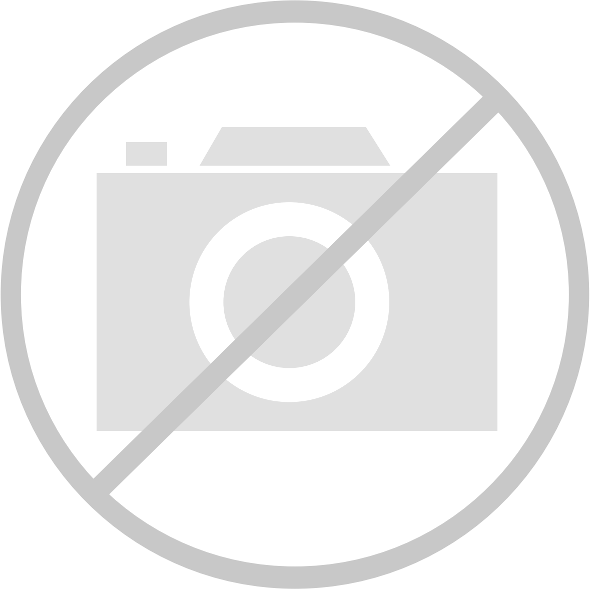
Motorola - MC68040RC40V Datasheet



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	182-BEPGA
Supplier Device Package	182-PGA (47.24x47.24)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68040rc40v

TABLE OF CONTENTS (Continued)

Paragraph Number | Title | Page Number

Section 6
IEEE 1149.1 Test Access Port (JTAG)

Table 1-4. Instruction Set Summary (Continued)

Opcode	Operation	Syntax
BTST	–(bit number of Destination) ø Z;	BTST Dn,<ea> BTST #<data>,<ea>

CAS	CAS Destination – Compare Operand ø cc;
	if Z, Update Operand ø Destination
	else Destination ø Compare Operand
		CAS Dc,Du,<ea>

CAS2	CAS2 Destination 1 – Compare 1 ø cc;
	if Z, Destination 2 – Compare ø cc;
	if Z, Update 1 ø Destination 1;
	Update 2 ø Destination 2
	else Destination 1 ø Compare 1;
	Destination 2 ø Compare 2
		CAS2 Dc1–Dc2,Du1–Du2,(Rn1)–(Rn2)

CHK	If Dn < 0 or Dn > Source then TRAP	CHK <ea>,Dn
CHK2	If Rn < LB or If Rn > UB then TRAP	CHK2 <ea>,Rn
CINV	If supervisor state then invalidate selected cache lines else TRAP	CINVL <caches>, (An) CINVP <caches>, (An) CINVA <caches>
CLR	0 ø Destination	CLR <ea>
CMP	Destination – Source ø cc	CMP <ea>,Dn
CMPA	Destination – Source	CMPA <ea>,An
CMPI	Destination – Immediate Data	CMPI #<data>,<ea>
CMPM	Destination – Source ø cc	CMPM (Ay)+,(Ax)+
CMP2	Compare Rn < LB or Rn > UB and Set Condition Codes	CMP2 <ea>,Rn
CPUSH	If supervisor state then if data cache push selected dirty data cache lines; invalidate selected cache lines else TRAP	CPUSHL <caches>, (An) CPUSHP <caches>, (An) CPUSHA <caches>
DBcc	If condition false then (Dn–1 ø Dn; If Dn ≠ –1 then PC + d_n ø PC)	DBcc Dn,<label>
DIVS, DIVSL	Destination ÷ Source ø Destination	DIVS.W <ea>,Dn 32 ÷ 16 ø 16r:16q DIVS.L <ea>,Dq 32 ÷ 32 ø 32q DIVS.L <ea>,Dr:Dq 64 ÷ 32 ø 32r:32q DIVSL.L <ea>,Dr:Dq 32 ÷ 32 ø 32r:32q
DIVU, DIVUL	Destination ÷ Source ø Destination	DIVU.W <ea>,Dn 32 ÷ 16 ø 16r:16q DIVU.L <ea>,Dq 32 ÷ 32 ø 32q DIVU.L <ea>,Dr:Dq 64 ÷ 32 ø 32r:32q DIVUL.L <ea>,Dr:Dq 32 ÷ 32 ø 32r:32q
EOR	Source ⊕ Destination ø Destination	EOR Dn,<ea>
EORI	Immediate Data ⊕ Destination ø Destination	EORI #<data>,<ea>
EORI to CCR	Source ⊕ CCR ø CCR	EORI #<data>,CCR
EORI to SR	If supervisor state then Source ⊕ SR ø SR else TRAP	EORI #<data>,SR

Table 1-4. Instruction Set Summary (Continued)

Opcode	Operation	Syntax
FNEG[2]	–(Source) ø FPn	FNEG.<fmt> <ea>,FPn FNEG.X FPm,FPn FNEG.X FPn FrNEG.<fmt> <ea>,FPn[3] FrNEG.X FPm,FPn[3] FrNEG.X FPn[3]
FNOP[2]	None	FNOP
FRESTORE[2]	If in supervisor state then FPU State Frame ø Internal State else TRAP	FRESTORE <ea>
FSAVE[2]	If in supervisor state then FPU Internal State ø State Frame else TRAP	FSAVE <ea>
FScc[2]	If condition true then 1s ø Destination else 0s ø Destination	FScc.SIZE <ea>
FSGLDIV	FPn ÷ Source ø FPn	FSGLDIV.<fmt> <ea>,FPn FSGLDIV.X FPm,FPn
FSGLMUL	Source × FPn ø FPn	FSGMUL.<fmt> <ea>,FPn FSGLMUL.X FPm, FPn
FSQRT[2]	Square Root of Source ø FPn	FSQRT.<fmt> <ea>,FPn FSQRT.X FPm,FPn FSQRT.X FPn FrSQRT.<fmt> <ea>,FPn[3] FrSQRT FPm,FPn[3] FrSQRT FPn[3]
FSUB[2]	FPn – Source ø FPn	FSUB.<fmt> <ea>,FPn FSUB.X FPm,FPn FrSUB.<fmt> <ea>,FPn[3] FrSUB.X FPm,FPn[3]
FTRAPcc[2]	If condition true then TRAP	FTRAPcc FTRAPcc.W #<data> FTRAPcc.L #<data>
FTST[2]	Condition Codes for Operand ø FPCC	FTST.<fmt> <ea> FTST.X FPm

ILLEGAL	SSP – 2 ø SSP; Vector Offset ø (SSP);
	SSP – 4 ø SSP; PC ø (SSP);
	SSp – 2 ø SSP; SR ø (SSP);
	Illegal Instruction Vector Address ø PC
		ILLEGAL

JMP	Destination Address ø PC	JMP <ea>
JSR	SP – 4 ø SP; PC ø (SP) Destination Address ø PC	JSR <ea>
LEA	<ea> ø An	LEA <ea>,An
LINK	SP – 4 ø SP; An ø(SP) SP ø An, SP+d øSP	LINK An,d_n

3.1.3 Transparent Translation Registers

The data transparent translation registers (DTTR0 and DTTR1) and instruction transparent translation registers (ITTR0 and ITTR1) are 32-bit registers that define blocks of logical address space. The TTRs operate independently of the E-bit in the TCR and the state of the MDIS signal. Data transfers to and from these registers are long-word transfers. The TTR fields are defined following Figure 3-5, which illustrates TTR format. Bits 12–10, 7, 4, 3, 1, and 0 always read as zero.

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

Figure 3-5. Transparent Translation Register Format

Logical Address Base

This 8-bit field is compared with address bits A31–A24. Addresses that match in this comparison (and are otherwise eligible) are transparently translated.

Logical Address Mask

Since this 8-bit field contains a mask for the Logical Address Mask field, setting a bit in this field causes the corresponding bit in the Logical Address Base field to be ignored. Blocks of memory larger than 16 Mbytes can be transparently translated by setting some of the logical address mask bits to ones. The low-order bits of this field can be set to define contiguous blocks larger than 16 Mbytes.

E—Enable

This bit enables or disables transparent translation of the block defined by this register:

0 = Transparent translation disabled
1 = Transparent translation enabled

S—Supervisor Mode

This field specifies the way FC2 is used in matching an address:

00 = Match only if FC2 = 0 (user mode access)
01 = Match only if FC2 = 1 (supervisor mode access)
1X = Ignore FC2 when matching

U0, U1—User Page Attributes

The user defines these bits, and the M68040 does not interpret them. U0 and U1 are echoed to the UPA0 and UPA1 signals, respectively, if an external bus transfer results from an access. These bits can be programmed by the user to support external addressing, bus snooping, or other applications.

read resulting from a write miss in copyback mode is cache inhibited, the write access misses in the cache and writes through to memory.

4.6.2 Cache Pushes

When the cache controller selects a dirty data cache line for replacement, memory must be updated with the dirty data before the line is replaced. This occurs when a CPUSH instruction execution explicitly selects the cache and when a cache inhibit access hits in the cache. To reduce the requested data's latency in the new line, the dirty line being replaced is temporarily placed in a push buffer while the new line is fetched from memory. When a line is allocated to the push buffer, an alternate bus master can snoop it, but the execution units cannot access it. After the bus transfer for the new line successfully completes, the dirty cache line is copied back to memory, and the push buffer is invalidated. If the operation to access the replacement line is abnormally terminated or signaled as cache inhibited, the line in the push buffer is copied back into its original position in the cache, and the processor continues operation as described in the previous paragraphs.

The number of dirty long words in the line to be pushed determines the size of the push transfer on the bus, minimizing bus bandwidth required for the push. A single long word is written to memory using a long-word push transfer if it is dirty. A push transfer is distinguished from a normal write transfer by an encoding of 000 on the transfer modifier signals (TM2–TM0) for the push. Asserting TA and TEA retries the transfer; a bus-error-asserted TEA terminates it. If a bus error terminates a push transfer, the processor immediately takes an exception.

A line containing two or more dirty long words is copied back to memory, using a line push transfer. For a line push, the bus controller requests a burst write transfer by indicating a line access with SIZ1 and SIZ0. The responding device sequentially accepts four long words of data. If the responding device does not support the burst mode, it should assert TBI for the first long word of the line access. The bus controller responds by terminating the line access and completes the remainder of the line push as three, sequential, long-word writes. The first cycle of the burst can be retried, but the bus controller interprets a retry for any of the three remaining cycles as a bus error. If a bus error occurs in any cycle in the line push transfer, the processor immediately takes an exception.

A dirty cache line hit by a cache-inhibited access is pushed before the external bus access occurs. If the access is part of a locked transfer sequence for TAS, CAS, or CAS2 operand accesses or translation table updates, the LOCK signal is also asserted for the push access.

4.7 CACHE OPERATION SUMMARY

The instruction and data caches function independently when servicing access requests from the IU. The following paragraphs discuss the operational details for the caches and present state diagrams depicting the cache line state transitions.

Table 5-1. Signal Index (Continued)

Signal Name	Mnemonic	Function
Processor Clock	PCLK[4]	Clock input used for internal logic timing. The PCLK frequency is exactly 2 × the BCLK frequency.
Test Clock	TCK	Clock signal for the IEEE P1149.1 Test Access Port (TAP).
Test Mode Select	TMS	Selects the principle operations of the test-support circuitry.
Test Data Input	TDI	Serial data input for the TAP.
Test Data Output	TDO	Serial data output for the TAP.
Test Reset	TRST[4]	Provides an asynchronous reset of the TAP controller.
Power Supply	V_{CC}	Power supply.
Ground	GND	Ground connection.

NOTES:

1. This signal is only available on the MC68040.
2. This signal is not available on the MC68EC040 and the MC68EC040V.
3. These signals are different on power-up for the MC68LC040 and MC68EC040.
4. These signals are not available on the MC68040V and MC68EC040V.

The following examples are for PSTx encodings:

1. An access error terminates an instruction such that the instruction execution stage is not reached. In this case, an ‘end current instruction’ is not indicated. Exception processing starts, the exception stacking status is indicated, and then the virtual JMP causes the ‘supervisor, branch taken/end current instruction’ encoding.
2. An FTRAPcc that does not take an exception ending with the ‘end current instruction’ encoding. The exception stacking status is indicated and then reaches the ‘supervisor, branch taken/end current instruction’ encoding if the FTRAPcc ends in an exception.
3. Two simultaneous interrupt exception processing sequences follow an ADD instruction. The ADD instruction ends with ‘end current instruction’, followed by exception stacking, followed by ‘branch taken/end current instruction’, followed by exception stacking, followed by ‘branch taken/end current instruction’.
4. An RTE instruction follows an ADD instruction. The ‘end current instruction’ is followed by RTE executing followed by a branch taken/end current instruction.

5.9.2 Bus Clock (BCLK)

This input signal is used as a reference for all bus timing. It is a TTL-compatible signal and cannot be gated off. Refer to **Section 11 MC68040 Electrical and Thermal Characteristics** for electrical specifications.

5.9.3 Processor Clock (PCLK)—Not on MC68040V and MC68EC040V

PCLK is used to derive all internal timing. This clock is also TTL compatible and cannot be gated off. Refer to **Section 11 MC68040 Electrical and Thermal Characteristics** for electrical specifications.

5.10 MMU DISABLE (MDIS)—NOT ON MC68EC040

The MMU disable signal dynamically disables the translation of addresses by the MMUs. The assertion of MDIS does not flush the address translation caches (ATCs); ATC entries become available again when MDIS is negated. During a processor reset, the level on MDIS is latched and used to select the normal data latch mode (MDIS high) or DLE mode (MDIS low). Refer to **Section 3 Memory Management Unit (Except MC68EC040 and MC68EC040V)** for a description of address translation and to **Section 7 Bus Operation** for information about DLE mode.

5.11 DATA LATCH ENABLE (DLE)—ONLY ON MC68040

This input signal is used in DLE mode to latch the input data bus on read transfers. DLE mode can be used to support asynchronous memory interfaces by allowing the interface to specify when data should be latched instead of requiring data to be valid on the rising edge of BCLK.

6.1 OVERVIEW

Figure 6-1 illustrates a block diagram of the M68040 implementation 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
Capture-IR
Update-IR
Shift-IR
Run-Test/Idle
Capture-DR
Update-DR
Shift-DR

The TAP controller provides access to five dedicated signal pins:

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.

TDI—A test data input with an internal pullup resistor sampled on the rising edge of TCK.

TDO—A three-state test data output actively driven only in the shift-IR and shift-DR controller states that changes on the falling edge of TCK.

TRST —An active-low asynchronous reset with an internal pullup resistor that forces the TAP controller into the test-logic-reset state.

The test logic also includes an instruction shift register and two test data registers, a boundary scan register and a bypass register. The boundary scan register links all device signal pins into the instruction shift register.

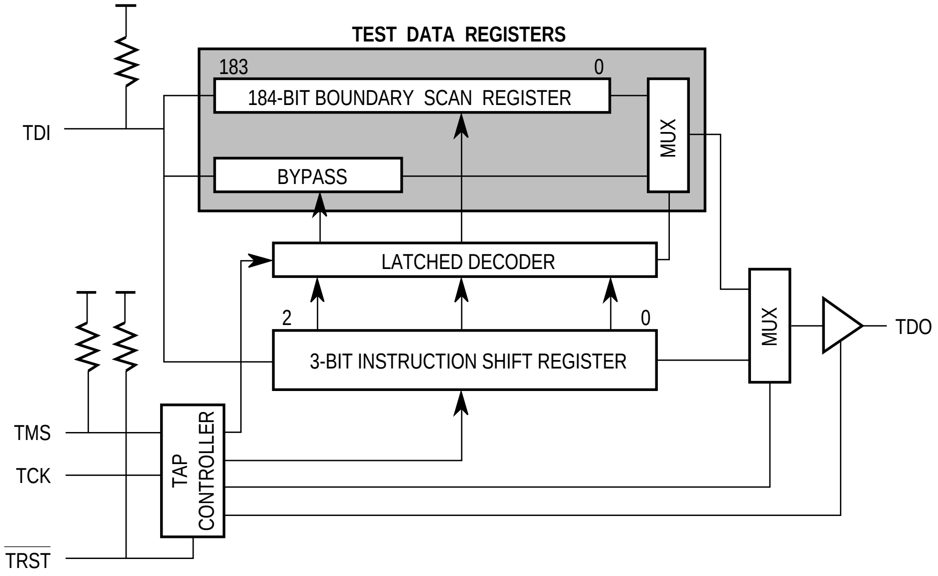


Figure 6-1. M68040 Test Logic Block Diagram

generates the vector number, which is the sum of the interrupt priority level plus 24 ($18). There are seven distinct autovectors that can be used, corresponding to the seven levels of interrupts available with $\overline{\text{IPL2}}$–$\overline{\text{IPL0}}$ signals. Figure 7-23 illustrates a functional timing diagram for an autovector operation.

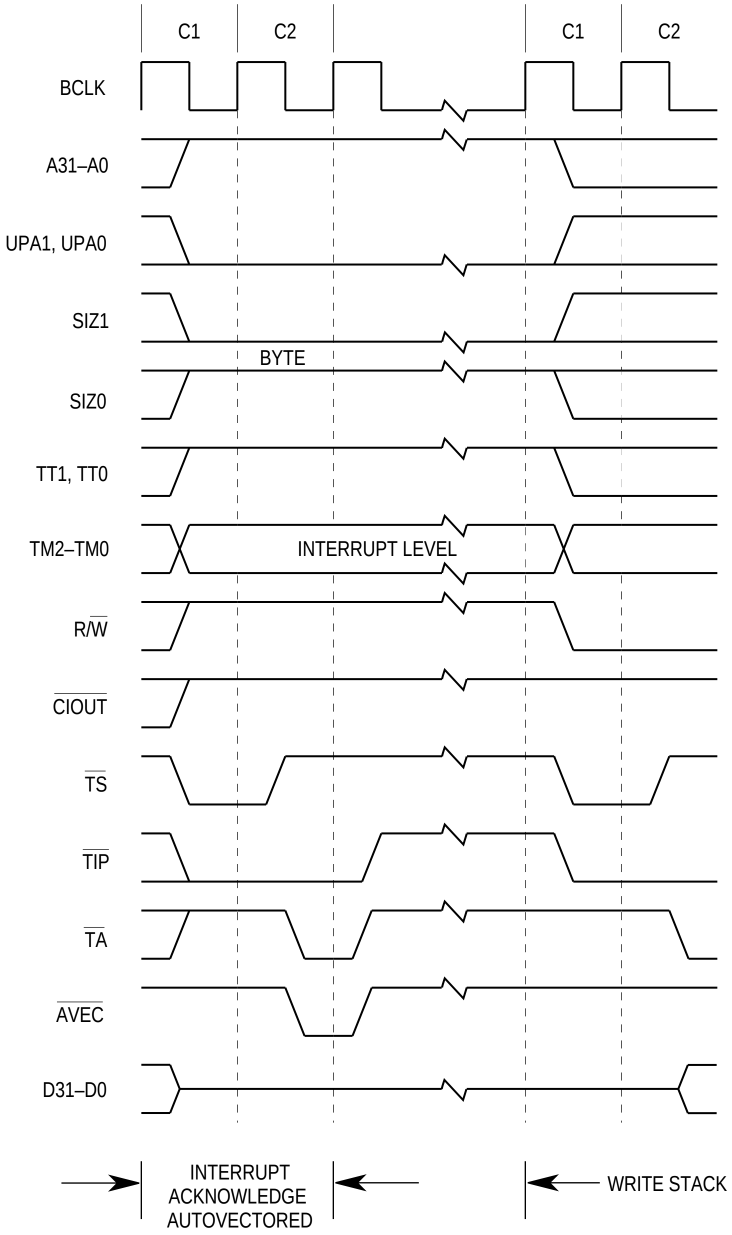


Figure 7-23. Autovector Interrupt Acknowledge Bus Cycle Timing

7.5.1.3 SPURIOUS INTERRUPT ACKNOWLEDGE BUS CYCLE. When a device does not respond to an interrupt acknowledge bus cycle with $\overline{\text{TA}}$, or $\overline{\text{AVEC}}$ and $\overline{\text{TA}}$, the external logic typically returns the transfer error acknowledge signal ($\overline{\text{TEA}}$). In this case, the M68040 automatically generates the spurious interrupt vector number 24 ($18) instead of the interrupt vector number. If $\overline{\text{TA}}$ and $\overline{\text{TEA}}$ are both asserted, the processor retries the cycle.

on the operand read itself can cause the instruction to be aborted, preventing multiple reads. It is important to note that when memory accesses are serialized noncachable, FMOVE will cause two identical writes to the same location to occur if the next instruction prefetch receives a bus error.

Since write cycles can be deferred indefinitely, many subsequent instructions can be executed, resulting in seemingly nonsequential instruction execution. When this action is not desired and the system depends on sequential execution following bus activity, the NOP instruction can be used. The NOP instruction forces instruction and bus synchronization because it freezes instruction execution until all pending bus cycles have completed.

A write operation of control information to an external register in which the external hardware attempts to control program execution based on the data that is written with the conditional assertion of $\overline{\text{TEA}}$ is one situation where the NOP instruction can be used to prevent multiple executions. If the data cache is enabled and the write cycle results in a hit in the data cache, the cache is updated. That data, in turn, may be used in a subsequent instruction before the external write cycle completes. Since the M68040 cannot process the bus error until the end of the bus cycle, the external hardware cannot successfully interrupt program execution. To prevent a subsequent instruction from executing until the external cycle completes, the NOP instruction can be inserted after the instruction causing the write. In this case, access error exception processing proceeds immediately after the write before subsequent instructions are executed. This is an irregular situation, and the use of the NOP instruction for this purpose is not required by most systems.

Note that the NOP instruction can also be used to force access serialization by placing NOP before the instruction that reads an I/O device. This practice eliminates the need to specify the entire page as serialized noncachable but does not prevent the instruction from being aborted by an exception condition.

7.8 BUS ARBITRATION AND EXAMPLES

The bus design of the M68040 provides for one bus master at a time, either the M68040 or an external device. More than one device having the capability to control the bus can be attached to the bus. An external arbiter prioritizes requests and determines which device is granted access to the bus. Bus arbitration is the protocol by which the processor or an external device becomes the bus master. When the M68040 is the bus master, it uses the bus to read instructions and data not contained in its internal caches from memory and to write data to memory. When an alternate bus master owns the bus, the M68040 is able to monitor the alternate bus master's transfer and intervene when necessary to maintain cache coherency. This capability is discussed in more detail in **7.9 Bus Snooping Operation.**

Unlike earlier members of the M68000 family, the M68040 implements an arbitration method in which an external arbiter controls bus arbitration and the processor acts as a slave device requesting ownership of the bus from the arbiter. Since the user defines the functionality of the external arbiter, it can be configured to support any desired priority scheme. For systems in which the processor is the only possible bus master, the bus can

7.11 SPECIAL MODES OF OPERATION

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

7.11.1 Output Buffer Impedance Selection

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

7.11.2 Multiplexed Bus Mode

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

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

they can be used to read the FPIAR in an exception handler without changing the previous value. A reset or a restore operation of the null state clears the FPIAR.

9.3 FLOATING-POINT DATA FORMATS AND DATA TYPES

The M68000 floating-point model (MC68881, MC68882, MC68040) supports the following data formats: single precision, double precision, extended precision, and packed decimal. The M68000 floating-point model supports the following data types: normalized, zeros, infinities, denormalized numbers, and NANs. The MC68040 supports part of the M68000 floating-point model in hardware. Table 9-2 lists the data formats and data types supported by the MC68040. Tables 9-3 through 9-6 summarize the floating-point data formats and data types details. For further information on the data formats and data types, refer to the M68000UM/AD, *M68000 Family Programmer's Reference Manual*.

Table 9-2. MC68040 FPU Data Formats and Data Types

	Data Formats						
Number Types	**Single-Precision Real**	**Double-Precision Real**	**Extended-Precision Real**	**Packed-Decimal Real**	**Byte Integer**	**Word Integer**	**Long-Word Integer**
Normalized	*	*	*	†	*	*	*
Zero	*	*	*	†	*	*	*
Infinity	*	*	*	†			
NAN	*	*	*	†			
Denormalized	†	†	†	†			
Unnormalized			†	†			

*Data Format/Type Supported by On-Chip MC68040 FPU Hardware
†Data Format/Type Supported by Software (MC68040FPSP)

Table 10.1. Instruction Timing Index (Continued)

Instruction	Page	Instruction	Page	Instruction	Page
FSAVE <ea>	10-33	MOVEP	10-11	ROL	10-26
FScc	10-32	MOVEQ	10-11	ROR	10-26
FSQRT	10-30,36	MOVES <ea>,An	10-24	ROXL	10-27
FSUB	10-30,35	MOVES <ea>,Dn	10-24	ROXR	10-27
FTRAPcc	10-29	MOVES Rn,<ea>	10-24	RTD	10-11
FTST <ea>, FPn	10-30	MULS.W/L	10-25	RTE	10-11
ILLEGAL	10-11	MULU.W/L	10-25	RTR	10-11
JMP	10-20	NBCD	10-25	RTS	10-11
JSR	10-21	NEG	10-26	SBCD	10-11
LEA	10-21	NEGX	10-26	Scc	10-27
LINK	10-11	NOP	10-11	SUB	10-13
LSL	10-14	NOT	10-26	SUBA	10-27
LSR	10-14	OR	10-13	SUBI	10-13
MOVE	10-9,10	ORI	10-13	SUBQ	10-14
MOVE from CCR	10-21	ORI #<xxx>,CCR	10-11	SUBX	10-11
MOVE from SR	10-22	ORI #<xxx>,SR	10-11	SWAP	10-11
MOVE to CCR	10-22	PACK	10-11	TAS	10-28
MOVE to SR	10-22	PEA	10-26	TRAP#	10-11
MOVE USP	10-11	PFLUSH	10-11	TRAPcc	10-11
MOVE16	10-11	PFLUSHA	10-11	TRAPV	10-11
MOVEA.L	10-23	PFLUSHAN	10-11	TST	10-13
MOVEC	10-11	PFLUSHN (An)	10-11	UNLK	10-11
MOVEM <list>,<ea>	10-23	PTESTR, PTESTW	10-11	UNPK	10-11
MOVEM.L <ea>,<list>	10-23	RESET	10-11		

10.6 INTEGER UNIT INSTRUCTION TIMINGS (Continued)

Addressing Mode	DIVS.W, DIVU.W*		DIVS.L, DIVU.L, DIVSL.L, DIVUL.L*		JMP	
	<ea> Calculate	Execute	<ea> Calculate	Execute	<ea> Calculate	Execute
Dn	8	27	9	44	—	—
An	—	—	—	—	—	—
(An)	8	27	9	44	3	$2_L + 1$
(An)+	8	27	9	44	—	—
–(An)	8	27	9	44	—	—
(d$_{16}$,An)	8	27	11	$2_L + 44$	4	$3_L + 1$
(d$_{16}$,PC)	11	$3_L + 27$	12	$3_L + 44$	6	$5_L + 1$
(xxx).W, (xxx).L	8	27	11	$2_L + 44$	3	$2_L + 1$
#<xxx>	8	27	10	$1_L + 44$	—	—
(d$_8$,An,Xn)	11	30	12	47	6	6
(d$_8$,PC,Xn)	12	$1_L + 30$	13	$1_L + 47$	7	$1_L + 6$
(BR,Xn)	13	$1_L + 31$	14	$1_L + 48$	8	$1_L + 7$
(bd,BR,Xn)	14	$1_L + 32$	15	$1_L + 49$	9	$1_L + 8$
([bd,BR,Xn])	17	$1_L + 35$	18	$1_L + 52$	12	$1_L + 11$
([bd,BR,Xn],od)	18	$1_L + 36$	19	$1_L + 53$	12	$1_L + 11$
([bd,BR],Xn)	18	$3_L + 34$	19	$3_L + 51$	13	$3_L + 10$
([bd,BR],Xn,od)	19	$3_L + 35$	20	$3_L + 52$	14	$3_L + 11$

*This instruction interlocks the <ea> calculate and execute stages. Execution time for a DIV/0 exception taken and exception processing is approximately 16 + <ea> calculate clocks. For example, DIV.W #0,Dn takes approximately 24 clocks in both the <ea> calculate and execute times to execute the divide instruction, perform exception stacking, fetch the exception vector, and prefetch the next instruction.

11.7 INPUT AC TIMING SPECIFICATIONS (see Figures 11-3 to 11-7)

		25 MHz		33 MHz		40 MHz		
Num	**Characteristic**	**Min**	**Max**	**Min**	**Max**	**Min.**	**Max.**	**Unit**
15	Data-In Valid to BCLK (Setup)	5	—	4	—	3	—	ns
16	BCLK to Data-In Invalid (Hold)	4	—	4	—	3	—	ns
17	BCLK to Data-In High Impedance (Read Followed by Write)	—	49	—	36.5	—	30.25	ns
22a	$\overline{\text{TA}}$ Valid to BCLK (Setup)	10	—	10	—	8	—	ns
22b	$\overline{\text{TEA}}$ Valid to BCLK (Setup)	10	—	10	—	9	—	ns
22c	$\overline{\text{TCI}}$ Valid to BCLK (Setup)	10	—	10	—	9	—	ns
22d	$\overline{\text{TBI}}$ Valid to BCLK (Setup)	11	—	10	—	9	—	ns
23	BCLK to $\overline{\text{TA}}$, $\overline{\text{TEA}}$, $\overline{\text{TCI}}$, $\overline{\text{TBI}}$ Invalid (Hold)	2	—	2	—	2	—	ns
24	$\overline{\text{AVEC}}$ Valid to BCLK (Setup)	5	—	5	—	5	—	ns
25	BCLK to $\overline{\text{AVEC}}$ Invalid (Hold)	2	—	2	—	2	—	ns
31	DLE Width High	8	—	8	—	8	—	ns
32	Data-In Valid to DLE (Setup)	2	—	2	—	2	—	ns
33	DLE to Data-In Invalid (Hold)	8	—	8	—	8	—	ns
34	BCLK to DLE Hold	3	—	3	—	3	—	ns
35	DLE High to BCLK	16	—	12	—	12	—	ns
36	Data-In Valid to BCLK (DLE Mode Setup)	5	—	5	—	5	—	ns
37	BCLK to Data-In Invalid (DLE Mode Hold)	4	—	4	—	4	—	ns
41a	$\overline{\text{BB}}$ Valid to BCLK (Setup)	7	—	7	—	7	—	ns
41b	$\overline{\text{BG}}$ Valid to BCLK (Setup)	8	—	7	—	7	—	ns
41c	$\overline{\text{CDIS}}$, $\overline{\text{MDIS}}$ Valid to BCLK (Setup)	10	—	8	—	8	—	ns
41d	$\overline{\text{IPLx}}$ Valid to BCLK (Setup)	4	—	3	—	3	—	ns
42	BCLK to $\overline{\text{BB}}$, $\overline{\text{BG}}$, $\overline{\text{CDIS}}$, $\overline{\text{IPLx}}$, $\overline{\text{MDIS}}$ Invalid (Hold)	2	—	2	—	2	—	ns
44a	Address Valid to BCLK (Setup)	8	—	7	—	7	—	ns
44b	SIZx Valid to BCLK (Setup)	12	—	8	—	8	—	ns
44c	TTx Valid to BCLK (Setup)	6	—	8.5	—	8.5	—	ns
44d	R/$\overline{\text{W}}$ Valid to BCLK (Setup)	6	—	5	—	5	—	ns
44e	SCx Valid to BCLK (Setup)	10	—	11	—	8	—	ns
45	BCLK to Address,SIZx, TTx, R/$\overline{\text{W}}$, SCx Invalid (Hold)	2	—	2	—	2	—	ns
46	$\overline{\text{TS}}$ Valid to BCLK (Setup)	5	—	9	—	7	—	ns
47	BCLK to $\overline{\text{TS}}$ Invalid (Hold)	2	—	2	—	2	—	ns
49	BCLK to $\overline{\text{BB}}$ High Impedance (MC68040 Assumes Bus Mastership)	—	9	—	9	—	9	ns
51	$\overline{\text{RSTI}}$ Valid to BCLK	5	—	4	—	4	—	ns
52	BCLK to $\overline{\text{RSTI}}$ Invalid	2	—	2	—	2	—	ns
53	Mode Select Setup to $\overline{\text{RSTI}}$ Negated	20	—	20	—	20	—	ns
54	$\overline{\text{RSTI}}$ Negated to Mode Selects Invalid	2	—	2	—	2	—	ns

rated at a particular maximum operating frequency is dependent upon the power-up mode. Therefore, the MC68040 is rated at a maximum power dissipation for either the large or small buffers at a particular frequency. This allows for control of some of the thermal management upon reset. The following equation provides a rough method to calculate the maximum power consumption for a chosen output buffer mode:

$$P_D = P_{DSB} + (P_{DLB} - P_{DSB}) \times (PINS_{LB} \div PINS_{CLB})$$

where:

P_D = Maximum Power Dissipation for Output Buffer Mode Selected
P_{DSB} = Maximum Power Dissipation for Small Buffer Mode (All Outputs)
P_{DLB} = Maximum Power Dissipation for Large Buffer Mode (All Outputs)
$PINS_{LB}$ = Number of Pins Large Buffer Mode
$PINS_{CLB}$ = Number of Pins Capable of the Large Buffer Mode

Table 11-1 lists the simplified relationship on the maximum power dissipation for eight possible configurations of output buffer modes.

Table 11-1. Maximum Power Dissipation for Output Buffer Mode Configurations

Output Configuration			
Data Bus	**Address Bus and Transfer Attributes**	**Control Signals**	**Maximum Power Dissipation**
Small*	Small	Small	P_{DSB}
Small	Small	Large	$P_{DSB} + (P_{DLB} - P_{DSB}) \times 13\%$
Small	Large	Small	$P_{DSB} + (P_{DLB} - P_{DSB}) \times 52\%$
Small	Large	Large	$P_{DSB} + (P_{DLB} - P_{DSB}) \times 65\%$
Large	Small	Small	$P_{DSB} + (P_{DLB} - P_{DSB}) \times 35\%$
Large	Small	Large	$P_{DSB} + (P_{DLB} - P_{DSB}) \times 48\%$
Large	Large	Small	$P_{DSB} + (P_{DLB} - P_{DSB}) \times 87\%$
Large	Large	Large	$P_{DSB} + (P_{DLB} - P_{DSB}) \times 100\%$

*The MC68LC040 and MC68EC040 only utilize this row of information.

To calculate the specific power dissipation of a design, the termination method of each signal must be considered. For example, a signal output that is not connected would not dissipate any additional power if it were configured in the large rather than the small buffer mode. Since the maximum operating junction temperature is specified as 110°C, the maximum case temperature (T_C) in °C can be obtained from the following equation:

$$T_C = T_J - P_D \times \theta_{JC}$$

where:

T_C = Maximum Case Temperature
T_J = Maximum Junction Temperature
P_D = Maximum Power Dissipation of the Device
θ_{JC} = Thermal Resistance between the Junction of the Die and the Case

Table C-1. Additional MC68040V and MC68EC040V Signals

Signal Name	Mnemonic	Function
Low Frequency Operation	$\overline{\text{LFO}}$	Used to enter the low frequency mode of operation.
Loss of Clock	LOC	Indicates loss of BCLK input, a reset is required
System Clock Disable	$\overline{\text{SCD}}$	Indicates normal operation is suspended and low-power stop mode is active, system logic may remove or change the frequency of the BCLK input.

C.1.1 Low Frequency Operation ($\overline{\text{LFO}}$)

When asserted, this input signal allows the frequency of BCLK to be changed instantaneously (0 to 16 MHz) providing minimum pulse width constraints are met (see **C.7 MC68040V and MC68EC040V Electrical Characteristics**. $\overline{\text{LFO}}$ is only recognized during low-power stop mode and reset.

C.1.2 Loss of Clock (LOC)

Whenever the internal clock circuitry detects either a phase lock error or a loss of BCLK, this output signal is driven high (only during normal mode of clocking operation). LOC is also three-stated during reset, low-power stop, or low frequency operation. There should be a pull-down resistor on the system board to ground.

C.1.3 System Clock Disable ($\overline{\text{SCD}}$)

When asserted this output signal indicates, when asserted, that the BCLK input can be disabled or changed in frequency. $\overline{\text{SCD}}$ is asserted upon termination of the LPSTOP broadcast cycle. BCLK must be stable when $\overline{\text{SCD}}$ is negated, in accordance with the specifications in **C.7 MC68040V and MC68EC040V Electrical Characteristics**.

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**.

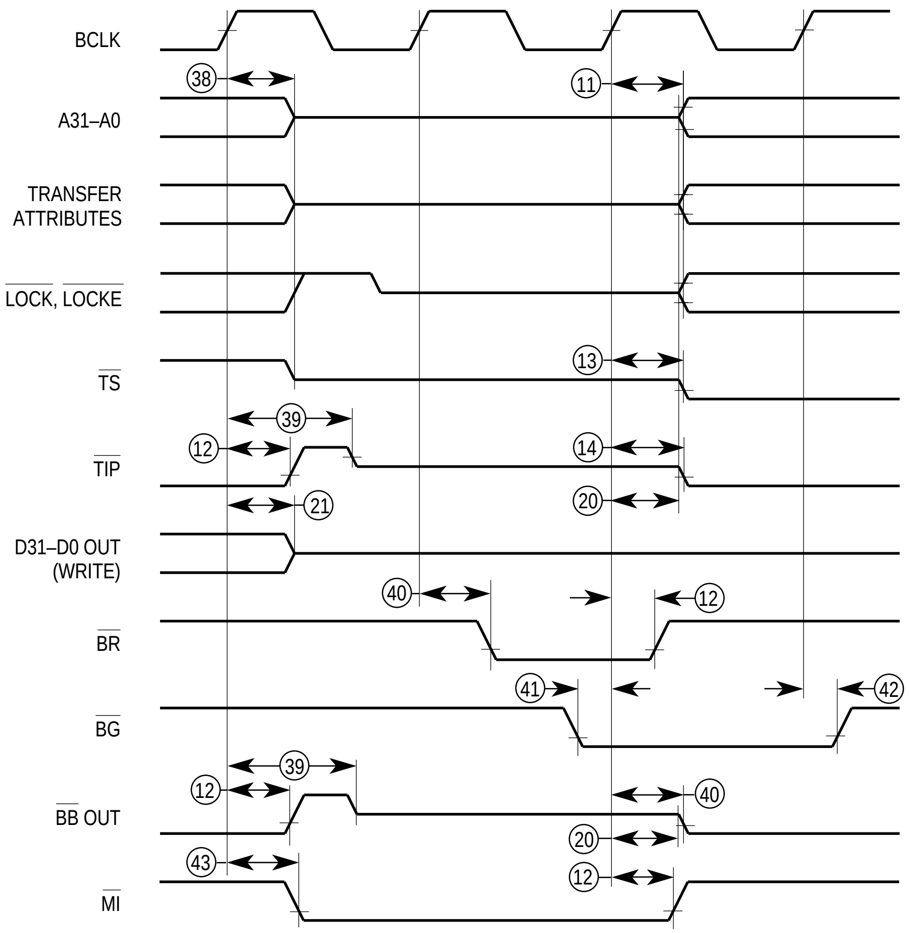
Figure C-4 illustrates a block diagram of the MC68040V and MC68EC040V implementations of IEEE standard 1149.1A. The test logic includes a 16-state dedicated TAP controller. These 16 controller states are defined in detail in the IEEE standard 1149.1A, but only 8 are included in this section.

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.



NOTE: Transfer Attribute Signals = UPAx, SIZx, TTx, TMx, TLNx, R/W, CIOUT

Figure C-14. Bus Arbitration Timing

INDEX