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	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	132-BQFP Bumpered
Supplier Device Package	132-PQFP (46x46)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68302eh16cb1

on bytes, words, or long words, and most instructions can use any of the 14 addressing modes.

Combining instruction types, data types, and addressing modes provides over 1000 useful instructions. These instructions include signed and unsigned multiply and divide, quick arithmetic operations, BCD arithmetic, and expanded operations (through traps).

Table 2-1. M68000 Data Addressing Modes

Mode	Generation
Register Direct Addressing Data Register Direct Address Register Direct	 EA = Dn EA = An
Absolute Data Addressing Absolute Short Absolute Long	 EA = (Next Word) EA = (Next Two Words)
Program Counter Relative Addressing Relative with Offset Relative with Index and Offset	 EA = (PC) + d_{16} EA = (PC) + Xn + d8
Register Indirect Addressing Register Indirect Postincrement Register Indirect Predecrement Register Indirect Register Indirect with Offset Indexed Register Indirect with Offset	 EA = (An) EA = (An), An ⇐ An + N EA = ← An - N, EA = (An) EA = (An) + d_{16} EA = (An) + (Xn) + d_8
Immediate Data Addressing Immediate Quick Immediate	 DATA = Next Word(s) Inherent Data
Implied Addressing Implied Register	 EA = SR, USP, SSP, PC

NOTES:

EA	=	Effective Address
An	=	Address Register
Dn	=	Data Register
Xn	=	Address or Data Register Used as an Index Register
SR	=	Status Register
PC	=	Program Counter
( )	=	Contents of
d_8	=	8-Bit Offset (Displacement)
d16	=	16-Bit Offset (Displacement)
N	=	1 for byte, 2 for word, and 4 for long word. If An is the stack pointer and the operand size is byte, N = 2 to keep the stack pointer on a word boundary.
←	=	Replaces

Once the processor is in the user state and executing instructions, only exception processing can change the privilege state. During exception processing, the current state of the S bit in the SR is saved and the S bit is asserted, putting the processor in the supervisor state. Therefore, when instruction execution resumes at the address specified to process the exception, the processor is in the supervisor privilege state. The transition from the supervisor to user state can be accomplished by any of four instructions: return from exception (RTE), move to status register (MOVE to SR), AND immediate to status register (ANDI to SR), and exclusive OR immediate to status register (EORI to SR).

2.4 EXCEPTION PROCESSING

The processing of an exception occurs in four steps, with variations for different exception causes. During the first step, a temporary copy of the SR is made, and the SR is set for exception processing. During the second step, the exception vector is determined; during the third step, the current processor context is saved. During the fourth step, a new context is obtained, and the processor switches to instruction processing.

2.4.1 Exception Vectors

Exception vectors are memory locations from which the processor fetches the address of a routine to handle that exception. All exception vectors are two words long except for the reset vector, which is four words. All exception vectors lie in the supervisor data space except for the reset vector, which is in the supervisor program space. A vector number is an 8-bit number which, when multiplied by four, gives the offset of the exception vector. Vector numbers are generated internally or externally, depending on the cause of the exception. In the case of interrupts, during the interrupt acknowledge bus cycle, a peripheral may provide an 8-bit vector number to the processor on data bus lines D7–D0. Alternatively, the peripheral may assert autovector ($\overline{\text{AVEC}}$) instead of data transfer acknowledge ($\overline{\text{DTACK}}$) to request an autovector for that priority level of interrupt. The exception vector assignments for the M68000 processor are shown in Table 2-5.

Table 2-5. M68000 Exception Vector Assignment

Vector Number	Decimal	Address Hex	Space	Assignment
0	0	000	SP	Reset: Initial SSP[2]
1	4	004	SP	Reset: Initial PC[2]
2	8	008	SD	Bus Error
3	12	00C	SD	Address Error
4	16	010	SD	Illegal Instruction
5	20	014	SD	Zero Divide
6	24	018	SD	CHK Instruction
7	28	01C	SD	TRAPV Instruction
8	32	020	SD	Privilege Violation
9	36	024	SD	Trace
10	40	028	SD	Line 1010 Emulator

3.1.6 DMA Bus Arbitration

The IDMA controller uses the M68000 bus arbitration protocol to request bus mastership before entering the DMA mode of operation. The six SDMA channels have priority over the IDMA and can transfer data between any two IDMA bus cycles with $\overline{\text{BGACK}}$ remaining continuously low. Once the processor has initialized and started a DMA channel, an operand transfer request is made pending by either an external device or by using an internal request.

When the IDMA channel has an operand transfer request pending and $\overline{\text{BCLR}}$ is not asserted, the IDMA will request bus mastership from the internal bus arbiter using the internal signal IDBR (see Figure 3-12). The arbiter will assert the internal M68000 core bus request ($\overline{\text{CBR}}$) signal and will monitor the core bus grant ($\overline{\text{CBG}}$) and external $\overline{\text{BR}}$ to determine when it may grant the IDMA mastership. The IDMA will monitor the address strobe ($\overline{\text{AS}}$), $\overline{\text{HALT}}$, bus error ($\overline{\text{BERR}}$), and bus grant acknowledge ($\overline{\text{BGACK}}$) signals. These signals must be negated to indicate that the previous bus cycle has completed and the previous bus master has released the bus. When these conditions are met, the IDMA only asserts $\overline{\text{BGACK}}$ to indicate that it has taken control of the bus. When all operand transfers have occurred, the IDMA will release control of the bus by negating $\overline{\text{BGACK}}$.

Internally generated IDMA requests are affected by a mechanism supported to reduce the M68000 core interrupt latency and external bus master arbitration latency (see 3.8.5 Bus Arbitration Logic). The IDMA is forced to relinquish the bus when an external bus master requests the bus ($\overline{\text{BR}}$ is asserted) or when the M68000 core has an unmasked pending interrupt request. In these cases, the on-chip arbiter sends an internal bus-clear signal to the IDMA. In response, any operand transfer in progress will be fully completed (up to four bus cycles depending on the configuration), and bus ownership will be released.

When the IDMA regains the bus, it will continue transferring where it left off. If the core caused the bus to be relinquished, no further IDMA bus cycles will be started until IPA in the SCR is cleared. If the cause was an external request, no further IDMA bus cycles will be started while $\overline{\text{BR}}$ remains asserted. When $\overline{\text{BR}}$ is externally negated, if a transfer request is pending and IPA is cleared, the IDMA will arbitrate for the bus and continue normal operation.

3.1.7 Bus Exceptions

In any computer system, the possibility always exists that an error will occur during a bus cycle due to a hardware failure, random noise, or an improper access. When an asynchronous bus structure, such as that supported by the M68000 is used, it is easy to make provisions allowing a bus master to detect and respond to errors during a bus cycle. The IDMA recognizes the same bus exceptions as the M68000 core: reset, bus error, halt, and retry.

NOTE

These exceptions also apply to the SDMA channels except that the bus error reporting method is different. See 4.5.8.4 Bus Error on SDMA Access for further details.

When working in the MC68008 mode (BUSW is low), writing the high byte of TRR1 and TRR2 will disable the timer's compare logic until the low byte is written.

TRR1 and TRR2 are set to all ones by reset. The reference value is not “reached” until TCN increments to equal TRR.

3.5.2.3 Timer Capture Registers (TCR1, TCR2)

Each TCR is a 16-bit register used to latch the value of the counter during a capture operation when an edge occurs on the respective TIN1 or TIN2 pin. TCR1 and TCR2 appear as memory-mapped read-only registers to the user.

When working in the MC68008 mode (BUSW is low), reading the high byte of TCR1 and TCR2 will disable the timer's capture logic until the low byte is read.

TCR1 and TCR2 are cleared at reset.

3.5.2.4 Timer Counter (TCN1, TCN2)

TCN1 and TCN2 are 16-bit up-counters. Each is memory-mapped and can be read and written by the user. A read cycle to TCN1 and TCN2 yields the current value of the timer and does not affect the counting operation.

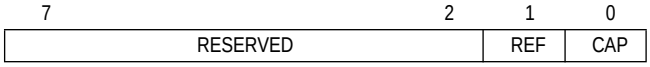
When working in the MC68008 mode (BUSW is low), reading the high byte of TCN1 and TCN2 will latch the low byte into a temporary register; a subsequent read cycle on the low byte yields the value of the temporary register.

A write cycle to TCN1 and TCN2 causes both the counter register and the corresponding prescaler to be reset to zero. In MC68008 mode (BUSW is low), a write cycle to either the high or low byte of the TCN will reset the counter register and the corresponding prescaler to zero.

3.5.2.5 Timer Event Registers (TER1, TER2)

Each TER is an 8-bit register used to report events recognized by any of the timers. On recognition of an event, the timer will set the appropriate bit in the TER, regardless of the corresponding interrupt enable bits (ORI and CE) in the TMR. TER1 and TER2, which appear to the user as memory-mapped registers, may be read at any time.

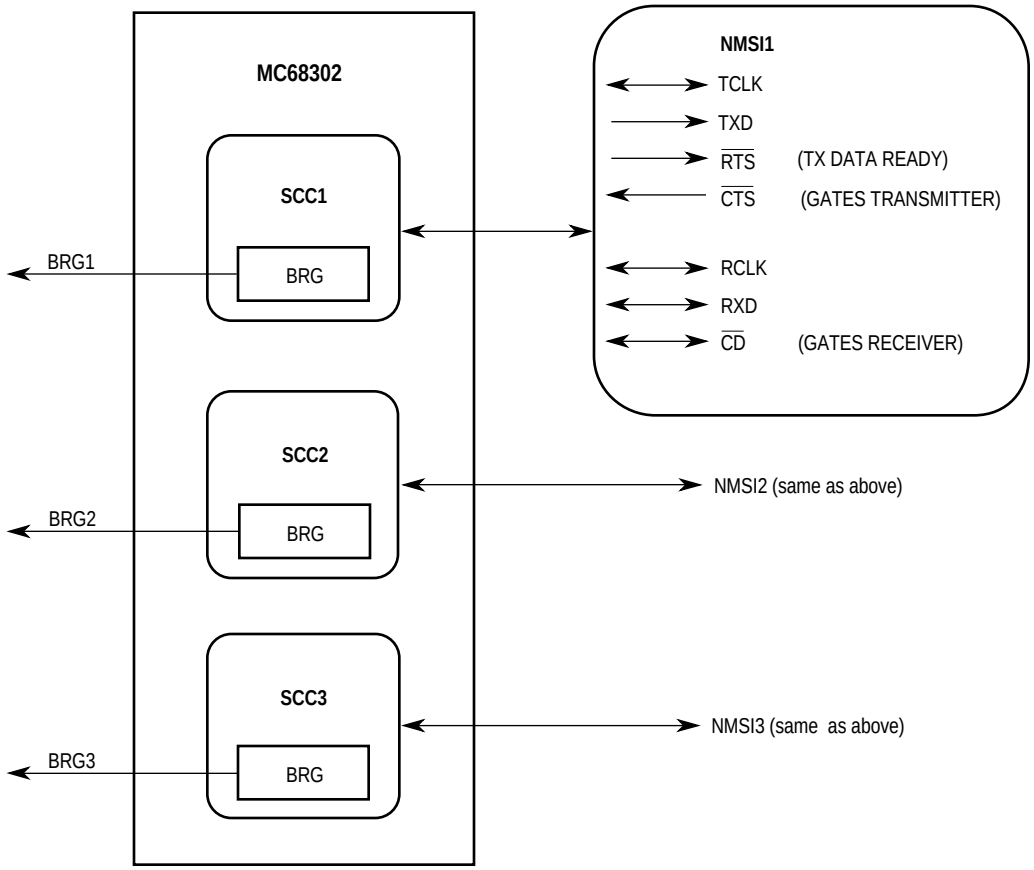
A bit is cleared by writing a one to that bit (writing a zero does not affect a bit's value). More than one bit may be cleared at a time. Both bits must be cleared before the timer will negate the INRQ to the interrupt controller. This register is cleared at reset.



CAP—Capture Event

The counter value has been latched into the TCR. The CE bits in the TMR are used to enable the interrupt request caused by this event.

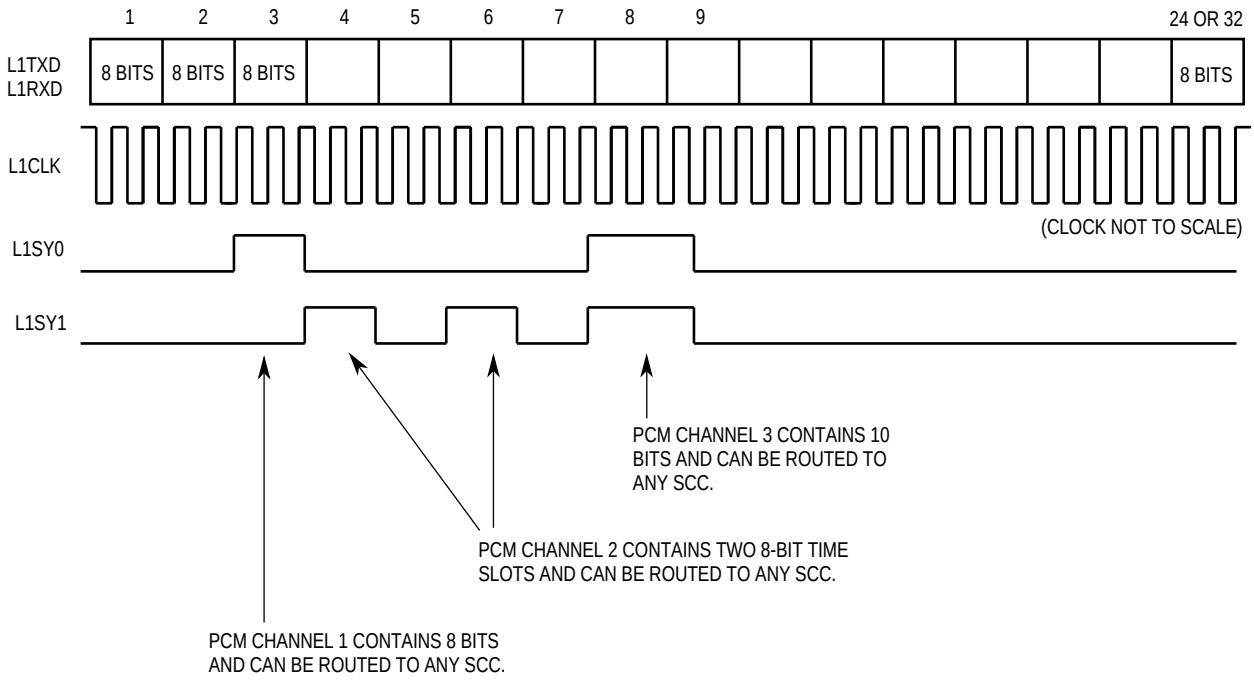
$\overline{RTS}$, $\overline{CTS}$, and $\overline{CD}$ are multiplexed with the SCP. See 4.6 Serial Communication Port (SCP) for more details on the SCP.



NMSI — Nonmultiplexed serial interface (also called the modem I/F).

Figure 4-3. NMSI Physical Interface

The other three physical interfaces, PCM, IDL, and GCI here are called multiplexed interfaces since they allow data from one, two, or all three SCCs to be time multiplexed together on the same pins. Note that if a multiplexed mode is chosen, the first SCC to use that mode must be SCC1 since the three multiplexed modes share pins with SCC1. After choosing a multiplexed mode, the user may decide whether SCC2 and SCC3 should be part of the multiplexed interface or whether they should have their own set of NMSI pins (see Figure 4-4). If SCC2 or SCC3 is part of the multiplexed interface, all NMSI2 and NMSI3 pins may be used for other functions such as parallel I/O. If a multiplexed mode is chosen, the baud rate generator clock is output on the BRG or TCLK pin, depending on whether the NMSI mode or multiplexed mode, respectively, was chosen for that SCC.



NOTE: Whenever the syncs are active, data from that SCC is transmitted and received using L1CLK edges.

Figure 4-10. PCM Channel Assignment on a T1/CEPT Line

4.4.4 Nonmultiplexed Serial Interface (NMSI)

The IMP supports the NMSI with modem signals. In this case, the serial interface connects the seven serial lines of the NMSI1/ISDN interface (RXD1, TXD1, RCLK1, TCLK1, CD1, CTS1, and RTS1) directly to the SCC1 controller. NMSI pins associated with SCC2 and SCC3 can be used as desired or left as general-purpose I/O port pins. See 3.3 Parallel I/O Ports for an example. $\overline{\text{RTS}}$ is an output of the transmitter, while $\overline{\text{CTS}}$ and $\overline{\text{CD}}$ are inputs to the transmitter and receiver, respectively. See 4.5.3 SCC Mode Register (SCM) and 4.4 Serial Channels Physical Interface for additional information.

$\overline{\text{CTS}}$ and $\overline{\text{CD}}$ may be programmed to control transmission and reception automatically or to just generate interrupts.

4.4.5 Serial Interface Registers

There are two serial interface registers: SIMODE and SIMASK. The SIMODE register is a 16-bit register used to define the serial interface operation modes. The SIMASK register is a 16-bit register used to determine which bits are active in the B1 and B2 channels of ISDN.

4.4.5.1 Serial Interface Mode Register (SIMODE)

If the IDL or GCI mode is used, this register allows the user to support any or all of the ISDN channels independently. Any extra SCC channel can then be used for other purposes in

The purpose of the control characters table is to enable automatic recognition (by the BISYNC controller) of the end of the current block. See 4.5.13.14 Programming the BISYNC Controllers for more information. Since the BISYNC controller imposes no restrictions on the format of the BISYNC blocks, user software must respond to the received characters and inform the BISYNC controller of mode changes and certain protocol events (e.g., resetting the BCS). However, correct use of the control characters table allows the remainder of the block to be received without interrupting the user software.

Up to eight control characters may be defined. These characters inform the BISYNC controller that the end of the current block has been reached and whether a BCS is expected following this character. For example, the end of text (ETX) character implies both an end of block (ETB) and a BCS should be received. An enquiry (ENQ) character designates end of block without a subsequent BCS. All the control characters are written into the data buffer.

The BISYNC controller uses a table of 16-bit entries to support control character recognition. Each entry consists of the control character, an end-of-table bit, a BCS expected bit, and a hunt mode bit. The control characters table is shown in Figure 4-31. To disable the entire control characters table, write $8000 to the first table entry.

	15	14	13	12	11	10	9	8	7 … 0
OFFSET + 0	E	B	H						CHARACTER1
OFFSET +2	E	B	H						CHARACTER2
OFFSET + 4	E	B	H						CHARACTER3
•									
•									
•									
OFFSET + E	E	B	H						CHARACTER8

Figure 4-31. BISYNC Control Characters Table

CHARACTER8–CHARACTER1—Control Character Value

These fields define control characters.

NOTE

When using 7-bit characters with parity, the parity bit should be included in the control character value.

E—End of Table

0 = This entry is valid. The lower eight bits will be checked against the incoming character.

1 = The entry is not valid. No valid entries exist beyond this entry.

CR—BCS Error

BCS error (CR) is updated every time a byte is written into the buffer. The CR bit includes the calculation for the current byte. By clearing the RBCS bit in the BISYNC mode register within eight serial clocks, the user can exclude the current character from the message BCS calculation. The data length field may be read to determine the current character's position.

OV—Overrun

A receiver overrun occurred during message reception.

CD—Carrier Detect Lost

The carrier detect signal was negated during message reception.

Data Length

The data length is the number of octets that the CP has written into this BD's data buffer, including the BCS (if selected). In BISYNC mode, the data length should initially be set to zero by the user and is incremented each time a received character is written to the data buffer.

NOTE

The actual buffer size should be greater than or equal to the MR-BLR.

Rx Buffer Pointer

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

NOTE

For correct operation of the function codes, the upper 8 bits of the pointer must be initialized to zero.

4.5.13.11 BISYNC Transmit Buffer Descriptor (Tx BD).

Data is presented to the CP for transmission on an SCC channel by arranging it in buffers referenced by the channel's Tx BD table. The CP confirms transmission (or indicates error conditions) using the BDs to inform the processor that the buffers have been serviced. The Tx BD is shown in Figure 4-33.

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

Figure 4-33. BISYNC Transmit Buffer Descriptor

The first word of the Tx BD contains status and control bits. These bits are prepared by the user before transmission and are set by the CP after the buffer has been transmitted.

R—Ready

0 = This buffer is not currently ready for transmission. The user is free to manipulate this BD (or its associated buffer). The CP clears this bit after the buffer has been fully transmitted or after an error condition has been encountered.
1 = The data buffer has been prepared for transmission by the user (but not yet transmitted). No fields of this BD may be written by the user once this bit is set.

X—External Buffer

0 = The buffer associated with this BD is in internal dual-port RAM.
1 = The buffer associated with this BD is in external memory.

W—Wrap (Final BD in Table)

0 = This is not the last BD in the Tx BD table.
1 = This is the last BD in the Tx BD table. After this buffer has been used, the CP will transmit data from the first BD in the table.

NOTE

The user is required to set the wrap bit in one of the first eight BDs; otherwise, errant behavior may occur.

I—Interrupt

0= No interrupt is generated after this buffer has been serviced.
1= Either TX or TXE in the BISYNC event register will be set when this buffer has been serviced by the CP, which can cause an interrupt.

L—Last in Message

0 = The last character in the buffer is not the last character in the current block.
1 = The last character in the buffer is the last character in the current block. The transmitter will enter (remain in) normal mode after sending the last character in the buffer and the BCS (if enabled).

TB—Transmit BCS

This bit is valid only when the L bit is set.

0 = Transmit the SYN1–SYN2 sequence or IDLE (according to the SYNF bit in the BISYNC mode register) after the last character in the buffer.
1 = Transmit the BCS sequence after the last character. The BISYNC controller will also reset the BCS generator after transmitting the BCS sequence.

B—BCS Enable

0 = Buffer consists of characters to be excluded from the BCS accumulation.
1 = Buffer consists of characters to be included in the BCS accumulation.

BR—BCS Reset

0 = The BCS accumulation is not reset.
1 = The transmitter BCS accumulation is reset (used for STX or SOH) before sending the data buffer.

5.12 TYPICAL SERIAL INTERFACE PIN CONFIGURATIONS

Table 5-4 shows typical configurations of the physical layer interface pins for an ISDN environment. Table 5-6 shows potential configurations of the physical layer interface pins for a non-ISDN environment. The IDMA, IACK, and timer pins can be used in all applications either as dedicated functions or as PIO pins.

Table 5-5. Typical ISDN Configurations

Pins	Connected To	Used As
NMSI1 or ISDN I/F	SCC1 and SCC3	SCC1 Used as ISDN D-ch SCC3 Used as ISDN B2-ch
NMSI2	SCC2	SCC2 is Connected to Terminal
NMSI3	PA12–PA8 SCP	PIO (Extra Modem Signals and SCP Select Signals) Status/Control Exchange

NOTES:
1. ISDN environment with SCP port for status/control exchange and with existing terminal (for rate adaption).
2. D-ch is used for signaling.
3. B1-ch is used for voice (external CODEC required).
4. B2-ch is used for data transfer.

Table 5-6. Typical Generic Configurations

Pins	Connected To	Used As
NMSI1 or ISDN I/F	SCC1	Terminal with Modem
NMSI2	SCC2	Terminal with Modem
NMSI3 (5)	SCC3	Terminal without Modem
NMSI3 (3)	SCP	Status/Control Exchange

NOTE: Generic environment with three SCC ports (any protocol) and the SCP port. SCC3 does not use modem control signals.

5.13 NMSI1 OR ISDN INTERFACE PINS

The NMSI1 or ISDN interface pins are shown in Figure 5-10.

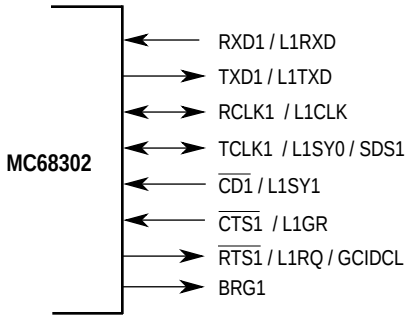


Figure 5-10. NMSI1 or ISDN Interface Pins

SECTION 6
ELECTRICAL CHARACTERISTICS

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

6.1 MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	- 0.3 to + 7.0	V
Input Voltage	V_{in}	- 0.3 to + 7.0	V
Operating Temperature Range MC68302 MC68302C	T_A	 0 to 70 - 40 to 85	°C
Storage Temperature Range	T_{stg}	- 55 to + 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to his high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD})

6.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance for PGA	θ_{JA}	25	°C/W
	θ_{JC}	2	°C/W
Thermal Resistance for CQFP	θ_{JA}	40	°C/W
	θ_{JC}	15	°C/W
Thermal Resistance for PQFP	θ_{JA}	42	°C/W
	θ_{JC}	20	°C/W

$T_J = T_A + (P_D \cdot {}_A)$
$P_D = (V_{DD} \cdot I_{DD}) + P_{I/O}$

where:
$P_{I/O}$ is the power dissipation on pins.
For T_A = 70°C and $P_{I/O}$ + 0 W, 16.67 MHz, 5.5 V, and CQFP package, the worst case value of T_J is:
T_J = 70°C + (5.5 V · 30 mA · 40°C/W) = 98.65C

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.

Signals come from the MC68302 only.

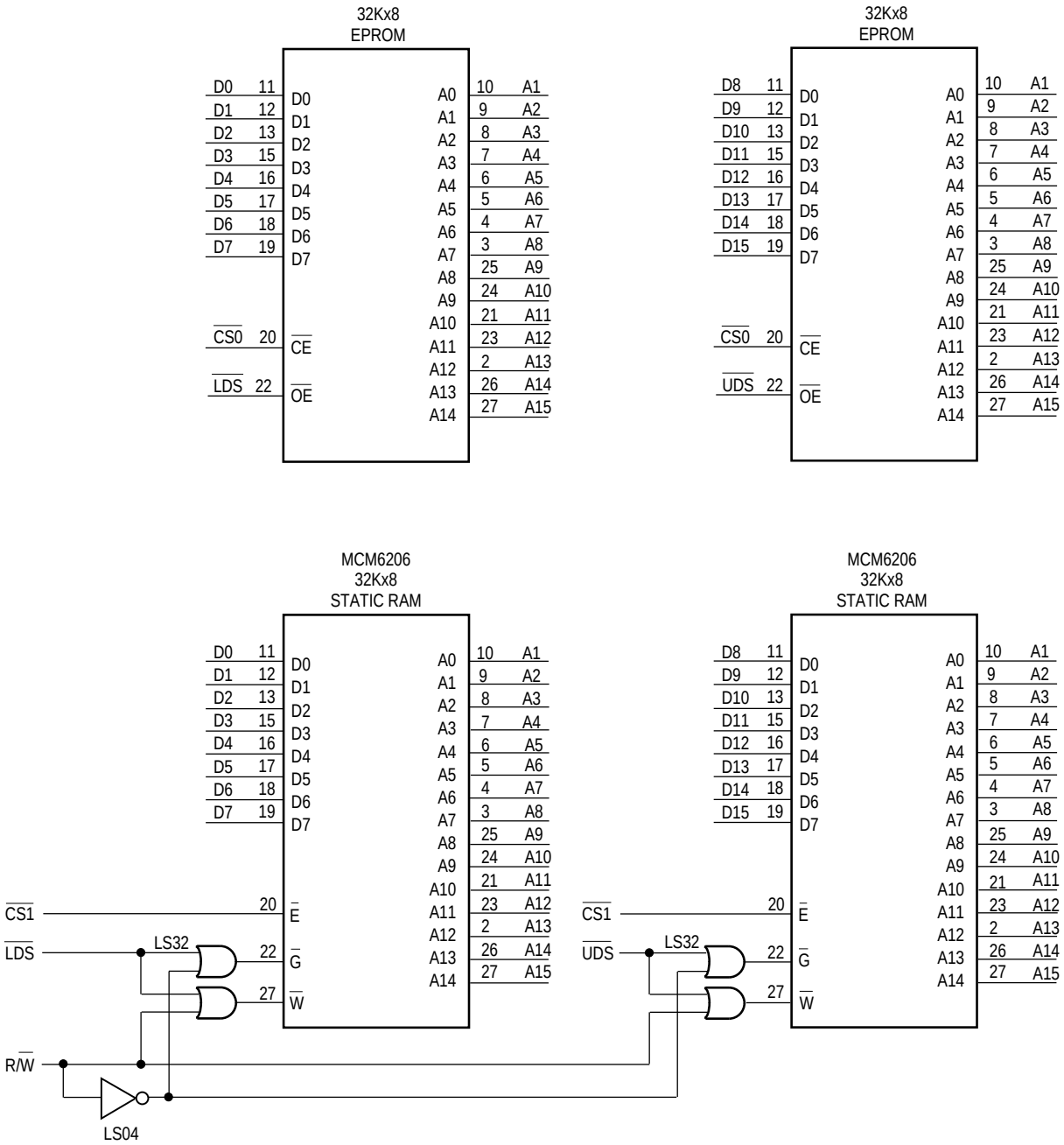


Figure D-2. MC68302 Minimum System Configuration (Sheet 2 of 2)

D.1.2 Reset Circuit

$\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ are both open-drain signals. When both signals are externally asserted, the entire MC68302 is reset. $\overline{\text{HALT}}$ is asserted by the MC68302 when the HALT instruction is executed, and $\overline{\text{RESET}}$ is asserted when the RESET instruction is executed.

The reset circuit shown uses an MC1455 timer to generate a 0.5-sec pulse *until* DIS reaches the 2/3 V_{CC} threshold level. After that, DIS discharges the 0.47-µF capacitor. When DIS falls below 2/3 V_{CC}, then the output pin (O) is pulled low, ending the reset to the MC68302. This

2-1/2 clocks –data setup time–$\overline{CS}$ maximum asserted time = 150 - 7 - 40 = 103 ns.

For proper timing of write cycles, it is important to know how much time elapses between $\overline{CS}$ negated and the data out hold time. With a 16.67-MHz MC68302, this value is a 10-ns minimum. Thus, on a $\overline{CS}$-controlled RAM write cycle, the RAM required data hold time must be <= 10 ns. For the MCM6206, it is 0 ns.

D.2 SWITCHING THE EXTERNAL ROM AND RAM USING THE MC68302

When designing with the MC68302 (or simply the M68000), one of the tasks often required is to switch the locations of the ROM and RAM in the system. This is because the reset vector needs to be supplied from ROM; whereas, the rest of the exception vectors are normally included in RAM so that they can be modified. The exception vector area extends from $0 to $FF.

The MC68302 makes this switch easy to perform, requiring no additional glue logic. Two chip-select lines and a few temporary locations of the dual-port RAM are used. For this example, 256 kbytes of EPROM and 64 kbytes of RAM are used.

D.2.1 Conditions at Reset

We physically connect $\overline{CS0}$ to the ROM and $\overline{CS1}$ to the RAM. After reset, the MC68302 defaults to having only one chip select enabled ($\overline{CS0}$), with a base address of $0 and a range of 8 kbytes. The ROM should be programmed with an initial set of interrupt vectors, and the reset vector should point to some location within the first 8 k of ROM. $\overline{CS1}$ is disabled at reset so the RAM cannot be accessed. Also, the base address register (BAR) in the MC68302 has not been written; therefore, the dual-port RAM and other on-chip peripherals cannot be accessed. The status register (SR) of the M68000 core is $2700, putting the core in supervisor mode with interrupts masked. The SR is left in this condition until the ROM and RAM are switched.

D.2.2 First Things First

The following situation now exists:

- ROM—Begins at $0; only first 8 kbytes are valid for programming.
- RAM—Undefined.
- MC68302—Undefined (except BAR at $0F2 and SCR at $0F4).

Once the reset vector has been taken, the BAR at location $0F2 should be written to locate the MC68302 dual-port RAM and on-chip peripheral registers in memory. In this example, $700000 is chosen as the starting point of the 4 k block of the MC68302 internal address space; thus, we write $0700 to BAR. BAR exists internal to the MC68302 and is easily accessed even though it overlaps memory with ROM. $\overline{CS0}$ *will not* activate on accesses to the BAR.

Now that the MC68302 can be fully accessed, the ROM and RAM chip selects can be modified. First, the option register (OR) of $\overline{CS0}$ can be extended to include all 256 kbytes by writing the base address mask in OR0 with 11111100000b. At this time, the DTACK field may be changed to reduce the number of wait states from six to something lower. Also, while writ-

D.3 MC68302 BUFFER PROCESSING AND INTERRUPT HANDLING

The following paragraphs describe how to build an algorithm to process the buffers for the MC68302 serial communication controller (SCC) channels.

D.3.1 Buffer Descriptors Definition

Data buffers used by the MC68302 are controlled by buffer descriptors (BDs). The general structure of a BD is shown in Figure D-3. The processing of buffers by software is done by examining BDs. Thus, BDs are the focus of this discussion.

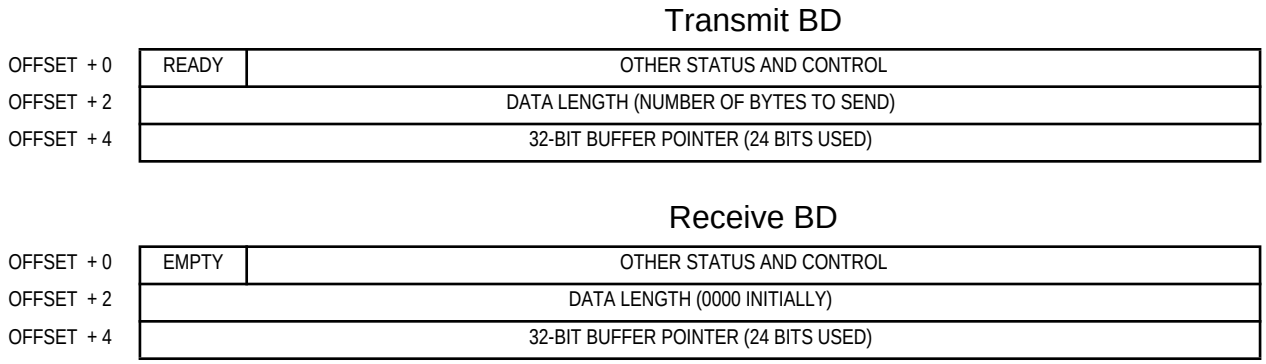


Figure D-3. Transmit and Receive BD Structure

Each transmit BD has a very important bit called the “ready” bit. This bit is set by the M68000 user program to signify to the SCC that the BD has data ready for sending. Similarly, the “empty” bit tells whether a receive BD is empty and can be used by the SCC for locating an empty buffer to store incoming data.

In the MC68302, up to 8 receive BDs and 8 transmit BDs can be defined per SCC. These BDs are stored in predefined places in the MC68302 dual-port RAM. The “wrap” bit is set in the last BD, causing the SCC to wrap back around to the first BD when processing of the last BD is complete. Thus, each set of BDs form a circular queue. An example is shown in Figure D-4.

of the physical interface configuration. Similarly, all bits in the receive buffer will be filled with real transparent data (full packing is always performed), regardless of the physical interface configuration.

If no data is available to transmit, transparent mode will transmit ones. The decision of whether to set the last (L) bit in the Tx BD is left to the user. If multiple buffers are to be sent back-to-back with no gaps in between, the L bit should be cleared in all buffers except for the last buffer. In this case, failure to provide buffers in time will result in a transmit underrun. If the L bit is set, the frame will end without error, and the transmission of ones will resume.

The transmit byte count and buffer alignment need not be even, but the SDMA channel will always read words on an even-byte boundary, even if it has to discard one of the two bytes. For example, if a transmit buffer begins on an odd-byte boundary and is 10 bytes in length (worst case), six word reads will result, even though only 10 bytes will be transmitted.

The receive buffer length (stored in MRBLR) and starting address must be even. All transfers to memory will be of word length and, unless an error occurs, a buffer will not be closed until it contains MRBLR/two words (the byte count will be equal to MRBLR). This raises an important point. Data received will only be transmitted to memory every 16 clocks. If a non-multiple of 16 bits is sent in a frame, the residue bits will not be transmitted to memory until additional bits arrive, and it will be impossible to demarcate frames unless their length is predetermined. (If a SYNC character is received with the data, the BISYNC mode can be used to receive an odd number of bytes with odd-length receive buffers and pointers allowed. (For more detailed information, refer to D.8.6 Other NMSI Modes**.)**

When the enable transmitter (ENT) bit is set, the process of polling the Tx BD begins by the RISC. The frequency of this polling is determined by the SCC's transmit clock. If the clock is stopped, no polling will occur. When the ready bit of the first Tx BD is set, the RISC initiates the SDMA activity of filling up the transmit FIFO with three words of data. Once the FIFO is full, the $\overline{\text{RTS}}$ signal is asserted, and the physical interface signals take control to determine the exact timing of the transmitted data. Once the physical interface says “go”, typically one final $FF is transmitted before data begins; however, whether $FF is transmitted depends on the mode chosen.

When the enable receiver (ENR) bit is set and 16 bits of valid data (as defined by the physical interface signals) have been clocked into the receiver, the RISC checks to see if the first receive buffer is available and, if the buffer is available, begins moving the data to it. The receive FIFO is three words deep, but a single open entry in the FIFO causes an SDMA service request. There are three types of receive errors: overrun (receive FIFO overflow), busy (new data arrived without a receive buffer being available), and $\overline{\text{CD}}$ lost (which is not possible in any example configuration discussed in this appendix). These errors are reported in the SCC event register (SCCE) or the Rx BD.

Whenever a buffer has been transmitted with the interrupt (I) bit set in the Tx BD, the TX event in the SCCE register will be set. This TX bit can cause an interrupt if the corresponding bit in the SCCM is set. Similarly, whenever a buffer has been received with the interrupt (I) bit set in the Rx BD, the RX event in the SCCE register will be set. Also, whenever a word of data is written to the receive buffer, the RCH bit is set in the SCCE.

There is nothing to synchronize on the receive side. As soon as the ENR bit is set, the first time slot for this SCC will begin the reception process. As with the NMSI mode, a word is not written to the buffer until the 9th clock after the serial clock that clocked in the last bit of this word (see D.8.7 Gating Clocks in NMSI Mode). Recall that clocks can only be counted *during* time slots.

D.8.9 PCM Mode Final Thoughts

Since all synchronous protocols work with PCM mode, it is possible to use the regular BISYNC mode to send syncs in the data stream, as described earlier.

When using totally transparent mode with PCM, both the NTSYN and EXSYN bits should normally be set. The DIAG1-DIAG0 bits can be set for either normal mode or software operation with no difference in behavior.

PCM mode does affect the SCCS register. In the SCCS register, the $\overline{CD}$ and $\overline{CTS}$ bits are always zero when the ENR and ENT bits, respectively, are set. The ID bit is not valid in transparent mode, regardless of the physical interface chosen.

Care should always be taken to avoid glitches or ringing on the L1CLK line. If a glitched or ringing L1CLK line causes an extra clock to be inserted during a time slot, there is no way to resynchronize the byte alignment in envelope mode until the ENT synchronization algorithm described previously is followed. This potential problem does lead to one slight advantage of the one-clock-prior method over the envelope sync. With the one-clock-prior method, it is more likely that the glitched clock will only misalign the transfer/reception of a single byte of data, rather than the whole data stream. (However, this cannot be guaranteed—predicting device behavior out-of-spec is extremely difficult.)

D.8.10 Using Transparent Mode with IDL and GCI

Transparent mode can be freely used with the ISDN physical interfaces. Using transparent mode with the ISDN interfaces is especially useful in the B-channels, since the D channel LAPD protocol is typically supported with the SCC in HDLC mode. Transparent data may be sent and received over the 64-kbps B1 channel, the 64-kbps B2 channel, a combined B1-B2 channel with a 128-kbps bandwidth, or subportions of either the B1 or B2 channel or both. (The desired subportions are defined in the SIMASK register.)

With the ISDN interfaces, as with the other types of interfaces, if the SCC is not transmitting data, it will transmit $FFs. If the NTSYN and the EXSYN bits are set in the SCM, data will be byte-aligned within the B1 or B2 channels. Thus, it will only be transmitted once the SCC transmit FIFO is filled and the *beginning* of the B1 or B2 channel occurs.

A special case occurs when the B1 and B2 channels are combined into a single 128-kbps channel. In this case, although data will only appear on byte boundaries, the transmit buffer's data could begin in either the B1 or B2 channels, depending on the timing involved. If this is a problem, the following rule may be observed. If the ENT bit is set at a consistent time during the GCI/IDL frame and if ready bit of the Tx BD is set at a consistent time relative to the GCI/IDL frame (preferably before the ENT bit is set), a consistent starting point of byte alignment (either B1 or B2) can be obtained. If data is then transmitted in a continuous

I—Interrupt

0 = The TXB bit in the event register is not set when this buffer is closed.
1 = The TXB bit in the event register is set if this buffer closed without an error. If an error occurred, then TXE is set.

L—Last in Frame

0 = This buffer is not the last buffer in a frame.
1 = This buffer is the last buffer in a frame.

TC—Tx CRC

0 = Transmit the closing flag after the last data byte.
1 = Transmit the CRC sequence after the last data byte.

Bits 9-2—Reserved for future use

UN—Underrun

0 = No transmitter underrun occurred.
1 = A transmitter underrun condition occurred while transmitting the associated data buffer.

CT—CTS Lost

0 = No CTS or L1GR lost was detected during frame transmission.
1 = CTS in NMSI mode or L1GR in IDL/GCI mode was lost during frame transmission.

E.1.1.5.2 Transmit Buffer Data Length. This 16-bit value is written by the user to indicate the number of data bytes to be transmitted from the data butter.

E.1.1.5.3 Transmit Buffer Pointer. This 32-bit value is written by the user to indicate the address of the first byte of data in the data buffer.

E.1.2 Programming the SCC for HDLC

This section gives a generic algorithm for programming an SCC to handle HDLC. The algorithm is intended to show what must be done and in what order to initialize the SCC and prepare the SCC for transmission and reception. The algorithm is not specific and assumes that the IMP and other on-chip peripherals have been initialized as required by the system hardware (timers, chip selects, etc.).

E.1.2.1 CP INITIALIZATION.

1. Write the port A and port B control registers (PACNT and PBCNT) to configure SCC2 or SCC3 serial interface pins as peripheral pins, if SCC2 or SCC3 is used.
2. Write SIMODE to configure the SCCs physical interface.
3. Write SIMASK if IDL or GCI multiplexed mode was selected in SIMODE.

E.1.2.2 GENERAL AND HDLC PROTOCOL-SPECIFIC RAM INITIALIZATION.

4. Write RFCR/TFCR.

OPCODE—Command Opcode

00 = STOP TRANSMIT Command.
01 = RESTART TRANSMIT Command.
10 = ENTER HUNT MODE Command.
11 = Reset receiver BCS generator (used only in BISYNC mode).

Bit 3—Reserved (should be set to zero by the user when the command register is written)

CH. NUM.—Channel Number

00 = SCC1.
01 = SCC2.
10 = SCC3.
11 = Reserved.

FLG—Command Semaphore Flag (set by the user and cleared by the CP upon command completion)

0 = CP is ready to receive a new command (should be checked before issuing next command to the CP).
1 = Command register contains a command to be executed or one that is currently being executed.

E.3.1.1.2 Serial Interface Mode Register (SIMODE). This 16-bit register is located at offset $8B4. The SIMODE register is used to configure the serial interface operation.

15	14	13	12	11	10	9	8
SETZ	SYNC/SCIT	SDIAG1	SDIAG0	SDC2	SDC1	B2RB	B2RA

7	6	5	4	3	2	1	0
B1RB	B1RA	DRB	DRA	MSC3	MSC2	MS1	MS0

SETZ—Set L1TXD to Zero (valid only for the GCI interface)

0 = Normal Operation
1 = L1TXD output set to a logic zero (used in GCI activation)

SYNC/SCIT—SYNC Mode/SCIT Select Support

0 = One pulse wide prior to the 8-bit data.
1 = N pulses wide and envelopes the N-bit data.

SDIAG1, SDIAG0—Serial Interface Diagnostic Mode

00 = Normal operation.
01 = Automatic Echo.
10 = Internal loopback.
11 = Loopback Control.

E.3.1.2.6 Transparent Status Register (SCCS). This 8-bit register is located at offset $88C (SCC1), $89C (SCC2), and $8AC (SCC3)on D15-D8 of a 16-bit data bus. The SCCS register reflects the current status of the CD and CTS lines as seen by the SCC.

7	6	5	4	3	2	1	0
—	—	—	—	—	—	CD	CTS

$\overline{CD}$—Carrier Detect Status Changed (valid only when the ENR bit is set and the receive clock is running)

0 = CD is asserted.
1 = CD is not asserted.

$\overline{CTS}$—Clear-To-Send Status Changed (valid only when the ENT bit is set and the transmit clock is running)

0 = $\overline{CTS}$ is asserted.
1 = $\overline{CTS}$ is not asserted.

E.3.1.3 GENERAL AND TRANSPARENT PROTOCOL-SPECIFIC PARAMETER RAM. Each SCC has 32 words of parameter RAM used to configure receive and transmit operation, store temporary parameters for the CP, and maintain counters. The first 14 words are general parameters, which are the same for each protocol. The last 18 words are specific to the protocol selected. The following subsections discuss the parameters that the user must initialize to configure the transparent operation.

E.3.1.3.1 RFCR/TFCR—Rx Function Code/Tx Function Code. This 16-bit parameter contains the function codes of the receive data buffers and transmit data buffers. The user must initialize the function codes (FC2-FC0) to a value less than 7.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	FC2	FC1	FC0	0	0	0	0	0	FC2	FC1	FC0	0	0	0	0

E.3.1.3.2 MRBLR—Maximum Rx Buffer Length. This 16-bit parameter defines the maximum receiver buffer length for each of the eight receive buffer descriptors.

E.3.1.4 RECEIVE BUFFER DESCRIPTORS. Each SCC has eight receive buffer descriptors. Each buffer descriptor consists of four words as shown below. Reserved bits in registers should be written as zeros.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	E	X	W	I	—	—	—	—	—	—	—	—	—	—	OV	CD
OFFSET +2	DATA LENGTH															
OFFSET +4	RX BUFFER POINTER															
OFFSET +6																

E.3.1.4.1 Receive BD Control/Status Word. To initialize the buffer, the user should write bits15-12 and clear bits1-0. The IMP clears bit 15 when the buffer is closed and sets bits 5-0 depending on which error occurred.