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	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	132-BCQFP
Supplier Device Package	132-CQFP (24x24)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68ec030fe40c

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.

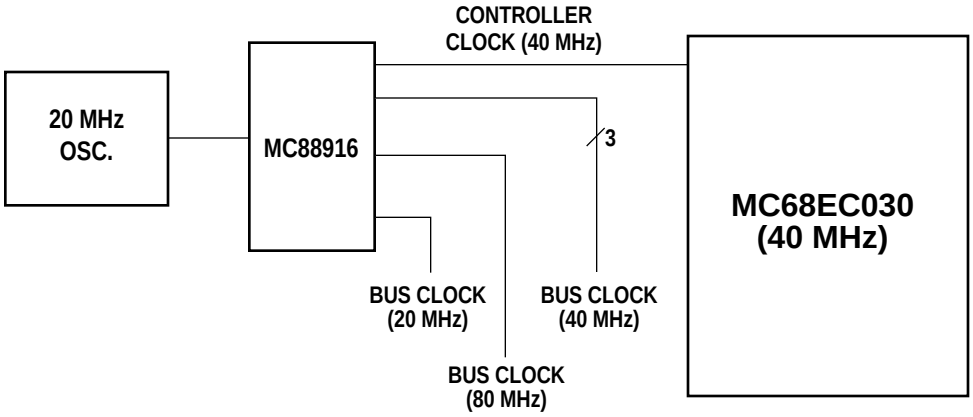


Figure 1. MC68EC030 Clock Circuitry

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

The ACU contains two access control registers that are used to define memory segments ranging in size from 16 Mbytes to 2 Gbytes each. Each segment is definable in terms of address, read/write access, and function code. Each segment can be marked as cacheable or non cacheable to control cache accesses to that memory space.

PROGRAMMING MODEL

As shown in the programming models (see Figures 3 and 4), the MC68EC030 has 16 32-bit general-purpose registers, a 32-bit program counter, two 32-bit supervisor stack pointers, a 16-bit status register, a 32-bit vector base register, two 3-bit alternate function code registers, two 32-bit cache handling (address and control) registers, and two 32-bit transparent translation registers. Registers D0–D7 are used as data registers for bit and bit field (1 to 32 bit), byte (8 bit), word (16 bit), long-word (32 bit), and quad-word (64 bit) operations. Registers A0–A6 and the user, interrupt, and master stack pointers are address registers that may be used as software stack pointers or base address registers. In addition, the address registers may be used for word and long-word operations. All 16 general-purpose registers (D0–D7, A0–A7) can be used as index registers.

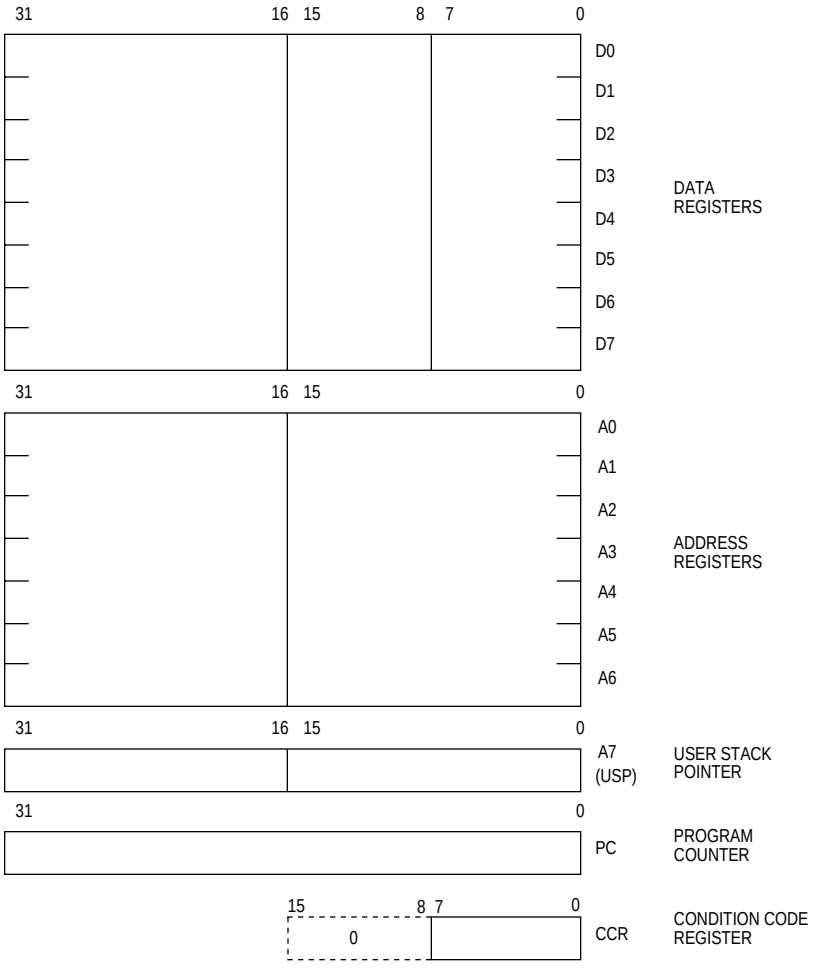


Figure 3. User Programming Model

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.

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

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.

SYNCHRONOUS TRANSFERS

Synchronous bus cycles are terminated by asserting STERM, which automatically indicates that the bus transfer is for 32 bits. Since this input is not synchronized internally, two-clock-cycle bus accesses can be performed if the signal is valid one clock after the beginning of the bus cycle with the appropriate setup time. However, the bus cycle may be lengthened by delaying STERM (inserting wait states in one-clock increments) until the device being accessed is able to terminate the cycle. After the assertion of STERM, these cycles may be aborted upon the assertion of BERR, or they may be retried with the simultaneous assertion of BERR and HALT.

BURST READ CYCLES

The MC68EC030 provides support for burst filling of its on-chip instruction and data caches, adding to the overall system performance. The on-chip caches are organized with a line size of four long words; there is only one tag for the four long words in a line. Since locality of reference is present to some degree in most programs, filling of all four entries when a single entry misses can be advantageous, especially if the time spent filling the additional entries is minimal. When the caches are burst filled, data can be latched by the controller in as little as one clock for each 32 bits. Burst read cycles can be performed only when the MC68EC030 requests them (with the assertion of CBREQ) and only when the first cycle is a synchronous cycle as previously described. If the cache burst acknowledge (CBACK) input is valid at the appropriate time in the synchronous bus cycle, the controller keeps the original AS, DS, R/W, address, function code, and size outputs asserted and latches 32 bits from the data bus at the end of each subsequent clock cycle that has STERM asserted. This procedure continues until the burst is complete (the entire block has been transferred), BERR is asserted in lieu of or after STERM, the cache inhibit in (CIIN) input is asserted, or the CBACK input is negated. The cache preloading allowed by the bursting enables the MC68EC030 to take advantage of cost-effective DRAM technology with minimal performance impact.

EXCEPTIONS

The types of exceptions and the exception processing sequence are discussed in the following paragraphs.

TYPES OF EXCEPTIONS

Exceptions can be generated by either internal or external causes. The externally generated exceptions are interrupts, BERR, and RESET. Interrupts are requests from peripheral devices for controller action; whereas, BERR and RESET are used for access control and controller restart. The internally generated exceptions come from instructions, address errors, tracing, or breakpoints. The TRAP, TRAPcc, TRAPVcc, cpTRAPcc, CKH, CKH2, and DIV instructions can all generate exceptions as part of instruction execution. Tracing behaves like a very high-priority, internally generated interrupt whenever it is processed. The other internally generated exceptions are caused by illegal instructions, instruction fetches from odd addresses, and privilege violations.

EXCEPTION PROCESSING SEQUENCE

Exception processing occurs in four steps. During the first step, an internal copy is made of the status register. After the copy is made, the special controller state bits in the status register are changed. The S-bit is set, putting the controller into the supervisor state. Also, the T1 and T0 bits are negated, allowing the exception handler to execute unhindered by tracing. For the reset and interrupt exceptions, the interrupt priority mask is also updated.

In the second step, the vector number of the exception is determined. For interrupts, the vector number is obtained by a controller read that is classified as an interrupt acknowledge cycle. For coprocessor-detected exceptions, the vector number is included in the coprocessor exception primitive response. For all other exceptions, internal logic provides the vector number. This vector number is then used to generate the address of the exception vector.

The third step is to save the current controller status. The exception stack frame is created and filled on the current supervisor stack. To minimize the amount of machine state that is saved, various stack frame sizes are used to contain the controller state, depending on the type of exception and where it occurred during instruction execution. If the exception is an interrupt and the M-bit is set, the M-bit is then cleared, and the short four-word exception stack frame that is saved on the master stack is also saved on the interrupt stack. If the exception is a reset, the M-bit is simply cleared, and the reset vector is accessed.

The MC68EC030 provides the same extensions to the exception stacking process as the MC68020, MC68030, and MC68040. If the M-bit is set, the master stack pointer (MSP) is used for all task-related exceptions. When a nontask-related exception occurs (i.e., an interrupt), the M bit is cleared, and the interrupt stack pointer (ISP) is used. This feature allows all the task's stack area to be carried within a single controller control block, and new tasks can be initiated by simply reloading the MSP and setting the M-bit.

The fourth and last step of exception processing is the same for all exceptions. The exception vector offset is determined by multiplying the vector number by four. This offset is then added to the contents of the vector base register (VBR) to determine the memory address of the exception vector. The new program counter is fetched from the exception vector. The instruction at the address given in the exception vector is fetched, and normal instruction decoding and execution is started.

STATUS and REFILL

The MC68EC030 provides the STATUS and REFILL signals to identify internal microsequencer activity associated with the processing of data pipelined in the pipeline. Since bus cycles are independently controlled and scheduled by the bus controller, information concerning the processing state of the microsequencer is not available by monitoring bus signals by themselves. The internal activity identified by the STATUS and REFILL signals include instruction boundaries, some exception conditions, when the microsequencer has halted, and instruction pipeline refills. STATUS and REFILL track only the internal microsequencer activity and are not directly related to bus activity.

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.

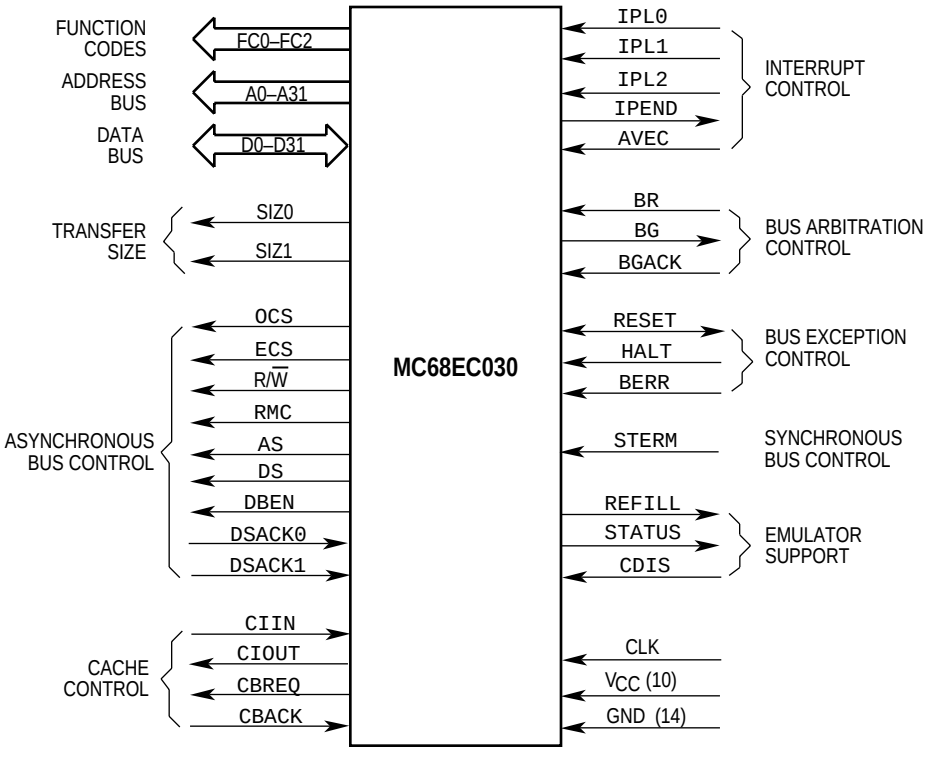


Figure 8. Functional Signal Groups

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.

The total thermal resistance of a package (θ_{JA}) can be separated into two components, θ_{JC} and θ_{CA}, representing the barrier to heat flow from the semiconductor junction to the package (case) surface (θ_{JC}) and from the case to the outside ambient air (θ_{CA}). These terms are related by the equation:

$$\theta_{JA}=\theta_{JC}+\theta_{CA} \quad (4)$$

θ_{JC} is device related and cannot be influenced by the user. However, θ_{CA} is user dependent and can be minimized by such thermal management techniques as heat sinks, ambient air cooling, and thermal convection. Thus, good thermal management on the part of the user can significantly reduce θ_{CA} so that θ_{JA} approximately equals; θ_{JC}. Substitution of θ_{JC} for θ_{JA} in equation (1) results in a lower semiconductor junction temperature.

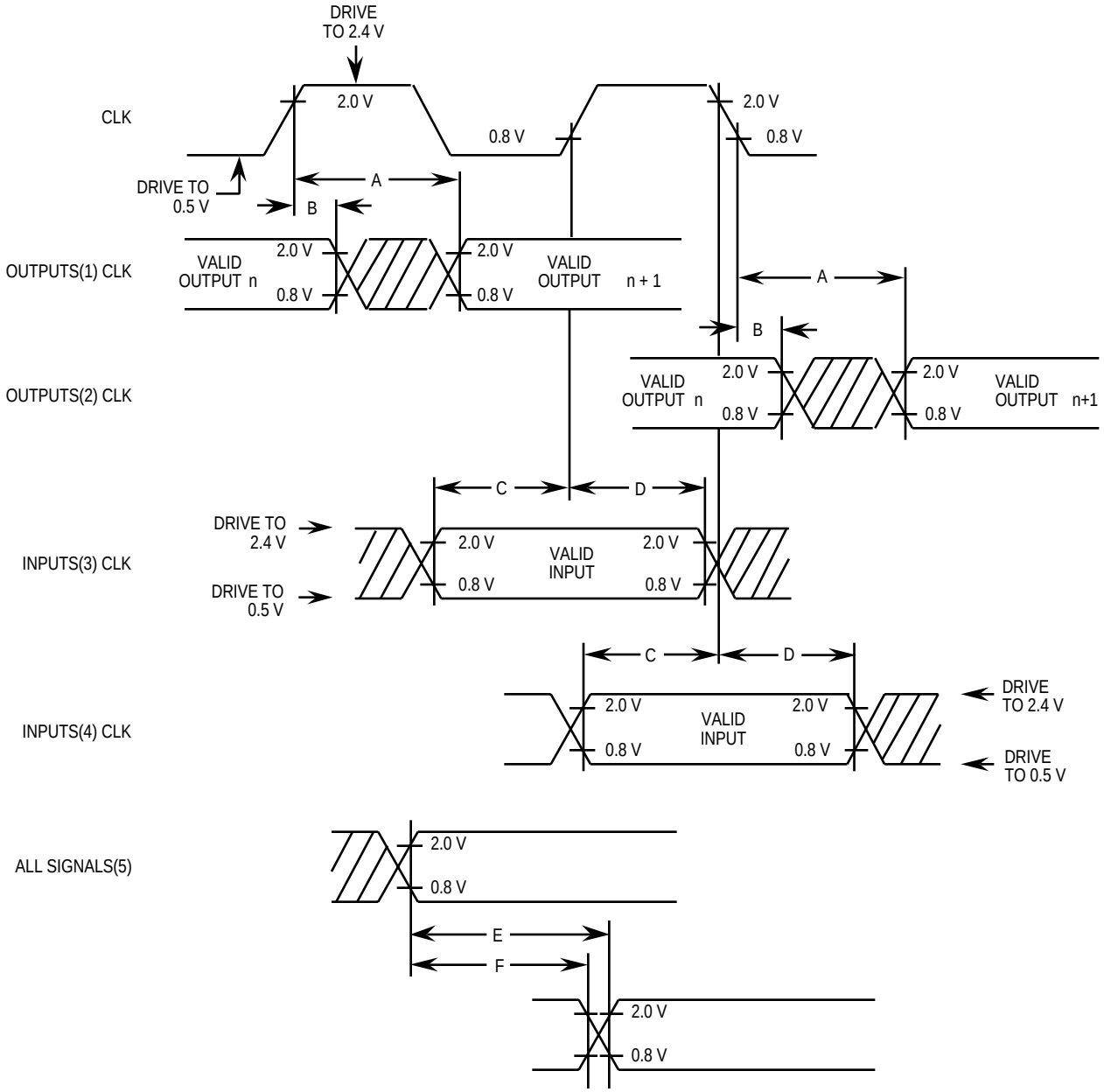
Values for thermal resistance presented in this document, unless estimated, were derived using the procedure described in Motorola Reliability Report 7843, "Thermal Resistance Measurement Method for MC68XX Microcomponent Devices," and are provided for design purposes only. Thermal measurements are complex and dependent on procedure and setup. User derived values for thermal resistance may differ.

AC ELECTRICAL SPECIFICATION DEFINITIONS

The AC specifications presented consist of output delays, input setup and hold times, and signal skew times. All signals are specified relative to an appropriate edge of the clock and possibly to one or more other signals.

The measurement of the AC specifications is defined by the waveforms shown in Figure 9. To test the parameters guaranteed by Motorola, inputs must be driven to the voltage levels specified in Figure 9. Outputs are specified with minimum and/or maximum limits, as appropriate, and are measured as shown in Figure 9. Inputs are specified with minimum setup and hold times, and are measured as shown. Finally, the measurement for signal-to-signal specifications is also shown.

Note that the testing levels used to verify conformance to the AC specifications does not affect the guaranteed DC operation of the device as specified in the DC electrical specifications.



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

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

AC ELECTRICAL SPECIFICATIONS — READ AND WRITE CYCLES
(Concluded)

Num.	Characterstics		25 MHz		40 MHz		Unit
			Min	Max	Min	Max	
44	R/W Low to DBEN Asserted (Write)		7	—	5	—	ns
45[5]	DBEN Width Asserted	Asynchronous Read Asynchronous Write	40 80	— —	22 45	— —	ns
45A[9]	DBEN Width Asserted	Synchronous Read Synchronous Write	5 40	— —	5 22	— —	ns
46	R/W Width Asserted (Asynchronous Write or Read)		100	—	50	—	ns
46A	R/W Width Asserted (Synchronous Write or Read)		60	—	30	—	ns
47A	Asynchronous Input Setup Time to Clock Low		2	—	2	—	ns
47B	Asynchronous Input Hold Time from Clock Low		8	—	6	—	ns
48[4]	DSACKx Asserted to BERR, HALT Asserted		—	25	—	14	ns
53	Data-Out Hold from Clock High		3	—	2	—	ns
55	R/W Asserted to Data Bus Impedance Change		20	—	11	—	ns
56	RESET Pulse Width (Reset Instruction)		512	—	512	—	Clks
57	BERR Negated to HALT Negated (Rerun)		0	—	0	—	ns
58[10]	BGACK Negated to Bus Driven		1	—	1	—	Clks
59[10]	BG Negated to Bus Driven		1	—	1	—	Clks
60[13]	Synchronous Input Valid to Clock High (Setup Time)		2	—	2	—	ns
61[13]	Clock High to Synchronous Input Invalid (Hold Time)		8	—	6	—	ns
62	Clock Low to STATUS, REFILL Asserted		0	20	0	15	ns
63	Clock Low to STATUS, REFILL Negated		0	20	0	15	ns

NOTES:

1. This number can be reduced to 5 ns if strobes have equal loads.
2. If the asynchronous setup time (#47A) requirements are satisfied, the DSACKx low to data setup time (#31) and DSACKx low to BERR low setup time (#48) can be ignored. The data must only satisfy the data-in clock low setup time (#27) for the following clock cycle and BERR must only satisfy the late BERR low to clock low setup time (#27A) for the following clock cycle.
3. This parameter specifies the maximum allowable skew between DSACK0 to DSACK1 asserted or DSACK1 to DSACK0 asserted; specification #47A must be met by DSACK0 or DSACK1.
4. This specification applies to the first (DSACK0 or DSACK1) DSACKx signal asserted. In the absence of DSACKx, BERR is an asynchronous input using the asynchronous input setup time (#47A).
5. DBEN may stay asserted on consecutive write cycles.
6. The minimum values must be met to guarantee proper operation. If this maximum value is exceeded, BG may be reasserted.
7. This specification indicates the minimum high time for ECS and OCS in the event of an internal cache hit followed immediately by another cache hit, a cache miss, or an operand cycle.
8. This specification guarantees operation with the MC68881/MC68882, which specifies a minimum time for DS negated to AS asserted (specification #13A in the *MC68881/MC68882 User's Manual*). Without this specification, incorrect interpretation of specifications #9A and #15 would indicate that the MC68EC030 does not meet the MC68881/MC68882 requirements.
9. This specification allows a system designer to guarantee data hold times on the output side of data buffers that have output enable signals generated with DBEN. The timing on DBEN precludes its use for synchronous READ cycles with no wait states.
10. These specifications allow system designers to guarantee that an alternate bus master has stopped driving the bus when the MC68EC030 regains control of the bus after an arbitration sequence.
11. DS will not be asserted for synchronous write cycles with no wait states.
12. These hold times are specified with respect to strobes (asynchronous) and with respect to the clock (synchronous). The designer is free to use either time.
13. Synchronous inputs must meet specifications #60 and #61 with stable logic levels for ***all*** rising edges of the clock while AS is asserted. These values are specified relative to the high level of the rising clock edge. The values originally published were specified relative to the low level of the rising clock edge.
14. This specification allows system designers to qualify the CS signal of an MC68881/MC68882 with AS (allowing 7 ns for a gate delay) and still meet the CS to DS setup time requirement (spec 8B of the *MC68881/MC68882 User's Manual)*.

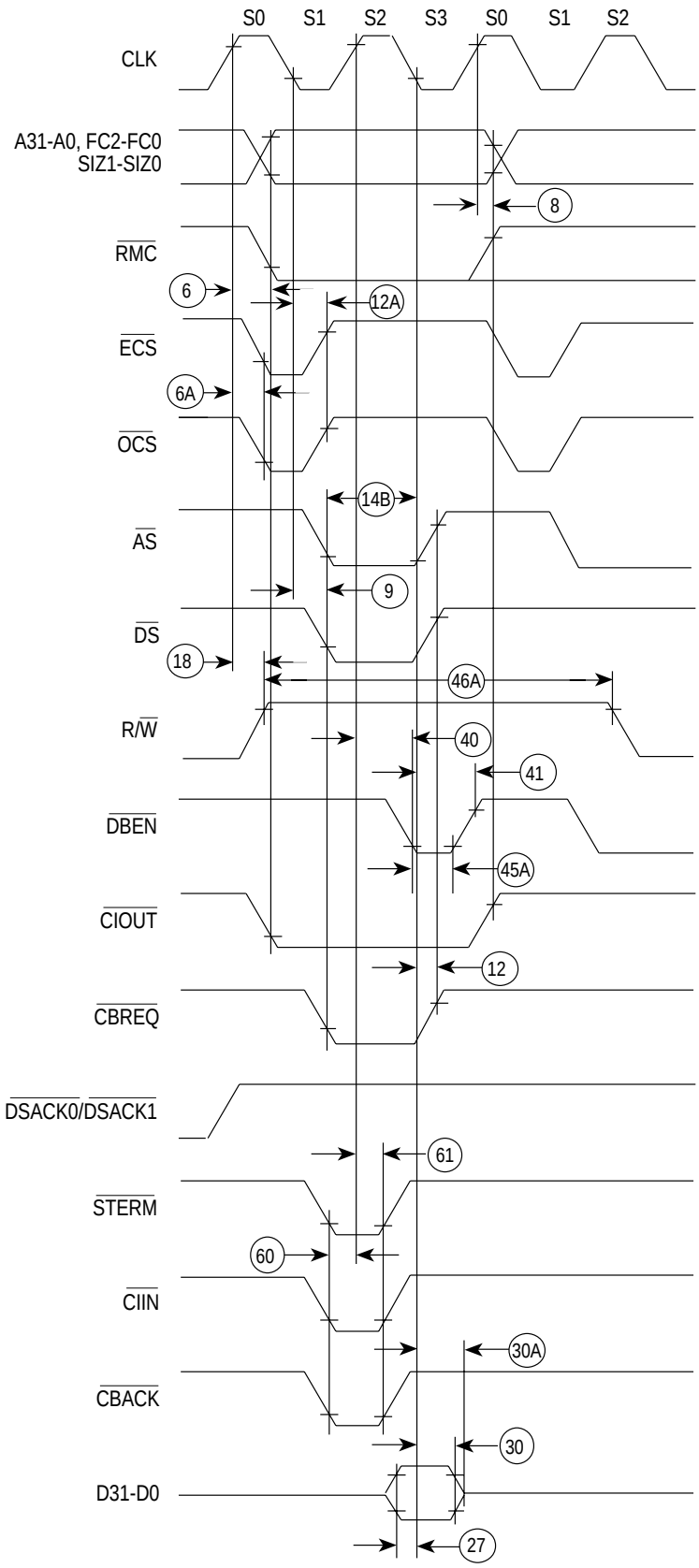
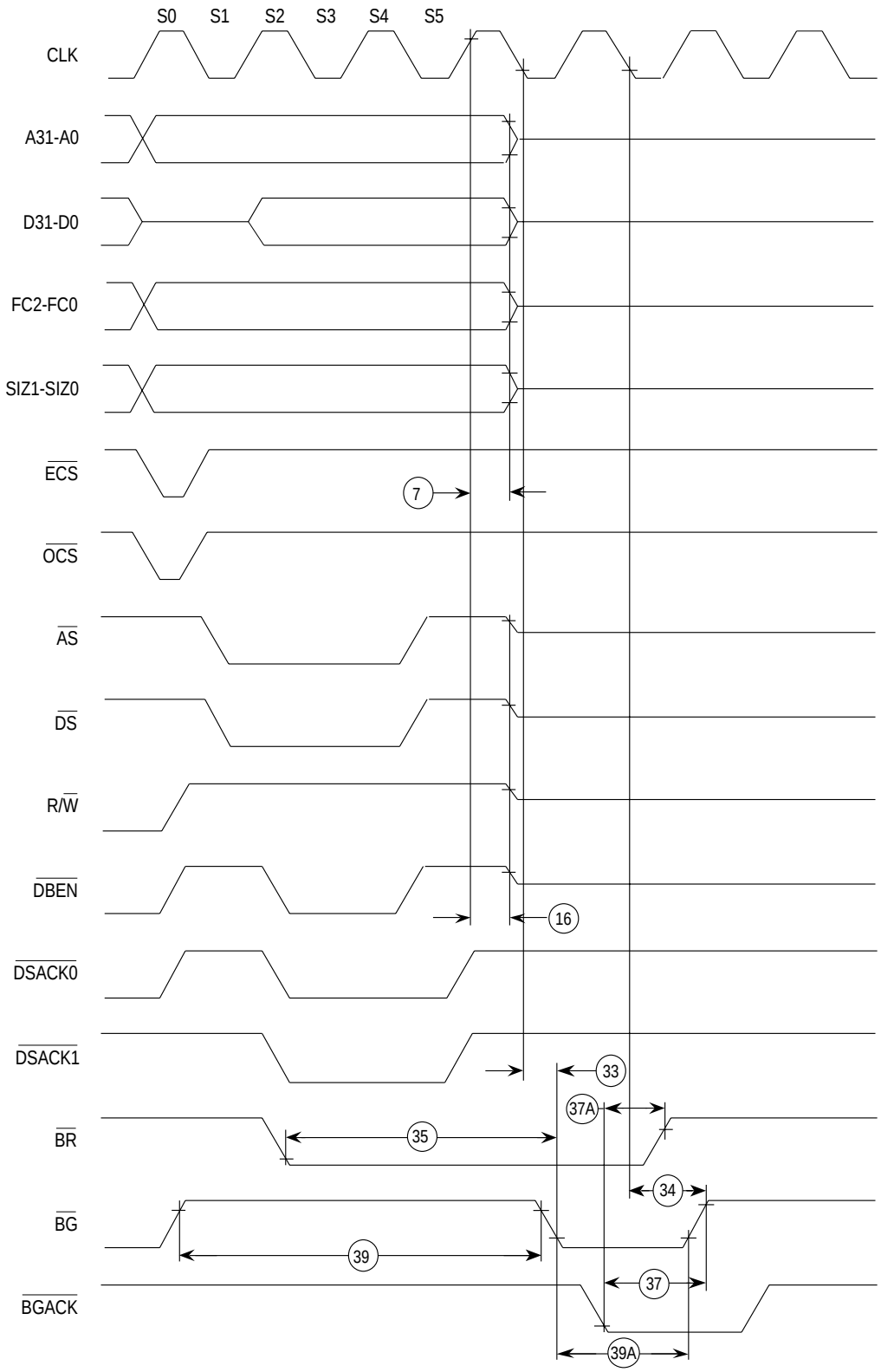


Figure 13. Synchronous Read Cycle Timing Diagram



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

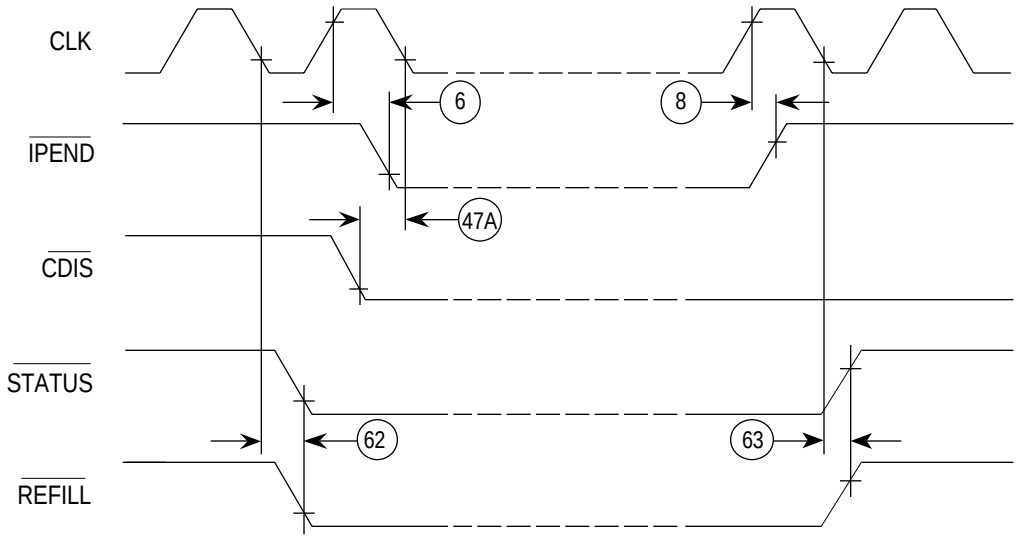
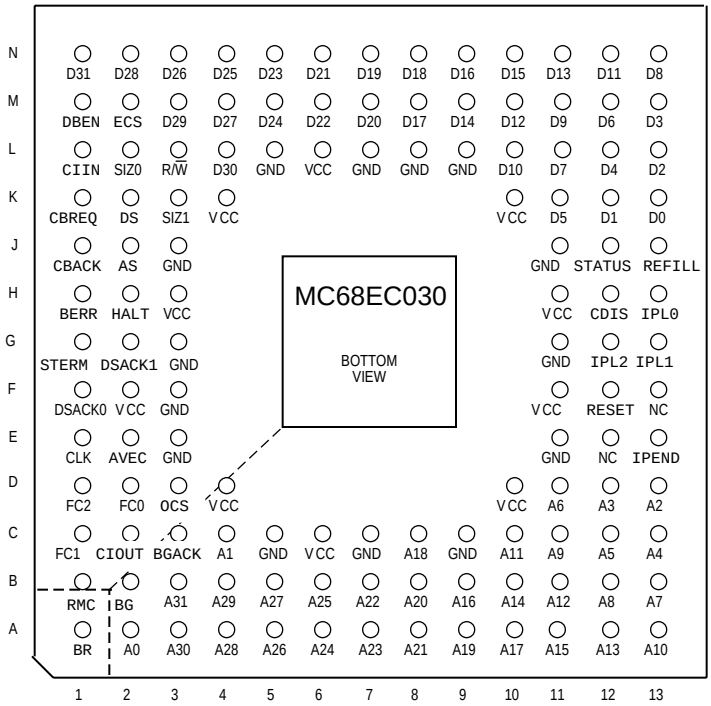


Figure 16. Other Signal Timings

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

Literature Distribution Centers:
USA: Motorola Literature Distribution: P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan.
ASIA-PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.