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	25MHz
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/pro/item?MUrl=&PartUrl=mc68340ag25e

TABLE OF CONTENTS (Continued)

Paragraph Number	Title	Page Number
7.1.5	Comparison of Serial Module to MC68681	7-4
7.2	Serial Module Signal Definitions	7-4
7.2.1	Crystal Input or External Clock (X1)	7-5
7.2.2	Crystal Output (X2)	7-5
7.2.3	External Input (SCLK)	7-6
7.2.4	Channel A Transmitter Serial Data Output (TxDA)	7-6
7.2.5	Channel A Receiver Serial Data Input (RxDA)	7-6
7.2.6	Channel B Transmitter Serial Data Output (TxDB)	7-6
7.2.7	Channel B Receiver Serial Data Input (RxDB)	7-6
7.2.8	Channel A Request-To-Send (RTSA)	7-6
7.2.8.1	RTSA	7-6
7.2.8.2	OP0	7-6
7.2.9	Channel B Request-To-Send (RTSB)	7-6
7.2.9.1	RTSB	7-7
7.2.9.2	OP1	7-7
7.2.10	Channel A Clear-To-Send (CTSA)	7-7
7.2.11	Channel B Clear-To-Send (CTSB)	7-7
7.2.12	Channel A Transmitter Ready (T≈RDYA)	7-7
7.2.12.1	T≈RDYA	7-7
7.2.12.2	OP6	7-7
7.2.13	Channel A Receiver Ready (R≈RDYA)	7-7
7.2.13.1	R≈RDYA	7-7
7.2.13.2	FFULLA	7-7
7.2.13.3	OP4	7-7
7.3	Operation	7-8
7.3.1	Baud Rate Generator	7-8
7.3.2	Transmitter and Receiver Operating Modes	7-8
7.3.2.1	Transmitter	7-10
7.3.2.2	Receiver	7-11
7.3.2.3	FIFO Stack	7-12
7.3.3	Looping Modes	7-14
7.3.3.1	Automatic Echo Mode	7-14
7.3.3.2	Local Loopback Mode	7-14
7.3.3.3	Remote Loopback Mode	7-14
7.3.4	Multidrop Mode	7-15
7.3.5	Bus Operation	7-17
7.3.5.1	Read Cycles	7-17
7.3.5.2	Write Cycles	7-17
7.3.5.3	Interrupt Acknowledge Cycles	7-17
7.4	Register Description and Programming	7-17
7.4.1	Register Description	7-17
7.4.1.1	Module Configuration Register (MCR)	7-19

Port B4, B2, B1, AVEC

This signal group functions as three bits of parallel I/O and the autovector input. AVEC requests an automatic vector during an interrupt acknowledge cycle.

2.6 INTERRUPT REQUEST LEVEL (IRQ7, IRQ6, IRQ5, IRQ3)

These pins can be programmed to be either prioritized interrupt request lines or port B parallel I/O.

IRQ7, IRQ6, IRQ5, IRQ3

IRQ7, the highest priority, is nonmaskable. IRQ6–IRQ1 are internally maskable interrupts. Refer to **Section 5 CPU32** for more information on interrupt request lines.

Port B7, B6, B5, B3

These pins can be used as port B parallel I/O. Refer to **Section 4 System Integration Module** for more information on parallel I/O signals.

2.7 BUS CONTROL SIGNALS

These signals control the bus transfer operations of the MC68340. Refer to **Section 3 Bus Operation** for more information on these signals.

2.7.1 Data and Size Acknowledge (DSACK1, DSACK0)

These two active-low input signals allow asynchronous data transfers and dynamic data bus sizing between the MC68340 and external devices as listed in Table 2-3. During bus cycles, external devices assert DSACK1 and/or DSACK0 as part of the bus protocol. During a read cycle, this signals the MC68340 to terminate the bus cycle and to latch the data. During a write cycle, this indicates that the external device has successfully stored the data and that the cycle may terminate.

Table 2-3. DSACK≈ Encoding

DSACK 1	DSACK 0	Result
1	1	Insert Wait States in Current Bus Cycle
1	0	Complete Cycle—Data Bus Port Size Is 8 Bits
0	1	Complete Cycle—Data Bus Port Size Is 16 Bits
0	0	Reserved—Defaults to 16-Bit Port Size Can Be Used for 32-Bit DMA Cycles

2.7.2 Address Strobe (AS)

AS is an output timing signal that indicates the validity of both an address on the address bus and many control signals. AS is asserted approximately one-half clock cycle after the beginning of a bus cycle.

2.8.4 Read-Modify-Write Cycle (RMC)

This output signal identifies the bus cycle as part of an indivisible read-modify-write operation. It remains asserted during all bus cycles of the read-modify-write operation to indicate that bus ownership cannot be transferred.

2.9 EXCEPTION CONTROL SIGNALS

These signals are used by the MC68340 to recover from an exception.

2.9.1 Reset (RESET)

This active-low, open-drain, bidirectional signal is used to initiate a system reset. An external reset signal (as well as a reset from the SIM40) resets the MC68340 and all external devices. A reset signal from the CPU32 (asserted as part of the RESET instruction) resets external devices; the internal state of the CPU32 is not affected. The on-chip modules are reset, except for the SIM40. However, the module configuration register for each on-chip module is not altered. When asserted by the MC68340, this signal is guaranteed to be asserted for a minimum of 512 clock cycles. Refer to **Section 3 Bus Operation** for a description of bus reset operation and **Section 5 CPU32** for information about the reset exception.

2.9.2 Halt (HALT)

This active-low, open-drain, bidirectional signal is asserted to suspend external bus activity, to request a retry when used with BERR, or to perform a single-step operation. As an output, HALT indicates a double bus fault by the CPU32. Refer to **Section 3 Bus Operation** for a description of the effects of HALT on bus operation.

2.9.3 Bus Error (BERR)

This active-low input signal indicates that an invalid bus operation is being attempted or, when used with HALT, that the processor should retry the current cycle. Refer to **Section 3 Bus Operation** for a description of the effects of BERR on bus operation.

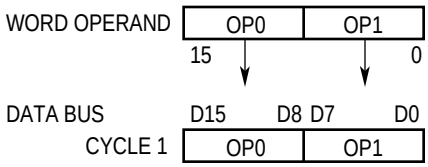
2.10 CLOCK SIGNALS

These signals are used by the MC68340 for controlling or generating the system clocks. See **Section 4 System Integration Module** for more information on the various clocking methods and frequencies.

2.10.1 System Clock (CLKOUT)

This output signal is the system clock output and is used as the bus timing reference by external devices. CLKOUT can be varied in frequency or slowed in low power stop mode to conserve power.

3.2.3.5 WORD OPERAND TO 16-BIT PORT, ALIGNED. The MC68340 drives the address bus with the desired address and the size pins to indicate a word operand.

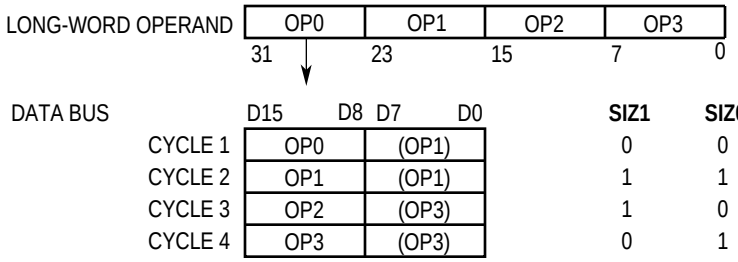


SIZ1	SIZ0	A0	$\overline{\text{DSACK1}}$	$\overline{\text{DSACK0}}$
1	0	0	0	X

For a read operation, the slave responds by placing the data on bits 15–0 of the data bus and asserting DSACK1 to indicate a 16-bit port. When DSACK1 is asserted, the MC68340 reads the data on the data bus and terminates the cycle.

For a write operation, the MC68340 drives the word operand on bits 15–0 of the data bus. The slave device then reads the entire operand from bits 15–0 of the data bus and asserts DSACK1 to terminate the bus cycle.

3.2.3.6 LONG-WORD OPERAND TO 8-BIT PORT, ALIGNED. The MC68340 drives the address bus with the desired address and the SIZx pins to indicate a long-word operand.



SIZ1	SIZ0	A0	$\overline{\text{DSACK1}}$	$\overline{\text{DSACK0}}$
0	0	0	1	0
1	1	1	1	0
1	0	0	1	0
0	1	1	1	0

For a read operation, shown in Figure 3-3, the slave responds by placing the most significant byte of the operand on bits 15–8 of the data bus and asserting DSACK0 to indicate an 8-bit port. The MC68340 reads the most significant byte of the operand (byte 0) from bits 15–8 and ignores bits 7–0. The MC68340 then decrements the transfer size counter, increments the address, initiates a new cycle, and reads byte 1 of the operand from bits 15–8 of the data bus. The MC68340 repeats the process of decrementing the transfer size counter, incrementing the address, initiating a new cycle, and reading a byte to transfer the remaining two bytes.

For a write operation, shown in Figure 3-4, the MC68340 drives the two most significant bytes of the operand on bits 15–0 of the data bus. The slave device then reads only the most significant byte of the operand (byte 0) from bits 15–8 of the data bus and asserts DSACK0 to indicate reception and an 8-bit port. The MC68340 then decrements the transfer size counter, increments the address, and writes byte 1 of the operand to bits 15–8 of the data bus. The MC68340 continues to decrement the transfer size counter, increment the address, and write a byte to transfer the remaining two bytes to the slave device.

4.2.2.6.2 Using the Periodic Timer as a Real-Time Clock. The periodic interrupt timer can be used as a real-time clock interrupt by setting it up to generate an interrupt with a one-second period. Rearranging the periodic timer period equation to solve for the desired count value:

$$\text{PITR count value} = \frac{\text{(PIT period) (EXTAL frequency)}}{\text{(Prescaler value) } (2^2)}$$

$$\text{PITR count value} = \frac{\text{(1) (32768)}}{\text{(512) } (2^2)}$$

$$\text{PITR count value} = 16 \text{ (decimal)}$$

Therefore, when using a 32.768-kHz crystal, the PITR should be loaded with a value of $10 with the prescaler enabled to generate interrupts at a one-second rate.

4.2.2.7 SIMULTANEOUS INTERRUPTS BY SOURCES IN THE SIM40. If multiple interrupt sources at the same interrupt level are simultaneously asserted in the SIM40, it will prioritize and service the interrupts in the following order: 1) software watchdog, 2) periodic interrupt timer, and 3) external interrupts.

4.2.3 Clock Synthesizer Operation

The clock synthesizer can operate with either an external crystal or an external oscillator for reference, using the internal phase-locked loop (PLL) and voltage-controlled oscillator (VCO), or an external clock can directly drive the clock signal at the operating frequency. The four modes of clock operation are listed in Table 4-1.

Table 4-1. Clock Operating Modes

Mode	Description	MODCK Reset Value	V_{CCSYN} Operating Value
Crystal Mode	External crystal or oscillator used with the on-chip PLL and VCO to generate a system clock and CLKOUT of programmable rates.	5 V	5 V
External Clock Mode without PLL	The desired operating frequency is driven into EXTAL resulting in a system clock and CLKOUT of the same frequency, not tightly coupled.	0 V	0 V
External Clock Mode with PLL	The desired operating frequency is driven into EXTAL, resulting in a system clock and CLKOUT of the same frequency, with a tight skew between input and output signals.	0 V	5 V
Limp Mode	Upon input signal loss for either clock mode using the PLL, operation continues at approximately one-half operating speed (affected by the value of the X-bit in the SYNCR).	X	5 V

In crystal mode (see Figure 4-4), the clock synthesizer can operate from the on-chip PLL and VCO, using a parallel resonant crystal connected between the EXTAL and XTAL pins, or an external oscillator connected to EXTAL as a reference frequency source. The oscillator circuit is shown in Figure 4-5. A 32.768-kHz watch crystal provides an inexpensive reference, but the reference crystal or external oscillator frequency can be any frequency in the range specified in **Section 11 Electrical Characteristics**. When

LOC—Loss of Clock Reset

1 = The last reset was caused by a loss of frequency reference to the clock synthesizer. This reset can only occur if the RSTEN bit in the SYNCR is set and the VCO is enabled.

SYS—System Reset

1 = The last reset was caused by the CPU32 executing a RESET instruction. The system reset does not load a reset vector or affect any internal CPU32 registers, SIM40 configuration registers, or the MCR in each internal peripheral module (DMA, timers, and serial modules). It will, however, reset external devices and all other registers in the peripheral modules.

4.3.2.4 SOFTWARE INTERRUPT VECTOR REGISTER (SWIV). The SWIV contains the 8-bit vector that is returned by the SIM40 during an IACK cycle in response to an interrupt generated by the software watchdog. This register can be read or written at any time. This register is set to the uninitialized vector, $0F, at reset.

SWIV $020

7	6	5	4	3	2	1	0
SWIV7	SWIV6	SWIV5	SWIV4	SWIV3	SWIV2	SWIV1	SWIV0
RESET:							
0	0	0	0	1	1	1	1

Supervisor Only

4.3.2.5 SYSTEM PROTECTION CONTROL REGISTER (SYPCR). The SYPCR controls the system monitors, the prescaler for the software watchdog, and the bus monitor timing. This register can be read at any time, but can be written only once after reset.

SYPCR $021

7	6	5	4	3	2	1	0
SWE	SWRI	SWT1	SWT0	DBFE	BME	BMT1	BMT0
RESET:							
0	0	0	0	0	0	0	0

Supervisor Only

SWE—Software Watchdog Enable

1 = Software watchdog is enabled.
0 = Software watchdog is disabled.

See **4.2.2.5 Software Watchdog** for more information.

SWRI—Software Watchdog Reset/Interrupt Select

1 = Software watchdog causes a system reset.
0 = Software watchdog causes a level 7 interrupt to the CPU32.

Table 5-2. Instruction Set Summary (Continued)

Opcode	Operation	Syntax
CMP2	Compare Rn < lower-bound or Rn > upper-bound and Set Condition Codes	CMP2 ⟨ea⟩,Rn
DBcc	If condition false then (Dn – 1 ⇒ Dn; If Dn ≠ –1 then PC + d ⇒ PC)	DBcc Dn,⟨label⟩
DIVS DIVSL	Destination/Source ⇒ Destination	DIVS.W ⟨ea⟩,Dn 32/16 ⇒ 16r:16q DIVS.L ⟨ea⟩,Dq 32/32 ⇒ 32q DIVS.L ⟨ea⟩,Dr:Dq 64/32 ⇒ 32r:32q DIVSL.L ⟨ea⟩,Dr:Dq 32/32 ⇒ 32r:32q
DIVU DIVUL	Destination/Source ⇒ Destination	DIVU.W ⟨ea⟩,Dn 32/16 ⇒ 16r:16q DIVU.L ⟨ea⟩,Dq 32/32 ⇒ 32q DIVU.L ⟨ea⟩,Dr:Dq 64/32 ⇒ 32r:32q DIVUL.L ⟨ea⟩,Dr:Dq 32/32 ⇒ 32r:32q
EOR	Source ⊕ Destination ⇒ Destination	EOR Dn,⟨ea⟩
EORI	Immediate Data ⊕ Destination ⇒ Destination	EORI #⟨data⟩,⟨ea⟩
EORI to CCR	Source ⊕ CCR ⇒ CCR	EORI #⟨data⟩,CCR
EORI to SR	If supervisor state the Source ⊕ SR ⇒ SR else TRAP	EORI #⟨data⟩,SR
EXG	Rx ⇔ Ry	EXG Dx,Dy EXG Ax,Ay EXG Dx,Ay EXG Ay,Dx
EXT EXTB	Destination Sign-Extended ⇒ Destination	EXT.W Dn extend byte to word EXT.L Dn extend word to long word EXTB.L Dn extend byte to long word
LLEGAL	SSP – 2 ⇒ SSP; Vector Offset ⇒ (SSP); SSP – 4 ⇒ SSP; PC ⇒ (SSP); SSp – 2 ⇒ SSP; SR ⇒ (SSP); Illegal Instruction Vector Address ⇒ PC	ILLEGAL
JMP	Destination Address ⇒ PC	JMP ⟨ea⟩
JSR	SP–4 ⇒ SP; PC ⇒ (SP) Destination Address ⇒ PC	JSR ⟨ea⟩
LEA	⟨ea⟩ ⇒ An	LEA ⟨ea⟩,An
LINK	SP – 4 ⇒ SP; An ⇒ (SP) SP ⇒ An, SP + d ⇒ SP	LINK An,#⟨displacement⟩
LPSTOP	If supervisor state Immediate Data ⇒ SR Interrupt Mask ⇒ External Bus Interface (EBI) STOP else TRAP	LPSTOP #⟨data⟩
LSL,LSR	Destination Shifted by ⟨count⟩ ⇒ Destination	LSd[1] Dx,Dy LSd[1] #⟨data⟩,Dy LSd[1] ⟨ea⟩
MOVE	Source ⇒ Destination	MOVE ⟨ea⟩,⟨ea⟩
MOVEA	Source ⇒ Destination	MOVEA ⟨ea⟩,An
MOVE from CCR	CCR ⇒ Destination	MOVE CCR,⟨ea⟩

Table 5-2. Instruction Set Summary (Concluded)

Opcode	Operation	Syntax
ROXL,ROXR	Destination Rotated with X by ⟨count⟩ ⇒ Destination	ROXd[1] Rx,Dy ROXd[1] #⟨data⟩,Dy ROXd[1] ⟨ea⟩
RTD	(SP) ⇒ PC; SP + 4 + d ⇒ SP	RTD #⟨displacement⟩
RTE	If supervisor state the (SP) ⇒ SR; SP + 2 ⇒ SP; (SP) ⇒ PC; SP + 4 ⇒ SP; restore state and deallocate stack according to (SP) else TRAP	RTE
RTR	(SP) ⇒ CCR; SP + 2 ⇒ SP; (SP) ⇒ PC; SP + 4 ⇒ SP	RTR
RTS	(SP) ⇒ PC; SP + 4 ⇒ SP	RTS
SBCD	$Destination_{10} - Source_{10} - X \Rightarrow$ Destination	SBCD Dx,Dy SBCD –(Ax),–(Ay)
Scc	If Condition True then 1s ⇒ Destination else 0s ⇒ Destination	Scc ⟨ea⟩
STOP	If supervisor state then Immediate Data ⇒ SR; STOP else TRAP	STOP #⟨data⟩
SUB	Destination – Source ⇒ Destination	SUB ⟨ea⟩,Dn SUB Dn,⟨ea⟩
SUBA	Destination – Source ⇒ Destination	SUBA ⟨ea⟩,An
SUBI	Destination – Immediate Data ⇒ Destination	SUBI #⟨data⟩,⟨ea⟩
SUBQ	Destination – Immediate Data ⇒ Destination	SUBQ #⟨data⟩,⟨ea⟩
SUBX	Destination – Source – X ⇒ Destination	SUBX Dx,Dy SUBX –(Ax),–(Ay)
SWAP	Register [31:16] ⇔ Register [15:0]	SWAP Dn
TAS	Destination Tested ⇒ Condition Codes; 1 ⇒ bit 7 of Destination	TAS ⟨ea⟩
TBLS	ENTRY(n) + {(ENTRY(n + 1) – ENTRY(n)) * Dx[7:0]} / 256 ⇒ Dx	TBLS.⟨size⟩ ⟨ea⟩, Dx TBLS.⟨size⟩ Dym:Dyn, Dx
TBLSN	ENTRY(n) × 256 + {(ENTRY(n + 1) – ENTRY(n)) * Dx [7:0]} ⇒ Dx	TBLSN.⟨size⟩ ⟨ea⟩,Dx TBLSN.⟨size⟩ Dym:Dyn, Dx
TBLU	ENTRY(n) + {(ENTRY(n + 1) – ENTRY(n)) * Dx[7:0]} / 256 ⇒ Dx	TBLU.⟨size⟩ ⟨ea⟩,Dx TBLU.⟨size⟩ Dym:Dyn, Dx
TBLUN	ENTRY(n) • 256 + {(ENTRY(n + 1) – ENTRY(n)) • Dx[7:0]} ⇒ Dx	TBLUN.⟨size⟩ ⟨ea⟩,Dx TBLUN.⟨size⟩ Dym:Dyn,Dx
TRAP	SSP – 2 ⇒ SSP; Format/Offset ⇒ (SSP); SSP – 4 ⇒ SSP; PC ⇒ (SSP); SSP – 2 ⇒ SSP; SR ⇒ (SSP); Vector Address ⇒ PC	TRAP #⟨vector⟩
TRAPcc	If cc then TRAP	TRAPcc TRAPcc.W #⟨data⟩ TRAPcc.L #⟨data⟩
TRAPV	If V then TRAP	TRAPV
TST	Destination Tested ⇒ Condition Codes	TST ⟨ea⟩
UNLK	An ⇒ SP; (SP) ⇒ An; SP + 4 ⇒ SP	UNLK An

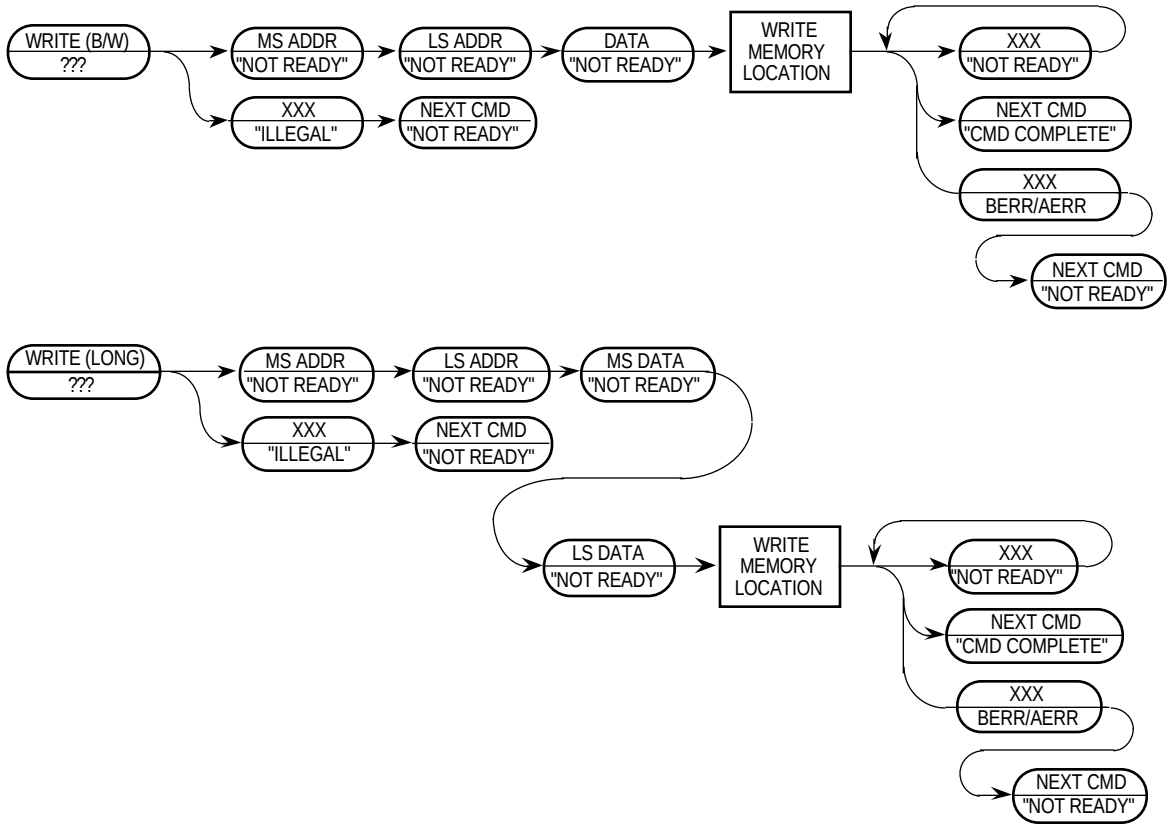
NOTE 1: d is direction, L or R.

space accessed. Only absolute addressing is supported. Valid data sizes include byte, word, and long word.

Command Format:

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

Command Sequence:



Operand Data:

Two operands are required for this instruction. The first operand is a long-word absolute address that specifies a location to which the operand data is to be written. The second operand is the data. Byte data is transmitted as a 16-bit word, justified in the least significant byte; 16- and 32-bit operands are transmitted as 16 and 32 bits, respectively.

Result Data:

Successful write operations return a status of $0FFFF. Bus or address errors on the write cycle are indicated by the assertion of bit 16 in the status message and by a data pattern of $0001.

5.6.2.8.10 Dump Memory Block (DUMP). DUMP is used in conjunction with the READ command to dump large blocks of memory. An initial READ is executed to set up the

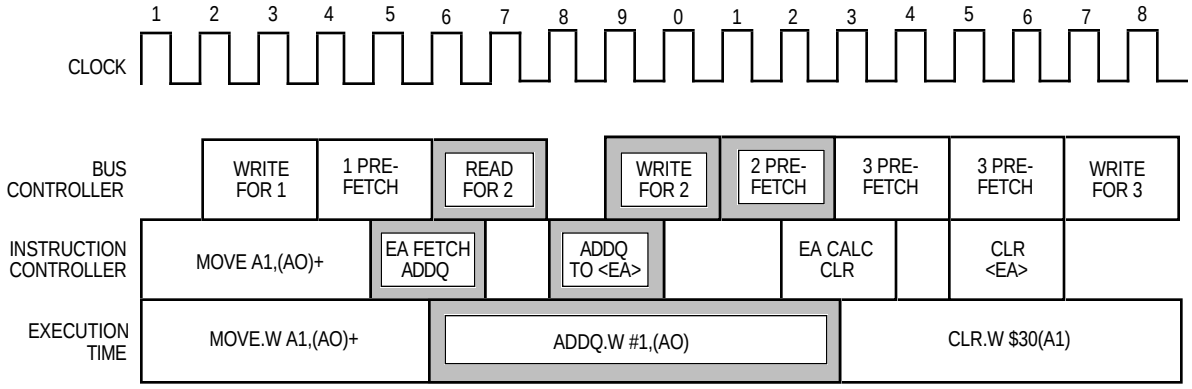


Figure 5-33. Example 1—Instruction Stream

5.7.2.2 TIMING EXAMPLE 2—BRANCH INSTRUCTIONS. Example 2 shows what happens when a branch instruction is executed for both the taken and not-taken cases. (see Figures 5-34 and 5-35). The instruction stream is for a simple limit check with the variable already in a data register.

Instructions

```
MOVEQ     #7, D1
CMP.L     D1, D0
BLE.B     NEXT
MOVE.L    D1, (A0)
```

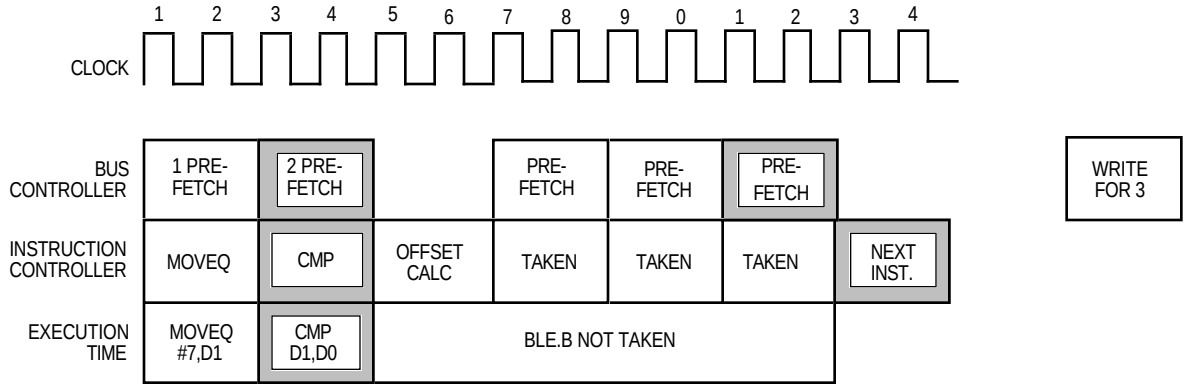


Figure 5-34. Example 2—Branch Taken

address. In the single-address mode with the source (read) device requesting mode of operation, this register is not used.

The manner in which the SAR and DAR change after each cycle depends upon the values in the CCR SSIZE and DSIZE fields and SAPI and DAPI bits, and the starting address in the SAR and DAR. If programmed to increment, the increment value is 1, 2, or 4 for byte, word, or long-word operands, respectively. If the address register is programmed to remain unchanged (no count), the register is not incremented after the operand transfer. The SAR and DAR are incremented if a bus error terminates the transfer. Therefore, either the SAR or the DAR contain the next address after the one that caused the bus error.

The BTC must be loaded with the number of byte transfers that are to occur. This register is decremented by 1, 2, or 4 at the end of each transfer. The FCR must be loaded with the source and destination function codes. Although these function codes may not be used in the address decode for the memory or peripheral, they are provided if needed. The CSR must be cleared for channel startup.

Once the channel has been initialized, it is started by writing a one to the STR bit in the CCR. Programming the channel for internal request causes the channel to request the bus and start transferring data immediately. If the channel is programmed for external request, DREQ≈ must be asserted before the channel requests the bus. The DREQ≈ input is ignored until the channel is started, since the channel does not recognize transfer requests until it is active.

If any fields in the CCR are modified while the channel is active, that change is effective immediately. To avoid any problems with changing the setup for the DMA channel, a zero should be written to the STR bit in the CCR to halt the DMA channel at the end of the current bus cycle.

6.6.2 Data Transfers

Each operand transfer requires from one to five bus cycles to complete. Once a bus request is recognized and the operand transfer begins, both the source (read) cycle and/or the destination (write) cycle occur before a new bus request may be honored, even if the new bus request is of higher priority.

6.6.2.1 INTERNAL REQUEST TRANSFERS. Internally generated request transfers are accessed as two-clock bus cycles. (The IMB can access on-chip peripherals in two clocks.) The percentage of bus bandwidth utilization can be limited for internal request transfers.

6.6.2.2 EXTERNAL REQUEST TRANSFERS. In single-address mode, only one bus cycle is run for each request. Since the operand size must be equal to the device port size in single-address mode, the number of normally terminated bus cycles executed during a transfer operation is always equal to the value programmed into the corresponding size field of the CCR. The sequencing of the address bus follows the programming of the CCR and address register (SAR or DAR) for the channel.

7.4.1.15 OUTPUT PORT CONTROL REGISTER (OPCR). The OPCR individually configures four bits of the 8-bit parallel OP for general-purpose use or as an auxiliary function serving the communication channels. This register can only be written when the serial module is enabled (i.e., the STP bit in the MCR is cleared).

OPCR $71D

7	6	5	4	3	2	1	0
OP7 T≈RDYB	OP6 T≈RDYA	OP5 R≈RDYB	OP4 R≈RDYA	OP3	OP2	OP1 RTSB	OP0 RTSA
RESET: 0	0	0	0	0	0	0	0

Write Only — Supervisor/User

NOTE

OP bits 7, 5, 3, and 2 are not pinned out on the MC68340; thus changing bits 7, 5, 3, and 2 of this register has no effect.

OP6—Output Port 6/T≈RDYA

1 = The OP6/T≈RDYA pin functions as the transmitter-ready signal for channel A. The signal reflects the complement of the value of bit 2 of the SRA; thus, T≈RDYA is a logic zero when the transmitter is ready.
0 = The OP6/T≈RDYA pin functions as a dedicated output. The signal reflects the complement of the value of bit 6 of the OP.

OP4—Output Port 4/R≈RDYA

1 = The OP4/R≈RDYA pin functions as the FIFO-full or receiver-ready signal for channel A (depending on the value of bit 6 of MR1A). The signal reflects the complement of the value of ISR bit 1; thus, R≈RDYA is a logic zero when the receiver is ready.
0 = The OP4/R≈RDYA pin functions as a dedicated output. The signal reflects the complement of the value of bit 4 of the OP.

OP1—Output Port 1/RTSB

1 = The OP1/RTSB pin functions as the ready-to-send signal for channel B. The signal is asserted and negated according to the configuration programmed by RxRTS bit 7 in the MR1B for the receiver and TxRTS bit 5 in the MR2B for the transmitter.
0 = The OP1/RTSB pin functions as a dedicated output. The signal reflects the complement of the value of bit 1 of the OP.

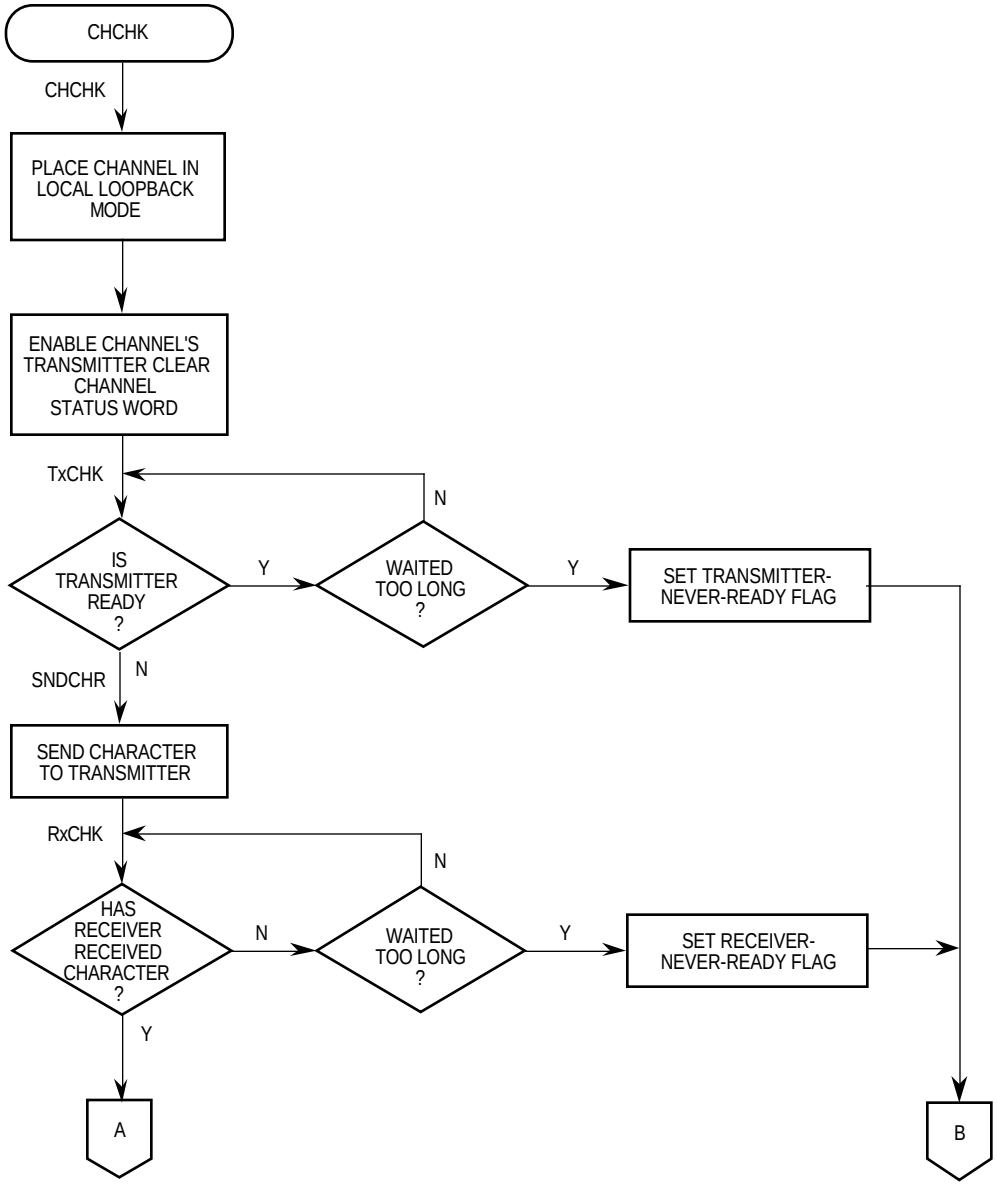


Figure 7-10. Serial Module Programming Flowchart (2 of 5)

Disabled—TOUTx is disabled and three-stated.

Toggle Mode—If the timer is disabled (SWR = 0) when this encoding is programmed, TOUTx is immediately set to zero. If the timer is enabled (SWR = 1), timeout events (counter reaches $0000) toggle TOUTx. In the input capture/output compare mode, TOUTx is immediately set to zero if the timer is disabled (SWR = 0). If the timer is enabled (SWR = 1), timer compare events toggle TOUTx. (Timer compare events occur when the counter reaches the value stored in the COM.)

Zero Mode—If the timer is disabled (SWR = 0) when this encoding is programmed, TOUTx is immediately set to zero. If the timer is enabled (SWR = 1), TOUTx will be set to zero at the next timeout. In the input capture/output compare mode, TOUTx is immediately set to zero if the timer is disabled (SWR = 0). If the timer is enabled (SWR = 1), TOUTx will be set to zero at timeouts and set to one at timer compare events. If the COM is $0000, TOUTx will be set to zero at the timeout/timer compare event.

One Mode—If the timer is disabled (SWR = 0) when this encoding is programmed, TOUTx is immediately set to one. If the timer is enabled (SWR = 1), TOUTx will be set to one at the next timeout. In the input capture/output compare mode, TOUTx is immediately set to one if the timer is disabled (SWR = 0). If the timer is enabled (SWR = 1), TOUTx will be set to one at timeouts and set to zero at timer compare events. If the COM is $0000, TOUTx will be set to one at the timeout/timer compare event.

8.4.4 Status Register (SR)

The SR contains timer status information as well as the state of the prescaler. This register is updated on the rising edge of the system clock when a read of its location is not in progress, allowing the most current information to be contained in this register. The register can be read, and the TO, TG, and TC bits can be written when the timer module is enabled (i.e., the STP bit in the MCR is cleared).

SR $608, $648

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ	TO	TG	TC	TGL	ON	OUT	COM	PO7	PO6	PO5	PO4	PO3	PO2	PO1	PO0
RESET (TGATE≈ NEGATED):															
0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1
RESET (TGATE≈ ASSERTED):															
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Supervisor/User

IRQ—Interrupt Request bit

The positioning of this bit in the most significant location in this register allows it it be conditionally tested as if it were a signed binary integer.

1 = An interrupt condition has occurred. This bit is the logical OR of the enabled TO, TG, and TC interrupt bits.

0 = The bit(s) that caused the interrupt condition has been cleared. If an IRQ≈ signal has been asserted, it is negated when this bit is cleared.

TO—Timeout Interrupt

1 = The counter has transitioned from $0001 to $0000, and the counter has rolled over. This bit does not affect the programmed IRQ≈ signal if the IE2 bit in the CR is cleared.

0 = This bit is cleared by the timer whenever the RESET signal is asserted on the IMB, regardless of the mode of operation. This bit may also be cleared by writing a one to it. Writing a zero to this bit does not alter its contents. This bit is not affected by disabling the timer (SWR = 0).

TG—Timer Gate Interrupt

1 = This bit is set whenever the CR TGE bit is set and the TGATE≈ signal transitions in the manner to which the particular mode of operation responds. Refer to **8.3 Operating Modes** for more details. This bit does not affect the programmed IRQ≈ signal if the IE1 bit in the CR is cleared.

0 = This bit is cleared by the timer whenever the RESET signal is asserted on the IMB, regardless of the mode of operation. This bit may also be cleared by writing a one to it. Writing a zero to this bit does not alter its contents. This bit is not affected by disabling the timer (SWR = 0).

TC—Timer Compare Interrupt

1 = This bit is set when the counter transitions (off a clock/event falling edge) to the value in the COM. This bit does not affect the programmed IRQ≈ signal if the IE0 bit in the CR is cleared.

0 = This bit is cleared by the timer whenever the RESET signal is asserted on the IMB, regardless of the mode of operation. This bit may also be cleared by writing a one to it. Writing a zero to this bit does not alter its contents. This bit is not affected by disabling the timer (SWR = 0).

TGL—TGATE≈ Level

1 = The TGATE≈ signal is negated.

0 = The TGATE≈ signal is asserted.

ON—Counter Enabled

1 = This bit is set whenever the SWR and CPE bits are set in the CR. If the CR TGE bit is set, TGATE≈ must also be asserted (except in the input capture/output compare mode) since this signal then controls the enabling and disabling of the counter. If all these conditions are met, the counter is enabled and begins counting down.

0 = The counter is not enabled and does not begin counting down.

OUT—Output Level

1 = TOUTx is a logic one.

0 = TOUTx is a logic zero, or the pin is three-stated.

COM—Compare Bit

This bit is used to indicate when the counter output value is at or between the value in the COM and $0000 (timeout).

```
* Module configuration register:
* Timer1 module is set for normal operation, ignore FREEZE.
* Supervisor/user timer1 registers unrestricted.
* Interrupt arbitration at priority $03.
        MOVE.W      #$0003,MCR1(A0)

* Initialize timer1 interrupt level to 2 and vector to $0F
        MOVE.W      #$020F,IR1(A0)

* Initialize preload 1 to 3
        MOVE.W      #$0003,PRLD11(A0)


* Initialize the compare register to 0
        CLR.W       COM1(A0)

* Control register 1:
* Enable timer1, no interrupts are enabled, TGATE signal has no effect.
* Use the selected clock for the counter clock, and enable it.
* Selected clock is 1/2 system's freq. Square-wave generation, toggle TOUT.
        MOVE.W      #$8205,CR1(A0)
*******************************************************************************
        END
*******************************************************************************



*******************************************************************************
* MC68340 basic timer module register initialization example code.
* This code is used to initialize the 68340's internal timer module
* registers, providing basic functions for operation.
* It sets up timer1 for pulse-width measurement. In this mode, the number
* of clock cycles during a particular event are counted. The event is
* defined by the assertion and negation of TGATE.
*******************************************************************************
*******************************************************************************
* equates
*******************************************************************************
MBAR        EQU   $0003FF00   Address of SIM40 Module Base Address Reg.
MODBASE     EQU   $FFFFF000   SIM40 MBAR address value

*******************************************************************************
* Timer1 module equates
TIMER1      EQU   $600        Offset from MBAR for timer1 module regs
MCR1        EQU   $0          MCR for timer1
```

```
* Timer1 register offsets from timer1 base address
IR1             EQU   $604          interrupt register timer1
CR1             EQU   $606          control register timer1
SR1             EQU   $608          status register timer1
CNTR1           EQU   $60A          counter register timer1
COM1            EQU   $610          compare register timer1


*************************************************************************
*************************************************************************
* Initialize Timer1
*************************************************************************
        LEA    MODBASE+TIMER1,A0    Pointer to timer1 module

* Disable timer1
        CLR.W         CR1(A0)

* Allow TGATE to negate and assert so that an accurate count will result.
* If SR1 TGL bit=1, continue looping. TGATE is negated.
        LOOP1         BTST.B        #$3,SR1(A0)
        BNE.B         LOOP1
* If TGL bit=0, continue looping. TGATE is asserted.
        LOOP2         BTST.B        #$3,SR1(A0)
        BEQ.B         LOOP2

* Ready to initialize timer1, TGATE is negated.

* Module configuration register:
* Timer1 module is set for normal operation, ignore FREEZE.
* Supervisor/user timer1 registers unrestricted.
* Interrupt arbitration at priority $03.
        MOVE.W       #$0003,MCR1(A0)

* Initialize timer1 interrupt level to 2 and vector to $0F
        MOVE.W       #$020F,IR1(A0)

* Initialize the compare register to 0
        CLR.W         COM1(A0)

* Clear the SR1 TG bit (by writing a 1) to use as a flag
        MOVE.B       #$20,SR1(A0)

* Control register 1:
* Enable timer1, no interrupts are enabled, TGATE signal used to control
* the counter. Use the selected clock for the counter clock, and enable it.
* Selected clock is 1/2 system's freq. Pulse-width measurement,
* disable TOUT.
```

includes a 3-bit instruction register without parity, consisting of a shift register with three parallel outputs. Data is transferred from the shift register to the parallel outputs during the update-IR controller state. The three bits are used to decode the four unique instructions listed in Table 9-3.

The parallel output of the instruction register is reset to all ones in the test-logic-reset controller state. Note that this preset state is equivalent to the BYPASS instruction.

Table 9-3. Instructions

Code			Instruction
B2	B1	B0	
0	0	0	EXTEST
0	0	1	SAMPLE/PRELOAD
X	1	X	BYPASS
1	0	0	HI-Z
1	0	1	BYPASS

During the capture-IR controller state, the parallel inputs to the instruction shift register are loaded with the standard 2-bit binary value (01) into the two least significant bits and the loss-of-crystal (LOC) status signal into bit 2. The parallel outputs, however, remain unchanged by this action since an update-IR signal is required to modify them.

The LOC status bit of the instruction register indicates whether an internal clock is detected when operating with a crystal clock source. The LOC bit is clear when a clock is detected and set when it is not. The LOC bit is always clear when an external clock is used. The LOC bit can be used to detect faulty connectivity when a crystal is used to clock the device.

9.4.1 EXTEST (000)

The external test (EXTEST) instruction selects the 132-bit boundary scan register. EXTEST asserts internal reset for the MC68340 system logic to force a predictable benign internal state while performing external boundary scan operations.

By using the TAP, the register is capable of a) scanning user-defined values into the output buffers, b) capturing values presented to input pins, c) controlling the direction of bidirectional pins, and d) controlling the output drive of three-state output pins. For more details on the function and uses of EXTEST, please refer to the IEEE 1149.1 document.

9.4.2 SAMPLE/PRELOAD (001)

The SAMPLE/PRELOAD instruction selects the 132-bit boundary scan register and provides two separate functions. First, it provides a means to obtain a snapshot of system data and control signals. The snapshot occurs on the rising edge of TCK in the capture-DR controller state. The data can be observed by shifting it transparently through the boundary scan register.

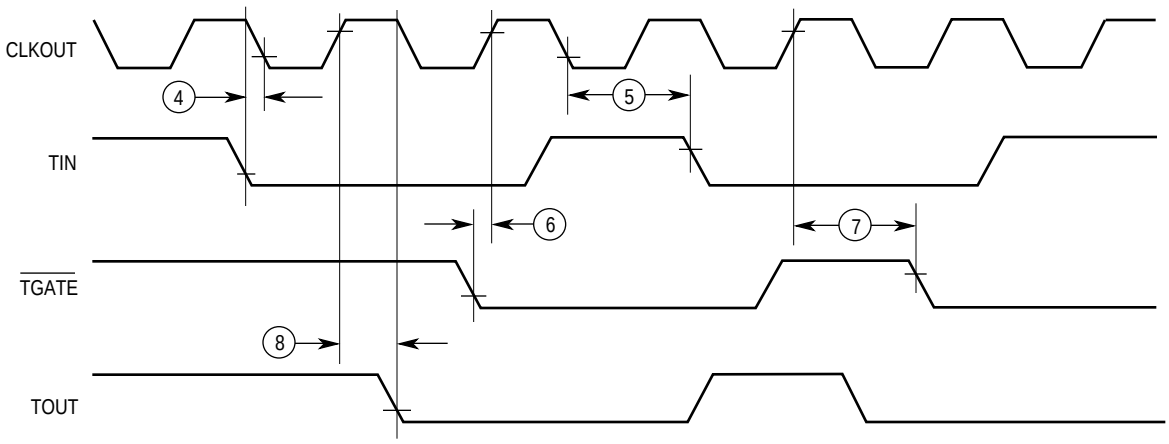


Figure 11-14. Timer Module Signal Timing Diagram

— J —

JTAG, 4-2

— L —

Late Bus Error, 3-34
LG Bit, 5-56–5-57
Limp Mode, 4-19, 4-29
Local Loopback Mode, 7-14, 7-38
Location of Modules, 4-2–4-3, 4-20
Logical Instructions, 4-48
Long-Word
  Read
    8-Bit Port, Timing, 3-11
    16-Bit Port, Timing, 3-13
  Write
    8-Bit Port, Timing, 3-12
    16-Bit Port, Timing, 3-13
Looping Modes, 7-14–7-15
Loss of Input Signal, 4-9, 4-11, 4-29
Low Power Stop
  Mode, 3-23, 4-13, 4-17, 4-29, 10-12
  Low-Voltage, 10-10, 10-11
LPSTOP Cycle, 3-23

— M —

MAID Bits, 6-25, 6-36
Master Station, 7-15
Maximum Rating, 11-1
MC68681, 7-4
Memory
  Access Times, 10-7
  Interfacing, 10-5, 10-10
Memory-to-Memory Transfer, 6-1, 6-3, 6-5
Microbus Controller, 5-89, 5-91
Microsequencer Operation, 5-89–5-90
Misaligned Operands, 3-7
MISC Bits, 7-28
MODCK Signal, 2-9, 4-7, 4-35
MODE Bits, 8-6, 8-8–8-10, 8-12–8-14, 8-16, 8-22, 8-28
Mode Register 1, 7-13, 7-16–7-17, 7-22, 7-34, 7-47
Mode Register 2, 7-4, 7-17, 7-38, 7-47
Module Base Address Register, 4-2, 4-20, 4-36
  Access, 3-27
Module
  Configuration Register, 4-21, 4-36, 6-23, 7-19, 7-46, 8-18, 8-27
  Locations, 4-3, 4-5
MOVE Instruction Timing Table, 5-101–5-102
MOVEM
  Faults, 5-56, 5-58–5-59, 5-61
MOVEP Faults, 5-55–5-56
Multidrop Mode, 7-15–7-16, 7-23
  Timing, 7-16
Multiprocessor Systems, 5-61

— N —

NCS Bit, 4-31
Negate RTS Command, 7-29
Negative Tails, 5-93–5-94
No Operation Command, 5-86

— O —

OC Bits, 8-6–8-8, 8-10, 8-22, 8-28
OE Bit, 7-13, 7-25, 7-28
ON Bit, 8-6, 8-8, 8-11, 8-24
One Mode, 8-23
OP0, 7-6, 7-36–7-38
OP1, 7-7, 7-36–7-38
OP4, 7-7, 7-36–7-37
OP6, 7-7, 7-36–7-37
Opcode Tracking in Loop Mode, 5-88
Operand
  Faults, 5-56, 5-58, 5-61
  Misalignment, 3-7
  Size Field, 5-73
Operation Field, 5-73
Ordering Information, 12-1
OUT Bit, 8-7–8-8, 8-10, 8-24
Output Port
  Control Register, 7-36, 7-46
  Data Register, 7-6–7-7, 7-22, 7-37
Overrun Error, 7-11, 7-25

— P —

Package Dimensions, 12-6–12-7
Package Types, 1-9, 12-1–12-2, 12-4
Parity
  Error, 7-11, 7-24
  Mode, 7-23
  Type, 7-23
PCLK Bit, 8-21, 8-22, 8-27
PE Bit, 7-11, 7-13, 7-24, 7-28
Period Measurement, 8-13
Periodic Interrupt
  Control Register, 4-7, 4-26, 4-37
  Generation, 8-6, 8-8, 8-9
  Timer Register, 4-7, 4-27, 4-37
  Timer, 4-1, 4-4, 4-7, 4-9, 4-17
Periodic Timer Period Calculation, 4-8
Phase Comparator, 4-11–4-12
Phase-Locked Loop, 4-9–4-12, 10-1–10-2
Pin Group, 12-3, 12-5
PIRQL Bits, 4-7, 4-26, 4-37
PITR Bits, 4-27, 4-37
PIV Bits, 4-26
PM Bits, 7-23, 7-47
PO Bits, 8-25
Port A
  Data Direction Register, 4-34
  Data Register, 4-34