NXP USA Inc. - MC68HC000CEI8 Datasheet

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	8MHz
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-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.21x24.21)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc000cei8

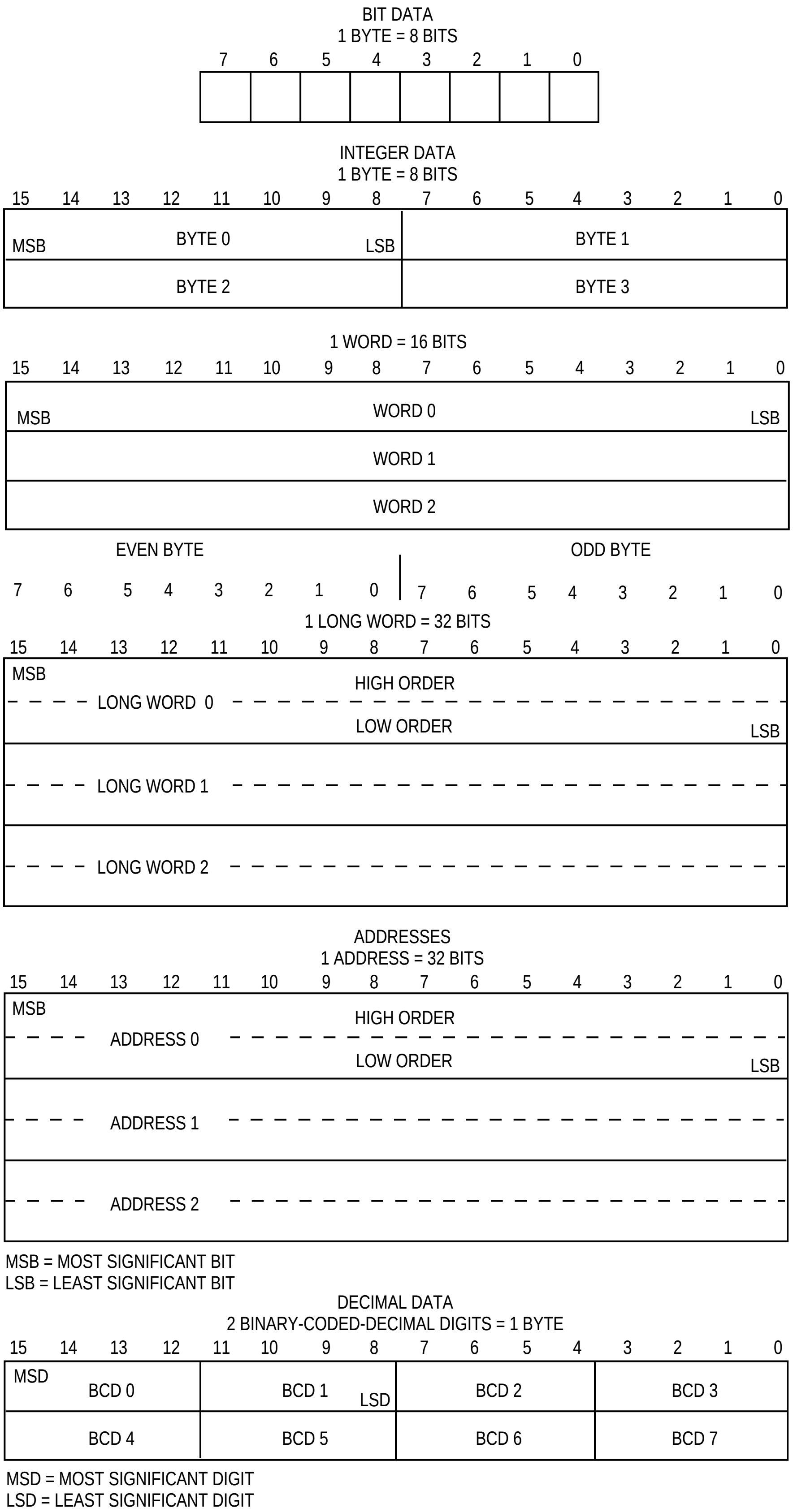


Figure 2-6. Data Organization in Memory

Table 2-2. Instruction Set Summary (Sheet 4 of 4)

Opcode	Operation	Syntax

RTE	If supervisor state
	then (SP) → SR; SP + 2 → SP; (SP) → PC;
	SP + 4 → SP;
	restore state and deallocate stack according to (SP)
	else TRAP
		RTE

RTR	(SP) → CCR; SP + 2 → SP; (SP) → PC; SP + 4 → SP	RTR
RTS	(SP) → PC; SP + 4 → SP	RTS
SBCD	Destination$_{10}$ – Source$_{10}$ – X → Destination	SBCD Dx,Dy SBCD –(Ax),–(Ay)
Scc	If condition true then 1s → Destination else 0s → Destination	Scc <ea>
STOP	If supervisor state then Immediate Data → SR; STOP else TRAP	STOP # <data>
SUB	Destination – Source → Destination	SUB <ea>,Dn SUB Dn,<ea>
SUBA	Destination – Source → Destination	SUBA <ea>,An
SUBI	Destination – Immediate Data → Destination	SUBI # <data>,<ea>
SUBQ	Destination – Immediate Data → Destination	SUBQ # <data>,<ea>
SUBX	Destination – Source – X → Destination	SUBX Dx,Dy SUBX –(Ax),–(Ay)
SWAP	Register [31:16] ↔ Register [15:0]	SWAP Dn
TAS	Destination Tested → Condition Codes; 1 → bit 7 of Destination	TAS <ea>

TRAP	SSP – 2 → SSP; Format/Offset → (SSP);
	SSP – 4 → SSP; PC → (SSP); SSP–2 → SSP;
	SR → (SSP); Vector Address → PC
		TRAP # <vector>

TRAPV	If V then TRAP	TRAPV
TST	Destination Tested → Condition Codes	TST <ea>
UNLK	An → SP; (SP) → An; SP + 4 → SP	UNLK An

NOTE: d is direction, L or R.

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.

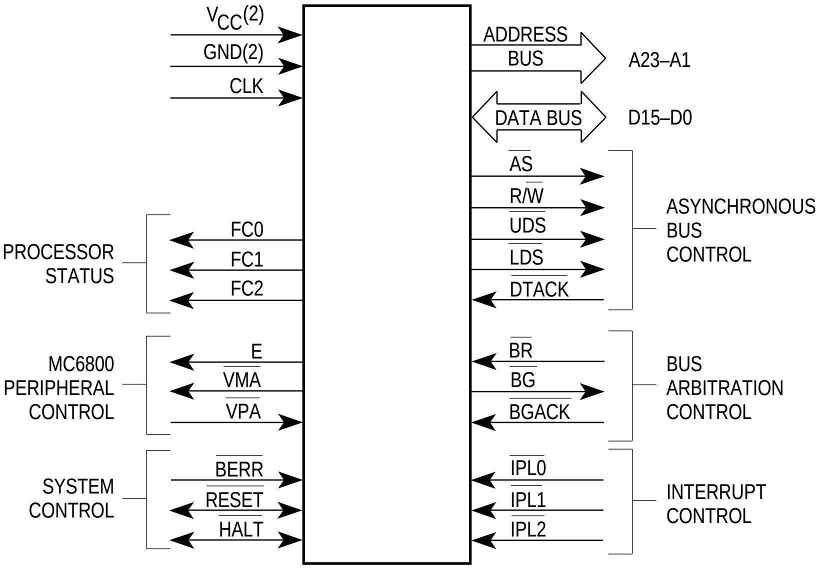


Figure 3-1. Input and Output Signals (MC68000, MC68HC000 and MC68010)

Address Bus (A23–A0)

This 24-bit, unidirectional, three-state bus is capable of addressing 16 Mbytes of data. This bus provides the address for bus operation during all cycles except interrupt acknowledge cycles and breakpoint cycles. During interrupt acknowledge cycles, address lines A1, A2, and A3 provide the level number of the interrupt being acknowledged, and address lines A23–A4 and A0 are driven to logic high. In 16-Bit mode, A0 is always driven high.

MC68008 Address Bus

The unidirectional, three-state buses in the two versions of the **MC68008** differ from each other and from the other processor bus only in the number of address lines and the addressing range. The 20-bit address (A19–A0) of the 48-pin version provides a 1-Mbyte address space; the 52-pin version supports a 22-bit address (A21–A0), extending the address space to 4 Mbytes. During an interrupt acknowledge cycle, the interrupt level number is placed on lines A1, A2, and A3. Lines A0 and A4 through the most significant address line are driven to logic high.

3.2 DATA BUS (D15–D0; MC68008: D7–D0)

This bidirectional, three-state bus is the general-purpose data path. It is 16 bits wide in the all the processors except the **MC68008** which is 8 bits wide. The bus can transfer and accept data of either word or byte length. During an interrupt acknowledge cycle, the external device supplies the vector number on data lines D7–D0. The MC68EC000 and MC68HC001 use D7–D0 in 8-bit mode, and D15–D8 are undefined.

3.3 ASYNCHRONOUS BUS CONTROL

Asynchronous data transfers are controlled by the following signals: address strobe, read/write, upper and lower data strobes, and data transfer acknowledge. These signals are described in the following paragraphs.

Address Strobe ($\overline{\text{AS}}$).

This three-state signal indicates that the information on the address bus is a valid address.

Read/Write (R/$\overline{\text{W}}$).

This three-state signal defines the data bus transfer as a read or write cycle. The R/$\overline{\text{W}}$ signal relates to the data strobe signals described in the following paragraphs.

Upper And Lower Data Strobes ($\overline{\text{UDS}}$, $\overline{\text{LDS}}$).

These three-state signals and R/$\overline{\text{W}}$ control the flow of data on the data bus. Table 3-1 lists the combinations of these signals and the corresponding data on the bus. When the R/$\overline{\text{W}}$ line is high, the processor reads from the data bus. When the R/$\overline{\text{W}}$ line is low, the processor drives the data bus. In 8-bit mode, $\overline{\text{UDS}}$ is always forced high and the $\overline{\text{LDS}}$ signal is used.

3.11 SIGNAL SUMMARY

Table 3-4 summarizes the signals discussed in the preceding paragraphs.

Table 3-4. Signal Summary

Signal Name	Mnemonic	Input/Output	Active State	Hi-Z	
				On $\overline{\mathrm{HALT}}$	On Bus Relinquish
Address Bus	A0–A23	Output	High	Yes	Yes
Data Bus	D0–D15	Input/Output	High	Yes	Yes
Address Strobe	$\overline{\mathrm{AS}}$	Output	Low	No	Yes
Read/Write	R/$\overline{\mathrm{W}}$	Output	Read-High Write-Low	No	Yes
Data Strobe	$\overline{\mathrm{DS}}$	Output	Low	No	Yes
Upper and Lower Data Strobes	$\overline{\mathrm{UDS}}$, $\overline{\mathrm{LDS}}$	Output	Low	No	Yes
Data Transfer Acknowledge	$\overline{\mathrm{DTACK}}$	Input	Low	No	No
Bus Request	$\overline{\mathrm{BR}}$	Input	Low	No	No
Bus Grant	$\overline{\mathrm{BG}}$	Output	Low	No	No
Bus Grant Acknowledge	$\overline{\mathrm{BGACK}}$	Input	Low	No	No
Interrupt Priority Level	$\overline{\mathrm{IPL}}$0, $\overline{\mathrm{IPL}}$1, $\overline{\mathrm{IPL}}$2	Input	Low	No	No
Bus Error	$\overline{\mathrm{BERR}}$	Input	Low	No	No
Mode	MODE	Input	High	—	—
Reset	$\overline{\mathrm{RESET}}$	Input/Output	Low	No*	No*
Halt	$\overline{\mathrm{HALT}}$	Input/Output	Low	No*	No*
Enable	E	Output	High	No	No
Valid Memory Address	$\overline{\mathrm{VMA}}$	Output	Low	No	Yes
Valid Peripheral Address	$\overline{\mathrm{VPA}}$	Input	Low	No	No
Function Code Output	FC0, FC1, FC2	Output	High	No	Yes
Clock	CLK	Input	High	No	No
Power Input	V_{CC}	Input	—	—	—
Ground	GND	Input	—	—	—

*Open drain.

SECTION 5
16-BIT BUS OPERATION

The following paragraphs describe control signal and bus operation for 16-bit bus operations during data transfer operations, bus arbitration, bus error and halt conditions, and reset operation. The 16-bit bus operation devices are the MC68000, MC68HC000, MC68010, and the MC68HC001 and MC68EC000 in 16-bit mode. The MC68HC001 and MC68EC000 select 16-bit mode by pulling mode high or leave it floating during reset.

5.1 DATA TRANSFER OPERATIONS

Transfer of data between devices involves the following signals:

1. Address bus A1 through highest numbered address line
2. Data bus D0 through D15
3. Control signals

The address and data buses are separate parallel buses used to transfer data using an asynchronous bus structure. In all cases, the bus master must deskew all signals it issues at both the start and end of a bus cycle. In addition, the bus master must deskew the acknowledge and data signals from the slave device.

The following paragraphs describe the read, write, read-modify-write, and CPU space cycles. The indivisible read-modify-write cycle implements interlocked multiprocessor communications. A CPU space cycle is a special processor cycle.

5.1.1 Read Cycle

During a read cycle, the processor receives either one or two bytes of data from the memory or from a peripheral device. If the instruction specifies a word or long-word operation, the MC68000, MC68HC000, MC68HC001, MC68EC000, or MC68010 processor reads both upper and lower bytes simultaneously by asserting both upper and lower data strobes. When the instruction specifies byte operation, the processor uses the internal A0 bit to determine which byte to read and issues the appropriate data strobe. When A0 equals zero, the upper data strobe is issued; when A0 equals one, the lower data strobe is issued. When the data is received, the processor internally positions the byte appropriately.

The word read-cycle flowchart is shown in Figure 5-1 and the byte read-cycle flowchart is shown in Figure 5-2. The read and write cycle timing is shown in Figure 5-3 and the word and byte read-cycle timing diagram is shown in Figure 5-4.

The word and byte write-cycle timing diagram and flowcharts in Figures 5-5, 5-6, and 5-7 applies directly to the MC68000, the MC68HC000, the MC68HC001 (in 16-bit mode), the MC68EC000 (in 16-bit mode), and the MC68010.

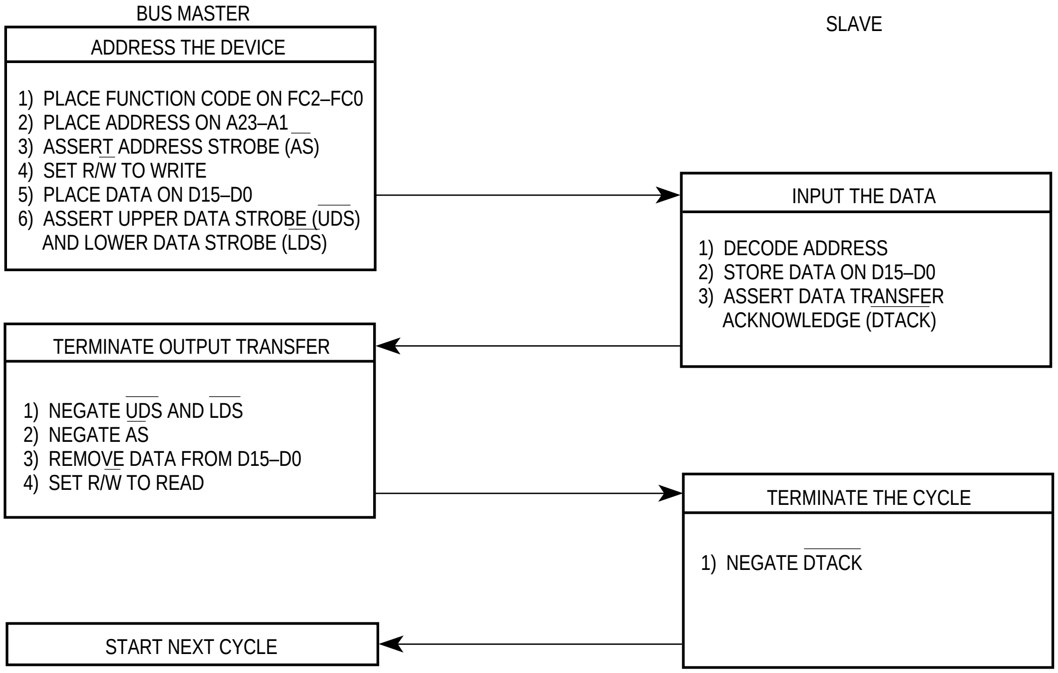

BUS MASTER
SLAVE
ADDRESS THE DEVICE
1) PLACE FUNCTION CODE ON FC2–FC0
2) PLACE ADDRESS ON A23–A1
3) ASSERT ADDRESS STROBE (AS)
4) SET R/W TO WRITE
5) PLACE DATA ON D15–D0
6) ASSERT UPPER DATA STROBE (UDS) AND LOWER DATA STROBE (LDS)
INPUT THE DATA
1) DECODE ADDRESS
2) STORE DATA ON D15–D0
3) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
TERMINATE OUTPUT TRANSFER
1) NEGATE UDS AND LDS
2) NEGATE AS
3) REMOVE DATA FROM D15–D0
4) SET R/W TO READ
TERMINATE THE CYCLE
1) NEGATE DTACK
START NEXT CYCLE

Figure 5-5. Word Write-Cycle Flowchart

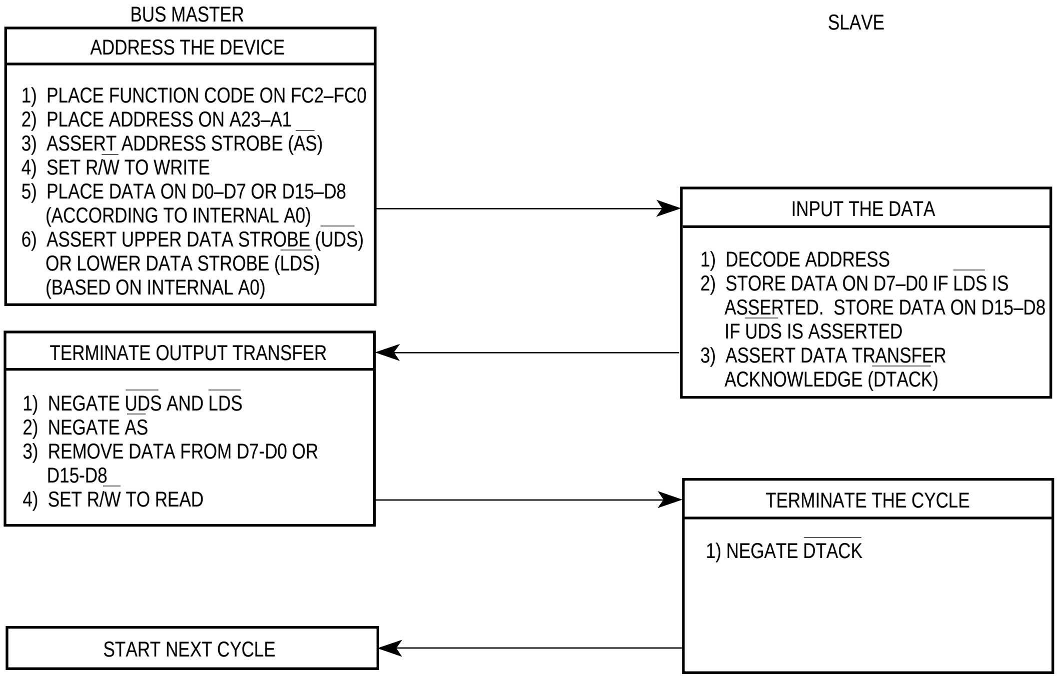

BUS MASTER
SLAVE
ADDRESS THE DEVICE
1) PLACE FUNCTION CODE ON FC2–FC0
2) PLACE ADDRESS ON A23–A1
3) ASSERT ADDRESS STROBE (AS)
4) SET R/W TO WRITE
5) PLACE DATA ON D0–D7 OR D15–D8 (ACCORDING TO INTERNAL A0)
6) ASSERT UPPER DATA STROBE (UDS) OR LOWER DATA STROBE (LDS) (BASED ON INTERNAL A0)
INPUT THE DATA
1) DECODE ADDRESS
2) STORE DATA ON D7–D0 IF LDS IS ASSERTED. STORE DATA ON D15–D8 IF UDS IS ASSERTED
3) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
TERMINATE OUTPUT TRANSFER
1) NEGATE UDS AND LDS
2) NEGATE AS
3) REMOVE DATA FROM D7-D0 OR D15-D8
4) SET R/W TO READ
TERMINATE THE CYCLE
1) NEGATE DTACK
START NEXT CYCLE

Figure 5-6. Byte Write-Cycle Flowchart

STATE 7 On the falling edge of the clock entering S7, the processor negates AS, UDS, or LDS. As the clock rises at the end of S7, the processor places the address and data buses in the high-impedance state, and drives R/W high. The device negates DTACK or BERR at this time.

5.1.3 Read-Modify-Write Cycle.

The read-modify-write cycle performs a read operation, modifies the data in the arithmetic logic unit, and writes the data back to the same address. The address strobe (AS) remains asserted throughout the entire cycle, making the cycle indivisible. The test and set (TAS) instruction uses this cycle to provide a signaling capability without deadlock between processors in a multiprocessing environment. The TAS instruction (the only instruction that uses the read-modify-write cycle) only operates on bytes. Thus, all read-modify-write cycles are byte operations. The read-modify-write flowchart shown in Figure 5-8 and the timing diagram in Figure 5-9, applies to the MC68000, the MC68HC000, the MC68HC001 (in 16-bit mode), the MC68EC000 (in 16-bit mode), and the MC68010.

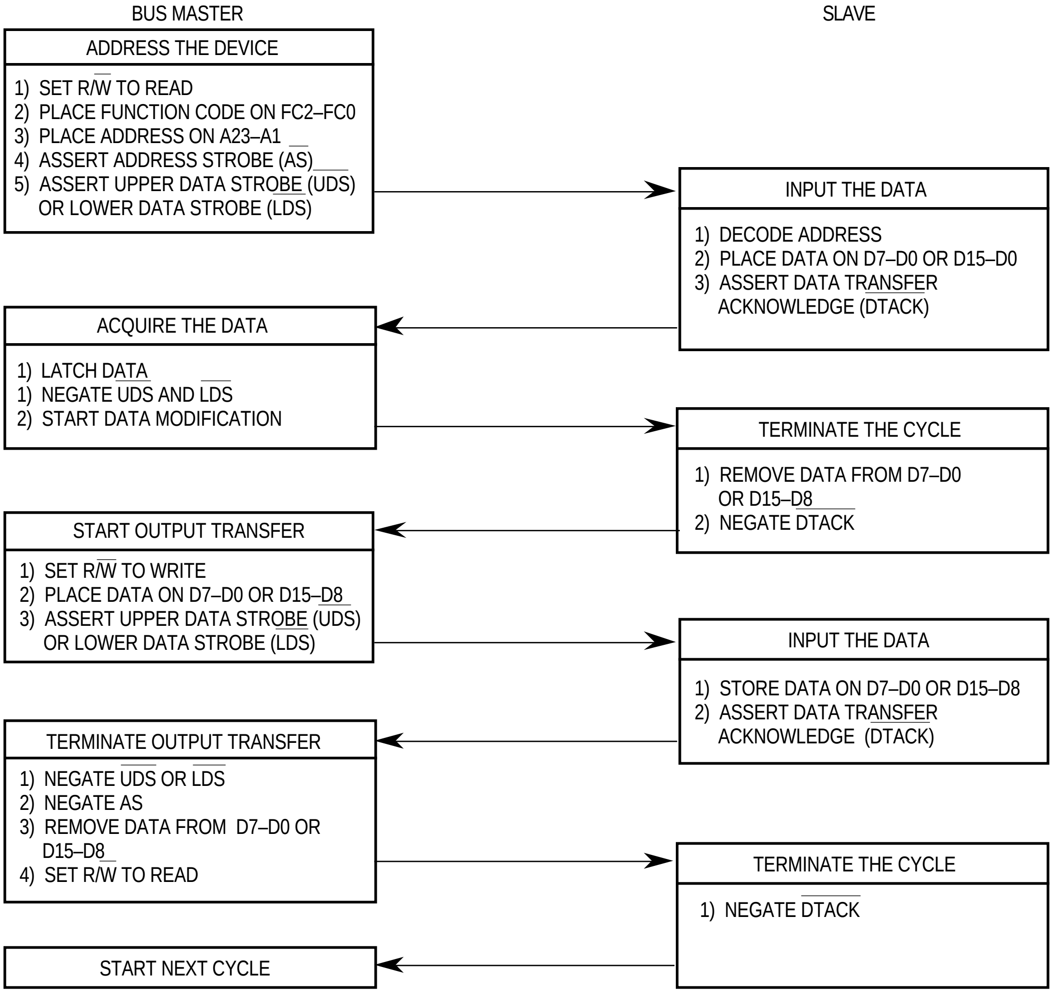

BUS MASTER
SLAVE
ADDRESS THE DEVICE
1) SET R/W TO READ
2) PLACE FUNCTION CODE ON FC2–FC0
3) PLACE ADDRESS ON A23–A1
4) ASSERT ADDRESS STROBE (AS)
5) ASSERT UPPER DATA STROBE (UDS) OR LOWER DATA STROBE (LDS)
INPUT THE DATA
1) DECODE ADDRESS
2) PLACE DATA ON D7–D0 OR D15–D0
3) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
ACQUIRE THE DATA
1) LATCH DATA
1) NEGATE UDS AND LDS
2) START DATA MODIFICATION
TERMINATE THE CYCLE
1) REMOVE DATA FROM D7–D0 OR D15–D8
2) NEGATE DTACK
START OUTPUT TRANSFER
1) SET R/W TO WRITE
2) PLACE DATA ON D7–D0 OR D15–D8
3) ASSERT UPPER DATA STROBE (UDS) OR LOWER DATA STROBE (LDS)
INPUT THE DATA
1) STORE DATA ON D7–D0 OR D15–D8
2) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
TERMINATE OUTPUT TRANSFER
1) NEGATE UDS OR LDS
2) NEGATE AS
3) REMOVE DATA FROM D7–D0 OR D15–D8
4) SET R/W TO READ
TERMINATE THE CYCLE
1) NEGATE DTACK
START NEXT CYCLE

Figure 5-8. Read-Modify-Write Cycle Flowchart

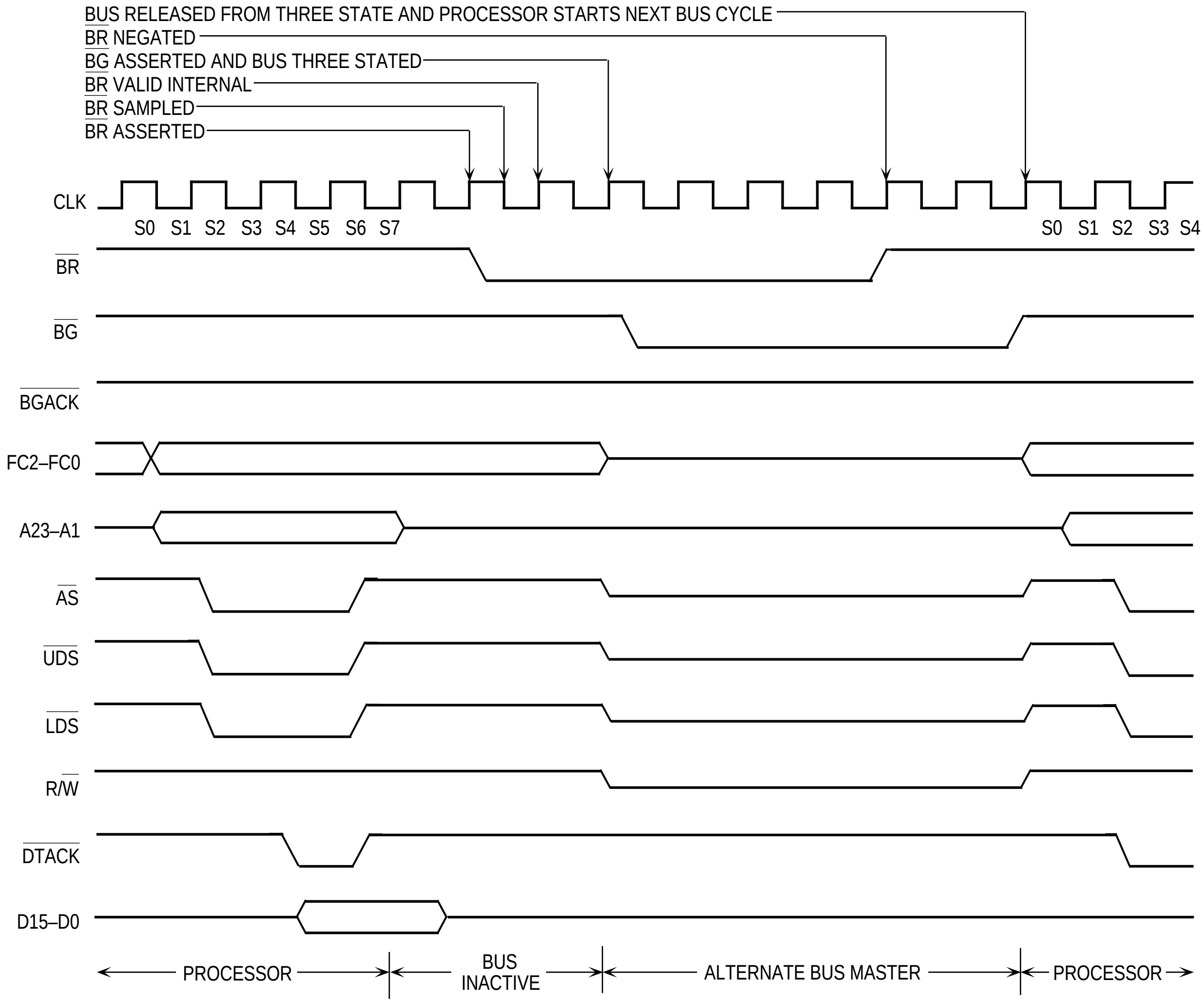


Figure 5-23. 2-Wire Bus Arbitration Timing Diagram—Bus Inactive

executing a reset instruction is ignored. Since the processor asserts the $\overline{\text{RESET}}$ signal for 124 clock cycles during execution of a reset instruction, an external reset should assert $\overline{\text{RESET}}$ for at least 132 clock periods.

5.6 THE RELATIONSHIP OF $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, AND $\overline{\text{HALT}}$

To properly control termination of a bus cycle for a retry or a bus error condition, $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, and $\overline{\text{HALT}}$ should be asserted and negated on the rising edge of the processor clock. This relationship assures that when two signals are asserted simultaneously, the required setup time (specification #47, **Section 9 Electrical Characteristics**) for both of them is met during the same bus state. External circuitry should be designed to incorporate this precaution. A related specification, #48, can be ignored when $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, and $\overline{\text{HALT}}$ are asserted and negated on the rising edge of the processor clock.

The possible bus cycle termination can be summarized as follows (case numbers refer to Table 5-5).

Normal Termination: $\overline{\text{DTACK}}$ is asserted. $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ remain negated (case 1).

Halt Termination: $\overline{\text{HALT}}$ is asserted coincident with or preceding $\overline{\text{DTACK}}$, and $\overline{\text{BERR}}$ remains negated (case 2).

Bus Error Termination: $\overline{\text{BERR}}$ is asserted in lieu of, coincident with, or preceding $\overline{\text{DTACK}}$ (case 3). In the MC68010, the late bus error also, $\overline{\text{BERR}}$ is asserted following $\overline{\text{DTACK}}$ (case 4). $\overline{\text{HALT}}$ remains negated and $\overline{\text{BERR}}$ is negated coincident with or after $\overline{\text{DTACK}}$.

Retry Termination: $\overline{\text{HALT}}$ and $\overline{\text{BERR}}$ asserted in lieu of, coincident with, or before $\overline{\text{DTACK}}$ (case 5). In the MC68010, the late retry also, $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ are asserted following $\overline{\text{DTACK}}$ (case 6). $\overline{\text{BERR}}$ is negated coincident with or after $\overline{\text{DTACK}}$. $\overline{\text{HALT}}$ must be held at least one cycle after $\overline{\text{BERR}}$.

Table 5-1 shows the details of the resulting bus cycle termination in the M68000 microprocessors for various combinations of signal sequences.

Table 5-1. DTACK, BERR, and 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	DTACK BERR HALT	A NA NA	S NA X	Normal cycle terminate and continue.	Normal cycle terminate and continue.
2	DTACK BERR HALT	A NA A/S	S NA S	Normal cycle terminate and halt. Continue when HALT negated.	Normal cycle terminate and halt. Continue when HALT negated.
3	DTACK BERR HALT	X A NA	X S NA	Terminate and take bus error trap.	Terminate and take bus error trap.
4	DTACK BERR HALT	A NA NA	S A NA	Normal cycle terminate and continue.	Terminate and take bus error trap.
5	DTACK BERR HALT	X A A/S	X S S	Terminate and retry when HALT removed.	Terminate and retry when HALT removed.
6	DTACK BERR HALT	A NA NA	S A A	Normal cycle terminate and continue.	Terminate and retry when 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 BERR and HALT under several conditions is shown in Table 5-6. (DTACK is assumed to be negated normally in all cases; for reliable operation, both DTACK and 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 BERR after timeout (case 3).

EXAMPLE B:

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

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

A signed divide (DIVS) or unsigned divide (DIVU) instruction forces an exception if a division operation is attempted with a divisor of zero.

6.3.6 Illegal and Unimplemented Instructions

Illegal instruction is the term used to refer to any of the word bit patterns that do not match the bit pattern of the first word of a legal M68000 instruction. If such an instruction is fetched, an illegal instruction exception occurs. Motorola reserves the right to define instructions using the opcodes of any of the illegal instructions. Three bit patterns always force an illegal instruction trap on all M68000-Family-compatible microprocessors. The patterns are: $4AFA, $4AFB, and $4AFC. Two of the patterns, $4AFA and $4AFB, are reserved for Motorola system products. The third pattern, $4AFC, is reserved for customer use (as the take illegal instruction trap (ILLEGAL) instruction).

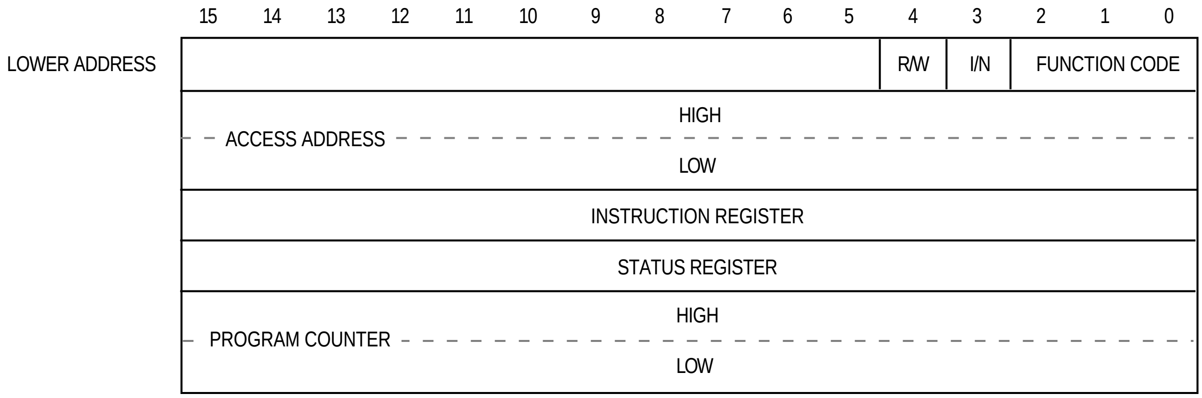
NOTE

In addition to the previously defined illegal instruction opcodes, the MC68010 defines eight breakpoint (BKPT) instructions with the bit patterns $4848–$484F. These instructions cause the processor to enter illegal instruction exception processing as usual. However, a breakpoint acknowledge bus cycle, in which the function code lines (FC2–FC0) are high and the address lines are all low, is also executed before the stacking operations are performed. The processor does not accept or send any data during this cycle. Whether the breakpoint acknowledge cycle is terminated with a $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, or $\overline{\text{VPA}}$ signal, the processor continues with the illegal instruction processing. The purpose of this cycle is to provide a software breakpoint that signals to external hardware when it is executed.

Word patterns with bits 15–12 equaling 1010 or 1111 are distinguished as unimplemented instructions, and separate exception vectors are assigned to these patterns to permit efficient emulation. Opcodes beginning with bit patterns equaling 1111 (line F) are implemented in the MC68020 and beyond as coprocessor instructions. These separate vectors allow the operating system to emulate unimplemented instructions in software.

Exception processing for illegal instructions is similar to that for traps. After the instruction is fetched and decoding is attempted, the processor determines that execution of an illegal instruction is being attempted and starts exception processing. The exception stack frame for group 2 is then pushed on the supervisor stack, and the illegal instruction vector is fetched.

protect memory contents from erroneous accesses. Only an external reset operation can restart a halted processor.



R/W (Read/Write): Write=0, Read=1. I/N (Instruction/Not): Instruction=0, Not=1

Figure 6-7. Supervisor Stack Order for Bus or Address Error Exception

6.3.9.2 BUS ERROR (MC68010). Exception processing for a bus error follows a slightly different sequence than the sequence for group 1 and 2 exceptions. In addition to the four steps executed during exception processing for all other exceptions, 22 words of additional information are placed on the stack. This additional information describes the internal state of the processor at the time of the bus error and is reloaded by the RTE instruction to continue the instruction that caused the error. Figure 6-8 shows the order of the stacked information.

Table 7-3. Move Word Instruction Execution Times

Source	Destination								
	Dn	An	(An)	(An)+	–(An)	(d_{16}, An)	(d_8, An, Xn)*	(xxx).W	(xxx).L
Dn	**8**(2/0)	**8**(2/0)	**16**(2/2)	**16**(2/2)	**16**(2/2)	**24**(4/2)	**26**(4/2)	**24**(4/2)	**32**(6/2)
An	**8**(2/0)	**8**(2/0)	**16**(2/2)	**16**(2/2)	**16**(2/2)	**24**(4/2)	**26**(4/2)	**24**(4/2)	**32**(6/2)
(An)	**16**(4/0)	**16**(4/0)	**24**(4/2)	**24**(4/2)	**24**(4/2)	**32**(6/2)	**34**(6/2)	**32**(6/2)	**40**(8/2)
(An)+	**16**(4/0)	**16**(4/0)	**24**(4/2)	**24**(4/2)	**24**(4/2)	**32**(6/2)	**34**(6/2)	**32**(6/2)	**40**(8/2)
–(An)	**18**(4/0)	**18**(4/0)	**26**(4/2)	**26**(4/2)	**26**(4/2)	**34**(6/2)	**32**(6/2)	**34**(6/2)	**42**(8/2)
(d_{16}, An)	**24**(6/0)	**24**(6/0)	**32**(6/2)	**32**(6/2)	**32**(6/2)	**40**(8/2)	**42**(8/2)	**40**(8/2)	**48**(10/2)
(d_8, An, Xn)*	**26**(6/0)	**26**(6/0)	**34**(6/2)	**34**(6/2)	**34**(6/2)	**42**(8/2)	**44**(8/2)	**42**(8/2)	**50**(10/2)
(xxx).W	**24**(6/0)	**24**(6/0)	**32**(6/2)	**32**(6/2)	**32**(6/2)	**40**(8/2)	**42**(8/2)	**40**(8/2)	**48**(10/2)
(xxx).L	**32**(8/0)	**32**(8/0)	**40**(8/2)	**40**(8/2)	**40**(8/2)	**48**(10/2)	**50**(10/2)	**48**(10/2)	**56**(12/2)
(d_{16}, PC)	**24**(6/0)	**24**(6/0)	**32**(6/2)	**32**(6/2)	**32**(6/2)	**40**(8/2)	**42**(8/2)	**40**(8/2)	**48**(10/2)
(d_8, PC, Xn)*	**26**(6/0)	**26**(6/0)	**34**(6/2)	**34**(6/2)	**34**(6/2)	**42**(8/2)	**44**(8/2)	**42**(8/2)	**50**(10/2)
#<data>	**16**(4/0)	**16**(4/0)	**24**(4/2)	**24**(4/2)	**24**(4/2)	**32**(6/2)	**34**(6/2)	**32**(6/2)	**40**(8/2)

*The size of the index register (Xn) does not affect execution time.

Table 7-4. Move Long Instruction Execution Times

Source	Destination								
	Dn	An	(An)	(An)+	–(An)	(d_{16}, An)	(d_8, An, Xn)*	(xxx).W	(xxx).L
Dn	**8**(2/0)	**8**(2/0)	**24**(2/4)	**24**(2/4)	**24**(2/4)	**32**(4/4)	**34**(4/4)	**32**(4/4)	**40**(6/4)
An	**8**(2/0)	**8**(2/0)	**24**(2/4)	**24**(2/4)	**24**(2/4)	**32**(4/4)	**34**(4/4)	**32**(4/4)	**40**(6/4)
(An)	**24**(6/0)	**24**(6/0)	**40**(6/4)	**40**(6/4)	**40**(6/4)	**48**(8/4)	**50**(8/4)	**48**(8/4)	**56**(10/4)
(An)+	**24**(6/0)	**24**(6/0)	**40**(6/4)	**40**(6/4)	**40**(6/4)	**48**(8/4)	**50**(8/4)	**48**(8/4)	**56**(10/4)
–(An)	**26**(6/0)	**26**(6/0)	**42**(6/4)	**42**(6/4)	**42**(6/4)	**50**(8/4)	**52**(8/4)	**50**(8/4)	**58**(10/4)
(d_{16}, An)	**32**(8/0)	**32**(8/0)	**48**(8/4)	**48**(8/4)	**48**(8/4)	**56**(10/4)	**58**(10/4)	**56**(10/4)	**64**(12/4)
(d_8, An, Xn)*	**34**(8/0)	**34**(8/0)	**50**(8/4)	**50**(8/4)	**50**(8/4)	**58**(10/4)	**60**(10/4)	**58**(10/4)	**66**(12/4)
(xxx).W	**32**(8/0)	**32**(8/0)	**48**(8/4)	**48**(8/4)	**48**(8/4)	**56**(10/4)	**58**(10/4)	**56**(10/4)	**64**(12/4)
(xxx).L	**40**(10/0)	**40**(10/0)	**56**(10/4)	**56**(10/4)	**56**(10/4)	**64**(12/4)	**66**(12/4)	**64**(12/4)	**72**(14/4)
(d_{16}, PC)	**32**(8/0)	**32**(8/0)	**48**(8/4)	**48**(8/4)	**48**(8/4)	**56**(10/4)	**58**(10/4)	**56**(10/4)	**64**(12/4)
(d_8, PC, Xn)*	**34**(8/0)	**34**(8/0)	**50**(8/4)	**50**(8/4)	**50**(8/4)	**58**(10/4)	**60**(10/4)	**58**(10/4)	**66**(12/4)
#<data>	**24**(6/0)	**24**(6/0)	**40**(6/4)	**40**(6/4)	**40**(6/4)	**48**(8/4)	**50**(8/4)	**48**(8/4)	**56**(10/4)

*The size of the index register (Xn) does not affect execution time.

7.3 STANDARD INSTRUCTION EXECUTION TIMES

The numbers of clock periods shown in Table 7-5 indicate the times required to perform the operations, store the results, and read the next instruction. 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 9-18. Miscellaneous Instruction Execution Times

Instruction	Size	Register	Memory	Register→ Destination**	Source**→ Register
ANDI to CCR	—	**16**(2/0)	—	—	—
ANDI to SR	—	**16**(2/0)	—	—	—
CHK	—	**8**(1/0)+	—	—	—
EORI to CCR	—	**16**(2/0)	—	—	—
EORI to SR	—	**16**(2/0)	—	—	—
EXG	—	**6**(1/0)	—	—	—
EXT	Word	**4**(1/0)	—	—	—
	Long	**4**(1/0)	—	—	—
LINK	—	**16**(2/2)	—	—	—
MOVE from CCR	—	**4**(1/0)	**8**(1/1)+*	—	
MOVE to CCR	—	**12**(2/0)	**12**(2/0)+	—	—
MOVE from SR	—	**4**(1/0)	**8**(1/1)+*	—	—
MOVE to SR	—	**12**(2/0)	**12**(2/0)+	—	—
MOVE from USP	—	**6**(1/0)	—	—	—
MOVE to USP	—	**6**(1/0)	—	—	—
MOVEC	—	—	—	**10**(2/0)	**12**(2/0)
MOVEP	Word	—	—	**16**(2/2)	**16**(4/0)
	Long	—	—	**24**(2/4)	**24**(6/0)
NOP	—	**4**(1/0)	—	—	—
ORI to CCR	—	**16**(2/0)	—	—	—
ORI to SR	—	**16**(2/0)	—	—	—
RESET	—	**130**(1/0)	—	—	—
RTD	—	**16**(4/0)	—	—	—
RTE	Short	**24**(6/0)	—	—	—
	Long, Retry Read	**112**(27/10)	—	—	—
	Long, Retry Write	**112**(26/1)	—	—	—
	Long, No Retry	**110**(26/0)	—	—	—
RTR	—	**20**(5/0)	—	—	—
RTS	—	**16**(4/0)	—	—	—
STOP	—	**4**(0/0)	—	—	—
SWAP	—	**4**(1/0)	—	—	—
TRAPV	—	**4**(1/0)	—	—	—
UNLK	—	**12**(3/0)	—	—	—

+Add effective address calculation time.
+Use nonfetching effective address calculation time.
**Source or destination is a memory location for the MOVEP instruction and a control register for the MOVEC instruction.

Table 10-1. Power Dissipation and Junction Temperature vs Temperature ($\theta_{JC}=\theta_{JA}$)

Package	T_A Range	θ_{JC} (°C/W)	P_D (W) @ T_A Min.	T_J (°C) @ T_A Min.	P_D (W) @ T_A Max.	T_J (°C) @ T_A Max.
L/LC	0°C to 70°C -40°C to 85°C 0°C to 85°C	15 15 15	1.5 1.7 1.5	23 -14 23	1.2 1.2 1.2	88 103 103
P	0°C to 70°C	15	1.5	23	1.2	88
R/RC	0°C to 70°C -40°C to 85°C 0°C to 85°C	15 15 15	1.5 1.7 1.5	23 -14 23	1.2 1.2 1.2	88 103 103
FN	0°C to 70°C	25	1.5	38	1.2	101

NOTE: Table does not include values for the MC68000 12F.
Does not apply to the MC68HC000, MC68HC001, and MC68EC000.

Table 10-2. Power Dissipation and Junction Temperature vs Temperature ($\theta_{JC}\neq\theta_{JC}$)

Package	T_A Range	θ_{JA} (°C/W)	P_D (W) @ T_A Min.	T_J (°C) @ T_A Min.	P_D (W) @ T_A Max.	T_J (°C) @ T_A Max.
L/LC	0°C to 70°C -40°C to 85°C 0°C to 85°C	30 30 30	1.5 1.7 1.5	23 -14 23	1.2 1.2 1.2	88 103 103
P	0°C to 70°C	30	1.5	23	1.2	88
R/RC	0°C to 70°C -40°C to 85°C 0°C to 85°C	33 33 33	1.5 1.7 1.5	23 -14 23	1.2 1.2 1.2	88 103 103
FN	0°C to 70°C	40	1.5	38	1.2	101

NOTE: Table does not include values for the MC68000 12F.
Does not apply to the MC68HC000, MC68HC001, and MC68EC000.

Values for thermal resistance presented in this manual, unless estimated, were derived using the procedure described in Motorola Reliability Report 7843 “Thermal Resistance Measurement Method for MC68XXX Microcomponent Devices”' and are provided for design purposes only. Thermal measurements are complex and dependent on procedure and setup. User-derived values for thermal resistance may differ.

10.4 CMOS CONSIDERATIONS

The MC68HC000, MC68HC001, and MC68EC000, with it significantly lower power consumption, has other considerations. The CMOS cell is basically composed of two complementary transistors (a P channel and an N channel), and only one transistor is turned on while the cell is in the steady state. The active P-channel transistor sources current when the output is a logic high and presents a high impedance when the output is logic low. Thus, the overall result is extremely low power consumption because no power

10.6 MC68000/68008/68010 DC ELECTRICAL CHARACTERISTICS

(V_{CC}=5.0 VDC±5%; GND=0 VDC; T_A=T_L TO T_H)

Characteristic		Symbol	Min	Max	Unit
Input High Voltage		V_{IH}	2.0	V_{CC}	V
Input Low Voltage		V_{IL}	GND-0.3	0.8	V
Input Leakage Current @ 5.25 V	$\overline{BERR}$, $\overline{BGACK}$, $\overline{BR}$, $\overline{DTACK}$, CLK, $\overline{IPL0}$—$\overline{IPL2}$, $\overline{VPA}$ $\overline{HALT}$, $\overline{RESET}$	I_{IN}	— —	2.5 20	µA
Three-State (Off State) Input Current @ 2.4 V/0.4 V	$\overline{AS}$, A1—A23, D0—D15, FC0—FC2, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$	I_{TSI}	—	20	µA
Output High Voltage (I_{OH} = –400 µA) (I_{OH} = -400 µA)	E* $\overline{AS}$, A1–A23, $\overline{BG}$, D0–D15, FC0–FC2, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$	V_{OH}	V_{CC} –0.75 2.4	— 2.4	V
Output Low Voltage (I_{OL}= 1.6 mA) (I_{OL} = 3.2 mA) (I_{OL} = 5.0 mA) (I_{OL} = 5.3 mA)	 $\overline{HALT}$ A1—A23, $\overline{BG}$, FC0-FC2 $\overline{RESET}$ E, $\overline{AS}$, D0—D15, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$	V_{OL}	 — — — —	 0.5 0.5 0.5 0.5	V
Power Dissipation (see POWER CONSIDERATIONS)		P_D***	—	—	W
Capacitance (V_{in}=0 V, T_A=25°C, Frequency=1 MHz)**		C_{in}	—	20.0	pF
Load Capacitance	$\overline{HALT}$ All Others	C_L	— —	70 130	pF

*With external pullup resistor of 1.1 Ω.

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

***During normal operation, instantaneous V_{CC} current requirements may be as high as 1.5 A.

10.12 AC ELECTRICAL SPECIFICATIONS — BUS ARBITRATION (V_{CC}=5.0 VDC±5%; GND=0 VDC, T_A=T_L TO T_H; See Figure s 10-7 – 10-11) (Applies To All Processors Except The MC68EC000)

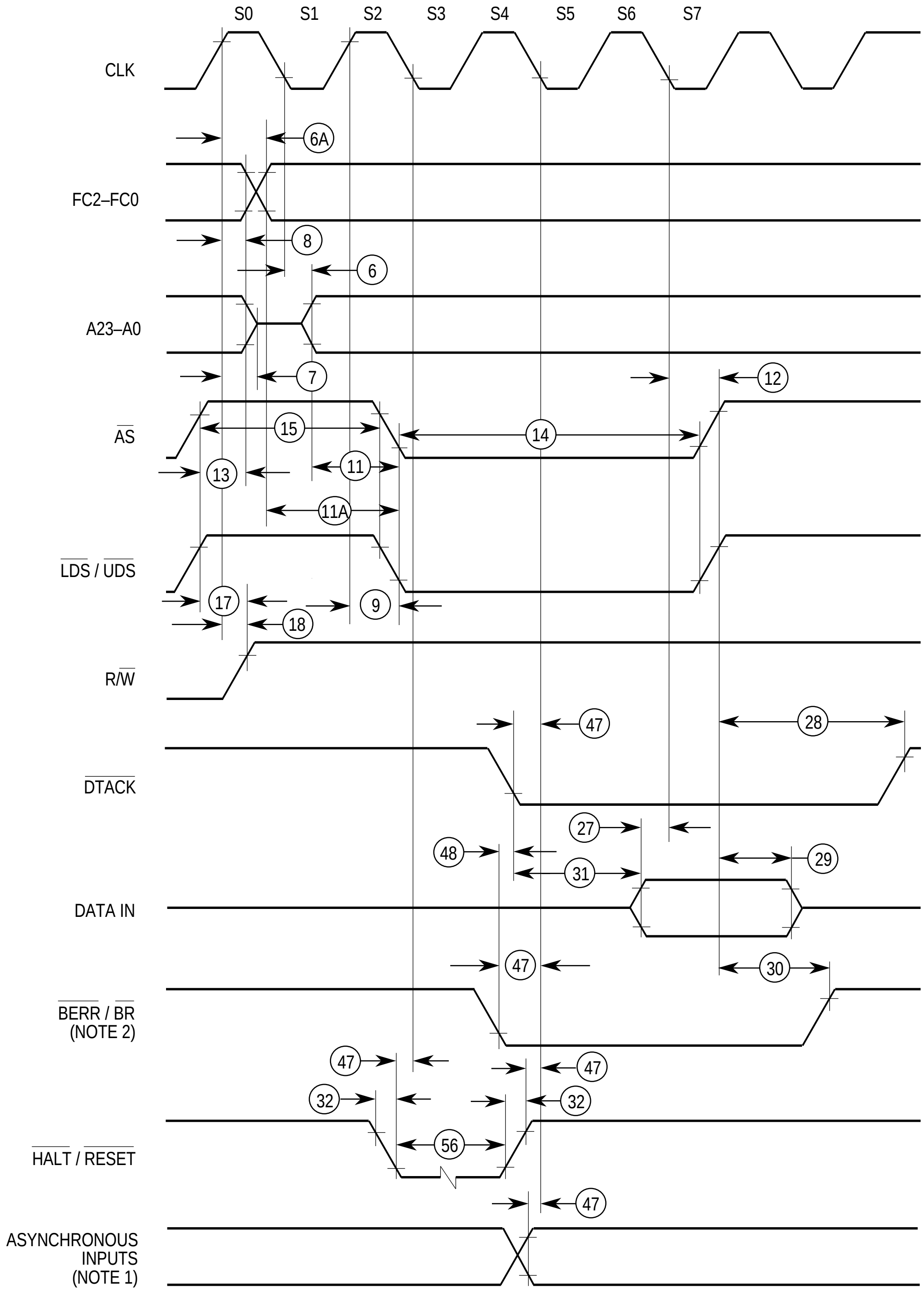
Num	Characteristic	8 MHz*		10 MHz*		12.5 MHz*		16.67 MHz 12F		16 MHz		20 MHz**		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
7	Clock High to Address, Data Bus High Impedance (Maximum)	—	80	—	70	—	60	—	50	—	50	—	42	ns
16	Clock High to Control Bus High Impedance	—	80	—	70	—	60	—	50	—	50	—	42	ns
33	Clock High to $\overline{BG}$ Asserted	—	62	—	50	—	40	0	40	0	30	0	25	ns
34	Clock High to $\overline{BG}$ Negated	—	62	—	50	—	40	0	40	0	30	0	25	ns
35	$\overline{BR}$ Asserted to $\overline{BG}$ Asserted	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	Clks
36[1]	$\overline{BR}$ Negated to $\overline{BG}$ Negated	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	Clks
37	$\overline{BGACK}$ Asserted to $\overline{BG}$ Negated	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	Clks
37A[2]	$\overline{BGACK}$ Asserted to $\overline{BR}$ Negated	20	1.5 Clks	20	1.5 Clks	20	1.5 Clks	10	1.5 Clks	10	1.5 Clks	10	1.5 Clks	Clks/ ns
38	$\overline{BG}$ Asserted to Control, Address, Data Bus High Impedance ($\overline{AS}$ Negated)		80		70		60	—	50	—	50	—	42	ns
39	$\overline{BG}$ Width Negated	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	Clks
46	$\overline{BGACK}$ Width Low	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	Clks
47	Asynchronous Input Setup Time	10	—	10	—	10	—	5	—	5	—	5	—	ns
57	$\overline{BGACK}$ Negated to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ Driven	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	Clks
57A	$\overline{BGACK}$ Negated to FC, $\overline{VMA}$ Driven	1	—	1	—	1	—	1	—	1	—	1	—	Clks
58[1]	$\overline{BR}$ Negated to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ Driven	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	Clks
58A[1]	$\overline{BR}$ Negated to FC, $\overline{VMA}$ Driven	1	—	1	—	1	—	1	—	1	—	1	—	Clks

*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.

** Applies only to the MC68HC000 and MC68HC001.

NOTES:

1. Setup time for the synchronous inputs $\overline{BGACK}$, $\overline{IPL0}$-$\overline{IPL2}$, and $\overline{VPA}$ guarantees their recognition at the next falling edge of the clock.
2. $\overline{BR}$ need fall at this time only in order to insure being recognized at the end of the bus cycle.
3. Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be lienar between 0.8 volt and 2.0 volts.
4. The processor will negate $\overline{BG}$ and begin driving the bus again if external arbitration logic negates $\overline{BR}$ before asserting $\overline{BGACK}$.
5. The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{BG}$ may be reasserted.



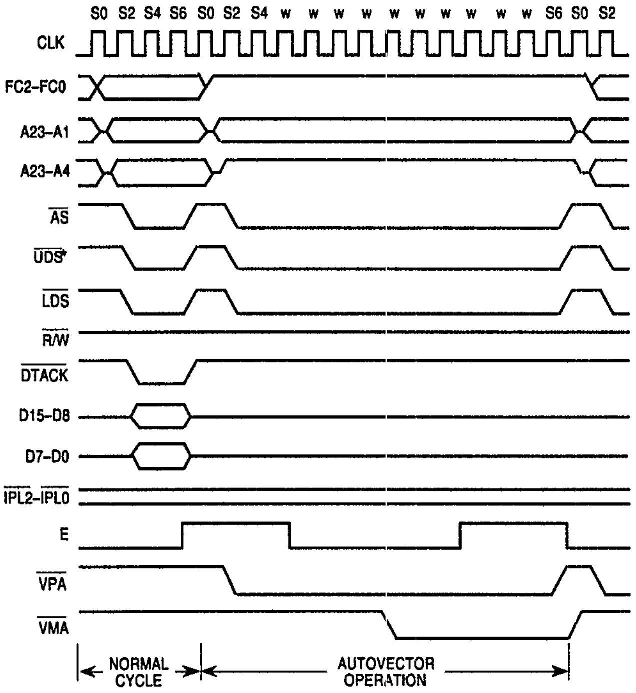
NOTES:

1. Setup time for the asynchronous inputs IPL2–IPL0 and AVEC (#47) guarantees their recognition at the next falling edge of the clock.
2. BR need fall at this time only to insure being recognized at the end of the bus cycle.
3. Timing measurements are referenced to and from a low voltage of 0.8 V and a high voltage of 2.0 V, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall is linear between 0.8 V and 2.0 V.

Figure 10-12. MC68EC000 Read Cycle Timing Diagram

interrupt service routine can be located anywhere within the supervisor program address space because the user assigns the vectors in the vector table.

Since $\overline{\text{VMA}}$ is asserted during an autovector operation, care should be taken to prevent an unintended access to the device. An unintended access could occur if the peripheral address were on the address bus during the autovector operation.



* generated internally.

Figure B-6. Autovector Operation Timing Diagram