Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	M68000
Number of Cores/Bus Width	1 Core, 8/16-Bit
Speed	16MHz
Co-Processors/DSP	Communications; RISC CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	3.3V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/kmc68302ag16vc

CFC—Compare Function Code

0 = The FC bits in the BAR are ignored. Accesses to the IMP 4K-byte block occur without comparing the FC bits.
1 = The FC bits in the BAR are compared. The address space compare logic uses the FC bits to detect address matches.

Bits 11–0—Base Address

The high address field is contained in bit 11–0 of the BAR. These bits are used to set the starting address of the dual-port RAM. The address compare logic uses only the most significant bits to cause an address match within its block size.

2.8 MC68302 MEMORY MAP

The following tables show the additional registers added to the M68000 to make up the MC68302. All of the registers are memory-mapped. Four entries in the M68000 exception vectors table (located in low RAM) are reserved for addresses of system configuration registers (see Table 2-6) that reside on-chip. These registers have fixed addresses of $0F0–$0FF. All other on-chip peripherals occupy a 4K-byte relocatable address space. When an on-chip register or peripheral is accessed, the internal access (IAC) pin is asserted.

Table 2-6. System Configuration Register

Address	Name	Width	Description	Reset Value
$0F0	RES	16	Reserved	
$0F2*	BAR	16	Base Address Register	BFFF
$0F4*	SCR	32	System Control Register	0000 0F00
$0F8	RES	16	Reserved	
$0FA	CKCR	16	Clock Control Register	0000
$0FC	RES	32	Reserved	

*Reset only upon a total system reset.

The internal 1176-byte dual-port RAM has 576 bytes of system RAM (see Table 2-7) and 576 bytes of parameter RAM (see Table 2-8).

Table 2-7. System RAM

Address	Width	Block	Description
Base + 000 • • • Base + 23F	576 Bytes	RAM	User Data Memory
Base +240 • • • Base + 3FF			Reserved (Not Implemented)

The parameter RAM contains the buffer descriptors for each of the three SCC channels, the SCP, and the two SMC channels. The memory structures of the three SCC channels are

NOTE

While in disable CPU mode, during the host processor interrupt acknowledge cycle for $\overline{IRQ1}$, if $\overline{IRQ1}$ is not continuously asserted, the interrupt controller will still provide the vector number (and $\overline{DTACK}$) according to the IV1 bit. The $\overline{IACK6}$ falling edge can be used externally to negate $\overline{IRQ1}$.

1 = Edge-triggered. An interrupt is made pending when $\overline{IRQ1}$ changes from one to zero (falling edge).

V7–V5—Interrupt Vector Bits 7–5

These three bits are concatenated with five bits provided by the interrupt controller, which indicate the specific interrupt source, to form an 8-bit interrupt vector number. If these bits are not written, the vector $0F is provided.

Note:

These three bits should be greater than or equal to '010' in order to put the interrupt vector in the area of the exception vector table for user vectors.

Bits 11 and 4–0—Reserved for future use.

3.2.5.2 Interrupt Pending Register (IPR)

Each bit in the 16-bit IPR corresponds to an INRQ interrupt source. When an INRQ interrupt is received, the interrupt controller sets the corresponding bit in the IPR.

In a vectored interrupt environment, the interrupt controller clears the IPR bit when the vector number corresponding to the INRQ interrupt source is passed to the M68000 core during an interrupt acknowledge cycle, unless an event register exists for that INRQ interrupt. In a polled interrupt scheme, the user must periodically read the IPR. When a pending interrupt is handled, the user should clear the corresponding bit in the IPR by writing a one to that bit. (If an event register exists, the unmasked event register bits should be cleared instead, causing the IPR bit to be cleared.) Since the user can only clear bits in this register, the bits that are written as zeros will not be affected. The IPR is cleared at reset.

NOTE

The ERR bit is set if the user drives the $\overline{IPL2}$–$\overline{IPL0}$ lines to interrupt level 4 and no INRQ interrupt is pending.

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

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

to the DRAM bank. The PAL generates the RAS and CAS lines for the DRAM chips and controls the address multiplexing in the external address buffers. One of the MC68000 chip-select lines can be used as the DRAM bank enable signal, if desired.

The refresh operation is a byte read operation. Thus, $\overline{UDS}$ or $\overline{LDS}$ will be asserted from the MC68302, but not both. A refresh to an odd address will assert $\overline{LDS}$; whereas, a refresh to an even address will assert $\overline{UDS}$.

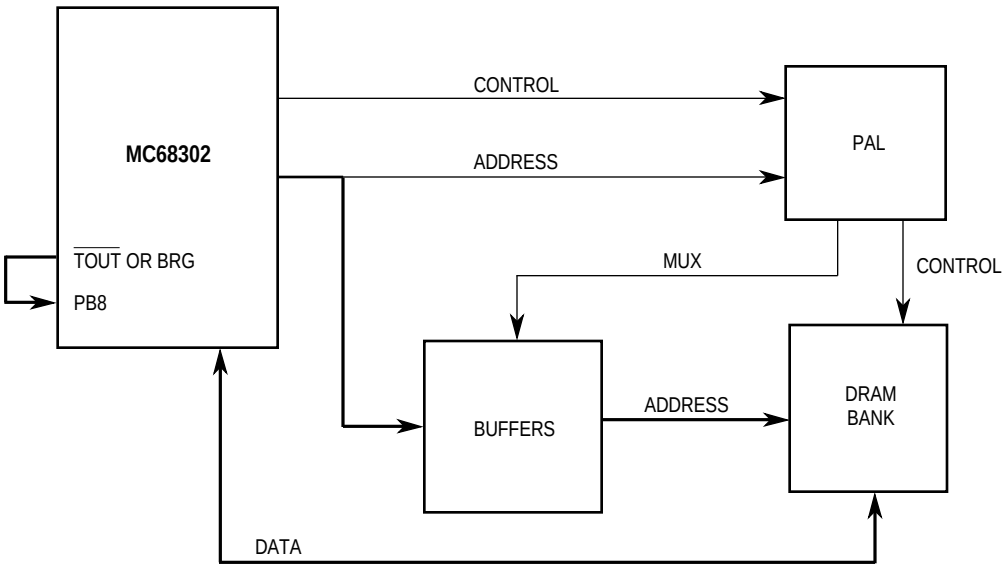


Figure 3-13. DRAM Control Block Diagram

3.10.2 DRAM Refresh Controller Bus Timing

The DRAM refresh controller bus cycles are actually SDMA byte read accesses (see 4.2 SDMA Channels for more details). All timings, signals, and arbitration characteristics of SDMA accesses apply to the DRAM refresh controller accesses. For example, DRAM refresh cycles activate the $\overline{BCLR}$ signal, just like the SDMA. Note that the function code bits may be used to distinguish DRAM refresh cycles from SDMA cycles, if desired.

A bus error on a DRAM refresh controller access causes the $\overline{BERR}$ channel number at offset BASE + $67C to be written with a $0001. This is also the value written if the SCC1 receive SDMA channel experiences a bus error; thus, these two sources cannot be distinguished upon a bus error. The DRAM refresh SDMA channel and SCC1 receive SDMA channel are separate and independent in all other respects.

3.10.3 Refresh Request Calculations

A typical 1-Mbyte DRAM needs one refresh cycle every 15.625 µs. The DRAM refresh controller is configured to execute one refresh cycle per request; thus, the PB8 pin should see a high-to-low transition every 15.625 µs. This is once every 260 cycles for a 16.67-MHz clock. Note that one refresh per request minimizes the speed loss on the SCC channels.

	15	0
OFFSET + 0	STATUS AND CONTROL	
OFFSET + 2	DATA LENGTH	
OFFSET + 4	HIGH-ORDER DATA BUFFER POINTER (only lower 8 bits used, upper 8 bits must be 0)	
OFFSET + 6	LOW-ORDER DATA BUFFER POINTER	

Figure 4-16. SCC Buffer Descriptor Format

For frame-oriented protocols (HDLC, BISYNC, DDCMP, V.110), a frame may reside in as many buffers as are necessary (transmit or receive). Each buffer has a maximum length of 64K–1 bytes. The CP does not assume that all buffers of a single frame are currently linked to the BD table, but does assume that the unlinked buffers will be provided by the processor in time to be either transmitted or received. Failure to do so will result in a TXE error being reported by the CP.

For example, assume the first six buffers of the transmit BD table have been transmitted and await processing by the M68000 core (with all eight buffers used in the circular queue), and a three-buffer frame awaits transmission. The first two buffers may be linked to the remaining two entries in the table as long as the user links the final buffer into the first entry in the BD table before the IMP attempts its transmission. If the final buffer is not linked in time to the BD table by the time the CP attempts its transmission, the CP will report an underrun error.

Buffers allocated to an SCC channel may be located in either internal or external memory. Memory allocation occurs for each BD individually. If internal memory is selected, the CP uses only the lower 11 address bits (A10–A0) as an offset to the internal dual-port RAM. Accesses to the internal memory by the CP are one clock cycle long and occur without arbitration. If external memory is selected, the pointers to the data buffers are used by the CP as 24 bits of address.

Extra caution should be used if function codes are included in the decoding of the external buffer address (e.g., in the on-chip chip select logic). The function code of this SCC channel must be set before external buffers can be accessed; it can then be changed only when the user is sure that the CP is not currently accessing external buffers for that channel. There are six separate function code registers located in the parameter RAM for the three SCC channels: three for receive data buffers (RFCR) and three for transmit data buffers (TFCR).

NOTE

The RFCR and TFCR function codes should never be initialized to “111.”

The CP processes the transmit BDs in a straightforward fashion. Once the transmit side of an SCC is enabled, it starts with the first BD in that SCC's transmit BD table, periodically checking a bit to see if that BD is "ready". Once it is ready, it will process that BD, reading a word at a time from its associated buffer, doing certain required protocol processing on the data, and moving resultant data to the SCC transmit FIFO. When the first buffer has been processed, the CP moves on to the next BD, again waiting for that BD's “ready” bit to be set. Thus, the CP does no look-ahead BD processing, nor does it skip over BDs that are not ready. When the CP sees the “wrap” bit set in a BD, it goes back to the beginning of the BD

table, after processing of this BD is complete. After using a BD, the CP sets the “ready” bit to not-ready; thus, the CP will never use a BD twice until the BD has been confirmed by the M68000 core.

The CP uses the receive BDs in a similar fashion. Once the receive side of an SCC is enabled, it starts with the first BD in that SCC's receive BD table. Once data arrives from the serial line into the SCC, the CP performs certain required protocol processing on the data and moves the resultant data (either bytes or words at a time depending on the protocol) to the buffer pointed to by the first BD. Use of a BD is complete when there is no more room left in the buffer or when certain events occur, such as detection of an error or an end-of-frame. Whatever the reason, the buffer is then said to be “closed,” and additional data will be stored using the next BD. Whenever the CP needs to begin using a BD because new data is arriving, it will check the “empty” bit of that BD. If the current BD is not empty, it will report a” busy” error. However, it will not move from the current BD until it becomes empty. When the CP sees the “wrap” bit set in a BD, it goes back to the beginning of the BD table, after use of this BD is complete. After using a BD, the CP sets the “empty” bit to not-empty; thus, the CP will never use a BD twice until the BD has been “processed” by the M68000 core.

In general, each SCC has eight transmit BDs and eight receive BDs. However, it is possible in one special case to assign up to 16 receive BDs at the expense of all transmit BDs. Since the transmit BDs directly follow the receive BDs in the memory map for each SCC, if an SCC is configured exclusively for half-duplex reception, it is possible to have up to 16 receive BDs available for that SCC.

If the DRAM refresh unit is used, SCC2 has six transmit BDs rather than the normal eight. SCC3 normally only has four transmit BDs. However, it is actually possible to regain additional Tx BDs for SCC3 as follows. The Tx BD table may be extended by two BDs to six BDs if the SMCs are not used. Additionally, all eight Tx BDs for SCC3 may be used if the following is considered: 1) the SCP and SMCs must not be used; 2) various words within the last two BDs will be changed by the CP during the initialization routine following any reset; and 3) the BERR channel number value will be written into the last BD after any SDMA bus error (see 4.5.8.4 Bus Error on SDMA Access), but this is not a major concern since the CP must be reset after any SDMA bus error.

4.5.6 SCC Parameter RAM Memory Map

Each SCC maintains a section in the dual-port RAM called the parameter RAM. Each SCC parameter RAM area begins at offset $80 from each SCC base area ($400, $500, or $600) and continues through offset $BF. Refer to Table 2-8 for the placement of the three SCC parameter RAM areas. Part of each SCC parameter RAM (offset $80–$9A), which is identical for each protocol chosen, is shown in Table 4-6. Offsets $9C–$BF comprise the protocol-specific portion of the SCC parameter RAM and are discussed relative to the particular protocol chosen.

gardless of what happens externally. This signifies that the corresponding SCCS bit is now valid.

7	3	2	1	0
RESERVED		ID	CD	CTS

Bits 7–3—Reserved for future use.

ID—Idle Status on the Receiver Line

This bit is meaningful only if the SCC is programmed to HDLC or UART mode. In HDLC mode, this bit is a one after 15 continuous ones are received on the line. This bit will be zero after a single zero occurs on the line. If flags, rather than idles, are received between frames, the ID bit will remain zero between frames.

In UART mode, this bit is a one after one idle character (9 to 13 bits) is received and is a zero after a single zero occurs on the line (e.g., a start bit).

If the DIAG1–DIAG0 bits in the SCM are programmed to normal mode, then the $\overline{CD}$ signal is an enable signal for ID status. In this case, if $\overline{CD}$ is not asserted, the ID bit will always be one, regardless of the activity on the line.

If the DIAG1–DIAG0 bits in the SCM are programmed to software operation mode, then the ID bit will always reflect line activity, regardless of the state of the $\overline{CD}$ pin.

The ID bit is valid in both the multiplexed and nonmultiplexed modes, once the ENR bit is set.

$\overline{CD}$—Carrier Detect Status on the Channel Pin

This bit has the same polarity as the external pin. In the multiplexed modes, it is always zero. $\overline{CD}$ is undefined until ENR is set.

$\overline{CTS}$ —Clear-to-Send Status on the Channel Pin.

This bit has the same polarity as the external pin. In the PCM highway mode, it is always zero. In the GCI and IDL mode, if the SCC is connected to the D channel, then this bit is valid; otherwise, it is always zero. $\overline{CTS}$ is undefined until the ENT bit is set.

When the $\overline{CTS}$ and $\overline{CD}$ lines are programmed to software control in the SCC mode register, these lines do not affect the SCC and can be used for other purposes such as a data set ready (DSR), a data terminal ready (DTR) line, or an interrupt source in the SCCE register according to the behavior just described.

4.5.8.4 Bus Error on SDMA Access

When a bus error occurs on an access by the SDMA channel, the CP generates a unique interrupt (see 3.2 Interrupt Controller). The interrupt service routine should read the bus error channel number from the parameter RAM at BASE + 67C as follows:

0—SCC1 Tx Channel

1—SCC1 Rx Channel or DRAM Refresh Cycle

2—SCC2 Tx Channel

3—SCC2 Rx Channel

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.

CT—Clear-to-Send Lost

This status bit indicates that the $\overline{CTS}$ signal was negated during transmission of this character. If this occurs, the CTS bit in the UART event register will also be set.

NOTE

If the $\overline{CTS}$ signal was negated during transmission, and the CP transmits this character in the middle of buffer transmission, the $\overline{CTS}$ signal could actually have been negated either during this character's transmission (i.e., CHARACTER8) or during a buffer character's transmission. In this case, the CP sets the CT bit both here and in the Tx BD status word.

A—Address

When working in a multidrop configuration, the user should include the address bit in this position.

CHARACTER8—Flow Control Character Value

Any 7- or 8-bit character value may be transmitted. This value may be modified only while the REA bit is cleared. A 7-bit character should comprise bits 6–0.

4.5.11.8 Send Break

A break is an all-zeros character without stop bits—i.e., 9 to 13 continuous zeros. A break is sent by issuing the STOP TRANSMIT command. The UART completes transmission of any outstanding data in the FIFO and then sends 9 to 13 zeros (depending on the UM1–UM0, SL, PEN, and CL bits in the UART mode register). The UART transmits a programmable number of break characters according to the value of the break count register (BRKCR), and then reverts to idle or sends data if the RESTART TRANSMIT command was given before completion. Upon transmission of the entire set of break characters, the transmitter sends at least one high bit before transmitting any data to guarantee recognition of a valid start bit.

4.5.11.9 Send Preamble (IDLE)

A preamble sequence gives the programmer a convenient way of ensuring that the line goes idle before starting a new message. The preamble sequence length is 9 to 13 consecutive ones (depending on the UM1–UM0, SL, PEN, and CL bits in the UART mode register). If the preamble bit in a BD is set, the SCC will send a preamble sequence before transmitting that data buffer.

4.5.11.10 Wakeup Timer

By issuing the ENTER HUNT MODE command, the user can temporarily disable the UART receiver. It will remain inactive until an idle or address character is recognized (depending on the setting of UM1–UM0).

If the UART is still in the process of receiving a message that the user has already decided to discard, the message may be aborted by issuing the ENTER HUNT MODE command. The UART receiver will be re-enabled when the message is finished by detecting one idle

4.5.11.13 UART Mode Register.

Each SCC mode register is a 16-bit, memory- mapped, read-write register that controls the SCC operation. The term UART mode register refers to the protocol-specific bits (15–6) of the SCC mode register when that SCC is configured as a UART. The read-write UART mode register is cleared by reset.

15	14	13	12	11	10	9	8	7	6	5 … 0
TPM1	TPM0	RPM	PEN	UM1	UM0	FRZ	CL	RTSM	SL	COMMON SCC MODE BITS

TPM1–TPM0—Transmitter Parity Mode

TMP1—TMP0 select the type of parity to be performed.

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

When odd parity is selected, the receiver will count the number of ones in the data word. If the total number of ones is not an odd number, the parity bit is set to one to produce an odd number of ones. If the receiver counts an even number of ones, an error in transmission has occurred. Similarly, for even parity, an even number of ones must result from the calculation performed at both ends of the line.

PEN—Parity Enable

0 = No parity
1 = Parity is enabled for the transmitter and receiver as determined by the parity mode bits.

UM1–UM0—UART Mode 1–0

00 = Normal UART operation. Multidrop mode is disabled for point-to-point operation and an idle-line wakeup is selected. In the idle-line wakeup mode, the UART receiver is re-enabled by an idle string of 9 to 13 consecutive ones (depending on character length and parity mode).
01 = In the multidrop mode, an additional address/data bit is transmitted with each character. The multidrop asynchronous modes are compatible with the Motorola MC68681 DUART, the Motorola MC68HC11 SCI interface, and the Motorola DSP56000 SCI interface. UM0 is also used to select the wakeup mode before enabling the receiver or issuing the ENTER HUNT MODE command.
Multidrop mode is enabled and an address bit wakeup is selected. In the address bit wakeup mode, the UART receiver is re-enabled when the last data bit (the 8th or 9th) in a character is one. This configuration means that the received character is an address, which should be processed by all inactive processors. The IMP re-

I—Interrupt

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

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

C—Control Character

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

A—Address

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

M—Address Match

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

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

ID—Buffer Closed on Reception of Idles

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

Bits 7–6, 2—Reserved for future use.

BR—Break Received

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

FR—Framing Error

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

PR—Parity Error

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

OV—Overrun

A receiver overrun occurred during message reception.

the SCC mode register when that SCC is configured for HDLC. The read-write HDLC mode register is cleared by reset.

15	14	13	12	11	10	9	8	7	6	5 0
NOF3	NOF2	NOF1	NOF0	C32	FSE	—	RTE	FLG	ENC	COMMON SCC MODE BITS

NOF3–NOF0—Minimum Number of Flags between Frames or before Frames (0 to 15 Flags)

If NOF3–NOF0 = 0000, then no flags will be inserted between frames. Thus, the closing flag of one frame will be followed immediately by the opening flag of the next frame in the case of back-to-back frames.

C32—CRC16/CRC32

0 = 16-bit CCITT CRC ($X^{16} + X^{12} + X^5 + 1$)
1 = 32-bit CCITT CRC ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + 1$)

FSE—Flag Sharing Enable

0 = Normal operation
1 = If NOF3–NOF0 = 0000, then a single shared flag is transmitted between back- to-back frames. Other values of NOF3–NOF0 are decremented by one when FSE is set. This is useful in Signaling System #7 applications.

Bit 9—Reserved for future use.

RTE—Retransmit Enable

0 = No automatic retransmission will be performed.
1 = Automatic retransmit enabled

Automatic retransmission occurs if a $\overline{\text{CTS}}$ lost condition happens on the first or second buffer of the frame. See 4.5.12.8 HDLC Error-Handling Procedure.

FLG—Transmit Flags/Idles between Frames and Control the RTS Pin

0 = Send ones between frames; $\overline{\text{RTS}}$ is negated between frames. If NOF–NOF0 is greater than zero, $\overline{\text{RTS}}$ will be negated for a multiple of eight transmit clocks. The HDLC controller can transmit ones in both the NRZ and NRZI data encoding formats. The CP polls the Tx BD ready bit every 16 transmit clocks.
1 = Send flags between frames. $\overline{\text{RTS}}$ is always asserted. The CP polls the Tx BD ready bit every eight transmit clocks.

NOTE

This bit may be dynamically modified. If toggled from a one to a zero between frames, a maximum of two additional flags will be transmitted before the idle condition will begin. Toggling FLG will never result in partial flags being transmitted.

4.5.15.3 Adaption for Asynchronous Rates up to 19.2 kbps

The V.110 asynchronous bit rate adaption block diagram within the terminal adaptor is shown in Figure 4-39.

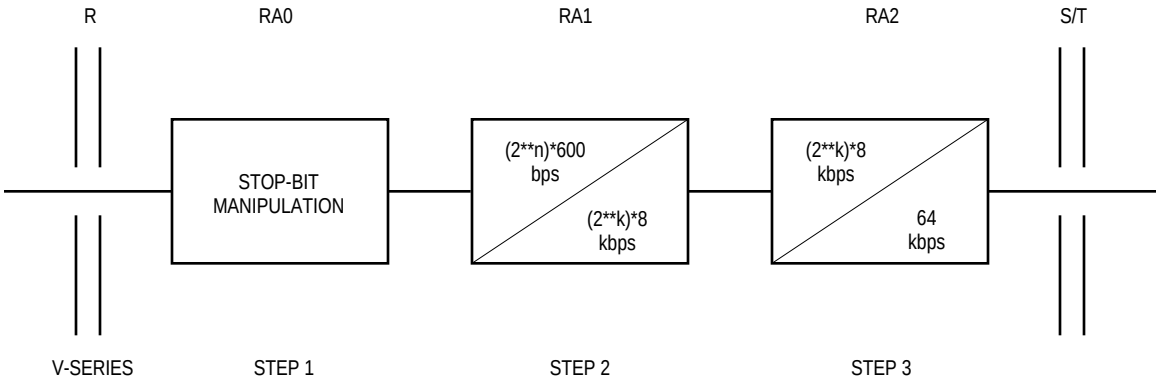


Figure 4-39. Three-Step Asynchronous Bit Rate Adaption

This function may be implemented in two SCCs. One SCC operates as a UART; the other SCC operates as a V.110 controller. The M68000 core formats the data for transmission by the V.110 at the 64 kbps data rate. Thus, the RA1 step is hidden in software.

4.5.15.4 V.110 Controller Overview.

By the appropriate setting of its SCC mode register, any of the SCC channels may be configured to function as a V.110 controller. MODE1–MODE0 bits the SCC mode register should be programmed to DDCMP, and the V.110 bit in the DDCMP mode register should be set. The V.110 controller has the ability to receive and transmit V.110 80-bit frames. The processing of those frames is handled by the M68000 core in software.

The V.110 receiver will synchronize on the 17-bit alignment pattern of the frame:

00000000	1xxxxxxx	1xxxxxxx	1xxxxxxx	1xxxxxxx
1xxxxxxx	1xxxxxxx	1xxxxxxx	1xxxxxxx	1xxxxxxx

After achieving frame synchronization, the receiver will transfer the frame data to a receive buffer (the leading one will be the MSB so that the programmer does not have to swap the bits). The V.110 controller will write nine bytes of data to the buffer (discarding the first byte of all zeros). The M68000 core should unformat the data in memory according to the V.110 protocol to create the data buffer; it may then use another SCC controller to transmit this data to the R interface.

The V.110 transmitter will transmit a data buffer transparently with a bit swap (the MSB will be transmitted first) onto a B channel. The data buffer should contain the 17-bit alignment pattern. Another SCC controller may be used to receive data from the R interface. The M68000 core should then format the data according to the V.110 protocol to create the V.110 80-bit frame data buffer. The V.110 controller will then transmit it onto the B channel.

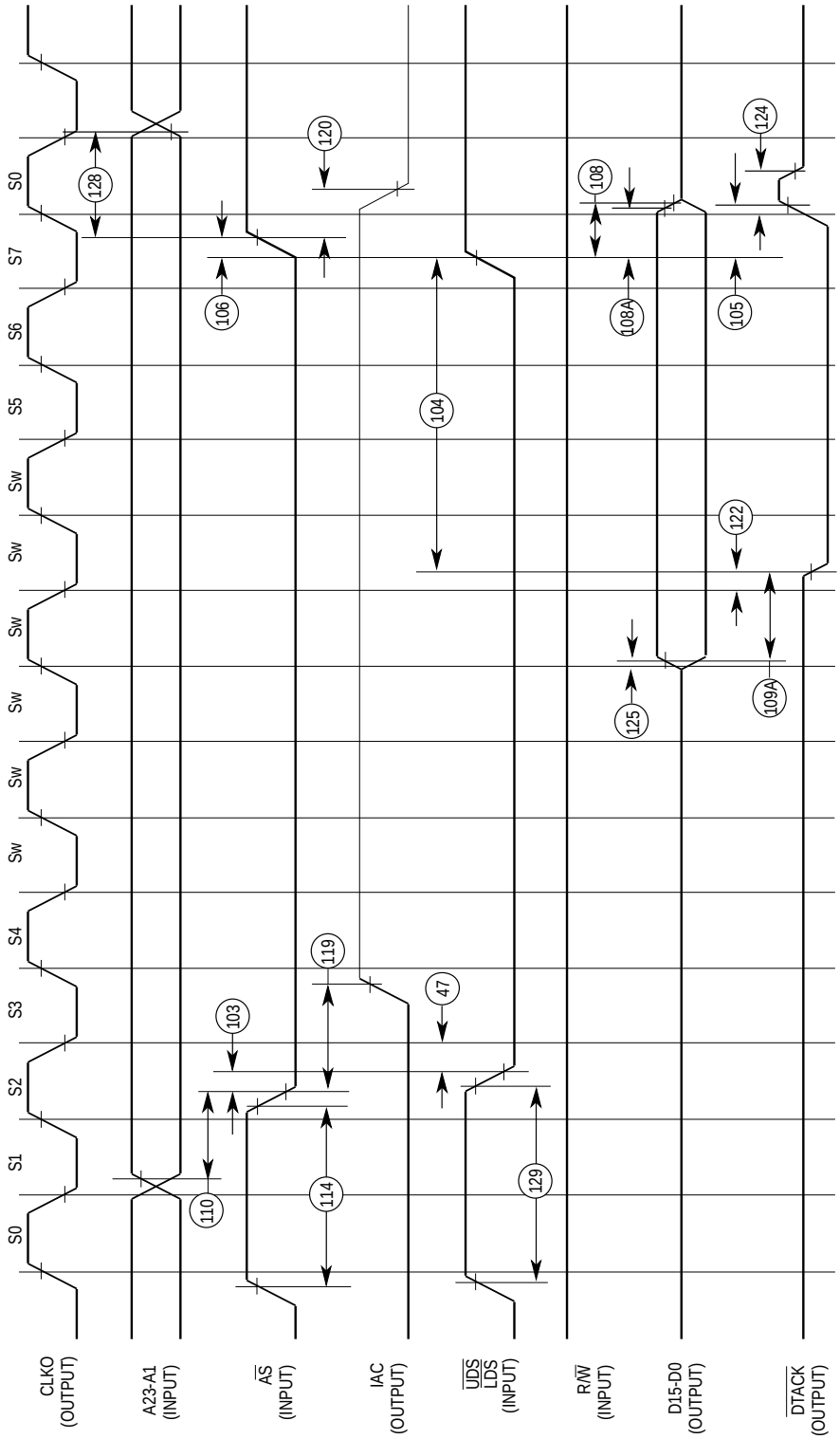


Figure 6-8. External Master Internal Asynchronous Read Cycle Timing Diagram

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

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

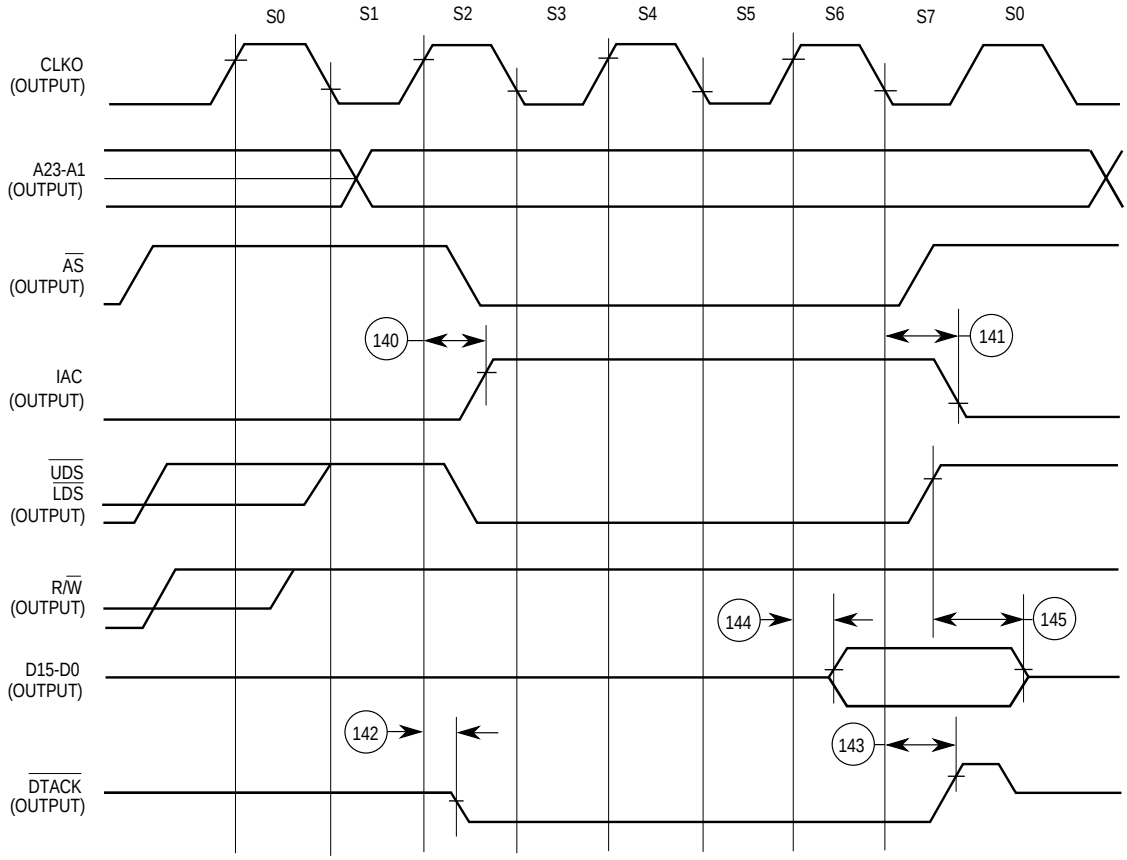
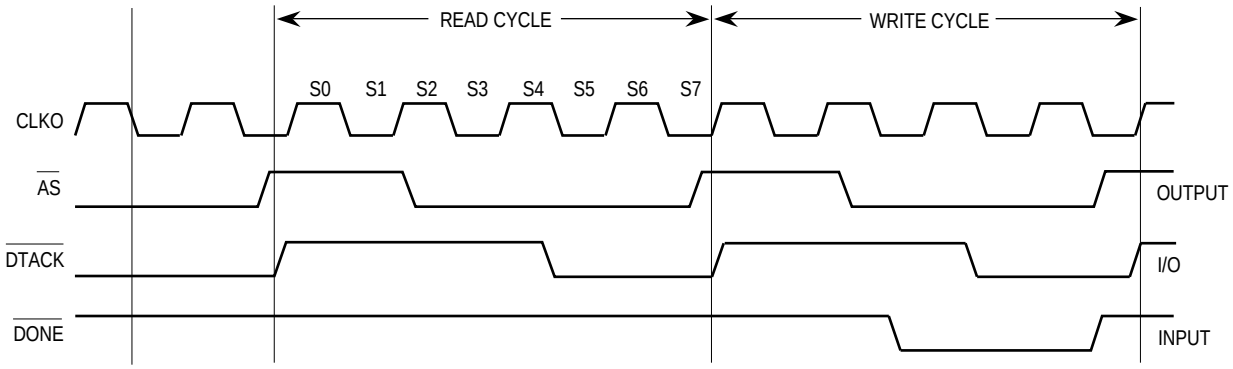


Figure 6-13. Internal Master Internal Read/Write Cycle Timing Diagram



NOTE: If the byte count had reached zero, DONE would be asserted by the IDMA, indicating normal transfer termination.

Figure D-8. Typical IDMA External Cycles Showing Block Transfer Termination

Table D-3. Channel Status Register Bits

Operation	Bit	7	6	5	4	3	2	1	0
Done Not Synchronized—DONE is asserted as an input during the first access of an 8-bit peripheral when operating with 1 6-bit memory.	DNS					1			
Bus Error Source—A bus error occurred during the read portion of an IDMA cycle.	BES	Reserved					1*		
Bus Error Destination—A bus error occurred during the write portion of an IDMA cycle.	BED							1*	
Normal Channel Transfer Done—The BCR decremented to zero or the external peripheral asserted DONE and no errors occurred during any IDMA cycle.	DONE								1*

* These bits are cleared by writing a one or setting RST in the CMR.

Figure D-9 depicts the typical cycles used on a 16-bit bus when the source data size and destination data size are not equal. In this example, the source size is byte and the destination size is word. The IDMA performs two read cycles to obtain data and then performs a write cycle to place data into the destination location. If the CMR SAPI bit was set, then each byte read increments the SAPR by two. Hence, the SAPR is always pointing to the leftmost or rightmost byte of the 16-bit bus. This type of transfer duplicates the function of an M68000 MOVEP instruction.

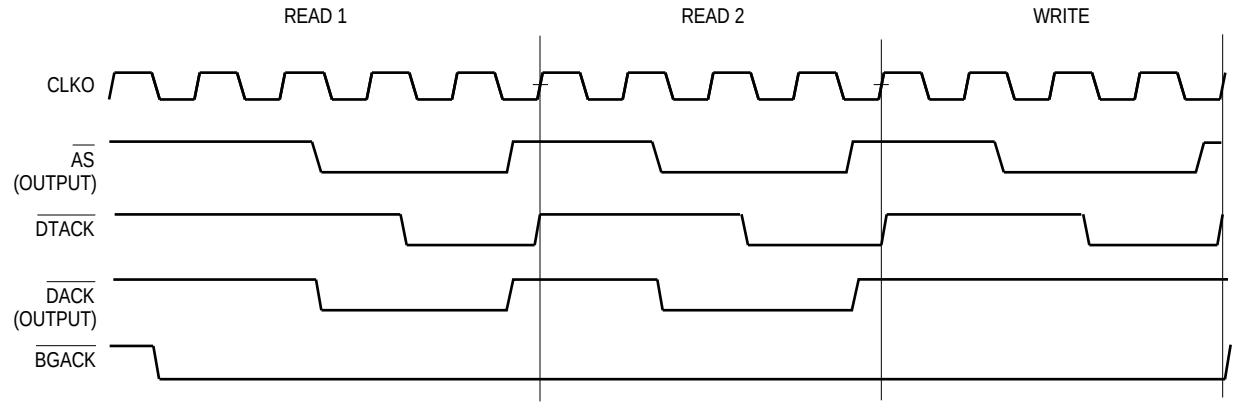


Figure D-9. Typical IDMA Source to Word Destination IDMA Cycles

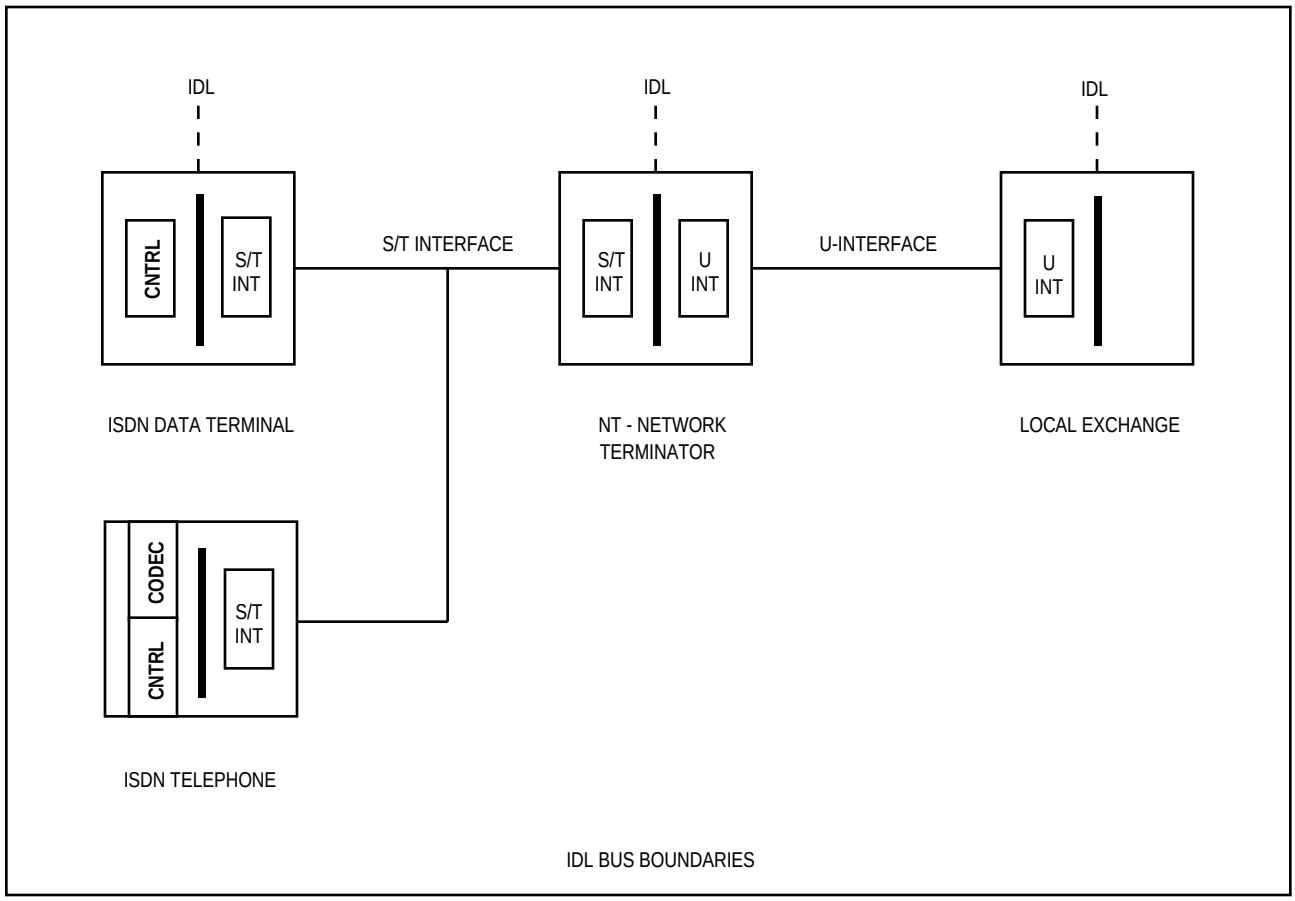


Figure D-12. IDL Bus Boundaries

Motorola offers a full line of ISDN/IDL compatible chips that enable modular and portable design of ISDN equipment:

- MC145472 — ISDN U interface transceiver conforms to the American Standard for ISDN basic access.
- MC145474 — ISDN S/T interface transceiver conforms to CCITT 1.430 and ANSI T1.605 recommendations.
- MC145475 — ISDN S/T interface transceiver conforms to CCITT 1.430 and ANSI T1.605 recommendations. Supports NT1 Star Operation.
- MC145554/7— Mu-Law and A-Law Companding PCM CODEC Filter (16-pin package).
- MC145564/7— Mu-Law and A-Law Companding PCM CODEC Filter (20-pin package).

D.6.5 IDL Bus Specification

The IDL bus (see Figure D-12) consists of four wires and provides data exchange with 125-µsec frame period with clock speeds of 1.544 to 2.56 MHz. The IDL supports a total of 160 kbps of full-duplex data consisting of two B-channels (2 x 64 kbps), one D-channel (16 kbps), one M-channel (8 kbps), and one A-channel (8 kbps). The A and M channels are used to convey maintenance and auxiliary data.

Since the L bit is set, once the frame ends, the synchronization process must occur once again. Also, to force resynchronizations instead of waiting for the transmission to finish, a STOP TRANSMIT command can be given, followed by a RESTART TRANSMIT command; however, the STOP TRANSMIT command will abort the current buffer.

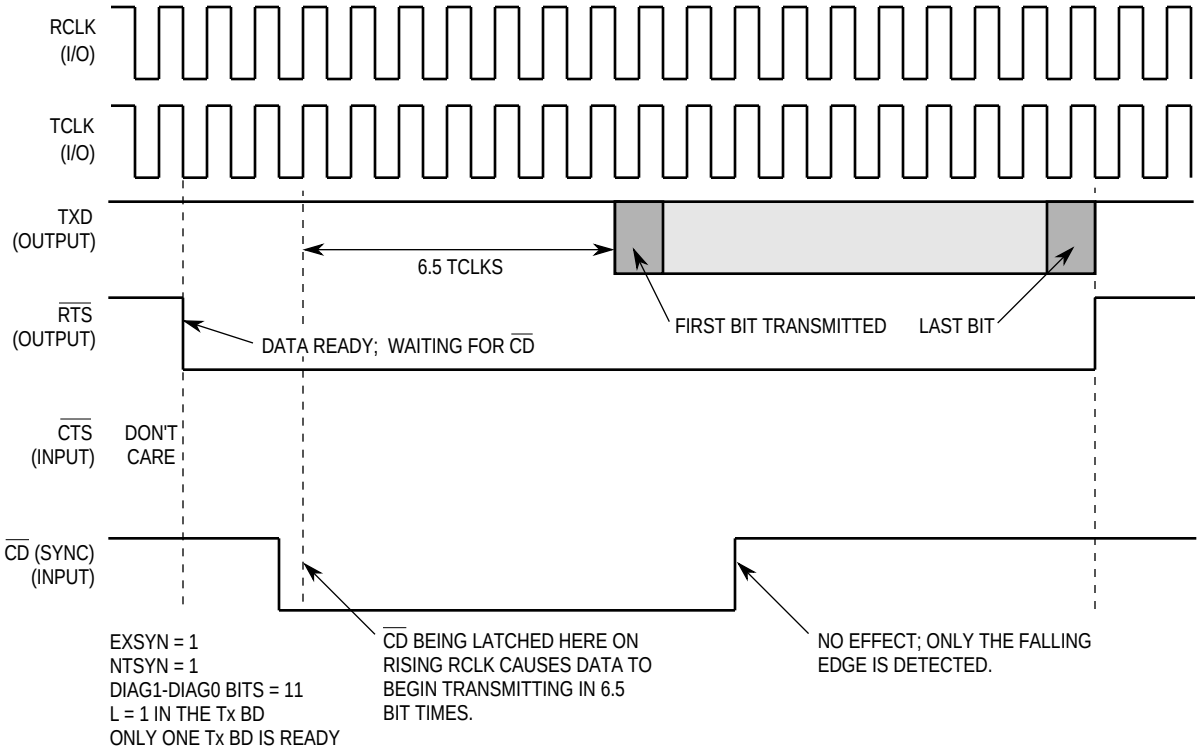


Figure D-26. Using $\overline{CD}$ (Sync) In the NMSI Transmit Case

Figure D-27 shows how $\overline{CD}$ (sync) is used in the NMSI receive case. Setting the EXSYN bit causes $\overline{CD}$ (sync) to control the reception of data. $\overline{CD}$ (sync) should be latched low on the rising clock (RCLK) of the second *bit of the frame.* (Latching $\overline{CD}$ (sync) during the 2nd bit of the frame allows external BISYNC sync detection logic, which also uses EXSYN, extra time to present the external sync to the SCC.) Once synchronization is achieved, it will never be lost unless an ENTER HUNT MODE command is given, a receive overrun occurs, or the receiver is disabled and re-enabled (ENR bit is cleared, ENTER HUNT MODE command is issued, and ENR is set). Once synchronization is lost, a new frame can be resynchronized using $\overline{CD}$ (sync).

Notice that we have been discussing the receive and transmit cases separately. The receive and transmit halves of the SCC really are separate and distinct; however, in transparent mode, the receive and transmit halves of the SCC share the $\overline{CD}$ (sync) pin, which is not true in normal NMSI.

The PCM highway interface has three $\overline{RTS}$ signals. One of these signals is asserted when an SCC wants to transmit over the PCM Highway just like in NMSI mode), and stays continuously asserted until the entire frame is transmitted (regardless of how many time slots the transmission takes). Which $\overline{RTS}$ signal asserts depends on which SCC is transmitting; there is one $\overline{RTS}$ signal for each SCC. Notice, however, that there is no $\overline{CTS}$ signal, so there is nothing to hold off the transfer. If the $\overline{RTS}$ signals are not needed, they can be ignored or reassigned as parallel I/O lines.

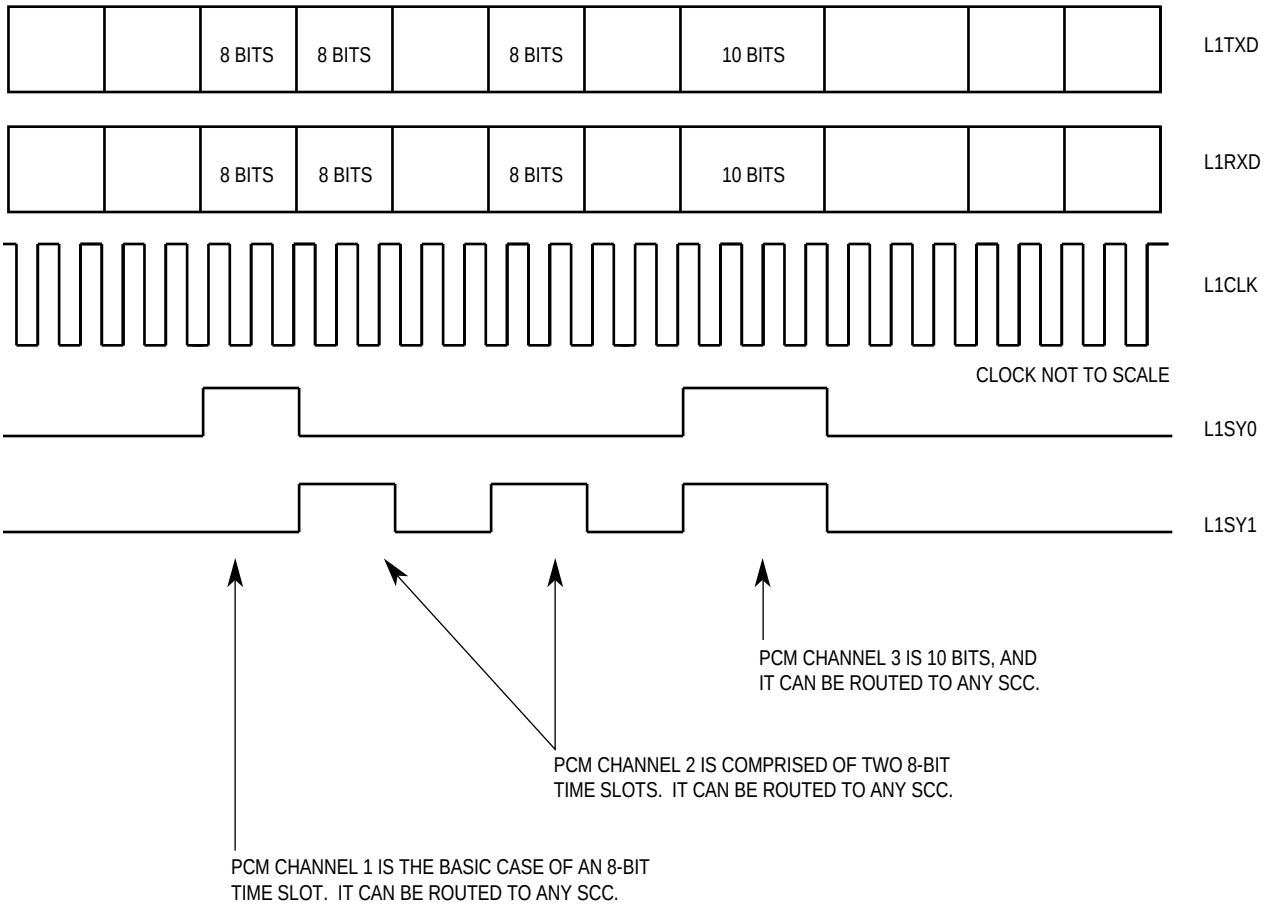


Figure D-29. Routing Channels in PCM Envelope Mode

What other signals are missing from PCM mode? First, there is no $\overline{CD}$ signal for the receiver. The receiver is enabled whenever the ENR bit is set. However, you could say there is a $\overline{CD}$ (sync) of sorts that is implemented with the L1SY1 and L1SY0 pins. Two pins are used since not only is the timing important, but also the selection of the PCM channel as well.

The way transparent mode works with a PCM highway interface is very similar to the operation of the gated clocks example discussed previously. Whether or not a time slot environment is present, PCM mode gives greater control over what intervals transparent data can be transmitted and received. However, in PCM mode, the clocks are gated by the physical interface on the MC68302 as opposed to external hardware.

RXB—Rx Buffer

0 = No interrupt
1 = A buffer that was not a complete frame was received on the HDLC channel (set only if the I bit in the Rx buffer descriptor is set).

E.1.1.2.5 HDLC Mask Register (SCCM). This 8-bit register is located at offset $88A (SCC1), $89A (SCC2), and $8AA (SCC3) on D15-D8 of a 16-bit data bus. The SCCM is used to enable and disable interrupt events reported by the SCCE. The mask bits correspond to the interrupt event bit shown in the SCCE. A bit should be set to one to enable the corresponding interrupt in the SCCE.

7	6	5	4	3	2	1	0
CTS	CD	IDL	TXE	RXF	BSY	TXB	RXB

E.1.1.2.6 HDLC Status Register (SCCS). This 8-bit register is located at offset $88C (SCC1), $89C (SCC2), and $8AC (SCC3) on D15-D8 of a 16-bit data bus. The SCCS register reflects the current status of the RXD, $\overline{CD}$, and $\overline{CTS}$ lines as seen by the SCC.

7	6	5	4	3	2	1	0
—	—	—	—	—	ID	CD	CTS

ID—Idle Status on the Receiver Line (valid only when the ENR bit is set and the receive clock is running)

0 = Receiver line is not idling.
1 = Either $\overline{CD}$ is not asserted or the receiver line is idling while $\overline{CD}$ is asserted.

$\overline{CD}$—Carrier Detect Status Changed (valid only when the ENR bit is set and the receive clock is running)

0 = $\overline{CD}$ is asserted.
1 = $\overline{CD}$ is not asserted.

$\overline{CTS}$—Clear-To-Send Status Changed (valid only when the ENT bit is set and the transmit clock is running)

0 = $\overline{CTS}$ is asserted.
1 = $\overline{CTS}$ is not asserted.

E.1.1.3 GENERAL AND HDLC PROTOCOL-SPECIFIC PARAMETER RAM. Each SCC has 32 words of parameter RAM used to configure receive and transmit operation, store temporary parameters for the CP, and maintain counters. The first 14 words are general parameters, which are the same for each protocol. The last 18 words are specific to the protocol selected. The following sections discuss the parameters that the user must initialize to configure the HDLC operation.

E.1.1.3.1 RFCR/TFCR—Rx Function Code/Tx Function Code. This 16-bit parameter contains the function codes of the receive data buffers and transmit data buffers. The user must initialize the function codes (FC2-FC0) to a value less than 7.

R—Ready

0 = This data buffer is not currently ready for transmission.
1 = This data buffer has been prepared by the user for transmission but has not yet been fully transmitted. Must be set by the user to enable transmission of the 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 transmit BD table.
1 = This is the last BD in the transmit BD table.

I—Interrupt

0 = No interrupt is generated when this buffer is closed.
1 = The TX bit in the event register is set if this buffer closed without an error. If an error occurred, then TXE is set.

L—Last in Frame

0 = This buffer is not the last buffer in the transmitted block.
1 = This buffer is the last buffer in the transmitted block.

Bits 10-2—Reserved for future use; should be written with zero by the user.

UN—Underrun

0 = No transmitter underrun occurred.
1 = A transmitter underrun condition occurred while transmitting the associated data buffer.

CT—CTS Lost

0 = No CTS or L1GR lost was detected during frame transmission.
1 = CTS in NMSI mode or L1GR in IDL/GCI mode was lost during frame transmission.

E.3.1.5.2 Transmit Buffer Data Length. This 16-bit value is written by the user to indicate the number of data bytes to be transmitted from the data buffer.

E.3.1.5.3 Transmit Buffer Pointer. This 32-bit value is written by the user to indicate the address of the first byte of data in the data buffer.

E.3.2 Programming the SCC for Transparent

This section gives a generic algorithm for programming an SCC to handle the transparent protocol. The algorithm is intended to show what must be done and in what order to initialize the SCC and prepare the SCC for transmission and reception. The algorithm is not specific and assumes that the IMP and other on-chip peripherals have been initialized as required by the system hardware (timers, chip selects, etc.).

E.3.2.1 CP INITIALIZATION.