Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	68040
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	20MHz
Co-Processors/DSP	-
RAM Controllers	-
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	179-BEPGA
Supplier Device Package	179-PGA (47.24x47.24)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68ec040rc20a

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS (Continued)

more common case of the branch taken, and both execution paths of the branch are fetched and decoded to minimize refilling of the instruction pipeline.

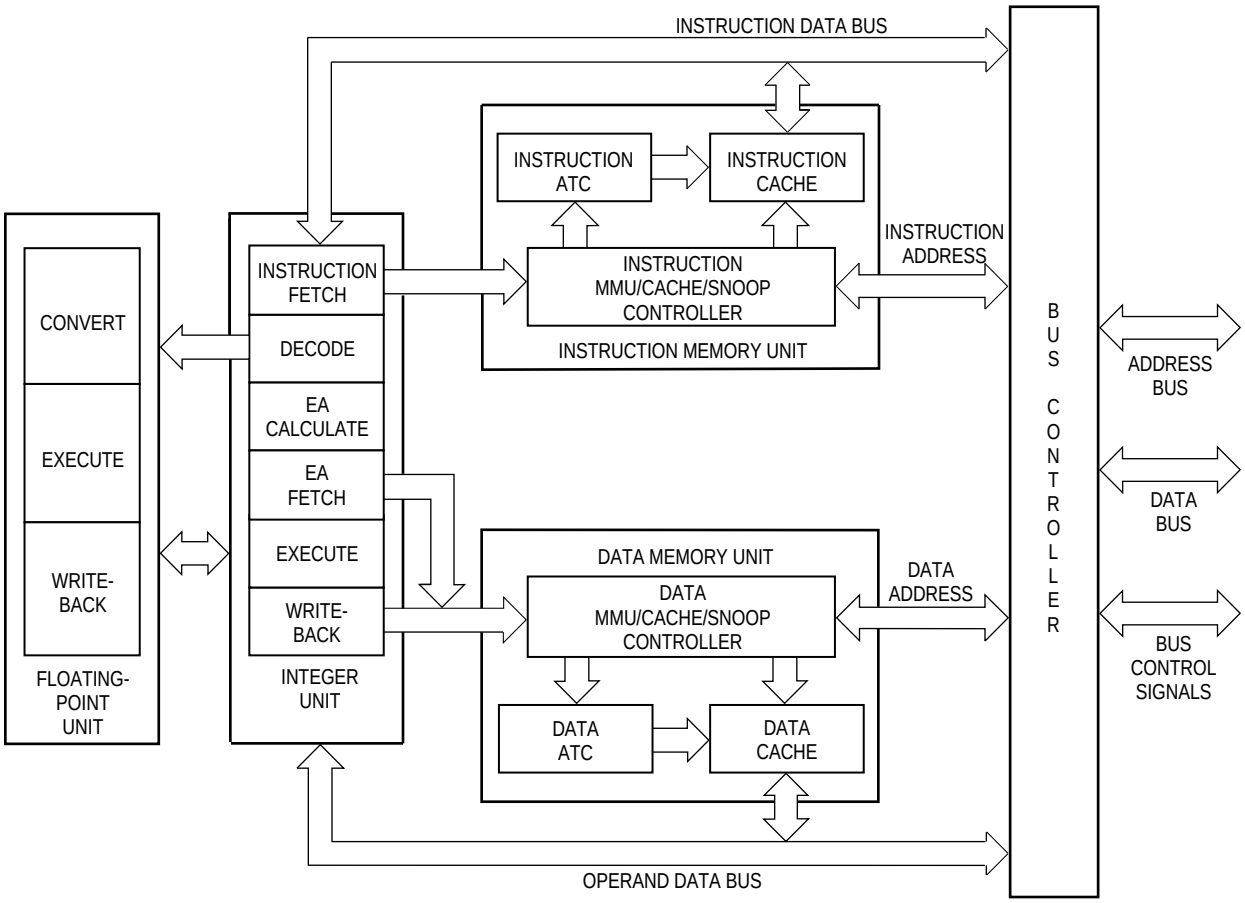


Figure 1-1. Block Diagram

To improve memory management, the M68040 includes separate, independent paged MMUs for instruction and data accesses. Each MMU stores recently used address mappings in separate 64-entry address translation caches (ATCs). Each MMU also has two transparent translation registers that define a one-to-one mapping for address space segments ranging in size from 16 Mbytes to 4 Gbytes each.

Two memory units independently interface with the IU and FPU. Each unit consists of an MMU, an ATC, a main cache, and a snoop controller. The MMUs perform memory management on a demand-page basis. By translating logical-to-physical addresses using translation tables stored in memory, the MMUs support virtual memory systems. Each MMU stores recently used address mappings in an ATC, reducing the average translation time.

Separate on-chip instruction and data caches operate independently and are accessed in parallel with address translation. The caches improve the overall performance of the system by reducing the number of bus transfers required by the processor to fetch information from memory and by increasing the bus bandwidth available for alternate bus

Table 1-3. Notational Conventions (Concluded)

Register Codes	
*	General Case.
C	Carry Bit in CCR
cc	Condition Codes from CCR
FC	Function Code
N	Negative Bit in CCR
U	Undefined, Reserved for Motorola Use.
V	Overflow Bit in CCR
X	Extend Bit in CCR
Z	Zero Bit in CCR
—	Not Affected or Applicable.
Stack Pointers	
ISP	Supervisor/Interrupt Stack Pointer
MSP	Supervisor/Master Stack Pointer
SP	Active Stack Pointer
SSP	Supervisor (Master or Interrupt) Stack Pointer
USP	User Stack Pointer
Miscellaneous	
<ea>	Effective Address
<label>	Assemble Program Label
<list>	List of registers, for example D3–D0.
LB	Lower Bound
m	Bit m of an Operand
m–n	Bits m through n of Operand
UB	Upper Bound

1.10 INSTRUCTION SET OVERVIEW

The instruction set is tailored to support high-level languages and is optimized for those instructions most commonly executed. The floating-point instructions for the M68040 are a commonly used subset of the MC68881/MC68882 instruction set with new arithmetic instructions to explicitly select single- or double-precision rounding. The remaining unimplemented instructions are less frequently used and are efficiently emulated in the M68040FPSP, maintaining compatibility with the MC68881/MC68882 floating-point coprocessors. The M68040 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. Table 1-4 provides an alphabetized listing of the M68040 instruction set's opcode, operation, and syntax. Refer to Table 1-3 for notations used in Table 1-4. The left operand in the syntax is always the source operand, and the right operand is the destination operand. Refer to M68000PM/AD, *M68000 Family Programmer's Reference Manual,* for details on instructions used by the M68040.

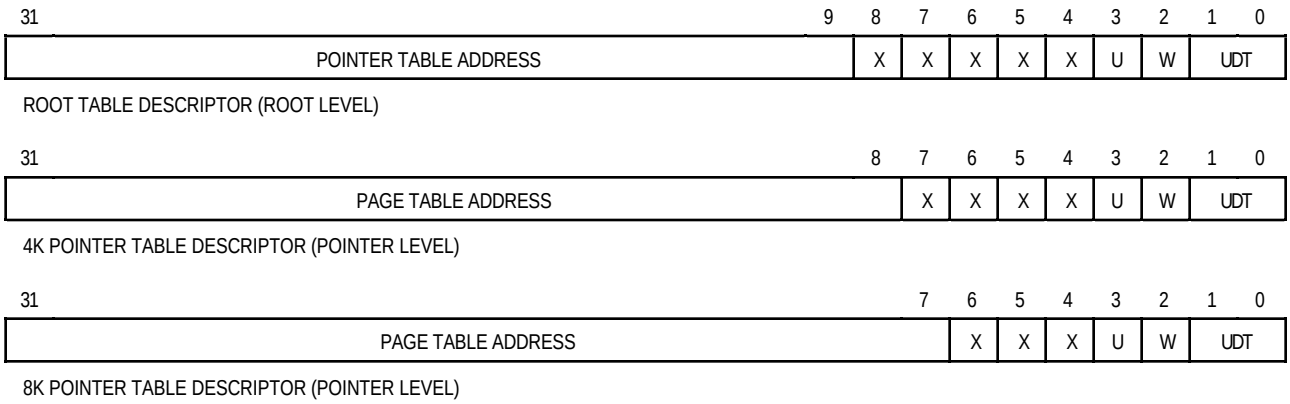


Figure 3-11. Table Descriptor Formats

3.2.2.2 PAGE DESCRIPTORS. Figure 3-12 illustrates the page descriptors for both 4-Kbyte and 8-Kbyte page sizes. Refer to **Section 4 Instruction and Data Caches** for details concerning caching page descriptors.

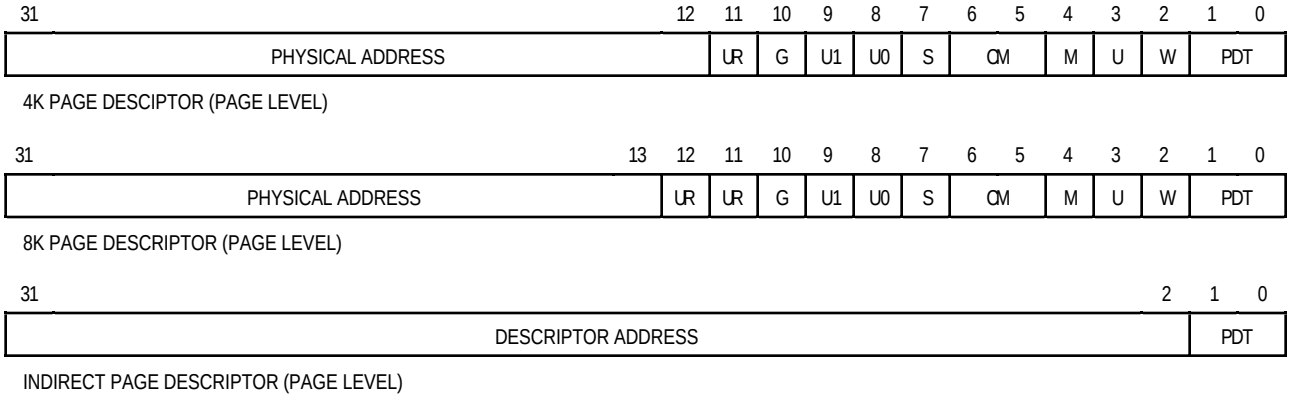


Figure 3-12. Page Descriptor Formats

3.2.2.3 DESCRIPTOR FIELD DEFINITIONS. The field definitions for the table- and page-level descriptors are listed in alphabetical order:

CM—Cache Mode

This field selects the cache mode and accesses serialization as follows:

00 = Cachable, Write-through
01 = Cachable, Copyback
10 = Noncachable, Serialized
11 = Noncachable

Section 4 Instruction and Data Caches provides detailed information on caching modes, and **Section 7 Bus Operation** provides information on serialization.

3.3 ADDRESS TRANSLATION CACHES

The ATCs in the MMUs are four-way set-associative caches that each store 64 logical-to-physical address translations and associated page information similar in form to the corresponding page descriptors in memory. The purpose of the ATC is to provide a fast mechanism for address translation by avoiding the overhead associated with a table search of the logical-to-physical mapping of recently used logical addresses. Figure 3-20 illustrates the organization of the ATC.

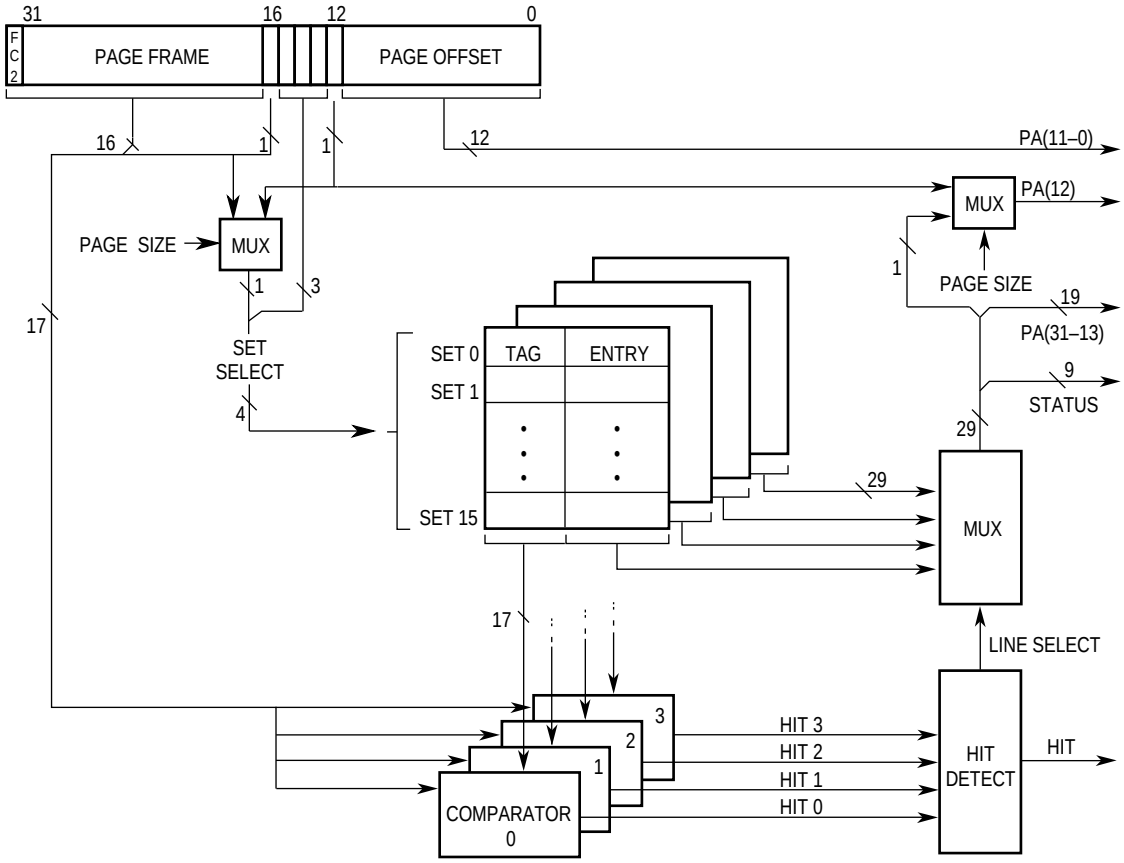


Figure 3-20. ATC Organization

Clock 2 (C2)

During the first half of the first clock cycle after C1, the processor negates $\overline{TS}$. The selected device uses R/$\overline{W}$, SIZ1, SIZ0, A1, and A0 to place its information on the data bus. With the exception of R/$\overline{W}$, these signals also select any or all of the bytes (D24–D31, D16–D23, D15–D8, and D7–D0). Concurrently, the selected device asserts $\overline{TA}$. At the end of the first clock cycle after C1, the processor samples the level of $\overline{TA}$ and latches the current value on the data bus. If $\overline{TA}$ is asserted, the read transfer terminates, and the latched data is passed to the appropriate memory unit. If $\overline{TA}$ is not recognized asserted, the processor ignores the data and appends a wait state instead of terminating the transfer. The processor continues to sample $\overline{TA}$ on successive rising edges of BCLK until $\overline{TA}$ is recognized as asserted. The latched data is then passed to the appropriate memory unit. If more than one read cycle is required to read in the operand(s), C1 and C2 are repeated accordingly.

When the processor recognizes $\overline{TA}$ at the end of the last read transfer for the locked bus cycle, it negates $\overline{TIP}$ during the first half of the next clock.

Clock Idle (CI)

The processor does not assert any new control signals during the idle clock states, but it may begin the modify portion of the cycle at this time. The R/$\overline{W}$ signal remains in the read mode until C3 to prevent bus conflicts with the preceding read portion of the cycle; the data bus is not driven until C4.

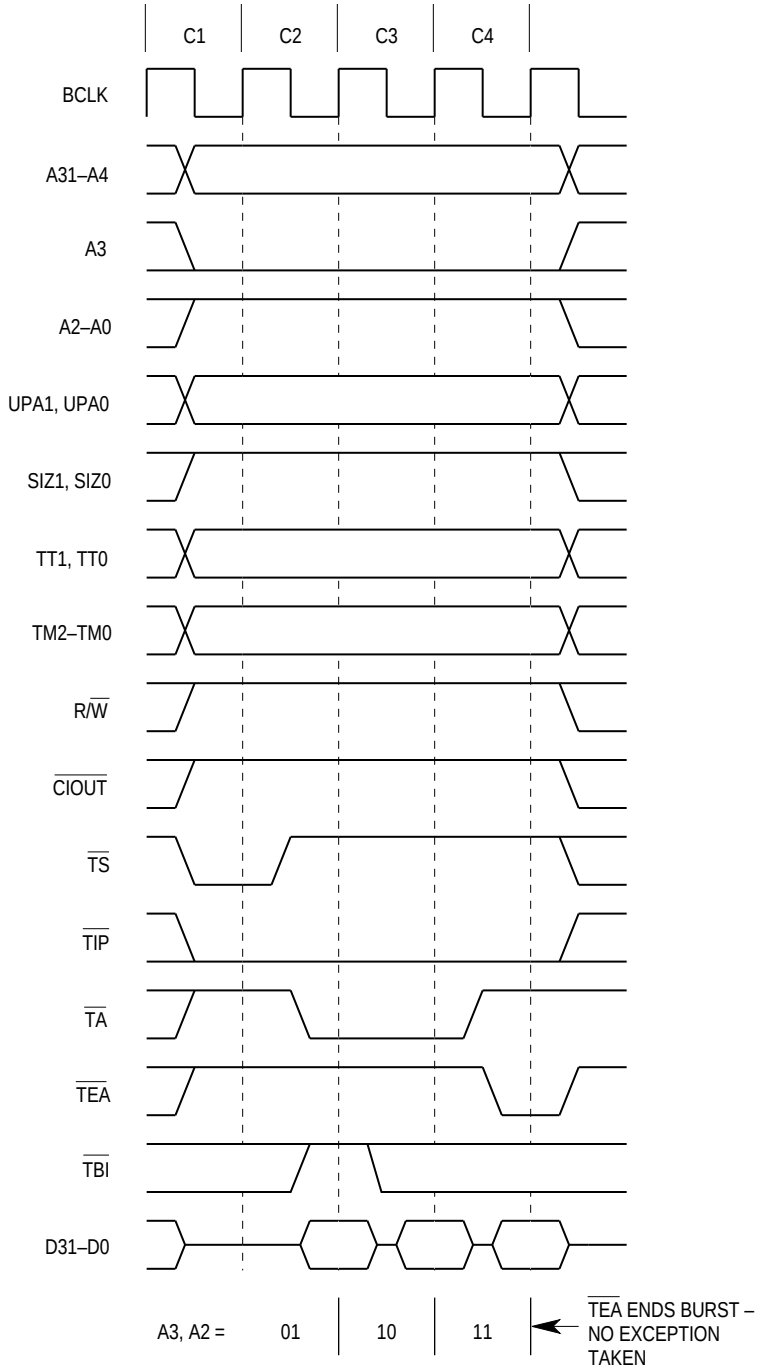
Clock 3 (C3)

During the first half of C3, the processor places valid values on the address bus and transfer attributes and drives R/$\overline{W}$ low for a write cycle. The processor asserts $\overline{TS}$ to indicate the beginning of a bus cycle. The $\overline{TIP}$ signal is also asserted at this time to indicate that a bus cycle is active.

$\overline{LOCKE}$ is asserted during C3 for the last write transfer of the locked sequence. If multiple write transfers are required for misaligned operands or multiple operands, $\overline{LOCKE}$ is asserted only for the final write transfer. The external arbiter can use this indication to distinguish between two back-to-back locked bus cycles and allow arbitration between them.

Clock 4 (C4)

During the first half of C4, the processor negates $\overline{TS}$ and drives the appropriate bytes of the data bus with the data to be written. All other bytes are driven with undefined values. The selected device uses R/$\overline{W}$, SIZ1, SIZ0, A1, and A0 to latch the information on the data bus. Any or all of the bytes (D31–D24, D23–D16, D15–D8, and D7–D0) are selected by SIZ1, SIZ0, A1, and A0. Concurrently, the selected device asserts $\overline{TA}$. At the end of C4, the processor samples the level of $\overline{TA}$; if $\overline{TA}$ is asserted, the bus cycle terminates. If $\overline{TA}$ is not recognized asserted at the end of C4, the processor appends a wait state instead of terminating the transfer. The processor continues to sample the $\overline{TA}$ signal on successive rising edges of BCLK until it is recognized asserted.



NOTE: The selected device increments the value on A3 and A2.

Figure 7-27. Line Read Access Terminated with TEA Timing

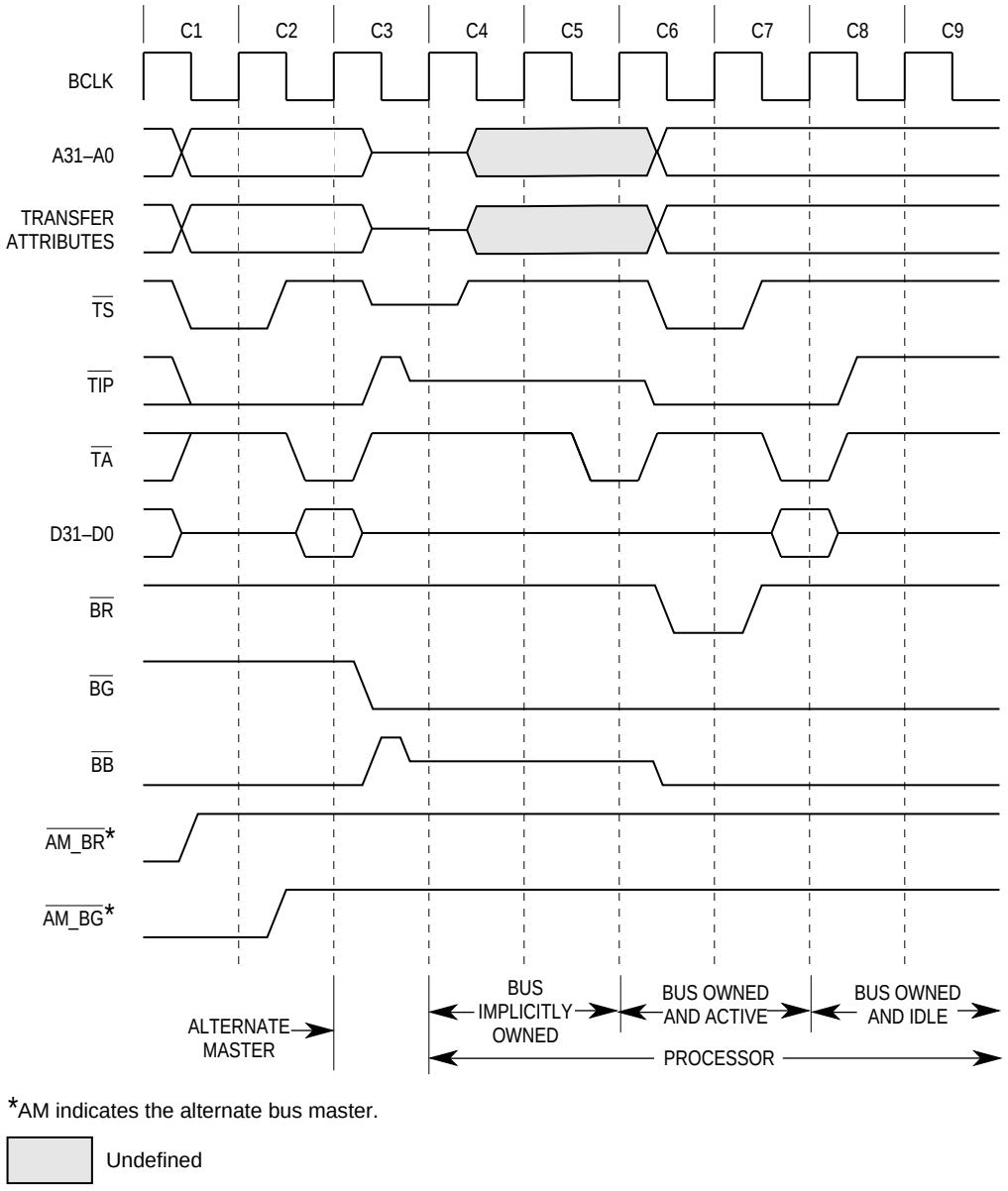


Figure 7-34. Implicit Bus Ownership Arbitration Timing

7.8.2 Bus Arbitration Examples

The following paragraphs illustrate the behavior of the M68040 bus arbitration scheme and provide examples of how an external bus arbiter can be designed to keep the integrity of locked bus operations. The examples include the previously mentioned indeterminate and disregard request conditions.

7.8.2.1 DUAL M68040 FAIRNESS ARBITRATION. The following state diagram illustrates a fairness algorithm using two MC68040s and assigning the least priority to the processor that owns the bus. If both processors keep their respective $\overline{BR}$ signals asserted, bus ownership alternates between the two processors so that each processor can run at least one bus cycle during its tenure. Each processor is allowed to own the bus without relinquishing it to maintain the integrity of locked transfers. This example also illustrates

7.11 SPECIAL MODES OF OPERATION

The MC68LC040 and MC68EC040 do not support the following three modes of operation, which for the M68040 are selectively enabled during processor reset and remain in effect until the next processor reset. Refer to **Appendix A MC68LC040** and **Appendix B MC68EC040** for differences in the special modes of operation for the MC68LC040 and MC68EC040.

7.11.1 Output Buffer Impedance Selection

All output drivers in the M68040 can be configured to operate in either a large buffer mode (low-impedance driver) or small buffer mode (high-impedance driver). Large buffers have a nominal output impedance of 6 Ω for both high and low drive, resulting in minimum output delays. Signal traces driven by large buffers usually require transmission line effects to be considered in their design, including the use of signal termination. Small buffers have a nominal impedance of 25 Ω for high and low drive, resulting in longer output delays and less critical board-design requirements. Refer to **Section 11 MC68040 Electrical and Thermal Characteristics** for further information on electrical specifications, buffer characteristics, and transmission line design examples. The output drivers are configured in three groups. Each group of signals is configured depending on the corresponding $\overline{\text{IPLx}}$ signal level during processor reset (see Table 5-5).

7.11.2 Multiplexed Bus Mode

The multiplexed bus mode changes the timing of the three-state control logic for the address and data buses to support generation of a multiplexed address/data bus. When the M68040 is operating in this mode, the address and data bus signals can be hardwired together to form a single 32-bit bus, with address and data information time-multiplexed on the bus. This configuration minimizes the number of pins required to interface to peripheral devices without requiring additional discrete multiplexing logic. This mode is enabled during a processor reset by a logic zero on the $\overline{\text{CDIS}}$ signal.

Figure 7-46 illustrates a line write with multiplexed bus mode enabled. The address bus drivers are enabled during C1 and disabled during C2. Later in C2, the data bus drivers are enabled to drive the data bus with the data to be written. The address bus is only driven for the BCLK rising edge at the start of each bus cycle.

Table 9-4. Double-Precision Real Format Summary

Data Format 63 62 52 51 0 s | e | f	
Field Size (in Bits)	
Sign (s)	1
Biased Exponent (e)	11
Fraction (f)	52
Total	64
Interpretation of Sign	
Positive Fraction	s = 0
Negative Fraction	s = 1
Normalized Numbers	
Bias of Biased Exponent	+1023 ($3FF)
Range of Biased Exponent	0 < e < 2047 ($7FF)
Range of Fraction	Zero or Nonzero
Fraction	1.f
Relation to Representation of Real Numbers	$(-1)^s \times 2^{e-1023} \times 1.f$
Denormalized Numbers	
Biased Exponent Format Minimum	0 ($000)
Bias of Biased Exponent	+1022 ($3FE)
Range of Fraction	Nonzero
Fraction	0.f
Relation to Representation of Real Numbers	$(-1)^s \times 2^{-1022} \times 0.f$
Signed Zeros	
Biased Exponent Format Minimum	0 ($00)
Fraction (Mantissa/Significand)	0.f = 0.0
Signed Infinities	
Biased Exponent Format Maximum	2047 ($7FF)
Fraction	0.f = 0.0
NANs	
Sign	0 or 1
Biased Exponent Format Maximum	255 ($7FF)
Fraction	Nonzero
Representation of Fraction Nonsignaling Signaling Nonzero Bit Pattern Created by User Fraction When Created by FPCP	 1xxxx…xxxx 0xxxx…xxxx xxxxx…xxxx 11111…1111
Approximate Ranges	
Maximum Positive Normalized	1.8×10^{308}
Minimum Positive Normalized	2.2×10^{-308}
Minimum Positive Denormalized	4.9×10^{-324}

precision mode is selected, the exponent value is in the correct range even if it is stored in extended-precision format. If the destination is a memory location, the FPSR PREC bits are ignored. In this case, a number in the extended-precision format is taken from the source floating-point data register, rounded to the destination format precision, and then written to memory.

Depending on the selected rounding mode or destination data format in effect, the location of the least significant bit of the mantissa and the locations of the guard, round, and sticky bits in the 67-bit intermediate result mantissa varies. The guard and round bits are always calculated exactly. The sticky bit is used to create the illusion of an infinitely wide intermediate result. As the arrow illustrates in Figure 9-7, the sticky bit is the logical OR of all the bits in the infinitely precise result to the right of the round bit. During the calculation stage of an arithmetic operation, any non-zero bits generated that are to the right of the round bit set the sticky bit to one. Because of the sticky bit, the rounded intermediate result for all required IEEE arithmetic operations in the RN mode is in error by no more than one-half unit in the last place.

9.4.2 Rounding The Result

Range control is the process of rounding the mantissa of the intermediate result to the specified precision and checking the 16-bit intermediate exponent to ensure that it is within the representable range of the selected rounding-precision format. Range control ensures correct emulation of a device that only supports single- or double-precision arithmetic. If the intermediate result's exponent exceeds the range of the selected precision, the exponent value appropriate for an underflow or overflow is stored as the result in the 16-bit extended-precision format exponent. For example, if the data format and rounding mode is single-precision RM and the result of an arithmetic operation overflows the magnitude of the single-precision format, the largest normalized single-precision value is stored as an extended-precision number in the destination floating-point data register (i.e., an unbiased 15-bit exponent of $00FF and a mantissa of $FFFFFF0000000000). If an infinity is the appropriate result for an underflow or overflow, the infinity value for the destination data format is stored as the result (i.e., an exponent with the maximum value and a mantissa of zero).

Figure 9-8 illustrates the algorithm that is used to round an intermediate result to the selected rounding precision and destination data format. If the destination is a floating-point data register, either the selected rounding precision specified by the FPCR PREC bits or by the instruction itself determines the rounding boundary. For example, FSADD and FDADD specify single- and double-precision rounding regardless of the precision specified in the FPCR PREC bits. If the destination is external memory or an integer data register, the destination data format determines the rounding boundary. If the rounded result of an operation is not exact, then the INEX2 bit is set in the FPSR EXC byte.

Table 9-16. State Frame Field Information (Continued)

FSAVE State Frame Field	Contents
SNAN (For Opclass 011)	
CMDREG1B	FMOVE Instruction Command Word
ETEMP	Unrounded Source Operand from Floating-Point Register, with SNAN bit set.
STAG	Source Operand Tag, indicated NAN.
E1	Always 1
T	Always 1
OPERR (For Opclass 000 and 010)	
CMDREG1B	Exception Instruction Command Word
ETEMP	Source operand is converted to extended precision.
STAG	Source Operand Tag
FPTEMP	Destination operand, if any, is converted to extended precision.
DTAG	Destination operand tag, if any.
E1	Always 1
T	Always 0
OPERR (For Opclass 011)	
CMDREG1B	FMOVE Instruction Command Word
ETEMP	Unrounded Source Operand from Floating-Point Register
STAG	Source Operand Tag
WBTEMP	Contains the rounded integer used to check for erroneous integer overflow.
E1	Always 1
T	Always 1
OVFL (FMOVE to Register, FABS, and FNEG)	
CMDREG1B	Exception Instruction Command Word
FPTEMP	Intermediate result with mantissa rounded to correct precision.
STAG	Source Operand Tag = Normalized
E1	Always 1
T	Always 0
OVFL (FADD, FSUB, FMUL, FDIV, and FSQRT)	
CMDREG3B	Encoded Exception Instruction Command Word
WBTEMP	WBTS, WBTE, and WBTM equal the intermediate result with mantissa rounded to the correct precision.
WBTE15	Bit 15 of the intermediate result's 16-bit exponent = 0 for overflow.
E3	Always 1
T	Either 1 or 0

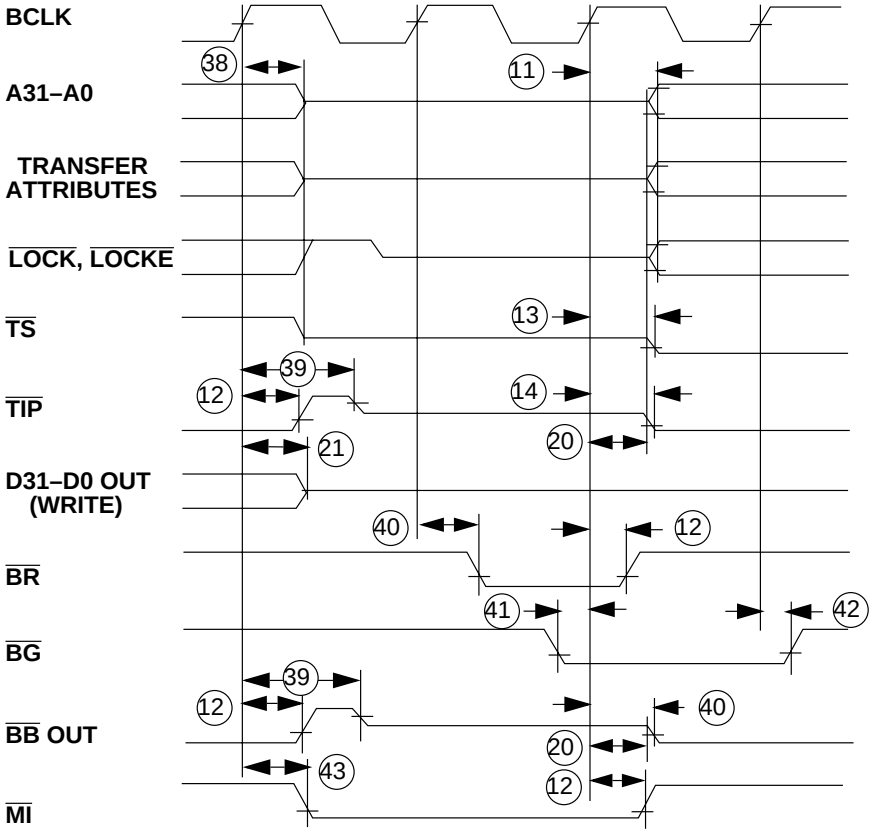
Table 9-16. State Frame Field Information (Concluded)

FSAVE State Frame Field	Contents
INEX (FMOVE to Register, FABS, and FNEG)	
CMDREG1B	Exception Instruction Command Word
FPTEMP	Unrounded, Extended-Precision Intermediate Result
STAG	Source Operand Tag = Normalized
E1	Always 1
T	Always 0
INEX (FADD, FSUB, FMUL, FDIV, and FSQRT)	
CMDREG3B	Encoded Exception Instruction Command Word
WBTEMP	WBTS, WBTE, and WBTM = intermediate result sign, biased 15-bit exponent, and 64-bit mantissa prior to rounding.
WBTE15	Either 1 or 0, generally useless for INEX exceptions.
WBTM1, WBTM0, SBIT	Guard, round, and sticky of intermediate result's 67-bit mantissa.
E3	Always 1
T	Either 1 or 0
INEX (FMOVE to Memory)	
CMDREG1B	FMOVE Instruction Command Word
FPTEMP	Intermediate result with mantissa prior to rounding.
STAG	Source Operand Tag = Normalized
E1	Always 1
T	Always 1

NOTE: If the M68040FPSP unimplemented exception handler is used, the above state frame information applies. The CMDREG1B or CMDREG3B fields of the state frame are modified as appropriate to encode the unimplemented instruction opcode. It is the user exception handler's responsibility to use the E3 and E1 field encodings to recognize which state frame information applies. When E3 = 1 and E1 = 1, E3 takes priority and the state frame information for E3 = 1 must be used.

10.6 INTEGER UNIT INSTRUCTION TIMINGS (Continued)

Addressing Mode	JSR		LEA		MOVE from CCR	
	<ea> Calculate	Execute	<ea> Calculate	Execute	<ea> Calculate	Execute
Dn	—	—	—	—	1	2
An	—	—	—	—	—	—
(An)	3	2_L + 1	1	1	1	2
(An)+	—	—	—	—	1	2
–(An)	—	—	—	—	1	2
(d_{16},An)	4	3_L + 1	2	1_L + 1	1	2
(d_{16},PC)	6	5_L + 1	4	3_L + 1	—	—
(xxx).W, (xxx).L	3	2_L + 1	1	1	1	2
#<xxx>	—	—	—	—	—	—
(d_8,An,Xn)	6	6	4	4	3	4
(d_8,PC,Xn)	7	1_L + 6	5	1_L + 4	—	—
(BR,Xn)	8	1_L + 7	6	1_L + 5	6	1_L + 6
(bd,BR,Xn)	9	1_L + 8	7	1_L + 6	7	1_L + 7
([bd,BR,Xn])	12	1_L + 11	9	1_L + 8	10	1_L + 10
([bd,BR,Xn],od)	13	1_L + 12	10	1_L + 9	11	1_L + 11
([bd,BR],Xn)	13	3_L + 10	10	3_L + 7	11	3_L + 9
([bd,BR],Xn,od)	14	3_L + 11	11	3_L + 8	12	3_L + 10



NOTE: Transfer Attribute Signals = UPAx, SIZx, TTx, TMx, TLNx, R/$\overline{W}$, and $\overline{CIOUT}$

Figure A-6. Bus Arbitration Timing

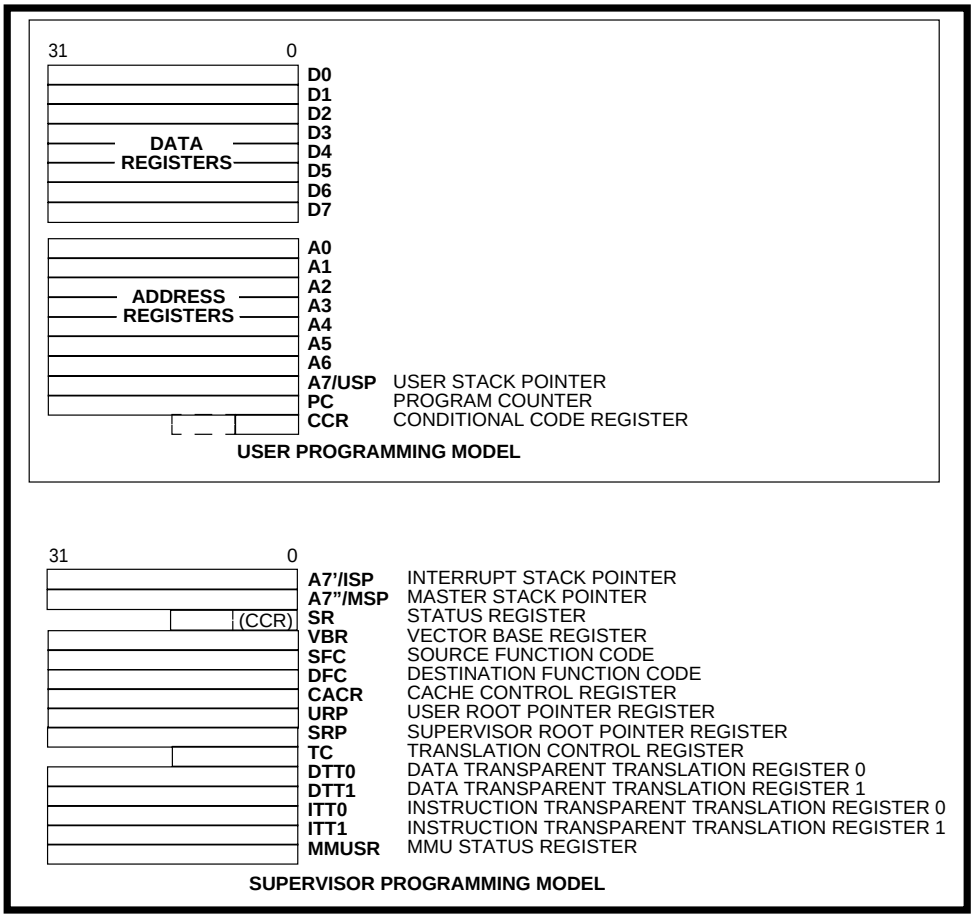


Figure B-2. MC68EC040 Programming Model

- PTEST and PFLUSH instructions cause an indeterminate result (i.e., an undetermined number of bus cycles); the user should not execute them on the MC68EC040.
- The MC68EC040 does not contain an FPU which causes unimplemented floating-point exceptions to occur using a new stack frame format.
- The DLE and $\overline{\text{MDIS}}$ pin names have been changed to JS0 and JS1, respectively.
- The MC68EC040 does not implement the DLE mode, multiplexed, or output buffer impedance selection modes of operation. The MC68EC040 implements only the small output buffer mode of operation. All timing and drive capabilities of the MC68EC040 are equivalent to those of the MC68040 in the small buffer mode of operation.

B.2 JTAG SCAN (JS1–JS0)

The MC68040 $\overline{\text{MDIS}}$ and DLE pin names have been changed to JS1 and JS0 respectively. During normal operation, the JS1 and JS0 pin cannot float, they must be tied to GND or Vcc directly or through a resistor. During board testing, these pins retain the functionality of the JTAG scan of the MC68040 for compatibility purposes. Refer to **Section 6 IEEE 1149.1A**

B.7.3 DC Electrical Specifications (V_{CC} = 5.0 Vdc ±5 %)

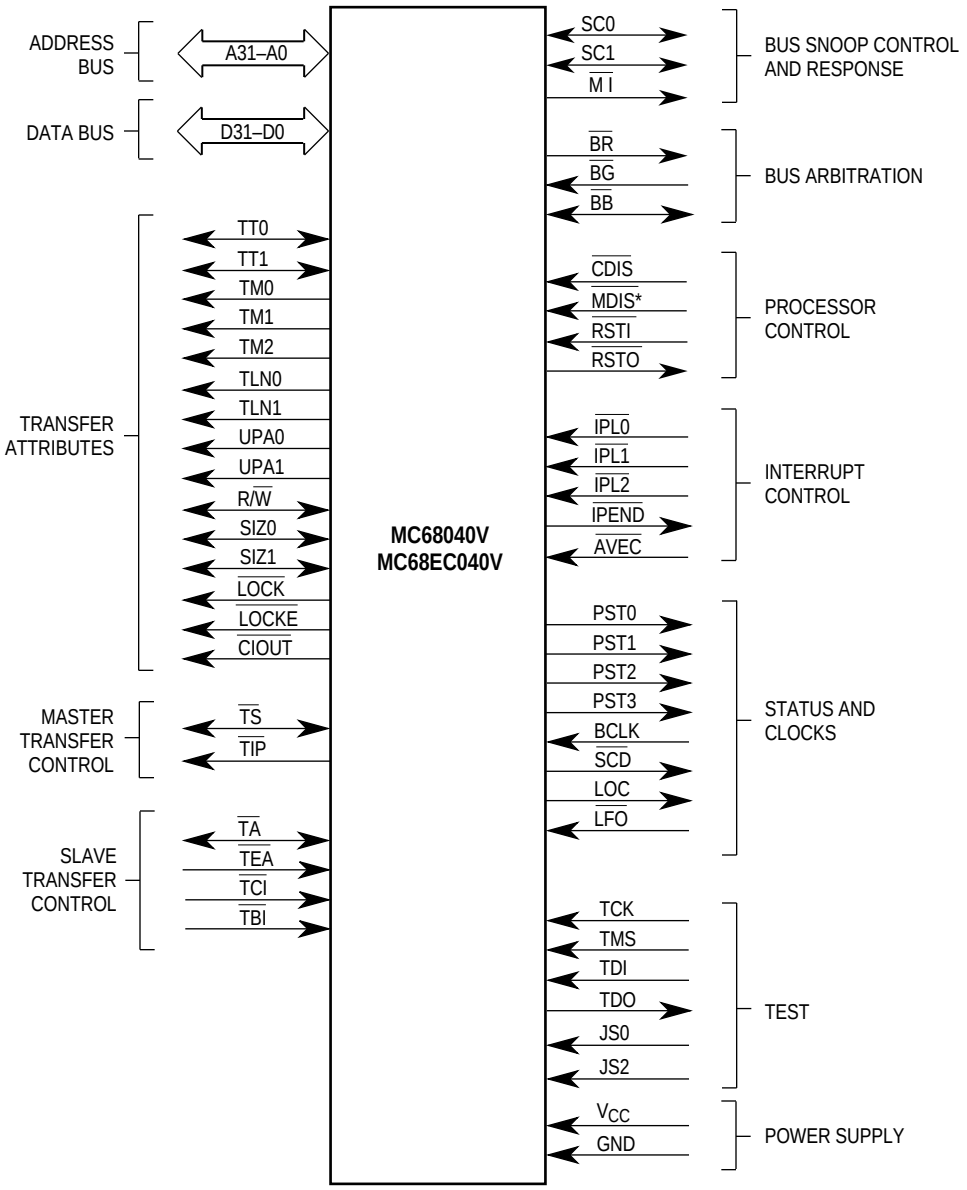
Characteristic	Symbol	Min	Max	Unit
Input High Voltage	V_{IH}	2	V_{CC}	V
Input Low Voltage	V_{IL}	GND	0.8	V
Undershoot	—	—	0.8	V
Input Leakage Current @ 0.5–2.4 V $\overline{AVEC}$, BCLK, $\overline{BG}$, $\overline{CDIS}$, $\overline{IPLA}$, PCLK, $\overline{RSTI}$, SCx, $\overline{TBI}$, TLNx, $\overline{TCI}$, TCK, $\overline{TEA}$	I_{in}	20	20	mA
Hi-Z (Off-State) Leakage Current @ 0.5–2.4 V An, $\overline{BB}$, $\overline{CIOUT}$, Dn, $\overline{LOCK}$, $\overline{LOCKE}$, R/$\overline{W}$, SIZx, $\overline{TA}$, TDO, $\overline{TIP}$, TMx, TLNx, $\overline{TS}$, TTx, UPAx	I_{TSI}	20	20	mA
Signal Low Input Current, V_{IL} = 0.8 V TMS, TDI, $\overline{TRST}$	I_{IL}	–1.1	–0.18	mA
Signal High Input Current, V_{IH} = 2.0 V TMS, TDI, $\overline{TRST}$	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
Capacitance*, V_{in} = 0 V, f = 1 MHz	C_{in}	—	25	pF

*Capacitance is periodically sampled rather than 100% tested.

B.7.4 Power Dissipation

Frequency	Watts
Maximum Values (V_{CC} = 5.25 V, T_A = 0°C)	
20 MHz	3.2
25 MHz	3.9
33 MHz	4.9
40 MHz	5.5
Typical Values (V_{CC} = 5 V, T_A = 25°C)*	
20 MHz	2.0
25 MHz	2.4
33 MHz	3.0
40 MHz	3.5

*This information is for system reliability purposes.



NOTE: *This signal is JS1 on the MC68EC040V.

Figure C-1. MC68040V and MC68EC040V Functional Signal Groups

C.2 LOW-POWER STOP MODE

The low-power stop mode is a reduced power mode of operation, that causes the MC68040V and MC68EC040V to remain quiescent until either a reset or non-masked interrupt occurs. This mode of operation has four phases of operation and is triggered by the low-power stop (LPSTOP) instruction:

1. Perform a LPSTOP broadcast cycle.
2. End integer unit (IU) instruction pipeline sequencing, which is similar to the STOP instruction sequence (IMM data ➧ SR), at termination of the LPSTOP broadcast cycle.

C.2.1 Bus Arbitration and Snooping

Bus arbitration and snooping are not allowed during low-power stop mode. If an alternate bus master requires ownership, arbitration must occur before the processor is allowed to enter low-power stop mode. This is achieved by externally decoding the LPSTOP broadcast cycle and negating the $\overline{BG}$ signal before the termination of the cycle, allowing bus arbitration to complete at the end of the cycle.

If the MC68040V or the MC68EC040V is the bus master during low-power stop mode, lowest power consumption cannot be achieved due to the DC loads on the processor output pins. To achieve maximum power savings, arbitrate bus mastership away from the processor during the LPSTOP broadcast cycle.

In a single bus master system the caches do not need to be shut down prior to the execution of LPSTOP. In a multi-master system, the programmer is responsible for providing a shut down sequence for the caches.

C.2.2 Low Frequency Operation

In addition to the low-power mode of operation the MC68040V and MC68EC040V provide a low frequency mode of operation. This mode of operation can be entered one of in two ways: directly from reset by asserting $\overline{LFO}$ prior to negating $\overline{RSTI}$; or by asserting $\overline{LFO}$ prior to generating the interrupt or reset when exiting the low-power stop mode. In the former case, the BCLK input can be changed as long as the frequency is 0–16 MHz and the minimum pulse width constraints are met. Normal operation can be resumed through the low-power stop mode and deasserting $\overline{LFO}$.

C.2.3 Changing BCLK Frequency

The frequency of the BCLK input can be changed only during the low-power stop or low frequency modes of operation. Once in the low-power stop mode and $\overline{SCD}$ is asserted, BCLK can be disabled or its frequency can be changed. Reducing the frequency or removing the BCLK input is not required for proper operation, but is an additional power saving measure. BCLK can be removed during the low-power stop mode as an additional system power saving measure. However, it is not necessary for normal operation and has no effect on the MC68040V's or MC68EC040V's power consumption.