Welcome to E-XFL.COM

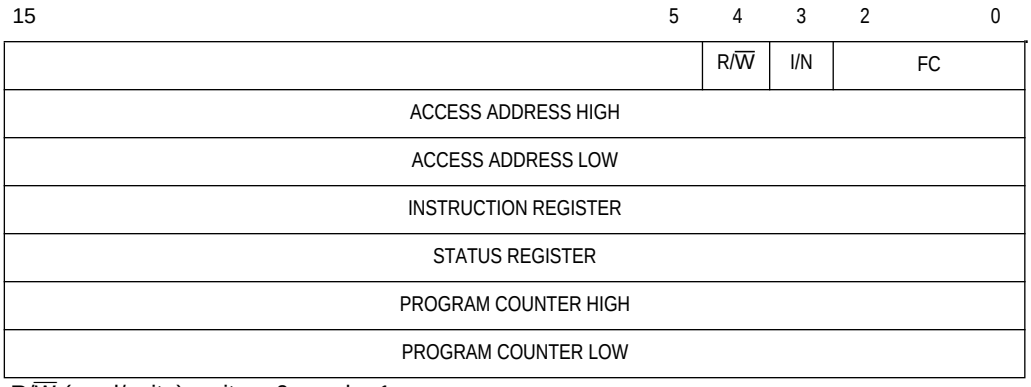
Understanding Embedded - Microprocessors

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	5.0V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	132-BQFP Bumpered
Supplier Device Package	132-PQFP (46x46)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68302cfc16c



R/$\overline{W}$ (read/write): write = 0, read = 1
I/N (instruction not): instruction = 0, not =1
FC: Function Code

Figure 2-3. M68000 Bus/Address Error Exception Stack Frame

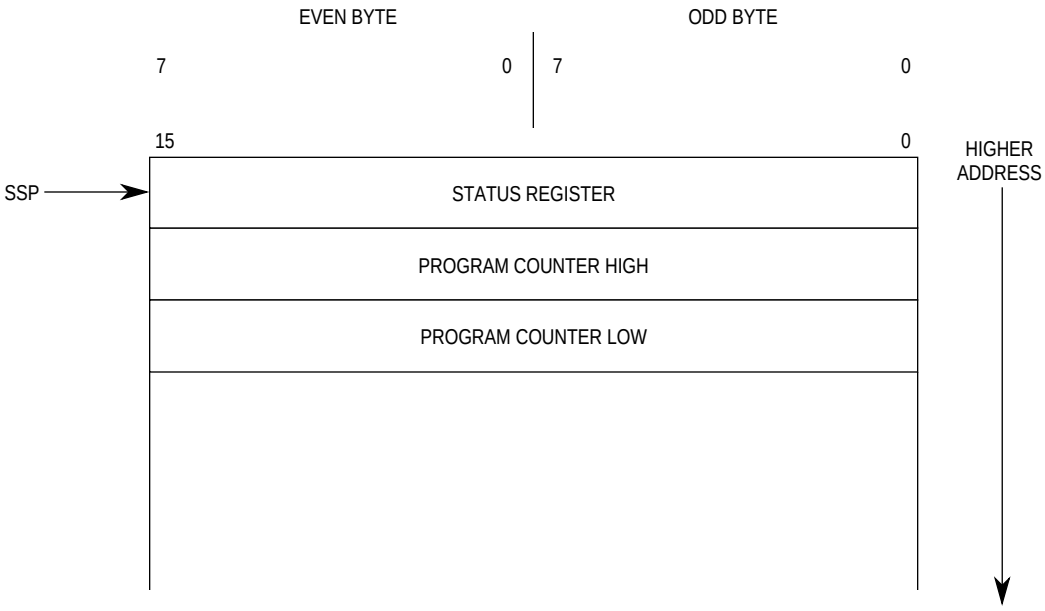


Figure 2-4. M68000 Short-Form Exception Stack Frame

NOTE

The MC68302 uses the exact same exception stack frames as the MC68000.

For exception processing times and instruction execution times, refer to MC68000UM/AD, *8-/16-/32-Bit Microprocessor User's Manual*.

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).

have to be contiguous in the PCM highway, but rather can be separated by other time slots. Also, PCM channel time slots need not be an even multiple of eight bits in envelope mode. Although not shown in the figure, it is also possible to route multiple PCM channels to a single SCC, causing the SCC to process one higher speed data stream.

The PCM highway interface also supports the $\overline{\text{RTS}}$ signals. They will be asserted (just like in NMSI mode) when an SCC desires to transmit over the PCM highway and will stay asserted until the entire frame is transmitted (regardless of how many time slots that takes). The $\overline{\text{RTS}}$ signal that asserts corresponds to the SCC that desires to transmit. The $\overline{\text{RTS}}$ signals may be useful in debugging but are not required for proper PCM highway operation. If the $\overline{\text{RTS}}$ signals are not needed, they can be ignored or reassigned as parallel I/O.

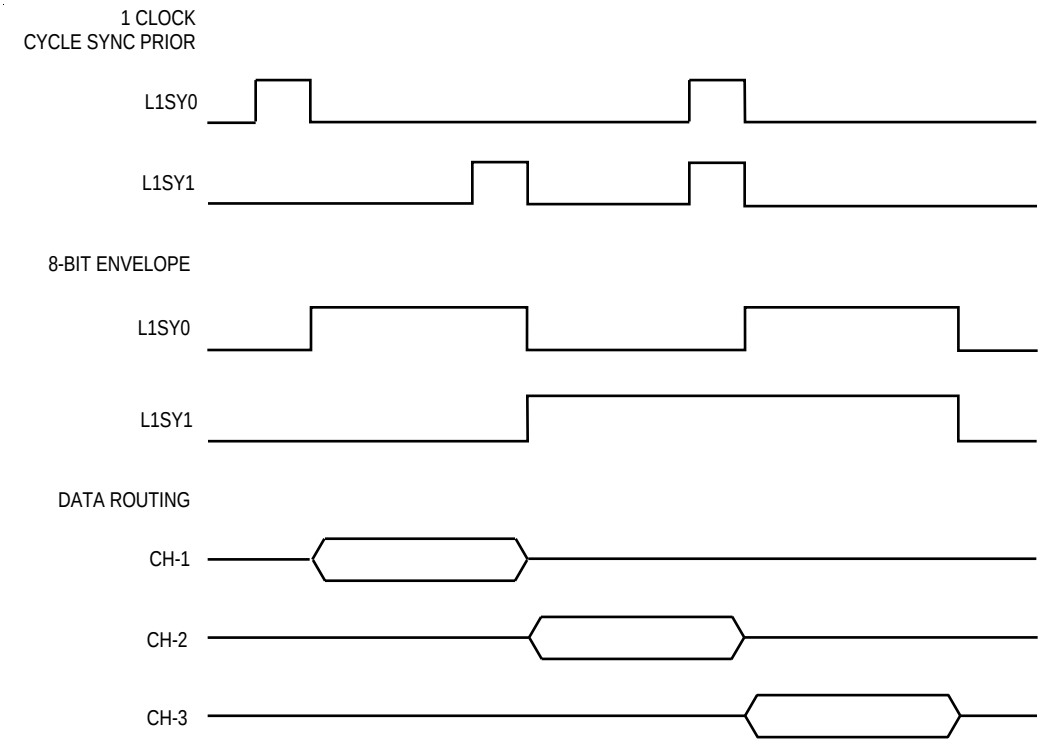


Figure 4-9. Two PCM Sync Methods

Three characters should first be entered into the UART control character table:

1. End of Line—The empty (E) bit is cleared; the reject (R) bit is cleared. When an end-of-line character is received, the current buffer is closed (the next BD taken by the IMP) and made available to the M68000 core for processing. This buffer contains an entire S record, which the processor can now check and copy to memory or disk as required.
2. XOFF—E should be cleared and R should be set. Whenever the M68000 core receives a control character received interrupt and the receive control character register contains XOFF, the software should immediately stop transmitting to the other station by setting the FRZ bit in the UART mode register. This prevents data from being lost by the other station when it runs out of receive buffers.
3. XON—XON should be received after XOFF. E should be cleared and R should be set. The FRZ bit on the transmitter should now be cleared. The IMP automatically resumes transmission of the serial line at the point at which it was previously stopped. Like XOFF, the XON character is not stored in the receive buffer.

To receive the S records, the M68000 core must only wait for the RX interrupt, indicating the reception of a complete S-record buffer. Transmission requires assembling S records into data buffers and linking them to the transmit buffer table (transmission may be temporarily halted by reception of an XOFF character). This scheme minimizes the number of interrupts received by the M68000 core (one per S record) and relieves it from the task of continually scanning for control characters.

4.5.12 HDLC Controller

Layer 2 of the seven-layer OSI model is the data link layer. One of the most common layer 2 protocols is HDLC. Many other common layer 2 protocols are heavily based on HDLC, particularly its framing structure: namely, SDLC, SS#7, LAPB, and LAPD. The framing structure of HDLC is shown in Figure 4-24.

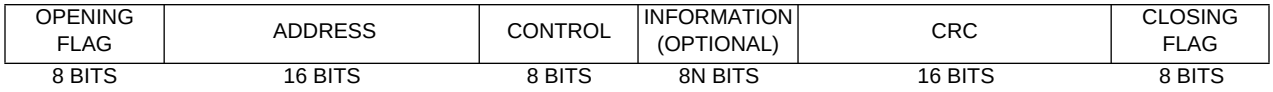


Figure 4-24. Typical HDLC Frame

HDLC uses a zero insertion/deletion process (commonly known as bit-stuffing) to ensure that the bit pattern of the delimiter flag does not occur in the fields between flags. The HDLC frame is synchronous and therefore relies on the physical layer to provide a method of clocking and synchronizing the transmitter and receiver.

Since the layer 2 frame can be transmitted over a point-to-point link, a broadcast network, or packet and circuit-switched systems, an address field is needed to carry the frame's destination address. The length of this field is commonly 0, 8, or 16 bits, depending on the data link layer protocol. For instance, SDLC and LAPB use an 8-bit address. SS#7 has no address field at all because it is always used in point-to-point signaling links. LAPD further divides its 16-bit address into different fields to specify various access points within one piece of equipment. It also defines a broadcast address. Some HDLC-type protocols also allow for extended addressing beyond 16-bits.

The purpose of the control characters table is to enable automatic recognition (by the BISYNC controller) of the end of the current block. See 4.5.13.14 Programming the BISYNC Controllers for more information. Since the BISYNC controller imposes no restrictions on the format of the BISYNC blocks, user software must respond to the received characters and inform the BISYNC controller of mode changes and certain protocol events (e.g., resetting the BCS). However, correct use of the control characters table allows the remainder of the block to be received without interrupting the user software.

Up to eight control characters may be defined. These characters inform the BISYNC controller that the end of the current block has been reached and whether a BCS is expected following this character. For example, the end of text (ETX) character implies both an end of block (ETB) and a BCS should be received. An enquiry (ENQ) character designates end of block without a subsequent BCS. All the control characters are written into the data buffer.

The BISYNC controller uses a table of 16-bit entries to support control character recognition. Each entry consists of the control character, an end-of-table bit, a BCS expected bit, and a hunt mode bit. The control characters table is shown in Figure 4-31. To disable the entire control characters table, write $8000 to the first table entry.

	15	14	13	12	11	10	9	8	7 … 0
OFFSET + 0	E	B	H						CHARACTER1
OFFSET +2	E	B	H						CHARACTER2
OFFSET + 4	E	B	H						CHARACTER3
•									
•									
•									
OFFSET + E	E	B	H						CHARACTER8

Figure 4-31. BISYNC Control Characters Table

CHARACTER8–CHARACTER1—Control Character Value

These fields define control characters.

NOTE

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

E—End of Table

0 = This entry is valid. The lower eight bits will be checked against the incoming character.

1 = The entry is not valid. No valid entries exist beyond this entry.

The channel will enter hunt mode immediately. It is possible that a SYN1–SYN2-(SOH,DLE,ENQ) sequence in data will be incorrectly interpreted as the start of the next header, but this “header” will have a CRC error.

NOTE

This error can occur only on asynchronous links.

6. Parity Error. When a parity error occurs, the channel writes the received character to the buffer, closes the buffer, sets the parity error (PR) bit in the BD, and generates the RBK interrupt (if enabled).
   The channel will enter hunt mode immediately. It is possible that a SYN1–SYN2-(SOH,DLE,ENQ) sequence in data will be incorrectly interpreted as the start of the next header, but this “header” will have a CRC error.

NOTE

This error can occur only on asynchronous links.

Error Counters

The CP maintains four 16-bit (modulo 2**16) error counters for each DDCMP controller. They can be initialized by the user when the channel is disabled. The counters are as follows:

—CRC1EC—CRC1 Error Counter
—CRC2EC—CRC2/CRC3 Error Counter
—NMARC — Nonmatching Address Received Counter (updated only when the frame is error-free)
—DISMC — Discarded Messages (received messages when there are no free buffers and the frame is error-free)

4.5.14.9 DDCMP Mode Register

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

15	14	13	12	11	10	9	8	7	6	5 0
NOS3	NOS2	NOS1	NOS0	—	V.110	—	—	SYNF	ENC	COMMON SCC MODE BITS

NOS3–NOS0—Minimum Number of SYN1—SYN2 Pairs between or before Messages (1 to 16 SYNC Pairs)

If NOS3–NOS0 = 0000, then 1 SYNC pair will be transmitted; if NOS3–NOS0 = 1111, then 16 SYNC pairs will be transmitted.

NOTE

With appropriate programming of the transmit BD (TC = 1 and L = 0), it is possible to transmit back-to-back messages.

$\overline{FRZ}$—Freeze Activity

The $\overline{FRZ}$ pin is used to freeze the activity of selected peripherals. This is useful for system debugging purposes. Refer to 3.8 System Control for more details on which peripherals are affected. $\overline{FRZ}$ should be continuously negated during total system reset.

5.5 ADDRESS BUS PINS (A23–A1)

The address bus pins are shown in Figure 5-4.

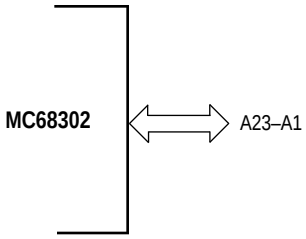


Figure 5-4. Address Bus Pins

A23—A1 form a 24-bit address bus when combined with $\overline{UDS}$/A0. The address bus is a bi-directional, three-state bus capable of addressing 16M bytes of data (including the IMP internal address space). It provides the address for bus operation during all cycles except CPU space cycles. In CPU space cycles, the CPU reads a peripheral device vector number.

These lines are outputs when the IMP (M68000 core, SDMA or IDMA) is the bus master and are inputs otherwise.

5.6 DATA BUS PINS (D15—D0)

The data bus pins are shown in Figure 5-5.

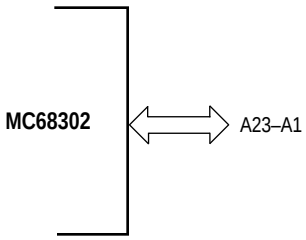


Figure 5-5. Data Bus Pins

This 16-bit, bidirectional, three-state bus is the general-purpose data path. It can transmit and accept data in either word or byte lengths. For all 16-bit IMP accesses, byte 0, the high-order byte of a word, is available on D15–D8, conforming to the standard M68000 format.

6.13 AC ELECTRICAL SPECIFICATIONS—CHIP-SELECT TIMING INTERNAL MASTER (see Figure 6-14)

			16.67 MHz		20 MHz		25 MHz		
Num.	Characteristic	Symbol	Min	Max	Min	Max	Min	Max	Unit
150	Clock High to $\overline{CS}$, $\overline{IACK}$ Low (see Note 2)	$t_{CHCSIAKL}$	0	40	0	35	0	27	ns
151	Clock Low to $\overline{CS}$, $\overline{IACK}$ High (see Note 2)	$t_{CLCSIAKH}$	0	40	0	35	0	27	ns
152	$\overline{CS}$ Width Negated	t_{CSH}	60	—	50	—	40	—	ns
153	Clock High to $\overline{DTACK}$ Low (0 Wait State)	t_{CHDTKL}	—	45	—	40	—	30	ns
154	Clock Low to $\overline{DTACK}$ Low (1–6 Wait States)	t_{CLDTKL}	—	30	—	25	—	20	ns
155	Clock Low to $\overline{DTACK}$ High	t_{CLDTKH}	—	40	—	35	—	27	ns
156	Clock High to $\overline{BERR}$ Low (see Note 1)	t_{CHBERL}	—	40	—	35	—	27	ns
157	Clock Low to $\overline{BERR}$ High Impedance (see Note 1)	t_{CLBERH}	—	40	—	35	—	27	ns
158	$\overline{DTACK}$ High to $\overline{DTACK}$ High Impedance	$t_{DTKHDTKZ}$	—	15	—	15	—	10	ns
171	Input Data Hold Time from S6 Low	t_{IDHCL}	5	—	5	—	5	—	ns
172	$\overline{CS}$ Negated to Data-Out Invalid (Write)	t_{CSNDOI}	10	—	10	—	7	—	ns
173	Address, FC Valid to $\overline{CS}$ Asserted	t_{AFVCSA}	15	—	15	—	15	—	ns
174	$\overline{CS}$ Negated to Address, FC Invalid	t_{CSNAFI}	15	—	15	—	12	—	ns
175	$\overline{CS}$ Low Time (0 Wait States)	t_{CSLT}	120	—	100	—	80	—	ns
176	$\overline{CS}$ Negated to R/$\overline{W}$ Invalid	t_{CSNRWI}	10	—	10	—	7	—	ns
177	$\overline{CS}$ Asserted to R/$\overline{W}$ Low (Write)	t_{CSARWL}	—	10	—	10	—	8	ns
178	$\overline{CS}$ Negated to Data-In Invalid (Hold Time on Read)	t_{CSNDII}	0	—	0	—	0	—	ns

NOTE:
1. This specification is valid only when the ADCE or WPVE bits in the SCR are set.
2. For loading capacitance less than or equal to 50 pF, subtract 4 ns from the maximum value given.
3. Since $\overline{AS}$ and $\overline{CS}$ are asserted/negated on the same CLKO edges, no $\overline{AS}$ to $\overline{CS}$ relative timings can be specified. However, $\overline{CS}$ timings are given relative to a number of other signals, in the same manner as $\overline{AS}$. See Figure 6-2 and Figure 6-3 for diagrams.

6.17 AC ELECTRICAL SPECIFICATIONS—TIMERS

(see Figure 6-18)

			16.67 MHz		20 MHz		25 MHz		
Num.	Characteristic	Symbol	Min	Max	Min	Max	Min	Max	Unit
200	Timer Input Capture Pulse Width	t_{TPW}	50	—	42	—	34	—	ns
201	TIN Clock Low Pulse Width	t_{TICLT}	50	—	42	—	34	—	ns
202	TIN Clock High Pulse Width and Input Capture High Pulse Width	t_{TICHT}	2	—	2	—	2	—	clk
203	TIN Clock Cycle Time	t_{cyc}	3	—	3	—	3	—	clk
204	Clock High to TOUT Valid	t_{CHTOV}	—	35	—	30	—	24	ns
205	$\overline{FRZ}$ Input Setup Time (to Clock High) (see Note 1)	t_{FRZSU}	20	—	20	—	14	—	ns
206	$\overline{FRZ}$ Input Hold Time (from Clock High)	t_{FRZHT}	10	—	10	—	7	—	ns

NOTES:
1. $\overline{FRZ}$ should be negated during total system reset.
2. The TIN specs above do not apply to the use of TIN1 as a baud rate generator input clock. In such a case, specifications 1–3 may be used.

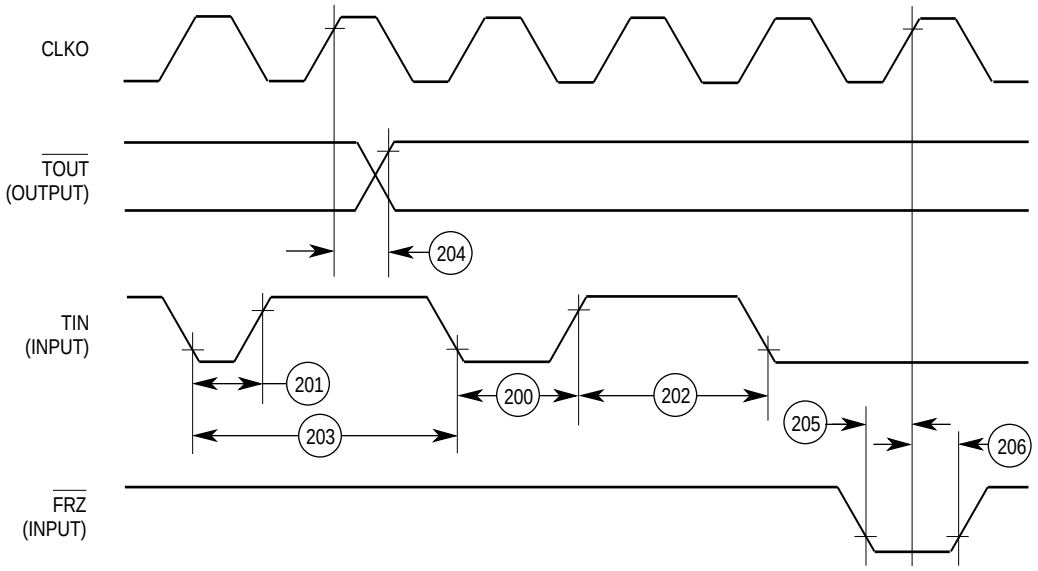


Figure 6-18. Timers Timing Diagram

designed to comply with the requirements of the Hayes AT command set; however, it can be used with simpler character schemes as well (such as a carriage return).

The SCC receiver synchronizes on the falling edge of the START bit. Once a start bit is detected, each bit received is processed by the AutoBaud controller. The AutoBaud controller measures the length of the START bit to determine the receive baudrate and compares the length to values in a user supplied lookup table. After the baudrate is determined, the AutoBaud controller assembles the character and compares it against two user-defined characters. If a match is detected, the AutoBaud controller interrupts the host and returns the determined nominal start value from the lookup table. The AutoBaud controller continues to assemble the characters and interrupt the host until the host stops the reception process. The incoming message should contain a mixture of even and odd characters so that the user has enough information to decide on the proper character format (length and parity). The host then uses the returned nominal start value from the lookup table, modifies the SCC Configuration Register (SCON) to generate the correct baudrate, and reprograms the SCC to UART mode.

Many rates are supported including: 150, 300, 600, 1200, 2400, 4800, 9600, 14.4K, 19.2K, 38.4K, 57.6K, 64K, 96K, and 115.2K. To estimate the performance of the AutoBaud microcode package, the performance table in Appendix A of the MC68302 user's manual can be used. The maximum full-duplex rate for a BISYNC channel is one-tenth of the system clock rate. So a 16.67 MHz 68302 can support 115.2k autobaudrate with another low-speed channel (<50 kbps) and a 20 MHz MC68302 can support 115.2k AutoBaudrate with 2 low-speed channels. The performance can vary depending on system loading, configuration, and echoing mode.

C.6 MICROCODE FROM RAM INITIALIZATION SEQUENCE

1. Perform a total system reset of the MC68302.
2. Write $0700 to the BAR. The base address of the internal dual-port RAM after this action is $700000 (hex). If a different base address is desired, the S-record file addresses should be modified to the desired address.
3. Load the S-record file data into the internal dual-port RAM. (In a production environment, the microcode may be copied from EPROM directly to the internal dual-port RAM.)
4. Write $0001 to address $0F8 in supervisory space.
5. Write a software reset command to the CR.
6. Continue with the normal initialization sequence.

```
IMR         EQU       BASE +  $816      ;Interrupt Mask Register
ISR         EQU       BASE +  $818      ;In-Service Register
SIMODE      EQU       BASE +  $8B4      ;Serial Interface Mode Register
SCON1       EQU       BASE +  $882      ;SCC1 Configuration Register
SCM1        EQU       BASE +  $884      ;SCC1 Mode Register
SCCE1       EQU       BASE +  $888      ;SCC1 Event Register
SCCM 1      EQU       BASE +  $88A      ;SCC1 Mask Register
EN_SCC      EQU       $0C               ;ENR and ENT bits in SCM

***SCC1 Parameter Table ***
ST_BD       EQU       0                 ;Status and Control in BD
SS_BD       EQU       1                 ;Status in BD
LN_BD       EQU       2                 ;Data Length in BD
PT_BD       EQU       4                 ;Buffer pointer in BD
SZ_BD       EQU       $08               ;Size of BD = 8 bytes
FCR_1       EQU       BASE+$0480        ;RFCR and TFCR for SCC1
MRBLR_1     EQU       BASE+$0482        ;Max Rx Buffer Length
RXBD_01     EQU       BASE+$0400        ;RX BD 0 in SCC1
TXBD_01     EQU       BASE+$0440        ;TX BD 0 in SCC1
READY       EQU       $07               ;Ready bit in the 1st byte of TX BD
EMPTY       EQU       $07               ;Empty bit in the 1st byte of RX BD
WRAP        EQU       $05               ;Wrap bit in the 1st byte of BD
* The following values are application dependent for this example
RXBF_01     EQU       $030000           ;Address of the first RX buffer
TXBF_01     EQU       $030080           ;Address of the first TX buffer
BD_CNT      EQU       $08               ;Number of BDs used
SZ_BF       EQU       $10               ;Size of buffer =16 bytes
N_DATA      EQU       6                 ;Number of data to be sent in a buffer
* SCC1 HDLC Parameters
CMSKL_1     EQU       BASE+$04A0        ;CRC Mask Low
CMSKH_1     EQU       BASE+$04A2        ;CRC Mask High
DISFC_1     EQU       BASE+$04A8        ;Discard Frame Counter
CRCEC_1     EQU       BASE+$04AA        ;CRC Error Counter
ABTSC_1     EQU       BASE+$04AC        ;Abort Sequence Counter
NMARC _ 1   EQU       BASE+$04AE        ;Nonmatching Address Receive Counter
RETRC_1     EQU       BASE+$04B0        ;Frame Retransmission Counter
MFLR_1      EQU       BASE+$04B2        ;Max Frame Length Register
HMASK 1     EQU       BASE+$04B6        ;User-Defined Frame Address Mask

********************** Typical M68302 Initialization Code********************
* Register Initialized values in ADS board before execution
* USP = 00080000 ISP = 000040000 (Stack pointer not used)
            ORG       INIT              ;PC = 00030300
            MOVE.W    #$2700,SR         ;SR = 2700, mask off interrupts
* Set Base Address = $700000
* Now all 68302 on-chip peripherals begin at address $700xxx
            MOVE.W    #$0700,BAR        ;BAR = 0700
* Set System Control Register
            MOVE.L    #0,SCR            ;Nothing special for this example
***Setups for interrupt ***
            MOVE.W    #$0A0,GIMR        ;Normal mode, v7–v5 = 5
            MOVE.W    #0,IMR            ;Mask off all for now
            MOVE.W    #$FFFF,IPR        ;Clear IPR

*** Set up Serial Interface Connection ***
```

```
* 00700608 1000 0000 0003 0003 (wrap bit set)


BAR             EQU       $0F2
GIMR            EQU       $700812
IPR             EQU       $700814
IMR             EQU       $700816
ISR             EQU       $700818
PACNT           EQU       $70081e
SIMODE          EQU       $7008b4
SCON3           EQU       $7008a2
SCM3            EQU       $700&4
SCCE3           EQU       $7008a8
SCCM3           EQU       $7008aa

*SCC3 Initialization Code
          ORG       $30300
          MOVE.W #$700.BAR            ;BAR = 0700
* Base Address - S700000, so ALL MC68302 on-chip peripherals begin at
* address S700xxx.

          MOVE.W   #$00A0, GIMR       ;GIMR = 00a0
          MOVE.W   #$FFFF,IPR         ;clear IPR
          MOVE.L   #$30500,$2a0       ;SCC3 vector initialization
          MOVE.W   #$0300,PACNT       ;PACNT = 0300
* Causes the SCC3 TXD3 and RXD3 pins to be enabled. TCLK3 and RCLK3
* pins are left as parallel l/O pins.
          MOVE.W   #50,SIMODE         ;SIMODE = 0000 (its reset value)
* SCC3 is set up for NMSI (i.e. modem) operation. No multiplexed
* modes are used on the other SCCs.
          MOVE.W   #$00d8,SCON3       ;SCON3=00d8 for ~9600 baud at 16.67 MHz
* Baud Rate generator is used for transmit and receive. Rate is 9556bps.
          MOVE.W   #$171,SCM3         ;SCM3 = 0171
* No parity. Normal UART operation. 8-bit characters. 2 Stop bits.
* The CD and CTS lines not used to enable reception and transmission,
* but do cause a status change in the SCC3 Event register.
          MOVE.L  #$50000000,S700640  ;Set up Tx BD 0 Status and Count
          MOVE.L  #$30000,$700644     ;Set up Tx BD 0 Buffer Address
          MOVE.L  #$70000000,$700648  ;Set up Tx BD 1 Status and Count
          MOVE.L  #$30001,$70064c     ;Set up Tx BD 1 Buffer Address
* Set up 2 Tx BDs
          MOVE.L  #$d0000000,$700600  ;Set up Rx BD 0 Status and Count
          MOVE.L  #$30002,$700604     ;Set up Rx BD 0 Buffer Address
          MOVE.L  #$f0000000,$700608  ;Set up Rx BD 1 Status and Count
          MOVE.L  #$30003,$70060c     ;Set up Rx BD 1 Buffer Address
* Set up 2 Rx BDs
          MOVE.W  #$0,$700680         ;clear RFCR/TFCR (Function code setup)
* Must be initialized to a value other than 7, or won't work with chip selects
          MOVE.W  #$1,$700682         ;MRBLR = 0001 (one-byte receive
                                       buffers)
* This combined with the “l” bit set in the Rx BD, gives interrupts on each
* character received.

          MOVE.W #$4,$70069c          ;MAX_IDL don't care since MRBLR =1.
* Normally set to a small value, it closes a receive buffer if a certain number
```

SCCs on the MC68302 can do this very efficiently because of their sophisticated DMA capability, and very little MC68000 core intervention is required.

Third, some applications require the switching of data without interfering with the protocol encoding itself. For instance, in a multiplexer, data from a high-speed time-multiplexed serial stream is multiplexed into multiple low-speed data streams. In this case, the idea is to switch the data path but not alter the protocol encoded on that data path.

Finally, some applications require a special protocol that does not fall under the category of HDLC, UART, etc. In some instances, transparent mode can be used; however, care should be taken to understand the capabilities of transparent mode before trying this. The most important issue is how this new protocol recognizes its frames on the receiving end. Transparent mode on the MC68302 was designed to work well receiving continuous streams of data (no gaps in the data exist over time). This is different from receiving transparent frames; although there is some support for this, it is limited on the MC68302.

D.8.3 Physical Interface to Accompany Transparent Mode

Before discussing the details of transparent mode timing, we need to chose the physical interface to go with the transparent mode. The timings associated with transparent mode differ based on the physical interface chosen.

The MC68302 supports the following four physical interfaces:

- Nonmultiplexed Serial Interface—NMSI
- Pulse Code Modulation Highway—PCM
- Interchip Digital Link—IDL
- General Circuit Interface—GCI

You will probably chose either an NMSI or PCM highway interface, unless you are designing an ISDN-based system. If you are designing an ISDN-based system, you will probably choose either an IDL or GCI interface. The following paragraphs discuss all the interfaces, but special attention is given to the NMSI.

NOTE

The following discussion assumes some knowledge of the interfaces. For more applications information on these interfaces, refer to 4.4 Serial Channels Physical Interface**.**

The most commonly used physical interface on the MC68302 is the nonmultiplexed serial interface (NMSI). The NMSI consists of seven basic modem (RS-232) signals: $\overline{\text{TXD}}$, $\overline{\text{TCLK}}$, $\overline{\text{RXD}}$, $\overline{\text{RCLK}}$, $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{CD}}$. Each of the three SCCs can have its own set of these signals, as shown in Figure D-21.

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

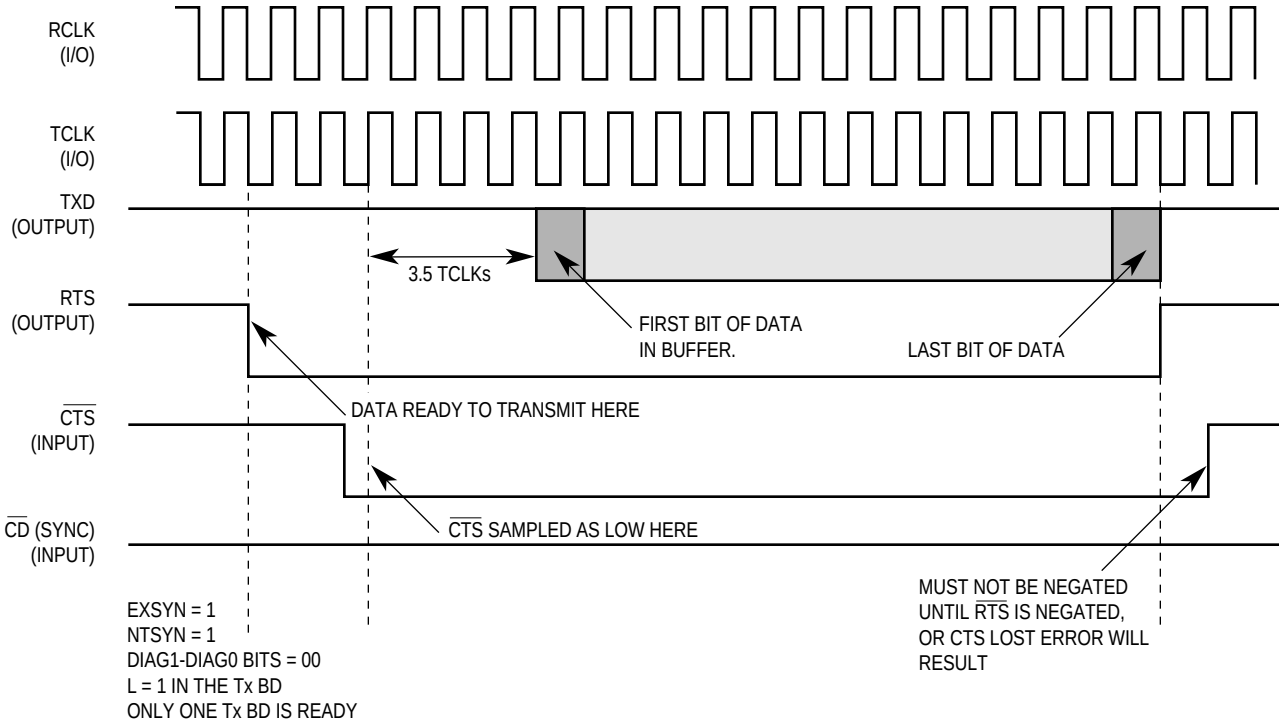


Figure D-25. Using $\overline{CTS}$ In the NMSI Transmit Case

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

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	FC2	FC1	FC0	0	0	0	0	0	FC2	FC1	FC0	0	0	0	0

E.1.1.3.2 MRBLR—Maximum Rx Buffer Length. This16-bit parameter defines the maximum receiver buffer length for each of the eight receive buffer descriptors.

E.1.1.3.3 CRC Mask_L and CRC Mask_H. This 32-bit parameter contains the constant values used for the 16-bit and 32-bit CRC calculation. For a 16-bit CRC, CRC_MASK_L should be set to $F0B8 and CRC_MASK_H is not used. For a 32-bit CRC, the user should set CRC_MASK_L = $DEBB and CRC_MASK_H = $20E3.

E.1.1.3.4 DISFC—Discard Frame Counter. This 16-bit parameter is incremented when a frame is discarded due to lack of receive buffers.

E.1.1.3.5 CRCEC—CRC Error Counter. This 16-bit parameter is incremented when a CRC error is detected in an incoming frame.

E.1.1.3.6 ABTSC—Abort Sequence Counter. This 16-bit parameter is incremented when an abort sequence is detected in an incoming frame,

E.1.1.3.7 NMARC—Nonmatching Address Received Counter. This 16-bit parameter is incremented when an error-free frame that does not match the user-defined addresses is detected.

E.1.1.3.8 RETRC—Frame Retransmission Counter. This 16-bit parameter is incremented when a frame is retransmitted due to a collision.

E.1.1.3.9 MFLR—Maximum Frame Length Register. This16-bit parameter defines the maximum length of an incoming receive frame.

E.1.1.3.10 HMASK—HDLC Frame Address Mask. This 16-bit parameter is the user-defined frame address mask register. A one should be written to each bit for which the address comparison is to occur. Bits 15-8 contain the least significant address byte, and bits 7-0 contain the most significant address byte.

E.1.1.3.11 HADDR1, HADDR2, HADDR3, and HADDR4—HDLC Frame Address. These four 16-bit parameters are the user-defined frame address registers. Bits 15-8 contain the least significant address byte, and bits 7-0 contain the most significant address byte.

E.1.1.4 RECEIVE BUFFER DESCRIPTORS. Each SCC has eight receive buffer descriptors. Each buffer descriptor consists of four words as shown below. Reserved bits in registers should be written as zeros.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	E	X	W	I	L	F	—	—	—	—	LG	NO	AB	CR	OV	CD
OFFSET +2	DATA LENGTH															
OFFSET +4	RX BUFFER POINTER															
OFFSET +6																

E.2.1.1 COMMUNICATIONS PROCESSOR (CP) REGISTERS. The CP has one set of three registers that configure the operation of the serial interface for all three SCCs. These registers are discussed in the next three subsections.

E.2.1.1.1 Command Register (CR). The command register is an 8-bit register located at offset $860 (on D15-D8 of a 16-bit data bus). This register is used to issue commands to the CP. The user should set the FLG bit when a command is written to the command register. The CP clears the FLG bit during command processing to indicate that it is ready for the next command.

7	6	5	4	3	2	1	0
RST	GCI	OPCODE		—	CH. NUM.		FLG

RST—Software Reset Command (set by the user and cleared by the CP)

0 = No software reset command issued or cleared by CP during software reset sequence.
1 = Software reset command (FLG bit should also be set if it is not already set).

GCI—GCI Commands

0 = Normal operation.
1 = The OPCODE bits are used for GCI commands (user should set CH. NUM. to 10 and FLG to 1).

OPCODE—Command Opcode

00 = STOP TRANSMIT Command.
01 = RESTART TRANSMIT Command.
10 = ENTER HUNT MODE Command.
11 = Reset receiver BCS generator (used only in BISYNC mode).

BIT 3—Reserved (should be set to zero by the user when the command register is written)

CH. NUM.—Channel Number

00 = SCC1.
01 = SCC2.
10 = SCC3.
11 = Reserved.

FLG—Command Semaphore Flag (set by the user and cleared by the CP upon command completion)

0 = CP is ready to receive a new command (should be checked before issuing the next command to the CP).
1 = Command register contains a command to be executed or one that is currently being executed.

E.2.1.2.3 SCC Data Synchronization Register (DSR). This 16-bit register is located at offset $886 (SCC1) $896 (SCC2) and $8A6 (SCC3). Bits 14-12 of the DSR are used to program the length of the last stop bit transmitted. These bits are decoded as follows:

DSR(14-12)—Fractional Stop Bits

111 16/16 (default value after reset).
110 15/16.
101 14/16.
100 13/16.
011 12/16.
010 11/16.
001 10/16.
000 9/16.

E.2.1.2.4 UART Event Register (SCCE). This 8-bit register is located at offset $888 (SCC1) $898 (SCC2) and $8A8 (SCC3) on D15-D8 of a 16-bit data bus. The SCCE is used to report events recognized by the UART channel. Bits must be cleared by the user to avoid missing interrupt events. Bits are cleared by writing ones to the corresponding bit positions.

7	6	5	4	3	2	1	0
CTS	CD	IDL	BRK	CCR	BSY	TX	RX

CTS—Clear-To-Send Status Changed

0 = No interrupt.
1 = A change in the status of $\overline{\text{CTS}}$ was detected.

CD—Carrier Detect Status Changed

0 = No interrupt.
1 = A change in the status of $\overline{\text{CD}}$ was detected.

IDL—IDLE Sequence Status Changed

0 = No interrupt.
1 = A change in the status of the receive data serial line was detected.

BRK—Break Character Received

0 = No interrupt.
1 = Break character received.

CCR—Control Character Received

0 = No interrupt
1 = Control character received (with reject (R) character = 1) and stored in the receive control character register (RCCR).

BSY—Busy Condition

0 = No interrupt.
1 = A character was received and discarded due to lack of buffers.

T

W

X