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	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/product-detail/nxp-semiconductors/mc68302cpv16vc

The maximum transfer rate is calculated from the fact that 16 bits are moved every 8 clocks. The calculation is as follows:

$$\frac{\text{16 bits x 16M clocks/sec}}{\text{(2 bus cycles) x (4 clocks/bus cycle)}} = \frac{\text{2 bytes x 16M clocks/sec}}{\text{8 clocks}} = \frac{\text{4M bytes}}{\text{sec}}$$

The IDMA controller block diagram is shown in Figure 3-1.

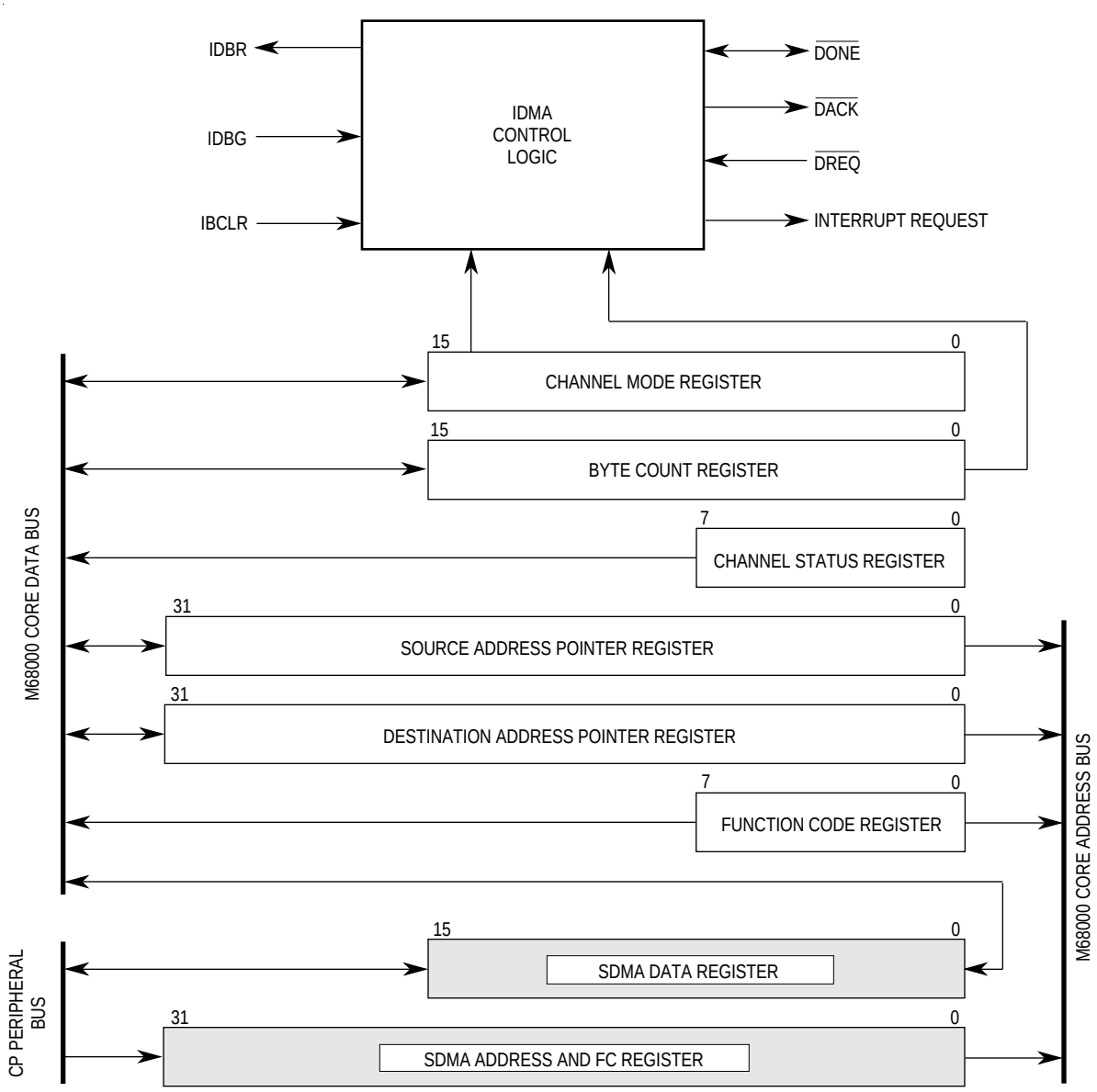

Figure 3-1. IDMA Controller Block Diagram

3.1.2 IDMA Registers (Independent DMA Controller)

The IDMA has six registers that define its specific operation. These registers include a 32-bit source address pointer register (SAPR), a 32-bit destination address pointer register

Table 3-3 indicates the interrupt levels available in both normal and dedicated modes. This table also shows the $\overline{IPL2}$–$\overline{IPL0}$ encoding that should be provided by external logic for each EXRQ interrupt level in normal mode. For the dedicated mode, this table shows the IMP input pins ($\overline{IRQ7}$, $\overline{IRQ6}$, and $\overline{IRQ1}$) that should be asserted by an external device according to the desired interrupt priority level.

Table 3-3. EXRQ and INRQ Prioritization

Priority Level	Normal Mode $\overline{IPL2}$–$\overline{IPL0}$	Dedicated Mode $\overline{IRQ7}$, $\overline{IRQ6}$, $\overline{IRQ1}$	Interrupt Source
7 (Highest)	000	$\overline{IRQ7}$	EXRQ
6	001	$\overline{IRQ6}$	EXRQ
5	010	*	EXRQ
4	*	*	INRQ
3	100	*	EXRQ
2	101	*	EXRQ
1 (Lowest)	110	$\overline{IRQ1}$	EXRQ

* Priority level not available to an external device in this mode.

3.2.2.2 INRQ Interrupt Source Priorities

Although all INRQ interrupts are presented at level 4, the interrupt controller further organizes interrupt servicing of the 15 INRQ interrupts according to the priorities illustrated in Table 3-4. The interrupt from the port B pin 11 (PB11) has the highest priority, and the interrupt from the port B pin 8 (PB8) has the lowest priority. A single interrupt priority within level 4 is associated with each table entry. The IDMA entry is associated with the general-purpose DMA channel only, and not with the SDMA channels that service the SCCs. Those interrupts are reported through each individual SCC channel or, in the case of a bus error, through the SDMA channels bus error entry.

Table 3-4. INRQ Prioritization within Interrupt Level 4

Priority Level	Interrupt Source Description	Multiple Interrupt Events
Highest	General-Purpose Interrupt 3 (PB11)	No
	General-Purpose Interrupt 2 (PB10)	No
	SCC1	Yes
	SDMA Channels Bus Error	No
	IDMA Channel	Yes
	SCC2	Yes
	Timer 1	Yes
	SCC3	Yes
	General-Purpose Interrupt 1 (PB9)	No
	Timer 2	Yes
	SCP	No
	Timer 3	No
	SMC1	No
	SMC2	No
	General-Purpose Interrupt 0 (PB8)	No
Lowest	Error	—

3.2.2.3 Nested Interrupts

The following rules apply to nested interrupts:

1. The interrupt controller responds to all EXRQ and INRQ interrupts based upon their assigned priority level. The highest priority interrupt request is presented to the M68000 core for servicing. After the vector number corresponding to this interrupt is passed to the core during an interrupt acknowledge cycle, an INRQ interrupt request is cleared in IPR. (EXRQ requests must be cleared externally.) The remaining interrupt

REF—Output Reference Event

The counter has reached the TRR value. The ORI bit in the TMR is used to enable the interrupt request caused by this event.

Bits 7–2—Reserved for future use.

3.5.2.6 General Purpose Timer Example

This section gives two examples on how to program the general purpose timers.

3.5.2.6.1 Timer Example 1

Generate an interrupt every 10 mS using the 20 MHz system clock.

1. Take the desired interrupt period and divide by the timer clock period to get an initial count value to calculate prescaler.

$$\frac{Tout}{Tin} = \frac{10ms}{\frac{1}{20MHz}} = Count = 200,000$$

2. To calculate the value for the clock divider, divide the count by 65536 (2^{16}).

$$\frac{Count}{65536} = Divider = 3.05176$$

3. The divider must be rounded up to the next integer value. A clock divider of 4 then changes the input timer period to Tin*4. A new count is calculated based on the new timer period, and this value will be written to the TRR. The prescaler in the TMR is equal to the clock divider minus 1 (or 4-1 = 3).

$$\frac{Tout}{Tin(Divider)} = \frac{10ms}{50ns(4)} = 50,000$$

4. Program the TRR to $C350 (= 50000 decimal).
5. Program the TMR to $031B (prescaler = 3, ORI =1 to enable interrupt, FRR = 1 to re-start counter after reference is reached, ICLK = 01 to use the master clock, and RST = 1 to enabled the timer).

Fine adjustments can be made to the timer by varying the TRR up or down.

3.5.2.6.2 Timer Example 2

Generate a 100 Hz square wave using the 20 MHz system clock. As in Timer Example 1, the period is 10 mS, so we can use the same Prescaler and Reference values. When OM is set, the TOUT pin only toggles when the reference value is reached. Therefore the reference value must be divided by two in order to generate two edges every 100 mS.

1. Program the Port B control register to change the port pin from a general purpose input pin to TOUT.
2. Program the TRR to $61A8 (= 50000/2).
3. Program the TMR to $321B (prescaler = 3, OM =1 to toggle TOUT, FRR = 1 to restart

When an external master desires to gain ownership, the standard M68000 bus arbitration protocol should be used:

1. Issue $\overline{BR}$ (to the IMP on-chip bus arbiter).
2. Wait for $\overline{BG}$ (from the IMP on-chip bus arbiter).
3. When $\overline{BG}$ is asserted, wait for the negation of both $\overline{AS}$ and $\overline{BGACK}$.
4. Assert $\overline{BGACK}$ and begin external master bus cycles.
5. Negate $\overline{BR}$ (to the IMP on-chip bus arbiter), causing $\overline{BG}$ to be negated by the IMP on-chip bus arbiter.
6. Negate $\overline{BGACK}$ after the external master bus cycles have completed.

This protocol is also followed by the on-chip bus masters (IDMA, SDMA, and DRAM refresh) except that they request the bus internally from the on-chip bus arbiter.

In the disable CPU mode, the IMP makes requests for the bus rather than granting the bus. In such a system, the IMP functions as an external master, and the external processor (e.g., an MC68030) need not assert $\overline{BGACK}$ as it accesses the IMP's on-chip RAM and registers.

NOTE

When the IMP's BUSW pin is low causing the M68000 core to operate as an MC68008, the $\overline{BGACK}$ signal should still be used in bus arbitration control. On the original MC68008, the $\overline{BGACK}$ signal was not available externally, and therefore could not be used.

3.8.6 Hardware Watchdog

The hardware watchdog logic is used to assert $\overline{BERR}$ and set HWT when a bus cycle is not terminated by $\overline{DTACK}$ and after a programmable number of clock cycles has elapsed. The hardware watchdog logic has a 10-bit downcounter and a 4-bit prescaler. When enabled, the watchdog timer commences counting clock cycles as $\overline{AS}$ is asserted (for internal or external bus masters). The count is terminated normally by the negation of $\overline{AS}$; however, if the count reaches zero before $\overline{AS}$ is negated, $\overline{BERR}$ will be asserted until $\overline{AS}$ is negated. The hardware watchdog will operate with internal as well as external bus masters.

The hardware watchdog logic uses four bits in the SCR.

HWDEN—Hardware Watchdog Enable

0 = The hardware watchdog is disabled.
1 = The hardware watchdog is enabled.

After system reset, this bit defaults to one to enable the hardware watchdog.

L1CLK	IDL clock; input to the IMP.
L1TXD	IDL transmit data; output from the IMP. Valid only for the bits that are supported by the IDL; three-stated otherwise.
L1RXD	IDL receive data; input to the IMP. Valid for the 20 bits of the IDL; ignored for other signals that may be present.
L1SY1	IDL SYNC signal; input to the IMP. This signal indicates that the 20 clock periods following the pulse designate the IDL frame.
L1RQ	Request permission to transmit on the D channel; output from the IMP.
L1GR	Grant permission to transmit on the D channel; input to the IMP.
SDS1	Serial data strobe 1
SDS2	Serial data strobe 2

NOTE

The IDL bus signals, L1TXD and L1RXD, require pull-up resistors in order to ensure proper operation with transceivers.

In addition to the 144-kbps ISDN 2B + D channels, IDL provides channels for maintenance and auxiliary bandwidth. The IDL bus has five channels:

B1	64-kbps Bearer Channel
B2	64-kbps Bearer Channel
D	16-kbps Signaling Channel
M	8-kbps Maintenance Channel (not required by IDL)
A	8-kbps Auxiliary Channel (not required by IDL)

The IMP supports all five channels of the IDL bus. The following table shows where each channel can be routed. The two B channels can be concatenated and routed to the same SCC channel.

IDL Channel	Serial Controllers
D	SCC1, SCC2, SCC3
B1	SCC1, SCC2, SCC3
B2	SCC1, SCC2, SCC3
M	SMC1
A	SMC2

The IMP supports the request-grant method for contention detection on the D channel. When the IMP has data to transmit on the D channel, it asserts L1RQ. The physical layer device monitors the physical layer bus for activity on the D channel and indicates that the channel is free by asserting L1GR. The IMP samples the L1GR signal when L1SY1 is asserted. If L1GR is high (active), the IMP transmits the first zero of the opening flag in the first bit of the D channel. If a collision is detected on the D channel, the physical layer device negates L1GR. The IMP then stops its transmission and retransmits the frame when L1GR is asserted again. This is handled automatically for the first two buffers of the frame.

SDC2—Serial Data Strobe Control 2

0 = SDS2 signal is asserted during the B2 channel
1 = SDS1 signal is asserted during the B2 channel

SDC1—Serial Data Strobe Control 1

0 = SDS1 signal is asserted during the B1 channel
1 = SDS2 signal is asserted during the B1 channel

B2RB, B2RA—B2 Channel Route in IDL/GCI Mode or CH-3 Route in PCM Mode

00 = Channel not supported
01 = Route channel to SCC1
10 = Route channel to SCC2 (if MSC2 is cleared)
11 = Route channel to SCC3 (if MSC3 is cleared)

B1RB, B1RA—B1 Channel Route in IDL/GCI Mode or CH-2 Route in PCM Mode

00 = Channel not supported
01 = Route channel to SCC1
10 = Route channel to SCC2 (if MSC2 is cleared)
11 = Route channel to SCC3 (if MSC3 is cleared)

DRB, DRA—D-Channel Route in IDL/GCI Mode or CH-1 Route in PCM Mode

00 = Channel not supported
01 = Route channel to SCC1
10 = Route channel to SCC2 (if MSC2 is cleared)
11 = Route channel to SCC3 (if MSC3 is cleared)

MSC3—SCC3 Connection

0 = SCC3 is connected to the multiplexed serial interface (PCM, IDL, or GCI) chosen in MS1–MS0. NMSI3 pins are all available for other purposes.
1 = SCC3 is not connected to a multiplexed serial interface but is connected directly to the NMSI3 pins or SCP pins or is not used. The choice of general-purpose I/O port pins versus SCC3 functions is made in the port A control register. The choice of SCP pins versus SCC3 functions is made in the SPMODE register.

MSC2—SCC2 Connection

0 = SCC2 is connected to the multiplexed serial interface (PCM, IDL, or GCI) chosen in MS1–MS0. NMSI2 pins are all available for other purposes.
1 = SCC2 is not connected to a multiplexed serial interface but is either connected directly to the NMSI2 pins or not used. The choice of general-purpose I/O port pins versus SCC2 functions is made in the port A control register.

MS1—MS0—Mode Supported

00 = NMSI Mode
When working in NMSI mode, SCC1 is connected directly to the seven NMSI1 pins (RXD1, TXD1, RCLK1, TCLK1, $\overline{\text{CD1}}$, $\overline{\text{CTS1}}$, and $\overline{\text{RTS1}}$). SCC2 functions can be routed to port A as NMSI functions or configured instead as PA6–PA0. Four of the SCC3 functions can be routed to port A or retained as PA11–PA8. The other

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 lie beyond this entry.

NOTE

In tables with eight receive control characters, E is always zero.

R—Reject Character

0 = The character is not rejected but is written into the receive buffer. The buffer is then closed, and a new receive buffer is used if there is more data in the message. A maskable interrupt is generated in the RX bit of the UART event register.

1 = If this character is recognized, it will not be written to the receive buffer. Instead, it is written to the RCCR, and a maskable interrupt is generated in the CCR bit in the UART event register. The current buffer is not closed when a control character is received with R set.

Transmission of out-of-sequence characters is also supported and is normally used for the transmission of flow control characters such as XON or XOFF. This is performed using the last (eighth) entry in the UART control characters table. The UART will poll this character whenever the transmitter is enabled for UART operation: during freeze, during buffer transmission, and when no buffer is ready for transmission. The character is transmitted at a higher priority than the other characters in the transmit buffer (if any), but does not pre-empt characters already in the transmit FIFO.

CHARACTER8—Control Character Value

The eighth entry in the UART control characters table is defined as follows:

E—Empty

Must be one to use this entry as a flow control transmission character. To use this entry instead as a receive control characters entry, this E bit (and all other E bits in the table) should be zero.

R—Reject

Must be zero to use this entry as a flow control transmission character. For a receive control characters entry, it maintains its functionality as previously defined.

REA—Ready

This bit is set by the M68000 core when the character is ready for transmission and will remain one while the character is being transmitted. The CP clears this bit after transmission.

I—Interrupt

If set, the M68000 core will be interrupted when this character has been transmitted. (The TX bit will be set in the UART event register.)

I—Interrupt

0 = No interrupt is generated after this buffer has been filled.
1 = The RX bit in the UART event register will be set when this buffer has been completely filled by the CP, indicating the need for the M68000 core to process the buffer. The RX bit can cause an interrupt.

The following bits contain status information written by the CP after it has finished receiving data in the associated data buffer.

C—Control Character

0 = This buffer does not contain a control character.
1 = This buffer contains a user-defined control character in the last byte location.

A—Address

0 = The buffer contains data only.
1 = When working in nonautomatic multidrop mode (UM1–UM0 = 01), this bit indicates that the first byte of this buffer contains an address byte. The address comparison should be implemented in software. In automatic multidrop mode, this bit indicates that the BD contains a message received immediately following an address recognized in UADDR1 or UADDR2. This address is not written into the receive buffer.

M—Address Match

This bit is meaningful only if the A bit (bit 10) is set and UM1–UM0 = 11 in the UART mode register. Following an address match, this bit defines which address character matched the user-defined address character, enabling the UART to receive the data.

0 = The address-matched user-defined UADDR2
1 = The address-matched user-defined UADDR1

ID—Buffer Closed on Reception of Idles

The buffer was closed due to the reception of the programmable number of consecutive IDLE sequences (defined in MAX_IDL).

Bits 7–6, 2—Reserved for future use.

BR—Break Received

A break sequence was received while receiving data into this buffer.

FR—Framing Error

A character with a framing error was received and is located in the last byte of this buffer. A framing error is detected by the UART controller when no stop bit is detected in the receive data string.

PR—Parity Error

A character with a parity error was received and is located in the last byte of this buffer.

OV—Overrun

A receiver overrun occurred during message reception.

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

Memory Map for the placement of the three SCC parameter RAM areas and Table 4-5 for the other parameter RAM values.

Table 4-9. BISYNC Specific Parameter RAM

Address	Name	Width	Description
SCC Base + 9C SCC Base + 9E SCC Base + A0 # SCC Base + A2 SCC Base + A4 # SCC Base + A6 SCC Base + A8	RCRC CRCC PRCRC TCRC PTCRC RES RES	Word Word Word Word Word Word Word	Temp Receive CRC CRC Constant Preset Receiver CRC 16/LRC Temp Transmit CRC Preset Transmitter CRC 16/LRC Reserved Reserved
SCC Base + AA # SCC Base + AC # SCC Base + AE #	PAREC BSYNC BDLE	Word Word Word	Receive Parity Error Counter BISYNC SYNC Character BISYNC DLE Character
SCC Base + B0 # SCC Base + B2 # SCC Base + B4 # SCC Base + B6 # SCC Base + B8 # SCC Base + BA # SCC Base + BC # SCC Base + BE #	CHARACTER1 CHARACTER2 CHARACTER3 CHARACTER4 CHARACTER5 CHARACTER6 CHARACTER7 CHARACTER8	Word Word Word Word Word Word Word Word	CONTROL Character 1 CONTROL Character 2 CONTROL Character 3 CONTROL Character 4 CONTROL Character 5 CONTROL Character 6 CONTROL Character 7 CONTROL Character 8

Initialized by the user (M68000 core).

The M68000 core configures each SCC to operate in one of four protocols by the MODE1–MODE0 bits in the SCC mode register. MODE1–MODE0 = 11 selects the BISYNC mode of operation. The SYN1–SYN2 synchronization characters are programmed in the data synchronization register (see 4.5.4 SCC Data Synchronization Register (DSR)).

The BISYNC controller uses the same basic data structure as the other protocol controllers. Receive and transmit errors are reported through their respective BDs. The status of the line is reflected in the SCC status register, and a maskable interrupt is generated upon each status change.

There are two basic ways of handling the BISYNC channels. First, data may be inspected on a per-byte basis, with the BISYNC controller interrupting the M68000 core upon receipt of every byte of data. Second, the BISYNC controller may be operated so that software is only necessary for handling the first two to three bytes of data; subsequent data (until the end of the block) can be handled by the BISYNC controller without interrupting the M68000 core. See 4.5.13.14 Programming the BISYNC Controllers for more information.

4.5.13.4 BISYNC 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 using 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 STOP TRANSMIT command aborts transmission after the contents of the FIFO are transmitted (up to three bytes) without waiting until the end of the buffer is reached. The TBD# is not advanced. SYNC characters consisting of SYNC-SYNC or DLE-SYNC pairs (according to the transmitter mode) will be continually transmitted until transmission is re-enabled by issuing the RESTART TRANSMIT command. The STOP TRANSMIT com-

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.

I—Interrupt

0 = No interrupt is generated after this buffer has been used.
1 = When this buffer has been closed by the transparent controller, the RX bit in the transparent event register will be set, which can cause an interrupt.

The following status bits are written by the CP after the received data has been placed into the associated data buffer.

Bits 11–2—Reserved for future use. Should be written with zero by the user.

OV—Overrun

A receiver overrun occurred during reception.

CD—Carrier Detect Lost

The carrier detect signal was negated during buffer reception.

Data Length

The data length is the number of octets that the CP has written into this BD's data buffer. It is written only once by the CP as the buffer is closed.

NOTE

The actual buffer size should be greater than or equal to the MRBLR.

Rx Buffer Pointer

The receive buffer pointer, which always points to the first location of the associated data buffer, must be even. 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.

4.5.16.9 Transparent 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 4-43.

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

Figure 4-43. Transparent 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.

Table 5-1. Signal Definitions

Chip Select	CS3–CS0	4
Testing	$\overline{FRZ}$ (2 Spare)	3
V_{DD}		8
GND		13

All pins except EXTAL, CLKO, and the layer 1 interface pins in IDL mode support TTL levels. EXTAL, when used as an input clock, needs a CMOS level. CLKO supplies a CMOS level output. The IDL interface is specified as a CMOS electrical interface.

All outputs (except CLKO and the GCI pins) drive 130 pF. CLKO is designed to drive 50 pF. The GCI output pins drive 150 pF.

5.2 POWER PINS

The IMP has 21 power supply pins. Careful attention has been paid to reducing IMP noise, potential crosstalk, and RF radiation from the output drivers. Inputs may be +5 V when V_{DD} is 0 V without damaging the device.

- V_{DD} (8)—There are 8 power pins.
- GND (13)—There are 13 ground pins.

NOTE

The Input High Voltage and Input Low Voltage for EXTAL and the values for power are specified in the DC Electrical Characteristics.A valid clock signal oscillates between a low voltage of between V_{SS}-0.3 and .6 volts and a high voltage of between 4.0 and V_{DD} volts. This EXTAL signal must be present within 20 mS after V_{DD} reaches its minimum specified level of 4.5 volts.

6.19 AC ELECTRICAL SPECIFICATIONS—IDL TIMING (All timing measurements, unless otherwise specified, are referenced to the L1CLK at 50% point of V_{DD}) (see Figure 6-20).

		16.67 MHz		20 MHz		25 MHz		
Num.	**Characteristic**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
260	L1CLK (IDL Clock) Frequency (see Note 1)	—	6.66	—	8	—	10	MHz
261	L1CLK Width Low	55	—	45	—	37	—	ns
262	L1CLK Width High (see Note 3)	P+10	—	P+10	—	P+10	—	ns
263	L1TXD, L1RQ, SDS1–SDS2 Rising/Falling Time	—	20	—	17	—	14	ns
264	L1SY1 (sync) Setup Time (to L1CLK Falling Edge)	30	—	25	—	20	—	ns
265	L1SY1 (sync) Hold Time (from L1CLK Falling Edge)	50	—	40	—	34	—	ns
266	L1SY1 (sync) Inactive Before 4th L1CLK	0	—	0	—	0	—	ns
267	L1TxD Active Delay (from L1CLK Rising Edge)	0	75	0	65	0	50	ns
268	L1TxD to High Impedance (from L1CLK Rising Edge) (see Note 2)	0	50	0	42	0	34	ns
269	L1RxD Setup Time (to L1CLK Falling Edge)	50	—	42	—	34	—	ns
270	L1RxD Hold Time (from L1CLK Falling Edge)	50	—	42	—	34	—	ns
271	Time Between Successive IDL syncs	20	—	20	—	20	—	L1CLK
272	L1RQ Valid before Falling Edge of L1SY1	1	—	1	—	1	—	L1CLK
273	L1GR Setup Time (to L1SY1 Falling Edge)	50	—	42	—	34	—	ns
274	L1GR Hold Time (from L1SY1 Falling Edge)	50	—	42	—	34	—	ns
275	SDS1–SDS2 Active Delay from L1CLK Rising Edge	10	75	10	65	7	50	ns
276	SDS1–SDS2 Inactive Delay from L1CLK Falling Edge	10	75	10	65	7	50	ns

NOTES:

1. The ratio CLKO/L1CLK must be greater than 2.5/1.
2. High impedance is measured at the 30% and 70% of V_{DD} points, with the line at $V_{DD}/2$ through 10K in parallel with130 pF.
3. Where P = 1/CLKO. Thus, for a 16.67-MHz CLKO rate, P = 60 ns.

```
          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?
```

```
          CMPI.W    #0,D2             ;If they are set
          BNE.B     TX_ INT           ;Handle Transmitter's interrupt
* The handling of other events, e.g., CTS, CD IDL, BSY, is left  to
* the users as desired.
OthrlNT   MOVE.W #$2000,1SR           ;Clear SCC1 bit in ISR
          RTE

***************Receiver portion of SCC1 interrupt routine***************
* This routine handles received (nonempty) BD: set data length = 0,
* clear status bits, set empty =1, and update PRD
* Clear the identified events as soon as possible, so no lost
* events occur during the interrupt handling
RX_INT    MOVE.B    #9,SCCE1          ;Clear RXF and RXB in SCCE1
* While Not-Empty continue to process the next Rx BD, Else Exit.
NxtPRD    BTST.B    #EMPTY,ST_BD(A3)  ;Test PRD -, Empty Bit
*         BNE.B     EXIT_RX           ;Don't need to process if the
*                                     ;Rx BD is still empty.
*** Check status in RXBD for erratic events ***
* If status bits are all 0 then continue, else SHUTDOWN the receiving
* process. This in turn shuts down the whole program, since all of
* Rx BDs will soon be unavailable (all BDs Empty = 8). Thus, the
* status of this BD will be saved for examination later.
          CMPI.B    #0,SS_BD(A3)      ;Check status bits
          BNE.B     EXIT_RX
* Status bits are all 0
          CLR.W     LN_BD(A3)         ;data length = 0
          CLR.B     SS_BD(A3)         ;Clear out all status bits
          BSET.B    #EMPTY,ST_BD(A3)  ;Empty = 1
          BTST.B    #WRAP,ST_BD(A3)   ;Test Wrap bit
          BNE.B     Wrap_R
          ADDQ.W    #SZ_BD,A3         ;Increment PRD to next BD
          BRA.B     NxtPRD            ;Back to while loop
Wrap_R    LEA.L     RXBD_01,A3        ;Wrap back to the first Rx BD
          BRA.B     NxtPRD            ;Back to the while loop
EXIT_RX   JMP       CK_TX             ;Exit receiver potion of the handler

***************Conflrmer portion of SCC1 Interrupt routine ***************
* This routine handles transmitted (Not-ready) BD: set data length = 0,
* clear status bits, set ready = 1, and update CTD.
* Same as the Rx Interrupt handler, the first thing to do is to
* clear the identified events
TX_INT    MOVE.B    #$1 2,SCCE1       ;Clear TXF and TXB in SCCE1
* While Not-Ready continue to process the next Rx BD, Else Exit.
* The Ready bit should be cleared by the CP
NxtCTD    BTST.B    #READY,ST BD(A1)  ;Test PRD-> Ready Bit
          BNE.B     EXIT_TX           ;Don't need to process if the
*                                     ;Tx BD is Ready.
* Check data length, length must be > 0 to continue the confirming process
          CMPI.W    #0,LN_BD(A1)      ;Test CTD –> data length
          BEQ.B     EXIT_TX
          CLR.W     LN_BD(A1)         ;data length = 0

***Check status in TXBD for erratic events ***
* If status bits are all 0 then continue, else SHUTDOWN the confirming
* process. This in turn shuts down the whole program, since soon
```

Figure D-25 shows how $\overline{CTS}$ can be used in the NMSI transmit case. NTSYN and EXSYN are set to enable transparent mode. Instead of software operation for $\overline{CTS}$ and $\overline{CD}$, normal (automatic) operation is chosen. $\overline{RTS}$ is asserted when the transmit FIFO is full. From then on, data is held off until $\overline{CTS}$ is sampled low. From that sample point, there is a 3.5 TCLK delay before the first bit of the data buffer is transmitted. Ones are transmitted until the first bit of the data buffer is transmitted.

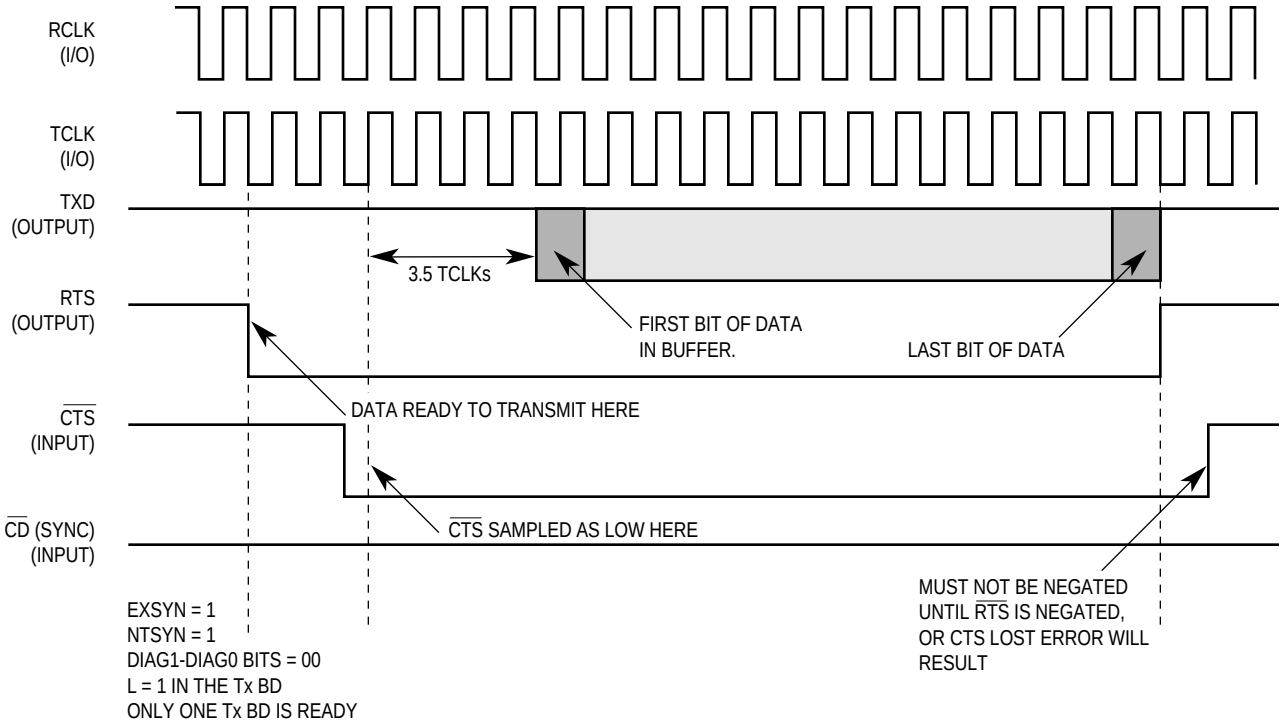

Figure D-25. Using $\overline{CTS}$ In the NMSI Transmit Case

In the case shown in Figure D-25, it is important that $\overline{CTS}$ not go high for the duration of the buffer transmission. If multiple buffers are all ready with their L bits cleared, transmission of frames will continue back-to-back. If $\overline{CTS}$ negated during any of these buffers, transmission will cease, and that buffer will report a $\overline{CTS}$ lost condition. Ones will be transmitted at that time. Once a restart transmit command is given, transmission of the next buffer can begin once $\overline{CTS}$ is reasserted.

Once $\overline{CTS}$ deasserts after $\overline{RTS}$, the $\overline{RTS}$-$\overline{CTS}$ protocol can begin again as soon as the next buffer is made ready, but a minimum of 17 idle bits will occur between frames, regardless of how soon $\overline{CTS}$ is reasserted. Remember that when EXSYN is set, $\overline{CD}$ (sync) must be low for transmission to begin. In this case, it is grounded; whereas, in the following case, EXSYN is actively switching.

Figure D-26 shows how $\overline{CD}$ (sync) can be used to control transmission. EXSYN and NTSYN are once again set to enable transparent mode, and the L bit is set. Since software operation mode (DIAG1 = 1 and DIAG0 = 1) is chosen, the $\overline{CTS}$ pin value is ignored. Once $\overline{CD}$ (sync) is latched low, data begins transmission in 6.5 TCLKs. Notice that the rising edge of $\overline{CD}$ (sync) and subsequent falling edges of $\overline{CD}$ (sync) (not shown) have no effect, since synchronization has already been achieved.