MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

Order this document
by MC68LC040/D

MC68LC040

Preliminary Technical Summary

Third-Generation 32-Bit Microprocessor

The MC68LC040 is Motorola's interger only version of the MC68040 third-generation, M68000-compatible, high-performance, 32-bit microprocessors. The MC68LC040 is a virtual memory microprocessor with a highly integrated architecture which provides a very high performance in a monolithic HCMOS device. On a single chip, the MC68LC040 integrates an MC68040-compatible integer unit and fully independent instruction and data demand-paged memory management units (MMUs), including independent 4K-byte instruction and data caches. A high degree of instruction execution parallelism is achieved through the use of a multistage instruction pipeline, multiple internal buses, and a full internal Harvard architecture, including separate physical caches for both instruction and data accesses. The MC68LC040 also directly supports cache coherency in multimaster applications with dedicated on-chip bus snooping logic. For detailed information on the MC68LC040, refer to MC68040UM/AD, *MC68040 32-Bit Microprocessor User's Manual.*

The MC68LC040 achieves the high performance through the use of the MC68040 integer unit. The multi-stage pipeline operates on up to six instructions concurrent with MMU, cache, and bus controller operations. Multiple internal buses, separate data and instruction caches, and a sophisticated bus controller allow internal units to operate concurrently and decouple the MC68LC040 from the external bus. The internal caches and the decoupling of the external bus allow for an external memory subsystem to be built from slower and less expensive memories with minimal impact to the overall system performance. The potential for a low-cost system design with the price/performance of the MC68LC040 makes it a good choice for embedded microprocessor applications as well as central processor applications.

The MC68LC040 is user-object-code compatible with previous members of the M68000 family and is specifically optimized to reduce the execution time of compiler-generated code. The high level of performance is ideal for integer-intensive applications. The MC68LC040 is implemented in Motorola's latest HCMOS technology, providing an ideal balance between speed, power, and physical device size. Independent data and instruction MMUs control the main caches and the address translation caches (ATCs). The ATCs speed up logical-to-physical address translations by storing recently used translations. The bus snooper circuit ensures cache coherency in multimaster and multiprocessing applications.

The main features of the MC68LC040 include:

- 22 MIPS Integer Performance at 25 MHz
- Independent Instruction and Data MMUs
- 4K-Byte Physical Instruction Cache and 4K-Byte Physical Data Cache Accessed Simultaneously
- 32-Bit, Nonmultiplexed External Address and Data Buses with Synchronous Interface
- User-Object-Code Compatibility with All Earlier M68000 Microprocessors
- Multimaster/Multiprocessor Support Via Bus Snooping
- Concurrent Integer Unit, MMU, Bus Controller, and Bus Snooper Maximize Throughput
- 4G-byte Direct Addressing Range
- Software Support Including Optimizing C Compiler and UNIX® System V Port

UNIX® is a registered trademark of AT&T Bell Laboratories.

INTRODUCTION

The MC68LC040 is an enhanced, 32-bit, HCMOS microprocessor that combines the high-performance integer unit processing capabilities of the MC68040 microprocessor with independent 4K-byte data and instruction caches. The MC68LC040 maintains the 32-bit registers available with the entire M68000 family as well as the 32-bit address and data paths, rich instruction set, and versatile addressing modes. Instruction execution proceeds in parallel with accesses to the internal caches, MMU operations, and bus controller activity. Additionally, the integer unit is optimized for high-level language environments. Figure 1 illustrates a simplified block diagram of the MC68LC040.

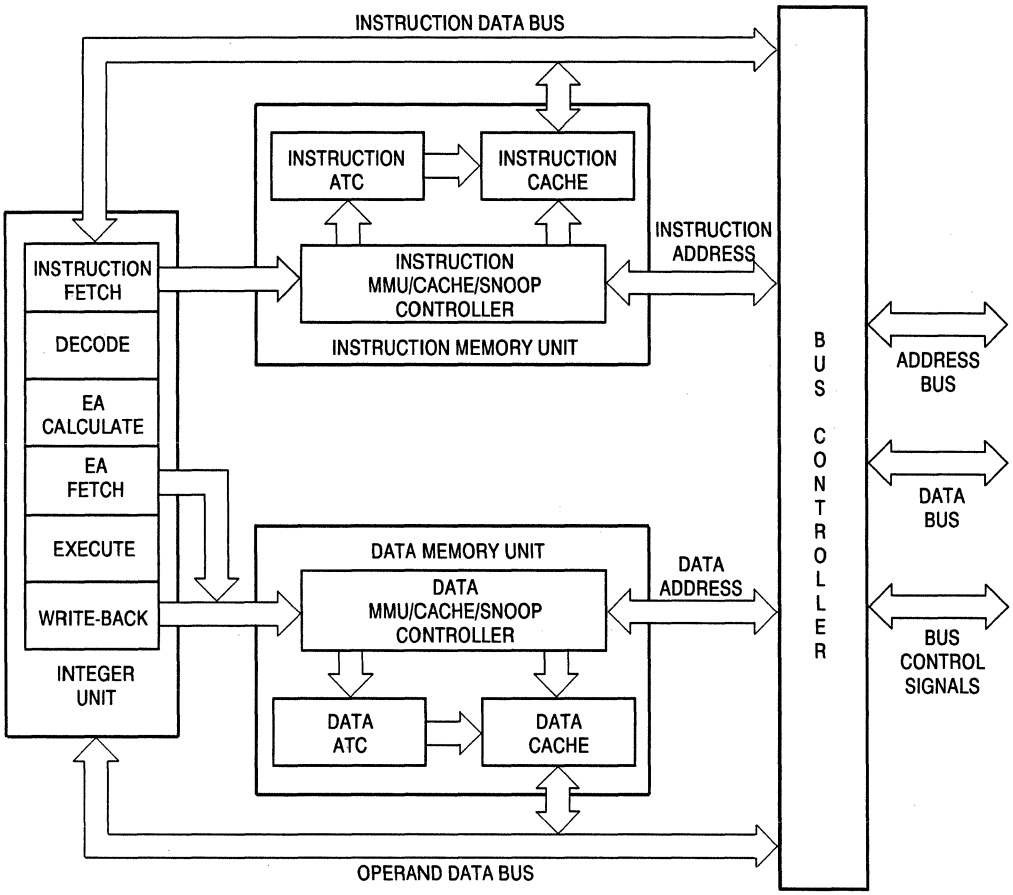


Figure 1. MC68LC040 Block Diagram

The MMUs support multiprocessing, virtual memory systems by translating logical addresses to physical addresses using translation tables stored in memory. The MMUs store recently used address mappings in two separate ATCs on-chip. When an ATC contains the physical address for a bus cycle requested by the processor, a translation table search is avoided and the physical address is supplied immediately, incurring no delay for address translation. Each MMU has two transparent translation registers that define a one-to-one mapping for address space segments ranging in size from 16M-bytes to 4G-bytes each.

Each MMU provides read-only and supervisor-only protections on a page basis. Also, processes can be given isolated address spaces by assigning each a unique table structure and updating the root pointer upon a task swap. Isolated address spaces protect the integrity of independent processes.

The instruction and data caches operate independently from the rest of the machine, storing information for fast access by the execution units. Each cache resides on its own internal address bus and internal data bus, allowing simultaneous access to both. The data cache provides writethrough or copyback write modes that can be configured on a page-by-page basis.

The MC68LC040 bus controller supports a high-speed, nonmultiplexed, synchronous external bus interface, which allows the following transfer sizes: byte, word (2 bytes), long word (4 bytes), and line (16 bytes). Line accesses are performed using burst transfers for both reads and writes to provide high data transfer rates.

PROGRAMMING MODEL

The MC68LC040 integrates the functions of the integer unit and MMU. Figure 2 illustrates the registers depicted in the programming model that provide access and control for the three units. The registers are partitioned into two levels of privilege: user and supervisor. User programs, executing in the user mode, can only use the resources of the user model. System software, executing in the supervisor mode, has unrestricted access to all processor resources.

The integer portion of the user programming model, consisting of 16, general-purpose, 32-bit registers and two control registers, is compatible with the M68000 family. The supervisor programming model is used exclusively by MC68LC040 system programmers to implement operating system functions, I/O control, and memory management subsystems. This supervisor/user distinction in the M68000 architecture was carefully planned so that all application software can be written to execute in the nonprivileged user mode and is upward compatible with the MC68LC040 from any M68000 platform without modification. MC68040 code is also compatible; however, all floating-point instructions will cause an exception. Since system software is usually modified by system designers when porting to a new design, the control features are properly placed in the supervisor programming model. For example, the transparent translation registers of the MC68LC040 can only be read or written by the supervisor software; the programming resources of user application programs are unaffected by the existence of the transparent translation registers.

Registers D0–D7 are data registers containing operands for bit and bit field (1- to 32-bits), byte (8-bit), word (16-bit), long-word (32-bit), and quad-word (64-bit) operations. Registers A0–A6 and the stack pointer registers (user, interrupt, and master) are address registers that may be used as software stack pointers or base address registers. Register A7 is the user stack pointer in user mode and is either the interrupt or master stack pointer (A7' or A7") in supervisor mode. In supervisor mode, the active stack pointer (interrupt or master) is selected based on a bit in the status register (SR). The address registers can be used for word and long-word operations, and all 16 general-purpose registers (D0–D7, A0–A7 in Figure 2) can be used as index registers.

The program counter (PC) usually contains the address of the instruction being executed by the MC68LC040. During instruction execution and exception processing, the processor automatically increments the contents of the PC or places a new value in the PC, as appropriate. The SR in the supervisor programming model contains the condition codes that reflect the results of a previous operation and can be used for conditional instruction execution in a program. The lower byte of the SR is accessible in user mode as the condition code register (CCR). Access to the upper byte of the SR is restricted to the supervisor mode.

As part of exception processing, the vector number of the exception provides an index into the exception vector table. The base address of the exception vector table is stored in the vector base register (VBR). The displacement of an exception vector is added to the value in the VBR when the MC68LC040 accesses the vector table during exception processing.

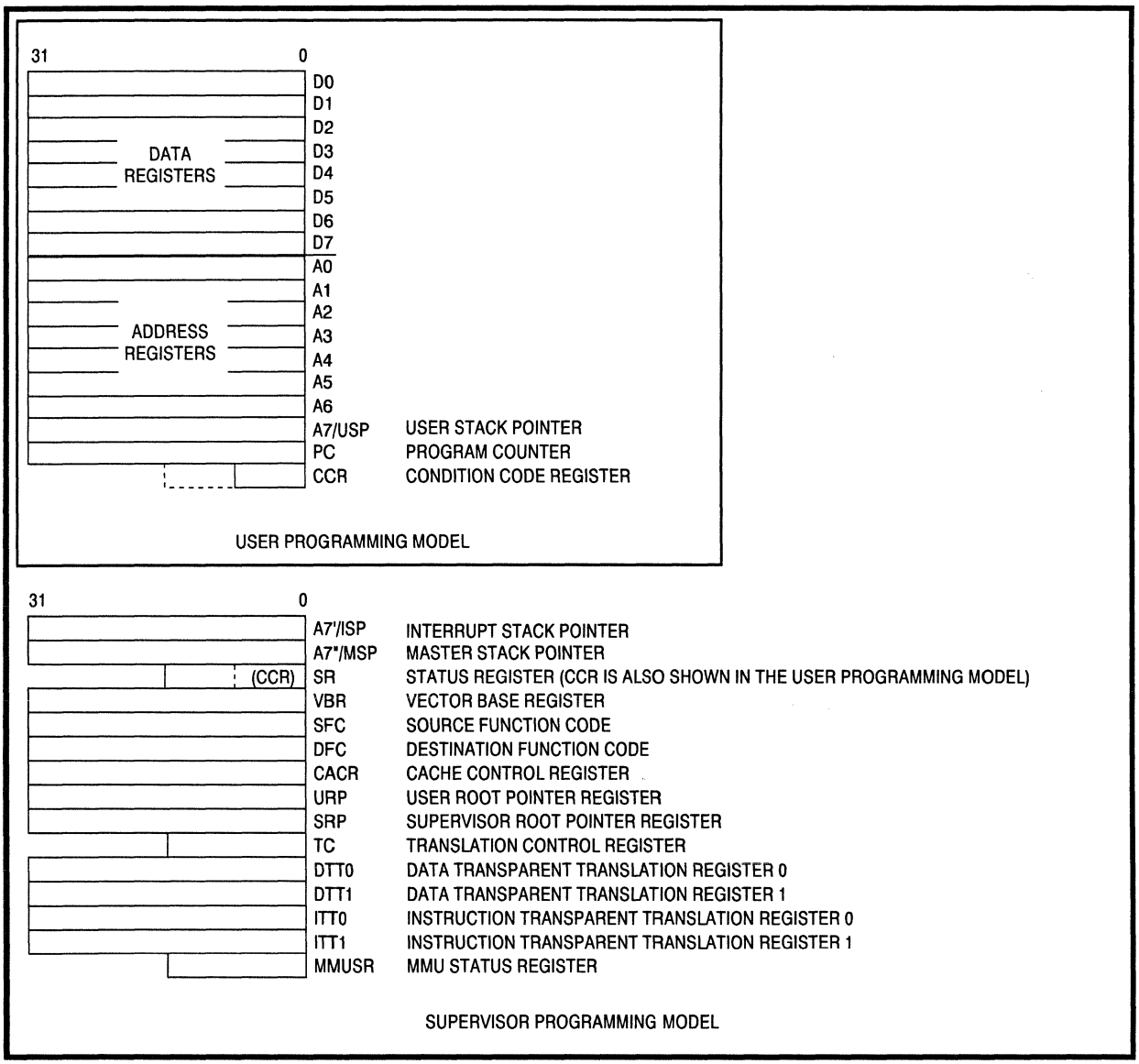


Figure 2. Programming Model

Source function code (SFC) and destination function code (DFC) registers contain 3-bit function codes. Function codes can be considered extensions of the 32-bit linear address. Function codes are automatically generated by the processor to select address spaces for data and program accesses at the user and supervisor modes. The SFC and DFC are used by certain instructions to explicitly specify the function codes for various operations.

The cache control register (CACR) controls enabling of the on-chip instruction and data caches of the MC68LC040.

The supervisor root pointer (SRP) and user root pointer (URP) registers point to the root of the address translation table tree to be used for supervisor mode and user mode accesses. The URP is used if FC2 of the logical address is zero, and the SRP is used if FC2 is one.

The translation control (TC) register enables logical-to-physical address translation and selects either 4K or 8K page sizes. Figure 2 illustrates four transparent translation registers —ITT0 and ITT1 for instruction accesses and DTT0 and DTT1 for data accesses. These registers allow portions of the logical address space to be transparently mapped and accessed without the use of resident descriptors in an ATC. The MMU status register (MMUSR) contains status information from the execution of a PTEST instruction. The PTEST instruction searches the translation tables for the logical address as specified by this instruction's effective address field and the DFC.

DATA TYPES AND ADDRESSING MODES

The MC68LC040 integer unit supports the basic data types listed in Table 1. The instruction set also supports operations on other data types such as memory addresses. The three integer data formats that are common to the integer unit (byte, word, and long word) are the standard twos-complement data formats defined in the M68000 family architecture.

Table 1. Data Types

Operand Data Type	Size	Notes
Bit	1 Bit	—
Bit Field	1–32 Bits	Field of Consecutive Bits
BCD	8 Bits	Packed: 2 Digits/Byte Unpacked: 1 Digit/Byte
Byte Integer	8 Bits	—
Word Integer	16 Bits	—
Long-Word Integer	32 Bits	—
Quad-Word Integer	64 Bits	Any Two Data Registers
16-Byte	128 Bits	Memory-Only, Aligned 16-Byte Boundary

The MC68LC040 addressing modes are listed in Table 2. The register indirect addressing modes support postincrement, predecrement, offset, and indexing, which are particularly useful for handling data structures common to sophisticated applications and high-level languages. The program counter indirect mode also has indexing and offset capabilities; this addressing mode is typically required to support position-independent software. In addition to these addressing modes, the MC68LC040 provides index sizing and scaling features that enhance software performance. Data formats are supported orthogonally by all arithmetic operations and by all appropriate addressing modes.

Table 2. Addressing Modes

Addressing Modes	Syntax
Register Direct	
Data Register Direct	Dn
Address Register Direct	An
Register Indirect	
Address Register Indirect	(An)
Address Register Indirect with Postincrement	(An)+
Address Register Indirect with Predecrement	–(An)
Address Register Indirect with Displacement	(d_{16},An)
Register Indirect with Index	
Address Register Indirect with Index (8-Bit Displacement)	(d_8,An,Xn)
Address Register Indirect with Index (Base Displacement)	(bd,An,Xn)
Memory Indirect	
Memory Indirect Post-indexed	([bd,An],Xn,od)
Memory Indirect Pre-indexed	([bd,An,Xn],od)
Program Counter Indirect with Displacement	(d_{16},PC)
Program Counter Indirect with Index	
PC Indirect with Index (8-Bit Displacement)	(d_8,PC,Xn)
PC Indirect with Index (Base Displacement)	(bd,PC,Xn)
Program Counter Memory Indirect	
PC Memory Indirect Post-indexed	([bd,PC],Xn,od)
PC Memory Indirect Pre-indexed	([bd,PC,Xn],od)
Absolute Data Addressing	
Absolute Short	xxx.W
Absolute Long	xxx.L
Immediate	#<data>

NOTES:

Dn = Data Register, D0–D7

An = Address Register, A0–A7

d_8, d_{16} = A twos-complement or sign-extended displacement; added as part of the effective address calculation; size is 8 (d_8) or 16 (d_{16}) bits; when omitted, assemblers use a value of zero.

Xn = Address or data register used as an index register; form is Xn.SIZE*SCALE, where SIZE is .W or .L (indicates index register size) and SCALE is 1, 2, 4, or 8 (index register is multiplied by SCALE); use of SIZE and/or SCALE is optional.

bd = A twos-complement base displacement; when present, size can be 16 or 32 bits.

od = Outer displacement added as part of effective address calculation after any memory indirection; use is optional with a size of 16 or 32 bits.

PC = Program Counter

<data> = Immediate value of 8, 16, or 32 bits

() = Effective Address

[] = Used as indirect address to long-word address

INSTRUCTION SET OVERVIEW

The instructions provided by the MC68LC040 are listed in Table 3. The instruction set has been tailored to support high-level languages and is optimized for those instructions most commonly executed; however, all instructions listed are fully supported. Many instructions operate on bytes, words, and long words, and most instructions can use any of the addressing modes listed in Table 2.

The MC68LC040 instruction set includes MOVE16, a new user instruction that allows high-speed transfers of 16-byte blocks between external devices, such as memory to memory or coprocessor to memory. For detailed information on the MC68LC040 integer and instruction set, refer to M68000PM/AD, *M68000 Programmer's Reference Manual,* for instructions listed under MC68040 with the exception of floating-point instructions.

INSTRUCTION AND DATA CACHES

Studies have shown that typical programs spend much of their execution time in a few main routines or tight loops. Earlier members of the M68000 family took advantage of this locality-of-reference phenomenon to varying degrees. The MC68LC040 takes further advantage of cache technology with its two, independent, on-chip, physical address space caches, one for instructions and one for data. The caches reduce the processor's external bus activity and increase CPU throughput by lowering the effective memory access time. For a typical system design, the large caches of the MC68LC040 yield a very high hit rate, providing a substantial increase in system performance. Additionally, the caches are automatically burst-filled from the external bus whenever a cache miss occurs.

The autonomous nature of the caches allows instruction stream fetches, data stream fetches, and a third external access to occur simultaneously with instruction execution. For example, if the MC68LC040 requires both an instruction stream access and an external peripheral access and if the instruction is resident in the on-chip cache, the peripheral access proceeds unimpeded rather than being queued behind the instruction fetch. If a data operand is also required and is resident in the data cache, it can also be accessed without hindering either the instruction access from its cache or the peripheral access external to the chip. The parallelism inherent in the MC68LC040 also allows multiple instructions that do not require any external accesses to execute concurrently while the processor is performing an external access for a previous instruction.

CACHE ORGANIZATION

The four-way set-associative instruction and data caches have 64 sets of four 16-byte lines for a total cache storage of 4K bytes each. Each 16-byte line contains an address tag and state information (see Figure 3). State information for each entry consists of a valid flag for the entire line in both instruction and data caches and write status for each long word in the data cache. The write status in the data cache signifies whether or not the long-word data is dirty (meaning that the data in the cache has been modified but has not been written back to external memory) for data in copyback pages.

The caches are accessed by physical addresses from the on-chip MMUs. The translation of the upper bits of the logical address occurs concurrently with the accesses into the set array in the cache by the lower address bits. The output of the ATC is compared with the tag field in the cache to determine if one of the lines in the selected set matches the translated physical address. If the tag matches and the entry is valid, then the cache has a hit.

Table 3. Instruction Set Summary

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
ADDA	Add Address
ADDI	Add Immediate
ADDQ	Add Quick
ADDX	Add with Extend
AND	Logical AND
ANDI	Logical AND Immediate
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
Bcc	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BFCHG	Test Bit Field and Change
BFCLR	Test Bit Field and Clear
BFEXTS	Signed Bit Field Extract
BFEXTU	Unsigned Bit Field Extract
BFFFO	Bit Field Find First One
BFINS	Bit Field Insert
BFSET	Test Bit Field and Set
BFTST	Test Bit Field
BKPT	Breakpoint
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CAS	Compare and Swap Operands
CAS2	Compare and Swap Dual Operands
CHK	Check Register against Bounds
CHK2	Check Register against Upper and Lower Bounds
CINV*	Invalidate Cache Entries
CLR	Clear Operand
CMP	Compare
CMPA	Compare Address
CMPI	Compare Immediate
CMPM	Compare Memory to Memory
CMP2	Compare Register Against Upper and Lower Bounds
CPUSH†	Push then Invalidate Cache Entries
DBcc	Test Condition, Decrement and Branch
DIVS, DIVSL	Signed Divide
DIVU, DIVUL	Unsigned Divide
EOR	Logical Exclusive OR
EORI	Logical Exclusive OR Immediate
EXG	Exchange Registers
EXT, EXTB	Sign Extend
ILLEGAL	Take Illegal Instruction Trap
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL, LSR	Logical Shift Left and Right
MOVE	Move
MOVE16†	16-Byte Block Move
MOVEA	Move Address
MOVE CCR	Move Condition Code Register
MOVE SR	Move Status Register
MOVE USP	Move User Stack Pointer
MOVEC†	Move Control Register
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral Data
MOVEQ	Move Quick
MOVES†	Move Alternate Address Space
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NEGX	Negate with Extend
NOP	No Operation
NOT	Ones Complement
OR	Logical OR
ORI	Logical OR Immediate
PACK	Pack BCD
PEA	Push Effective Address
PFLUSH†	Flush Entry(ies) in the ATCs
PTEST†	Test Logical Address
RESET	Reset External Devices
ROL, ROR	Rotate Left and Right
ROXL, ROXR	Rotate with Extend Left and Right
RTD	Return and Deallocate
RTE	Return from Exception
RTR	Return and Restore Codes
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Scc	Set Conditionally
STOP	Stop
SUB	Subtract
SUBA	Subtract Address
SUBI	Subtract Immediate
SUBQ	Subtract Quick
SUBX	Subtract with Extend
SWAP	Swap Data Register Words
TAS	Test and Set Operand
TRAP	Trap
TRAPcc	Trap Conditionally
TRAPV	Trap on Overflow
TST	Test Operand
UNLK	Unlink
UNPK	Unpack BCD

†MC68LC040 additions or alterations to the MC68030 and MC68EC030 instruction sets.

If the cache hits and the access is a read, the appropriate long word from the cache line is multiplexed onto the appropriate internal bus. If the cache hits and the access is a write, the data, regardless of size, is written to the appropriate portion of the corresponding long-word entry in the cache.

When a data cache miss occurs and a previously valid cache line is needed to cache the new line, any dirty data in the old line will be internally buffered and copied back to memory after the new cache line has been loaded. Pushing of dirty data can be forced by the CPUSH instruction.

Cachability of data in each memory page is controlled by two bits in the page descriptor for each page. Cachable pages may be either writethrough or copyback, with no write-allocate for misses to writethrough pages. Noncachable pages may also be specified as noncachable I/O, forcing accesses to these pages to occur in order of instruction execution.

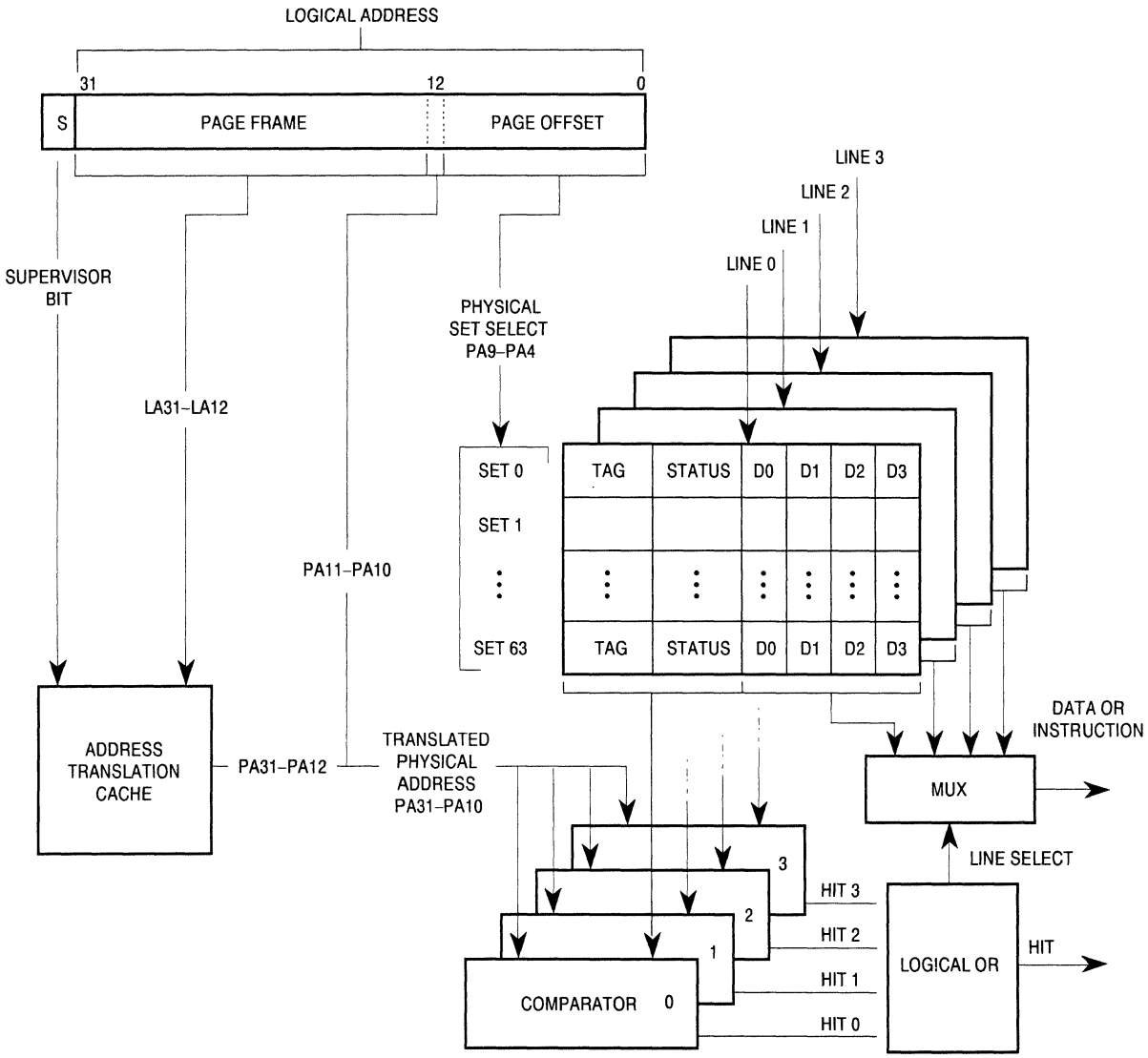


Figure 3. Cache Organization Overview

CACHE COHERENCY

The MC68LC040 has the ability to snoop the external bus during accesses by other bus masters to maintain coherency between the MC68LC040 caches and external memory systems. External write cycles are snooped by both the instruction cache and data cache; whereas, external read cycles are snooped only by the data cache. In addition, external cycles can be flagged on the bus as snoopable or nonsnoopable. When an external cycle is marked as snoopable, the bus snooper checks the caches for a coherency conflict based on the state of the corresponding cache line and the type of external cycle.

Although the internal execution units and the bus snooper circuit all have access to the on-chip caches, the snooper has priority over the execution units to allow the snooper to resolve coherency discrepancies immediately.

CACHE INSTRUCTIONS

The MC68LC040 supports the following instructions for cache maintenance. Both instructions may selectively operate on the data and/or instruction cache.

CINV—Invalidates a single line, all lines in a physical page, or the entire cache.

CPUSH—Pushes selected dirty data cache lines to memory, then invalidates all selected lines.

OPERAND TRANSFER MECHANISMS

The MC68LC040 external synchronous bus supports multiple masters and overlaps arbitration with data transfers. The bus is optimized to perform high-speed transfers to and from an external cache or memory. The data and address buses are each 32 bits wide.

TRANSFER TYPES

The MC68LC040 provides two signals (TT1, TT0) that define four types of bus transfers: normal access, MOVE16 access, alternate access, and interrupt acknowledge access. Normal accesses identify normal memory references; MOVE16 accesses are memory accesses by a MOVE16 instruction; and alternate accesses identify accesses to the undefined address spaces (function code values of 0, 3, 4, 7). The interrupt acknowledge access is used to fetch an interrupt vector during interrupt exception processing.

BURST TRANSFER OPERATION

During burst read/write to cache transfers, the values on the address and transfer type signals do not change; they are the address of the first requested item of the cache line. When the MC68LC040 requests a burst read transfer of a cache line, the address bus indicates the address of the long word in the line needed first, but the memory system is expected to provide data in the following order (modulo 4): 0, 1, 2, 3 (long-word offsets). The first address needed may not be from offset 0; nevertheless, all four long words must be transferred. Burst writes occur in a similar manner.

BUS SNOOPING

Bus snooping ensures that data in main memory is consistent with data in the on-chip caches. If an alternate bus master is performing a read transfer on the bus and snooping is enabled and if the snoop logic determines that the on-chip data cache has dirty data (data valid but not consistent with memory) for this transfer, memory is prevented from responding to the read request, and the MC68LC040 supplies the data directly to the master. If the alternate master is performing a write transfer on the bus and snooping is enabled and if the snooper logic determines that one of the on-chip caches has a valid line for this request, the snooper logic may either invalidate or update the line as selected by the snoop control signals.

EXCEPTION PROCESSING

The MC68LC040 provides the same extensions to the exception stacking process as the MC68030 and MC68040. If the M-bit in the SR is set, the master stack pointer is used for all task-related exceptions. When a nontask-related exception occurs (i.e., an interrupt), the M-bit is cleared, and the interrupt stack pointer is used. This feature allows a task's stack area to be carried within a single processor control block, and new tasks may be initiated by simply reloading the master stack pointer and setting the M-bit.

The externally generated exceptions are interrupts, bus errors, and reset conditions. The interrupts are requests from external devices for processor action; whereas, the bus error and reset signals are used for access control and processor initialization. The internally generated exceptions come from instructions, address errors, tracing, or breakpoints. The TRAP, TRAPcc, TRAPVcc, FTRAPcc, CHK, CHK2, and DIV instructions can all generate exceptions as part of their instruction execution. Tracing behaves like a very high-priority, internally generated interrupt whenever it is processed. The other internally generated exceptions are caused by unimplemented floating-point instructions, illegal instructions, instruction fetches from odd addresses, and privilege violations. Finally, the MMU can generate exceptions for access violations and for invalid descriptors encountered during table searches.

Exception processing for the MC68LC040 occurs in the following sequence: 1) an internal copy is made of the SR, 2) the vector number of the exception is determined, 3) current processor status is saved, and 4) the exception vector offset is determined by multiplying the vector number by four. This offset is then added to the contents of the VBR to determine the memory address of the exception vector. The instruction at the address given in the exception vector is fetched, and normal instruction decoding and execution is started.

All MC68040 and MC68881/MC68882 floating-point instructions trap as floating-point unimplemented instructions. Illegal F-line instructions trap as illegal instructions. Figure 4 illustrates floating-point unimplemented instructions that are handled by a new eight-word floating-point stack frame. The unimplemented stack frame is not recognized by the MC68040. For a complete description of this stack frame, refer to **floating-point exceptions**.

FLOATING-POINT EXCEPTIONS

Unimplemented floating-point instructions generate an exception that causes a new stack frame. The new stack frame contains the effective address, PC, vector offset, PC of faulted instruction, and SR. The effective address field contains the calculated effective address of the operand for floating-point instruction for addressing modes in which the effective address is calculated. For immediate and register direct addressing modes, this field is $0. The PC field contains the address of the next instruction to be executed by the MC68LC040. This value will be restored during RTE execution. The stack frame number and vector offset format number will be four for this eight-word stack. Note that an MC68040 cannot correctly handle a format word of four. The PC of the faulted instruction contains the PC of the floating-point instruction that caused the trap to occur. The information is provided so that the instruction is available to software emulation of floating-point instructions.

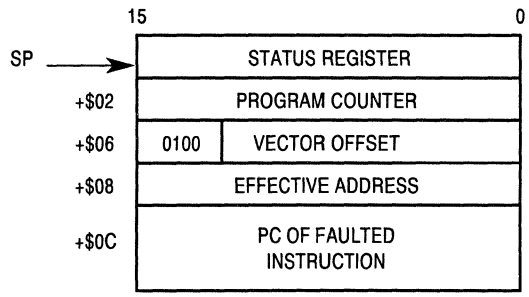


Figure 4. Floating-Point Unimplemented Instruction Stack Frame

MEMORY MANAGEMENT UNITS

The full addressing range of the MC68LC040 is 4G bytes (4,294,967,296 bytes). However, most MC68LC040 systems implement a much smaller physical memory. Nonetheless, by using virtual memory techniques, the system can be made to appear to have a full 4G bytes of physical memory available to each user program. The independent instruction and data MMUs fully support demand-paged virtual-memory operating systems with either 4K or 8K page sizes. In addition to its main function of memory management, each MMU protects supervisor areas from accesses by user programs and also provides write protection on a page-by-page basis. For maximum efficiency, each MMU operates in parallel with other processor activities.

TRANSLATION MECHANISM

The logical-to-physical address translation has been optimized because it is one of the most frequently executed operations of the MC68LC040 MMUs. Each MMU initiates address translation by searching for a descriptor containing the address translation information in the ATC. If the descriptor does not reside in the ATC, then the MMU performs external bus cycles via the bus controller to search the translation tables in physical memory. After being located, the page descriptor is loaded into the ATC, and the address is correctly translated for the access, provided no exception conditions are encountered.

ADDRESS TRANSLATION CACHE

An integral part of the translation function previously described is the dual cache memory that stores recently used logical-to-physical address translation information (page descriptors) for instruction and data accesses. These caches are 64-entry, four-way, set-associative. Each ATC compares the logical address of the incoming access against its entries. If one of the entries matches, there is a hit, and the ATC sends the physical address to the bus controller, which then starts the external bus cycle (provided no hit occurred in the corresponding cache for the access).

TRANSLATION TABLES

The translation tables of the MC68LC040 have a three-level tree structure and reside in main memory. Since only a portion of the complete tree needs to exist at any one time, the tree structure minimizes the amount of memory necessary to set up the tables for most programs. Figure 5 illustrates how either the user root pointer or the supervisor root pointer points to the first level table, depending on the value of the function code for an access. Table entries at the second level of the tree (pointer tables) contain pointers to the third level (page tables). Entries in the page tables contain either page descriptors or indirect pointers to page

descriptors. The mechanism for performing table search operations uses portions of the logical address (as indices) at each level of the search. All addresses in the translation table entries are physical addresses.

There are two variations of table searches for both 4K and 8K page sizes: normal searches and indirect searches. An indirect search differs in that the entry in the third-level page table contains a pointer to a page descriptor rather than the page descriptor itself.

Entries in the translation tables contain control and status information in addition to the physical address information. Control bits specify write protection, limit accesses to supervisor only, and determine cachability of data in each memory page. Each page descriptor also has two user-programmable bits that appear on the UPA0 and UPA1 signals during an external access for use as address modifier bits.

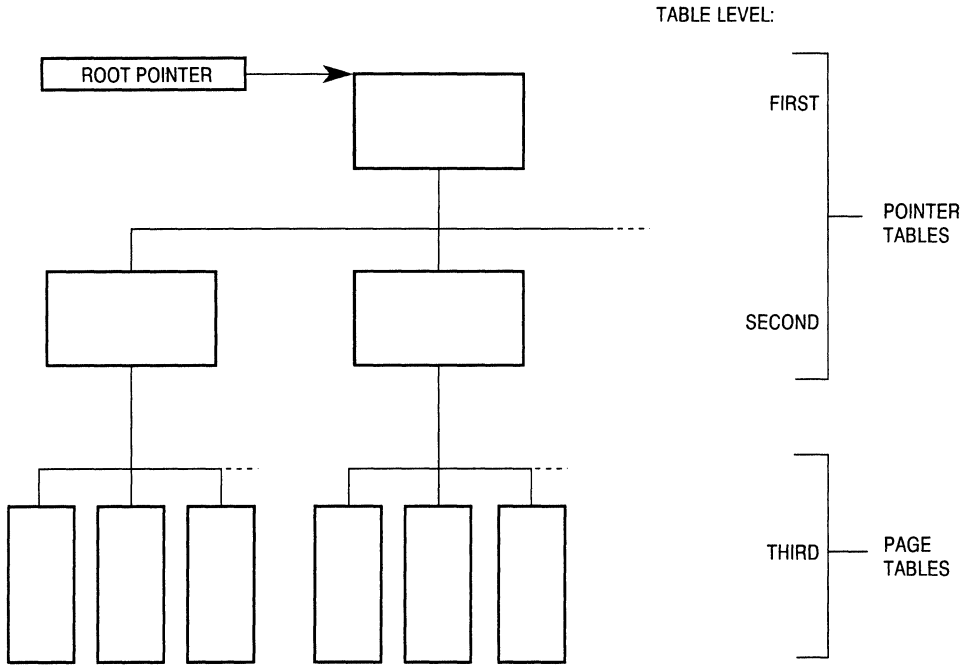


Figure 5. Translation Table Structure

A global bit can be set in each page descriptor to prevent flushing of the ATC entry for that page by some PFLUSH instruction variants, allowing system ATC entries to remain resident during task swaps. If these special PFLUSH instructions are not used, this bit can be user defined. The MMUs automatically maintain access history information for the pages by updating the used (U) and modified (M) status bits.

MMU INSTRUCTIONS

The MMU instructions supported by the MC68LC040 are as follows:

- PFLUSH—Allows flushing of either selected ATC entries by function code and logical address or the entire ATCs.
- PTEST—Takes an address and function code and searches the translation tables for the corresponding entry, which is then loaded into the ATC. The results of the search are available in the MMUSR and are often useful in determining the cause of a fault.

All MC68LC040 MMU instructions are privileged and can only be executed from the supervisor mode.

TRANSPARENT TRANSLATION

Four transparent translation registers, two each for instruction and data accesses, are provided on the MC68LC040 MMU to allow portions of the logical address space to be transparently mapped and accessed without the need for corresponding entries resident in the ATC. Each register can be used to define a range of logical addresses from 16M bytes to 4G bytes with a base address and a mask. All addresses within these ranges are not mapped and are optionally protected against user or supervisor accesses and write accesses. Logical addresses in these areas become the physical addresses for memory access. The transparent translation feature allows rapid movement of large blocks of data in memory or I/O space without disturbing the context of the on-chip ATCs or incurring delays associated with translation table searches.

SIGNAL DESCRIPTION

Figure 6 illustrates the functional signal groups, and Table 4 lists the signals on the MC68LC040 signal functions.

The test signals, TRST, TMS, TCK, TDI, and TDO, comply with subset P1149.1 of the IEEE testability bus standard. The MC68LC040 is pin compatible with the MC68040. DLE has been modified and renamed JS0 to maintain compatibility with previous MC68040 designs and JTAG scan patterns. The JTAG scan compatibility pin JS0 should be tied to V_{CC} or GND during normal operation. An existing design currently using a pullup or pulldown resistor is acceptable. Due to power considerations, this pin may not float during normal operation.

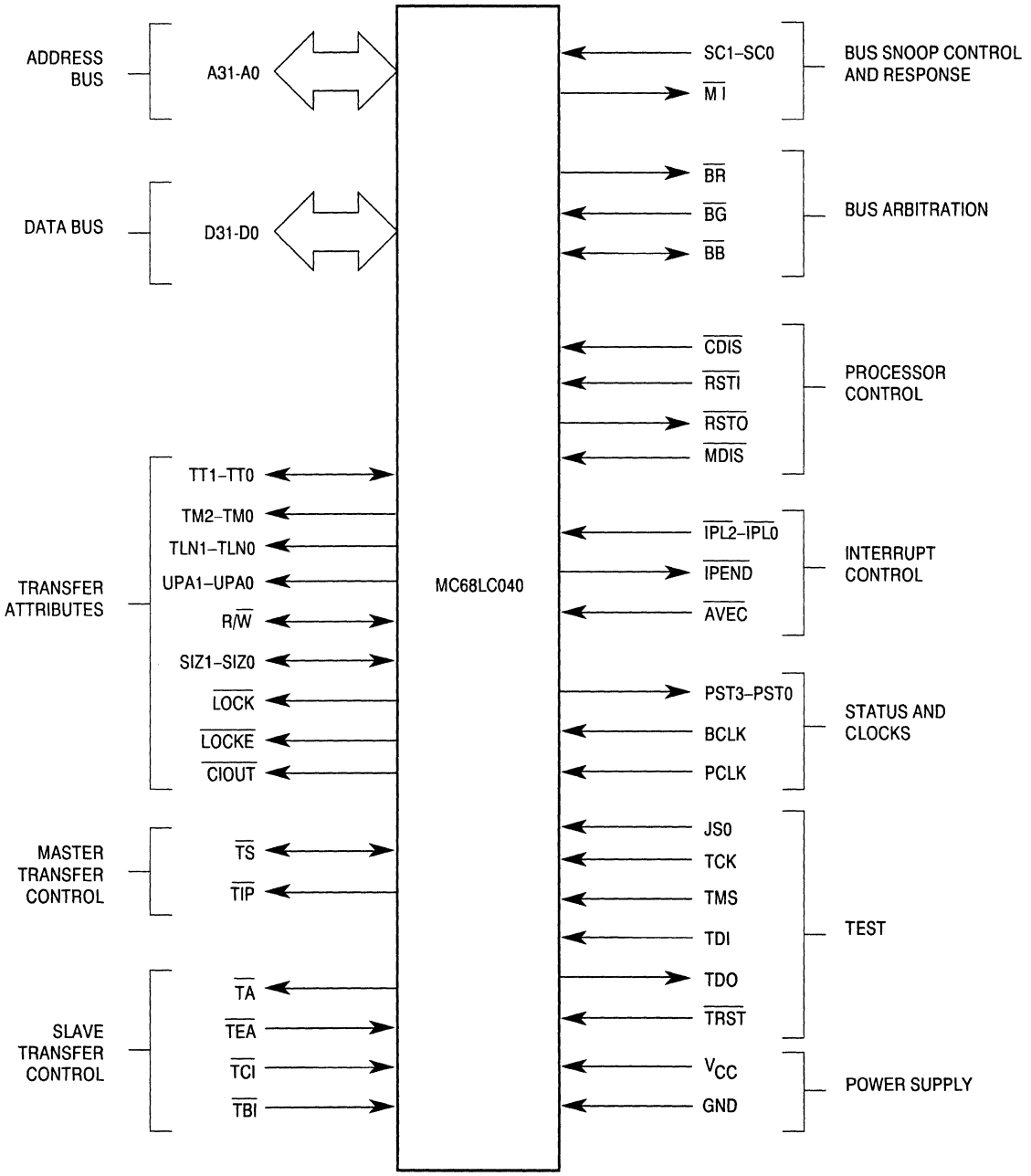


Figure 6. Functional Signal Groups

Table 4. Signal Index

Signal Name	Mnemonic	Function
Address Bus	A31–A0	32-bit address bus used to address any of 4 Gbytes.
Data Bus	D31–D0	32-bit data bus used to transfer up to 32 bits of data per bus transfer.
Transfer Type	TT1,TT0	Indicates the general transfer type: normal, MOVE16, alternate logical function code, and acknowledge.
Transfer Modifier	TM2–TM0	Indicates supplemental information about the access.
Read/Write	R/W	Identifies the transfer as a read or write.
Transfer Size	SIZ1, SIZ0	Indicates the data transfer size. These signals, together with A0 and A1, define the active sections of the data bus.
Transfer Line Number	TLN1, TLN0	Indicates which cache line in a set is being pushed or loaded by the current line transfer.
User Programmable Attributes	UPA1, UPA0	User-defined signals, controlled by the corresponding user attribute bits from the address translation entry.
Bus Lock	LOCK	Indicates a bus transfer is part of a read-modify-write operation, and that the sequence of transfers should not be interrupted.
Bus Lock End	LOCKE	Indicates the current transfer is the last in a locked sequence of transfers.
Cache Inhibit Out	CIOUT	Indicates the processor will not cache the current bus transfer.
Transfer Start	TS	Indicates the beginning of a bus transfer.
Transfer in Progress	TIP	Asserted for the duration of a bus transfer.
Transfer Acknowledge	TA	Asserted to acknowledge a bus transfer.
Transfer Error Acknowledge	TEA	Indicates an error condition exists for a bus transfer.
Transfer Cache Inhibit	TCI	Indicates the current bus transfer should not be cached.
Transfer Burst Inhibit	TBI	Indicates the slave cannot handle a line burst access.
Snoop Control	SC1,SC0	Indicates the snooping operation required during an alternate master access.
Memory Inhibit	MI	Inhibits memory devices from responding to an alternate master access during snooping operations.
Bus Request	BR	Asserted by the processor to request bus mastership.
Bus Grant	BG	Asserted by an arbiter to grant bus mastership to the processor.
Bus Busy	BB	Asserted by the current bus master to indicate it has assumed ownership of the bus.
Cache Disable	CDIS	Dynamically disables the internal caches to assist emulator support.
MMU Disable	MDIS	Disables the translation mechanism of the MMUs.
Reset In	RSTI	Processor reset.
Reset Out	RSTO	Asserted during execution of a RESET instruction to reset external devices.
Interrupt Priority Level	IPL2–IPL0	Provides an encoded interrupt level to the processor.
Interrupt Pending	IPEND	Indicates an interrupt is pending.
Autovector	AVEC	Used during an interrupt acknowledge transfer to request internal generation of the vector number.
Processor Status	PST3–PST0	Indicates internal processor status.
Bus Clock	BCLK	Clock input used to derive all bus signal timing.
Processor Clock	PCLK	Clock input used for internal logic timing. The PCLK frequency is exactly 2X the BCLK frequency.
Test Clock	TCK	Clock signal for the IEEE P1149.1 test access port (TAP).
Test Mode Select	TMS	Selects the principle operations of the test-support circuitry.

Table 4. Signal Index (Continued)

Test Data Input	TDI	Serial data input for the TAP.
Test Data Output	TDO	Serial data output for the TAP.
Test Reset	TRST	Provides an asynchronous reset of the TAP controller.
Power Supply	V_{CC}	Power supply.
JTAG Scan 0	JS0	MC68040 JTAG scan compatibility pin.
Ground	GND	Ground connection.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Signal Name	Symbol	Value	Unit
Supply Voltage	V_{CC}	–0.3 to +7.0	V
Input Voltage	V_{in}	–0.8 to +7.0	V
Maximum Operating Junction Temperature	T_J	110	°C
Storage Temperature Range	T_{stg}	–55 to 150	°C

This device contains protective circuitry against damage due to high static voltages or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{CC}).

THERMAL CHARACTERISTICS—PGA PACKAGE

Characteristic	Symbol	Value	Rating
Thermal Resistance— Junction to Case	R_{JC}	3	°C/W

DC ELECTRICAL SPECIFICATIONS (V_{CC} = 5.0 VDC ±5 %; GND = 0 VDC)

Characteristic	Symbol	Min	Max	Unit
Input High Voltage	V_{IH}	2	V_{CC}	V
Input Low Voltage	V_{IL}	GND	0.8	V
Undershoot	—	—	–0.8	V
Input Leakage Current AVEC, BCLK, BG, CDIS, @ 0.5/2.4 V MDIS, IPLn, PCLK, RSTI, SCn,TBI, TLNn, TCI, TCK, TEA	I_{in}	20	20	µA
Hi-Z (Off-State) Leakage Current An, BB, CIOUT, Dn, LOCK, @ 0.5/2.4 V LOCKE, R/W, SIZn, TA, TDO, TIP, TMn, TLNn, TS, TTn, UPAn	I_{TSI}	20	20	µA
Signal Low Input Current TMS, TDI, TRST, V_{IL} = 8 V	I_{IL}	–1.1	–0.18	mA
Signal High Input Current TMS, TDI, TRST, V_{IH} = 2.0 V	I_{IH}	–0.94	–0.16	mA
Output High Voltage, I_{OH} = 5 mA	V_{OH}	2.4	—	V
Output Low Voltage, I_{OL} = 5 mA	V_{OL}	—	0.5	V
Power Dissipation (T_J = 110°C)	P_D	TBD	TBD	W
Capacitance[1], V_{in} = 0 V, f = 1-MHz	C_{in}	—	20	pF

Notes:

1. Capacitance is periodically sampled rather than 100% tested.

TBD—To Be Determined

CLOCK AC TIMING SPECIFICATIONS (see Figure 7)

Num	Characteristic	20 MHz		25 MHz		33 MHz		Unit
		Min	Max	Min	Max	Min	Max	
	Frequency of Operation	16.67	20	16.67	25	16.6	33.3	MHz
1	PCLK Cycle Time	25	30	20	30	15	30	ns
2	PCLK Rise Time	—	1.7	—	1.7	—	1.7	ns
3	PCLK Fall Time	—	1.6	—	1.6	—	1.6	ns
4	PCLK Duty Cycle Measured at 1.5 V	48	52	47.5	52.5	46.67	53.33	%
4a[1]	PCLK Pulse Width High Measured at 1.5 V	12	13	9.5	10.5	7	8	ns
4b[1]	PCLK Pulse Width Low Measured at 1.5 V	12	13	9.5	10.5	7	8	ns
5	BCLK Cycle Time	50	60	40	60	30	60	ns
6,7	BCLK Rise and Fall Time	—	4	—	4	—	3	ns
8	BCLK Duty Cycle Measured at 1.5 V	40	60	40	60	40	60	%
8a[1]	BCLK Pulse Width High Measured at 1.5 V	20	30	16	24	12	18	ns
8b[1]	BCLK Pulse Width Low Measured at 1.5 V	20	30	16	24	12	18	ns
9	PCLK, BCLK Frequency Stability	—	1000	—	1000	—	1000	ppm
10	PCLK to BCLK Skew	—	12	—	9	—	n/a	ns

1. Specification value at maximum frequency of operation.

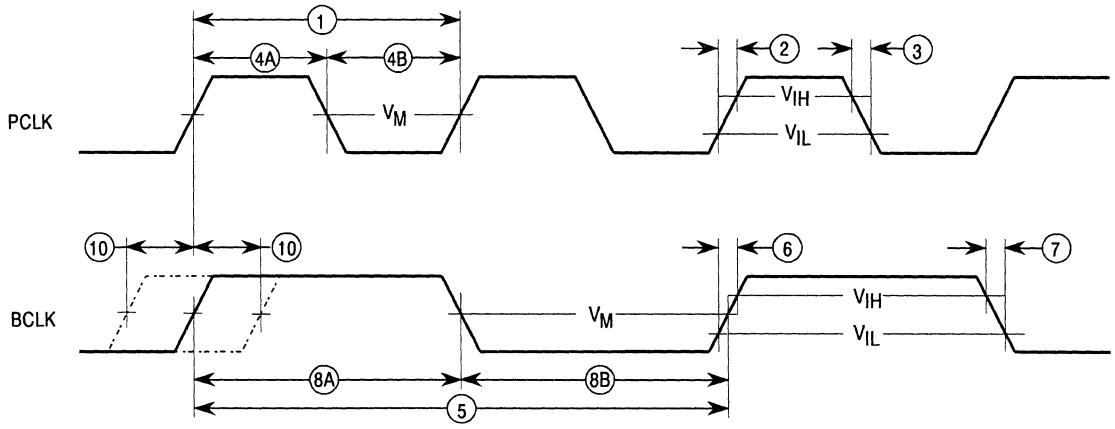


Figure 7. Clock Input Timing Diagram

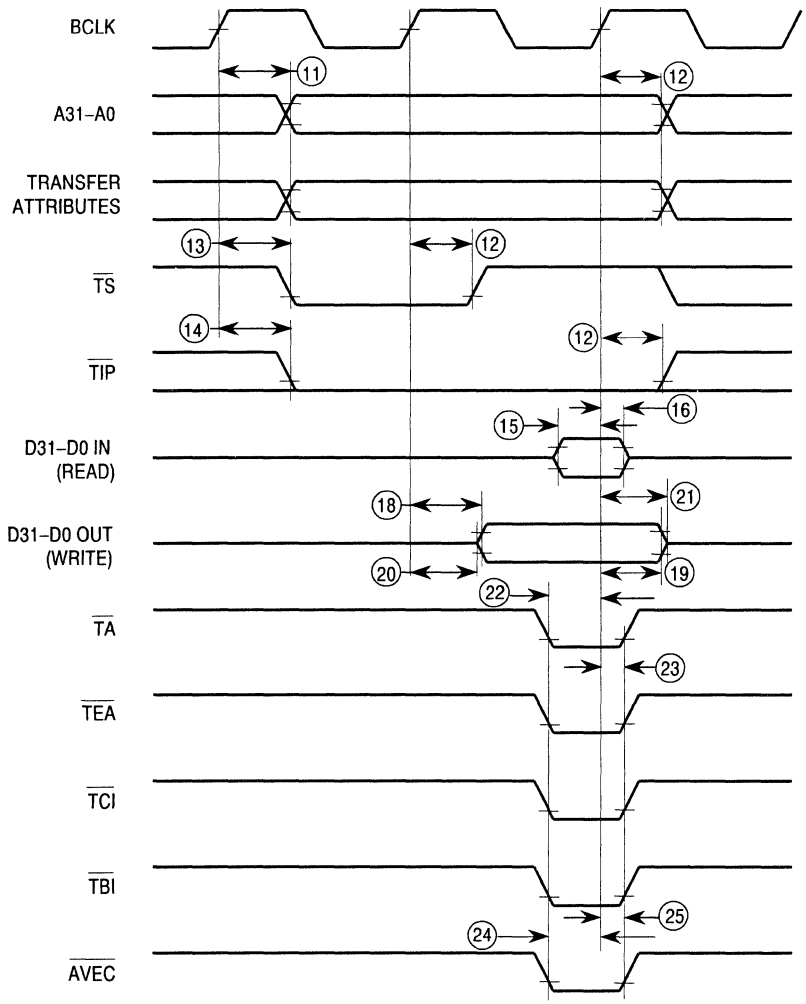
OUTPUT AC TIMING SPECIFICATIONS (see Figures 8–12[1])

Num	Characteristic	20 MHz		25 MHz		33 MHz		Unit
		Min	Max	Min	Max	Min	Max	
11	BCLK to Address CIOUT, LOCK, LOCKE, R/W, SIZn, TLNn,TMn, TTn, UPAn Valid	11.5	35	9	30	6.5	25	ns
12	BCLK to Output Invalid (Output Hold)	11.5	—	9	—	6.5	—	ns
13	TCLK to TS Valid	11.5	35	9	30	6.5	25	ns
14	BCLK to TIP Valid	11.5	35	9	30	6.5	25	ns
18	BCLK to Data-Out Valid	11.5	37	9	32	6.5	27	ns
19	BCLK to Data-Out Invalid (Output Hold)	11.5	—	9	—	6.5	—	ns
20	BCLK to Output Low Impedance	11.5	—	9	—	6.5	—	ns
21	BCLK to Data-Out High Impedance	11.5	25	9	20	6.5	17	ns
38	BCLK to Address, CIOUT, LOCK, LOCKE, R/W, SIZn, TS, TLNn, TMn, TTn, UPAn High Impedance	11.5	23	9	18	6.5	15	ns
39	BCLK to BB, TA, TIP High Impedance	23	33	19	28	6.5	23	ns
40	BCLK to BR, BB Valid	11.5	35	9	30	14	25	ns
43	BCLK to MI Valid	11.5	35	9	30	6.5	25	ns
48	BCLK to TA Valid	11.5	35	9	30	6.5	25	ns
50	BCLK to IPEND, PSTn, RSTO Valid	11.5	35	9	30	6.5	25	ns

1. Output timing is specified for a valid signal measured at the pin. Timing is specified driving an unterminated 30-Ω transmission line with a length characterized by a 2.5-ns one-way propagation delay. Buffer output impedance is typically 30 Ω; the buffer specifications include approximately 5 ns for the signal to propagate the length of the transmission line and back. Refer to MC68040DH/AD, *MC68040 Designer's Handbook*, for further information on transmission line environments.

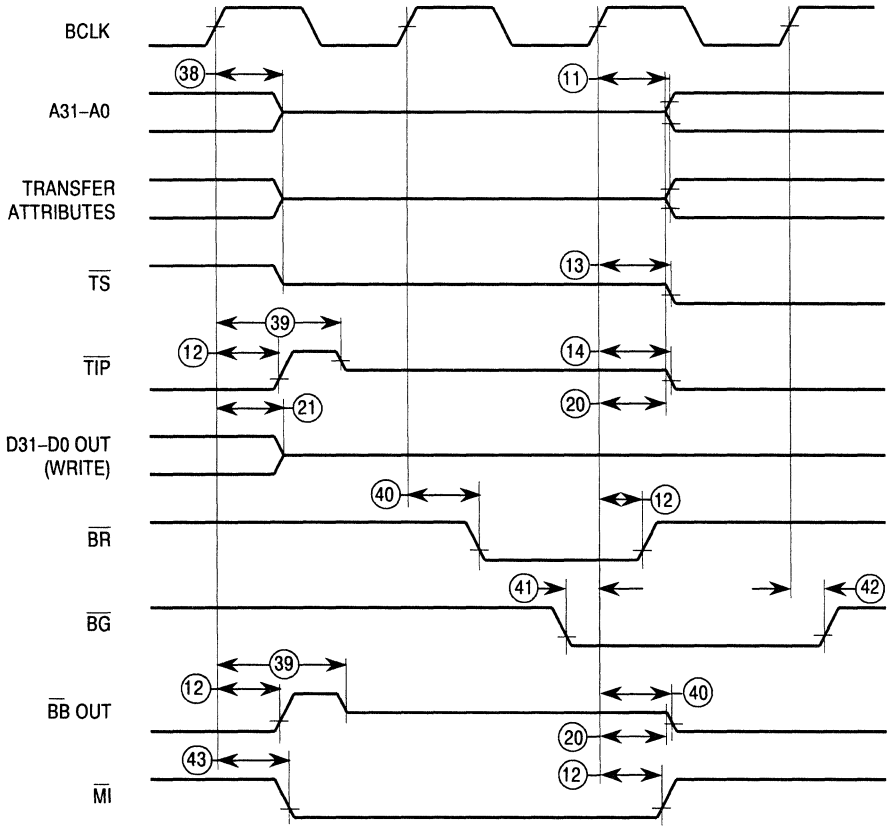
INPUT AC TIMING SPECIFICATIONS (see Figures 7–12)

Num	Characteristic	20 MHz		25 MHz		33 MHz		Unit
		Min	Max	Min	Max	Min	Max	
15	Data-In Valid to BCLK (Setup)	6	—	5	—	4	—	ns
16	BCLK to Data-In Invalid (Hold)	5	—	4	—	4	—	ns
17	BCLK to Data-In High Impedance (Read Followed by Write)	—	61	—	49	—	36.5	ns
22a	TA Valid to BCLK (Setup)	12.5	—	10	—	10	—	ns
22b	TEA Valid to BCLK (Setup)	12.5	—	10	—	10	—	ns
22c	TCI Valid to BCLK (Setup)	12.5	—	10	—	10	—	ns
22d	TBI Valid to BCLK (Setup)	14	—	11	—	10	—	ns
23	BCLK to TA, TEA, TCI, TBI Invalid (Hold)	2.5	—	2	—	2	—	ns
24	AVEC Valid to BCLK (Setup)	6	—	5	—	5	—	ns
25	BCLK to AVEC Invalid (Hold)	2.5	—	2	—	2	—	ns
41a	BB Valid to BCLK (Setup)	8	—	7	—	7	—	ns
41b	BG Valid to BCLK (Setup)	10	—	8	—	7	—	ns
41c	CDIS, MDIS Valid to BCLK (Setup)	12.5	—	10	—	8	—	ns
41d	IPLn Valid to BCLK (Setup)	5	—	4	—	3	—	ns
42	BCLK to BB, BG, CDIS, MDIS, IPLn Invalid (Hold)	2.5	—	2	—	2	—	ns
44a	Address Valid to BCLK (Setup)	10	—	8	—	7	—	ns
44b	SIZn Valid to BCLK (Setup)	15	—	12	—	8	—	ns
44c	TTn Valid to BCLK (Setup)	7.5	—	6	—	8.5	—	ns
44d	R/W Valid to BCLK (Setup)	7.7	—	6	—	5	—	ns
44e	SCn Valid to BCLK (Setup)	12.5	—	10	—	11	—	ns
45	BCLK to Address SIZn, TTn, R/W, SCn Invalid (Hold)	2.5	—	2	—	2	—	ns
46	TS Valid to BCLK (Setup)	6	—	5	—	9	—	ns
47	BCLK to TS Invalid (Hold)	2.5	—	2	—	2	—	ns
49	BCLK to BB High Impedance (MC68LC040 Assumes Bus Mastership)	—	11	—	9	—	9	ns
51	RSTI Valid to BCLK	6	—	5	—	4	—	ns
52	BCLK to RSTI Invalid	2.5	—	2	—	2	—	ns
53	Mode Select Setup to RSTI Negated	25	—	20	—	20	—	ns
54	RSTI Negated to Mode Selects Invalid	2.5	—	2	—	2	—	ns



NOTE: Transfer Attribute Signals = UPAn, SIZn, TTn, TMn, TLNn, R/$\overline{W}$, $\overline{LOCK}$, $\overline{LOCKE}$, $\overline{CIOUT}$

Figure 8. Read/Write Timing



NOTE: Transfer Attribute Signals = UPAn, SIZn, TTn, TMn, TLNn, R/W, LOCK, LOCKE, CIOUT

Figure 9. Bus Arbitration Timing

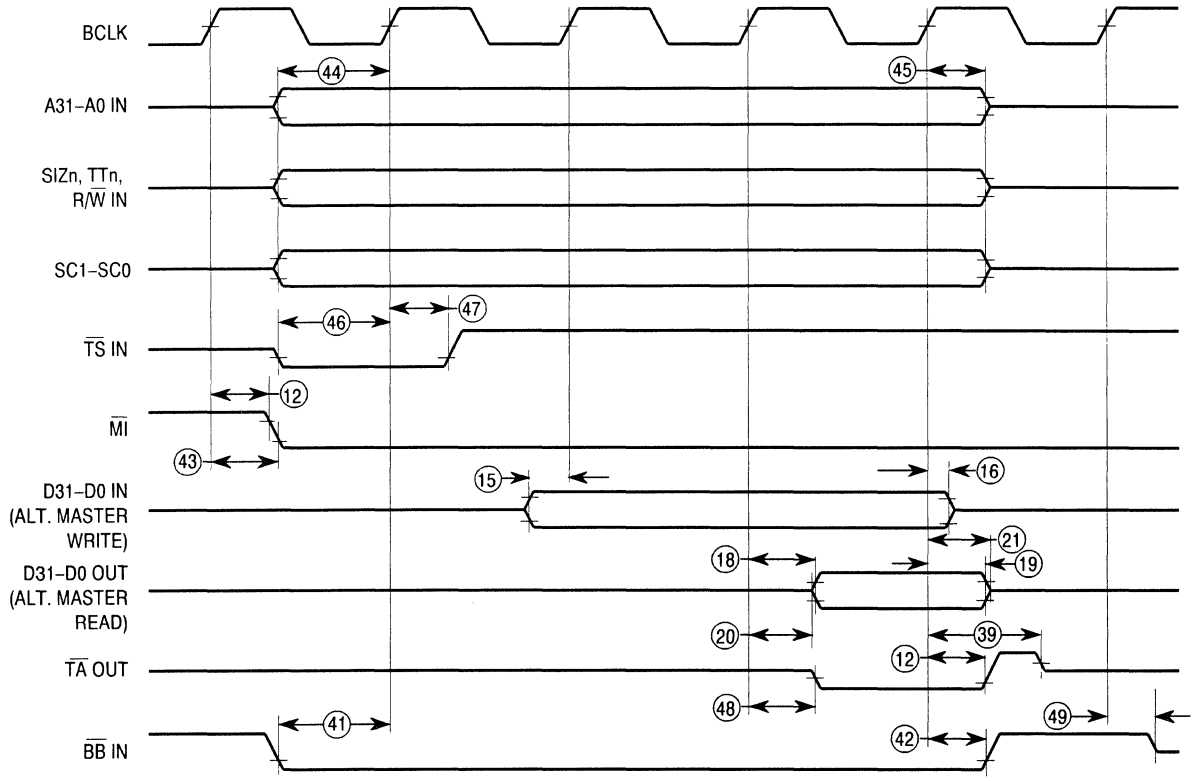


Figure 10. Snoop Hit Timing

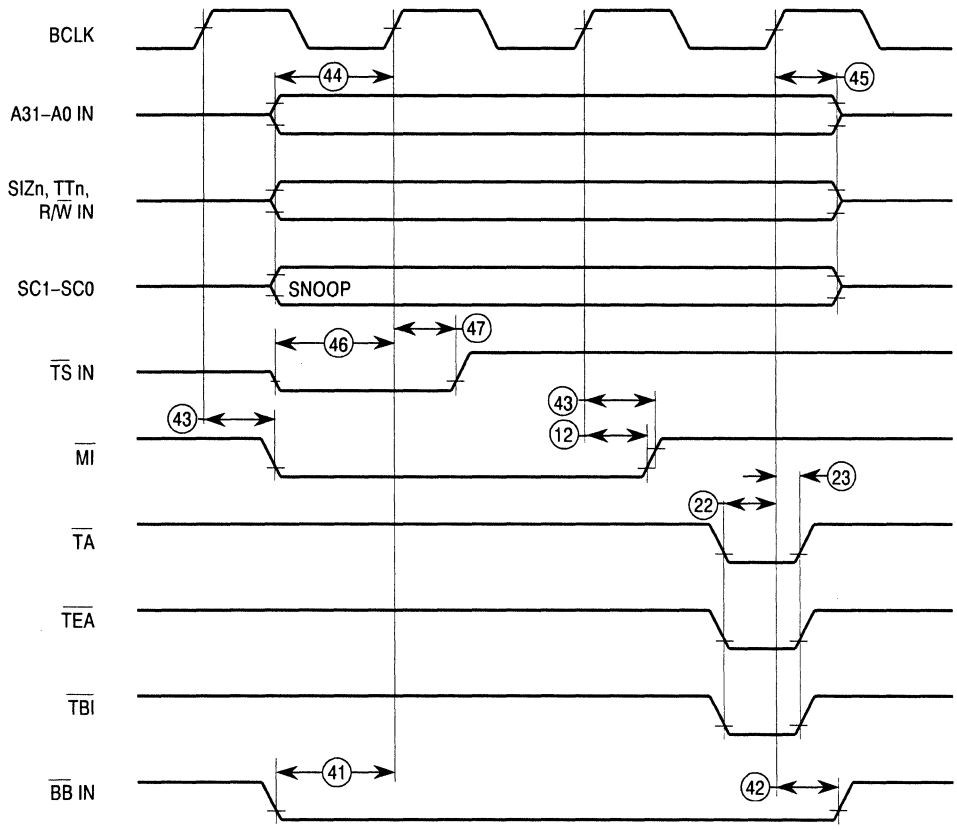


Figure 11. Snoop Miss Timing

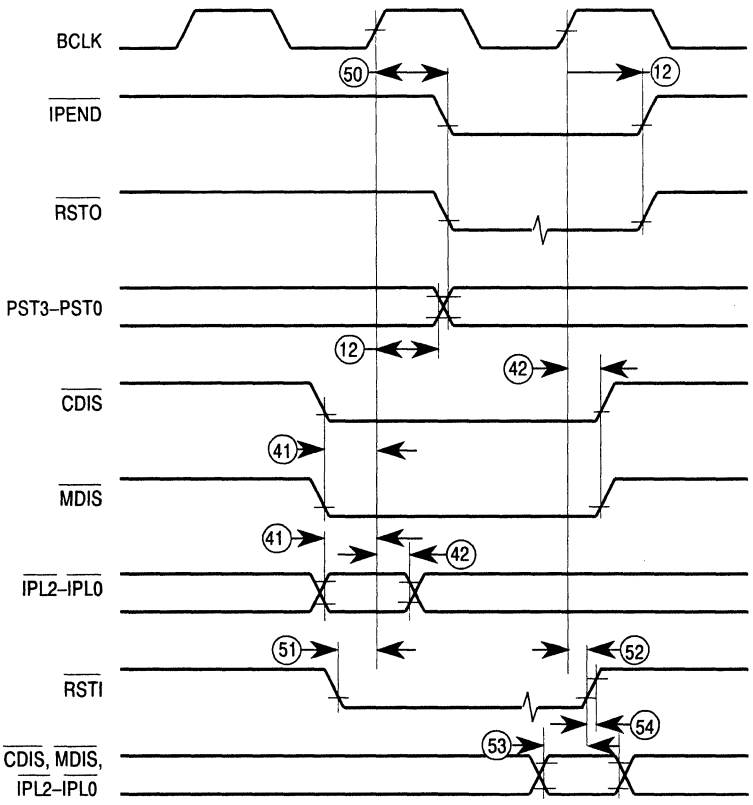
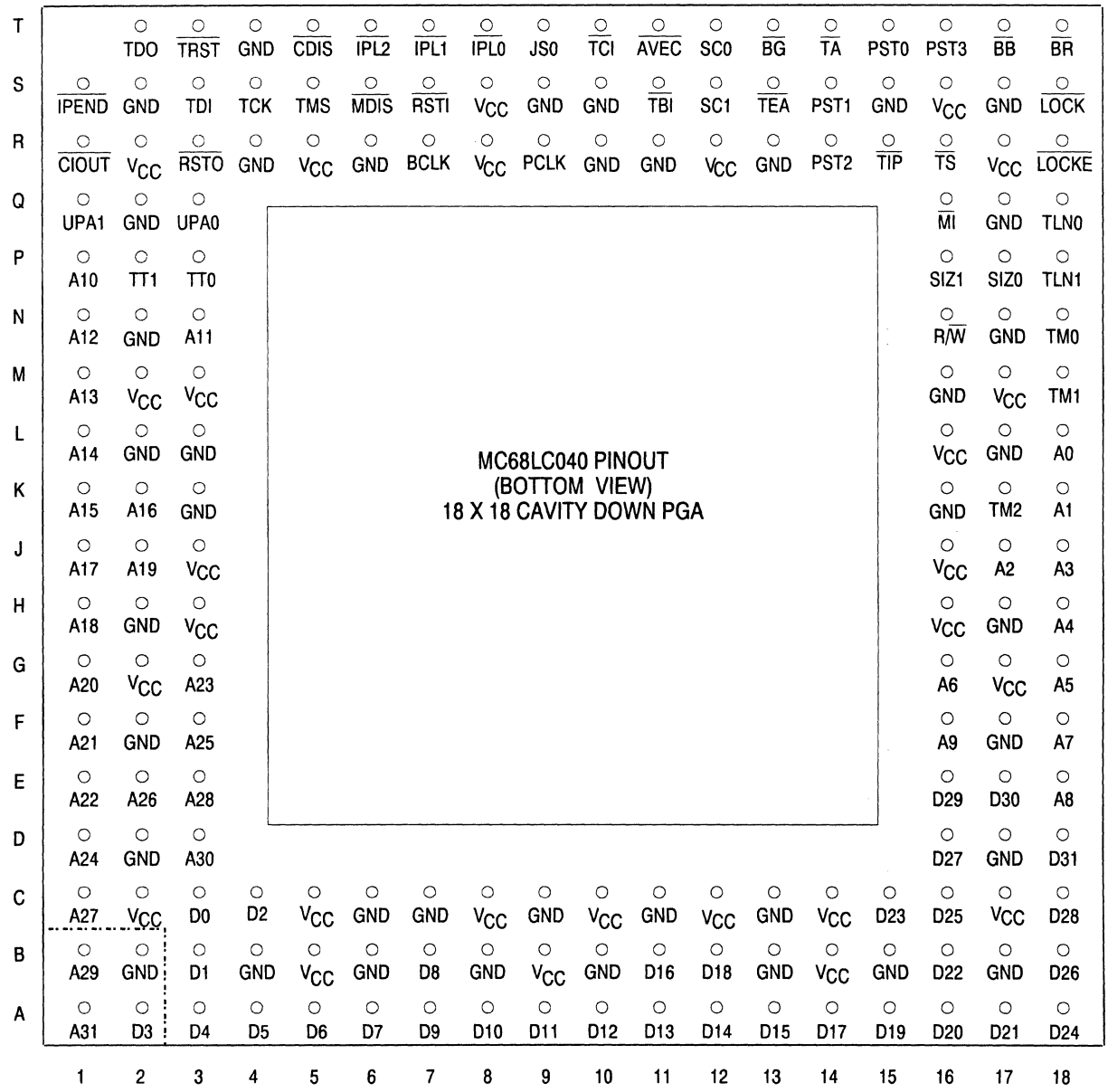


Figure 12. Other Signal Timing

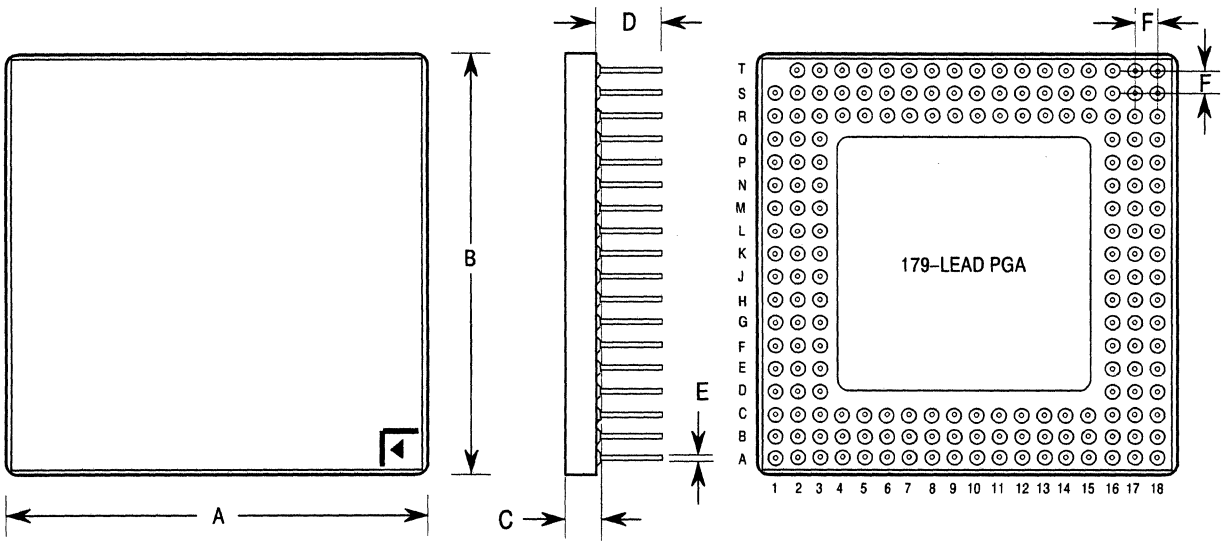
MECHANICAL DATA

PIN ASSIGNMENT



PLL	GND	V_{CC}
	S9, R6, R10	R8, S8
Internal Logic	C6, C7, C9, C11, C13, K3, K16, L3, M16, R4, R11, R13, S10, S6, S10, T4	C5, C8, C10, C12, C14, H3, H16, J3, J16, L16, M3, R5, R12
Output Drivers	B2, B4, B6, B8, B10, B13, B15, B17, D2, D17, F2, F17, H2, H17, L2, L17, N2, N17, Q2, Q17, S2, S15, S17	B5, B9, B14, C2, C17, G2, G17, M2, M17, R2, R17, S16

PACKAGE DIMENSIONS



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	46.863	47.625	1.845	1.875
B	46.863	47.625	1.845	1.875
C	2.3876	2.9464	.094	.116
D	4.318	4.826	.170	.190
E	0.44	0.55	0.017	0.022
F	2.54 BSC		0.100 BSC	

Case To Be Determined

Literature Distribution Centers:

USA: Motorola Literature Distribution: P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan.
ASIA-PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.

A30261 PRINTED IN USA 8/91 IMPERIAL LITHO 80658 18,000 MPU YGABAA