

---

**PART NUMBER****VQ80960MC-25-ROC**

---

**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-38535
  - 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.*

## **MG80960MC-25, MG80960MC-25/B**

### *Embedded 32-Bit Microprocessor with Integrated Floating-Point Unit and Memory Management Unit*

The 80960MC is the enhanced military member of Intel's new 32-bit microprocessor family, the 960 series, which is designed especially for embedded applications. It is based on the family's high performance, common core architecture, and includes a 512-byte instruction cache, a built-in interrupt controller, an integrated floating-point unit and a memory management unit. The 80960MC has a large register set, multiple parallel execution units, and a high-bandwidth, burst bus.

#### **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**

# 80960MC EMBEDDED 32-BIT MICROPROCESSOR WITH INTEGRATED FLOATING-POINT UNIT AND MEMORY MANAGEMENT UNIT

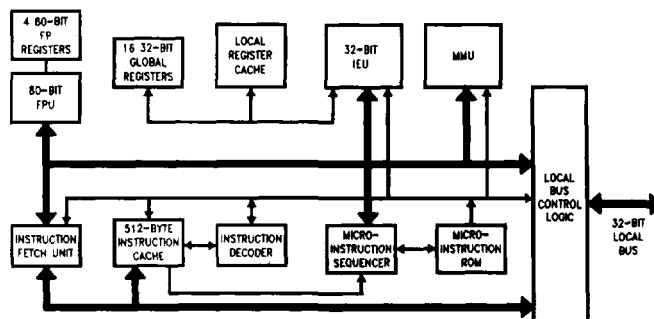
*Military*

- **High-Performance Embedded Architecture**
  - 25 MIPS Burst Execution at 20 MHz
  - 9.4 MIPS\* Sustained Execution at 25 MHz
- **On-Chip Floating-Point Unit**
  - Supports IEEE 754 Floating-Point Standard
  - Full Transcendental Support
  - Four 80-Bit Registers
  - 5.2 Million Whetstones/Second at 25 MHz
- **512-Byte On-Chip Instruction Cache**
  - Direct Mapped
  - Parallel Load/Decode for Uncached Instructions
- **Multiple Register Sets**
  - Sixteen Global 32-Bit Registers
  - Sixteen Local 32-Bit Registers
  - Four Local Register Sets Stored On-Chip (Sixteen 32-Bit Registers per Set)
  - Register Scoreboarding
- **On-Chip Memory Management Unit**
  - 4 Gigabyte Virtual Address Space per Task
  - 4 Kbyte Pages with Supervisor/User Protection
- **Built-In Interrupt Controller**
  - 32 Priority Levels
  - 248 Vectors
  - Supports M8259A
  - 3.4  $\mu$ s Latency
- **Easy to Use, High Bandwidth 32-Bit Bus**
  - 66.7 MBytes/s Burst
  - Up to 16-Bytes Transferred per Burst
- **Multitasking and Multiprocessor Support**
  - Automatic Task Dispatching
  - Prioritized Task Queues
- **Advanced Package Technology**
  - 132 Lead Ceramic Pin Grid Array
  - 164 Lead Ceramic Quad Flatpack
- **Military Temperature Range**
  - -55°C to +125°C (T<sub>C</sub>)

5

The 80960MC is the enhanced military member of Intel's new 32-bit microprocessor family, the 960 series, which is designed especially for embedded applications. It is based on the family's high performance, common core architecture, and includes a 512-byte instruction cache, a built-in interrupt controller, an integrated floating-point unit and a memory management unit. The 80960MC has a large register set, multiple parallel execution units, and a high-bandwidth, burst bus. Using advanced RISC technology, this high performance processor can respond to interrupts in under 3.4  $\mu$ s and is capable of execution rates in excess of 9.4 million instructions per second.\* The 80960MC is well-suited for a wide range of military and other high reliability applications, including avionics, airborne radar, navigation, and instrumentation.

\*Relative to Digital Equipment Corporation's VAX-11/780\*\* at 1 MIPS



**Figure 1. The 80960MC's Highly Parallel Microarchitecture**

271080-1

\*\*VAX-11™ is a trademark of Digital Equipment Corporation.

**THE 960 SERIES**

The 80960MC is the enhanced military member of a new family of 32-bit microprocessors from Intel known as the 960 Series. This series was especially designed to serve the needs of embedded applications. The embedded market includes applications as diverse as industrial automation, avionics, image processing, graphics, robotics, telecommunications, and automobiles. These types of applications require high integration, low power consumption, quick interrupt response times, and high performance. Since time to market is critical, embedded microprocessors need to be easy to use in both hardware and software designs.

All members of the 80960 series share a common core architecture which utilizes RISC technology so that, except for special functions, the family members are object code compatible. Each new processor in the series will add its own special set of functions to the core to satisfy the needs of a specific application or range of applications in the embedded market. For example, future processors may include a DMA controller, a timer, or an A/D converter.

The 80960MC includes an integrated Floating Point Unit (FPU), a Memory Management Unit (MMU), multitasking support, and multiprocessor support. There are also two commercial members of the family: the 80960KB processor with integrated FPU and the 80960KA without floating-point.

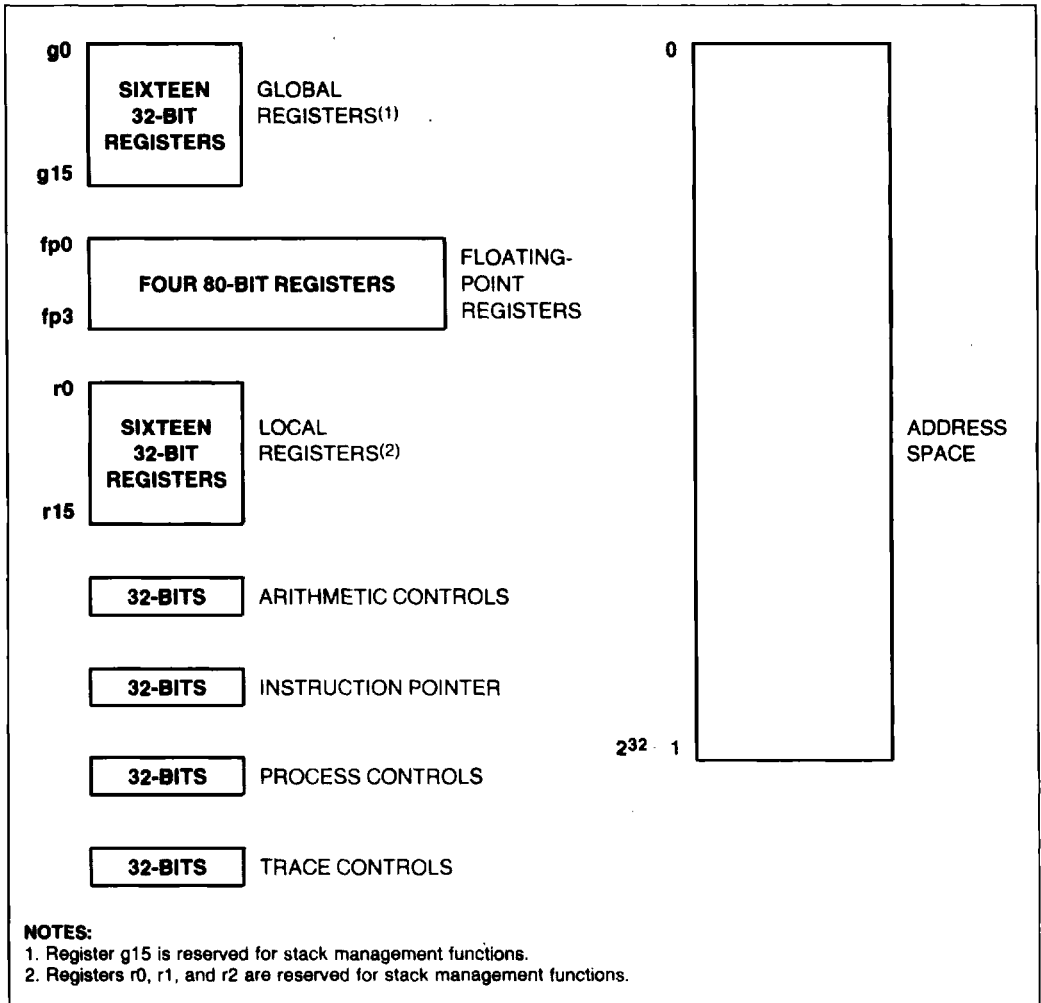


Figure 2. Register Set

**KEY PERFORMANCE FEATURES**

The 80960MC's architecture is based on the most recent advances in RISC technology and is grounded in Intel's long experience in designing embedded controllers. Many features contribute to the 80960MC's exceptional performance:

**1. Large Register Set.** Having a large number of registers reduces the number of times that a processor needs to access memory. Modern compilers can take advantage of this feature to optimize execution speed. For maximum flexibility, the 80960MC provides thirty-two 32-bit registers (sixteen local and sixteen global) and four 80-bit floating-point global registers. (See Figure 2.)

**2. Fast Instruction Execution.** Simple functions make up the bulk of instructions in most programs,

so that execution speed can be greatly improved by ensuring that these core instructions execute in as short a time as possible. The most-frequently executed instructions such as register-register moves, add/subtract, logical operations, and shifts execute in one to two cycles (Table 1 contains a list of instructions.)

**3. Load/Store Architecture.** Like other processors based on RISC technology, the 80960MC has a Load/Store architecture, only the LOAD and STORE instructions reference memory; all other instructions operate on registers. This type of architecture simplifies instruction decoding and is used in combination with other techniques to increase parallelism.

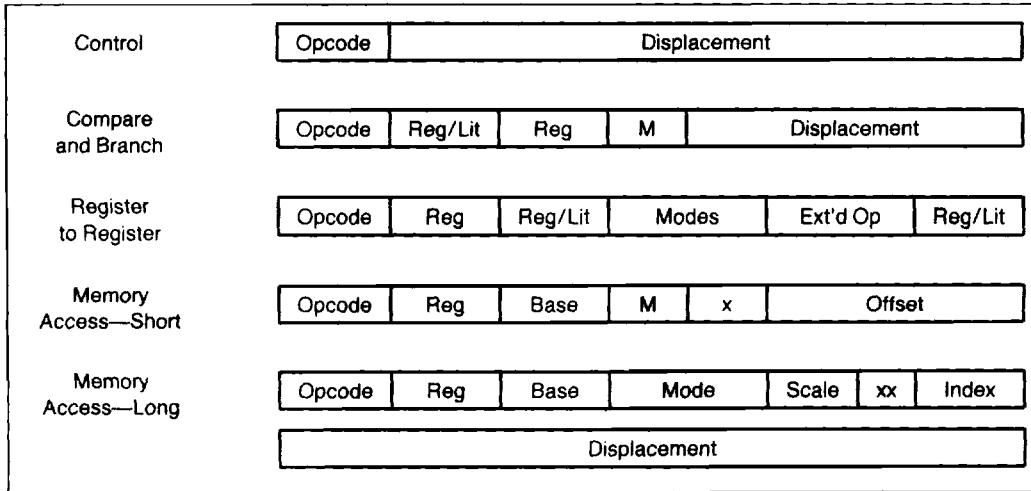


Figure 3. Instruction Formats

**Table 1. 80960MC Instruction Set**

<b>Data Movement</b>	<b>Arithmetic</b>	<b>Floating Point</b>	<b>Logical</b>
Load Store Move Load Address Load Physical Address	Add Subtract Multiply Divide Remainder Modulo Shift	Add Subtract Multiply Divide Remainder Scale Round Square Root Sine Cosine Tangent Arctangent Log Log Binary Log Natural Exponent Classify Copy Real Extended Compare	And Not And And Not Or Exclusive Or Not Or Or Not Nor Exclusive Nor Not Nand Rotate
<b>Comparison</b>	<b>Branch</b>	<b>Bit and Bit Field</b>	<b>String</b>
Compare Conditional Compare Compare and Increment Compare and Decrement	Unconditional Branch Conditional Branch Compare and Branch	Set Bit Clear Bit Not Bit Check Bit Alter Bit Scan for Bit Scan over Bit Extract Modify	Move String Move Quick String Fill String Compare String Scan Byte for Equal
<b>Conversion</b>	<b>Decimal</b>	<b>Call/Return</b>	<b>Process Management</b>
Convert Real to Integer Convert Integer to Real	Move Add with Carry Subtract with Carry	Call Call Extended Call System Return Branch and Link	Schedule Process Saves Process Resume Process Load Process Time Modify Process Controls Wait Conditional Wait Signal Receive Conditional Receive Send Send Service Atomic Add Atomic Modify
<b>Fault</b>	<b>Debug</b>	<b>Miscellaneous</b>	
Conditional Fault Synchronize Faults	Modify Trace Controls Mark Force Mark	Flush Local Registers Inspect Access Modify Arithmetic Controls Test Condition Code	

**4. Simple Instruction Formats.** All instructions in the 80960MC are 32-bits long and must be aligned on word boundaries. This alignment makes it possible to eliminate the instruction-alignment stage in the pipeline. To simplify the instruction decoder further, there are only five instruction formats and each instruction uses only one format. (See Figure 3.)

**5. Overlapped Instruction Execution.** A load operation allows execution of subsequent instructions to continue before the data has been returned from memory, so that these instructions can overlap the load. The 80960MC manages this process transparently to software through the use of a register scoreboard. Conditional instructions also make use of a scoreboard so that subsequent unrelated instructions can be executed while the conditional instruction is pending.

**6. Integer Execution Optimization.** When the result of an operation is used as an operand in a subsequent calculation, the value is sent immediately to its destination register. Yet at the same time, the value is put back on a bypass path to the ALU, thereby saving the time that otherwise would be required to retrieve the value for the next operation.

**7. Bandwidth Optimizations.** The 80960MC gets optimal use of its memory bus bandwidth because the bus is tuned for use with the cache: the line size of the instruction cache matches the maximum burst size for instruction fetches. The 80960MC automatically fetches four words in a burst and stores them directly in the cache. Due to the size of the cache and the fact that it is continually filled in anticipation of needed instructions in the program flow, the 80960MC is exceptionally insensitive to memory wait states. In fact, each wait state causes only a 7% degradation in system performance. The benefit is that the 80960MC will deliver outstanding performance even with a low cost memory system.

**8. Cache Bypass.** If there is a cache miss, the processor fetches the needed instruction, then sends it on to the instruction decoder at the same time it updates the cache. Thus, no extra time is taken to load and read the cache.

## Memory Space and Addressing Modes

The 80960MC allows each task (process) to address a logical memory space of up to 4 Gbytes. In turn, each task's address space is divided into four 1-Gbyte regions and each region can be mapped to physical addresses by zero, one, or two levels of page tables. The region with the highest addresses (Region 3) is common to all tasks.

In keeping with RISC design principles, the number of addressing modes has been kept to a minimum but includes all those necessary to ensure efficient execution of high-level languages such as Ada, C, and Fortran. Table 2 lists the memory addressing modes.

## Data Types

The 80960MC recognizes the following data types:

### Numeric:

- 8-, 16-, 32- and 64-bit ordinals
- 8-, 16, 32- and 64-bit integers
- 32-, 64- and 80-bit real numbers

### Non-Numeric:

- Bit
- Bit Field
- Triple-Word (96 bits)
- Quad-Word (128 bits)

## Large Register Set

The programming environment of the 80960MC includes a large number of registers. In fact, 36 registers are available at any time. The availability of this many registers greatly reduces the number of memory accesses required to execute most programs, which leads to greater instruction processing speed.

There are two types of general-purpose registers: local and global. The 20 global registers consist of sixteen 32-bit registers (G0 through G15) and four 80-bit registers (FP0 through FP3). These registers perform the same function as the general-purpose registers provided in other popular microprocessors. The term global refers to the fact that these registers retain their contents across procedure calls.

The local registers, on the other hand, are procedure specific. For each procedure call, the 80960MC allocates 16 local registers (R0 through R15). Each local register is 32 bits wide. Any register can also be used for floating-point operations; the 80-bit floating-point registers are provided for extended precision.

## Multiple Register Sets

To further increase the efficiency of the register set, multiple sets of local registers are stored on-chip. This cache holds up to four local register frames, which means that up to three procedure calls can be made without having to access the procedure stack resident in memory.

**Table 2. Memory Addressing Modes**

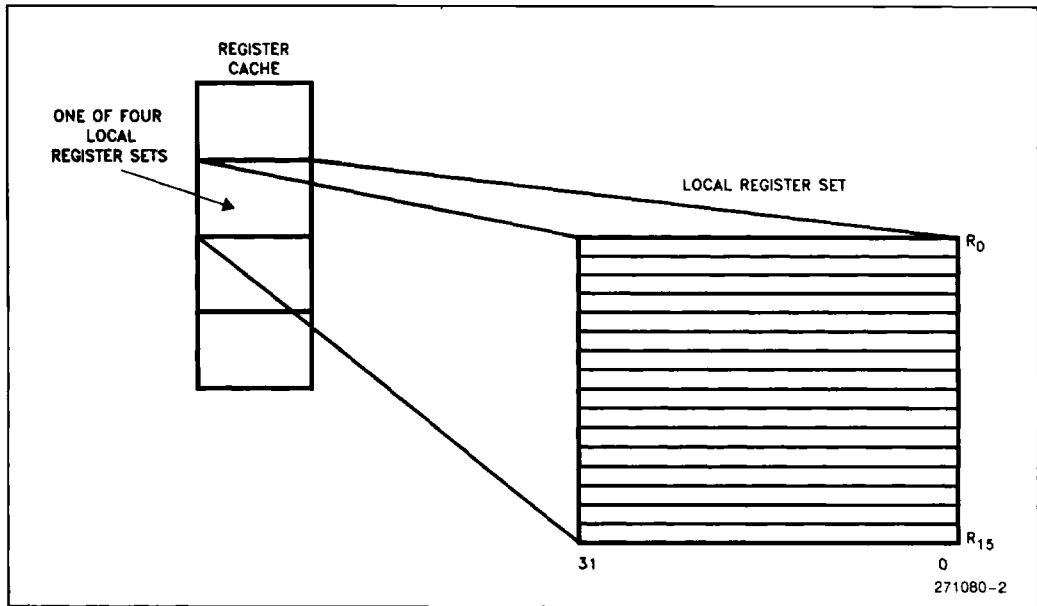
<ul style="list-style-type: none"> <li>• 12-Bit Offset</li> <li>• 32-Bit Offset</li> <li>• Register-Indirect</li> <li>• Register + 12-Bit Offset</li> <li>• Register + 32-Bit Offset</li> <li>• Register + (Index-Register × Scale-Factor)</li> <li>• Register × Scale Factor + 32-Bit Displacement</li> <li>• Register + (Index-Register × Scale-Factor) + 32-Bit Displacement</li> </ul> <p>Scale-Factor is 1, 2, 4, 8 or 16</p>
--

Although programs may have procedure calls nested many calls deep, a program typically oscillates back and forth between only two or three levels. As a result, with four stack frames in the cache, the probability of there being a free frame on the cache when a call is made is very high. In fact, runs of representative C-language programs show that 80% of the calls are handled without needing to access memory.

procedure stack in memory to make room for a new set of registers. Global register G15 is used by the processor as the frame pointer (FP) for the procedure stack.

Note that the global and floating-point registers are not exchanged on a procedure call, but retain their contents, making them available to all procedures for fast parameter passing. An illustration of the register cache is shown in Figure 4.

If there are four or more active procedures and a new procedure is called, the processor moves the oldest set of local registers in the register cache to a



**Figure 4. Multiple Register Sets Are Stored On-Chip**



### Instruction Cache

To further reduce memory accesses, the 80960MC includes a 512-byte on-chip instruction cache. The instruction cache is based on the concept of locality of reference; that is, most programs are not usually executed in a steady stream but consist of many branches and loops that lead to jumping back and forth within the same small section of code. Thus, by maintaining a block of instructions in a cache, the number of memory references required to read instructions into the processor can be greatly reduced.

To load the instruction cache, instructions are fetched in 16-byte blocks, so that up to four instructions can be fetched at one time. An efficient prefetch algorithm increases the probability that an instruction will already be in the cache when it is needed.

Code for small loops will often fit entirely within the cache, leading to a great increase in processing speed since further memory references might not be necessary until the program exits the loop. Similarly, when calling short procedures, the code for the calling procedure is likely to remain in the cache, so it will be there on the procedure's return.

### Register Scoreboarding

The instruction decoder has been optimized in several ways. One of these optimizations is the ability to do instruction overlapping by means of register scoreboarding.

Register scoreboarding occurs when a LOAD instruction is executed to move a variable from memory into a register. When the instruction is initiated, a scoreboard bit on the target register is set. When the register is actually loaded, the bit is reset. In between, any reference to the register contents is accompanied by a test of the scoreboard bit to insure that the load has completed before processing continues. Since the processor does not have to wait for the LOAD to be completed, it can go on to execute additional instructions placed in between the LOAD instruction and the instruction that uses the register contents, as shown in the following example:

```
LOAD R4, address 1
LOAD R5, address 2
Unrelated instruction
Unrelated instruction
ADD R4, R5, R6
```

In essence, the two unrelated instructions between the LOAD and ADD instructions are executed for

free (i.e., take no apparent time to execute) because they are executed while the register is being loaded. Up to three LOAD instructions can be pending at one time with three corresponding scoreboard bits set. By exploiting this feature, system programmers and compilers have a useful tool for optimizing execution speed.

### Memory Management and Protection

The 80960MC will be especially useful for multitasking applications that require software protection and a very large address space. To ensure the highest level of performance possible, the memory management unit and translation look-aside buffer (TLB) are contained on-chip.

The 80960MC supports a conventional form of demand-paged virtual memory in which the address space is divided into 4 Kbyte pages. Studies have shown that a 4 Kbyte page is the optimum size for a broad range of applications.

Each page table entry includes a 2-bit page rights field that specifies whether the page is a no-access, read-only, or read-write page. This field is interpreted differently depending on whether the current task (process) is executing in user or supervisor mode, as shown below:

Rights	User	Supervisor
00	No Access	Read-Only
01	No Access	Read-Write
10	Read-Only	Read-Write
11	Read-Write	Read-Write



### Floating-Point Arithmetic

In the 80960MC, floating-point arithmetic has been made an integral part of the architecture. Having the floating-point unit integrated on-chip provides two advantages. First, it improves the performance of the chip for floating-point applications, since no additional bus overhead is associated with floating-point calculations, thereby leaving more time for other bus operations such as I/O. Second, the cost of using floating-point operations is reduced because a separate coprocessor chip is not required.

The 80960MC floating-point (real number) data types include single-precision (32-bit), double-precision (64-bit), and extended precision (80-bit) floating-point numbers. Any register may be used to execute floating-point operations.

The processor provides hardware support for both mandatory and recommended portions of IEEE Standard 754 for floating-point arithmetic, including all arithmetic, exponential, logarithmic, and other transcendental functions. Table 3 shows execution times for some representative instructions.

**Table 3. Sample Floating-Point Execution Times ( $\mu$ s) at 25 MHz**

	32-Bit	64-Bit
Add	0.4	0.5
Subtract	0.4	0.5
Multiply	0.7	1.3
Divide	1.3	2.9
Square Root	3.7	3.9
Arctangent	10.1	13.1
Exponent	11.3	12.5
Sine	15.2	16.6
Cosine	15.2	16.6

## Multitasking Support

Multitasking programs commonly involve the monitoring and control of an external operation, such as the activities of a process controller or the movements of a machine tool. These programs generally consist of a number of processes that run independently of one another, but share a common database or pass data among themselves.

The 80960MC offers several hardware functions designed to support multitasking systems. One unique feature, called self-dispatching, allows a processor to switch itself automatically among scheduled tasks. When self-dispatching is used, all the operating system is required to do is place the task in the scheduling queue.

When the processor becomes available, it dispatches the task from the beginning of the queue and then executes it until it becomes blocked, interrupted, or until its time-slice expires. It then returns the task to the end of the queue (i.e., automatically reschedules it) and dispatches the next ready task.

During these operations, no communication between the processor and the operating system is necessary until the running task is complete or an interrupt is issued.

## Synchronization and Communication

The 80960MC also offers instructions to set up and test semaphores to ensure that concurrent tasks remain synchronized and no data inconsistency results. Special data structures, known as communication ports, provide the means for exchanging parameters and data structures. Transmission of information by means of communication ports is asynchronous and automatically buffered by the processor.

Communication between tasks by means of ports can be carried out independently of the operating system. Once the ports have been set up by the programmer, the processor handles the message passing automatically.

## High Bandwidth Local Bus

An 80960MC CPU resides on a high-bandwidth address/data bus known as the local bus (L-Bus). The L-Bus provides a direct communication path between the processor and the memory and I/O subsystem interfaces. The processor uses the local bus to fetch instructions, manipulate memory, and respond to interrupts. Its features include:

- 32-bit multiplexed address/data path
- Four-word burst capability, which allows transfers from 1 to 16 bytes at a time
- High bandwidth reads and writes at 66.7 MBytes per second
- Special signal to indicate whether a memory transaction can be cached

Figure 5 identifies the groups of signals which constitute the L-Bus. Table 4 lists the function of the L-Bus and other processor-support signals, such as the interrupt lines.

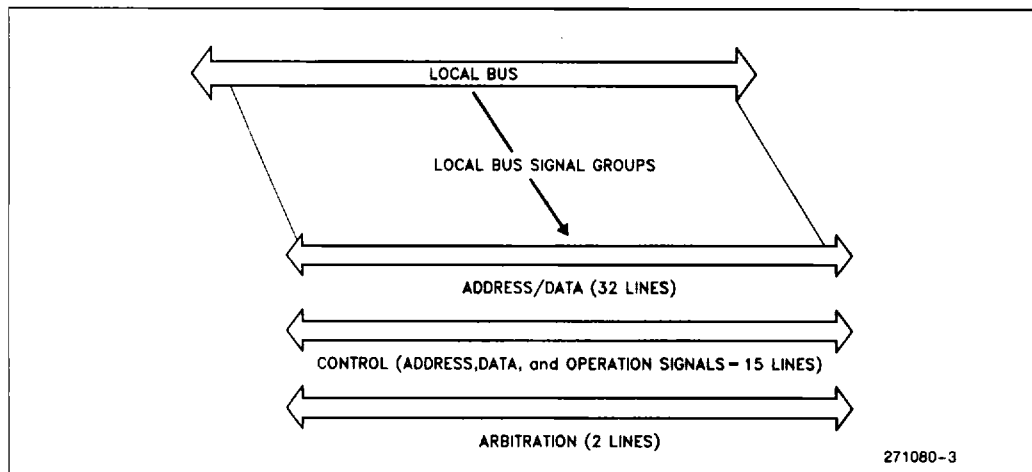


Figure 5. Local Bus Signal Groups

## Multiple Processor Support

One means of increasing the processing power of a system is to run two or more processors in parallel. Since microprocessors are not generally designed to run in tandem with other processors, designing such a system is usually difficult and costly.

The 80960MC solves this problem by offering a number of functions to coordinate the actions of multiple processors. First, messages can be passed between processors to initiate actions such as flushing a cache, stopping or starting another processor, or preempting a task. The messages are passed on the bus and allow multiple processors to run together smoothly, with rare need to lock the bus or memory.

Second, a set of synchronization instructions help maintain the coherency of memory. These instructions permit several processors to modify memory at the same time without inserting inaccuracies or ambiguities into shared data structures.

The self-dispatching mechanism, in addition to being used in single-processor systems, provides the means to increase the performance of a system merely by adding processors. Each processor can either work on the same pool of tasks (sharing the same queue with other processors) or can be restricted to its own queue.

When processors perform system operations, they synchronize themselves by using atomic operations and sending special messages between each other. And changing the number of processors in a system

never requires a software change. Software will execute correctly regardless of the number of processors in the system; systems with more processors simply execute faster.

## Interrupt Handling

The 80960MC can be interrupted in one of two ways: by the activation of one of four interrupt pins or by sending a message on the processor's data bus.

The 80960MC is unusual in that it automatically handles interrupts on a priority basis and tracks pending interrupts through its on-chip interrupt controller. Two of the interrupt pins can be configured to provide M8259A handshaking for expansion beyond four interrupt lines.

An interrupt message is made up of a vector number and an interrupt priority. If the interrupt priority is greater than that of the currently running task, the processor accepts the interrupt and uses the vector as an index into the interrupt table. If the priority of the interrupt message is below that of the current task, the processor saves the information in a section of the interrupt table reserved for pending interrupts.

## Debug Features

The 80960MC has built-in debug capabilities. There are two types of breakpoints and six different trace modes. The debug features are controlled by two

internal 32-bit registers, the Process-Controls Word and the Trace-Controls Word. By setting bits in these control words, a software debug monitor can closely control how the processor responds during program execution.

The 80960MC has both hardware and software breakpoints. It provides two hardware breakpoint registers on-chip which can be set by a special command to any value. When the instruction pointer matches the value in one of the breakpoint registers, the breakpoint will fire, and a breakpoint handling routine is called automatically.

The 80960MC also provides software breakpoints through the use of two instructions, MARK and FMARK. These instructions can be placed at any point in a program and will cause the processor to halt execution at that point and call the breakpoint handling routine. The breakpoint mechanism is easy to use and provides a powerful debugging tool.

Tracing is available for instructions (single-step execution), calls and returns, and branching. Each different type of trace may be enabled separately by a special debug instruction. In each case, the 80960MC executes the instruction first and then calls a trace handling routine (usually part of a software debug monitor). Further program execution is halted until the trace routine is completed. When the trace event handling routine is completed, instruction execution resumes at the next instruction. The 80960MC's tracing mechanisms, which are implemented completely in hardware, greatly simplify the task of testing and debugging software.

## FAULT DETECTION

The 80960MC has an automatic mechanism to handle faults. There are ten fault types including trace, arithmetic, and floating-point faults. When the processor detects a fault, it automatically calls the appropriate fault handling routine and saves the current instruction pointer and necessary state information to make efficient recovery possible. The processor posts diagnostic information on the type of fault to a Fault Record. Like interrupt handling routines, fault handling routines are usually written to meet the needs of a specific application and are often included as part of the operating system or kernel.

For each of the ten fault types, there are numerous subtypes that provide specific information about a fault. For example, a floating-point fault may have its subtype set to an Overflow or Zero-Divide fault. The fault handler can use this specific information to respond correctly to the fault.

## Interagent Communications (IAC)

In order to coordinate their actions, processors in a multiple processor system need a means for communicating with each other. The 80960MC does this through a mechanism known as Interagent Communication messages or IACs.

IAC messages cause a variety of actions including starting and stopping processors, flushing instruction caches and TLBs, and sending interrupts to other processors in the system. The upper 16 Mbytes of the processor's physical memory space is reserved for sending and receiving IAC messages.

## BUILT-IN TESTABILITY

Upon reset, the 80960MC automatically conducts an exhaustive internal test of its major blocks of logic.

Then, before executing its first instruction, it does a zero check sum on the first eight words in memory to ensure that the system has been loaded correctly. If a problem is discovered at any point during the self-test, the 80960MC will assert its FAILURE pin and will not begin program execution. The self-test takes approximately 47,000 cycles to complete.

System manufacturers can use the 80960MC's self-test feature during incoming parts inspection. No special diagnostic programs need to be written, and the test is both thorough and fast. The self-test capability helps ensure that defective parts will be discovered before systems are shipped, and once in the field, the self-test makes it easier to distinguish between problems caused by processor failure and problems resulting from other causes.

## COMPATIBILITY WITH 80960K-SERIES

Application programs written for the 80960K-Series microprocessors can be run on the 80960MC without modification. The 80960K-Series instruction set forms the core of the 80960MC's instructions, so binary compatibility is assured.

**CHMOS**

The 80960MC is fabricated using Intel's CHMOS IV (Complementary High Speed Metal Oxide Semiconductor) process. This advanced technology eliminates the frequency and reliability limitations of older

CMOS processes and opens a new era in micro-processor performance. It combines the high performance capabilities of Intel's industry-leading HMOS technology with the high density and low power characteristics of CMOS. The 80960MC is available at 16, 20 and 25 MHz.

**Table 4a. 80960MC Pin Description: L-Bus Signals**

Symbol	Type	Name and Function															
CLK2	I	<b>SYSTEM CLOCK</b> provides the fundamental timing for 80960MC systems. It is divided by two inside the 80960MC to generate the internal processor clock. CLK2 is shown in Figure 9.															
LAD <sub>31</sub> -LAD <sub>0</sub>	I/O T.S.	<p><b>LOCAL ADDRESS/DATA BUS</b> carries 32-bit physical addresses and data to and from memory. During an address (<math>T_a</math>) cycle, bits 2-31 contain a physical word address (bits 0-1 indicate SIZE; see below). During a data (<math>T_d</math>) cycle, bits 0-31 contain read or write data. The LAD lines are active HIGH and float to a high impedance state when not active.</p> <p><b>SIZE</b>, which is comprised of bits 0-1 of the LAD lines during a <math>T_a</math> cycle, specifies the size of a transfer in words for a burst transaction.</p> <table border="0" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;"><u>LAD</u> 1</td> <td style="text-align: center;"><u>LAD</u> 0</td> <td></td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1 Word</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">2 Words</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">3 Words</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">4 Words</td> </tr> </table>	<u>LAD</u> 1	<u>LAD</u> 0		0	0	1 Word	0	1	2 Words	1	0	3 Words	1	1	4 Words
<u>LAD</u> 1	<u>LAD</u> 0																
0	0	1 Word															
0	1	2 Words															
1	0	3 Words															
1	1	4 Words															
ALE	O T.S.	<b>ADDRESS-LATCH ENABLE</b> indicates the transfer of a physical address. ALE is asserted during a $T_a$ cycle and deasserted before the beginning of the $T_d$ state. It is active LOW and floats to a high impedance state when the processor is idle or is at the end of any bus access.															
ADS	O O.D.	<b>ADDRESS STATUS</b> indicates an address state. ADS is asserted every $T_a$ state and deasserted during the the following $T_d$ state. For a burst transaction, ADS is asserted again every $T_d$ state where READY was asserted in the previous cycle.															
W/R	O O.D.	<b>WRITE/READ</b> specifies, during a $T_a$ cycle, whether the operation is a write or read. It is latched on-chip and remains valid during $T_d$ and $T_w$ states.															
DT/R	O O.D.	<b>DATA TRANSMIT/RECEIVE</b> indicates the direction of data transfer to and from the L-Bus. It is low during $T_a$ , $T_w$ and $T_d$ cycles for a read or interrupt acknowledgement; it is high during $T_a$ , $T_w$ and $T_d$ cycles for a write. DT/R never changes state when DEN is asserted (see Timing Diagrams).															
DEN	O O.D.	<b>DATA ENABLE</b> is asserted during $T_d$ and $T_w$ cycles and indicates transfer of data on the LAD bus lines.															
READY	I	<b>READY</b> indicates that data on LAD lines can be sampled or removed. If READY is not asserted during a $T_d$ cycle, the $T_d$ cycle is extended to the next cycle by inserting wait states ( $T_w$ ), and ADS is not asserted in the next cycle.															
LOCK	I/O O.D.	<p><b>BUS LOCK</b> prevents other bus masters from gaining control of the L-Bus following the current cycle (if they would assert LOCK to do so). LOCK is used by the processor or any bus agent when it performs indivisible Read/Modify/Write (RMW) operations.</p> <p>For a read that is designated as a RMW-read, LOCK is examined. if asserted, the processor waits until it is not asserted; if not asserted, the processor asserts LOCK during the <math>T_a</math> cycle and leaves it asserted.</p> <p>A write that is designated as an RMW-write deasserts LOCK in the <math>T_a</math> cycle.</p>															

I/O = Input/Output, O = Output, I = Input, O.D. = Open-Drain, T.S. = three state  
 $T_a$  =  $T_{Address}$ ,  $T_d$  =  $T_{Data}$ ,  $T_w$  =  $T_{Wait}$ ,  $T_r$  =  $T_{Recovery}$ ,  $T_i$  =  $T_{Idle}$ ,  $T_h$  =  $T_{Hold}$

Table 4a. 80960MC Pin Description: L-Bus Signals (Continued)

Symbol	Type	Name and Function
$\overline{BE}_3 - \overline{BE}_0$	O O.D.	<p><b>BYTE ENABLE LINES</b> specify which data bytes (up to four) on the bus take part in the current bus cycle. <math>\overline{BE}_3</math> corresponds to LAD<sub>31</sub>-LAD<sub>24</sub> and <math>\overline{BE}_0</math> corresponds to LAD<sub>7</sub>-LAD<sub>0</sub>.</p> <p>The byte enables are provided in advance of data. The byte enables asserted during <math>T_a</math> specify the bytes of the first data word. The byte enables asserted during <math>T_d</math> specify the bytes of the next data word (if any), that is, the word to be transmitted following the next assertion of <b>READY</b>. The byte enables during the <math>T_d</math> cycles preceding the last assertion of <b>READY</b> are undefined. The byte enables are latched on-chip and remain constant from one <math>T_d</math> cycle to the next when <b>READY</b> is not asserted.</p> <p>For reads, the byte enables specify the byte(s) that the processor will actually use. 80960MC's will assert only adjacent byte enables (e.g., asserting just <math>\overline{BE}_0</math> and <math>\overline{BE}_2</math> is not permitted), and are required to assert at least one byte enable. Accesses must also be naturally aligned (e.g., asserting <math>\overline{BE}_1</math> and <math>\overline{BE}_2</math> is not allowed even though they are adjacent). To produce address bits <math>A_0</math> and <math>A_1</math> externally, they can be decoded from the byte enables.</p>
HOLD (HLDAR)	I	<p><b>HOLD</b> indicates a request from a secondary bus master to acquire the bus. If the processor is initialized as the primary bus master this input will be interpreted as <b>HOLD</b>. When the processor receives <b>HOLD</b> and grants another master control of the bus, it floats its three-state bus lines, asserts <b>HOLD ACKNOWLEDGE</b>, and enters the <math>T_h</math> state. When <b>HOLD</b> is deasserted, the processor will deassert <b>HOLD ACKNOWLEDGE</b> and go to either the <math>T_i</math> or <math>T_a</math> state.</p> <p><b>HOLD ACKNOWLEDGE RECEIVED</b> indicates that the processor has acquired the bus. If the processor is initialized as the secondary bus master this input is interpreted as <b>HLDAR</b>.</p> <p>HOLD timing is shown in Figure 11.</p>
HLDA (HOLDR)	O T.S.	<p><b>HOLD ACKNOWLEDGE</b> relinquishes control of the bus to another bus master. If the processor is initialized as the primary bus master this output will be interpreted as <b>HLDA</b>. When <b>HOLD</b> is deasserted, the processor will deassert <b>HLDA</b> and go to either the <math>T_i</math> or <math>T_a</math> state.</p> <p><b>HOLD REQUEST</b> indicates a request to acquire the bus. If the processor is initialized as the secondary bus master this output will be interpreted as <b>HOLDR</b>.</p> <p>HOLD timing is shown in Figure 11.</p>
CACHE	O T.S.	<p><b>CACHE</b> indicates if an access is cacheable during a <math>T_a</math> cycle. The <b>CACHE</b> signal floats to a high impedance state when the processor is idle.</p>

I/O = Input/Output, O = Output, I = Input, O.D. = Open-Drain, T.S. = three state

$T_a$  =  $T_{Address}$ ,  $T_d$  =  $T_{Data}$ ,  $T_W$  =  $T_{Wait}$ ,  $T_r$  =  $T_{Recovery}$ ,  $T_i$  =  $T_{Idle}$ ,  $T_h$  =  $T_{Hold}$

Table 4b. 80960MC Pin Description: Module Support Signals

Symbol	Type	Name and Function
BADAC	I	<p><b>BAD ACCESS</b>, if asserted in the cycle following the one in which the last <b>READY</b> of a transaction is asserted, indicates that an unrecoverable error has occurred on the current bus transaction, or that a synchronous load/store instruction has not been acknowledged.</p> <p><b>STARTUP:</b> During system reset, the <b>BADAC</b> signal is interpreted differently. If the signal is high, it indicates that this processor will perform system initialization. If it is low, another processor in the system will perform system initialization instead.</p>
RESET	I	<p><b>RESET</b> clears the internal logic of the processor and causes it to re-initialize.</p> <p>During <b>RESET</b> assertion, the input pins are ignored (except for <b>BADAC</b> and <b>IAC/INT<sub>0</sub></b>), the tri-state output pins are placed in a high impedance state, and other output pins are placed in their non-asserted state.</p> <p><b>RESET</b> must be asserted for at least 41 CLK2 cycles for a predictable <b>RESET</b>. The HIGH to LOW transition of <b>RESET</b> should occur after the rising edge of both CLK2 and the external bus CLK, and before the next rising edge of CLK2.</p> <p><b>RESET</b> timing is shown in Figure 10.</p>
FAILURE	O O.D.	<p><b>INITIALIZATION FAILURE</b> indicates that the processor has failed to initialize correctly. After <b>RESET</b> is deasserted and before the first bus transaction begins, <b>FAILURE</b> is asserted while the processor performs a self-test. If the self-test completes successfully, then <b>FAILURE</b> is deasserted. Next, the processor performs a zero checksum on the first eight words of memory. If it fails, <b>FAILURE</b> is asserted for a second time and remains asserted; if it passes, system initialization continues and <b>FAILURE</b> remains deasserted.</p>
N.C.	N/A	<p><b>NOT CONNECTED</b> indicates pins should not be connected. Never connect any pin marked N.C.</p>
$\overline{\text{IAC}}$ ( $\overline{\text{INT}}_0$ )	I	<p><b>INTERAGENT COMMUNICATION REQUEST/INTERRUPT 0</b> indicates either that there is a pending IAC message for the processor or an interrupt. The bus interrupt control register determines in which way the signal should be interpreted. To signal an interrupt or IAC request in a synchronous system, this pin (as well as the other interrupt pins) must be enabled by being deasserted for at least one bus cycle and then asserted for at least one additional bus cycle; in an asynchronous system, the pin must remain deasserted for at least two bus cycles and then be asserted for at least two more bus cycles.</p> <p><b>LOCAL PROCESSOR NUMBER:</b> This signal is interpreted differently during system reset. If the signal is at a high voltage level, it indicates that this processor is a primary bus master (Local Processor Number = 0); if it is at a low voltage level, it indicates that this processor is a secondary bus master (Local Processor Number = 1).</p>
INT <sub>1</sub>	I	<p><b>INTERRUPT 1</b>, like <math>\overline{\text{INT}}_0</math>, provides direct interrupt signaling.</p>
INT <sub>2</sub> (INTR)	I	<p><b>INTERRUPT 2/INTERRUPT REQUEST:</b> The bus control registers determines how this pin is interpreted. If INT<sub>2</sub>, it has the same interpretation as the <math>\overline{\text{INT}}_0</math> and INT<sub>1</sub> pins. If INTR, it is used to receive an interrupt request from an external interrupt controller.</p>
$\overline{\text{INT}}_3$ ( $\overline{\text{INTA}}$ )	I/O O.D.	<p><b>INTERRUPT 3/INTERRUPT ACKNOWLEDGE:</b> The bus interrupt control register determines how this pin is interpreted. If <math>\overline{\text{INT}}_3</math>, it has the same interpretation as the <math>\overline{\text{INT}}_0</math>, INT<sub>1</sub>, and INT<sub>2</sub> pins. If <math>\overline{\text{INTA}}</math>, it is used as an output to control interrupt-acknowledge bus transactions. The <math>\overline{\text{INTA}}</math> output is latched on-chip and remains valid during T<sub>d</sub> cycles; as an output, it is open-drain.</p>

I/O = Input/Output, O = Output, I = Input, O.D. = Open-Drain, T.S. = three state  
 T<sub>a</sub> = T<sub>Address</sub>, T<sub>d</sub> = T<sub>Data</sub>, T<sub>w</sub> = T<sub>Wait</sub>, T<sub>r</sub> = T<sub>Recovery</sub>, T<sub>1</sub> = T<sub>Idle</sub>, T<sub>h</sub> = T<sub>Hold</sub>

## ELECTRICAL SPECIFICATIONS

### Power and Grounding

The 80960MC is implemented in CHMOS III technology and has modest power requirements. Its high clock frequency and numerous output buffers (address/data, control, error and arbitration signals) can cause power surges as multiple output buffers drive new signal levels simultaneously. For clean on-chip power distribution at high frequency, 12  $V_{CC}$  and 13  $V_{SS}$  pins separately feed functional units of the 80960MC.

Power and ground connections must be made to all power and ground pins of the 80960MC. On the circuit board, all  $V_{CC}$  pins must be strapped closely together, preferably on a power plane. Likewise, all  $V_{SS}$  pins should be strapped together, preferably on a ground plane.

### Power Decoupling Recommendations

Liberal decoupling capacitance should be placed near the 80960MC. The processor can cause transient power surges when driving the L-Bus, particularly when it is connected to a large capacitive load.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance can be reduced by shortening the board traces between the processor and decoupling capacitors as much as possible.

### Connection Recommendations

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, if

one or more interrupt lines are not used, they should be pulled up or down to their respective deasserted states. No inputs should ever be left floating.

All open-drain outputs require a pullup device. While in some cases a simple pullup resistor will be adequate, we recommend a network of pullup and pull-down resistors biased to a valid  $V_{IH}$  ( $\geq 3.4V$ ) and terminated in the characteristic impedance of the circuit board. Figure 6 shows our recommendations for the resistor values for both a low and high current drive network, which assumes that the circuit board has a characteristic impedance of  $100\Omega$ . The advantage of terminating the output signals in this fashion is that it limits signal swing and reduces AC power consumption.

### Characteristic Curves

Figure 7 shows the typical supply current requirements over the operating temperature range of the processor at supply voltage ( $V_{CC}$ ) of 5V. Figure 8 shows the typical power supply current ( $I_{CC}$ ) required by the 80960MC at various operating frequencies when measured at three input voltage ( $V_{CC}$ ) levels.

Figure 9 shows the typical capacitive derating curve for the 80960MC measured from 1.5V on the system clock (CLK) to 0.8V on the falling edge and 2.0V on the rising edge of the L-Bus address/data (LAD) signals.

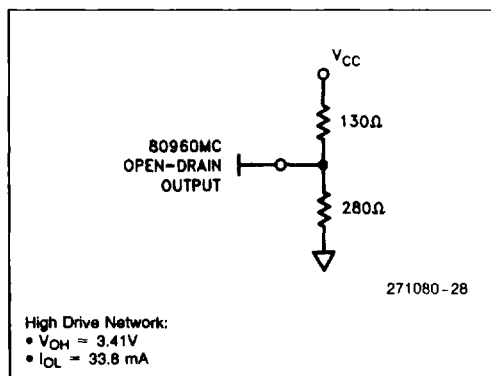
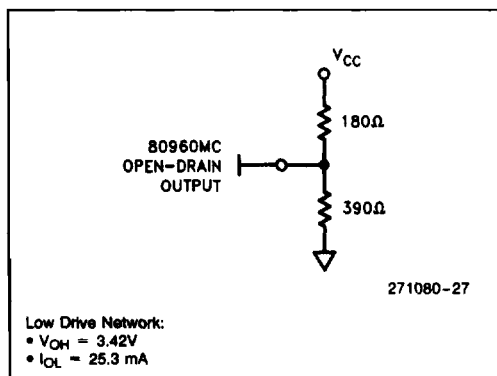


Figure 6. Connection Recommendations for Low and High Current Drive Networks



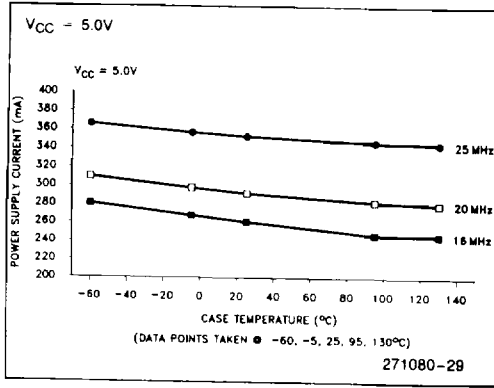


Figure 7. Typical Supply Current (I<sub>CC</sub>)

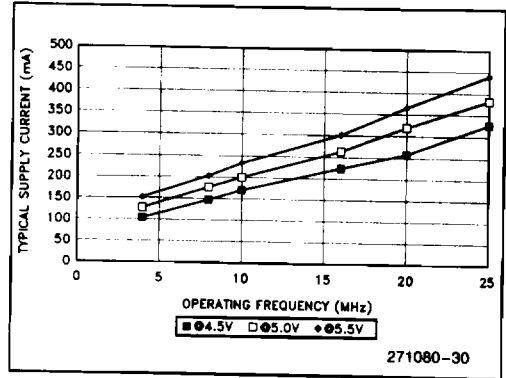


Figure 8. Typical Current vs Frequency

Test Load Circuit

Figure 10 illustrates the load circuit used to test the 80960MC's tristate pins, and Figure 11 shows the load circuit used to test the open drain outputs. The open drain test uses an active load circuit in the form of a matched diode bridge. Since the open-drain outputs sink current, only the I<sub>OL</sub> legs of the bridge are necessary and the I<sub>OH</sub> legs are not used. When the 80960MC driver under test is turned off, the output pin is pulled up to V<sub>REF</sub> (i.e., V<sub>OH</sub>). Diode D<sub>1</sub> is turned off and the I<sub>OL</sub> current source flows through diode D<sub>2</sub>.

When the 80960MC open-drain driver under test is on, diode D<sub>1</sub> is also on, and the voltage on the pin being tested drops to V<sub>OL</sub>. Diode D<sub>2</sub> turns off and I<sub>OL</sub> flows through diode D<sub>1</sub>.

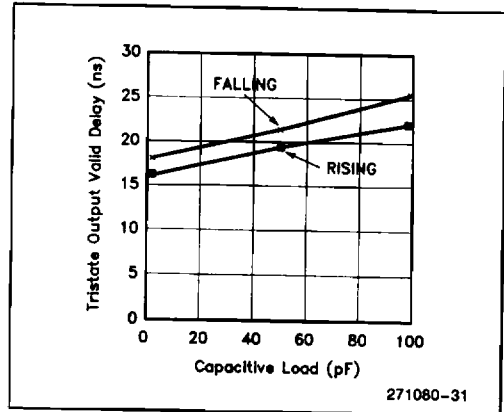


Figure 9. Capacitive Derating Curve

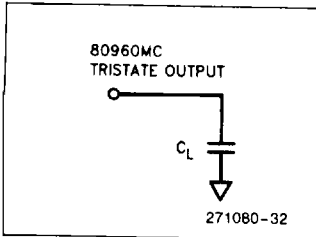


Figure 10. Test Load Circuit for TRI-STATE Output Pins

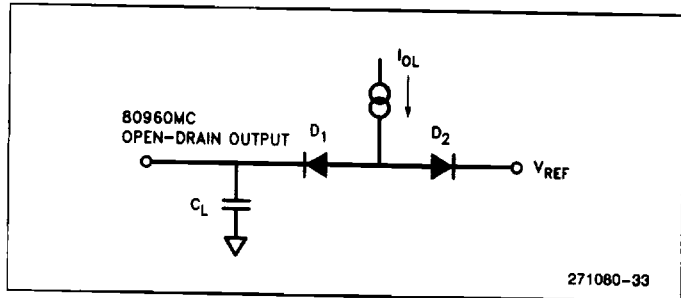


Figure 11. Test Load Circuit for Open-Drain Output Pins

5

**ABSOLUTE MAXIMUM RATINGS\***

Case Temperature under Bias<sup>(7)</sup> ..... -55°C to +125°C  
 Storage Temperature ..... -65°C to +150°C  
 Voltage on Any Pin ..... -0.5V to V<sub>CC</sub> + 0.5V  
 Power Dissipation ..... 2.6W (25 MHz)

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice.

*\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

**D.C. CHARACTERISTICS**

80960MC: T<sub>CASE</sub><sup>(6)</sup> = -55°C to +125°C, V<sub>CC</sub> = 5V ± 5%

Symbol	Parameter	Min	Max	Units	Test Conditions
V <sub>IL</sub>	Input Low Voltage	-0.3	+0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0	V <sub>CC</sub> + 0.3	V	
V <sub>CL</sub>	CLK2 Input Low Voltage	-0.3	+0.8	V	
V <sub>CH</sub>	CLK2 Input High Voltage	0.55 V <sub>CC</sub>	V <sub>CC</sub> + 0.3	V	
V <sub>OL</sub>	Output Low Voltage		0.45	V	(1, 5)
V <sub>OH</sub>	Output High Voltage	2.4		V	(2, 4)
I <sub>CC</sub>	Power Supply Current: 16 MHz 20 MHz 25 MHz		375 420 480	mA	
I <sub>LI</sub>	Input Leakage Current		±15	µA	0 ≤ V <sub>O</sub> ≤ V <sub>CC</sub>
I <sub>LO</sub>	Output Leakage Current		±15	µA	0.45 ≤ V <sub>O</sub> ≤ V <sub>CC</sub>
C <sub>IN</sub>	Input Capacitance		10	pF	f <sub>C</sub> = 1 MHz <sup>(3)</sup>
C <sub>O</sub>	I/O or Output Capacitance		12	pF	f <sub>C</sub> = 1 MHz <sup>(3)</sup>
C <sub>CLK</sub>	Clock Capacitance		10	pF	f <sub>C</sub> = 1 MHz <sup>(3)</sup>
θ <sub>JA</sub>	Thermal Resistance (Junction-to-Ambient) Pin Grid Array Ceramic Quad Flatpack		21 29	°C/W °C/W	
θ <sub>JC</sub>	Thermal Resistance (Junction-to-Case) Pin Grid Array Ceramic Quad Flatpack		4 8	°C/W °C/W	

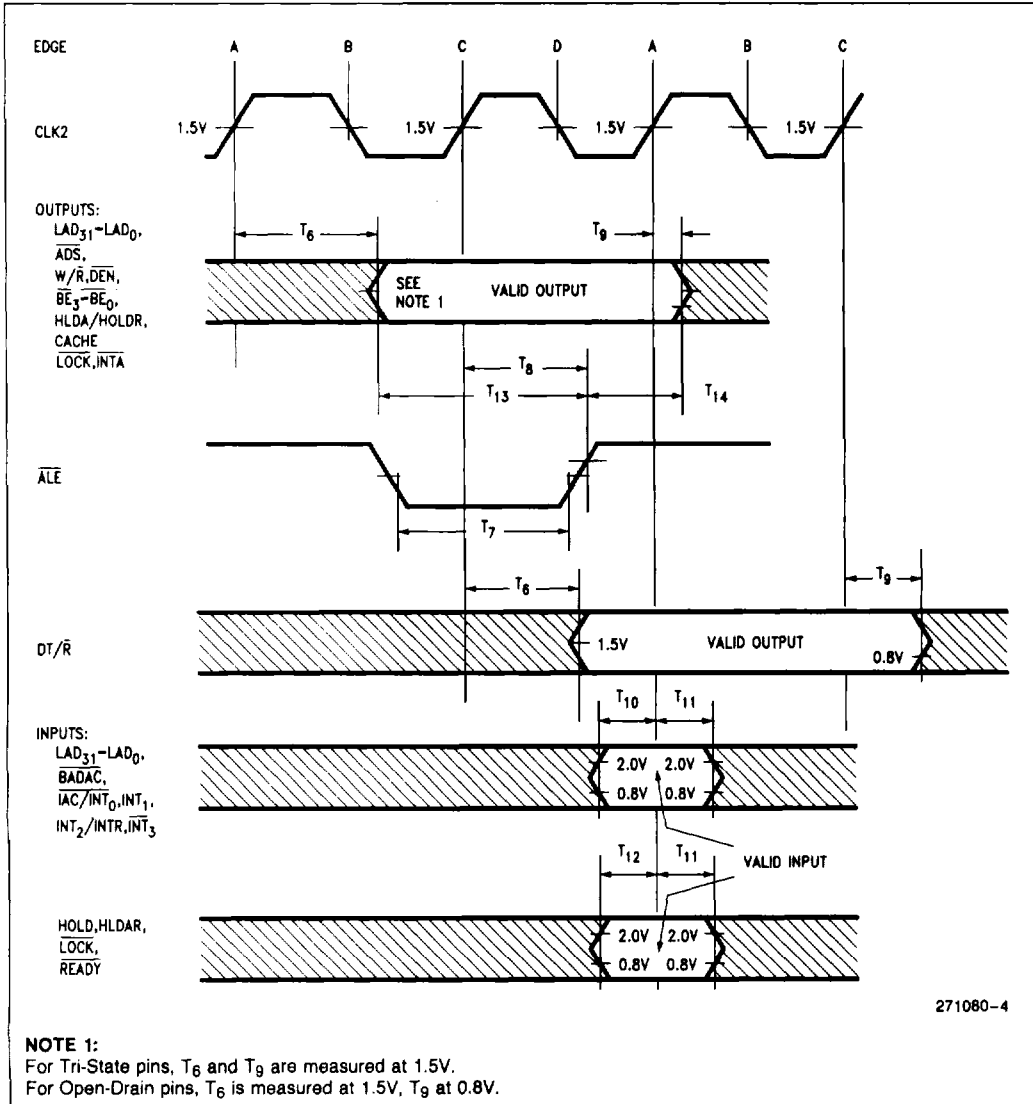
**NOTES:**

1. For three-state outputs, this parameter is measured at:  
 Address/Data ..... 4.0 mA  
 Controls ..... 5.0 mA
2. This parameter is measured at:  
 Address/Data ..... 1.0 mA  
 Controls ..... 0.9 mA  
 ALE ..... 5.0 mA
3. Input, output, and clock capacitance are not tested.
4. Not measured on open-drain outputs.
5. For open-drain outputs ..... 25 mA
6. Case temperatures are "instant on".

**AC SPECIFICATIONS**

This section describes the AC specifications for the 80960MC pins. All input and output timings are specified relative to the 1.5V level of the rising edge of CLK2, and refer to the time at which the signal

reaches (for output delay and input setup) or leaves (for hold time) the TTL levels of LOW (0.8V) or HIGH (2.0V). All AC testing should be done with input voltages of 0.4V and 2.4V, except for the clock (CLK2), which should be tested with input voltages of 0.45V and 0.55 V<sub>CC</sub>.



**Figure 12. Drive Levels and Timing Relationships for 80960MC Signals**

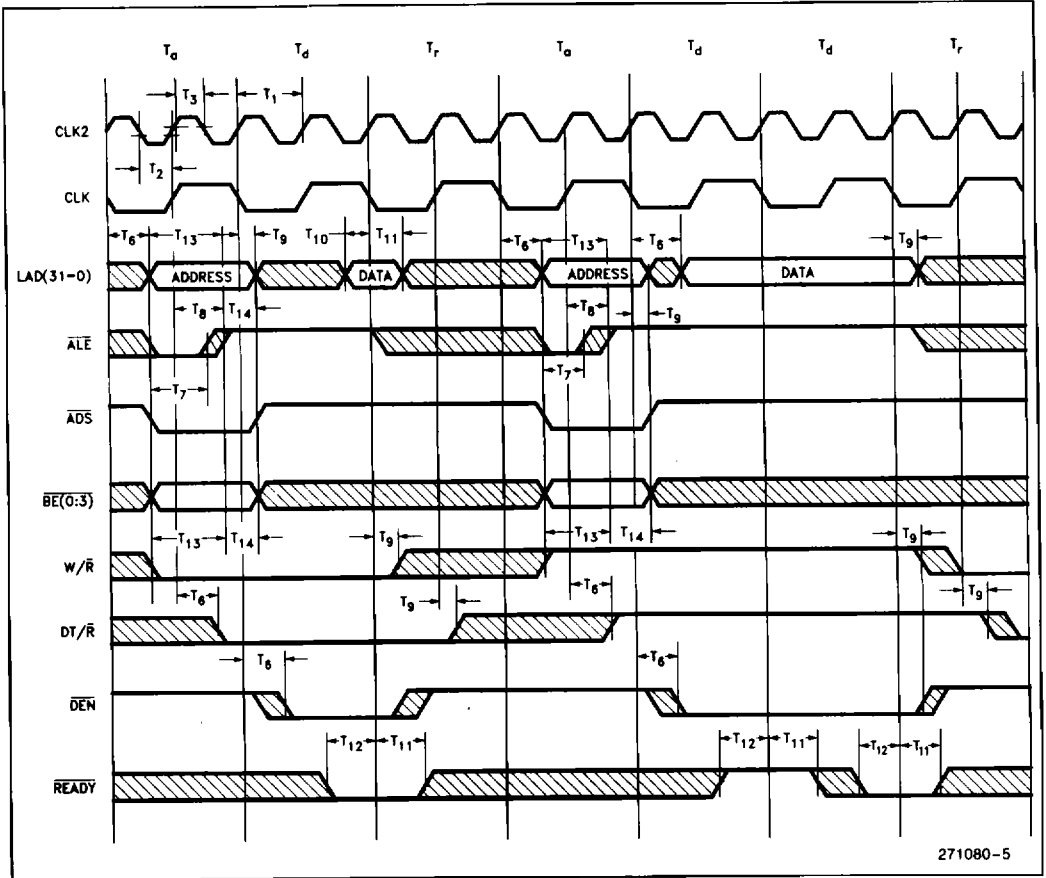


Figure 13. Timing Relationship of L-Bus Signals

**A.C. Specification Tables**

80960MC A.C. Characteristics (16 MHz)

 $T_{CASE}^{(3)} = -55^{\circ}\text{C to } +125^{\circ}\text{C}, V_{CC} = 5\text{V} \pm 5\%$ 

Symbol	Parameter	Min	Max	Units	Test Conditions
T <sub>1</sub>	Processor Clock Period (CLK2)	31.25	125	ns	V <sub>IN</sub> = 1.5V
T <sub>2</sub>	Processor Clock Low Time (CLK2)	8		ns	V <sub>IL</sub> = 10% Point = 1.2V
T <sub>3</sub>	Processor Clock High Time (CLK2)	8		ns	V <sub>IH</sub> = 90% Point = 0.1V + 0.5 V <sub>CC</sub>
T <sub>4</sub>	Processor Clock Fall Time (CLK2)		10	ns	V <sub>IN</sub> = 90% Point to 10% Point
T <sub>5</sub>	Processor Clock Rise Time (CLK2)		10	ns	V <sub>IN</sub> = 10% Point to 90% Point
T <sub>6</sub>	Output Valid Delay	2	25	ns	C <sub>L</sub> = 100 pF (LAD) C <sub>L</sub> = 75 pF (Controls)
T <sub>6H</sub>	HOLDA Output Valid Delay	4	31	ns	C <sub>L</sub> = 75 pF
T <sub>7</sub>	ALE Width	15		ns	C <sub>L</sub> = 75 pF
T <sub>8</sub>	ALE Invalid Delay	0	20	ns	C <sub>L</sub> = 75 pF <sup>(2)</sup>
T <sub>9</sub>	Output Float Delay	2	20	ns	C <sub>L</sub> = 100 pF (LAD) C <sub>L</sub> = 75 pF (Controls) <sup>(2)</sup>
T <sub>9H</sub>	HOLDA Output Float Delay	4	20	ns	C <sub>L</sub> = 75 pF
T <sub>10</sub>	Input Setup 1	3		ns	(Note 1)
T <sub>11</sub>	Input Hold	5		ns	(Note 1)
T <sub>11H</sub>	HOLD Input Hold	4		ns	
T <sub>12</sub>	Input Setup 2	8		ns	
T <sub>13</sub>	Setup to ALE Inactive	10		ns	C <sub>L</sub> = 100 pF (LAD) C <sub>L</sub> = 75 pF (Controls)
T <sub>14</sub>	Hold after ALE Inactive	8		ns	C <sub>L</sub> = 100 pF (LAD) C <sub>L</sub> = 75 pF (Controls)
T <sub>15</sub>	Reset Hold	3		ns	
T <sub>16</sub>	Reset Setup	5		ns	
T <sub>17</sub>	Reset Width	1281		ns	41 CLK2 Periods Minimum

**NOTES:**

 1. IAC/INT<sub>0</sub>, INT<sub>1</sub>, INT<sub>2</sub>/INTR, INT<sub>3</sub> can be asynchronous.

 2. A float condition occurs when the maximum output current becomes less than I<sub>LO</sub>. Float delay is not tested, but should be no longer than the valid delay.

3. Case temperatures are "instant on".

**A.C. Specification Tables** (Continued)

80960MC A.C. Characteristics (20 MHz)

 $T_{CASE}^{(3)} = -55^{\circ}\text{C to } +125^{\circ}\text{C}, V_{CC} = 5\text{V} \pm 5\%$ 

Symbol	Parameter	Min	Max	Units	Test Conditions
T <sub>1</sub>	Processor Clock Period (CLK2)	25	125	ns	V <sub>IN</sub> = 1.5V
T <sub>2</sub>	Processor Clock Low Time (CLK2)	6		ns	V <sub>IL</sub> = 10% Point = 1.2V
T <sub>3</sub>	Processor Clock High Time (CLK2)	6		ns	V <sub>IH</sub> = 90% Point = 0.1V + 0.5 V <sub>CC</sub>
T <sub>4</sub>	Processor Clock Fall Time (CLK2)		10	ns	V <sub>IN</sub> = 90% Point to 10% Point
T <sub>5</sub>	Processor Clock Rise Time (CLK2)		10	ns	V <sub>IN</sub> = 10% Point to 90% Point
T <sub>6</sub>	Output Valid Delay	2	20	ns	C <sub>L</sub> = 60 pF (LAD) C <sub>L</sub> = 50 pF (Controls)
T <sub>6H</sub>	HOLDA Output Valid Delay	4	26	ns	C <sub>L</sub> = 50 pF
T <sub>7</sub>	ALE Width	12		ns	C <sub>L</sub> = 50 pF
T <sub>8</sub>	ALE Invalid Delay	0	20	ns	C <sub>L</sub> = 50 pF <sup>(2)</sup>
T <sub>9</sub>	Output Float Delay	2	20	ns	C <sub>L</sub> = 60 pF (LAD) C <sub>L</sub> = 50 pF (Controls) <sup>(2)</sup>
T <sub>9H</sub>	HOLDA Output Float Delay	4	20	ns	C <sub>L</sub> = 50 pF
T <sub>10</sub>	Input Setup 1	3		ns	(Note 1)
T <sub>11</sub>	Input Hold	5		ns	(Note 1)
T <sub>11H</sub>	HOLD Input Hold	4		ns	
T <sub>12</sub>	Input Setup 2	7		ns	
T <sub>13</sub>	Setup to ALE Inactive	10		ns	C <sub>L</sub> = 60 pF (LAD) C <sub>L</sub> = 50 pF (Controls)
T <sub>14</sub>	Hold after ALE Inactive	8		ns	C <sub>L</sub> = 60 pF (LAD) C <sub>L</sub> = 50 pF (Controls)
T <sub>15</sub>	Reset Hold	3		ns	
T <sub>16</sub>	Reset Setup	5		ns	
T <sub>17</sub>	Reset Width	1025		ns	41 CLK2 Periods Minimum

**NOTES:**

 1. IAC/INT<sub>0</sub>, INT<sub>1</sub>, INT<sub>2</sub>/INTR, INT<sub>3</sub> can be asynchronous.

 2. A float condition occurs when the maximum output current becomes less than I<sub>LO</sub>. Float delay is not tested, but should be no longer than the valid delay.

3. Case temperatures are "instant on".

**A.C. Specification Tables** (Continued)

80960MC A.C. Characteristics (25 MHz)

 $T_{CASE}^{(3)} = -55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 5\%$ 

Symbol	Parameter	Min	Max	Units	Test Conditions
$T_1$	Processor Clock Period (CLK2)	20	125	ns	$V_{IN} = 1.5\text{V}$
$T_2$	Processor Clock Low Time (CLK2)	5		ns	$V_{IL} = 10\%$ Point = 1.2V
$T_3$	Processor Clock High Time (CLK2)	5		ns	$V_{IH} = 90\%$ Point = $0.1\text{V} + 0.5 V_{CC}$
$T_4$	Processor Clock Fall Time (CLK2)		10	ns	$V_{IN} = 90\%$ Point to 10% Point
$T_5$	Processor Clock Rise Time (CLK2)		10	ns	$V_{IN} = 10\%$ Point to 90% Point
$T_6$	Output Valid Delay	2	19	ns	$C_L = 60\text{ pF}$ (LAD) $C_L = 50\text{ pF}$ (Controls)
$T_{6H}$	HOLDA Output Valid Delay	4	24	ns	$C_L = 50\text{ pF}$
$T_7$	ALE Width	12		ns	$C_L = 50\text{ pF}$
$T_8$	ALE Invalid Delay	0	20	ns	$C_L = 50\text{ pF}^{(2)}$
$T_9$	Output Float Delay	2	19	ns	$C_L = 60\text{ pF}$ (LAD) $C_L = 50\text{ pF}$ (Controls) <sup>(2)</sup>
$T_{9H}$	HOLDA Output Float Delay	4	20	ns	$C_L = 50\text{ pF}$
$T_{10}$	Input Setup 1	3		ns	(Note 1)
$T_{11}$	Input Hold	5		ns	(Note 1)
$T_{11H}$	HOLD Input Hold	4		ns	
$T_{12}$	Input Setup 2	7		ns	
$T_{13}$	Setup to ALE Inactive	8		ns	$C_L = 60\text{ pF}$ (LAD) $C_L = 50\text{ pF}$ (Controls)
$T_{14}$	Hold after ALE Inactive	8		ns	$C_L = 60\text{ pF}$ (LAD) $C_L = 50\text{ pF}$ (Controls)
$T_{15}$	Reset Hold	3		ns	
$T_{16}$	Reset Setup	5		ns	
$T_{17}$	Reset Width	820		ns	41 CLK2 Periods Minimum

**NOTES:**1. IAC/INT<sub>0</sub>, INT<sub>1</sub>, INT<sub>2</sub>/INTR, INT<sub>3</sub> can be asynchronous.2. A float condition occurs when the maximum output current becomes less than  $I_{LO}$ . Float delay is not tested, but should be no longer than the valid delay.

3. Case temperatures are "instant on".

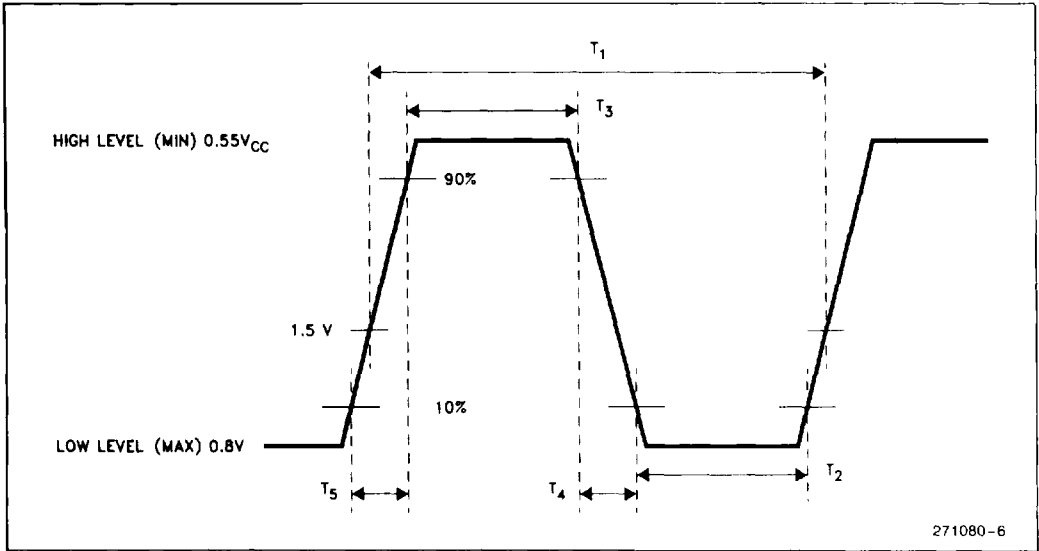


Figure 14. Processor Clock Pulse (CLK2)

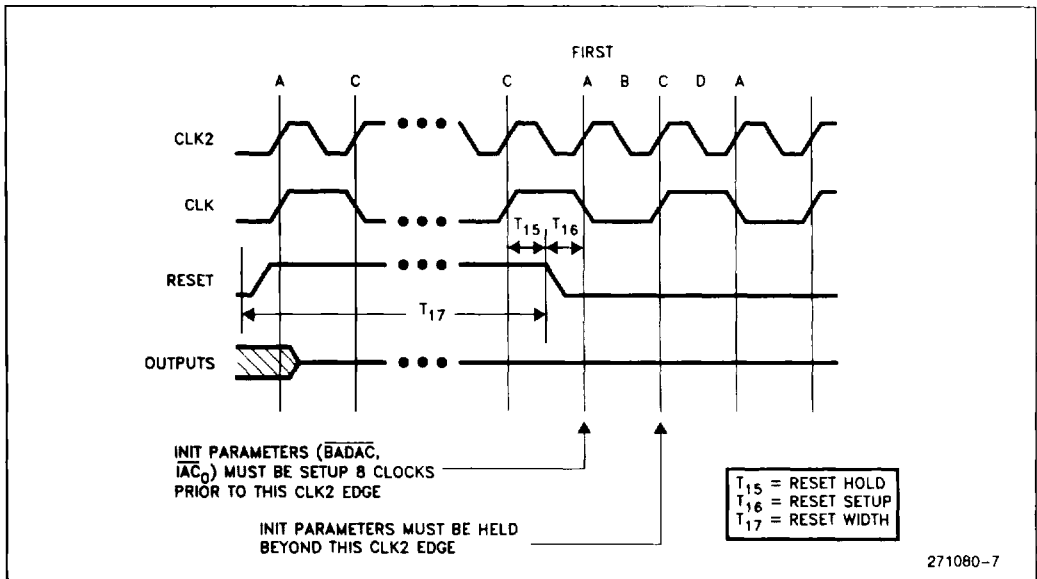


Figure 15. RESET Signal Timing



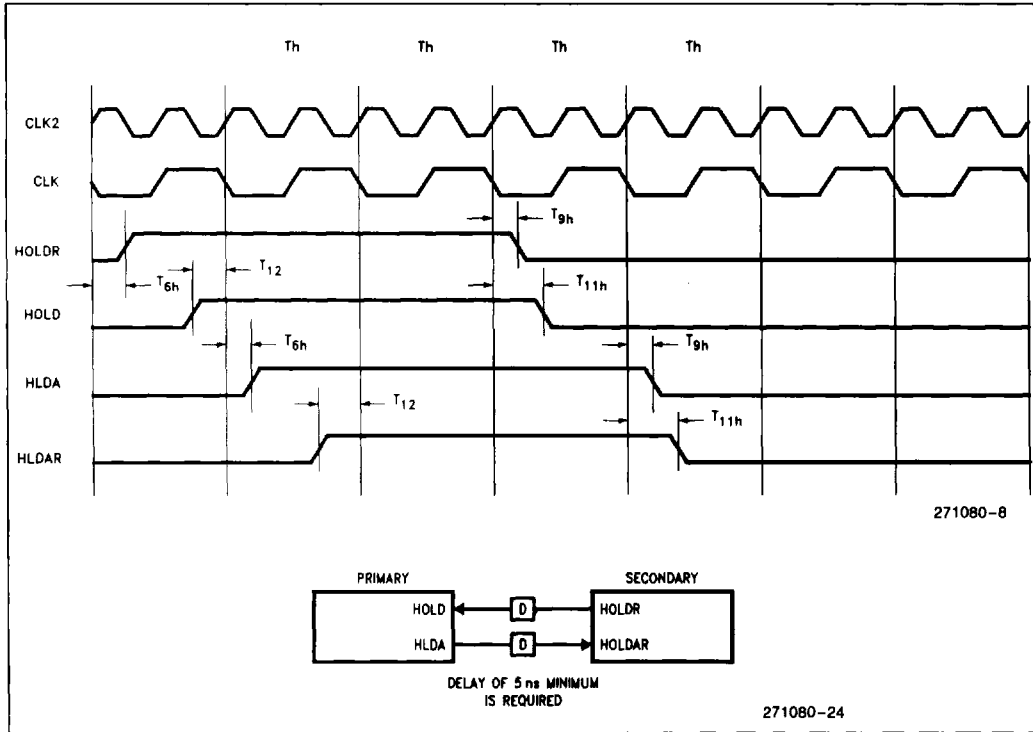


Figure 16. Hold Timing

**Design Considerations**

Input hold times can be disregarded by the designer whenever the input is removed because a subsequent output from the processor is deasserted (e.g.,  $\overline{DEN}$  becomes deasserted).

In other words, whenever the processor generates an output that indicates a transition into a subsequent state, the processor must have sampled any inputs for the previous state.

Similarly, whenever the processor generates an output that indicates a transition into a subsequent state, any outputs that are specified to be three stated in this new state are guaranteed to be three stated.

**Designing for the ICE-960MC**

The 80960MC In-Circuit Emulator assists in debugging 80960MC hardware and software designs. The product consists of a probe module, cable, and control unit. Because of the high operating frequency of 80960MC systems, the probe module connects directly to the 80960MC socket.

When designing an 80960MC hardware system that uses the ICE-960MC to debug the system, several electrical and mechanical characteristics should be considered. These considerations include capacitive loading, drive requirement, power requirement, and physical layout.

The ICE-960MC probe module increases the load capacitance of each line by up to 25 pF. It also adds one standard Schottky TTL load on the CLK2 line, up to one advanced low-power Schottky TTL load for each control signal line, and one advanced low-power Schottky TTL load for each address/data and byte enable line. These loads originate from the probe module and are driven by the 80960MC processor.

To achieve high noise immunity, the ICE-960MC probe is powered by the user's system. The high-speed probe circuitry draws up to 1.1A plus the maximum current ( $I_{CC}$ ) of the 80960MC processor.

The mechanical considerations are shown in Figure 17, which illustrates the lateral clearance requirements for the ICE-960MC probe as viewed from above the socket of the 80960MC processor.

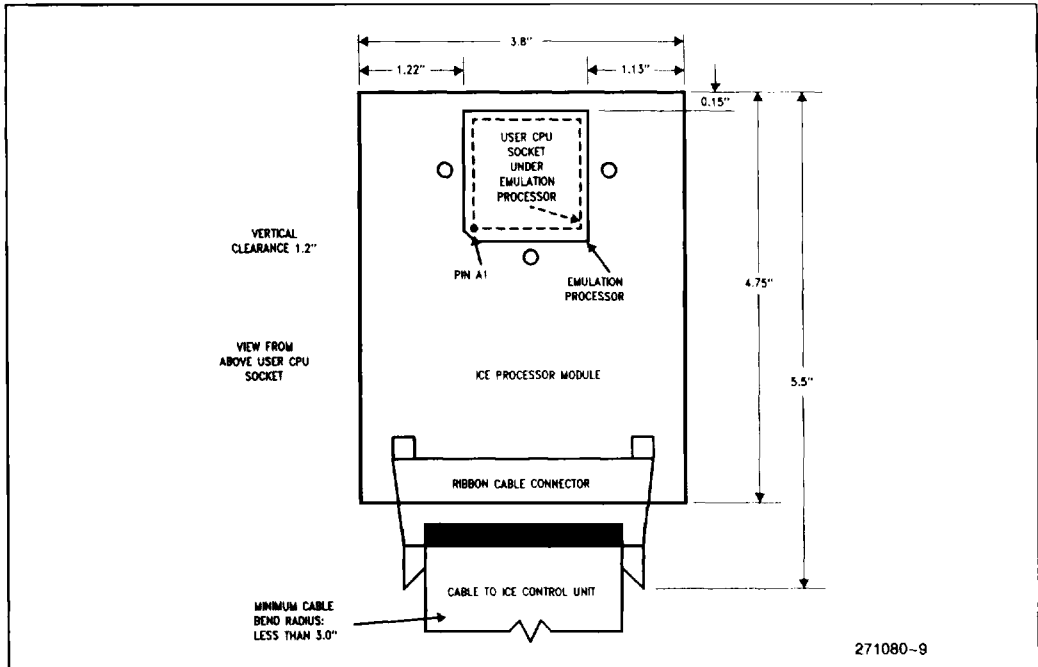


Figure 17. ICE-960MC Lateral Clearance Requirements

**MECHANICAL DATA**

**Pin Assignment**

The 80960MC is packaged in a 132-lead ceramic pin grid array and a 164-lead ceramic quad flatpack. The 80960MC pin grid array pinout as viewed from the substrate side of the component is shown in Figure 18 and from the pin side in Figure 19. The 80960MC ceramic quad flatpack pinout as viewed from the top of the package is shown in Figure 20.

V<sub>CC</sub> and GND connections must be made to multiple V<sub>CC</sub> and GND pins. Each V<sub>CC</sub> and GND pin must be connected to the appropriate voltage or ground and externally strapped close to the package. Preferably, the circuit board should include power and ground planes for power distribution. Tables 5, 6, 7 and 8 list the function of each pin.

**NOTE:**

Pins identified as N.C., "No Connect," should never be connected under any circumstances.

**Package Dimensions and Mounting**

Pins in the pin grid array package are arranged 0.100 inch (2.54mm) center-to-center, in a 14 by 14 matrix, three rows around. (See Figure 21.)

A wide variety of available sockets allow low-insertion or zero-insertion force mountings, and a choice of terminals such as soldertail, surface mount, or wire wrap. Several applicable sockets are shown in Figure 22.

**Package Thermal Specification**

The 80960MC is specified for operation when its case temperature is within the range of -55°C to +125°C. The PGA case temperature should be measured at the center of the top surface opposite the pins as shown in Figure 23. The ceramic quad flatpack case temperature should be measured at the center of the lid on the top surface of the package.

**WAVEFORMS**

Figures 24 through 30 show the waveforms for various transactions on the 80960MC's local bus.

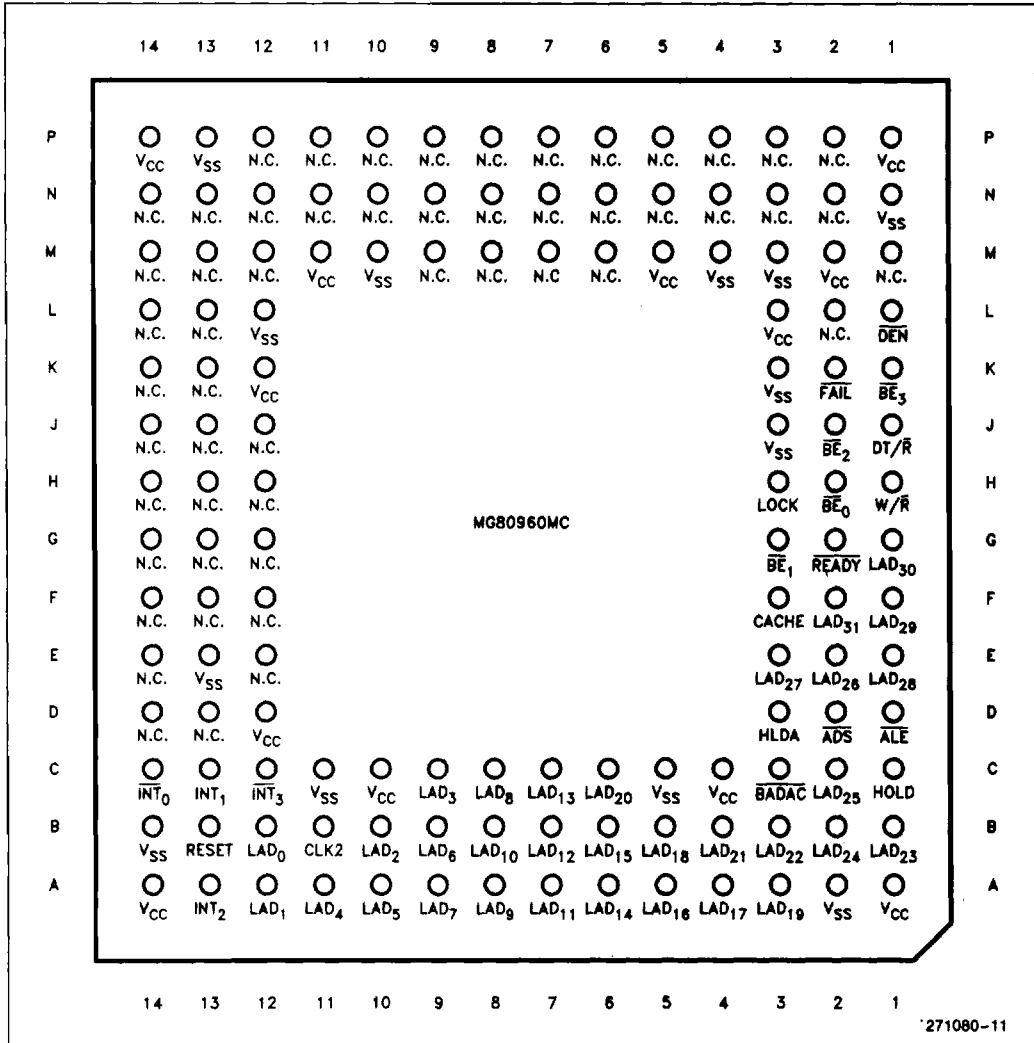


Figure 18. MG80960MC Pinout—View from Top (Pins Facing Down)

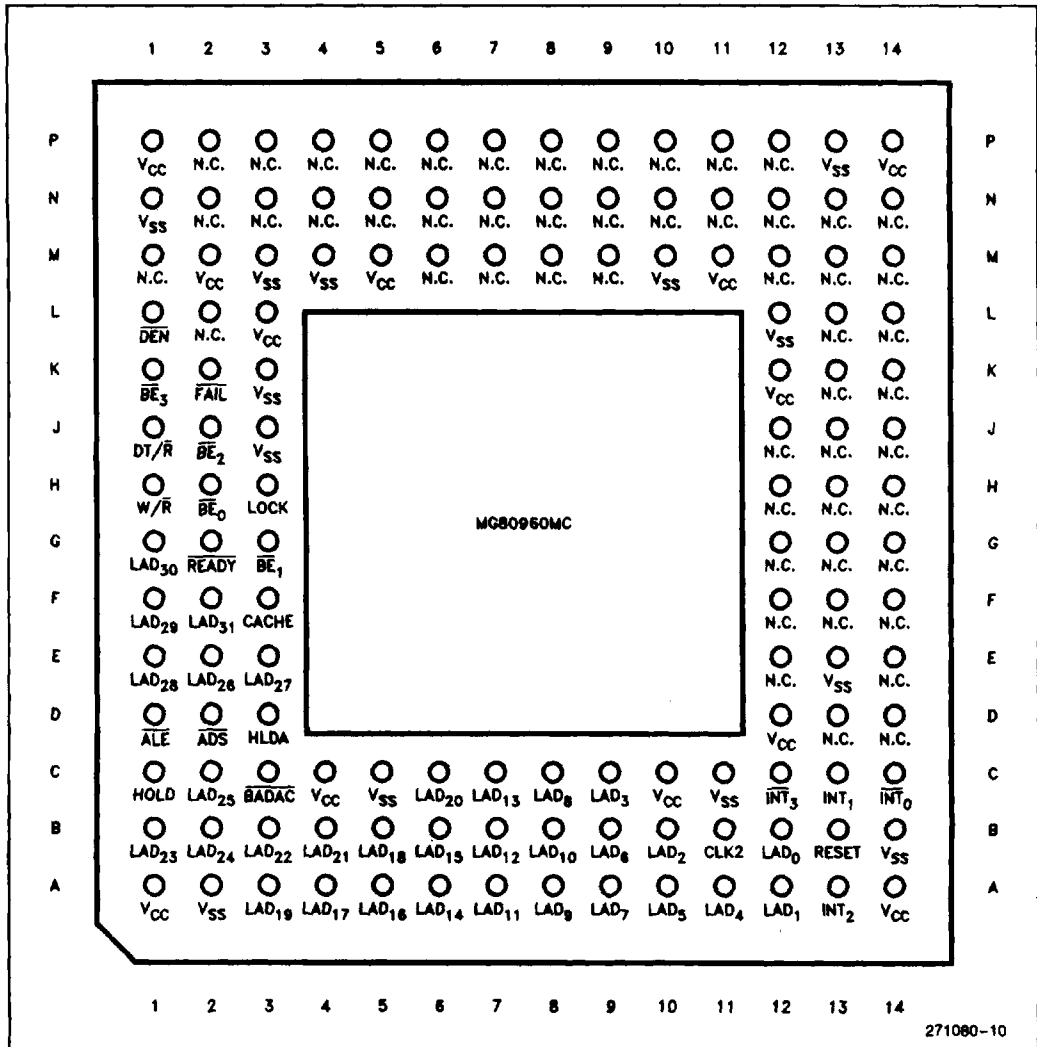
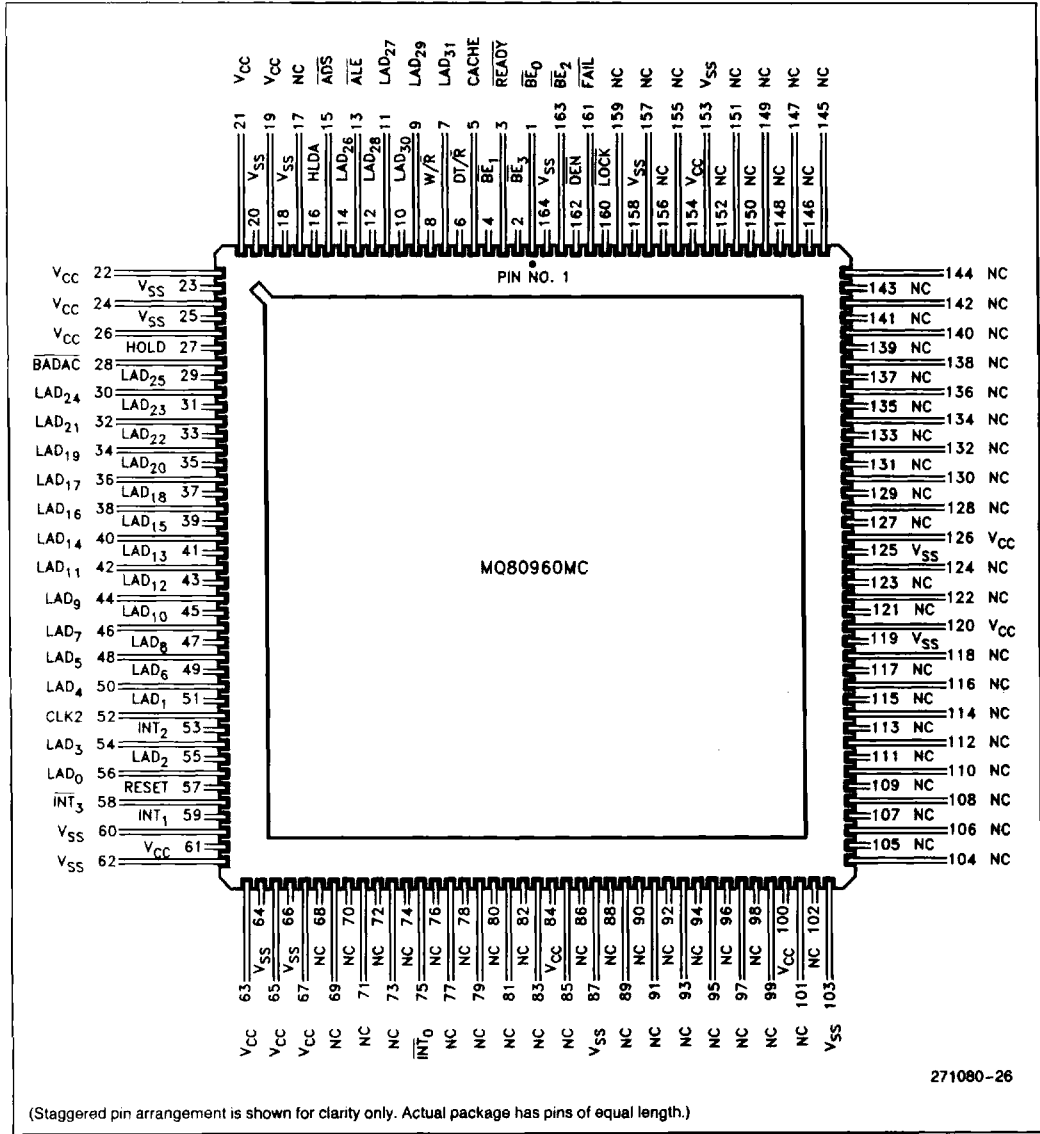


Figure 19. MG80960MC Pinout—View from Bottom (Pins Facing Up)

271080-10



(Staggered pin arrangement is shown for clarity only. Actual package has pins of equal length.)

Figure 20. MQ80960MC Pinout—View from Top of Package

5

Table 5. MG80960MC (PGA) Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	V <sub>CC</sub>	C6	LAD <sub>20</sub>	H1	W/R	M10	V <sub>SS</sub>
A2	V <sub>SS</sub>	C7	LAD <sub>13</sub>	H2	$\overline{BE}_0$	M11	V <sub>CC</sub>
A3	LAD <sub>19</sub>	C8	LAD <sub>8</sub>	H3	$\overline{LOCK}$	M12	N.C.
A4	LAD <sub>17</sub>	C9	LAD <sub>3</sub>	H12	N.C.	M13	N.C.
A5	LAD <sub>16</sub>	C10	V <sub>CC</sub>	H13	N.C.	M14	N.C.
A6	LAD <sub>14</sub>	C11	V <sub>SS</sub>	H14	N.C.	N1	V <sub>SS</sub>
A7	LAD <sub>11</sub>	C12	$\overline{INT}_3/\overline{INTA}$	J1	DT/R	N2	N.C.
A8	LAD <sub>9</sub>	C13	$\overline{INT}_1$	J2	$\overline{BE}_2$	N3	N.C.
A9	LAD <sub>7</sub>	C14	$\overline{IAC}/\overline{INT}_0$	J3	V <sub>SS</sub>	N4	N.C.
A10	LAD <sub>5</sub>	D1	$\overline{ALE}$	J12	N.C.	N5	N.C.
A11	LAD <sub>4</sub>	D2	$\overline{ADS}$	J13	N.C.	N6	N.C.
A12	LAD <sub>1</sub>	D3	HLDA/HLDR	J14	N.C.	N7	N.C.
A13	$\overline{INT}_2/\overline{INTR}$	D12	V <sub>CC</sub>	K1	$\overline{BE}_3$	N8	N.C.
A14	V <sub>CC</sub>	D13	N.C.	K2	FAILURE	N9	N.C.
B1	LAD <sub>23</sub>	D14	N.C.	K3	V <sub>SS</sub>	N10	N.C.
B2	LAD <sub>24</sub>	E1	LAD <sub>28</sub>	K12	V <sub>CC</sub>	N11	N.C.
B3	LAD <sub>22</sub>	E2	LAD <sub>26</sub>	K13	N.C.	N12	N.C.
B4	LAD <sub>21</sub>	E3	LAD <sub>27</sub>	K14	N.C.	N13	N.C.
B5	LAD <sub>18</sub>	E12	N.C.	L1	$\overline{DEN}$	N14	N.C.
B6	LAD <sub>15</sub>	E13	V <sub>SS</sub>	L2	N.C.	P1	V <sub>CC</sub>
B7	LAD <sub>12</sub>	E14	N.C.	L3	V <sub>CC</sub>	P2	N.C.
B8	LAD <sub>10</sub>	F1	LAD <sub>29</sub>	L12	V <sub>SS</sub>	P3	N.C.
B9	LAD <sub>6</sub>	F2	LAD <sub>31</sub>	L13	N.C.	P4	N.C.
B10	LAD <sub>2</sub>	F3	CACHE	L14	N.C.	P5	N.C.
B11	CLK2	F12	N.C.	M1	N.C.	P6	N.C.
B12	LAD <sub>0</sub>	F13	N.C.	M2	V <sub>CC</sub>	P7	N.C.
B13	RESET	F14	N.C.	M3	V <sub>SS</sub>	P8	N.C.
B14	V <sub>SS</sub>	G1	LAD <sub>30</sub>	M4	V <sub>SS</sub>	P9	N.C.
C1	HOLD/HLDR	G2	READY	M5	V <sub>CC</sub>	P10	N.C.
C2	LAD <sub>25</sub>	G3	$\overline{BE}_1$	M6	N.C.	P11	N.C.
C3	$\overline{BADAC}$	G12	N.C.	M7	N.C.	P12	N.C.
C4	V <sub>CC</sub>	G13	N.C.	M8	N.C.	P13	V <sub>SS</sub>
C5	V <sub>SS</sub>	G14	N.C.	M9	N.C.	P14	V <sub>CC</sub>

**NOTE:**

Pins identified as N.C. ("No Connect") should never be connected under any circumstances.

Table 6. MG80960MC (PGA) Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
$\overline{ADS}$	D2	LAD <sub>15</sub>	B6	N.C.	J14	N.C.	P89
$\overline{ALE}$	D1	LAD <sub>16</sub>	A5	N.C.	K13	N.C.	P10
BADAC	C3	LAD <sub>17</sub>	A4	N.C.	K14	N.C.	P11
$\overline{BE}_0$	H2	LAD <sub>18</sub>	B5	N.C.	L13	N.C.	P12
$\overline{BE}_1$	G3	LAD <sub>19</sub>	A3	N.C.	L14	N.C.	L2
$\overline{BE}_2$	J2	LAD <sub>20</sub>	C6	N.C.	M1	N.C.	G2
$\overline{BE}_3$	K1	LAD <sub>21</sub>	B4	N.C.	M6	READY	B13
CACHE	F3	LAD <sub>22</sub>	B3	N.C.	M7	RESET	A1
CLK2	B11	LAD <sub>23</sub>	B1	N.C.	M8	V <sub>CC</sub>	A14
$\overline{DEN}$	L1	LAD <sub>24</sub>	B2	N.C.	M9	V <sub>CC</sub>	C4
DT/ $\overline{R}$	J1	LAD <sub>25</sub>	C2	N.C.	M12	V <sub>CC</sub>	C10
FAILURE	K2	LAD <sub>26</sub>	E2	N.C.	M13	V <sub>CC</sub>	D12
HLDA/HOLDR	D3	LAD <sub>27</sub>	E3	N.C.	M14	V <sub>CC</sub>	K12
HOLD/HLDAR	C1	LAD <sub>28</sub>	E1	N.C.	N2	V <sub>CC</sub>	L3
$\overline{IAC}/\overline{INT}_0$	C14	LAD <sub>29</sub>	F1	N.C.	N3	V <sub>CC</sub>	M2
INT <sub>1</sub>	C13	LAD <sub>30</sub>	G1	N.C.	N4	V <sub>CC</sub>	M5
INT <sub>2</sub> /INTR	A13	LAD <sub>31</sub>	F2	N.C.	N5	V <sub>CC</sub>	M11
$\overline{INT}_3/\overline{INTA}$	C12	$\overline{LOCK}$	H3	N.C.	N6	V <sub>CC</sub>	P1
LAD <sub>0</sub>	B12	N.C.	D13	N.C.	N7	V <sub>CC</sub>	P14
LAD <sub>1</sub>	A12	N.C.	D14	N.C.	N8	V <sub>SS</sub>	A2
LAD <sub>2</sub>	B10	N.C.	E12	N.C.	N9	V <sub>SS</sub>	B14
LAD <sub>3</sub>	C9	N.C.	E14	N.C.	N10	V <sub>SS</sub>	C5
LAD <sub>4</sub>	A11	N.C.	F12	N.C.	N11	V <sub>SS</sub>	C11
LAD <sub>5</sub>	A10	N.C.	F13	N.C.	N12	V <sub>SS</sub>	E13
LAD <sub>6</sub>	B9	N.C.	F14	N.C.	N13	V <sub>SS</sub>	J3
LAD <sub>7</sub>	A9	N.C.	G12	N.C.	N14	V <sub>SS</sub>	K3
LAD <sub>8</sub>	C8	N.C.	G13	N.C.	P2	V <sub>SS</sub>	L12
LAD <sub>9</sub>	A8	N.C.	G14	N.C.	P3	V <sub>SS</sub>	M3
LAD <sub>10</sub>	B8	N.C.	H12	N.C.	P4	V <sub>SS</sub>	M4
LAD <sub>11</sub>	A7	N.C.	H13	N.C.	P5	V <sub>SS</sub>	M10
LAD <sub>12</sub>	B7	N.C.	H14	N.C.	P6	V <sub>SS</sub>	N1
LAD <sub>13</sub>	C7	N.C.	J12	N.C.	P7	V <sub>SS</sub>	P13
LAD <sub>14</sub>	A6	N.C.	J13	N.C.	P8	W/ $\overline{R}$	H1

**NOTE:**

Pins identified as N.C. ("No Connect") should never be connected under any circumstances.

Table 7. MQ80960MC (CQP) Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	$\overline{BE}_0$	42	LAD <sub>11</sub>	83	N.C.	124	N.C.
2	$\overline{BE}_3$	43	LAD <sub>12</sub>	84	V <sub>CC</sub>	125	V <sub>SS</sub>
3	READY	44	LAD <sub>9</sub>	85	N.C.	126	V <sub>CC</sub>
4	$\overline{BE}_1$	45	LAD <sub>10</sub>	86	N.C.	127	N.C.
5	CACHE	46	LAD <sub>7</sub>	87	V <sub>SS</sub>	128	N.C.
6	DT/R	47	LAD <sub>8</sub>	88	N.C.	129	N.C.
7	LAD <sub>31</sub>	48	LAD <sub>5</sub>	89	N.C.	130	N.C.
8	W/R	49	LAD <sub>6</sub>	90	N.C.	131	N.C.
9	LAD <sub>29</sub>	50	LAD <sub>4</sub>	91	N.C.	132	N.C.
10	LAD <sub>30</sub>	51	LAD <sub>1</sub>	92	N.C.	133	N.C.
11	LAD <sub>27</sub>	52	CLK <sub>2</sub>	93	N.C.	134	N.C.
12	LAD <sub>28</sub>	53	INT <sub>2</sub>	94	N.C.	135	N.C.
13	ALE	54	LAD <sub>3</sub>	95	N.C.	136	N.C.
14	LAD <sub>26</sub>	55	LAD <sub>2</sub>	96	N.C.	137	N.C.
15	ADS	56	LAD <sub>0</sub>	97	N.C.	138	N.C.
16	HLDA	57	RESET	98	N.C.	139	N.C.
17	N.C.	58	$\overline{INT}_3$	99	N.C.	140	N.C.
18	V <sub>SS</sub>	59	INT <sub>1</sub>	100	V <sub>CC</sub>	141	N.C.
19	V <sub>CC</sub>	60	V <sub>SS</sub>	101	N.C.	142	N.C.
20	V <sub>SS</sub>	61	V <sub>CC</sub>	102	N.C.	143	N.C.
21	V <sub>CC</sub>	62	V <sub>SS</sub>	103	V <sub>SS</sub>	144	N.C.
22	V <sub>CC</sub>	63	V <sub>CC</sub>	104	N.C.	145	N.C.
23	V <sub>SS</sub>	64	V <sub>SS</sub>	105	N.C.	146	N.C.
24	V <sub>CC</sub>	65	V <sub>CC</sub>	106	N.C.	147	N.C.
25	V <sub>SS</sub>	66	V <sub>SS</sub>	107	N.C.	148	N.C.
26	V <sub>CC</sub>	67	V <sub>CC</sub>	108	N.C.	149	N.C.
27	HOLD	68	N.C.	109	N.C.	150	N.C.
28	BADAC	69	N.C.	110	N.C.	151	N.C.
29	LAD <sub>25</sub>	70	N.C.	111	N.C.	152	N.C.
30	LAD <sub>24</sub>	71	N.C.	112	N.C.	153	V <sub>SS</sub>
31	LAD <sub>23</sub>	72	N.C.	113	N.C.	154	V <sub>CC</sub>
32	LAD <sub>21</sub>	73	N.C.	114	N.C.	155	N.C.
33	LAD <sub>22</sub>	74	N.C.	115	N.C.	156	N.C.
34	LAD <sub>19</sub>	75	$\overline{INT}_0$	116	N.C.	157	N.C.
35	LAD <sub>20</sub>	76	N.C.	117	N.C.	158	V <sub>SS</sub>
36	LAD <sub>17</sub>	77	N.C.	118	N.C.	159	N.C.
37	LAD <sub>18</sub>	78	N.C.	119	V <sub>SS</sub>	160	LOCK
38	LAD <sub>16</sub>	79	N.C.	120	V <sub>CC</sub>	161	FAIL
39	LAD <sub>15</sub>	80	N.C.	121	N.C.	162	DEN
40	LAD <sub>14</sub>	81	N.C.	122	N.C.	163	$\overline{BE}_2$
41	LAD <sub>13</sub>	82	N.C.	123	N.C.	164	V <sub>SS</sub>

**NOTE:**

Pins identified as N.C. ("No Connect") should never be connected under any circumstances.



Table 8. MQ80960MC (CQP) Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
ADS	15	LAD <sub>23</sub>	31	N.C.	102	N.C.	148
$\overline{\text{ALE}}$	13	LAD <sub>24</sub>	30	N.C.	104	N.C.	149
BADAC	28	LAD <sub>25</sub>	29	N.C.	105	N.C.	150
$\overline{\text{BE}}_0$	1	LAD <sub>26</sub>	14	N.C.	106	N.C.	151
$\overline{\text{BE}}_1$	4	LAD <sub>27</sub>	11	N.C.	107	N.C.	152
$\overline{\text{BE}}_2$	163	LAD <sub>28</sub>	12	N.C.	108	N.C.	155
$\overline{\text{BE}}_3$	2	LAD <sub>29</sub>	9	N.C.	109	N.C.	156
CACHE	5	LAD <sub>30</sub>	10	N.C.	110	N.C.	157
CLK2	52	LAD <sub>31</sub>	7	N.C.	111	N.C.	159
$\overline{\text{DEN}}$	162	$\overline{\text{LOCK}}$	160	N.C.	112	READY	3
$\overline{\text{DT/R}}$	6	N.C.	17	N.C.	113	RESET	57
FAILURE	161	N.C.	68	N.C.	114	V <sub>CC</sub>	19
HLDA/HOLDR	16	N.C.	69	N.C.	115	V <sub>CC</sub>	21
HOLD/HLDAR	27	N.C.	70	N.C.	116	V <sub>CC</sub>	22
$\overline{\text{IAC/INT}}_0$	75	N.C.	71	N.C.	117	V <sub>CC</sub>	24
INT <sub>1</sub>	59	N.C.	72	N.C.	118	V <sub>CC</sub>	26
INT <sub>2</sub> /INTR	53	N.C.	73	N.C.	121	V <sub>CC</sub>	61
INT <sub>3</sub> /INTA	58	N.C.	74	N.C.	122	V <sub>CC</sub>	63
LAD <sub>0</sub>	56	N.C.	76	N.C.	123	V <sub>CC</sub>	65
LAD <sub>1</sub>	51	N.C.	77	N.C.	124	V <sub>CC</sub>	67
LAD <sub>2</sub>	55	N.C.	78	N.C.	127	V <sub>CC</sub>	84
LAD <sub>3</sub>	54	N.C.	79	N.C.	128	V <sub>CC</sub>	100
LAD <sub>4</sub>	50	N.C.	80	N.C.	129	V <sub>CC</sub>	120
LAD <sub>5</sub>	48	N.C.	81	N.C.	130	V <sub>CC</sub>	126
LAD <sub>6</sub>	49	N.C.	82	N.C.	131	V <sub>CC</sub>	154
LAD <sub>7</sub>	46	N.C.	83	N.C.	132	V <sub>SS</sub>	18
LAD <sub>8</sub>	47	N.C.	85	N.C.	133	V <sub>SS</sub>	20
LAD <sub>9</sub>	44	N.C.	86	N.C.	134	V <sub>SS</sub>	23
LAD <sub>10</sub>	45	N.C.	88	N.C.	135	V <sub>SS</sub>	25
LAD <sub>11</sub>	42	N.C.	89	N.C.	136	V <sub>SS</sub>	60
LAD <sub>12</sub>	43	N.C.	90	N.C.	137	V <sub>SS</sub>	62
LAD <sub>13</sub>	41	N.C.	91	N.C.	138	V <sub>SS</sub>	64
LAD <sub>14</sub>	40	N.C.	92	N.C.	139	V <sub>SS</sub>	66
LAD <sub>15</sub>	39	N.C.	93	N.C.	140	V <sub>SS</sub>	87
LAD <sub>16</sub>	38	N.C.	94	N.C.	141	V <sub>SS</sub>	103
LAD <sub>17</sub>	36	N.C.	95	N.C.	142	V <sub>SS</sub>	119
LAD <sub>18</sub>	37	N.C.	96	N.C.	143	V <sub>SS</sub>	125
LAD <sub>19</sub>	34	N.C.	97	N.C.	144	V <sub>SS</sub>	153
LAD <sub>20</sub>	35	N.C.	98	N.C.	145	V <sub>SS</sub>	158
LAD <sub>21</sub>	32	N.C.	99	N.C.	146	V <sub>SS</sub>	164
LAD <sub>22</sub>	33	N.C.	101	N.C.	147	W/R	8

**NOTE:**

Pins identified as N.C. ("No Connect") should never be connected under any circumstances.

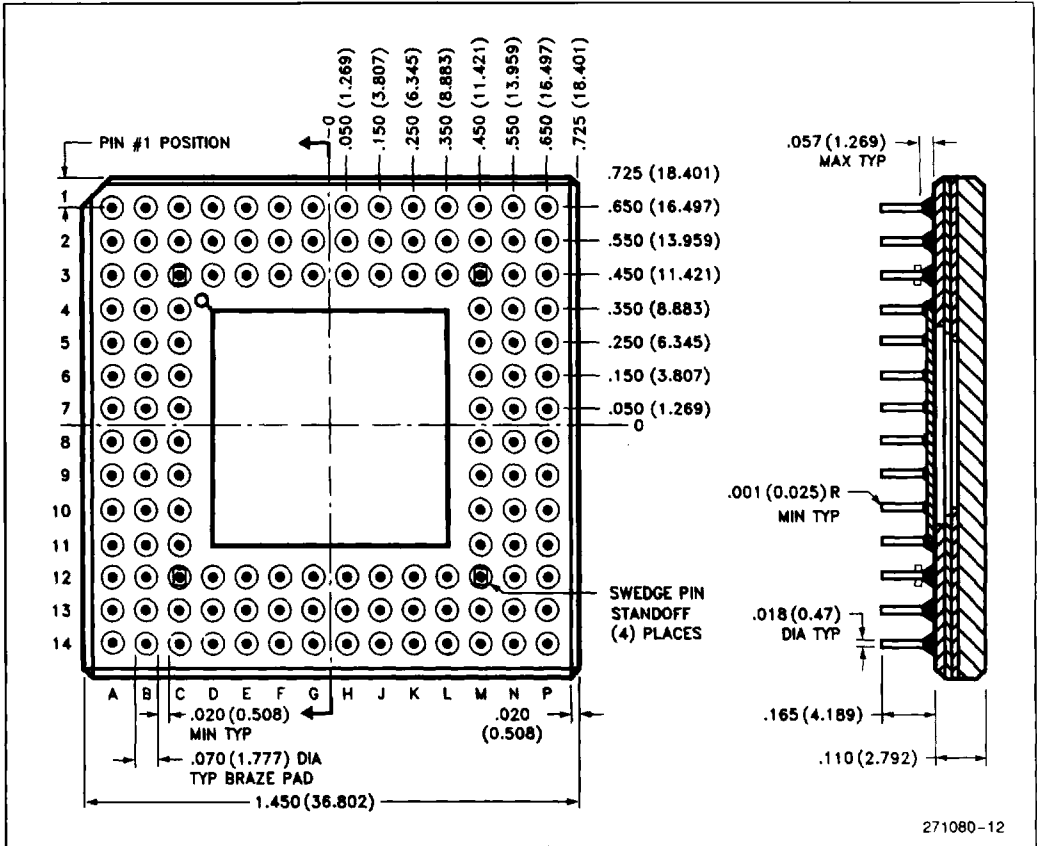
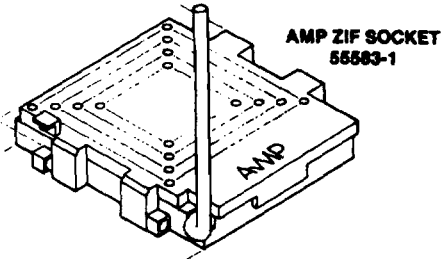
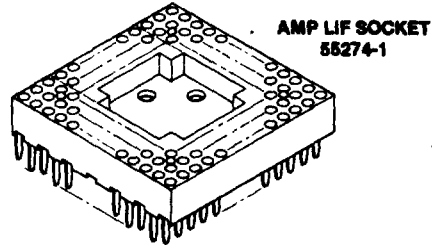


Figure 21. A 132-Lead Pin-Grid Array (PGA) Used to Package the MG80960MC

- Low insertion force (LIF) soldertail 55274-1
- Amp tests indicate 50% reduction in insertion force compared to machined sockets
- Other socket options
- Zero insertion force (ZIF) soldertail 55583-1
- Zero insertion force (ZIF) Burn-in version 55573-2

**Amp Incorporated**  
 (Harrisburg, PA 17105 U.S.A  
 Phone 717-564-0100)



271080-13

Cam handle locks in low profile position when MG80960MC is installed (handle UP for open and DOWN for closed positions).

Courtesy Amp Incorporated

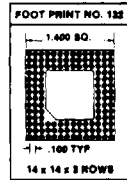
Peel-A-Way\* Mylar and Kapton Socket Terminal Carriers

- Low insertion force surface mount CS132-37TG
- Low insertion force soldertail CS132-01TG
- Low insertion force wire-wrap CS132-02TG (two-level) CS132-03TG (three-level)
- Low insertion force press-fit CS132-05TG

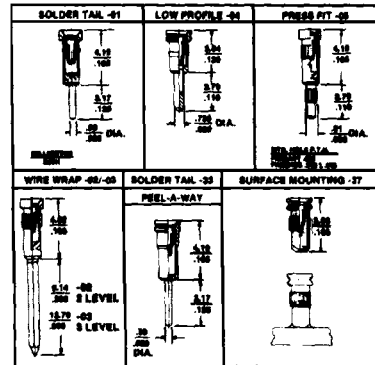
**Advanced Interconnections**  
 (5 Division Street)  
 Warwick, RI 02818 U.S.A.  
 Phone 401-885-0485)

Peel-A-Way Carrier No. 132: Kapton Carrier is KS132 Mylar Carrier is MS132

Molded Plastic Body KS132 is shown below:



271080-14



271080-15

Courtesy Advanced Interconnections  
 (Peel-A-Way Terminal Carriers  
 U.S. Patent No. 4442838)

\*Peel-A-Way is a trademark of Advanced Interconnections

Figure 22. Several Socket Options for Mounting the MG80960MC

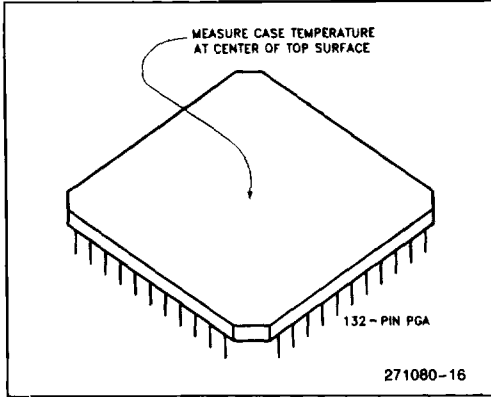


Figure 23. Measuring MG80960MC PGA Case Temperature ( $T_C$ )

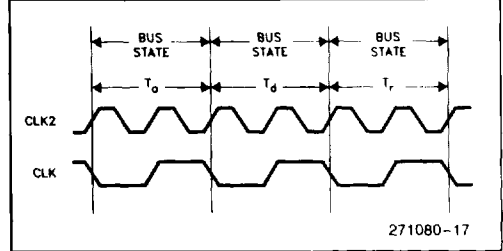


Figure 24. System and Processor Clock Relationship

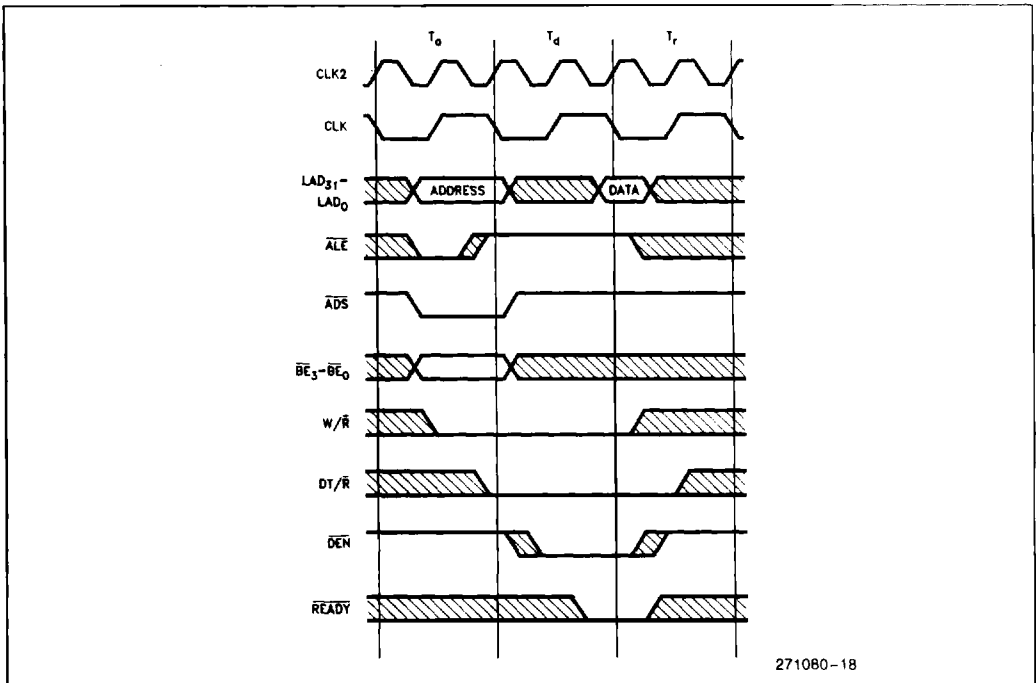


Figure 25. Read Transaction

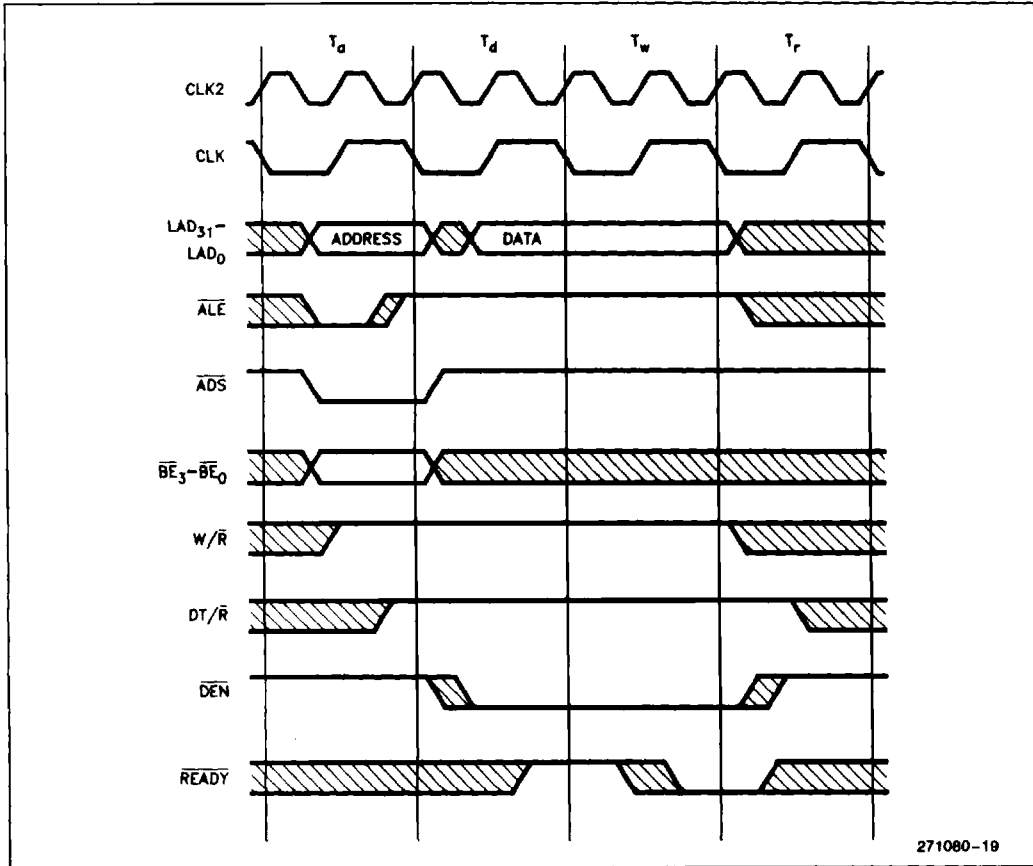
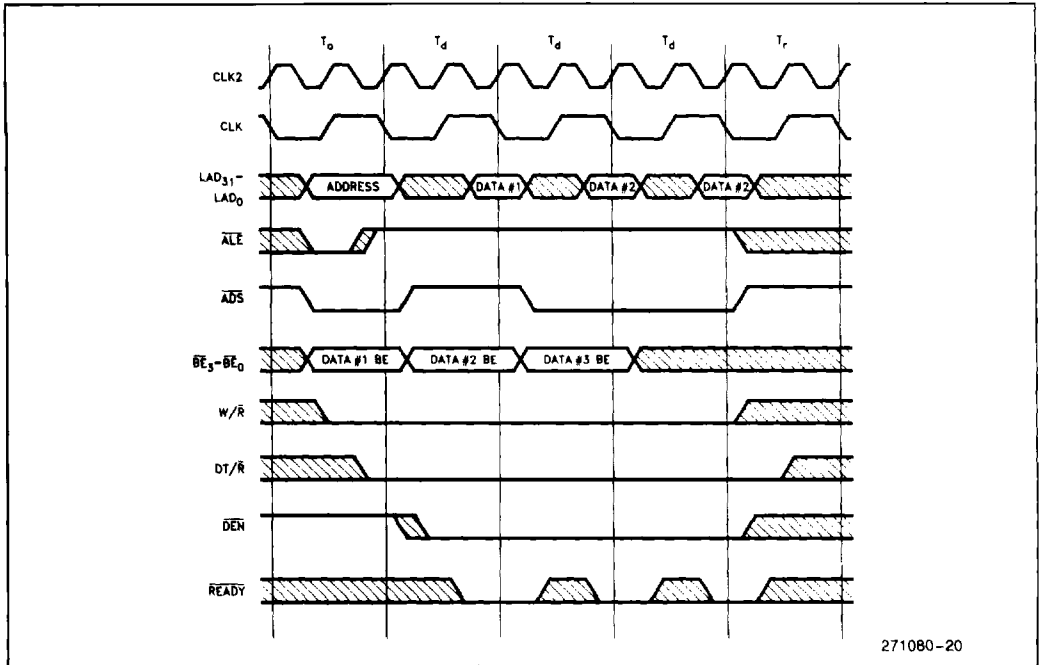
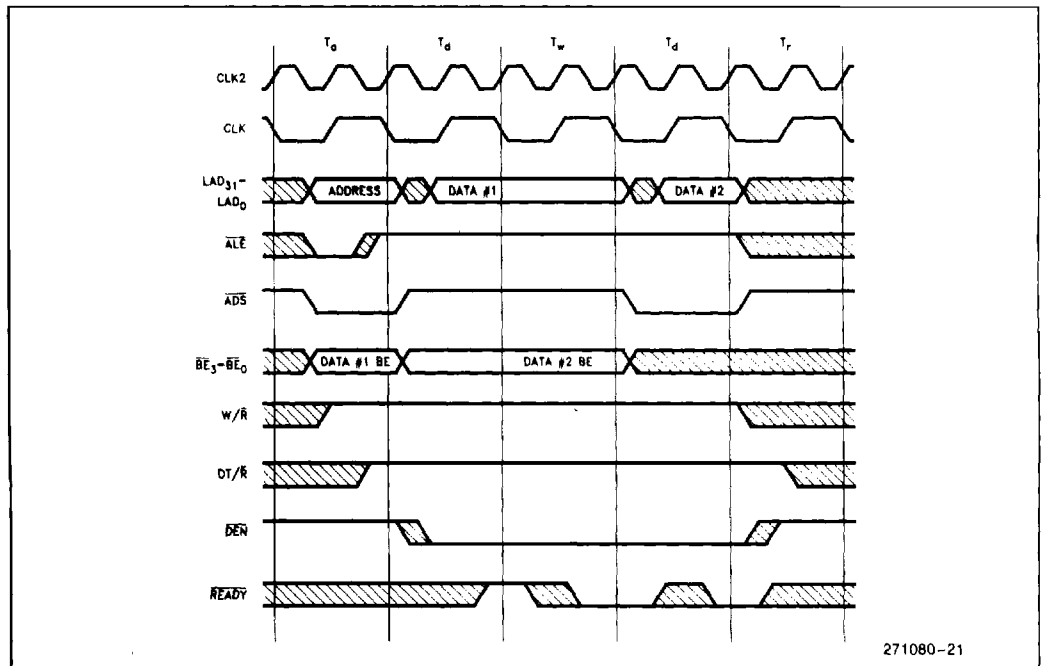


Figure 26. Write Transaction with One Wait State



271080-20

Figure 27. Burst Read Transaction



271080-21

Figure 28. Burst Write Transaction with One Wait State

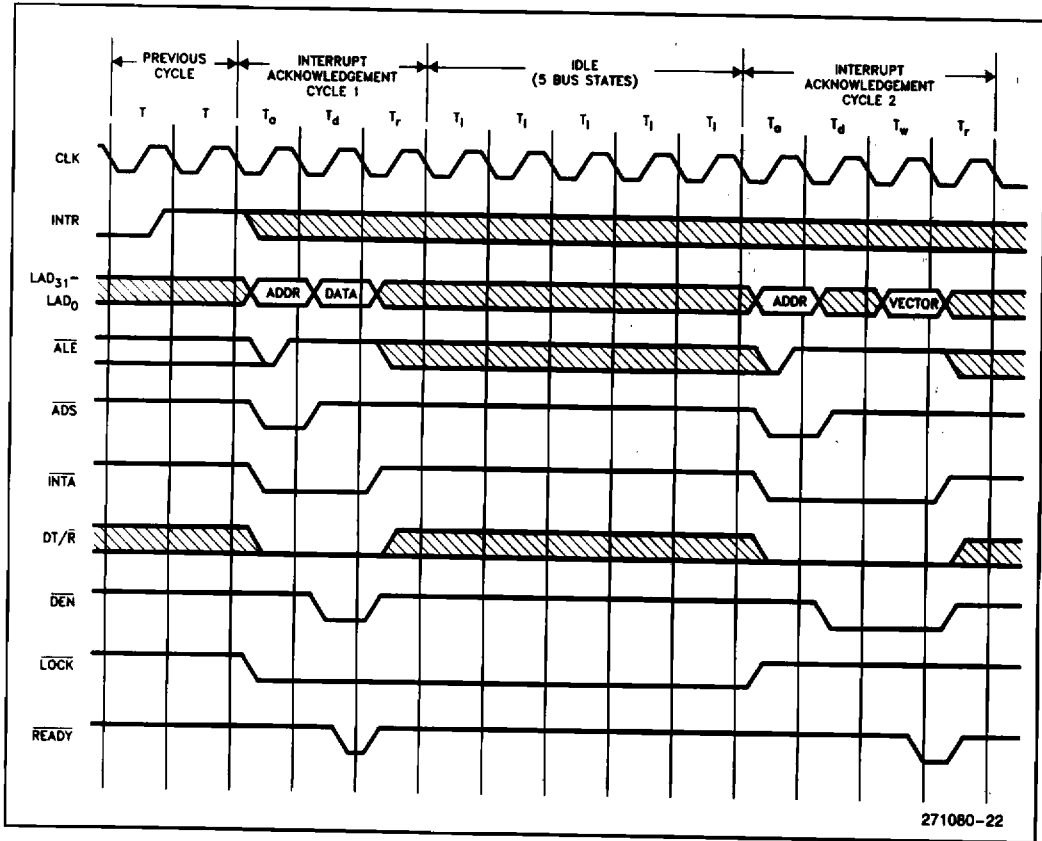


Figure 29. Interrupt Acknowledge Transaction

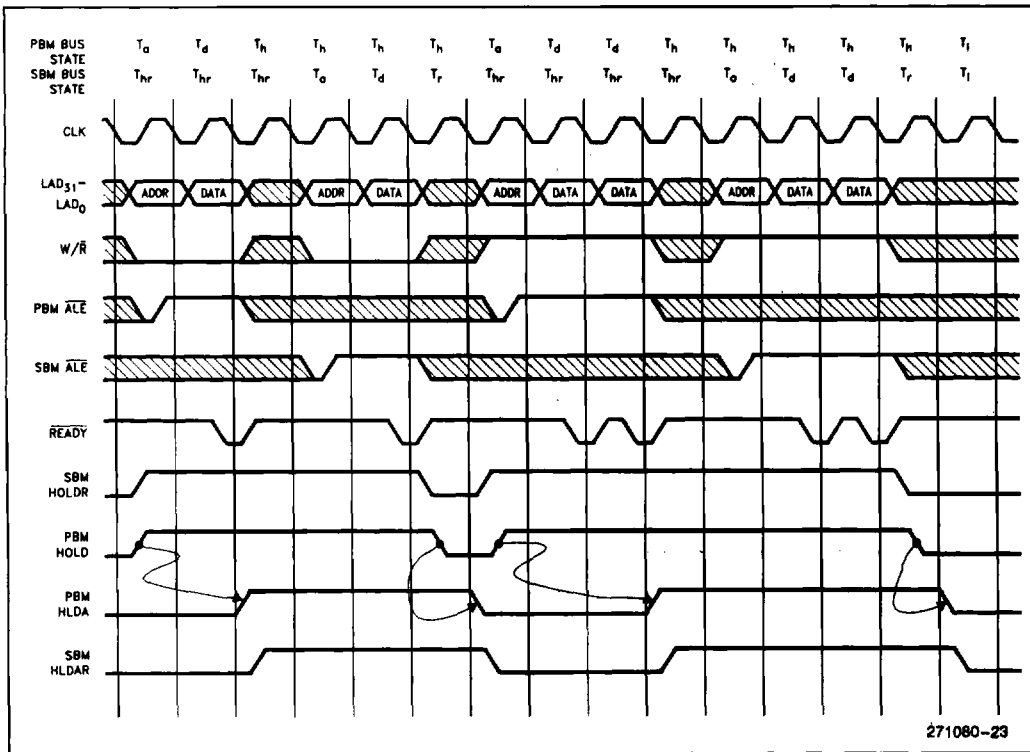


Figure 30. Bus Exchange Transaction (PBM = Primary Bus Master, SBM = Secondary Bus Master)

**Revision History**

1. 20 MHz timing specifications were added.
2. Pin 158, ceramic quad pack, (see Figure 20) changed from NC (No Connect) to  $V_{SS}$ .