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	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/kmc68302ag16c

PREFACE

The complete documentation package for the MC68302 consists of the M68000PM/AD, *MC68000 Family Programmer's Reference Manual,* MC68302UM/AD, *MC68302 Integrated Multiprotocol Processor User's Manual,* and the MC68302/D, *MC68302 Integrated Multiprotocol Processor Product Brief*.

The *MC68302 Integrated Multiprotocol Processor User's Manual* describes the programming, capabilities, registers, and operation of the MC68302; the *MC68000 Family Programmer's Reference Manual* provides instruction details for the MC68302; and the *MC68302 Low Power Integrated Multiprotocol Processor Product Brief* provides a brief description of the MC68302 capabilities.

This user's manual is organized as follows:

Section 1	General Description
Section 2	MC68000/MC68008 Core
Section 3	System Integration Block (SIB)
Section 4	Communications Processor (CP)
Section 5	Signal Description
Section 6	Electrical Characteristics
Section 7	Mechanical Data And Ordering Information
Appendix B	Development Tools and Support
Appendix C	RISC Microcode from RAM
Appendix D	MC68302 Applications
Appendix E	SCC Programming Reference
Appendix F	Design Checklist

ELECTRONIC SUPPORT:

The Technical Support BBS, known as AESOP (Application Engineering Support Through On-Line Productivity), can be reach by modem or the internet. AESOP provides commonly asked application questons, latest device errata, device specs, software code, and many other useful support functions.

Modem: Call 1-800-843-3451 (outside US or Canada 512-891-3650) on a modem that runs at 14,400 bps or slower. Set your software to N/8/1/F emulating a vt100.

Internet: This access is provided by telneting to pirs.aus.sps.mot.com [129.38.233.1] or through the World Wide Web at http://pirs.aus.sps.mot.com.

— Sales Offices —

For questions or comments pertaining to technical information, questions, and applications, please contact one of the following sales offices nearest you.

TABLE OF CONTENTS

Paragraph Number	Title	Page Number
	Section 1 General Description	
1.1	Block Diagram	1-1
1.2	Features	1-3
1.3	MC68302 System Architecture	1-4
1.4	NMSI Communications-Oriented Environment	1-5
1.5	Basic Rate ISDN or Digital Voice/Data Terminal	1-6
	Section 2 MC68000/MC68008 Core	
2.1	Programming Model	2-1
2.2	Instruction Set Summary	2-3
2.3	Address Spaces	2-6
2.4	Exception Processing	2-8
2.4.1	Exception Vectors	2-8
2.4.2	Exception Stacking Order	2-9
2.5	Interrupt Processing	2-11
2.6	M68000 Signal Differences	2-11
2.7	MC68302 IMP Configuration Control	2-12
2.8	MC68302 Memory Map	2-14
2.9	Event Registers	2-19
	Section 3 System Integration Block (SIB)	
3.1	DMA Control	3-2
3.1.1	Key Features	3-2
3.1.2	IDMA Registers (Independent DMA Controller)	3-3
3.1.2.1	Channel Mode Register (CMR)	3-4
3.1.2.2	Source Address Pointer Register (SAPR)	3-6
3.1.2.3	Destination Address Pointer Register (DAPR)	3-6
3.1.2.4	Function Code Register (FCR)	3-7
3.1.2.5	Byte Count Register (BCR)	3-7
3.1.2.6	Channel Status Register (CSR)	3-7
3.1.3	Interface Signals	3-8
3.1.3.1	$\overline{\text{DREQ}}$ and $\overline{\text{DACK}}$	3-8
3.1.3.2	$\overline{\text{DONE}}$	3-8
3.1.4	IDMA Operational Description	3-9
3.1.4.1	Channel Initialization	3-9
3.1.4.2	Data Transfer	3-9

Paragraph Number	Title	Page Number
4.4.5	Serial Interface Registers	4-19
4.4.5.1	Serial Interface Mode Register (SIMODE)	4-19
4.4.5.2	Serial Interface Mask Register (SIMASK)	4-22
4.5	Serial Communication Controllers (SCCs)	4-22
4.5.1	SCC Features	4-24
4.5.2	SCC Configuration Register (SCON)	4-24
4.5.2.1	Asynchronous Baud Rate Generator Examples	4-26
4.5.2.2	Synchronous Baud Rate Generator Examples	4-27
4.5.3	SCC Mode Register (SCM)	4-27
4.5.4	SCC Data Synchronization Register (DSR)	4-31
4.5.5	Buffer Descriptors Table	4-32
4.5.6	SCC Parameter RAM Memory Map	4-34
4.5.6.1	Data Buffer Function Code Register (TFCR, RFCR)	4-35
4.5.6.2	Maximum Receive Buffer Length Register (MRBLR)	4-36
4.5.6.3	Receiver Buffer Descriptor Number (RBD#)	4-36
4.5.6.4	Transmit Buffer Descriptor Number (TBD#)	4-36
4.5.6.5	Other General Parameters	4-37
4.5.7	SCC Initialization	4-37
4.5.8	Interrupt Mechanism	4-38
4.5.8.1	SCC Event Register (SCCE)	4-38
4.5.8.2	SCC Mask Register (SCCM)	4-39
4.5.8.3	SCC Status Register (SCCs)	4-39
4.5.8.4	Bus Error on SDMA Access	4-40
4.5.9	SCC Transparent Mode	4-41
4.5.10	Disabling the SCCs	4-42
4.5.11	UART Controller	4-43
4.5.11.1	Normal Asynchronous Mode	4-45
4.5.11.2	Asynchronous DDCMP MODE	4-46
4.5.11.3	UART Memory Map	4-46
4.5.11.4	UART Programming Model	4-48
4.5.11.5	UART Command Set	4-49
4.5.11.6	UART Address Recognition	4-50
4.5.11.7	UART Control Characters and Flow Control	4-51
4.5.11.8	Send Break	4-53
4.5.11.9	Send Preamble (IDLE)	4-53
4.5.11.10	Wakeup Timer	4-53
4.5.11.11	UART Error-Handling Procedure	4-54
4.5.11.12	Fractional Stop Bits	4-55
4.5.11.13	UART Mode Register	4-56
4.5.11.14	UART Receive Buffer Descriptor (Rx BD)	4-57
4.5.11.15	UART Transmit Buffer Descriptor (Tx BD)	4-61
4.5.11.16	UART Event Register	4-63
4.5.11.17	UART MASK Register	4-65
4.5.11.18	S-Records Programming Example	4-65
4.5.12	HDLC Controller	4-66

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

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

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

NOTE

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

Bits 12–2—Base Address

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

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

NOTE

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

RW—Read/Write

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

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

NOTE

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

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

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

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.

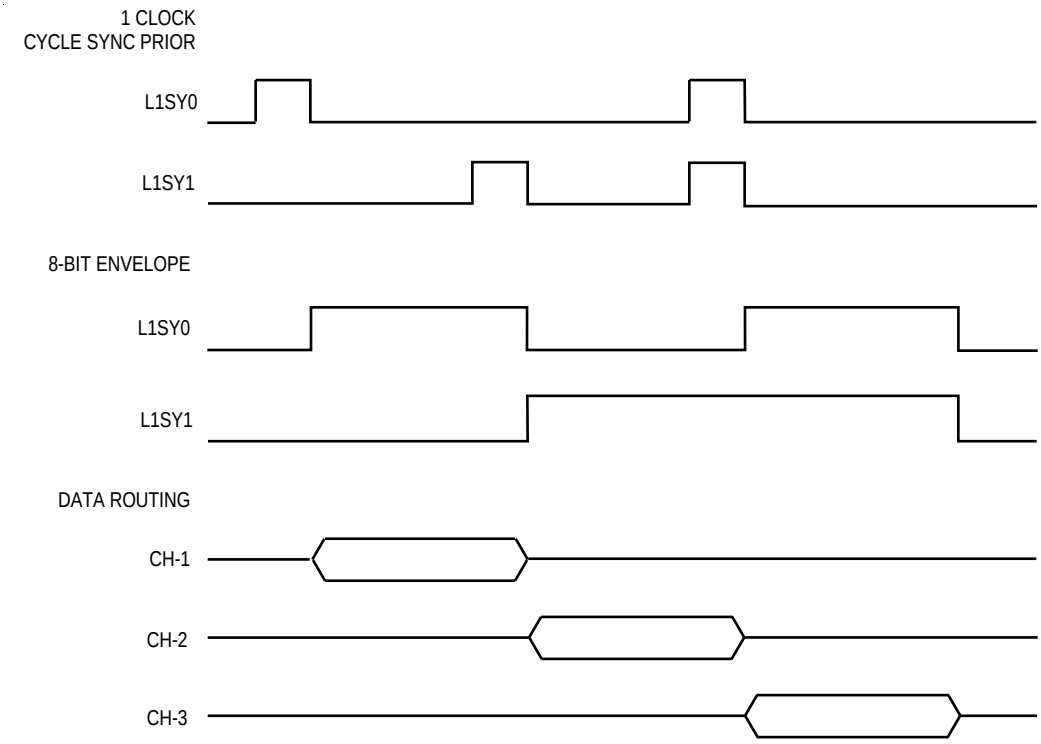

Figure 4-9. Two PCM Sync Methods

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.

MAX_IDL

The UART controller watches the receive line, regardless of whether or not actual data is being received. If the line is idle, the UART controller counts how many idle characters have been received. An idle character is defined as 9 to 13 consecutive ones. For a given application, the number of bits in the idle character is calculated as follows:

1 + data length (either 7 or 8) + (1 if address bit used) + (1 if parity bit used) + number of stop bits (either 1 or 2)

MAX_IDL is programmed with a value from 1 (MAX_IDL = 1) to 65536 (MAX_IDL = 0).

Once a character of data is received on the line, the UART controller begins counting any idle characters received. If a MAX_IDL number of idle characters is received before the next data character is received, an idle timeout occurs, and the buffer is closed. This, in turn, can produce an interrupt request to the M68000 core to receive the data from the buffer. MAX_IDL then provides a convenient way to demarcate frames in the UART mode (see also 4.5.11.11 UART Error-Handling Procedure.

NOTE

Program MAX_IDL to $0001 for the minimum timeout value; program MAX_IDL to $0000 for the maximum timeout value.

IDLC

This value is used by the RISC to store the current idle counter value in the MAX_IDL timeout process. IDLC is a down counter. It does not need to be initialized or accessed by the user.

BRKCR

The UART controller will send a break character sequence whenever a STOP TRANSMIT command is given. The number of break characters sent by the UART controller is determined by the value in BRKCR. The length of one break character is 9 to 13 zeros depending on the configuration. The same equation applies for BRKCR as that used for MAX_IDL. See 4.5.11.8 Send Break for more details. BRKCR is programmed with a value from 0 (BRKCR = 0) to 65535 (BRKCR = 65535).

PAREC, FRMEC, NOSEC, BRKEC

These counters are initialized by the user. When the associated condition occurs, they will be incremented by the RISC controller. See 4.5.11.11 UART Error-Handling Procedure for more details.

UADDR1, UADDR2

In the multidrop mode, the UART controller can provide automatic address recognition of two addresses. In this case, the lower order byte of UADDR1 and UADDR2 are programmed by the user with the two desired addresses. See 4.5.11.6 UART Address Recognition for more details.

4.5.11.5 UART Command Set

These commands are issued to the command register described in 4.3 Command Set.

STOP TRANSMIT Command

After a hardware or software reset and the enabling of the channel by writing the SCC mode register, the channel is in the transmit enable mode and starts polling the first BD in the table approximately every eight transmit clocks.

The channel STOP TRANSMIT command disables the transmission of characters on the transmit channel. If this command is received by the UART controller during message transmission, transmission of that message is aborted. The UART completes transmission of any data already transferred to the UART FIFO (up to three characters) and then stops transmitting data. The TBD# is not advanced.

The UART transmitter will transmit a programmable number of break sequences and then start to transmit idles. The number of break sequences (which may be zero) should be written to the break count register (BRKCR) before this command is given to the UART controller.

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

RESTART TRANSMIT Command

The channel RESTART TRANSMIT command re-enables the transmission of characters on the transmit channel. This command is expected by the UART in three situations: after issuing a STOP TRANSMIT command, after issuing a STOP TRANSMIT and then disabling the channel using the SCC mode register, or after transmitter errors (CTS lost). The UART controller will resume transmission from the current transmitter BD number (TBD#) in the channel's Tx BD table.

If the transmitter 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 by its 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 UART controller to abort reception of the current message, generate an RX interrupt (if enabled) as the buffer is closed, and enter the hunt mode. The UART controller will resume reception using the next BD once an address character or a single idle character is received. In multidrop hunt mode, the UART controller continually scans the input data stream for the address character. While not in multidrop mode, the UART controller will wait for a single IDLE character. In the UART mode, none of the data received in the FIFO is lost when ENTER HUNT MODE command is issued; however, this command does reset the receive FIFO in other protocols, e.g., HDLC.

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. Reception will then begin with the next BD.

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 lie beyond this entry.

NOTE

In tables with eight receive control characters, E is always zero.

R—Reject Character

0 = The character is not rejected but is written into the receive buffer. The buffer is then closed, and a new receive buffer is used if there is more data in the message. A maskable interrupt is generated in the RX bit of the UART event register.

1 = If this character is recognized, it will not be written to the receive buffer. Instead, it is written to the RCCR, and a maskable interrupt is generated in the CCR bit in the UART event register. The current buffer is not closed when a control character is received with R set.

Transmission of out-of-sequence characters is also supported and is normally used for the transmission of flow control characters such as XON or XOFF. This is performed using the last (eighth) entry in the UART control characters table. The UART will poll this character whenever the transmitter is enabled for UART operation: during freeze, during buffer transmission, and when no buffer is ready for transmission. The character is transmitted at a higher priority than the other characters in the transmit buffer (if any), but does not pre-empt characters already in the transmit FIFO.

CHARACTER8—Control Character Value

The eighth entry in the UART control characters table is defined as follows:

E—Empty

Must be one to use this entry as a flow control transmission character. To use this entry instead as a receive control characters entry, this E bit (and all other E bits in the table) should be zero.

R—Reject

Must be zero to use this entry as a flow control transmission character. For a receive control characters entry, it maintains its functionality as previously defined.

REA—Ready

This bit is set by the M68000 core when the character is ready for transmission and will remain one while the character is being transmitted. The CP clears this bit after transmission.

I—Interrupt

If set, the M68000 core will be interrupted when this character has been transmitted. (The TX bit will be set in the UART event register.)

CD—Carrier Detect Lost

The carrier detect signal was negated during message reception.

Data Length

Data length contains the number of octets written by the CP into this BD's data buffer. It is written by the CP once as the BD is closed.

NOTE

The actual amount of memory allocated for this buffer should be greater than or equal to the contents of maximum receive buffer length register (MRBLR).

Rx Buffer Pointer

The receive buffer pointer, which always points to the first location of the associated data buffer, may be even or odd. The buffer may reside in either internal or external memory.

NOTE

For correct operation of the function codes, the upper 8 bits of the pointer must be initialized to zero.

4.5.11.15 UART Transmit Buffer Descriptor (Tx BD)

Data is presented to the CP for transmission on an SCC channel by arranging it in buffers referenced by the channel's Tx BD table. The CP confirms transmission (or indicates error conditions) through the BDs to inform the M68000 core that the buffers have been serviced. The Tx BD shown in Figure 4-22.

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

Figure 4-22. UART Transmit Buffer Descriptor

The first word of the Tx BD contains status and control bits. The following bits are prepared by the user before transmission and set by the CP after the buffer has been transmitted.

R—Ready

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

1 = The data buffer, which has been prepared for transmission by the user, has not been transmitted or is currently transmitting. No fields of this BD may be written by the user once this bit is set.

The HDLC controller uses the same data structure as the UART, BISYNC, and DDCMP controllers. This data structure supports multibuffer operation and address comparisons.

The receive errors (overrun, nonoctet aligned frame, $\overline{CD}$ lost, aborted frame, and CRC error) are reported through the receive BD. The transmit errors (underrun and $\overline{CTS}$ lost) are reported through the transmit BD. An indication about the status of the lines (idle, $\overline{CD}$, and $\overline{CTS}$) is reported through the SCC status register (SCCS), and a maskable interrupt is generated upon a status change in any one of those lines.

4.5.12.5 HDLC Command Set

The following commands are issued to the command register.

STOP TRANSMIT Command

After a hardware or software reset and the enabling of the channel in the SCC mode register, the channel is in the transmit enable mode and starts polling the first BD in the table approximately every eight or sixteen transmit clocks as determined by the FLG bit in the HDLC mode register.

The channel STOP TRANSMIT command disables the transmission of frames on the transmit channel. If this command is received by the HDLC controller during frame transmission, transmission of that frame is aborted after the contents of the FIFO are transmitted (up to four words). The TBD# is not advanced. No new BD is accessed, and no new frames are transmitted for this channel. The transmitter will transmit an abort sequence (if the command was given during frame transmission) and then begin to transmit flags or idles as indicated by the HDLC mode register. The abort sequence on transmit is a zero followed by seven ones (01111111).

This command is useful for performing frame retransmission. The M68000 core may issue the STOP TRANSMIT command, reorganize the transmit BD table, and issue the RESTART TRANSMIT command. The STOP TRANSMIT command may also be used in the X.25 protocol to send a reject frame or a link reset command.

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 HDLC controller after a STOP TRANSMIT command, after a STOP TRANSMIT command and disabling the channel in its SCC mode register, or after transmitter error (underrun or $\overline{CTS}$ lost when no automatic frame retransmission is performed). The HDLC controller will resume transmission from the current transmitter BD (TBD#) in the channel's transmit BD table.

If the transmitter 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.

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

Figure 4-40. V.110 Receive Buffer Descriptor

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

E—Empty

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

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

X—External Buffer

0 = The buffer associated with this BD is in internal dual-port RAM.

1 = The buffer associated with this BD is in external memory.

W—Wrap (Final BD in Table)

0 = This is not the last BD in the Rx BD table.

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

NOTE

The user is required to set the wrap bit in one of the first eight BDs; otherwise, errant behavior may occur.

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

SE—Synchronization Error

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

OV—Overrun

A receiver overrun occurred during message reception.

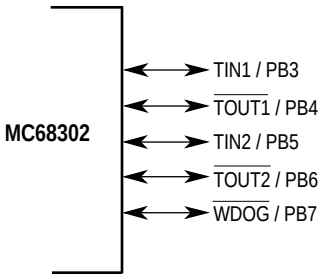

Figure 5-15. Timer Pins

Each of these five pins can be used either as a dedicated timer function or as a general-purpose port B I/O port pin. Note that the timers do not require the use of external pins. The input buffers have Schmitt triggers.

TIN1/PB3—Timer 1 Input

This input is used as a timer clock source for timer 1 or as a trigger for the timer 1 capture register. TIN1 may also be used as the external clock source for any or all three SCC baud rate generators.

$\overline{\text{TOUT1}}$/PB4—Timer 1 Output

This output is used as an active-low pulse timeout or an event overflow output (toggle) from timer 1.

TIN2/PB5—Timer 2 Input

This input can be used as a timer clock source for timer 2 or as a trigger for the timer 2 capture register.

$\overline{\text{TOUT2}}$/PB6—Timer 2 Output

This output is used as an active-low pulse timeout or as an event overflow output (toggle) from timer 2.

$\overline{\text{WDOG}}$/PB7—Watchdog Output

This active-low, open-drain output indicates expiration of the watchdog timer. $\overline{\text{WDOG}}$ is asserted for a period of 16 clock (CLKO) cycles and may be externally connected to the $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ pins to reset the MC68302. $\overline{\text{WDOG}}$ is never asserted by the on-chip hardware watchdog (see the $\overline{\text{BERR}}$ signal description). The $\overline{\text{WDOG}}$ pin function is enabled after a total system reset. It may be reassigned as the PB7 I/O pin in the PBCNT register.

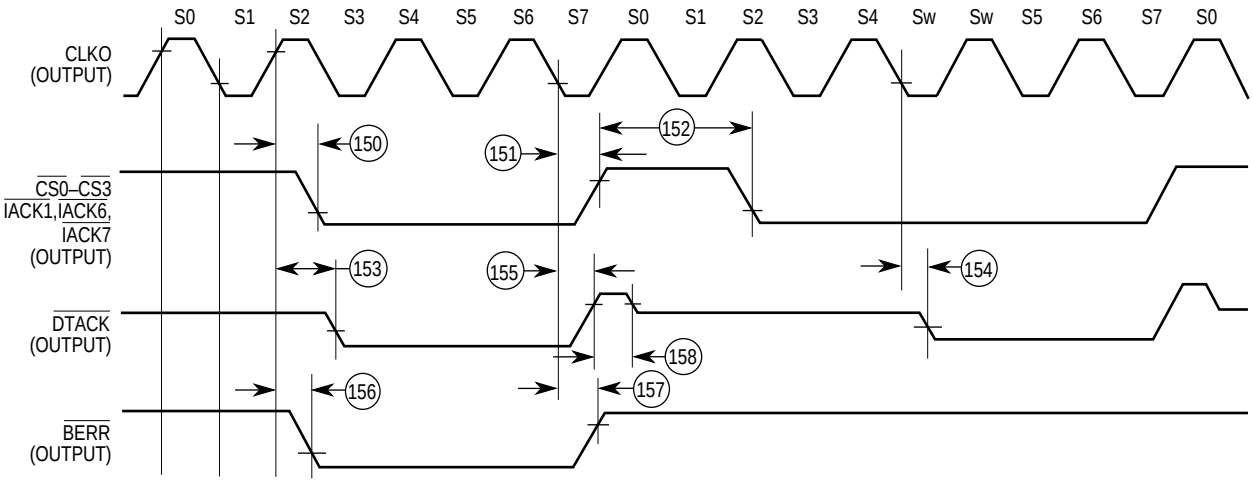

Figure 6-14. Internal Master Chip-Select Timing Diagram

6.14 AC ELECTRICAL SPECIFICATIONS—CHIP-SELECT TIMING EXTERNAL MASTER (see Figure 6-15)

			16.67 MHz		20 MHz		25 MHz		
Num.	**Characteristic**	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
154	Clock Low to $\overline{DTACK}$ Low (1-6 Wait States)	t_{CLDTKL}	—	30	—	25	—	20	ns
160	$\overline{AS}$ Low to $\overline{CS}$ Low	t_{ASLCSL}	—	30	—	25	—	20	ns
161	$\overline{AS}$ High to $\overline{CS}$ High	t_{ASHCSH}	—	30	—	25	—	20	ns
162	Address Valid to $\overline{AS}$ Low	t_{AVASL}	15	—	12	—	10	—	ns
163	R/$\overline{W}$ Valid to $\overline{AS}$ Low (see Note 1)	t_{RWVASL}	15	—	12	—	10	—	ns
164	$\overline{AS}$ Negated to Address Hold Time	t_{ASHAI}	0	—	0	—	0	—	ns
165	$\overline{AS}$ Low to $\overline{DTACK}$ Low (0 Wait State)	$t_{ASLDTKL}$	—	45	—	40	—	30	ns
167	$\overline{AS}$ High to $\overline{DTACK}$ High	$t_{ASHDTKH}$	—	30	—	25	—	20	ns
168	$\overline{AS}$ Low to $\overline{BERR}$ Low (see Note 2)	$t_{ASLBERL}$	—	30	—	25	—	20	ns
169	$\overline{AS}$ High to $\overline{BERR}$ High Impedance (see Notes 2 and 3)	$t_{ASHBERH}$	—	30	—	25	—	20	ns

NOTES:
1. The minimum value must be met to guarantee write protection operation.
2. This specification is valid when the ADCE or WPVE bits in the SCR are set.
3. Also applies after a timeout of the hardware watchdog.

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.

```
* Set up PACNT, PBCNT, etc., ignore for this example, only SCC1 is used
* Select Serial Interface Mode: normal operation, NMSI mode
          MOVE.W    #0,SIMODE          ;Same as default after reset

*** SCC1 Initialization ***
*  Interrupt Vector: SCC1 interrupt handler is at INT_VEC = $31000
*  v7-v5 =5, v4-v0 = $ad -> vector = $ad -> Exception vector = ($ad<<2) = $2b4
          MOVE.L    #INT_VEC, $02B4

* Determine Configuration
* Use Baud Rate Generator for transmit and receive, Rate is 130 kbps.
          MOVE.W    #$07E,SCON1

* Select SCC Mode
* HDLC, Loopback mode, CRC16, RTS negate between frames, NRZ mode.
          MOVE.W    #$10,SCM1

* Set up Parameter RAM
          MOVE.W    #0,FCR_1           ; Clear RFCR and TFCR
          MOVE.W    #$08,MRBLR_1       ; Max Buffer Length = 8
          MOVE.W    #$F0B8,CMSKL_1     ; 16 bit CRC
          MOVE.W    #$070,MFLR_1       ; Max Frame Length = $70 bytes
          MOVE.W    #0,HMASK_1         ; Do not check address
          MOVE.W    #0,DISFC_1         ; Clear the counter
          MOVE.W    #0,CRCEC_1         ; Clear the counter
          MOVE.W    #0,ABTSC 1         ; Clear the counter
          MOVE.W    #0,NMARC_1         ; Clear the counter
          MOVE.W    #0,RETRC_1         ; Clear the counter

* Clear Event Register
          MOVE.B    #$FF,SCCE1

* Determine Maskable Interrupt Events by setting SCCM
* Allow the following interrupt: TXE, RFX, TXB, and RFB
          MOVE.B    #$1B,SCCM1

* Clear M68000 data registers
          CLR.L     D0
          CLR.L     D1
          CLR.L     D2
          CLR.L     D3
          CLR.L     D4
          CLR.L     D5

***Prepare Buffer Descriptors ***
*SCC1 Rx Buffer Descriptors Initialization values before execution:
*00700400 D000 0000 0003 0000 D000 0000 0003 0010
*00700410 D000 0000 0003 0020 D000 0000 0003 0030
*00700420 D000 0000 0003 0040 D000 0000 0003 0050
*00700430 D000 0000 0003 0060 F000 0000 0003 0070
          LEA.L     RXBD_01,A0         ;A0 points to the first RXBD of SCC1
          LEA.L     RXBF_01,A1         ;A1 points to the first buffer
          MOVE.W    #$D000,D1          ;D1 is used for setting the status of BD
*                                      ;Empty = 1, External = 1, Int=1
          MOVE.W    #$F000,D2          ;D2 is for the last BD, Wrap = 1
```

D.6.9 Parallel I/O Port A Configuration

To implement a DCE interface for the non-ISDN terminal, PA2 and PA5 should be configured as general-purpose outputs (drive $\overline{\text{CTS}}$ and $\overline{\text{CD}}$).

PA7 and PA8 should be configured as general-purpose outputs (drive SCP_EN and RESET).

PA4 and PA6 may be configured either as general-purpose inputs or as dedicated modem pins.

If a general-purpose input is chosen, the state of the terminal $\overline{\text{DTR}}$ and $\overline{\text{RTS}}$ lines will be polled by application software by reading the parallel I/O data register.

If PA4 and PA6 are to be used as general-purpose inputs, the following registers should be set:

Register	Value	Comments
PACNT	$000B	Set PA0 PA1, and PA3 to the dedicated mode. (RXD2, TXD2, and TCLK2 connected to SCC2.)
PADDR	$01A4	Set PA2, PA5, PA7, and PA8 as output pins. ($\overline{\text{RCLK2}}$ and $\overline{\text{RTS2}}$ are outputs.)
PADAT		Will be set by the application software to drive PA2, PA5, PA7, and PA8 to the proper state. Will be read to check $\overline{\text{DTR}}$ and $\overline{\text{RTS}}$.

If the dedicated mode is chosen, the software can use the delta mechanism of the SCC to generate an interrupt upon any change in the state of $\overline{\text{RTS}}$ or $\overline{\text{DTR}}$. In this case, the SCC can be configured to use its $\overline{\text{CTS}}$ (connected to the terminal $\overline{\text{DTR}}$ line) and $\overline{\text{CD}}$ (connected to the terminal $\overline{\text{RTS}}$ line) under automatic control or under software control by setting the DIAG bits in the mode register.

If $\overline{\text{CTS2}}$ (PA4) and $\overline{\text{CD2}}$ (PA6) are to be used as dedicated pins, the following registers should be set:

Register	Value	Comments
PACNT	$005B	Set PA0, PA1, PA3, PA4, and PA5 to the dedicated mode. (RXD2, TXD2, TCLK2, CTS2 and CD2 connected to SCC2.)
PADDR	$01A4	Set PA2, PA5, PA7, and PA8 as output pins. ($\overline{\text{RCLK2}}$ and $\overline{\text{RTS2}}$ are outputs.)
PADAT		Will be set by the application software to drive PA2, PA5, PA7, and PA8 to the proper state.

D.6.10 SCP Bus

The SCP (see Figure D-15) is an industry standard bus used for controlling and programming external devices. The SCP is a four-wire bus consisting of transmit path, receive path, associated clock, and enable signal (see Figure D-16). The clock determines the rate of the exchange of data in both the transmit and receive directions, and the enable signal governs when this exchange occurs.

using transparent mode with the EXSYN bit set in SCM2.
DSR2 = $7E7E

6. Setting SCCE2 to $FF clears out any current status in the event register. SCCE2 = $FF
7. You must indicate in the SCCM2 register which (if any) transparent events you wish to cause interrupts. Bit 5 should be set to zero. Bit 3 is only valid in BISYNC mode and has no meaning in transparent mode. All other bits are valid. For this example, we will disable interrupts, but all events will still be set in SCCE2.
SCCM2 = $00
8. Setting IMR to $0400 allows interrupts from SCC2 to be enabled in the interrupt controller; however, since SCCM2 = $00, any SCC2 interrupt requests are prevented from reaching the interrupt controller, and this step has no effect.
IMR = $0400
9. Initialize the receive and transmit function codes to 000, and set the receive buffer size to 10 (hex) bytes. These are the general-purpose parameter RAM values for SCC2.
RFCR = $00
TFCR = $00
MRBLR = $0010
10. Initialize the BISYNC parameters. These parameters do not involve transparent mode; however, it is a good idea to initialize them in case BISYNC mode is ever accidentally entered by clearing the NTSYN bit in SCM2. The values for BSYNC and BDLE are arbitrary and were chosen so that the two registers have different values. The control characters table is disabled for good measure, although this too is not used in transparent mode.
PRCRC = $0000
PTCRC = $0000
PAREC = $0000
BSYNC = $0033
BDLE = $0044
CHARACTER1 = $8000
11. The Tx BD buffer starts at address $30000. It is 18 (hex) bytes long. (Notice that the MRBLR value equal to $0010 does not restrict the transmit buffer size.) The status $D800 says that the buffer is ready and in external RAM. Since the I bit is set, the TX bit in the SCCE2 register will be set upon completion, and this is the “last” buffer in this transmission. The next Tx BD is set up so that it is not ready; transmission will halt after one Tx BD.
Tx BD = $D800 $0018 $0003 $0000
Tx BD = $5800 $xxxx $xxxx $xxxx (This Tx BD is not yet ready.)
12. Two empty Rx BDs are needed to receive the transmit frame. Both are currently empty, and data is to be stored in external RAM buffers. Since the I bits are set, the RX bit in the SCCE2 register will be set when each buffer is filled with data. The third Rx BD is not ready yet. If it was ready, it would be filled with all $FFs (idles) after the first two buffers were filled.
Rx BD = $D000 $0000 $0004 $0000
Rx BD = $D000 $0000 $0004 $0010

E.1.1.2 PER SCC REGISTERS. Each of the three SCCs has a set of the following six registers. These registers configure the SCC and the protocol operation. Some parameters and register bits are protocol independent. The HDLC functions have been given for those parameters and bits that are protocol specific.

E.1.1.2.1 Serial Configuration Register (SCON). This 16-bit register is located at offset $882 (SCC1), $892 (SCC2), and $8A2 (SCC3). The SCON register is used to select the clock source and baud rate for the SCC.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WOMS	EXTC	TCS	RCS	CD10	CD9	CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0	DIV4

WOMS—Wired-OR Mode Select

0 = TXD driver operates normally.
1 = TXD driver functions as an open-drain output and may be wired together with other TXD pins.

EXTC—External Clock Source

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

TCS—Transmit Clock Source

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

RCS—Receive Clock Source

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

CD10—CD0—Clock Divider

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

DIV4—SCC Clock Prescaler Divide by 4

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

E.1.1.2.2 SCC Mode Register (SCM). This 16-bit register is located at offset $884 (SCC1), $894 (SCC2), and $8A4 (SCC3). The SCM register configures the operation of the SCC and defines HDLC specific parameters. Note that reserved bits in registers should be written as zeros.

15	14	13	12	11	10	9	8
NOF3	NOF2	NOF1	NOF0	C32	FSE	—	RTE

7	6	5	4	3	2	1	0
FLG	ENC	DIAG1	DIAG0	ENR	ENT	MODE1	M0DE0

E.2.2.5 SCC INTERRUPT HANDLING.

1. Read the SCC event register.
2. Clear any unmasked bits that will be used in this interrupt routine.
3. Handle the interrupt events as required by the system.
4. Clear the appropriate SCC bit in the in-service register (ISR) of the interrupt controller.
5. Return from the interrupt.

E.3 TRANSPARENT PROGRAMMING REFERENCE SECTION

This subsection discusses the registers and parameters required to program an SCC for transparent operation. At the end of this subsection is a generic algorithm for programming the SCC.

E.3.1 Transparent Programming Model

The programming model and memory map for the transparent protocol is shown in E-1. The offsets for each SCC are given above each table. Some parameters are common to all protocols. The transparent parameters are shown for those entries that are protocol specific. Table E-3(b) depicts the general and protocol-specific parameter RAM for each SCC. The SCC registers are shown in Table E-3(c), and the communications processor registers are shown in Table E-3(d). Note that reserved bits in registers should be written as zeros.