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	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	0°C ~ 70°C (TA)
Security Features	-
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68340ag16e

PREFACE

The complete documentation package for the MC68340 consists of the MC68340UM/AD, *MC68340 Integrated Processor with DMA User's Manual*, M68000PM/AD, *MC68000 Family Programmer's Reference Manual,* and the MC68340P/D, *MC68340 Integrated Processor with DMA Product Brief*.

The *MC68340 Integrated with DMA Processor User's Manual* describes the programming, capabilities, registers, and operation of the MC68340; the *MC68000 Family Programmer's Reference Manual* provides instruction details for the MC68340; and the *MC68340 Integrated Processor with DMA Product Brief* provides a brief description of the MC68340 capabilities.

This user's manual is organized as follows:

Section 1	Device Overview	Section 8	Timer Modules
Section 2	Signal Descriptions	Section 9	IEEE 1149.1 Test Access Port
Section 3	Bus Operation		
Section 4	System Integration Module	Section 10	Applications
Section 5	CPU32	Section 11	Electrical Characteristics
Section 6	DMA Controller Module	Section 12	Ordering Information and Mechanical Data
Section 7	Serial Module		

68K FAX-IT
FAX 512-891-8593

The Motorola High-End Technical Publication Department provides a FAX number for you to submit any questions and comments about this document. We welcome your suggestions for improving our documentation or any questions concerning our products.

Please provide the part number and revision number (located in upper right-hand corner on the cover), and the title of the document when submitting. When referring to items in the manual please reference by the page number, paragraph number, figure number, table number, and line number if needed. Reference the line number from the top of the page.

When we receive a FAX between the hours of 7:30 AM and 5:00 PM EST, Monday through Friday, we will respond within two hours. If the FAX is received after 5:00 PM or on the weekend, we will respond within two hours on the first working day following receipt of the FAX.

When sending a FAX, please provide your name, company, FAX number, and voice number including area code (so we can talk to a real person if needed).

Commands are received over a dedicated, high-speed, full-duplex serial interface. Commands allow the manual reading or writing of CPU32 registers, reading or writing of external memory locations, and diversion to user-specified patch code. This background debug mode permits a much simpler emulation environment while leaving the processor chip in the target system, running its own debugging operations.

1.3 ON-CHIP PERIPHERALS

To improve total system throughput and reduce part count, board size, and cost of system implementation, the M68300 family integrates on-chip, intelligent peripheral modules and typical glue logic. These functions on the MC68340 include the SIM40, a DMA controller, a serial module, and two timers.

The processor communicates with these modules over the on-chip intermodule bus (IMB). This backbone of the chip is similar to traditional external buses with address, data, clock, interrupt, arbitration, and handshake signals. Because bus masters (like the CPU32 and DMA), peripherals, and the SIM40 are all on the chip, the IMB ensures that communication between these modules is fully synchronized and that arbitration and interrupts can be handled in parallel with data transfers, greatly improving system performance. Internal accesses across the IMB may be monitored from outside of the chip, if desired.

Each module operates independently. No direct connections between peripheral modules are made inside the chip; however, external connections could, for instance, link a serial output to a DMA control line. Modules and their registers are accessed in the memory map of the CPU32 (and DMA) for easy access by general M68000 instructions and are relocatable. Each module may be assigned its own interrupt level, response vector, and arbitration priority. Since each module is a self-contained design and adheres to the IMB interface specifications, the modules may appear on other M68300 family products, retaining the investment in the software drivers for the module.

1.3.1 System Integration Module

The MC68340 SIM40 provides the external bus interface for both the CPU32 and the DMA. It also eliminates much of the glue logic that typically supports the microprocessor and its interface with the peripheral and memory system. The SIM40 provides programmable circuits to perform address decoding and chip selects, wait-state insertion, interrupt handling, clock generation, bus arbitration, watchdog timing, discrete I/O, and power-on reset timing. A boundary scan test capability is also provided.

1.3.1.1 EXTERNAL BUS INTERFACE. The external bus interface (EBI) handles the transfer of information between the internal CPU32 or DMA controller and memory, peripherals, or other processing elements in the external address space. Based on the MC68030 bus, the external bus provides up to 32 address lines and 16 data lines. Address extensions identify each bus cycle as CPU32 or DMA initiated, supervisor or user privilege level, and instruction or data access. The data bus allows dynamic sizing for 8- or 16-bit bus accesses (plus 32 bits for DMA). Synchronous transfers from the CPU32 or the DMA can be made in as little as two clock cycles. Asynchronous transfers allow the

3.3.2 Write Cycle

During a write cycle, the MC68340 transfers data to memory or a peripheral device. Figure 3-8 is a flowchart of a word write cycle.

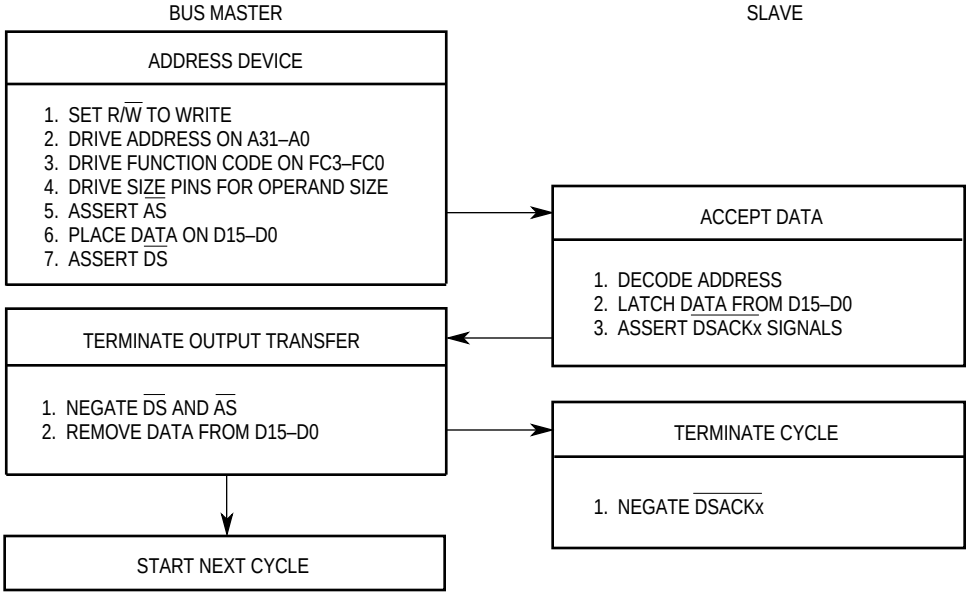


Figure 3-8. Word Write Cycle Flowchart

State 0—The write cycle starts in S0. During S0, 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 cycle. The MC68340 drives R/W low for a write cycle. SIZ1/SIZ0 become valid, indicating the number of bytes to be transferred.

State 1—One-half clock later during 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, and samples DSACK≈ at the end of S2.

State 3—The MC68340 asserts DS during S3, indicating that data is stable 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 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 DSACK≈ on the falling edges of the clock until one is recognized. The selected device uses R/W, SIZ1/SIZ0, and A0 to latch data from the appropriate byte(s) of D15–D8 and D7–D0. SIZ1/SIZ0 and A0 select the bytes of the data bus. If it has not already done so, the device asserts DSACK≈ to signal that it has successfully stored the data.

3. INTRST (internal reset) goes to all other internal circuits.

Synchronous reset sources are not asserted until the end of the current bus cycle, whether or not RMC is asserted. The internal bus monitor is automatically enabled for synchronous resets; therefore, if the current bus cycle does not terminate normally, the bus monitor terminates it. Only single-byte or word transfers are guaranteed valid for synchronous resets. An external or clock reset is a synchronous reset source.

Asynchronous reset sources indicate a catastrophic failure, and the reset controller logic immediately resets the system. Resetting the MC68340 causes any bus cycle in progress to terminate as if DSACK≈ or BERR had been asserted. In addition, the MC68340 appropriately initializes registers for a reset exception. Asynchronous reset sources include power-up, software watchdog, double bus fault resets, and execution of the RESET instruction.

If an external device drives RESET low, RESET should be asserted for at least 590 clock periods to ensure that the MC68340 resets. The reset control logic holds reset asserted internally until the external RESET is released. When the reset control logic detects that external RESET is no longer being driven, it drives both internal and external reset low for an additional 512 cycles to guarantee this length of reset to the entire system. Figure 3-27 shows the RESET timing.

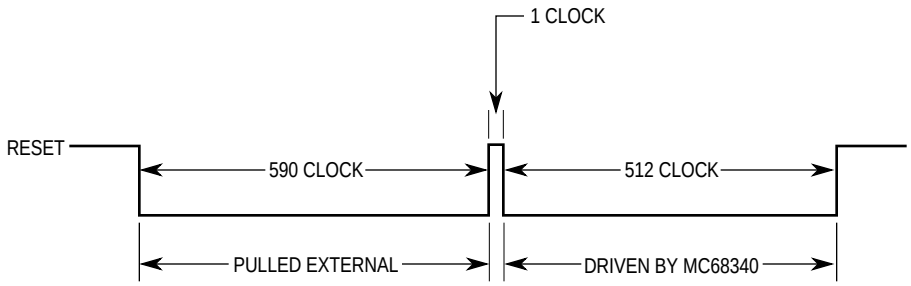


Figure 3-27. Timing for External Devices Driving RESET

If reset is asserted from any other source, the reset control logic asserts RESET for 328 input clock periods plus 512 output clock periods, and until the source of reset is negated.

After any internal reset occurs, a 14-cycle rise time is allowed before testing for the presence of an external reset. If no external reset is detected, the CPU32 begins its vector fetch.

Figure 3-28 is a timing diagram of the power-up reset operation, showing the relationships between RESET, V_{CC}, and bus signals. During the reset period, the entire bus three-states except for non-three-statable signals, which are driven to their inactive state. Once RESET negates, all control signals are driven to their inactive state, the data bus is in read mode, and the address bus is driven. After this, the first bus cycle for RESET exception processing begins.

4.3.4.1 BASE ADDRESS REGISTERS. There are four 32-bit base address registers in the chip select function, one for each chip select signal.

Base Address 1 $044, $04C, $054, $05C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

Supervisor Only

Base Address 2 $046, $04E, $056, $05E

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BFC3	BFC2	BFC1	BFC0	WP	FTE	NCS	V
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	0	0

U = Unaffected by reset Supervisor Only

BA31–BA8—Base Address Bits 31–8

The base address field, the upper 24 bits of each base address register, selects the starting address for the chip select. The specified base address must be on a multiple of the selected block size. The corresponding bits, AM31–AM8, in the address mask register define the size of the block for the chip select. The base address field (and the base function code field) is compared to the address on the address bus to determine if a chip select should be generated.

BFC3–BFC0—Base Function Code Bits 3–0

The value programmed into this field causes a chip select to be asserted for a certain address space type. There are nine function code address spaces (see **Section 3 Bus Operation**) specified as either user or supervisor, program or data, CPU, and DMA. These bits should be used to allow access to one type of address space. If access to more than one type of address space is desired, the FCMx bits should be used in addition to the BFCx bits. To prevent access to CPU space, set the NCS bit.

WP—Write Protect

This bit can restrict write accesses to the address range in a base address register. An attempt to write to the range of addresses specified in a base address register that has this bit set returns BERR.

1 = Only read accesses are allowed.
0 = Either read or write accesses are allowed.

FTE—Fast-Termination Enable

This bit causes the cycle to terminate early with an internal DSACK≈, giving a fast two-clock external access. When clear, all external cycles are at least three clocks. If fast termination is enabled, the DD bits of the corresponding address mask register are overridden (see **Section 3 Bus Operation**).

1 = Fast termination cycle enabled (termination determined by PS bits).
0 = Fast termination cycle disabled (termination determined by DD and PS bits).

To specify conditions for change in program control, condition codes must be substituted for the letters "cc" in conditional program control opcodes. Condition test mnemonics are given below. Refer to **5.3.3.10 Condition Tests** for detailed information on condition codes.

CC — Carry clear
CS — Carry set
EQ — Equal
F — False*
GE — Greater or equal
GT — Greater than
HI — High
LE — Less or equal
LS — Low or same
LT — Less than
MI — Minus
NE — Not equal
PL — Plus
T — True
VC — Overflow clear
VS — Overflow set

*Not applicable to the Bcc instruction

5.3.3.9 SYSTEM CONTROL INSTRUCTIONS. Privileged instructions, trapping instructions, and instructions that use or modify the CCR provide system control operations. All of these instructions cause the processor to flush the instruction pipeline. Table 5-11 summarizes the instructions. The preceding list of condition tests also applies to the TRAPcc instruction. Refer to **5.3.3.10 Condition Tests** for detailed information on condition codes.

First, the results of each TBL are rounded with the TBLS round-to-nearest-even algorithm. The following values would be returned by TBLS:

```
TBL # 1        0010  0000 .
TBL # 2        0011  1111 .
TBL # 3        0000  0001 .
```

Summing, the following result is obtained:

```
0010  0000 .
0011  1111 .
0000  0001 .
0110  0000 .
```

Now, using the same TBL results, the sum is first calculated and then rounded according to the same algorithm:

```
0010  0000 . 0111  0000
0011  1111 . 0111  0000
0000  0001 . 0111  0000
0110  0001 . 0101  0000
```

Rounding yields:

```
0110  0001 .
```

The second result is preferred. The following code sequence illustrates how addition of a series of table interpolations can be performed without loss of precision in the intermediate results:

```
L0:
    TBLSN.B     ⟨ea⟩, Dx
    TBLSN.B     ⟨ea⟩, Dx
    TBLSN.B     ⟨ea⟩, Dl
    ADD.L       Dx, Dm      Long addition avoids problems with carry
    ADD.L       Dm, Dl
    ASR.L       #8, Dl      Move radix point
    BCC.B       L1          Fraction MSB in carry
    ADDQ.B      #1, Dl
L1: . . .
```

All exception processing is performed at the supervisor level. All bus cycles generated during exception processing are supervisor references, and all stack accesses use the SSP.

Instructions that have important system effects can only be executed at supervisor level. For instance, user programs are not permitted to execute STOP, LPSTOP, or RESET instructions. To prevent a user program from gaining privileged access, except in a controlled manner, instructions that can alter the S-bit in the SR are privileged. The TRAP #n instruction provides controlled user access to operating system services.

5.4.2.2 USER PRIVILEGE LEVEL. If the S-bit in the SR is cleared, the processor executes instructions at the user privilege level. The bus cycles for an instruction executed at the user privilege level are classified as user references, and the values of the function codes on FC2–FC0 specify user address spaces. While the processor is at the user level, implicit references to the system SP and explicit references to address register seven (A7) refer to the USP.

5.4.2.3 CHANGING PRIVILEGE LEVEL. To change from user privilege level to supervisor privilege level, a condition that causes exception processing must occur. When exception processing begins, the current values in the SR, including the S-bit, are saved on the supervisor stack, and then the S-bit is set to enable supervisory access. Execution continues at supervisor privilege level until exception processing is complete.

To return to user access level, a system routine must execute one of the following instructions: MOVE to SR, ANDI to SR, EORI to SR, ORI to SR, or RTE. These instructions execute only at supervisor privilege level and can modify the S-bit of the SR. After these instructions execute, the instruction pipeline is flushed, then refilled from the appropriate address space.

The RTE instruction causes a return to a program that was executing when an exception occurred. When RTE is executed, the exception stack frame saved on the supervisor stack can be restored in either of two ways.

If the frame was generated by an interrupt, breakpoint, trap, or instruction exception, the SR and PC are restored to the values saved on the supervisor stack, and execution resumes at the restored PC address, with access level determined by the S-bit of the restored SR.

If the frame was generated by a bus error or an address error exception, the entire processor state is restored from the stack.

5.5 EXCEPTION PROCESSING

An exception is a special condition that preempts normal processing. Exception processing is the transition from normal mode program execution to execution of a routine that deals with an exception. The following paragraphs discuss system resources related to exception handling, exception processing sequence, and specific features of individual exception processing routines.

starting address of the block and to retrieve the first result. Subsequent operands are retrieved with the DUMP command. The initial address is incremented by the operand size (1, 2, or 4) and saved in a temporary register. Subsequent DUMP commands use this address, increment it by the current operand size, and store the updated address back in the temporary register.

NOTE

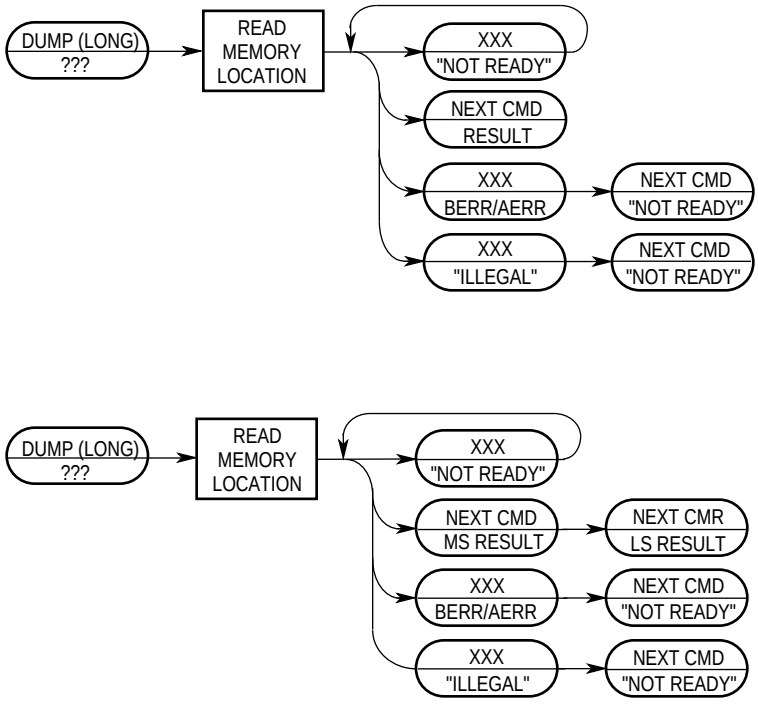
The DUMP command does not check for a valid address in the temporary register—DUMP is a valid command only when preceded by another DUMP or by a READ command. Otherwise, the results are undefined. The NOP command can be used for intercommand padding without corrupting the address pointer.

The size field is examined each time a DUMP command is given, allowing the operand size to be altered dynamically.

Command Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	1	0	1	OP SIZE		0	0	0	0	0	0

Command Sequence:



6.1 DMA MODULE OVERVIEW

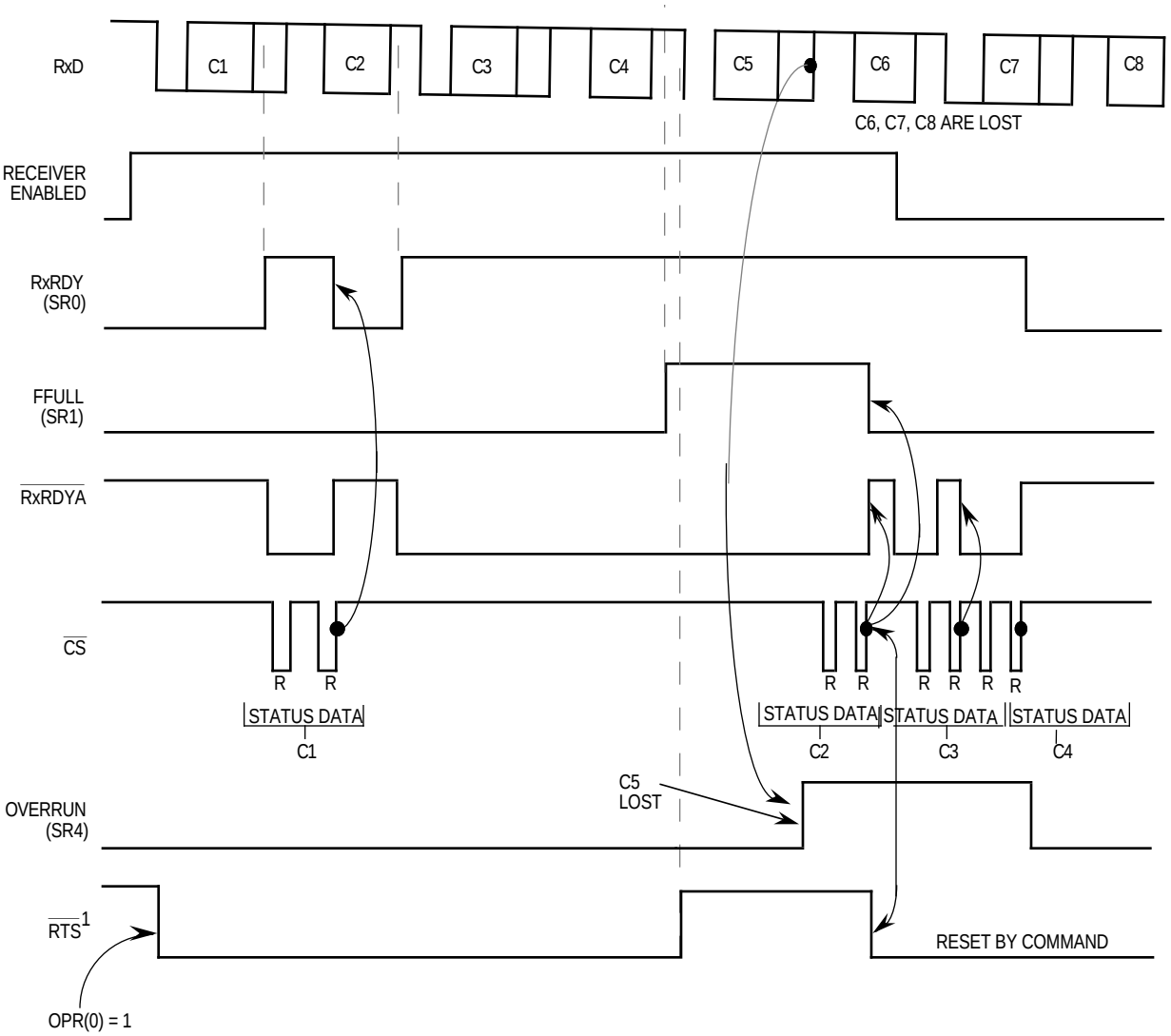
The main purpose of the DMA controller module is to transfer data at very high rates, usually much faster than the CPU32 under software control can handle. The term DMA is used to refer to the ability of a peripheral device to access memory in a system in the same manner as a microprocessor does. DMA operations can greatly increase overall system performance.

The MC68340 DMA module consists of two, independent, programmable channels. The term DMA is used throughout this section to reference either channel 1 or channel 2 since the two are functionally equivalent. Each channel has independent request, acknowledge, and done signals. However, both channels cannot own the bus at the same time. Therefore, it is impossible to implicitly address both DMA channels at the same time. The MC68340 on-chip peripherals do not support the single-address transfer mode.

DMA requests may be internally generated by the channel or externally generated by a device. For an internal request, the amount of bus bandwidth allocated for the DMA can be programmed. The DMA channels support two external request modes: burst mode and cycle steal mode.

The DMA controller supports single- and dual-address transfers. In single-address mode, a channel supports 32 bits of address and 32 bits of data. Only an external request can be used to start a transfer in the single-address mode. The DMA provides address and control signals during a single-address transfer. The requesting device either sends or receives data to or from the specified address (see Figure 6-2). In dual-address mode, a channel supports 32 bits of address and 16 bits of data. The dual-address transfers can be started by either the internal request mode or by an external device using the request signal. In this mode, two bus transfers occur, one from a source device and the other to a destination device (see Figure 6-3). In dual-address mode, operands are packed or unpacked according to port sizes and addresses.

Any operation involving the DMA will follow the same basic steps: channel initialization, data transfer, and channel termination. In the channel initialization step, the DMA channel registers are loaded with control information, address pointers, and a byte transfer count. The channel is then started. During the data transfer step, the DMA accepts requests for operand transfers and provides addressing and bus control for the transfers. The channel termination step occurs after operation is complete. The channel indicates the status of the operation in the channel status register.



NOTES:
1. Timing shown for MR1(7) = 1
2. Timing shown for OPCR(4) = 1 and MR1(6) = 0
3. R = Read
4. C_N = Received Character

Figure 7-8. Multidrop Mode Timing Diagram

A transmitted character from the master station consists of a start bit, a programmed number of data bits, an address/data (A/D) bit flag, and a programmed number of stop bits. The A/D bit identifies the type of character being transmitted to the slave station. The character is interpreted as an address character if the A/D bit is set or as a data character if the A/D bit is cleared. The polarity of the A/D bit is selected by programming bit 2 of the MR1. The MR1 should be programmed before enabling the transmitter and loading the corresponding data bits into the transmit buffer.

In multidrop mode, the receiver continuously monitors the received data stream, regardless of whether it is enabled or disabled. If the receiver is disabled, it sets the

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.

RxRDYB—Channel B Receiver Ready or FIFO full

1 = Enable interrupt
0 = Disable interrupt

TxRDYB—Channel B Transmitter Ready

1 = Enable interrupt
0 = Disable interrupt

Bit 3—Reserved

DBA—Delta Break A

1 = Enable interrupt
0 = Disable interrupt

RxRDYA—Channel A Receiver Ready or FIFO full

1 = Enable interrupt
0 = Disable interrupt

TxRDYA—Channel A Transmitter Ready

1 = Enable interrupt
0 = Disable interrupt

7.4.1.14 INPUT PORT (IP). The IP register shows the current state of the CTS≈ inputs. This register can only be read when the serial module is enabled (i.e., the STP bit in the MCR is cleared).

IP $71D

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

Read Only Supervisor/User

CTSB, CTSA—Current State

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

The information contained in these bits is latched and reflects the state of the input pins at the time that the IP is read.

NOTE

These bits have the same function and value of the IPCR bits 1 and 0.

— Pulse-Width Measurement

— Period Measurement

— Event Counting

• Seven Maskable Interrupt Conditions Based on Programmable Events

8.1.1 Timer and Counter Functions

The term 'timer' is used to reference either timer 1 or timer 2, since the two are functionally equivalent.

The timer can perform virtually any application traditionally assigned to timers and counters. The timer can be used to generate timed events that are independent of the timing errors to which real-time programmed microprocessors are susceptible—for example, those of dynamic memory refreshing, DMA cycle steals, and interrupt servicing.

The timer has several functional areas: an 8-bit countdown prescaler, a 16-bit downcounter, timeout logic, compare logic, and clock selection logic. Figure 8-2 shows a functional diagram of the timer module.

8.1.1.1 PRESCALER AND COUNTER. The counter can be driven directly by the selected clock or the prescaler output. Both the counter and prescaler are updated on the falling edge of the clock. During reset, the prescaler is set to $FF, and the counter is set to $0000. The counter is loaded with a programmed value on the first falling edge of the counter clock after the timer is enabled and again when a timeout occurs (counter reaches $0000). The prescaler and counter can be used as one 24-bit counter by enabling the prescaler and selecting the divide-by-256 prescaler output. Refer to **8.4 Register Description** for additional information on how to program the timer.

8.1.1.2 TIMEOUT DETECTION. Timeout is achieved when all 16 stages of the counter transition to zero, a counter value of $0000. Timeout is a defined counter event which triggers specific actions depending upon the programmed mode of operation. Refer to **8.3 Operating Modes** for descriptions of the individual modes.

8.1.1.3 COMPARATOR. The comparator block compares the value in the 16-bit compare register (COM) with the output of the 16-bit counter. When an exact match is detected, bits in the status register (SR) are set to indicate this condition. When in the input capture/output compare mode, a match is a defined counter event that can affect the output of the timer (TOUTx). Refer to **8.3.1 Input Capture/Output Compare** for additional information on this 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.

Table 9-2. Boundary Scan Bit Definitions

Bit Num	Cell Type	Pin/Cell Name	Pin Type	Output CTL Cell	Bit Num	Cell Type	Pin/Cell Name	Pin Type	Output CTL Cell
0	IO.Cell	FC3	I/O*	ab.ctl	35	O.Latch	R≈RDYA	Output	—
1	IO.Cell	FC2	I/O*	ab.ctl	36	O.Latch	T≈RDYA	Output	—
2	IO.Cell	FC1	I/O*	ab.ctl	37	I.Pin	RxDB	Input	—
3	IO.Cell	FC0	I/O*	ab.ctl	38	O.Latch	TxDB	Output	—
4	IO.Cell	A23	I/O*	ab.ctl	39	O.Latch	RTSB	Output	—
5	IO.Cell	A22	I/O*	ab.ctl	40	I.Pin	CTSB	Input	—
6	IO.Cell	A21	I/O*	ab.ctl	41	I.Pin	SCLK	Input	—
7	IO.Cell	A20	I/O*	ab.ctl	42	I.Pin	X1	Input	—
8	IO.Cell	A19	I/O*	ab.ctl	43	I.Pin	DREQ1	Input	—
9	IO.Cell	A18	I/O*	ab.ctl	44	O.Latch	DACK1	Output	—
10	IO.Cell	A17	I/O*	ab.ctl	45	O.Latch	DONE1	OD-I/O	—
11	IO.Cell	A16	I/O*	ab.ctl	46	I.Pin	DONE1	OD-I/O	—
12	IO.Cell	A15	I/O*	ab.ctl	47	I.Pin	DREQ2	Input	—
13	IO.Cell	A14	I/O*	ab.ctl	48	O.Latch	DACK2	Output	—
14	IO.Cell	A13	I/O*	ab.ctl	49	O.Latch	DONE2	OD-I/O	—
15	IO.Cell	A12	I/O*	ab.ctl	50	I.Pin	DONE2	OD-I/O	—
16	IO.Cell	A11	I/O*	ab.ctl	51	IO.Cell	IRQ7	I/O	irq7.ctl
17	IO.Cell	A10	I/O*	ab.ctl	52	IO.Ctl0	irq7.ctl	—	—
18	IO.Cell	A9	I/O*	ab.ctl	53	IO.Cell	IRQ6	I/O	irq6.ctl
19	IO.Cell	A8	I/O*	ab.ctl	54	IO.Ctl0	irq6.ctl	—	—
20	IO.Cell	A7	I/O*	ab.ctl	55	IO.Cell	IRQ5	I/O	irq5.ctl
21	IO.Cell	A6	I/O*	ab.ctl	56	IO.Ctl0	irq5.ctl	—	—
22	IO.Cell	A5	I/O*	ab.ctl	57	IO.Cell	CS3	I/O	cs3.ctl
23	IO.Cell	A4	I/O*	ab.ctl	58	IO.Ctl0	cs3.ctl	—	—
24	IO.Cell	A3	I/O*	ab.ctl	59	IO.Cell	IRQ3	I/O	irq3.ctl
25	IO.Cell	A2	I/O*	ab.ctl	60	IO.Ctl0	irq3.ctl	—	—
26	IO.Cell	A1	I/O*	ab.ctl	61	IO.Cell	CS2	I/O	cs2.ctl
27	I.Pin	TGATE2	Input	—	62	IO.Ctl0	cs2.ctl	—	—
28	O.Latch	TOUT2	TS-Output	tout2.ctl	63	IO.Cell	CS1	I/O	cs1.ctl
29	IO.Ctl0	tout2.ctl	—	—	64	IO.Ctl0	cs1.ctl	—	—
30	I.Pin	TIN2	Input	—	65	IO.Cell	CS0	I/O	cs0.ctl
31	I.Pin	RxDA	Input	—	66	IO.Ctl0	cs0.ctl	—	—
32	O.Latch	TxDA	Output	—	67	IO.Cell	D0	I/O	db.ctl
33	O.Latch	RTSA	Output	—	68	IO.Cell	D1	I/O	db.ctl
34	I.Pin	CTSA	Input	—	69	IO.Cell	D2	I/O	db.ctl

NOTE

Since there is no internal synchronization between the IEEE 1149.1 clock (TCK) and the system clock (CLKOUT), the user must provide some form of external synchronization to achieve meaningful results.

The second function of SAMPLE/PRELOAD is to initialize the boundary scan register output bits prior to selection of EXTEST. This initialization ensures that known data will appear on the outputs when entering the EXTEST instruction.

9.4.3 BYPASS (X1X, 101)

The BYPASS instruction selects the single-bit bypass register as shown in Figure 9-9. This creates a shift-register path from TDI to the bypass register and, finally, to TDO, circumventing the 132-bit boundary scan register. This instruction is used to enhance test efficiency when a component other than the MC68340 becomes the device under test.

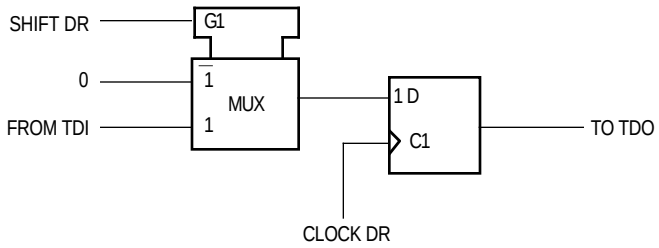


Figure 9-9. Bypass Register

When the bypass register is selected by the current instruction, the shift-register stage is set to a logic zero on the rising edge of TCK in the capture-DR controller state. Therefore, the first bit to be shifted out after selecting the bypass register will always be a logic zero.

9.4.4 HI-Z (100)

The HI-Z instruction is not included in the IEEE 1149.1 standard. It is provided as a manufacturer's optional public instruction to prevent having to backdrive the output pins during circuit-board testing. When HI-Z is invoked, all output drivers, including the two-state drivers, are turned off (i.e., high impedance). The instruction selects the bypass register.

9.5 MC68340 RESTRICTIONS

The control afforded by the output enable signals using the boundary scan register and the EXTEST instruction requires a compatible circuit-board test environment to avoid device-destructive configurations. The user must avoid situations in which the MC68340 output drivers are enabled into actively driven networks. Overdriving the TDO driver when it is active is not recommended.

10.2.4 Interfacing an 8-Bit Device to 16-Bit Memory Using Single-Address DMA Mode

One of the requirements of single-address mode is that the source and destination must be the same port size. However, the MC68340 can perform direct memory accesses in single-address mode between an 8-bit device and 16-bit memory. The port size must be specified as 8 bits, and some external logic is required as shown in Figure 10-14.

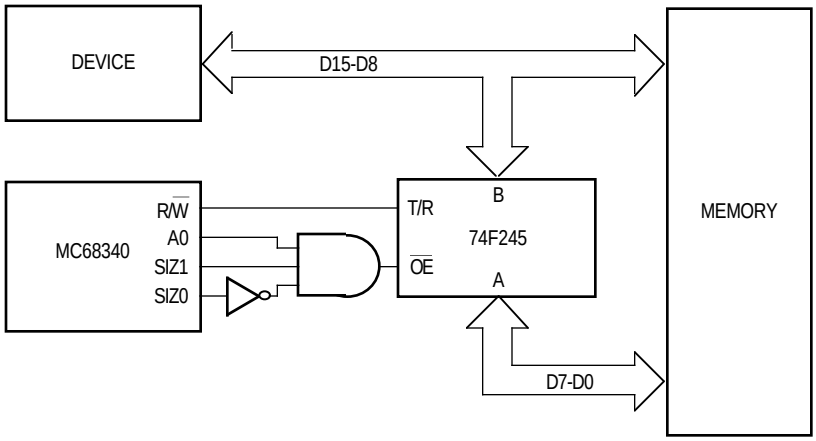


Figure 10-14. Circuitry for Interfacing 8-Bit Device to 16-Bit Memory in Single-Address DMA Mode

During even-byte accesses, the data is transferred directly on D15–D8. However, during odd-byte accesses, the data must be routed on D15–D8 for the 8-bit device and on D7–D0 for the 16-bit memory.

10.3 POWER CONSUMPTION CONSIDERATIONS

The MC68340 can be designed into low-power applications that involve high-performance processing capability (32-bits), high functional density, small size, portable capability, and battery operation.

The MC68340 fits into the following types of applications:

- "Palmtop" Computers
  - — Stylus Input
  - — Voice Input
  - — Image Input
- Transaction Tracking
  - — Car Rental
  - — Cargo
  - — Courier
  - — Handheld
- Bar Code Scanners
- Telephony
  - — Cordless Phones
  - — Cellular Phones
- CD-I, CD-ROM
- Defense Industry
  - — Guidance Systems
  - — Tracking Systems
- Data Entry
- Instruments
- Handheld Games

— E —

— F —