Rochester Electronics®

Datasheet

# 80C186

## *CHMOS High Integration 16-Bit Microprocessor*

The 80C186 is a CHMOS high integration microprocessor. In 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 80C186 is 100% pin-for-pin compatible with the NMOS 80186 (except for 8087 applications). The “enhanced” mode of operation allows the full feature set of the 80C186 to be used. The 80C186 is upward compatible with 8086 and 8088 software and fully compatible with 80186 and 80188 software.

### Rochester Electronics Manufactured Components

Rochester branded components are manufactured using either die/wafers purchased from the original suppliers or Rochester wafers recreated from the original IP. All re-creations are done with the approval of the Original Component Manufacturer (OCM).

Parts are tested using original factory test programs or Rochester developed test solutions to guarantee product meets or exceeds the OCM data sheet.

### Quality Overview

- ISO-9001
- AS9120 certification
- Qualified Manufacturers List (QML) MIL-PRF-35835
    - Class Q Military
    - Class V Space Level
- Qualified Suppliers List of Distributors (QSLD)
    - Rochester is a critical supplier to DLA and meets all industry and DLA standards.

Rochester Electronics, LLC is committed to supplying products that satisfy customer expectations for quality and are equal to those originally supplied by industry manufacturers.

*The original manufacturer’s datasheet accompanying this document reflects the performance and specifications of the Rochester manufactured version of this device. Rochester Electronics guarantees the performance of its semiconductor products to the original OCM specifications. ‘Typical’ values are for reference purposes only. Certain minimum or maximum ratings may be based on product characterization, design, simulation, or sample testing.*

FOR REFERENCE ONLY

INTEL CORP (UP/PRPHLS)

PRELIMINARY

T-49-17-15

# 80C186
# CHMOS HIGH INTEGRATION 16-BIT MICROPROCESSOR

- **Operation Modes Include:**
  - **Enhanced Mode Which Has**
    - **DRAM Refresh Control Unit**
    - **Power-Save Mode**
    - **Direct Interface to New Numerics Coprocessor**
  - **Compatible Mode**
    - **NMOS 80186 Pin-for-Pin Replacement for Non-Numerics Applications**
- **Integrated Feature Set**
  - **Enhanced 80C86/C88 CPU**
  - **Clock Generator**
  - **2 Independent DMA Channels**
  - **Programmable Interrupt Controller**
  - **3 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 16 MHz (80C186-16), 12.5 MHz (80C186-12) and 10 MHz (80C186) Versions**
- **Direct Addressing Capability to 1 Mbyte Memory and 64 Kbyte I/O**
- **Completely Object Code Compatible with All Existing 8086/8088 Software and Also Has 10 Additional Instructions Over 8086/8088**
- **Complete System Development Support**
  - **All 8086 and NMOS 80186 Software Development Tools Can Be Used for 80C186 System Development**
    - **ASM 86 Assembler, PL/M-86, Pascal-86, FORTRAN-86, C-86 and System Utilities**
    - **In-Circuit-Emulator (ICE™-186)**
- **High Performance Numeric Coprocessing Capability through 80C187 Interface**
- **Available in 68-Pin:**
  - **Plastic Leaded Chip Carrier (PLCC)**
  - **Ceramic Pin Grid Array (PGA)**
  - **Ceramic Leadless Chip Carrier (JEDEC A Package)**

  (See Packaging Outlines and Dimensions, Order Number 231369)
- **Available in EXPRESS Extended Temperature Range (−40°C to +85°C)**
- **Available in Military:**
  - **Different Specifications**
  - **10 MHz (M80C186-10) and 12.5 MHz (M80C186-12) Versions**

  (See M80C186 data sheet, Order Number 270500 for specifications)

24

The Intel 80C186 is a CHMOS high integration microprocessor. In 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 80C186 is 100% pin-for-pin compatible with the NMOS 80186 (except for 8087 applications). The "enhanced" mode of operation allows the full feature set of the 80C186 to be used. The 80C186 is upward compatible with 8086 and 8088 software and fully compatible with 80186 and 80188 software.



October 1991
Order Number: 270354-005

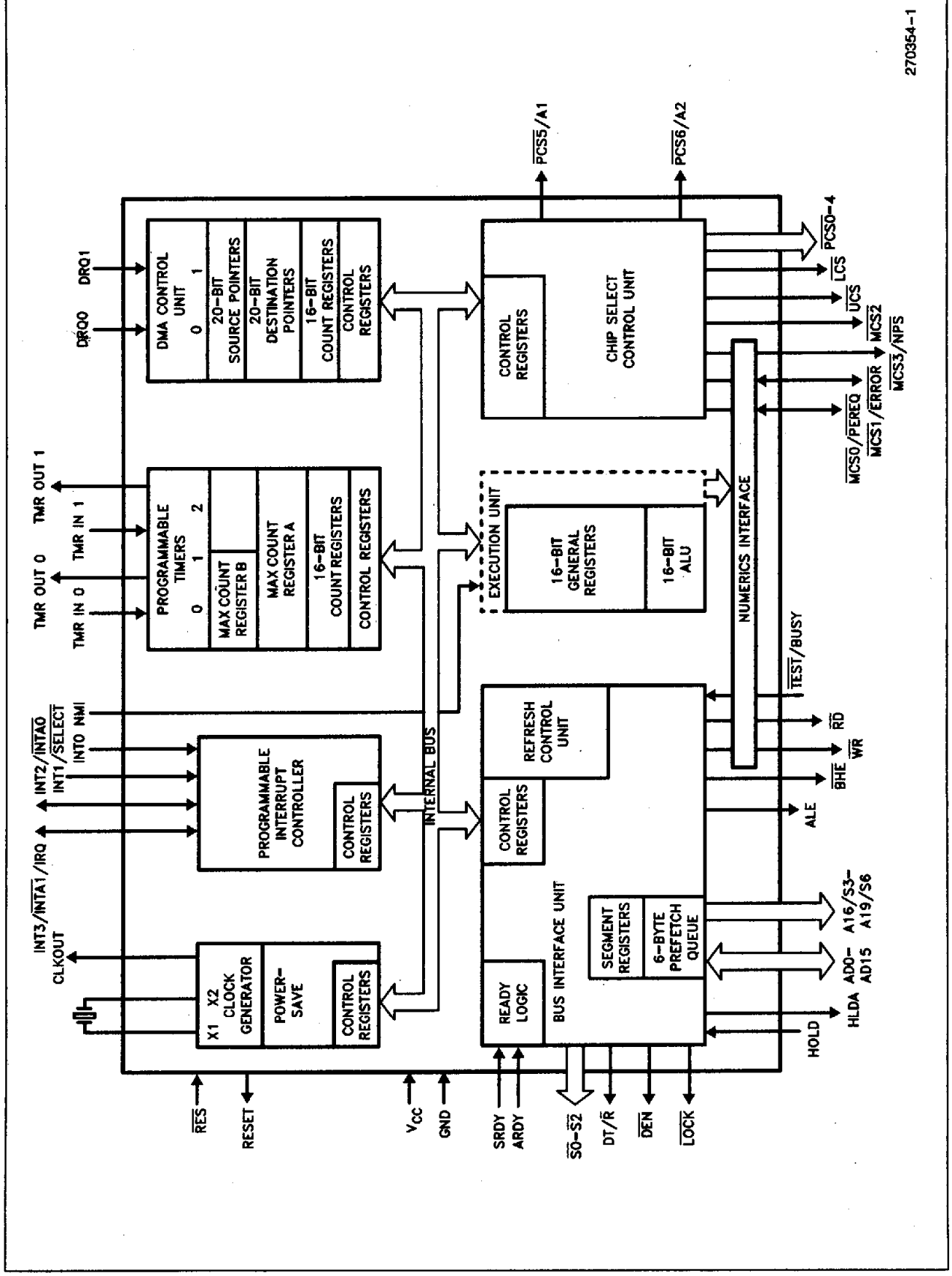
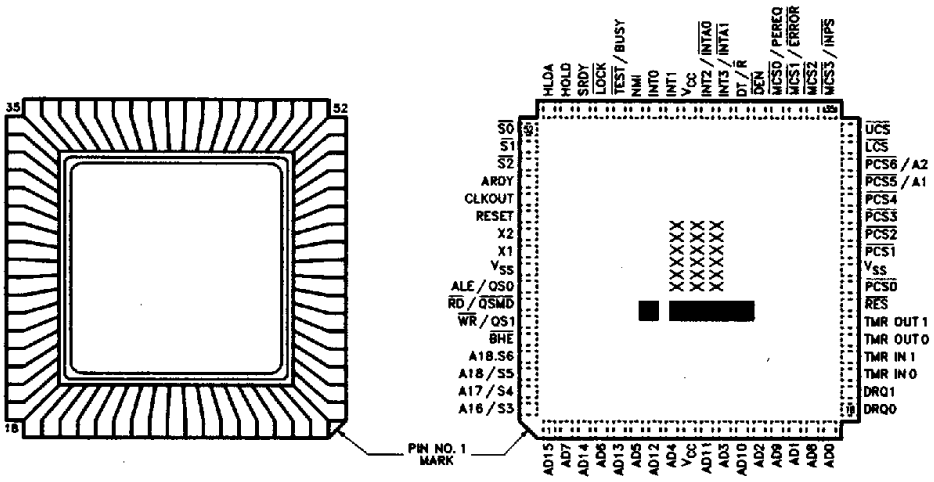



**Figure 1. 80C186 Block Diagram**

### Ceramic Leadless Chip Carrier (JEDEC Type A)

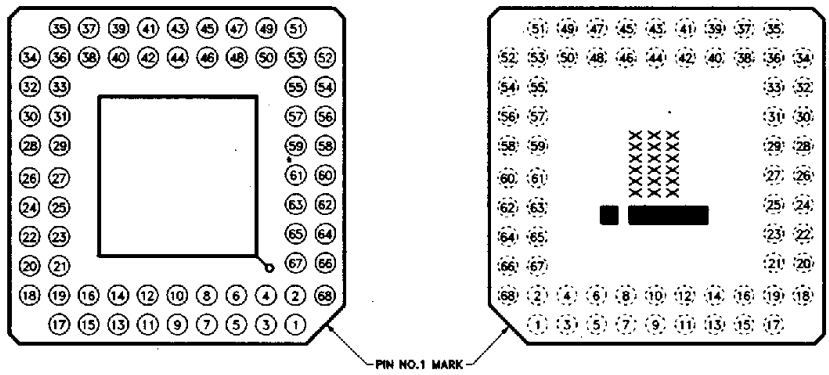
**Contacts Facing Up** | **Contacts Facing Down**




270354-2

### Ceramic Pin Grid Array

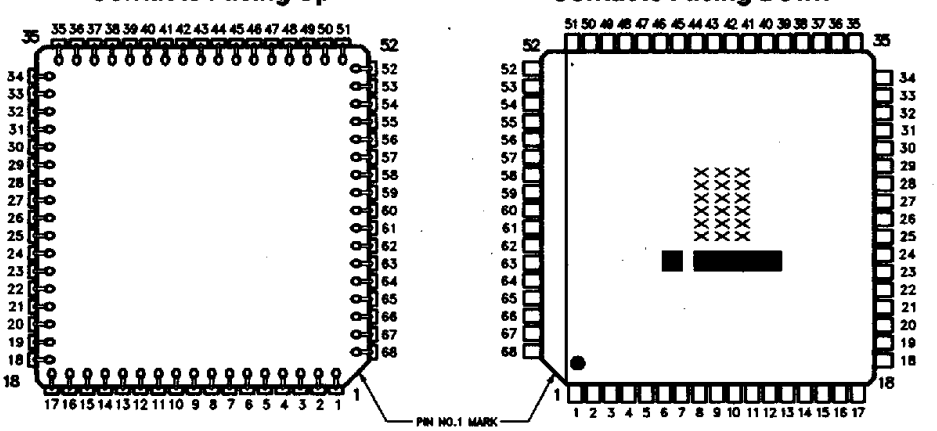
**Pins Facing Up** | **Pins Facing Down**




270354-3

### Plastic Leaded Chip Carrier

**Contacts Facing Up** | **Contacts Facing Down**




270354-19

**Figure 2. 80C186 Pinout Diagrams**

24

**Table 1. 80C186 Pin Description**

| Symbol | Pin No. | Type | Name and Function |
|---|---|---|---|
| $V_{CC}$ | 9<br>43 | I<br>I | System Power: +5 volt power supply. |
| $V_{SS}$ | 26<br>60 | I<br>I | System Ground. |
| RESET | 57 | O | RESET Output indicates that the 80C186 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 $\overline{RES}$ signal. Reset goes inactive 2 clockout periods after $\overline{RES}$ goes inactive. When tied to the $\overline{TEST}$/BUSY pin, RESET forces the 80C186 into enhanced mode. RESET is not floated during bus hold. |
| X1<br>X2 | 59<br>58 | I<br>O | 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). |
| CLKOUT | 56 | O | 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. |
| $\overline{RES}$ | 24 | I | An active $\overline{RES}$ causes the 80C186 to immediately terminate its present activity, clear the internal logic, and enter a dormant state. This signal may be asynchronous to the 80C186 clock. The 80C186 begins fetching instructions approximately 6½ clock cycles after $\overline{RES}$ 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 $\overline{RES}$ held LOW. $\overline{RES}$ is internally synchronized. This input is provided with a Schmitt-trigger to facilitate power-on $\overline{RES}$ generation via an RC network. |
| $\overline{TEST}$/BUSY | 47 | I/O | The $\overline{TEST}$ pin is sampled during and after reset to determine whether the 80C186 is to enter Compatible or Enhanced Mode. Enhanced Mode requires $\overline{TEST}$ to be HIGH on the rising edge of $\overline{RES}$ and LOW four CLKOUT cycles later. Any other combination will place the 80C186 in Compatible Mode. During power-up, active $\overline{RES}$ is required to configure $\overline{TEST}$/BUSY as an input. A weak internal pullup ensures a HIGH state when the input is not externally driven.<br>$\overline{TEST}$—In Compatible Mode this pin is configured to operate as $\overline{TEST}$. This pin is examined by the WAIT instruction. If the $\overline{TEST}$ input is HIGH when WAIT execution begins, instruction execution will suspend. $\overline{TEST}$ will be resampled every five clocks until it goes LOW, at which time execution will resume. If interrupts are enabled while the 80C186 is waiting for $\overline{TEST}$, interrupts will be serviced.<br>BUSY—In Enhanced Mode, this pin is configured to operate as BUSY. The BUSY input is used to notify the 80C186 of Numerics Processor Extension activity. Floating point instructions executing in the 80C186 sample the BUSY pin to determine when the Numerics Processor is ready to accept a new command. BUSY is active HIGH. |
| TMR IN 0<br>TMR IN 1 | 20<br>21 | I<br>I | 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<br>TMR OUT 1 | 22<br>23 | O<br>O | Timer outputs are used to provide single pulse or continous waveform generation, depending upon the timer mode selected. These outputs are not floated during a bus hold. |

**Table 1. 80C186 Pin Description** (Continued)

| Symbol | Pin No. | Type | Name and Function |
|---|---|---|---|
| DRQ0<br>DRQ1 | 18<br>19 | I<br>I | 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. |
| NMI | 46 | I | 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. |
| INT0<br>INT1/$\overline{\text{SELECT}}$<br>INT2/$\overline{\text{INTA0}}$<br>INT3/$\overline{\text{INTA1}}$/IRQ | 45<br>44<br>42<br>41 | I<br>I<br>I/O<br>I/O | 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 data sheet). |
| A19/S6<br>A18/S5<br>A17/S4<br>A16/S3 | 65<br>66<br>67<br>68 | O<br>O<br>O<br>O | Address Bus Outputs (16–19) and Bus Cycle Status (3–6) indicate the four most significant address bits during $T_1$. These signals are active HIGH.<br>During $T_2$, $T_3$, $T_W$, and $T_4$, 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. |
| AD15<br>AD14<br>AD13<br>AD12<br>AD11<br>AD10<br>AD9<br>AD8<br>AD7<br>AD6<br>AD5<br>AD4<br>AD3<br>AD2<br>AD1<br>AD0 | 1<br>3<br>5<br>7<br>10<br>12<br>14<br>16<br>2<br>4<br>6<br>8<br>11<br>13<br>15<br>17 | I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O | Address/Data Bus (0–15) signals constitute the time multiplexed memory or I/O address ($T_1$) and data ($T_2$, $T_3$, $T_W$, and $T_4$) bus. The bus is active HIGH. $A_0$ is analogous to $\overline{\text{BHE}}$ for the lower byte of the data bus, pins $D_7$ through $D_0$. It is LOW during $T_1$ when a byte is to be transferred onto the lower portion of the bus in memory or I/O operations. These pins are floated during a bus hold or reset. |

24

**Table 1. 80C186 Pin Description** (Continued)

| Symbol | Pin No. | Type | Name and Function |
|---|---|---|---|
| $\overline{BHE}$ | 64 | O | The $\overline{BHE}$ (Bus High Enable) signal is analogous to A0 in that it is used to enable data on to the most significant half of the data bus, pins D15–D8. $\overline{BHE}$ will be LOW during $T_1$ when the upper byte is transferred and will remain LOW through $T_3$ AND $T_W$. $\overline{BHE}$ does not need to be latched. $\overline{BHE}$ will float during HOLD or RESET.<br>In Enhanced Mode, $\overline{BHE}$ will also be used to signify DRAM refresh cycles. A refresh cycle is indicated by both $\overline{BHE}$ and A0 being HIGH. |

**$\overline{BHE}$ and A0 Encodings**

| $\overline{BHE}$ Value | A0 Value | Function |
|---|---|---|
| 0 | 0 | Word Transfer |
| 0 | 1 | Byte Transfer on upper half of data bus (D15–D8) |
| 1 | 0 | Byte Transfer on lower half of data bus ($D_7$–$D_0$) |
| 1 | 1 | Refresh |

| | | | |
|---|---|---|---|
| ALE/QS0 | 61 | O | Address Latch Enable/Queue Status 0 is provided by the 80C186 to latch the address. ALE is active HIGH, with addresses guaranteed valid on the trailing edge. |
| $\overline{WR}$/QS1 | 63 | O | 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 80C186 is in queue status mode, the ALE/QS0 and $\overline{WR}$/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 |

| | | | |
|---|---|---|---|
| $\overline{RD}$/$\overline{QSMD}$ | 62 | O/I | Read Strobe is an active LOW signal which indicates that the 80C186 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 $\overline{RD}$/$\overline{QSMD}$ is HIGH during RESET. Following RESET the pin is sampled to determine whether the 80C186 is to provide ALE, $\overline{RD}$, and $\overline{WR}$, or queue status information. To enable Queue Status Mode, $\overline{RD}$ must be connected to GND. $\overline{RD}$ will float during bus HOLD. |
| ARDY | 55 | I | Asynchronous Ready informs the 80C186 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 80C186 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. |
| SRDY | 49 | I | Synchronous Ready informs the 80C186 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 synchonize 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. |

**Table 1. 80C186 Pin Description** (Continued)

T-49-17-15

| Symbol | Pin No. | Type | Name and Function |
|---|---|---|---|
| $\overline{LOCK}$ | 48 | O | $\overline{LOCK}$ output indicates that other system bus masters are not to gain control of the system bus. $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ 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 $\overline{LOCK}$ is asserted. $\overline{LOCK}$ floats during bus hold or reset. |
| $\overline{S0}$<br>$\overline{S1}$<br>$\overline{S2}$ | 52<br>53<br>54 | O<br>O<br>O | Bus cycle status $\overline{S0}$–$\overline{S2}$ are encoded to provide bus-transaction information:<br>**80C186 Bus Cycle Status Information**<br>$\overline{S2}$ $\overline{S1}$ $\overline{S0}$ — Bus Cycle Initiated<br>0 0 0 — Interrupt Acknowledge<br>0 0 1 — Read I/O<br>0 1 0 — Write I/O<br>0 1 1 — Halt<br>1 0 0 — Instruction Fetch<br>1 0 1 — Read Data from Memory<br>1 1 0 — Write Data to Memory<br>1 1 1 — Passive (no bus cycle)<br>The status pins float during HOLD.<br>$\overline{S2}$ may be used as a logical M/$\overline{IO}$ indicator, and $\overline{S1}$ as a DT/$\overline{R}$ indicator. |
| HOLD<br>HLDA | 50<br>51 | I<br>O | HOLD indicates that another bus master is requesting the local bus. The HOLD input is active HIGH. The 80C186 generates HLDA (HIGH) in response to a HOLD request. Simultaneous with the issuance of HLDA, the 80C186 will float the local bus and control lines. After HOLD is detected as being LOW, the 80C186 will lower HLDA. When the 80C186 needs to run another bus cycle, it will again drive the local bus and control lines.<br>In Enhanced Mode, HLDA will go low when a DRAM refresh cycle is pending in the 80C186 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 80C186 may execute the refresh cycle. |
| $\overline{UCS}$ | 34 | O/I | 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. $\overline{UCS}$ does not float during bus hold. The address range activating $\overline{UCS}$ is software programmable.<br>$\overline{UCS}$ and $\overline{LCS}$ are sampled upon the rising edge of $\overline{RES}$. If both pins are held low, the 80C186 will enter ONCE Mode. In ONCE Mode all pins assume a high impedance state and remain so until a subsequent RESET. $\overline{UCS}$ has a weak internal pullup that is active during RESET to ensure that the 80C186 does not enter ONCE mode inadvertently. |

24

**Table 1. 80C186 Pin Description** (Continued) T-49-17-15

| Symbol | Pin No. | Type | Name and Function |
|---|---|---|---|
| $\overline{LCS}$ | 33 | O/I | Lower Memory Chip Select is active LOW whenever a memory reference is made to the defined lower portion (1K–256K) of memory. $\overline{LCS}$ does not float during bus HOLD. The address range activating $\overline{LCS}$ is software programmable.<br>$\overline{UCS}$ and $\overline{LCS}$ are sampled upon the rising edge of $\overline{RES}$. If both pins are held low, the 80C186 will enter ONCE Mode. In ONCE Mode all pins assume a high impedance state and remain so until a subsequent RESET. $\overline{LCS}$ has a weak internal pullup that is active only during RESET to ensure that the 80C186 does not enter ONCE mode inadvertently. |
| $\overline{MCS0}$/PEREQ<br>$\overline{MCS1}$/$\overline{ERROR}$<br>$\overline{MCS2}$<br>$\overline{MCS3}$/$\overline{NPS}$ | 38<br>37<br>36<br>35 | O/I<br>O/I<br>O<br>O | 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 $\overline{MCS0}$–3 are software programmable.<br>In Enhanced Mode, $\overline{MCS0}$ becomes a PEREQ input (Processor Extension Request). When connected to the Numerics Processor Extension, this input is used to signal the 80C186 when to make numeric data transfers to and from the NPX. $\overline{MCS3}$ becomes $\overline{NPS}$ (Numeric Processor Select) which may only be activated by communication to the Numerics Processor Extension. $\overline{MCS1}$ becomes $\overline{ERROR}$ in enhanced mode and is used to signal numerics coprocessor errors.<br>$\overline{MCS0}$/PEREQ and $\overline{MCS1}$/$\overline{ERROR}$ have weak internal pullups which are active during reset. |
| $\overline{PCS0}$<br>$\overline{PCS1}$<br>$\overline{PCS2}$<br>$\overline{PCS3}$<br>$\overline{PCS4}$ | 25<br>27<br>28<br>29<br>30 | O<br>O<br>O<br>O<br>O | Peripheral Chip Select signals 0–4 are active LOW when a reference is made to the defined peripheral area (64K byte I/O or 1 MByte memory space). These lines do not float during bus HOLD. The address ranges activating $\overline{PCS0}$–4 are software programmable. |
| $\overline{PCS5}$/A1 | 31 | O | 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 $\overline{PCS5}$ is software-programmable. $\overline{PCS5}$/A1 does not float during bus HOLD. When programmed to provide latched A1, this pin will retain the previously latched value during HOLD. |
| $\overline{PCS6}$/A2 | 32 | O | 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 $\overline{PCS6}$ is software-programmable. $\overline{PCS6}$/A2 does not float during bus HOLD. When programmed to provide latched A2, this pin will retain the previously latched value during HOLD. |
| DT/$\overline{R}$ | 40 | O | Data Transmit/Receive controls the direction of data flow through an external data bus transceiver. When LOW, data is transferred to the 80C186. When HIGH the 80C186 places write data on the data bus. DT/$\overline{R}$ floats during a bus hold or reset. |
| $\overline{DEN}$ | 39 | O | Data Enable is provided as a data bus transceiver output enable. $\overline{DEN}$ is active LOW during each memory and I/O access (including 80C187 access). $\overline{DEN}$ is HIGH whenever DT/$\overline{R}$ changes state. During RESET, $\overline{DEN}$ is driven HIGH for one clock, then floated. $\overline{DEN}$ also floats during HOLD. |

# FUNCTIONAL DESCRIPTION

## Introduction

The following Functional Description describes the base architecture of the 80C186. The 80C186 is a very high integration 16-bit microprocessor. It combines 15–20 of the most common microprocessor system components onto one chip. The 80C186 is object code compatible with the 8086/8088 microprocessors and adds 10 new instruction types to the 8086/8088 instruction set.

The 80C186 has two major modes of operation, Compatible and Enhanced. In Compatible Mode the 80C186 is completely compatible with NMOS 80186, with the exception of 8087 support. The Enhanced mode adds three new features to the system design. These are Power-Save control, Dynamic RAM refresh, and an asynchronous Numerics Coprocessor interface.

## 80C186 BASE ARCHITECTURE

The 8086, 8088, 80186, and 80188 family all contain the same basic set of registers, instructions, and addressing modes. The 80C186 processor is upward compatible with the 8086 and 8088 CPUs.

## Register Set

The 80C186 base architecture has fourteen registers as shown in Figures 3a and 3b. 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.

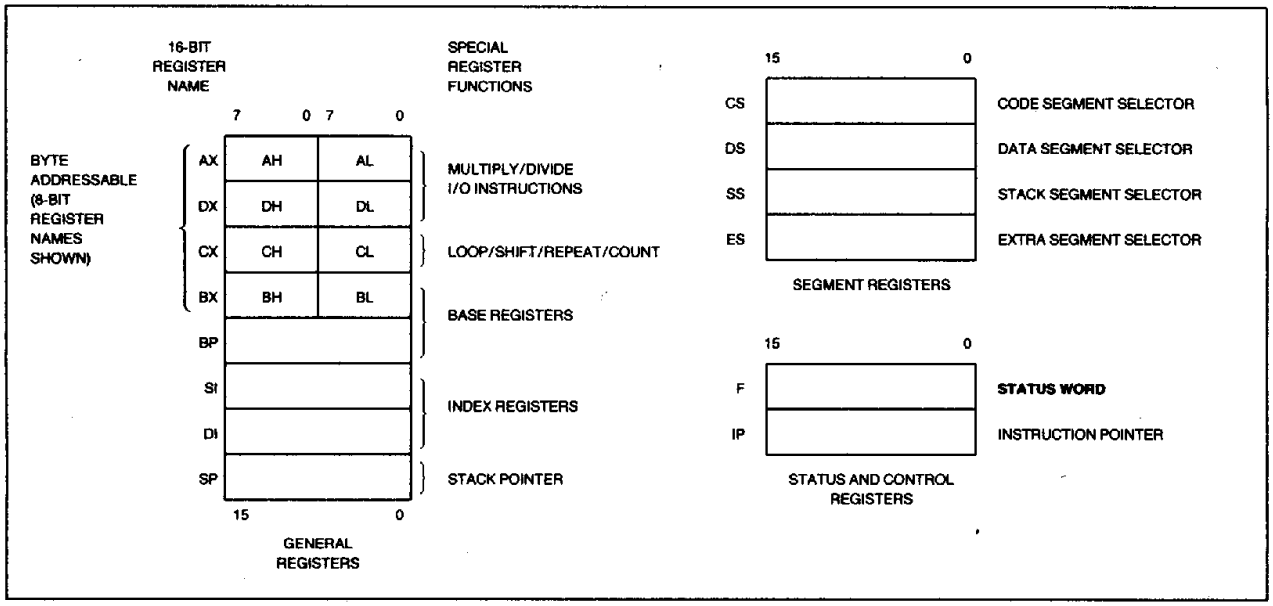



**Figure 3a. 80C186 Register Set**

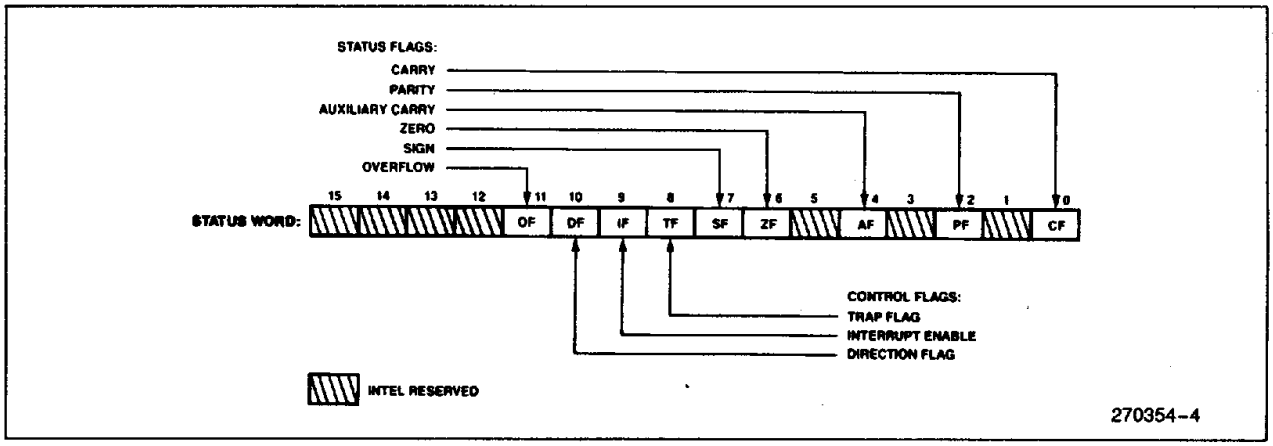



**Figure 3b. Status Word Format**

### Segment Registers

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

### Base and Index Registers

Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode 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 80C186 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 3a and 3b).

## 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 2.

**Table 2. 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 | Set on carry from or borrow to the low order four bits of AL; cleared otherwise |
| 6 | ZF | Zero Flag—Set if result is zero; cleared otherwise |
| 7 | SF | Sign Flag—Set equal to high-order bit of result (0 if positive, 1 if negative) |
| 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 Figure 4.

An 80C186 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 data sheet.

## 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 register (code, data, stack, extra). The physical address is calculated by shifting the base value LEFT by four bits and adding the 16-bit offset value to yield a 20-bit physical address (see Figure 5). This allows for a 1 MByte 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 6) as independent modules that require areas for code and data, a stack, and access to external data areas.

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

| 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 to 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 $\overline{\text{TEST}}$ pin active |
| ESC | Escape to extension processor |
| 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 |

**Figure 4. 80C186 Instruction Set**

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

| | |
|---|---|
| JO | Jump if overflow |
| JP/JPE | Jump if parity/parity even |
| JS | Jump if sign |
| **UNCONDITIONAL TRANSFERS** | |
| CALL | Call procedure |
| RET | Return from procedure |
| JMP | Jump |
| **ITERATION CONTROLS** | |
| LOOP | Loop |
| LOOPE/LOOPZ | Loop if equal/zero |
| LOOPNE/LOOPNZ | Loop if not equal/not zero |
| JCXZ | Jump if register CX = 0 |
| **INTERRUPTS** | |
| INT | Interrupt |
| INTO | Interrupt if overflow |
| IRET | Interrupt return |

**Figure 4. 80C186 Instruction Set** (Continued)

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.

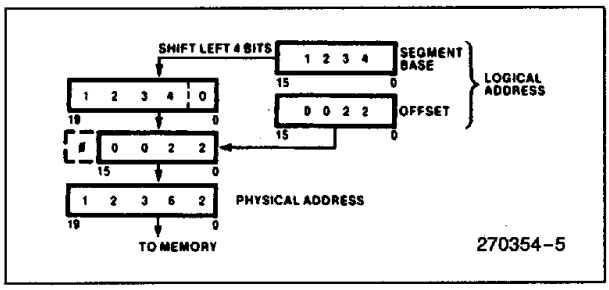



**Figure 5. Two Component Address**

**Table 3. 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. |

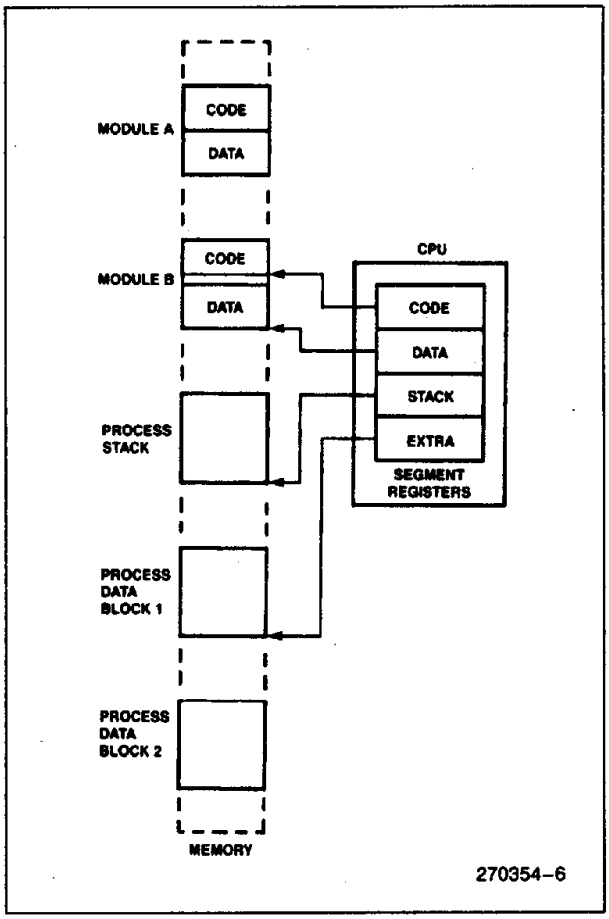



**Figure 6. Segmented Memory Helps Structure Software**

## Addressing Modes

The 80C186 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 general registers.
- *Immediate Operand Mode:* The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: 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 indexed Mode with Displacement:* The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

## Data Types

The 80C186 directly supports the following data types:

- *Integer:* A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation. Signed 32- and 64-bit integers are supported using a Numeric Data Coprocessor with the 80C186.
- *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.
- *Floating Point:* A signed 32-, 64-, or 80-bit real number representation. (Floating point operands are supported using a Numeric Data Coprocessor with the 80C186.)

In general, individual data elements must fit within defined segment limits. Figure 7 graphically represents the data types supported by the 80C186.

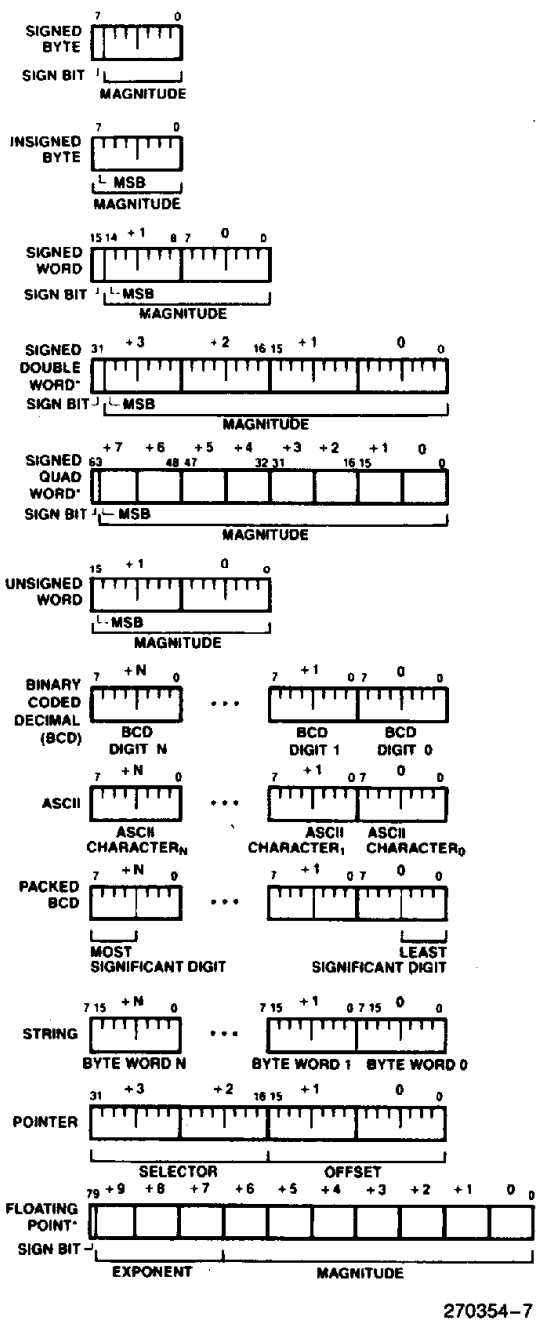
24

## 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. 8-bit port addresses are zero extended such that $A_{15}$–$A_8$ are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.

## 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.




**NOTE:**
*Supported by using a Numeric Data Coprocessor with the 80C186.

**Figure 7. 80C186 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 4 shows the 80C186 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 noncascaded 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 80C186 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, INT0, 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 4), 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 80C186 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 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

**Table 4. 80C186 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 (Coprocessor) | 1, 3 |
| Timer 0 Interrupt | 8 | 20H | 2A | | 4 |
| Timer 1 Interrupt | 18 | 48H | 2B | | 4, 6 |
| Timer 2 Interrupt | 19 | 4CH | 2C | | 4, 6 |
| Reserved | 9 | 24H | 3 | | |
| DMA 0 Interrupt | 10 | 28H | 4 | | 6 |
| DMA 1 Interrupt | 11 | 2CH | 5 | | 6 |
| INT0 Interrupt | 12 | 30H | 6 | | |
| INT1 Interrupt | 13 | 34H | 7 | | |
| INT2 Interrupt | 14 | 38H | 8 | | |
| INT3 Interrupt | 15 | 3CH | 9 | | |
| Numerics Coprocessor Exception | 16 | 40H | 1 | ESC Opcodes (Numerics Coprocessor) | 1, 5 |
| Reserved | 17 | 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 8086.
3. An ESC (coprocessor) opcode will cause a trap if the 80C186 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 80C186 is not directly compatible with the 80186 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. Numerics coprocessor exceptions are detected by the 80C186 upon execution of a subsequent numerics instruction.
6. The vector type numbers for these sources are programmable in Slave Mode.



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 bit. No external interrupt acknowledge sequence is performed. The IF bit is cleared at the beginning of an 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 one-byte version of the INT instruction. It uses 12 as an index into the service routine address table (because it is a type 3 interrupt).

### INTO DETECTED OVERFLOW EXCEPTION (TYPE4)

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 (D8H–DFH). In compatible mode operation, ESC opcodes will always generate this exception. In enhanced mode operation, the exception will be generated only if a bit in the relocation register is set. 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:**
80C186 processing of ESC (numerics coprocessor) opcodes differs substantially from the 80186.

**NUMERICS COPROCESSOR EXCEPTION (TYPE 16)**

An interrupt generated in response to an unmasked error in the 80C187 Numerics Coprocessor Extension. In general, the 80C187 does not detect an error until the instruction after the error occurred. A numerics coprocessor error is signalled to the 80C187 on its $\overline{\text{ERROR}}$ input pin.

Hardware-generated interrupts are divided into two groups: maskable interrupts and non-maskable interrupts. The 80C186 provides maskable hardware interrupt request pins INT0–INT3. In addition, maskable interrupts may be generated by the 80C186 integrated DMA controller and the integrated timer unit. The vector types for these interrupts is shown in Table 4. 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 data sheet.

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 bit. If the interrupt return re-enables interrupts, and another interrupt is pending, the 80C186 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 is accomplished by driving the $\overline{\text{RES}}$ input pin LOW. $\overline{\text{RES}}$ must be LOW during power-up to ensure proper device initialization. $\overline{\text{RES}}$ forces the 80C186 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as $\overline{\text{RES}}$ is active. After $\overline{\text{RES}}$ becomes inactive and an internal processing interval elapses, the 80C186 begins execution with the instruction at physical location FFFF0(H). $\overline{\text{RES}}$ also sets some registers to predefined values as shown in Table 5.

**Table 5. 80C186 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) |

## 80C186 CLOCK GENERATOR

The 80C186 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 80C186 oscillator circuit 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 8c. This is used as the time base for the 80C186. 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 80C186. The two recommended crystal configurations are shown in Figures 8a and 8b. When used in third-overtone mode the tank circuit shown in Figure 8b is recommended for stable operation. The sum of the stray capacitances and load-

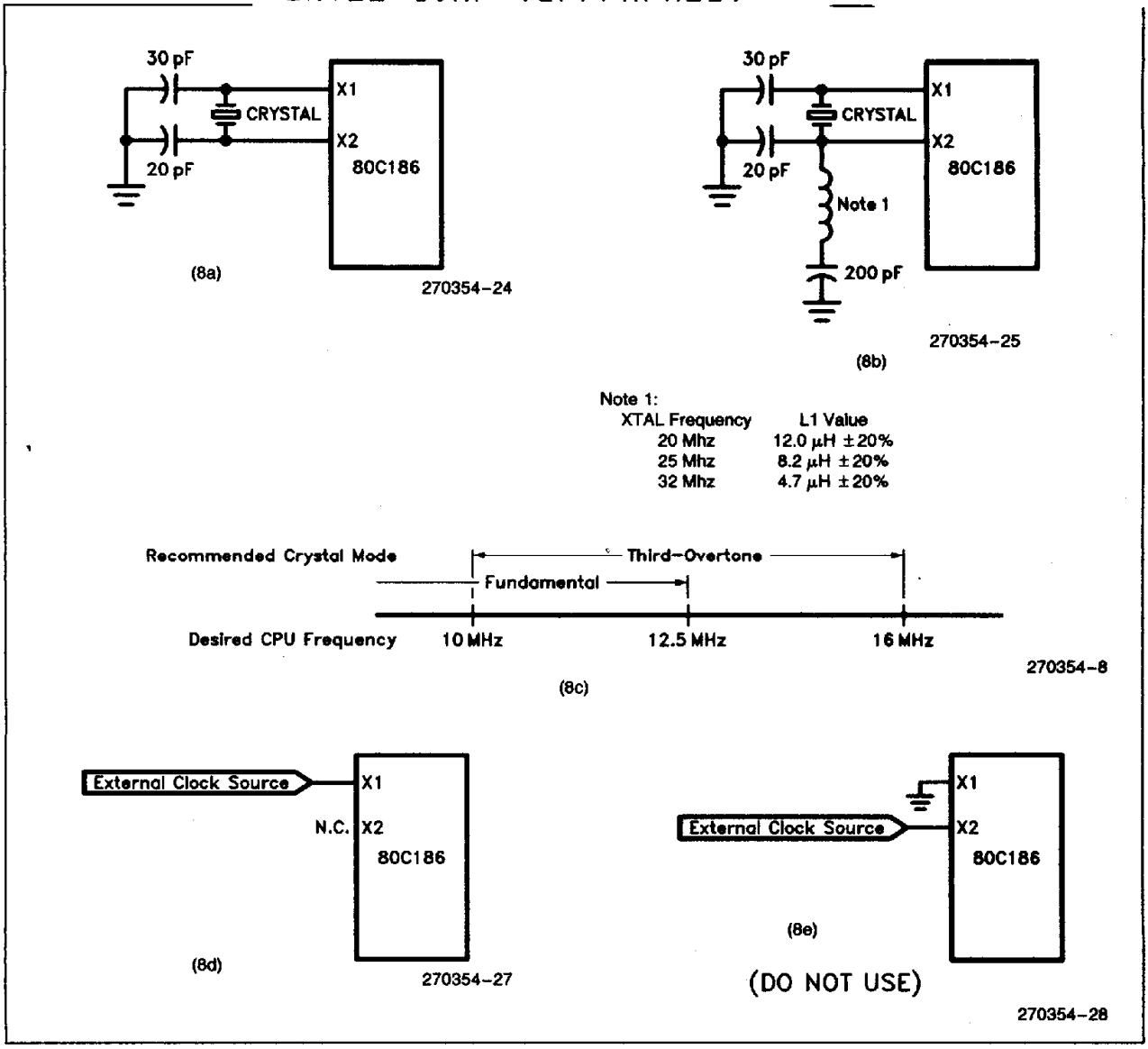



**Figure 8. 80C186 Oscillator Configurations (see text)**

ing 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 8d. The configuration shown in Figure 8e is not recommended.

Intel recommends the following values for crystal selection parameters.

| | |
|---|---|
| Temperature Range: | 0 to 70°C |
| ESR (Equivalent Series Resistance): | 40Ω max |
| $C_0$ (Shunt Capacitance of Crystal): | 7.0 pF max |
| $C_1$ (Load Capacitance): | 20 pF ± 2 pF |
| Drive Level: | 1 mW max |

## Clock Generator

The 80C186 clock generator provides the 50% duty cycle processor clock for the 80C186. 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 80C186. This may be used to drive other system components. All timings are referenced to the output clock.

## READY Synchronization

The 80C186 provides both synchronous and asynchronous ready inputs. Asynchronous ready synchronization is accomplished by circuitry which samples ARDY in the middle of $T_2$, $T_3$, and again in the middle of each $T_W$ 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 $T_2$, $T_3$, **or** $T_W$, or at the falling edge of $T_3$ or $T_W$.

A second ready input (SRDY) is provided to interface with externally synchronized ready signals. This input is sampled at the end of $T_2$, $T_3$ and again at the end of each $T_W$ 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 80C186, 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 80C186 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 80C186 is provided with hysteresis in order to facilitate power-on Reset generation via an RC network. RESET output 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 80C186 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 80C186 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 80C186 or to strobe data from the 80C186 to memory or I/O. The ALE line provides a strobe to latch the address when it is valid. The 80C186 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), make the memory and I/O spaces nonoverlapping, or use only the integrated chip-select circuitry.

## Transceiver Control

The 80C186 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/$\overline{R}$ and $\overline{DEN}$, are generated to control the flow of data through the transceivers. The operation of these signals is shown in Table 6.

**Table 6. Transceiver Control Signals Description**

| Pin Name | Function |
|---|---|
| $\overline{DEN}$ (Data Enable) | Enables the output drivers of the transceivers. It is active LOW during memory, I/O, numeric processor extension, or INTA cycles. |
| DT/$\overline{R}$ (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 80C186 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 80C186 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 80C186 relinquishes control of the local bus, it floats $\overline{DEN}$, $\overline{RD}$, $\overline{WR}$, $\overline{S0}$–$\overline{S2}$, $\overline{LOCK}$, AD0–AD15, A16–A19, $\overline{BHE}$, and DT/$\overline{R}$ to allow another master to drive these lines directly.

The 80C186 HOLD latency time, i.e., 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 80C186 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 trans-

fer 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 80C186 has relinquished the bus and a refresh request is pending, HLDA is removed (driven low) to signal the remote processor that the 80C186 wishes to regain control of the bus. The 80C186 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 $\overline{DEN}$, $\overline{RD}$, and $\overline{WR}$ HIGH for one clock cycle, then float them.
- Drive $\overline{S0}$–$\overline{S2}$ to the inactive state (all HIGH) and then float.
- Drive $\overline{LOCK}$ HIGH and then float.
- Float AD0–15, A16–19, $\overline{BHE}$, DT/$\overline{R}$.
- Drive ALE LOW
- Drive HLDA LOW.

$\overline{RD}$/$\overline{QSMD}$, $\overline{UCS}$, $\overline{LCS}$, $\overline{MCS0}$/PEREQ, $\overline{MCS1}$/$\overline{ERROR}$, and $\overline{TEST}$/BUSY pins have internal pullup devices which are active while $\overline{RES}$ is applied. Excessive loading or grounding certain of these pins causes the 80C186 to enter an alternative mode of operation:

- $\overline{RD}$/$\overline{QSMD}$ low results in Queue Status Mode.
- $\overline{UCS}$ and $\overline{LCS}$ low results in ONCE™ Mode.
- $\overline{TEST}$/BUSY low (and high later) results in Enhanced Mode.

## INTERNAL PERIPHERAL INTERFACE

All the 80C186 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 $\overline{RD}$, $\overline{WR}$, status, address, data, etc., lines will be driven as in a normal bus cycle), but $D_{15-0}$, 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 zeros). All of the defined registers within this control block may be read or written by the 80C186 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 9). 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 data sheet). Any access to the 256 bytes of the control block activates an internal chip select.

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 10.

## CHIP-SELECT/READY GENERATION LOGIC



The 80C186 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 80C186 provides 6 memory chip select outputs for 3 address areas; upper memory, lower memory, and midrange memory. One each is provided for upper memory and lower memory, while four are provided for midrange 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

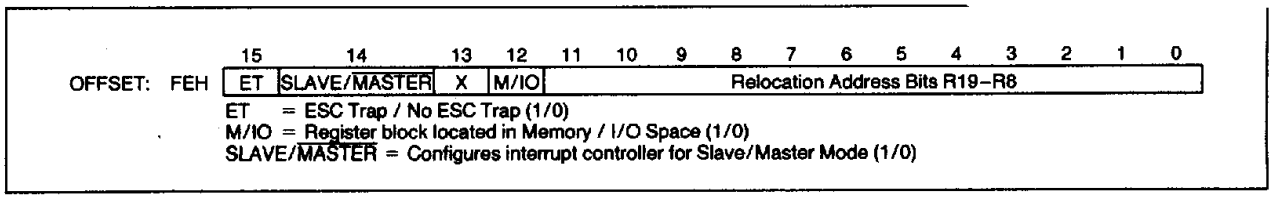



**Figure 9. Relocation Register**

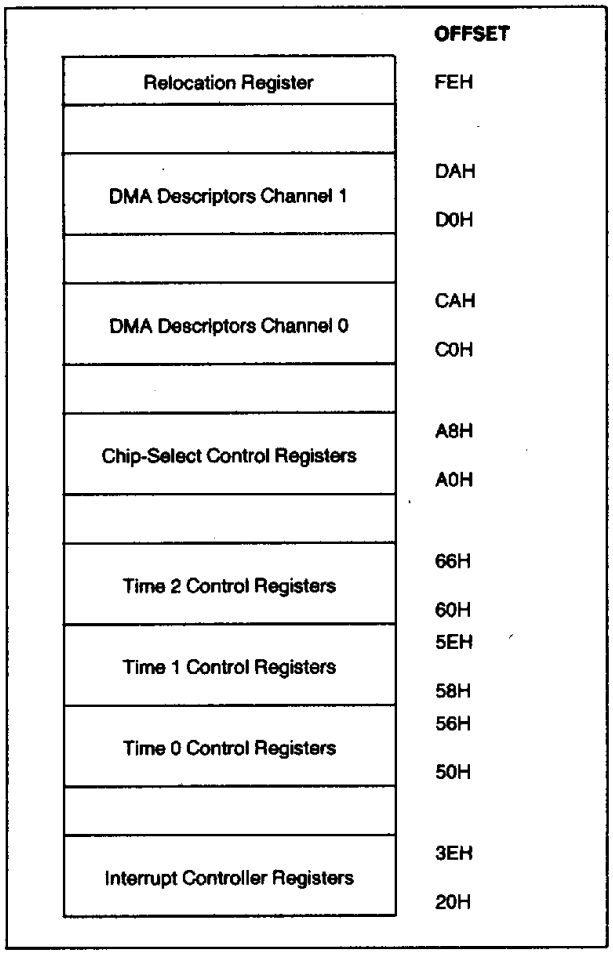



**Figure 10. Internal Register Map**

of the midrange 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 80C186 memory is arranged in words. This means that if, for example, 16 64K x 1 memories are used, the memory block size will be 128K, not 64K.

## Upper Memory $\overline{CS}$

The 80C186 provides a chip select, called $\overline{UCS}$, for the top of memory. The top of memory is usually used as the system memory because after reset the 80C186 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 7 shows the relationship between the base address selected and the size of the memory block obtained.

**Table 7. 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 11). This register is at offset A0H in the internal control block. The legal values for bits 6–13 and the resulting starting address and memory block sizes are given in Table 7. Any combination of bits 6–13 not shown in Table 7 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 0–5 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 80C186 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 8 shows the relationship between the upper address selected and the size of the memory block obtained.

**Table 8. 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 12) at offset A2H in the internal control block. The legal values for bits 6–15 and the resulting upper address and memory block sizes are given in Table 8. Any combination of bits 6–15 not shown in Table 8 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 0–5 "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 80C186 provides four $\overline{MCS}$ lines which are active within a user-locatable memory block. This block can be located within the 80C186 1M byte 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 9, is determined by bits 8–14 of the MPCS register (see Figure 13). This register is at location A8H in the internal control block. One and only one of bits 8–14 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 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 9. 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 14). This register is at offset A6H in the internal control block. 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 $\overline{MCS}$ 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 $\overline{MCS}$ lines will be active until both the MMCS and MPCS registers are accessed.



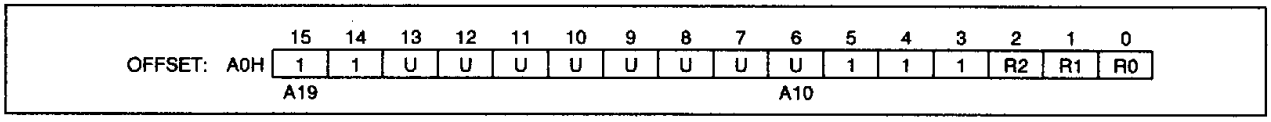



**Figure 11. UMCS Register**

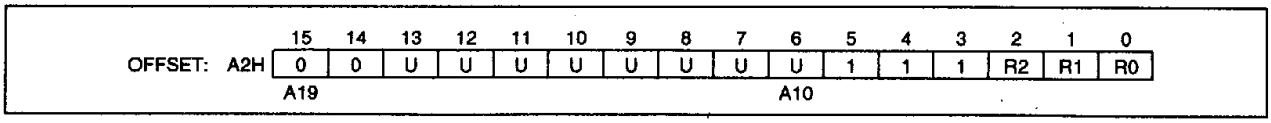



**Figure 12. LMCS Register**

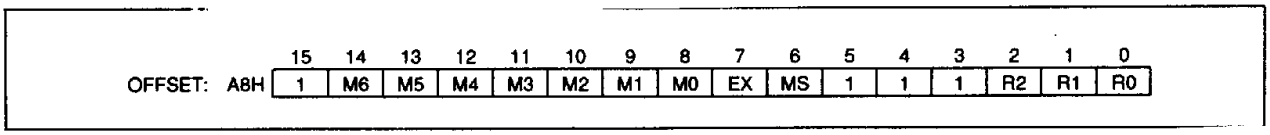



**Figure 13. MPCS Register**

T-49-17-15

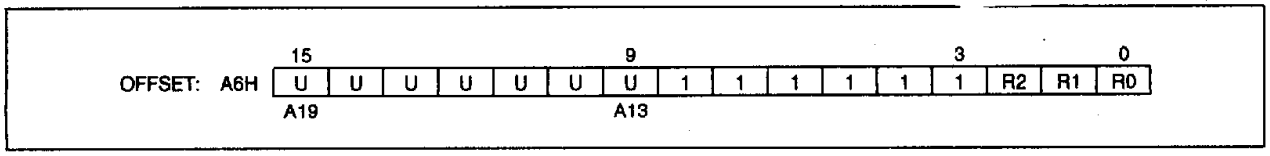



**Figure 14. MMCS Register**

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 $\overline{LCS}$ line was programmed, there would be an internal conflict between the $\overline{LCS}$ ready generation logic and the $\overline{MCS}$ ready generation logic. Likewise, if the base address were programmed at 80000H, there would be a conflict with the $\overline{UCS}$ ready generation logic. Since the $\overline{LCS}$ chip-select line does not become active until programmed, while the $\overline{UCS}$ line is active at reset, the memory base can be set only at 00000H. If this base address is selected, however, the $\overline{LCS}$ range must not be programmed.

In Enhanced Mode, three of the four $\overline{MCS}$ pins become handshaking pins for the 80C187 Numerics Processor Extension. $\overline{MCS2}$ is still available as a chip select covering one-fourth the mid-range address block, subject to the usual programming of the MPCS and MMCS registers.

## Peripheral Chip Selects

The 80C186 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 $\overline{CS}$ lines called $\overline{PCS0}$–6 are generated by the 80C186. 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.

$\overline{PCS5}$ and $\overline{PCS6}$ 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 external 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 15). 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 zeros. If the chip-select block is located in I/O space, bits 12–15 must be programmed zero, since the I/O address is only 16 bits wide. Table 10 shows the address range of each peripheral chip select with respect to the PBA contained in PACS register.

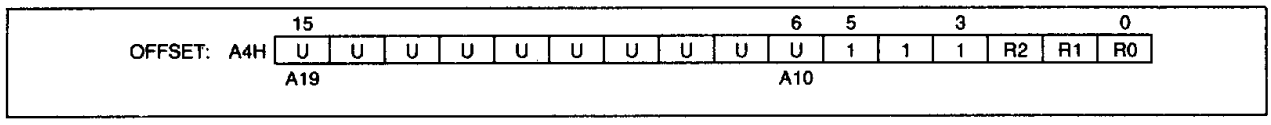



**Figure 15. PACS Register**

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

**Table 10. PCS Address Ranges**

| $\overline{PCS}$ Line | Active between Locations |
|---|---|
| $\overline{PCS0}$ | PBA —PBA+127 |
| $\overline{PCS1}$ | PBA+128—PBA+255 |
| $\overline{PCS2}$ | PBA+256—PBA+383 |
| $\overline{PCS3}$ | PBA+384—PBA+511 |
| $\overline{PCS4}$ | PBA+512—PBA+639 |
| $\overline{PCS5}$ | PBA+640—PBA+767 |
| $\overline{PCS6}$ | 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 13). The register is located at offset A8H in the internal control block. Bit 7 is used to select the function of $\overline{PCS5}$ and $\overline{PCS6}$, while bit 6 is used to select whether the peripheral chip selects are mapped into memory or I/O space. Table 11 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 11. MS, EX Programming Values**

| Bit | Description |
|---|---|
| MS | 1 = Peripherals mapped into memory space. |
| | 0 = Peripherals mapped into I/O space. |
| EX | 0 = 5 $\overline{PCS}$ lines. A1, A2 provided. |
| | 1 = 7 $\overline{PCS}$ lines. A1, A2 are not provided. |

MPCS bits 0–2 specify the READY mode for $\overline{PCS4}$–$\overline{PCS6}$ as outlined below.

## READY Generation Logic

The 80C186 can generate a READY signal internally for each of the memory or peripheral $\overline{CS}$ lines. The number of WAIT states to be inserted for each peripheral or memory is programmable to provide 0–3 wait states for all accesses to the area for which the chip select is active. In addition, the 80C186 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 $\overline{CS}$ line or group of lines generated by the 80C186. The interpretation of the READY bits is shown in Table 12.

**Table 12. 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 $\overline{PCS0}$–3 READY mode, R2–R0 of MPCS set the $\overline{PCS4}$–6 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 $\overline{UCS}$ 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 FFFBH).
- 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 80C186 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 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 13. 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 17). 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 13. DMA Control Block Format**

| Register Name | Register Address | |
|---|---|---|
| | Ch. 0 | Ch. 1 |
| Control Word | CAH | DAH |
| Transfer Count | C8H | D8H |
| Destination Pointer (upper 4 bits) | C6H | D6H |
| Destination Pointer | C4H | D4H |
| Source Pointer (upper 4 bits) | C2H | D2H |
| Source Pointer | C0H | D0H |

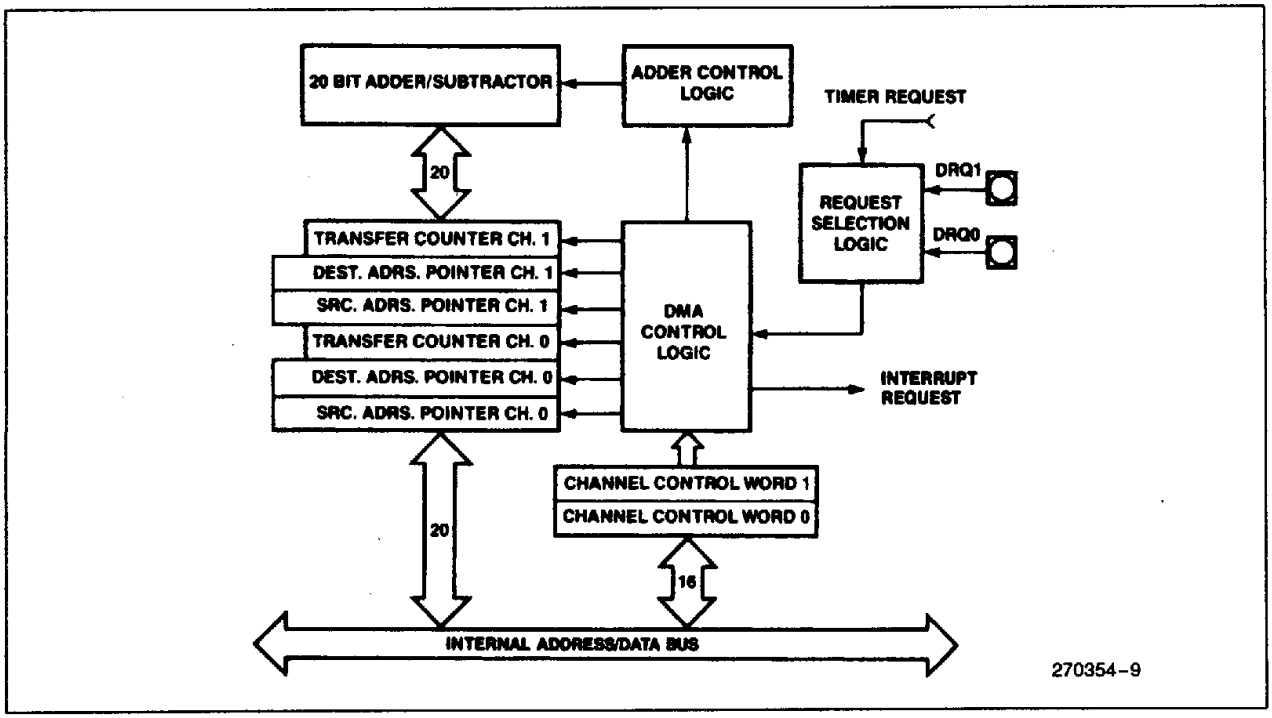



**Figure 16. DMA Unit Block Diagram**

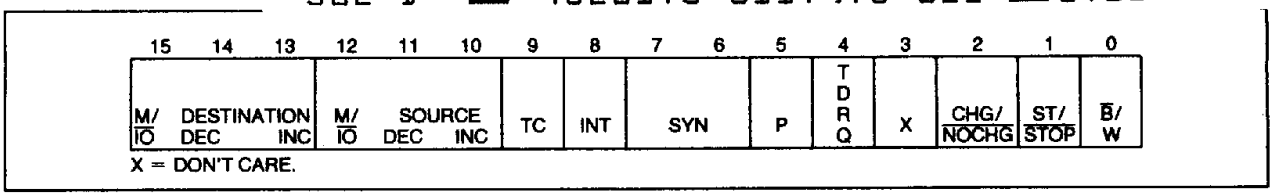



**Figure 17. DMA Control Register**

## DMA Channel Control Word Register

Each DMA Channel Control Word determines the mode of operation for the particular 80C186 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.

## 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 or 2 (depends on $\overline{B}$/W) after each transfer.

INC Increment destination pointer by 1 or 2 (depends on $\overline{B}$/W) 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 or 2 (depends on $\overline{B}$/W) after each transfer.

INC Increment source pointer by 1 or 2 (depends on $\overline{B}$/W) 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 zero. The ST/$\overline{STOP}$ bit will also be reset at this point. If cleared, the DMA controller will decrement the transfer count register for each DMA cycle, but DMA transfers will not stop when the transfer count register reaches zero.

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

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

$\overline{B}$/W: Byte/Word (0/1) transfers.

24

## 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 18). 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). The 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 count register reaches zero.

## 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 eight 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 14 shows the maximum DMA transfer rates.

**Table 14. Maximum DMA Transfer Rates at CLKOUT = 16 MHz**

| Type of Synchronization Selected | CPU Running | CPU Halted |
|---|---|---|
| Unsynchronized | 4.0MBytes/sec | 4.0MBytes/sec |
| Source Synch | 4.0MBytes/sec | 4.0MBytes/sec |
| Destination Synch | 2.7MBytes/sec | 3.2MBytes/sec |

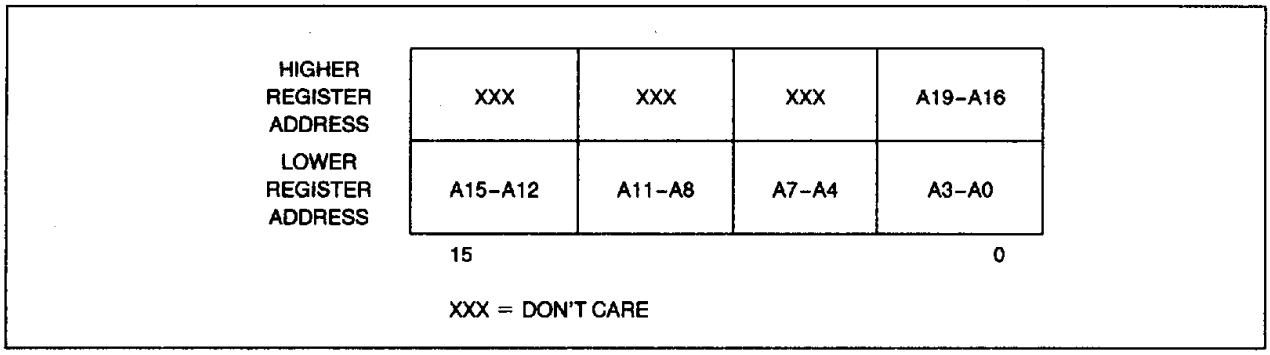



**Figure 18. DMA Pointer Register Format**

## 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 to give one channel priority over the other, or they may be programmed 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; also 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/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.

T-49-17-15

## 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 80C186 provides three internal 16-bit programmable timers (see Figure 19). 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 nonrepetitive 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, the third timer can be used as a prescaler to the other two, or as a DMA request source.

24

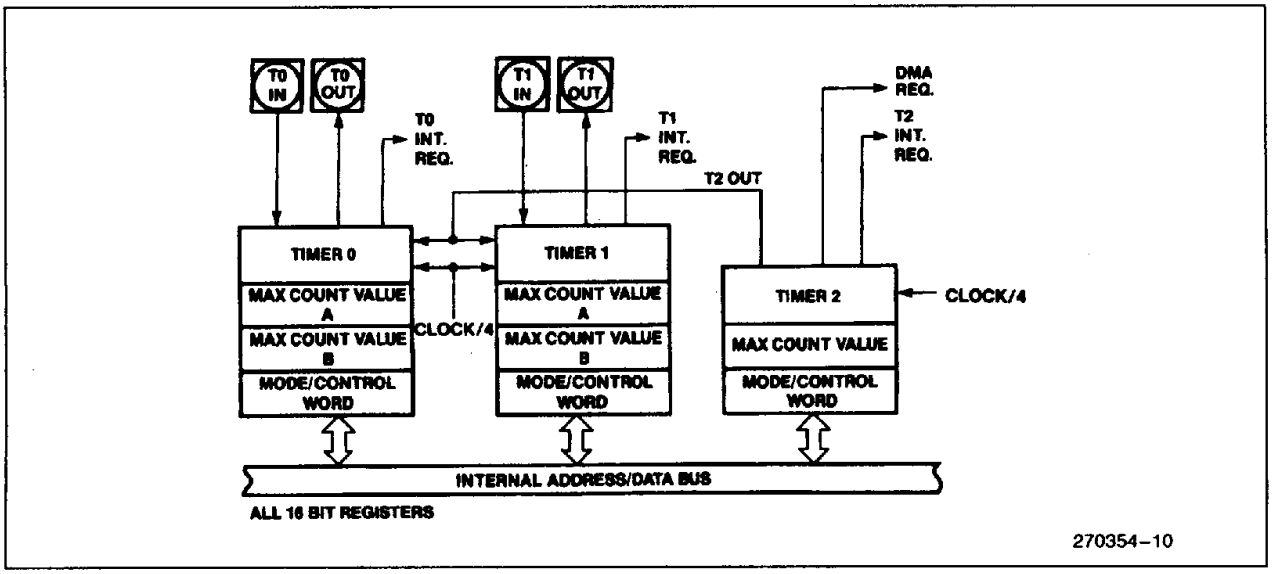



**Figure 19. Timer Block Diagram**

## Timer Operation

The timers are controlled by 11 16-bit registers in the peripheral control block. The configuration of these registers is shown in Table 15. 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 zero during that same clock, i.e., 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 single clock, 1 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 6 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. Any Read or Write access to the timers will add one wait state to the minimum four-clock bus cycle, however. This is needed to synchronize and coordinate the internal data flows between the internal timers and the internal bus.

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 20) allows the user to program the specific mode of operation or check the current programmed status for any of the three integrated timers.

**Table 15. 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 transistions during the time EN is zero will be ignored. If CONT is zero, the EN bit is automatically cleared upon maximum count.

**$\overline{\text{INH}}$:**

The inhibit bit allows for selective updating of the enable (EN) bit. If $\overline{\text{INH}}$ is a one 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 zero 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

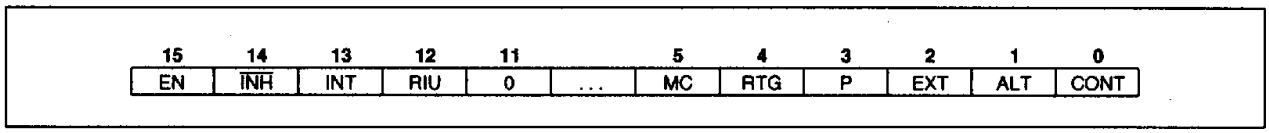



**Figure 20. Timer Mode/Control Register**

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 zero 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 zero.

**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.

Programmer 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 80C186 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 zero on the first clock, and then incrementing thereafter. Further transitions on the input pin will again reset the timer to zero, 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 zero, the timer will count at one-fourth the internal CPU clock rate. If the P bit is a one, 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 80C186 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 6 clocks. However, clock inputs may be pipelined as closely together as every 4 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 zero, the output pin will go LOW for one clock, the clock after the maximum count is reached. If ALT is one, 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

## 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 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 state of the timers will be as follows:

- 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 80C186 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 80C186 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 be interrupted by higher priority interrupts. A block diagram of the interrupt controller is shown in Figure 21.

The 80C186 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 80C186 on the second cycle. The capability to interface to external 82C59A programmable interrupt controllers is provided when the inputs are configured in Cascade Mode.

## Interrupt Controller Modes of Operation

The basic modes of operation of the interrupt controller in Master Mde 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 22. 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.

T-49-17-15

### Cascade Mode

The 80C186 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 23. 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 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 80C186 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.

24

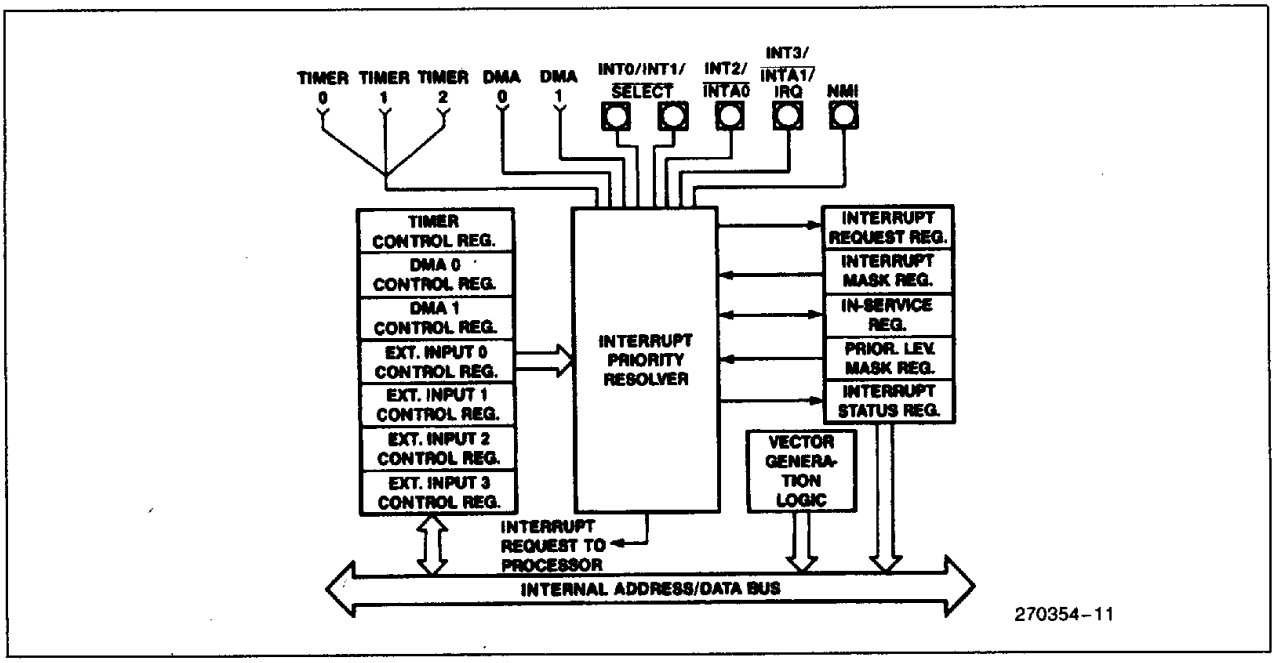



**Figure 21. Interrupt Controller Block Diagram**

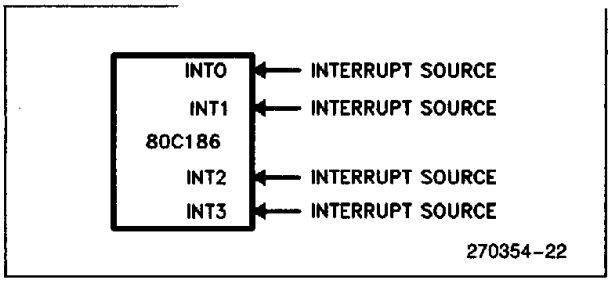



**Figure 22. Fully Nested (Direct) Mode Interrupt Controller Connections**

### 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 80C186 interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt will not be recognized by the 80C186 controller until the 80C186 in-service bit is reset. In Special Fully Nested Mode, the 80C186 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 80C186 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 80C186 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 32). Bit 15 in the poll word indicates to the processor that an interrupt of high enough priority is requesting service. Bits 0–4 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, i.e., not set the indicated in-service bit. The 80C186 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.

T-49-17-15

## 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 4).

If two requests with the same programmed priority level are pending at once, the priority ordering scheme shown in Table 4 is used. If the serviced interrupt routine reenables interrupts, other interrupt requests can be serviced.

### 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 nonspecific. The nonspecific 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

80C186 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 re-enable 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 Modes. The interrupt vectors generated are fixed and cannot be changed (see Table 4).

## Interrupt Controller Registers

The Interrupt Controller register model is shown in Figure 24. It contains 15 registers. All registers can both 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 25. 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 I0–I3 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.

T-49-17-15

### Interrupt Request Register

The internal interrupt sources have interrupt request bits inside the interrupt controller. The format of this register is shown in Figure 25. 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 25. A one in a bit position corre-

24

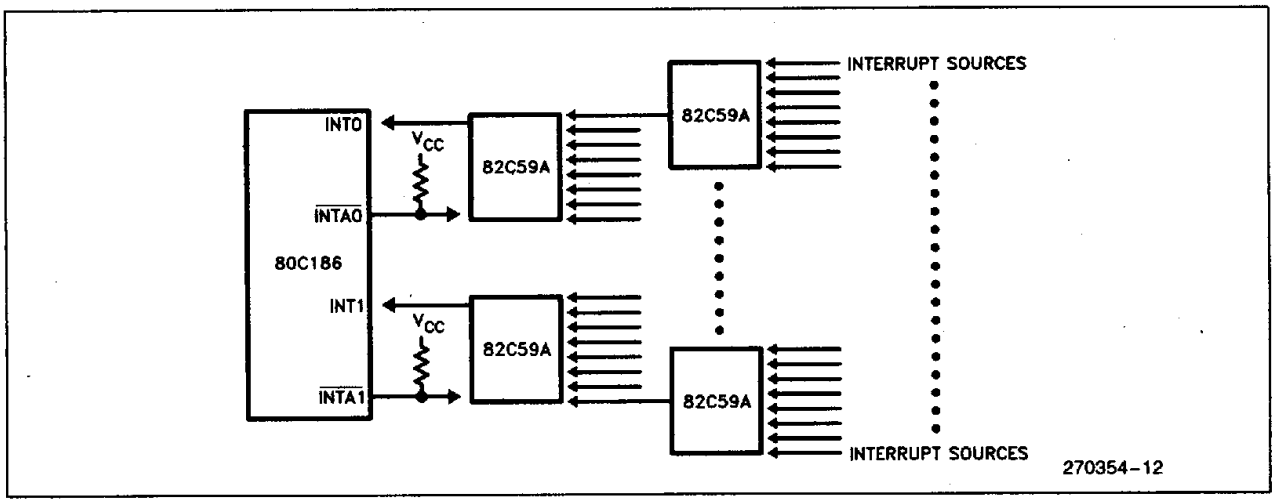



**Figure 23. Cascade and Special Fully Nested Mode Interrupt Controller Connections**

sponding to a particular source masks 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.

| | 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 |

**Figure 24. Interrupt Controller Registers (Master Mode)**

### Priority Mask Register

T-49-17-15

This register masks all interrupts below a particular interrupt priority level. The format of this register is shown in Figure 26. 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.

### Interrupt Status Register

This register contains general interrupt controller status information. The format of this register is shown in Figure 27. 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 request. Note that setting any one of these three bits initiates an interrupt request to the interrupt controller.

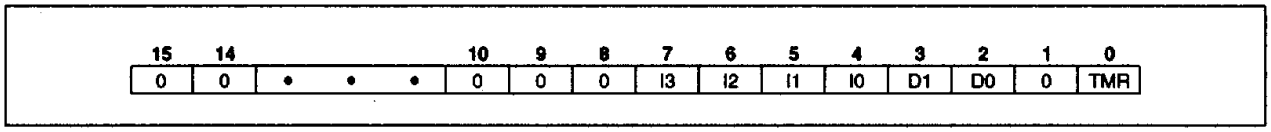



**Figure 25. In-Service, Interrupt Request, and Mask Register Formats**

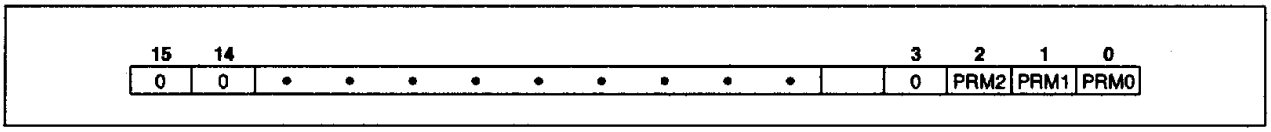



**Figure 26. Priority Mask Register Format**

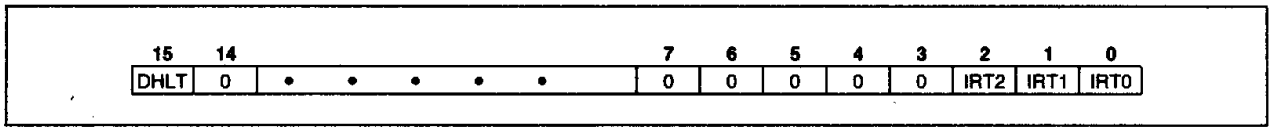



**Figure 27. Interrupt Status Register Format (Master Mode)**

### Timer, DMA 0, 1; Control Register

These registers are the control words for all the internal interrupt sources. The format for these registers is shown in Figure 28. The three bit positions PR0, PR1, and PR2 represent the programmable priority level of the interrupt source. The MSK bit inhibits its interrupt requests from the interrupt source. The MSK bits in the individual control registers are the exact same bits as are in the Mask Register; modifying them in the individual control registers will also modify them in the Mask Register, and vice versa.

### INT0-INT3 Control Registers

These registers are the control words for the four external input pins. Figure 29 shows the format of the INT0 and INT1 Control registers; Figure 30 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:

PRO-2: 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 31. It initiates an EOI command when written to by the 80C186 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 4. 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.



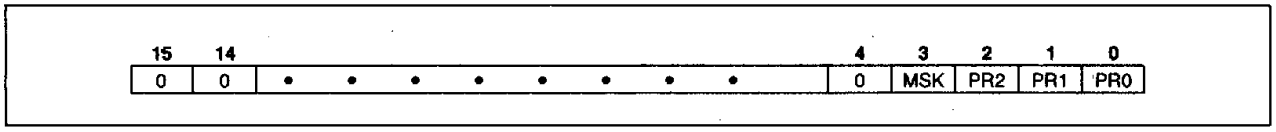



**Figure 28. Timer/DMA Control Registers Formats**

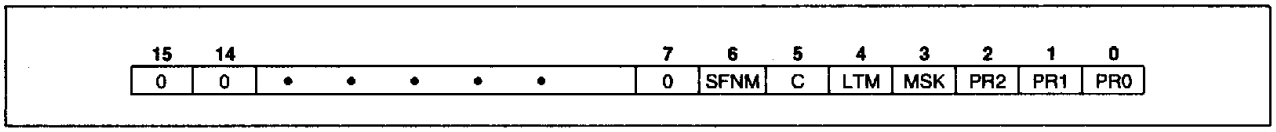



**Figure 29. INT0/INT1 Control Register Formats**

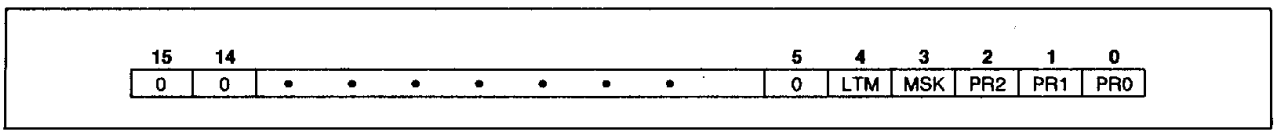



**Figure 30. INT2/INT3 Control Register Formats**

**Poll and Poll Status Registers**

These registers contain polling information. The format of these registers is shown in Figure 32. 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 80C186 interrupt controller will be used as a slave controller to an external master interrupt controller. The internal 80C186 resources will be monitored by the internal interrupt controller, while the external controller functions as the system master interrupt controller.

Upon reset, the 80C186 will be in master mode. To provide for slave mode operation bit 14 of the relocation register should be set.

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 80C186 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 80C186 with respect to an external 82C59A master is shown in Figure 33. The INT0 (Pin 45) input is used as the 80C186 CPU interrupt input. IRQ (Pin 41) functions as an output to send the 80C186 slave-interrupt-request to one of the 8 master-PIC-inputs.

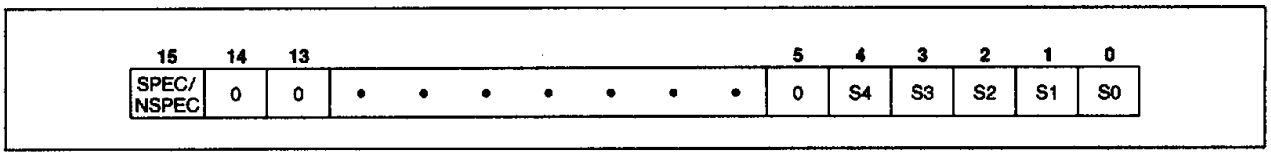



**Figure 31. EOI Register Format**

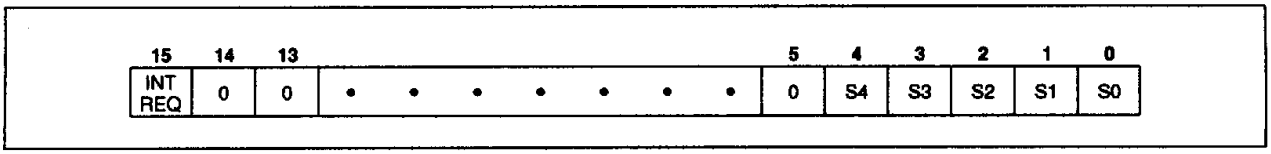



**Figure 32. Poll and Poll Status Register Format**

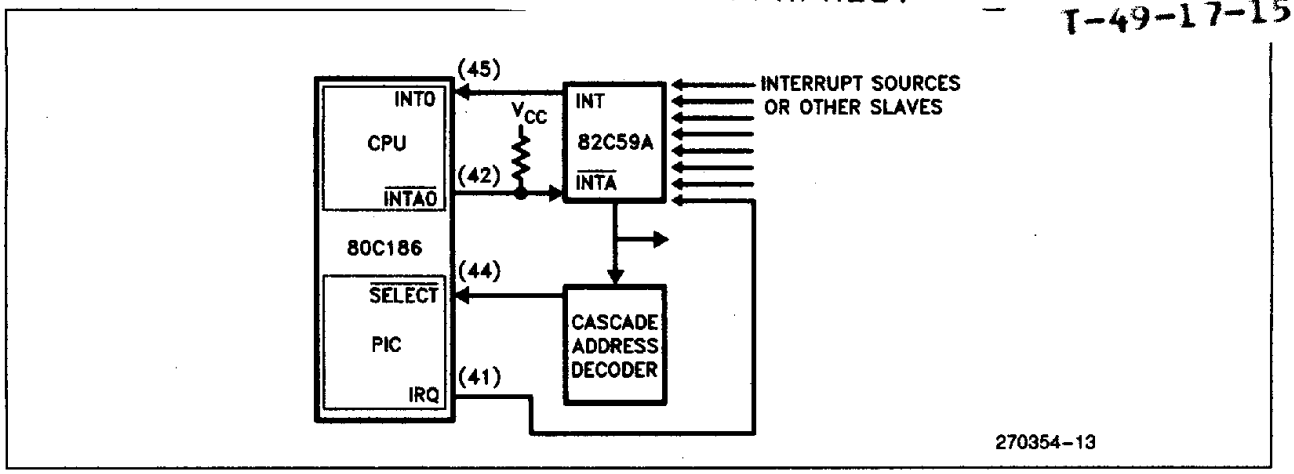



**Figure 33. Slave Mode Interrupt Controller Connections**

Correct master-slave interface requires decoding of the slave addresses (CAS0-2). Slave 82C59As do this internally. Because of pin limitations, the 80C186 slave address will have to be decoded externally. $\overline{\text{SELECT}}$ (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.

$\overline{\text{INTA0}}$ (Pin 42) is used as an acknowledge output, suitable to drive the $\overline{\text{INTA}}$ input of an 82C59A.

## 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 to use 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 34. 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 34 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 35. It initiates an EOI command when written by the 80C186 CPU.

The bits in the EOI register are encoded as follows:

$VT_X$: Three least-significant vector type bits corresponding to the source for which the IS bit is to be reset. Figure 34 indicates these bits.

24

### In-Service Register

This register can be read from or written into. It contains the in-service bit for each of the internal interrupt sources. The format for this register is shown in Figure 36. 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 36. The interrupt request bits are set when a request arrives from an internal source, and are reset when the processor acknowledges the request. 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 36. 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, i.e., 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 37. 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.

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 |
| DMA1 CONTROL REGISTER (VECTOR TYPE XXXXX011) | 36H |
| DMA0 CONTROL REGISTER (VECTOR TYPE XXXXX010) | 34H |
| TIMER0 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 |

**Figure 34. Interrupt Controller Registers (Slave Mode)**

| 15 | 14 | 13 | • | • | • | • | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | • | • | • | • | 0 | 0 | 0 | 0 | 0 | 0 | VT2 | VT1 | VT0 |

**Figure 35. 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 |

**Figure 36. In-Service, Interrupt Request, and Mask Register Format**

### Interrupt Vector Register

This register provides the upper five bits of the interrupt vector address. The format of this register is shown in Figure 38. 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.

### Interrupt Status Register

This register is defined as in Master Mode except that DHLT is not implemented (see Figure 27).

## 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.

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

**Figure 37. 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 |

**Figure 38. 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 |

**Figure 39. Priority Level Mask Register**

## Enhanced Mode Operation

In Compatible Mode the 80C186 operates with all the features of the NMOS 80186, 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 80C186 will operate with Power-Save, DRAM refresh, and numerics coprocessor support in addition to all the Compatible Mode features.

## Entering Enhanced Mode

If connected to a numerics coprocessor, this mode will be invoked automatically. Without an NPX, this mode can be entered by tying the RESET output signal from the 80C186 to the TEST/BUSY 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 80C186 will reconfigure the ALE and $\overline{WR}$ pins to be QS0 and QS1 respectively. This mode is available on the 80C186 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 80C186 will deactivate the HLDA pin in order to perform a refresh cycle. The circuit external to the 80C186 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 zeros.

## DRAM Refresh Addresses

The address generated during a refresh cycle is determined by the contents of the MDRAM register (see Figure 40) and the contents of a 9-bit counter. Figure 41 illustrates the origin of each bit.

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MDRAM: Offset E0H | M6 | M5 | M4 | M3 | M2 | M1 | M0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0–8: Reserved, read back as 0.

Bits 9–15: M0–M6, are address bits A13–A19 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 on RESET.

**Figure 40. 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 41. Addresses Generated by RCU**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CDRAM: Offset E2H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | C8 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |

Bits 0–8: C0–C8, clock divisor register, holds the number of CLKOUT cycles between each refresh request.

Bits 9–15: Reserved, read back as 0.

**Figure 42. Clock Pre-Scaler Register**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EDRAM: Offset E4H | E | 0 | 0 | 0 | 0 | 0 | 0 | T8 | T7 | T6 | T5 | T4 | T3 | T2 | T1 | T0 |

Bits 0–8: T0–T8, refresh clock counter outputs. Read only.

Bits 9–14: Reserved, read back as 0.

Bit 15: Enable RCU, set to 0 on RESET.

**Figure 43. Enable RCU Register**

## Refresh Control Unit Programming and Operation

After programming the MDRAM and the CDRAM registers (Figures 40 and 42), the RCU is enabled by setting the "E" bit in the EDRAM register (Figure 43). The clock counter (T0–T8 of EDRAM) will be loaded from C0–C8 of CDRAM during $T_3$ 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 80C186, 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 two-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 re-programmed 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 80C186 will begin to be divided during the $T_3$ state of the instruction cycle that sets the enable bit. Clearing the enable bit will restore full speed in the $T_3$ state of that instruction.

At no time should the internal clock frequency be allowed to fall below 0.5 MHz. This is the minimum operational frequency of the 80C186. For example, an 80C186 running with a 12 MHz crystal (6 MHz CLOCKOUT) should never have a clock divisor greater than eight.

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PDCON: Offset F0H | E | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | F1 | F0 |

Bits 0–1: Clock Divisor Select

| F1 | F0 | Division Factor |
|---|---|---|
| 0 | 0 | divide by 1 |
| 0 | 1 | divide by 4 |
| 1 | 0 | divide by 8 |
| 1 | 1 | divide by 16 |

Bits 2–14: Reserved, read back as zero.

Bit 15: Enable Power Save Mode. Set to zero on RESET.

**Figure 44. Power-Save Control Register**

## Interface for 80C187 Numeric Processor Extension

In Enhanced Mode, three of the mid-range memory chip selects are redefined according to Table 16 for use with the 80C187. The fourth chip select, $\overline{MCS2}$ functions as in compatible mode, and may be programmed for activity with ready logic and wait states accordingly. As in compatible mode, $\overline{MCS2}$ will function for one-fourth a programmed block size.

**Table 16. $\overline{MCS}$ Assignments**

| Compatible Mode | Enhanced Mode | |
|---|---|---|
| $\overline{MCS0}$ | PEREQ | Processor Extension Request |
| $\overline{MCS1}$ | $\overline{ERROR}$ | NPX Error |
| $\overline{MCS2}$ | $\overline{MCS2}$ | Mid-Range Chip Select |
| $\overline{MCS3}$ | $\overline{NPS}$ | Numeric Processor Select |

Four port addresses are assigned to the 80C186/80C187 interface for 16-bit reads and writes. Table 17 shows the port definitions. These ports are not accessible by using the 80C186 I/O instructions. However, numerics operations will cause a $\overline{PCS}$ line to be activated if it is properly programmed for this I/O range.

**Table 17. Numerics Coprocessor I/O Port Assignments**

| I/O Address | Read Definition | Write Definition |
|---|---|---|
| 00F8H | Status/Control | Opcode |
| 00FAH | Data | Data |
| 00FCH | reserved | CS:IP, DS:EA |
| 00FEH | Opcode Status | reserved |

## ONCE™ Test Mode

To facilitate testing and inspection of devices when fixed into a target system, the 80C186 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 80C186 will put all pins in the high-impedance state until RESET.

The ONCE mode is selected by tying 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 the $\overline{LCS}$ pins have weak internal pull-up resistors similar to the $\overline{RD}$ and $\overline{TEST}$/BUSY pins to guarantee normal operation.

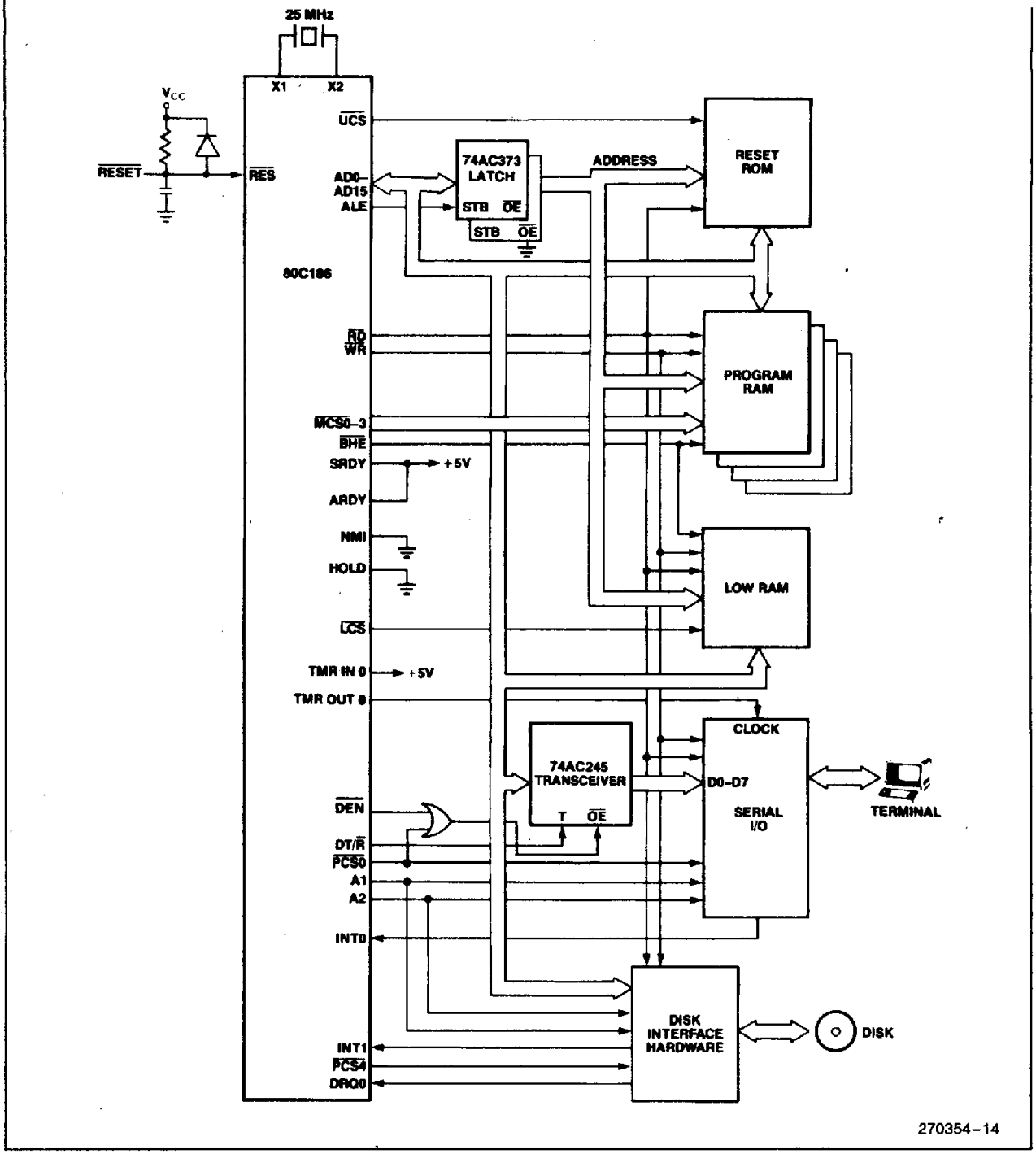



**Figure 45. Typical 80C186 Computer**

## ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias ....0°C to +70°C

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

Voltage on Any Pin with Respect to Ground .....−1.0V to +7.0V

Package Power Dissipation .....1W
Not to exceed the maximum allowable die temperature based on thermal resistance of the package.

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

*NOTICE: This data sheet is only valid for devices indicated in the Specification Level Markings section. Specifications contained in the following tables are subject to change.*

## D.C. CHARACTERISTICS

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

| Symbol | Parameter | Min | 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 X1, $\overline{RES}$, 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 (X1) | 3.9 | $V_{CC}$ + 0.5 | V | |
| $V_{OL}$ | Output Low Voltage | | 0.45 | V | $I_{OL}$ = 2.5 mA (S0, 1, 2)<br>$I_{OL}$ = 2.0 mA (others) |
| $V_{OH}$ | Output High Voltage | 2.4 | $V_{CC}$ | V | $I_{OH}$ = −2.4 mA @ 2.4V (4) |
| | | $V_{CC}$ − 0.5 | $V_{CC}$ | V | $I_{OH}$ = −200 µA @ $V_{CC}$ −0.5(4) |
| $I_{CC}$ | Power Supply Current | | 150 | mA | @ 16 MHz, 0°C<br>$V_{CC}$ = 5.25V (3) |
| | | | 120 | mA | @ 12.5 MHz, 0°C<br>$V_{CC}$ = 5.5V (3) |
| | | | 100 | mA | @ 10 MHz, 0°C<br>$V_{CC}$ = 5.5V (3) |
| $I_{LI}$ | Input Leakage Current | | ±10 | µA | @ 0.5 MHz,<br>0.45V ≤ $V_{IN}$ ≤ $V_{CC}$ |
| $I_{LO}$ | Output Leakage Current | | ±10 | µA | @ 0.5 MHz,<br>0.45V ≤ $V_{OUT}$ ≤ $V_{CC}$(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 + 5.0V or 0.45V. 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. $\overline{RD}$/QSMD, $\overline{UCS}$, $\overline{LCS}$, $\overline{MCS0}$/PEREQ, $\overline{MCS1}$/$\overline{ERROR}$, and $\overline{TEST}$/BUSY pins have internal pullup devices. Loading some of these pins above $I_{OH}$ = −200 µA can cause the 80C186 to go into alternative modes of operation. See the section on Local Bus Controller and Reset for details.

## 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 $I_{CC}$ = 8.4 mA × freq. (MHz) + 15 mA.

Typical current is given by $I_{CC}$ (typical) = 6.4 mA × freq. (MHz) + 4.0 mA. "Typicals" are based on a limited number of samples taken from early manufacturing lots measured at $V_{CC}$ = 5V and room temperature. "Typicals" are not guaranteed.

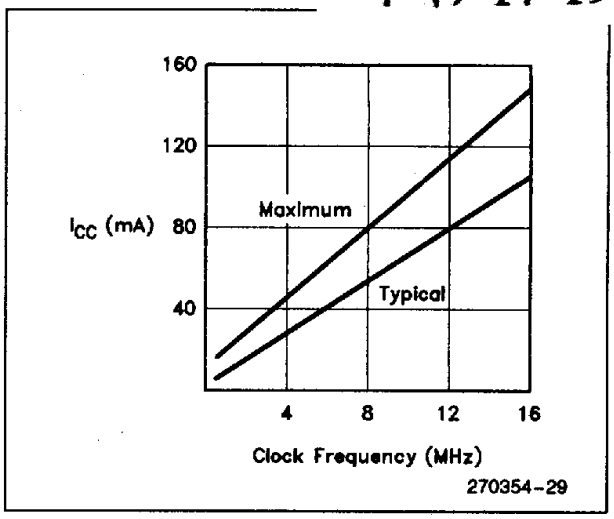



**Figure 46. $I_{CC}$ vs Frequency**

## A. C. CHARACTERISTICS

### MAJOR CYCLE TIMINGS (READ CYCLE)

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

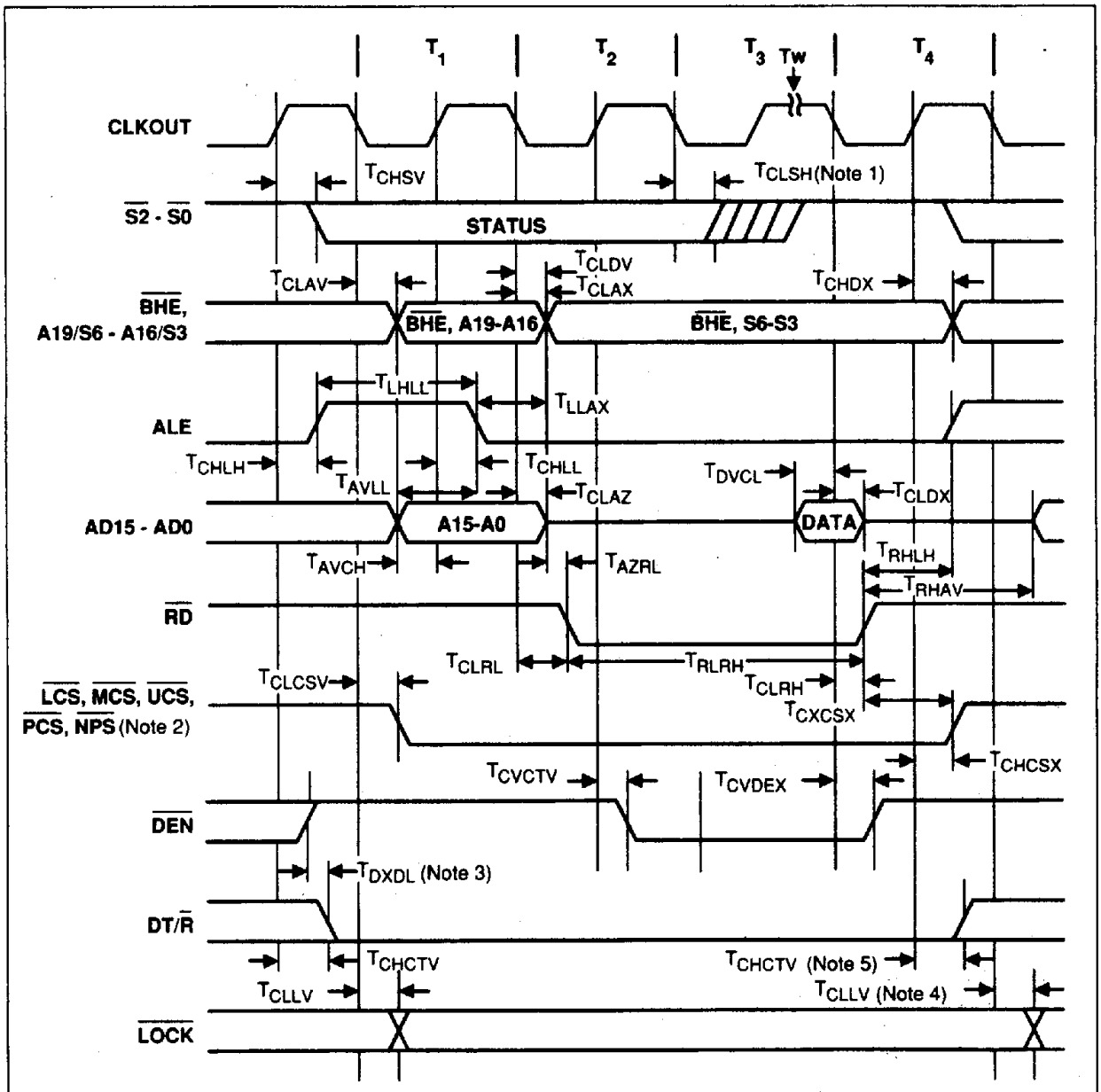
All timings are measured at 1.5V and 100 pF loading on CLKOUT unless otherwise noted.
All output test conditions are with $C_L$ = 50-200 pF (10 MHz) and $C_L$ = 50-100 pF (12.5-16 MHz).
For A.C. tests, input $V_{IL}$ = 0.45V and $V_{IH}$ = 2.4V except at X1 where $V_{IH} = V_{CC} - 0.5V$.

| Symbol | Parameter | 80C186 | | 80C186-12 | | 80C186-16 | | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| **80C186 GENERAL TIMING REQUIREMENTS (Listed More Than Once)** | | | | | | | | | |
| $T_{DVCL}$ | Data in Setup (A/D) | 15 | | 15 | | 15 | | ns | |
| $T_{CLDX}$ | Data in Hold (A/D) | 3 | | 3 | | 3 | | ns | |
| **80C186 GENERAL TIMING RESPONSES (Listed More Than Once)** | | | | | | | | | |
| $T_{CHSV}$ | Status Active Delay | 5 | 45 | 5 | 35 | 5 | 31 | ns | |
| $T_{CLSH}$ | Status Inactive Delay | 5 | 46 | 5 | 35 | 5 | 30 | ns | |
| $T_{CLAV}$ | Address Valid Delay | 5 | 44 | 5 | 36 | 5 | 33 | ns | |
| $T_{CLAX}$ | Address Hold | 0 | | 0 | | 0 | | ns | |
| $T_{CLDV}$ | Data Valid Delay | 5 | 40 | 5 | 36 | 5 | 33 | ns | |
| $T_{CHDX}$ | Status Hold Time | 10 | | 10 | | 10 | | ns | |
| $T_{CHLH}$ | ALE Active Delay | | 30 | | 25 | | 20 | ns | |
| $T_{LHLL}$ | ALE Width | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | ns | |
| $T_{CHLL}$ | ALE Inactive Delay | | 30 | | 25 | | 20 | ns | |
| $T_{AVLL}$ | Address Valid to ALE Low | $T_{CLCH}$ - 18 | | $T_{CLCH}$ - 15 | | $T_{CLCH}$ - 15 | | ns | Equal Loading |
| $T_{LLAX}$ | Address Hold from ALE Inactive | $T_{CHCL}$ - 15 | | $T_{CHCL}$ - 15 | | $T_{CHCL}$ - 15 | | ns | Equal Loading |
| $T_{AVCH}$ | Address Valid to Clock High | 0 | | 0 | | 0 | | ns | |
| $T_{CLAZ}$ | Address Float Delay | $T_{CLAX}$ | 30 | $T_{CLAX}$ | 25 | $T_{CLAX}$ | 20 | ns | |
| $T_{CLCSV}$ | Chip-Select Active Delay | 3 | 42 | 3 | 33 | 3 | 30 | ns | |
| $T_{CXCSX}$ | Chip-Select Hold from Command Inactive | $T_{CLCH}$ - 10 | | $T_{CLCH}$ - 10 | | $T_{CLCH}$ - 10 | | ns | Equal Loading |
| $T_{CHCSX}$ | Chip-Select Inactive Delay | 5 | 35 | 5 | 30 | 5 | 25 | ns | |
| $T_{DXDL}$ | $\overline{DEN}$ Inactive to DT/$\overline{R}$ Low | 0 | | 0 | | 0 | | ns | Equal Loading |
| $T_{CVCTV}$ | Control Active Delay 1 | 3 | 44 | 3 | 37 | 3 | 31 | ns | |
| $T_{CVDEX}$ | $\overline{DEN}$ Inctive Delay | 5 | 44 | 5 | 37 | 5 | 31 | ns | |
| $T_{CHCTV}$ | Control Active Delay 2 | 5 | 44 | 5 | 37 | 5 | 31 | ns | |
| $T_{CLLV}$ | $\overline{LOCK}$ Valid/Invalid Delay | 3 | 40 | 3 | 37 | 3 | 35 | ns | |
| **80C186 TIMING RESPONSES (Read Cycle)** | | | | | | | | | |
| $T_{AZRL}$ | Address Float to $\overline{RD}$ Active | 0 | | 0 | | 0 | | ns | |
| $T_{CLRL}$ | $\overline{RD}$ Active Delay | 5 | 44 | 5 | 37 | 5 | 31 | ns | |
| $T_{RLRH}$ | $\overline{RD}$ Pulse Width | $2T_{CLCL}$ - 30 | | $2T_{CLCL}$ - 25 | | $2T_{CLCL}$ - 25 | | ns | |
| $T_{CLRH}$ | $\overline{RD}$ Inactive Delay | 5 | 44 | 5 | 37 | 5 | 31 | ns | |
| $T_{RHLH}$ | $\overline{RD}$ Inactive to ALE High | $T_{CLCH}$ - 14 | | $T_{CLCH}$ - 14 | | $T_{CLCH}$ - 14 | | ns | Equal Loading |
| $T_{RHAV}$ | $\overline{RD}$ Inactive to Address Active | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | ns | Equal Loading |

270354-34

## A. C. CHARACTERISTICS

### READ CYCLE WAVEFORMS




**NOTES:**

1. Status inactive in state preceding $T_4$.
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.

270354-35

24

## A. C. CHARACTERISTICS

T-49-17-15

### MAJOR CYCLE TIMINGS (WRITE CYCLE)

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

All timings are measured at 1.5V 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 (12.5-16 MHz).
For A.C. tests, input $V_{IL}$ = 0.45V and $V_{IH}$ = 2.4V except at X1 where $V_{IH} = V_{CC} - 0.5V$.

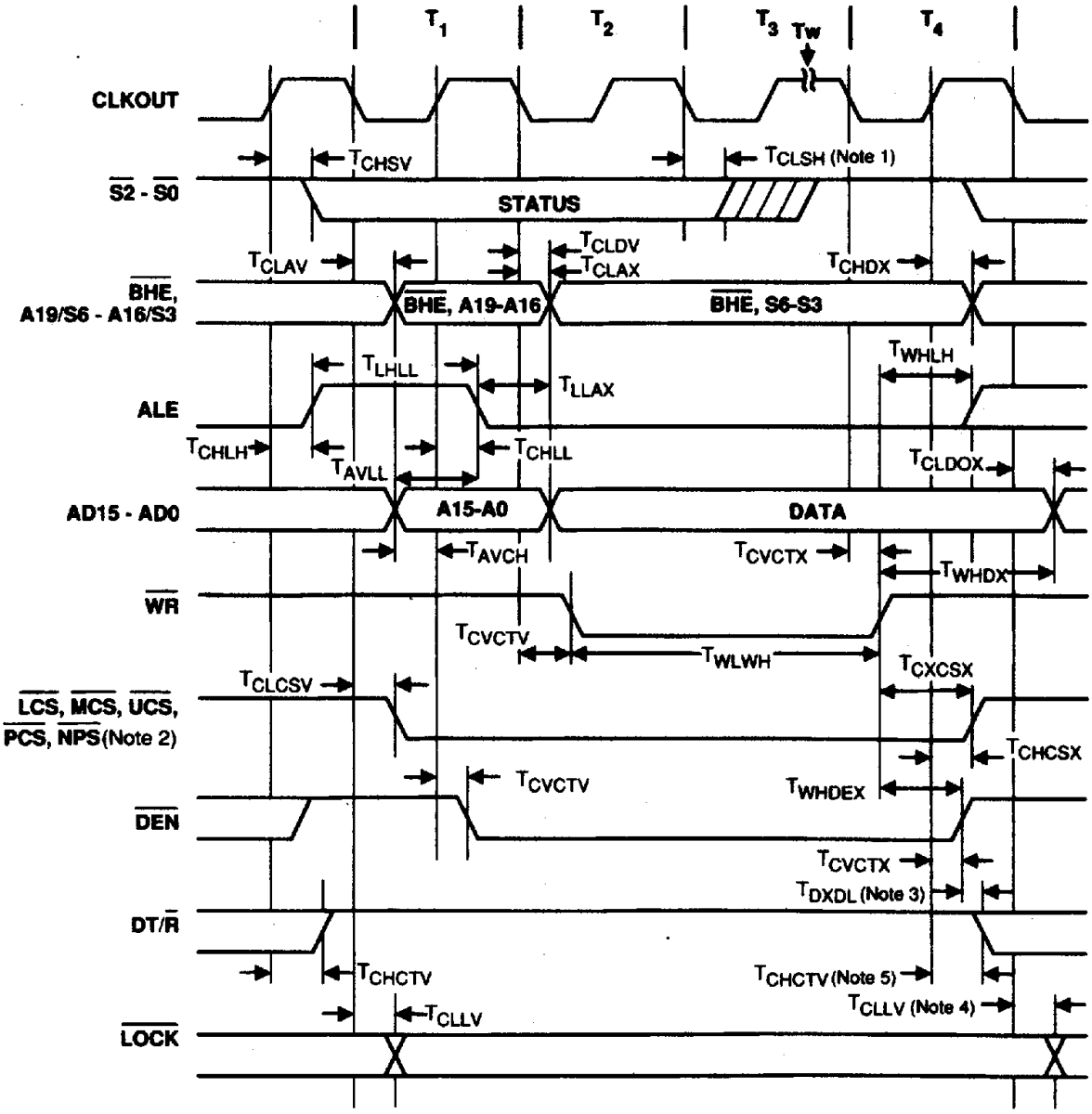
| Symbol | Parameter | 80C186 | | 80C186-12 | | 80C186-16 | | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| **80C186 GENERAL TIMING RESPONSES (Listed More Than Once)** | | | | | | | | | |
| $T_{CHSV}$ | Status Active Delay | 5 | 45 | 5 | 35 | 5 | 31 | ns | |
| $T_{CLSH}$ | Status Inactive Delay | 5 | 46 | 5 | 35 | 5 | 30 | ns | |
| $T_{CLAV}$ | Address Valid Delay | 5 | 44 | 5 | 36 | 5 | 33 | ns | |
| $T_{CLAX}$ | Address Hold | 0 | | 0 | | 0 | | ns | |
| $T_{CLDV}$ | Data Valid Delay | 5 | 40 | 5 | 36 | 5 | 33 | ns | |
| $T_{CHDX}$ | Status Hold Time | 10 | | 10 | | 10 | | ns | |
| $T_{CHLH}$ | ALE Active Delay | | 30 | | 25 | | 20 | ns | |
| $T_{LHLL}$ | ALE Width | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | ns | |
| $T_{CHLL}$ | ALE Inactive Delay | | 30 | | 25 | | 20 | ns | |
| $T_{AVLL}$ | Address Valid to ALE Low | $T_{CLCH}$ - 18 | | $T_{CLCH}$ - 15 | | $T_{CLCH}$ - 15 | | ns | Equal Loading |
| $T_{LLAX}$ | Address Hold from ALE Inactive | $T_{CHCL}$ - 15 | | $T_{CHCL}$ - 15 | | $T_{CHCL}$ - 15 | | ns | Equal Loading |
| $T_{AVCH}$ | Address Valid to Clock High | 0 | | 0 | | 0 | | ns | |
| $T_{CLDOX}$ | Data Hold Time | 3 | | 3 | | 3 | | ns | |
| $T_{CVCTV}$ | Control Active Delay 1 | 3 | 44 | 3 | 37 | 3 | 31 | ns | |
| $T_{CVCTX}$ | Control Inactive Delay | 3 | 44 | 3 | 37 | 3 | 31 | ns | |
| $T_{CLCSV}$ | Chip-Select Active Delay | 3 | 42 | 3 | 33 | 3 | 30 | ns | |
| $T_{CXCSX}$ | Chip-Select Hold from Command Inactive | $T_{CLCH}$ - 10 | | $T_{CLCH}$ - 10 | | $T_{CLCH}$ - 10 | | ns | Equal Loading |
| $T_{CHCSX}$ | Chip-Select Inactive Delay | 5 | 35 | 5 | 30 | 5 | 25 | ns | |
| $T_{DXDL}$ | $\overline{DEN}$ Inactive to DT/$\overline{R}$ Low | 0 | | 0 | | 0 | | ns | Equal Loading |
| $T_{CLLV}$ | $\overline{LOCK}$ Valid/Invalid Delay | 3 | 40 | 3 | 37 | 3 | 35 | ns | |
| **80C186 TIMING RESPONSES (Write Cycle)** | | | | | | | | | |
| $T_{WLWH}$ | $\overline{WR}$ Pulse Width | $2T_{CLCL}$ - 30 | | $2T_{CLCL}$ - 25 | | $2T_{CLCL}$ - 25 | | ns | |
| $T_{WHLH}$ | $\overline{WR}$ Inactive to ALE High | $T_{CLCH}$ - 14 | | $T_{CLCH}$ - 14 | | $T_{CLCH}$ - 14 | | ns | Equal Loading |
| $T_{WHDX}$ | Data Hold After $\overline{WR}$ | $T_{CLCL}$ - 34 | | $T_{CLCL}$ - 20 | | $T_{CLCL}$ - 20 | | ns | Equal Loading |
| $T_{WHDEX}$ | $\overline{WR}$ Inactive to $\overline{DEN}$ Inactive | $T_{CLCH}$ - 10 | | $T_{CLCH}$ - 10 | | $T_{CLCH}$ - 10 | | ns | Equal Loading |

270354-36

## A. C. CHARACTERISTICS

### WRITE CYCLE WAVEFORMS

T-49-17-15




**NOTES:**

1. Status inactive in state preceding $T_4$.
2. If latched $A_1$ and $A_2$ are selected instead of $\overline{PCS5}$ and $\overline{PCS6}$, only $T_{CLCSV}$ is applicable.
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.

270354-37

24

## A. C. CHARACTERISTICS

T-49-17-15

### MAJOR CYCLE TIMINGS (INTERRUPT ACKNOWLEDGE CYCLE)

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

All timings are measured at 1.5V 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 (12.5-16 MHz).
For A.C. tests, input $V_{IL}$ = 0.45V and $V_{IH}$ = 2.4V except at X1 where $V_{IH} = V_{CC} - 0.5V$.

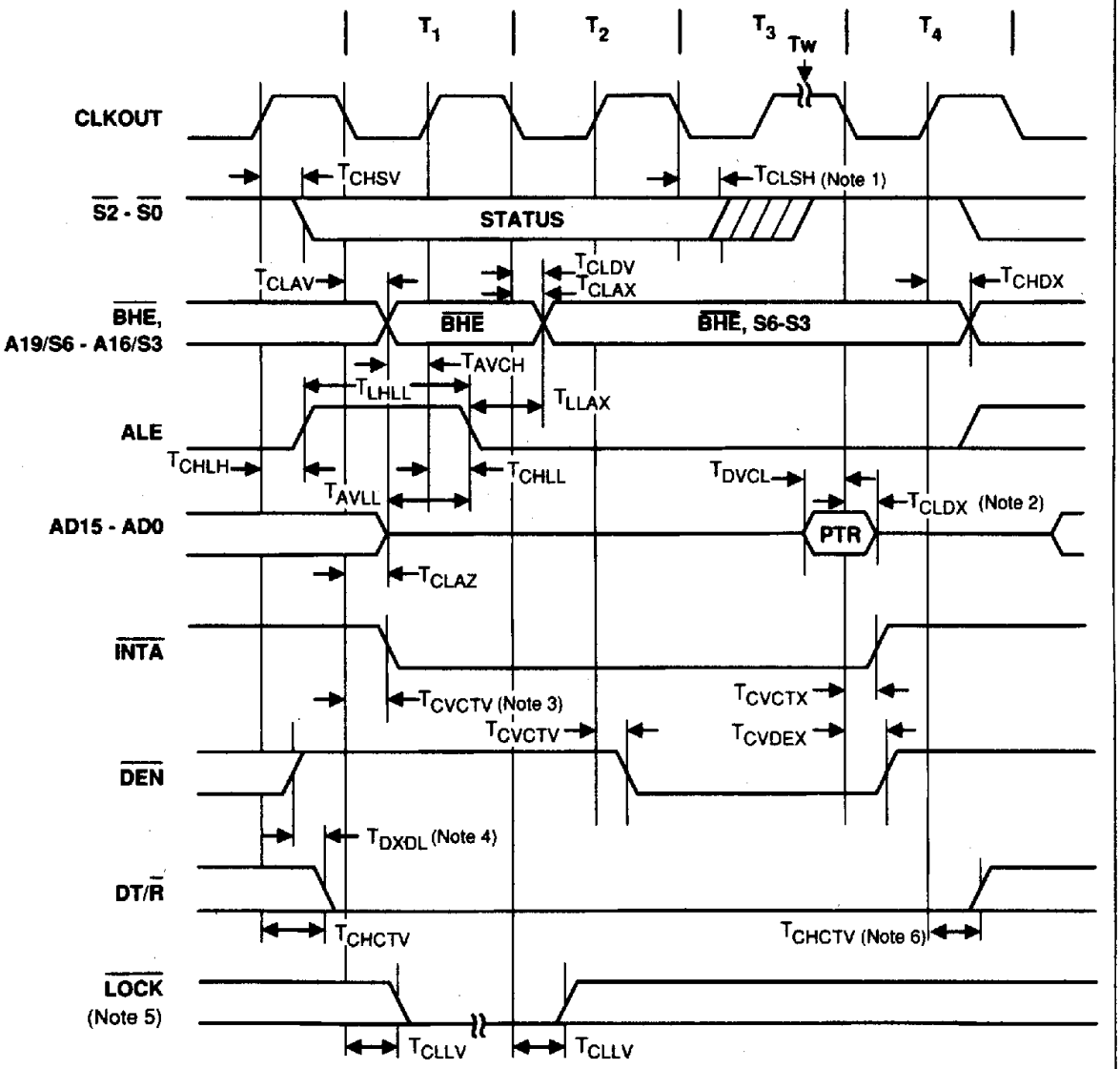
| Symbol | Parameter | 80C186 | | 80C186-12 | | 80C186-16 | | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| **80C186 GENERAL TIMING REQUIREMENTS (Listed More Than Once)** | | | | | | | | | |
| $T_{DVCL}$ | Data in Setup (A/D) | 15 | | 15 | | 15 | | ns | |
| TCLDX | Data in Hold (A/D) | 3 | | 3 | | 3 | | ns | |
| **80C186 GENERAL TIMING RESPONSES (Listed More Than Once)** | | | | | | | | | |
| $T_{CHSV}$ | Status Active Delay | 5 | 45 | 5 | 35 | 5 | 31 | ns | |
| $T_{CLSH}$ | Status Inactive Delay | 5 | 46 | 5 | 35 | 5 | 30 | ns | |
| $T_{CLAV}$ | Address Valid Delay | 5 | 44 | 5 | 36 | 5 | 33 | ns | |
| $T_{AVCH}$ | Address Valid to Clock High | 0 | | 0 | | 0 | | ns | |
| $T_{CLAX}$ | Address Hold | 0 | | 0 | | 0 | | ns | |
| $T_{CLDV}$ | Data Valid Delay | 5 | 40 | 5 | 36 | 5 | 33 | ns | |
| $T_{CHDX}$ | Status Hold Time | 10 | | 10 | | 10 | | ns | |
| $T_{CHLH}$ | ALE Active Delay | | 30 | | 25 | | 20 | ns | |
| $T_{LHLL}$ | ALE Width | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | ns | |
| $T_{CHLL}$ | ALE Inactive Delay | | 30 | | 25 | | 20 | ns | |
| $T_{AVLL}$ | Address Valid to ALE Low | $T_{CLCH}$ - 18 | | $T_{CLCH}$ - 15 | | $T_{CLCH}$ - 15 | | ns | Equal Loading |
| $T_{LLAX}$ | Address Hold to ALE Inactive | $T_{CHCL}$ - 15 | | $T_{CHCL}$ - 15 | | $T_{CHCL}$ - 15 | | ns | Equal Loading |
| $T_{CLAZ}$ | Address Float Delay | $T_{CLAX}$ | 30 | $T_{CLAX}$ | 25 | $T_{CLAX}$ | 20 | ns | |
| $T_{CVCTV}$ | Control Active Delay 1 | 3 | 44 | 3 | 37 | 3 | 31 | ns | |
| $T_{CVCTX}$ | Control Inactive Delay | 3 | 44 | 3 | 37 | 3 | 31 | ns | |
| $T_{DXDL}$ | $\overline{DEN}$ Inactive to DT/$\overline{R}$ Low | 0 | | 0 | | 0 | | ns | Equal Loading |
| $T_{CHCTV}$ | Control Active Delay 2 | 5 | 44 | 5 | 37 | 5 | 31 | ns | |
| $T_{CVDEX}$ | $\overline{DEN}$ Inctive Delay (Non-Write Cycles) | 5 | 44 | 5 | 37 | 5 | 31 | ns | |
| $T_{CLLV}$ | $\overline{LOCK}$ Valid/Invalid Delay | 3 | 40 | 3 | 37 | 3 | 35 | ns | |

270354-38

## A. C. CHARACTERISTICS

T-49-17-15

### INTERRUPT ACKNOWLEDGE CYCLE WAVEFORMS




**NOTES:**

1. Status inactive in state preceding $T_4$.
2. The data hold time lasts only until $\overline{INTA}$ goes inactive, even if the $\overline{INTA}$ transition occurs prior to $T_{CLDX}$ (min).
3. $\overline{INTA}$ occurs one clock later in Slave Mode.
4. For write cycle followed by interrupt acknowledge cycle.
5. $\overline{LOCK}$ is active upon $T_1$ of the first interrupt acknowledge cycle and inactive upon $T_2$ of the second interrupt acknowledge cycle.

270354-39

24

## A. C. CHARACTERISTICS



### SOFTWARE HALT CYCLE TIMINGS

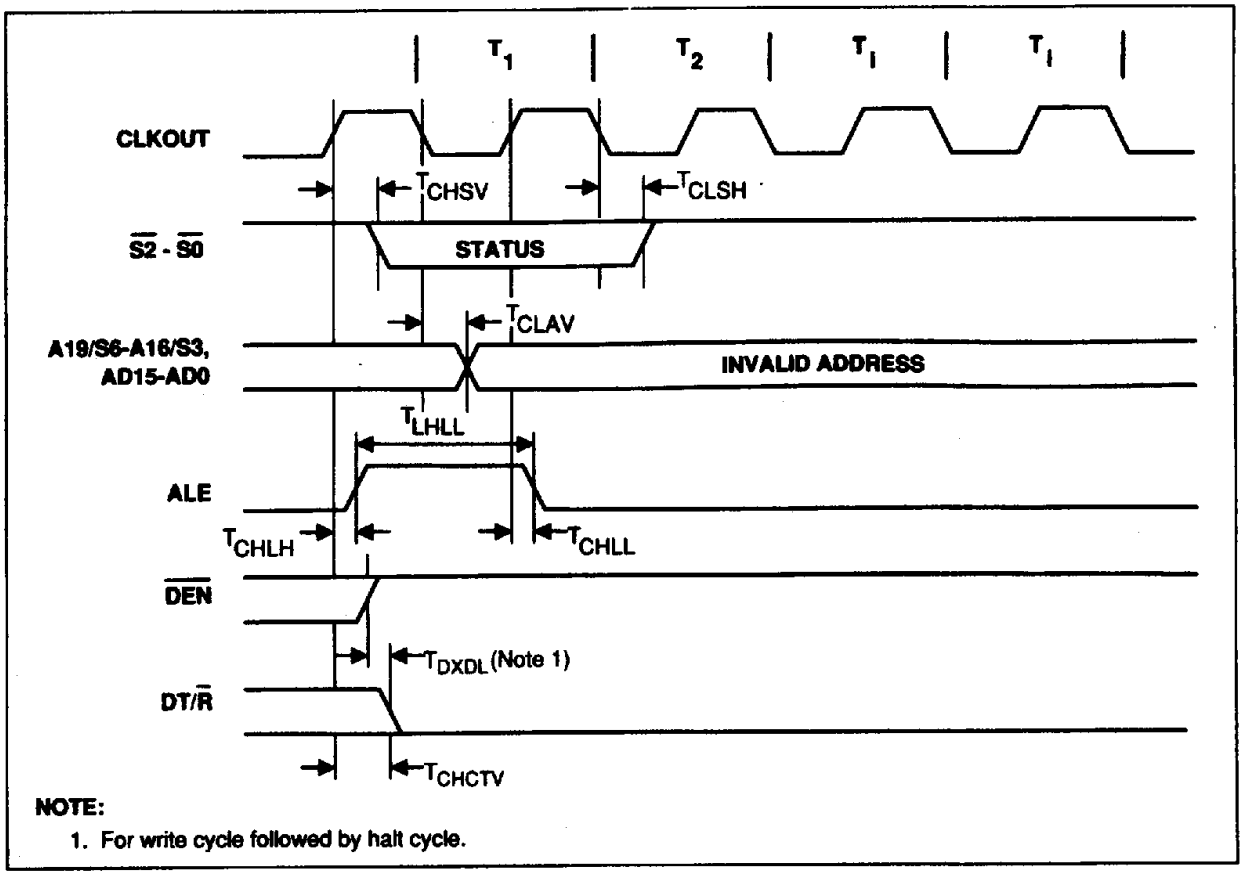


$T_A = 0^\circ C$ to $+70^\circ C$, $V_{CC}$ = 5V ± 10% except $V_{CC}$ = 5V ± 5% at f > 12.5 MHz

All timings are measured at 1.5V 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 (12.5-16 MHz).
For A.C. tests, input $V_{IL}$ = 0.45V and $V_{IH}$ = 2.4V except at X1 where $V_{IH} = V_{CC} - 0.5V$.

| Symbol | Parameter | 80C186 | | 80C186-12 | | 80C186-16 | | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| **80C186 GENERAL TIMING RESPONSES (Listed More Than Once)** | | | | | | | | | |
| $T_{CHSV}$ | Status Active Delay | 5 | 45 | 5 | 35 | 5 | 31 | ns | |
| $T_{CLSH}$ | Status Inactive Delay | 5 | 46 | 5 | 35 | 5 | 30 | ns | |
| $T_{CLAV}$ | Address Valid Delay | 5 | 44 | 5 | 36 | 5 | 33 | ns | |
| $T_{CHLH}$ | ALE Active Delay | | 30 | | 25 | | 20 | ns | |
| $T_{LHLL}$ | ALE Width | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | $T_{CLCL}$ - 15 | | ns | |
| $T_{CHLL}$ | ALE Inactive Delay | | 30 | | 25 | | 20 | ns | |
| $T_{DXDL}$ | $\overline{DEN}$ Inactive to DT/$\overline{R}$ Low | | 0 | | 0 | | 0 | ns | Equal Loading |
| $T_{CHCTV}$ | Control Active Delay 2 | 5 | 44 | 5 | 37 | 5 | 31 | ns | |

### SOFTWARE HALT CYCLE WAVEFORMS




**NOTE:**
1. For write cycle followed by halt cycle.

270354-40

## A. C. CHARACTERISTICS

T-49-17-15

### CLOCK TIMINGS

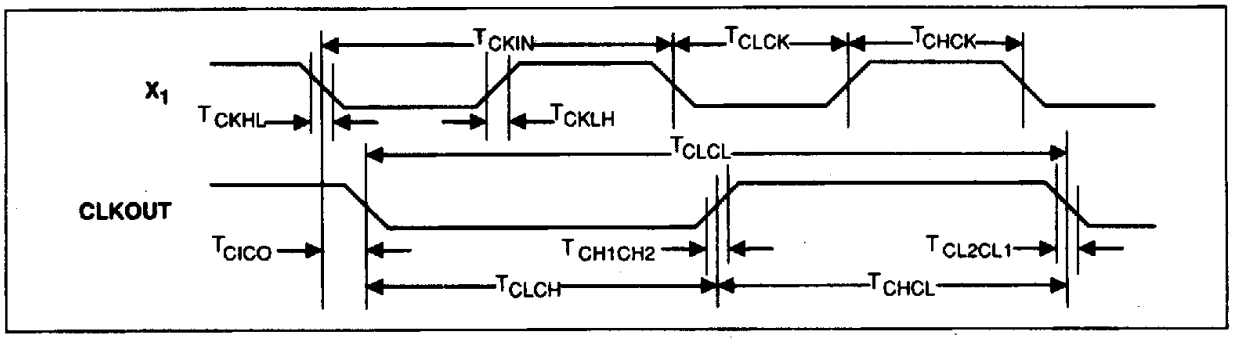
$T_A = 0^\circ C$ to $+70^\circ C$, $V_{CC}$ = 5V ± 10% except $V_{CC}$ = 5V± 5% at f > 12.5 MHz

All timings are measured at 1.5V 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 (12.5-16 MHz).
For A.C. tests, input $V_{IL}$ = 0.45V and $V_{IH}$ = 2.4V except at X1 where $V_{IH} = V_{CC}$ - 0.5V.

| Symbol | Parameter | 80C186 | | 80C186-12 | | 80C186-16 | | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| **80C186 CLKIN REQUIREMENTS** Measurements taken with following conditions: External clock input to X1 and X2 not connected (float) | | | | | | | | | |
| $T_{CKIN}$ | CLKIN Period | 50 | 1000 | 40 | 1000 | 31.25 | 1000 | ns | |
| $T_{CLCK}$ | CLKIN Low Time | 20 | | 16 | | 13 | | ns | 1.5 V(2) |
| $T_{CHCK}$ | CLKIN High Time | 20 | | 16 | | 13 | | ns | 1.5 V(2) |
| $T_{CKHL}$ | CLKIN Fall Time | | 5 | | 5 | | 5 | ns | 3.5 to 1.0V |
| $T_{CKLH}$ | CLKIN Rise Time | | 5 | | 5 | | 5 | ns | 1.0 to 3.5V |
| **80C186 CLKOUT TIMING** | | | | | | | | | |
| $T_{CICO}$ | CLKIN to CLKOUT Skew | | 25 | | 21 | | 17 | ns | |
| $T_{CLCL}$ | CLKOUT Period | 100 | 2000 | 80 | 2000 | 62.5 | 2000 | ns | |
| $T_{CLCH}$ | CLKOUT Low Time | 0.5 $T_{CLCL}$ -8 | | 0.5 $T_{CLCL}$ -7 | | 0.5 $T_{CLCL}$ -7 | | ns | $C_L$=100pF(2) |
| | | 0.5 $T_{CLCL}$ -6 | | 0.5 $T_{CLCL}$ -5 | | 0.5 $T_{CLCL}$ -5 | | ns | $C_L$=50pF(3) |
| $T_{CHCL}$ | CLKOUT High Time | 0.5 $T_{CLCL}$ -8 | | 0.5 $T_{CLCL}$ -7 | | 0.5 $T_{CLCL}$ -7 | | ns | $C_L$=100pF(4) |
| | | 0.5 $T_{CLCL}$ -6 | | 0.5 $T_{CLCL}$ -5 | | 0.5 $T_{CLCL}$ -5 | | ns | $C_L$=50pF(3) |
| $T_{CH1CH2}$ | CLKOUT Rise Time | | 10 | | 10 | | 10 | ns | 1.0 to 3.5V |
| $T_{CL2CL1}$ | CLKOUT Fall Time | | 10 | | 10 | | 10 | ns | 3.5 to 1.0V |

**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: $V_{CC}$ = 5.5V (5.25V @ 16 MHz). $T_A = 70^\circ C$.
3. Not Tested.
4. Tested under worst case conditions: $V_{CC}$ = 4.5V (4.75V @ 16 MHz). $T_A = 0^\circ C$.

### CLOCK WAVEFORMS




270354-41

24

## A. C. CHARACTERISTICS

### READY, PERIPHERAL, AND QUEUE STATUS TIMINGS



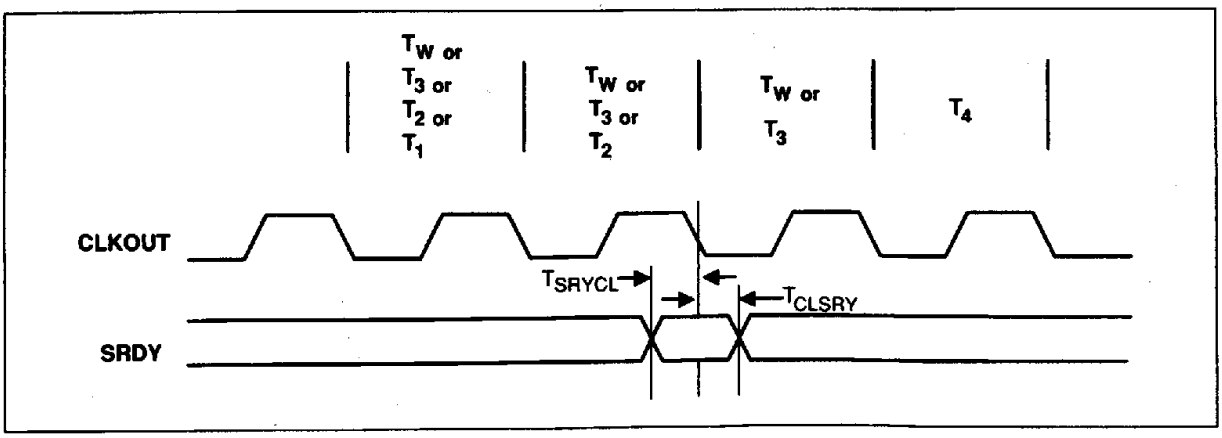
$T_A = 0^\circ C$ to $+70^\circ C$, $V_{CC}$ = 5V ± 10% except $V_{CC}$ = 5V± 5% at f > 12.5 MHz

All timings are measured at 1.5V 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 (12.5-16 MHz).
For A.C. tests, input $V_{IL}$ = 0.45V and $V_{IH}$= 2.4V except at X1 where $V_{IH} = V_{CC}$- 0.5V.

| Symbol | Parameter | 80C186 | | 80C186-12 | | 80C186-16 | | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| **80C186 READY AND PERIPHERAL TIMING REQUIREMENTS** | | | | | | | | | |
| $T_{SRYCL}$ | Synchronous Ready(SRDY) Transition Setup Time (1) | 15 | | 15 | | 15 | | ns | |
| $T_{CLSRY}$ | SRDY Transition Hold Time (1) | 15 | | 15 | | 15 | | ns | |
| $T_{ARYCH}$ | ARDY Resolution Transition Setup Time (2) | 15 | | 15 | | 15 | | ns | |
| $T_{CLARX}$ | ARDY Active Hold Time (1) | 15 | | 15 | | 15 | | ns | |
| $T_{ARYCHL}$ | ARDY Inactive Holding Time | 15 | | 15 | | 15 | | ns | |
| $T_{ARYLCL}$ | Asynchronous Ready (ARDY) Setup Time (1) | 25 | | 25 | | 25 | | ns | |
| $T_{INVCH}$ | INTx, NMI, $\overline{TEST}$/BUSY, TMR IN Setup Time (2) | 15 | | 15 | | 15 | | ns | |
| $T_{INVCL}$ | DRQ0, DRQ1 Setup Time (2) | 15 | | 15 | | 15 | | ns | |
| **80C186 PERIPHERAL AND QUEUE STATUS TIMING RESPONSES** | | | | | | | | | |
| $T_{CLTMV}$ | Timer Output Delay | | 40 | | 33 | | 27 | ns | |
| $T_{CHQSV}$ | Queue Status Delay | | 37 | | 32 | | 30 | ns | |

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

### SYNCHRONOUS READY (SRDY) WAVEFORMS

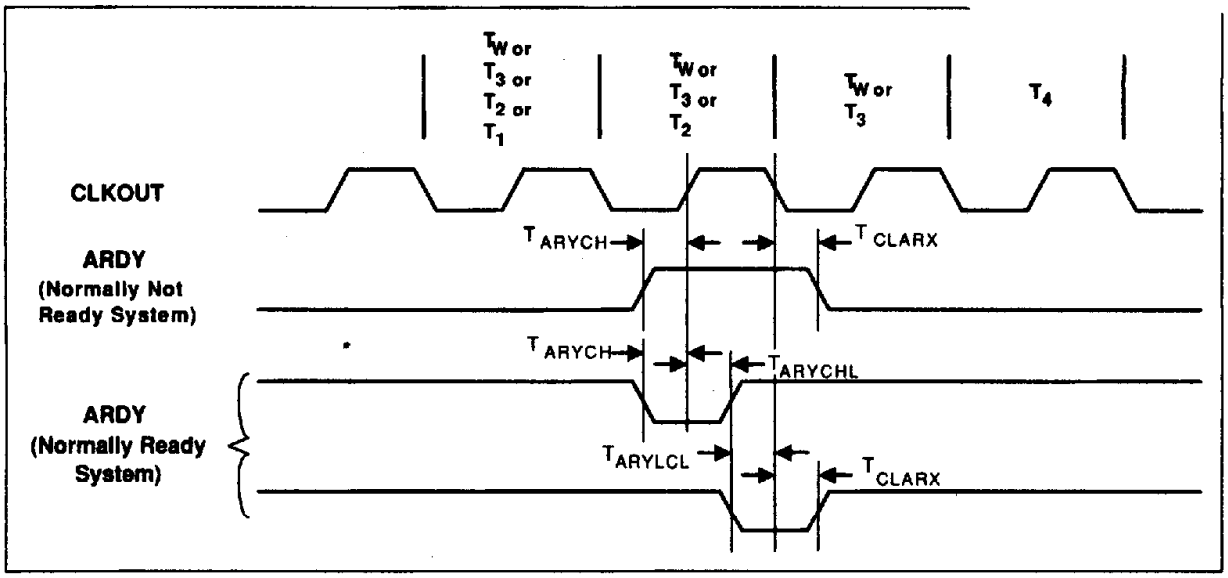



270354-42

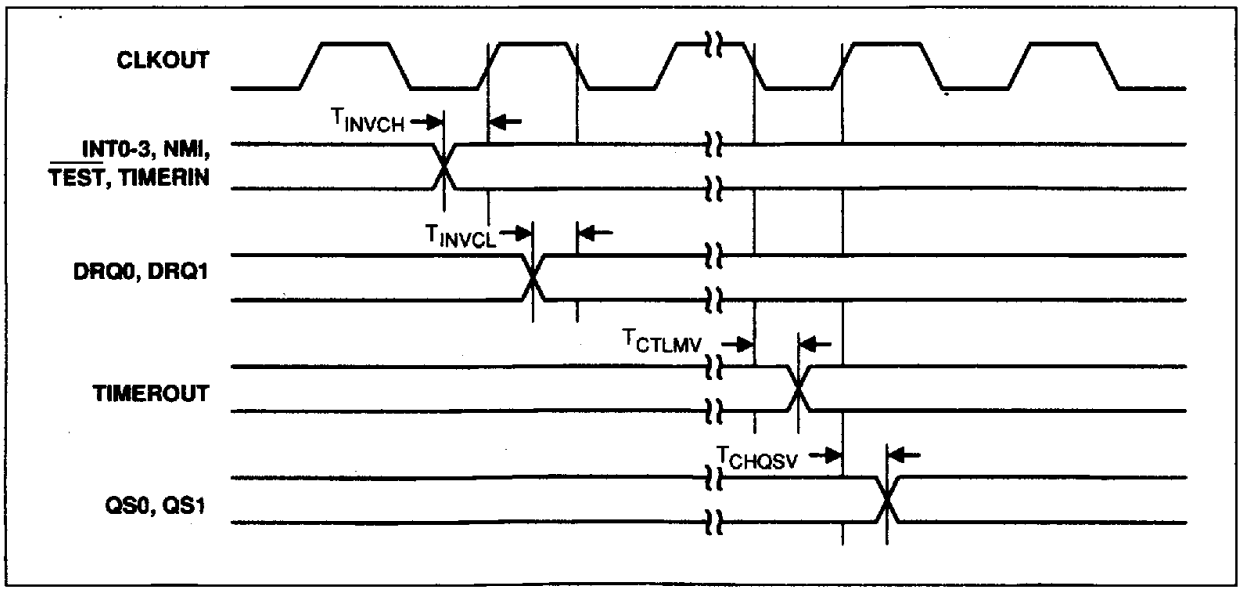
## A. C. CHARACTERISTICS

### ASYNCHRONOUS READY (ARDY) WAVEFORMS

T-49-17-15




### PERIPHERAL AND QUEUE STATUS WAVEFORMS




270354-43

24

## A. C. CHARACTERISTICS

T-49-17-15

### RESET AND HOLD/HLDA TIMINGS

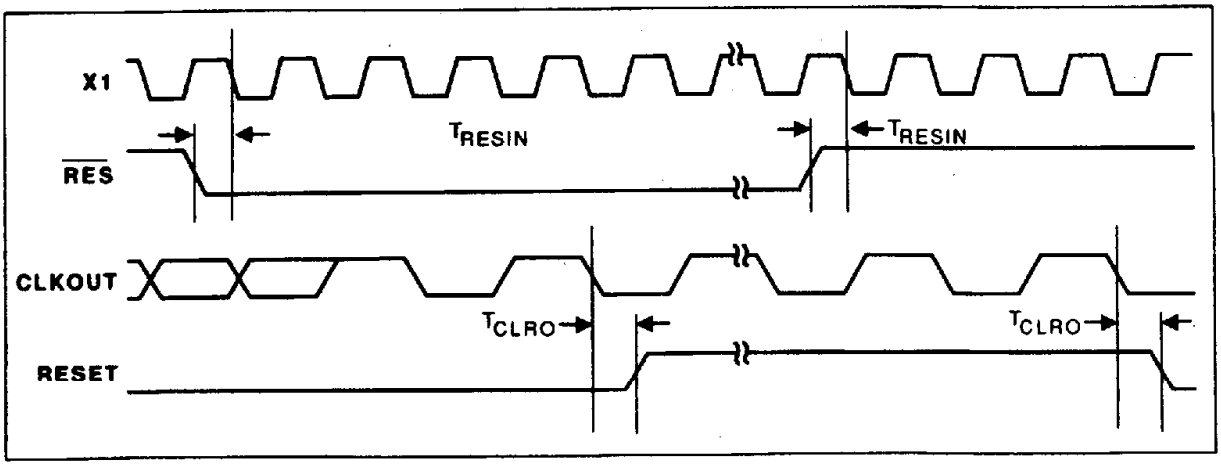
$T_A = 0^\circ C$ to $+70^\circ C$, $V_{CC} = 5V \pm 10\%$ except $V_{CC} = 5V \pm 5\%$ at $f > 12.5$ MHz

All timings are measured at 1.5V 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 (12.5-16 MHz).
For A.C. tests, input $V_{IL}$ = 0.45V and $V_{IH}$ = 2.4V except at X1 where $V_{IH} = V_{CC} - 0.5V$.

| Symbol | Parameter | 80C186 | | 80C186-12 | | 80C186-16 | | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| **80C186 RESET AND HOLD/HLDA TIMING REQUIREMENTS** | | | | | | | | | |
| $T_{RESIN}$ | $\overline{RES}$ Setup | 15 | | 15 | | 15 | | ns | |
| $T_{HVCL}$ | HOLD Setup (1) | 15 | | 15 | | 15 | | ns | |
| **80C186 GENERAL TIMING RESPONSES (Listed More Than Once)** | | | | | | | | | |
| $T_{CLAZ}$ | Address Float Delay | $T_{CLAX}$ | 30 | $T_{CLAX}$ | 25 | $T_{CLAX}$ | 20 | ns | |
| $T_{CLAV}$ | Address Valid Delay | 5 | 44 | 5 | 36 | 5 | 33 | ns | |
| **80C186 RESET AND HOLD/HLDA TIMING RESPONSES** | | | | | | | | | |
| $T_{CLRO}$ | Reset Delay | | 40 | | 33 | | 27 | ns | |
| $T_{CLHAV}$ | HLDA Valid Delay | 3 | 40 | 3 | 33 | 3 | 25 | ns | |
| $T_{CHCZ}$ | Command Lines Float Delay | | 40 | | 33 | | 28 | ns | |
| $T_{CHCV}$ | Command Lines Valid Delay (after Float) | | 44 | | 36 | | 32 | ns | |

**NOTE:**
1. To guarantee recognition at next clock.

### RESET WAVEFORMS

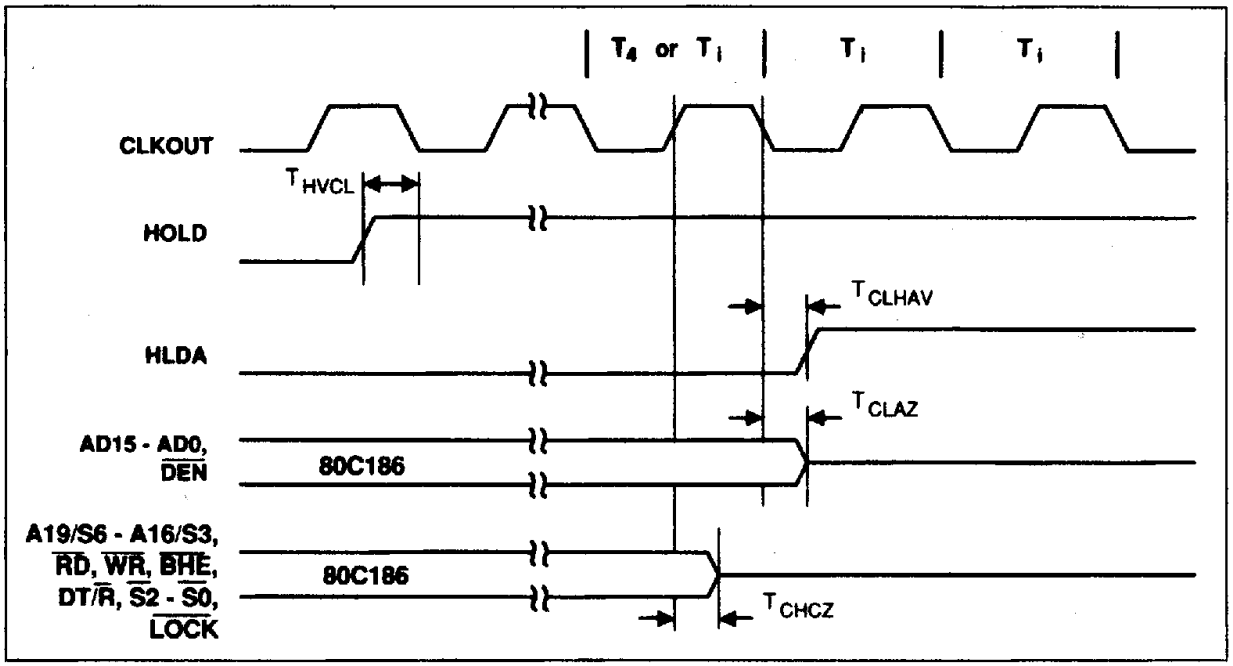



270354-44

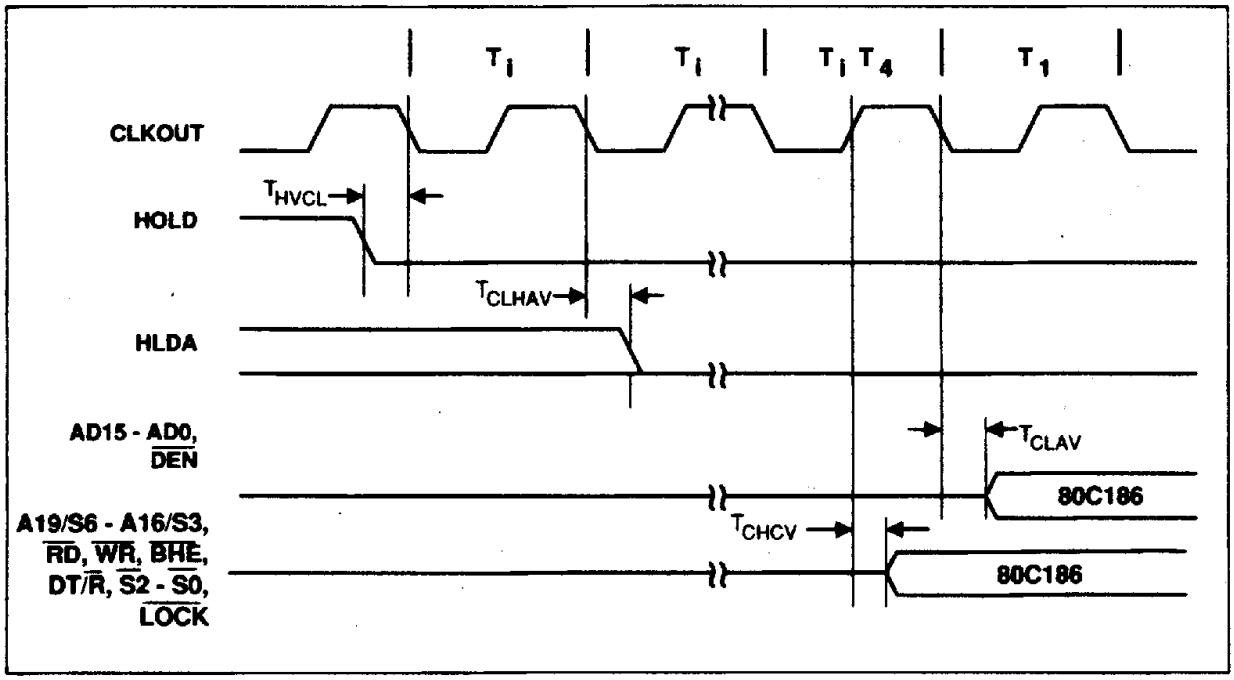
## A. C. CHARACTERISTICS

### HOLD/HLDA WAVEFORMS (Entering Hold)

T-49-17-15




### HOLD/HLDA WAVEFORMS (Leaving Hold)




270354-45

24

## EXPLANATION OF THE AC SYMBOLS

T-49-17-15

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 ($\overline{S0}$, $\overline{S1}$, $\overline{S2}$)
SRY: Synchronous Ready Input
V: Valid
W: WR Signal
X: No Longer a Valid Logic Level
Z: Float

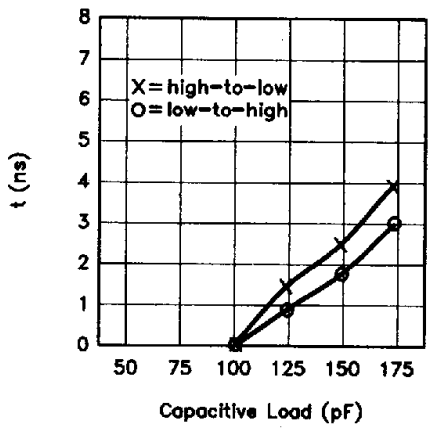
Examples:

$T_{CLAV}$ — 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

T-49-17-15

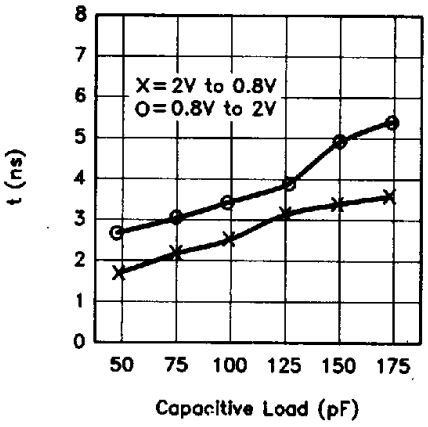
**Typical Output Delay Capacitive Derating**




270354-31

**Figure 47. Capacitive Derating Curve**

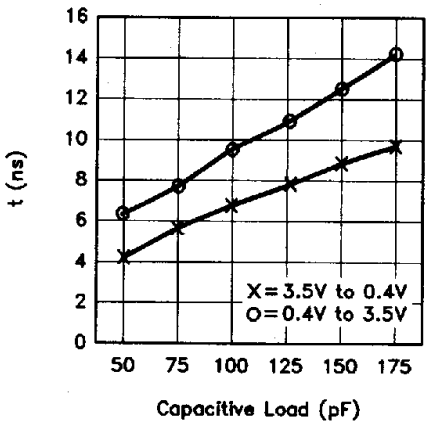
**Typical Rise and Fall Times for TTL Voltage Levels**




270354-32

**Figure 48. TTL Level Rise and Fall Times for Output Buffers**

**Typical Rise and Fall Times for CMOS Voltage Levels**




270354-33

**Figure 49. CMOS Level Rise and Fall Times for Output Buffers**

24

## 80C186 EXPRESS

The Intel EXPRESS system offers enhancements to the operational specifications of the 80C186 microprocessor. EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The 80C186 EXPRESS program includes an extended temperature range. With the commercial standard temperature range, operational characteristics are guaranteed over the temperature range of 0°C to +70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of −40°C to +85°C.

Package types and EXPRESS versions are identified by a one or two-letter prefix to the part number. The prefixes are listed in Table 18. All AC and DC specifications not mentioned in this section are the same for both commercial and EXPRESS parts.

**Table 18. Prefix Identification**

| Prefix | Package Type | Temperature Range |
|---|---|---|
| A | PGA | Commercial |
| N | PLCC | Commercial |
| R | LCC | Commercial |
| TA | PGA | Extended |
| TN | PLCC | Extended |
| TR | LCC | Extended |

**NOTE:**
Extended temperature versions of the 80C186 are not available at 16 MHz.

## 80C186 EXECUTION TIMINGS

A determination of 80C186 program execution timing must consider the 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 word-data is located on even-address boundaries.

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 accesses 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.

With a 16-bit BIU, the 80C186 has sufficient bus performance to ensure that an adequate number of prefetched bytes will reside in the queue most of the time. Therefore, actual program execution time will not be substantially greater than that derived from adding the instruction timings shown.

## INSTRUCTION SET SUMMARY

| Function | Format | | | | Clock Cycles | Comments |
|---|---|---|---|---|---|---|
| **DATA TRANSFER** | | | | | | |
| **MOV = Move:** | | | | | | |
| Register to Register/Memory | 1000100w | mod reg r/m | | | 2/12 | |
| Register/memory to register | 1000101w | mod reg r/m | | | 2/9 | |
| Immediate to register/memory | 1100011w | mod 000 r/m | data | data if w=1 | 12–13 | 8/16-bit |
| Immediate to register | 1011w reg | data | data if w=1 | | 3–4 | 8/16-bit |
| Memory to accumulator | 1010000w | addr-low | addr-high | | 8 | |
| Accumulator to memory | 1010001w | addr-low | addr-high | | 9 | |
| Register/memory to segment register | 10001110 | mod 0 reg r/m | | | 2/9 | |
| Segment register to register/memory | 10001100 | mod 0 reg r/m | | | 2/11 | |
| **PUSH = Push:** | | | | | | |
| Memory | 11111111 | mod 1 1 0 r/m | | | 16 | |
| Register | 01010 reg | | | | 10 | |
| Segment register | 000reg110 | | | | 9 | |
| Immediate | 011010s0 | data | data if s=0 | | 10 | |
| PUSHA = Push All | 01100000 | | | | 36 | |
| **POP = Pop:** | | | | | | |
| Memory | 10001111 | mod 0 0 0 r/m | | | 20 | |
| Register | 01011 reg | | | | 10 | |
| Segment register | 000reg111 | (reg≠01) | | | 8 | |
| POPA = Pop All | 01100001 | | | | 51 | |
| **XCHG = Exchange:** | | | | | | |
| Register/memory with register | 1000011w | mod reg r/m | | | 4/17 | |
| Register with accumulator | 10010 reg | | | | 3 | |
| **IN = Input from:** | | | | | | |
| Fixed port | 1110010w | port | | | 10 | |
| Variable port | 1110110w | | | | 8 | |
| **OUT = Output to:** | | | | | | |
| Fixed port | 1110011w | port | | | 9 | |
| Variable port | 1110111w | | | | 7 | |
| **XLAT** = Translate byte to AL | 11010111 | | | | 11 | |
| **LEA** = Load EA to register | 10001101 | mod reg r/m | | | 6 | |
| **LDS** = Load pointer to DS | 11000101 | mod reg r/m | (mod≠11) | | 18 | |
| **LES** = Load pointer to ES | 11000100 | mod reg r/m | (mod≠11) | | 18 | |
| **LAHF** = Load AH with flags | 10011111 | | | | 2 | |
| **SAHF** = Store AH into flags | 10011110 | | | | 3 | |
| **PUSHF** = Push flags | 10011100 | | | | 9 | |
| **POPF** = Pop flags | 10011101 | | | | 8 | |

Shaded areas indicate instructions not available in 8086, 8088 microsystems.

24

## INSTRUCTION SET SUMMARY (Continued)

T-49-17-15

| Function | Format | | | | Clock Cycles | Comments |
|---|---|---|---|---|---|---|
| **DATA TRANSFER** (Continued) | | | | | | |
| **SEGMENT = Segment Override:** | | | | | | |
| **CS** | 00101110 | | | | 2 | |
| **SS** | 00110110 | | | | 2 | |
| **DS** | 00111110 | | | | 2 | |
| **ES** | 00100110 | | | | 2 | |
| **ARITHMETIC** | | | | | | |
| **ADD = Add:** | | | | | | |
| Reg/memory with register to either | 000000dw | mod reg r/m | | | 3/10 | |
| Immediate to register/memory | 100000sw | mod 000 r/m | data | data if s w=01 | 4/16 | |
| Immediate to accumulator | 0000010w | data | data if w=1 | | 3/4 | 8/16-bit |
| **ADC = Add with carry:** | | | | | | |
| Reg/memory with register to either | 000100dw | mod reg r/m | | | 3/10 | |
| Immediate to register/memory | 100000sw | mod 010 r/m | data | data if s w=01 | 4/16 | |
| Immediate to accumulator | 0001010w | data | data if w=1 | | 3/4 | 8/16-bit |
| **INC = Increment:** | | | | | | |
| Register/memory | 1111111w | mod 000 r/m | | | 3/15 | |
| Register | 01000 reg | | | | 3 | |
| **SUB = Subtract:** | | | | | | |
| Reg/memory and register to either | 001010dw | mod reg r/m | | | 3/10 | |
| Immediate from register/memory | 100000sw | mod 101 r/m | data | data if s w=01 | 4/16 | |
| Immediate from accumulator | 0010110w | data | data if w=1 | | 3/4 | 8/16-bit |
| **SBB = Subtract with borrow:** | | | | | | |
| Reg/memory and register to either | 000110dw | mod reg r/m | | | 3/10 | |
| Immediate from register/memory | 100000sw | mod 011 r/m | data | data if s w=01 | 4/16 | |
| Immediate from accumulator | 0001110w | data | data if w=1 | | 3/4 | 8/16-bit |
| **DEC = Decrement** | | | | | | |
| Register/memory | 1111111w | mod 001 r/m | | | 3/15 | |
| Register | 01001 reg | | | | 3 | |
| **CMP = Compare:** | | | | | | |
| Register/memory with register | 0011101w | mod reg r/m | | | 3/10 | |
| Register with register/memory | 0011100w | mod reg r/m | | | 3/10 | |
| Immediate with register/memory | 100000sw | mod 111 r/m | data | data if s w=01 | 3/10 | |
| Immediate with accumulator | 0011110w | data | data if w=1 | | 3/4 | 8/16-bit |
| **NEG** = Change sign register/memory | 1111011w | mod 011 r/m | | | 3/10 | |
| **AAA** = ASCII adjust for add | 00110111 | | | | 8 | |
| **DAA** = Decimal adjust for add | 00100111 | | | | 4 | |
| **AAS** = ASCII adjust for subtract | 00111111 | | | | 7 | |
| **DAS** = Decimal adjust for subtract | 00101111 | | | | 4 | |
| **MUL** = Multiply (unsigned): | 1111011w | mod 100 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, 8088 microsystems.

## INSTRUCTION SET SUMMARY (Continued)

T-49-17-15

| Function | Format | | | | Clock Cycles | Comments |
|---|---|---|---|---|---|---|
| **ARITHMETIC** (Continued) | | | | | | |
| **IMUL** = Integer multiply (signed): | 1111011w | mod101 r/m | | | | |
| Register-Byte | | | | | 25-28 | |
| Register-Word | | | | | 34-37 | |
| Memory-Byte | | | | | 31-34 | |
| Memory-Word | | | | | 40-43 | |
| **IMUL** = Integer immediate multiply (signed) | 011010s1 | mod reg r/m | data | data if s=0 | 22-25/ 29-32 | |
| **DIV** = Divide (unsigned): | 1111011w | mod110 r/m | | | | |
| Register-Byte | | | | | 29 | |
| Register-Word | | | | | 38 | |
| Memory-Byte | | | | | 35 | |
| Memory-Word | | | | | 44 | |
| **IDIV** = Integer divide (signed): | 1111011w | mod111 r/m | | | | |
| Register-Byte | | | | | 44-52 | |
| Register-Word | | | | | 53-61 | |
| Memory-Byte | | | | | 50-58 | |
| Memory-Word | | | | | 59-67 | |
| **AAM** = ASCII adjust for multiply | 11010100 | 00001010 | | | 19 | |
| **AAD** = ASCII adjust for divide | 11010101 | 00001010 | | | 15 | |
| **CBW** = Convert byte to word | 10011000 | | | | 2 | |
| **CWD** = Convert word to double word | 10011001 | | | | 4 | |
| **LOGIC** | | | | | | |
| **Shift/Rotate Instructions:** | | | | | | |
| Register/Memory by 1 | 1101000w | mod TTT r/m | | | 2/15 | |
| Register/Memory by CL | 1101001w | mod TTT r/m | | | 5+n/17+n | |
| Register/Memory by Count | 1100000w | mod TTT r/m | count | | 5+n/17+n | |
| | **TTT Instruction**<br>000 ROL<br>001 ROR<br>010 RCL<br>011 RCR<br>100 SHL/SAL<br>101 SHR<br>111 SAR | | | | | |
| **AND** = **And:** | | | | | | |
| Reg/memory and register to either | 001000dw | mod reg r/m | | | 3/10 | |
| Immediate to register/memory | 1000000w | mod100 r/m | data | data if w=1 | 4/16 | |
| Immediate to accumulator | 0010010w | data | data if w=1 | | 3/4 | 8/16-bit |
| **TEST** = **And function to flags, no result:** | | | | | | |
| Register/memory and register | 1000010w | mod reg r/m | | | 3/10 | |
| Immediate data and register/memory | 1111011w | mod000 r/m | data | data if w=1 | 4/10 | |
| Immediate data and accumulator | 1010100w | data | data if w=1 | | 3/4 | 8/16-bit |
| **OR** = **Or:** | | | | | | |
| Reg/memory and register to either | 000010dw | mod reg r/m | | | 3/10 | |
| Immediate to register/memory | 1000000w | mod001 r/m | data | data if w=1 | 4/16 | |
| Immediate to accumulator | 0000110w | data | data if w=1 | | 3/4 | 8/16-bit |

Shaded areas indicate instructions not available in 8086, 8088 microsystems.

24

## INSTRUCTION SET SUMMARY (Continued)

| Function | Format | | | | Clock Cycles | Comments |
|---|---|---|---|---|---|---|
| **LOGIC** (Continued) | | | | | | |
| **XOR = Exclusive or:** | | | | | | |
| Reg/memory and register to either | 001100dw | mod reg r/m | | | 3/10 | |
| Immediate to register/memory | 1000000w | mod 110 r/m | data | data if w=1 | 4/16 | |
| Immediate to accumulator | 0011010w | data | data if w=1 | | 3/4 | 8/16-bit |
| **NOT** = Invert register/memory | 1111011w | mod 010 r/m | | | 3/10 | |
| **STRING MANIPULATION** | | | | | | |
| **MOVS** = Move byte/word | 1010010w | | | | 14 | |
| **CMPS** = Compare byte/word | 1010011w | | | | 22 | |
| **SCAS** = Scan byte/word | 1010111w | | | | 15 | |
| **LODS** = Load byte/wd to AL/AX | 1010110w | | | | 12 | |
| **STOS** = Store byte/wd from AL/AX | 1010101w | | | | 10 | |
| **INS** = Input byte/wd from DX port | 0110110w | | | | 14 | |
| **OUTS** = Output byte/wd to DX port | 0110111w | | | | 14 | |
| Repeated by count in CX (REP/REPE/REPZ/REPNE/REPNZ) | | | | | | |
| **MOVS** = Move string | 11110010 | 1010010w | | | 8+8n | |
| **CMPS** = Compare string | 1111001z | 1010011w | | | 5+22n | |
| **SCAS** = Scan string | 1111001z | 1010111w | | | 5+15n | |
| **LODS** = Load string | 11110010 | 1010110w | | | 6+11n | |
| **STOS** = Store string | 11110010 | 1010101w | | | 6+9n | |
| **INS** = Input string | 11110010 | 0110110w | | | 8+8n | |
| **OUTS** = Output string | 11110010 | 0110111w | | | 8+8n | |
| **CONTROL TRANSFER** | | | | | | |
| **CALL = Call:** | | | | | | |
| Direct within segment | 11101000 | disp-low | disp-high | | 15 | |
| Register/memory indirect within segment | 11111111 | mod 010 r/m | | | 13/19 | |
| Direct intersegment | 10011010 | segment offset | | | 23 | |
| | | segment selector | | | | |
| Indirect intersegment | 11111111 | mod 011 r/m | (mod ≠ 11) | | 38 | |
| **JMP = Unconditional jump:** | | | | | | |
| Short/long | 11101011 | disp-low | | | 14 | |
| Direct within segment | 11101001 | disp-low | disp-high | | 14 | |
| Register/memory indirect within segment | 11111111 | mod 100 r/m | | | 11/17 | |
| Direct intersegment | 11101010 | segment offset | | | 14 | |
| | | segment selector | | | | |
| Indirect intersegment | 11111111 | mod 101 r/m | (mod ≠ 11) | | 26 | |

Shaded areas indicate instructions not available in 8086, 8088 microsystems.

## INSTRUCTION SET SUMMARY (Continued)

T-49-17-15

| Function | Format | | | | Clock Cycles | Comments |
|---|---|---|---|---|---|---|
| **CONTROL TRANSFER** (Continued) | | | | | | |
| **RET** = **Return from CALL:** | | | | | | |
| Within segment | 11000011 | | | | 16 | |
| Within seg adding immed to SP | 11000010 | data-low | data-high | | 18 | |
| Intersegment | 11001011 | | | | 22 | |
| Intersegment adding immediate to SP | 11001010 | data-low | data-high | | 25 | |
| **JE/JZ** = Jump on equal/zero | 01110100 | disp | | | 4/13 | JMP not taken/JMP taken |
| **JL/JNGE** = Jump on less/not greater or equal | 01111100 | disp | | | 4/13 | |
| **JLE/JNG** = Jump on less or equal/not greater | 01111110 | disp | | | 4/13 | |
| **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 taken/LOOP taken |
| **LOOPZ/LOOPE** = Loop while zero/equal | 11100001 | disp | | | 6/16 | |
| **LOOPNZ/LOOPNE** = Loop while not zero/equal | 11100000 | disp | | | 6/16 | |
| **ENTER** = Enter Procedure | 11001000 | data-low | data-high | L | | |
| L = 0 | | | | | 15 | |
| L = 1 | | | | | 25 | |
| L > 1 | | | | | 22 + 16(n - 1) | |
| **LEAVE** = Leave Procedure | 11001001 | | | | 8 | |
| **INT** = **Interrupt:** | | | | | | |
| Type specified | 11001101 | type | | | 47 | |
| Type 3 | 11001100 | | | | 45 | if INT. taken/ if INT. not taken |
| **INTO** = Interrupt on overflow | 11001110 | | | | 48/4 | |
| **IRET** = Interrupt return | 11001111 | | | | 28 | |
| **BOUND** = Detect value out of range | 01100010 | mod reg r/m | | | 33-35 | |

Shaded areas indicate instructions not available in 8086, 8088 microsystems.

24

## INSTRUCTION SET SUMMARY (Continued)

| Function | Format | Clock Cycles | Comments |
|---|---|---|---|
| **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 | |
| | (TTT LLL are opcode to processor extension) | | |

Shaded areas indicate instructions not available in 8086, 8088 microsystems.

## FOOTNOTES

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

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

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

*except if mod = 00 and r/m = 110 then 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.

## REVISION HISTORY

**The sections significantly revised since version -004 are:**

| | |
|---|---|
| Pin Description Table | Added note to $\overline{TEST}$/BUSY pin requiring proper RESET at power-up to configure pin as input.<br>Renamed pin 44 to INT1/$\overline{SELECT}$ and pin 41 to INT3/$\overline{INTA1}$/IRQ to better describe their functions in Slave Mode. |
| Initialization and Processor Reset | Added reminder to drive $\overline{RES}$ pin LOW during power-up. |
| Read and Write Cycle Waveforms | Clarified applicability of $T_{CLCSV}$ to latched A1 and A2 in footnotes. |
| Slave Mode Operation | The three low order bits associated with vector generation and performing EOI are not alterable; however, the priority levels are programmable. This information is a clarification only. |

**The sections significantly revised since version -003 are:**

| | |
|---|---|
| Front Page | Deleted references to burn-in devices. |
| Local Bus Controller and Reset | Clarified effects of excessive loading on pins with internal pullup devices. Equivalent resistance no longer shown. |
| D.C. Characteristics | Renamed $V_{CLI}$ to $V_{IL1}$. Renamed $V_{CHI}$ to $V_{IH3}$. Changed $V_{OH}$ (min.) from 0.8 $V_{CC}$ to $V_{CC}$ − 0.5V. Changed $I_{CC}$ (max.) from 180 mA to 150 mA at 16 MHz, 150 mA to 120 mA at 12.5 mA, and 100 mA to 120 mA at 10 MHz. Changed $V_{CLO}$ (max.) from 0.5V to 0.45V. Changed $V_{CHO}$ (min.) from 0.8 $V_{CC}$ to $V_{CC}$ − 0.5V. Clarified effect of excessive loading on pins with internal pullup devices. |
| Power Supply Current | Added equation and graph for maximum current. |
| A.C. Characteristics | Many timings changed (all listed in ns): $T_{DVCL}$ (min.) at 16 MHz from 10 to 15; $T_{CLDX}$ (min.) from 5 to 3; $T_{CLAV}$ (max.) at 10 MHz from 50 to 44; $T_{CHCV}$ (max.) from 45 to 44 at 10 MHz, and from 37 to 36 at 12.5 MHz; $T_{LHLL}$ (min.) from $T_{CLCL}$ − 30 to $T_{CLCL}$ − 15; $T_{LLAX}$ (min.) at 10 MHz from $T_{CHCL}$ − 20 to $T_{CHCL}$ − 15; $T_{CVCTV}$ (max.) from 56 to 44 at 10 MHz and from 47 to 37 at 12.5 MHz; $T_{CVDEX}$ (max.) from 56 to 44 at 10 MHz, 47 to 37 at 12.5 MHz, and from 35 to 31 at 16 MHz; $T_{RHAV}$ (min.) from $T_{CLCL}$ − 40 at 10 MHz and $T_{CLCL}$ − 20 at 12.5 MHz and 16 MHz to $T_{CLCL}$ − 15 at all frequencies; $T_{RLRH}$ (min.) from 2 $T_{CLCL}$ − 46 to 2 $T_{CLCL}$ − 30 at 10 MHz, from 2 $T_{CLCL}$ − 40 to 2 $T_{CLCL}$ − 25 at 12.5 MHz, and 2 $T_{CLCL}$ − 30 to 2 $T_{CLCL}$ − 25 at 16 MHz; $T_{WLWH}$ (min.) from 2 $T_{CLCL}$ − 34 to 2 $T_{CLCL}$ − 30 at 10 MHz, and 2 $T_{CLCL}$ − 30 to 2 $T_{CLCL}$ − 25 at 12.5 MHz; $T_{AVLL}$ (min.) from $T_{CLCH}$ − 19 to $T_{CLCH}$ − 18 at 10 MHz; $T_{CLSH}$ (max.) at 10 MHz from 50 to 46; $T_{CLTMV}$ (max.) from 48 to 40 at 10 MHz, 40 to 33 at 12.5 MHz, and 30 to 27 at 16 MHz; $T_{CLRO}$ (max.) from 48 to 40 at 10 MHz, 40 to 33 at 12.5 MHz, and 30 to 27 at 16 MHz; $T_{CHQSV}$ (max.) from 28 to 37 at 10 MHz, 28 to 32 at 12.5 MHz, and 25 to 30 at 16 MHz; $T_{CHDX}$ (min.) from 5 to 10; $T_{CLLV}$ (max.) at 10 MHz from 45 to 40 and at 12.5 MHz from 40 to 37; $T_{CLCSV}$ (max.) from 45 to 42 at 10 MHz; $T_{CHCSX}$ (max.) from 32 to 35 at 10 MHz, 28 to 30 at 12.5 MHz, and 23 to 25 at 16 MHz; and $T_{CH1CH2}$ and $T_{CL2CL1}$ (max.) at 16 MHz from 8 to 10. Added new timings for $T_{WHDEX}$, $T_{RHLH}$, and $T_{WHLH}$. Established min. timing for $T_{CLCSV}$. |
| Timing Waveforms | Section rearranged to show waveforms on same or facing page relative to corresponding tabular data. $T_{CLSRY}$ drawn to same clock edge as $T_{SRYCL}$. Drawing changed to indicate one less clock between HOLD inactive and HLDA inactive. |
| Specification Level Markings | New Section. |

24

**The sections significantly revised since version -002 are:**

| | |
|---|---|
| Block Diagram | Redrawn to illustrate numerics coprocessor interface. |
| Pin Description Table | Various descriptions rewritten for clarity. |
| Interrupt Vector Table | Redrawn for clarity. Interrupt Type 16 listed. |
| ESC Opcode Exception Description | Note added concerning ESC trap. |
| Oscillator Configurations | Deleted drive of X2 with inverted X1. |
| RESET Logic | Deleted paragraph concerning setup times for synchronization of multiple processors. |
| Local Bus Arbitration | Added description of HLDA when a refresh cycle is pending. |
| Local Bus Controller and Reset | Added description of pullup devices for appropriate pins. |
| DMA Controller | Added reminder to initialize transfer count registers and pointer registers. |
| Timers | Added reminder to intialize count registers. |
| DRAM Refresh Addresses | Refresh address counter described in figure. |
| D.C. Characteristics | $V_{IH2}$ indicated for SRDY, ARDY. $I_{CC}$ (max.) now indicated for all devices. |
| Power Supply Current | Typical $I_{CC}$ indicated. |
| A.C. Characteristics | Input $V_{IH}$ test condition at X1 added. $T_{CLDOX}$, $T_{CVCTV}$, $T_{CVCTX}$, $T_{CLHAV}$, and $T_{CLLV}$ minimums reduced from 5 ns to 3 ns. $T_{CLCH}$ (min.) and $T_{CHCL}$ (min.) relaxed by 2 ns. Added reminder that $T_{SRYCL}$ and $T_{CLSRY}$ must be met. |
| Explanation of the A.C. Symbols | New Section. |
| Major Cycle Timing Waveforms | $T_{DXDL}$ indicated in Read Cycle. $T_{CLRO}$ indicated. |
| Rise/Fall and Capacitive Derating Curves | New Figures added. |
| Instruction Set Summary | ESC instruction clock count deleted. |

**The sections significantly revised since version -001 are:**

| | |
|---|---|
| Pin Description Table | Noted $\overline{RES}$ to be low more than 4 clocks. |
| Oscillator Configurations | Added reminder not to drive X2. |
| DMA Transfer Rate Table | Corrected to reflect 16 MHz capability. |
| DMA Control Bit Descriptions | Moved and clarified note concerning TC condition for ST/$\overline{STOP}$ clearing during unsynchronized transfers. |
| Interrupt Controller, etc. | Renamed iRMX Mode to Slave Mode. |
| Interrupt Request Register | Noted that D0 and D1 are read/write, others read-only. |
| DRAM Refresh Addresses | Added figure to explain refresh address bits. |
| A.C. Characteristics | Many timings changed (all listed in ns): $T_{CLDX}$ (min.) from 8 to 5; $T_{SRYCL}$ (min.) from 20 to 15; $T_{HVCL}$ (min.) from 20 to 15; $T_{INVCH}$ (min.) from 25 to 15; $T_{INVCL}$ (min.) from 20 to 15; $T_{CLAV}$ at 12.5 MHz from 4–33 to 5–36; $T_{CLAV}$ at 16 MHz from 4–30 to 5–33; $T_{CLAX}$ (min.) to 0; $T_{CLDV}$ (min.) at 10 MHz from 10 to 5; $T_{CLDV}$ (min.) at 12.5 MHz from 10–33 to 5–36; $T_{CLDV}$ (min.) at 16 MHz from 10–30 to 5–33; $T_{CLDOX}$ (min.) from 10 at 10 MHz and 8 at 12.5 MHz to 5 at both frequencies; $T_{CVCTV}$ (max.) and $T_{CHCTV}$ (max.) at 16 MHz from 25 to 31; $T_{CHCTV}$ (min.) and $T_{CVDEX}$ (min.) both from 10 at 10 MHz and 8 at 12.5 MHz to 5 at both frequencies; $T_{CVCTX}$ (max.) at 16 MHz from 25 to 33; $T_{CLRL}$ at 10 MHz from 10–56 to 5–44; $T_{CLRL}$ at 12.5 MHz from 8–47 to 5–35; $T_{CLRL}$ (max.) at 16 MHz from 25 to 31; $T_{CLRH}$ (min.) at 10 MHz from 10 to 5 and at 12.5 MHz from 8 to 5; $T_{CHSV}$ (min.) at 10 MHz from 10 to 5 and at 12.5 MHz from 8 to 5; $T_{CHSV}$ (max.) at 16 MHz from 25 to 31; $T_{CLSH}$ (min.) at 10 MHz from 10 to 5 and at 12.5 MHz from 8 to 5; $T_{CHQSV}$ (max.) at 12.5 MHz from 23 to 28 and at 16 MHz from 23 to 25; $T_{CHDX}$ (min.) at 10 MHz from 10 to 5 and at 12.5 MHz from 8 to 5; $T_{AVCH}$ (min.) to 0; $T_{CLLV}$ (max.) at 10 MHz from 60 to 45 and at 12.5 MHz from 55 to 40 and at 16 MHz from 40 to 35; $T_{DXDL}$ (min.) to 0; $T_{CXCSX}$ (min.) from 35 at 10 MHz and 29 at 12.5 MHz and 25 at 16 MHz to $T_{CLCH} - 10$ at all frequencies; $T_{CHCSX}$ (min.) at 12.5 MHz and 16 MHz from 4–23 to 5–28 and 5–23 respectively. |
| Execution Timings | Clarified effect of bus width. |

## SPECIFICATION LEVEL MARKINGS

Current 80C186 devices bear backside lot code information consisting of seven digits followed by letters. The second, third, and fourth digits comprise a manufacturing date code. This preliminary data sheet applies only to 80C186 devices with a date code corresponding to week 25 of 1989 (backside markings x925xxx XXX) or later.