Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Active
Core Processor	M68000
Number of Cores/Bus Width	1 Core, 8/16-Bit
Speed	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	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68302ag20c

SECTION 3
SYSTEM INTEGRATION BLOCK (SIB)

The MC68302 contains an extensive SIB that simplifies the job of both the hardware and software designer. It integrates the M68000 core with the most common peripherals used in an M68000-based system. The independent direct memory access (IDMA) controller relieves the hardware designer of the extra effort and board logic needed to connect an external DMA controller. The interrupt controller can be used in a dedicated mode to generate interrupt acknowledge signals without external logic. Also, the chip-select signals and their associated wait-state logic eliminate the need to generate chip-select signals externally. The three timers simplify control and improve reliability. These and other features in the SIB conserve board space and cost while decreasing development time.

The SIB includes the following functions:

- IDMA Controller with Three Handshake Signals: $\overline{\text{DREQ}}$, $\overline{\text{DACK}}$, and $\overline{\text{DONE}}$
- Interrupt Controller with Two Modes of Operation
- Parallel Input/Output (I/O) Ports, Some with Interrupt Capability
- On-Chip 1152-Byte Dual-Port RAM
- Three Timers Including a Software Watchdog Timer
- Four Programmable Chip-Select Lines with Wait-State Generator Logic
- On-Chip Clock Generator with Output Signal
- System Control

  —System Status and Control Logic
  —Disable CPU Logic (M68000)
  —Bus Arbitration Logic with Low-Interrupt Latency Support
  —Hardware Watchdog for Monitoring Bus Activity
  —Low-Power (Standby) Modes
  —Freeze Control for Debugging
- Clock Control

  —Adjustable CLKO Drive
  —Three-state RCLK1 and TCLK1
  —Disable BRG1
- DRAM Refresh Controller

$\overline{\text{DREQ}}$ input to the IDMA is level-sensitive and is sampled at certain points to determine when a valid request is asserted by the device. The device requests service by asserting $\overline{\text{DREQ}}$ and leaving it asserted. In response, the IDMA arbitrates for the system bus and begins to perform an operand transfer. During each access to the device, the IDMA will assert $\overline{\text{DACK}}$ to indicate to the device that a request is being serviced. If $\overline{\text{DREQ}}$ remains asserted when the IDMA completes the peripheral cycle (the cycle during which $\overline{\text{DACK}}$ is asserted by the IDMA) one setup time (see specification. 80) before the S5 falling edge (i.e., before or with $\overline{\text{DTACK}}$), then a valid request for another operand transfer is recognized, and the IDMA will service the next request immediately. If $\overline{\text{DREQ}}$ is negated one setup time (see specification 80) before the S5 falling edge, a new request will not be recognized, and the IDMA will relinquish the bus.

NOTE:

If 8 to 16 bit packing occurs, then the $\overline{\text{DREQ}}$ is sampled during the last 8-bit cycle.

External Cycle Steal

For external devices that generate a pulsed signal for each operand to be transferred, the external cycle steal mode uses $\overline{\text{DREQ}}$ as a falling edge-sensitive input. The IDMA will respond to cycle-steal requests in the same manner as for all other requests. However, if subsequent $\overline{\text{DREQ}}$ pulses are generated before $\overline{\text{DACK}}$ is asserted in response to each request, they will be ignored. If $\overline{\text{DREQ}}$ is asserted after the IDMA asserts $\overline{\text{DACK}}$ for the previous request but one setup time (see specification 80) before the S5 falling edge, then the new request will be serviced before the bus is relinquished. If a new request has not been generated by one setup time (see specification 80) before the S5 falling edge, the bus will be released to the next bus master.

3.1.4.5 Block Transfer Termination

The user may stop the channel by clearing STR. Additionally, the channel operation can be terminated for any of the following reasons: transfer count exhausted, external device termination, or error termination. This is independent of how requests are generated to the IDMA.

Transfer Count Exhausted

When the channel begins an operand transfer, if the current value of the BCR is one or two (according to the operand size in the CMR), $\overline{\text{DONE}}$ is asserted during the last bus cycle to the device to indicate that the channel operation will be terminated when the current operand transfer has successfully completed. In the memory to memory case, $\overline{\text{DONE}}$ is asserted during the last access to memory (source or destination) as defined by the ECO bit. When the operand transfer has completed and the BCR has been decremented to zero, the channel operation is terminated, STR is cleared, and an interrupt is generated if INTN is set. The SAPR and/or DAPR are also incremented in the normal fashion.

NOTE

If the channel is started with BCR value set to zero, the channel will transfer 64K bytes.

ET7—$\overline{\text{IRQ7}}$ Edge-/Level-Triggered

This bit is valid only in the dedicated mode.

0 = Level-triggered. An interrupt is made pending when $\overline{\text{IRQ7}}$ is low.

NOTE

The M68000 always treats level 7 as an edge-sensitive interrupt. Normally, users should not select the level-triggered option. The level-triggered option is useful when it is desired to make the negation of $\overline{\text{IRQ7}}$ cause the IOUT2–IOUT0 pins to cease driving a level 7 interrupt request when the MC68302 is used in the disable CPU mode. This situation is as follows:

For a slave-mode MC68302, when it is triggered by $\overline{\text{IRQ1}}$, $\overline{\text{IRQ6}}$, or $\overline{\text{IRQ7}}$ to generate an interrupt, its interrupt controller will output the interrupt request on pins IOUT2–IOUT0 to another processor (MC68302, MC68020, etc.) For cases when the slave MC68302 does not generate a level 4 vector (i.e., the VGE bit is cleared), one must set the ET1, ET6, and ET7 bits to level-triggered and then negate the $\overline{\text{IRQ1}}$, $\overline{\text{IRQ6}}$, and $\overline{\text{IRQ7}}$ lines externally in the interrupt handler code. If the ET1, ET6, and ET7 bits are set to edge-triggered and the VGE bit is clear, the IOUT2–IOUT0 pins will never be cleared.

1 = Edge-triggered. An interrupt is made pending when $\overline{\text{IRQ7}}$ changes from one to zero (falling edge).

ET6—IRQ6 Edge-/Level-Triggered

This bit is valid only in the dedicated mode.

0 = Level-triggered. An interrupt is made pending when $\overline{\text{IRQ6}}$ is low.

NOTE

While in disable CPU mode during the host processor interrupt acknowledge cycle for $\overline{\text{IRQ6}}$, if $\overline{\text{IRQ6}}$ is not continuously asserted, the interrupt controller will still provide the vector number (and $\overline{\text{DTACK}}$) according to the IV6 bit. The $\overline{\text{IACK6}}$ falling edge may be used externally to negate $\overline{\text{IRQ6}}$.

1 = Edge-triggered. An interrupt is made pending when $\overline{\text{IRQ6}}$ changes from one to zero (falling edge).

ET1—$\overline{\text{IRQ1}}$ Edge-/Level-Triggered

This bit is valid only in the dedicated mode.

0 = Level-triggered. An interrupt is made pending when $\overline{\text{IRQ1}}$ is low.

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

address match exists within its address space and, therefore, whether to assert the chip-select line.

111 = Not supported; reserved. Chip select will not assert if this value is chosen.
110 = Value may be used.
•
•
•
000 = Value may be used.

After system reset, the FC field in BR3–BR0 defaults to supervisor program space (FC = 110) to select a ROM device containing the reset vector. Because of the priority mechanism and the EN bit, only the $\overline{CS0}$ line is active after a system reset.

NOTE

The FC bits can be masked and ignored by the chip-select logic using CFC in the OR.

Bits 12–2—Base Address

These bits are used to set the starting address of a particular address space. The address compare logic uses only A23–A13 to cause an address match within its block size. The base address should be located on a block boundary. For example, if the block size is 64k bytes, then the base address should be a multiple of 64k.

After system reset, the base address defaults to zero to select a ROM device on which the reset vector resides. All base address values default to zero on system reset, but, because of the priority mechanism, only $\overline{CS0}$ will be active.

NOTE

All address bits can be masked and ignored by the chip-select logic through the base address mask in the OR.

RW—Read/Write

0 = The chip-select line is asserted for read operations only.
1 = The chip-select line is asserted for write operations only.

After system reset, this bit defaults to zero (read-only operation).

NOTE

This bit can be masked and ignored by the read-write compare logic, as determined by MRW in the OR. The line is then asserted for both read and write cycles.

On write protect violation cycles (RW = 0 and MRW = 1), $\overline{BERR}$ will be generated if WPVE is set, and WPV will be set.

If the write protect mechanism is used by an external master, the R/$\overline{W}$ low to $\overline{AS}$ asserted timing should be 16 ns minimum.

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.

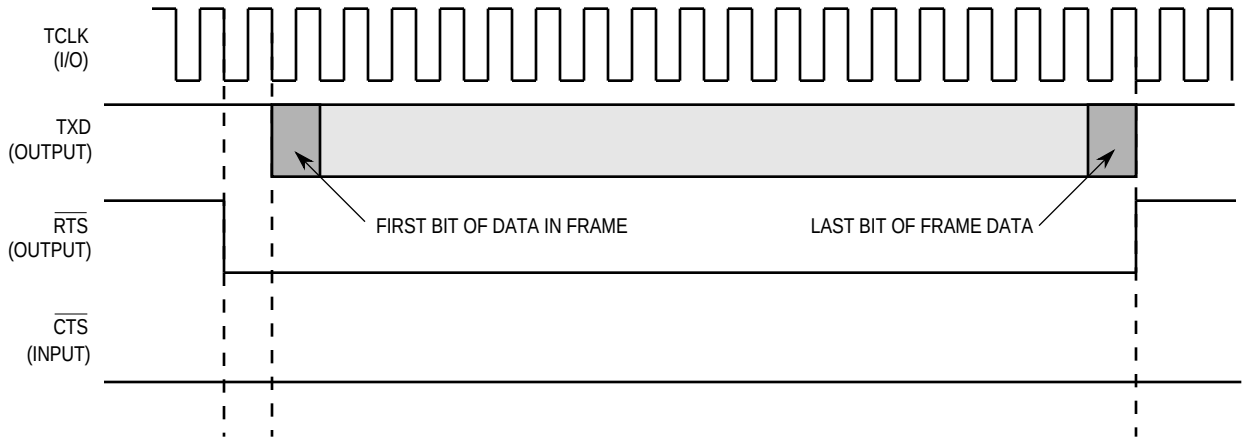
In the deactivated state, the clock pulse is disabled, and the data line is a logic one. The layer-1 device activates the IMP by enabling the clock pulses and by an indication in the channel 0 C/I channel. The IMP will then report to the M68000 core by a maskable interrupt that a valid indication is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

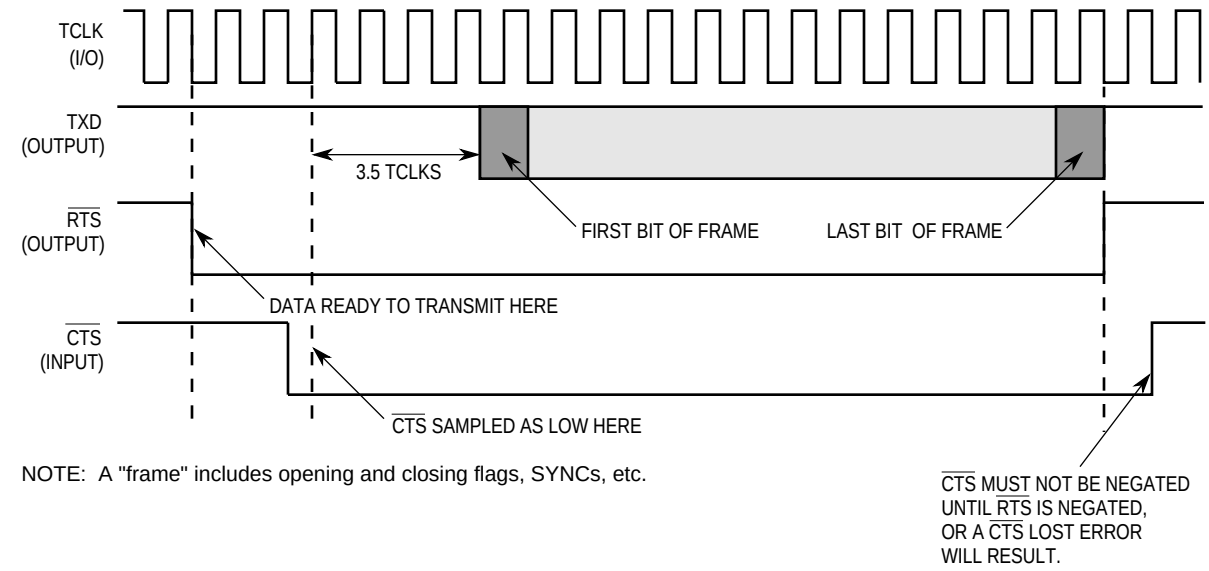
In PCM highway mode, one, two, or all three SCCs can be multiplexed together to support various time-division multiplexed interfaces. PCM highway supports the standard T1 and CEPT interfaces as well as user-defined interfaces. In this mode, the NMSI1 pins have new names and functions (see Table 4-2).

allel I/O lines in the PACNT register. To cause the TXD and $\overline{RTS}$ pins to simply remain high in NMSI1, NMSI2, and NMSI3 modes, use this loopback mode in conjunction with setting the SDIAG1–SDIAG0 bits in the SIMODE register to loopback control.



NOTE: A "frame" includes opening and closing flags in HDLC and SYNCs in BISYNC and DDCMP.

Figure 4-13. Output Delays from RTS Low, Synchronous Protocol



NOTE: A "frame" includes opening and closing flags, SYNCs, etc.

Figure 4-14. Output Delays from $\overline{CTS}$ Low, Synchronous Protocol

If an internal loopback is desired when this SCC is configured to a multiplexed physical interface, then only the SDIAG1–SDIAG0 bits need be set.
When using loopback mode, the clock source for the transmitter and the receiver (as set in the TCS and RCS bits in the SCON register), must be the same. Thus,

4.5.6.2 Maximum Receive Buffer Length Register (MRBLR)

Each SCC has one MRBLR that is used to define the receive buffer length for that SCC. The MRBLR defines the maximum number of bytes that the IMP will write to a receive buffer on that SCC before moving to the next buffer. The IMP may write fewer bytes to the buffer than MRBLR if a condition such as an error or end of frame occurs, but it will never write more bytes than the MRBLR value. Thus, buffers supplied by the user for use by the IMP should always be of size MRBLR (or greater) in length.

The transmit buffers for an SCC are not affected in any way by the value programmed into MRBLR. Transmit buffers may be individually chosen to have varying lengths, as needed. The number of bytes to be transmitted is chosen by programming the data length field in the Tx BD.

NOTE

MRBLR was not intended to be changed dynamically while an SCC is operating. However, if it is modified in a single bus cycle with one 16-bit move (NOT two 8-bit back-to-back bus cycles), then a dynamic change in receive buffer length can be successfully achieved, which occurs when the CP moves control to the next Rx BD in the table. Thus, a change to MRBLR will not have an immediate effect. To guarantee the exact Rx BD on which the change will occur, the user should change MRBLR only while the SCC receiver is disabled (see 4.5.6 SCC Parameter RAM Memory Map).

NOTE

The MRBLR value should be greater than zero in all modes. In the HDLC and transparent modes, the MRBLR should have an even value.

4.5.6.3 Receiver Buffer Descriptor Number (RBD#)

The RBD# for each SCC channel defines the next BD to which the receiver will move data when it is in the IDLE state or defines the current BD during frame processing. The RBD# is the BD offset from the SCC base in the Rx BD table. For Rx BD 0, RBD# = $00; for Rx BD 1, RBD# = $08, etc. Upon reset, the CP main controller sets this register to zero. The user can change this register only after the ENR bit is clear and after the ENTER HUNT MODE command has been issued. In most applications, this parameter will never need to be modified by the user.

4.5.6.4 Transmit Buffer Descriptor Number (TBD#)

The TBD# for each SCC channel defines the next BD from which the transmitter will move data when it is in the IDLE state or defines the current BD during frame transmission. The TBD# is the BD offset from the SCC base in the Tx BD table. For Tx BD 0, TBD# = $40; for Tx BD 1, TBD# = $48, etc. Upon reset, the CP main controller sets this register to $40. The user can change this register only after the STOP TRANSMIT command has been issued. In most applications, this parameter will never need to be modified by the user.

RCCR, CHARACTER

The UART controller can automatically recognize special characters and generate interrupts. It also allows a convenient method for inserting flow control characters into the transmit stream. See 4.5.11.7 UART Control Characters and Flow Control for more details.

If neither of these capabilities are desired, initialize CHARACTER1 to $8000 and CHARACTER8 to $0000 to disable both functions.

4.5.11.4 UART Programming Model

An SCC configured as a UART uses the same data structure as the other protocols. The UART data structure supports multibuffer operation. The UART may also be programmed to perform address comparison whereby messages not destined for a given programmable address are discarded. Also, the user can program the UART to accept or reject control characters. If a control character is rejected, an interrupt may be generated. The UART enables the user to transmit break and preamble sequences. Overrun, parity, noise, and framing errors are reported using the buffer descriptor (BD) table and/or error counters. An indication of the status of the line (idle) is reported through the status register, and a maskable interrupt is generated upon a status change.

In its simplest form, the UART can function in a character-oriented environment. Each character is transmitted with accompanying stop bits and parity (as configured by the user) and is received into separate one-byte buffers. Reception of each buffer may generate a maskable interrupt.

Many applications may want to take advantage of the message-oriented capabilities supported by the UART using linked buffers to receive or transmit data. In this case, data is handled in a message-oriented environment; users can work on entire messages rather than operating on a character-by-character basis. A message may span several linked buffers. For example, rather than being interrupted after the reception of each character, a terminal driver may want to wait until an end-of-line character has been typed by a user before handling the input data.

As another example, when transmitting ASCII files, the data may be transferred as messages ending on the end-of-line character. Each message could be both transmitted and received as a circular list of buffers without any intervention from the M68000 core. This technique achieves both ease in programming and significant savings in processor overhead.

On the receive side, the user may define up to eight control characters. Each control character may be configured to designate the end of a message (such as end of line) or to generate a maskable interrupt without being stored in the data buffer. This latter option is useful when flow-control characters such as XON or XOFF need to alert the M68000 core, yet do not belong to the message being received. Flow-control characters may also be transmitted at any time.

In the message-oriented environment, the data stream is divided into buffers. However, the physical format of each character (stop bits, parity, etc.) is not altered.

ceives the address character and writes it to a new buffer. No address recognition is performed.

10 = The DDCMP protocol is implemented over the asynchronous channel.

11 = Multidrop mode is enabled as in the 01 case, and the IMP automatically checks the address of the incoming address character and either accepts or discards the data following the address.

FRZ—Freeze Transmission

This bit allows the user to halt the UART transmitter and to continue transmission from the next character in the buffer at a later time.

0 = Normal operation (or resume transmission after FRZ is set).

1 = The UART completes transmission of any data already transferred to the UART FIFO (up to three characters) and then stops transmitting data. The UART continues to receive normally.

CL—Character Length

0 = 7-bit character length. On receive, bit 7 in memory is written as zero. On transmit, bit 7 in memory is a don't care.

1 = 8-bit character length

RTSM—RTS Mode

0 = $\overline{\text{RTS}}$ is asserted whenever the transmitter is enabled and there are characters to transmit. $\overline{\text{RTS}}$ is negated after the last stop bit of a transmitted character when both the shift register and the transmit FIFO are empty. RTS is also negated at the end of a buffer to guarantee accurate reporting of the CTS bit in the BD.

1 = $\overline{\text{RTS}}$ is asserted whenever the transmitter is enabled (the ENT bit is set).

SL—Stop Length

This bit selects the number of the stop bits transmitted by the UART. The receiver is always enabled for one stop bit. Fractional stop bits are configured in the DSR (see 4.5.11.12 Fractional Stop Bits).

0 = One stop bit

1 = Two stop bits

COMMON SCC MODE BITS—see 4.5.3 SCC Mode Register (SCM) for a description of the DIAG1, DIAG0, ENR, ENT, MODE1, and MODE0 bits.

4.5.11.14 UART Receive Buffer Descriptor (Rx BD)

The CP reports information about each buffer of received data by its BDs. The Rx BD is shown in Figure 4-20. The CP closes the current buffer, generates a maskable interrupt, and starts to receive data in the next buffer due to any of the following events:

1. Reception of a user-defined control character (when reject (R) bit = 0)
2. Detection of an error during message processing
3. Detection of a full receive buffer
4. Reception of a programmable number of consecutive IDLE characters

and the frame length in the last BD. MFLR is defined as all the in-frame bytes between the opening flag and the closing flag (address, control, data, and CRC). MAX_CNT is a temporary downcounter used to track the frame length.

4.5.12.8 HDLC Error-Handling Procedure

The HDLC controller reports frame reception and transmission error conditions using the channel BDs, the error counters, and the HDLC event register. The modem interface lines can also be directly monitored in the SCC status register.

Transmission Errors:

1. Transmitter Underrun. When this error occurs, the channel terminates buffer transmission, closes the buffer, sets the underrun (UN) bit in the BD, and generates the TXE interrupt (if enabled). The channel will resume transmission after the reception of the RESTART TRANSMIT command. The transmit FIFO size is four words.
2. Clear-To-Send Lost (Collision) During Frame 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 TXE interrupt (if enabled). The channel will resume transmission after the RESTART TRANSMIT command is given.

NOTE

If this error occurs on the first or second buffer of the frame and the retransmit enable (RTE) bit in the HDLC mode register is set, the channel will retransmit the frame when the $\overline{\text{CTS}}$ line becomes active again. When using this feature, users should design transmit frames to fit within two buffers or less. When working in ISDN mode with D-channel collision possibility, to ensure the retransmission method functions properly, the first and second data buffers should contain more than 10 bytes of data if multiple buffers per frame are used. (Small frames consisting of a single buffer are not subject to this requirement). The channel will also increment the retransmission counter (RETRC).

Reception Errors:

1. Overrun Error. The HDLC controller maintains an internal three-word 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 in the FIFO. When a receive FIFO overrun occurs, the channel writes the received data byte to the internal FIFO over the previously received byte. The previous data byte and the frame status are lost. Then the channel closes the buffer with the overrun (OV) bit in the BD set and generates the RXF interrupt (if enabled). The receiver then enters the hunt mode.
   Even if the overrun occurs during a frame whose address is not matched in the address recognition logic, a BD of length two will be opened to report the overrun, and the RXB interrupt will be generated (if enabled).
2. Carrier Detect Lost During Frame Reception. When this error occurs and the channel

Rx Buffer Pointer

This pointer contains the address of the associated data buffer and 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.

4.5.14.11 DDCMP Transmit Buffer Descriptor (Tx BD)

Data is presented to the CP for transmission over 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 M68000 core that the buffers have been serviced. The Tx BD is shown in Figure 4-37.

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

Figure 4-37. DDCMP Transmit Buffer Descriptor

The first word contains status and control bits. Bits 15–9 are prepared by the user before transmission. Bits 1–0 are set by the DDCMP controller after the buffer has been transmitted. Bit 15 is set by the user when the buffer and BD have been prepared and is cleared by the DDCMP controller when the message is transmitted.

R—Ready

0 = This buffer is not currently ready for transmission. The user is free to manipulate this BD (or its associated buffer). The DDCMP controller clears this bit after the buffer has been completely transmitted (or after an error condition is 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 DDCMP controller 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.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	E	X	W	—	—	—	—	—	—	—	—	—	SE	—	OV	—
OFFSET + 2	DATA LENGTH															
OFFSET + 4	RX BUFFER POINTER (24-bits used, upper 8 bits must be 0)															
OFFSET +6																

Figure 4-40. V.110 Receive Buffer Descriptor

The first word of the Rx BD contains control and status bits. Bits 15–13 are written by the user before the buffer is linked to the Rx BD table, and bits 1 and 3 are set by the IMP following message reception. Bit 15 is set by the M68000 core when the buffer is available to the V.110 controller and is cleared by the V.110 controller after filling the buffer.

E—Empty

0 = The data buffer associated with this BD has been filled with received data, or data reception has been aborted due to an error condition. The M68000 core is free to examine or write to any fields of the BD.

1 = The data buffer associated with this BD is empty. This bit signifies that the BD and its associated buffer are available to the V.110 controller. The M68000 core should not write to any fields of this BD after it sets this bit. The empty bit will remain set while the V.110 controller is currently filling the buffer with received data.

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 Rx BD table.

1 = This is the last BD in the Rx BD table. After this buffer has been used, the V.110 controller receives incoming data by placing it in 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.

Bits 12–4, 2, 0—Reserved for future use.

SE—Synchronization Error

A frame with a synchronization error was received. A synchronization error is detected by the V.110 controller when the MSB of a byte (except the all-zeros byte) is not one.

OV—Overrun

A receiver overrun occurred during message reception.

5.12 TYPICAL SERIAL INTERFACE PIN CONFIGURATIONS

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

Table 5-5. Typical ISDN Configurations

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

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

Table 5-6. Typical Generic Configurations

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

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

5.13 NMSI1 OR ISDN INTERFACE PINS

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

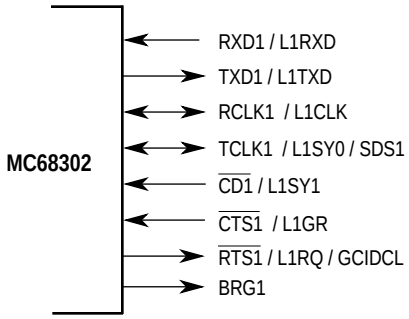
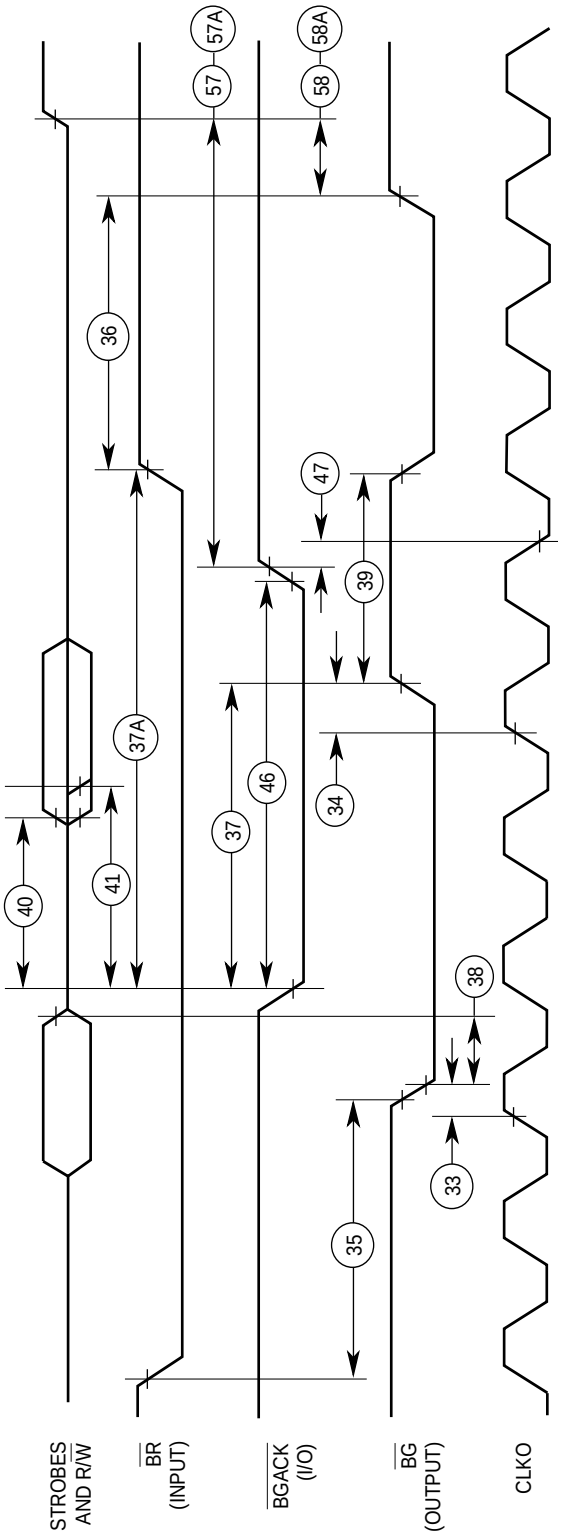


Figure 5-10. NMSI1 or ISDN Interface Pins



NOTE: Setup time to the clock (#47) for the asynchronous inputs BERR, BGACK, BR, DTACK, AND IPL2–IPL0 guarantees their recognition at the next falling edge of the clock.

Figure 6-5. Bus Arbitration Timing Diagram

D.3 MC68302 BUFFER PROCESSING AND INTERRUPT HANDLING

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

D.3.1 Buffer Descriptors Definition

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

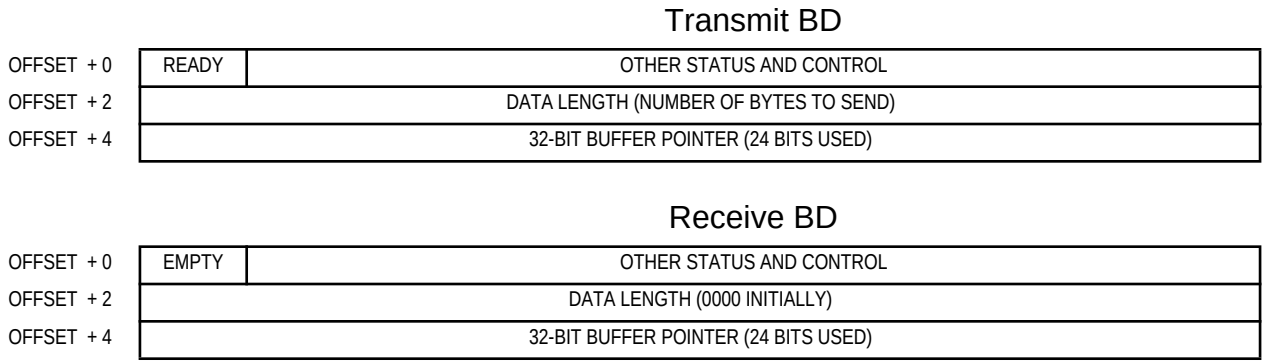


Figure D-3. Transmit and Receive BD Structure

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

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

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

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



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

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

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

EXTC—External Clock Source

0 = The internal main clock is the source for the baud rate generator.
1 = The external clock on the TIN1 pin is the source for the baud rate generator.

TCS—Transmit Clock Source

0 = Transmit clock source is the baud rate generator output.
1 = Transmit clock source is the clock signal on TCLK pin.

RCS—Receive Clock Source

0 = Receive clock source is the baud rate generator output.
1 = Receive clock source is the clock signal on TCLK pin.

CD10-CD0—Clock Divide

Used to preset the 11-bit counter that is decremented at the prescaler output rate.

DIV4—SCC Clock Prescaler Divide by 4

0 = Divide-by-1 prescaler.
1 = Divide-by-4 prescaler.

E.2.1.2.2 SCC Mode Register (SCM). This 16-bit register is located at offset $884 (SCC1), $894 (SCC2), and $8A4 (SCC3). The SCM register configures the operation of the SCC and defines UART specific parameters.

15	14	13	12	11	10	9	8
TPM1	TPM0	RPM	PEN	UM1	UM0	FRZ	CL

7	6	5	4	3	2	1	0
RSTM	SL	DIAG1	DIAG0	ENR	ENT	MODE1	MODE0

TPM1,TMP0—Transmitter Parity Mode

00 = Odd parity; always send an odd number of ones.
01 = Force low parity; always send a zero in the parity bit position.
10 = Even parity; always send an even number of ones.
11 = Force high parity; always send a one in the parity bit position.

RPM—Receiver Parity Mode

0 = Odd parity.
1 = Even parity.

PEN—Parity Enable

0 = No parity.
1 = Parity is enabled for the transmitter and receiver.