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	M68000
Number of Cores/Bus Width	1 Core, 8/16-Bit
Speed	16MHz
Co-Processors/DSP	Communications; RISC CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 85°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=mc68302cag16vc

Table 2-2. M68000 Instruction Set Summary

Mnemonic	Description
ABCD ADD AND ASL ASR	Add Decimal with Extend Add Logical AND Arithmetic Shift Left Arithmetic Shift Right
Bcc BCHG BCLR BRA BSET BSR BTST	Branch Conditionally Bit Test and Change Bit Test and Clear Branch Always Bit Test and Set Branch to Subroutine Bit Test
CHK CLR CMP	Check Register Against Bounds Clear Operand Compare
DBcc DIVS DIVU	Decrement and Branch Conditionally Signed Divide Unsigned Divide
EOR EXG EXT	Exclusive OR Exchange Registers Sign Extend
JMP JSR	Jump Jump to Subroutine
LEA LINK LSL LSR	Load Effective Address Link Stack Logical Shift Left Logical Shift Right
MOVE MULS MULU	Move Source to Destination Signed Multiply Unsigned Multiply
NBCD NEG NOP NOT	Negate Decimal with Extend Negate No Operation Ones Complement
OR	Logical OR
PEA	Push Effective Address
RESET ROL ROR ROXL ROXR RTE RTR RTS	Reset External Devices Rotate Left without Extend Rotate Right without Extend Rotate Left with Extend Rotate Right with Extend Return from Exception Return and Restore Return from Subroutine
SBCD Scc STOP SUB SWAP	Subtract Decimal with Extend Set Conditionally Stop Subtract Swap Data Register Halves
TAS TRAP TRAPV TST	Test and Set Operand Trap Trap on Overflow Test
UNLK	Unlink

CFC—Compare Function Code

0 = The FC bits in the BAR are ignored. Accesses to the IMP 4K-byte block occur without comparing the FC bits.
1 = The FC bits in the BAR are compared. The address space compare logic uses the FC bits to detect address matches.

Bits 11–0—Base Address

The high address field is contained in bit 11–0 of the BAR. These bits are used to set the starting address of the dual-port RAM. The address compare logic uses only the most significant bits to cause an address match within its block size.

2.8 MC68302 MEMORY MAP

The following tables show the additional registers added to the M68000 to make up the MC68302. All of the registers are memory-mapped. Four entries in the M68000 exception vectors table (located in low RAM) are reserved for addresses of system configuration registers (see Table 2-6) that reside on-chip. These registers have fixed addresses of $0F0–$0FF. All other on-chip peripherals occupy a 4K-byte relocatable address space. When an on-chip register or peripheral is accessed, the internal access (IAC) pin is asserted.

Table 2-6. System Configuration Register

Address	Name	Width	Description	Reset Value
$0F0	RES	16	Reserved	
$0F2*	BAR	16	Base Address Register	BFFF
$0F4*	SCR	32	System Control Register	0000 0F00
$0F8	RES	16	Reserved	
$0FA	CKCR	16	Clock Control Register	0000
$0FC	RES	32	Reserved	

*Reset only upon a total system reset.

The internal 1176-byte dual-port RAM has 576 bytes of system RAM (see Table 2-7) and 576 bytes of parameter RAM (see Table 2-8).

Table 2-7. System RAM

Address	Width	Block	Description
Base + 000 • • • Base + 23F	576 Bytes	RAM	User Data Memory
Base +240 • • • Base + 3FF			Reserved (Not Implemented)

The parameter RAM contains the buffer descriptors for each of the three SCC channels, the SCP, and the two SMC channels. The memory structures of the three SCC channels are

Option 2. The external peripheral can generate the vector. In this case the external device must decode the interrupt acknowledge cycle, put out the 8-bit vector, and generate $\overline{\text{DTACK}}$. The decoding of the interrupt acknowledge cycle can be provided by the $\overline{\text{IACK7}}$, $\overline{\text{IACK6}}$, and $\overline{\text{IACK1}}$ signals (enabled in the PBCNT register) if either normal or dedicated mode is chosen. These signals eliminate the need for external logic to perform the decoding of the A19–A16, A3–A1, and FC2–FC0 pins externally to detect the interrupt acknowledge cycle. If the $\overline{\text{IACK}}$ signals are not needed, they can be regained as general purpose parallel I/O pins. The external device must generate $\overline{\text{DTACK}}$ in this mode, and $\overline{\text{DTACK}}$ is an input to the IMP.

Option 3. The external peripheral can assert the $\overline{\text{AVEC}}$ pin to cause the M68000 to use an autovector. In this case, $\overline{\text{DTACK}}$ should not be asserted by the external device. $\overline{\text{AVEC}}$ is recognized by the M68000 core on the falling edge of S4 and should meet the asynchronous setup time to the falling edge of S4. The $\overline{\text{IACK}}$ signals can be used to help generate the $\overline{\text{AVEC}}$ signal for priority levels 1, 6, and 7, if needed.

NOTE

If $\overline{\text{AVEC}}$ is asserted during an interrupt acknowledge cycle, an autovector is taken, regardless of the vector on the bus. $\overline{\text{AVEC}}$ should not be asserted during level 4 interrupt acknowledge cycles.

When the IMP generates the vector, the following procedure is used. The three most significant bits of the interrupt vector number are programmed by the user in the GIMR. These three bits are concatenated with five bits generated by the interrupt controller to provide an 8-bit vector number to the core. The interrupt controller's encoding of the five low-order bits of the interrupt vector is shown in Table 3-5. An example vector calculation is shown in Figure 3-4. When the core initiates an interrupt acknowledge cycle for level 4 and there is no internal interrupt pending, the interrupt controller encodes the error code 00000 onto the five low-order bits of the interrupt vector.

2. $\overline{BG}$ will be an input to the IDMA and SDMA from the external M68000 bus, rather than being an output from the MC68302. When BG is sampled as low by the MC68302, it waits for $\overline{AS}$, $\overline{BERR}$, $\overline{HALT}$, and $\overline{BGACK}$ to be negated, and then asserts $\overline{BGACK}$ and performs one or more bus cycles. See Section 6 for timing diagrams.
3. $\overline{BCLR}$ will be an input to the IDMA, but will remain an output from the SDMA.
4. The interrupt controller will output its interrupt request lines ($\overline{IPL0}$, $\overline{IPL1}$, $\overline{IPL2}$) normally sent to the M68000 core on pins $\overline{IOUT0}$, $\overline{IOUT1}$, and $\overline{IOUT2}$, respectively. $\overline{AVEC}$, $\overline{RMC}$, and $\overline{CS0}$, which share pins with $\overline{IOUT0}$, $\overline{IOUT1}$, and $\overline{IOUT2}$, respectively, are not available in this mode.

DISCPU should remain continuously high during disable CPU mode operation. Although the $\overline{CS0}$ pin is not available as an output from the device in disable CPU mode, it may be enabled to provide $\overline{DTACK}$ generation. In disable CPU mode, BR0 is initially $C000.

Accesses by an external master to the MC68302 RAM and registers may be asynchronous or synchronous to the MC68302 clock. (This feature is actually available regardless of disable CPU mode). See the SAM and EMWS bits in the SCR for details.

In disable CPU mode, the interrupt controller may be programmed to generate or not generate interrupt vectors during interrupt acknowledge cycles. When multiple MC68302 devices share a single M68000 bus, vector generation at level 4 should be prevented on all but one MC68302. When using disable CPU mode to implement an interface, such as between the MC68020 and a single MC68302, vector generation can be enabled. For this purpose, the VGE bit is defined.

VGE—Vector Generation Enable

0 = In disable CPU mode, the MC68302 will not output interrupt vectors during interrupt acknowledge cycles.
1 = In disable CPU mode, the MC68302 will output interrupt vectors for internal level 4 interrupts (and for levels 1, 6, and/or 7 as enabled in the interrupt controller) during interrupt acknowledge cycles.

NOTE

Do not use the function code value “111” during external accesses to the IMP, except during interrupt acknowledge cycles.

In disable CPU mode, the low-power modes will be entered immediately upon the setting of the LPEN bit in the SCR by an external master. In this case, low-power mode will continue until the LPEN bit is cleared. Users may wish to use a low-power mode in conjunction with disable CPU mode to save power consumed by the disabled M68000 core.

All MC68302 functionality not expressly mentioned in this section is retained in disable CPU mode and operates identically as before.

NOTE

Even without the use of the disable CPU logic, another processor can be granted access to the IMP on-chip peripherals by re-

CD10–CD0—Clock Divider

The clock divider bits and the prescaler determine the baud rate generator output clock rate. CD10–CD0 are used to preset an 11-bit counter that is decremented at the prescaler output rate. The counter is not otherwise accessible to the user. When the counter reaches zero, it is reloaded with the clock divider bits. Thus, a value of $7FF in CD10–CD0 produces the minimum clock rate (divide by 2048); a value of $000 produces the maximum clock rate (divide by 1).

NOTE

Because of SCC clocking restrictions, the maximum baud rate that may be used to clock an SCC is divide by 3.

When dividing by an odd number, the counter ensures a 50% duty cycle by asserting the terminal count once on a clock high and next on a clock low. The terminal count signals the counter expiration and toggles the clock.

DIV4—SCC Clock Prescaler Divide by 4

The SCC clock prescaler bit selects a divide-by-1 (DIV4 = 0) or divide-by-4 (DIV4 = 1) prescaler for the clock divider input. The divide-by-4 option is useful in generating very slow baud rates.

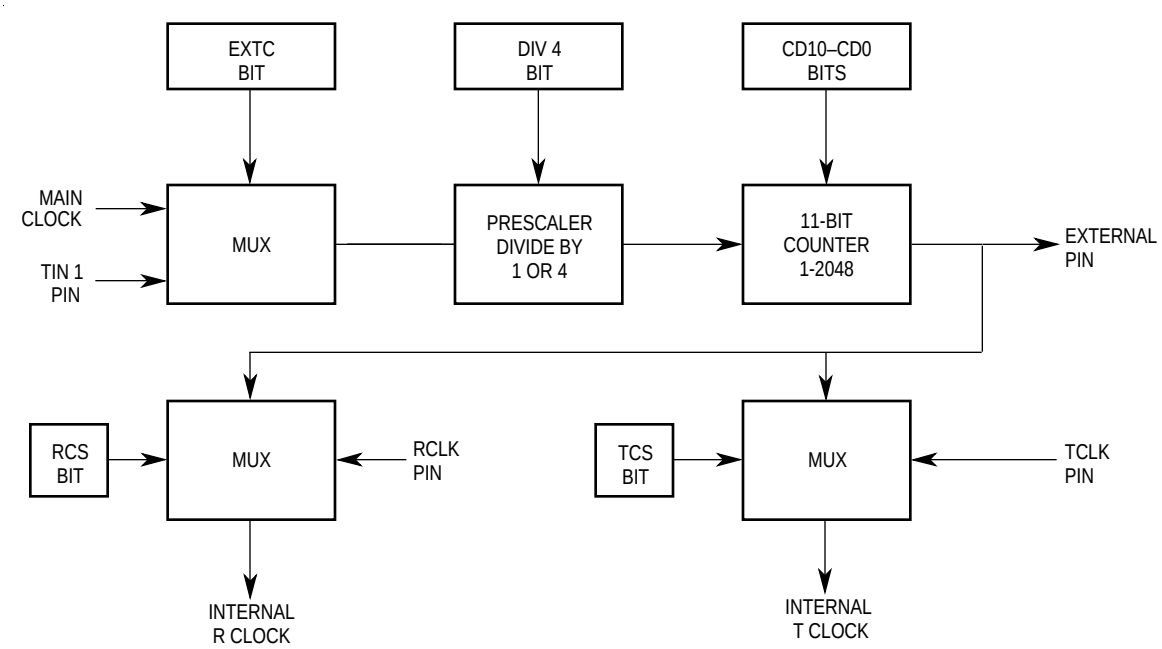


Figure 4-12. SCC Baud Rate Generator

4.5.2.1 Asynchronous Baud Rate Generator Examples

The UART circuitry always uses a clock that is 16x the baud rate. The ratio of the 16x UART clock to the system parallel clock must not exceed 1:2.5. For an internally supplied clock, an integer divider value must be used; therefore, the divider must be 3 or greater. Thus, using a clock divider value of 3 (programmed as 2 in the SCON) and a 16.67-MHz crystal gives a UART clock rate of 5.56 MHz and a baud rate of 347 kbaud. Assuming again a 16.67-MHz

	15	0
OFFSET + 0	STATUS AND CONTROL	
OFFSET + 2	DATA LENGTH	
OFFSET + 4	HIGH-ORDER DATA BUFFER POINTER (only lower 8 bits used, upper 8 bits must be 0)	
OFFSET + 6	LOW-ORDER DATA BUFFER POINTER	

Figure 4-16. SCC Buffer Descriptor Format

For frame-oriented protocols (HDLC, BISYNC, DDCMP, V.110), a frame may reside in as many buffers as are necessary (transmit or receive). Each buffer has a maximum length of 64K–1 bytes. The CP does not assume that all buffers of a single frame are currently linked to the BD table, but does assume that the unlinked buffers will be provided by the processor in time to be either transmitted or received. Failure to do so will result in a TXE error being reported by the CP.

For example, assume the first six buffers of the transmit BD table have been transmitted and await processing by the M68000 core (with all eight buffers used in the circular queue), and a three-buffer frame awaits transmission. The first two buffers may be linked to the remaining two entries in the table as long as the user links the final buffer into the first entry in the BD table before the IMP attempts its transmission. If the final buffer is not linked in time to the BD table by the time the CP attempts its transmission, the CP will report an underrun error.

Buffers allocated to an SCC channel may be located in either internal or external memory. Memory allocation occurs for each BD individually. If internal memory is selected, the CP uses only the lower 11 address bits (A10–A0) as an offset to the internal dual-port RAM. Accesses to the internal memory by the CP are one clock cycle long and occur without arbitration. If external memory is selected, the pointers to the data buffers are used by the CP as 24 bits of address.

Extra caution should be used if function codes are included in the decoding of the external buffer address (e.g., in the on-chip chip select logic). The function code of this SCC channel must be set before external buffers can be accessed; it can then be changed only when the user is sure that the CP is not currently accessing external buffers for that channel. There are six separate function code registers located in the parameter RAM for the three SCC channels: three for receive data buffers (RFCR) and three for transmit data buffers (TFCR).

NOTE

The RFCR and TFCR function codes should never be initialized to “111.”

The CP processes the transmit BDs in a straightforward fashion. Once the transmit side of an SCC is enabled, it starts with the first BD in that SCC's transmit BD table, periodically checking a bit to see if that BD is "ready". Once it is ready, it will process that BD, reading a word at a time from its associated buffer, doing certain required protocol processing on the data, and moving resultant data to the SCC transmit FIFO. When the first buffer has been processed, the CP moves on to the next BD, again waiting for that BD's “ready” bit to be set. Thus, the CP does no look-ahead BD processing, nor does it skip over BDs that are not ready. When the CP sees the “wrap” bit set in a BD, it goes back to the beginning of the BD

CR—Rx CRC Error

This frame contains a CRC error.

OV—Overrun

A receiver overrun occurred during frame reception.

CD—Carrier Detect Lost

The carrier detect signal was negated during frame reception. This bit is valid only when working in NMSI mode.

Data Length

The data length is the number of octets written to this BD's data buffer by the HDLC controller. It is written by the CP once as the BD is closed.

When this BD is the last BD in the frame (L = 1), the data length contains the total number of frame octets (including any previous linked receive data buffers and two or four bytes for the CRC) in the frame. This behavior is useful for determining the total number of octets received, even if MFLR was exceeded.

NOTE

The actual amount of memory allocated for this buffer should be even and greater than or equal to the contents of maximum receive buffer length register (MRBLR).

Rx Buffer Pointer

The receive buffer pointer, which always points to the first location of the associated data buffer, may reside in either internal or external memory.

NOTE

The Rx buffer pointer must be even, and the upper 8 bits must of the pointer must be zero for the function codes to operate correctly.

4.5.12.11 HDLC Transmit Buffer Descriptor (Tx BD)

Data is presented to the HDLC controller for transmission on an SCC channel by arranging it in buffers referenced by the channel's Tx BD table. The HDLC controller confirms transmission (or indicates error conditions) using the BDs to inform the M68000 core that the buffers have been serviced. The Tx BD is shown in Figure 4-28.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	R	X	W	I	L	TC	—	—	—	—	—	—	—	—	UN	CT
OFFSET + 2	DATA LENGTHTX BUFFER POINTER (24-bits used, upper 8 bits must be 0)															
OFFSET + 4																
OFFSET + 6																

Figure 4-28. HDLC Transmit Buffer Descriptor

sion of a block. For the receiver, the ENQ character designates the end of the block, but no CRC is expected.

Following control character reception (i.e., end of the block), the RCH bit in the BISYNC mask register should be set, re-enabling interrupts for each byte of received data.

4.5.14 DDCMP Controller

The byte-oriented digital data communications message protocol (DDCMP) was originated by DEC for use in networking products. The three classes of DDCMP frames are transparent (or maintenance) messages, data messages, and control messages (see Figure 4-34). Each class of frame starts with a standard two octet synchronization pattern and ends with a CRC. Depending upon the frame type, a separate CRC may be present for the header as well as the data portions of the frame. These CRCs use the same 16-bit generator polynomial as that used in HDLC.

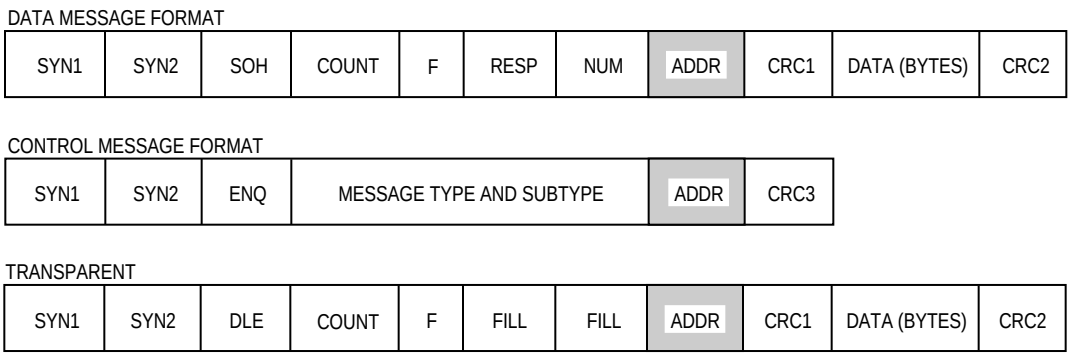


Figure 4-34. Typical DDCMP Frames

The most notable feature of the DDCMP frame is that the frame length is transmitted within the frame itself. Thus, any character pattern can be transmitted in the data field since the character count is responsible for ending the frame, not a special character. For this to work properly, the header containing the frame length must be protected, causing a need for a CRC in the frame header.

The bulk of the frame is divided into fields whose meaning depends on the frame type. Defined control characters are only used in the fixed-length frame headers (the fields between the synchronization octets and the first CRC). The following fields are one byte each: SYN1, SYN2, SOH, RESP, NUM, ADDR, ENQ, DLE, and FILL. The following fields are two bytes each: COUNT + F, CRC1, CRC2, and CRC3. The DATA field is a variable number of bytes, as defined in the COUNT field.

DDCMP communications can be either synchronous or asynchronous, with both types using the same frame format. Synchronous DDCMP frames require the physical layer to transmit the clock along with data over the link. Asynchronous DDCMP frames are composed of asynchronous UART characters, which together form the frame. The receiver and transmitter clocks are not linked; the receiver resynchronizes itself every byte using the start and stop bits of each UART character.

Table 4-10. DDCMP Specific Parameter RAM

Address	Name	Width	Description
SCC Base + 9C SCC Base + 9E SCC Base + A0 # SCC Base + A2	RCRC CRCC PCRC TCRC	Word Word Word Word	Temp Receive CRC CRC16 Constant Preset CRC16 Temp Transmit CRC
SCC Base + A4 # SCC Base + A5 # SCC Base + A6 SCC Base + A7 # SCC Base + A8 SCC Base + A9 #	DSYN1 DSOH Reserved DENQ Reserved DDLE	Byte Byte Byte Byte Byte Byte	DDCMP SYN1 Character DDCMP SOH Character DDCMP ENQ Character DDCMP DLE Character
SCC Base + AA # SCC Base + AC # SCC Base + AE # SCC Base + B0 #	CRC1EC CRC2EC NMARC DISMC	Word Word Word Word	CRC1 Error Counter CRC2 Error Counter Nonmatching Address Received Counter Discard Message Counter
SCC Base + B2 SCC Base + B4	RMLG RMLG_CNT	Word Word	Received Message Length Received Message Length Counter
SCC Base + B6 # SCC Base + B8 # SCC Base + BA # SCC Base + BC # SCC Base + BE #	DMASK DADDR1 DADDR2 DADDR3 DADDR4	Word Word Word Word Word	User-Defined Frame Address Mask User-Defined Frame Address User-Defined Frame Address User-Defined Frame Address User-Defined Frame Address

Initialized by the user (M68000 core).

4.5.14.4 DDCMP Programming Model

The M68000 core configures each SCC to operate in one of four protocols by the MODE1–MODE0 bits in the SCC mode register. If MODE1–MODE0 = 10, DDCMP operation is selected with synchronous links. For asynchronous links, MODE1–MODE0 = 01 (ASYNC) should be selected, and the DDCMP bit in the UART mode register should be set. The SYN1–SYN2 synchronization characters are programmed in the data synchronization register (DSR). See 4.5.4 SCC Data Synchronization Register (DSR) for more programming information. The DDCMP controller uses the same basic data structure as the UART, HDLC, and BISYNC controllers.

The DDCMP controller generates and checks the CRC16 message trailer. It can be preset to ones or zeros by writing to the preset CRC (PCRC) register before enabling the receiver or the transmitter. The received message length (RMLG) is the header byte count value as determined by the receiver, and the received message length counter (RMLG _ CNT) is the temporary received data downcounter.

Receive and transmit errors are reported in their respective BDs. The line status signals ($\overline{CD}$ and $\overline{CTS}$) may be read in the SCC status register and a maskable interrupt is generated upon each status change (see 4.5.2 SCC Configuration Register (SCON)).

4.5.14.5 DDCMP Command Set.

The following commands are issued to the command register:

STOP TRANSMIT Command

After a hardware or software reset and the enabling of the channel in the SCC mode register, the channel is in the transmit enable mode and starts polling the first BD in the table approximately every eight transmit clocks.

The channel STOP TRANSMIT command disables the transmission of messages on the transmit channel. If this command is received by the DDCMP controller during message

Bit 7—This bit is reserved and should be set to zero.

SMD3–SMD0—SMC Mode Support

X00X = GCI—The monitor channel is not used.
001X = GCI—The monitor channel data and the A and E control bits are internally controlled according to the monitor channel protocol.
101X = GCI—The monitor channel data and the A and E control bits are received and transmitted transparently by the IMP.
X100 = IDL—The M and A channels are in hunt-on-zero mode.
X101 = IDL—Only the M channel is in hunt-on-zero mode.
X110 = IDL—Only the A channel is in hunt-on-zero mode.
X111 = IDL—Regular operation; no channel is in hunt-on-zero mode.

LOOP—Local Loopback Mode

0 = Normal mode
1 = Local loopback mode. In GCI mode, EN1 and EN2 must also be set.

EN2—SMC2 Enable

0 = Disable SMC2
1 = Enable SMC2

EN1—SMC1 Enable

0 = Disable SMC1
1 = Enable SMC1

4.7.3 SMC Commands

The following commands issued to the CP command register (see 4.3 Command Set) are used only when GCI is selected for the serial channels physical interface.

TRANSMIT ABORT REQUEST Command

This receiver command may be issued when the IMP implements the monitor channel protocol. When issued, the IMP sends an abort request on the A bit.

TIMEOUT Command

This transmitter command may be issued when the IMP implements the monitor channel protocol. It is issued because the device is not responding or because GCI A bit errors are detected. When issued, the IMP sends an abort request on the E bit.

4.7.4 SMC Memory Structure and Buffers Descriptors

The CP uses several memory structures and memory-mapped registers to communicate with the M68000 core. All the structures detailed in the following paragraphs reside in the dual-port RAM of the IMP (see Figure 3-4). The SMC buffer descriptors allow the user to define one data byte at a time for each transmit channel and receive one data byte at a time for each receive channel.

These eight pins can be used either as NMSI1 in nonmultiplexed serial interface (NMSI) mode or as an ISDN physical layer interface in IDL, GCI, and PCM highway modes. The input buffers have Schmitt triggers.

Table 5-7 shows the functionality of each pin in NMSI, GCI, IDL, and PCM highway modes.

Table 5-7. Mode Pin Functions

Signal Name	NMSI1		GCI		IDL		PCM	
RXD1/L1RXD	I	RXD1	I	L1RXD	I	L1RXD	I	L1RXD
TXD1/L1TXD	O	TXD1	O	L1TXD	O	L1TXD	O	L1TXD
RCLK1/L1CLK	I/O	RCLK1	I	L1CLK	I	L1CLK	I	L1CLK
TCLK1/L1SY0	I/O	TCLK1	O	SDS1	O	SDS1	I	L1SY0
$\overline{\text{CD1}}$/L1SY1	I	$\overline{\text{CD1}}$	I	L1SYNC	I	L1SYNC	I	L1SY1
$\overline{\text{CTS1}}$/L1GR	I	$\overline{\text{CTS1}}$	I	L1GR	I	L1GR		
$\overline{\text{RTS1}}$/L1RQ	O	$\overline{\text{RTS1}}$	O	GCIDCL	O	L1RQ	O	$\overline{\text{RTS}}$
BRG1	O	BRG1	O	BRG1	O	BRG1	O	BRG1

NOTES:
1. In IDL and GCI mode, SDS2 is output on the PA7 pin.
2. $\overline{\text{CD1}}$ may be used as an external sync in NMSI mode.
3. $\overline{\text{RTS}}$ is the $\overline{\text{RTS1}}$, $\overline{\text{RTS2}}$, or RTS3 pin according to which SCCs are connected to the PCM highway.

RXD1/L1RXD—Receive Data/Layer-1 Receive Data

This input is used as the NMSI1 receive data in NMSI mode and as the receive data input in IDL, GCI, and PCM modes.

TXD1/L1TXD—Transmit Data/Layer-1 Transmit Data

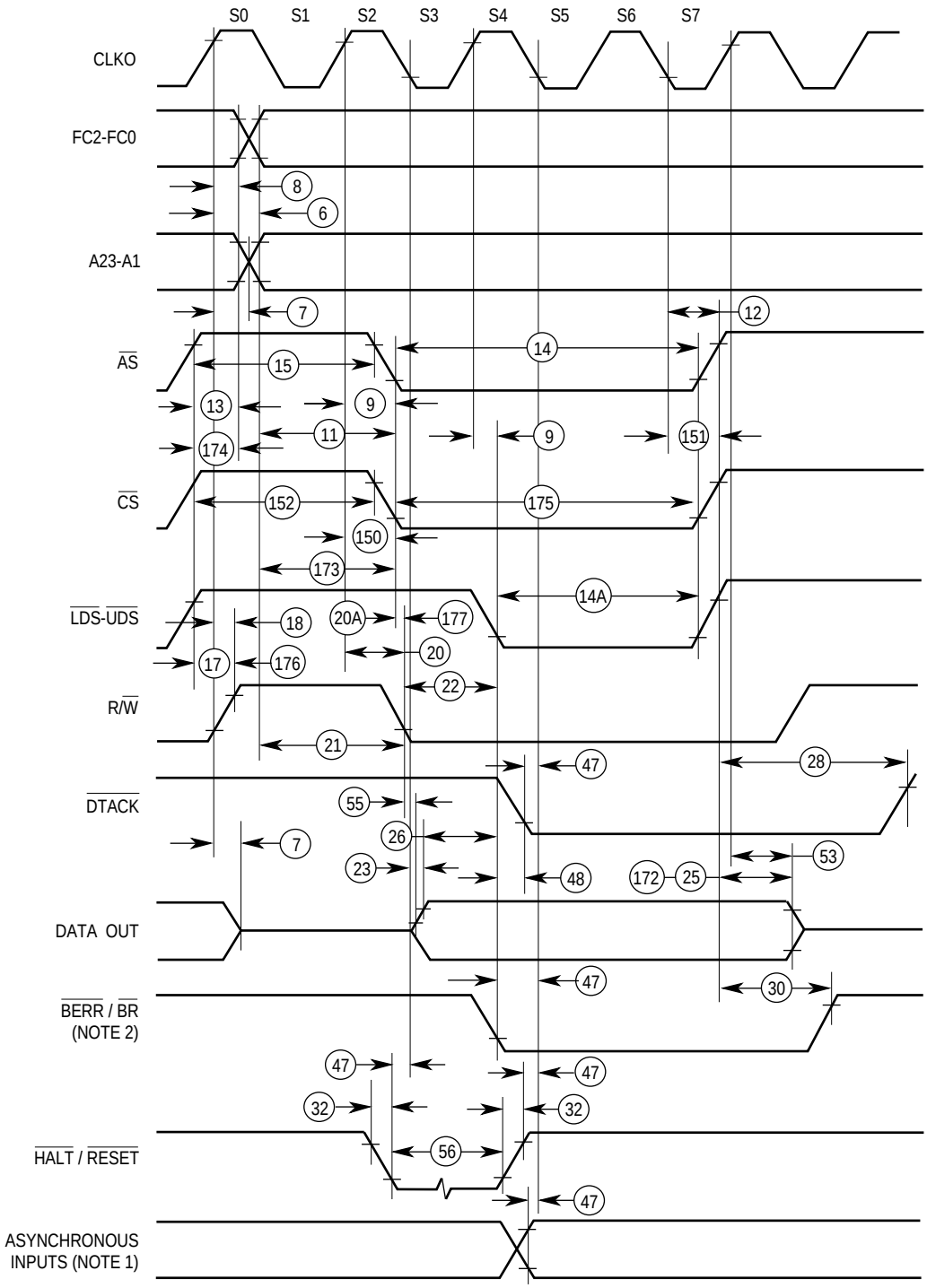
This output is used as NMSI1 transmit data in NMSI mode and as the transmit data output in IDL, GCI, and PCM modes. TXD1 may be configured as an open-drain output in NMSI mode. L1TXD in IDL and PCM mode is a three-state output. In GCI mode, it is an open-drain output.

RCLK1/L1CLK—Receive Clock/Layer-1 Clock

This pin is used as an NMSI1 bidirectional receive clock in NMSI mode or as an input clock in IDL, GCI, and PCM modes. In NMSI mode, this signal is an input when SCC1 is working with an external clock and is an output when SCC1 is working with its baud rate generator. The RCLK1 output can be three-stated by setting bit 12 in the CKCR register (see 3.9 Clock Control Register).

TCLK1/L1SY0/SDS1—Transmit Clock/PCM Sync/Serial Data Strobe 1

This pin is used as an NMSI1 bidirectional transmit clock in NMSI mode, as a sync signal in PCM mode, or as the SDS1 output in IDL/GCI modes. In NMSI mode, this signal is an input when SCC1 is working with an external clock and is an output when SCC1 is working with its baud rate generator. The TCLK1 output can be three-stated by setting bit 13 in the CKCR register (see 3.9 Clock Control Register).



NOTES:
1. Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall is linear between between 0.8 volt and 2.0 volts.
2. Because of loading variations, R/W may be valid after AS even though both are initiated by the rising edge of S2 (specification #20A)
3. Each wait state is a full clock cycle inserted between S4 and S5.

Figure 6-3. Write Cycle Timing Diagram

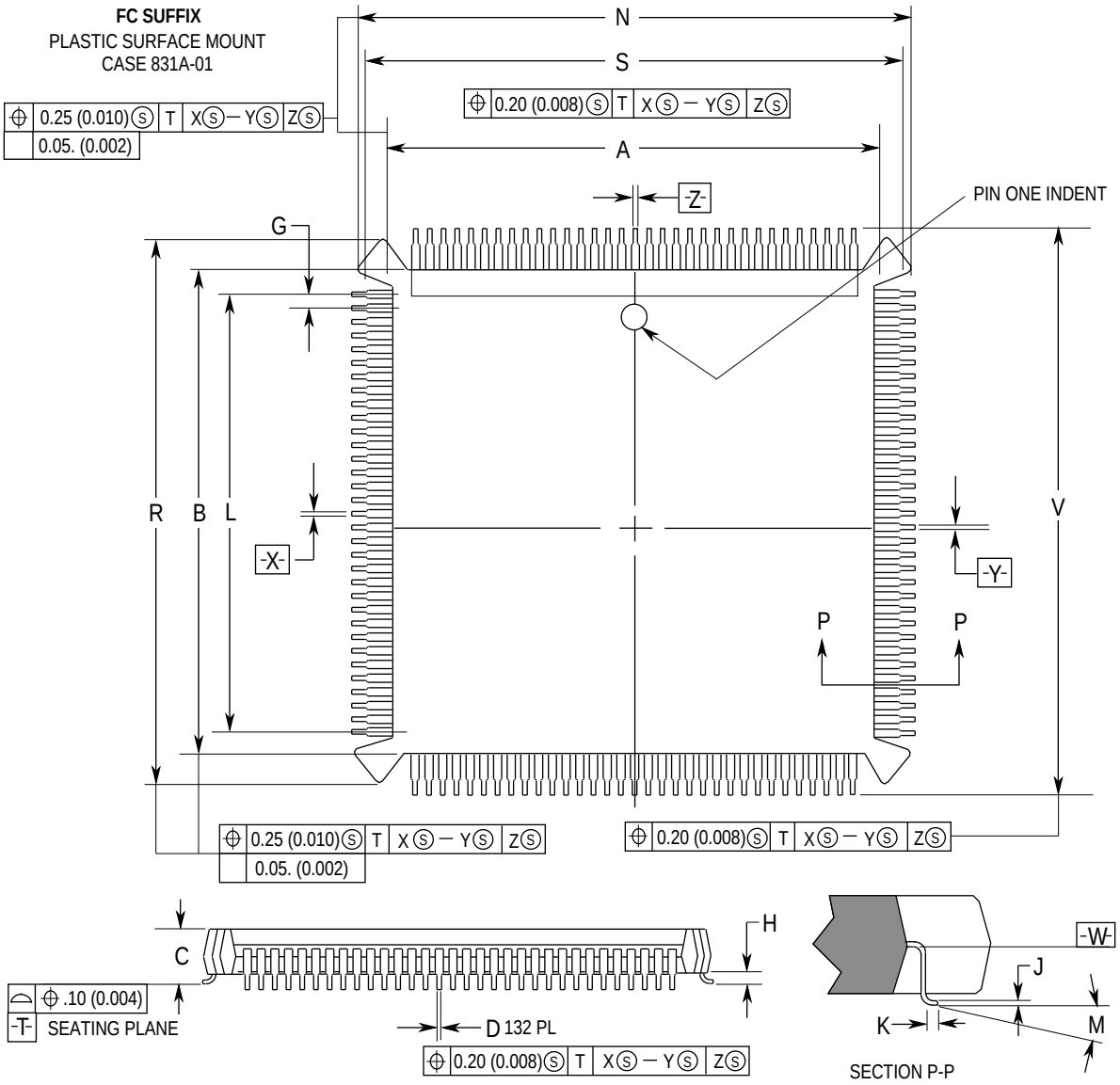
6.10 AC ELECTRICAL SPECIFICATIONS—EXTERNAL MASTER INTERNAL ASYNCHRONOUS READ/WRITE CYCLES

(see Figure 6-8 and Figure 6-9)

			16.67 MHz		20 MHz		25 MHz		
Num.	**Characteristic**	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
100	R/$\overline{W}$ Valid to $\overline{DS}$ Low	t_{RWVDSL}	0	—	0	—	0	—	ns
101	$\overline{DS}$ Low to Data-In Valid	t_{DSLDIV}	—	30	—	25	—	20	ns
102	$\overline{DTACK}$ Low to Data-In Hold Time	t_{DKLDH}	0	—	0	—	0	—	ns
103	$\overline{AS}$ Valid to $\overline{DS}$ Low	t_{ASVDSL}	0	—	0	—	0	—	ns
104	$\overline{DTACK}$ Low to $\overline{AS}$, $\overline{DS}$ High	t_{DKLDSH}	0	—	0	—	0	—	ns
105	$\overline{DS}$ High to $\overline{DTACK}$ High	t_{DSHDKH}	—	45	—	40	—	30	ns
106	$\overline{DS}$ Inactive to $\overline{AS}$ Inactive	t_{DSIASI}	0	—	0	—	0	—	ns
107	$\overline{DS}$ High to R/$\overline{W}$ High	t_{DSHRWH}	0	—	0	—	0	—	ns
108	$\overline{DS}$ High to Data High Impedance	t_{DSHDZ}	—	45	—	40	—	30	ns
108A	$\overline{DS}$ High to Data-Out Hold Time (see Note)	t_{DSHDH}	0	—	0	—	0	—	ns
109A	Data Out Valid to $\overline{DTACK}$ Low	t_{DOVDKL}	15	—	15	—	10	—	ns

NOTE: If $\overline{AS}$ is negated before $\overline{DS}$, the data bus could be three-stated (spec 126) before $\overline{DS}$ is negated.

7.2.2 Plastic Surface Mount (PQFP)



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	24.06	24.20	0.947	0.953
B	24.06	24.20	0.947	0.953
C	4.07	4.57	0.160	0.180
D	0.21	0.30	0.008	0.012
G	0.64 BSC		0.025 BSC	
H	0.51	1.01	0.020	0.040
J	0.16	0.20	0.006	0.008
K	0.51	0.76	0.020	0.030
M	0°	8°	0°	8°
N	27.88	28.01	1.097	1.103
R	27.88	28.01	1.097	1.103
S	27.31	27.55	1.075	1.085
V	27.31	27.55	1.075	1.085

NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCHES
3. DIM A, B, N, AND R DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION FOR DIMENSIONS A AND B IS 0.25 (0.010), FOR DIMENSIONS N AND R IS 0.18 (0.007).
4. DATUM PLANE -W- IS LOCATED AT THE UNDERSIDE OF LEADS WHERE LEADS EXIT PACKAGE BODY.
5. DATUMS X-Y AND Z TO BE DETERMINED WHERE CENTER LEADS EXIT PACKAGE BODY AT DATUM -W-.
6. DIM S AND V TO BE DETERMINED AT SEATING PLANE, DATUM -T-.
7. DIM A, B, N AND R TO BE DETERMINED AT DATUM PLANE -W-.

```
          BSET.B    #$7,(A2)          ;Set Ready bit of Tx BD
* The Tx BD sand data status is not checked since the only one is CTS lost,
* which is not applicable, since CTS is ignored in this application.
* The following updates A2 to point to the next Tx BD
          BTST.B    #$05,(A2)         ;test Wrap bit
          BNE.B     REINIT2           ;If set, reinit A2 to 700640
          ADDA.W    #$08,A2           ;else inc A2 by 8 to next Tx BD
          BRA.B     CONT              ;Jump to Continue on
REINIT2 MOVEA.L     #$700640,A2       ;Reinitialize A2
* Determine what the next byte to "echo" will be and then go to OUTERLOOP
 CONT     ADDO.W    #$1,A1            ;Increment A1 to next byte to send
          CMPA.L    #$30004,A1        ;Is A1 = 30004? ***
          BEQ.B     NEWA1             ;If so, go to NEWA1
          BRA.B     OUTLOOP           ;Jump back to outerloop and wait
NEWA1     SUBO.W    #$02.A1           ;Set A1 back to 30002 ?***
          BRA.B     OUTLOOP           ;Jump back to outerloop and wait
* The two lines with ***  above are dependent on the number of Rx BDs used.
* If the number is increased, these values should be increased by the same
* amount. These are the only lines dependent on the Rx BD or Tx BD setup.
*************************************************************************
*SCC3 Interrupt Routine
          ORG       $30500
          CLR.L     D1                ;clear D1
          MOVE.B    SCCE3,D1          ;Move SCCE3 status to D1
          MOVE.B    #$15,SCCE3        ;Clear only BRK. BSY and RX in SCCE3.

          BTST.B    #$2.D1            ;Is BSY set?
          BNE.B     BUSY              ;Jump to BUSY handler if set



* Test Break:
BRKTEST   BTST.B    #$4,D1            ;Is BRK set?
          BNE.B     BREAK             ;Jump to BREAK handler if set
*Test Receive:
RECTEST   BTST.B    #$0,D1            ;Is RX set?
          BNE.B     RECEIVE           ;Jump to RECEIVE handler if set
          JMP       ALMDONE           ;Jump to About Done (impossible)
* Busy handler:
BUSY      ADDQ.B    #1,D5             ;Inc Busy counter (no receive buffers)
          BSET.B    #$F,(A0)          ;set Empty bit of current Rx BD
          JMP       BRKTEST           ;Jump to test for BREAK

*Break handler:
BREAK     NOP                         ;This code ignores received breaks
* The UART BRKEC will record the number of breaks received
          JMP       RECTEST           ;Jump to test for RECEIVE
*Receive handler:
RECEIVE   ADDQ.W    #1,D3             ;Increment number of chars received
          ADDQ.B    #1,D6             ;D6 inc by 1 (character ready to send)

          ADDQ.W    #1,A0             ;Inc A0 to point to Rx BD byte status
          CMPI.B    #$0,(A0)          ;Does status = 00?
          BNE.B     BSTAT             ;Jump to Bad Status it not 00
INCPTR    SUBQ.W    #1,A0             ;Dec A0 to point to beginning of Rx BD
          ANDI.W    #$FF00,(A0)       ;Clear out Rx BD status
```

SCCs on the MC68302 can do this very efficiently because of their sophisticated DMA capability, and very little MC68000 core intervention is required.

Third, some applications require the switching of data without interfering with the protocol encoding itself. For instance, in a multiplexer, data from a high-speed time-multiplexed serial stream is multiplexed into multiple low-speed data streams. In this case, the idea is to switch the data path but not alter the protocol encoded on that data path.

Finally, some applications require a special protocol that does not fall under the category of HDLC, UART, etc. In some instances, transparent mode can be used; however, care should be taken to understand the capabilities of transparent mode before trying this. The most important issue is how this new protocol recognizes its frames on the receiving end. Transparent mode on the MC68302 was designed to work well receiving continuous streams of data (no gaps in the data exist over time). This is different from receiving transparent frames; although there is some support for this, it is limited on the MC68302.

D.8.3 Physical Interface to Accompany Transparent Mode

Before discussing the details of transparent mode timing, we need to chose the physical interface to go with the transparent mode. The timings associated with transparent mode differ based on the physical interface chosen.

The MC68302 supports the following four physical interfaces:

- Nonmultiplexed Serial Interface—NMSI
- Pulse Code Modulation Highway—PCM
- Interchip Digital Link—IDL
- General Circuit Interface—GCI

You will probably chose either an NMSI or PCM highway interface, unless you are designing an ISDN-based system. If you are designing an ISDN-based system, you will probably choose either an IDL or GCI interface. The following paragraphs discuss all the interfaces, but special attention is given to the NMSI.

NOTE

The following discussion assumes some knowledge of the interfaces. For more applications information on these interfaces, refer to 4.4 Serial Channels Physical Interface**.**

The most commonly used physical interface on the MC68302 is the nonmultiplexed serial interface (NMSI). The NMSI consists of seven basic modem (RS-232) signals: $\overline{\text{TXD}}$, $\overline{\text{TCLK}}$, $\overline{\text{RXD}}$, $\overline{\text{RCLK}}$, $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{CD}}$. Each of the three SCCs can have its own set of these signals, as shown in Figure D-21.

BIT 15—Reserved for future use; should be written with zero.

EXSYN—External Sync

When set, the SCC receiver uses the L1SY1/$\overline{CD1}$, $\overline{CD2}$, or $\overline{CD3}$ pins to synchronize the receiver and transmitter to the beginning of a transparent frame.

NTSYN—No Transmit SYNC

This bit must be set for the SCC to operate in the transparent mode.

REVD—Reverse Data

When this bit is set, the receiver and transmitter will reverse the character bit order, transmitting the most significant bit first.

Bits 11-6—Reserved for future use; should be written with zero.

DIAG1, DIAG0—Diagnostic Mode

00 = Normal operation.
01 = Loopback mode.
10 = Automatic echo.
11 = Software operation.

ENR—Enable Receiver

0 = Receiver is disabled.
1 = Receiver is enabled

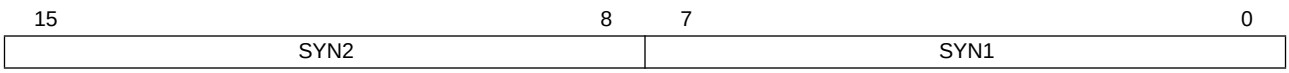
ENT—Enable Transmitter

0 = Transmitter is disabled.
1 = Transmitter is enabled.

MODE1, MODE0—Channel Mode

00 = HDLC.
01 = Asynchronous (UART and DDCMP).
10 = Synchronous DDCMP and V.110.
11 = BISYNC and Promiscuous (Transparent).

E.3.1.2.3 SCC Data Synchronization Register (DSR). This 16-bit register is located at offset $886 (SCC1), $896 (SCC2), and $8A6 (SCC3). The DSR specifies the pattern used for the receive frame synchronization procedure if the EXSYN bit is cleared. For transparent, the DSR may be set to any desired pattern. The DSR value after reset is $7E7E.



E.3.1.2.4 Transparent Event Register (SCCE). This 8-bit register is located at offset $888 (SCC1), $898 (SCC2), and $8A8 (SCC3) on D15-D8 of a 16-bit data bus. The SCCE is used to report events recognized by the transparent channel. Bits must be cleared by the user to avoid missing interrupt events. Bits are cleared by writing ones to the corresponding bit positions

C

D