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	40MHz
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=mc68ec040rc40a

The user programming model includes eight data registers, seven address registers, and a stack pointer register. The address registers and stack pointer can be used as base address registers or software stack pointers, and any of the 16 registers can be used as index registers. Two control registers are available in the user mode—the program counter (PC), which usually contains the address of the instruction that the MC68040 is executing, and the lower byte of the SR, which is accessible as the condition code register (CCR). The CCR contains the condition codes that reflect the results of a previous operation and can be used for conditional instruction execution in a program.

The supervisor programming model includes the upper byte of the SR, which contains operation control information. The vector base register (VBR) contains the base address of the exception vector table, which is used in exception processing. The source function code (SFC) and destination function code (DFC) registers contain 3-bit function codes. These function codes can be considered extensions to the 32-bit logical address. The processor automatically generates function codes to select address spaces for data and program accesses in the user and supervisor modes. Some instructions use the alternate function code registers to specify the function codes for various operations.

The cache control register (CACR) controls enabling of the on-chip instruction and data caches of the MC68040. The supervisor root pointer (SRP) and user root pointer (URP) registers point to the root of the address translation table tree to be used for supervisor and user mode accesses.

The translation control register (TCR) enables logical-to-physical address translation and selects either 4- or 8-Kbyte page sizes. There are four transparent translation registers, two for instruction accesses and two for data accesses. These registers allow portions of the logical address space to be transparently mapped and accessed without the use of resident descriptors in an ATC. The MMU status register (MMUSR) contains status information derived from the execution of a PTEST instruction. The PTEST instruction searches the translation tables for the logical address, specified by this instruction's effective address field and the DFC, and returns status information corresponding to the translation.

The user programming model can also access the entire floating-point programming model. The eight 80-bit floating-point data registers are analogous to the integer data registers. A 32-bit floating-point control register (FPCR) contains an exception enable byte that enables and disables traps for each class of floating-point exceptions and a mode byte that sets the user-selectable rounding and precision modes. A floating-point status register (FPSR) contains a condition code byte, quotient byte, exception status byte, and accrued exception byte. A floating-point exception handler can use the address in the 32-bit floating-point instruction address register (FPIAR) to locate the floating-point instruction that has caused an exception. Instructions that do not modify the FPIAR can be used to read the FPIAR in the exception handler without changing the previous value.

Table 1-3. Notational Conventions (Continued)

Data Format And Type	
+ inf	Positive Infinity
<fmt>	Operand Data Format: Byte (B), Word (W), Long (L), Single (S), Double (D), Extended (X), or Packed (P).
B, W, L	Specifies a signed integer data type (twos complement) of byte, word, or long word.
D	Double-precision real data format (64 bits).
k	A twos complement signed integer (–64 to +17) specifying a number's format to be stored in the packed decimal format.
P	Packed BCD real data format (96 bits, 12 bytes).
S	Single-precision real data format (32 bits).
X	Extended-precision real data format (96 bits, 16 bits unused).
– inf	Negative Infinity
Subfields and Qualifiers	
#<xxx> or #<data>	Immediate data following the instruction word(s).
( )	Identifies an indirect address in a register.
[ ]	Identifies an indirect address in memory.
bd	Base Displacement
ccc	Index into the MC68881/MC68882 Constant ROM
d_n	Displacement Value, n Bits Wide (example: d_{16} is a 16-bit displacement).
LSB	Least Significant Bit
LSW	Least Significant Word
MSB	Most Significant Bit
MSW	Most Significant Word
od	Outer Displacement
SCALE	A scale factor (1, 2, 4, or 8, for no-word, word, long-word, or quad-word scaling, respectively).
SIZE	The index register's size (W for word, L for long word).
{offset:width}	Bit field selection.
Register Names	
CCR	Condition Code Register (lower byte of status register)
DFC	Destination Function Code Register
FPcr	Any Floating-Point System Control Register (FPCR, FPSR, or FPIAR)
FPm, FPn	Any Floating-Point Data Register specified as the source or destination, respectively.
IC, DC, IC/DC	Instruction, Data, or Both Caches
MMUSR	MMU Status Register
PC	Program Counter
Rc	Any Non Floating-Point Control Register
SFC	Source Function Code Register
SR	Status Register

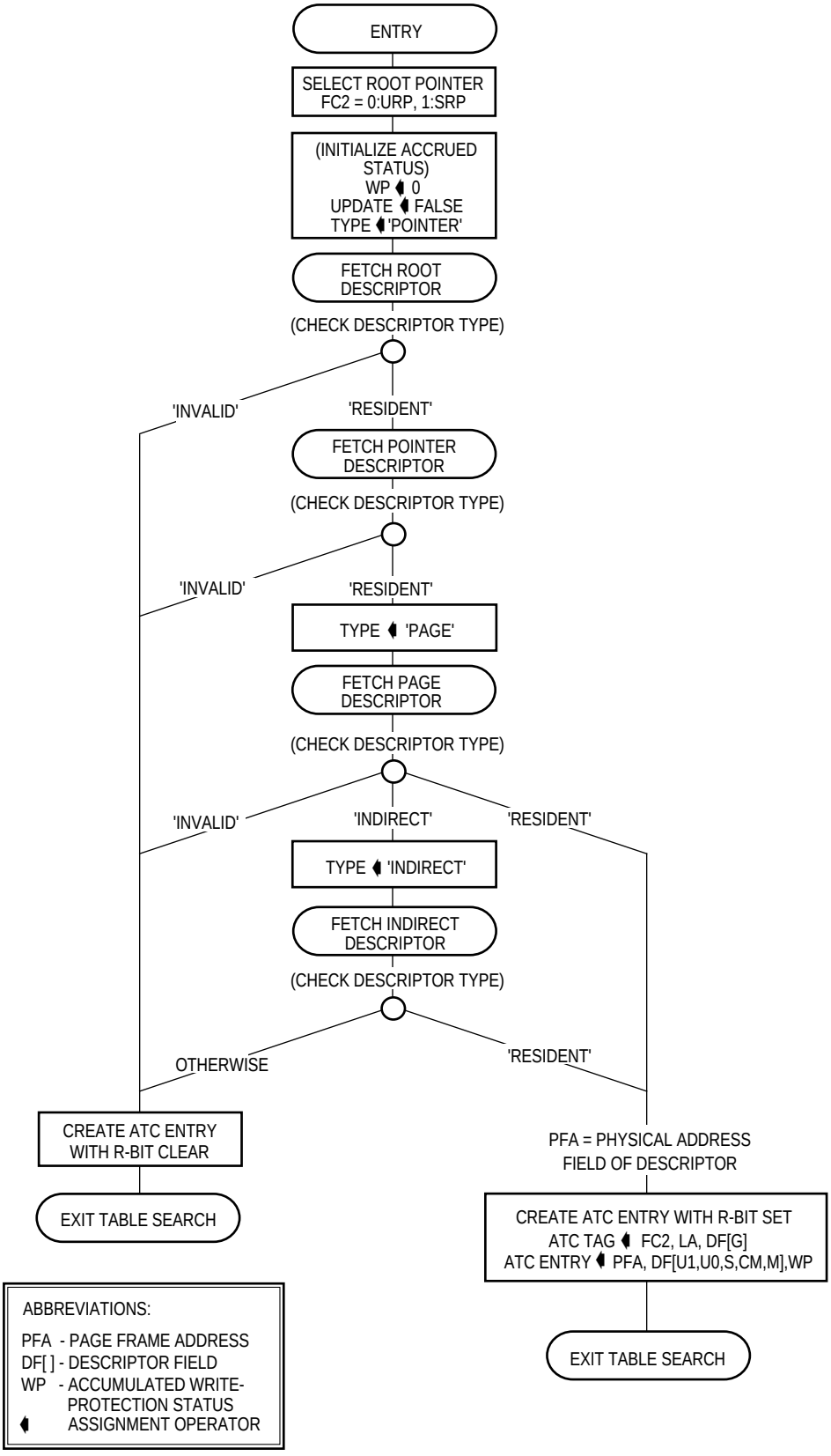


Figure 3-9. Detailed Flowchart of Table Search Operation

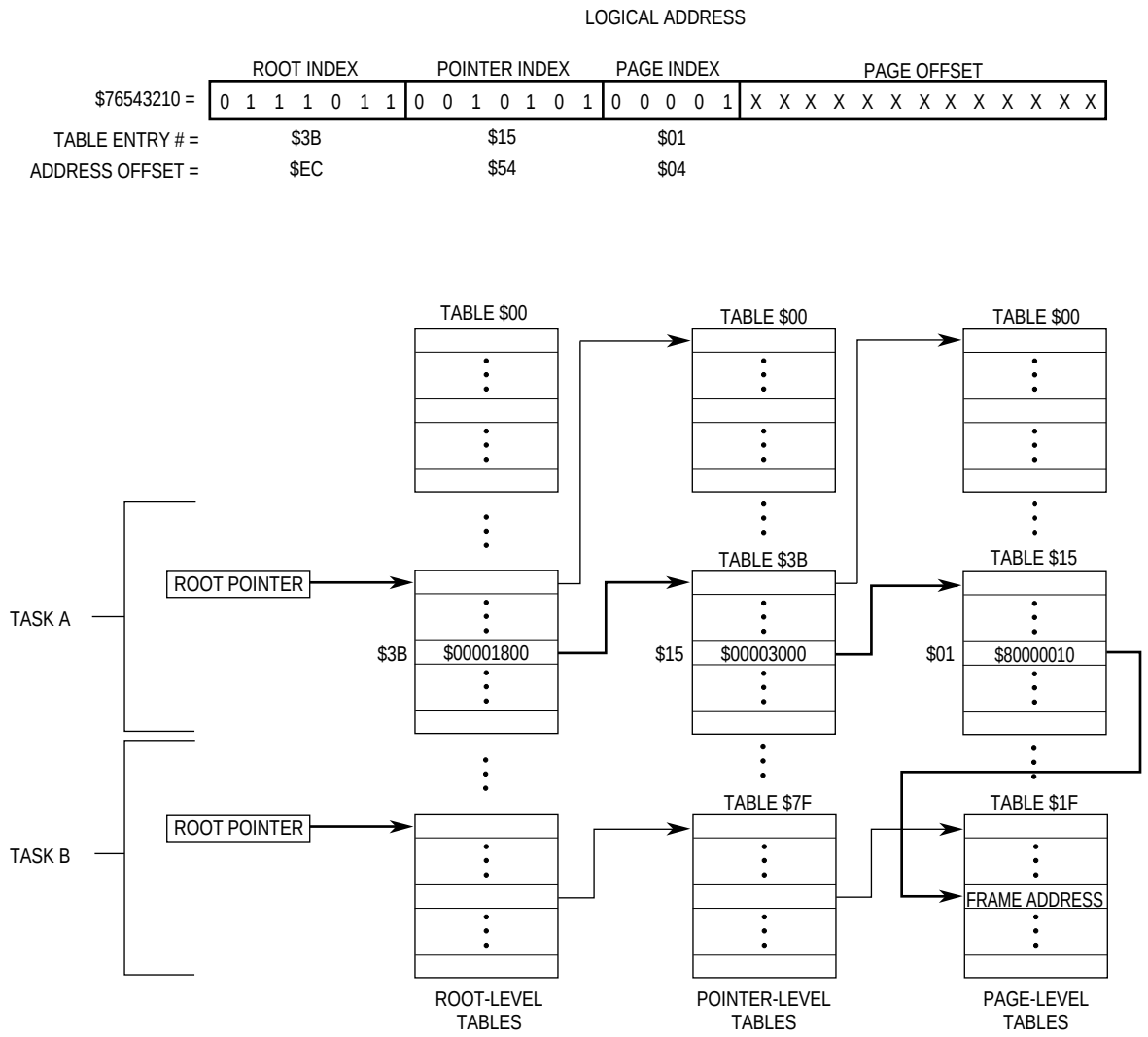


Figure 3-14. Translation Table Using Indirect Descriptors

3.2.4.2 TABLE SHARING BETWEEN TASKS. More than one task can share a pointer- or page-level table by placing a pointer to a shared table in the address translation tables. The upper (nonshared) tables can contain different write-protected settings, allowing different tasks to use the memory areas with different write permissions. In Figure 3-15, two tasks share the memory translated by the table at the pointer table level. Task A cannot write to the shared area; task B, however, has the W-bit clear in its pointer to the shared table so that it can read and write the shared area. Also, the shared area appears at different logical addresses for each task. Figure 3-15 illustrates shared tables in a translation table structure.

3.2.6 Address Translation Protection

The M68040 MMUs provide separate translation tables for supervisor and user address spaces. The translation tables contain both mapping and protection information. Each table and page descriptor includes a write-protect (W) bit that can be set to provide write protection at any level. Page descriptors also contain a supervisor-only (S) bit that can limit access to programs operating at the supervisor privilege level.

The protection mechanisms can be used individually or in any combination to protect:

- Supervisor address space from accesses by user programs.
- User address space from accesses by other user programs.
- Supervisor and user program spaces from write accesses (implicitly supported by designating all memory pages used for program storage as write protected).
- One or more pages of memory from write accesses.

3.2.6.1 SUPERVISOR AND USER TRANSLATION TABLES. One way of protecting supervisor and user address spaces from unauthorized accesses is to use separate supervisor and user translation tables. Separate trees protect supervisor programs and data from accesses by user programs and user programs and data from access by supervisor programs. Access is granted to the supervisor programs that can accesses any area of memory with MOVES. The translation table pointed to by the SRP is selected for all other supervisor mode accesses. This translation table can be common to all tasks. Figure 3-17 illustrates separate translation tables for supervisor accesses and for two user tasks that share the common supervisor space. Each user task has an translation table with unique mappings for the logical addresses in its user address space.

3.2.6.2 SUPERVISOR ONLY. A second mechanism protects supervisor programs and data without requiring segmenting of the logical address space into supervisor and user address spaces. Page descriptors contain S-bits to protect areas of memory from access by user programs. When a table search for a user access encounters an S-bit set in a page descriptor, the table search ends, and an ATC descriptor corresponding to the logical address is created with the S-bit set. A subsequent retry of the user access results in an access error exception being taken. The S-bit can be used to protect one or more pages from user program access. Supervisor and user mode accesses can share descriptors by using indirect descriptors or by sharing tables. The entire user and supervisor address spaces can be mapped together by loading the same root pointer address into both the SRP and URP registers.

5.12 TEST SIGNALS

The M68040 includes dedicated user-accessible test logic that is fully compatible with the IEEE 1149.1 *Standard Test Access Port and Boundary Scan Architecture*. Problems associated with testing high-density circuit boards have led to the development of this standard under the IEEE Test Technology Committee and Joint Test Action Group (JTAG) sponsorship. The M68040 implementation supports circuit board test strategies based on this standard. However, the JTAG interface is not intended to provide an in-circuit test to verify M68040 operations; therefore, it is impossible to test M68040 operations using this interface. **Section 6 IEEE 1149.1 Test Access Port (JTAG)** describes the M68040 implementation of the IEEE 1149.1 and is intended to be used with the supporting IEEE document.

5.12.1 Test Clock (TCK)

This input signal is used as a dedicated clock for the test logic. Since clocking of the test logic is independent of the normal operation of the MC68040, several other components on a board can share a common test clock with the processor even though each component may operate from a different system clock. The design of the test logic allows the test clock to run at low frequencies, or to be gated off entirely as required for test purposes.

5.12.2 Test Mode Select (TMS)

This input signal is decoded by the TAP controller and distinguishes the principle operationas of the test support circuitry.

5.12.3 Test Data In (TDI)

This input signal provides a serial data input to the TAP.

5.12.4 Test Data Out (TDO)

This three-state output signal provides a serial data output from the TAP. The TDO output can be placed in a high-impedance mode to allow parallel connection of board-level test data paths.

5.12.5 Test Reset ($\overline{TRST}$)—Not on MC68040V and MC68EC040V

This input signal provides an asynchronous reset of the TAP controller.

5.13 POWER SUPPLY CONNECTIONS

The M68040 requires connection to a V_{CC} power supply, positive with respect to ground. The V_{CC} and ground connections are grouped to supply adequate current to the various sections of the processor. **Section 12 Ordering Information and Mechanical Data** describes the groupings of V_{CC} and ground connections.

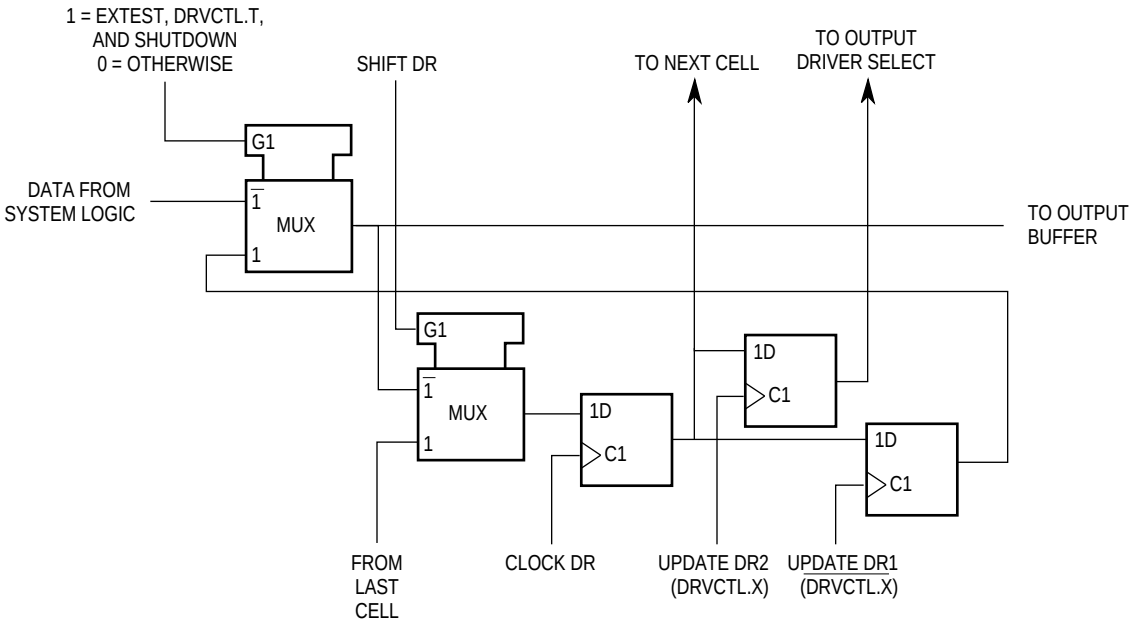


Figure 6-3. Output Latch Cell (O.Latch)

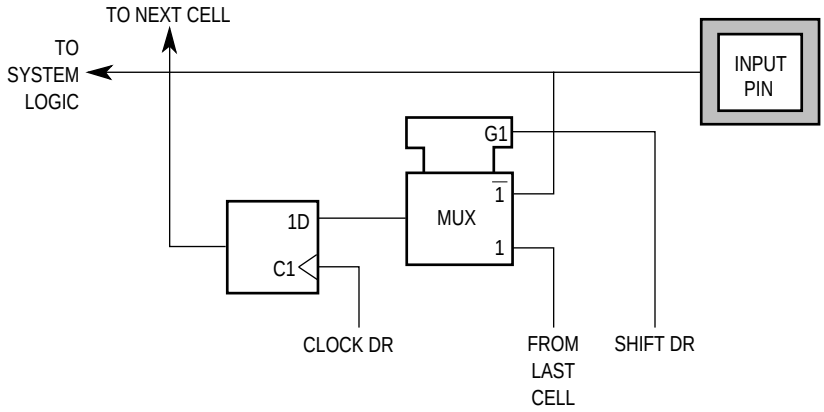


Figure 6-4. Input Pin Cell (I.Pin)

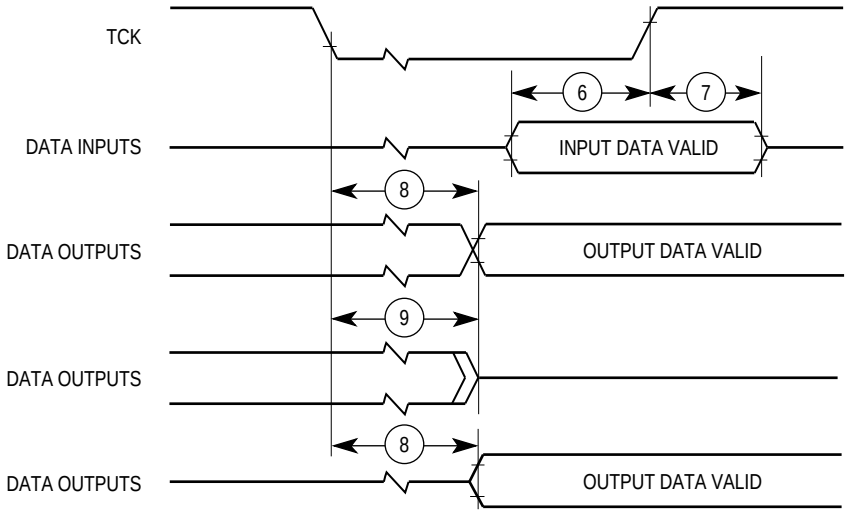


Figure 6-10. Boundary Scan Timing Diagram

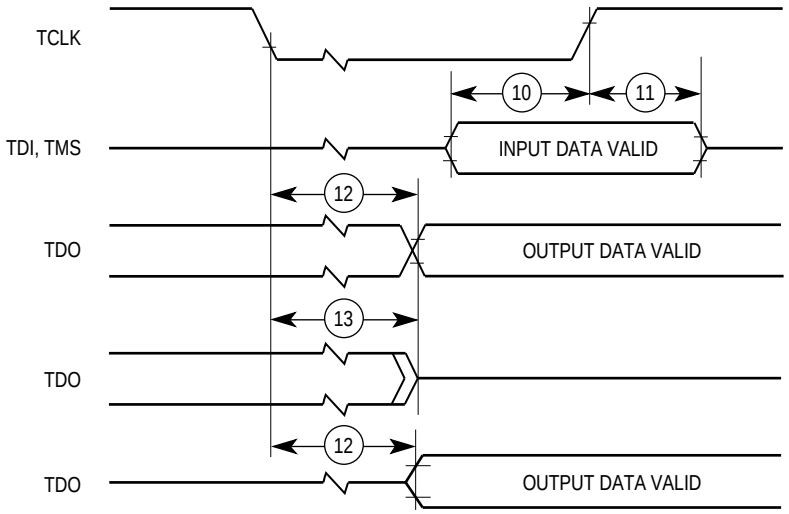


Figure 6-11. Test Access Port Timing Diagram

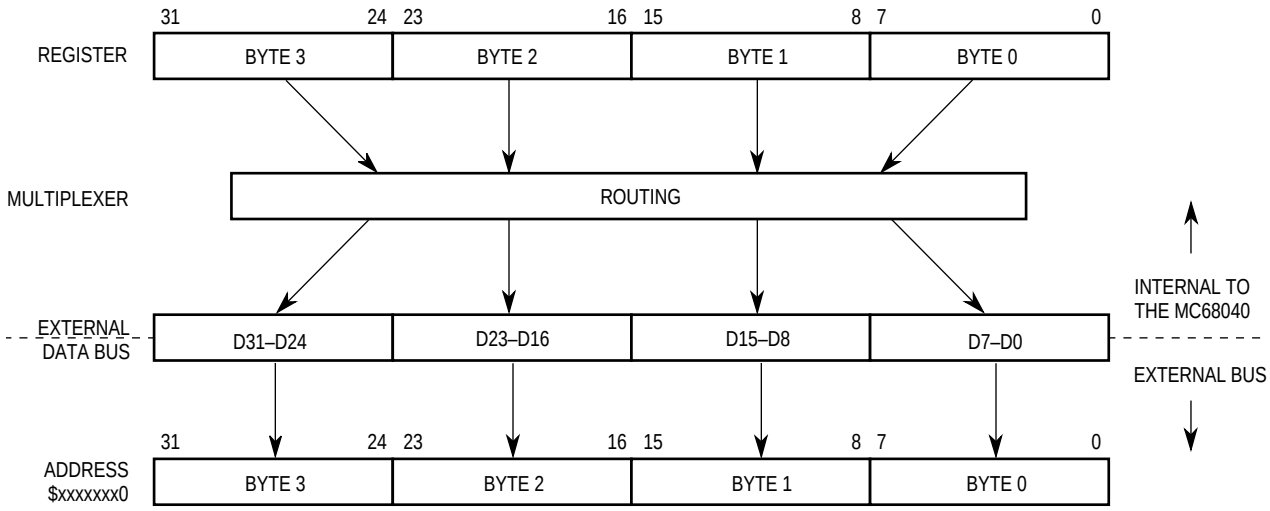


Figure 7-3. Data Multiplexing

Table 7-1 lists the combinations of the SIZx, A1, and A0 signals, collectively called byte enable signals, that are used for each of the four sections of the data bus. In the table, BYTEn indicates the data bus section that is active, the portion of the requested operand that is read or written during that bus transfer. For line transfers, all bytes are valid as listed and can correspond to portions of the requested operand or to data required to fill the remainder of the cache line. The bytes labeled with a dash are not required; they are ignored on read transfers and driven with undefined data on write transfers. Not selecting these bytes prevents incorrect accesses in sensitive areas such as I/O devices. Figure 7-4 illustrates a logic diagram for one method for generating byte enable signals from the SIZx, A1, and A0 and the associated PAL equation. These byte enable signals can be combined with the address decode logic.

Table 7-1. Data Bus Requirements for Read and Write Cycles

Transfer Size	Signal Encodings				Active Data Bus Sections			
	SIZ1	SIZ0	A1	A0	D31–D24	D23–D16	D15–D8	D7–D0
Byte	0	1	0	0	BYTEn	—	—	—
	0	1	0	1	—	BYTEn	—	—
	0	1	1	0	—	—	BYTEn	—
	0	1	1	1	—	—	—	BYTEn
Word	1	0	0	0	BYTEn	BYTEn	—	—
	1	0	1	0	—	—	BYTEn	BYTEn
Long Word	0	0	X	X	BYTEn	BYTE n	BYTEn	BYTEn
Line	1	1	X	X	BYTEn	BYTEn	BYTEn	BYTEn

Clock 5 (C5)

This clock is identical to C3 except that once $\overline{TA}$ is recognized, the latched value corresponds to the third long word of data for the burst. After the processor recognizes the last $\overline{TA}$ assertion and terminates the line read bus cycle, $\overline{TIP}$ remains asserted if the processor is ready to begin another bus cycle. Otherwise, the processor negates $\overline{TIP}$ during the first half of the next clock.

Figures 7-12 and 7-13 illustrate a flowchart and functional timing diagram for a burst-inhibited line read bus cycle.

MC68LC040 memory units and **Appendix B MC68EC040** for information on the MC68EC040 memory unit.

The processor asserts $\overline{TS}$ during C1 to indicate the beginning of a bus cycle. If not already asserted from a previous bus cycle, the $\overline{TIP}$ signal is also asserted at this time to indicate that a bus cycle is active.

Clock 2 (C2)

During the first half of the first clock after C1, the processor negates $\overline{TS}$ and drives the data bus with the data to be written. The selected device uses R/$\overline{W}$, SIZ1, and SIZ0 to latch the data on the data bus. Concurrently, the selected device asserts $\overline{TA}$ and either negates or asserts $\overline{TBI}$ to indicate it can or cannot support a burst transfer. At the end of the first clock after C1, the processor samples the level of $\overline{TA}$ and $\overline{TBI}$. If $\overline{TA}$ is asserted, the transfer terminates. If $\overline{TA}$ is not recognized asserted, the processor inserts wait states instead of terminating the transfer. The processor continues to sample $\overline{TA}$ and $\overline{TBI}$ on successive rising edges of BCLK until $\overline{TA}$ is recognized asserted.

If $\overline{TBI}$ was negated with $\overline{TA}$, the processor continues the cycle with C3. Otherwise, if $\overline{TBI}$ was asserted, the line transfer is burst inhibited, and the processor writes the remaining three long words using long-word write bus cycles. Only in this case does the processor increment A3 and A2 for each write, and the new address is placed on the address bus for each bus cycle. Refer to **7.4.3 Byte, Word, and Long-Word Write Transfers** for information on long-word writes. If no waits states are generated, a burst-inhibited line write completes in eight clocks instead of the five required for a burst write.

Clock 3 (C3)

The processor drives the second long word of data on the data bus and holds the address and transfer attribute signals constant during C3. The selected device increments A3 and A2 to reference the next long word, latches this data from the data bus, and asserts $\overline{TA}$. At the end of C3, the processor samples the level of $\overline{TA}$; if $\overline{TA}$ is asserted, the transfer terminates. If $\overline{TA}$ is not recognized asserted at the end of C3, the processor inserts wait states instead of terminating the transfer. The processor continues to sample $\overline{TA}$ on successive rising edges of BCLK until $\overline{TA}$ is recognized asserted.

Clock 4 (C4)

This clock is identical to C3 except that the value driven on the data bus corresponds to the third long word of data for the burst.

Clock 5 (C5)

This clock is identical to C3 except that the value driven on the data bus corresponds to the fourth long word of data for the burst. After the processor recognizes the last $\overline{TA}$ assertion and terminates the line write bus cycle, $\overline{TIP}$ remains asserted if the processor is ready to begin another bus cycle. Otherwise, the processor negates $\overline{TIP}$ during the first half of the next clock. The processor also three-states the data bus during the first half of the next clock following termination of the write cycle.

7.4.5 Read-Modify-Write Transfers (Locked Transfers)

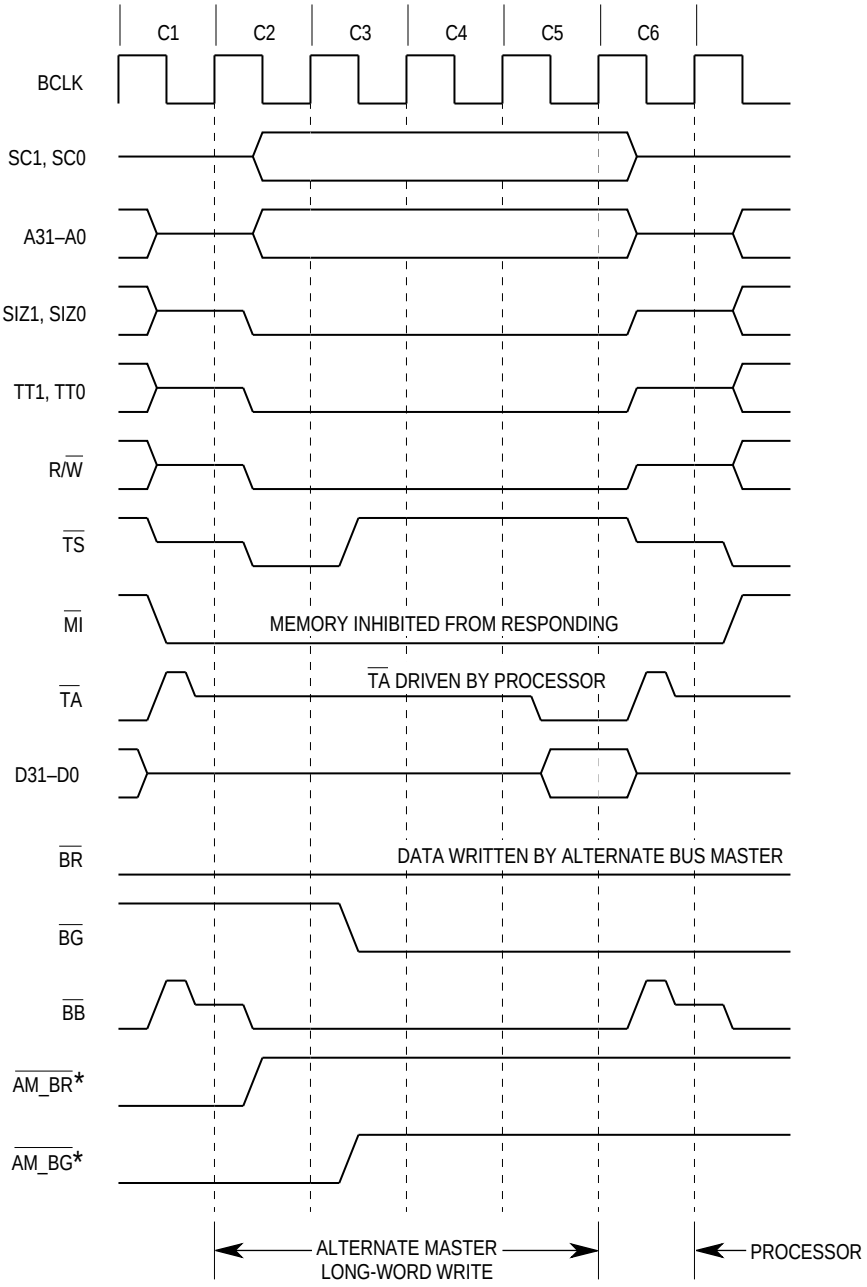
The read-modify-write transfer performs a read, conditionally modifies the data in the processor, and writes the data out to memory. In the M68040, this operation can be indivisible, providing semaphore capabilities for multiprocessor systems. During the entire read-modify-write sequence, the M68040 asserts the $\overline{\text{LOCK}}$ signal to indicate that an indivisible operation is occurring and asserts the $\overline{\text{LOCKE}}$ signal for the last transfer to indicate completion of the locked sequence. The external arbiter can use the $\overline{\text{LOCK}}$ and $\overline{\text{LOCKE}}$ signals to prevent arbitration of the bus during locked processor sequences. External bus arbitrations can use $\overline{\text{LOCKE}}$ to support bus arbitration between consecutive read-modify-write cycles. A read-modify-write operation is treated as noncachable. If the access hits in the data cache, it invalidates a matching valid entry and pushes a matching dirty entry. The read-modify-write transfer begins after the line push (if required) is complete; however, $\overline{\text{LOCK}}$ may assert during the line push bus cycle.

The TAS, CAS, and CAS2 instructions are the only M68040 instructions that utilize read-modify-write transfers. Some page descriptor updates during translation table searches also use read-modify-write transfers. Refer to **Section 3 Memory Management Unit (Except MC68EC040 and MC68EC040V)** for information about table searches.

The read-modify-write transfer for the CAS and CAS2 instructions in the M68040 differs from those used by previous members of the M68000 family. If an operand does not match one of these instructions, the M68040 still executes a single write transfer to terminate the locked sequence with $\overline{\text{LOCKE}}$ asserted. For the CAS instruction, the value read from memory is written back; for the CAS2 instruction, the second operand read is written back. Figure 7-18 illustrates a functional timing diagram for a TAS instruction read-modify-write bus transfer.

Clock 1 (C1)

The read cycle starts in C1. During the first half of C1, the processor places valid values on the address bus and transfer attributes. $\overline{\text{LOCK}}$ is asserted to identify a locked read-modify-write bus cycle. For user and supervisor mode accesses, which the corresponding memory unit translates, the UPAx signals are driven with the values from the matching U1 and U0 bits. The TTx and TMx signals identify the specific access type. R/$\overline{\text{W}}$ is driven high for a read cycle. $\overline{\text{CIOUT}}$ is asserted if the access is identified as noncachable. The processor asserts $\overline{\text{TS}}$ during C1 to indicate the beginning of a bus cycle. If not already asserted from a previous bus cycle, the $\overline{\text{TIP}}$ signal is also asserted at this time to indicate that a bus cycle is active. Refer to **Section 3 Memory Management Unit (Except MC68EC040 and MC68EC040V)** for information on the M68040 and MC68LC040 memory units and **Appendix B MC68EC040** for information on the MC68EC040 memory unit.



*AM indicates the alternate bus master.

Figure 7-43. Snooped Long-Word Write, Memory Inhibited

7.10 RESET OPERATION

An external device asserts the reset input signal ($\overline{RSTI}$) to reset the processor. When power is applied to the system, external circuitry should assert $\overline{RSTI}$ for a minimum of 10 BCLK cycles after V_{CC} is within tolerance. Figure 7-44 is a functional timing diagram of the power-on reset operation, illustrating the relationships among V_{CC}, $\overline{RSTI}$, mode selects, and bus signals. The BCLK and PCLK clock signals are required to be stable by the time V_{CC} reaches the minimum operating specification. The V_{IH} levels of the clocks

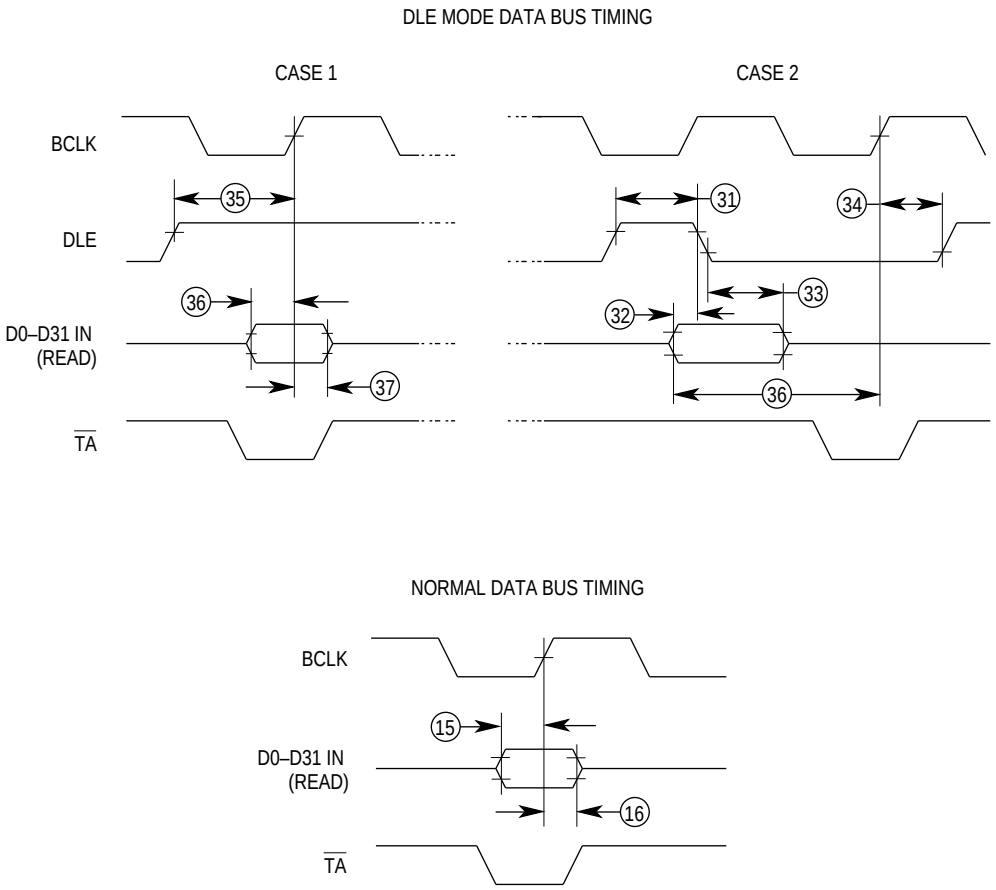


Figure 7-48. DLE versus Normal Data Read Timing

Case 1

If DLE is negated and meets setup time specification #35 to the rising edge of BCLK when the bus read is terminated, latch A is transparent, and the read data must meet setup and hold time specifications #36 and #37 to the rising edge of BCLK. Read timing is similar to normal timing for this case.

Case 2

If DLE is asserted, the data bus levels are latched and held internally. D31–D0 must meet setup and hold time specifications #32 and #33 to the falling edge of DLE, and can transition to a new level once DLE is asserted. D31–D0 must still meet setup time specification #36 to BCLK, but not hold time specification #37, since the data is internally held valid as long as DLE remains asserted low.

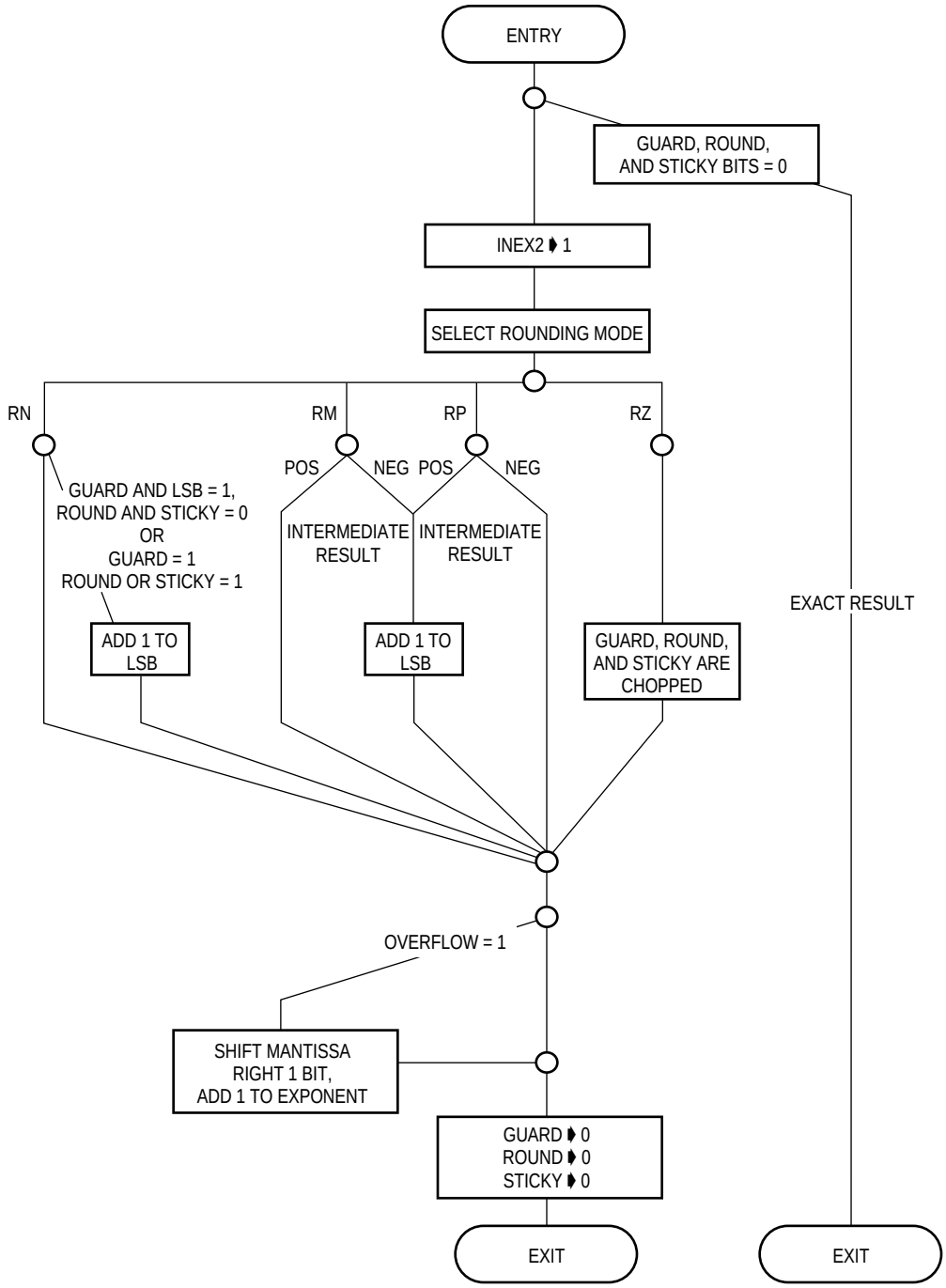
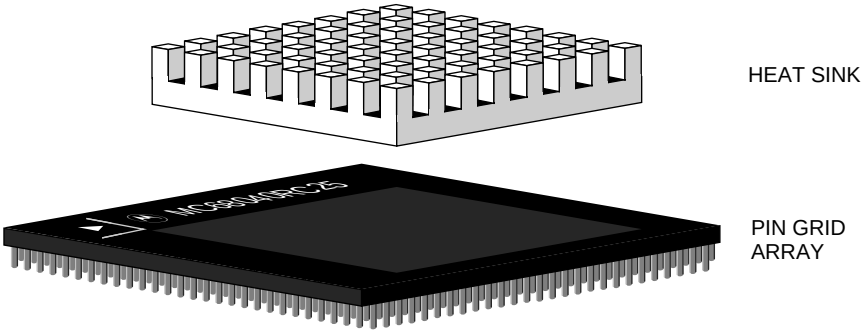


Figure 9-8. Rounding Algorithm Flowchart

The three additional bits beyond the extended-precision format, the difference between the intermediate result's 67-bit mantissa and the stored result's 64-bit mantissa, allow the FPU to perform all calculations as though it were performing calculations using a float engine with infinite bit precision. The result is always correct for the specified destination's data format before performing rounding (unless an overflow or underflow error occurs). The specified rounding operation then produces a number that is as close as possible to the infinitely precise intermediate value and still representable in the selected precision.



NOTE: Do not cover up microprocessor markings with an adhesive mounted heat sink.

Figure 11-10. Heat Sink with Adhesive

All pin-fin heat sinks tested were made from extrusion aluminum products. The planar face of the heat-sink matting to the package should have a good degree of planarity; if it has any curvature, the curvature should be convex at the central region of the heat-sink surface to provide intimate physical contact to the PGA surface. This heat sinks meet this criteria. Nonplanar, concave curvature in the central regions of the heat sink results in poor thermal contact to the package.

Although there are several ways to attach a heat sink to the package, it is easiest to use a demountable heat-sink attachment called “E-Z attach for PGA packages” (see Figure 11-33). A steel spring clamps the heat sink and the package to a plastic frame. Besides the height of the heat sink and plastic frame, no additional height is added to the package. The interface between the ceramic package and the aluminum heat sink was evaluated for both dry and wet interfaces in still air. The thermal grease reduced the θ_{CA} quite significantly (about 2.5 °C/W) in still air. An attachment with thermal grease provided about the same thermal performance as if a thermal epoxy had been used.

A.2 INTERRUPT PRIORITY LEVEL ($\overline{IPL2}$–$\overline{IPL0}$)

The $\overline{IPL2}$–$\overline{IPL0}$ pins do not have any affect on the selection of output buffer impedance.

A.3 JTAG SCAN (JS0)

The MC68040 DLE pin name has been changed to JS0. During normal operation, the JS0 pin cannot float, it must be tied to GND or Vcc directly or through a resistor. During board testing, this pin retains the functionality of the JTAG scan of the MC68040 for compatibility purposes. Refer to **Section 6 IEEE 1149.1A Test Access Port (JTAG)** for details concerning IEEE *1149.1 Standard Test Access Port and Boundary Scan Architecture*.

A.4 DATA LATCH AND MULTIPLEXED BUS MODES

The MC68LC040 does not implement the data latch or multiplexed modes of operation. The $\overline{CDIS}$ pin is ignored at the rising edge of reset. All timing and drive capabilities of the MC68LC040 are equivalent to those of the MC68040 in small output buffer impedance mode.

A.5 FLOATING-POINT UNIT (FPU)

The FPU is not implemented on the MC68LC040. All floating-point instructions cause an unimplemented floating-point exception to be taken with a new eight-word stack frame (format $4). The stack frame contains the status register (SR), program counter (PC), vector offset, effective address of the operand (where applicable), and PC value of the unimplemented floating-point instruction.

A.5.1 Unimplemented Floating-Point Instructions and Exceptions

All legal MC68040 and MC68881/MC68882 floating-point instructions are defined as unimplemented floating-point instructions on the MC68LC040. These instructions generate a format $4 stack frame during exception processing before taking an F-line exception. These instructions trap as an F-line exception, and the F-line exception handler can emulate them in software to maintain user-object-code compatibility.

The MC68LC040 assists the emulation process by distinguishing unimplemented floating-point instructions from other unimplemented F-line instructions. To aid emulation, the effective address is calculated and saved in the format $4 stack frame. This simplifies and speeds up the emulation process by eliminating the need for the emulation routine to determine the effective address and by providing information required to emulate the instruction on the exception stack frame in the supervisor address space. However, the floating-point instruction can reside in user space; therefore, the floating-point unimplemented exception handler may need to access user instruction space. The following processing steps occur for an unimplemented floating-point instruction:

1. When an unimplemented floating-point instruction is encountered, the instruction is partially decoded, and the effective address is calculated, if required.
2. The processor waits for all previous integer instructions, write-backs, and associated exception processing to complete before beginning exception processing for the unimplemented floating-point instruction. Any access error that occurs in completing the write-backs causes an access error exception, and the resulting stack frame indicates

ADDRESS MASK field equals $0F, the W-bit is set to one, and the S-field = $1. The inclusion of independent ACRs in both the instruction ACU (IACU) and data ACU (DACU provides an exception to the merged instruction and data address space, allowing different access control for instruction and operand accesses. Also, since the instruction memory unit is only used for instruction prefetches, different instruction and data ACRs can cause PC relative operand fetches to be translated differently from instruction prefetches.

Matching either of the ACRs in a corresponding ACU during an access to a memory unit completes the access with the ACU. If both registers match, the access uses the xACR0 status bits. Addresses are passed through without translation if there is no match in the ACRs and no table search occurs. The MC68EC040 does not perform table searches.

B.3.3 Effect of $\overline{\text{RSTI}}$ on the ACU

When the assertion of the reset input ($\overline{\text{RSTI}}$) signal resets the MC68EC040, the E-bits of the ACRs are cleared, disabling address access control.

B.4 SPECIAL MODES OF OPERATION

This part of the M68040 User's Manual does not apply to the MC68EC040. The MC68EC040 does not sample the $\overline{\text{IPL2}}$–$\overline{\text{IPL0}}$, $\overline{\text{CDIS}}$, JS0 (DLE on the MC68040), or JS1 ($\overline{\text{MDIS}}$ on the MC68040) pins on the rising edge of $\overline{\text{RSTI}}$.

An external device asserts $\overline{\text{RSTI}}$ to reset the processor. When power is applied to the system, external circuitry should assert $\overline{\text{RSTI}}$ for a minimum of 10 BCLK cycles after V_{CC} is within tolerance. Figure B-5 is a functional timing diagram of the power-on reset operation, illustrating the relationships between V_{CC}, $\overline{\text{RSTI}}$, and bus signals. The BCLK and PCLK clock signals are required to be stable by the time V_{CC} reaches the minimum operating specification. $\overline{\text{RSTI}}$ is internally synchronized for two BCLKS before being used, and must meet the specified setup and hold times to BCLK (specifications #51 and #52 in **MC68EC040 Electrical Characteristics**) only if recognition by a specific BCLK rising edge is required.

Once $\overline{\text{RSTI}}$ is negated, the processor is internally held in reset for another 128 clock cycles. During the reset period, all three-statable signals are three-stated, and the rest are driven to their inactive state. Once the internal reset signal negates, all bus signals remain in a high-impedance state until the processor is granted the bus. After this, the first bus cycle for reset exception processing begins. In Figure B-6, the processor assumes implicit ownership of the bus before the first bus cycle begins. The levels on the $\overline{\text{CDIS}}$, JS1 ($\overline{\text{MDIS}}$ on the MC68040), and $\overline{\text{IPL2}}$–$\overline{\text{IPL0}}$ signals are not sampled when $\overline{\text{RSTI}}$ is negated.

For processor resets after the initial power-on reset, should be asserted for at least 10 clock periods. Figure B-6 illustrates timing associated with a reset when the processor is executing bus cycles. Note that $\overline{\text{BB}}$ and $\overline{\text{TIP}}$ (and $\overline{\text{TA}}$ driven during a snooped access) are asserted before transitioning to a three-state level. Processor reset causes any bus cycle in progress to terminate as if $\overline{\text{TA}}$ or $\overline{\text{TEA}}$ had been asserted. Also, the processor initializes registers appropriately for a reset exception.

C.6 MC68040V AND MC68EC040V JTAG (PRELIMINARY)

The MC68040V and MC68EC040V include dedicated user-accessible test logic that is fully compatible with the IEEE standard 1149.1A *Standard Test Access Port and Boundary Scan Architecture*. Problems associated with testing high-density circuit boards have led to the standard's development under the sponsorship of the IEEE Test Technology Committee and the Joint Test Action Group (JTAG).

The following paragraphs are to be used in conjunction with the supporting IEEE document and includes those chip-specific items that the IEEE standard requires to be defined and additional information specific to the MC68040V and MC68EC040V implementations. For example, the IEEE standard 1149.1A test access port (TAP) controller states are referenced in this section but are not described. For these details and application information regarding the standard, refer to the IEEE standard 1149.1A document.

The MC68040V and MC68EC040V implementations support circuit board test strategies based on the standard. The test logic utilizes static logic design and is system logic independent of the device. The MC68040V and MC68EC040V implementations provide capabilities to:

a. Perform boundary scan operations to test circuit board electrical continuity,
b. Bypass the MC68040V and MC68EC040V by reducing the shift register path to a single cell,
c. Sample the MC68040V and MC68EC040V system pins during operation and transparently shift out the result,
d. Disable the output drive to output-only pins during circuit board testing.

NOTE

The IEEE standard 1149.1A test logic cannot be considered completely benign to those planning not to use this capability. Certain precautions must be observed to ensure that this logic does not interfere with system operation. Refer to **C.6.4 Disabling The IEEE Standard 1149.1A Operation**.

Figure C-4 illustrates a block diagram of the MC68040V and MC68EC040V implementations of IEEE standard 1149.1A. The test logic includes a 16-state dedicated TAP controller. These 16 controller states are defined in detail in the IEEE standard 1149.1A, but only 8 are included in this section.

Test-Logic-Reset	Run-Test/Idle
Capture-IR	Capture-DR
Update-IR	Update-DR
Shift-IR	Shift-DR

Four dedicated signal pins provides access to the TAP controller:

TCK—A test clock input that synchronizes the test logic.

TMS—A test mode select input with an internal pullup resistor sampled on the rising edge of TCK to sequence the TAP controller.

Table E-3 lists all the data formats and types supported by the MC68040 FPU. Also included are the data formats and types that the MC68040 FPU does not support but that are supported by the M68040FPSP.

Table E-3. Support for Data Types and Data Formats

Data Types	Data Formats						
	SGL	DBL	EXT	Decimal	Byte	Word	Long Word
Normalized	†	†	†	*	†	†	†
Zero	†	†	†	*	†	†	†
Infinity	†	†	†	*			
NAN	†	†	†	*			
Denormalized	‡	‡	*	*			
Unnormalized			*	*			

* Supported by M68040FPSP
† Supported by the MC68040 FPU
‡ Supported by M68040FPSP after being converted to extended precision by the MC68040 FPU

The M68040FPSP provides system designers with a simple path to port existing MC68881/MC68882 exceptions handlers to the MC68040. It also provides an entry point for the IEEE-defined exception conditions listed in Table E-4.

Table E-4. Exception Conditions

Mnemonic	Description
BSUN	Branch/Set on Unordered
SNAN	Signaling Not-a-Number
OPERR	Operand Error
OVFL	Overflow
UNFL	Underflow
DZ	Divide by Zero
INEX1/INEX2	Inexact Result 1/2

The M68040FPSP is written in M68000 family assembly code and comes with an installation guide. Tape contains both Motorola syntax and UNIX “as” syntax. Tape cartridge (M68040FPSPT) media is available in CPIO and TAR formats. Also available is 9-track (M68040FPSPP) media in high or low density as well as CPIO and TAR formats. A license is required to obtain rights to use and distribute the M68040FPSP. License terms include the right to use and modify source code and redistribute resulting object code.