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	25MHz
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/product-detail/nxp-semiconductors/mc68302eh25cb1

identical. When any SCC, SCP, or SMC channel buffer descriptors or parameters are not used, their parameter RAM area can be used for additional memory. For detailed information about the use of the buffer descriptors and protocol parameters in a specific protocol, see 4.5 Serial Communication Controllers (SCCs). Base + 67E contains the MC68302 revision number. Revision A parts (mask 1B14M) correspond to the value $0001. Revision B parts (mask 2B14M and 3B14M which are described in this manual) correspond to the value $0002. Revision C and D parts have revision number $0003.

Table 2-8. Parameter RAM

Address	Width	Block	Description
Base + 400	4 Word	SCC1	Rx BD 0
Base + 408	4 Word	SCC1	Rx BD 1
Base + 410	4 Word	SCC1	Rx BD 2
Base + 418	4 Word	SCC1	Rx BD 3
Base + 420	4 Word	SCC1	Rx BD 4
Base + 428	4 Word	SCC1	Rx BD 5
Base + 430	4 Word	SCC1	Rx BD 6
Base + 438	4 Word	SCC1	Rx BD 7
Base + 440	4 Word	SCC1	Tx BD 0
Base + 448	4 Word	SCC1	Tx BD 1
Base + 450	4 Word	SCC1	Tx BD 2
Base + 458	4 Word	SCC1	Tx BD 3
Base + 460	4 Word	SCC1	Tx BD 4
Base + 468	4 Word	SCC1	Tx BD 5
Base + 470	4 Word	SCC1	Tx BD 6
Base + 478	4 Word	SCC1	Tx BD 7
Base + 480 • • • Base + 4BF		SCC1 SCC1	Specific Protocol Parameters
Base + 4C0 • • • Base + 4FF			Reserved (Not Implemented)
Base + 500	4 Word	SCC2	Rx BD 0
Base + 508	4 Word	SCC2	Rx BD 1
Base + 510	4 Word	SCC2	Rx BD 2
Base + 518	4 Word	SCC2	Rx BD 3
Base + 520	4 Word	SCC2	Rx BD 4
Base + 528	4 Word	SCC2	Rx BD 5
Base + 530	4 Word	SCC2	Rx BD 6
Base + 538	4 Word	SCC2	Rx BD 7
Base + 540	4 Word	SCC2	Tx BD 0
Base + 548	4 Word	SCC2	Tx BD 1
Base + 550	4 Word	SCC2	Tx BD 2
Base + 558	4 Word	SCC2	Tx BD 3
Base + 560	4 Word	SCC2	Tx BD 4
Base + 568	4 Word	SCC2	Tx BD 5
Base + 570	4 Word	SCC2	Tx BD 6/DRAM Refresh
Base + 578	4 Word	SCC2	Tx BD 7/DRAM Refresh

The IDL interface supports the CCITT I.460 recommendation for data rate adaptation. The IDL interface can access each bit of the B channel as an 8-kbps channel. A serial interface mask register (SIMASK) for the B channels specifies which bits are supported by the IDL interface. The receiver will support only the bits enabled by SIMASK. The transmitter will transmit only the bits enabled by the mask register and will three-state L1TXD otherwise.

Refer to Figure 4-6 for an example of supporting two bits in the B1 channel and three bits in the B2 channel.

4.4.2 GCI Interface

The normal mode of the GCI (also known as ISDN-Oriented Modular rev 2.2 (IOM2)) ISDN bus is fully supported by the IMP. The IMP also supports channel 0 of the Special Circuit Interface T (SCIT) interface, and in channel 2 of SCIT, supports the D channel access control for S/T interface terminals, using the command/indication (C/I) field. The IMP does not support the Telecom IC (TIC) bus.

The GCI bus consists of four lines: two data lines, a clock, and a frame synchronization line. Usually an 8-kHz frame structure defines the various channels within the 256-kbps data rate as indicated in Figure 4-8. However, the interface can also be used in a multiplexed frame structure on which up to eight physical layer devices multiplex their GCI channels. L1SY1 must provide the channel SYNC. In this mode, the data rate would be 2048 kbps.

The GCI clock rate is twice the data rate. The clock rate for the IMP must not exceed the ratio of 1:2.5 serial clock to parallel clock. Thus, for a 16.67-MHz system clock, the serial clock rate must not exceed 6.67 MHz.

The IMP also supports another line for D-channel access control—the L1GR line. This signal is not part of the GCI interface definition and may be used in proprietary interfaces.

NOTE

When the L1GR line is not used, it should be pulled high. The IMP has two data strobe lines (SDS1 and SDS2) for selecting either or both of the B1 and B2 channels and the data rate clock (L1CLK). These signals are used for interfacing devices that do not support the GCI bus. They are configured with the SIMASK register and are active only for bits that are not masked.

The monitor channel is used to transfer data between layer-1 devices and the control unit (i.e., the M68000 core). The command/indication channel is used to control activation/deactivation procedures or for the switching of test loops by the control unit.

The IMP supports all five channels of the GCI channel 0. The following table shows where each channel can be routed. The two B channels can be concatenated and routed to the same SCC channel.

GCI Channel 0	Serial Controllers
D	SCC1, SCC2, SCC3
B1	SCC1, SCC2, SCC3
B2	SCC1, SCC2, SCC3
M	SMC1
C/I	SMC2

The GCI interface supports the CCITT I.460 recommendation for data rate adaptation. The GCI interface can access each bit of the B channel as an 8-kbps channel. The mask register (SIMASK) for the B channels specifies which bits are supported by the GCI interface. The receiver will receive only the bits that are enabled by SIMASK; the transmitter will transmit only the bits that are enabled by SIMASK and will not drive the L1TXD pin otherwise (L1TXD in GCI mode is an open-drain output).

The IMP supports contention detection on the D channel. When the IMP has data to transmit on the D channel, it checks bit 4 of the SCIT C/I channel 2. The physical layer device monitors the physical layer bus for activity on the D channel and indicates with this bit that the channel is free. If a collision is detected on the D channel, the physical layer device sets bit 4 of C/I channel 2 to logic high. The IMP then aborts its transmission and retransmits the frame when this bit is asserted again. This procedure is handled automatically for the first two buffers of a frame. The L1GR line may also be used for access to the S interface D channel. This signal is checked by the IMP, and the physical layer device should indicate that the S interface D channel is free by asserting L1GR.

In the deactivated state, the clock pulse is disabled, and the data line is a logic one. The layer-1 device activates the IMP by enabling the clock pulses and by an indication in the channel 0 C/I channel. The IMP will then report to the M68000 core by a maskable interrupt that a valid indication is in the SMC2 receive buffer descriptor.

When the M68000 core activates the line, it sets SETZ in the serial interface mode (SIMODE) register, causing the data output from L1TXD to become a logic zero. Code 0 (command timing TIM) will be transmitted on channel 0 C/I channel to the layer-1 device until the SETZ is reset. The physical layer device will resume transmitting the clock pulses and will give an indication in the channel 0 C/I channel. The M68000 core should reset SETZ to enable data output.

4.4.3 PCM Highway Mode

In PCM highway mode, one, two, or all three SCCs can be multiplexed together to support various time-division multiplexed interfaces. PCM highway supports the standard T1 and CEPT interfaces as well as user-defined interfaces. In this mode, the NMSI1 pins have new names and functions (see Table 4-2).

tools are 1) the ready bit in the transmit buffer descriptor, 2) the ENT bit, 3) the STOP TRANSMIT command, 4) the RESTART TRANSMIT command, and 5) the FRZ bit in the SCM (UART mode only).

ENR—Enable Receiver

When ENR is set, the receiver is enabled. When it is cleared, the receiver is disabled, and any data in the receive FIFO is lost. If ENR is cleared during data reception, the receiver aborts the current character. ENR may be set or cleared regardless of whether serial clocks are present. To restart reception, the ENTER HUNT MODE command should be issued before ENR is set again.

ENT—Enable Transmitter

When ENT is set, the transmitter is enabled; when ENT is cleared, the transmitter is disabled. If ENT is cleared, the transmitter will abort any data transmission, clear the transmit data FIFO and shift register, and force the TXD line high (idle). Data already in the transmit shift register will not be transmitted. ENT may be set or cleared regardless of whether serial clocks are present.

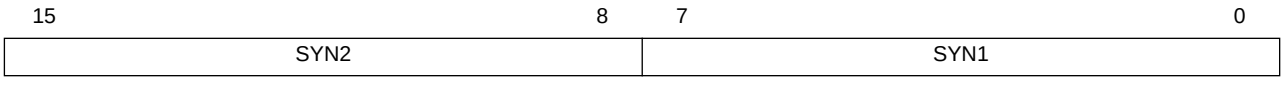
The STOP TRANSMIT command additionally aborts the current frame and would normally be given to the channel before clearing ENT. The command does not clear ENT automatically. In a similar manner, to restart transmission, the user should issue the RESTART TRANSMIT command and then set ENT. The command register is described in 4.3 Command Set. The specific actions taken with each command vary somewhat according to protocol and are discussed in each protocol section.

MODE1–MODE0—Channel Mode

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

4.5.4 SCC Data Synchronization Register (DSR)

Each DSR is a 16-bit, memory-mapped, read-write register. DSR specifies the pattern used in the frame synchronization procedure of the SCC in the synchronous protocols. In the UART protocol it is used to configure fractional stop bit transmission. After reset, the DSR defaults to $7E7E (two FLAGs); thus, no additional programming is necessary for the HDLC protocol. For BISYNC, DDCMP, and V.110, the contents of the DSR should be written before the channel is enabled. Note that for the DDCMP, SYN1 must equal SYN2 must equal DSYN1 for proper operation.



NOTE

The DSR register has no relationship to the RS-232 signal “data set ready,” which is also abbreviated DSR.

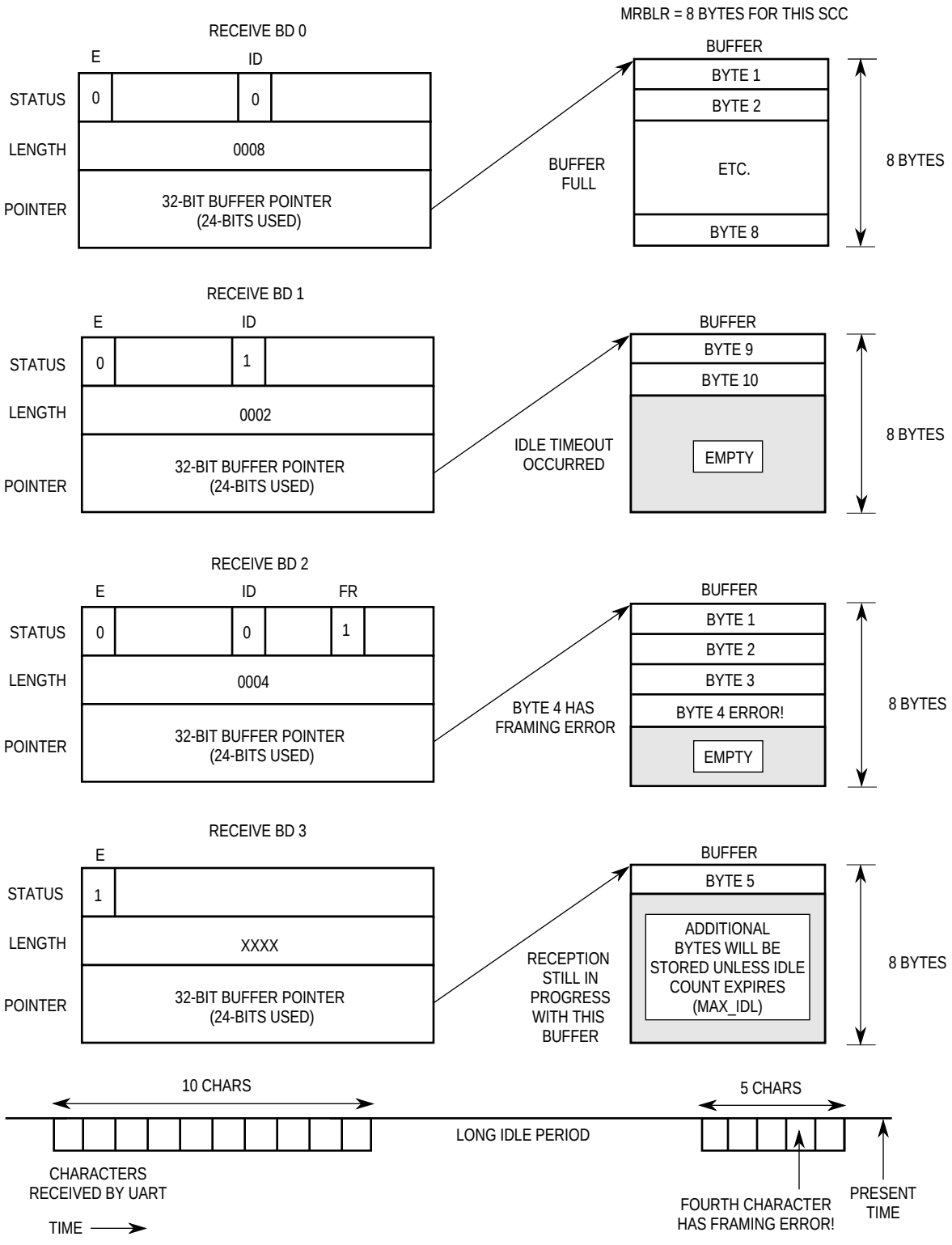


Figure 4-21. UART Rx BD Example

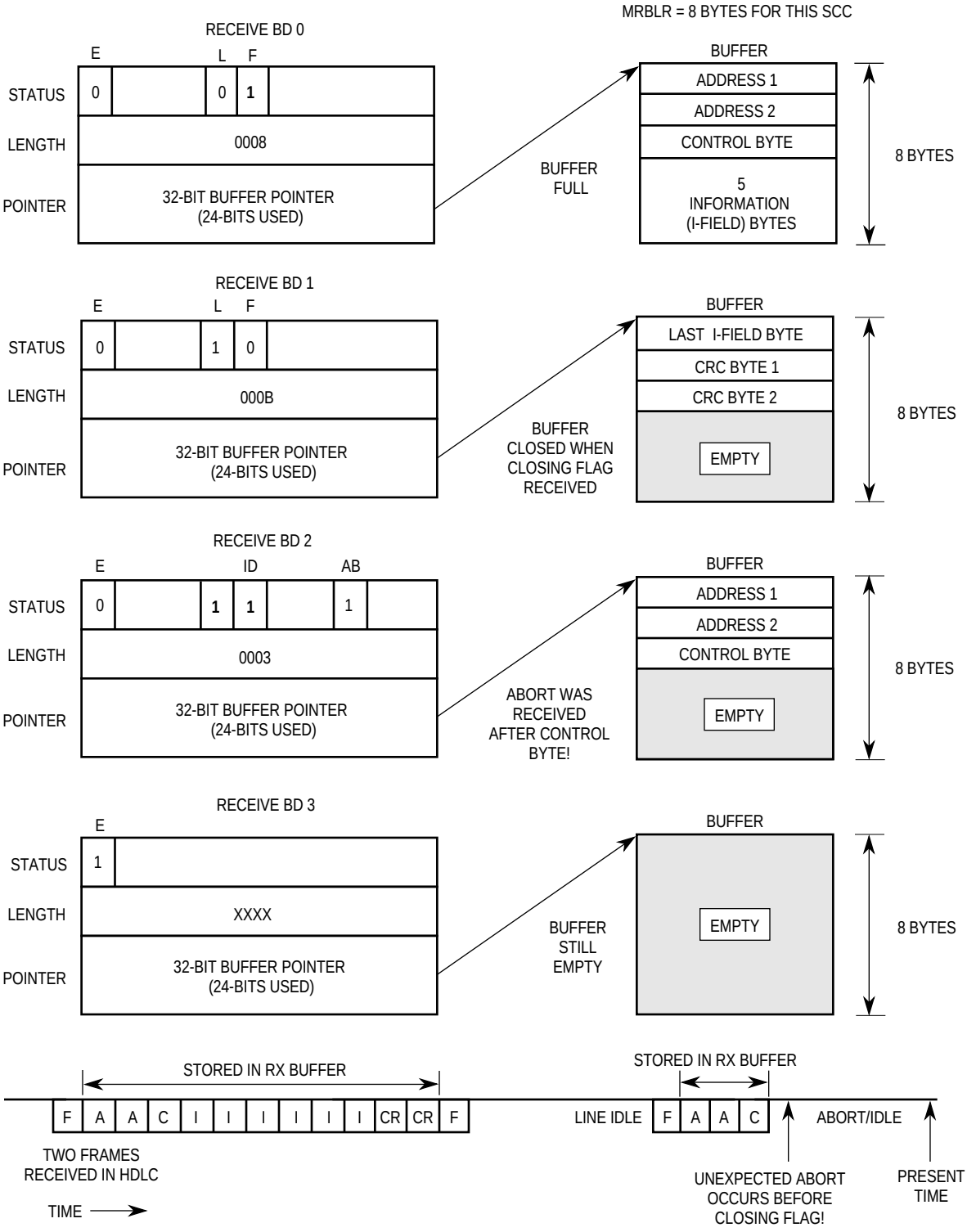


Figure 4-27. HDLC Receive BD Example

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.

By setting its SCC mode register (SCM), any of the SCC channels may be configured to function as a DDCMP controller. The DDCMP link can be either synchronous (by programming the MODE1–MODE0 bits of the SCC mode register to DDCMP) or asynchronous (by programming the MODE1–MODE0 bits of the SCC mode register to asynchronous and setting the DDCMP bit in the UART mode register). The DDCMP controller handles the basic functions of the DDCMP protocol in both cases.

The SCC in DDCMP mode can work in either IDL, GCI, PCM highway, or NMSI interfaces. When the SCC is used with a modem interface (NMSI), the serial outputs are connected directly to the external pins. The modem interface uses seven dedicated pins: transmit data (TXD), receive data (RXD), receive clock (RCLK), transmit clock (TCLK), carrier detect (CD), clear to send (CTS), and request to send (RTS). Other modem lines can be supported through the parallel I/O pins.

The DDCMP controller consists of separate transmit and receive sections whose operations are asynchronous with the M68000 core and may be either synchronous or asynchronous with respect to the other SCCs. Each clock can be supplied either from the baud rate generator or externally. More information on the baud rate generator is available in 4.5.2 SCC Configuration Register (SCON).

The DDCMP controller key features are as follows:

- Synchronous or Asynchronous DDCMP Links Supported
- Flexible Data Buffers
- Four Address Comparison Registers with Mask
- Automatic Frame Synchronization
- Automatic Message Synchronization by Searching for SOH, ENQ, or DLE
- CRC16 Generation/Checking
- NRZ/NRZI Data Encoding
- Maintenance of Four 16-Bit Error Counters

4.5.14.1 DDCMP Channel Frame Transmission Processing

The DDCMP transmitter is designed to work with almost no intervention from the M68000 core (see Figure 4-35).

When the M68000 core enables the DDCMP transmitter and the link is synchronous, it starts transmitting SYN1–SYN2 pairs (programmed in the data synchronization register) or IDLEs as determined in the DDCMP mode register. The DDCMP controller polls the first buffer descriptor (BD) in the channel's transmit BD table. When there is a message to transmit, the DDCMP controller fetches the data from memory and starts transmitting the message (after first transmitting the SYN1–SYN2 pair when the link is synchronous).

Monitor Channel Protocol

In this mode, SMC1 transmits the data and handles the A and E control bits according to the GCI monitor channel protocol. When using the monitor channel protocol, the user may issue the TIMEOUT command to solve deadlocks in case of bit errors in the A and E bit positions on data line. The IMP will transmit an abort on the E bit.

SMC1 Reception

The SMC1 receiver can be programmed to work in one of two modes:

Transparent Mode

In this mode, SMC1 receives the data, moves the A and E control bits transparently into the SMC1 receive BD, and generates a maskable interrupt. The SMC1 receiver discards new data when the M68000 core has not read the receive BD.

Monitor Channel Protocol

In this mode, SMC1 receives data and handles the A and E control bits according to the GCI monitor channel protocol. When a received data byte is stored by the CP in the SMC1 receive BD, a maskable interrupt is generated.
When using the monitor channel protocol, the user may issue the TRANSMIT ABORT REQUEST command. The IMP will then transmit an abort request on the A bit.

SMC2 Controls the GCI Command/Indication (C/I) Channel

SMC2 Transmission

The M68000 core writes the data byte into the SMC2 Tx BD. SMC2 will transmit the data continuously on the C/I channel to the physical layer device.

SMC2 Reception

The SMC2 receiver continuously monitors the C/I channel. When a change in data is recognized and this value is received in two successive frames, it will be interpreted as valid data. The received data byte is stored by the CP in the SMC2 receive BD, and a maskable interrupt is generated.
The receive and transmit clocks are derived from the same physical clock (L1CLK) and are only active while serial data is transferred between the SMC controllers and the serial interface.
When SMC loopback mode is chosen, SMC transmitted data is routed to the SMC receiver. Transmitted data appears on the L1TXD pin, unless the SDIAG1–SDIAG0 bits in the SIMODE register are programmed to “loopback control” (see 4.4 Serial Channels Physical Interface).

4.7.2 SMC Programming Model

The operating mode of both SMC ports is defined by SMC mode, which consists of the lower eight bits of SPMODE. As previously mentioned, the upper eight bits program the SCP.

7	6	5	4	3	2	1	0
—	SMD3	SMD2	SMD1	SMD0	LOOP	EN2	EN1

$\overline{\text{DREQ}}$/PA13—DMA Request

This input is asserted by a peripheral device to request an operand transfer between that peripheral device and memory. In the cycle steal request generation mode, this input is edge-sensitive. In burst mode, it is level-sensitive.

$\overline{\text{DACK}}$/PA14—DMA Acknowledge

This output, asserted by the IDMA, signals to the peripheral that an operand is being transferred in response to a previous transfer request.

$\overline{\text{DONE}}$/PA15—DONE

This bidirectional, open-drain signal is asserted by the IDMA or by a peripheral device during any IDMA bus cycle to indicate that the data being transferred is the last item in a block. The IDMA asserts this signal as an output during a bus cycle when the byte count register is decremented to zero. Otherwise, this pin is an input to the IDMA to terminate IDMA operation.

5.17 IACK OR PIO PORT B PINS

The IACK or PIO port B pins are shown in Figure 5-14.

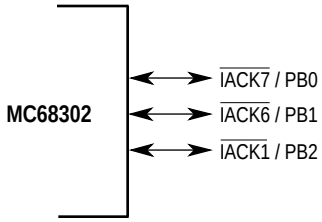


Figure 5-14. IACK or PIO Port B Pins

Each one of these three pins can be used either as an interrupt acknowledge signal or as a general-purpose parallel I/O port. Note that the IMP interrupt controller does not require the use of the IACK pins when it supplies the interrupt vector for the external source. The input buffers have Schmitt triggers.

$\overline{\text{IACK7}}$/PB0

$\overline{\text{IACK6}}$/PB1

$\overline{\text{IACK1}}$/PB2—Interrupt Acknowledge/Port B I/O

As $\overline{\text{IACK1}}$, $\overline{\text{IACK6}}$, and $\overline{\text{IACK7}}$, these active low output signals indicate to the external device that the MC68302 is executing an interrupt acknowledge cycle. The external device must then place its vector number on the lower byte of the data bus or use AVEC for autovectoring (unless internal vector generation is used).

5.18 TIMER PINS

The timer pins are shown in Figure 5-15.

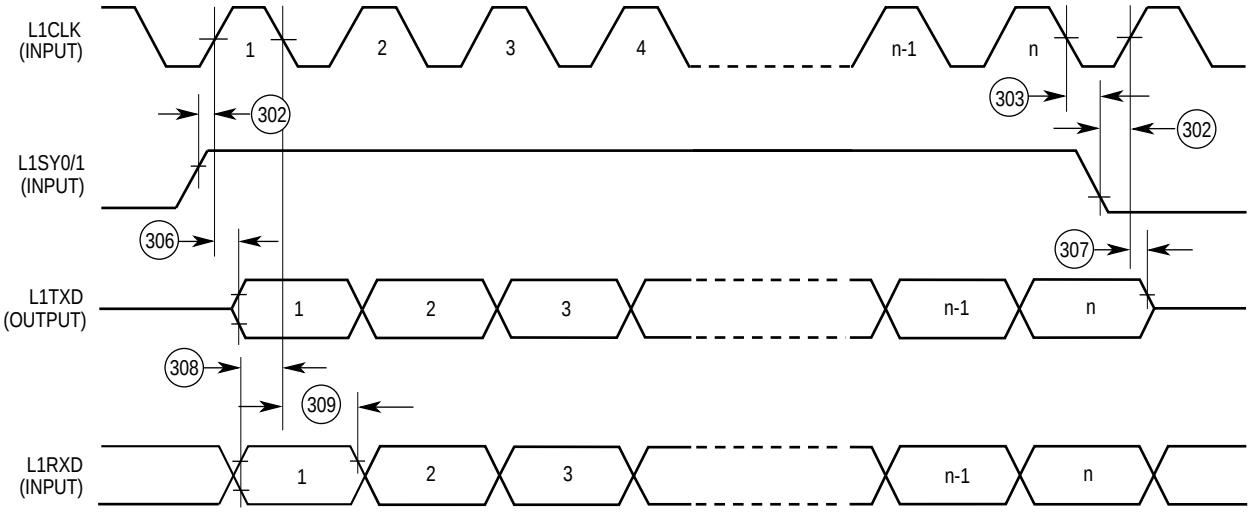
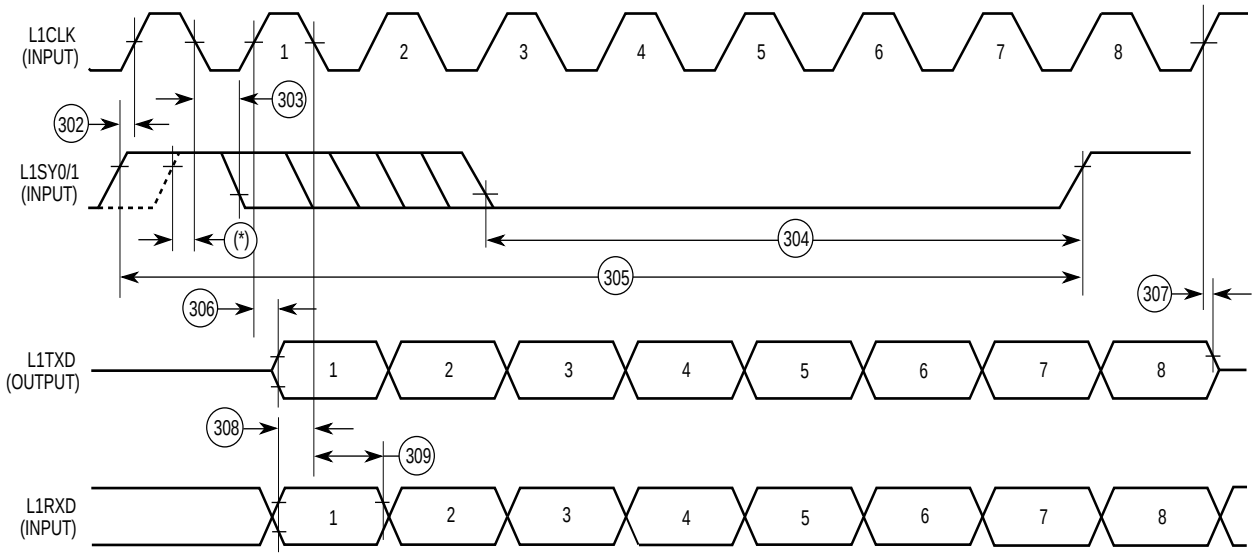


Figure 6-22. PCM Timing Diagram (SYNC Envelopes Data)



NOTE: (*) If L1SYn is guaranteed to make a smooth low to high transition (no spikes) while the clock is high, setup time can be defined as shown (min 20 ns).

Figure 6-23. PCM Timing Diagram (SYNC Prior to 8-Bit Data)

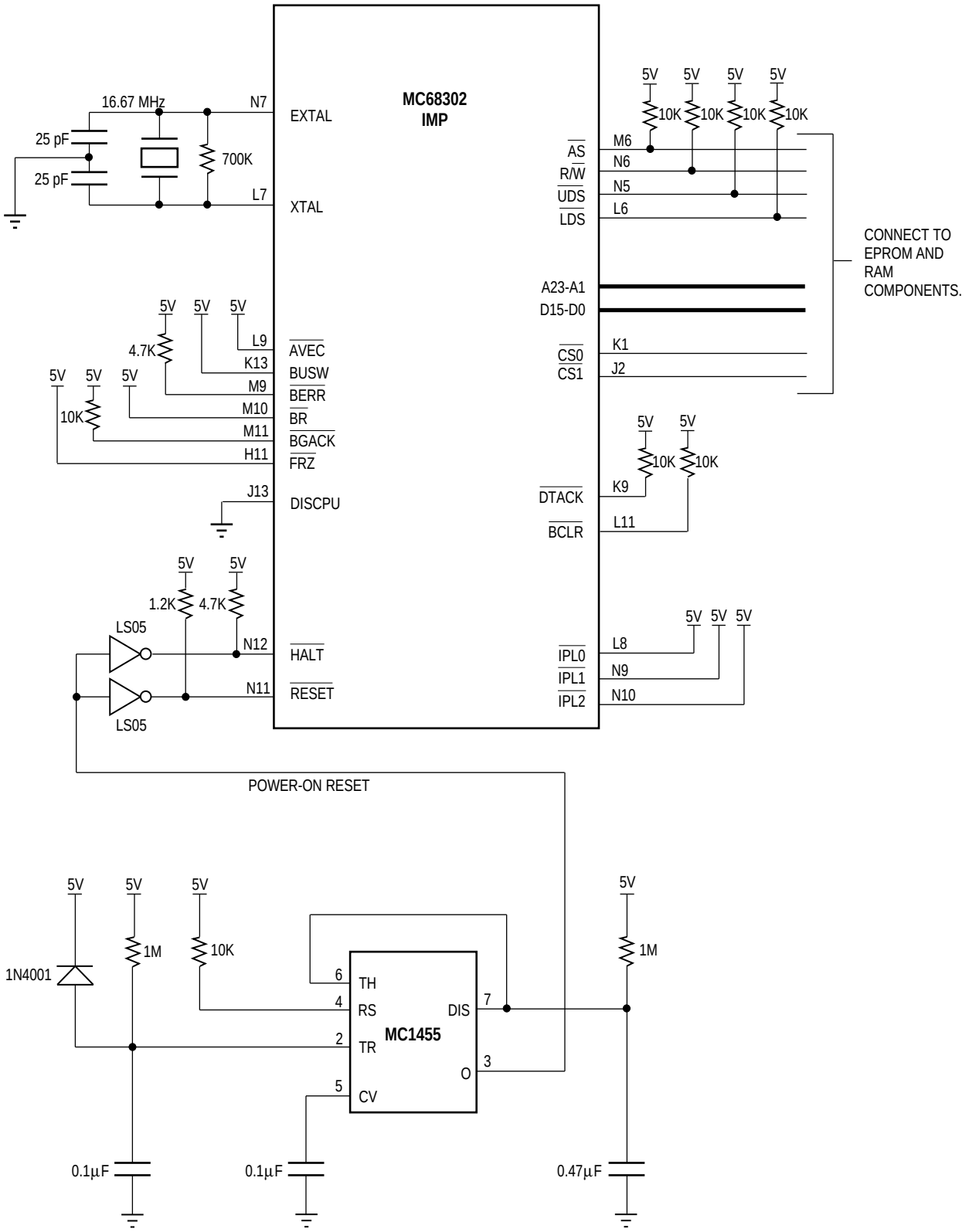


Figure D-1. MC68302 Minimum System Configuration (Sheet 1 of 2)

ing this register, the compare function code (CFC) bit should be cleared, since data stored in the program ROM will need to be accessed and moved to other areas of memory. (At reset, the function code for $\overline{CS0}$ defaults to 110b to select supervisor program, and the function code comparison is enabled). Thus, OR0 could be written with $5F80 for this example.

Next, the RAM addresses should be defined. To set the range of the RAM for 64 kbytes, to configure it for zero wait states, to allow both reads and writes, and to disable function code comparisons, OR1 should be set to $1FE0. To initially place the RAM at $400000, enable the RAM and set the function code to supervisor data; BR1 should be set to $A801.

Now that the RAM is enabled, an initial set of vectors may be placed in it (i.e., copied from the ROM). Of course, they will not be accessed as vectors until the RAM is moved to location 0. Any exceptions that occur during this time will still have their vectors derived from the ROM vector table.

The following situation now exists:

ROM—$0 to $03FFFF

RAM—$400000 to $40FFFF

MC68302—$700000 to $700FFF

D.2.3 Switching Process

To perform the switch, jump from the ROM to the dual-port RAM, reconfigure chip selects 1 and 0, and then jump back to the ROM. It is important that the RAM be moved first to ensure that an exception vector table is always present. During the brief period while the locations in $\overline{CS1}$ and $\overline{CS0}$ overlap, $\overline{CS0}$ will take precedence.

To perform the switch we need to execute a short dual-port RAM program. Thus, we copy the following data (instructions) from ROM to the dual-port RAM starting at location $700000:

```
MOVE.W     #$A001, ($700834).L   ; Place CS1 at $0 by writing to BR1
MOVE.W     #$C201, ($700830).L   ; Place CS0 at $100000 by writing to BR0
JMP       ($Address in ROM).L
```

Thus, first copy the binary coding of the preceding program from ROM to the dual-port RAM starting at location $700000.

Next, execute

```
JMP       ($700000).L            ; Jump to Dual-Port RAM
```

which then causes the following final situation:

RAM—$000000 to $00FFFF

ROM—$100000 to $13FFFF

MC68302—$700000 to $700FFF

The initialization of chip selects 0 and 1 and the switch of ROM and RAM are now complete.

```
          BSET.B    #$07,(A0)          ;Set Empty bit of Rx BD
          BTST.B    #$05,(A0)          ;test Wrap bit
          BNE.B     REINIT0            ;It set, reinit A0 to 700600
          ADDA.W    #$08,A0            ;else inc A0 by 8 to next Rx BD
          BRA.B     ALMDONE            ;Jump to Almost Done
REINIT0 MOVEA.L     #$700600,A0        ;Reinitialize A0
          BRA.B     ALMDONE            ;Jump to almost Done
*Bad Status:
BSTAT     NOP                          ;Bad status handler would go here
*Note that the UART FRMEC, NOSEC, and PAREC counters record bad status.
          BRA.B     INCPTR             ;Jump back to Receive handler
* Almost Done:
ALMDONE   MOVE.W    #$0100,1SR         ;Clear SCC3 bit in the ISR
          RTE                          ;end of interrupt handler

END
```

D.5 INDEPENDENT DMA IN THE MC68302

Moving of data between a high-speed peripheral controller and memory is optimized when a direct memory access (DMA) controller is used. The MC68302 contains an independent direct memory access (IDMA) controller. Engineers developing system architectures requiring both the M68000 microprocessor and DMA can use the MC68302 to obtain both building blocks in one package.

The registers associated with the programming of the IDMA are as follows: six IDMA registers, four interrupt controller registers, and one parallel I/O register (see Table D-1).

Table D-1. IDMA Registers

Acronym	Register	Address
CMR	Channel Mode Register	Base + $802
SAPR	Source Address Pointer Register	Base + $804
DAPR	Destination Address Pointer Register	Base + $808
BCR	Byte Count Register	Base + $80c
CSR	Channel Status Register	Base + $80e
FCR	Function Code Register	Base + $810
GIMR	Global Interrupt Mode Register	Base + $812
IPR	Interrupt Pending Register	Base + $814
IMR	Interrupt Mask Register	Base + $816
ISR	Interrupt In-Service Register	Base + $818
PACNT	Port A Control Register	Base + $81e

D.5.1 IDMA Overview

The IDMA can transfer data between any combination of memory and I/O, in either byte or word quantities with even or odd source and destination addresses. Each IDMA transfer requires two fundamental operations: reading data from a source address and writing data to a destination address. A pointer to the source data is contained in the source address point-

D.6.1 M68000 Core

The M68000 core processor on the MC68302 is instruction and timing compatible with the standard MC68000 (16-bit) or MC68008 (8-bit) versions of the M68000 Family. The core supports bus lock during read-modify-write cycles, a low latency interrupt mechanism, and bus width configuration. It does not support the older M6800 peripherals.

D.6.2 Communications Processor

The communication processor consists of a RISC processor, three serial communication controllers (SCCs), six DMA channels for the three SCCs, a programmable physical interface, a programmable serial communication port (SCP), and two serial management controllers (SMCs). The RISC processor, a separate processor from the M68000 core processor, is dedicated to the service of the SCCs, SCP, and SMCs.

The MC68302 supports three, full-duplex, independent SCCs, which support HDLC, UART, BISYNC, DDCMP, and V.110 protocols as well as transparent mode.

The physical interface supports a standard nonmultiplexed interface for each of the three SCCs (TXD, RXD, TCLK, RCLK, CTS, RTS, and CD) as well as several multiplexed modes.In multiplexed modes, up to three SCCs can be time-multiplexed onto the same serial channel. The multiplexed modes include IDL, GCI, and PCM highway.

The SCP is a full-duplex, synchronous, character-oriented channel that provides a three-wire interface. It is used to control and program SPI-type devices. The SCP implements a subset of Motorola's SPI interface.

The two SMCs are used to exchange control information multiplexed with the 2B + D data in the IDL or GCI buses.

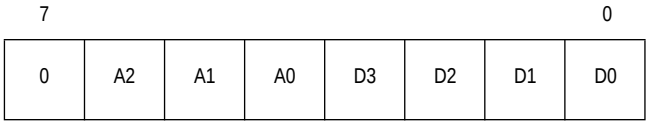
D.6.3 System Integration Block

The system integration block incorporates general-purpose peripherals that eliminate the glue logic found in most M68000 systems. It includes an independent DMA controller (IDMA), an interrupt controller, parallel I/O ports, 1152-byte dual-port RAM, two timers, one watchdog timer, chip-select lines and wait-state generation logic, a bus arbiter, low power modes, core disable logic, on-chip clock generator, and a hardware watchdog.

D.6.4 IDL Bus

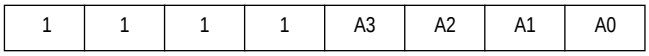
The IDL was developed to maximize the portability of the various chips required in an ISDN system. It provides a consistent interface definition across which a family of ISDN chips will be able to transport data (see Figure D-12).

A nibble register write is made by writing one byte with the following format:

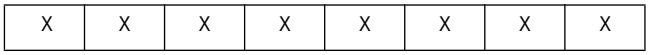


A3–A0 Address Register
D3–D0 Data Register

A byte register read is made by writing two bytes with the following format:



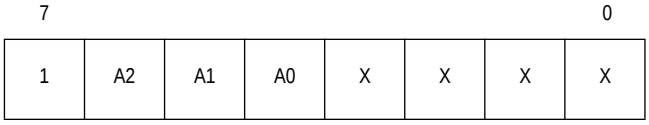
A3–A0 Address Register



X–Don't Care

Data read from the register will be received during the second transaction.

A nibble register read is made by writing one byte with the following format:



A2–A0 Address Register
X– Don't Care

Data read from the register will be received during the second transaction.

NOTE

The SCP_EN signal must be asserted prior to each SCP transaction and negated after completion.

D.6.13 Additional IMP To S/T Chip Connections

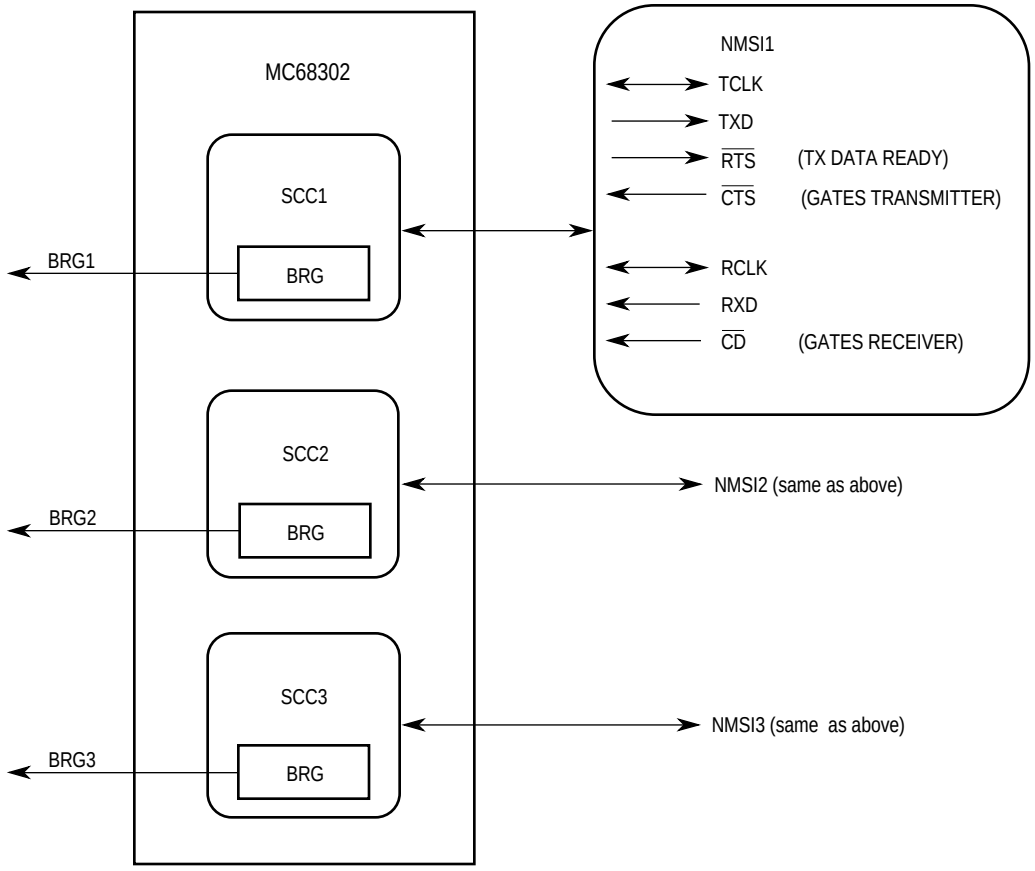
In addition to the IDL bus and the SCP bus, two discrete signals connect the MC145475 S/T chip to the MC68302 (see Figure D-17).

$\overline{\text{IRQ}}$ —The active-low signal sends an interrupt request from the MC145475 to the MC68302 core. This is an active-low signal that is asserted when one or more of the following events occurs:

— Change in the received information state (INFO n) of the S/T receiver.
— Multiframe reception.
D-channel collision.

Each event can be masked and/or cleared by a write/read operation on the corresponding register. The $\overline{\text{IRQ}}$ signal can be connected to the IRQ1 pin of the MC68302 to generate a level 1 interrupt.

$\overline{\text{RESET}}$—This active-low signal initializes the MC145475, forces all internal state machines to the initial state, and forces all internal nibble and byte registers (except BR4 and



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

Figure D-21. NMSI Pin Definitions

The other three physical interfaces, PCM highway, IDL, and GCI, are called multiplexed interfaces since they allow data from one, two, or all three SCCs to be time multiplexed together on the same pins. If a multiplexed interface is chosen, the first SCC to use that interface must be SCC1, since the three multiplexed modes share pins with SCC1 (see Figure D-22). After choosing a multiplexed mode, you can decide independently whether SCC2 and SCC3 should be part of the multiplexed interface or whether they should have their own set of NMSI pins.

If you are working in ISDN, transparent mode can be quite useful in sending and receiving transparent data over the 2B + D interface. IDL and GCI allow the SCCs to transmit and receive data on the two 64 kbps B channels and on the one 16 kbps D channel in basic rate ISDN. If you are not interfacing to a 2B + D ISDN environment, you can probably rule out using IDL and GCI.

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.

E.1.1.4.1 Receive BD Control/Status Word. To initialize the buffer, the user should write bits 15-12 and clear bits 11-10 and 5-0. 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 buffer 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 (The RXF bit in the event register is set when a complete frame is received, independent of the I bit.)

0 = The RXB bit in the event register is not set when this buffer is closed.
1 = The RXB bit (or RXF bit, if this is the last buffer in a frame) in the event register is set when this buffer is closed.

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.

F—First in Frame

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

Bits 9–6—Reserved for future use

LG—Rx Frame Length Violation

0 = No frame length violation occurred.
1 = A frame length violation was detected. Up to the number of bytes specified in the maximum frame length will be written to the buffer (or buffers, if multiple buffers per frame).

NO—Rx Nonoctet Aligned Frame

0 = An octet aligned frame was received.
1 = A nonoctet aligned frame was received.

AB—Rx Abort Sequence

0 = No abort was received.
1 = A minimum of seven ones was received during frame reception.

Table E-1. (a)Transparent Programming Model
Receive and Transmit Buffer Descriptors for SCCx

Initialized by User	Offset Hex	Name	Initialized by User	Offset Hex	Name
Yes	00	Rx BD 0 Control/Status	Yes	40	Tx BD 0 Control/Status
	02	Rx BD 0 Data Count	Yes	42	Tx BD 0 Data Count
Yes	04	Rx BD 0 Data Pointer (High Word)	Yes	44	Tx BD 0 Data Pointer (High Word)
Yes	06	Rx BD 0 Data Pointer (Low Word)	Yes	46	Tx BD 0 Data Pointer (Low Word)
Yes	08	Rx BD 1 Control/Status	Yes	48	Tx BD 1 Control/Status
	0A	Rx BD 1 Data Count	Yes	4A	Tx BD 1 Data Count
Yes	0C	Rx BD 1 Data Pointer (High Word)	Yes	4C	Tx BD 1 Data Pointer (High Word)
Yes	0E	Rx BD 1 Data Pointer (Low Word)	Yes	4E	Tx BD 1 Data Pointer (Low Word)
Yes	10	Rx BD 2 Control/Status	Yes	50	Tx BD 2 Control/Status
	12	Rx BD 2 Data Count	Yes	52	Tx BD 2 Data Count
Yes	14	Rx BD 2 Data Pointer (High Word)	Yes	54	Tx BD 2 Data Pointer (High Word)
Yes	16	Rx BD 2 Data Pointer (Low Word)	Yes	56	Tx BD 2 Data Pointer (Low Word)
Yes	18	Rx BD 3 Control/Status	Yes	58	Tx BD 3 Control/Status
	1A	Rx BD 3 Data Count	Yes	5A	Tx BD 3 Data Count
Yes	1C	Rx BD 3 Data Pointer (High Word)	Yes	5C	Tx BD 3 Data Pointer (High Word)
Yes	1E	Rx BD 3 Data Pointer (Low Word)	Yes	5E	Tx BD 3 Data Pointer (Low Word)
Yes	20	Rx BD 4 Control/Status	Yes	60	Tx BD 4 Control/Status
	22	Rx BD 4 Data Count	Yes	62	Tx BD 4 Data Count
Yes	24	Rx BD 4 Data Pointer (High Word)	Yes	64	Tx BD 4 Data Pointer (High Word)
Yes	26	Rx BD 4 Data Pointer (Low Word)	Yes	66	Tx BD 4 Data Pointer (Low Word)
Yes	28	Rx BD 5 Control/Status	Yes	68	Tx BD 5 Control/Status
	2A	Rx BD 5 Data Count	Yes	6A	Tx BD 5 Data Count
Yes	2C	Rx BD 5 Data Pointer (High Word)	Yes	6C	Tx BD 5 Data Pointer (High Word)
Yes	2E	Rx BD 5 Data Pointer (Low Word)	Yes	6E	Tx BD 5 Data Pointer (Low Word)
Yes	30	Rx BD 6 Control/Status	Yes	70	Tx BD 6 Control/Status
	32	Rx BD 6 Data Count	Yes	72	Tx BD 6 Data Count
Yes	34	Rx BD 6 Data Pointer (High Word)	Yes	74	Tx BD 6 Data Pointer (High Word)
Yes	36	Rx BD 6 Data Pointer (Low Word)	Yes	76	Tx BD 6 Data Pointer (Low Word)
Yes	38	Rx BD 7 Control/Status	Yes	78	Tx BD 7 Control/Status
	3A	Rx BD 7 Data Count	Yes	7A	Tx BD 7 Data Count
Yes	3C	Rx BD 7 Data Pointer (High Word)	Yes	7C	Tx BD 7 Data Pointer (High Word)
Yes	3E	Rx BD 7 Data Pointer (Low Word)	Yes	7E	Tx BD 7 Data Pointer (Low Word)

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

.

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

Initialized by User	Offset Hex	Name		Initialized by User	Offset Hex	Name
Yes	80	RFCR	TFCR		A2	Reserved
Yes	82	MRBLR			A4	Reserved
	84	Rx Internal State			A6	Reserved
	86	Reserved	Rx Internal Buffer No.		A8	Reserved
	88 8A	Rx Internal Data Pointer (High Word) Rx Internal Data Pointer (Low Word)			AA	Reserved
	8C	Rx Internal Byte Count			AC	Reserved
	8E	Rx Temp			AE	Reserved
	90	Tx Internal State			B0	Reserved
	92	Reserved	Tx Internal Buffer No.		B2	Reserved
	94 96	Tx Internal Data Pointer (High Word) TX Internal Data Pointer (Low Word)			B4	Reserved
	98	Tx Internal Byte Count			B6	Reserved
	9A	Tx Temp			B8	Reserved
	9C	Reserved			BA	Reserved
	9E	Reserved			BC	Reserved
	A0	Reserved			BE	Reserved

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