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	Obsolete
Core Processor	EC000
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	16MHz
Co-Processors/DSP	-
RAM Controllers	-
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	68-BCPGA
Supplier Device Package	68-PGA (26.92x26.92)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc000crc16

LIST OF TABLES (Concluded)

1.1 MC68000

The MC68000 is the first implementation of the M68000 16/-32 bit microprocessor architecture. The MC68000 has a 16-bit data bus and 24-bit address bus while the full architecture provides for 32-bit address and data buses. It is completely code-compatible with the MC68008 8-bit data bus implementation of the M68000 and is upward code compatible with the MC68010 virtual extensions and the MC68020 32-bit implementation of the architecture. Any user-mode programs using the MC68000 instruction set will run unchanged on the MC68008, MC68010, MC68020, MC68030, and MC68040. This is possible because the user programming model is identical for all processors and the instruction sets are proper subsets of the complete architecture.

1.2 MC68008

The MC68008 is a member of the M68000 family of advanced microprocessors. This device allows the design of cost-effective systems using 8-bit data buses while providing the benefits of a 32-bit microprocessor architecture. The performance of the MC68008 is greater than any 8-bit microprocessor and superior to several 16-bit microprocessors.

The MC68008 is available as a 48-pin dual-in-line package (plastic or ceramic) and 52-pin plastic leaded chip carrier. The additional four pins of the 52-pin package allow for additional signals: A20, A21, $\overline{\text{BGACK}}$, and $\overline{\text{IPL2}}$. The 48-pin version supports a 20-bit address that provides a 1-Mbyte address space; the 52-pin version supports a 22-bit address that extends the address space to 4 Mbytes. The 48-pin MC68008 contains a simple two-wire arbitration circuit; the 52-pin MC68008 contains a full three-wire MC68000 bus arbitration control. Both versions are designed to work with daisy-chained networks, priority encoded networks, or a combination of these techniques.

A system implementation based on an 8-bit data bus reduces system cost in comparison to 16-bit systems due to a more effective use of components and byte-wide memories and peripherals. In addition, the nonmultiplexed address and data buses eliminate the need for external demultiplexers, further simplifying the system.

The large nonsegmented linear address space of the MC68008 allows large modular programs to be developed and executed efficiently. A large linear address space allows program segment sizes to be determined by the application rather than forcing the designer to adopt an arbitrary segment size without regard to the application's individual requirements.

1.3 MC68010

The MC68010 utilizes VLSI technology and is a fully implemented 16-bit microprocessor with 32-bit registers, a rich basic instruction set, and versatile addressing modes. The vector base register (VBR) allows the vector table to be dynamically relocated

Notation for operands:

PC — Program counter
SR — Status register
V — Overflow condition code
Immediate Data — Immediate data from the instruction
Source — Source contents
Destination — Destination contents
Vector — Location of exception vector
+inf — Positive infinity
–inf — Negative infinity
<fmt> — Operand data format: byte (B), word (W), long (L), single (S), double (D), extended (X), or packed (P).
FPm — One of eight floating-point data registers (always specifies the source register)
FPn — One of eight floating-point data registers (always specifies the destination register)

Notation for subfields and qualifiers:

<bit> of <operand> — Selects a single bit of the operand
<ea>{offset:width} — Selects a bit field
(<operand>) — The contents of the referenced location
<operand>10 — The operand is binary-coded decimal, operations are performed in decimal
(<address register>) — The register indirect operator
–(<address register>) — Indicates that the operand register points to the memory
(<address register>)+ — Location of the instruction operand—the optional mode qualifiers are –, +, (d), and (d, ix)
#xxx or #<data> — Immediate data that follows the instruction word(s)

Notations for operations that have two operands, written <operand> <op> <operand>, where <op> is one of the following:

→ — The source operand is moved to the destination operand
↔ — The two operands are exchanged
+ — The operands are added
– — The destination operand is subtracted from the source operand
× — The operands are multiplied
÷ — The source operand is divided by the destination operand
< — Relational test, true if source operand is less than destination operand
> — Relational test, true if source operand is greater than destination operand
V — Logical OR
⊕ — Logical exclusive OR
Λ — Logical AND

SECTION 3
SIGNAL DESCRIPTION

This section contains descriptions of the input and output signals. The input and output signals can be functionally organized into the groups shown in Figure 3-1 (for the MC68000, the MC68HC000 and the MC68010), Figure 3-2 (for the MC68HC001), Figure 3-3 (for the MC68EC000), Figure 3-4 (for the MC68008, 48-pin version), and Figure 3-5 (for the MC68008, 52-pin version). The following paragraphs provide brief descriptions of the signals and references (where applicable) to other paragraphs that contain more information about the signals.

NOTE

The terms **assertion** and **negation** are used extensively in this manual to avoid confusion when describing a mixture of "active-low" and "active-high" signals. The term assert or assertion is used to indicate that a signal is active or true, independently of whether that level is represented by a high or low voltage. The term negate or negation is used to indicate that a signal is inactive or false.

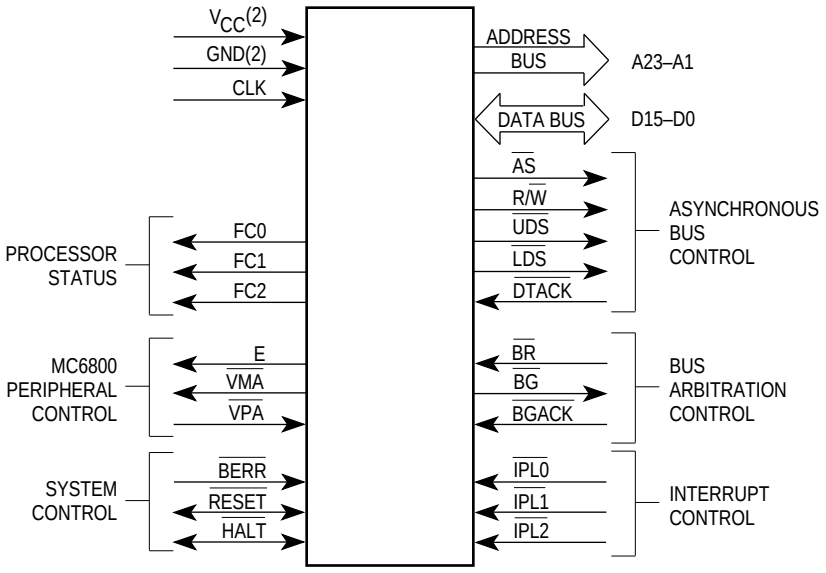


Figure 3-1. Input and Output Signals (MC68000, MC68HC000 and MC68010)

Bus Request ($\overline{BR}$).

This input can be wire-ORed with bus request signals from all other devices that could be bus masters. This signal indicates to the processor that some other device needs to become the bus master. Bus requests can be issued at any time during a cycle or between cycles.

Bus Grant ($\overline{BG}$).

This output signal indicates to all other potential bus master devices that the processor will relinquish bus control at the end of the current bus cycle.

Bus Grant Acknowledge ($\overline{BGACK}$).

This input indicates that some other device has become the bus master. This signal should not be asserted until the following conditions are met:

1. A bus grant has been received.
2. Address strobe is inactive, which indicates that the microprocessor is not using the bus.
3. Data transfer acknowledge is inactive, which indicates that neither memory nor peripherals are using the bus.
4. Bus grant acknowledge is inactive, which indicates that no other device is still claiming bus mastership.

The 48-pin version of the **MC68008** has no pin available for the bus grant acknowledge signal and uses a two-wire bus arbitration scheme instead. If another device in a system supplies a bus grant acknowledge signal, the bus request input signal to the processor should be asserted when either the bus request or the bus grant acknowledge from that device is asserted.

3.5 INTERRUPT CONTROL ($\overline{IPL0}$, $\overline{IPL1}$, $\overline{IPL2}$)

These input signals indicate the encoded priority level of the device requesting an interrupt. Level seven, which cannot be masked, has the highest priority; level zero indicates that no interrupts are requested. $\overline{IPL0}$ is the least significant bit of the encoded level, and $\overline{IPL2}$ is the most significant bit. For each interrupt request, these signals must remain asserted until the processor signals interrupt acknowledge (FC2–FC0 and A19–A16 high) for that request to ensure that the interrupt is recognized.

NOTE

The 48-pin version of the **MC68008** has only two interrupt control signals: $\overline{IPL0}$/$\overline{IPL2}$ and $\overline{IPL1}$. $\overline{IPL0}$/$\overline{IPL2}$ is internally connected to both $\overline{IPL0}$ and $\overline{IPL2}$, which provides four interrupt priority levels: levels 0, 2, 5, and 7. In all other respects, the interrupt priority levels in this version of the **MC68008** are identical to those levels in the other microprocessors described in this manual.

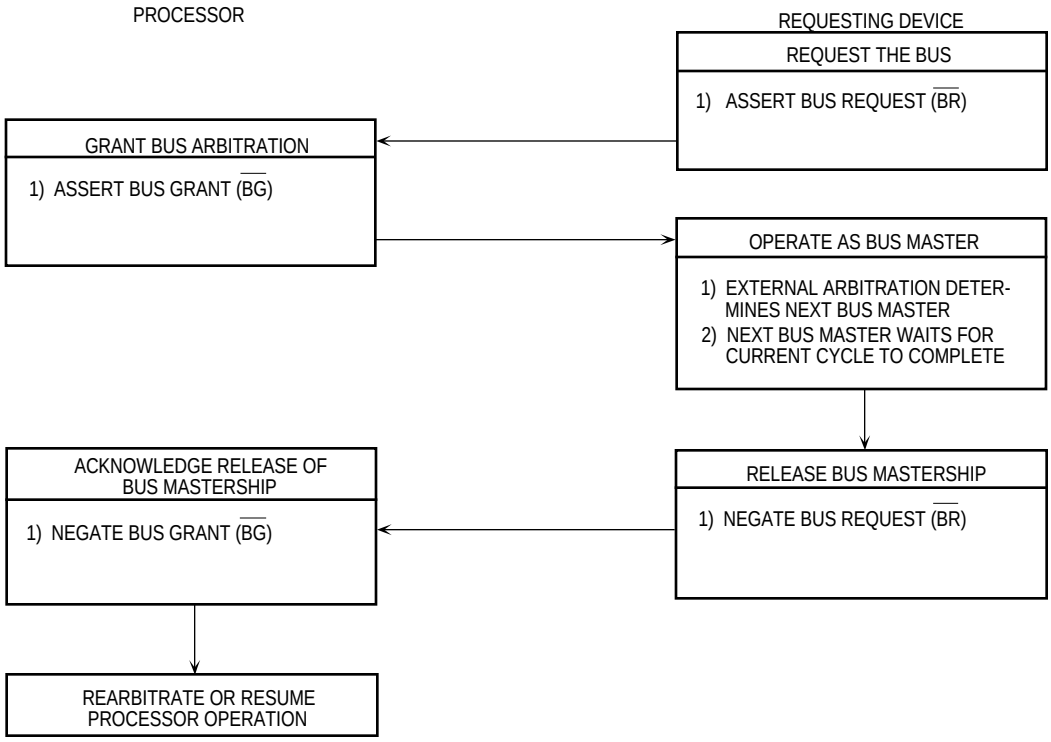


Figure 5-14. 2-Wire Bus Arbitration Cycle Flowchart

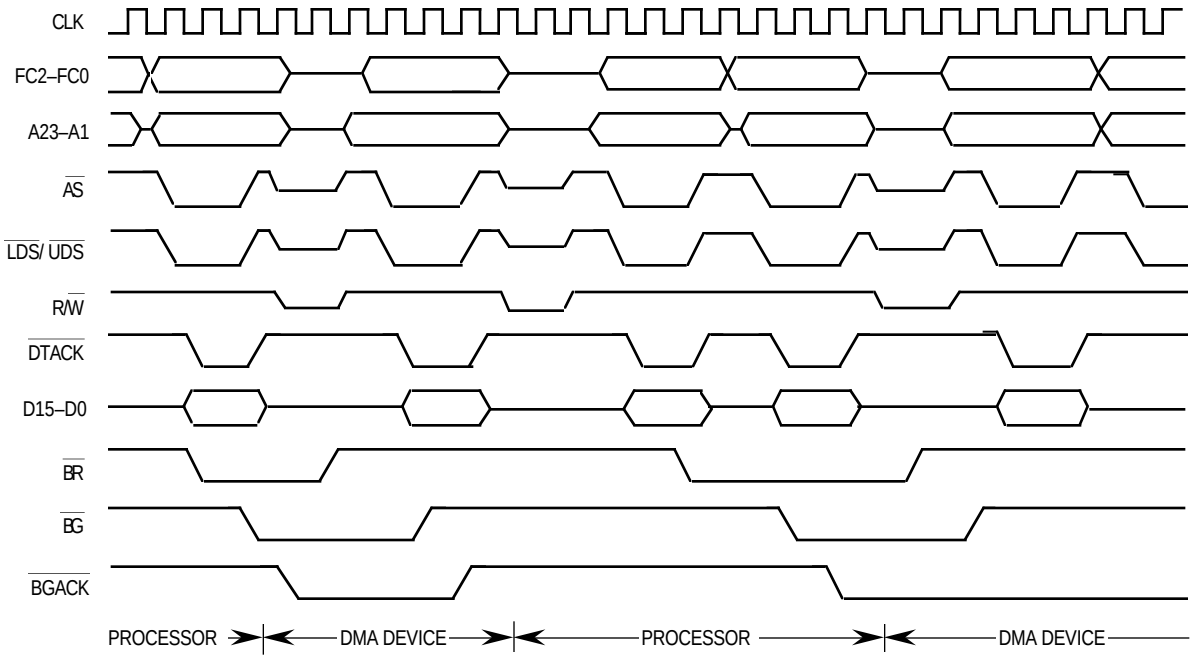


Figure 5-15. 3-Wire Bus Arbitration Timing Diagram (Not Applicable to 48-Pin MC68008 or MC68EC000)

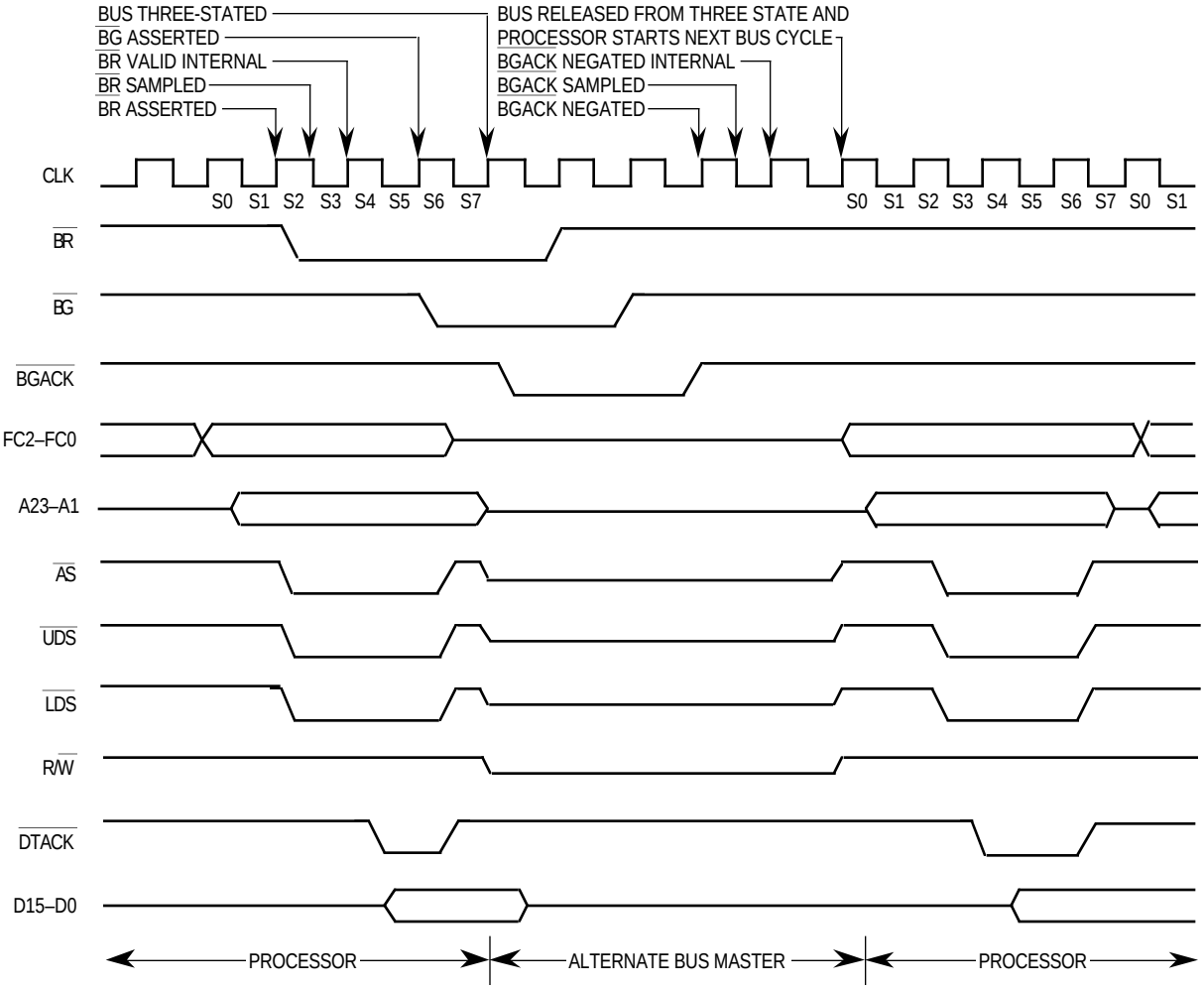


Figure 5-19. 3-Wire Bus Arbitration Timing Diagram—Processor Active

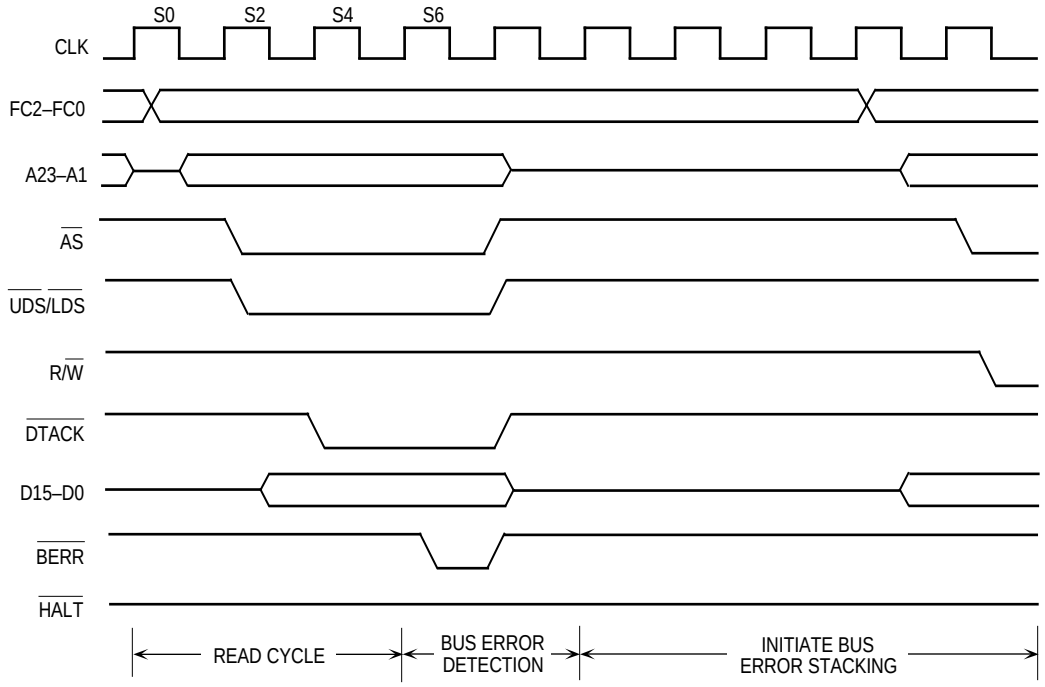


Figure 5-26. Delayed Bus Error Timing Diagram (MC68010)

After the aborted bus cycle is terminated and $\overline{\text{BERR}}$ is negated, the processor enters exception processing for the bus error exception. During the exception processing sequence, the following information is placed on the supervisor stack:

1. Status register
2. Program counter (two words, which may be up to five words past the instruction being executed)
3. Error information

The first two items are identical to the information stacked by any other exception. The error information differs for the MC68010. The MC68000, MC68HC000, MC68HC001, MC68EC000, and MC68008 stack bus error information to help determine and to correct the error. The MC68010 stacks the frame format and the vector offset followed by 22 words of internal register information. The return from exception (RTE) instruction restores the internal register information so that the MC68010 can continue execution of the instruction after the error handler routine completes.

After the processor has placed the required information on the stack, the bus error exception vector is read from vector table entry 2 (offset $08) and placed in the program counter. The processor resumes execution at the address in the vector, which is the first instruction in the bus error handler routine.

Table 5-1. $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, and $\overline{\text{HALT}}$ Assertion Results

Case No.	Control Signal	Asserted on Rising Edge of State		MC68000/MC68HC000/001 EC000/MC68008 Results	MC68010 Results
		N	N+2		
1	$\overline{\text{DTACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	A NA NA	S NA X	Normal cycle terminate and continue.	Normal cycle terminate and continue.
2	$\overline{\text{DTACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	A NA A/S	S NA S	Normal cycle terminate and halt. Continue when $\overline{\text{HALT}}$ negated.	Normal cycle terminate and halt. Continue when $\overline{\text{HALT}}$ negated.
3	$\overline{\text{DTACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	X A NA	X S NA	Terminate and take bus error trap.	Terminate and take bus error trap.
4	$\overline{\text{DTACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	A NA NA	S A NA	Normal cycle terminate and continue.	Terminate and take bus error trap.
5	$\overline{\text{DTACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	X A A/S	X S S	Terminate and retry when $\overline{\text{HALT}}$ removed.	Terminate and retry when $\overline{\text{HALT}}$ removed.
6	$\overline{\text{DTACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	A NA NA	S A A	Normal cycle terminate and continue.	Terminate and retry when $\overline{\text{HALT}}$ removed.

LEGEND:
- N — The number of the current even bus state (e.g., S4, S6, etc.)
- A — Signal asserted in this bus state
- NA — Signal not asserted in this bus state
- X — Don't care
- S — Signal asserted in preceding bus state and remains asserted in this state

NOTE: All operations are subject to relevant setup and hold times.

The negation of $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ under several conditions is shown in Table 5-6. ($\overline{\text{DTACK}}$ is assumed to be negated normally in all cases; for reliable operation, both $\overline{\text{DTACK}}$ and $\overline{\text{BERR}}$ should be negated when address strobe is negated).

EXAMPLE A:

A system uses a watchdog timer to terminate accesses to unused address space. The timer asserts $\overline{\text{BERR}}$ after timeout (case 3).

EXAMPLE B:

A system uses error detection on random-access memory (RAM) contents. The system designer may:

1. Delay $\overline{\text{DTACK}}$ until the data is verified. If data is invalid, return $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ simultaneously to retry the error cycle (case 5).
2. Delay $\overline{\text{DTACK}}$ until the data is verified. If data is invalid, return $\overline{\text{BERR}}$ at the same time as $\overline{\text{DTACK}}$ (case 3).
3. For an MC68010, return $\overline{\text{DTACK}}$ before data verification. If data is invalid, assert $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ to retry the error cycle (case 6).

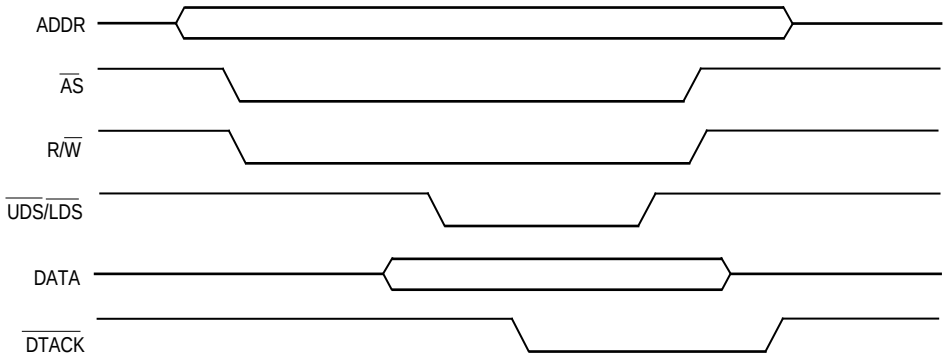


Figure 5-32. Fully Asynchronous Write Cycle

In the asynchronous mode, the accessed device operates independently of the frequency and phase of the system clock. For example, the MC68681 dual universal asynchronous receiver/transmitter (DUART) does not require any clock-related information from the bus master during a bus transfer. Asynchronous devices are designed to operate correctly with processors at any clock frequency when relevant timing requirements are observed.

A device can use a clock at the same frequency as the system clock (e.g., 8, 10, or 12.5, 16, and 20MHz), but without a defined phase relationship to the system clock. This mode of operation is pseudo-asynchronous; it increases performance by observing timing parameters related to the system clock frequency without being completely synchronous with that clock. A memory array designed to operate with a particular frequency processor but not driven by the processor clock is a common example of a pseudo-asynchronous device.

The designer of a fully asynchronous system can make no assumptions about address setup time, which could be used to improve performance. With the system clock frequency known, the slave device can be designed to decode the address bus before recognizing an address strobe. Parameter #11 (refer to **Section 10 Electrical Characteristics**) specifies the minimum time before address strobe during which the address is valid.

In a pseudo-asynchronous system, timing specifications allow $\overline{\text{DTACK}}$ to be asserted for a read cycle before the data from a slave device is valid. The length of time that $\overline{\text{DTACK}}$ may precede data is specified as parameter #31. This parameter must be met to ensure the validity of the data latched into the processor. No maximum time is specified from the assertion of $\overline{\text{AS}}$ to the assertion of $\overline{\text{DTACK}}$. During this unlimited time, the processor inserts wait cycles in one-clock-period increments until $\overline{\text{DTACK}}$ is recognized. Figure 5-33 shows the important timing parameters for a pseudo-asynchronous read cycle.

Parameter #47 of **Section 10 Electrical Characteristics** is the asynchronous input setup time. Signals that meet parameter #47 are guaranteed to be recognized at the next falling edge of the system clock. However, signals that do not meet parameter #47 are not guaranteed to be recognized. In addition, if $\overline{\text{DTACK}}$ is recognized on a falling edge, valid data is latched into the processor (during a read cycle) on the next falling edge, provided the data meets the setup time required (parameter #27). When parameter #27 has been met, parameter #31 may be ignored. If $\overline{\text{DTACK}}$ is asserted with the required setup time before the falling edge of S4, no wait states are incurred, and the bus cycle runs at its maximum speed of four clock periods.

The late $\overline{\text{BERR}}$ in an MC68010 that is operating in a synchronous mode must meet setup time parameter #27A. That is, when $\overline{\text{BERR}}$ is asserted after $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$ must be asserted before the falling edge of the clock, one clock cycle after $\overline{\text{DTACK}}$ is recognized. Violating this requirement may cause the MC68010 to operate erratically.

shown in the previously described format. The number of clock periods, the number of read cycles, and the number of write cycles, respectively, must be added to those of the effective address calculation where indicated by a plus sign (+).

In Table 7-6, the following notation applies:

— Immediate operand
Dn — Data register operand
An — Address register operand
M — Memory operand

Table 7-6. Immediate Instruction Execution Times

Instruction	Size	op #, Dn	op #, An	op #, M
ADDI	Byte	**16**(4/0)	—	**20**(4/1)+
	Word	**16**(4/0)	—	**24**(4/2)+
	Long	**28**(6/0)	—	**40**(6/4)+
ADDQ	Byte	**8**(2/0)	—	**12**(2/1)+
	Word	**8**(2/0)	**12**(2/0)	**16**(2/2)+
	Long	**12**(2/0)	**12**(2/0)	**24**(2/4)+
ANDI	Byte	**16**(4/0)	—	**20**(4/1)+
	Word	**16**(4/0)	—	**24**(4/2)+
	Long	**28**(6/0)	—	**40**(6/4)+
CMPI	Byte	**16**(4/0)	—	**16**(4/0)
	Word	**16**(4/0)	—	**16**(4/0)
	Long	**26**(6/0)	—	**24**(6/0)
EORI	Byte	**16**(4/0)	—	**20**(4/1)+
	Word	**16**(4/0)	—	**24**(4/2)+
	Long	**28**(6/0)	—	**40**(6/4)+
MOVEQ	Long	**8**(2/0)	—	—
ORI	Byte	**16**(4/0)	—	**20**(4/1)+
	Word	**16**(4/0)	—	**24**(4/2)+
	Long	**28**(6/0)	—	**40**(6/4)+
SUBI	Byte	**16**(4/0)	—	**12**(2/1)+
	Word	**16**(4/0)	—	**16**(2/2)+
	Long	**28**(6/0)	—	**24**(2/4)+
SUBQ	Byte	**8**(2/0)	—	**20**(4/1)+
	Word	**8**(2/0)	**12**(2/0)	**24**(4/2)+
	Long	**12**(2/0)	**12**(2/0)	**40**(6/4)+

+Add effective address calculation time.

7.5 SINGLE OPERAND INSTRUCTION EXECUTION TIMES

Table 7-7 lists the timing data for the single operand instructions. The total number of clock periods, the number of read cycles, and the number of write cycles are shown in the previously described format. The number of clock periods, the number of read cycles, and the number of write cycles, respectively, must be added to those of the effective address calculation where indicated by a plus sign (+).

7.7 BIT MANIPULATION INSTRUCTION EXECUTION TIMES

Table 7-9 lists the timing data for the bit manipulation instructions. The total number of clock periods, the number of read cycles, and the number of write cycles are shown in the previously described format. The number of clock periods, the number of read cycles, and the number of write cycles, respectively, must be added to those of the effective address calculation where indicated by a plus sign (+).

Table 7-9. Bit Manipulation Instruction Execution Times

Instruction	Size	Dynamic		Static	
		Register	Memory	Register	Memory
BCHG	Byte Long	— **12**(2/0)*	**12**(2/1)+ —	— **20**(4/0)*	**20**(4/1)+ —
BCLR	Byte Long	— **14**(2/0)*	**12**(2/1)+ —	— **22**(4/0)*	**20**(4/1)+ —
BSET	Byte Long	— **12**(2/0)*	**12**(2/1)+ —	— **20**(4/0)*	**20**(4/1)+ —
BTST	Byte Long	— **10**(2/0)	**8**(2/0)+	— **18**(4/0)	**16**(4/0)+ —

+Add effective address calculation time.
* Indicates maximum value; data addressing mode only.

7.8 CONDITIONAL INSTRUCTION EXECUTION TIMES

Table 7-10 lists the timing data for the conditional instructions. The total number of clock periods, the number of read cycles, and the number of write cycles are shown in the previously described format. The number of clock periods, the number of read cycles, and the number of write cycles, respectively, must be added to those of the effective address calculation where indicated by a plus sign (+).

Table 7-10. Conditional Instruction Execution Times

Instruction	Displacement	Trap or Branch Taken	Trap or Branch Not Taken
Bcc	Byte Word	**18**(4/0) **18**(4/0)	**12**(2/0) **20**(4/0)
BRA	Byte Word	**18**(4/0) **18**(4/0)	— —
BSR	Byte Word	**34**(4/4) **34**(4/4)	— —
DBcc	CC True CC False	— **18**(4/0)	**20**(4/0) **26**(6/0)
CHK	—	**68**(8/6)+*	**14**(2/0)
TRAP	—	**62**(8/6)	**—**
TRAPV	—	**66**(10/6)	**8**(2/0)

+Add effective address calculation time for word operand.
* Indicates maximum base value.

periods, the number of read cycles, and the number of write cycles, respectively, must be added to those of the effective address calculation where indicated by a plus sign (+).

Table 7-15. Exception Processing Execution Times

Exception	Periods
Address Error	**94**(8/14)
Bus Error	**94**(8/14)
CHK Instruction	**68**(8/6)+
Divide by Zero	**66**(8/6)+
Interrupt	**72**(9/6)*
Illegal Instruction	**62**(8/6)
Privilege Violation	**62**(8/6)
$\overline{\text{RESET}}$**	**64**(12/0)
Trace	**62**(8/6)
TRAP Instruction	**62**(8/6)
TRAPV Instruction	**66**(10/6)

\+ Add effective address calculation time.
** Indicates the time from when $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are first sampled as negated to when instruction execution starts.

Table 8-5. Immediate Instruction Execution Times

Instruction	Size	op #, Dn	op #, An	op #, M
ADDI	Byte, Word	**8**(2/0)	—	**12**(2/1)+
	Long	**16**(3/0)	—	**20**(3/2)+
ADDQ	Byte, Word	**4**(1/0)	**4**(1/0)*	**8**(1/1)+
	Long	**8**(1/0)	**8**(1/0)	**12**(1/2)+
ANDI	Byte, Word	**8**(2/0)	—	**12**(2/1)+
	Long	**14**(3/0)	—	**20**(3/2)+
CMPI	Byte, Word	**8**(2/0)	—	**8**(2/0)+
	Long	**14**(3/0)	—	**12**(3/0)+
EORI	Byte, Word	**8**(2/0)	—	**12**(2/1)+
	Long	**16**(3/0)	—	**20**(3/2)+
MOVEQ	Long	**4**(1/0)	—	—
ORI	Byte, Word	**8**(2/0)	—	**12**(2/1)+
	Long	**16**(3/0)	—	**20**(3/2)+
SUBI	Byte, Word	**8**(2/0)	—	**12**(2/1)+
	Long	**16**(3/0)	—	**20**(3/2)+
SUBQ	Byte, Word	**4**(1/0)	**8**(1/0)*	**8**(1/1)+
	Long	**8**(1/0)	**8**(1/0)	**12**(1/2)+

8.5 SINGLE OPERAND INSTRUCTION EXECUTION TIMES

Table 8-6 lists the timing data for the single operand instructions. The total number of clock periods, the number of read cycles, and the number of write cycles are shown in the previously described format. The number of clock periods, the number of read cycles, and the number of write cycles, respectively, must be added to those of the effective address calculation where indicated by a plus sign (+).

Table 10-1 summarizes maximum power dissipation and average junction temperature for the curve drawn in Figure 10-1, using the minimum and maximum values of ambient temperature for different packages and substituting $^{\theta}J_C$ for $^{\theta}J_A$ (assuming good thermal management). Table 10-2 provides the maximum power dissipation and average junction temperature assuming that no thermal management is applied (i.e., still air).

NOTE

Since the power dissipation curve shown in Figure 10-1 is negatively sloped, power dissipation declines as ambient temperature increases. Therefore, maximum power dissipation occurs at the lowest rated ambient temperature, but the highest average junction temperature occurs at the maximum ambient temperature where *power dissipation is lowest*.

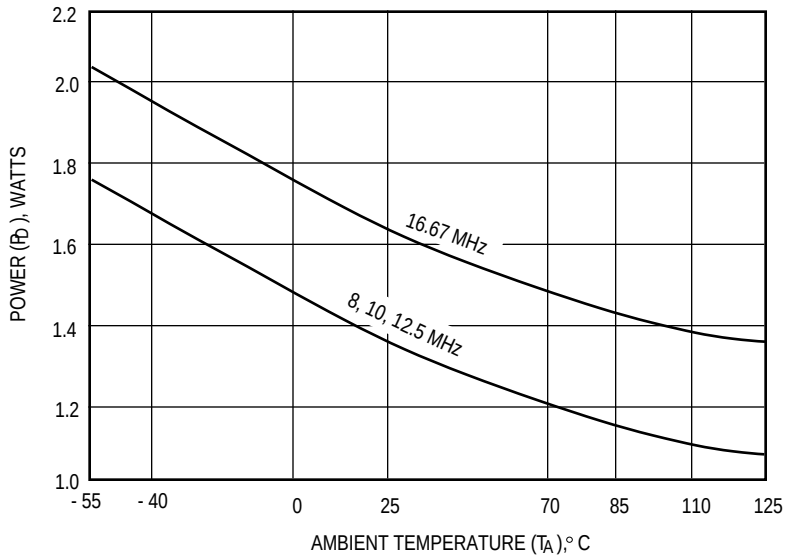
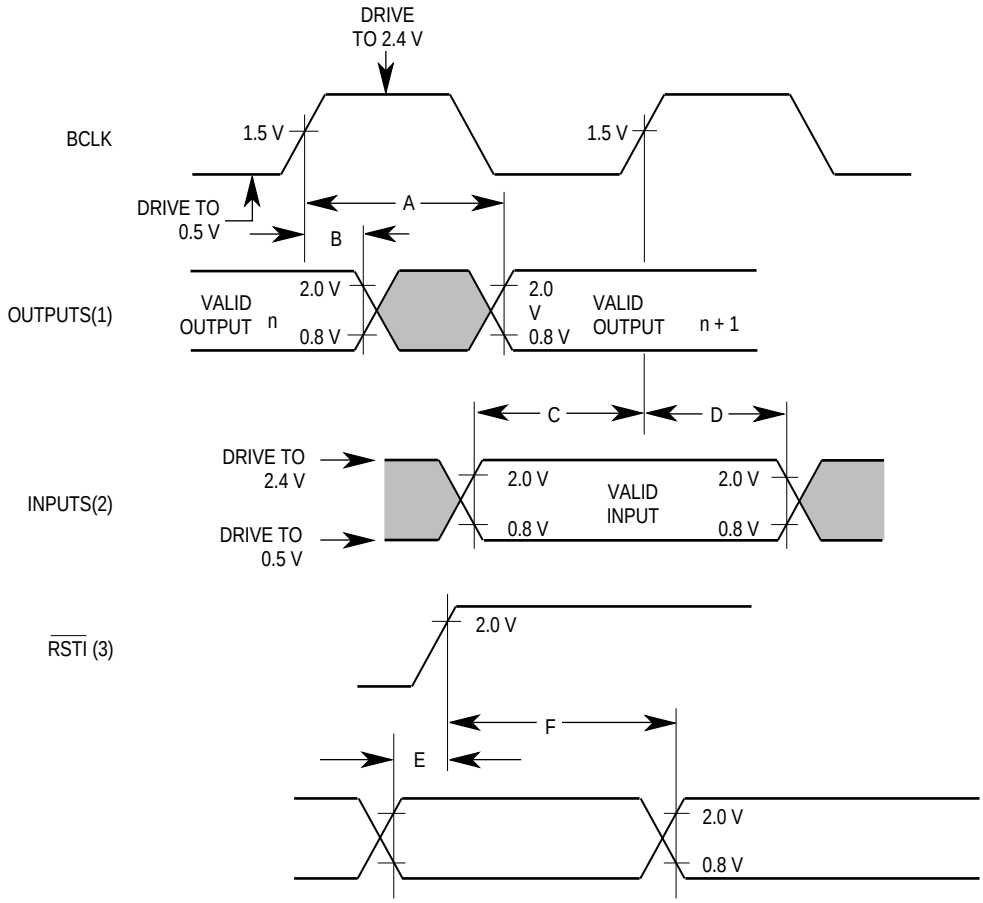


Figure 10-1. MC68000 Power Dissipation (P_D) vs Ambient Temperature (T_A) (Not Applicable to MC68HC000/68HC001/68EC000)



NOTES:

1. This output timing is applicable to all parameters specified relative to the rising edge of the clock.
2. This input timing is applicable to all parameters specified relative to the rising edge of the clock.
3. This timing is applicable to all parameters specified relative to the negation of the RESET signal.

LEGEND:

A. Maximum output delay specification.
B. Minimum output hold time.
C. Minimum input setup time specification.
D. Minimum input hold time specification.
E. Mode select setup time to RESET negated.
F. Mode select hold time from RESET negated.

Figure 10-2. Drive Levels and Test Points for AC Specifications

10.11 AC ELECTRICAL SPECIFICATIONS—MC68000 TO M6800 PERIPHERAL (V_{CC} = 5.0 Vdc ±5%; GND=0 Vdc; $T_A = T_L$ TO T_H; refer to figures 10-6)

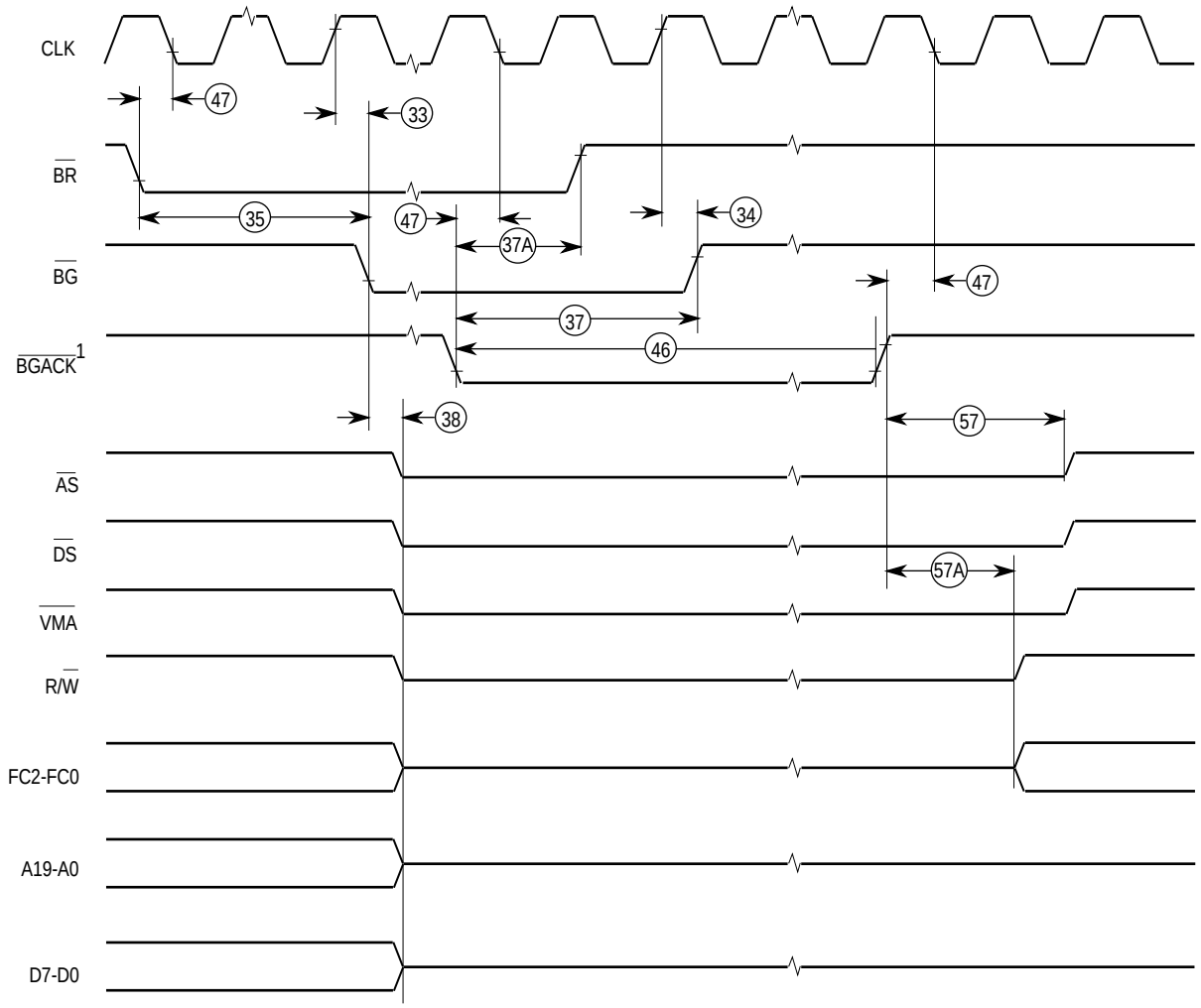
(Applies To All Processors Except The MC68EC000)

Num	Characteristic	8 MHz*		10 MHz*		12.5 MHz*		16.67 MHz `12F'		16 MHz		20 MHz**		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
12[1]	Clock Low to $\overline{AS}$, $\overline{DS}$ Negated	—	62	—	50	—	40	—	40	3	30	3	25	ns
18[1]	Clock High to R/$\overline{W}$ High (Read)	0	55	0	45	0	40	0	40	0	30	0	25	ns
20[1]	Clock High to R/$\overline{W}$ Low (Write)	0	55	0	45	0	40	0	40	0	30	0	25	ns
23	Clock Low to Data-Out Valid (Write)	—	62	—	50	—	50	—	50	—	30	—	25	ns
27	Data-In Valid to Clock Low (Setup Time on Read)	10	—	10	—	10	—	7	—	5	—	5	—	ns
29	$\overline{AS}$, $\overline{DS}$ Negated to Data-In Invalid (Hold Time on Read)	0	—	0	—	0	—	0	—	0	—	0	—	ns
40	Clock Low to $\overline{VMA}$ Asserted	—	70	—	70	—	70	—	50	—	50	—	40	ns
41	Clock Low to E Transition	—	55	—	45	—	35	—	35	—	35	—	30	ns
42	E Output Rise and Fall Time	—	15	—	15	—	15	—	15	—	15	—	12	ns
43	$\overline{VMA}$ Asserted to E High	200	—	150	—	90	—	80	—	80	—	60	—	ns
44	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{VPA}$ Negated	0	120	0	90	0	70	0	50	0	50	0	42	ns
45	E Low to Control, Address Bus Invalid (Address Hold Time)	30	—	10	—	10	—	10	—	10	—	10	—	ns
47	Asynchronous Input Setup Time	10	—	10	—	10	—	10	—	10	—	5	—	ns
49[2]	$\overline{AS}$, $\overline{DS}$, Negated to E Low	-70	70	-55	55	-45	45	-35	35	-35	35	–30	30	ns
50	E Width High	450	—	350	—	280	—	220	—	220	—	190	—	ns
51	E Width Low	700	—	550	—	440	—	340	—	340	—	290	—	ns
54	E Low to Data-Out Invalid	30	—	20	—	15	—	10	—	10	—	5	—	ns

*These specifications represent improvement over previously published specifications for the 8-, 10-, and 12.5-MHz MC68000 and are valid only for product bearing date codes of 8827 and later.

** This frequency applies only to MC68HC000 and MC68HC001.

NOTES:
1. For a loading capacitance of less than or equal to 50 pF, subtract 5 ns from the value given in the maximum columns.
2. The falling edge of S6 triggers both the negation of the strobes ($\overline{AS}$ and $\overline{DS}$) and the falling edge of E. Either of these events can occur first, depending upon the loading on each signal. Specificaton #49 indicates the absolute maximum skew that will occur between the rising edge of the strobes and the falling edge of the E clock.



NOTES: Waveform measurements for all inputs and outputs are specified at: logic high 2.0 V, logic low = 0.8 V.
1. MC68008 52-Pin Version only.

Figure 10-9. Bus Arbitration Timing — Idle Bus Case
(Applies To All Processors Except The MC68EC000)

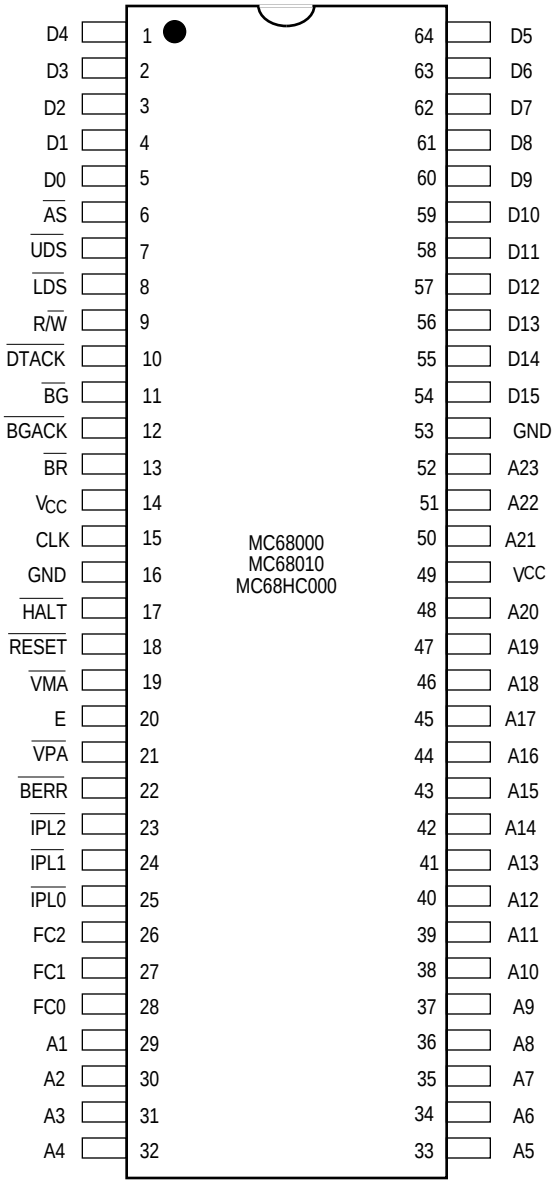


Figure 11-1. 64-Pin Dual In Line