Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	M68000
Number of Cores/Bus Width	1 Core, 8/16-Bit
Speed	20MHz
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-BPGA Exposed Pad
Supplier Device Package	132-PGA (34.5x34.5)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68302rc20c

3.2.5.4 Interrupt In-Service Register (ISR).

Each bit in the 16-bit ISR corresponds to an INRQ interrupt source. In a vectored interrupt environment, the interrupt controller sets the ISR bit when the vector number corresponding to the INRQ interrupt source is passed to the core during an interrupt acknowledge cycle. The user's interrupt service routine should clear this bit during the servicing of the interrupt. (If an event register exists for this peripheral, its bits should also be cleared by the user program.) To clear a bit in the ISR, the user writes a one to that bit. The user can only clear bits in this register, and bits that are written with zeros will not be affected. The ISR is cleared at reset.

This register may be read by the user to determine which INRQ interrupts are currently being processed. More than one bit in the ISR may be a one if the capability is used to allow higher priority level 4 interrupts to interrupt lower priority level 4 interrupts. See 3.2.2.3 Nested Interrupts for more details.

The user can control the extent to which level 4 interrupts may interrupt other level 4 interrupts by selectively clearing the ISR. A new INRQ interrupt will be processed if it has a higher priority than the highest priority INRQ interrupt having its ISR bit set. Thus, if an INRQ interrupt routine lowers the 3-bit mask in the M68000 core to level 3 and also clears its ISR bit at the beginning of the interrupt routine, then a lower priority INRQ interrupt can interrupt it as long as the lower priority is higher than any other ISR bits that are set.

If the INRQ error vector is taken, no bit in the ISR is set. Bit 0 of the ISR is always zero.

15	14	13	12	11	10	9	8
PB11	PB10	SCC1	SDMA	IDMA	SCC2	TIMER1	SCC3

7	6	5	4	3	2	1	0
PB9	TIMER2	SCP	TIMER3	SMC1	SMC2	PB8	0

3.2.6 Interrupt Handler Examples

The following examples illustrate proper interrupt handling on the IMP. Nesting of level 4 interrupts (a technique described earlier) is not implemented in the following examples.

Example 1—Timer 3 (Software Watchdog Timer) Interrupt Handler

1. Vector to interrupt handler.
2. (Handle Event)
3. Clear the TIMER3 bit in the ISR.
4. Execute RTE instruction.

Example 2— SCC1 Interrupt Handler

1. Vector to interrupt handler.

SCR), will not activate the chip-select lines. Thus, it is convenient to use one of the chip-select lines to select external ROM/RAM that overlaps these register addresses, since, in this way, bus contention is completely avoided during a read access to these addresses. If, in a given application, it is not possible to use the chip-select lines for this purpose, the IAC signal may be used externally to prevent bus contention.

NOTE

The chip-select logic does not allow an address match during interrupt acknowledge cycles.

A special case occurs when the locked read-modify-write test and set (TAS) instruction is executed in combination with the chip selects. The assertion of wait states on the write portion of the cycle will only occur if the RMCST bit in the SCR is set. Refer to 3.8.3 System Control Bits for more details.

3.6.1 Chip-Select Logic Key Features

Key features of the chip-select logic are as follows:

- Four Programmable Chip-Select Lines
- Various Block Sizes: 8K, 16K, 32K, 64K, 128K, 256K, 512K, 1M, 2M, 4M, 8M, and 16M Bytes
- Read-Only, Write-Only, or Read-Write Select
- Internal $\overline{\text{DTACK}}$ Generation with Wait-State Options
- Default Line ($\overline{\text{CS0}}$) to Select an 8K-Boot ROM Containing the Reset Vector and Initial Program

3.6.2 Chip-Select Registers

Each of the four chip-select units has two registers that define its specific operation. These registers are a 16-bit base register (BR) and a 16-bit option register (OR) (e.g., BR0 and OR0). These registers may be modified by the M68000 core. The BR should normally be programmed after the OR since the BR contains the chip-select enable bit.

3.6.2.1 Base Register (BR3–BR0)

These 16-bit registers consist of a base address field, a read-write bit, and a function code field.

15 13	12 2	1	0
FC2 –FC0	BASE ADDRESS (A23–A13)	RW	EN

FC2–FC0 —Function Code Field

This field is contained in bits 15–13 of each BR. These bits are used to set the address space function code. The address compare logic uses these bits to determine whether an

EN—Enable

0 = The chip-select line is disabled.
1 = The chip-select line is enabled.

After system reset, only $\overline{CS0}$ is enabled; $\overline{CS3}$–$\overline{CS1}$ are disabled. In disable CPU mode, $\overline{CS3}$–$\overline{CS0}$ are disabled at system reset. The chip select does not require disabling before changing its parameters.

3.6.2.2 Option Registers (OR3–OR0)

These four 16-bit registers consist of a base address mask field, a read/write mask bit, a compare function code bit, and a $\overline{DTACK}$ generation field.

Bits 15–12—DTACK Field

These bits are used to determine whether $\overline{DTACK}$ is generated internally with a programmable number of wait states or externally by the peripheral. With internal $\overline{DTACK}$ generation, zero to six wait states can be automatically inserted before the $\overline{DTACK}$ pin is asserted as an output (see Port A Control Register (PACNT)).

Table 3-8. DTACK Field Encoding

Bits			Description
15	14	13	
0	0	0	No Wait State
0	0	1	1 Wait State
0	1	0	2 Wait States
0	1	1	3 Wait States
1	0	0	4 Wait States
1	0	1	5 Wait States
1	1	0	6 Wait States
1	1	1	External DTACK

When all the bits in this field are set to one, $\overline{DTACK}$ must be generated externally, and the IMP or external bus master waits for $\overline{DTACK}$ (input) to terminate its bus cycle. After system reset, the bits of the DTACK field default to six wait states.

The DTACK generator uses the IMP internal clock to generate the programmable number of wait states. For asynchronous external bus masters, the programmable number of wait states is counted directly from the internal clock. When no wait state is programmed (DTACK = 000), the DTACK generator will generate $\overline{DTACK}$ asynchronously.

The $\overline{CS}$ lines are asserted slightly earlier for internal IMP master memory cycles than for an external master using the $\overline{CS}$ lines. Set external master wait state (EMWS) in the SCR whenever these timing differences require an extra memory wait state for external masters.

After system reset, this bit defaults to zero. If BCLM is set, then the typical maximum interrupt latency is about 78 clocks in a zero-wait-state system. This assumes a standard instruction mix, that the IDMA is just beginning a four-bus-cycle transfer when the interrupt becomes pending, and that an SDMA has an access pending (one bus cycle). Interrupt execution time is 44 clocks and includes the time to execute the interrupt acknowledge cycle, save the status register and PC value on the stack, and then vector to the first location of the interrupt service routine. Thus, the calculation is 78 = 14 (instruction completion) + 20 (DMAs) + 44 (interrupt execution).

SDMA operation is not affected by the BCLM bit. Note that the SDMA accesses only one byte/word of external memory at a time before giving up the bus and that accesses are relatively infrequent. External bus master operation may or may not be affected by the BCLM bit, depending on whether the $\overline{\text{BCLR}}$ signal is used to clear the external master off the bus.

Without using the BCLM bit, the maximum interrupt latency includes the maximum time that the IDMA or external bus master could use the bus in the worst case. Note that the IDMA can limit its bus usage if its requests are generated internally.

NOTE

The IPA status bit will be set, regardless of the BCLM value.

SAM—Synchronous Access Mode

This bit controls how external masters may access the MC68302 peripheral area. This bit is not relevant for applications that do not have external bus masters that access the MC68302. In applications such as disable CPU mode, in which the M68000 core is not operating, the user should note that SAM may be changed by an external master on the first access of the MC68302, but that first write access must be asynchronous with three wait states. (If $\overline{\text{DTACK}}$ is used to terminate bus cycles, this change need not influence hardware.)

0 = Asynchronous accesses. All accesses to the MC68302 internal RAM and registers (including BAR and SCR) by an external master are asynchronous to the MC68302 clock. Read and write accesses are with three wait states, and $\overline{\text{DTACK}}$ is asserted by the MC68302 assuming three wait-state accesses. This is the default value.

1 = Synchronous accesses. All accesses to the MC68302 internal RAM and registers (including BAR and SCR) must be synchronous to the MC68302 clock. Synchronous read accesses may occur with one wait state if EMWS is also set to one.

3.8.4 Disable CPU Logic (M68000)

The MC68302 can be configured to operate solely as a peripheral to an external processor. In this mode, the on-chip M68000 CPU should be disabled by strapping DISCPU high during system reset ($\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ asserted simultaneously). The internal accesses to the MC68302 peripherals and memory may be asynchronous or synchronous. During synchronous reads, one wait state may be used if required (EMWS bit set). The following pins change their functionality in this mode:

1. $\overline{\text{BR}}$ will be an output from the IDMA and SDMA to the external M68000 bus, rather than being an input to the MC68302.

HWDCN–HWDCN0—Hardware Watchdog Count 2–0

000 = $\overline{\text{BERR}}$ is asserted after 128 clock cycles (8 µs, 16-MHz clock)
001 = $\overline{\text{BERR}}$ is asserted after 256 clock cycles (16 µs, 16-MHz clock)
010 = $\overline{\text{BERR}}$ is asserted after 512 clock cycles (32 µs, 16-MHz clock)
011 = $\overline{\text{BERR}}$ is asserted after 1K clock cycles (64 µs, 16-MHz clock)
100 = $\overline{\text{BERR}}$ is asserted after 2K clock cycles (128 µs, 16-MHz clock)
101 = $\overline{\text{BERR}}$ is asserted after 4K clock cycles (256 µs, 16-MHz clock)
110 = $\overline{\text{BERR}}$ is asserted after 8K clock cycles (512 µs, 16-MHz clock)
111 = $\overline{\text{BERR}}$ is asserted after 16K clock cycles (1 ms, 16-MHz clock)

After system reset, these bits default to all ones; thus, $\overline{\text{BERR}}$ will be asserted after 1 ms for a 16-MHz system clock.

NOTE

Successive timeouts of the hardware watchdog may vary slightly in length. The counter resolution is 16 clock cycles.

3.8.7 Reducing Power Consumption

There are a number of ways to reduce power consumption on the IMP. They can be classified as general power-saving tips and low-power modes.

3.8.7.1 Power-Saving Tips

Without using any of the IMP low-power modes, power consumption may be reduced in the following ways.

1. The system clock frequency of the IMP may be reduced to the lower limit of its operating frequency range (e.g., 8 MHz) as specified in Section 6 Electrical Characteristics.
2. When not used, the SCCs should be disabled by clearing the ENT and ENR bits in the SCM registers.
3. If not needed, the SCC baud rate generators should be disabled or caused to clock at a low frequency. The baud rate generators are initialized to a very fast clock rate after reset, which can be reduced by programming the SCON register.
4. The two general-purpose timer prescalers should be set to the maximum divider value, and the timers should be disabled if not used.
5. Any unneeded peripheral output pins that are multiplexed with parallel I/O pins should be left configured as parallel I/O pins. The smaller the number of output transistors switching, the less current used.

3.8.7.2 Low-Power (Standby) Modes

The IMP also supports several types of low-power modes. The low-power modes on the IMP are used when no processing is required from the M68000 core and when it is desirable to reduce system power consumption to its minimum value. All low-power modes are entered by first setting the low-power enable (LPEN) bit, and then executing the M68000 STOP instruction.

UN—Underrun

The HDLC controller encountered a transmitter underrun condition while transmitting the associated data buffer.

CT—CTS Lost

CTS in NMSI mode or L1GR (layer-1 grant) in IDL/GCI mode was lost during frame transmission. If data from more than one buffer is currently in the FIFO when this error occurs, this bit will be set in the Tx BD that is currently open.

Data Length

The data length is the number of octets the HDLC controller should transmit from this BD's data buffer. It is never modified by the CP. The value of this field should be greater than zero.

Tx Buffer Pointer

The transmit buffer pointer, which contains the address 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.

4.5.12.12 HDLC Event Register

The SCC event register (SCCE) is called the HDLC event register when the SCC is operating as an HDLC controller. It is an 8-bit register used to report events recognized by the HDLC channel and to generate interrupts. Upon recognition of an event, the HDLC controller sets its corresponding bit in the HDLC event register. Interrupts generated by this register may be masked in the HDLC mask register.

The HDLC event register is a memory-mapped register that may be read at any time. A bit is cleared by writing a one; writing a zero does not affect a bit's value. More than one bit may be cleared at a time. All unmasked bits must be cleared before the CP will clear the internal interrupt request. This register is cleared at reset.

An example of the timing of various events in the HDLC event register is shown in Figure 4-29.

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.

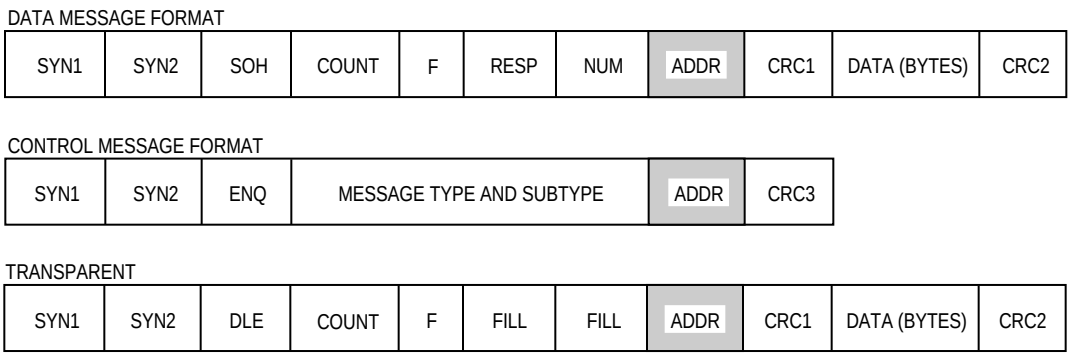


Figure 4-34. Typical DDCMP Frames

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.

NOTE

This error can occur only on synchronous links.

2. Clear-To-Send Lost (Collision) During Message Transmission. When this error occurs and the channel is not programmed to control this line with software, the channel terminates buffer transmission, closes the buffer, sets the CTS lost (CT) bit in the BD, and generates the transmit error (TXE) interrupt (if enabled). The channel resumes transmission after the reception of the RESTART TRANSMIT command.

Reception Errors:

1. Carrier Detect Lost During Message Reception. When this error occurs and the channel is not programmed to control this line with software, the channel terminates message reception, closes the buffer, sets the carrier detect lost (CD) bit in the BD, and generates the receive block (RBK) interrupt (if enabled). This error has the highest priority. The rest of the message is lost, and other errors in that message are not checked.
   The channel will enter hunt mode immediately. It is possible that a SYN1–SYN2-(SOH,DLE,ENQ) sequence in data will be incorrectly interpreted as the start of the next header, but this "header" will have a CRC error.
2. Overrun Error. The DDCMP controller maintains an internal three-byte FIFO for receiving data. The CP begins programming the SDMA channel (if the data buffer is in external memory) and updating the CRC when the first word is received into the FIFO. If the receive FIFO overrun error occurs, the channel writes the received data byte to the internal FIFO on top of the previously received byte. The previous data byte is lost. Then the channel closes the buffer, sets the overrun (OV) bit in the BD, and generates the receive block (RBK) interrupt (if enabled).
   The channel will enter hunt mode immediately. It is possible that a SYN1–SYN2-(SOH,DLE,ENQ) sequence in data will be incorrectly interpreted as the start of the next header, but this “header” will have a CRC error.
3. CRC1 (Header CRC) Error. When this error occurs, the channel writes the received CRC to the data buffer, closes the buffer, sets the CRC error (CR) bit in the BD, generates the RBK interrupt (if enabled), increments the error counter (CRC1EC), and enters hunt mode.
   When this error occurs on data-and maintenance-message header fields, the channel will enter hunt mode immediately. It is possible that a SYN1–SYN2-(SOH,DLE,ENQ) sequence in data will be incorrectly interpreted as the start of the next header, but this “header” will have a CRC error.
4. CRC2 (Data or Maintenance CRC) or CRC3 (Control Message) Error. When this error occurs, the channel writes the received CRC to the data buffer, closes the buffer, sets the CRC error (CR) bit in the BD, and generates the RBK interrupt (if enabled). The channel also increments the CRC2EC counter and enters hunt mode.
5. Framing Error. A framing error is detected by the DDCMP controller when no stop bit is detected in a received data string. When this error occurs, the channel writes the received character to the buffer, closes the buffer, sets the framing error (FR) bit in the BD, and generates the RBK interrupt (if enabled). When this error occurs, parity is not checked for this character.

ters. If DTACK is generated internally, then it is an output. It is an input when the IMP accesses an external device not within the range of the chip-select logic or when programmed to be generated externally.

RMC/IOUT1—Read-Modify-Write Cycle Indication/Interrupt Output 1

This signal functions as RMC in normal operation. RMC is an output signal that is asserted when a read-modify-write cycle is executed. It indicates that the cycle is indivisible.

When the M68000 core is disabled, this pin operates as IOUT1. IOUT2–IOUT0 provide the interrupt request output signals from the IMP interrupt controller to an external CPU when the M68000 core is disabled.

IAC—Internal Access

This output indicates that the current bus cycle accesses an on-chip location. This includes the on-chip 4K byte block of internal RAM and registers (both real and reserved locations), and the system configuration registers ($0F0–$0FF). The above-mentioned bus cycle may originate from the M68000 core, the IDMA, or an external bus master. Note that, if the SDMA accesses the internal dual-port RAM, it does so without arbitration on the M68000 bus; therefore, the IAC pin is not asserted in this case. The timing of IAC is identical to that of the CS3–CS0 pins.

IAC can be used to disable an external address/data buffer when the on-chip dual-port RAM and registers are accessed, thus preventing bus contention. Such a buffer is optional and is only required in larger systems. An external address/data buffer with its output enable (E) and direction control (dir) may be placed between the two bus segments as shown in Figure 5-7. The IAC signal saves the propagation delay and logic required to OR all the various system chip-select lines together to determine when to enable the external buffers.

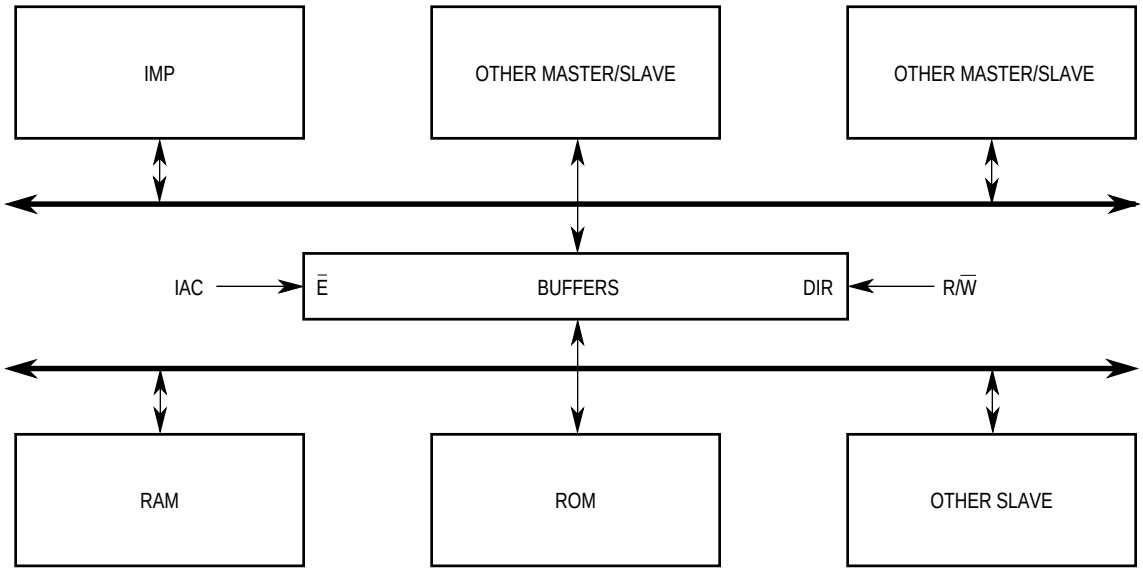


Figure 5-7. External Address/Data Buffer

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

6.8 AC ELECTRICAL SPECIFICATIONS—IMP BUS MASTER CYCLES

(see Figure 6-2, Figure 6-3, Figure 6-4, and Figure 6-5))

			16.67 MHz		20 MHz		25 MHz		
Num.	Characteristic	Symbol	Min	Max	Min	Max	Min	Max	Unit
6	Clock High to FC, Address Valid	$t_{CHFCADV}$	0	45	0	40	0	30	ns
7	Clock High to Address, Data Bus High Impedance (Maximum)	t_{CHADZ}	—	50	—	42	—	33	ns
8	Clock High to Address, FC Invalid (Minimum)	t_{CHAFI}	0	—	0	—	0	—	ns
9	Clock High to $\overline{AS}$, $\overline{DS}$ Asserted (see Note 1)	t_{CHSL}	3	30	3	25	3	20	ns
11	Address, FC Valid to $\overline{AS}$, $\overline{DS}$ Asserted (Read) $\overline{AS}$ Asserted Write (see Note 2)	t_{AFCVSL}	15	—	12	—	10	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$ Negated (see Note 1)	t_{CLSH}	—	30	—	25	—	20	ns
13	$\overline{AS}$, $\overline{DS}$ Negated to Address, FC Invalid (see Note 2)	t_{SHAFI}	15	—	12	—	10	—	ns
14	$\overline{AS}$ (and $\overline{DS}$ Read) Width Asserted (see Note 2)	t_{SL}	120	—	100	—	80	—	ns
14A	$\overline{DS}$ Width Asserted, Write (see Note 2)	t_{DSL}	60	—	50	—	40	—	ns
15	$\overline{AS}$, $\overline{DS}$ Width Negated (see Note 2)	t_{SH}	60	—	50	—	40	—	ns
16	Clock High to Control Bus High Impedance	t_{CHCZ}	—	50	—	42	—	33	ns
17	$\overline{AS}$, $\overline{DS}$ Negated to R/$\overline{W}$ Invalid (see Note 2)	t_{SHRH}	15	—	12	—	10	—	ns
18	Clock High to R/$\overline{W}$ High (see Note 1)	t_{CHRH}	—	30	—	25	—	20	ns
20	Clock High to R/$\overline{W}$ Low (see Note 1)	t_{CHRL}	—	30	—	25	—	20	ns
20A	$\overline{AS}$ Asserted to R/$\overline{W}$ Low (Write) (see Notes 2 and 6)	t_{ASRV}	—	10	—	10	—	7	ns
21	Address FC Valid to R/$\overline{W}$ Low (Write) (see Note 2)	t_{AFCVRL}	15	—	12	—	10	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$ Asserted (Write) (see Note 2)	t_{RLSL}	30	—	25	—	20	—	ns
23	Clock Low to Data-Out Valid	t_{CLDO}	—	30	—	25	—	20	ns
25	$\overline{AS}$, $\overline{DS}$, Negated to Data-Out Invalid (Write) (see Note 2)	t_{SHDOI}	15	—	12	—	10	—	ns
26	Data-Out Valid to $\overline{DS}$ Asserted (Write) (see Note 2)	t_{DOSL}	15	—	12	—	10	—	ns
27	Data-In Valid to Clock Low (Setup Time on Read) (see Note 5)	t_{DICL}	7	—	6	—	5	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DTACK}$ Negated (Asynchronous Hold) (see Note 2)	t_{SHDAH}	0	110	0	95	0	75	ns
29	$\overline{AS}$, $\overline{DS}$ Negated to Data-In Invalid (Hold Time on Read)	t_{SHDII}	0	—	0	—	0	—	ns
30	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{BERR}$ Negated	t_{SHBEH}	0	—	0	—	0	—	ns
31	$\overline{DTACK}$ Asserted to Data-In Valid (Setup Time) (see Notes 2 and 5)	t_{DALDI}	—	50	—	42	—	33	ns
32	$\overline{HALT}$ and $\overline{RESET}$ Input Transition Time	t_{RHr}, t_{RHf}	—	150	—	150	—	150	ns
33	Clock High to $\overline{BG}$ Asserted	t_{CHGL}	—	30	—	25	—	20	ns
34	Clock High to $\overline{BG}$ Negated	t_{CHGH}	—	30	—	25	—	20	ns
35	$\overline{BR}$ Asserted to $\overline{BG}$ Asserted (see Note 11)	t_{BRLGL}	2.5	4.5	2.5	4.5	2.5	4.5	clks
36	$\overline{BR}$ Negated to $\overline{BG}$ Negated (see Note 7)	t_{BRHGH}	1.5	2.5	1.5	2.5	1.5	2.5	clks

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.

6.12 AC ELECTRICAL SPECIFICATIONS—INTERNAL MASTER INTERNAL READ/WRITE CYCLES (see Figure 6-13)

			16.67 MHz		20 MHz		25 MHz		
Num.	Characteristic	Symbol	Min	Max	Min	Max	Min	Max	Unit
140	Clock High to IAC High	t_{CHIAH}	—	40	—	35	—	27	ns
141	Clock Low to IAC Low	t_{CLIAL}	—	40	—	35	—	27	ns
142	Clock High to $\overline{\text{DTACK}}$ Low	t_{CHDTL}	—	45	—	40	—	30	ns
143	Clock Low to $\overline{\text{DTACK}}$ High	t_{CLDTH}	—	40	—	35	—	27	ns
144	Clock High to Data-Out Valid	t_{CHDOV}	—	30	—	25	—	20	ns
145	$\overline{\text{AS}}$ High to Data-Out Hold Time	t_{ASHDOH}	0	—	0	—	0	—	ns

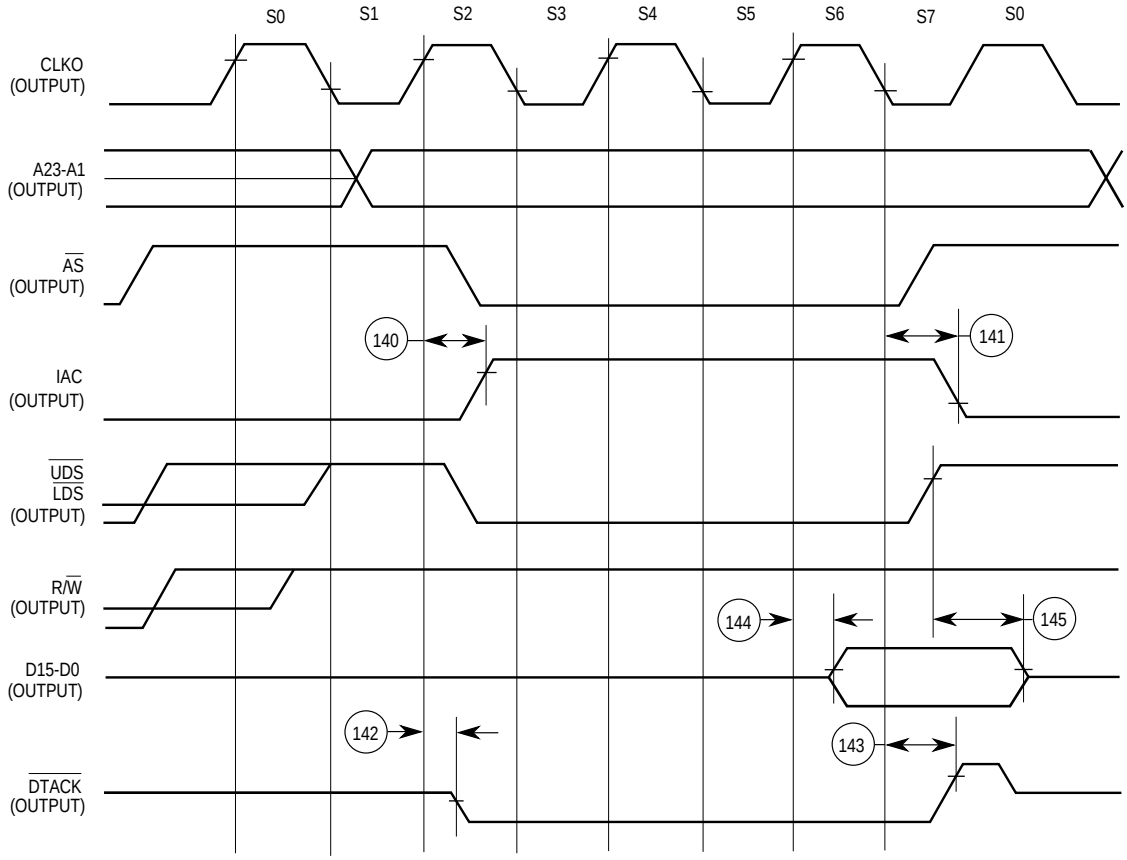


Figure 6-13. Internal Master Internal Read/Write Cycle Timing Diagram

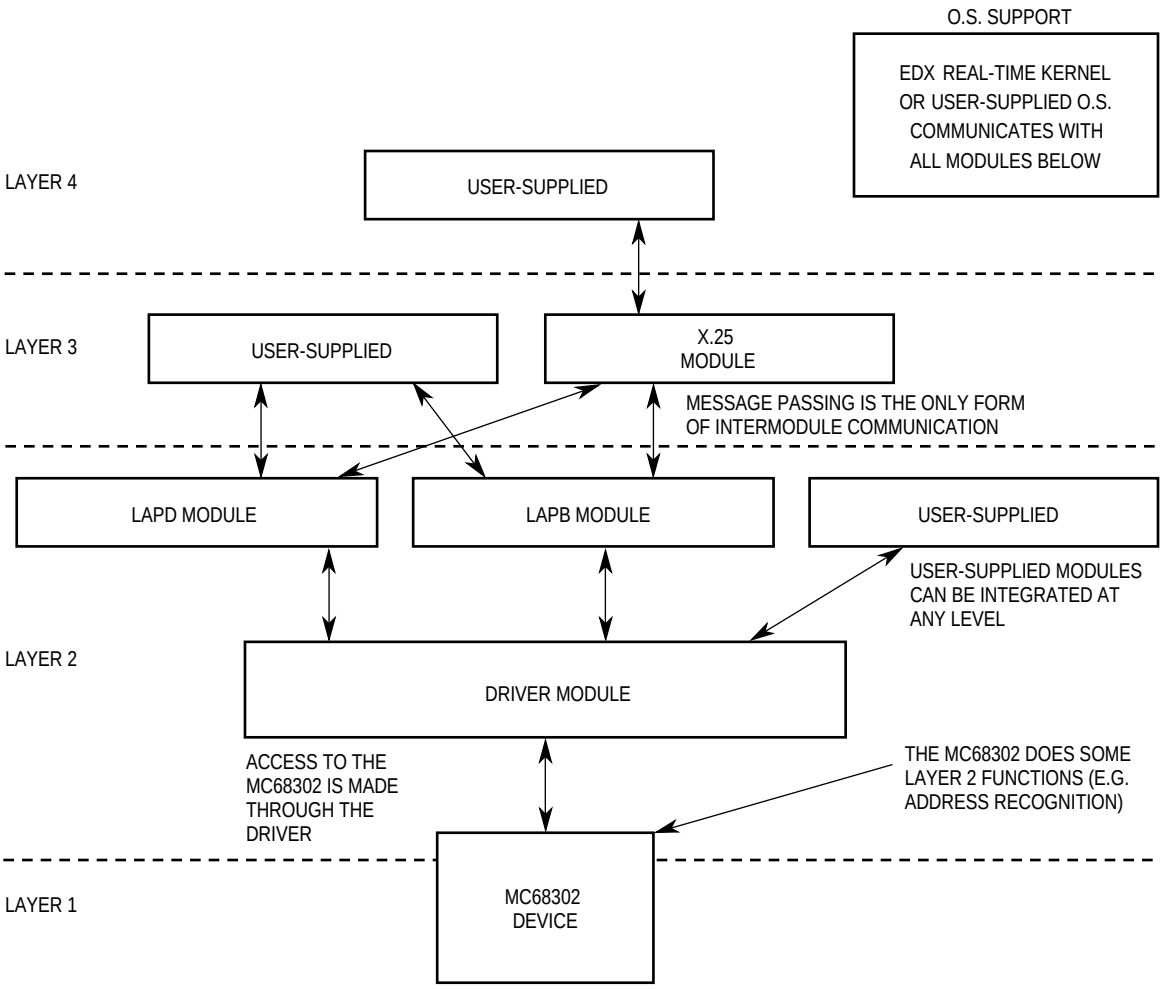


Figure B-1. Software Overview

The LAPD module features are as follows:

- Fully implements *1988 CCITT Recommendation Q.920/Q.921*
- Supports up to twelve physical channels—8192 logical links per channel
- Supports management and broadcast links
- Applicable for user and network applications
- Uses dedicated transmit pool for fast control frame generation
- Dynamic modification of protocol parameters
- Independent of layer 1 and layer 3 implementation
- Independent of layer 2 management implementation
- Message-oriented interface
- Independent configuration of upper and lower layer modules interfacing with each link
- Special mode for internal loopback between pairs of links

```
          BLT       NxtBT             ;Next Byte
          DBRA      D1,NxtBF          ;Next Buffer

***Now ready to go ***
***Set IMR and Enable SCC1 ***
          MOVE.W    #$2000,1MR        ;Allow SCC1 interrupt only
          MOVE.W    #$2000,SR         ;Unmask interrupts
          ORI.W     #EN_SCC,SCM1      ;ENT = ENR = 1
          JMP       MAIN              ;Go to Main routine for Tx and Rx

*****************************Main Routine ******************************
* Set up BD pointers
MAIN      LEA.L     TXBD_01,A1        ;A1 = CTD pointer
          LEA.L     TXBD_01,A2        ;A2 = NTD pointer
          LEA.L     RXBD_01,A3        ;A3= PRD pointer

* The following is an infinite loop that prepares data to be sent
* when a Tx BD is available to be used.
          CLR.L      D3               ;D3 is used to count Tx frames
*                                       transmitted in the loop
TxReady   BTST.B    #READY,(A2)       ;Test Ready Bit
          BNE.B     TxReady           ;If Ready = 0, the BD has been sent
Confirm   CMPI.W    #$0,LN_BD(A2)     ;test NTD ->data length
          BNE.B     Confirm           ;If  length = 0, the BD has been
*                                       confirmed

*Set TXBD if it is to be changed, e.g.,
*          ORI.W    #$5C00,ST_BD(A2)



* Mask off interrupt for the following operations
          MOVE.W    #$2700,SR
          MOVE.W    #N_DATA,LN_BD(A2) ;Set data length to 6
          BSET.B    #READY,ST_BD(A2)  ;Set Ready bit
          ADDO.L    #1,D3             ;Inc Tx frame count
          BTST.B    #WRAP,ST_BD(A2)   ;Test Wrap bit
          BNE.B     Wrapit            ;If Wrap . 1, wrap a
          ADDQ.W    #SZ_BD,A2         ;Move NTD to next BD
          BRA.B     Unmask
Wrapit    LEA.L     TXBD _ 01,A2      ;Wrap back to the first TX BD
Umask     MOVE.W    #$2000,SR         ;Unmask interrupt
          JMP       TxReady

*******************SCC1 interrupt handler *******************
          ORG       INT_VEC           ;Interrupt Vector for SCC1
* Check events:  Handle RX then TX then Errors
          CLR.L     D1                ;Clear D1
          MOVE.B    SCCE1,D1          ;SCCE1 =>, D1
          MOVE.L    D1,D2             ;SCCE1 =>, D2
          ANDI.W    #9,D2             ;Are RXF or RXB set?
          CMPI.W    #0,D2             ;If they are set
          BNE.B     RX_INT            ;Handle receiver's interrupt
CK_TX     MOVE.L    D1,D2             ;SCCE1 =>, D2
          ANDI.W    #$12,D2           ;Are TXF or TXB set?
```

D.6.9 Parallel I/O Port A Configuration

To implement a DCE interface for the non-ISDN terminal, PA2 and PA5 should be configured as general-purpose outputs (drive $\overline{\text{CTS}}$ and $\overline{\text{CD}}$).

PA7 and PA8 should be configured as general-purpose outputs (drive SCP_EN and RESET).

PA4 and PA6 may be configured either as general-purpose inputs or as dedicated modem pins.

If a general-purpose input is chosen, the state of the terminal $\overline{\text{DTR}}$ and $\overline{\text{RTS}}$ lines will be polled by application software by reading the parallel I/O data register.

If PA4 and PA6 are to be used as general-purpose inputs, the following registers should be set:

Register	Value	Comments
PACNT	$000B	Set PA0 PA1, and PA3 to the dedicated mode. (RXD2, TXD2, and TCLK2 connected to SCC2.)
PADDR	$01A4	Set PA2, PA5, PA7, and PA8 as output pins. ($\overline{\text{RCLK2}}$ and $\overline{\text{RTS2}}$ are outputs.)
PADAT		Will be set by the application software to drive PA2, PA5, PA7, and PA8 to the proper state. Will be read to check $\overline{\text{DTR}}$ and $\overline{\text{RTS}}$.

If the dedicated mode is chosen, the software can use the delta mechanism of the SCC to generate an interrupt upon any change in the state of $\overline{\text{RTS}}$ or $\overline{\text{DTR}}$. In this case, the SCC can be configured to use its $\overline{\text{CTS}}$ (connected to the terminal $\overline{\text{DTR}}$ line) and $\overline{\text{CD}}$ (connected to the terminal $\overline{\text{RTS}}$ line) under automatic control or under software control by setting the DIAG bits in the mode register.

If $\overline{\text{CTS2}}$ (PA4) and $\overline{\text{CD2}}$ (PA6) are to be used as dedicated pins, the following registers should be set:

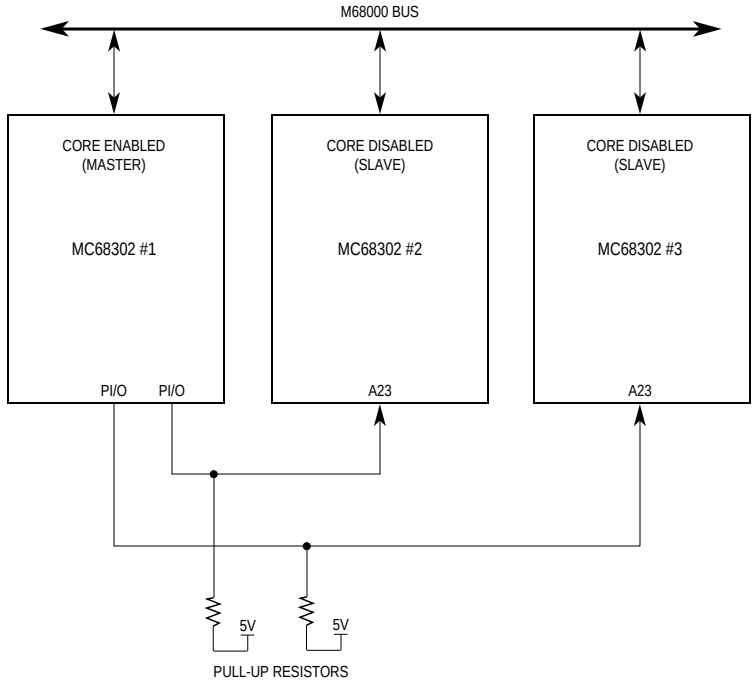
Register	Value	Comments
PACNT	$005B	Set PA0, PA1, PA3, PA4, and PA5 to the dedicated mode. (RXD2, TXD2, TCLK2, CTS2 and CD2 connected to SCC2.)
PADDR	$01A4	Set PA2, PA5, PA7, and PA8 as output pins. ($\overline{\text{RCLK2}}$ and $\overline{\text{RTS2}}$ are outputs.)
PADAT		Will be set by the application software to drive PA2, PA5, PA7, and PA8 to the proper state.

D.6.10 SCP Bus

The SCP (see Figure D-15) is an industry standard bus used for controlling and programming external devices. The SCP is a four-wire bus consisting of transmit path, receive path, associated clock, and enable signal (see Figure D-16). The clock determines the rate of the exchange of data in both the transmit and receive directions, and the enable signal governs when this exchange occurs.

fully functioning MC68302s, each having an isolated bus and the ability to send data and messages between them (e.g., through a shared RAM). However, another approach is possible.

By using the MC68302 “disable CPU logic” feature, enabled with the DISCPU pin, the MC68302 can be converted into an intelligent slave peripheral that no longer has its M68000 core operating. The SDMA channels and IDMA channel request the bus externally through the bus request ($\overline{BR}$) pin. (When not in slave mode, these channels request the bus internally to the on-chip bus arbiter, with no external indication visible.) A typical slave mode example is shown in Figure D-19. A single master MC68302 (i.e., one with the M68000 core enabled) can access and control one or more slave MC68302s. (i.e., ones with the M68000 core disabled.)



NOTE: A23 is not used by the slaves.

Figure D-19. Typical Slave Mode Example

Use of the “disable CPU logic” feature in a multiple MC68302 system depends mainly on the amount of protocol processing required by the M68000 core. If the data rates are high and the amount of protocol processing required on each channel is significant, the M68000 core may be the limiting factor in communications performance. Thus, further increases in serial rates will not yield additional packets/sec performance. In such a case, a faster processor (such as the MC68020/MC68030) could be used to control all three MC68302 devices in slave mode.

The bus utilization of the SDMA channels on the three MC68302 devices is not usually a significant factor. For instance, if three SCC channels are running full duplex at 64 kbps, the respective SDMA channels consume less than 1 percent of the M68000 bus. You can calculate this figure for your design by determining how often a bus cycle to memory is required

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.

Table E-1 (b). UART Programming Model (Continued)
General Parameter and UART Protocol-Specific RAM for SCCx

Initialized by User	Offset Hex	Name		Initialized by User	Offset Hex	Name
Yes	80	RFCR	TFCR	Yes	A2	Receive Parity Error Counter
Yes	82	MRBLR		Yes	A4	Receive Framing Error Counter
	84	Rx Internal State		Yes	A6	Receive Noise Counter
	86	Reserved	Rx Internal Buffer No.	Yes	A8	Receive Break Condition Counter
	88 8A	Rx Internal Data Pointer (High Word) Rx Internal Data Pointer (Low Word)		Yes	AA	UART Address Character 1
	8C	Rx Internal Byte Count		Yes	AC	UART Address Character 2
	8E	Rx Temp			AE	Receive Control Character Register
	90	Tx Internal State		Yes	B0	Control Character 1
	92	Reserved	Tx Internal Buffer No.	Yes	B2	Control Character 2
	94 96	Tx Internal Data Pointer (High Word) TX Internal Data Pointer (Low Word)		Yes	B4	Control Character 3
	98	Tx Internal Byte Count		Yes	B6	Control Character 4
	9A	Tx Temp		Yes	B8	Control Character 5
Yes	9C	Maximum IDLE Characters		Yes	BA	Control Character 6
	9E	Temporary Receive IDLE Counter		Yes	BC	Control Character 7
Yes	A0	Break Counter Register		Yes	BE	Control Character 8

NOTE: The offset is from the MC68302 base address + ($400 for SCC1, $500 for SCC2, or $600 for SCC3).

Table E-1 (c). SCCx Register Set

Initialized by User	Offset Hex	Name	
	00	Reserved	
Yes	02	SCC Configuration Register (SCON)	
Yes	04	SCC Mode Register (SCM)	
Yes	06	SCC Data Synchronization Register (DSR)	
Yes	08	Event Register (SCCE)	Reserved
Yes	0A	Mask Register (SCCM)	Reserved
	0C	Status Register (SCCS)	Reserved
	0E	Reserved	

NOTE: The offset is from the MC68302 base address + ($880 for SCC1, $890 for SCC2, or $8A0 for SCC3).

Table E-1 (d). General Registers (Only One Set)

Initialized by User	Offset Hex	Name	
	860	Command Register (CR)	Reserved
Yes	8B2	Serial Interface Mask Register (SIMASK)	
Yes	8B4	Serial Interface Mask Register (SIMODE)	

NOTE: The offset is from the MC68302 base address.

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

E.2.1.4.1 Receive BD Control/Status Word. To initialize the buffer, the user should write bits 15-12 and clear bits 11-0. The IMP clears bit 15 when the buffer is closed and sets bits 5-0 depending on which error occurred.

E—Empty

0 = This data buffer is full or has been closed due to an error condition.
1 = This data buffer is empty; must be set by the user to enable reception into this buffer.

X—External Buffer

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

W—Wrap (final BD in table)

0 = This is not the last BD in the receive BD table.
1 = This is the last BD in the receive BD table.

I—Interrupt

0 = The RX bit in the event register is not set when this buffer is closed.
1 = The RX bit in the event register is set when this buffer is closed.

C—Control Character

0 = This buffer does not contain a control character.
1 = The last byte of this buffer contains a control character.

A—Address

0 = This buffer contains data only.
1 = The first byte of this data buffer is an address byte.

M—Address Match

0 = The address byte matched UADDR2.
1 = The address byte matched UADDR1.

ID—IDLE Reception

0 = Buffer not closed due to reception of maximum number of IDLE characters (MAX_IDL).
1 = Buffer closed due to reception of maximum number of IDLE characters (MAX_IDL).