# 80C88

November 2004

## CMOS 8/16-Bit Microprocessor

### *Features*

- **Compatible with NMOS 8088**
- **Direct Software Compatibility with 80C86, 8086, 8088**
- **8-Bit Data Bus Interface; 16-Bit Internal Architecture**
- **Completely Static CMOS Design**
  - **DC . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5MHz (80C88)**
  - **DC . . . . . . . . . . . . . . . . . . . . . . . . . . . 8MHz (80C88-2)**
- **Low Power Operation**
  - **ICCSB . . . . . . . . . . . . . . . . . . . . . . . . 500μA Maximum**
  - **ICCOP . . . . . . . . . . . . . . . . . . . . 10mA/MHz Maximum**
- **1 Megabyte of Direct Memory Addressing Capability**
- **24 Operand Addressing Modes**
- **Bit, Byte, Word, and Block Move Operations**
- **8-Bit and 16-Bit Signed/Unsigned Arithmetic**
- **Bus-Hold Circuitry Eliminates Pull-up Resistors**
- **Wide Operating Temperature Ranges**
  - **C80C88 . . . . . . . . . . . . . . . . . . . . . . . . . 0°C to + 70°C**
  - **I80C88 . . . . . . . . . . . . . . . . . . . . . . . . . -40°C to +85°C**
  - **M80C88 . . . . . . . . . . . . . . . . . . . . . . -55°C to +125°C**

### *Description*

The Intersil 80C88 high performance 8/16-bit CMOS CPU is manufactured using a self-aligned silicon gate CMOS process (Scaled SAJI IV). Two modes of operation, MINimum for small systems and MAXimum for larger applications such as multiprocessing, allow user configuration to achieve the highest performance level.

Full TTL compatibility (with the exception of CLOCK) and industry-standard operation allow use of existing NMOS 8088 hardware and Intersil CMOS peripherals.

Complete software compatibility with the 80C86, 8086, and 8088 microprocessors allows use of existing software in new designs.

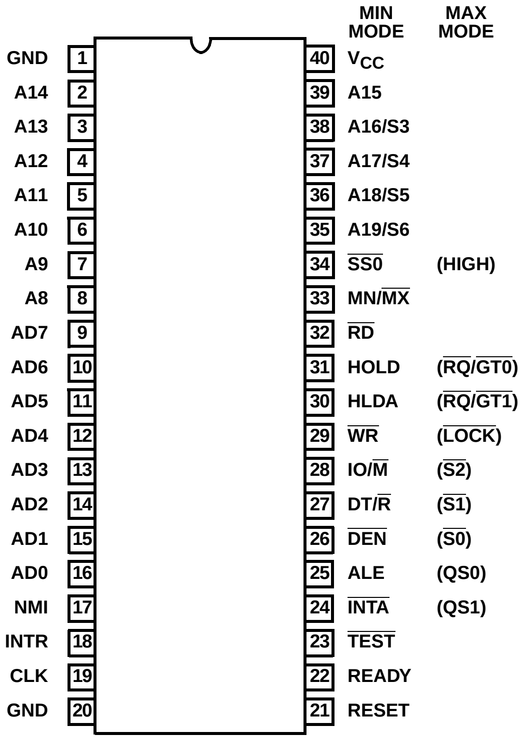
### *Ordering Information*

| PACKAGE | TEMPERATURE RANGE | 5MHz | 8MHz | PKG. DWG. # |
|---|---|---|---|---|
| PDIP | 0°C to +70°C | CP80C88 | CP80C88-2 | E40.6 |
| | -40°C to +85°C | IP80C88 | IP80C88-2 | E40.6 |
| CERDIP | -55°C to +125°C | MD80C88/B | | F40.6 |

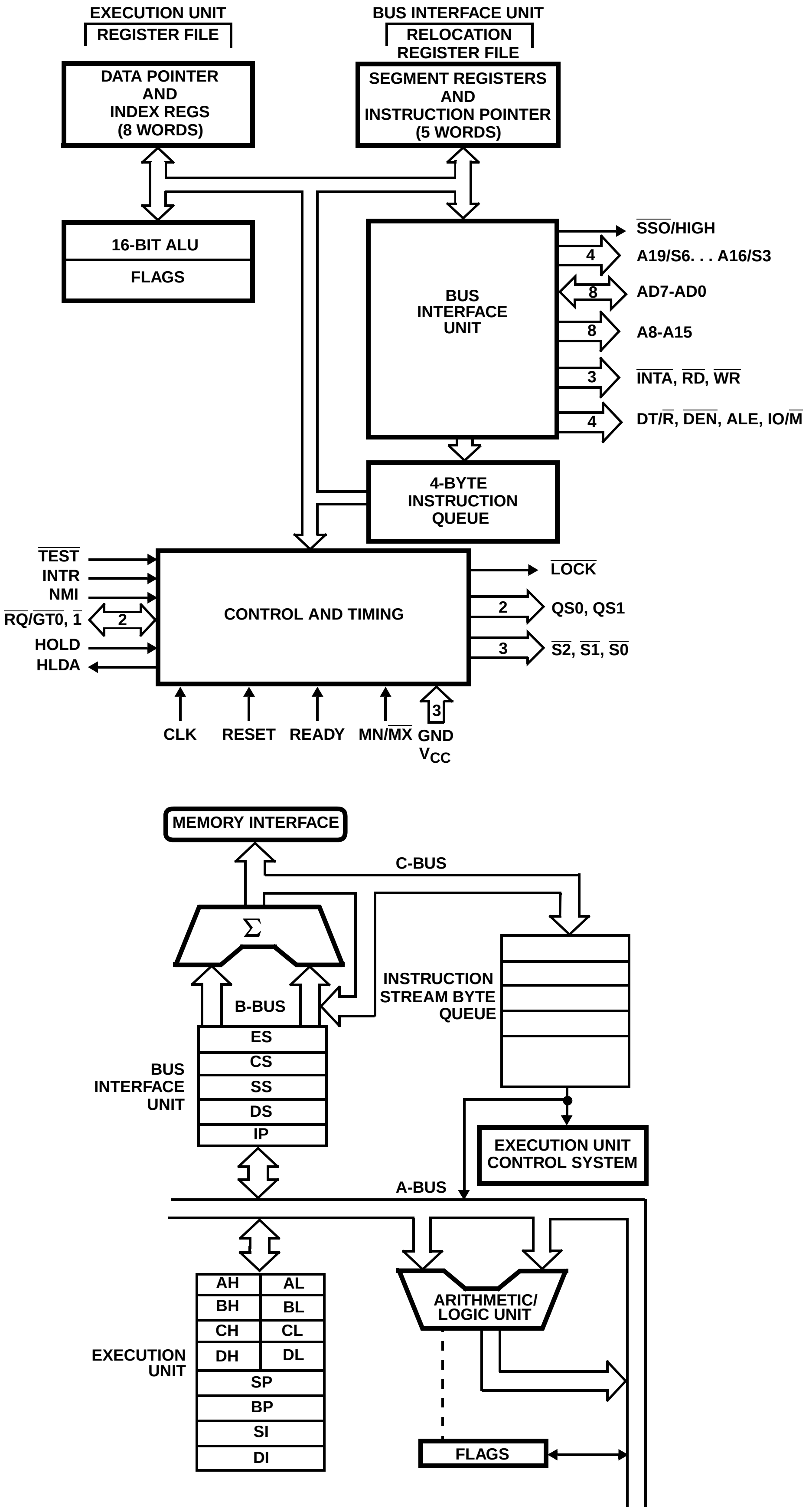
CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.
1-888-INTERSIL or 321-724-7143 | Intersil (and design) is a trademark of Intersil Americas Inc.

## Pinout

**80C88 (PDIP, CERIDP)**
TOP VIEW


MIN MODE
MAX MODE
GND 1
A14 2
A13 3
A12 4
A11 5
A10 6
A9 7
A8 8
AD7 9
AD6 10
AD5 11
AD4 12
AD3 13
AD2 14
AD1 15
AD0 16
NMI 17
INTR 18
CLK 19
GND 20
40 $V_{CC}$
39 A15
38 A16/S3
37 A17/S4
36 A18/S5
35 A19/S6
34 $\overline{SS0}$ (HIGH)
33 MN/$\overline{MX}$
32 $\overline{RD}$
31 HOLD ($\overline{RQ}$/$\overline{GT0}$)
30 HLDA ($\overline{RQ}$/$\overline{GT1}$)
29 $\overline{WR}$ ($\overline{LOCK}$)
28 IO/$\overline{M}$ ($\overline{S2}$)
27 DT/$\overline{R}$ ($\overline{S1}$)
26 $\overline{DEN}$ ($\overline{S0}$)
25 ALE (QS0)
24 $\overline{INTA}$ (QS1)
23 $\overline{TEST}$
22 READY
21 RESET

## Functional Diagram



## Pin Description

The following pin function descriptions are for 80C88 systems in either minimum or maximum mode. The “local bus” in these descriptions is the direct multiplexed bus interface connection to the 80C88 (without regard to additional bus buffers).

<table>
<tr><th>SYMBOL</th><th>PIN NUMBER</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>AD7-AD0</td><td>9-16</td><td>I/O</td><td>ADDRESS DATA BUS: These lines constitute the time multiplexed memory/IO address (T1) and data (T2,T3,Tw and T4) bus. These lines are active HIGH and are held at high impedance to the last valid level during interrupt acknowledge and local bus “hold acknowledge” or “grant sequence”</td></tr>
<tr><td>A15-A8</td><td>2-8, 39</td><td>O</td><td>ADDRESS BUS: These lines provide address bits 8 through 15 for the entire bus cycle (T1-T4). These lines do not have to be latched by ALE to remain valid. A15-A8 are active HIGH and are held at high impedance to the last valid logic level during interrupt acknowledge and local bus “hold acknowledge” or “grant sequence”.</td></tr>
<tr><td>A19/S6,<br>A18/S5,<br>A17/S4,<br>A16/S3</td><td>35<br>36<br>37<br>38</td><td>O<br>O<br>O<br>O</td><td>ADDRESS/STATUS: During T1, these are the four most significant address lines for memory operations. During I/O operations, these lines are LOW. During memory and I/O operations, status information is available on these lines during T2, T3, TW and T4. S6 is always LOW. The status of the interrupt enable flag bit (S5) is updated at the beginning of each clock cycle. S4 and S3 are encoded as shown.<br>This information indicates which segment register is presently being used for data accessing.<br>These lines are held at high impedance to the last valid logic level during local bus “hold acknowledge” or “grant Sequence”.
<table>
<tr><th>S4</th><th>S3</th><th>CHARACTERISTICS</th></tr>
<tr><td>0</td><td>0</td><td>Alternate Data</td></tr>
<tr><td>0</td><td>1</td><td>Stack</td></tr>
<tr><td>1</td><td>0</td><td>Code or None</td></tr>
<tr><td>1</td><td>1</td><td>Data</td></tr>
</table></td></tr>
<tr><td>$\overline{RD}$</td><td>32</td><td>O</td><td>READ: Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the IO/$\overline{M}$ pin or $\overline{S2}$. This signal is used to read devices which reside on the 80C88 local bus. $\overline{RD}$ is active LOW during T2, T3, Tw of any read cycle, and is guaranteed to remain HIGH in T2 until the 80C88 local bus has floated.<br>This line is held at a high impedance logic one state during “hold acknowledge” or “grant sequence”.</td></tr>
<tr><td>READY</td><td>22</td><td>I</td><td>READY: is the acknowledgment from the address memory or I/O device that it will complete the data transfer. The RDY signal from memory or I/O is synchronized by the 82C84A clock generator to from READY. This signal is active HIGH. The 80C88 READY input is not synchronized. Correct operation is not guaranteed if the set up and hold times are not met.</td></tr>
<tr><td>INTR</td><td>18</td><td>I</td><td>INTERRUPT REQUEST: is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.</td></tr>
<tr><td>$\overline{TEST}$</td><td>23</td><td>I</td><td>TEST: input is examined by the “wait for test” instruction. If the $\overline{TEST}$ input is LOW, execution continues, otherwise the processor waits in an “idle” state. This input is synchronized internally during each clock cycle on the leading edge of CLK.</td></tr>
<tr><td>NMI</td><td>17</td><td>I</td><td>NONMASKABLE INTERRUPT: is an edge triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.</td></tr>
<tr><td>RESET</td><td>21</td><td>I</td><td>RESET: cases the processor to immediately terminate its present activity. The signal must transition LOW to HIGH and remain active HIGH for at least four clock cycles. It restarts execution, as described in the instruction set description, when RESET returns LOW. RESET is internally synchronized.</td></tr>
<tr><td>CLK</td><td>19</td><td>I</td><td>CLOCK: provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.</td></tr>
<tr><td>$V_{CC}$</td><td>40</td><td></td><td>$V_{CC}$: is the +5V power supply pin. A 0.1μF capacitor between pins 20 and 40 recommended for decoupling.</td></tr>
<tr><td>GND</td><td>1, 20</td><td></td><td>GND: are the ground pins (both pins must be connected to system ground). A 0.1μF capacitor between pins 1 and 20 is recommended for decoupling.</td></tr>
<tr><td>MN/$\overline{MX}$</td><td>33‘</td><td>I</td><td>MINIMUM/MAXIMUM: indicates the mode in which the processor is to operate. The two modes are discussed in the following sections.</td></tr>
</table>

## Pin Description (Continued)

The following pin function descriptions are for 80C88 system in minimum mode (i.e., MN/$\overline{\text{MX}}$ = $V_{CC}$). Only the pin functions which are unique to the minimum mode are described; all other pin functions are as described above.

### MINIMUM MODE SYSTEM

<table>
<tr><th>SYMBOL</th><th>PIN NUMBER</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>IO/$\overline{\text{M}}$</td><td>28</td><td>O</td><td>STATUS LINE: is an inverted maximum mode $\overline{\text{S2}}$. It is used to distinguish a memory access from an I/O access. IO/$\overline{\text{M}}$ becomes valid in the T4 preceding a bus cycle and remains valid until the final T4 of the cycle (I/O = HIGH, M = LOW). IO/$\overline{\text{M}}$ is held to a high impedance logic one during local bus “hold acknowledge”.</td></tr>
<tr><td>$\overline{\text{WR}}$</td><td>29</td><td>O</td><td>Write: strobe indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the IO/$\overline{\text{M}}$ signal. $\overline{\text{WR}}$ is active for T2, T3, and Tw of any write cycle. It is active LOW, and is held to high impedance logic one during local bus “hold acknowledge”.</td></tr>
<tr><td>$\overline{\text{INTA}}$</td><td>24</td><td>O</td><td>INTA: is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T2, T3 and Tw of each interrupt acknowledge cycle. Note that $\overline{\text{INTA}}$ is never floated.</td></tr>
<tr><td>ALE</td><td>25</td><td>O</td><td>ADDRESS LATCH ENABLE: is provided by the processor to latch the address into the 82C82/82C83 address latch. It is a HIGH pulse active during clock low of T1 of any bus cycle. Note that ALE is never floated.</td></tr>
<tr><td>DT/$\overline{\text{R}}$</td><td>27</td><td>O</td><td>DATA TRANSMIT/RECEIVE: is needed in a minimum system that desires to use an 82C86/82C87 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically, DT/$\overline{\text{R}}$ is equivalent to $\overline{\text{S1}}$ in the maximum mode, and its timing is the same as for IO/$\overline{\text{M}}$ (T = HIGH, R = LOW). This signal is held to a high impedance logic one during local bus “hold acknowledge”.</td></tr>
<tr><td>$\overline{\text{DEN}}$</td><td>26</td><td>O</td><td>DATA ENABLE: is provided as an output enable for the 82C86/82C87 in a minimum system which uses the transceiver. $\overline{\text{DEN}}$ is active LOW during each memory and I/O access, and for $\overline{\text{INTA}}$ cycles. For a read or $\overline{\text{INTA}}$ cycle, it is active from the middle of T2 until the middle of T4, while for a write cycle, it is active from the beginning of T2 until the middle of T4. $\overline{\text{DEN}}$ is held to high impedance logic one during local bus “hold acknowledge”.</td></tr>
<tr><td>HOLD,<br>HLDA</td><td>31<br>30</td><td>I<br>O</td><td>HOLD: indicates that another master is requesting a local bus “hold”. To be acknowledged, HOLD must be active HIGH. The processor receiving the “hold” request will issue HLDA (HIGH) as an acknowledgment, in the middle of a T4 or T1 clock cycle. Simultaneous with the issuance of HLDA the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor lowers HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines.<br>Hold is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the set up time.</td></tr>
<tr><td>$\overline{\text{SS0}}$</td><td>34</td><td>O</td><td>STATUS LINE: is logically equivalent to $\overline{\text{S0}}$ in the maximum mode. The combination of $\overline{\text{SS0}}$, IO/$\overline{\text{M}}$ and DT/$\overline{\text{R}}$ allows the system to completely decode the current bus cycle status. $\overline{\text{SS0}}$ is held to high impedance logic one during local bus “hold acknowledge”.
<table>
<tr><th>IO/$\overline{\text{M}}$</th><th>DT/$\overline{\text{R}}$</th><th>$\overline{\text{SS0}}$</th><th>CHARACTERISTICS</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Interrupt Acknowledge</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Read I/O Port</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Write I/O Port</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Halt</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Code Access</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Read Memory</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Write Memory</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Passive</td></tr>
</table></td></tr>
</table>

## Pin Description (Continued)

The following pin function descriptions are for 80C88 system in maximum mode (i.e., MN/$\overline{\text{MX}}$ = GND). Only the pin functions which are unique to the maximum mode are described; all other pin functions are as described above.

**MAXIMUM MODE SYSTEM**

<table>
<tr><th>SYMBOL</th><th>PIN NUMBER</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>$\overline{\text{S0}}$<br>$\overline{\text{S1}}$<br>$\overline{\text{S2}}$</td><td>26<br>27<br>28</td><td>O<br>O<br>O</td><td>STATUS: is active during clock high of T4, T1 and T2, and is returned to the passive state (1, 1, 1) during T3 or during Tw when READY is HIGH. This status is used by the 82C88 bus controller to generate all memory and I/O access control signals. Any change by $\overline{\text{S2}}$, $\overline{\text{S1}}$ or $\overline{\text{S0}}$ during T4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T3 or Tw is used to indicate the end of a bus cycle.<br>These signals are held at a high impedance logic one state during “grant sequence”.<br>
<table>
<tr><th>$\overline{\text{S2}}$</th><th>$\overline{\text{S1}}$</th><th>$\overline{\text{S0}}$</th><th>CHARACTERISTICS</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Interrupt Acknowledge</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Read I/O Port</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Write I/O Port</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Halt</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Code Access</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Read Memory</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Write Memory</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Passive</td></tr>
</table></td></tr>
<tr><td>$\overline{\text{RQ}}$/$\overline{\text{GT0}}$,<br>$\overline{\text{RQ}}$/$\overline{\text{GT1}}$</td><td>31<br>30</td><td>I/O</td><td>REQUEST/GRANT: pins are used by other local bus masters to force the processor to release the local bus at the end of the processor’s current bus cycle. Each pin is bidirectional with $\overline{\text{RQ}}$/$\overline{\text{GT0}}$ having higher priority than $\overline{\text{RQ}}$/$\overline{\text{GT1}}$. $\overline{\text{RQ}}$/$\overline{\text{GT}}$ has internal bus-hold high circuitry and, if unused, may be left unconnected. The request/grant sequence is as follows (see $\overline{\text{RQ}}$/$\overline{\text{GT}}$ Timing Sequence):<br>1. A pulse of one CLK wide from another local bus master indicates a local bus request (“hold”) to the 80C88 (pulse 1).<br>2. During a T4 or T1 clock cycle, a pulse one clock wide from the 80C88 to the requesting master (pulse 2), indicates that the 80C88 has allowed the local bus to float and that it will enter the “grant sequence” state at the next CLK. The CPUs bus interface unit is disconnected logically from the local bus during “grant sequence”.<br>3. A pulse one CLK wide from the requesting master indicates to the 80C88 (pulse 3) that the “hold” request is about to end and that the 80C88 can reclaim the local bus at the next CLK. The CPU then enters T4 (or T1 if no bus cycles pending).<br>Each master-master exchange of the local bus is a sequence of three pulses. There must be one idle CLK cycle after bus exchange. Pulses are active LOW.<br>If the request is made while the CPU is performing a memory cycle, it will release the local bus during T4 of the cycle when all the following conjugations are met:<br>1. Request occurs on or before T2.<br>2. Current cycle is not the low bit of a word.<br>3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence.<br>4. A locked instruction is not currently executing.<br>If the local bus is idle when the request is made the two possible events will follow:<br>1. Local bus will be released during the next clock.<br>2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.</td></tr>
<tr><td>$\overline{\text{LOCK}}$</td><td>29</td><td>O</td><td>LOCK: indicates that other system bus masters are not to gain control of the system bus while $\overline{\text{LOCK}}$ is active (LOW). The $\overline{\text{LOCK}}$ signal is activated by the “LOCK” prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and is held at a high impedance logic one state during “grant sequence”. In Max Mode, $\overline{\text{LOCK}}$ is automatically generated during T2 of the first $\overline{\text{INTA}}$ cycle and removed during T2 of the second $\overline{\text{INTA}}$ cycle.</td></tr>
<tr><td>QS1, QS0</td><td>24, 25</td><td>O</td><td>QUEUE STATUS: provide status to allow external tracking of the internal 80C88 instruction queue.<br>The queue status is valid during the CLK cycle after which the queue operation is performed. Note that the queue status never goes to a high impedance statue (floated).<br>
<table>
<tr><th>QS1</th><th>QS0</th><th>CHARACTERISTICS</th></tr>
<tr><td>0</td><td>0</td><td>No Operation</td></tr>
<tr><td>0</td><td>1</td><td>First Byte of Opcode from Queue</td></tr>
<tr><td>1</td><td>0</td><td>Empty the Queue</td></tr>
<tr><td>1</td><td>1</td><td>Subsequent Byte from Queue</td></tr>
</table></td></tr>
<tr><td>-</td><td>34</td><td>O</td><td>Pin 34 is always a logic one in the maximum mode and is held at a high impedance logic one during a “grant sequence”.</td></tr>
</table>

## Functional Description

### Static Operation

All 80C88 circuitry is static in design. Internal registers, counters and latches are static and require not refresh as with dynamic circuit design. This eliminates the minimum operating frequency restriction placed on other microprocessors. The CMOS 80C88 can operate from DC to the specified upper frequency limit. The processor clock may be stopped in either state (high/low) and held there indefinitely. This type of operation is especially useful for system debug or power critical applications.

The 80C88 can be single stepped using only the CPU clock. This state can be maintained as long as is necessary. Single step clock operation allows simple interface circuitry to provide critical information for start-up.

Static design also allows very low frequency operation (as low as DC). In a power critical situation, this can provide extremely low power operation since 80C88 power dissipation is directly related to operation frequency. As the system frequency is reduced, so is the operating power until, at a DC input frequency, the power requirement is the 80C88 standby current.

### Internal Architecture

The internal functions of the 80C88 processor are partitioned logically into two processing units. The first is the Bus Interface Unit (BIU) and the second is the Execution Unit (EU) as shown in the CPU block diagram.

These units can interact directly but for the most part perform as separate asynchronous operational processors. The bus interface unit provides the functions related to instruction fetching and queuing, operand fetch and store, and address relocation. This unit also provides the basic bus control. The overlap of instruction pre-fetching provided by this unit serves to increase processor performance through improved bus bandwidth utilization. Up to 4 bytes of the instruction stream can be queued while waiting for decoding and execution.

The instruction stream queuing mechanism allows the BIU to keep the memory utilized very efficiently. Whenever there is space for at least 1 byte in the queue, the BIU will attempt a byte fetch memory cycle. This greatly reduces “dead time”: on the memory bus. The queue acts as a First-In-First-Out (FIFO) buffer, from which the EU extracts instruction bytes as required. If the queue is empty (following a branch instruction, for example), the first byte into the queue immediately becomes available to the EU.

The execution unit receives pre-fetched instructions from the BIU queue and provides unrelocated operand addresses to the BIU. Memory operands are passed through the BIU for processing by the EU, which passes results to the BIU for storage.

### Memory Organization

The processor provides a 20-bit address to memory which locates the byte being referenced. The memory is organized as a linear array of up to 1 million bytes, addressed as 00000(H) to FFFFF(H). The memory is logically divided into code, data, extra, and stack segments of up to 64K bytes each, with each segment falling on 16 byte boundaries. (See Figure 1).

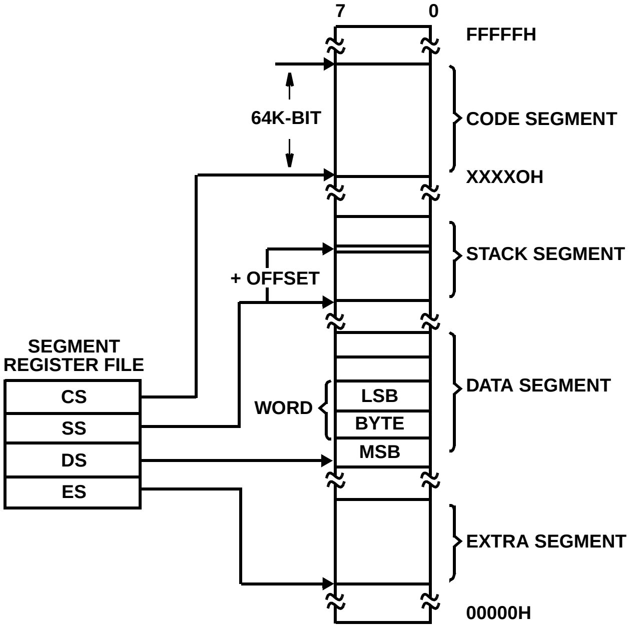


FIGURE 14. MEMORY ORGANIZATION

All memory references are made relative to base addresses contained in high speed segment registers. The segment types were chosen based on the addressing needs of programs. The segment register to be selected is automatically chosen according to specific rules as shown in Table 1. All information in one segment type share the same logical attributes (e.g., code or data). By structuring memory into relocatable areas of similar characteristics and by automatically selecting segment registers, programs are shorter, faster, and more structured.

TABLE 6.

| MEMORY REFERENCE NEED | SEGMENT REGISTER USED | SEGMENT SELECTION RULE |
|---|---|---|
| Instructions | CODE (CS) | Automatic with all instruction prefetch. |
| Stack | STACK (SS) | All stack pushes and pops. Memory references relative to BP base register except data references. |
| Local Data | DATA (DS) | Data references when: relative to stack, destination of string operation, or explicitly overridden. |
| External Data (Global) | EXTRA (ES) | Destination of string operations: Explicitly selected using a segment override. |

Word (16-bit) operands can be located on even or odd address boundaries. For address and data operands, the least significant byte of the word is stored in the lower valued address location and the most significant byte in the next higher address location.

The BIU will automatically execute two fetch or write cycles for 16-bit operands.

Certain locations in memory are reserved for specific CPU operations. (See Figure 2). Locations from addresses FFFF0H through FFFFFH are reserved for operations including a jump to initial system initialization routine. Following RESET, the CPU will always begin execution at location FFFF0H where the jump must be located. Locations 00000H through 003FFH are reserved for interrupt operations. Each of the 256 possible interrupt service routines is accessed through its own pair of 16-bit pointers - segment address pointer and offset address pointer. The first pointer, used as the offset address, is loaded into the IP, and the second pointer, which designates the base address, is loaded into the CS. At this point program control is transferred to the interrupt routine. The pointer elements are assumed to have been stored at their respective places in reserved memory prior to the occurrence of interrupts.

### Minimum and Maximum Modes

The requirements for supporting minimum and maximum 80C88 systems are sufficiently different that they cannot be done efficiently with 40 uniquely defined pins. Consequently, the 80C88 is equipped with a strap pin (MN/$\overline{\text{MX}}$) which defines the system configuration. The definition of a certain subset of the pins changes, dependent on the condition of the strap pin. When the MN/$\overline{\text{MX}}$ pin is strapped to GND, the 80C88 defines pins 24 through 31 and 34 in maximum mode. When the MN/$\overline{\text{MX}}$ pins is strapped to $V_{CC}$, the 80C88 generates bus control signals itself on pins 24 through 31 and 34.

The minimum mode 80C88 can be used with either a muliplexed or demultiplexed bus. This architecture provides the 80C88 processing power in a highly integrated form.

The demultiplexed mode requires one latch (for 64K addressability) or two latches (for a full megabyte of addressing). An 82C86 or 82C87 transceiver can also be used if data bus buffering is required. (See Figure 3). The 80C88 provides $\overline{\text{DEN}}$ and DT/$\overline{\text{R}}$ to control the transceiver, and ALE to latch the addresses. This configuration of the minimum mode provides the standard demultiplexed bus structure with heavy bus buffering and relaxed bus timing requirements.

The maximum mode employs the 82C88 bus controller (See Figure 4). The 82C88 decode status lines S0, S1 and S2, and provides the system with all bus control signals. Moving the bus control to the 82C88 provides better source and sink current capability to the control lines, and frees the 80C88 pins for extended large system features. Hardware lock, queue status, and two request/grant interfaces are provided by the 80C88 in maximum mode. These features allow coprocessors in local bus and remote bus configurations.

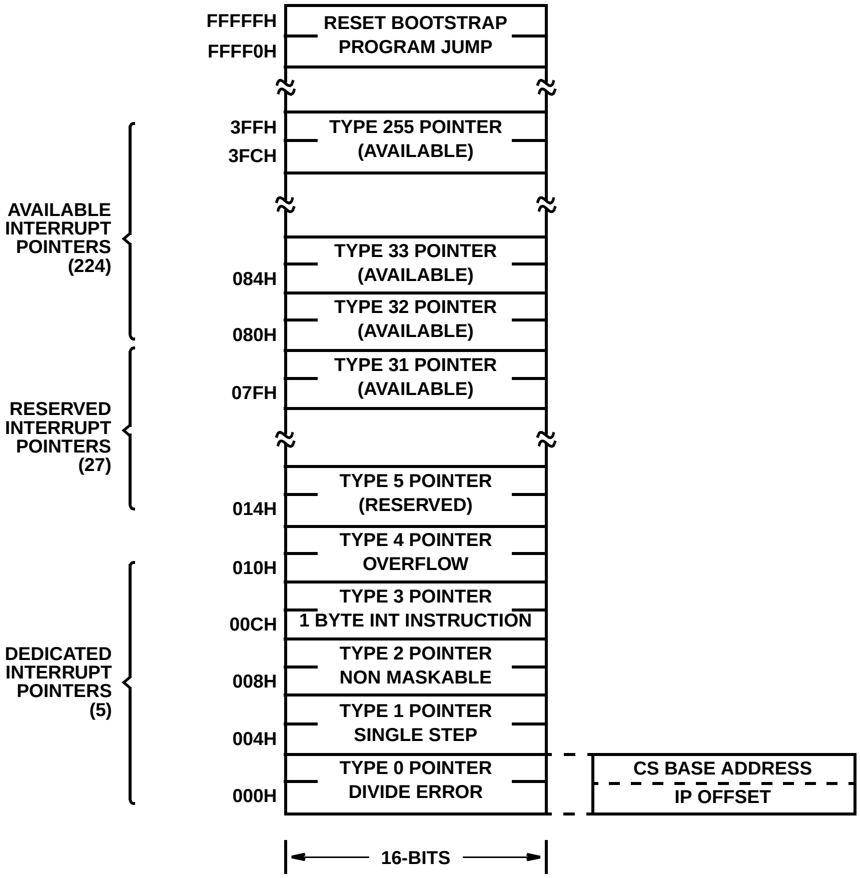

FFFFFH
FFFF0H
RESET BOOTSTRAP PROGRAM JUMP
3FFH
3FCH
TYPE 255 POINTER (AVAILABLE)
AVAILABLE INTERRUPT POINTERS (224)
TYPE 33 POINTER (AVAILABLE)
084H
TYPE 32 POINTER (AVAILABLE)
080H
TYPE 31 POINTER (AVAILABLE)
07FH
RESERVED INTERRUPT POINTERS (27)
TYPE 5 POINTER (RESERVED)
014H
TYPE 4 POINTER OVERFLOW
010H
TYPE 3 POINTER 1 BYTE INT INSTRUCTION
00CH
DEDICATED INTERRUPT POINTERS (5)
TYPE 2 POINTER NON MASKABLE
008H
TYPE 1 POINTER SINGLE STEP
004H
TYPE 0 POINTER DIVIDE ERROR
000H
CS BASE ADDRESS
IP OFFSET
16-BITS

FIGURE 15. RESERVED MEMORY LOCATIONS

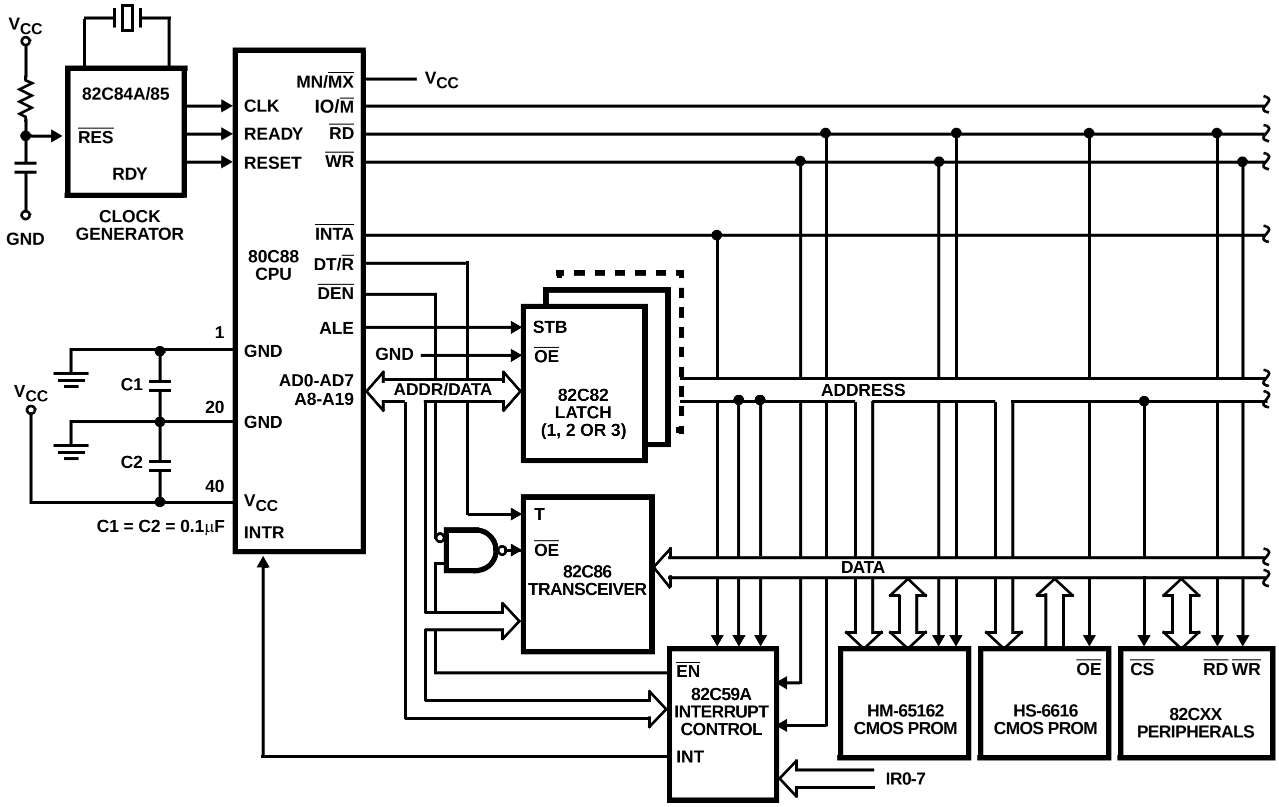


FIGURE 16. DEMULTIPLEXED BUS CONFIGURATION

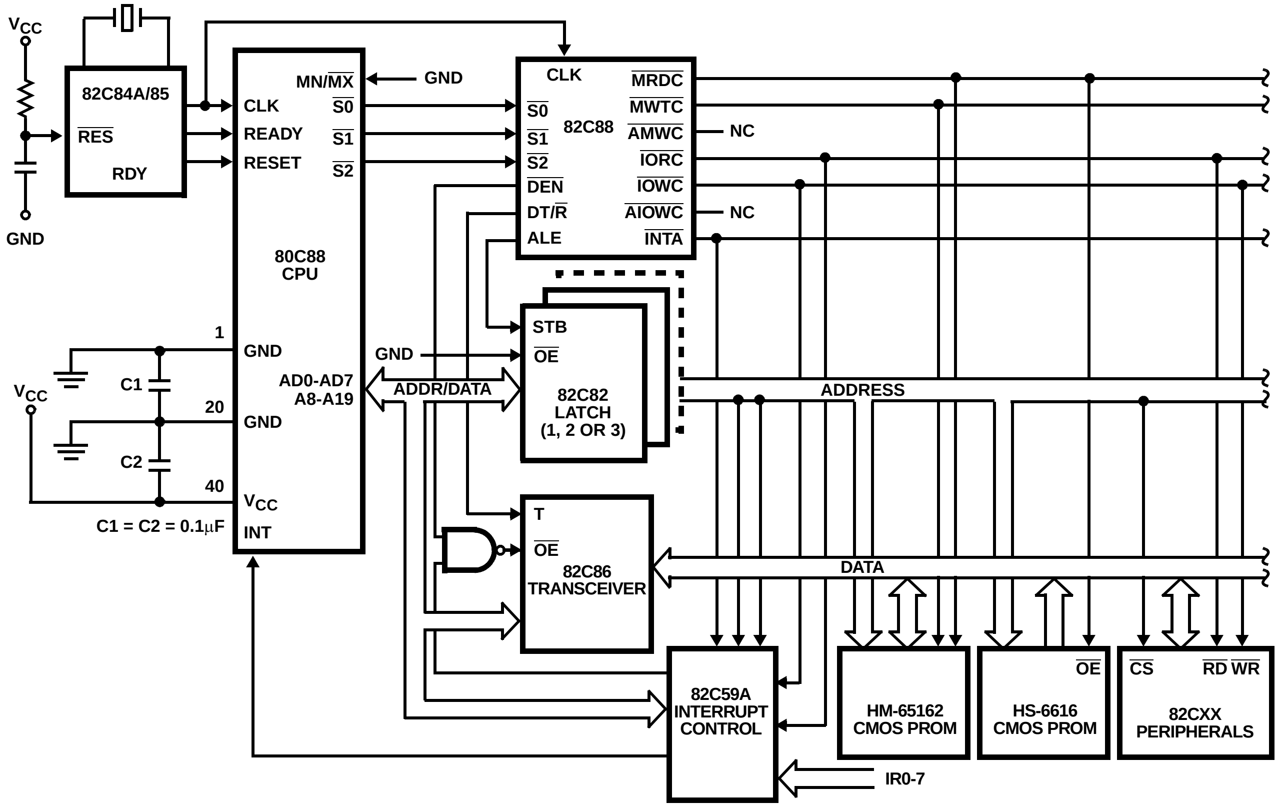


FIGURE 17. FULLY BUFFERED SYSTEM USING BUS CONTROLLER

### Bus Operation

The 80C88 address/data bus is broken into three parts: the lower eight address/data bits (AD0-AD7), the middle eight address bits (A8-A15), and the upper four address bits (A16-A19). The address/data bits and the highest four address bits are time multiplexed. This technique provides the most efficient use of pins on the processor, permitting the use of standard 40 lead package. The middle eight address bits are not multiplexed, i.e., they remain valid throughout each bus cycle. In addition, the bus can be demultiplexed at the processor with a single address latch if a standard, nonmultiplexed bus is desired for the system.

Each processor bus cycle consists of at least four CLK cycles. These are referred to as T1, T2, T3 and T4. (See Figure 5). The address is emitted from the processor during T1 and data transfer occurs on the bus during T3 and T4. T2 is used primarily for changing the direction of the bus during read operations. In the event that a "Not Ready" indication is given by the addressed device, "wait" states (TW) are inserted between T3 and T4. Each inserted "wait" state is of the same duration as a CLK cycle. Periods can occur between 80C88 driven bus cycles. These are referred to as "idle" states (TI), or inactive CLK cycles. The processor uses these cycles for internal housekeeping.

During T1 of any bus cycle, the ALE (Address latch enable) signal is emitted (by either the processor or the 82C88 bus controller, depending on the MN/$\overline{\text{MX}}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{\text{S0}}$, $\overline{\text{S1}}$, and $\overline{\text{S2}}$ are used by the bus controller, in maximum mode, to identify the type of bus transaction according to Table 2.

Status bits S3 through S6 are multiplexed with high order address bits and are therefore valid during T2 through T4. S3 and S4 indicate which segment register was used to this bus cycle in forming the address according to Table 3.

S5 is a reflection of the PSW interrupt enable bit. S6 is always equal to 0.

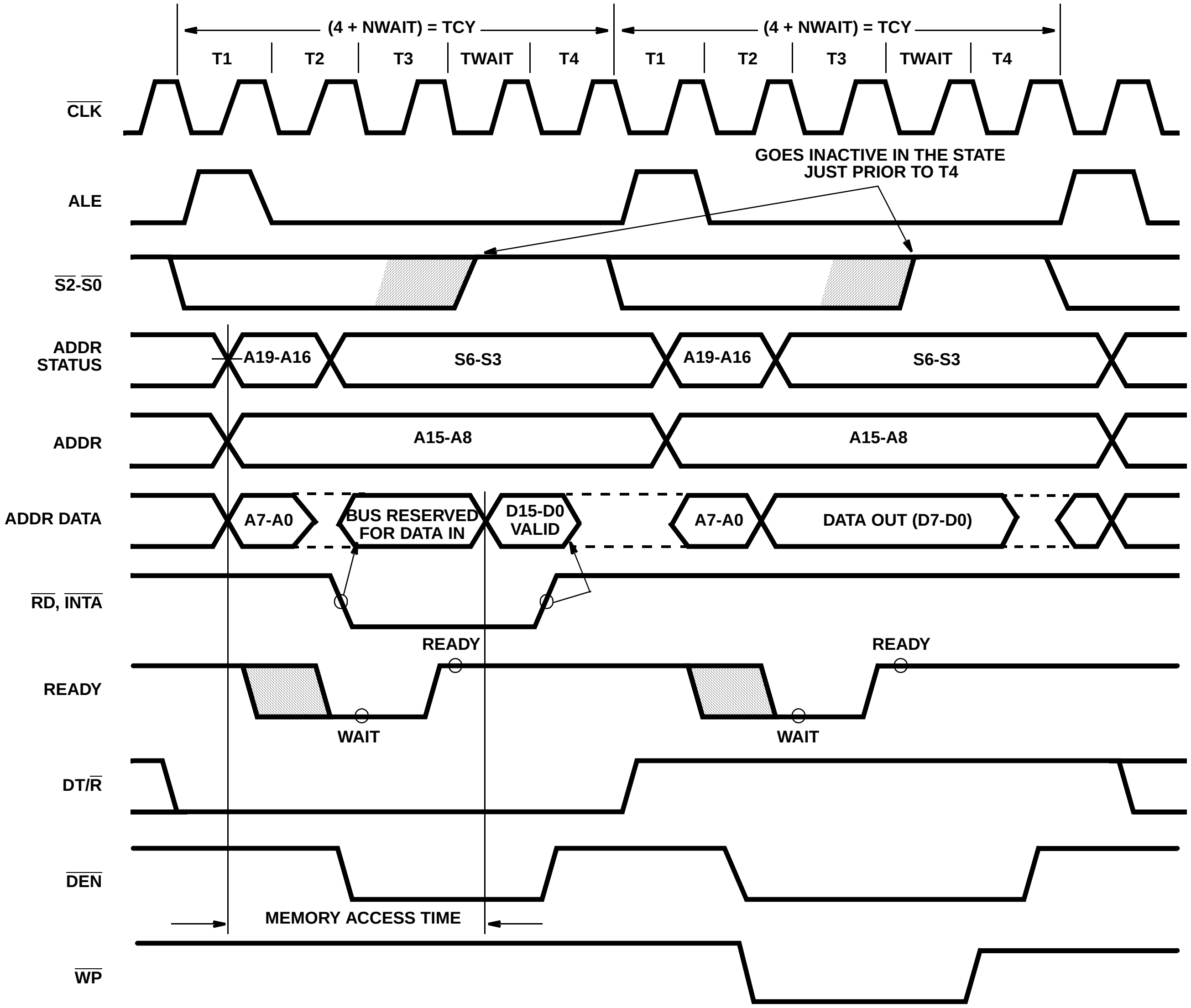


FIGURE 18. BASIC SYSTEM TIMING

TABLE 7.

| $\overline{S2}$ | $\overline{S1}$ | $\overline{S0}$ | CHARACTERISTICS |
|---|---|---|---|
| 0 | 0 | 0 | Interrupt Acknowledge |
| 0 | 0 | 1 | Read I/O |
| 0 | 1 | 0 | Write I/O |
| 0 | 1 | 1 | Halt |
| 1 | 0 | 0 | Instruction Fetch |
| 1 | 0 | 1 | Read Data from Memory |
| 1 | 1 | 0 | Write Data to Memory |
| 1 | 1 | 1 | Passive (No Bus Cycle) |

TABLE 8.

| S4 | S3 | CHARACTERISTICS |
|---|---|---|
| 0 | 0 | Alternate Data (Extra Segment) |
| 0 | 1 | Stack |
| 1 | 0 | Code or None |
| 1 | 1 | Data |

### I/O Addressing

In the 80C88, I/O operations can address up to a maximum of 64K I/O registers. The I/O address appears in the same format as the memory address on bus lines A15-A0. The address lines A19-A16 are zero in I/O operations. The variable I/O instructions, which use register DX as a pointer, have full address capability, while the direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

Designers familiar with the 8085 or upgrading an 8085 design should note that the 8085 addresses I/O with an 8-bit address on both halves of the 16-bit address bus. The 80C88 uses a full 16-bit address on its lower 16 address lines.

## External Interface

### Processor Reset and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 80C88 RESET is required to be HIGH for greater than four clock cycles. The 80C88 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 7 clock cycles. After this interval the 80C88 operates normally, beginning with the instruction in absolute location FFFFOH (see Figure 2). The RESET input is internally synchronized to the processor clock. At initialization, the HIGH to LOW transition of RESET must occur no sooner than 50µs after power up, to allow complete initialization of the 80C88.

NMI will not be recognized if asserted prior to the second CLK cycle following the end of RESET.

### Bus Hold Circuitry

To avoid high current conditions caused by floating inputs to CMOS devices and to eliminate the need for pull-up/down resistors, “bus-hold” circuitry has been used on 80C88 pins 2-16, 26-32 and 34-39 (see Figure 6A and 6B). These circuits maintain a valid logic state if no driving source is present (i.e., an unconnected pin or a driving source which goes to a high impedance state).

To override the “bus hold” circuits, an external driver must be capable of supplying 400µA minimum sink or source current at valid input voltage levels. Since this “bus hold” circuitry is active and not a “resistive” type element, the associated power supply current is negligible. Power dissipation is significantly reduced when compared to the use of passive pull-up resistors.

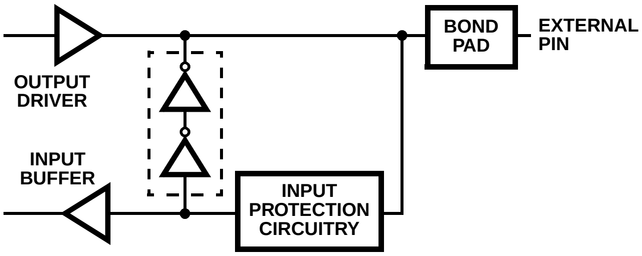


FIGURE 19A. BUS HOLD CIRCUITRY PIN 2-16, 35-39

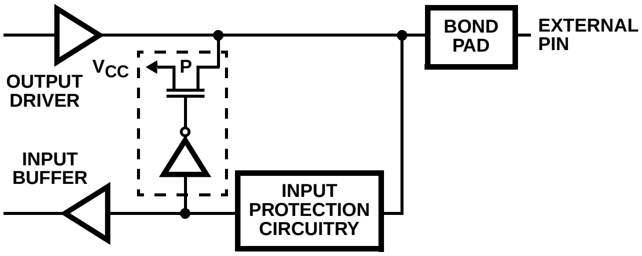


FIGURE 19B. BUS HOLD CIRCUITRY PIN 26-32, 34

### Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are specified in the instruction set description. Hardware interrupts can be classified as nonmaskable or maskable.

Interrupts result in a transfer of control to a new program location. A 256 element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (see Figure 2), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt “type”. An interrupting device supplies an 8-bit type number, during the interrupt acknowledge sequence, which is used to vector through the appropriate element to the new interrupt service program location.

### Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt (NMI) pin which has higher priority than the maskable interrupt request (INTR) pin. A typical use would be to activate a power failure routine. The NMI is edge-triggered on a LOW to High transition. The activation of this pin causes a type 2 interrupt.

NMI is required to have a duration in the HIGH state of greater than two clock cycles, but is not required to be synchronized to the clock. An high going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves (2 bytes in the case of word moves) of a block type instruction. Worst case response to NMI would be for multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during, or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure.

The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

### Maskable Interrupt (INTR)

The 80C88 provides a singe interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable (IF) flag bit. The interrupt request signal is level triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK.

To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block type instruction. INTR may be removed anytime after the falling edge of the first INTA signal. During interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single step). The FLAGS register, which is automatically pushed onto the stack, reflects the state of the processor prior to the interrupt. The enable bit will be zero until the old FLAGS register is restored, unless specifically set by an instruction.

During the response sequence (see Figure 7), the processor executes two successive (back-to-back) interrupt acknowledge cycles. The 80C88 emits to LOCK signal (maximum mode only) from T2 of the first bus cycle until T2 of the second. A local bus “hold” request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interrupt system (e.g., 82C59A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table.

An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. INTR may be removed anytime after the falling edge of the first INTA signal. The interrupt return instruction includes a flags pop which returns the status of the original interrupt enable bit when it restores the flags.

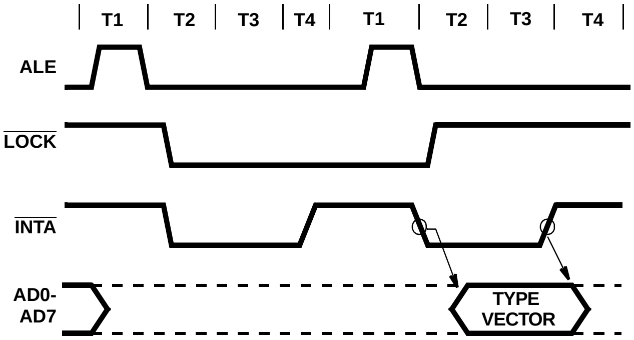


FIGURE 20. INTERRUPT ACKNOWLEDGE SEQUENCE

### Halt

When a software HALT instruction is executed, the processor indicates that it is entering the HALT state in one of two ways, depending upon which mode is strapped. In minimum mode, the processor issues ALE, delayed by one clock cycle, to allow the system to latch the halt status. Halt status is available on IO/M, DT/R, and SS0. In maximum mode, the processor issues appropriate HALT status on S2, S1 and S0, and the 82C88 bus controller issues one ALE. The 80C88 will not leave the HALT state when a local bus hold is entered while in HALT. In this case, the processor reissues the HALT indicator at the end of the local bus hold. An interrupt request or RESET will force the 80C88 out of the HALT state.

### Read/Modify/Write (Semaphore) Operations Via LOCK

The LOCK status information is provided by the processor when consecutive bus cycles are required during the execution of an instruction. This allows the processor to perform read/modify/write operations on memory (via the “exchange register with memory” instruction), without another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish “test and set lock” operations. The LOCK signal is activated (LOW) in the clock cycle following decoding of the LOCK prefix instruction. It is deactivated at the end of the last bus cycle of the instruction following the LOCK prefix. While LOCK is active, a request on a RQ/GT pin will be recorded, and then honored at the end of the LOCK.

### External Synchronization Via TEST

As an alternative to interrupts, the 80C88 provides a single software-testable input pin (TEST). This input is utilized by executing a WAIT instruction. The single WAIT instruction is repeatedly executed until the TEST input goes active (LOW). The execution of WAIT does not consume bus cycles once the queue is full.

If a local bus request occurs during WAIT execution, the 80C88 three-states all output drivers while inputs and I/O pins are held at valid logic levels by internal bus-hold circuits. If interrupts are enabled, the 80C88 will recognize interrupts and process them when it regains control of the bus.

### Basic System Timing

In minimum mode, the MN/$\overline{\text{MX}}$ pin is strapped to $V_{CC}$ and the processor emits bus control signals ($\overline{\text{RD}}$, $\overline{\text{WR}}$, IO/$\overline{\text{M}}$, etc.) directly. In maximum mode, the MN/$\overline{\text{MX}}$ pin is strapped to GND and the processor emits coded status information which the 82C88 bus controller uses to generate MULTIBUS™ compatible bus control signals.

### System Timing - Minimum System

The read cycle begins in T1 with the assertion of the address latch enable (ALE) signal (see Figure 5). The trailing (low going) edge of this signal is used to latch the address information, which is valid on the address data bus (ADO-AD7) at this time, into the 82C82/82C83 latch. Address lines A8 through A15 do not need to be latched because they remain valid throughout the bus cycle. From T1 to T4 the IO/$\overline{\text{M}}$ signal indicates a memory or I/O operation. At T2 the address is removed from the address data bus and the bus is held at the last valid logic state by internal bus-hold devices. The read control signal is also asserted at T2. The read ($\overline{\text{RD}}$) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later, valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again three-state its bus drivers. If a transceiver (82C86/82C87) is required to buffer the local bus, signals DT/$\overline{\text{R}}$ and $\overline{\text{DEN}}$ are provided by the 80C88.

A write cycle also begins with the assertion of ALE and the emission of the address. The IO/$\overline{\text{M}}$ signal is again asserted to indicate a memory or I/O write operation. In T2, immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until at least the middle of T4. During T2, T3, and Tw, the processor asserts the write control signal. The write ($\overline{\text{WR}}$) signal becomes active at the beginning of T2, as opposed to the read, which is delayed somewhat into T2 to provide time for output drivers to become inactive.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge ($\overline{\text{INTA}}$) signal is asserted in place of the read ($\overline{\text{RD}}$) signal and the address bus is held at the last valid logic state by internal bus-hold devices (see Figure 6). In the second of two successive $\overline{\text{INTA}}$ cycles, a byte of information is read from the data bus, as supplied by the interrupt system logic (i.e., 82C59A priority interrupt controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into the interrupt vector lookup table, as described earlier.

### Bus Timing - Medium Complexity Systems

For medium complexity systems, the MN/$\overline{\text{MX}}$ pin is connected to GND and the 82C88 bus controller is added to the system, as well as an 82C82/82C83 latch for latching the system address, and an 82C86/82C87 transceiver to allow for bus loading greater than the 80C88 is capable of handling (see Figure 8). Signals ALE, $\overline{\text{DEN}}$, and DT/$\overline{\text{R}}$ are generated by the 82C88 instead of the processor in this configuration, although their timing remains relatively the same. The 80C88 status outputs ($\overline{\text{S2}}$, $\overline{\text{S1}}$ and $\overline{\text{S0}}$) provide type of cycle information and become 82C88 inputs. This bus cycle information specifies read (code, data or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 82C88 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 82C88 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data is not valid at the leading edge of write. The 82C86/82C87 transceiver receives the usual T and $\overline{\text{OE}}$ inputs from the 82C88 DT/$\overline{\text{R}}$ and $\overline{\text{DEN}}$ outputs.

The pointer into the interrupt vector table, which is passed during the second $\overline{\text{INTA}}$ cycle, can derive from an 82C59A located on either the local bus or the system bus. If the master 82C59A priority interrupt controller is positioned on the local bus, the 82C86/82C87 transceiver must be disabled when reading from the master 82C59A during the interrupt acknowledge sequence and software "poll".

### The 80C88 Compared to the 80C86

The 80C88 CPU is a 8-bit processor designed around the 8086 internal structure. Most internal functions of the 80C88 are identical to the equivalent 80C86 functions. The 80C88 handles the external bus the same way the 80C86 does with the distinction of handling only 8-bits at a time. Sixteen-bit operands are fetched or written in two consecutive bus cycles. Both processors will appear identical to the software engineer, with the exception of execution time. The internal register structure is identical and all instructions have the same end result. Internally, there are three differences between the 80C88 and the 80C86. All changes are related to the 8-bit bus interface.

- The queue length is 4 bytes in the 80C88, whereas the 80C86 queue contains 6 bytes, or three words. The queue was shortened to prevent overuse of the bus by the BIU when prefetching instructions. This was required because of the additional time necessary to fetch instructions 8-bits at a time.
- To further optimize the queue, the prefetching algorithm was changed. The 80C88 BIU will fetch a new instruction to load into the queue each time there is a 1 byte space available in the queue. The 80C86 waits until a 2 byte space is available.
- The internal execution time of the instruction set is affected by the 8-bit interface. All 16-bit fetches and writes from/to memory take an additional four clock cycles. The CPU is also limited by the speed of instruction fetches. This latter problem only occurs when a series of simple operations occur. When the more sophisticated instructions of the 80C88 are being used, the queue has time to fill the execution proceeds as fast as the execution unit will allow.

MULTIBUS™ is a patented Intel bus.

The 80C88 and 80C86 are completely software compatible by virtue of their identical execution units. Software that is system dependent may not be completely transferable, but software that is not system dependent will operate equally as well on an 80C88 or an 80C86.

The hardware interface of the 80C88 contains the major differences between the two CPUs. The pin assignments are nearly identical, however, with the following functional changes:

- A8-A15: These pins are only address outputs on the 80C88. These address lines are latched internally and remain valid throughout a bus cycle in a manner similar to the 8085 upper address lines.
- $\overline{BHE}$ has no meaning on the 80C88 and has been eliminated.
- $\overline{SS0}$ provides the $\overline{S0}$ status information in the minimum mode. This output occurs on pin 34 in minimum mode only. DT/$\overline{R}$, IO/$\overline{M}$ and $\overline{SS0}$ provide the complete bus status in minimum mode.
- IO/$\overline{M}$ has been inverted to be compatible with the 8085 bus structure.
- ALE is delayed by one clock cycle in the minimum mode when entering HALT, to allow the status to be latched with

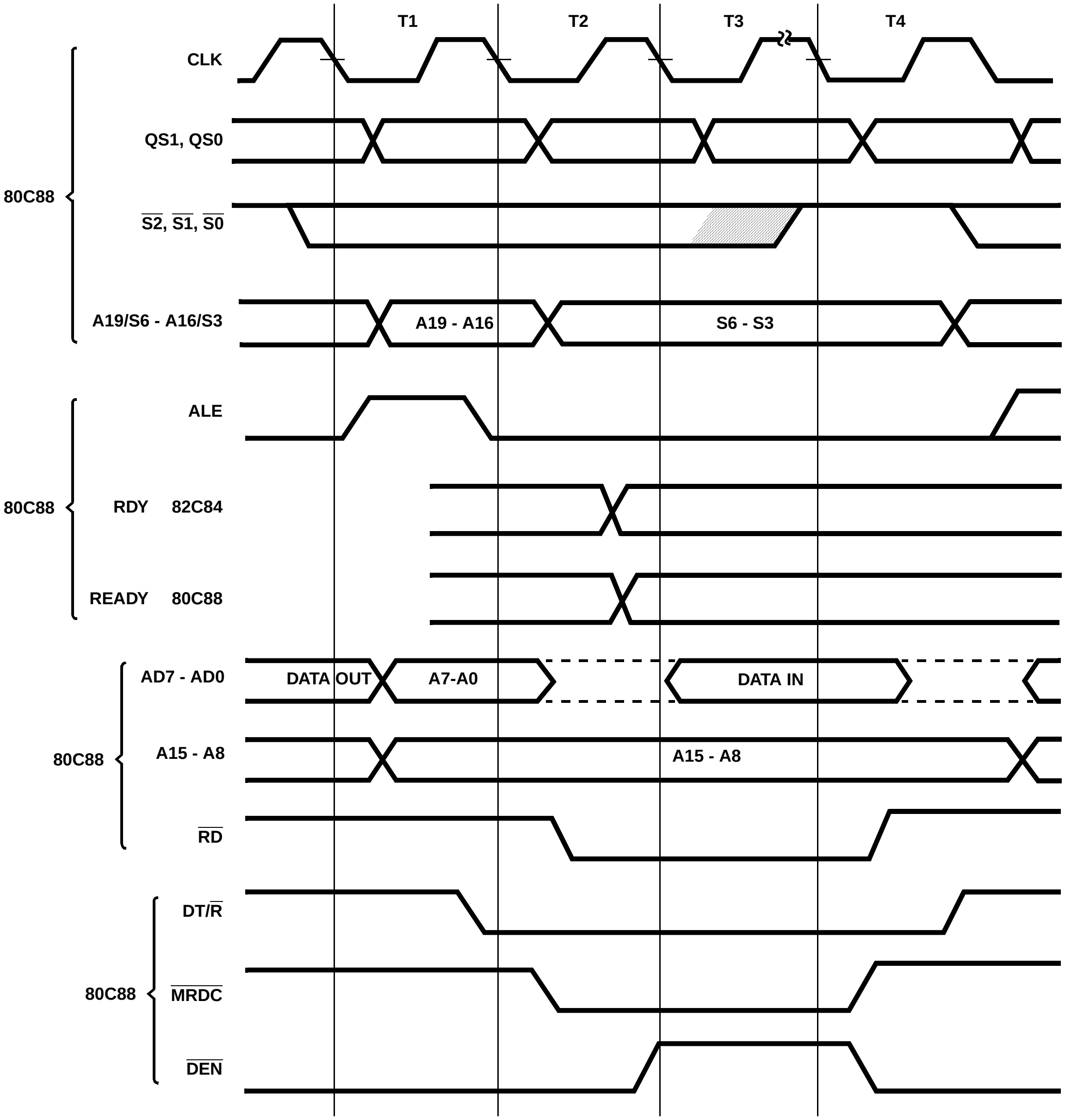


FIGURE 21. MEDIUM COMPLEXITY SYSTEM TIMING

## Absolute Maximum Ratings

| | |
|---|---|
| Supply Voltage | +8.0V |
| Input, Output or I/O Voltage | GND -0.5V to $V_{CC}$ +0.5V |
| ESD Classification | Class 1 |

## Operating Conditions

| | |
|---|---|
| Operating Voltage Range | +4.5V to +5.5V |
| M80C88-2 Only | +4.75V to +5.25V |
| Operating Temperature Range | |
| C80C88/-2 | 0°C to +70°C |
| I80C88/-2 | -40°C to +85°C |
| M80C88 | -55°C to +125°C |

## Thermal Information

| Thermal Resistance (Typical) | $\theta_{JA}$ (°C/W) |
|---|---|
| PDIP Package | 50 |
| CERDIP Package | 30 |
| Maximum Junction Temperature | |
| Ceramic Package | +175°C |
| Plastic Package | +150°C |
| Storage Temperature Range | -65°C to +150°C |
| Maximum Lead Temperature (Soldering 10s) | +300°C |

## Die Characteristics

| | |
|---|---|
| Gate Count | 9750 Gates |

*CAUTION: Stresses above those listed in “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.*

## DC Electrical Specifications

$V_{CC}$ = 5.0V, ±10%; $T_A$ = 0°C to +70°C (C80C88, C80C88-2)
$V_{CC}$ = 5.0V, ±10%; $T_A$ = -40°C to +85°C (I80C88, I80C88-2)
$V_{CC}$ = 5.0V, ±10%; $T_A$ = -55°C to +125°C (M80C88)

| SYMBOL | PARAMETER | MIN | MAX | UNITS | TEST CONDITION |
|---|---|---|---|---|---|
| $V_{IH}$ | Logical One Input Voltage | 2.0<br>2.2 | - | V<br>V | C80C88, I80C88 (Note 4)<br>M80C88 (Note 4) |
| $V_{IL}$ | Logical Zero Input Voltage | - | 0.8 | V | |
| VIHC | CLK Logical One Input Voltage | $V_{CC}$ -0.8 | - | V | |
| VILC | CLK Logical Zero Input Voltage | - | 0.8 | V | |
| $V_{OH}$ | Output High Voltage | 3.0<br>$V_{CC}$ -0.4 | - | V<br>V | IOH = -2.5mA<br>IOH = -100µA |
| $V_{OL}$ | Output Low Voltage | - | 0.4 | V | IOL = +2.5mA |
| $I_I$ | Input Leakage Current | -1.0 | 1.0 | µA | $V_{IN}$ = 0V or $V_{CC}$ Pins 17-19, 21-23, 33 |
| IBHH | Input Current-Bus Hold High | -40 | -400 | µA | $V_{IN}$ = - 3.0V (Note 1) |
| IBHL | Input Current-Bus Hold Low | 40 | 400 | µA | $V_{IN}$ = - 0.8V (Note 2) |
| $I_O$ | Output Leakage Current | - | -10.0 | µA | $V_{OUT}$ = 0V (Note 5) |
| ICCSB | Standby Power Supply Current | - | 500 | µA | $V_{CC}$ = 5.5V (Note 3) |
| ICCOP | Operating Power Supply Current | - | 10 | mA/MHz | FREQ = Max, $V_{IN}$ = $V_{CC}$ or GND, Outputs Open |

NOTES:

1. IBHH should be measured after raising $V_{IN}$ to $V_{CC}$ and then lowering to 3.0V on the following pins 2-16, 26-32, 34-39.
2. IBHL should be measured after lowering $V_{IN}$ to GND and then raising to 0.8V on the following pins: 2-16, 35-39.
3. ICCSB tested during clock high time after HALT instruction executed. $V_{IN}$ = $V_{CC}$ or GND, $V_{CC}$ = 5.5V, Outputs unloaded.
4. MN/$\overline{\text{MX}}$ is a strap option and should be held to $V_{CC}$ or GND.
5. IO should be measured by putting the pin in a high impedance state and then driving $V_{OUT}$ to GND on the following pins: 26-29 and 32.

## Capacitance $T_A$ = 25°C

| SYMBOL | PARAMETER | TYPICAL | UNITS | TEST CONDITIONS |
|---|---|---|---|---|
| CIN | Input Capacitance | 25 | pF | FREQ = 1MHz. All measurements are referenced to device GND |
| COUT | Output Capacitance | 25 | pF | FREQ = 1MHz. All measurements are referenced to device GND |
| CI/O | I/O Capacitance | 25 | pF | FREQ = 1MHz. All measurements are referenced to device GND |

## AC Electrical Specifications

$V_{CC}$ = 5.0V ±10%; $T_A$ = $0^oC$ to $+70^oC$ (C80C88, C80C88-2)
$V_{CC}$ = 5.0V ±10%; $T_A$ = $-40^oC$ to $+85^oC$ (I80C88, I80C88-2)
$V_{CC}$ = 5.0V ±10%; $T_A$ = $-55^oC$ to $+125^oC$ (M80C88)

**MINIMUM COMPLEXITY SYSTEM**

| SYMBOL | | PARAMETER | 80C88 | | 80C88-2 | | UNITS | TEST CONDITIONS |
|---|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | | |
| **TIMING REQUIREMENTS** | | | | | | | | |
| (1) | TCLCL | CLK Cycle Period | 200 | - | 125 | - | ns | |
| (2) | TCLCH | CLK Low Time | 118 | - | 68 | - | ns | |
| (3) | TCHCL | CLK High Time | 69 | - | 44 | - | ns | |
| (4) | TCH1CH2 | CLK Rise Time | - | 10 | - | 10 | ns | From 1.0V to 3.5V |
| (5) | TCL2CL1 | CLK Fall Time | - | 10 | - | 10 | ns | From 3.5V to 1.0V |
| (6) | TDVCL | Data In Setup Time | 30 | - | 20 | - | ns | |
| (7) | TCLDX1 | Data In Hold Time | 10 | - | 10 | - | ns | |
| (8) | TR1VCL | RDY Setup Time into 82C84A (Notes 6, 7) | 35 | - | 35 | - | ns | |
| (9) | TCLR1X | RDY Hold Time into 82C84A (Notes 6, 7) | 0 | - | 0 | - | ns | |
| (10) | TRYHCH | READY Setup Time into 80C88 | 118 | - | 68 | - | ns | |
| (11) | TCHRYX | READY Hold Time into 80C88 | 30 | - | 20 | - | ns | |
| (12) | TRYLCL | READY Inactive to CLK (Note 8) | -8 | - | -8 | - | ns | |
| (13) | THVCH | HOLD Setup Time | 35 | - | 20 | - | ns | |
| (14) | TINVCH | INTR, NMI, $\overline{\text{TEST}}$ Setup Time (Note 7) | 30 | - | 15 | - | ns | |
| (15) | TILIH | Input Rise Time (Except CLK) | - | 15 | - | 15 | ns | From 0.8V to 2.0V |
| (16) | TIHIL | Input Fall Time (Except CLK) | - | 15 | - | 15 | ns | From 2.0V to 0.8V |
| **TIMING RESPONSES** | | | | | | | | |
| (17) | TCLAV | Address Valid Delay | 10 | 110 | 10 | 60 | ns | CL = 100pF |
| (18) | TCLAX | Address Hold Time | 10 | - | 10 | - | ns | CL = 100pF |
| (19) | TCLAZ | Address Float Delay | TCLAX | 80 | TCLAX | 50 | ns | CL = 100pF |
| (20) | TCHSZ | Status Float Delay | - | 80 | - | 50 | ns | CL = 100pF |
| (21) | TCHSV | Status Active Delay | 10 | 110 | 10 | 60 | ns | CL = 100pF |
| (22) | TLHLL | ALE Width | TCLCH-20 | - | TCLCH-10 | - | ns | CL = 100pF |
| (23) | TCLLH | ALE Active Delay | - | 80 | - | 50 | ns | CL = 100pF |
| (24) | TCHLL | ALE Inactive Delay | - | 85 | - | 55 | ns | CL = 100pF |
| (25) | TLLAX | Address Hold Time to ALE Inactive | TCHCL-10 | - | TCHCL-10 | - | ns | CL = 100pF |

## AC Electrical Specifications

$V_{CC}$ = 5.0V ±10%; $T_A$ = $0^oC$ to $+70^oC$ (C80C88, C80C88-2)
$V_{CC}$ = 5.0V ±10%; $T_A$ = $-40^oC$ to $+85^oC$ (I80C88, I80C88-2)
$V_{CC}$ = 5.0V ±10%; $T_A$ = $-55^oC$ to $+125^oC$ (M80C88)

**MINIMUM COMPLEXITY SYSTEM**

| SYMBOL | | PARAMETER | 80C88 | | 80C88-2 | | UNITS | TEST CONDITIONS |
|---|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | | |
| (26) | TCLDV | Data Valid Delay | 10 | 110 | 10 | 60 | ns | CL = 100pF |
| (27) | TCLDX2 | Data Hold Time | 10 | - | 10 | - | ns | CL = 100pF |
| (28) | TWHDX | Data Hold Time After $\overline{\text{WR}}$ | TCLCL-30 | - | TCLCL-30 | - | ns | CL = 100pF |
| (29) | TCVCTV | Control Active Delay 1 | 10 | 110 | 10 | 70 | ns | CL = 100pF |
| (30) | TCHCTV | Control Active Delay 2 | 10 | 110 | 10 | 60 | ns | CL = 100pF |
| (31) | TCVCTX | Control Inactive Delay | 10 | 110 | 10 | 70 | ns | CL = 100pF |
| (32) | TAZRL | Address Float to READ Active | 0 | - | 0 | - | ns | CL = 100pF |
| (33) | TCLRL | $\overline{\text{RD}}$ Active Delay | 10 | 165 | 10 | 100 | ns | CL = 100pF |
| (34) | TCLRH | $\overline{\text{RD}}$ Inactive Delay | 10 | 150 | 10 | 80 | ns | CL = 100pF |
| (35) | TRHAV | $\overline{\text{RD}}$ Inactive to Next Address Active | TCLCL-45 | - | TCLCL-40 | - | ns | CL = 100pF |
| (36) | TCLHAV | HLDA Valid Delay | 10 | 160 | 10 | 100 | ns | CL = 100pF |
| (37) | TRLRH | $\overline{\text{RD}}$ Width | 2TCLCL-75 | - | 2TCLCL-50 | - | ns | CL = 100pF |
| (38) | TWLWH | $\overline{\text{WR}}$ Width | 2TCLCL-60 | - | 2TCLCL-40 | - | ns | CL = 100pF |
| (39) | TAVAL | Address Valid to ALE Low | TCLCH-60 | - | TCLCH-40 | - | ns | CL = 100pF |
| (40) | TOLOH | Output Rise Time | - | 15 | - | 15 | ns | From 0.8V to 2.0V |
| (41) | TOHOL | Output Fall Time | - | 15 | - | 15 | ns | From 2.0V to 0.8V |

NOTES:

6. Signal at 82C84A shown for reference only.
7. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
8. Applies only to T2 state (8ns into T3).

## *Waveforms*

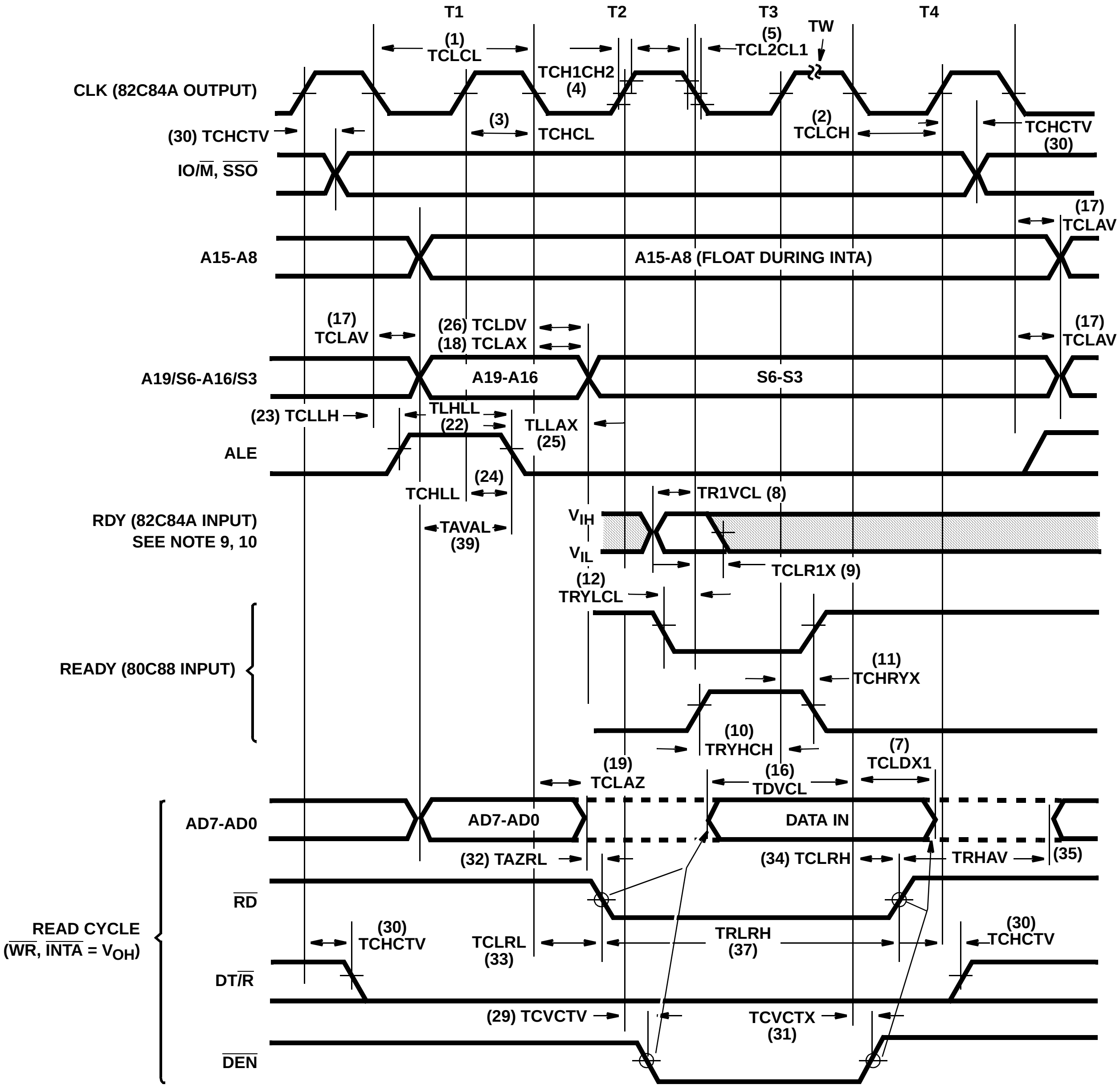


**FIGURE 22. BUS TIMING - MINIMUM MODE SYSTEM**

NOTES:

9. RDY is sampled near the end of T2, T3, TW to determine if TW machine states are to be inserted.
10. Signals at 82C84A are shown for reference only.

## Waveforms (Continued)

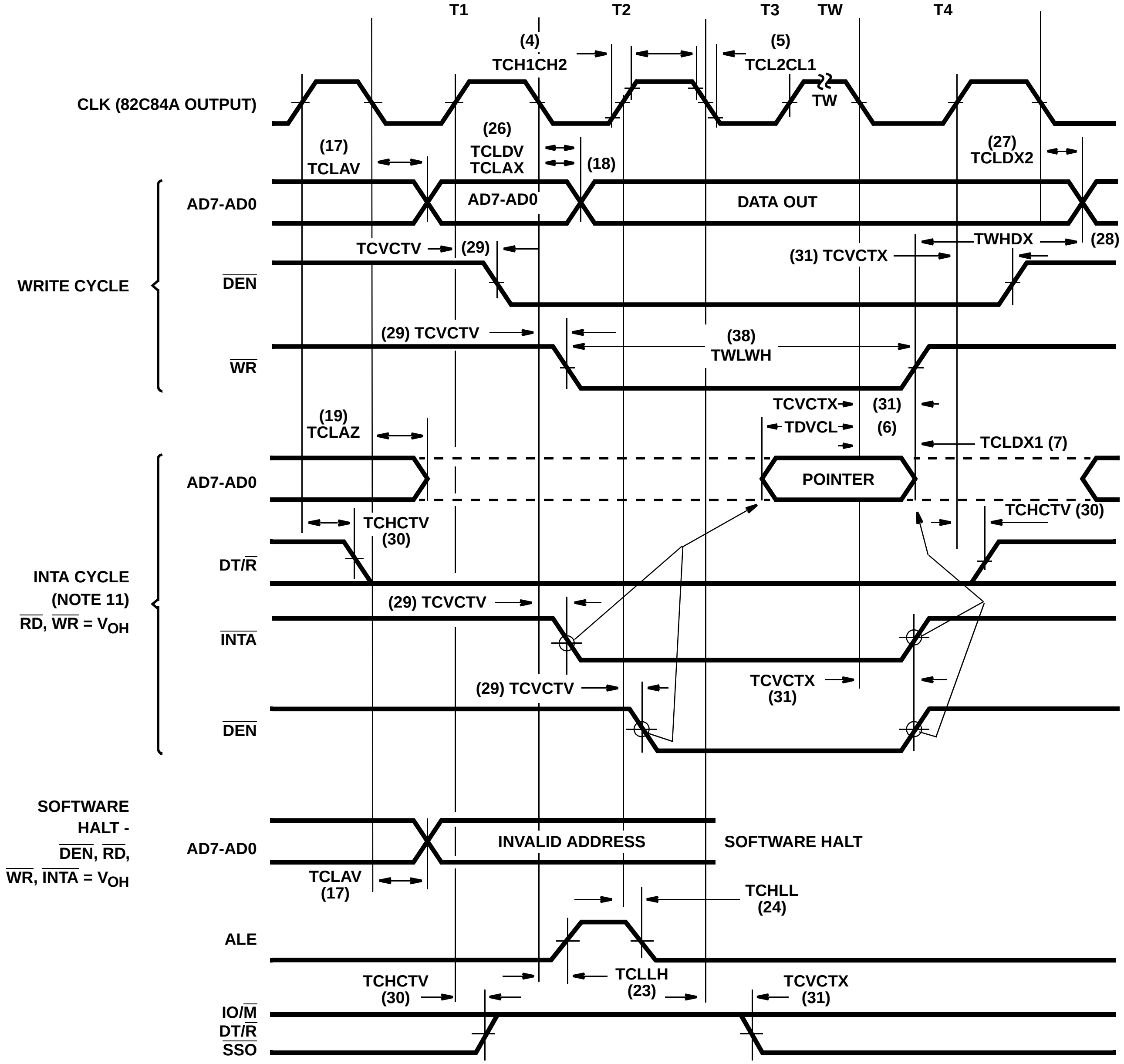


FIGURE 23. BUS TIMING - MINIMUM MODE SYSTEM (Continued)

NOTES:

11. Two INTA cycles run back-to-back. The 80C88 local ADDR/DATA bus is floating during both INTA cycles. Control signals are shown for the second INTA cycle.
12. Signals at 82C84A are shown for reference only.

## AC Electrical Specifications

$V_{CC}$ = 5.0V ±10%; $T_A$ = $0^oC$ to $+70^oC$ (C80C88, C80C88-2)
$V_{CC}$ = 5.0V ±10%; $T_A$ = $-40^oC$ to $+85^oC$ (I80C88, I80C88-2)
$V_{CC}$ = 5.0V ±10%; $T_A$ = $-55^oC$ to $+125^oC$ (M80C88)

**MAX MODE SYSTEM (USING 82C88 BUS CONTROLLER)**

| | | | 80C88 | | 80C88-2 | | | |
|---|---|---|---|---|---|---|---|---|
| SYMBOL | | PARAMETER | MIN | MAX | MIN | MAX | UNITS | TEST CONDITIONS |
| **TIMING REQUIREMENTS** | | | | | | | | |
| (1) | TCLCL | CLK Cycle Period | 200 | - | 125 | - | ns | |
| (2) | TCLCH | CLK Low Time | 118 | - | 68 | - | ns | |
| (3) | TCHCL | CLK High Time | 69 | - | 44 | - | ns | |
| (4) | TCH1CH2 | CLK Rise Time | - | 10 | - | 10 | ns | From 1.0V to 3.5V |
| (5) | TCL2CL1 | CLK Fall Time | - | 10 | - | 10 | ns | From 3.5V to 1.0V |
| (6) | TDVCL | Data in Setup Time | 30 | - | 20 | - | ns | |
| (7) | TCLDX1 | Data In Hold Time | 10 | - | 10 | - | ns | |
| (8) | TR1VCL | RDY Setup Time into 82C84 (Notes 13, 14) | 35 | - | 35 | - | ns | |
| (9) | TCLR1X | RDY Hold Time into 82C84 (Notes 13, 14) | 0 | - | 0 | - | ns | |
| (10) | TRYHCH | READY Setup Time into 80C88 | 118 | - | 68 | - | ns | |
| (11) | TCHRYX | READY Hold Time into 80C88 | 30 | - | 20 | - | ns | |
| (12) | TRYLCL | READY Inactive to CLK (Note 15) | -8 | - | -8 | - | ns | |
| (13) | TINVCH | Setup Time for Recognition (INTR, NMI, $\overline{\text{TEST}}$) (Note 14) | 30 | - | 15 | - | ns | |
| (14) | TGVCH | $\overline{\text{RQ}}$/$\overline{\text{GT}}$ Setup Time | 30 | - | 15 | - | ns | |
| (15) | TCHGX | $\overline{\text{RQ}}$ Hold Time into 80C88 (Note 16) | 40 | TCHCL+ 10 | 30 | TCHCL+ 10 | ns | |
| (16) | TILIH | Input Rise Time (Except CLK) | - | 15 | - | 15 | ns | From 0.8V to 2.0V |
| (17) | TIHIL | Input Fall Time (Except CLK) | - | 15 | - | 15 | ns | From 2.0V to 0.8V |
| **TIMING RESPONSES** | | | | | | | | |
| (18) | TCLML | Command Active Delay (Note 13) | 5 | 35 | 5 | 35 | ns | CL = 100pF for all 80C88 outputs in addition to internal loads. |
| (19) | TCLMH | Command Inactive (Note 13) | 5 | 35 | 5 | 35 | ns | |
| (20) | TRYHSH | READY Active to Status Passive (Notes 15, 17) | - | 110 | - | 65 | ns | |
| (21) | TCHSV | Status Active Delay | 10 | 110 | 10 | 60 | ns | |
| (22) | TCLSH | Status Inactive Delay (Note 17) | 10 | 130 | 10 | 70 | ns | |
| (23) | TCLAV | Address Valid Delay | 10 | 110 | 10 | 60 | ns | |
| (24) | TCLAX | Address Hold Time | 10 | - | 10 | - | ns | |
| (25) | TCLAZ | Address Float Delay | TCLAX | 80 | TCLAX | 50 | ns | |
| (26) | TCHSZ | Status Float Delay | - | 80 | - | 50 | ns | |
| (27) | TSVLH | Status Valid to ALE High (Note 13) | - | 20 | - | 20 | ns | |
| (28) | TSVMCH | Status Valid to MCE High (Note 13) | - | 30 | - | 30 | ns | |
| (29) | TCLLH | CLK Low to ALE Valid (Note 13) | - | 20 | - | 20 | ns | |
| (30) | TCLMCH | CLK Low to MCE High (Note 13) | - | 25 | - | 25 | ns | |
| (31) | TCHLL | ALE Inactive Delay (Note 13) | 4 | 18 | 4 | 18 | ns | |

## AC Electrical Specifications

$V_{CC}$ = 5.0V ±10%; $T_A$ = $0^oC$ to $+70^oC$ (C80C88, C80C88-2)
$V_{CC}$ = 5.0V ±10%; $T_A$ = $-40^oC$ to $+85^oC$ (I80C88, I80C88-2)
$V_{CC}$ = 5.0V ±10%; $T_A$ = $-55^oC$ to $+125^oC$ (M80C88)

**MAX MODE SYSTEM (USING 82C88 BUS CONTROLLER)**

| SYMBOL | | PARAMETER | 80C88 | | 80C88-2 | | UNITS | TEST CONDITIONS |
|---|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | | |
| (32) | TCLMCL | MCE Inactive Delay (Note 13) | - | 15 | - | 15 | ns | CL = 100pF for all 80C88 outputs in addition to internal loads. |
| (33) | TCLDV | Data Valid Delay | 10 | 110 | 10 | 60 | ns | |
| (34) | TCLDX2 | Data Hold Time | 10 | - | 10 | - | ns | |
| (35) | TCVNV | Control Active Delay (Note 13) | 5 | 45 | 5 | 45 | ns | |
| (36) | TCVNX | Control Inactive Delay (Note 13) | 10 | 45 | 10 | 45 | ns | |
| (37) | TAZRL | Address Float to Read Active | 0 | - | 0 | - | ns | |
| (38) | TCLRL | $\overline{\text{RD}}$ Active Delay | 10 | 165 | 10 | 100 | ns | |
| (39) | TCLRH | $\overline{\text{RD}}$ Inactive Delay | 10 | 150 | 10 | 80 | ns | |
| (40) | TRHAV | $\overline{\text{RD}}$ Inactive to Next Address Active | TCLCL -45 | - | TCLCL -40 | - | ns | |
| (41) | TCHDTL | Direction Control Active Delay (Note 13) | - | 50 | - | 50 | ns | |
| (42) | TCHDTH | Direction Control Inactive Delay (Note 1) | - | 30 | - | 30 | ns | |
| (43) | TCLGL | $\overline{\text{GT}}$ Active Delay | 0 | 85 | 0 | 50 | ns | |
| (44) | TCLGH | $\overline{\text{GT}}$ Inactive Delay | 0 | 85 | 0 | 50 | ns | |
| (45) | TRLRH | $\overline{\text{RD}}$ Width | 2TCLCL -75 | - | 2TCLCL -50 | - | ns | |
| (46) | TOLOH | Output Rise Time | - | 15 | - | 15 | ns | From 0.8V to 2.0V |
| (47) | TOHOL | Output Fall Time | - | 15 | - | 15 | ns | From 2.0V to 0.8V |

NOTES:

13. Signal at 82C84A or 82C88 shown for reference only.
14. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
15. Applies only to T2 state (8ns into T3).
16. The 80C88 actively pulls the $\overline{\text{RQ}}/\overline{\text{GT}}$ pin to a logic one on the following clock low time.
17. Status lines return to their inactive (logic one) state after CLK goes low and READY goes high.

## Waveforms

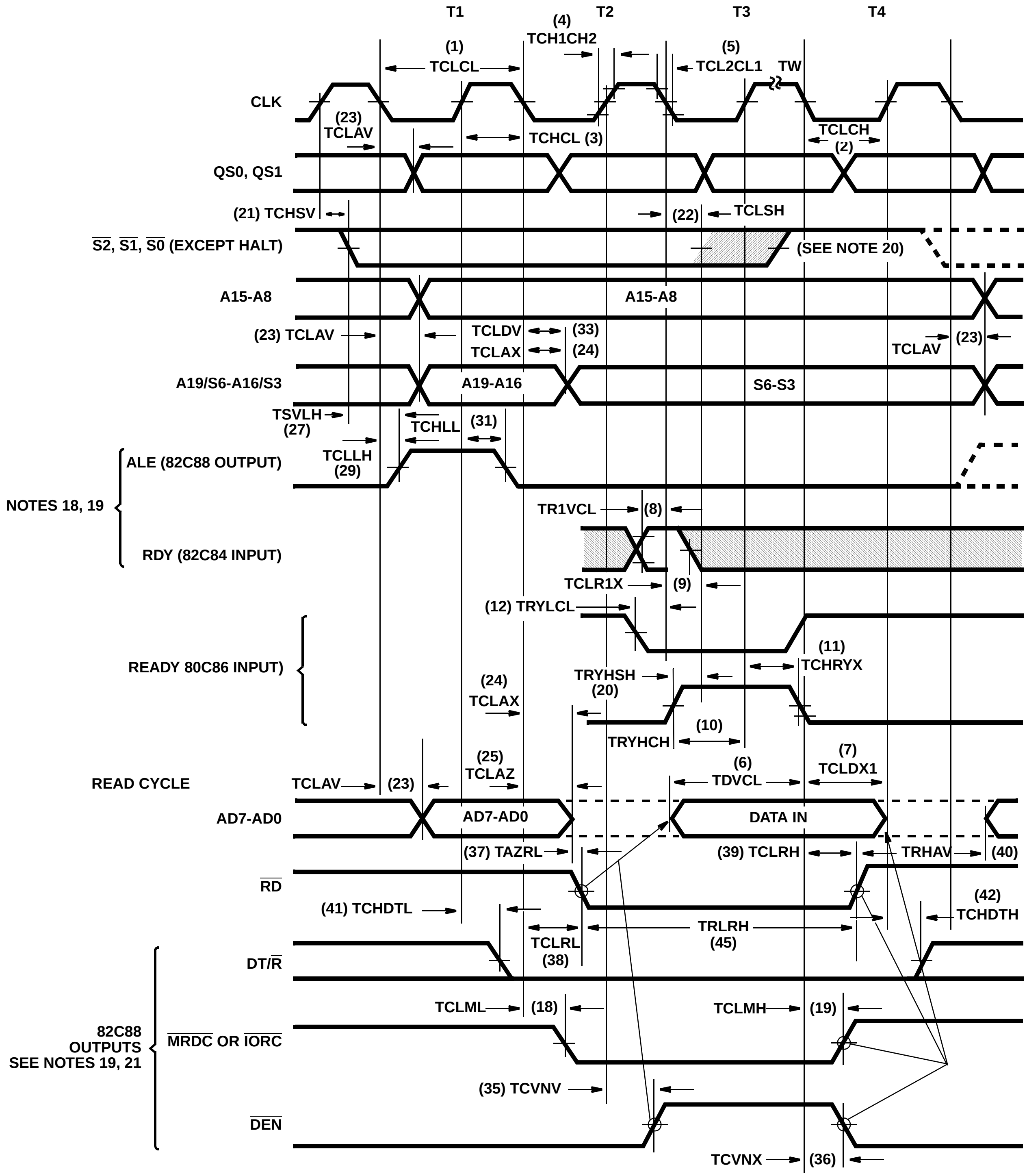


FIGURE 24. BUS TIMING - MAXIMUM MODE (USING 82C88)

NOTES:

18. RDY is sampled near the end of T2, T3, TW to determine if TW machine states are to be inserted.
19. Signals at 82C84A or 82C88 are shown for reference only.
20. Status inactive in state just prior to T4.
21. The issuance of the 82C88 command and control signals (MRDC, MWTC, AMWC, IORC, IOWC, AIOWC, INTA, and DEN) lags the active high 82C88 CEN.

## *Waveforms* (Continued)

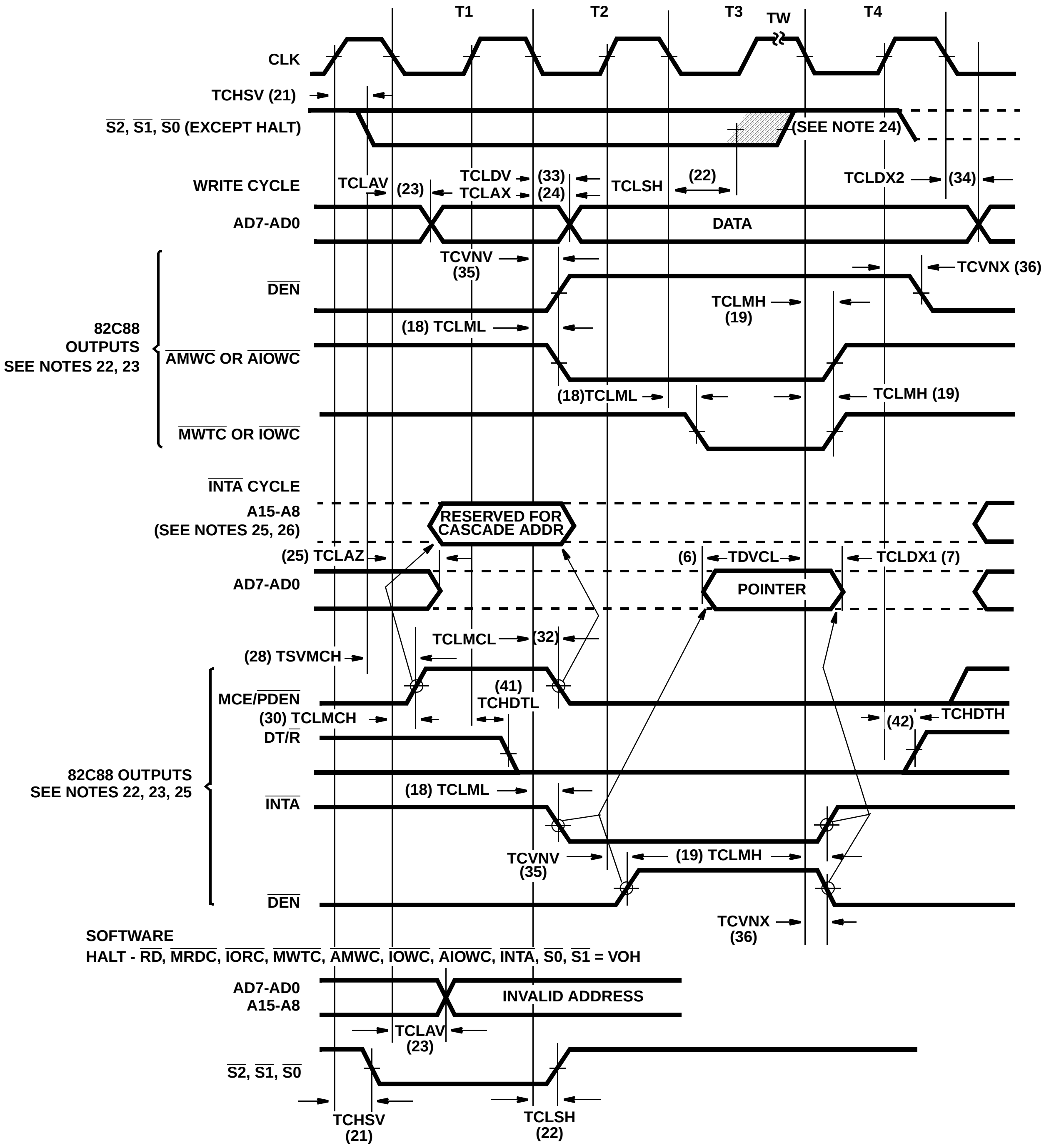


FIGURE 25. BUS TIMING - MAXIMUM MODE SYSTEM (USING 82C88) (Continued)

NOTES:

22. Signals at 82C84A or 82C86 are shown for reference only.
23. The issuance of the 82C88 command and control signals (MRDC, MWTC, AMWC, IORC, IOWC, AIOWC, INTA and DEN) lags the active high 82C88 CEN.
24. Status inactive in state just prior to T4.
25. Cascade address is valid between first and second INTA cycles.
26. Two INTA cycles run back-to-back. The 80C88 local ADDR/DATA bus is floating during both INTA cycles. Control for pointer address is shown for second INTA cycle.

## Waveforms (Continued)

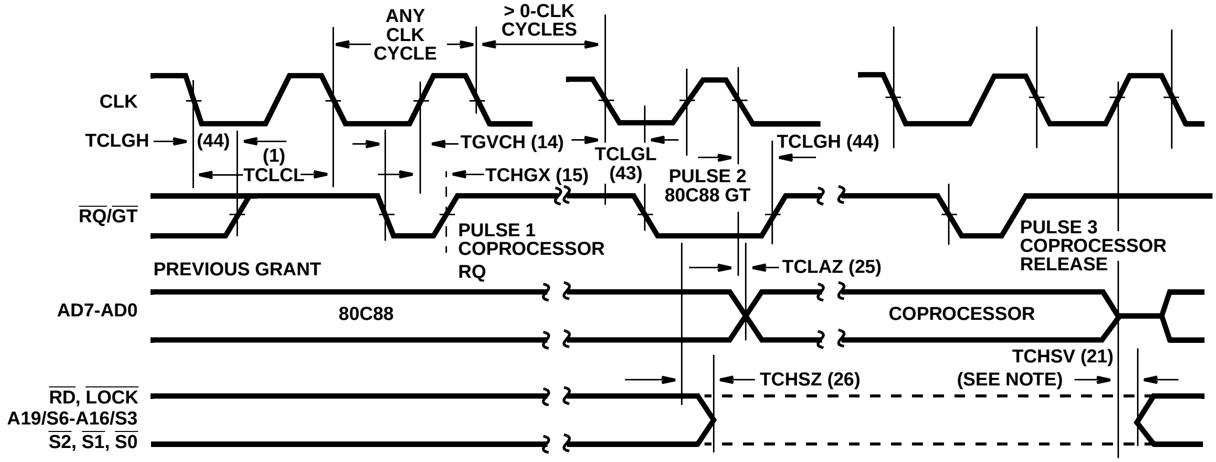


FIGURE 26. REQUEST/GRANT SEQUENCE TIMING (MAXIMUM MODE ONLY)

NOTE: The coprocessor may not drive the busses outside the region shown without risking contention.

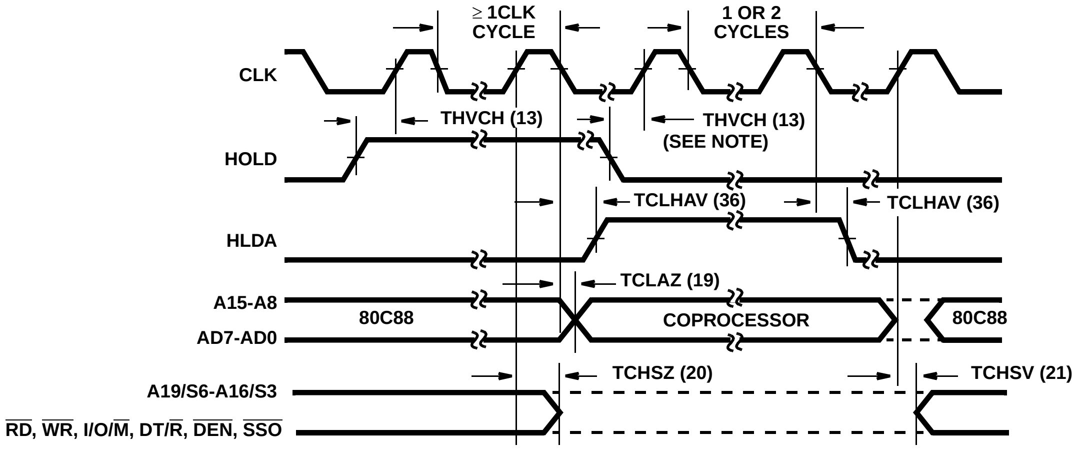


FIGURE 27. HOLD/HOLD ACKNOWLEDGE TIMING (MINIMUM MODE ONLY)

NOTE: Setup requirements for asynchronous signals only to guarantee recognition at next CLK.

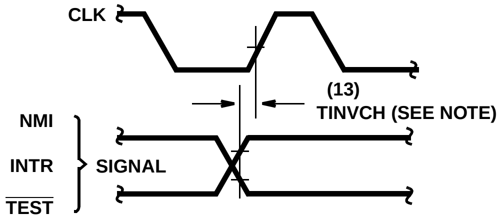


FIGURE 28. ASYNCHRONOUS SIGNAL RECOGNITION

NOTE: Setup requirements for asynchronous signals only to guarantee recognition at next CLK.

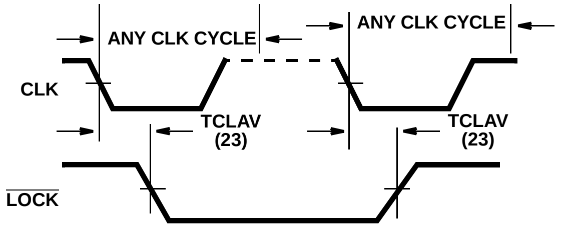


FIGURE 29. BUS LOCK SIGNAL TIMING (MAXIMUM MODE ONLY)

## Waveforms (Continued)

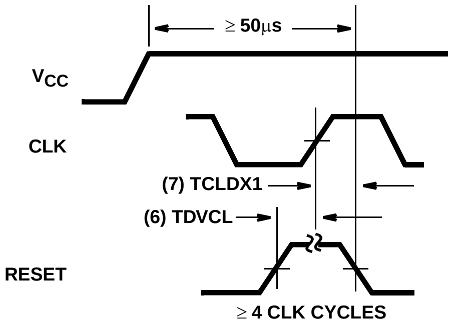
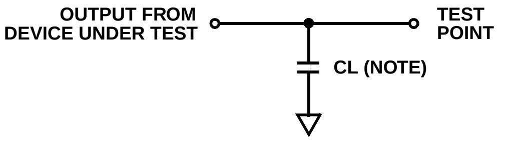


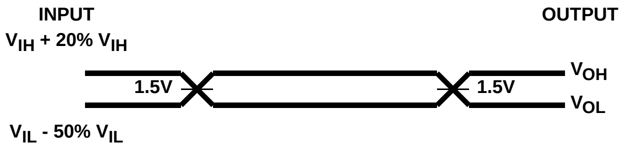
FIGURE 30. RESET TIMING

## AC Test Circuit



NOTE: Includes stay and jig capacitance.

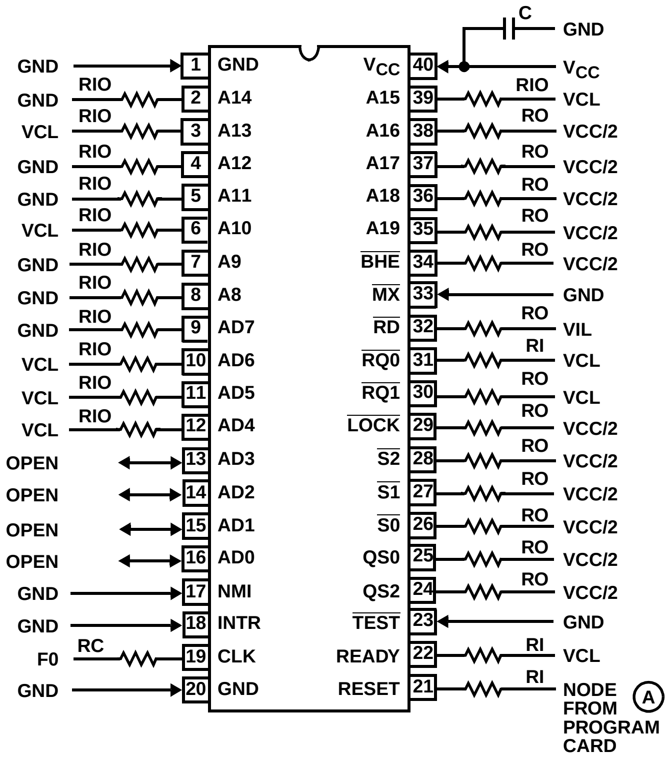
## AC Testing Input, Output Waveform



AC Testing: All input signals (other than CLK) must switch between $V_{ILMAX}$ -50% $V_{IL}$ and $V_{IHMIN}$ +20% $V_{IH}$. CLK must switch between 0.4V and $V_{CC}$ -0.4V. Input rise and fall times are driven at 1ns/V.

## Burn-In Circuits

MD80C88 (CERDIP)



## Burn-In Circuits (Continued)

NOTES:

1. $V_{CC}$ = 5.5V ±0.5V, GND = 0V.
2. Input voltage limits (except clock):
   $V_{IL}$ (Maximum) = 0.4V
   $V_{IH}$ (Minimum) = 2.6V, $V_{IH}$ (Clock) = $V_{CC}$ - 0.4V) minimum.
3. VCC/2 is external supply set to 2.7V ±10%.
4. $V_{CL}$ is generated on program card ($V_{CC}$ - 0.65V).
5. Pins 13 - 16 input sequenced instructions from internal hold devices, (DIP Only).
6. F0 = 100kHz ±10%.
7. Node Ⓐ = a 40µs pulse every 2.56ms.

COMPONENTS:

1. RI = 10kΩ ±5%, 1/4W
2. RO = 1.2kΩ ±5%, 1/4W
3. RIO = 2.7kΩ ±5%, 1/4W
4. RC = 1kΩ ±5%, 1/4W
5. C = 0.01µF (Minimum)

## Die Characteristics

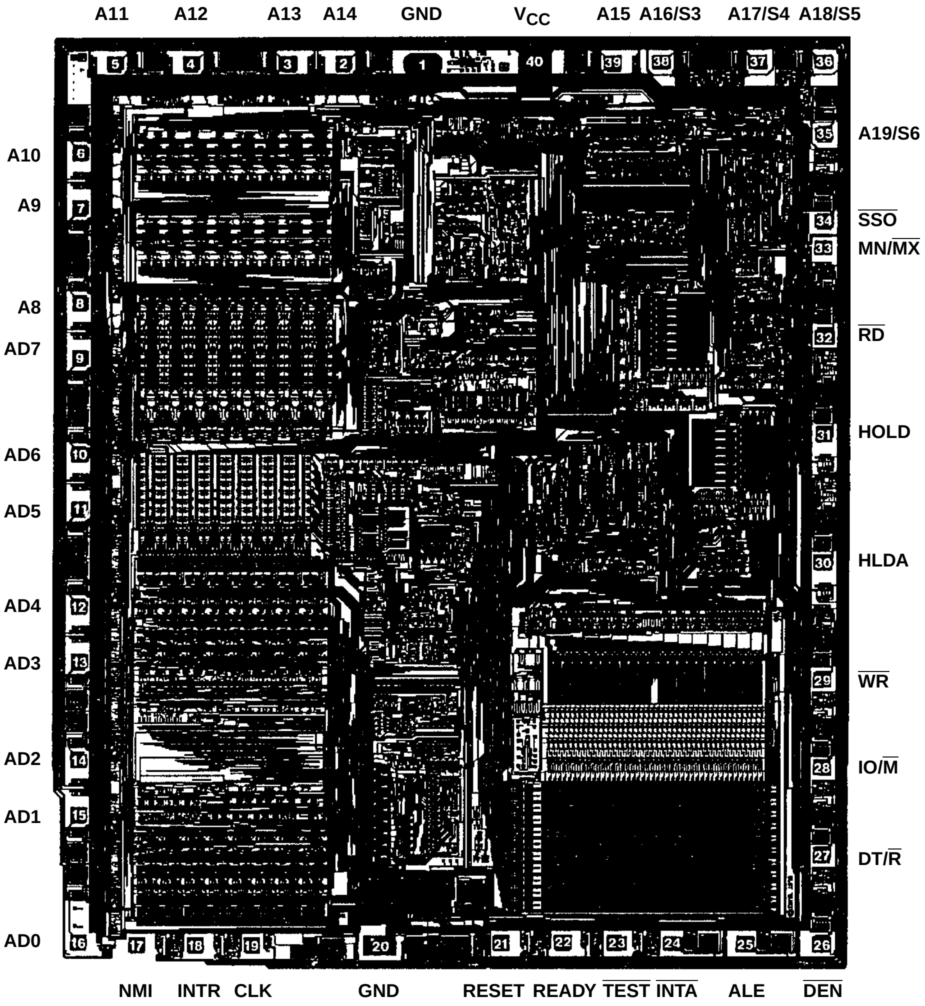
**METALLIZATION:**
Type: Silicon - Aluminum
Thickness: 11kÅ ±2kÅ

**GLASSIVATION:**
Type: $SiO_2$
Thickness: 8kÅ ±1kÅ

**WORST CASE CURRENT DENSITY:**
$1.5 \times 10^5$ A/$cm^2$

## Metallization Mask Layout

**80C88**



## Instruction Set Summary

| MNEMONIC AND DESCRIPTION | INSTRUCTION CODE | | | |
|---|---|---|---|---|
| | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 |
| **DATA TRANSFER**<br>**MOV = MOVE:** | | | | |
| Register/Memory to/from Register | 1 0 0 0 1 0 d w | mod reg r/m | | |
| Immediate to Register/Memory | 1 1 0 0 0 1 1 w | mod 0 0 0 r/m | data | data if w 1 |
| Immediate to Register | 1 0 1 1 w reg | data | data if w 1 | |
| Memory to Accumulator | 1 0 1 0 0 0 0 w | addr-low | addr-high | |
| Accumulator to Memory | 1 0 1 0 0 0 1 w | addr-low | addr-high | |
| Register/Memory to Segment Register †† | 1 0 0 0 1 1 1 0 | mod 0 reg r/m | | |
| Segment Register to Register/Memory | 1 0 0 0 1 1 0 0 | mod 0 reg r/m | | |
| **PUSH = Push:** | | | | |
| Register/Memory | 1 1 1 1 1 1 1 1 | mod 1 1 0 r/m | | |
| Register | 0 1 0 1 0 reg | | | |
| Segment Register | 0 0 0 reg 1 1 0 | | | |
| **POP = Pop:** | | | | |
| Register/Memory | 1 0 0 0 1 1 1 1 | mod 0 0 0 r/m | | |
| Register | 0 1 0 1 1 reg | | | |
| Segment Register | 0 0 0 reg 1 1 1 | | | |
| **XCHG = Exchange:** | | | | |
| Register/Memory with Register | 1 0 0 0 0 1 1 w | mod reg r/m | | |
| Register with Accumulator | 1 0 0 1 0 reg | | | |
| **IN = Input from:** | | | | |
| Fixed Port | 1 1 1 0 0 1 0 w | port | | |
| Variable Port | 1 1 1 0 1 1 0 w | | | |
| **OUT = Output to:** | | | | |
| Fixed Port | 1 1 1 0 0 1 1 w | port | | |
| Variable Port | 1 1 1 0 1 1 1 w | | | |
| **XLAT =** Translate Byte to AL | 1 1 0 1 0 1 1 1 | | | |
| **LEA =** Load EA to Register2 | 1 0 0 0 1 1 0 1 | mod reg r/m | | |
| **LDS =** Load Pointer to DS | 1 1 0 0 0 1 0 1 | mod reg r/m | | |
| **LES =** Load Pointer to ES | 1 1 0 0 0 1 0 0 | mod reg r/m | | |
| **LAHF =** Load AH with Flags | 1 0 0 1 1 1 1 1 | | | |
| **SAHF =** Store AH into Flags | 1 0 0 1 1 1 1 0 | | | |
| **PUSHF =** Push Flags | 1 0 0 1 1 1 0 0 | | | |
| **POPF =** Pop Flags | 1 0 0 1 1 1 0 1 | | | |
| **ARITHMETIC**<br>**ADD = Add:** | | | | |
| Register/Memory with Register to Either | 0 0 0 0 0 0 d w | mod reg r/m | | |
| Immediate to Register/Memory | 1 0 0 0 0 0 s w | mod 0 0 0 r/m | data | data if s:w = 01 |
| Immediate to Accumulator | 0 0 0 0 0 1 0 w | data | data if w = 1 | |
| **ADC = Add with Carry:** | | | | |
| Register/Memory with Register to Either | 0 0 0 1 0 0 d w | mod reg r/m | | |

## *Instruction Set Summary* (Continued)

| MNEMONIC AND DESCRIPTION | INSTRUCTION CODE | | | |
|---|---|---|---|---|
| | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 |
| Immediate to Register/Memory | 1 0 0 0 0 0 s w | mod 0 1 0 r/m | data | data if s:w = 01 |
| Immediate to Accumulator | 0 0 0 1 0 1 0 w | data | data if w = 1 | |
| **INC = Increment:** | | | | |
| Register/Memory | 1 1 1 1 1 1 1 w | mod 0 0 0 r/m | | |
| Register | 0 1 0 0 0 reg | | | |
| **AAA =** ASCII Adjust for Add | 0 0 1 1 0 1 1 1 | | | |
| **DAA =** Decimal Adjust for Add | 0 0 1 0 0 1 1 1 | | | |
| **SUB =** Subtract: | | | | |
| Register/Memory and Register to Either | 0 0 1 0 1 0 d w | mod reg r/m | | |
| Immediate from Register/Memory | 1 0 0 0 0 0 s w | mod 1 0 1 r/m | data | data if s:w = 01 |
| Immediate from Accumulator | 0 0 1 0 1 1 0 w | data | data if w = 1 | |
| **SBB = Subtract with Borrow** | | | | |
| Register/Memory and Register to Either | 0 0 0 1 1 0 d w | mod reg r/m | | |
| Immediate from Register/Memory | 1 0 0 0 0 0 s w | mod 0 1 1 r/m | data | data if s:w = 01 |
| Immediate from Accumulator | 0 0 0 1 1 1 0 w | data | data if w = 1 | |
| **DEC = Decrement:** | | | | |
| Register/Memory | 1 1 1 1 1 1 1 w | mod 0 0 1 r/m | | |
| Register | 0 1 0 0 1 reg | | | |
| **NEG = Change Sign** | 1 1 1 1 0 1 1 w | mod 0 1 1 r/m | | |
| **CMP = Compare:** | | | | |
| Register/Memory and Register | 0 0 1 1 1 0 d w | mod reg r/m | | |
| Immediate with Register/Memory | 1 0 0 0 0 0 s w | mod 1 1 1 r/m | data | data if s:w = 01 |
| Immediate with Accumulator | 0 0 1 1 1 1 0 w | data | data if w = 1 | |
| **AAS =** ASCII Adjust for Subtract | 0 0 1 1 1 1 1 1 | | | |
| **DAS =** Decimal Adjust for Subtract | 0 0 1 0 1 1 1 1 | | | |
| **MUL =** Multiply (Unsigned) | 1 1 1 1 0 1 1 w | mod 1 0 0 r/m | | |
| **IMUL =** Integer Multiply (Signed) | 1 1 1 1 0 1 1 w | mod 1 0 1 r/m | | |
| **AAM =** ASCII Adjust for Multiply | 1 1 0 1 0 1 0 0 | 0 0 0 0 1 0 1 0 | | |
| **DIV =** Divide (Unsigned) | 1 1 1 1 0 1 1 w | mod 1 1 0 r/m | | |
| **IDIV =** Integer Divide (Signed) | 1 1 1 1 0 1 1 w | mod 1 1 1 r/m | | |
| **AAD =** ASCII Adjust for Divide | 1 1 0 1 0 1 0 1 | 0 0 0 0 1 0 1 0 | | |
| **CBW =** Convert Byte to Word | 1 0 0 1 1 0 0 0 | | | |
| **CWD =** Convert Word to Double Word | 1 0 0 1 1 0 0 1 | | | |
| **LOGIC** | | | | |
| **NOT = Invert** | 1 1 1 1 0 1 1 w | mod 0 1 0 r/m | | |
| **SHL/SAL =** Shift Logical/Arithmetic Left | 1 1 0 1 0 0 v w | mod 1 0 0 r/m | | |
| **SHR =** Shift Logical Right | 1 1 0 1 0 0 v w | mod 1 0 1 r/m | | |
| **SAR =** Shift Arithmetic Right | 1 1 0 1 0 0 v w | mod 1 1 1 r/m | | |
| **ROL =** Rotate Left | 1 1 0 1 0 0 v w | mod 0 0 0 r/m | | |
| **ROR =** Rotate Right | 1 1 0 1 0 0 v w | mod 0 0 1 r/m | | |
| **RCL =** Rotate Through Carry Flag Left | 1 1 0 1 0 0 v w | mod 0 1 0 r/m | | |

## *Instruction Set Summary* (Continued)

| MNEMONIC AND DESCRIPTION | INSTRUCTION CODE | | | |
|---|---|---|---|---|
| | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 |
| **RCR =** Rotate Through Carry Right | 1 1 0 1 0 0 v w | mod 0 1 1 r/m | | |
| **AND = And:** | | | | |
| Reg./Memory and Register to Either | 0 0 1 0 0 0 0 d w | mod reg r/m | | |
| Immediate to Register/Memory | 1 0 0 0 0 0 0 w | mod 1 0 0 r/m | data | data if w = 1 |
| Immediate to Accumulator | 0 0 1 0 0 1 0 w | data | data if w = 1 | |
| **TEST = And Function to Flags, No Result:** | | | | |
| Register/Memory and Register | 1 0 0 0 0 1 0 w | mod reg r/m | | |
| Immediate Data and Register/Memory | 1 1 1 1 0 1 1 w | mod 0 0 0 r/m | data | data if w = 1 |
| Immediate Data and Accumulator | 1 0 1 0 1 0 0 w | data | data if w = 1 | |
| **OR = Or:** | | | | |
| Register/Memory and Register to Either | 0 0 0 0 1 0 d w | mod reg r/m | | |
| Immediate to Register/Memory | 1 0 0 0 0 0 0 w | mod 1 0 1 r/m | data | data if w = 1 |
| Immediate to Accumulator | 0 0 0 0 1 1 0 w | data | data if w = 1 | |
| **XOR = Exclusive or:** | | | | |
| Register/Memory and Register to Either | 0 0 1 1 0 0 d w | mod reg r/m | | |
| Immediate to Register/Memory | 1 0 0 0 0 0 0 w | mod 1 1 0 r/m | data | data if w = 1 |
| Immediate to Accumulator | 0 0 1 1 0 1 0 w | data | data if w = 1 | |
| **STRING MANIPULATION** | | | | |
| **REP =** Repeat | 1 1 1 1 0 0 1 z | | | |
| **MOVS =** Move Byte/Word | 1 0 1 0 0 1 0 w | | | |
| **CMPS =** Compare Byte/Word | 1 0 1 0 0 1 1 w | | | |
| **SCAS =** Scan Byte/Word | 1 0 1 0 1 1 1 w | | | |
| **LODS =** Load Byte/Word to AL/AX | 1 0 1 0 1 1 0 w | | | |
| **STOS =** Stor Byte/Word from AL/A | 1 0 1 0 1 0 1 w | | | |
| **CONTROL TRANSFER** | | | | |
| **CALL = Call:** | | | | |
| Direct Within Segment | 1 1 1 0 1 0 0 0 | disp-low | disp-high | |
| Indirect Within Segment | 1 1 1 1 1 1 1 1 | mod 0 1 0 r/m | | |
| Direct Intersegment | 1 0 0 1 1 0 1 0 | offset-low | offset-high | |
| | | seg-low | seg-high | |
| Indirect Intersegment | 1 1 1 1 1 1 1 1 | mod 0 1 1 r/m | | |
| **JMP = Unconditional Jump:** | | | | |
| Direct Within Segment | 1 1 1 0 1 0 0 1 | disp-low | disp-high | |
| Direct Within Segment-Short | 1 1 1 0 1 0 1 1 | disp | | |
| Indirect Within Segment | 1 1 1 1 1 1 1 1 | mod 1 0 0 r/m | | |
| Direct Intersegment | 1 1 1 0 1 0 1 0 | offset-low | offset-high | |
| | | seg-low | seg-high | |
| Indirect Intersegment | 1 1 1 1 1 1 1 1 | mod 1 0 1 r/m | | |
| **RET = Return from CALL:** | | | | |
| Within Segment | 1 1 0 0 0 0 1 1 | | | |
| Within Seg Adding Immed to SP | 1 1 0 0 0 0 1 0 | data-low | data-high | |

## *Instruction Set Summary* (Continued)

| MNEMONIC AND DESCRIPTION | INSTRUCTION CODE | | | |
|---|---|---|---|---|
| | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 |
| Intersegment | 1 1 0 0 1 0 1 1 | | | |
| Intersegment Adding Immediate to SP | 1 1 0 0 1 0 1 0 | data-low | data-high | |
| **JE/JZ =** Jump on Equal/Zero | 0 1 1 1 0 1 0 0 | disp | | |
| **JL/JNGE =** Jump on Less/Not Greater or Equal | 0 1 1 1 1 1 0 0 | disp | | |
| **JLE/JNG =** Jump on Less or Equal/ Not Greater | 0 1 1 1 1 1 1 0 | disp | | |
| **JB/JNAE =** Jump on Below/Not Above or Equal | 0 1 1 1 0 0 1 0 | disp | | |
| **JBE/JNA =** Jump on Below or Equal/Not Above | 0 1 1 1 0 1 1 0 | disp | | |
| **JP/JPE =** Jump on Parity/Parity Even | 0 1 1 1 1 0 1 0 | disp | | |
| **JO =** Jump on Overflow | 0 1 1 1 0 0 0 0 | disp | | |
| **JS =** Jump on Sign | 0 1 1 1 1 0 0 0 | disp | | |
| **JNE/JNZ =** Jump on Not Equal/Not Zero | 0 1 1 1 0 1 0 1 | disp | | |
| **JNL/JGE =** Jump on Not Less/Greater or Equal | 0 1 1 1 1 1 0 1 | disp | | |
| **JNLE/JG =** Jump on Not Less or Equal/Greater | 0 1 1 1 1 1 1 1 | disp | | |
| **JNB/JAE =** Jump on Not Below/Above or Equal | 0 1 1 1 0 0 1 1 | disp | | |
| **JNBE/JA =** Jump on Not Below or Equal/Above | 0 1 1 1 0 1 1 1 | disp | | |
| **JNP/JPO =** Jump on Not Par/Par Odd | 0 1 1 1 1 0 1 1 | disp | | |
| **JNO =** Jump on Not Overflow | 0 1 1 1 0 0 0 1 | disp | | |
| **JNS =** Jump on Not Sign | 0 1 1 1 1 0 0 1 | disp | | |
| **LOOP =** Loop CX Times | 1 1 1 0 0 0 1 0 | disp | | |
| **LOOPZ/LOOPE =** Loop While Zero/Equal | 1 1 1 0 0 0 0 1 | disp | | |
| **LOOPNZ/LOOPNE =** Loop While Not Zero/Equal | 1 1 1 0 0 0 0 0 | disp | | |
| **JCXZ =** Jump on CX Zero | 1 1 1 0 0 0 1 1 | disp | | |
| **INT =** Interrupt | | | | |
| Type Specified | 1 1 0 0 1 1 0 1 | type | | |
| Type 3 | 1 1 0 0 1 1 0 0 | | | |
| **INTO =** Interrupt on Overflow | 1 1 0 0 1 1 1 0 | | | |
| **IRET =** Interrupt Return | 1 1 0 0 1 1 1 1 | | | |
| **PROCESSOR CONTROL** | | | | |
| **CLC =** Clear Carry | 1 1 1 1 1 0 0 0 | | | |
| **CMC =** Complement Carry | 1 1 1 1 0 1 0 1 | | | |
| **STC =** Set Carry | 1 1 1 1 1 0 0 1 | | | |
| **CLD =** Clear Direction | 1 1 1 1 1 1 0 0 | | | |
| **STD =** Set Direction | 1 1 1 1 1 1 0 1 | | | |
| **CLI =** Clear Interrupt | 1 1 1 1 1 0 1 0 | | | |
| **ST =** Set Interrupt | 1 1 1 1 1 0 1 1 | | | |
| **HLT =** Halt | 1 1 1 1 0 1 0 0 | | | |
| **WAIT =** Wait | 1 0 0 1 1 0 1 1 | | | |
| **ESC =** Escape (to External Device) | 1 1 0 1 1 x x x | mod x x x r/m | | |
| **LOCK =** Bus Lock Prefix | 1 1 1 1 0 0 0 0 | | | |

## Instruction Set Summary (Continued)

| MNEMONIC AND DESCRIPTION | INSTRUCTION CODE | | | |
|---|---|---|---|---|
| | 76543210 | 76543210 | 76543210 | 76543210 |

NOTES:
AL = 8-bit accumulator
AX = 16-bit accumulator
CX = Count register
DS= Data segment
ES = Extra segment
Above/below refers to unsigned value.
Greater = more positive;
Less = less positive (more negative) signed values
if d = 1 then “to” reg; if d = 0 then “from” reg
if w = 1 then word instruction; if w = 0 then byte instruction
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 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.
†† MOV CS, REG/MEMORY not allowed.
if s:w = 01 then 16-bits of immediate data form the operand.
if s:w = 11 then an immediate data byte is sign extended to form the 16-bit operand.
if v = 0 then “count” = 1; if v = 1 then “count” in ($C_L$)
x = don't care
z is used for string primitives for comparison with ZF FLAG.

### SEGMENT OVERRIDE PREFIX

| 001 reg 11 0 |
|---|

REG is assigned according to the following table:

| 16-BIT (w = 1) | 8-BIT (w = 0) | SEGMENT |
|---|---|---|
| 000 AX | 000 AL | 00 ES |
| 001 CX | 001 CL | 01 CS |
| 010 DX | 010 DL | 10 SS |
| 011 BX | 011 BL | 11 DS |
| 100 SP | 100 AH | |
| 101 BP | 101 CH | |
| 110 SI | 110 DH | |
| 111 DI | 111 BH | |

Instructions which reference the flag register file as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS =

X:X:X:X:(OF):(DF):(IF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)

All Intersil U.S. products are manufactured, assembled and tested utilizing ISO9000 quality systems.
Intersil Corporation’s quality certifications can be viewed at www.intersil.com/design/quality

*Intersil products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design, software and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.*

For information regarding Intersil Corporation and its products, see www.intersil.com