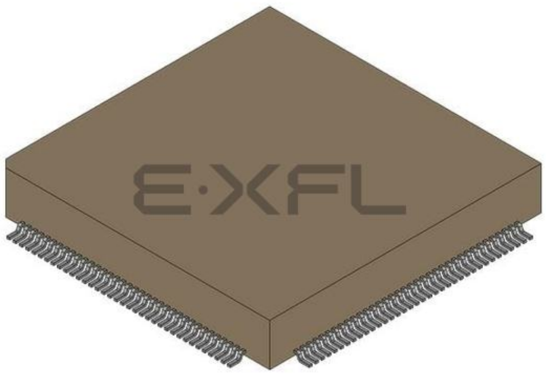
Freescale Semiconductor - MC68340CAB16E Datasheet



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	CPU32
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	16MHz
Co-Processors/DSP	-
RAM Controllers	DRAM
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	144-BQFP
Supplier Device Package	144-QFP (28x28)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68340cab16e

TABLE OF CONTENTS (Continued)

1.3.1.7 IEEE 1149.1 TEST ACCESS PORT. To aid in system diagnostics, the MC68340 includes dedicated user-accessible test logic that is fully compliant with the IEEE 1149.1 standard for boundary scan testability, often referred to as JTAG (Joint Test Action Group).

1.3.2 Direct Memory Access Module

The most distinguishing MC68340 characteristic is the high-speed 32-bit DMA controller, used to quickly move large blocks of data between internal peripherals, external peripherals, or memory without processor intervention. The DMA module consists of two, independent, programmable channels. Each channel has separate request, acknowledge, and done signals. Each channel can operate in a single-address or a dual-address (flyby) mode.

In single-address mode, only one (the source or the destination) address is provided, and a peripheral device such as a serial communications controller receives or supplies the data. An external request must start a single-address transfer. In this mode, each channel supports 32 bits of address and 8, 16, or 32 bits of data.

In dual-address mode, two bus transfers occur, one from a source device and the other to a destination device. Dual-address transfers can be started by either an internal or external request. In this mode, each channel supports 32 bits of address and 8 or 16 bits of data (32 bits require external logic). The source and destination port size can be selected independently; when they are different, the data will be packed or unpacked. An 8-bit disk interface can be read twice before the concatenated 16-bit result is passed into memory.

Byte, word, and long-word counts up to 32 bits can be transferred. All addresses and transfer counters are 32 bits. Addresses increment or remain constant, as programmed. The DMA channels support two external request modes, burst transfer and cycle steal. Internal requests can be programmed to occupy 25, 50, 75, or 100 percent of the data bus bandwidth. Interrupts can be programmed to postpone DMA completion.

The DMA module can sustain a transfer rate of 12.5 Mbytes/sec in dual-address mode and nearly 50 Mbytes/sec in single-address mode @ 25.16 MHz (8.4 and 33.3 Mbytes/sec @ 16.78 MHz, respectively). The DMA controller arbitrates with the CPU32 for the bus in parallel with existing bus cycles and is fully synchronized with the CPU32, eliminating all delays normally associated with bus arbitration by allowing DMA bus cycles to butt seamlessly with CPU bus cycles.

1.3.3 Serial Module

Most digital systems use serial I/O to communicate with host computers, operator terminals, or remote devices. The MC68340 contains a two-channel, full-duplex USART. An on-chip baud rate generator provides standard baud rates up to 76.8k baud independently to each channel's receiver and transmitter. The module is functionally equivalent to the MC68681/MC2681 DUART.

requires only a 3.3-V power supply, reduces current consumption by 40–60% in all modes of operation (as well as reducing noise emissions).

The MC68340 has many additional methods of dynamically controlling power consumption during operation. The frequency of operation can be lowered under software control to reduce current consumption when performance is less critical. Idle internal peripheral modules can be turned off to save power (5–10% each). Running a special low power stop (LPSTOP) instruction shuts down the active circuits in the CPU and peripheral modules, halting instruction execution. Power consumption in this standby mode is reduced to about 350 µW. Processing and power consumption can be resumed by resetting the part or by generating an interrupt with the SIM40's periodic interrupt timer.

1.5 PHYSICAL

The MC68340 is available as 0–16.78 MHz and 0–25.16 MHz, 0°C to +70°C and -40°C to +85°C, and 5.0 V ±5% and 3.3 V ±0.3 supply voltages (reduced frequencies at 3.3 V). Thirty-two power and ground leads minimize ground bounce and ensure proper isolation of different sections of the chip, including the clock oscillator. A 144 pins are used for signals and power. The MC68340 is available in a gull-wing ceramic quad flat pack (CQFP) with 25.6-mil (0.001-in) lead spacing or a 15 × 15 plastic pin grid array (PPGA) with 0.1-in pin spacing.

1.6 COMPACT DISC-INTERACTIVE

The MC68340 was designed to meet the needs of many markets, including compact disc-interactive (CD-I). CD-I is an emerging standard for a publishing medium that will bring multimedia to a broad general audience—the consumer. CD-I players combine television and stereo systems as output devices, with interactive control using a TV remote-control-like device to provide a multimedia experience selected from software "titles" contained in compressed form on standard compact discs.

The highly integrated MC68340 is ideal as the central processor for CD-I players. It provides the M68000 microprocessor code compatibility and DMA functions required by the *CD-I Green Book* specification as well as many other useful on-chip functions for a very cost-effective solution. The extra demands of full-motion video CD-I systems make the best use of the MC68340 high performance. The MC68340 is CD-I compliant and has been CD-I qualified. With its low voltage operation, the MC68340V is the only practical choice for portable CD-I.

State 0—The MC68340 asserts RMC in S0 to identify a read-modify-write cycle. The MC68340 places a valid address on A31–A0 and valid function codes on FC3–FC0. The function codes select the address space for the operation. SIZ1/SIZ0 become valid in S0 to indicate the operand size. The MC68340 drives R/W high for the read cycle.

State 1—One-half clock later during S1, the MC68340 asserts AS indicating a valid address on the address bus. The MC68340 also asserts DS during S1.

State 2—The selected device uses R/W, SIZ1/SIZ0, A0, and DS to place information on the data bus. Either or both of the bytes (D15–D8 and D7–D0) are selected by SIZ1/SIZ0 and A0. Concurrently, the selected device may assert DSACK≈.

State 3—As long as at least one of the DSACK≈ signals is recognized by the end of S2 (meeting the asynchronous input setup time requirement), data is latched on the next falling edge of the clock, and the cycle terminates. If DSACK≈ is not recognized by the start of S3, the MC68340 inserts wait states instead of proceeding to S4 and S5. To ensure that wait states are inserted, both DSACK1 and DSACK0 must remain negated throughout the asynchronous input setup and hold times around the end of S2. If wait states are added, the MC68340 continues to sample the DSACK≈ signals on the falling edges of the clock until one is recognized.

State 4—At the end of S4, the MC68340 latches the incoming data.

State 5—The MC68340 negates AS and DS during S5. If more than one read cycle is required to read in the operand(s), S0–S5 are repeated for each read cycle. When finished reading, the MC68340 holds the address, R/W, and FC3–FC0 valid in preparation for the write portion of the cycle. The external device keeps its data and DSACK≈ signals asserted until it detects the negation of AS or DS (whichever it detects first). The device must remove the data and negate DSACK≈ within approximately one clock period after sensing the negation of AS or DS. DSACK≈ signals that remain asserted beyond this limit may be prematurely detected for the next portion of the operation.

Idle States—The MC68340 does not assert any new control signals during the idle states, but it may internally begin the modify portion of the cycle at this time. S0–S5 are omitted if no write cycle is required. If a write cycle is required, R/W remains in the read mode until S0 to prevent bus conflicts with the preceding read portion of the cycle; the data bus is not driven until S2.

State 0—The MC68340 drives R/W low for a write cycle. Depending on the write operation to be performed, the address lines may change during S0.

State 1—In S1, the MC68340 asserts AS, indicating a valid address on the address bus.

State 2—During S2, the MC68340 places the data to be written onto D15–D0.

State 3—The MC68340 asserts DS during S3, indicating stable data on the data bus. As long as at least one of the DSACK≈ signals is recognized by the end of S2 (meeting the asynchronous input setup time requirement), the cycle terminates one clock later. If DSACK≈ is not recognized by the start of S3, the MC68340 inserts wait states instead of

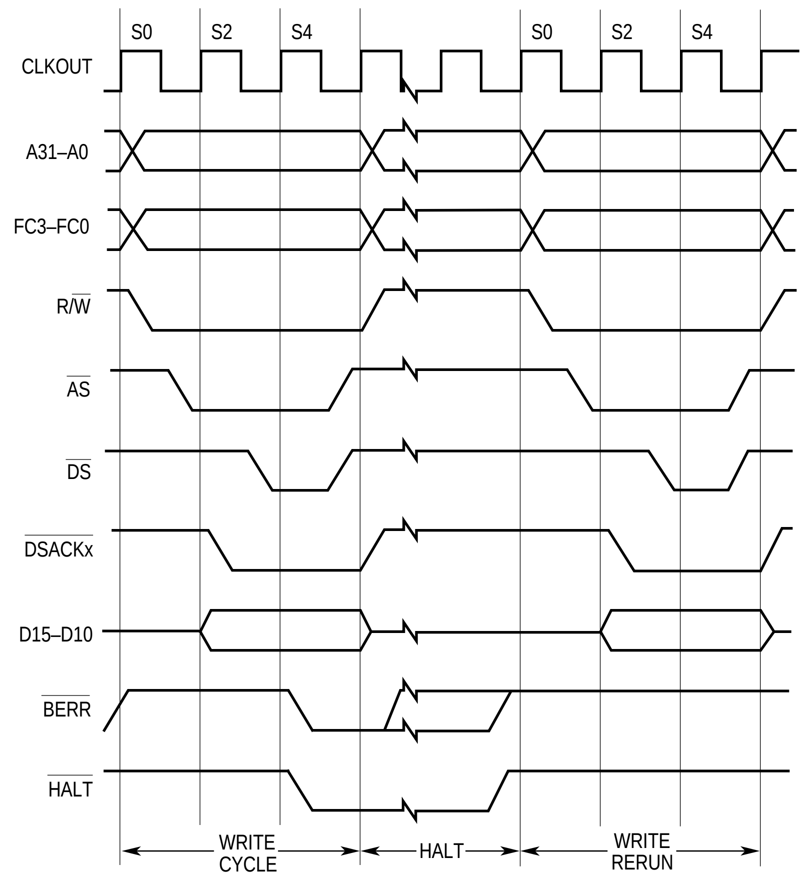


Figure 3-20. Late Retry Sequence

3.5.3 Halt Operation

When HALT is asserted and BERR is not asserted, the MC68340 halts external bus activity at the next bus cycle boundary (see Figure 3-21). HALT by itself does not terminate a bus cycle. Negating and reasserting HALT in accordance with the correct timing requirements provides a single-step (bus cycle to bus cycle) operation. Since HALT affects external bus cycles only, a program that does not require use of the external bus may continue executing. The single-cycle mode allows the user to proceed through (and debug) external MC68340 operations, one bus cycle at a time. Since the occurrence of a bus error while HALT is asserted causes a retry operation, the user must anticipate retry cycles while debugging in the single-cycle mode. The single-step operation and the software trace capability allow the system debugger to trace single bus cycles, single instructions, or changes in program flow.

When the MC68340 completes a bus cycle with HALT asserted, D15–D0 is placed in the high-impedance state, and bus control signals are negated (not high-impedance state); the A31–A0, FCx, SIZx, and R/W signals remain in the same state. The halt operation has no effect on bus arbitration (see **3.6 Bus Arbitration**). When bus arbitration occurs while the MC68340 is halted, the address and control signals are also placed in the high-impedance state. Once bus mastership is returned to the MC68340, if HALT is still

FRZ0—Freeze Bus Monitor Enable

1 = When FREEZE is asserted, the bus monitor is disabled.
0 = When FREEZE is asserted, the bus monitor continues to operate as programmed.

FIRQ—Full Interrupt Request Mode

1 = Configures port B for seven interrupt request lines, autovector, and no external chip selects.
0 = Configures port B for four interrupt request lines and four external chip selects.

See Table 4-5 for pin function selection.

SHEN1, SHEN0—Show Cycle Enable

These two control bits determine what the EBI does with the external bus during internal transfer operations (see Table 4-6). A show cycle allows internal transfers to be externally monitored. The address, data, and control signals (except for AS) are driven externally. DS is used to signal address strobe timing for show cycles. Data is valid on the next falling clock edge after DS is negated. However, data is not driven externally, and AS and DS are not asserted externally for internal accesses unless show cycles are enabled.

If external bus arbitration is disabled, the EBI will not recognize an external bus request until arbitration is enabled again. To prevent bus conflicts, external peripherals must not attempt to initiate cycles during show cycles with arbitration disabled.

Table 4-6. SHENx Control Bits

SHEN1	SHEN0	ACTION
0	0	Show cycles disabled, external arbitration enabled
0	1	Show cycles enabled, external arbitration disabled
1	X	Show cycles enabled, external arbitration enabled

SUPV—Supervisor/User Data Space

The SUPV bit defines the SIM40 registers as either supervisor data space or user (unrestricted) data space.

1 = The SIM40 registers defined as supervisor/user are restricted to supervisor data access (FC3–FC0 = $5). An attempted user-space write is ignored and returns BERR.
0 = The SIM40 registers defined as supervisor/user data are unrestricted (FC2 is a don't care).

IARB3–IARB0—Interrupt Arbitration Bits 3–0

These bits are used to arbitrate for the bus in the case that two or more modules simultaneously generate an interrupt at the same priority level. No two modules can share the same IARB value. The reset value of IARB is $F, allowing the SIM40 to arbitrate during an IACK cycle immediately after reset. The system software should initialize the IARB field to a value from $F (highest priority) to $1 (lowest priority). A

Result Data:

Always returns 32 bits of data, regardless of the size of the register being read. If the register is less than 32 bits, the result is returned zero extended.

Register Field:

The system control register is specified by the register field (see Table 5-24).

Table 5-24. Register Field for RSREG and WSREG

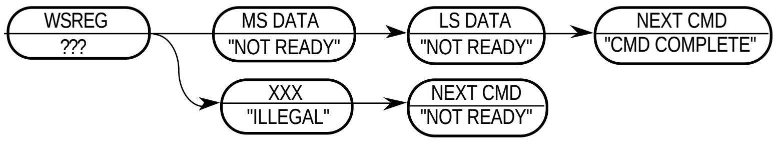
System Register	Select Code
Return Program Counter (RPC)	0000
Current Instruction Program Counter (PCC)	0001
Status Register (SR)	1011
User Stack Pointer (USP)	1100
Supervisor Stack Pointer (SSP)	1101
Source Function Code Register (SFC)	1110
Destination Function Code Register (DFC)	1111
Temporary Register A (ATEMP)	1000
Fault Address Register (FAR)	1001
Vector Base Register (VBR)	1010

5.6.2.8.7 Write System Register (WSREG). Operand data is written into the specified system control register. All registers that can be written in supervisor mode can be written in BDM. Several internal temporary registers are also accessible.

Command Format:

15	14	13	12	11	10	9	8	7	6	5	4	3 — 0
0	0	1	0	0	1	0	0	1	0	0	0	REGISTER

Command Sequence:



Operand Data:

The data to be written into the register is always supplied as a 32-bit long word. If the register is less than 32 bits, the least significant word is used.

Result Data:

"Command complete" status is returned when register write is complete.

Assertion of IPIPE for a single clock cycle indicates the use of data from IRB. Regardless of the presence of valid data in IRA, the contents of IRB are invalidated when IPIPE is asserted. If IRA contains valid data, the data is copied into IRB (IRA ⇒ IRB), and the IRB stage is revalidated.

Assertion of IPIPE for two clock cycles indicates the start of a new instruction and subsequent replacement of data in IRC. This action causes a full advance of the pipeline (IRB ⇒ IRC and IRA ⇒ IRB). IRA is refilled during the next instruction fetch bus cycle.

Data loaded into IRA propagates automatically through subsequent empty pipeline stages. Signals that show the progress of instructions through IRB and IRC are necessary to accurately monitor pipeline operation. These signals are provided by IRA and IRB validity bits. When a pipeline advance occurs, the validity bit of the stage being loaded is set, and the validity bit of the stage supplying the data is negated.

Because instruction execution is not timed to bus activity, IPIPE is synchronized with the system clock, not the bus. Figure 5-29 illustrates the timing in relation to the system clock.

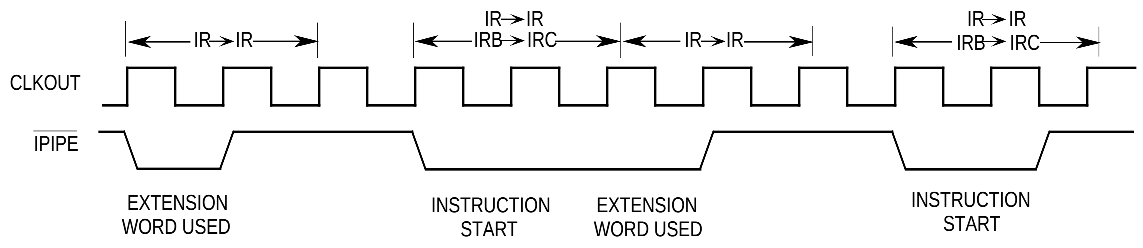


Figure 5-29. Instruction Pipeline Timing Diagram

IPIPE should be sampled on the falling edge of the clock. The assertion of IPIPE for a single cycle after one or more cycles of negation indicates use of the data in IRB (advance of IRA into IRB). Assertion for two clock cycles indicates that a new instruction has started (IRB ⇒ IRC and IRA ⇒ IRB transfers have occurred). Loading IRC always indicates that an instruction is beginning execution—the opcode is loaded into IRC by the transfer.

In some cases, instructions using immediate addressing begin executing and initiate a second pipeline advance simultaneously at the same time. IPIPE will not be negated between the two indications, which implies the need for a state machine to track the state of IPIPE. The state machine can be resynchronized during periods of inactivity on the signal.

5.6.3.3 OPCODE TRACKING DURING LOOP MODE. IPIPE and IFETCH continue to work normally during loop mode. IFETCH indicates all instruction fetches up through the point that data begins recirculating within the instruction pipeline. IPIPE continues to signal the start of instructions and the use of extension words even though data is being recirculated internally. IFETCH returns to normal operation with the first fetch after exiting loop mode.

6.2 DMA MODULE SIGNAL DEFINITIONS

This section contains a brief description of the DMA module signals used to provide handshake control for either a source or destination external device.

NOTE

The terms *assertion* and *negation* are used throughout this section to avoid confusion when dealing with a mixture of active-low and active-high signals. The term *assert* or *assertion* indicates that a signal is active or true, independent of the level represented by a high or low voltage. The term *negate* or *negation* indicates that a signal is inactive or false.

6.2.1 DMA Request (DREQ≈)

This active-low input is asserted by a peripheral device to request an operand transfer between that peripheral and memory. The assertion of DREQ≈ starts the DMA process. The assertion level in external burst mode is level sensitive; in external cycle steal mode, it is falling-edge sensitive.

6.2.2 DMA Acknowledge (DACK≈)

This active-low output is asserted by the DMA to signal to a peripheral that an operand is being transferred in response to a previous transfer request.

6.2.3 DMA Done (DONE≈)

This active-low bidirectional signal is asserted by the DMA or a peripheral device during any DMA bus cycle to indicate that the last data transfer is being performed. DONE≈ is an active input in any mode. As an output, DONE≈ is only active in external request mode. An external pullup resistor is required even if operating only in the internal request mode.

6.3 TRANSFER REQUEST GENERATION

The DMA channel supports two types of request generation methods: internal and external. Internally generated requests can be programmed to limit the amount of bus utilization. Externally generated requests can be either burst mode or cycle steal mode. The request generation method used for the channel is programmed by the channel control register (CCR) in the REQ field.

6.3.1 Internal Request Generation

Internal requests are accessed in two clocks by the intermodule bus (IMB). The channel is started as soon as the STR bit in the CCR is set. The channel immediately requests the bus and begins transferring data. Only internal requests can limit the amount of bus utilization. The percentage of the bandwidth that the DMA channel can use during a transfer can be selected by the CCR BB field.

Each operand transfer in dual-address mode requires from two to five bus cycles in response to each operand transfer request. If the source and destination operands are the same size, two cycles will transfer the complete operand. If the source and destination operands are different sizes, the number of cycles will vary. If the source is a long-word and the destination is a byte, there would be one bus cycle for the read and four bus cycles for the write. Once the DMA channel has started a dual-address operand transfer, it must complete that transfer before releasing ownership of the bus or servicing a request for another channel of equal or higher priority, unless one of the bus cycles is terminated with a bus error during the transfer.

6.6.3 Channel Termination

The channel can terminate by normal completion or from an error. The status of a DMA operation can be determined by reading the CSR. The DMA channel can also interrupt the processor to inform it of errors, normal transfer completion, or breakpoints. The fast termination option can also be used to provide a two-clock access for external requests.

6.6.3.1 CHANNEL TERMINATION. The channel operation can be terminated for several reasons: the BTC is decremented to zero, a peripheral device asserts DONE≈ during an operand transfer, the STR bit is cleared in the CCR, a bus cycle is terminated with a bus error, or a reset occurs.

6.6.3.2 INTERRUPT OPERATION. Interrupts can be generated by error termination of a bus cycle or by normal channel completion. Specifically, if the CCR interrupt error (INTE) bit is set and a bus error on source (CCR BES) bit, bus error on destination (CCR BED) bit, or configuration error (CCR CONF) bit is set, the CCR IRQ bit is set. In this case, clearing the INTE, BES, BED, or CONF bits causes the IRQ bit to be cleared. If the interrupt normal (CCR INTN) bit is set and the CCR DONE bit is set, the IRQ bit is set. In this case, clearing the INTN or the DONE bit causes the IRQ bit to be cleared. If the interrupt breakpoint (CCR INTB) and the CSR BRKP bits are set, the IRQ bit is set. Clearing INTB or BRKP clears IRQ.

6.6.3.3 FAST TERMINATION OPTION. Using the system integration module (SIM40) chip select logic, the fast termination option (Figure 6-13) can be employed to give a fast bus access of two clock cycles rather than the standard three-cycle access time for external requests. The fast termination option is described in **Section 3 Bus Operation** and **Section 4 System Integration Module**.

```
* Normal Operation, ignore FREEZE, dual-address mode. ISM field at 3. Make
* sure CPU32 SR I2-I0 bits are less than or equal to ISM bits for channel startup.
* Supervisor/user reg. unrestricted, MAID field at 3. IARB priority at 4.
          MOVE.W      #$0334,(A0)

* Clear channel control reg.
* Clear STR (start) bit to prevent the channel from starting a transfer early.
          CLR.W       DMACCR1(A0)

* Initialize interrupt reg.
* Interrupt priority at 7, interrupt vector at $42.
          MOVE.W      #$0742,DMAINT1(A0)

* Initialize channel status reg.
* Clear the DONE, BES, BED, CONF and BRKP bits to allow channel to startup.
          MOVE.B      #$7C,DMACSR1(A0)

* Initialize function code reg.
* DMA space, supervisor data space for source and destination.
          MOVE.B      #$DD,DMAFCR1(A0)

* Initialize source operand address
* Source address is equal to $6000.
          MOVE.L      SARADD,DMASAR1(A0)

* Initialize destination operand address
* Destination address is equal to $8000.
          MOVE.L      DARADD,DMADAR1(A0)

* Initialize the byte transfer count reg.
* The number of bytes to be transferred is $E or 7 words
          MOVE.L      NUMBYTE,DMABTC1(A0)

* Channel control reg. init. and Start DMA transfers
* No interrupts are enabled, destination (write) cycle. Increment source and
* destination addresses,source size is word, destination size is word.
* REQ is internal. 100% of bus bandwidth, dual-address transfers,
* start the DMA transfers.
          MOVE.W      #$0E8D,DMACCR1(A0)

*******************************************************************************
          END
*******************************************************************************
```

Example 3: Internal Request Generation, Memory Block Initialization.

```
*******************************************************************************
* MC68340 basic DMA channel register initialization example code.
```

SECTION 7 SERIAL MODULE

The MC68340 serial module is a dual universal asynchronous/synchronous receiver/transmitter that interfaces directly to the CPU32 processor via the intermodule bus (IMB). The serial module, shown in Figure 7-1, consists of the following major functional areas:

- Two Independent Serial Communication Channels (A and B)
- Baud Rate Generator Logic
- Internal Channel Control Logic
- Interrupt Control Logic

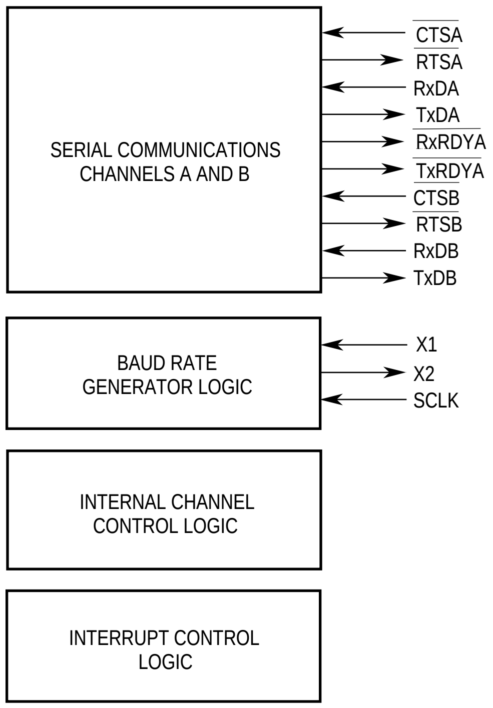

SERIAL COMMUNICATIONS CHANNELS A AND B
CTSA
RTSA
RxDA
TxDA
RxRDYA
TxRDYA
CTSB
RTSB
RxDB
TxDB
BAUD RATE GENERATOR LOGIC
X1
X2
SCLK
INTERNAL CHANNEL CONTROL LOGIC
INTERRUPT CONTROL LOGIC

Figure 7-1. Simplified Block Diagram

characters until the shift register is ready to accept more data. When the shift register is empty, it checks to see if the holding register has a valid character to be sent (TxRDY bit cleared). If there is a valid character, the shift register loads the character and reasserts the TxRDY bit in the channel's SR. Writes to the transmitter buffer when the channel's SR TxRDY bit is clear and when the transmitter is disabled have no effect on the transmitter buffer. This register can only be written when the serial module is enabled (i.e., the STP bit in the MCR is cleared).

TBA, TBB $713, $71B

7	6	5	4	3	2	1	0
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
RESET:							
0	0	0	0	0	0	0	0

Write Only Supervisor/User

TB7–TB0—These bits contain the character in the transmitter buffer.

7.4.1.10 INPUT PORT CHANGE REGISTER (IPCR). The IPCR shows the current state and the change-of-state for the CTSA and CTSB pins. This register can only be read when the serial module is enabled (i.e., the STP bit in the MCR is cleared).

IPCR $714

7	6	5	4	3	2	1	0
0	0	COSB	COSA	0	0	CTSB	CTSA
RESET:							
0	0	0	0	0	0	U	U

Read Only Supervisor/User

Bits 7, 6, 3, 2—Reserved

COSB, COSA—Change-of-State

1 = A change-of-state (high-to-low or low-to-high transition), lasting longer than 25–50 µs when using a crystal as the sampling clock or longer than one or two periods when using SCLK, has occurred at the corresponding CTS≈ input (MCR ICCS bit controls selection of the sampling clock for clear-to-send operation). When these bits are set, the ACR can be programmed to generate an interrupt to the CPU32.

0 = The CPU32 has read the IPCR. No change-of-state has occurred. A read of the IPCR also clears the ISR COS bit.

CTSB, CTSA—Current State

Starting two serial clock periods after reset, the CTS≈ bits reflect the state of the CTS≈ pins. If a CTS≈ pin is detected as asserted at that time, the associated COSx bit will be set, which will initiate an interrupt if the corresponding IECx bit of the ACR register is enabled.

1 = The current state of the respective CTS≈ input is negated.
0 = The current state of the respective CTS≈ input is asserted.

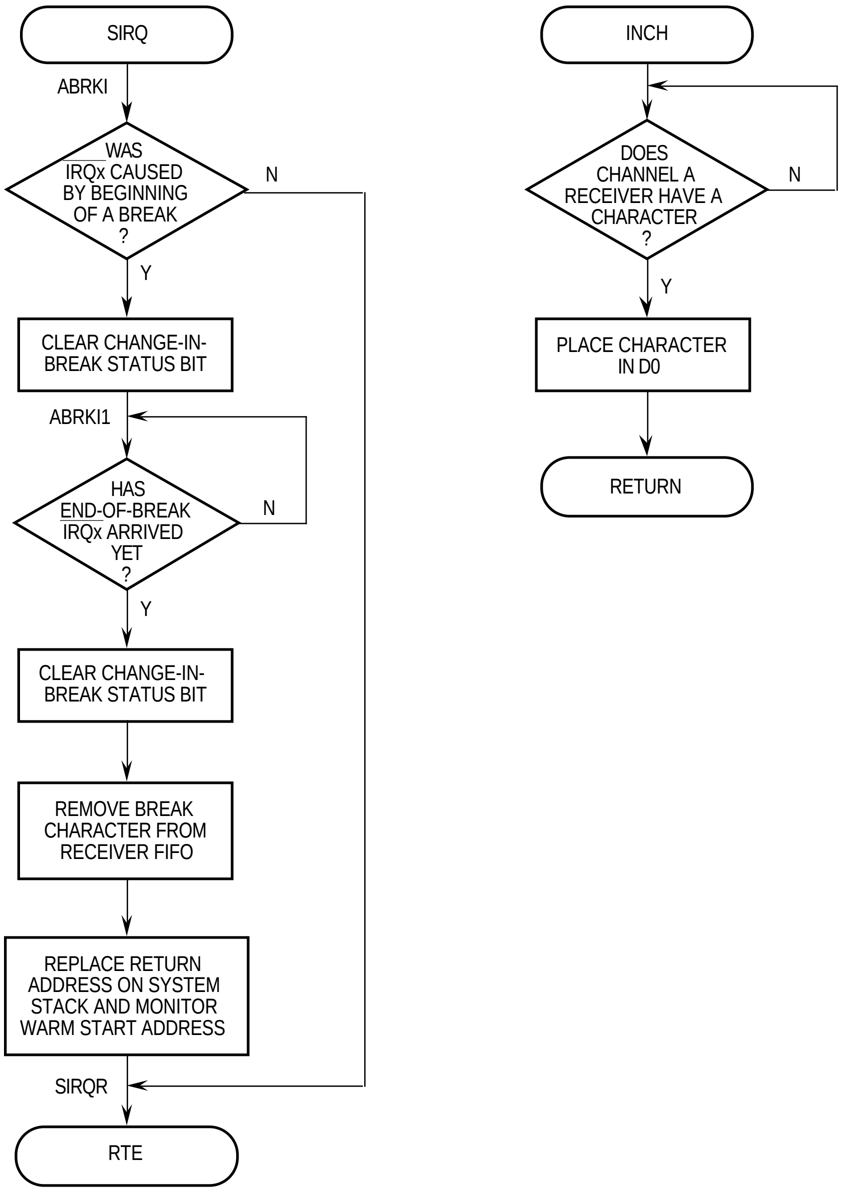

SIRQ
ABRKI
WAS IRQx CAUSED BY BEGINNING OF A BREAK ?
N
Y
CLEAR CHANGE-IN-BREAK STATUS BIT
ABRKI1
HAS END-OF-BREAK IRQx ARRIVED YET ?
N
Y
CLEAR CHANGE-IN-BREAK STATUS BIT
REMOVE BREAK CHARACTER FROM RECEIVER FIFO
REPLACE RETURN ADDRESS ON SYSTEM STACK AND MONITOR WARM START ADDRESS
SIRQR
RTE
INCH
DOES CHANNEL A RECEIVER HAVE A CHARACTER ?
N
Y
PLACE CHARACTER IN D0
RETURN

Figure 7-10. Serial Module Programming Flowchart (4 of 5)

8.4.2 Interrupt Register (IR)

The IR contains the priority level for the timer interrupt request and the 8-bit vector number of the interrupt. The register can be read or written to at any time while in supervisor mode and while the timer module is enabled (i.e., the STP bit in the MCR is cleared).

IR $604, $644

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	IL2	IL1	IL0	IVR7	IVR6	IVR5	IVR4	IVR3	IVR2	IVR1	IVR0
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Supervisor Only

Bits 15–11—Reserved

IL2–IL0—Interrupt Level Bits

Each module that can generate interrupts has an interrupt level field. The priority level encoded in these bits is sent to the CPU32 on the appropriate IRQ≈ signal. The CPU32 uses this value to determine servicing priority. See **Section 5 CPU32** for more information.

IV7–IV0—Interrupt Vector Bits

Each module that can generate interrupts has an interrupt vector (IV) field. This 8-bit number indicates the offset from the base of the vector table where the address of the exception handler for the specified interrupt is located. The IV field is reset to $0F, which indicates an uninitialized interrupt condition. See **Section 5 CPU32** for more information.

8.4.3 Control Register (CR)

The CR controls the operation of the timer. The register can always be read or written when the timer module is enabled (i.e., the STP bit in the MCR is cleared). Changing the contents of the CR should only be attempted when the timer is disabled (the SWR bit in the CR is cleared). Changing the CR while the timer is running may produce unpredictable results.

CR $606, $646

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SWR	IE2	IE1	IE0	TGE	PCLK	CPE	CLK	POT2	POT1	POT0	MODE2	MODE1	MODE0	OC1	OC0
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Supervisor/User

SWR—Software Reset

1 = Removes the software reset.

0 = A software reset is performed by first clearing this bit and then clearing the TO, TG, and TC bits in the SR. The prescaler is loaded with $FF, the counter is set to $0000, and the SR COM bit is cleared. When this bit is zero, the timer is disabled.

POT2–POT0—Prescaler Output Tap

If PCLK is set, these bits encode which of the prescaler's output taps act as the counter clock. A division of the selected clock is applied to the counter as listed in Table 8-4.

Table 8-4. POT Encoding

POT2	POT1	POT0	Division of Selected Clock
0	0	1	Divide by 2
0	1	0	Divide by 4
0	1	1	Divide by 8
1	0	0	Divide by 16
1	0	1	Divide by 32
1	1	0	Divide by 64
1	1	1	Divide by 128
0	0	0	Divide by 256

MODE2–MODE0—Operation Mode

These bits select one of the eight modes of operation for the timer as listed in Table 8-5. Refer to **8.3 Operating Modes** for more information on the individual modes.

Table 8-5. MODEx Encoding

MODE2	MODE1	MODE0	OPERATION MODE
0	0	0	Input Capture/Output Compare
0	0	1	Square-Wave Generator
0	1	0	Variable Duty-Cycle Square-Wave Generator
0	1	1	Variable-Width Single-Shot Pulse Generator
1	0	0	Pulse-Width Measurement
1	0	1	Period Measurement
1	1	0	Event Count
1	1	1	Timer Bypass (Simple Test Mode)

OC1–OC0—Output Control

These bits select the conditions under which TOUTx changes (see Table 8-6). These bits may have a different effect when in the input capture/output compare mode. Caution should be used when modifying the OC bits near timer events.

Table 8-6. OCx Encoding

OC1	OC0	TOUTx MODE
0	0	Disabled
0	1	Toggle Mode
1	0	Zero Mode
1	1	One Mode

PREL1 $60C, $64C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR1-15	PR1-14	PR1-13	PR1-12	PR1-11	PR1-10	PR1-9	PR1-8	PR1-7	PR1-6	PR1-5	PR1-4	PR1-3	PR1-2	PR1-1	PR1-0
RESET:															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Supervisor/User

For some modes of operation, this register is also used to reload the counter one falling clock edge after a timeout occurs. Refer to **8.3 Operating Modes** for more information on the individual modes.

8.4.7 Preload 2 Register (PREL2)

PREL2 is used in addition to PREL1 in the variable duty-cycle square-wave generator and variable-width single-shot pulse generator modes. When in either of these modes, the value in PREL1 is loaded into the counter on the first falling edge of the counter clock after the counter is enabled. After timeout, the value in PREL2 is loaded into the counter. This register can be be read and written when the timer module is enabled (i.e., the STP bit in the MCR is cleared). However, a write to this register must be completed before timeout for the new value to be reliably loaded into the counter.

PREL2 $60E, $64E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR2-15	PR2-14	PR2-13	PR2-12	PR2-11	PR2-10	PR2-9	PR2-8	PR2-7	PR2-6	PR2-5	PR2-4	PR2-3	PR2-2	PR2-1	PR2-0
RESET:															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Supervisor/User

8.4.8 Compare Register (COM)

The COM can be used in any mode. When the 16-bit counter reaches the value in the COM, the TC and COM bits in the SR are set. In the input capture/output compare mode, a compare event can be programmed to set, clear, or toggle TOUTx. The register can be be read and written when the timer module is enabled (i.e., the STP bit in the MCR is cleared).

COM $610, $650

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COM15	COM14	COM13	COM12	COM11	COM10	COM9	COM8	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Supervisor/User

The COM can be used to produce an interrupt when the SR TC bit has been enabled to produce an interrupt and the counter counts down to a preselected value. The COM can also be used to indicate that the timer is approaching timeout.

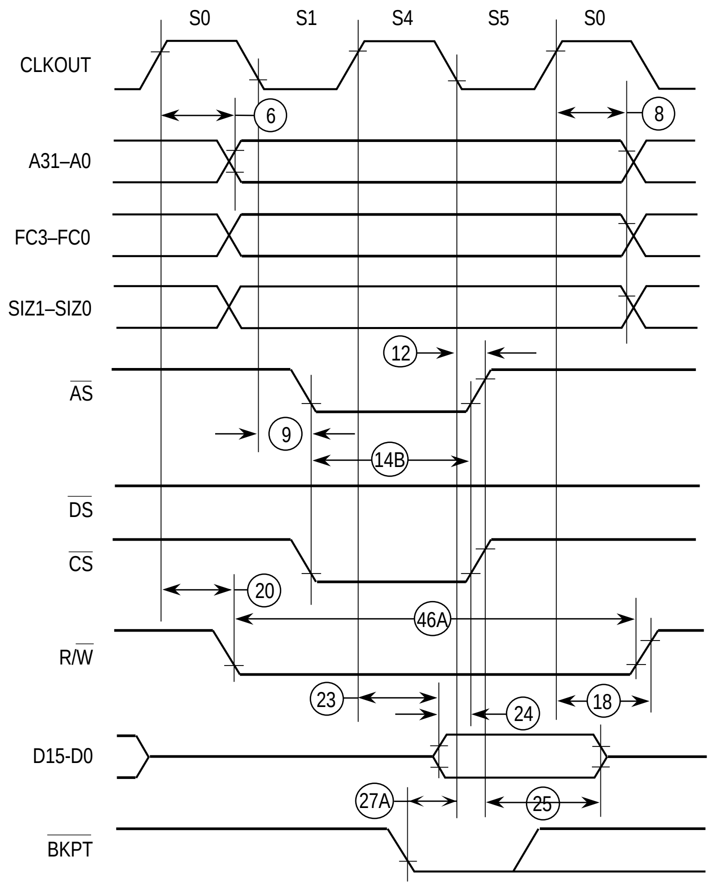
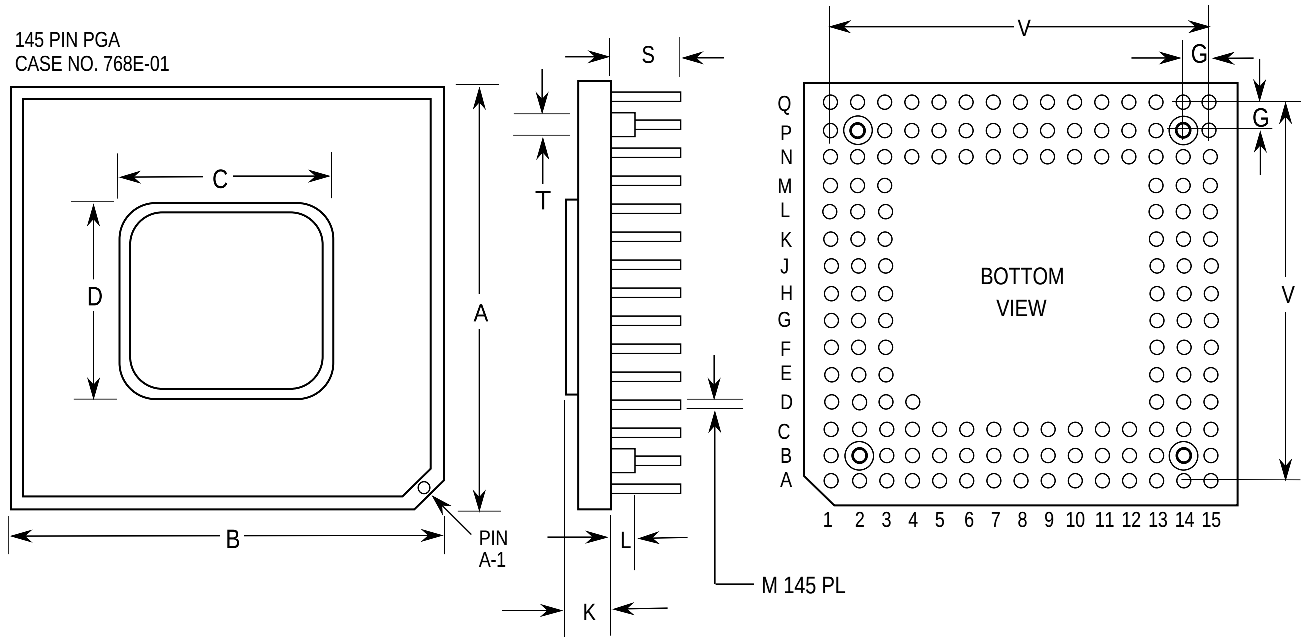


Figure 11-5. Fast Termination Write Cycle Timing Diagram

12.3.2 RP Suffix



	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	39.37	39.88	1.550	1.570
B	39.37	39.88	1.550	1.570
C	22.75	22.97	0.895	0.905
D	22.75	22.97	0.895	0.905
G	2.54 BASIC		0.100 BASIC	
K	2.92	3.43	0.115	0.135
L	1.02	1.52	0.040	0.060
M	0.43	0.55	0.017	0.022
S	4.32	4.95	0.170	0.195
V	35.56 BASIC		1.400 BASIC	