Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	68030
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	25MHz
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	132-BCQFP
Supplier Device Package	132-CQFP (24x24)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68ec030cfe25c

INTRODUCTION

The MC68EC030 is an integrated controller that incorporates the capabilities of the MC68030 integer unit, a data cache, an instruction cache, an access control unit (ACU), and an improved bus controller on one VLSI device. It maintains the 32-bit registers available with the entire M68000 Family as well as the 32-bit address and data paths, rich instruction set, versatile addressing modes, and flexible coprocessor interface provided with the MC68020 and MC68030. In addition, the internal operations of this integrated controller are designed to operate in parallel, allowing instruction execution to proceed in parallel with accesses to the internal caches and the bus controller.

The MC68EC030 fully supports the nonmultiplexed asynchronous bus of the MC68020 and MC68030 as well as the dynamic bus sizing mechanism that allows the controller to transfer operands to or from external devices while automatically determining device port size on a cycle-by-cycle basis. In addition to the asynchronous bus, the MC68EC030 also supports the fast synchronous bus of the MC68030 for off-chip caches and fast memories. Like the MC68030, the MC68EC030 bus is capable of fetching up to four long words of data in a burst mode compatible with DRAM chips that have burst capability. Burst mode can reduce (up to 50 percent) the time necessary to fetch the four long words. The four long words are used to prefill the on-chip instruction and data caches so that the hit ratio of the caches is improved and the average access time for operand fetches is minimized.

The MC68EC030 is specifically designed to sustain high performance while using low-cost (DRAM) memory subsystems. Coupled with the MC88916 clock generation and distribution circuit, the MC68EC030 provides simple interface to lower speed memory subsystems. The MC88916 (see Figure 1) provides the precise clock signals required to efficiently control memory subsystems, eliminating system design constraints due to clock generation and distribution.

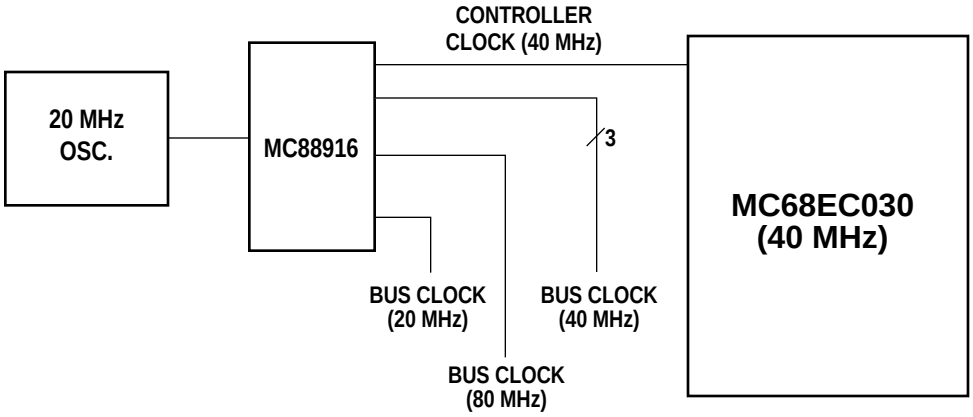


Figure 1. MC68EC030 Clock Circuitry

The block diagram shown in Figure 2 depicts the major sections of the MC68EC030 and illustrates the autonomous nature of these blocks. The bus controller consists of the address and data pads, the multiplexers required to support dynamic bus sizing, and a microbus controller that schedules the bus cycles on the basis of priority. The micromachine contains the execution unit and all related control logic. Microcode control is provided by a modified two-level store of microROM and nanoROM contained in the micromachine. Programmed logic arrays (PLAs) are used to provide instruction decode and sequencing

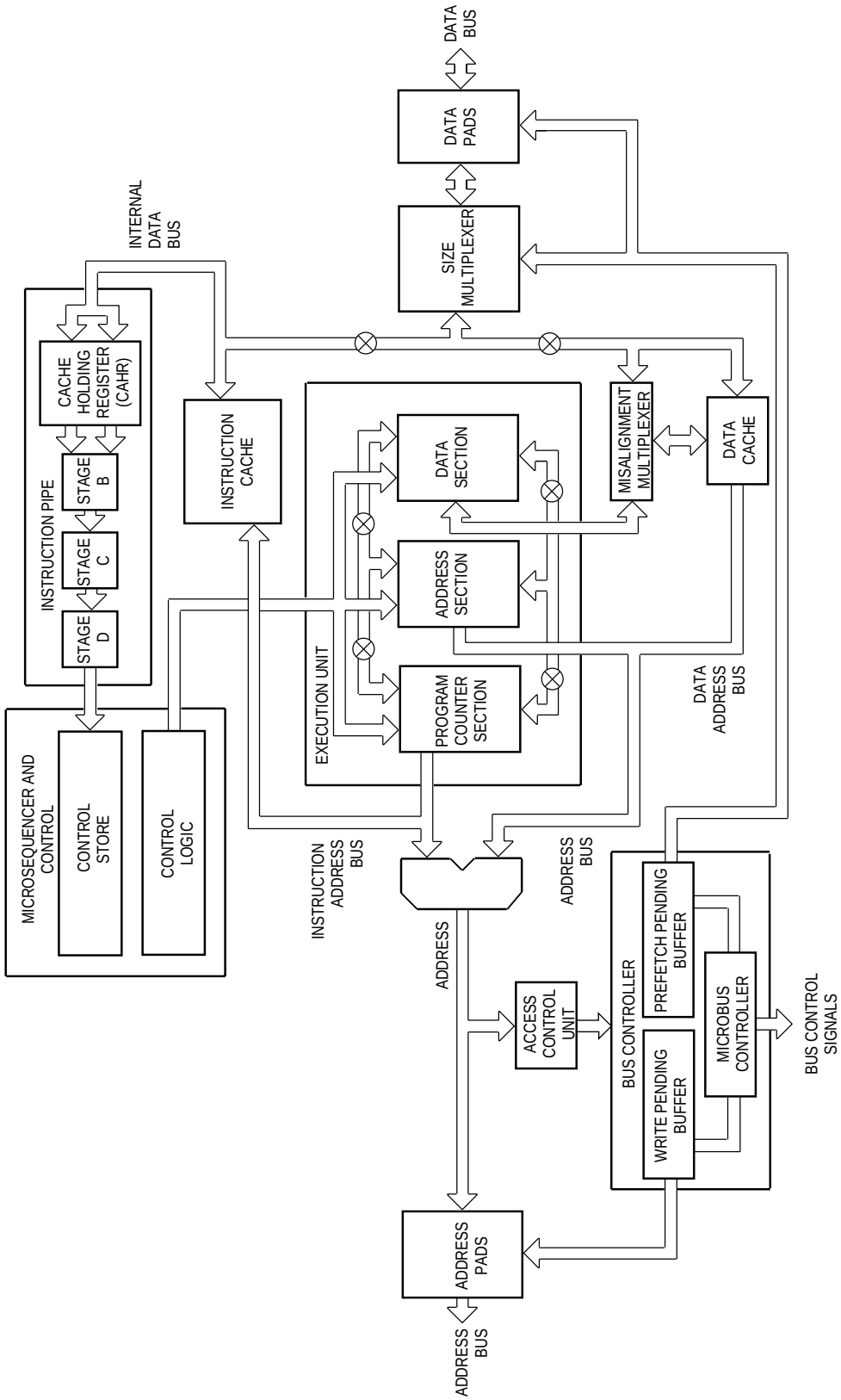


Figure 2. Block Diagram

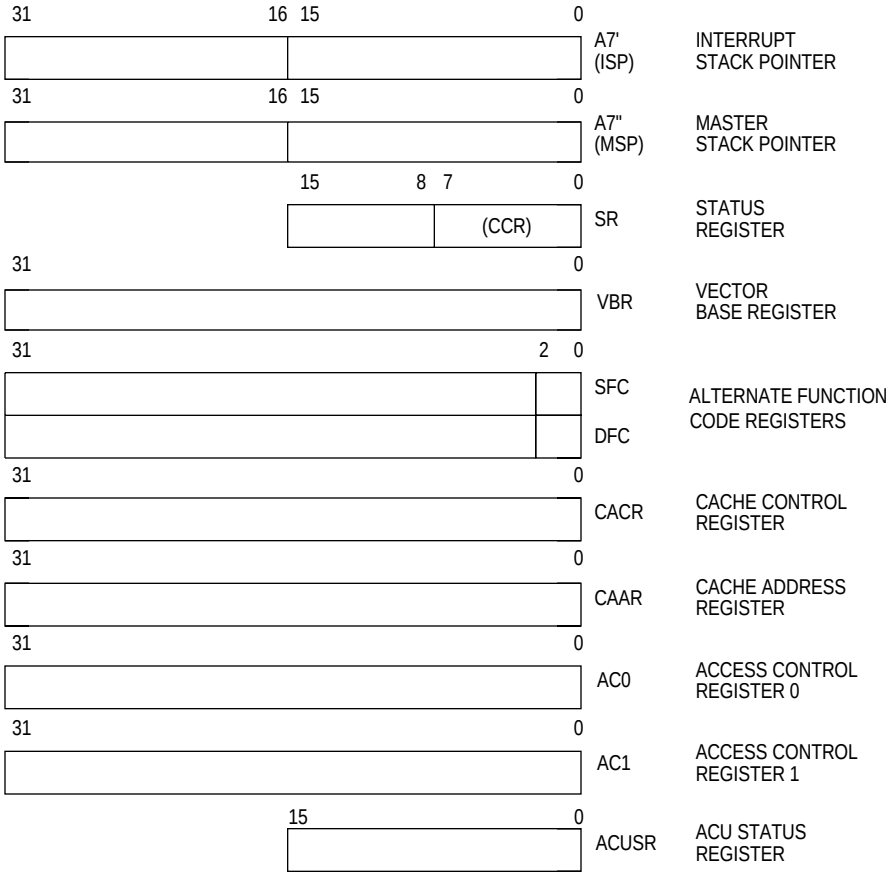


Figure 4. Supervisor Programming Model Supplement

The status register (see Figure 5) contains the interrupt priority mask (three bits) as well as the following condition codes: extend (X), negate (N), zero (Z), overflow (V), and carry (C). Additional control bits indicate that the controller is in the trace mode (T1 or T0), supervisor/user state (S), and master/interrupt state (M).

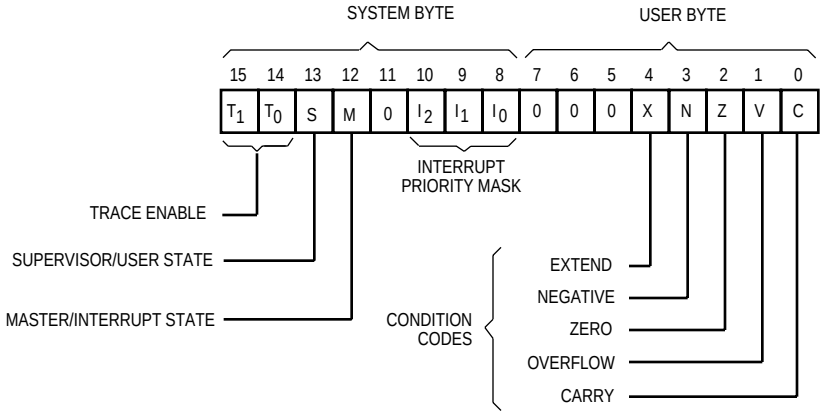


Figure 5. Status Register

In addition, operations on other data types, such as memory addresses, status word data, etc., are provided in the instruction set. The coprocessor mechanism allows direct support of floating-point data types with the MC68881/MC68882 floating-point coprocessors as well as specialized user-defined data types and functions. The 18 addressing modes, listed in Table 1, include nine basic types:

- Register Direct
- Register Indirect
- Register Indirect with Index
- Memory Indirect
- Program Counter Indirect with Displacement
- Program Counter Indirect with Index
- Program Counter Memory Indirect
- Absolute
- Immediate

The register indirect addressing modes support postincrement, predecrement, offset, and indexing. These capabilities are particularly useful for handling advanced data structures common to sophisticated applications and high-level languages. The program counter relative mode also has index and offset capabilities; this addressing mode is generally required to support position- independent software. In addition to these addressing modes, the MC68EC030 provides data operand sizing and scaling; these features provide performance enhancements to the programmer.

Table 1. MC68EC030 Addressing Modes

Addressing Modes	Syntax
Register Direct Addressing Data Register Direct Address Register Direct	 Dn An
Register Indirect Address Register Indirect Address Register Indirect with Postincrement Address Register Indirect with Predecrement Address Register Indirect with Displacement	 (An) (An);pl -(An) (d_{16},An)
Register Indirect with Index Address Register Indirect with Index (8-Bit Displacement) Address Register Indirect with Index (Base Displacement)	 (d_8,An,Xn) (bd,An,Xn)
Memory Indirect Memory Indirect Postindexed Memory Indirect Preindexed	 ([bd,An],Xn,od) ([bd,An,Xn],od)
Program Counter Indirect with Displacement	(d_{16},PC)
Program Counter Indirect with Index PC Indirect with Index (8-Bit Displacement) PC Indirect with Index (Base Displacement)	 (d_8,PC,Xn) (bd,PC,Xn)
Program Counter Memory Indirect PC Memory Indirect Postindexed PC Memory Indirect Preindexed	 ([bd,PC],Xn,od) ([bd,PC,Xn],od)
Absolute Data Addressing Absolute Short Absolute Long	 xxx.W xxx.L
Immediate	#<data>

NOTES:

Dn = Data Register, D0–D7
An = Address Register, A0–A7
d_8, d_{16} = A twos-complement or sign-extended displacement; added as part of the effective address calculation; size is 8 (d_8) or 16 (d_{16}) bits; when omitted, assemblers use a value of zero.
Xn = Address or data register used as an index register; form is Xn.SIZE*SCALE, where SIZE is .W or .L (indicates index register size) and SCALE is 1, 2, 4, or 8 (index register is multiplied by SCALE); use of SIZE and/or SCALE is optional.
bd = A twos-complement base displacement; when present, size can be 16 or 32 bits.
od = Outer displacement added as part of effective address calculation after any memory indirection; use is optional with a size of 16 or 32 bits.
PC = Program Counter
<data> = Immediate value of 8, 16, or 32 bits
( ) = Effective Address
[ ] = Used as indirect address to long-word address.

INSTRUCTION SET OVERVIEW

The MC68EC030 instruction set is listed in Table 2. Each instruction, with few exceptions, operates on bytes, words, and long words, and most instructions can use any of the 18 addressing modes. The MC68EC030 is upward source- and object-level code compatible with the M68000 Family because it supports all instructions of previous family members.

Table 2. Instruction Set

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
ADDA	Add Address
ADDI	Add Immediate
ADDQ	Add Quick
ADDX	Add with Extend
AND	Logical AND
ANDI	Logical AND Immediate
ASL,ASR	Arithmetic Shift Left and Right
Bcc	Branch Conditionally
BCHG	Test Bit and Change
BCLR	Test Bit and Clear
BFCHG	Test Bit Field and Change
BFCLR	Test Bit Field and Clear
BFEXTS	Signed Bit Field Extract
BEFXTU	Unsigned Bit Field Extract
BFFFO	Bit Field Find First One
BFINS	Bit Field Insert
BFSET	Test Bit Field and Set
BFTST	Test Bit Field
BKPT	Breakpoint
BRA	Branch
BSET	Test Bit and Set
BSR	Branch to Subroutine
BTST	Test Bit
CAS	Compare and Swap Operands
CAS2	Compare and Swap Dual Operands
CHK	Check Register Against Bound
CHK2	Check Register Against Upper and Lower Bounds
CLR	Clear
CMP	Compare
CMPA	Compare Address
CMPI	Compare Immediate
CMPM	Compare Memory to Memory
CMP2	Compare Register Against Upper and Lower Bounds
DBcc	Test Condition, Decrement and Branch
DIVS,DIVSL	Signed Divide
DIVU, DIVUL	Unsigned Divide
EOR	Logical Exclusive OR
EORI	Logical Exclusive OR Immediate
EXG	Exchange Registers
EXT, EXTB	Sign Extend
ILLEGAL	Take Illegal Instruction Trap
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link and Allocate
LSL, LSR	Logical Shift Left and Right

Mnemonic	Description
MOVE	Move
MOVEA	Move Address
MOVE CCR	Move Condition Code Register
MOVE SR	Move Status Register
MOVE USP	Move User Stack Pointer
MOVEC	Move Control Register
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral
MOVEQ	Move Quick
MOVES	Move Alternate Address Space
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NEGX	Negate with Extend
NOP	No Operation
NOT	Logical Complement
OR	Logical Inclusive OR
ORI	Logical Inclusive OR Immediate
PACK	Pack BCD
PEA	Push Effective Address
PFLUSH	No Effect
PLOAD	No Effect
PMOVE	Move to/from ACx Registers
PTEST	Test Address in ACx Registers
RESET	Reset External Devices
ROL, ROR	Rotate Left and Right
ROXL, ROXR	Rotate with Extend Left and Right
RTD	Return and Deallocate
RTE	Return from Exception
RTR	Return and Restore Codes
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Scc	Set Conditionally
STOP	Stop
SUB	Subtract
SUBA	Subtract Address
SUBI	Subtract Immediate
SUBQ	Subtract Quick
SUBX	Subtract with Extend
SWAP	Swap Register Words
TAS	Test Operand and Set
TRAP	Trap
TRAPcc	Trap Conditionally
TRAPV	Trap on Overflow
TST	Test Operand
UNLK	Unlink
UNPK	Unpack BCD

Coprocessor Instructions

cpBCC	Branch Conditionally	cpRESTORE	Restore Internal State of Coprocessor
cpDBcc	Test Coprocessor Condition, Decrement and Branch	cpSAVE	Save Internal State of Coprocessor
		cpScc	Set Conditionally
cpGEN	Coprocessor General Instruction	cpTRAPcc	Trap Conditionally

Included in the MC68EC030 set are the bit field operations, binary-coded decimal support, bounds checking, additional trap conditions, and additional multiprocessing support (CAS and CAS2 instructions) offered by the MC68020, MC68030, and MC68040. In addition, object code written for the MC68EC030 can be used on the MC68040 for even more performance. The memory management unit (MMU) instructions of the MC68030, and MC68040 are not supported by the MC68EC030.

INSTRUCTION AND DATA CACHES

Studies have shown that typical programs spend most of their execution time in a few main routines or tight loops. This phenomenon, known as locality of reference, has an impact on program performance. The MC68010 takes limited advantage of this phenomenon with the loop mode of operation that can be used with the DBcc instruction. The MC68EC030 takes further advantage of cache technology to provide the system with two on-chip caches, one for instructions and one for data.

MC68EC030 CACHE GOALS

Similar to the MC68020 and MC68030, there were two primary design goals for the MC68EC030 embedded controller caches. The first design goal was to reduce the external bus activity of the CPU even more than was accomplished with the MC68020. The second design goal was to increase effective CPU throughput as larger memory sizes or slower memories increased average access time. By placing a high-speed cache between the controller and the rest of the memory system, the effective memory access time becomes:

$$t_{acc} = R_h * t_{cache} + (1-R_h) * t_{ext}$$

where t_{acc} is the effective system access time, t_{cache} is the cache access time, t_{ext} is the access time of the rest of the system, and R_h is the hit ratio or the percentage of time that the data is found in the cache. Thus, for a given system design, the two MC68EC030 on-chip caches provide an even more substantial CPU performance increase over that obtainable with the MC68020 instruction cache. Alternately, slower and less expensive memories can be used for the same controller performance.

The throughput increase in the MC68EC030 is gained in three ways. First, the MC68EC030 caches are accessed in less time than is required for external accesses, providing improvement in the access time for items residing in the cache. Second, the burst filling of the caches allows instruction and data words to be found in the on-chip caches the first time they are accessed by the micromachine, minimizing the time required to bring those items into the cache. Utilizing burst fill capabilities lowers the average access time for items found in the caches even further. Third, the autonomous nature of the caches allows instruction stream fetches, data fetches, and external bus activity to occur simultaneously with instruction execution. The parallelism designed into the MC68EC030 also allows multiple instructions to execute concurrently so that several internal instructions (those that do not require any external accesses) can execute while the controller is performing an external access for a previous instruction.

INSTRUCTION CACHE

DATA CACHE

The organization of the data cache (see Figure 7) is similar to that of the instruction cache. However, the tag is composed of the upper 24 address bits, the four valid bits, and all three function code bits, explicitly specifying the address space associated with each line. The data cache employs a write-through policy with programmable write allocation of data writes— i.e., if a cache hit occurs on a write cycle, both the data cache and the external device are updated with the new data. If a write cycle generates a cache miss, the external device is updated, and a new data cache entry can be replaced or allocated for that address, depending on the state of the write-allocate (WA) bit in the CACR.

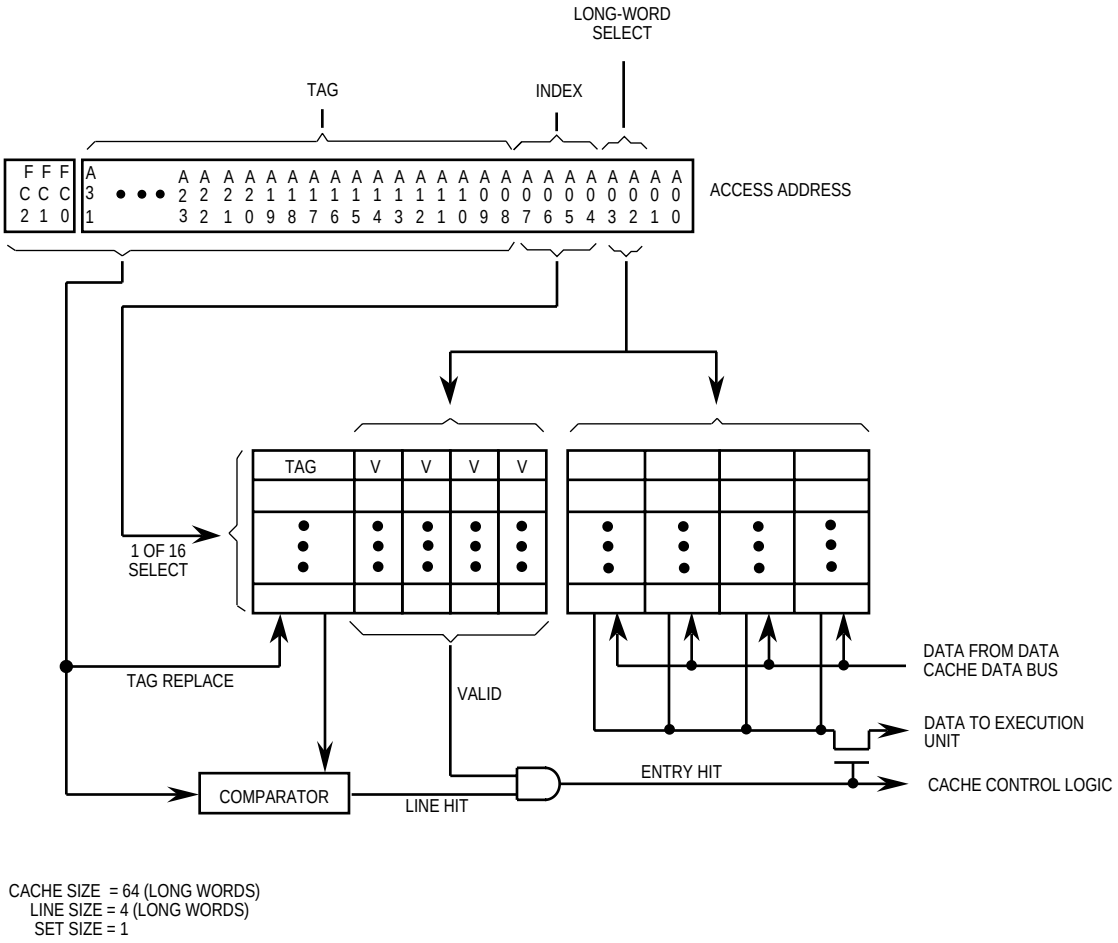


Figure 7. On-Chip Data Cache Organization

OPERAND TRANSFER MECHANISM

The MC68EC030 offers three different mechanisms by which data can be transferred into and out of the chip. Asynchronous bus cycles, compatible with the asynchronous bus on the MC68020 and MC68030, can transfer data in a minimum of three clock cycles; the amount of data transferred on each cycle is determined by the dynamic bus sizing mechanism on a cycle-by-cycle basis with the data transfer and size acknowledge (DSACKx) signals. Synchronous bus cycles, compatible with the synchronous bus on the MC68030, are terminated with the synchronous termination (STERM) signal and always transfer 32-bits of data in a minimum of two clock cycles, increasing the bus bandwidth available for other bus masters,

thereby increasing possible performance. Burst mode transfers can be used to fill lines of the instruction and data caches when the MC68EC030 asserts cache burst request (CBREQ). After completing the first cycle with STERM, subsequent cycles may accept data on every clock cycle where STERM is asserted until the burst is completed. Use of this mode can further increase the available bus bandwidth in systems that use DRAMs with page, nibble, or static-column mode operation.

ASYNCHRONOUS TRANSFERS

Though the MC68EC030 has a full 32-bit data bus, it offers the ability to automatically and dynamically downsize its bus to 8 or 16 bits if peripheral devices are unable to accommodate the entire 32 bits. This feature allows the programmer to write code that is not bus-width specific. For example, long-word (32 bit) accesses to peripherals may be used in the code; yet, the MC68EC030 will transfer only the amount of data that the peripheral can manage. This feature allows the peripheral to define its port size as 8, 16, or 32 bits wide, and the MC68EC030 will dynamically size the data transfer accordingly, using multiple bus cycles when necessary. Hence, programmers are not required to program for each device port size or know the specific port size before coding; hardware designers have the flexibility to choose hardware implementations regardless of software implementations.

The dynamic bus sizing mechanism is invoked by DSACKx and occurs on a cycle-by-cycle basis. For example, if the controller is executing an instruction that requires reading a long-word operand, it will attempt to read 32 bits during the first bus cycle to a long-word address boundary. If the port responds that it is 32 bits wide, the MC68EC030 latches all 32 bits of data and continues. If the port responds that it is 16 bits wide, the MC68EC030 latches the 16 valid bits of data and continues. An 8-bit port is handled similarly but has four bus read cycles. Each port is fixed in the assignment to particular sections of the data bus. However, the MC68EC030 has no restrictions concerning the alignment of operands in memory; long-word operands need not be aligned to long-word address boundaries. When misaligned data requires multiple bus cycles, the MC68EC030 automatically runs the minimum number of bus cycles. Instructions must still be aligned to word boundaries.

The timing of asynchronous bus cycles is also determined by the assertion of DSACKx on a cycle-by-cycle basis. If the DSACKx signals are valid 1.5 clocks after the beginning of the bus cycle (with the appropriate setup time), the cycle terminates in the minimum amount of time (corresponding to three-clock-cycle total). The cycle can be lengthened by delaying DSACKx (effectively inserting wait states in one-clock increments) until the device being accessed is able to terminate the cycle. This flexibility gives the controller the ability to communicate with devices of varying speeds while operating at the fastest rate possible for each device.

The asynchronous transfer mechanism allows external errors to abort cycles upon the assertion of bus error (BERR) or allows individual bus cycles to be retried with the simultaneous assertion of BERR and HALT.

ACCESS CONTROL

Two access control registers are provided on the MC68EC030 to control cachability of accesses for two independent blocks of memory. Each block can range in size from 16 Mbytes to 2 Gbytes, and is specified in the corresponding ACx register with a base address, a base mask, function code, function code mask, and read/write mask. A typical use for an access control register is to designate a block of memory containing I/O devices as non-cachable.

COPROCESSOR INTERFACE

The coprocessor interface is a mechanism for extending the instruction set of the M68000 Family. The interface provided on the MC68EC030 is the same as that on the MC68020 and MC68030. Examples of these extensions are the addition of specialized data operands for the existing data types or, for the case of floating point, the inclusion of new data types and operations implemented by the MC68881/MC68882 floating-point coprocessors.

SIGNAL DESCRIPTION

Figure 8 illustrates the functional signal groups, and Table 3 describe the signals and their function.

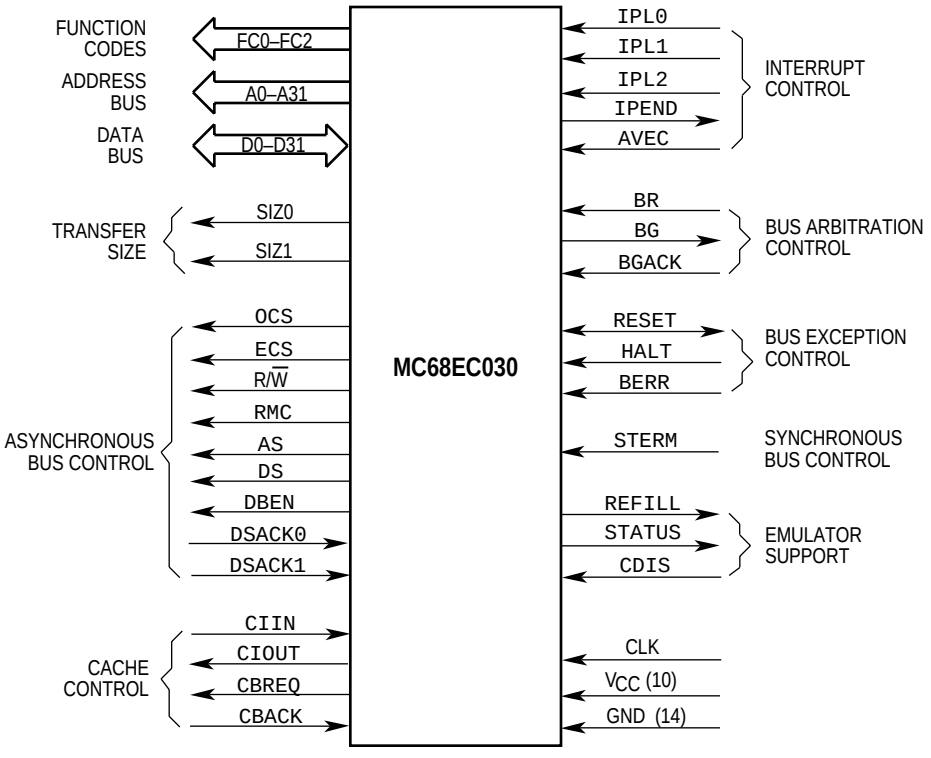


Figure 8. Functional Signal Groups

Table 3. Signal Index

Signal Name	Mnemonic	Function
Function Codes	FC0–FC2	3-bit function code used to identify the address space of each bus cycle.
Address Bus	A0–A31	32-bit address bus.
Data Bus	D0–D31	32-bit data bus used to transfer 8, 16, 24, or 32 bits of data per bus cycle.
Size	SIZ0–SIZ1	Indicates the number of bytes remaining to be transferred for this cycle. These signals, together with A0 and A1, define the active sections of the data bus.
Operand Cycle Start	OCS	Identical operation to that of ECS except that OCS is asserted only during the first bus cycle of an operand transfer
External Cycle Start	ECS	Provides an indication that a bus cycle is beginning.
Read/Write	R/W	Defines the bus transfer as a controller read or write.
Read-Modify-Write Cycle	RMC	Provides an indicator that the current bus cycle is part of an indivisible read-modify-write operation.
Address Strobe	AS	Indicates that a valid address is on the bus.
Data Strobe	DS	Indicates that valid data is to be placed on the data bus by an external device or has been replaced by the MC68EC030.
Data Buffer Enable	DBEN	Provides an enable signal for external data buffers.
Data Transfer and Size Acknowledge	DSACK0, DSACK1	Bus response signals that indicate the requested data transfer operation has completed. In addition, these two lines indicate the size of the external bus port on a cycle-by-cycle basis and are used for asynchronous transfers.
Synchronous Termination	STERM	Bus response signal that indicates a port size of 32 bits and that data may be latched on the next falling clock edge.
Cache Inhibit In	CIIN	Prevents data from being loaded into the MC68EC030 instruction and data caches.
Cache Inhibit Out	CIOUT	Reflects the CI bit in ACx registers; indicates that external caches should ignore these accesses.
Cache Burst Request	CBREQ	Indicates a burst request for the instruction or data cache.
Cache Burst Acknowledge	CBACK	Indicates that the accessed device can operate in burst mode.
Interrupt Priority Level	IPL0–IPL2	Provides an encoded interrupt level to the controller.
Interrupt Pending	IPEND	Indicates that an interrupt is pending.
Autovector	AVEC	Requests an autovector during an interrupt acknowledge cycle.
Bus Request	BR	Indicates that an external device requires bus mastership.
Bus Grant	BG	Indicates that an external device may assume bus mastership.
Bus Grant Acknowledge	BGACK	Indicates that an external device has assumed bus mastership.
Reset	RESET	System reset.
Halt	HALT	Indicates that the controller should suspended bus activity.
Bus Error	BERR	Indicates that an erroneous bus operation is being attempted.
Cache Disable	CDIS	Dynamically disables the on-chip cache to assist emulator support.
Pipe Refill	REFILL	Indicates when the MC68EC030 is beginning to fill pipeline.
Microsequencer Status	STATUS	Indicates the state of the microsequencer.

Clock	CLK	Clock input to the controller.

Table 3. Signal Index – Continued

Signal Name	Mnemonic	Function
Power Supply	V_{CC}	Power supply.
Ground	GND	Ground connection.
No Connect	NC	Do not connect.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 to +7.0	V
Input Voltage	V_{in}	-0.5 to +7.0	V
Operating Temperature Range Minimum Ambient Temperature Maximum Ambient Temperature	 T_A T_A	 0 70	℃
Storage Temperature Range	T_{stg}	-55 to 150	℃

The device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, normal precautions should be taken to avoid application of voltages higher than maximum-rated voltages to these high-impedance circuits. Tying unused inputs to the appropriate logic voltage level (e.g., either GND or V_{CC}) enhances reliability of operation.

THERMAL CHARACTERISTICS-- PGA PACKAGE

Characteristic	Symbol	Value	Rating
Thermal Resistance - Plastic Junction to Ambient Junction to case	 θ_{JA} θ_{JC}	 32 TBD	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, TJ, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θJ_A = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications, $P_{I/O} < P_{INT}$ and can be neglected.

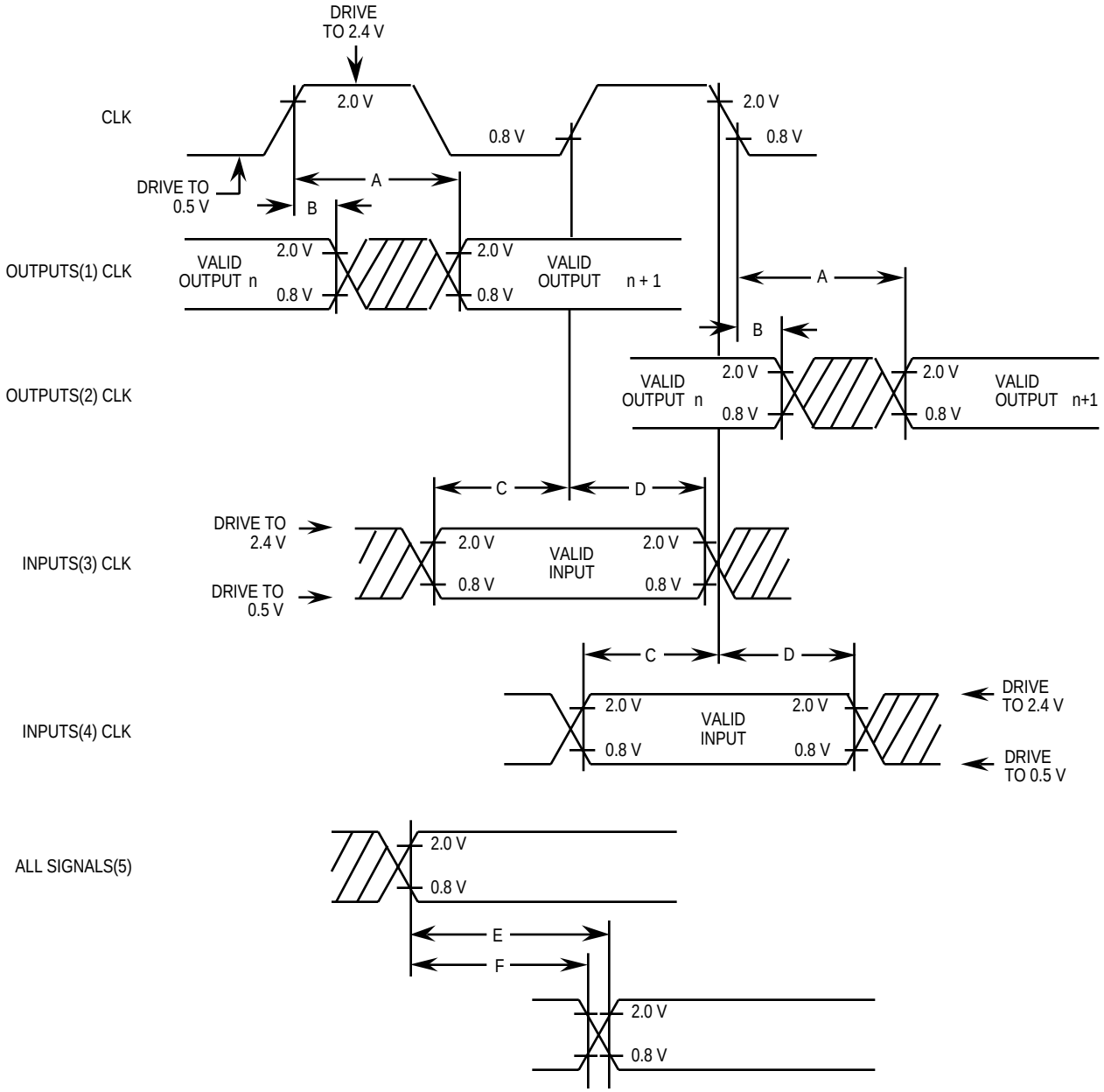
The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving Equations (1) and (2) for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

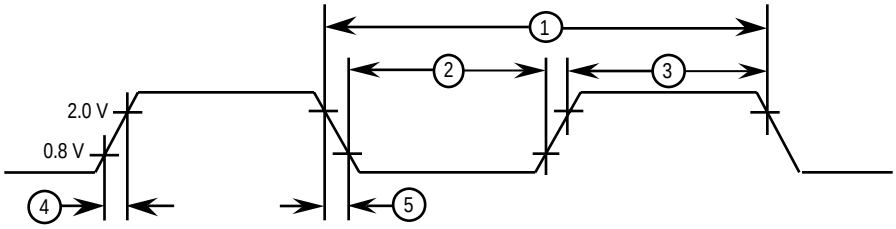
where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at thermal equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.



NOTES:
1. This output timing is applicable to all parameters specified relative to the rising edge of the clock.
2. This output timing is applicable to all parameters specified relative to the falling edge of the clock.
3. This input timing is applicable to all parameters specified relative to the rising edge of the clock.
4. This input timing is applicable to all parameters specified relative to the falling edge of the clock.
5. This timing is applicable to all parameters specified relative to the assertion/negation of another signal.

LEGEND:
A. Maximum output delay specification.
B. Minimum output hold time.
C. Minimum input setup time specification.
D. Minimum input hold time specification.
E. Signal valid to signal valid specification (maximum or minimum).
F. Signal valid to signal invalid specification (maximum or minimum).

Figure 9. Drive Levels and Test Points for AC Specifications



NOTE: 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 so that the rise or fall will be linear between 0.8 V and 2.0 V.

Figure 10. Clock Input Timing Diagram

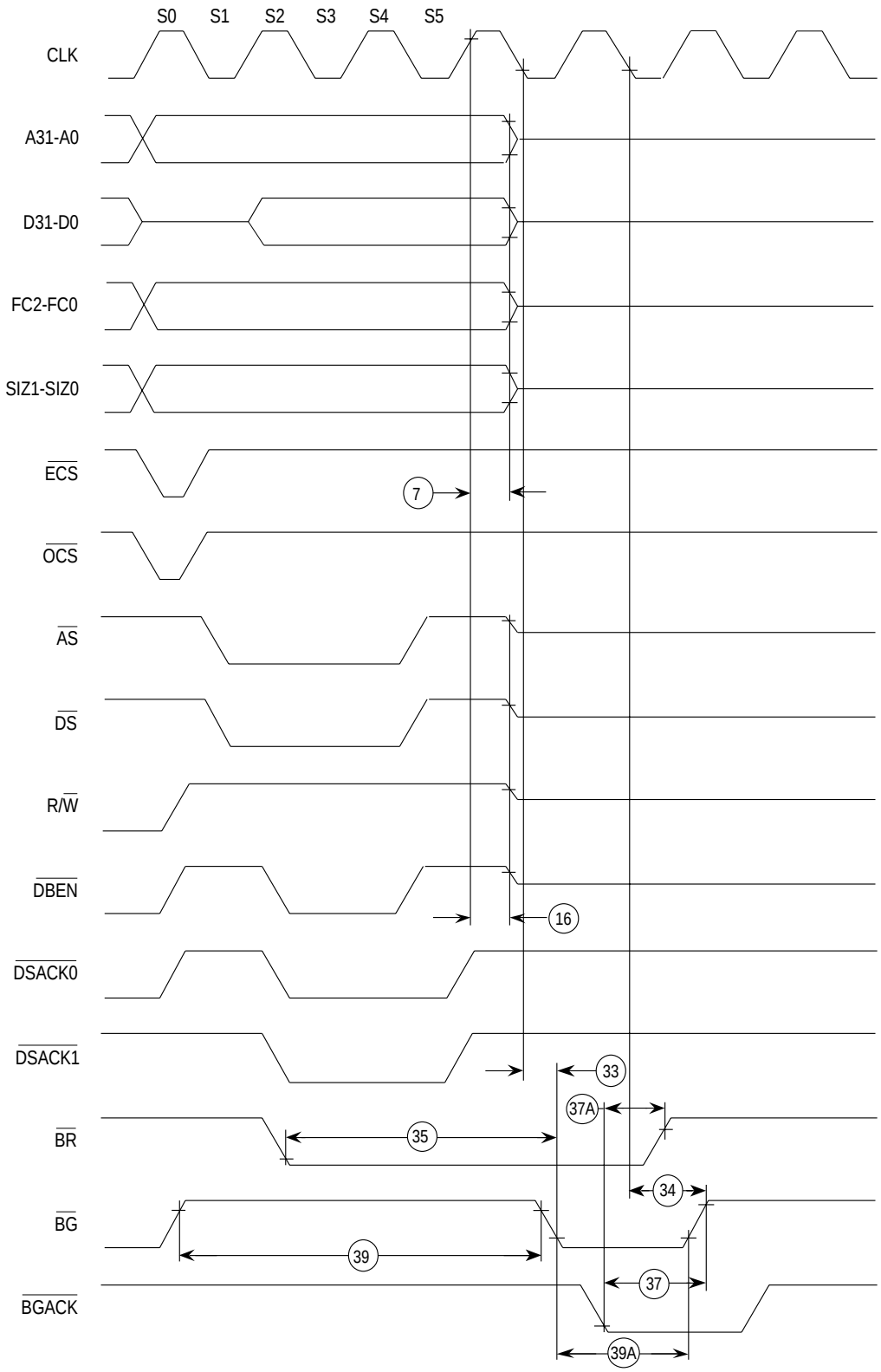
AC ELECTRICAL SPECIFICATIONS -- READ AND WRITE CYCLES

(V_{CC}=5.0Vdc ± 5%; GND=0 Vdc; temperature in defined ranges; see Figures 11–16)

Num.	Characterstics	25MHz		40 MHz		Unit
		Min	Max	Min	Max	
6	Clock High to Function Code, Size, RMC, IPEND,CIOUT, Address Valid	0	20	0	14	ns
6A	Clock High to ECS, OCS Asserted	0	15	0	10	ns
6B	Function Code, Size, RMC, IPEND, CIOUT, Address Valid to Negating Edge of ECS	3	—	3	—	ns
7	Clock High to Function Code Size, RMC, CIOUT, Address Data High Impedance	0	40	0	25	ns
8	Clock High to Function Code Size, RMC, IPEND, CIOUT, Address Invalid	0	—	0	—	ns
9	Clock Low to AS, DS Asserted, CBREQ Valid	3	18	2	10	ns
9A[1]	AS to DS Assertion Skew (Read)	-10	10	-6	6	ns
9B[14]	AS Asserted to DS Asserted (Write)	27	—	16	—	ns
10	ECS Width Asserted	10	—	5	—	ns
10A	OCS Width Asserted	10	—	5	—	ns
10B[7]	ECS, OCS Width Negated	5	—	5	—	ns
11	Function Code, Size, RMC, CIOUT, Address Valid to AS Asserted (and DS Asserted, Read)	7	—	5	—	ns
12	Clock Low to AS, DS, CBREQ Negated	0	18	0	10	ns
12A	Clock Low to ECS/OCS Negated	0	18	0	12	ns
13	AS, DS Negated to Function Code, Size, RMC CIOUT, Address Invalid	7	—	3	—	ns
14	AS (and DS Read) Width Asserted (Asynchronous Cycle)	70	—	30	—	ns
14A[11]	DS Width Asserted (Write)	30	—	18	—	ns
14B	AS (and DS, Read) Width Asserted (Synchronous Cycle)	30	—	18	—	ns
15	AS, DS Width Negated	30	—	18	—	ns

AC ELECTRICAL SPECIFICATIONS — READ AND WRITE CYCLES (Continued)

Num.	Characterstics	25MHz		40 MHz		Unit
		Min	Max	Min	Max	
15A[8]	DS Negated to AS Asserted	25	—	16	—	ns
16	Clock High to AS, DS, R/W, DBEN, CBREQ High Impedance	—	40	—	25	ns
17	AS, DS Negated to R/W Invalid	7	—	3	—	ns
18	Clock High to R/W High	0	20	0	14	ns
20	Clock High to R/W Low	0	20	0	14	ns
21	R/W High to AS Asserted	7	—	5	—	ns
22	R/W Low to DS Asserted (Write)	47	—	24	—	ns
23	Clock High to Data-Out Valid	—	20	—	14	ns
24	Data-Out Valid to Negating Edge of AS	5	—	3	—	ns
25[11]	AS, DS Negated to Data-Out Invalid	7	—	3	—	ns
25A[9,11]	DS Negated to DBEN Negated (Write)	7	—	3	—	ns
26[11]	Data-Out Valid to DS Asserted (Write)	7	—	3	—	ns
27	Data-In Valid to Clock Low (Setup)	2	—	1	—	ns
27A	Late BERR/HALT Asserted to Clock Low (Setup)	5	—	3	—	ns
28[12]	AS, DS Negated to DSACKx, BERR, HALT, AVEC Negated (Asynchronous Hold)	0	40	0	20	ns
28A[12]	Clock Low to DSACKx, BERR, HALT, AVEC Negated (Synchronous Hold)	8	70	6	40	ns
29[12]	AS, DS Negated to Data-In Invalid (Asynchronous Hold)	0	—	0	—	ns
29A[12]	AS, DS Negated to Data-In High Impedance	—	40	—	25	ns
30[12]	Clock Low to Data-In Invalid (Synchronous Hold)	8	—	6	—	ns
30A[12]	Clock Low to Data-In High Impedance (Read followed by Write)	—	60	—	30	ns
31[2]	DSACKx Asserted to Data-In Valid (Asynchronous Data Setup)	—	28	—	14	ns
31A[3]	DSACKx Asserted to DSACKx Valid (Skew)	—	7	—	3	ns
32	RESET Input Transition Time	—	1.5	—	1.5	Clks
33	Clock Low to BG Asserted	0	20	0	14	ns
34	Clock Low to BG Negated	0	20	0	14	Clks
35	BR Asserted to BG Asserted (RMC Not Asserted)	1.5	3.5	1.5	3.5	Clks
37	BGACK Asserted to BG Negated	1.5	3.5	1.5	3.5	Clks
37A[6]	BGACK Asserted to BR Negated	0	1.5	0	1.5	ns
39	BG Width Negated	60	—	30	—	ns
39A	BG Width Asserted	60	—	30	—	ns
40	Clock High to DBEN Asserted (Read)	0	20	0	16	ns
41	Clock Low to DBEN Negated (Read)	0	20	0	16	ns
42	Clock Low to DBEN Asserted (Write)	0	20	0	16	ns
43	Clock High to DBEN Negated (Write)	0	20	0	16	ns



NOTE: 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 so that the rise or fall will be linear between 0.8 V and 2.0 V.

Figure 15. Bus Arbitration Timing Diagram

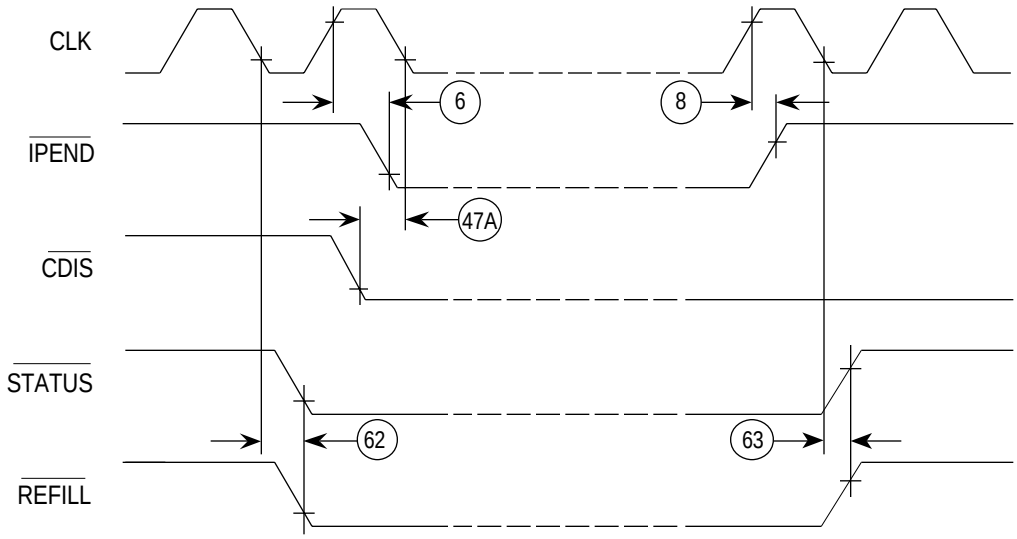
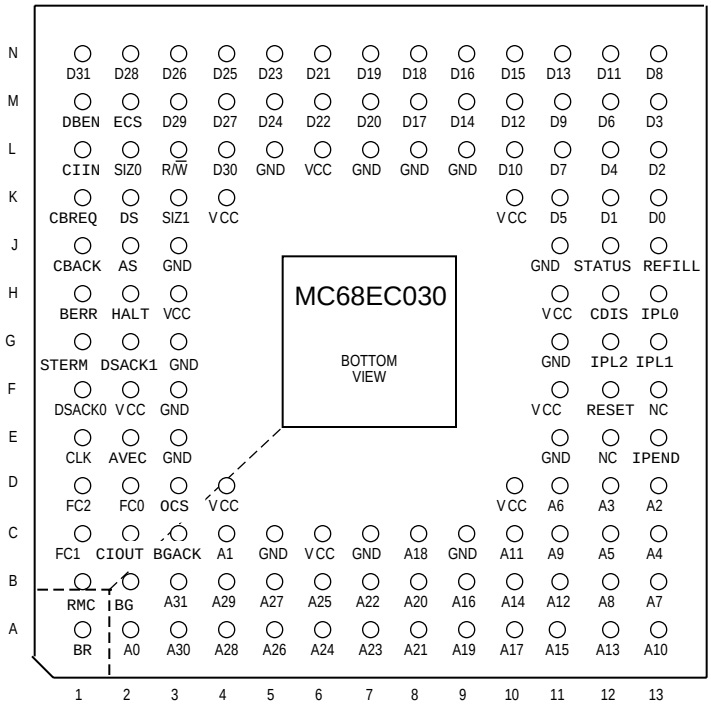


Figure 16. Other Signal Timings

MECHANICAL DATA

PIN ASSIGNMENTS — PIN GRID ARRAY (RC SUFFIX)



NOTE

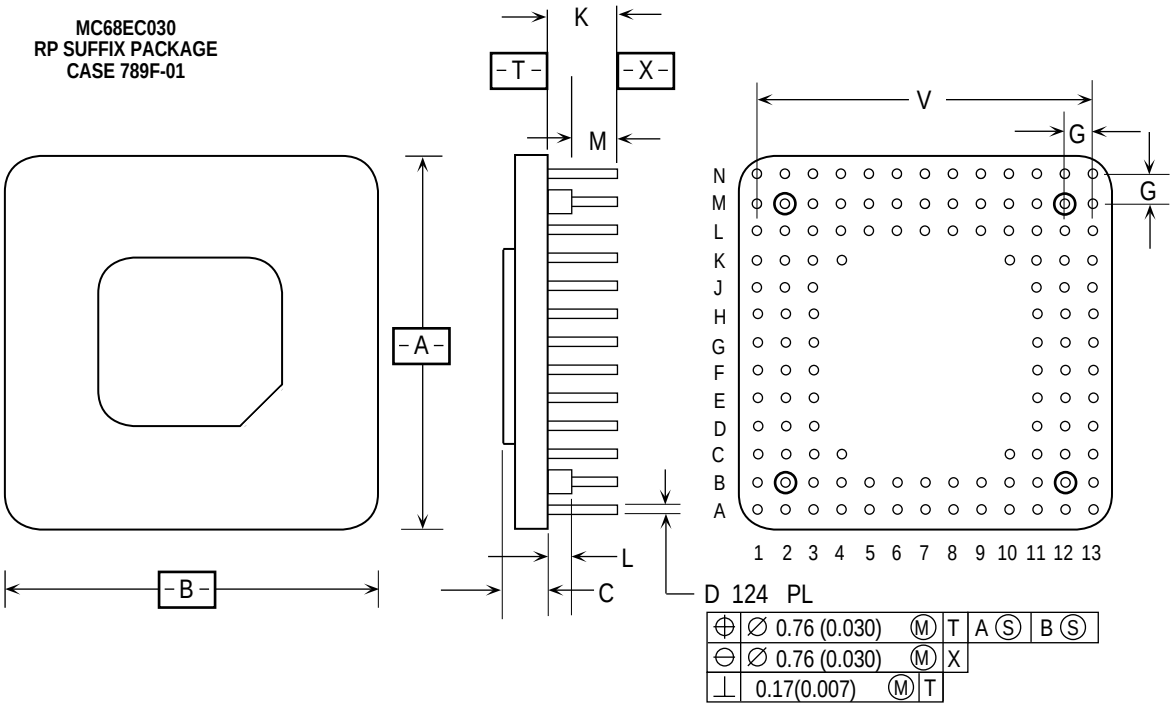
The MC68030 has four additional guide pins not present on the MC68EC030. Therefore, an MC68EC030 fits in a socket designed for the MC68030, but the MC68030 does not necessary fit in a socket intended for the MC68EC030.

The Vcc and GND pins are separated into three groups to provide individual power supply connections for the address bus buffers, data bus buffers, and all other output buffers and internal logic

Pin Group	VCC	GND
Address Bus	C6, D10	C5, C7, C9, E11
Data Bus	L6, K10	J11, L9, L7, L5
ECS, SIZx, DS, AS, DBEN, CBREQ, R/W	K4	J3
FC0–FC2, RMS, OCS, CIOUT, BG	D4	E3
Internal Logic, RESET, STATUS, REFILL, Misc	H3, F2, F11, H11	L8, G3, F3, G11

PACKAGE DIMENSIONS

MC68EC030
RP SUFFIX PACKAGE
CASE 789F-01



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	34.04	35.05	1.340	1.380
B	34.04	35.05	1.340	1.380
C	2.92	3.18	0.115	0.135
D	0.44	0.55	0.017	0.022
G	2.54 BSC		0.100 BSC	
K	4.32	4.95	0.170	0.195
L	1.02	1.52	0.040	0.060
M	2.79	3.81	0.110	0.150
V	30.48 BSC		1.200 BSC	

NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH
3. DIMENSION D INCLUDES LEAD FINISH.