Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	M68000
Number of Cores/Bus Width	1 Core, 8/16-Bit
Speed	20MHz
Co-Processors/DSP	Communications; RISC CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	132-BPGA Exposed Pad
Supplier Device Package	132-PGA (34.5x34.5)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68302crc20c

Appendix F
Design Checklist

The watchdog timer has the following features:

- A 16-Bit Counter and Reference Register
- Maximum Period of 16.78 Seconds (at 16 MHz)
- 0.5 ms Resolution (at 16 MHz)
- Output Signal (WDOG)
- Interrupt Capability

3.5.2 General Purpose Timer Units

The clock input to the prescaler may be selected from the main clock (divided by 1 or by 16) or from the corresponding timer input (TIN) pin. TIN is internally synchronized to the internal clock. The clock input source is selected by the ICLK bits of the corresponding TMR. The prescaler is programmed to divide the clock input by values from 1 to 256. The output of the prescaler is used as an input to the 16-bit counter.

The resolution of the timer is one clock cycle (60 ns at 16.67 MHz). The maximum period (when the reference value is all ones) is 268,435,456 cycles (16.78 seconds at 16.00 MHz).

Each timer may be configured to count until a reference is reached and then either resets to zero on the next clock or continues to run. The free run/restart (FRR) bit of the corresponding TMR selects each mode. Upon reaching the reference value, the corresponding TER bit is set, and an interrupt is issued if the output reference interrupt enable (ORI) bit in TMR is set.

Each timer may output a signal on the timer output ($\overline{\text{TOUT1}}$ or $\overline{\text{TOUT2}}$) pin when the reference value is reached, as selected by the output mode (OM) bit of the corresponding TMR. This signal can be an active-low pulse for one clock cycle or a toggle of the current output. The output can also be used as an input to the other timer, resulting in a 32-bit timer.

Each timer has a 16-bit TCR, which is used to latch the value of the counter when a defined transition (of TIN1 or TIN2) is sensed by the corresponding input capture edge detector. The type of transition triggering the capture is selected by the capture edge and enable interrupt (CE) bits in the corresponding TMR. Upon a capture or reference event, the corresponding TER bit is set, and a maskable interrupt is issued.

The timer registers may be modified at any time by the user.

3.5.2.1 Timer Mode Register (TMR1, TMR2)

TMR1 and TMR2 are identical 16-bit registers. TMR1 and TMR2, which are memory-mapped read-write registers to the user, are cleared by reset.

15 … 8	7 6	5	4	3	2 1	0
PRESCALER VALUE (PS)	CE	OM	ORI	FRR	ICLK	RST

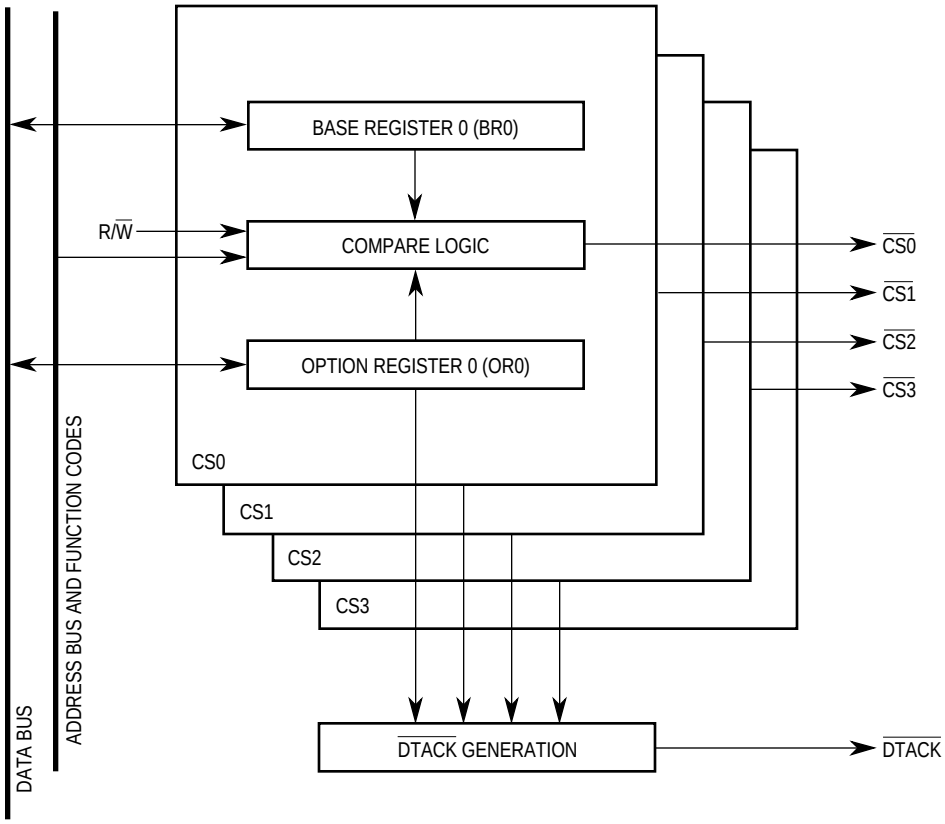


Figure 3-9. Chip-Select Block Diagram

The user should not normally program more than one chip-select line to the same area. When this occurs, the address compare logic will set address decode conflict (ADC) in the system control register (SCR) and generate $\overline{\text{BERR}}$ if address decode conflict enable (ADCE) is set. Only one chip-select line will be driven because of internal line priorities. $\overline{\text{CS0}}$ has the highest priority, and $\overline{\text{CS3}}$ the lowest. $\overline{\text{BERR}}$ will not be asserted on write accesses to the chip-select registers.

If one chip select is programmed to be read-only and another chip select is programmed to be write-only, then there will be no overlap conflict between these two chip selects, and the ADC bit will not be set.

When a bus master attempts to write to a read-only location, the chip-select logic will set write protect violation (WPV) in the SCR and generate $\overline{\text{BERR}}$ if write protect violation enable (WPVE) is set. The $\overline{\text{CS}}$ line will not be asserted.

NOTE

The chip-select logic is reset only on total system reset (assertion of $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$). Accesses to the internal RAM and registers, including the system configuration registers (BAR and

4.4.1 IDL Interface

The IDL interface is a full-duplex ISDN interface used to interconnect a physical layer device (such as the Motorola ISDN S/T transceiver MC145474) to the integrated multiprotocol processor (IMP). Data on five channels (B1, B2, D, A, and M) is transferred in a 20-bit frame every 125 µs, providing 160-kbps full-duplex bandwidth. The IMP is an IDL slave device that is clocked by the IDL bus master (physical layer device). The IMP provides direct connections to the MC145474. Refer to Figure 4-6 for the IDL bus signals.

The IMP supports 10-bit IDL as shown in Figure 4-6; it does not support 8-bit IDL.

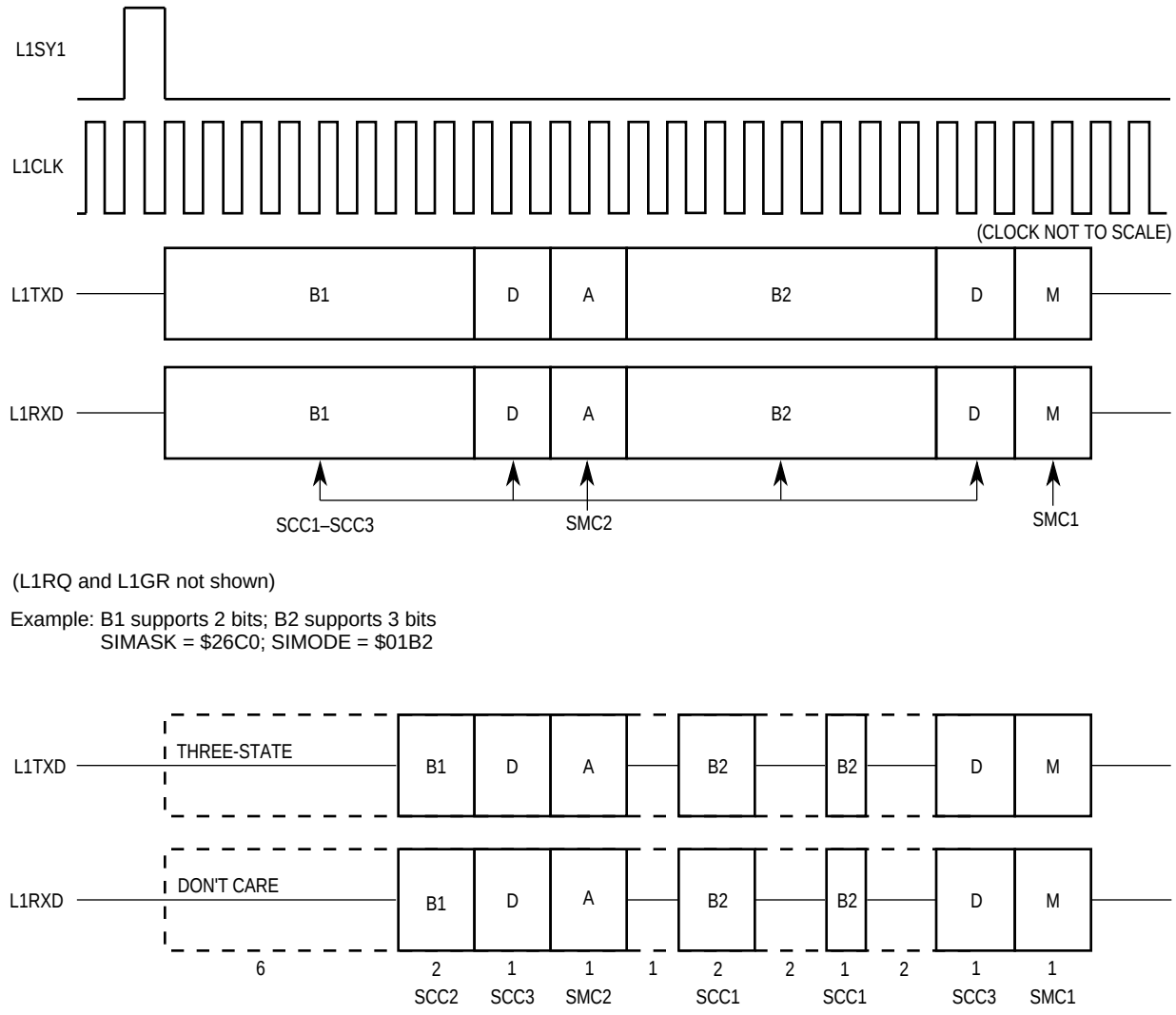


Figure 4-6. IDL Bus Signals

An application of the IDL interface is to build a basic rate ISDN terminal adaptor (see Figure 4-7). In such an application, the IDL interface is used to connect the 2B + D channels between the IMP, CODEC, and S/T transceiver. One of the IMP SCCs would be configured to HDLC mode to handle the D channel; another IMP SCC would be used to rate adapt the

NMSI mode. The SIMODE register is a memory-mapped read-write register cleared by reset.

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

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

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

0 = Normal operation
1 = L1TXD output set to a logic zero (used in GCI activation, refer to 4.4.2 GCI Interface)

SYNC/SCIT—SYNC Mode/SCIT Select Support

SYNC is valid only in PCM mode.

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

The SCIT (Special Circuit Interface T) interface mode is valid only in GCI mode.

0 = SCIT support disabled
1 = SCIT D-channel collision enabled. Bit 4 of channel 2 C/I used by the IMP for receiving indication on the availability of the S interface D channel.

SDIAG1–SDIAG0—Serial Interface Diagnostic Mode (NMSI1 Pins Only)

00 = Normal operation
01 = Automatic echo
The channel automatically retransmits the received data on a bit-by-bit basis. The receiver operates normally, but the transmitter can only retransmit received data. In this mode, L1GR is ignored.
10 = Internal loopback
The transmitter output (L1TXD) is internally connected to the receiver input (L1RXD). The receiver and the transmitter operate normally. Transmitted data appears on the L1TXD pin, and any external data received on L1RXD pin is ignored. In this mode, L1RQ is asserted normally, and L1GR is ignored.
11 = Loopback control
In this mode, the transmitter output (TXD1/L1TXD) is internally connected to the receiver input (RXD1/L1RXD). The TXD1/L1TXD, TXD2, TXD3, $\overline{\text{RTS1}}$, $\overline{\text{RTS2}}$, and $\overline{\text{RTS3}}$ pins will be high, but L1TXD will be three-stated in IDL and PCM modes. This mode may be used to accomplish multiplex mode loopback testing without affecting the multiplexed layer 1 interface. It also prevents an SCC's individual loopback (configured in the SCM) from affecting the pins of its associated NMSI interface.

for an internal clock, TCS and RCS may both be zero, or, for an external clock, they may both be one. The other two combinations are not allowed in this mode.

NOTE

If external loopback is desired (i.e., external to the MC68302), then the DIAG1–DIAG0 bits should be set for either normal or software operation, and an external connection should be made between the TXD and RXD pins. Clocks may be generated internally, externally, or an internally generated TCLK may be externally connected to RCLK. If software operation is used, the $\overline{\text{RTS}}$, $\overline{\text{CD}}$, and $\overline{\text{CTS}}$ pins need not be externally connected. If normal operation is used, the $\overline{\text{RTS}}$ pin may be externally connected to the $\overline{\text{CD}}$ pin, and the $\overline{\text{CTS}}$ pin may be grounded.

NOTE

Do not use this mode for loopback operation of IDL in the Serial Interface. Instead program the diag bits to Normal Operation, and (1) assert the L1GR pin externally from the S/T chip, or (2) configure the SDIAG1-0 bits in the SIMODE to Internal Loopback or Loopback Control.

10 = Automatic echo

In this mode, the channel automatically retransmits the received data on a bit-by-bit basis. The receiver operates normally, but the transmitter simply retransmits the received data. The $\overline{\text{CD}}$ pin must be asserted for the receiver to receive data, and the $\overline{\text{CTS}}$ line is ignored. The data is echoed out the TXD pin with a few nanosecond delay from RXD. No transmit clock is required, and the ENT bit in the SCC mode register does not have to be set.

NOTE

The echo function may also be accomplished in software by receiving buffers from an SCC, linking them to transmit buffer descriptors, and then transmitting them back out of that SCC.

11 = Software operation (CTS, CD lines under software control)

In this mode, the CTS and CD lines are just inputs to the SCC event (SCCE)and status (SCCS) registers. The SCC controller does not use these lines to enable/disable reception and transmission, but leaves low (i.e., active) in this mode. Transmission delays from RTS low are zero TCLKs (asynchronous protocols) or one TCLK (synchronous protocols).

NOTE

The MC68302 provides several tools for enabling and disabling transmission and/or reception. Choosing the right tool is application and situation dependent. For the receiver, the tools are 1) the empty bit in the receive buffer descriptor, 2) the ENR bit, and 3) the ENTER HUNT MODE command. For the transmitter, the

The SCC should be disabled and re-enabled if any change is made to the SCC's parallel I/O or serial channels physical interface configuration. The SCC does not need to be disabled if only a change to a parameter RAM value is made. See 4.5.6 SCC Parameter RAM Memory Map for a discussion of when parameter RAM values may be modified.

To save power, the SCCs may simply be disabled. Clearing the enable transmitter (ENT) bit in the SCC mode register causes the SCC transmitter to consume the least possible power; clearing the ENR bit causes a similar action for the SCC receiver.

The above statement on saving power is independent of a decision to use the low-power modes (see 3.8 System Control). If a low-power mode is desired, the SCC may be disabled before entering the low-power mode, or it may be left enabled so that an SCC interrupt may bring the IMP out of the low-power mode. One common use of the low-power mode is to disable the transmitter but leave the receiver enabled (i.e., in the hunt mode) so that an arriving frame destined for this station will cause an interrupt, waking the IMP from its low-power mode.

The low-power mode affects the M68000 core, not the SCCs. Since the SCCs are usually clocked at a far lower rate than the M68000 core, significant power savings may still be achieved with the SCCs fully enabled and the M68000 core in the low-power mode.

4.5.11 UART Controller

Many applications need a simple method of communicating low-speed data between equipment. The universal asynchronous receiver transmitter (UART) protocol is the *de-facto* standard for such communications. The term asynchronous is used because it is not necessary to send clocking information with the data that is sent. UART links are typically 38400 baud or less in speed and are character oriented (i.e., the smallest unit of data that can be correctly received or transmitted is a character). Typical applications of asynchronous links are connections between terminals and computer equipment. Even in applications where synchronous communication is required, the UART is often used for a local debugging port to run board debugger software. The character format of the UART protocol is shown in Figure 4-17.

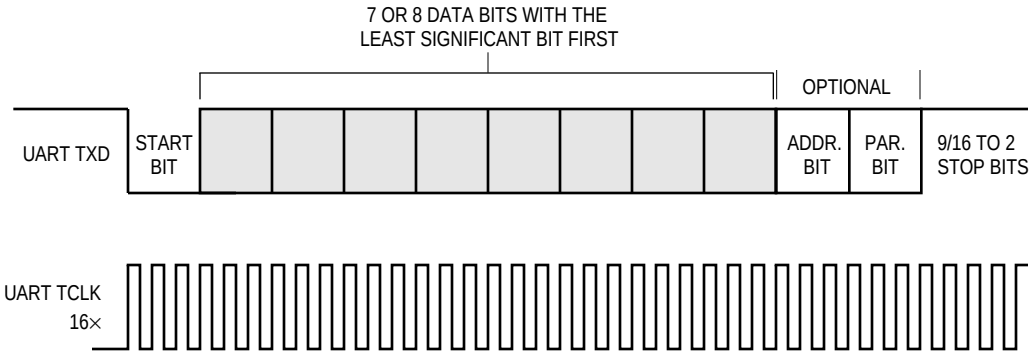


Figure 4-17. UART Frame Format

cludes it from the BCS. If the second character is a DLE, the BISYNC controller will write it to the buffer and include it in the BCS. If the character is not a DLE or SYNC, the BISYNC controller will examine the control characters table and act accordingly. If the character is not in the table, the buffer will be closed with the DLE follow character error (DL) bit set. If the V bit is not set, the receiver will treat the character as a normal character.

NOTE

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V	0	0	0	0	0	0	0	DLE							

4.5.13.8 BISYNC Error-Handling Procedure

The BISYNC controller reports message reception and transmission error conditions using the channel BDs, the error counters, and the BISYNC 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 resumes transmission after the reception of the RESTART TRANSMIT command. Underrun cannot occur between frames or during a DLE-XXX pair in transparent mode. The FIFO size is three bytes in BISYNC.
2. Clear-To-Send Lost During Message Transmission. When this error occurs and the channel is not programmed to control this line with software, the channel terminates buffer transmission, closes the buffer, sets the CTS lost (CT) bit in the BD, and generates the TXE interrupt (if enabled). The channel will resume transmission after the reception of the RESTART TRANSMIT command.

Reception Errors:

1. Overrun Error. The BISYNC controller maintains an internal three-byte FIFO for receiving data. The CP begins programming the SDMA channel (if the data buffer is in external memory) and updating the CRC when the first word is received into the FIFO. If a FIFO overrun occurs, the BISYNC controller writes the received data byte to the internal FIFO over the previously received byte. The previous character and its status bits are lost. Following this, the channel closes the buffer, sets the overrun (OV) bit in the BD, and generates the RX interrupt (if enabled). The receiver then enters hunt mode immediately.
2. Carrier Detect Lost During Message Reception. When this error occurs and the channel is not programmed to control this line with software, the channel terminates message reception, closes the buffer, sets the carrier detect lost (CD) bit in the BD, and generates the RX interrupt (if enabled). This error is the highest priority; the rest of the message is lost and no other errors are checked in the message. The receiver then enters hunt mode immediately.
3. Parity Error. When this error occurs, the channel writes the received character to the

transmission, message transmission is aborted after the contents of the FIFO (up to three bytes) are transmitted. The TBD# is not advanced. No new BD is accessed, and no new messages are transmitted for this channel. Upon receipt of this command, the transmitter aborts the message transmission (if currently transmitting) and then transmits SYN1–SYN2 pairs or IDLEs as determined by the DDCMP mode register.

The STOP TRANSMIT command must be issued before the SCC mode register is used to disable the transmitter if the transmitter is to be re-enabled at a later time.

RESTART TRANSMIT Command

The RESTART TRANSMIT command re-enables the transmission of characters on the transmit channel. This command is expected by the DDCMP controller after a STOP TRANSMIT command, after a STOP TRANSMIT command followed by the disabling of the channel in its SCC mode register, or after a transmitter error (underrun or CTS lost during data or maintenance message header fields). The DDCMP controller will resume transmission from the current transmitter BD number (TBD#) in the channel's transmit BD table.

If the channel is being re-enabled, the RESTART TRANSMIT command must be used and should be followed by the enabling of the transmitter in the SCC mode register.

ENTER HUNT MODE Command

After a hardware or software reset and the enabling of the channel in the SCC mode register, the channel is in the receive enable mode and will use the first BD in the table.

The ENTER HUNT MODE command is used to force the DDCMP controller to abort reception of the current message, generate an RBD interrupt (if enabled) as the buffer is closed, and enter hunt mode. In hunt mode, the DDCMP controller continually scans the input data stream for the SYN1–SYN2 sequence on synchronous links. Then for synchronous or asynchronous links, the DDCMP controller scans the input bytes for the starting byte of one of the messages. After receiving the command, the current receive buffer is closed, and the CRC is reset. Message reception continues using the next BD.

If an enabled receiver has been disabled (by clearing ENR in the SCC mode register), the ENTER HUNT MODE command must be given to the channel before setting ENR again.

4.5.14.6 DDCMP Control Character Recognition

The DDCMP controller can recognize three special control characters. These characters are used to synchronize the message and allow the DDCMP controller to function in a DMA-controlled environment.

DSYN1—DDCMP Sync Character Register

The 8-bit DSYN1 register should be written with the same value that was written in the SYN1 byte of the data synchronization register (DSR). DSYN1 is a memory-mapped read-write register.

NOTE

For correct operation of DDCMP, DSYN1, SYN1, and SYN2 must be the same value.

Bits 11, 9–8—Reserved for future use.

V.110—V.110 Mode

0 = DDCMP mode; synchronous DDCMP is chosen.
1 = V.110 mode; the V.110 protocol description is in 4.5.15 V.110 Controller.

SYNF—Transmit SYN1–SYN2 or IDLE between Messages and Control the RTS Pin

0 = Send ones between messages. $\overline{\text{RTS}}$ is negated between messages.

NOTE

The DDCMP controller can transmit ones in both NRZ and NRZI data encoded formats. The minimum number of ones transmitted is 17.

1 = Send SYN1–SYN2 pairs between messages. $\overline{\text{RTS}}$ is always asserted. Note that SYN1 and SYN2 may be the same character.

ENC—Data Encoding Format

0 = Nonreturn to Zero (NRZ). A one is a high level; a zero is a low level.
1 = Nonreturn to Zero Inverted (NRZI). A one is represented by no change in the level; a zero is represented by a change in the level. The receiver decodes NRZI, but a clock must be supplied. The transmitter encodes NRZI.

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.

4.5.14.10 DDCMP Receive Buffer Descriptor (Rx BD)

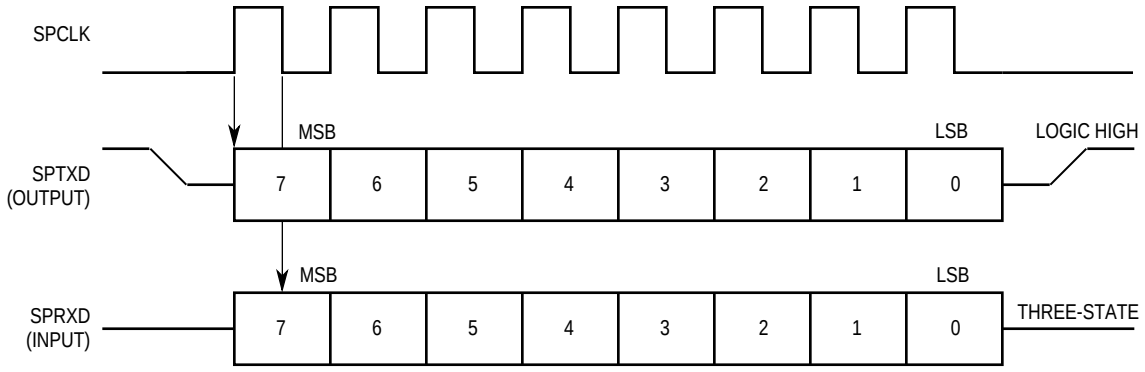
The CP reports information about the received data for each buffer using the BDs. The Rx BD is shown in Figure 4-36. The CP closes the current buffer, generates a maskable interrupt, and starts to receive data in the next buffer after any of the following events:

- Receiving the received message length number of bytes (RMLG)
- Detecting an error
- Detecting a full receive buffer
- Issuing the ENTER HUNT MODE command

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	E	X	W	I	L	H	T2	T1	—	—	CF	FR	PR	CR	OV	CD
OFFSET + 2	DATA LENGTH															
OFFSET + 4	RX BUFFER POINTER (24-bits used, upper 8 bits must be 0)															
OFFSET +6																

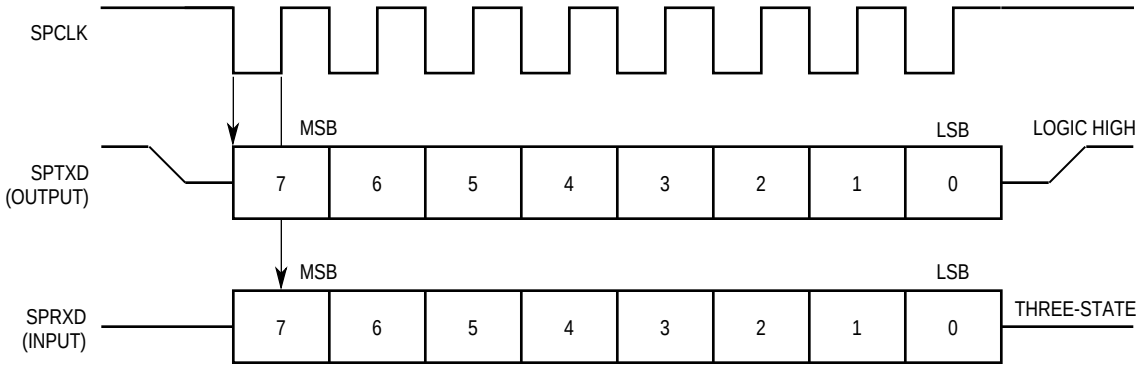
Figure 4-36. DDCMP Receive Buffer Descriptor

The first word of the Rx BD contains control and status bits. Bits 15–12 are written by the user before the buffer is linked to the Rx BD table, and bits 5–0 and 11–8 are set by the IMP



NOTE: Transmitted data bits shift on rising edges; received bits are sampled on falling edges.

(a) CI=0



NOTE: Transmitted data bits shift on falling edges; received bits are sampled on rising edges.

(b) CI=1

Figure 4-44. SCP Timing

The SCP can be configured to operate in a local loopback mode, which is useful for local diagnostic functions.

Note that the least significant bit of the SCP is labeled as data bit 0 on the serial line; whereas, other devices, such as the MC145554 CODEC, may label the most significant bit as data bit 0. The MC68302 SCP bit 7 (most significant bit) is shifted out first.

The SCP key features are as follows:

- Three-Wire Interface (SPTXD, SPRXD, and SPCLK)
- Full-Duplex Operation
- Clock Rate up to 4.096 MHz
- Programmable Clock Generator
- Local Loopback Capability for Testing

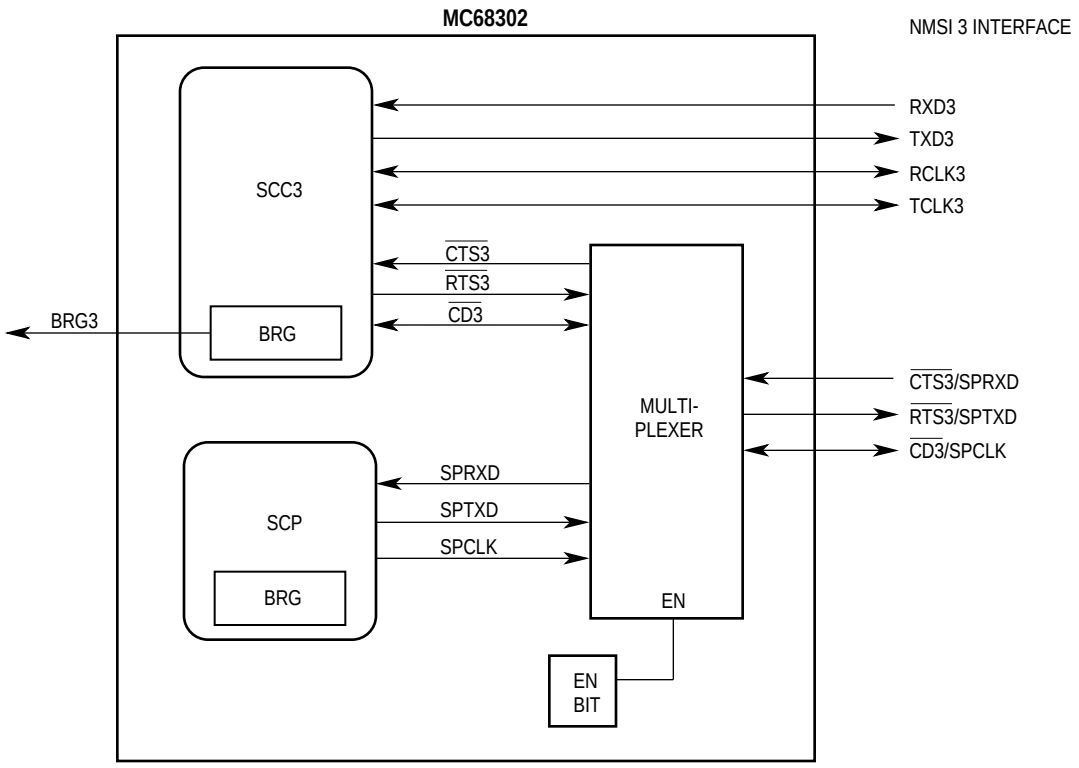


Figure 4-45. SCP vs. SCC Pin Multiplexing

4.6.2 SCP Transmit/Receive Buffer Descriptor

The transmit/receive BD contains the data to be transmitted (written by the M68000 core) and the received data (written by the SCP). The done (D) bit indicates that the received data is valid and is cleared by the SCP.

15	14 … 8	7 … 0
D	RESERVED	DATA

4.6.3 SCP Transmit/Receive Processing

The IMP SCP always functions in the master mode. Thus, in a typical exchange of messages, the IMP transmits a message to an external peripheral (SCP slave) which, in turn, sends back a reply. When the IMP works with more than one slave, it can use the general-purpose parallel I/O pins as enable (select) signals. To begin the data exchange, the M68000 core writes the data to be transmitted into the transmit/receive BD, and sets the done bit. The M68000 core should then set the start transmit (STR) bit in the SPMODE register to start transmission of data. STR is cleared by hardware after one system clock cycle.

Upon recognizing the STR bit, the SCP also begins receiving eight bits of data. It writes the data into the transmit/receive BD, clears the done bit, and issues a maskable interrupt to the IMP interrupt controller. When working in a polled environment, the done bit should be set

$\overline{CS3}$–$\overline{CS1}$—Chip Selects 3–1

These three active-low output pins function as chip selects for external devices or memory. $\overline{CS3}$—$\overline{CS0}$ do not activate on accesses to the internal RAM or registers (including the BAR SCR, or CKCR registers).

5.21 NO-CONNECT PINS

NC1 and NC3 output high values and are reserved for future use.

5.22 WHEN TO USE PULLUP RESISTORS

Pins that are output-only do not require external pullups. The bidirectional bus control signals require pullups since they are three-stated by the MC68302 when they are not being driven. Open-drain signals always require pullups.

Unused inputs should not be left floating. If they are input-only, they may be tied directly to V_{CC} or ground, or a pullup or pulldown resistor may be used. Unused outputs may be left unconnected. Unused I/O pins may be configured as outputs after reset and left unconnected.

If the MC68302 is to be held in reset for extended periods of time in an application (other than what occurs in normal power-on reset or board test sequences) due to a special application requirement (such as V_{DD} dropping below required specifications, etc.), then three-stated signals and inputs should be pulled up or down. This decreases stress on the device transistors and saves power.

See the $\overline{RESET}$ pin description for the condition of all pins during reset.

D.5.6 External Cycles Examples

If the MC68302 is the current bus master and no other internal or external resources are arbitrating for the bus, then the IDMA will obtain bus mastership and perform the data movement cycles when the $\overline{\text{DREQ}}$ signal meets the asynchronous setup time prior to the falling edge of clock.

Figure D-7 shows the sequence of a peripheral requesting a data transfer, the IDMA reading data from memory and then writing the data to the peripheral. The source and destination size are the same for this case. $\overline{\text{DREQ}}$ is sampled on the falling edge of CLK and causes the $\overline{\text{BGACK}}$ signal to assert at the conclusion of the current M68000 core bus cycle. The IDMA performs a read cycle using the SAPR to obtain data and then performs a write cycle to place data in the address specified in the DAPR. The fact that $\overline{\text{DACK}}$ asserts in the write cycle indicates that data is being transferred to the requesting peripheral. Note that the $\overline{\text{BR}}$ and $\overline{\text{BG}}$ pins are not affected by the IDMA in this example. They only activate when the device is in the disable CPU mode (see 3.8.4 Disable CPU Logic (M68000)).

D.5.8 Final Notes

Only three signals, $\overline{\text{DREQ}}$, $\overline{\text{DACK}}$, and $\overline{\text{DONE}}$, are used to control the handshake between the peripheral and the IDMA. $\overline{\text{DONE}}$ is not needed unless the number of bytes to transmit is not known. The $\overline{\text{DACK}}$ signal can select the peripheral device when data is being transferred to or from it, and wait-state logic can generate $\overline{\text{DTACK}}$ to indicate when valid data is on the bus. Software controls data transfer by setting bits in the CMR as shown in Table D-2 and loading the SAPR and DAPR. It then waits for completion of the transfer by either monitoring the bits in the CSR (see Table D-3) or by using interrupt processing.

D.6 MC68302 MULTIPROTOCOL CONTROLLER TIED TO IDL BUS FORMS AND ISDN VOICE/DATA TERMINAL

The following paragraphs discuss how the MC68302 can be tied to the interchip digital link (IDL) bus, which enables connectivity to a family of ISDN chips. The IDL bus connects the MC68302 integrated controller with the MC145475 S/T interface and the MC145554 CO-DEC to form a basic rate ISDN voice/data terminal (see Figure D-11).

The MC68302 is the first device to combine the benefits of the M68000 microprocessor with a flexible communications architecture. This CMOS device incorporates an MC68000 or MC68008 core processor, a communications RISC processor with associated peripherals, and a system integration block.

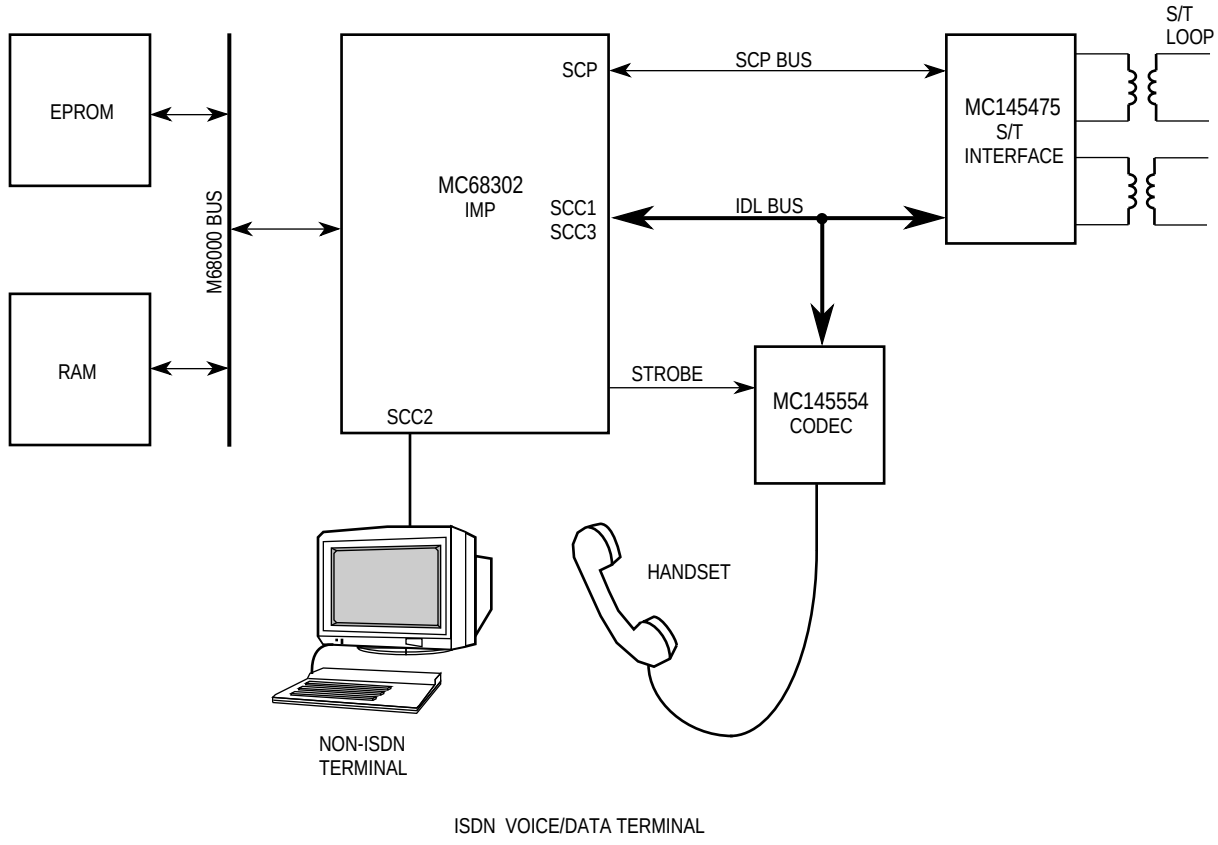


Figure D-11. ISDN Voice/Data Terminal

D.6.7 Serial Interface Configuration

To allow the MC68302 to be tied to the IDL bus, the serial interface (see Figure D-15) must be configured to the IDL mode. A value of $0C76 written to the SIMR will set the following configuration: IDL mode, D-channel routed to SCC3, B1-channel routed to SCC1, SCC2 working in NMSI mode (routed to its external pins), and SDS1 enveloping B2-channel (CODEC strobe).

This configuration is suitable for terminal adaptor configuration, in which the non-ISDN terminal is connected to SCC2. The output/input frames of the rate adaption protocol are transmitted/received by SCC1 over B1 channel, and SCC3 handles the signaling over the D-channel (call establishment, disconnect, etc.). The B2 channel is routed to a CODEC by the serial data strobe 1 and is used for voice calls

The serial interface mask register allows selection of subrates of the 64-kbps B-channel capacity by selecting the used bits of the eight B-channel bits.

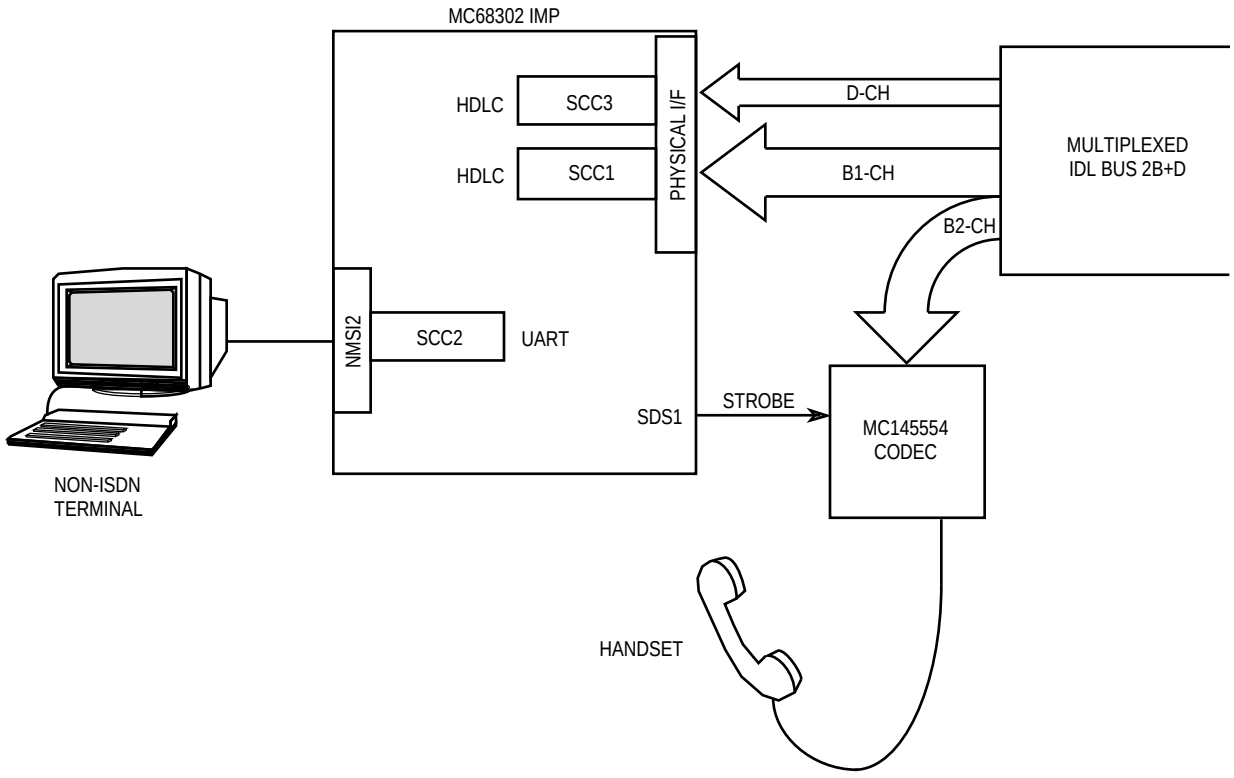


Figure D-15. Serial Interface Configuration

D.9.1 Overview of the Board

The board interfaces two of the MC68302 SCCs to the AppleTalk LAN. SCC1 and SCC2 are used for this example, but any of the three SCCs could have been originally chosen for use with the board. The MC68195 LA provides the FM0 encoding/decoding and digital phase-locked loop functionality. It also provides a direct interface to the MC68302 and the line driver/receiver chips, as shown. The LA input clock is required to be 10x the desired data rate. Thus, for AppleTallk, a 2.304-MHz crystal was used.

The physical portion of the board was modeled after the Macintosh™ Plus serial interface. Thus, the RS-422 driver/receiver function was implemented with the 26LS32 receiver and 26LS30 line driver. The connectors used were the standard mini-DIN 8, which is the same as those used in the Macintosh. The connections to this connector followed the Macintosh Plus serial interface diagram, except that HSKo (pin 1) was simply pulled high through a 100 Ω resistor. This does not inhibit LocalTalk functionality.

D.9.2 Important Side Notes

The reset circuit chosen was a simple RC delay. Normally, the reset function would be part of the system reset circuitry, but this function was not available through the original connector on the ADS302 board, so it was built on the LA board. The LA has an internal Schmitt trigger on the reset input to facilitate this configuration.

Note that there is a pullup on the SCC2 $\overline{\text{RTS2}}$ pin because SCC2 on the MC68302, unlike SCC1, has its pins multiplexed with parallel I/O pins. These pins default to the input state upon reset. If this pullup is omitted, the $\overline{\text{RTS2}}$ pin might be low between the moment of reset and the moment at which the software configures this pin to the $\overline{\text{RTS2}}$ function (i.e., writing the PACNT register). During this window of time, a low value on $\overline{\text{RTS2}}$ would cause the LA to illegally transfer out onto the LocalTalk network. The pullup safely avoids this problem. A pullup is not needed on the MC68302 $\overline{\text{RTS1}}$ or $\overline{\text{RTS3}}$ since they reset to the inactive (high) state.

Five parallel I/O signals (PA7-PA11) are used to configure the LA into various loopback, bypass, or clock enable modes, making it easy for the MC68302 to put the LA into various modes for testing. In a final implementation, some of these signals could be pulled directly high or low. As previously described, these signals float until initialized in software by the MC68302; however, since these briefly floating inputs do not cause a problem in this system, pullups and pulldowns were not added.

The channel enable signals ($\overline{\text{CHEN}}$) were pulled continuously high in this application since there was no need to disable LA operation. In a final system, these could be easily connected to MC68302 I/O pins as needed.

·The general-purpose inputs (GPI) were simply pulled high fort his example. They could have been used to support asynchronous operation over the mini-DIN 8 connector, if desired.

AppleTalk is a registered trademark of Apple Computer, Inc.
Macintosh is a trademark of Apple Computer, Inc.

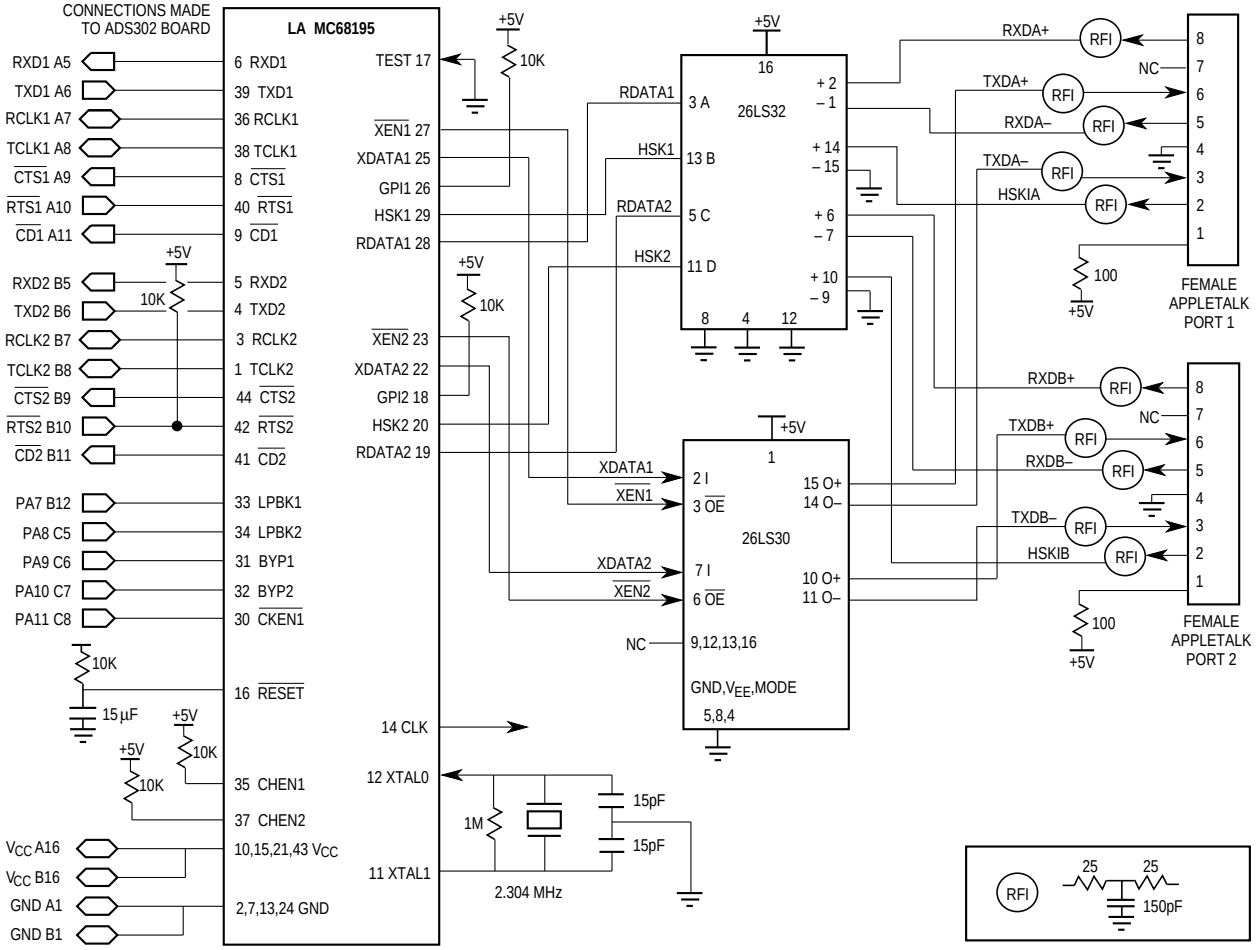


Figure D-32. Local Talk Adaptor Board

Table E-1 (a). HDLC Programming Mode
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). HDLC Programming Model (Continued)
General Parameter and HDLC Protocol-Specific RAM for SCCx

Initialized by User	Offset Hex	Name		Initialized by User	Offset Hex	Name
Yes	80	RFCR	TFCR	Yes	A2	CRC_Mask_H
Yes	82	MRBLR			A4	Temp Transmit CRC Low
	84	Rx Internal State			A6	Temp Transmit CRC High
	86	Reserved	Rx Internal Buffer No.	Yes	A8	DISFC
	88 8A	Rx Internal Data Pointer (High Word) Rx Internal Data Pointer (Low Word)		Yes	AA	CRCEC
	8C	Rx Internal Byte Count		Yes	AC	ABTSC
	8E	Rx Temp		Yes	AE	NMARC
	90	Tx Internal State		Yes	B0	RETRC
	92	Reserved	Tx Internal Buffer No.	Yes	B2	MFLR
	94 96	Tx Internal Data Pointer (High Word) TX Internal Data Pointer (Low Word)			B4	MAX_cnt
	98	Tx Internal Byte Count		Yes	B6	HMASK
	9A	Tx Temp		Yes	B8	HADDR1
	9C	Temp Receive CRC Low		Yes	BA	HADDR2
	9E	Temp Receive CRC High		Yes	BC	HADDR3
Yes	A0	CRC_Mask_L		Yes	BE	HADDR4

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