Order this document
by MC68EC040/D

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68EC040

Preliminary Technical Summary

Third-Generation 32-Bit Embedded Controller

The MC68EC040 is Motorola's third generation of M68000-compatible, high-performance, 32-bit embedded controllers. The MC68EC040 is an embedded controller employing a highly integrated architecture to provide very high performance in a monolithic HCMOS device. On a single chip, the MC68EC040 integrates an MC68040-compatible integer unit, an access control unit (ACU), and independent 4K-byte instruction and 4K-byte 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 MC68EC040 also directly supports cache coherency in multimaster applications with dedicated on-chip bus snooping logic. For detailed information on the MC68EC040, refer to MC68EC040UM/AD, *MC68EC040 32-Bit Embedded Controller User's Manual.*

The MC68EC040 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 MC68EC040 is designed to take full advantage of existing M68000 Family code while providing the improved performance of the MC68040 architecture. The increased performance is well suited to CPU-intensive embedded control applications. The MC68EC040 is implemented in Motorola's latest HCMOS technology, providing an ideal balance between speed, power, and physical device size.

Figure 1 is a simplified block diagram of the MC68EC040. Instruction execution is pipelined in the integer unit. Bus snooping ensures cache coherency in multimaster and multiprocessing embedded control applications.

The main features of the MC68EC040 are as follows:

- 20 MIPS Integer Performance at 25 MHz, 16 MIPS Integer Performance at 20 MHz
- 4K-Byte Instruction Cache and 4K-Byte Data Cache Accessed Simultaneously
- 32-Bit, Nonmultiplexed External Address and Data Buses with Synchronous Interface
- User-Object-Code Compatible with All M68000 Microprocessors
- Concurrent Integer Unit, ACU, Bus Controller, and Bus Snooper Maximize Throughput
- Independent Instruction and Data AC Registers
- 4-Gbyte Direct Addressing Range
- Software Support Including Optimizing C Compiler

This document contains information on a new product. Specifications and information herein are subject to change without notice.

MOTOROLA

Rev 1
(Replaces NP522)

INTRODUCTION

The MC68EC040 is an enhanced, 32-bit, HCMOS embedded controller that combines the high-performance integer processing unit of the MC68040 microprocessor with independent 4K-byte data and instruction caches. The MC68EC040 maintains the 32-bit registers available with the entire M68000 Family as well as the 32-bit address and data paths, a rich instruction set, and versatile addressing modes. Instruction execution proceeds in parallel with accesses to the internal caches, ACU and bus controller activity. Additionally, the integer unit is optimized for high-level language environments. The full addressing range of the MC68EC040 is 4 Gbytes (4,294,967,296 bytes); however, most MC68EC040 systems implement a much smaller physical memory.

The high degree of parallelism in the instruction execution unit is achieved through a multi-stage pipeline, the bus controller, and separate instruction and data caches. The multi-stage pipeline will operate on up to six instructions concurrent with bus controller operations, ACU, and separate instruction and data cache accesses. Multiple internal buses, separate instruction and data caches, and a sophisticated bus controller decouples the high-performance MC68EC040 integer unit from the external bus. The copyback cache mode provides even higher performance by reducing the external bus activity for data writes. The MC68EC040 will burst fill four 32-bit long words into the instruction and data caches, when needed by the integer unit, to increase the decoupling of the controller from the external bus. The sophisticated bus controller can delay writes when the external bus is needed to fetch instruction or data which is not present in the internal caches. By reducing the external bus utilization, the MC68EC040 can sustain a much higher performance without an expensive external memory system. The decoupling of the external bus allows the external memory system to be slower and much less expensive without a significant impact on the MC68EC040 performance.

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 data bus, allowing simultaneous access to both. The Harvard-style architecture allows a combined internal transfer rate of 200 Mbytes per second at a 25-MHz clock rate when accessing both the data and instruction caches simultaneously. The data cache provides writethrough or copyback write modes that can be configured on a 16-Mbyte to 4-Gbyte block basis by using the access control unit. Cache coherency is maintained by the bus snooping of the bus controller on the MC68EC040.

The MC68EC040 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. The bus controller unit is capable of handling 50 Mbyte per second of long word transfers and burst transfers of 80 Mbytes per second.

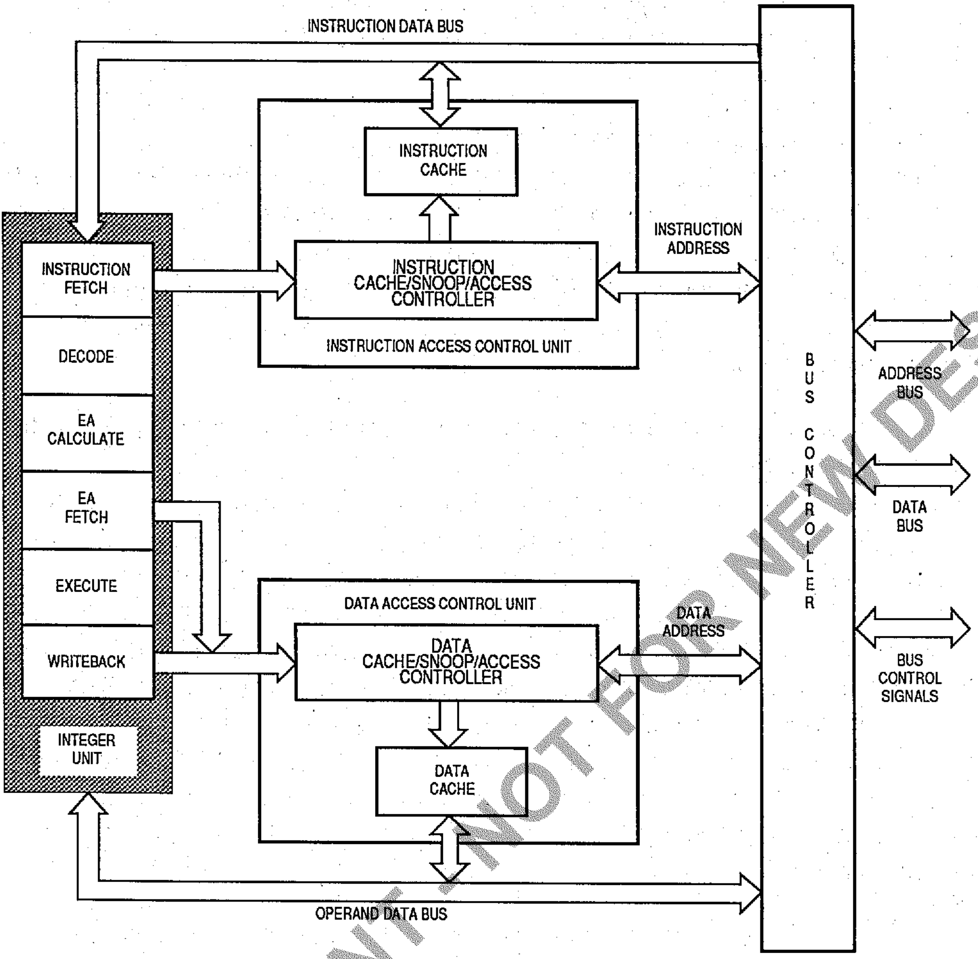


Figure 1. Block Diagram

PROGRAMMING MODEL

The MC68EC040 integrates the functions of the integer unit, ACU, and caches. As shown in Figure 2, the registers depicted in the programming model 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 controller resources.

The integer portion of the user programming model consists of 16 general-purpose 32-bit registers and two control registers. The MC68EC040 user programming model is compatible with the entire M68000 Family. The supervisor programming model is used exclusively by MC68EC040 system programmers to implement operating system functions, ACU, and I/O control. 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 migrate to the MC68EC040 from any M68000 platform without modification. 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 access control registers of the MC68EC040 can only be read or written by the supervisor software; the programming resources of user application programs are unaffected by the existence of the access control 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.

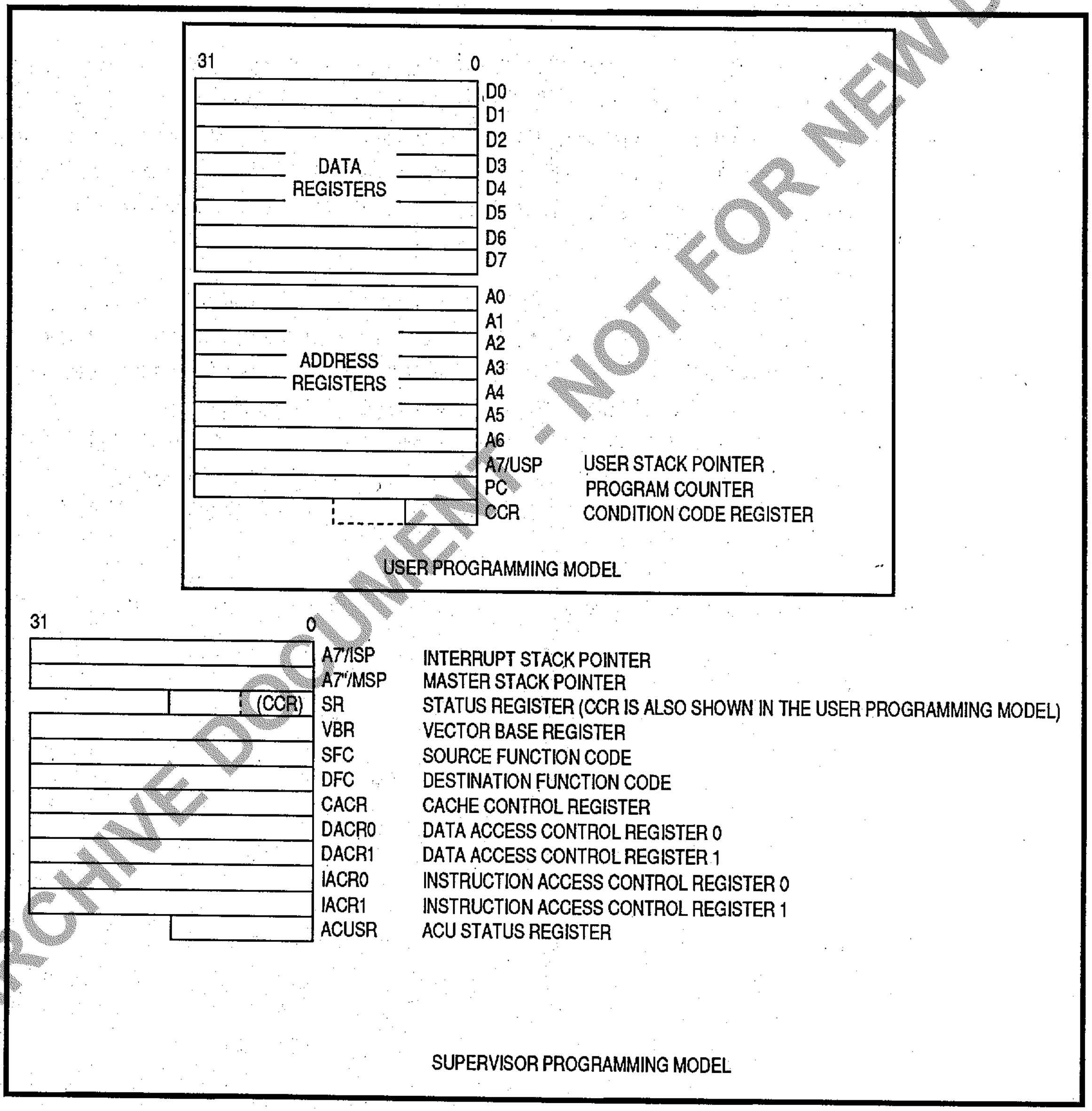


Figure 2. Programming Model

The program counter (PC) contains the address of the instruction being executed by the MC68EC040. During instruction execution and exception processing, the controller 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 MC68EC040 accesses the vector table during exception processing.

Alternate function code registers, SFC and DFC (source and destination), contain 3-bit function codes. Function codes can be considered extensions of the 32-bit linear address. Function codes are automatically generated by the controller to select address spaces for data and program accesses at the user and supervisor modes. The alternate function code registers 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 MC68EC040.

The access control registers (DACR0, DACR1, IACR0, and IACR1) can each specify separate blocks of memory as accessible with restrictions. As shown in Figure 2, there are four access control registers: IACR0 and IACR1 for instruction accesses and DACR0 and DACR1 for data accesses. These registers allow portions of the address space to be accessed with cachability, user/supervisor space and read/write controls. The eight most significant bits of the address are used to define the area of memory controlled. The access control feature allows addresses that match the nACx registers to be cache inhibited and writes to be serialized, which is useful for marking I/O space as noncachable. The nACx register controls the user-programmable access (UPAn) pins.

The ACU status register (ACUSR) contains access control status information resulting from a test of the access control registers.

DATA TYPES AND ADDRESSING MODES

The MC68EC040 supports the basic data types listed in Table 1. In addition, the instruction set supports operations on other data types such as memory addresses.

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 MC68EC040 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 MC68EC040 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 Addressing	
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 Postindexed	([bd,An],Xn,od)
Memory Indirect Preindexed	([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 Postindexed	([bd,PC],Xn,od)
PC Memory Indirect Preindexed	([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 MC68EC040 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 of Table 2.

The MC68EC040 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 MC68EC040 integer and instruction set, refer to M68000PM/AD, *M68000 Programmer's Reference Manual,* for instructions listed under MC68040 or the MC68EC040. For detailed information on the MC68EC040 ACU instruction set refer to MC68EC040UM/AD, *MC68EC040 User's Manual.*

INSTRUCTION AND DATA CACHES

Studies have shown that typical embedded controller 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 MC68EC040 takes further advantage of cache technology with its two, independent, on-chip, caches, one for instructions and one for data. The caches reduce the processor's external bus activity and increase central processor unit (CPU) throughput by lowering the effective memory access time. For a typical system design, the large caches of the MC68EC040 yield a 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 MC68EC040 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 or the external peripheral access. The parallelism inherent in the MC68EC040 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.

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 Codes
MOVE SR	Move to 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	OR Immediate
PACK	Pack BCD
PEA	Push Effective Address
**PTEST	Test Address in ACU Register
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
UNLK	Unlink
UNPK	Unpack BCD

*MC68EC040 additions or alterations to the MC68030 and MC68EC030 instruction sets.
**MC68EC040 alterations to the MC68030, MC68EC030, and MC68040 instruction sets.

CACHE ORGANIZATION

The four-way set-associative instruction and data caches have 64 sets of four 16-byte lines for total cache storage of 4K bytes each. As shown in Figure 3, each 16-byte line contains an address tag and state information. The cache 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 blocks.

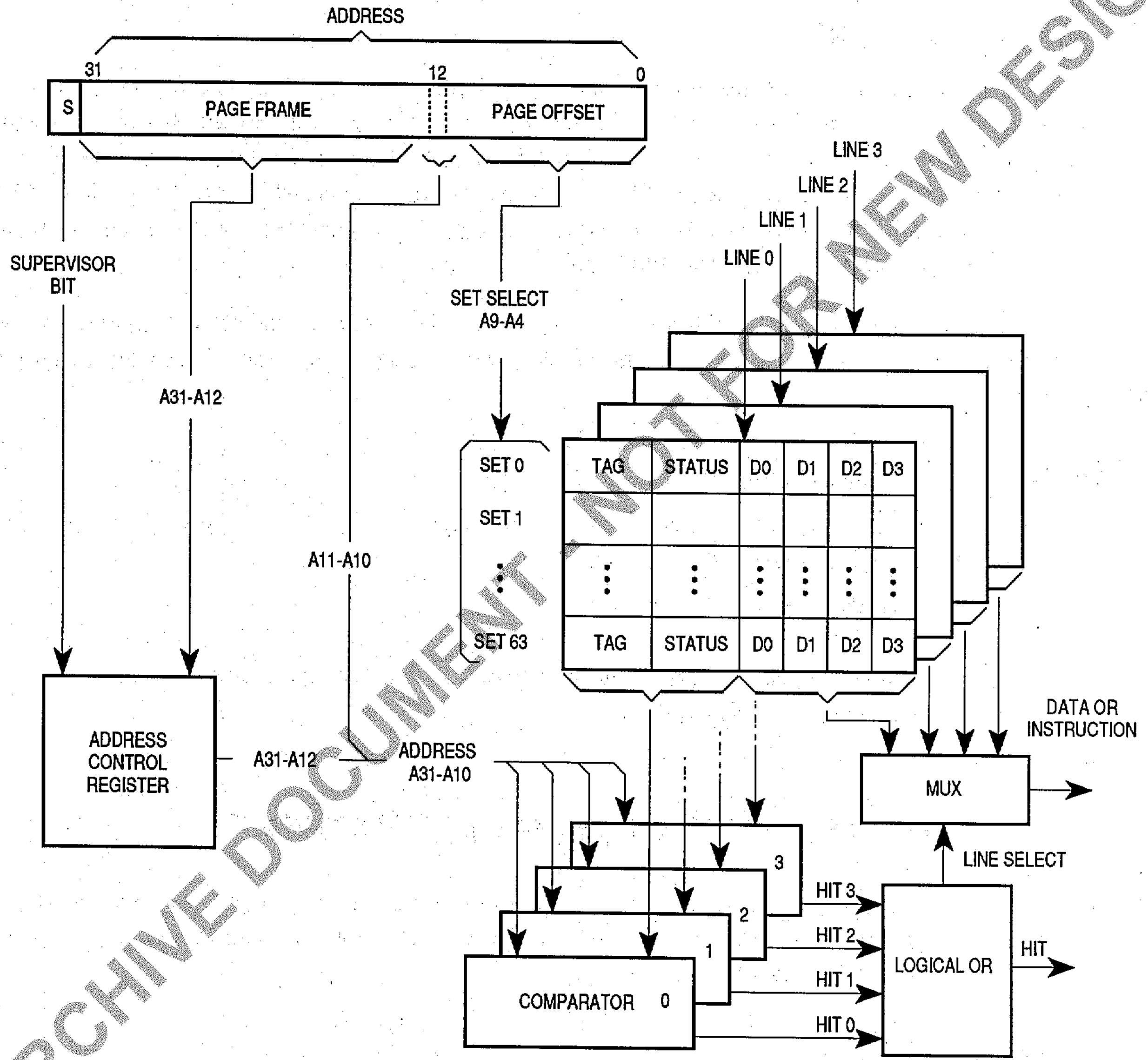


Figure 3. Internal Cache Organization

If the tag matches and the entry is valid, then the cache has a hit. If the cache hits and the access is a read, the appropriate data 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 data entry in the cache.

When a data cache miss occurs and a previously valid cache line is replaced by 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 memory is controlled by two bits in the data access control register. Cachable blocks may be either writethrough or copyback, with no write-allocate for misses to writethrough blocks. Noncachable blocks may also be specified as noncachable I/O, forcing accesses to these blocks to occur in order of instruction execution.

CACHE COHERENCY

The MC68EC040 has the ability to snoop the external bus during accesses by other bus masters to maintain coherency between the MC68EC040 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 unit and the bus snooper circuit both have access to the on-chip caches, the snooper has priority over the execution units to allow the snooper to immediately resolve coherency discrepancies.

CACHE INSTRUCTIONS

The MC68EC040 supports the following instructions for cache maintenance. Both instructions may selectively operate on the data and/or instruction cache.

CINV Invalidates a single line or the entire cache.

CPUSH Pushes selected dirty data cache lines,or all of the cache to memory, then invalidates all selected lines.

OPERAND TRANSFER MECHANISMS

The MC68EC040 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 MC68EC040 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 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 MC68EC040 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 might not be from offset 0; nevertheless, all four long words must be transferred in a wraparound mode. 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 MC68EC040 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 snoop logic determines that one of the on-chip caches has a valid line for this request, the snoop logic may either invalidate or update the line as selected by the snoop control signals.

EXCEPTION PROCESSING

The MC68EC040 provides the same extensions to the exception stacking process as the MC68030 and MC68EC030. 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 can 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 ACU can generate exceptions for access violations.

Exception processing for the MC68EC040 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.

ACCESS CONTROL UNIT

The access control unit (ACU) consist of five registers: The access control unit status register (ACUSR), two instruction access control registers (IACRn), and two data access control registers (DACRn). The ACSR is a 32-bit register that contains the status information returned by execution of the PTEST instruction. The PTEST instruction searches the nACRx to determine status information about the accessibility of a specified address. The status of the ACx registers can be tested and the results can be stored in the ACUSR. The ACUSR is shown in the following Figure 4.

31 12	11	10	9	8	7	6 5	4	3	2	1	0
PHYSICAL ADDRESS	B	G	U1	U0	S	CM	M	0	W	T	R

Figure 4. ACU Status Register

The data ACU registers (DACR0 and DACR1) and instruction ACU registers (IACR0 and IACR1) are 32-bit registers that define blocks of address space. These registers are provided on the MC68EC040 to allow portions of the address space to have particular attributes such as cacheability, supervisor space, and write protection. Address are used with two user-defined page attributes and optional write protection. Each register is used to define an address range from 16 Mbytes to 4 Gbytes with a base address and a mask. All addresses within these ranges are optionally protected against user or supervisor accesses and write accesses. A significant use of these registers would be to provide a separate serialized noncachable address space for peripheral ports. The format of the AC registers is shown in Figure 5.

31						24	23						16
	ADDRESS SPACE							ADDRESS MASK					
E	S FIELD	0	0	0	U1	U0	0	CM	0	0	W	0	0
15	14 13	12	11	10	9	8	7	6 5	4	3	2	1	0

Figure 5. AC Register

SIGNAL DESCRIPTION

Figure 6 illustrates the functional signal groups, and Table 4 describe the signals on the MC68EC040 signal functions. The test signals, $\overline{\text{TRST}}$, TMS, TCK, TDI, and TDO, comply with subset P-1149.1 of the IEEE testability bus standard.

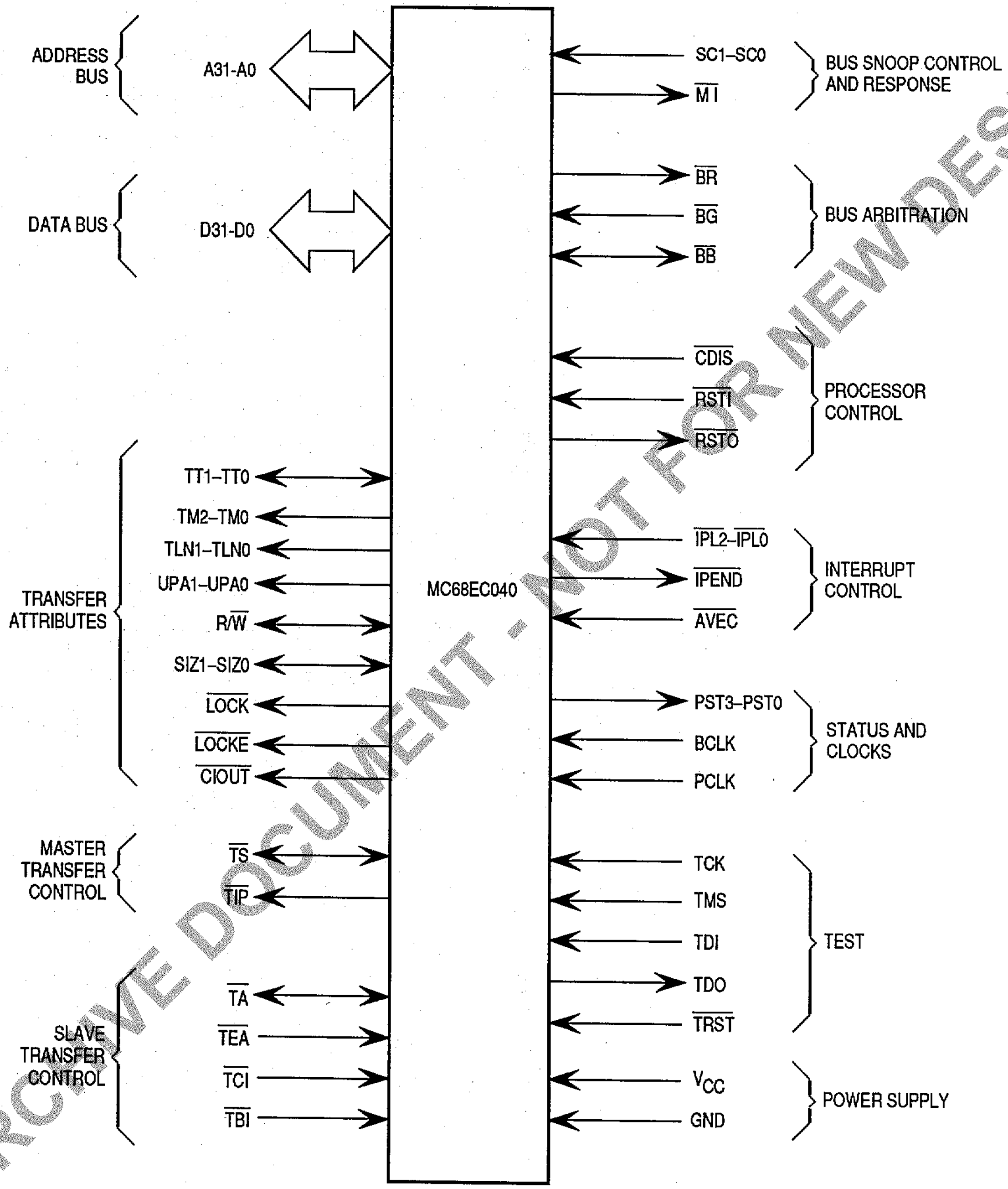


Figure 6. Functional Signal Groups

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/$\overline{\text{W}}$	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 access control registers.
Bus Lock	$\overline{\text{LOCK}}$	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	$\overline{\text{LOCKE}}$	Indicates the current transfer is the last in a locked sequence of transfers.
Cache Inhibit Out	$\overline{\text{CIOUT}}$	Indicates the processor will not cache the current bus transfer.
Transfer Start	$\overline{\text{TS}}$	Indicates the beginning of a bus transfer.
Transfer in Progress	$\overline{\text{TIP}}$	Asserted for the duration of a bus transfer.
Transfer Acknowledge	$\overline{\text{TA}}$	Asserted to acknowledge a bus transfer.
Transfer Error Acknowledge	$\overline{\text{TEA}}$	Indicates an error condition exists for a bus transfer.
Transfer Cache Inhibit	$\overline{\text{TCI}}$	Indicates the current bus transfer should not be cached.
Transfer Burst Inhibit	$\overline{\text{TBI}}$	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	$\overline{\text{MI}}$	Inhibits memory devices from responding to an alternate master access during snooping operations.
Bus Request	$\overline{\text{BR}}$	Asserted by the processor to request bus mastership.
Bus Grant	$\overline{\text{BG}}$	Asserted by an arbiter to grant bus mastership to the processor.
Bus Busy	$\overline{\text{BB}}$	Asserted by the current bus master to indicate it has assumed ownership of the bus.
Cache Disable	$\overline{\text{CDIS}}$	Dynamically disables the internal caches to assist emulator support.
Reset In	$\overline{\text{RSTI}}$	Processor reset.
Reset Out	$\overline{\text{RSTO}}$	Asserted during execution of a RESET instruction to reset external devices.
Interrupt Priority Level	$\overline{\text{IPL2}}$–$\overline{\text{IPL0}}$	Provides an encoded interrupt level to the processor.
Interrupt Pending	$\overline{\text{IPEND}}$	Indicates an interrupt is pending.
Autovector	$\overline{\text{AVEC}}$	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.
Test Data Input	TDI	Serial data input for the TAP.
Test Data Output	TDO	Serial data output for the TAP.
Test Reset	$\overline{\text{TRST}}$	Provides an asynchronous reset of the TAP controller.
Power Supply	V_{CC}	Power supply.
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 in 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 @ 0.5/2.4 V	$\overline{\text{AVEC}}$, BCLK, $\overline{\text{BG}}$, $\overline{\text{CDIS}}$, $\overline{\text{IPLn}}$, PCLK, $\overline{\text{RSTI}}$, SCn $\overline{\text{TBI}}$, TLNn, $\overline{\text{TCI}}$, TCK, $\overline{\text{TEA}}$	I_{in}	20	20	µA
Hi-Z (Off-State) Leakage Current @ 0.5/2.4 V	An, $\overline{\text{BB}}$, $\overline{\text{CIOUT}}$, Dn, $\overline{\text{LOCK}}$, $\overline{\text{LOCKE}}$, R/$\overline{\text{W}}$, SIZn, $\overline{\text{TA}}$, TDO $\overline{\text{TIP}}$, TMn, TLNn, $\overline{\text{TS}}$, TTn, UPAn	I_{TSI}	20	20	µA
Signal Low Input Current V_{IL} = 8 V	TMS, TDI, $\overline{\text{TRST}}$	I_{IL}	–1.1	–0.18	mA
Signal High Input Current V_{IH} = 2.0 V	TMS, TDI, $\overline{\text{TRST}}$	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 (see Note) V_{in} = 0 V, f = 1 MHz		C_{in}	—	20	pF

NOTE: Capacitance is periodically sampled rather than 100% tested.

TBD — To Be Determined

CLOCK AC TIMING SPECIFICATIONS (see Figure 7)

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

NOTE 1: Specification value at maximum frequency of operation.

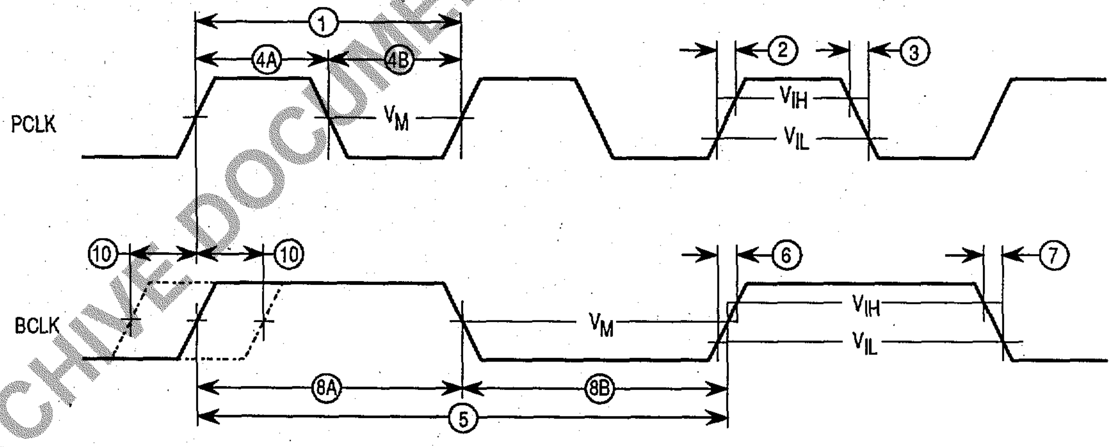


Figure 7. Clock Input Timing Diagram

OUTPUT AC TIMING SPECIFICATIONS

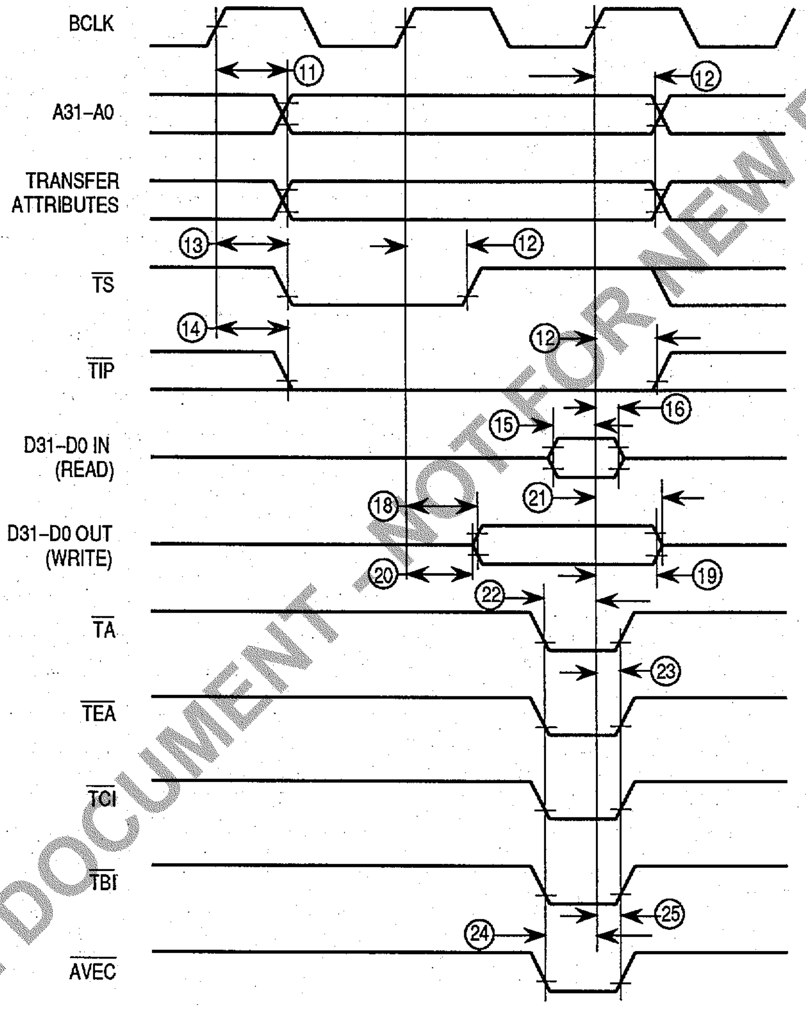
(See Note 1; see Figures 8–12) These output specifications are for only 25 MHz; they must be scaled for lower operating frequencies. Refer to MC68040DH/AD, *MC68040 Designer's Handbook* for further information.

Num	Characteristic	25 MHz[1]		Unit
		Min	Max	
11	BCLK to Address $\overline{\text{CIOUT}}$, $\overline{\text{LOCK}}$, $\overline{\text{LOCKE}}$, R/$\overline{\text{W}}$, SIZn, TLNn,TMn, TTn, UPAn Valid	9	30	ns
12	BCLK to Output Invalid (Output Hold)	9	—	ns
13	TCLK to $\overline{\text{TS}}$ Valid	9	30	ns
14	BCLK to $\overline{\text{TIP}}$ Valid	9	30	ns
18	BCLK to Data-Out Valid	9	32	ns
19	BCLK to Data-Out Invalid (Output Hold)	9	—	ns
20	BCLK to Output Low Impedance	9	—	ns
21	BCLK to Data-Out High Impedance	9	20	ns
38	BCLK to Address, $\overline{\text{CIOUT}}$, $\overline{\text{LOCK}}$, $\overline{\text{LOCKE}}$, R/$\overline{\text{W}}$, SIZn, $\overline{\text{TS}}$, TLNn, TMn, TTn, UPAn High Impedance	9	18	ns
39	BCLK to $\overline{\text{BB}}$, $\overline{\text{TA}}$, $\overline{\text{TIP}}$ High Impedance	19	28	ns
40	BCLK to $\overline{\text{BR}}$, $\overline{\text{BB}}$ Valid	9	30	ns
43	BCLK to $\overline{\text{MI}}$ Valid	9	30	ns
48	BCLK to $\overline{\text{TA}}$ Valid	9	30	ns
50	BCLK to $\overline{\text{IPEND}}$, PSTn, $\overline{\text{RSTO}}$ Valid	9	30	ns

NOTE 1: Output timing is specified for a valid signal measured at the pin. Buffer timing is specified driving an unterminated 30-ΩV transmission line with a length characterized by a 2.5-ns one-way propagation delay. Buffer output impedance is typically 30 ΩV; 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.

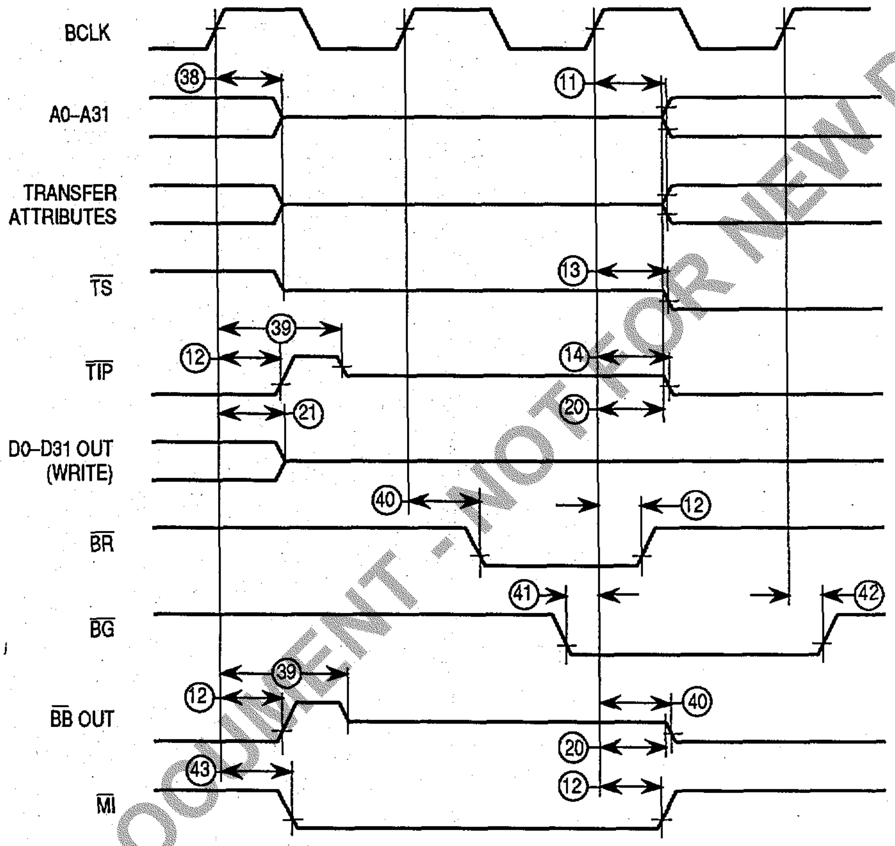
INPUT AC TIMING SPECIFICATIONS (see Figures 8–12)

Num	Characteristic	25 MHz		Unit
		Min	Max	
15	Data-In Valid to BCLK (Setup)	5	—	ns
16	BCLK to Data-In Invalid (Hold)	4	—	ns
17	BCLK to Data-In High Impedance (Read Followed by Write)	—	49	ns
22a	$\overline{\text{TA}}$ Valid to BCLK (Setup)	10	—	ns
22b	$\overline{\text{TEA}}$ Valid to BCLK (Setup)	10	—	ns
22c	$\overline{\text{TCI}}$ Valid to BCLK (Setup)	10	—	ns
22d	$\overline{\text{TBI}}$ Valid to BCLK (Setup)	11	—	ns
23	BCLK to $\overline{\text{TA}}$, $\overline{\text{TEA}}$, $\overline{\text{TCI}}$, $\overline{\text{TBI}}$ Invalid (Hold)	2	—	ns
24	$\overline{\text{AVEC}}$ Valid to BCLK (Setup)	5	—	ns
25	BCLK to $\overline{\text{AVEC}}$ Invalid (Hold)	2	—	ns
41a	$\overline{\text{BB}}$ Valid to BCLK (Setup)	7	—	ns
41b	$\overline{\text{BG}}$ Valid to BCLK (Setup)	8	—	ns
41c	$\overline{\text{CDIS}}$ Valid to BCLK (Setup)	10	—	ns
41d	$\overline{\text{IPLn}}$ Valid to BCLK (Setup)	4	—	ns
42	BCLK to $\overline{\text{BB}}$, $\overline{\text{BG}}$, $\overline{\text{CDIS}}$, $\overline{\text{IPLn}}$ Invalid (Hold)	2	—	ns
44a	Address Valid to BCLK (Setup)	8	—	ns
44b	SIZn Valid to BCLK (Setup)	12	—	ns
44c	TTn Valid to BCLK (Setup)	6	—	ns
44d	R/$\overline{\text{W}}$ Valid to BCLK (Setup)	6	—	ns
44e	SCn Valid to BCLK (Setup)	10	—	ns
45	BCLK to Address SIZn, TTn, R/$\overline{\text{W}}$, SCn Invalid (Hold)	2	—	ns
46	$\overline{\text{TS}}$ Valid to BCLK (Setup)	5	—	ns
47	BCLK to $\overline{\text{TS}}$ Invalid (Hold)	2	—	ns
49	BCLK to $\overline{\text{BB}}$ High Impedance (MC68EC040 Assumes Bus Mastership)	—	9	ns
51	$\overline{\text{RSTI}}$ Valid to BCLK	5	—	ns
52	BCLK to $\overline{\text{RSTI}}$ Invalid	2	—	ns
53	Mode Select Setup to $\overline{\text{RSTI}}$ Negated	20	—	ns
54	$\overline{\text{RSTI}}$ Negated to Mode Selects Invalid	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/$\overline{\text{W}}$, $\overline{\text{LOCK}}$, $\overline{\text{LOCKE}}$, $\overline{\text{CIOUT}}$

Figure 9. Bus Arbitration Timing

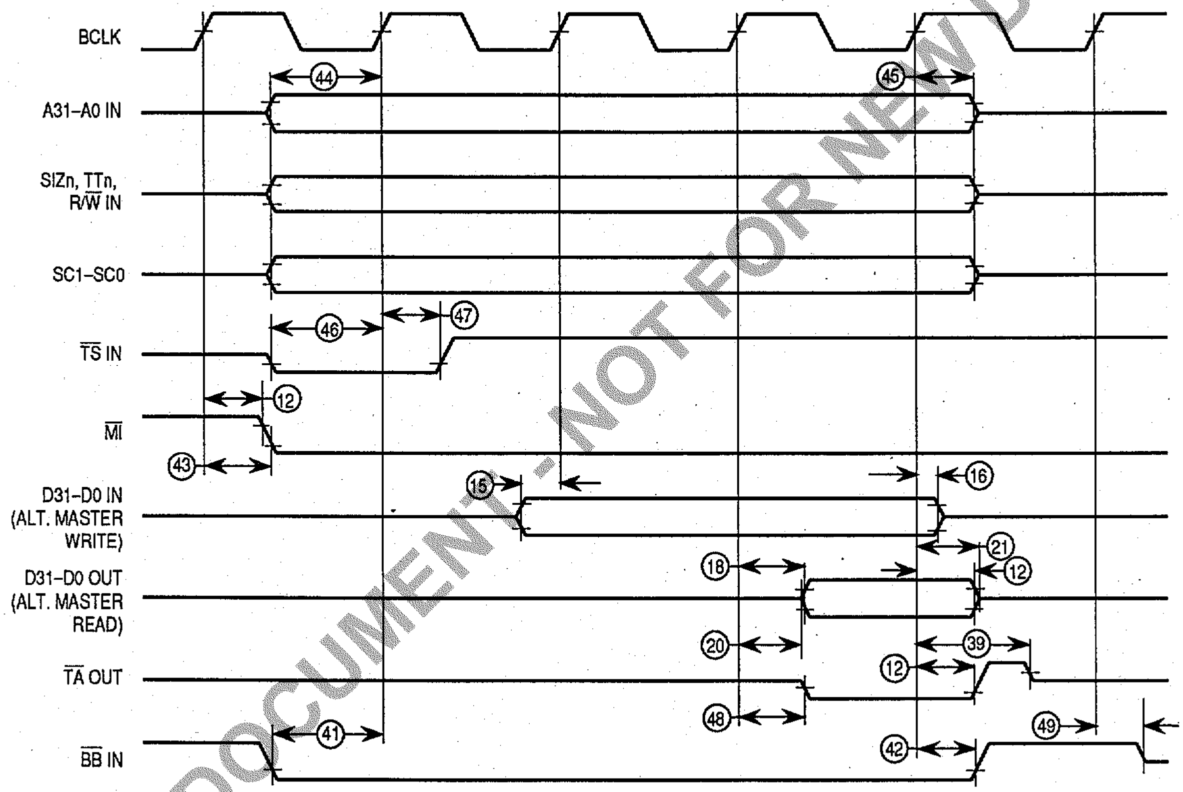


Figure 10. Snoop Hit Timing

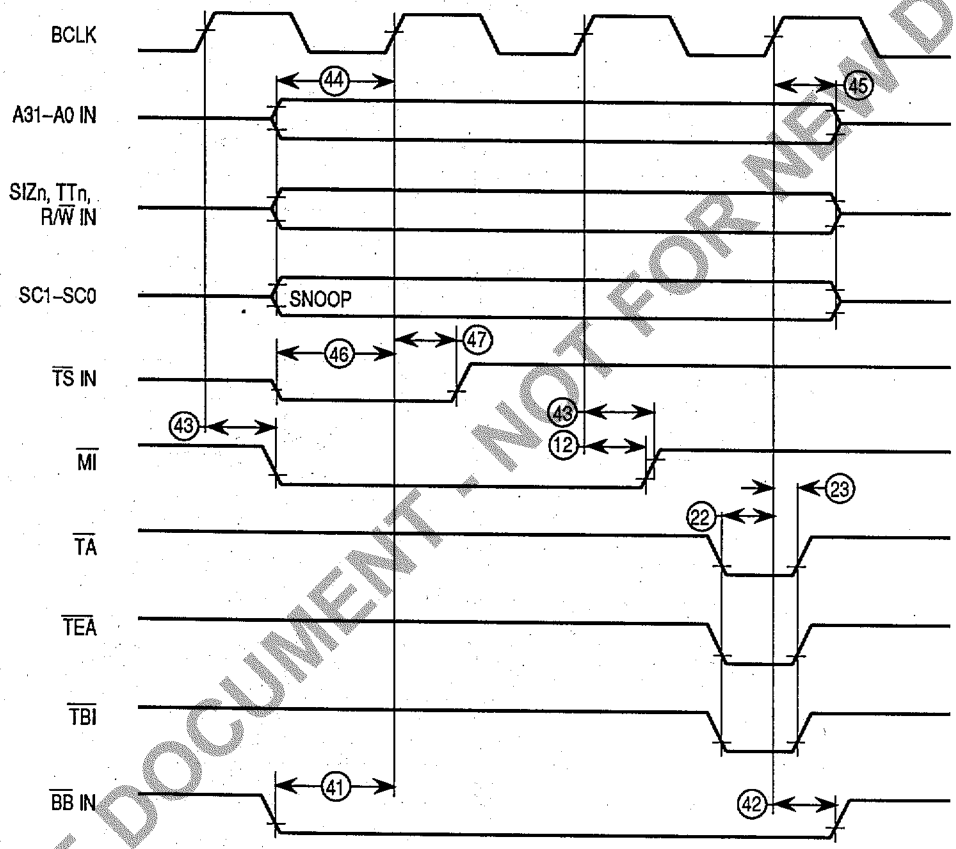


Figure 11. Snoop Miss Timing

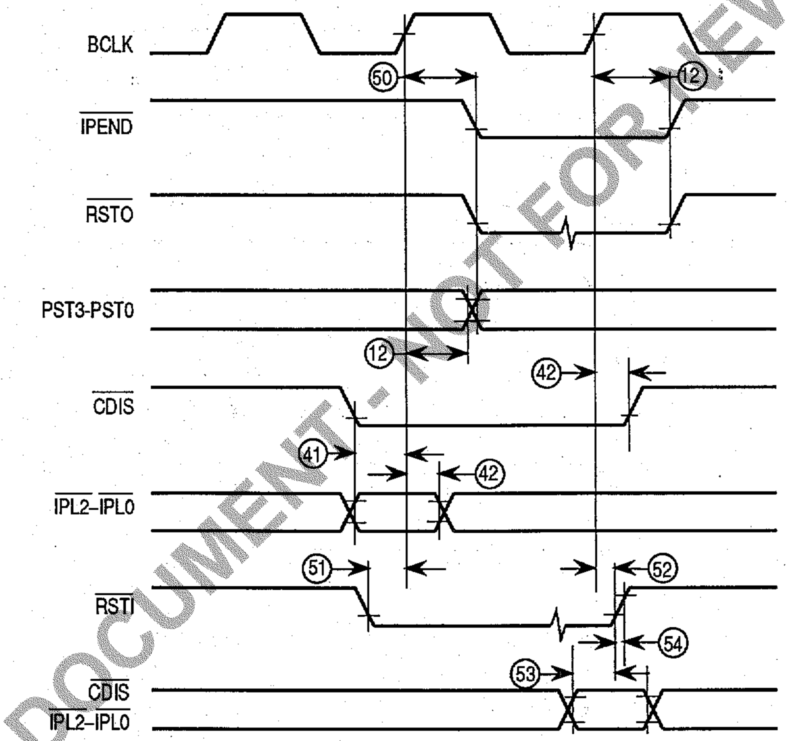
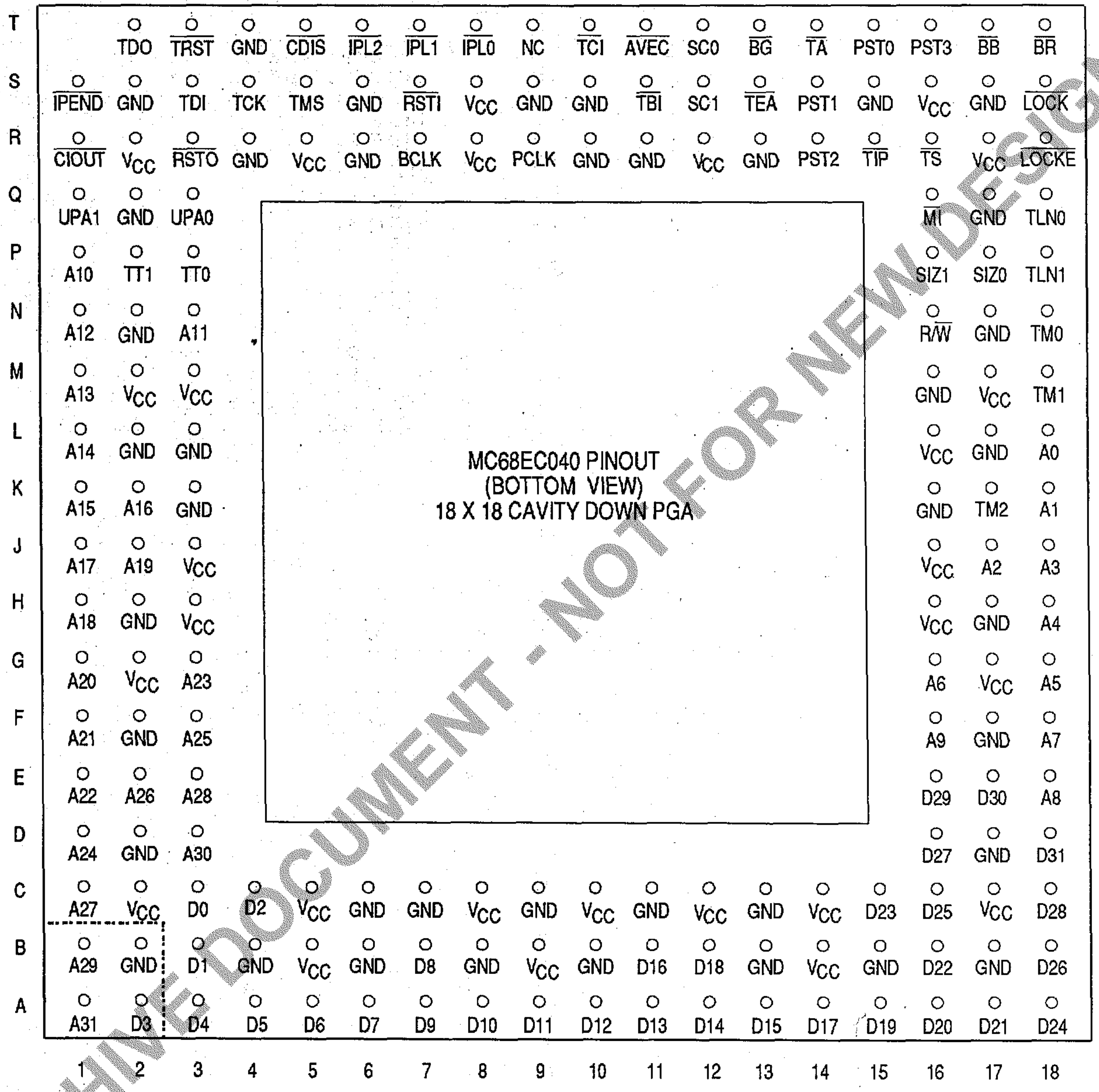


Figure 12. Other Signal Timing

MECHANICAL DATA

PIN ASSIGNMENT



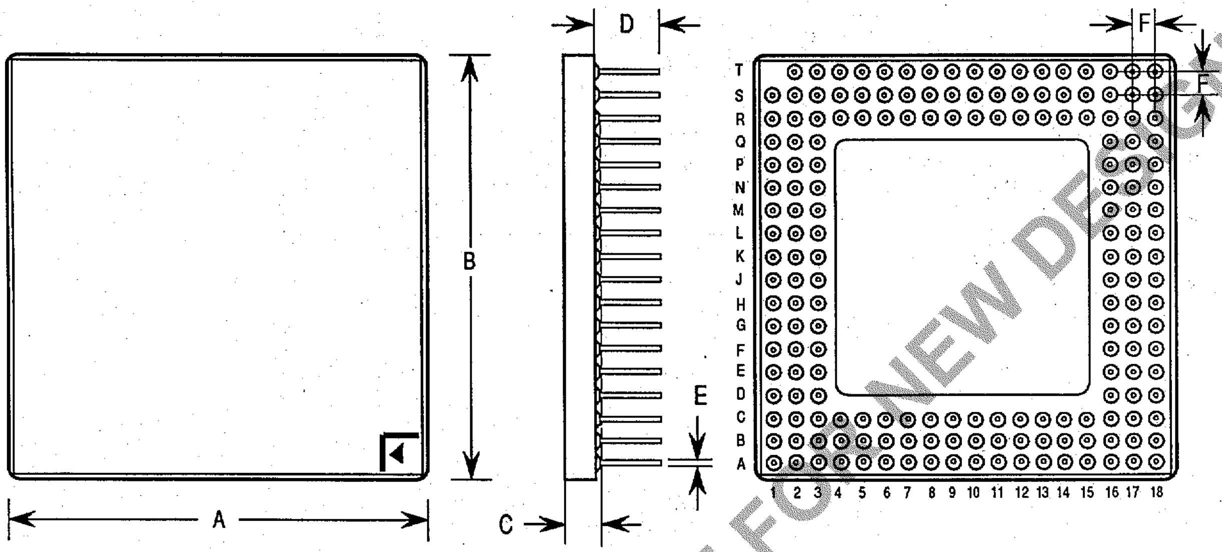
MC68EC040 PINOUT
(BOTTOM VIEW)
18 X 18 CAVITY DOWN PGA

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
T		TDO	TRST	GND	CDIS	IPL2	IPL1	IPL0	NC	TCI	AVEC	SC0	BG	TA	PST0	PST3	BB	BR
S	IPEND	GND	TDI	TCK	TMS	GND	RSTI	V_{CC}	GND	GND	TBI	SC1	TEA	PST1	GND	V_{CC}	GND	LOCK
R	CIOUT	V_{CC}	RSTO	GND	V_{CC}	GND	BCLK	V_{CC}	PCLK	GND	GND	V_{CC}	GND	PST2	TIP	TS	V_{CC}	LOCKE
Q	UPA1	GND	UPA0													MI	GND	TLN0
P	A10	TT1	TT0													SIZ1	SIZ0	TLN1
N	A12	GND	A11													R/W	GND	TM0
M	A13	V_{CC}	V_{CC}													GND	V_{CC}	TM1
L	A14	GND	GND													V_{CC}	GND	A0
K	A15	A16	GND													GND	TM2	A1
J	A17	A19	V_{CC}													V_{CC}	A2	A3
H	A18	GND	V_{CC}													V_{CC}	GND	A4
G	A20	V_{CC}	A23													A6	V_{CC}	A5
F	A21	GND	A25													A9	GND	A7
E	A22	A26	A28													D29	D30	A8
D	A24	GND	A30													D27	GND	D31
C	A27	V_{CC}	D0	D2	V_{CC}	GND	GND	V_{CC}	GND	V_{CC}	GND	V_{CC}	GND	V_{CC}	D23	D25	V_{CC}	D28
B	A29	GND	D1	GND	V_{CC}	GND	D8	GND	V_{CC}	GND	D16	D18	GND	V_{CC}	GND	D22	GND	D26
A	A31	D3	D4	D5	D6	D7	D9	D10	D11	D12	D13	D14	D15	D17	D19	D20	D21	D24

	GND	V_{CC}
PLL	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

Case To Be Determined



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	

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

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.

MOTOROLA

A30112-0 PRINTED IN USA 4/91 IMPERIAL LITHO 78567 18,000 HPU YGABAA