1	1	1	Divide by 256

Figure 42. Power-Save Control Register

13087D-043

ONCE™ Test Mode

To facilitate testing and inspection of devices when fixed into a target system, the 80C188 has a test mode available which allows all pins to be placed in a high-impedance state. ONCE stands for "ON Circuit Emulation." When placed in this mode, the 80C188 will put all pins in the high-impedance state until RESET.

The ONCE mode is selected by typing the $\overline{UCS}$ and the $\overline{LCS}$ Low during RESET. These pins are sampled on the low-to-high transition of the $\overline{RES}$ pin. The $\overline{UCS}$ and $\overline{LCS}$ pins have weak internal pull-up resistors, similar to the $\overline{RD}$ and $\overline{TEST}$ pins, to guarantee normal operation.

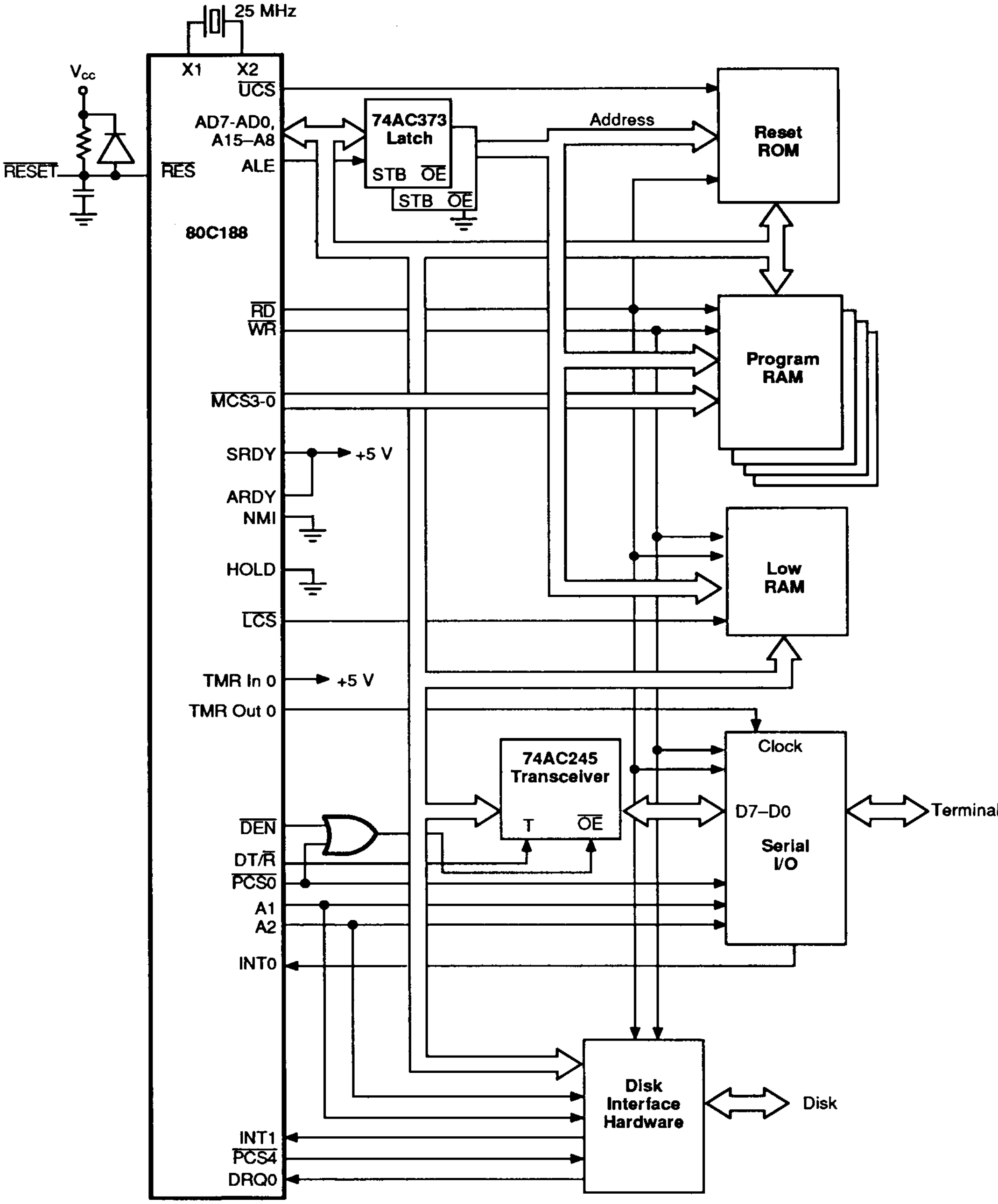


Figure 43. Typical 80C188 System

13087D-044

ABSOLUTE MAXIMUM RATINGS

Ambient temperature under bias (T_A) . . 0°C to +70°C
Storage temperature . . . . –65°C to +150°C
Voltage on any pin with respect to ground . . . . –1.0 V to +7.0 V
Package power dissipation . . . . 1 W

Not to exceed the maximum allowable die temperature based on thermal resistance of the package.

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

DC CHARACTERISTICS over operating ranges

T_A = 0°C to +70°C, V_{CC} = 5 V ±10% except V_{CC} = 5 V ±5% at f > 12.5 MHz

Symbol	Parameter	Preliminary Min	Preliminary Max	Units	Test Conditions
V_{IL}	Input Low Voltage (Except X1)	–0.5	0.2 V_{CC} – 0.3	V	
V_{IL1}	Clock Input Low Voltage (X1)	–0.5	0.6	V	
V_{IH}	Input High Voltage (All except X_1, $\overline{RES}$, ARDY, and SRDY)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage ($\overline{RES}$)	3.0	V_{CC} + 0.5	V	
V_{IH2}	Input High Voltage (SRDY, ARDY)	0.2 V_{CC} + 1.1	V_{CC} + 0.5	V	
V_{IH3}	Clock Input High Voltage (X_1)	3.9	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.5 mA (S0, 1, 2) I_{OL} = 2.0 mA (others)
V_{OH}	Output High Voltage	2.4	V_{CC}	V	I_{OH} = –2.4 mA @ 2.4 V(4)
		V_{CC} – 0.5	V_{CC}	V	I_{OH} = –200 µA @ 0.8 V_{CC}(4)
I_{CC}	Power Supply Current		175	mA	@ 20 MHz, 0°C V_{CC} = 5.25 V(3)
			143	mA	@ 16 MHz, 0°C V_{CC} = 5.25 V(3)
			115	mA	@ 12.5 MHz, 0°C V_{CC} = 5.5 V(3)
			95	mA	@ 10 MHz, 0°C V_{CC} = 5.5 V(3)
			250	µA	@ DC, 0°C V_{CC} = 5.5 V
I_{LI}	Input Leakage Current		±10	µA	@ 0.5 MHz, 0.45 V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current		±10	µA	@ 0.5 MHz, 0.45 V ≤ V_{OUT} ≤ V_{CC}(1)
V_{CLO}	Clock Output Low		0.45	V	I_{CLO} = 4.0 mA
V_{CHO}	Clock Output High	V_{CC} – 0.5		V	I_{CHO} = –500 µA
C_{IN}	Input Capacitance		10	pF	@ 1 MHz(2)
C_{IO}	Output or I/O Capacitance		20	pF	@ 1 MHz(2)

Notes:
1. Pins being floated during HOLD or by invoking the ONCE mode.
2. Characterization conditions are a) Frequency = 1 MHz; b) Unmeasured pins at GND; c) V_{IN} at + 5.0 V or 0.45 V. This parameter is not tested.
3. Current is measured with the device in RESET with X1 and X2 driven and all other non-power pins open.
4. $\overline{RD}/\overline{QSMD}$, $\overline{UCS}$, $\overline{LCS}$, and $\overline{TEST}$ pins have internal pull-up devices. Loading some of these pins above I_{OH} = –200 µA can cause the 80C188 to go into alternative modes of operation (e.g., Queue Status, ONCE) upon reset.

Power Supply Current

Current is linearly proportional to clock frequency and is measured with the device in RESET with X1 and X2 driven and all other non-power pins open.

Maximum current is given by I_{CC} = 8 mA × freq. (MHz) +15 mA.

Typical current is given by I_{CC} (typical) = 5 mA × freq. (MHz) + 0.5 mA. "Typicals" are based on a limited number of samples taken from early manufacturing lots measured at V_{CC} = 5 V and room temperature. "Typicals" are not guaranteed.

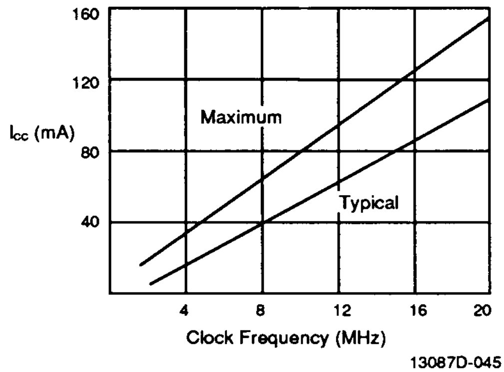


Figure 44. I_{CC} versus Frequency

Parameter Number With Description

Parameter Number	Symbol Name	Parameter Description
1	t_{DVCL}	Data In Setup (A/D)
2	t_{CLDX}	Data in Hold (A/D)
3	t_{CHSV}	Status Active Delay
4	t_{CLSH}	Status Inactive Delay
5	t_{CLAV}	Address Valid Delay
6	t_{CLAX}	Address Hold
7	t_{CLDV}	Data Valid Delay
8	t_{CHDX}	Status Hold Time
9	t_{CHLH}	ALE Active Delay
10	t_{LHLL}	ALE Width
11	t_{CHLL}	ALE Inactive Delay
12	t_{AVLL}	Address Valid to ALE Low
13	t_{LLAX}	Address Hold from ALE Inactive
14	t_{AVCH}	Address Valid to Clock High
15	t_{CLAZ}	Address Float Delay
16	t_{CLCSV}	Chip-Select Active Delay
17	t_{CXCSX}	Chip-Select Hold from Command Inactive
18	t_{CHCSX}	Chip-Select Inactive Delay
19	t_{DXDL}	$\overline{DEN}$ Inactive to DT/$\overline{R}$ Low
20	t_{CVCTV}	Control Active Delay 1
21	t_{CVDEX}	$\overline{DEN}$ Inactive Delay
22	t_{CHCTV}	Control Active Delay 2
23	t_{CLLV}	$\overline{LOCK}$ Valid/Invalid Delay
24	t_{AZRL}	Address Float to $\overline{RD}$ Active
25	t_{CLRL}	$\overline{RD}$ Active Delay
26	t_{RLRH}	$\overline{RD}$ Pulse Width
27	t_{CLRH}	$\overline{RD}$ Inactive Delay
28	t_{RHLH}	$\overline{RD}$ Inactive to ALE High

Parameter Number With Description (continued)

Parameter Number	Symbol Name	Parameter Description
29	t_{RHAV}	$\overline{RD}$ Inactive to Address Active
30	t_{CLDOX}	Data Hold Time
31	t_{CVCTX}	Control Inactive Delay
32	t_{WLWH}	$\overline{WR}$ Pulse Width
33	t_{WHLH}	$\overline{WR}$ Inactive to ALE High
34	t_{WHDX}	Data Hold after $\overline{WR}$
35	t_{WHDEX}	$\overline{WR}$ Inactive to $\overline{DEN}$ Inactive
36	t_{CKIN}	CLKIN Period
37	t_{CLCK}	CLKIN Low Time
38	t_{CHCK}	CLKIN High Time
39	t_{CKHL}	CLKIN Fall Time
40	t_{CKLH}	CLKIN Rise Time
41	t_{CICO}	CLKIN to CLKOUT Skew
42	t_{CLCL}	CLKOUT Period
43	t_{CLCH}	CLKOUT Low Time
44	t_{CHCL}	CLKOUT High Time
45	t_{CH1CH2}	CLKOUT Rise Time
46	t_{CL2CL1}	CLKOUT Fall Time
47	t_{SRYCL}	Synchronous Ready (SRDY) Transition Setup Time
48	t_{CLSRY}	SRDY Transition Hold Time
49	t_{ARYCH}	ARDY Resolution Transition Setup Time
50	t_{CLARX}	ARDY Active Hold Time
51	t_{ARYCHL}	ARDY Inactive Holding Time
52	t_{ARYLCL}	Asynchronous Ready (ARDY) Setup Time
53	t_{INVCH}	INTx, NMI, $\overline{TEST}$, TMR IN Setup Time
54	t_{INVCL}	DRQ0, DRQ1, Setup Time
55	t_{CLTMV}	Timer Output Delay
56	t_{CHQSV}	Queue Status Delay
57	t_{RESIN}	$\overline{RES}$ Setup
58	t_{HVCL}	HOLD Setup
59	t_{CLAZ}	Address Float Delay
60	t_{CLAV}	Address Valid Delay
61	t_{CLRO}	Reset Delay
62	t_{CLHAV}	HLDA Valid Delay
63	t_{CHCZ}	Command Lines Float Delay
64	t_{CHCV}	Command Lines Valid Delay (after Float)

SWITCHING CHARACTERISTICS

Major Cycle Timings (Read Cycle)

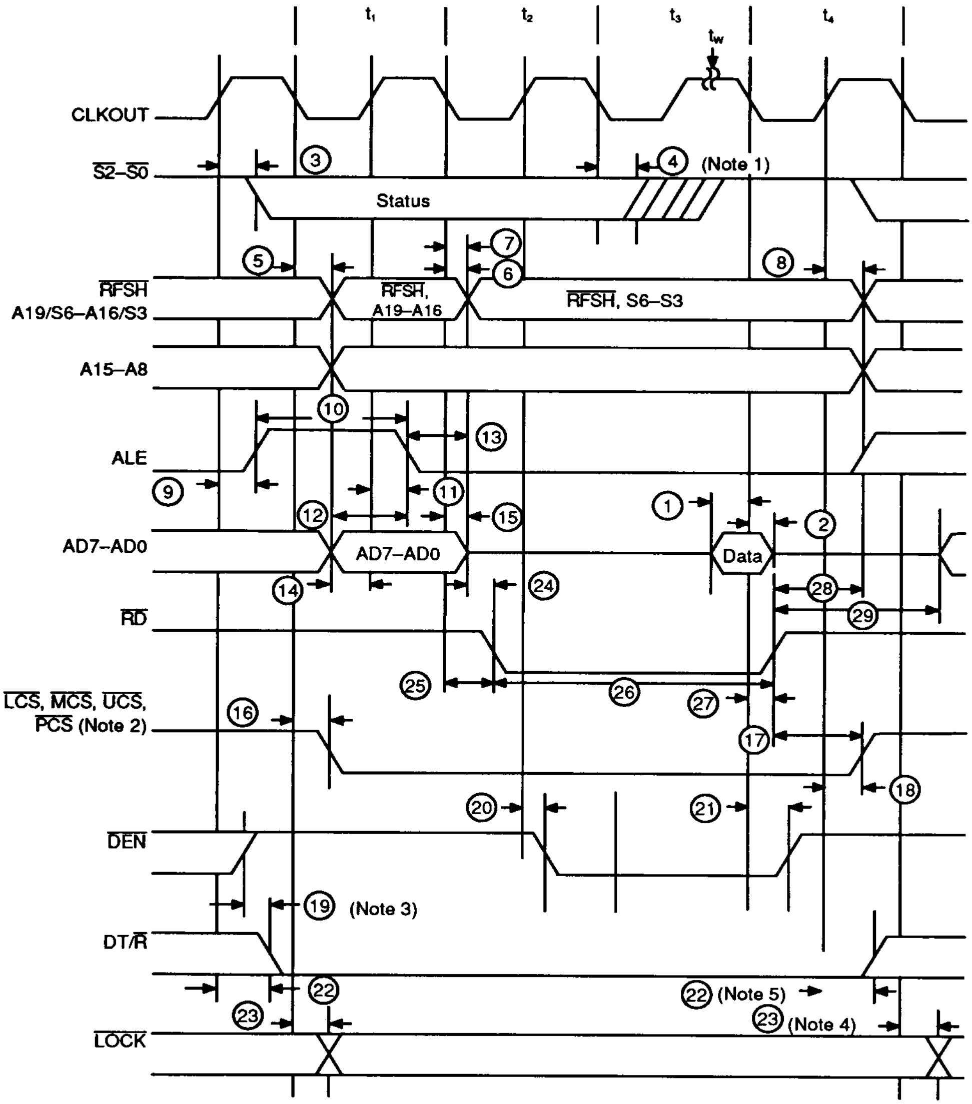
T_A = 0°C to +70°C, V_{CC} = 5 V ±10% except V_{CC} = 5 V ±5% at f > 12.5 MHz

No	Symbol	Parameter	Preliminary 80C188 Min	80C188 Max	80C188-12 Min	80C188-12 Max	80C188-16 Min	80C188-16 Max	80C188-20 Min	80C188-20 Max	Unit
80C188 General Timing Requirements (Listed More Than Once)											
1	t_{DVCL}	Data in Setup (A/D)	15		15		15		10		ns
2	t_{CLDX}	Data in Hold (A/D)	3		3		3		3		ns
80C188 General Timing Responses (Listed More Than Once)											
3	t_{CHSV}	Status Active Delay	5	45	5	35	5	31	5	25	ns
4	t_{CLSH}	Status Inactive Delay	5	46	5	35	5	30	5	25	ns
5	t_{CLAV}	Address Valid Delay	5	44	5	36	5	33	5	30	ns
6	t_{CLAX}	Address Hold	0		0		0		0		ns
7	t_{CLDV}	Data Valid Delay	5	40	5	36	5	33	5	30	ns
8	t_{CHDX}	Status Hold Time	10		10		10		10		ns
9	t_{CHLH}	ALE Active Delay		30		25		20		20	ns
10	t_{LHLL}	ALE Width	$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-15$		ns
11	t_{CHLL}	ALE Inactive Delay		30		25		20		20	ns
12	t_{AVLL}	Address Valid to ALE Low*	$t_{CLCH}-18$		$t_{CLCH}-15$		$t_{CLCH}-15$		$t_{CLCH}-5$		ns
13	t_{LLAX}	Address Hold from ALE Inactive*	$t_{CHCL}-15$		$t_{CHCL}-15$		$t_{CHCL}-15$		$t_{CHCL}-10$		ns
14	t_{AVCH}	Address Valid to Clock High	0		0		0		0		ns
15	t_{CLAZ}	Address Float Delay	t_{CLAX}	30	t_{CLAX}	25	t_{CLAX}	20	t_{CLAX}	17	ns
16	t_{CLCSV}	Chip-Select Active Delay	3	42	3	33	3	30	3	25	ns
17	t_{CXCSX}	Chip-Select Hold from Command Inactive*	$t_{CLCH}-10$		$t_{CLCH}-10$		$t_{CLCH}-10$		$t_{CLCH}-10$		ns
18	t_{CHCSX}	Chip-Select Inactive Delay	5	35	5	30	5	25	5	20	ns
19	t_{DXDL}	$\overline{DEN}$ Inactive to DT/$\overline{R}$ Low*	0		0		0		0		ns
20	t_{CVCTV}	Control Active Delay 1	3	44	3	37	3	31	3	27	ns
21	t_{CVDEX}	$\overline{DEN}$ Inactive Delay	5	44	5	37	5	31	5	30	ns
22	t_{CHCTV}	Control Active Delay 2	5	44	5	37	5	31	5	27	ns
23	t_{CLLV}	$\overline{LOCK}$ Valid/Invalid Delay	3	40	3	37	3	35	3	30	ns
80C188 Timing Responses (Read Cycle)											
24	t_{AZRL}	Address Float to $\overline{RD}$ Active	0		0		0		0		ns
25	t_{CLRL}	$\overline{RD}$ Active Delay	5	44	5	37	5	31	5	27	ns
26	t_{RLRH}	$\overline{RD}$ Pulse Width	$2t_{CLCL}-30$		$2t_{CLCL}-25$		$2t_{CLCL}-25$		$2t_{CLCL}-25$		ns
27	t_{CLRH}	$\overline{RD}$ Inactive Delay	5	44	5	37	5	31	5	27	ns
28	t_{RHLH}	$\overline{RD}$ Inactive to ALE High*	$t_{CLCH}-14$		$t_{CLCH}-14$		$t_{CLCH}-14$		$t_{CLCH} = 14$		ns
29	t_{RHAV}	$\overline{RD}$ Inactive to Address Active*	$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-15$		ns

*Equal Loading

All timings are measured at 1.5 V and 100 pF loading on CLKOUT unless otherwise noted.
All output test conditions are with $C_L = 50–200$ pF (10 MHz) and $C_L = 50–100$ pF (12.5–20 MHz).
For AC tests, input $V_{IL} = 0.45$ V and $V_{IH} = 2.4$ V except at X_1 where $V_{IH} = V_{CC} - 0.5$ V.

Read Cycle Waveforms



Notes: 1. Status inactive in state preceding t_4.
2. If latched A_1 and A_2 are selected instead of PCS5 and PCS6.
3. For write cycle followed by read cycle.
4. t_1 of next bus cycle.
5. Changes in t-state preceding next bus cycle if followed by write.

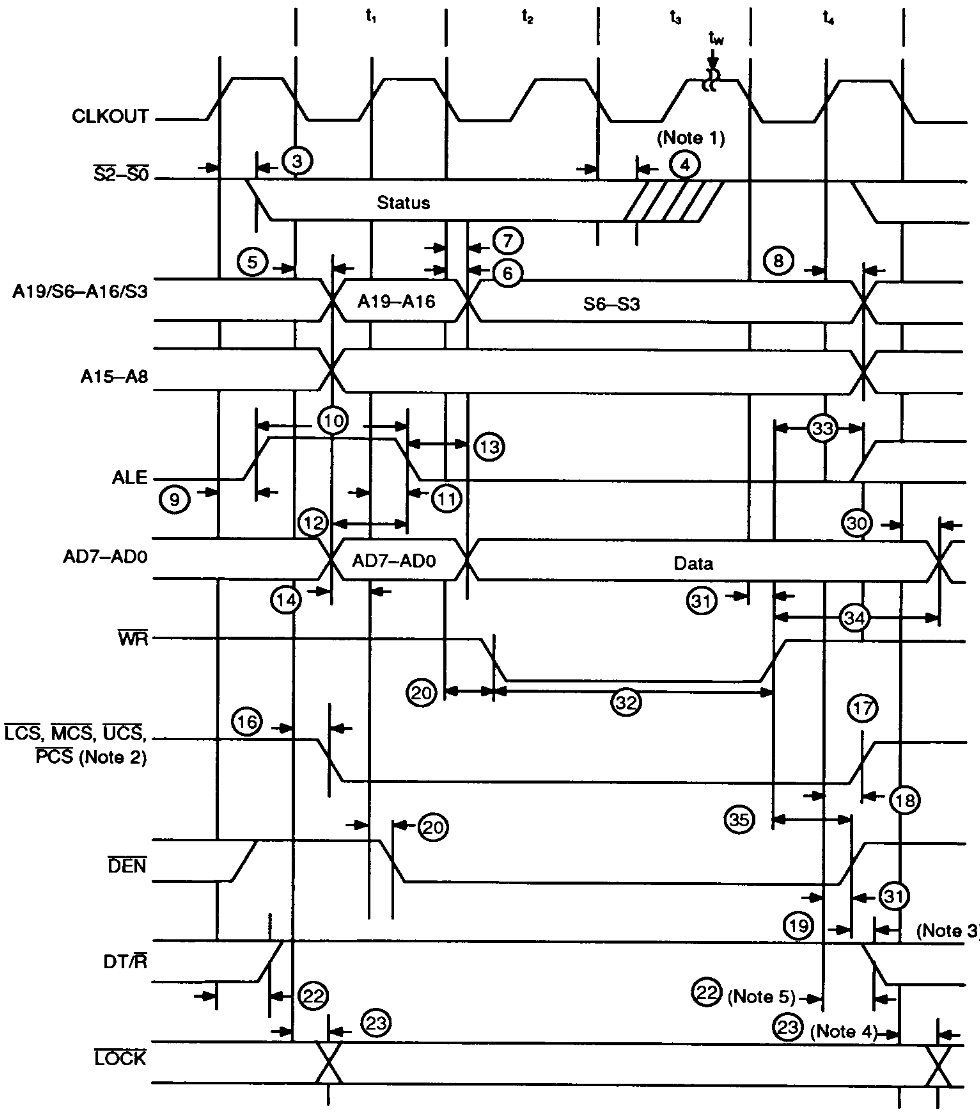
SWITCHING CHARACTERISTICS (continued)
Major Cycle Timings (Write cycle)

$T_A = 0°C$ to $+70°C$, $V_{CC} = 5$ V ±10% except $V_{CC} = 5$ V ±5% at f > 12.5 MHz

No	Symbol	Parameter	Preliminary 80C188 Min	80C188 Max	80C188-12 Min	80C188-12 Max	80C188-16 Min	80C188-16 Max	80C188-20 Min	80C188-20 Max	Unit
80C188 General Timing Responses (Listed More Than Once)											
3	t_{CHSV}	Status Active Delay	5	45	5	35	5	31	5	25	ns
4	t_{CLSH}	Status Inactive Delay	5	46	5	35	5	30	5	25	ns
5	t_{CLAV}	Address Valid Delay	5	44	5	36	5	33	5	30	ns
6	t_{CLAX}	Address Hold	0		0		0		0		ns
7	t_{CLDV}	Data Valid Delay	5	40	5	36	5	33	5	30	ns
8	t_{CHDX}	Status Hold Time	10		10		10		10		ns
9	t_{CHLH}	ALE Active Delay		30		25		20		20	ns
10	t_{LHLL}	ALE Width	$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-15$		ns
11	t_{CHLL}	ALE Inactive Delay		30		25		20		20	ns
12	t_{AVLL}	Address Valid to ALE Low*	$t_{CLCH}-18$		$t_{CLCH}-15$		$t_{CLCH}-15$		$t_{CLCH}-13$		ns
13	t_{LLAX}	Address Hold from ALE Inactive*	$t_{CHCL}-15$		$t_{CHCL}-15$		$t_{CHCL}-15$		$t_{CHCL}-10$		ns
14	t_{AVCH}	Address Valid to Clock High	0		0		0		0		ns
16	t_{CLCSV}	Chip-Select Active Delay	3	42	3	33	3	30	3	25	ns
17	t_{CXCSX}	Chip-Select Hold from Command Inactive*	$t_{CLCH}-10$		$t_{CLCH}-10$		$t_{CLCH}-10$		$t_{CLCH}-10$		ns
18	t_{CHCSX}	Chip-Select Inactive Delay	5	35	5	30	5	25	5	20	ns
19	t_{DXDL}	$\overline{DEN}$ Inactive to DT/$\overline{R}$ Low*	0		0		0		0		ns
20	t_{CVCTV}	Control Active Delay 1	3	44	3	37	3	31	3	27	ns
23	t_{CLLV}	$\overline{LOCK}$ Valid/Invalid Delay	3	40	3	37	3	35	3	30	ns
30	t_{CLDOX}	Data Hold Time	3		3		3		3		ns
31	t_{CVCTX}	Control Inactive Delay	3	44	3	37	3	31	3	27	ns
80C188 Timing Responses (Write Cycle)											
32	t_{WLWH}	$\overline{WR}$ Pulse Width	$2t_{CLCL}-30$		$2t_{CLCL}-25$		$2t_{CLCL}-25$		$2t_{CLCL}-20$		ns
33	t_{WHLH}	$\overline{WR}$ Inactive to ALE High*	$t_{CLCH}-14$		$t_{CLCH}-14$		$t_{CLCH}-14$		$t_{CLCH}-14$		ns
34	t_{WHDX}	Data Hold after $\overline{WR}$*	$t_{CLCL}-34$		$t_{CLCL}-20$		$t_{CLCL}-20$		$t_{CLCL}-17$		ns
35	t_{WHDEX}	$\overline{WR}$ Inactive to $\overline{DEN}$ Inactive*	$t_{CLCH}-10$		$t_{CLCH}-10$		$t_{CLCH}-10$		$t_{CLCH}-10$		ns

All timings are measured at 1.5 V and 100 pF loading on CLKOUT unless otherwise noted.
All output test conditions are with $C_L = 50–200$ pF (10 MHz) and $C_L = 50–100$ pF (12.5–20 MHz).
For AC tests, input $V_{IL} = 0.45$ V and $V_{IH} = 2.4$ V except at X_1 where $V_{IH} = V_{CC} - 0.5$ V.
*Equal Loading

Write Cycle Waveforms



Notes:
1. Status inactive in state preceding t_4.
2. If latched A_1 and A_2 are selected instead of $\overline{PCS5}$ and $\overline{PCS6}$,.
3. For write cycle followed by read cycle.
4. t_1 of next bus cycle.
5. Changes in t-state preceding next bus cycle if followed by read, INTA, or halt.

SWITCHING CHARACTERISTICS (continued)

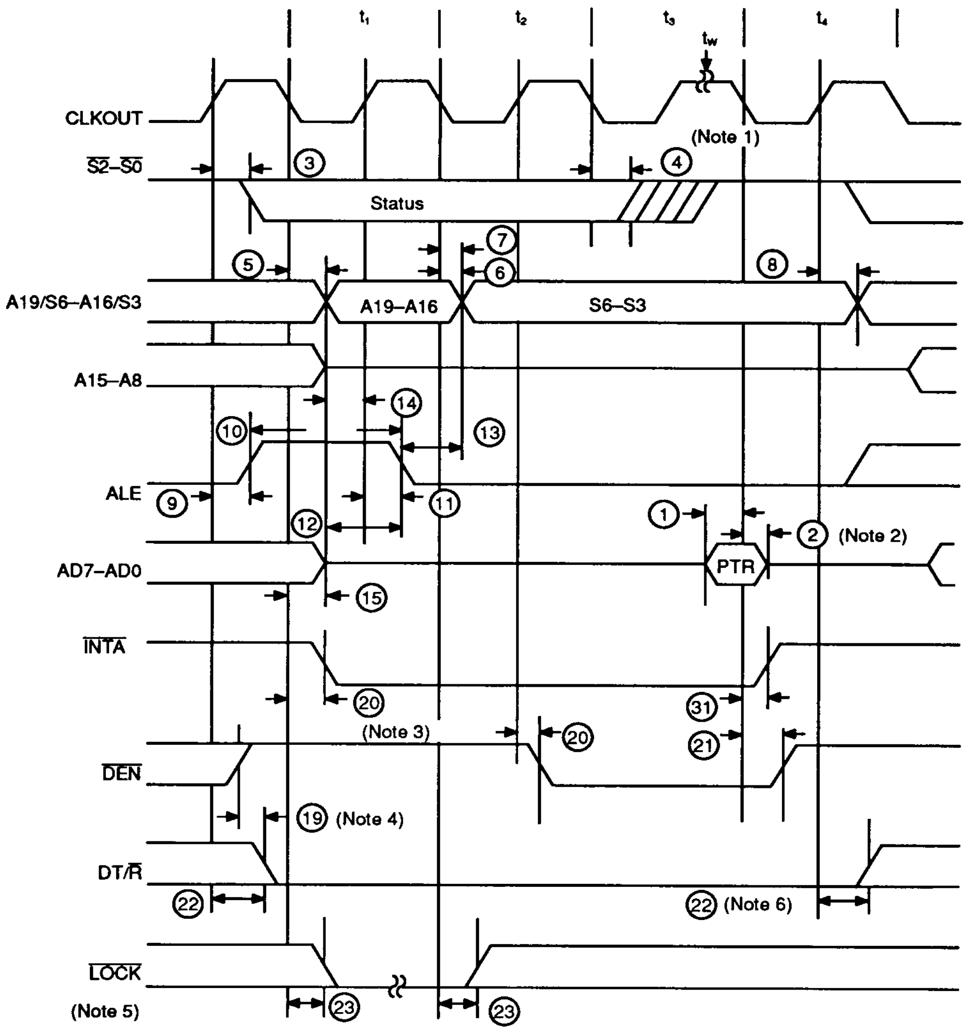
Major Cycle Timings (Interrupt Acknowledge Cycle)

T_A = 0°C to +70°C, V_{CC} = 5 V ±10% except V_{CC} = 5 V ±5% at f > 12.5 MHz

			Preliminary								
			80C188		80C188-12		80C188-16		80C188-20		
No	Symbol	Parameter	Min	Max	Min	Max	Min	Max	Min	Max	Unit
80C188 General Timing Requirements (Listed More Than Once)											
1	t_{DVCL}	Data in Setup (A/D)	15		15		15		10		ns
2	t_{CLDX}	Data in Hold (A/D)	3		3		3		3		ns
80C188 General Timing Responses (Listed More Than Once)											
3	t_{CHSV}	Status Active Delay	5	45	5	35	5	31	5	25	ns
4	t_{CLSH}	Status Inactive Delay	5	46	5	35	5	30	5	25	ns
5	t_{CLAV}	Address Valid Delay	5	44	5	36	5	33	5	30	ns
6	t_{CLAX}	Address Hold	0		0		0		0		ns
7	t_{CLDV}	Data Valid Delay	5	40	5	36	5	33	5	30	ns
8	t_{CHDX}	Status Hold Time	10		10		10		10		ns
9	t_{CHLH}	ALE Active Delay		30		25		20		20	ns
10	t_{LHLL}	ALE Width	$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-13$		ns
11	t_{CHLL}	ALE Inactive Delay		30		25		20		20	ns
12	t_{AVLL}	Address Valid to ALE Low*	$t_{CLCH}-18$		$t_{CLCH}-15$		$t_{CLCH}-15$		$t_{CLCH}-13$		ns
13	t_{LLAX}	Address Hold to ALE Inactive*	$t_{CHCL}-15$		$t_{CHCL}-15$		$t_{CHCL}-15$		$t_{CHCL}-10$		ns
14	t_{AVCH}	Address Valid to Clock High	0		0		0		0		ns
15	t_{CLAZ}	Address Float Delay	t_{CLAX}	30	t_{CLAX}	25	t_{CLAX}	20	t_{CLAX}	17	ns
19	t_{DXDL}	$\overline{DEN}$ Inactive to DT/$\overline{R}$ Low*	0		0		0		0		ns
20	t_{CVCTV}	Control Active Delay 1	3	44	3	37	3	31	3	27	ns
21	t_{CVDEX}	$\overline{DEN}$ Inactive Delay (Non-Write Cycles)	5	44	5	37	5	31	5	30	ns
22	t_{CHCTV}	Control Active Delay 2	5	44	5	37	5	31	5	27	ns
23	t_{CLLV}	$\overline{LOCK}$ Valid/Invalid Delay	3	40	3	37	3	35	3	30	ns
31	t_{CVCTX}	Control Inactive Delay	3	44	3	37	3	31	3	27	ns

All timings are measured at 1.5 V and 100 pF loading on CLKOUT unless otherwise noted.
All output test conditions are with C_L = 50–200 pF (10 MHz) and C_L = 50–100 pF (12.5–20 MHz).
For AC tests, input V_{IL} = 0.45 V and V_{IH} = 2.4 V except at X_1 where $V_{IH} = V_{CC} - 0.5$ V.
*Equal Loading

Interrupt Acknowledge Cycle Waveforms



Notes: 1. Status inactive in state preceding t_4.

2. The data hold time lasts only until INTA goes inactive, even if the INTA transition occurs prior to t_{CLDX} (min).

3. INTA occurs one clock later in Slave Mode.

4. For write cycle followed by interrupt acknowledge cycle.

5. LOCK is active upon t_1 of the first interrupt acknowledge cycle and inactive upon t_2 of the second interrupt acknowledge cycle.

6. Changes in t-state preceding next bus cycle if followed by write.

SWITCHING CHARACTERISTICS (continued)

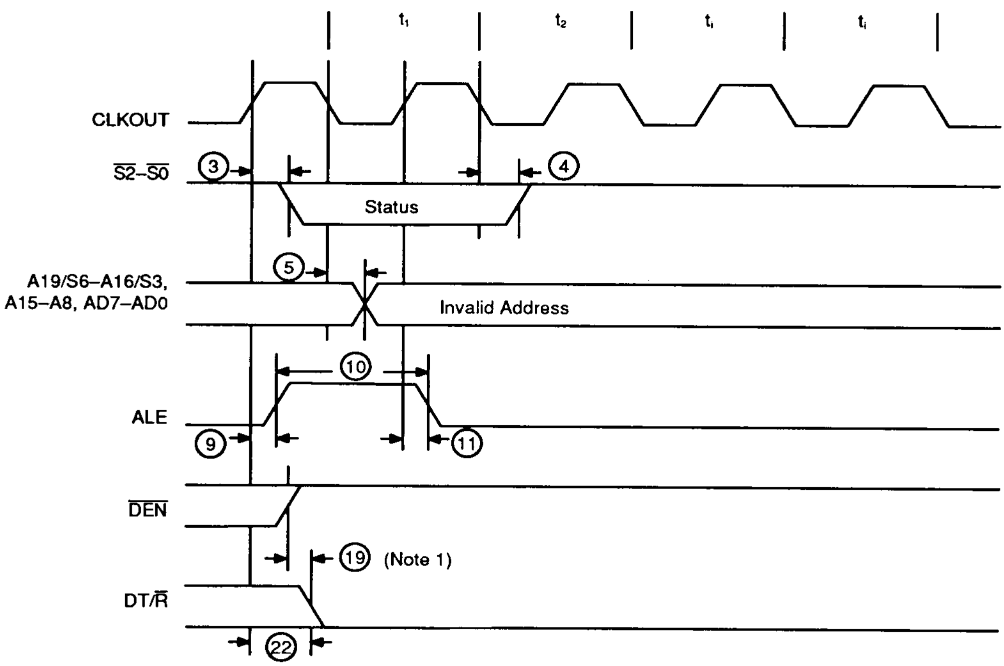
Software Halt Cycle Timings

T_A = 0°C to +70°C, V_{CC} = 5 V ±10% except V_{CC} = 5 V ±5% at f > 12.5 MHz

No	Symbol	Parameter	Preliminary								Unit
			80C188		80C188-12		80C188-16		80C188-20		
			Min	Max	Min	Max	Min	Max	Min	Max	
80C188 General Timing Responses (Listed More Than Once)											
3	t_{CHSV}	Status Active Delay	5	45	5	35	5	31	5	25	ns
4	t_{CLSH}	Status Inactive Delay	5	46	5	35	5	30	5	25	ns
5	t_{CLAV}	Address Valid Delay	5	44	5	36	5	33	5	30	ns
9	t_{CHLH}	ALE Active Delay		30		25		20		20	ns
10	t_{LHLL}	ALE Width	$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-15$		$t_{CLCL}-15$		ns
11	t_{CHLL}	ALE Inactive Delay		30		25		20		20	ns
19	t_{DXDL}	$\overline{DEN}$ Inactive to DT/$\overline{R}$ Low*		0		0		0	0	0	ns
22	t_{CHCTV}	Control Active Delay 2	5	44	5	37	5	31	5	27	ns

All timings are measured at 1.5 V and 100 pF loading on CLKOUT unless otherwise noted.
All output test conditions are with C_L = 50–200 pF (10 MHz) and C_L = 50–100 pF (12.5–20 MHz).
For AC tests, input V_{IL} = 0.45 V and V_{IH} = 2.4 V except at X_1 where V_{IH} = V_{CC} − 0.5 V.
*Equal Loading

Software Halt Cycle Waveforms



Note: 1. For write cycle followed by halt cycle.

SWITCHING CHARACTERISTICS (continued)

Clock Timings

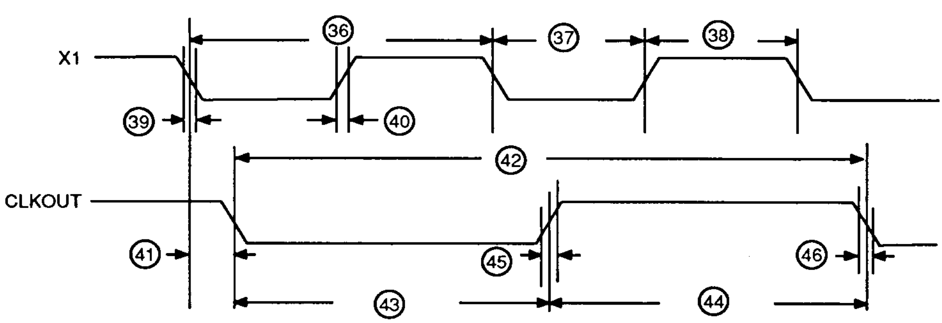
T_A = 0°C to +70°C, V_{CC} = 5 V ±10% except V_{CC} = 5 V ±5% at f > 12.5 MHz

No	Symbol	Parameter	Preliminary								Unit
			80C188		80C188-12		80C188-16		80C188-20		
			Min	Max	Min	Max	Min	Max	Min	Max	
80C188 CLKIN Requirements Measurements taken with following conditions. External clock input to X1 and X2 not connected (float)											
36	t_{CKIN}	CLKIN Period	50		40		31.25		25		ns
37	t_{CLCK}	CLKIN Low Time 1.5 V(2)	20		16		13		7		ns
38	t_{CHCK}	CLKIN High Time 1.5 V(2)	20		16		13		8		ns
39	t_{CKHL}	CLKIN Fall Time 3.5 to 1.0 V		5		5		5		5	ns
40	t_{CKLH}	CLKIN Rise Time 1.0 to 3.5 V		5		5		5		5	ns
80C188 CLKOUT Timing											
41	t_{CICO}	CLKIN to CLKOUT Skew		25		21		17		13	ns
42	t_{CLCL}	CLKOUT Period	100		80		62.5		50		ns
43	t_{CLCH}	CLKOUT Low Time C_L = 100 pF(2)	0.5 t_{CLCL}−8		0.5 t_{CLCL}−7		0.5 t_{CLCL}−7		0.5 t_{CLCL}−7		ns
		C_L = 50 pF(3)	0.5 t_{CLCL}−6		0.5 t_{CLCL}−5		0.5 t_{CLCL}−5		0.5 t_{CLCL}−5		ns
44	t_{CHCL}	CLKOUT High Time C_L = 100 pF(4)	0.5 t_{CLCL}−8		0.5 t_{CLCL}−7		0.5 t_{CLCL}−7		0.5 t_{CLCL}−7		ns
		C_L = 50 pF(3)	0.5 t_{CLCL}−6		0.5 t_{CLCL}−5		0.5 t_{CLCL}−5		0.5 t_{CLCL}−5		ns
45	t_{CH1CH2}	CLKOUT Rise Time 1.0 to 3.5 V		10		10		10		10	ns
46	t_{CL2CL1}	CLKOUT Fall Time 3.5 to 1.0 V		10		10		10		10	ns

All timings are measured at 1.5 V and 100 pF loading on CLKOUT unless otherwise noted.
All output test conditions are with C_L = 50–200 pF (10 MHz) and C_L = 50–100 pF (12.5–20 MHz).
For AC tests, input V_{IL} = 0.45 V and V_{IH} = 2.4 V except at X_1 where V_{IH} = V_{CC} − 0.5 V.

Notes:
1. t_{CLCK} and t_{CHCK} (CLKIN Low and High times) should not have a duration less than 40% of t_{CKIN}.
2. Tested under worst case conditions: V_{CC} = 5.5 V (5.25 V @ 20 MHz), T_A = 70°C.
3. Not tested.
4. Tested under worst case conditions: V_{CC} = 4.5 V (4.75 V @ 20 MHz), T_A = 0°C.

Clock Waveforms



SWITCHING CHARACTERISTICS (continued)

Ready, Peripheral, and Queue Status Timings

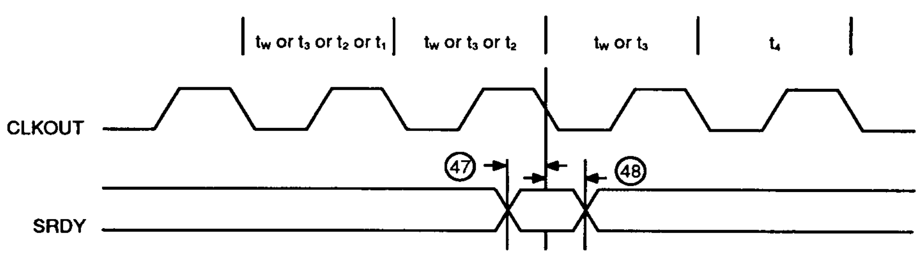
$T_A = 0°C$ to $+70°C$, $V_{CC} = 5$ V ±10% except $V_{CC} = 5$ V ±5% at f > 12.5 MHz

			Preliminary								
			80C188		80C188-12		80C188-16		80C188-20		
No	Symbol	Parameter	Min	Max	Min	Max	Min	Max	Min	Max	Unit
80C188 Ready and Peripheral Timing Requirements											
47	t_{SRYCL}	Synchronous Ready (SRDY) Transition Setup Time(1)	15		15		15		10		ns
48	t_{CLSRY}	SRDY Transition Hold Time(1)	15		15		15		10		ns
49	t_{ARYCH}	ARDY Resolution Transition Setup Time(2)	15		15		15		10		ns
50	t_{CLARX}	ARDY Active Hold Time(1)	15		15		15		10		ns
51	t_{ARYCHL}	ARDY Inactive Holding Time	15		15		15		10		ns
52	t_{ARYLCL}	Asynchronous Ready (ARDY) Setup Time(1)	25		25		25		20		ns
53	t_{INVCH}	INTx, NMI, $\overline{TEST}$, TMR IN Setup Time(2)	15		15		15		15		ns
54	t_{INVCL}	DRQ0, DRQ1, Setup Time(2)	15		15		15		15		ns
80C188 Peripheral and Queue Status Timing Responses											
55	t_{CLTMV}	Timer Output Delay		40		33		27		25	ns
56	t_{CHQSV}	Queue Status Delay		37		32		30		23	ns

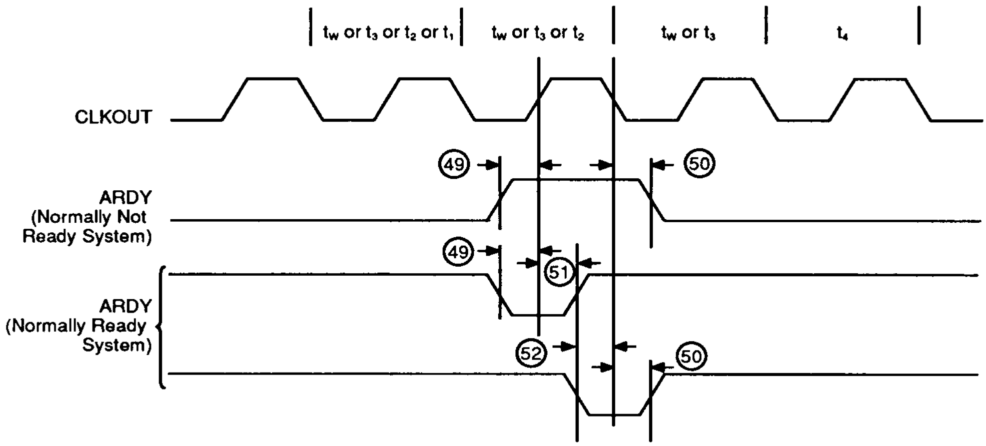
All timings are measured at 1.5 V and 100 pF loading on CLKOUT unless otherwise noted.
All output test conditions are with $C_L = 50–200$ pF (10 MHz) and $C_L = 50–100$ pF (12.5–20 MHz).
For AC tests, input $V_{IL} = 0.45$ V and $V_{IH} = 2.4$ V except at X_1 where $V_{IH} = V_{CC} - 0.5$ V.

Notes: 1. To guarantee proper operation.
2. To guarantee recognition at clock edge.

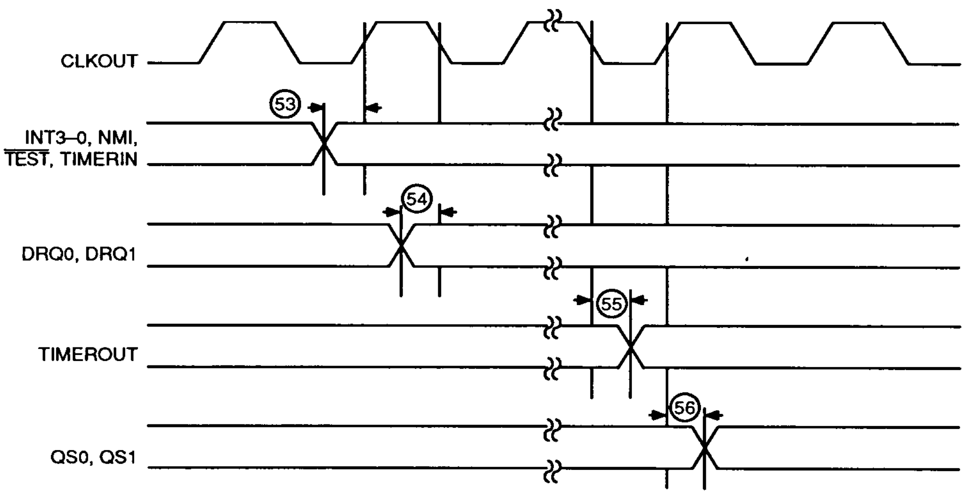
Synchronous Ready (SRDY) Waveforms



Asynchronous Ready (ARDY) Waveforms



Peripheral and Queue Status Waveforms



SWITCHING CHARACTERISTICS (continued)
RESET and HOLD/HLDA Timings

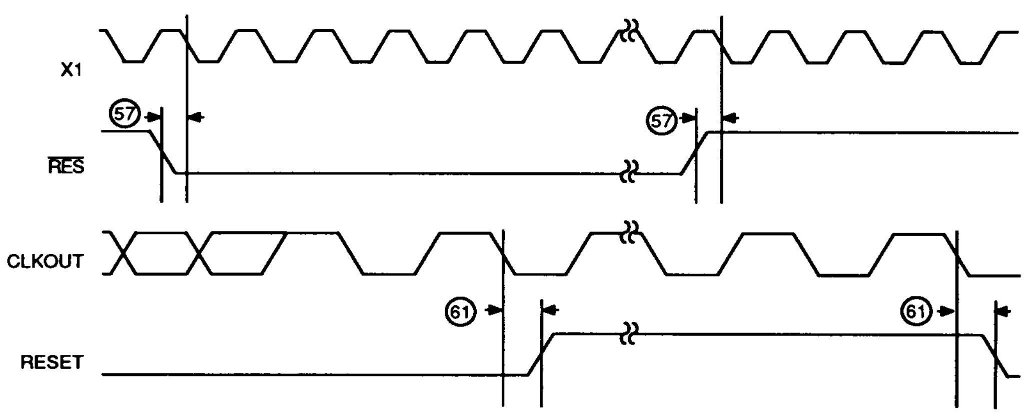
T_A = 0°C to +70°C, V_{CC} = 5 V ±10% except V_{CC} = 5 V ±5% at f > 12.5 MHz

			Preliminary								
			80C188		80C188-12		80C188-16		80C188-20		
No	Symbol	Parameter	Min	Max	Min	Max	Min	Max	Min	Max	Unit
80C188 RESET and HOLD/HLDA Timing Requirements											
57	t_{RESIN}	$\overline{RES}$ Setup	15		15		15		15		ns
58	t_{HVCL}	HOLD Setup(1)	15		15		15		15		ns
80C188 General Timing Responses (Listed More Than Once)											
59	t_{CLAZ}	Address Float Delay	t_{CLAX}	30	t_{CLAX}	25	t_{CLAX}	20	t_{CLAX}	17	ns
60	t_{CLAV}	Address Valid Delay	5	44	5	36	5	33	5	30	ns
80C188 RESET and HOLD/HLDA Timing Responses											
61	t_{CLRO}	Reset Delay		40		33		27		25	ns
62	t_{CLHAV}	HLDA Valid Delay	3	40	3	33	3	25	3	25	ns
63	t_{CHCZ}	Command Lines Float Delay		40		33		28		25	ns
64	t_{CHCV}	Command Lines Valid Delay (after Float)		44		36		32		30	ns

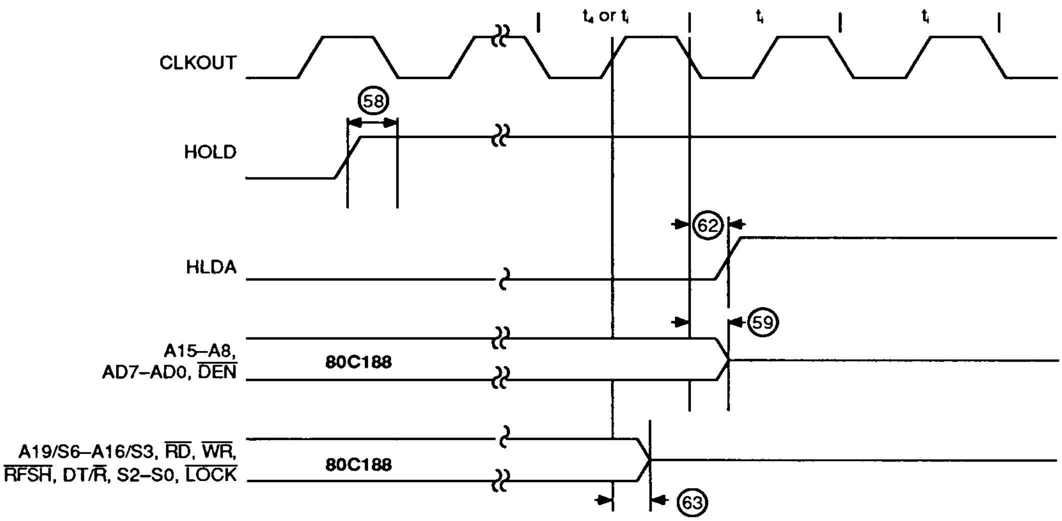
All timings are measured at 1.5 V and 100 pF loading on CLKOUT unless otherwise noted.
All output test conditions are with C_L = 50–200 pF (10 MHz) and C_L = 50–100 pF (12.5–20 MHz).
For AC tests, input V_{IL} = 0.45 V and V_{IH} = 2.4 V except at X_1 where V_{IH} = V_{CC} − 0.5 V.

Note: 1. To guarantee recognition at next clock.

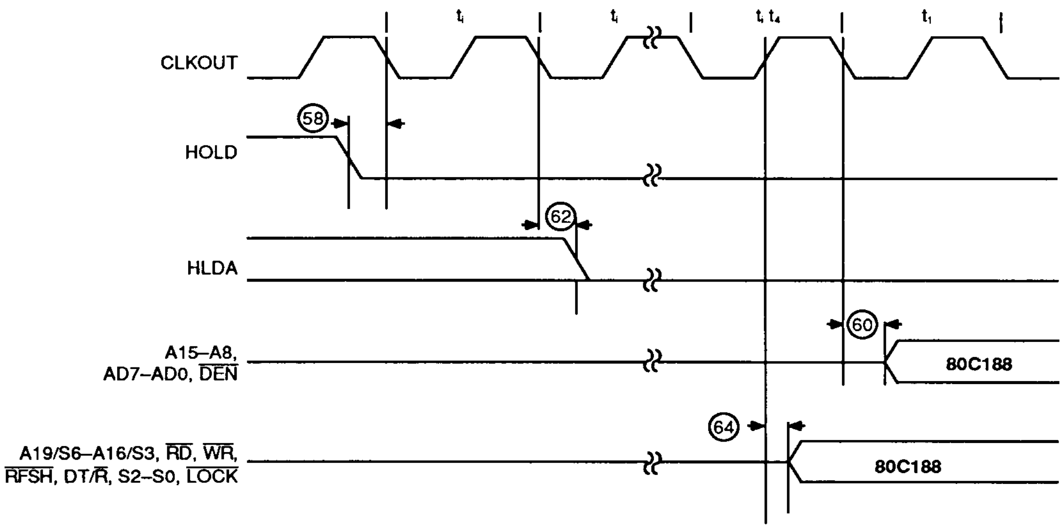
RESET Waveforms



HOLD/HLDA Waveforms (Entering HOLD)



HOLD/HLDA Waveforms (Leaving HOLD)



EXPLANATION OF THE SWITCHING SYMBOLS

Each timing symbol has from 5 to 7 characters. The first character is always a "t" (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A:	Address
ARY:	Asynchronous Ready Input
C:	Clock Output
CK:	Clock Input
CS:	Chip Select
CT:	Control (DT/$\overline{R}$, $\overline{DEN}$, . . .)
D:	Data Input
DE:	$\overline{DEN}$
H:	Logic Level High
IN:	Input (DRQ0, TIM0, . . .)
L:	Logic Level Low or ALE
O:	Output
QS:	Queue Status (QS1, QS2)
R:	$\overline{RD}$ Signal, RESET Signal
S:	Status ($\overline{S2}$, $\overline{S1}$, $\overline{S0}$)
SRY:	Synchronous Ready Input
V:	Valid
W:	$\overline{WR}$ Signal
X:	No Longer a Valid Logic Level
Z:	Float

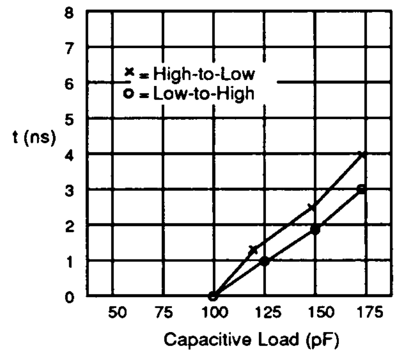
Examples:

t_{CLAV}—Time from Clock Low to Address Valid

t_{CHLH}—Time from Clock High to ALE High

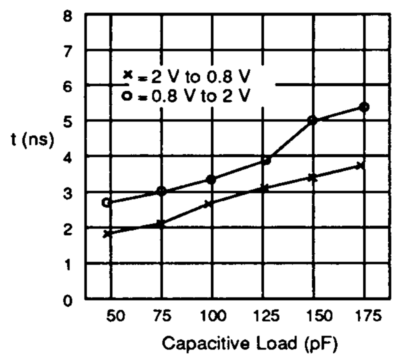
t_{CLCSV}—Time from Clock Low to Chip Select Valid

Waveforms (continued)



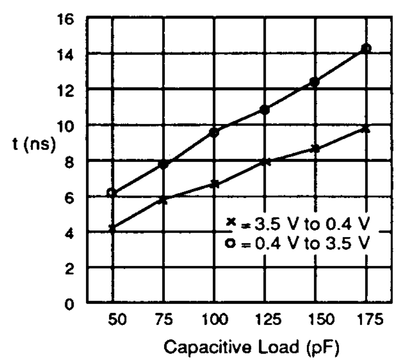
13087D-046

Figure 45. Capacitive Derating Curve for Typical Output Delay



13087D-047

Figure 46. TTL Voltage Level Rise and Fall Times for Output Buffers



13087D-048

Figure 47. CMOS Voltage Level Rise and Fall Times for Output Buffers

80C188 EXECUTION TIMINGS

A determination of 80C188 program execution timing must consider bus cycles necessary to prefetch instructions, as well as the number of execution unit cycles necessary to execute instructions. The following instruction timings represent the minimum execution time in clock cycles for each instruction. The timings given are based on the following assumptions:

- The opcode, along with any data or displacement required for execution of a particular instruction, has been prefetched and resides in the queue at the time it is needed.
- No wait states or bus HOLDs occur.

All jumps and calls include the time required to fetch the opcode of the next instruction at the destination address.

All instructions which involve memory access can require one or two additional clocks above the minimum timings shown due to the asynchronous handshake between the bus interface unit (BIU) and execution unit.

The 80C188 8-bit BIU is noticeably limited in its performance relative to the execution unit. A sufficient number of prefetched bytes may not reside in the prefetch queue much of the time. Therefore, actual program execution time will be substantially greater than that derived from adding the instruction timings shown.

INSTRUCTION SET SUMMARY

Function	Format				Clock Cycles	Comment
DATA TRANSFER						
MOV = Move:						
Register to register/memory	1 0 0 0 1 0 0 w	mod reg r/m			2/12*	
Register/memory to register	1 0 0 0 1 0 1 w	mod reg r/m			2/9*	
Immediate to register/memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1	12–13	8/16-bit
Immediate to register	1 0 1 1 w reg	data	data if w = 1		3-4	8/16-bit
Memory to accumulator	1 0 1 0 0 0 0 w	addr-low	addr-high		8*	
Accumulator to memory	1 0 1 0 0 0 1 w	addr-low	addr-high		9*	
Register/memory to segment register	1 0 0 0 1 1 1 0	mod 0 reg r/m			2/13	
Segment register to register/memory	1 0 0 0 1 1 0 0	mod 0 reg r/m			2/15	
PUSH = Push:						
Memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m			20	
Register	0 1 0 1 0 reg				14	
Segment register	0 0 0 reg 1 1 0				13	
Immediate	0 1 1 0 1 0 s 1	data	data if s = 0		14	
PUSHA = Push All	0 1 1 0 0 0 0 0				68	
POP = Pop:						
Memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m			24	
Register	0 1 0 1 1 reg				14	
Segment register	0 0 0 reg 1 1 1	(reg ≠ 01)			12	
POPA = Pop All	0 1 1 0 0 0 0 1				83	
XCHG = Exchange:						
Register/memory with register	1 0 0 0 0 1 1 w	mod reg r/m			4/17*	
Register with accumulator	1 0 0 1 0 reg				3	
IN = Input from:						
Fixed port	1 1 1 0 0 1 0 w	port			10*	
Variable port	1 1 1 0 1 1 0 w				8*	
OUT = Output to:						
Fixed port	1 1 1 0 0 1 1 w	port			9*	
Variable port	1 1 1 0 1 1 1 w				7*	
XLAT = Translate byte to AL	1 1 0 1 0 1 1 1				15	
LEA = Load EA to register	1 0 0 0 1 1 0 1	mod reg r/m			6	
LDS = Load pointer to DS	1 1 0 0 0 1 0 1	mod reg r/m	(mod ≠ 11)		26	
LES = Load pointer to ES	1 1 0 0 0 1 0 0	mod reg r/m	(mod ≠ 11)		26	
LAHF = Load AH with flags	1 0 0 1 1 1 1 1				2	
SAHF = Store AH into flags	1 0 0 1 1 1 1 0				3	
PUSHF = Push flags	1 0 0 1 1 1 0 0				13	
POPF = Pop flags	1 0 0 1 1 1 0 1				12	

Shaded areas indicate instructions not available in 8086 or 8088 microsystems.
*Note: Clock cycles shown for byte transfer. For word operations, add 4-clock cycles for all memory transfers.

INSTRUCTION SET SUMMARY (continued)

Function	Format				Clock Cycles	Comment
DATA TRANSFER (Continued)						
SEGMENT = Segment Override:						
CS	0 0 1 0 1 1 1 0				2	
SS	0 0 1 1 0 1 1 0				2	
DS	0 0 1 1 1 1 1 0				2	
ES	0 0 1 0 0 1 1 0				2	
ARITHMETIC:						
ADD = Add:						
Reg/memory with register to either	0 0 0 0 0 0 d w	mod reg r/m			3/10*	
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s w = 01	4/16*	
Immediate to accumulator	0 0 0 0 0 1 0 w	data	data if w = 1		3/4	8/16-bit
ADC = Add with carry:						
Reg/memory with register to either	0 0 0 1 0 0 d w	mod reg r/m			3/10*	
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s w = 01	4/16*	
Immediate to accumulator	0 0 0 1 0 1 0 w	data	data if w = 1		3/4	8/16-bit
INC = Increment:						
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m			3/15*	
Register	0 1 0 0 0 reg				3	
SUB = Subtract:						
Reg/memory and register to either	0 0 1 0 1 0 d w	mod reg r/m			3/10*	
Immediate from register/memory	1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s w = 1	4/16*	
Immediate from accumulator	0 0 0 1 0 1 1 0 w	data	data if w = 1		3/4	8/16-bit
SBB = Subtract with borrow:						
Reg/memory and register to either	0 0 0 1 1 0 d w	mod reg r/m			3/10*	
Immediate from register/memory	1 0 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s w = 01	4/16*	
Immediate from accumulator	0 0 0 1 1 1 0 w	data	data if w = 1		3/4	8/16-bit
DEC = Decrement:						
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m			3/15*	
Register	0 1 0 0 1 reg				3	
CMP = Compare:						
Register/memory with register	0 0 1 1 1 0 1 w	mod reg r/m			3/10*	
Register with register/memory	0 0 1 1 1 0 0 w	mod reg r/m			3/10*	
Immediate with register/memory	1 0 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s w = 01	3/10*	
Immediate with accumulator	0 0 1 1 1 1 0 w	data	data if w = 1		3/4	8/16–bit
NEG = Change sign register/memory	1 1 1 1 0 1 1 w	mod 0 1 1 r/m			3/10*	
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1				8	
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1				4	
AAS = ASCII adjust for subtract	0 0 1 1 1 1 1 1				7	
DAS = Decimal adjust for subtract	0 0 1 0 1 1 1 1				4	
MUL = Multiply (unsigned)	1 1 1 1 0 1 1 w	mod 1 0 0 r/m				
Register-Byte					26–28	
Register-Word					35–37	
Memory-Byte					32–34	
Memory-Word					41–43*	

Shaded areas indicate instructions not available in 8086 or 8088 microsystems.
*Note: Clock cycles shown for byte transfer. For word operations, add 4-clock cycles for all memory transfers.

INSTRUCTION SET SUMMARY (continued)

Function	Format				Clock Cycles	Comment
ARITHMETIC (Continued)						
IMUL = Integer multiply (signed)	1 1 1 1 0 1 1 w	mod 1 0 1 r/m				
Register-Byte Register-Word Memory-Byte Memory-Word					25–28 34–37 31–34 40–43*	
IMUL = Integer Immediate multiply (signed)	0 1 1 0 1 0 s 1	mod reg r/m	data	data if s = 0	22–25/ 29–32	
DIV = Divide (unsigned):	1 1 1 1 0 1 1 w	mod 1 1 0 r/m				
Register-Byte Register-Word Memory-Byte Memory-Word					29 38 35 44*	
IDIV = Integer divide (signed)	1 1 1 1 0 1 1 w	mod 1 1 1 r/m				
Register-Byte Register-Word Memory-Byte Memory-Word					44–52 53–61 50–58 59–67*	
AAM = ASCII adjust for multiply	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0			19	
AAD = ASCII adjust for divide	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0			15	
CBW = Convert byte to word	1 0 0 1 1 0 0 0				2	
CWD = Convert word to double word	1 0 0 1 1 0 0 1				4	
LOGIC **Shift/Rotate Instructions:**						
Register/Memory by 1	1 1 0 1 0 0 0 w	mod TTT r/m			2/15	
Register/Memory by CL	1 1 0 1 0 0 1 w	mod TTT r/m			5 + n/17 + n	
Register/Memory by Count	1 1 0 0 0 0 0 w	mod TTT r/m	count		5 + n/17 + n	

TTT	Instruction
0 0 0	ROL
0 0 1	ROR
0 1 0	RCL
0 1 1	RCR
1 0 0	SHL/SAL
1 0 1	SHR
1 1 1	SAR

Function	Format				Clock Cycles	Comment
AND = And:						
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m			3/10*	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1	4/16*	
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1		3/4	8/16-bit
TEST = And function to flags, no result:						
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m			3/10*	
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1	4/10*	
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1		3/4	8/16-bit
OR = Or:						
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m			3/10*	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1	4/16*	
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1		3/4	8/16-bit
XOR = Exclusive or:						
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m			3/10*	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1	4/16*	
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1		3/4	8/16-bit
NOT = Invert register/memory:	1 1 1 1 0 1 1 w	mod 0 1 0 r/m			3/10*	

Shaded areas indicate instructions not available in 8086 or 8088 microsystems.
*Note: Clock cycles shown for byte transfer. For word operations, add 4-clock cycles for all memory transfers.

INSTRUCTION SET SUMMARY (continued)

Function	Format			Clock Cycles	Comment
STRING MANIPULATION:					
MOVS = Move byte/word	1 0 1 0 0 1 0 w			14*	
CMPS = Compare byte/word	1 0 1 0 0 1 1 w			22*	
SCAS = Scan byte/word	1 0 1 0 1 1 1 w			15*	
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w			12*	
STOS = Store byte/wd from AL/A	1 0 1 0 1 0 1 w			10*	
INS = Input byte/wd from DX port	0 1 1 0 1 1 0 w			14	
OUTS = Output byte/wd to DX port	0 1 1 0 1 1 1 w			14	
Repeated by count in CX (REP/REPE/REPZ/PEPNE/PEPNZ)					
MOVS = Move string	1 1 1 1 0 0 1 0	1 0 1 0 0 1 0 w		8 + 8n*	
CMPS = Compare string	1 1 1 1 0 0 1 z	1 0 1 0 0 1 1 w		5 + 22n*	
SCAS = Scan string	1 1 1 1 0 0 1 z	1 0 1 0 1 1 1 w		5 + 15n*	
LODS = Load string	1 1 1 1 0 0 1 0	1 0 1 0 1 1 0 w		6 + 11n*	
STOS = Store string	1 1 1 1 0 0 1 0	1 0 1 0 1 0 1 w		6 + 9n*	
INS = Input string	1 1 1 1 0 0 1 0	0 1 1 0 1 1 0 w		8 + 8n*	
OUTS = Output string	1 1 1 1 0 0 1 0	0 1 1 0 1 1 1 w		8 + 8n*	
CONTROL TRANSFER					
CALL = **Call:**					
Direct within segment	1 1 1 0 1 0 0 0	disp-low	disp-high	19	
Register memory indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0 r/m		17/27	
Direct intersegment	1 0 0 1 1 0 1 0	segment offset		31	
		segment selector			
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m	(mod ≠ 11)	54	
JMP = **Unconditional jump:**					
Short/long	1 1 1 0 1 0 1 1	disp-low		14	
Direct within segment	1 1 1 0 1 0 0 1	disp-low	disp-high	14	
Register/mem indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0 r/m		11/21	
Direct intersegment	1 1 1 0 1 0 1 0	segment offset		14	
		segment selector			
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1 r/m	(mod ≠ 11)	34	
RET = **Return from CALL:**					
Within segment	1 1 0 0 0 0 1 1			20	
Within seg adding immed to SP	1 1 0 0 0 0 1 0	data-low	data-high	22	
Intersegment	1 1 0 0 1 0 1 1			30	
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data-low	data-high	33	

Shaded areas indicates instructions not available in 8086 or 8088 microsystems.
*Note: Clock cycles shown for byte transfer. For word operations, add 4-clock cycles for all memory transfers.

INSTRUCTION SET SUMMARY (continued)

Function	Format				Clock Cycles	Comment
CONTROL TRANSFER (Continued):						
JE/JZ = Jump on equal zero	01110100	disp			4/13	JMP not
JL/JNGE = Jump on less/ not greater or equal	01111100	disp			4/13	taken/JMP
JLE/JNG = Jump on less/ or equal not greater	01111110	disp			4/13	taken
JB/JNAE = Jump on below/ not above or equal	01110010	disp			4/13	
JBE/JNA = Jump on below or equal/not above	01110110	disp			4/13	
JP/JPE = Jump on parity/ parity even	01111010	disp			4/13	
JO = Jump on overflow	01110000	disp			4/13	
JS = Jump on sign	01111000	disp			4/13	
JNE/JNZ = Jump on not equal/ not zero	01110101	disp			4/13	
JNL/JGE = Jump on not less greater or equal	01111101	disp			4/13	
JNLE/JG = Jump on not less/ or equal/greater	01111111	disp			4/13	
JNB/JAE = Jump on not below above or equal	01110011	disp			4/13	
JNBE/JA = Jump on not below or equal/above	01110111	disp			4/13	
JNP/JPO = Jump on not par/par odd	01111011	disp			4/13	
JNO = Jump on not overflow	01110001	disp			4/13	
JNS = Jump on not sign	01111001	disp			4/13	
JCXZ = Jump on CX zero	11100011	disp			5/15	
LOOP = Loop CX Times	11100010	disp			6/16	LOOP not
LOOPZ/LOOPE = Loop while zero/equal	11100001	disp			6/16	taken/LOOP
LOOPNZ/LOOPNE = Loop while not zero/equal	11100000	disp			6/16	taken
ENTER = Enter Procedure	11001000	data-low	data-high	L		
L = 0					19	
L = 1					29	
L > 1					26 + 20(n − 1)	
LEAVE = Leave Procedure	11001001				8	
INT = **Interrupt:**						
Type specified	11001101	type			47	
Type 3	11001100				45	if INT. taken/
INTO = Interrupt on overflow	11001110				48/4	if INT. not taken
IRET = Interrupt return	11001111				28	
BOUND = Detect value out of range	01100010	mod reg r/m			33–35	

Shaded areas indicates instructions not available in 8086 or 8088 microsystems.

INSTRUCTION SET SUMMARY (continued)

Function	Format	Clock Cycles	Comment
PROCESSOR CONTROL			
CLC = Clear carry	11111000	2	
CMC = Complement carry	11110101	2	
STC = Set carry	11111001	2	
CLD = Clear direction	11111100	2	
STD = Set direction	11111101	2	
CLI = Clear interrupt	11111010	2	
STI = Set interrupt	11111011	2	
HLT = Halt	11110100	2	
WAIT = Wait	10011011	6	if $\overline{TEST}$ = 0
LOCK = Bus lock prefix	11110000	2	
NOP = No Operation	10010000	3	

Footnotes

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

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low
if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP
DISP follows 2nd byte of instruction (before data if required)
*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

EA calculation time is 4-clock cycles for all modes, and is included in the execution times given whenever appropriate.

Segment Override Prefix

0	0	1	reg	1	1	0

reg is assigned according to the following:

reg	Segment Register
00	ES
01	CS
10	SS
11	DS

REG is assigned according to the following table:

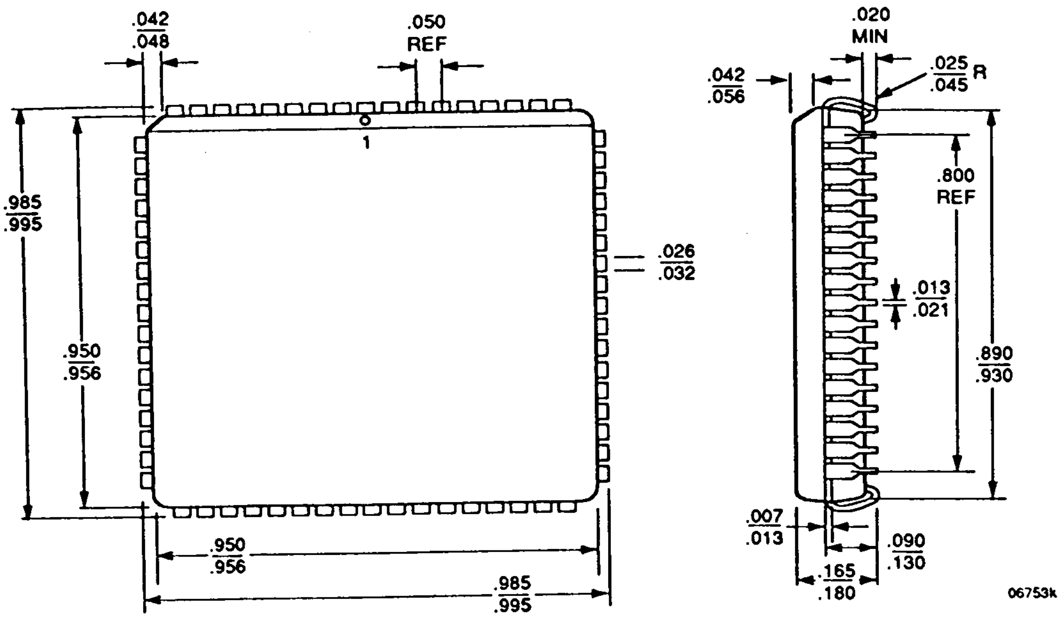
16-Bit (w = 1)	8-Bit (w = 0)
000 AX	000 AL
001 CX	001 CL
010 DX	010 DL
011 BX	011 BL
100 SP	100 AH
101 BP	101 CH
110 SI	110 DH
111 DI	111 BH

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

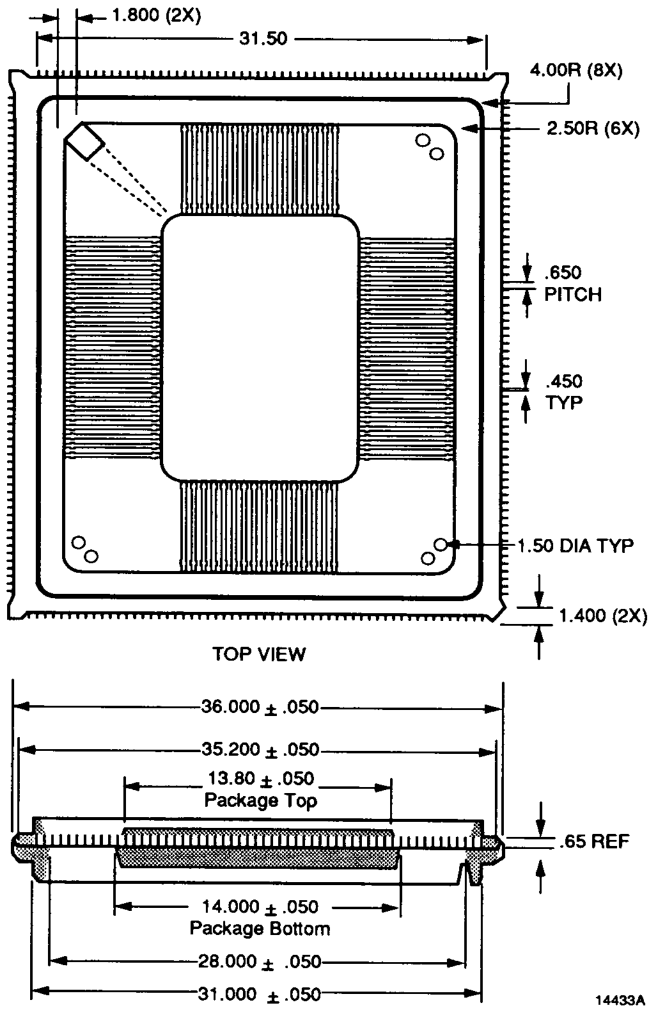
PHYSICAL DIMENSIONS

For reference only. All dimensions measured in inches. BSC is an ANSI standard for Basic Space Centering. Preliminary; package in development.

PL 068

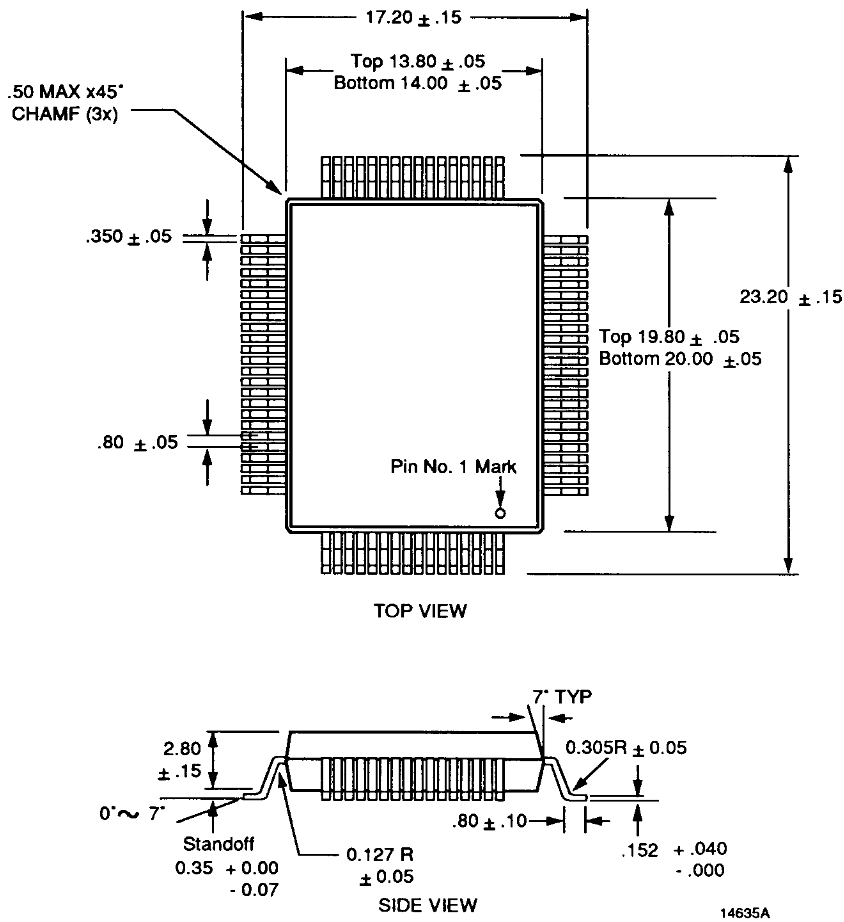


QFP080(MCQFP)
MOLDED CARRIER QUAD FLAT PACK



SIDE VIEW

QFP080 (PQFP)
(EIAJ)



AMD is a registered trademark of Advanced Micro Devices, Inc.
ONCE is a trademark of Intel Corporation.
TapePak is a registered trademark of National Semiconductor Corporation.

Sales Offices

North American

ALABAMA	(205) 882-9122
ARIZONA	(602) 242-4400
CALIFORNIA,	
Culver City	(213) 645-1524
Newport Beach	(714) 752-6262
Sacramento(Roseville)	(916) 786-6700
San Diego	(619) 560-7030
San Jose	(408) 452-0500
Woodland Hills	(818) 992-4155
CANADA, Ontario,	
Kanata	(613) 592-0060
Willowdale	(416) 224-5193
COLORADO	(303) 741-2900
CONNECTICUT	(203) 264-7800
FLORIDA,	
Clearwater	(813) 530-9971
Ft. Lauderdale	(305) 776-2001
Orlando (Longwood)	(407) 862-9292
GEORGIA	(404) 449-7920
ILLINOIS,	
Chicago (Itasca)	(708) 773-4422
Naperville	(708) 505-9517
KANSAS	(913) 451-3115
MARYLAND	(301) 381-3790
MASSACHUSETTS	(617) 273-3970
MINNESOTA	(612) 938-0001
NEW JERSEY,	
Cherry Hill	(609) 662-2900
Parsippany	(201) 299-0002
NEW YORK,	
Liverpool	(315) 457-5400
Brewster	(914)279-8323
Rochester	(716) 272-9020
NORTH CAROLINA	
Harrisburg	(704) 455-1010
Raleigh	(919) 878-8111
OHIO,	
Columbus (Westerville)	(614) 891-6455
Dayton	(513) 439-0268
OREGON	(503) 245-0080
PENNSYLVANIA	(215) 398-8006
SOUTH CAROLINA	(803) 772-6760
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 785-9001
UTAH	(801) 264-2900

International

BELGIUM, Bruxelles	TEL	(02) 771-91-42
	FAX	(02) 762-37-12
	TLX	846-61028
FRANCE, Paris	TEL	(1) 49-75-10-10
	FAX	(1) 49-75-10-13
	TLX	263282F
WEST GERMANY,		
Hannover area	TEL	(0511) 736085
	FAX	(0511) 721254
	TLX	922850
München	TEL	(089) 4114-0
	FAX	(089) 406490
	TLX	523883
Stuttgart	TEL	(0711) 62 33 77
	FAX	(0711) 625187
	TLX	721882
HONG KONG,	TEL	852-8654525
Wanchai	FAX	852-8654335
	TLX	67955AMDAPHX
ITALY, Milan	TEL	(02) 3390541
		(02) 3533241
	FAX	(02) 3498000
	TLX	843-315286
JAPAN,		
Atsugi	TEL	462-29-8460
	FAX	462-29-8458
Kanagawa	TEL	462-47-2911
	FAX	462-47-1729
Tokyo	TEL	(03) 346-7550

International (Continued)

	FAX	(03) 342-5196
	TLX	J24064AMDTKOJ
Osaka	TEL	06-243-3250
	FAX	06-243-3253
KOREA, Seoul	TEL	822-784-0030
	FAX	822-784-8014
LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
	TLX	5109554261 AMDFTL
NORWAY, Hovik	TEL	(03) 010156
	FAX	(02) 591959
	TLX	79079
SINGAPORE	TEL	65-3481188
	FAX	65-3480161
	TLX	55650 AMDMMI
SWEDEN,		
Stockholm	TEL	(08) 733 03 50
(Sundbyberg)	FAX	(08) 733 22 85
	TLX	11602
TAIWAN	TEL	886-2-7213393
	FAX	886-2-7723422
	TLX	886-2-7122066
UNITED KINGDOM,		
Manchester area	TEL	(0925) 828008
(Warrington)	FAX	(0925) 827693
	TLX	851-628524
London area	TEL	(0483) 740440
(Woking)	FAX	(0483) 756196
	TLX	851-859103

North American Representatives

CANADA	
Burnaby, B.C. - DAVETEK MARKETING	(604) 430-3680
Calgary, Alberta - DAVETEK MARKETING	(403) 291-4984
Kanata, Ontario - VITEL ELECTRONICS	(613) 592-0060
Mississauga, Ontario - VITEL ELECTRONICS	(416) 676-9720
Lachine, Quebec - VITEL ELECTRONICS	(514) 636-5951
IDAHO	
INTERMOUNTAIN TECH MKTG, INC	(208) 888-6071
ILLINOIS	
HEARTLAND TECH MKTG, INC	(312) 577-9222
INDIANA	
Huntington - ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3450
Indianapolis - ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3450
IOWA	
LORENZ SALES	(319) 377-4666
KANSAS	
Merriam – LORENZ SALES	(913) 469-1312
Wichita – LORENZ SALES	(316) 721-0500
KENTUCKY	
ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3452
MICHIGAN	
Birmingham - MIKE RAICK ASSOCIATES	(313) 644-5040
Holland – COM-TEK SALES, INC	(616) 392-7100
Novi – COM-TEK SALES, INC	(313) 344-1409
MINNESOTA	
Mel Foster Tech. Sales, Inc.	(612) 941-9790
MISSOURI	
LORENZ SALES	(314) 997-4558
NEBRASKA	
LORENZ SALES	(402) 475-4660
NEW MEXICO	
THORSON DESERT STATES	(505) 293-8555
NEW YORK	
East Syracuse – NYCOM, INC	(315) 437-8343
Woodbury – COMPONENT CONSULTANTS, INC	(516) 364-8020
OHIO	
Centerville – DOLFUSS ROOT & CO	(513) 433-6776
Columbus – DOLFUSS ROOT & CO	(614) 885-4844
Strongsville – DOLFUSS ROOT & CO	(216) 238-0300
OREGON	
ELECTRA TECHNICAL SALES, INC	(503) 643-5074
PENNSYLVANIA	
DOLFUSS ROOT & CO	(412) 221-4420
PUERTO RICO	
COMP REP ASSOC, INC	(809) 746-6550
UTAH, R² MARKETING	(801) 595-0631
WASHINGTON	
ELECTRA TECHNICAL SALES	(206) 821-7442
WISCONSIN	
HEARTLAND TECH MKTG, INC	(414) 792-0920

Advanced Micro Devices, Inc. 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA
Tel: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450
APPLICATIONS HOTLINE & LITERATURE ORDERING • TOLL FREE: (800) 222-9323 • (408) 749-5703

7/27/90
BTA/10M/9-90 Printed in USA